International Journal of Computer Engineering in Research Trends

Multidisciplinary, Open Access, Peer-Reviewed and fully refereed

Research Paper Volume-7, Issue-7 , July 2020 Regular Edition E-ISSN: 2349-7084

Inverter-based MUX: A Low Overhead Approach for Logic Encryption

Ghobad Zarrinchian[1*]

[1*] *Dept. of Computer Engineering, Islamic Azad University of Yadegar-e-Imam Khomeini (rah) Shahr-e-Rey, Tehran, Iran*

e-mail: zarrinchian@iausr.ac.ir

**Corresponding Author: zarrinchian@iausr.ac.ir,*

Available online at: http://www.ijcert.org

Received: 14/07/2020 Revised: 24/07/2020 Accepted: 02/08/2020 Published: 08/08/2020

Abstract:

IC overproduction and design theft have been a concern in recent decades for the revenue loss of digital design companies. Logic encryption is a well-known approach to address this problem by locking the functionality of digital designs. In logic encryption techniques, key-gates are added to the design whose functionality is to lock (or obfuscate) the operation of the circuit. Correct functionality is achieved by applying a correct key, which is only known to the IC designer, to these key-gates. The key-gates, however, may incur a considerable overhead to the area and performance of the design. In this paper, a new technique based on simple inverter cells is proposed, which can provide the required locking functionality with low overhead. The results on a set of ISCAS'89 benchmarks reveal that the proposed approach incurs about 2% to 19% area overhead, which is less than any other technique, as well as low power overhead.

Keywords: Overproduction, design theft, logic encryption, key-gate, obfuscation, inverter

1. Introduction

IC piracy and digital design theft have been recognized as serious concerns for the revenue loss of design houses. According to published results, IP and IC design companies are facing billions of dollars revenue loss a year in the form of design theft, reverse engineering, cloning, recycling, and so on [1]. Outsourcing digital designs to foundries and lack of designers' control over the IC manufacturing phase have raised even more concerns regarding IC overproduction and malicious modifications of the design (known as hardware Trojans) [2].

Nowadays, design companies must face numerous security concerns to ensure that their products are secure and cannot be illegally cloned or reused by other third-party groups.

Illegally copying of ICs, which can be done through IC reverse engineering or IC overbuilding, is one of the most important security concerns that has been the focus of attention in the literature. In IC reverse engineering, an intact IC is de-layered, and its internal structure is extracted by the help of imaging or other applicable techniques. The extracted map of the IC can then be used to illegally fabricate and sell the same design (probably with a new brand).

On the other hand, IC overproduction is easily feasible as the fabrication company has full access to the design files and can illegally produce excess ICs to sell them in the market, causing a huge revenue loss to the design companies.

To address illegal copying of the IC and IP designs, logic encryption techniques (also known as active metering techniques) have been introduced [1]. In these techniques, the normal functionality of design is locked by the use of a security key which is only known to the designer. The design will function correctly only if the correct unlocking key is loaded into the chip. Logic encryption makes IC copying useless as none of the illegally fabricated or cloned chips will function correctly without the unlocking key.

https://doi.org/10.22362/ijcert/2020/v7/i07/v7i0706

Logic encryption techniques embed key-gates in the design to lock it. These key-gates, which can range from XOR/XNOR cells to LUTs, however, incur considerable overhead to the design. Especially, when the security is an important factor, and a large number of key-gates should be embedded, this overhead may be prohibitive. Few works have been studied in the literature to reduce the overhead caused by logic encryption. In this paper, a new approach is proposed to address this issue. In the proposed method, typical inverter cells from standard cell libraries are embedded in the design as key-gates in a way that provides the required locking functionality. These cells are generally the smallest in terms of size compared to other cells in every standard cell library.

The proposed method does not require any new specialized standard cell library and can be implemented easily with current CAD tools and technology. It only requires some treatments with signal wires in the physical design stage.

By reducing the hardware overhead, the proposed method provides the capability to embed more key-gates in a circuit, which can result in higher security as a side advantage.

The present work has the following contributions:

- Introducing a low-overhead key-gate to lock integrated circuits.
- Analyzing the area and performance overhead of the proposed approach on a set of ISCAS'89 benchmarks.

The rest of the paper is organized as follows: Section 2 presents a literature review on available logic encryption techniques. The proposed method is introduced in Sections 3. In Section 4, simulation results and performance of the proposed technique are discussed, and finally, Section 5 concludes the paper.

2. Related Work

Preventing design theft requires mechanisms to enable the designers to gain control over their manufactured chips. In these mechanisms, which are known as logic encryption or active metering techniques, the logical specification of the design is modified such that the design is initially inactive (locked) and can be activated only by applying a correct unlocking key. The designer is the only entity who knows the unlocking key and can activate the design to perform its normal operation. As a result, every reverse engineered or overproduced IC would not work without its activation key.

Current logic encryption techniques are generally classified into two groups: Sequential locking and combinational locking techniques. In sequential locking techniques, the FSM (Finite State Machine) structure of design is modified such that the design is initially powered-up in a non-functional and locked state [3-4]. To enter the normal functional states, the user should apply a key (a sequence of inputs) to the circuit by which the circuit traverses through a set of pre-defined states and then arrives at its first functional state. From now on, the circuit operates normally. There are some important issues with sequential locking techniques: First of all, they incur a significant hardware overhead. Secondly, they require a huge effort to modify the FSM structure of the original design, which is costly and time-consuming. Lastly, the level of security they offer is relatively unknown to the community as few security analysis has been done in the literature on these techniques.

For the reasons mentioned above, the main focus on logic encryption techniques in the literature has been on combinational locking approaches [3-10]. In these methods, combinational parts of the design are modified to implement the locking functionality by embedding key-gates at different parts of the design. Currently, various methods have been proposed for this purpose, and the difference between these methods is based on the type of key-gates they use to lock the design.

In [5], simple XOR/XNOR cells are embedded as key-gates in some nodes of the designs to obfuscate their original functionality (See Fig. 1 as an example). When propagating a given input to the output, these logic cells act as either a buffer or an inverter based on the logic value of the second input. This input, thus, represents the key bit that should be applied to the key-gate.

MUXs (Multiplexers) can also be used to obfuscate the design [6]. When using MUXs, a selection is made between a given original net (true wire) and a secondary net (false wire). In this scheme, which is illustrated in Fig. 2, the selection bit of the MUX represents the key bit that determines which of the two inputs is the correct signal and should be propagated.

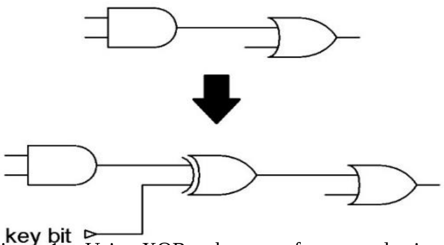


Figure 1. Using XOR as key-gate for a sample circuit

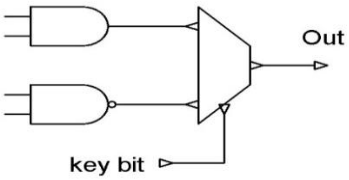


Figure 2. Using MUX with true and false wires as key-gate for a sample circuit

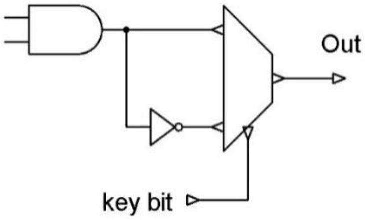
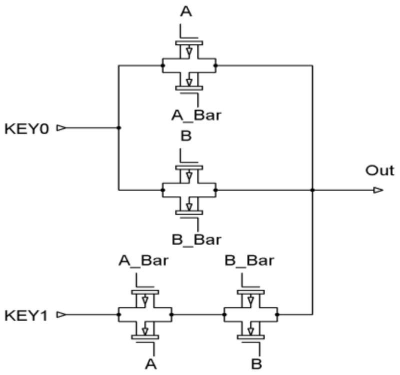


Figure 3. Using MUX and Inverter as key-gate for a sample circuit

Using MUXs to choose between a set of driving signals is the basic idea of approaches in which the focus is on scrambling circuit signals instead of directly affecting the logic values. Scrambling bus wires [7] or internal wires of design [8] are two approaches with this technique in mind in which the signals will be scrambled in case of applying a wrong unlocking key.

A technique similar to the MUX-based method is proposed in [9] where a selection is made between a given signal and its inverted version (See Fig. 3). This technique ensures that applying a wrong key would result in erroneous function. In contrast, in the former case with true and false wires, an incorrect key does not necessarily result in an erroneous behavior as the logical value of the two wires may be the same.



(a)

KEY0	KEY1	A	B	Out
0	0	X	X	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	X	X	1

(b)

Figure 4. (a) AND/NAND gate structure proposed in [11], (b) The truth tale for the AND/NAND gate

Baumgarten et al. in [10] proposed to obfuscate the design by removing some parts of the design from the netlist and implementing them in the form of reconfigurable logics. In this scheme, some parts of the netlist are implemented with the help of Lookup Tables (LUTs), and the bitstream to be loaded into the LUTs represents the activation key.

The main issue with combinational logic encryption techniques described above is the hardware overhead they incur to the circuit. This overhead may be prohibitive, especially when the high level of security is required and thus a large number of key-gates should be embedded (e.g., 128 MUXs should be embedded when a 128-bit security key is required). The large number of key-gates may incur significant area and power overhead. They can also result in performance (timing) overhead when placed in critical paths of the design.

Few works have been done to address the overhead issue. From the view point of security, some studies have tried to select the location of embedded key-gates such that it requires lower key-gate count while achieving an acceptable level of security [6]. This approach is independent of the type of key-gates and can be used with any locking structure.

From the view point of technology, the only work proposed in the literature, to the best of our knowledge, is the work presented in [11]. This study proposes to implement special key-gates with Transmission Gate (TG) technology to reduce the overhead. In this study, key-gates with dual functionality are proposed whose functionality is determined by their input key bits. An example structure is shown in Fig. 4(a). This structure implements an AND/NAND gate, and its functionality is determined by the two key bits KEY0 and KEY1. Fig. 4(b) illustrates the functionality of the mentioned AND/NAND gate. According to the figure, when KEY0=0 and KEY1=1, the gate functions as an AND gate, and when KEY0=1 and KEY1=0, the gate functions as a NAND gate. In other conditions (KEY0 and KEY1 equal to 00 or 11), a constant 0 or 1 value is produced at the output. Other hybrid gates such as OR/NOR gates can also be realized with minor modifications to the same structure.

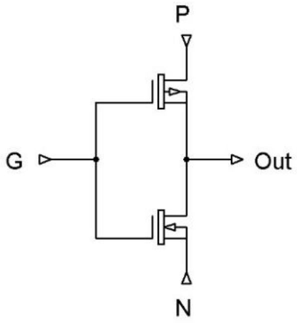
The proposed TG-based key-gates can reduce the overhead by reducing the number of required transistors per each key-gate. According to Fig. 4, the AND/NAND structure requires eight transistors plus two inverters to produce complemented signals (A_Bar and B_Bar), resulting in 12 transistors in total. Since two key bits are used in this structure, the share per each key bit is six transistors. This is while an XOR/XNOR or MUX cell requires 10 to 12 transistors on average as a key-gate with one key bit, which means about 50% reduction in overhead.

The main issue with the TG-based key-gate structure is that it cannot be realized with current CMOS-based standard cell libraries and requires a huge effort to produce a new library with TG cells.

In the next section, a new logic encryption technique is proposed, which not only further reduces the overhead but also can be fully realized with typical standard cell libraries and requires no special technology.

3. Proposed approach

To realize key-gates with low overhead, this paper proposes to use simple inverter cells (NOT gates). A CMOS inverter cell is constructed by connecting its pull-up PMOS transistor to VDD and pull-down NMOS transistor to GND. However, as discussed in [12], this structure can be used in a way that pull-up and pull-down transistors can be connected to any.



			Function
			$\bar{A}B$
			AB
			$A+B$
			$\bar{A}+B$
			$\bar{A}B+AC$
			$\bar{A}$

(a) (b)

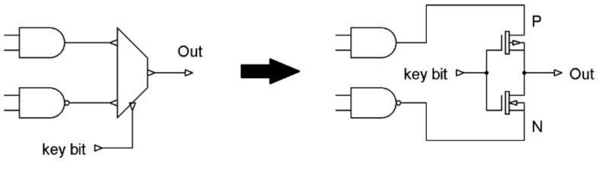
(a) A typical CMOS inverter structure, (b) Different functions that can be implemented with the inverter Logic value.

In other words, the inverter cell can be seen as a three-pin structure in which each pin can be independently connected to a logic value. In this way, different functions can be implemented using the same structure based on the logic values of the pins. Fig. 5 illustrates this concept. Different functions that can be implemented with this structure are also shown in the figure.

Based on the figure, the inverter structure can be used to implement a MUX. In this case, the input to inverter (G) acts as the selector signal of the MUX which determines which of the two pin values P or N should be propagated to the output. Since MUXs can be used as key-gates for logic encryption, the inverter cells with MUX functionality can be used for this purpose. This results in a considerable reduction in the overhead of key-gates.

The aforementioned approach, which is called hereafter as Inverter-based MUX, is a MUX cell with pass transistor logic. The main advantage of this logic style is its low number of transistors required to implement different logics, which provides much lower overhead compared to other logic styles. The main problem with this logic style, however, is its inability to propagate complete voltage levels. It is generally known that NMOS and PMOS transistors show a weak performance in propagating 1 and 0, respectively. In a multi-level logical circuit, this can result in a gradual decrease in voltage swing and finally produce incorrect voltage levels. This problem can be solved by the use of a transmission gate logic style (using a pair of NMOS and PMOS transistors to pass a voltage level). This solution is achieved at the cost of doubling the overhead of the logical circuit.

In the proposed logic encryption technique, key-gates are implemented with the inverter-based MUXs while the rest of the circuit is realized with the typical CMOS style.



Key gate with typical CMOS MUX

Key gate with inverter-based MUX

Fig. 6 shows an example in this regard.

Instead of using a CMOS-Replacing MUXs by inverters for a sample circuit based MUX, an inverter-based MUX is embedded in the design as a key-gate. The internal signals that selection should be made among are connected to the pins P and N (instead of connecting these pins to VDD and GND).

Since key-gates are not directly connected (connecting two key-gates has no advantage from the security perspective), the issue with voltage swing happens only once (only for one circuit level), which is compensated by the cell in the next level (the next cell retrieves the complete swing). As a result, this design methodology does not affect the correct functionality of the circuit.

Based on the proposed technique, the overhead required to embed one key-gate in the design is equivalent to the overhead of one inverter cell (i.e., two transistors), which has by far the least overhead among other encryption techniques.

To realize the proposed logic encryption technique, typical inverter cells are used as key-gates. This means that no special technology library is required for this purpose. The only required manipulation at the physical design stage is to connect VDD and GND pins of these inverter cells to internal signal nets, instead of connecting to VDD and GND nets. This can be done manually or using a script to do this in the CAD tool.

4. Experimental results

To evaluate the performance of the proposed technique, we used a set of ISCAS'89 benchmarks. The technology library used to synthesize the experimental circuits is Nangate 45nm technology. To evaluate the overhead incurred by the proposed technique, we considered a security key of 64-bit length. This means that 64 key-gates should be embedded in the design. Table 1 depicts the percent of area overhead in the set of benchmarks. The area overhead has been reported for the proposed technique as well as four other logic encryption techniques, namely: XOR/XNOR, MUX-TF, MUX-NC, and TG. XOR/XNOR represents the method in which XOR/XNOR gates are embedded as key-gates. MUX-TF represents the method in which MUXs with true and false.

TABLE I. AREA OVERHEAD OF DIFFERENT LOGIC ENCRYPTION TECHNIQUES

	XOR/XNOR	MUX-TF	MUX-NC	TG[11]	Proposed inverter-based MUX
s641	58.7%	68.5%	88%	35.2%	19.5%
s1423	16.5%	19.3%	24.8%	9.9%	5.5%
s1488	29.8%	34.8%	44.7%	17.8%	9.9%
s5378	7.2%	8.4%	10.8%	4.3%	2.4%
s9234	9.2%	10.7%	13.8%	5.5%	3%

TABLE II. POWER OVERHEAD OF MUX-TF AND PROPOSED INVERTER-BASED MUX TECHNIQUES

	MUX-TF	Proposed inverter-based MUX
s641	5.35%	1.95%
s1423	2.08%	3.21%
s1488	13.93%	8.86%
s5378	0.48%	1.67%
s9234	0.2%	0.01%

Wires are embedded in the design. MUX-NC represents the method in which a MUX is used to select among a signal net and its inverted version. Finally, TG represents the technique proposed in [11] in which TG-based cells are embedded as key-gates.

As expected, the proposed inverter-based MUX technique has resulted in the lowest area overhead compared to other methods. It should be noted that the area overhead incurred by different techniques depends on the transistor-level structure of their key-gates. As this structure is fixed and pre-determined for all types of key-gates, the overhead can be measured based on the security key length specified for the design and the information available for the technology library, without the need to implementing a given logic encryption approach.

Since no library for transmission gate logic style was available, the results for the TG-based approach are reported relative to the XOR/XNOR approach. In the XOR/XNOR approach, one key-gate requires ten transistors (in the target 45nm technology node), while a key-gate in the TG-based approach requires only six transistors. As a result, we assumed that the TG-based approach results in 40% lower area overhead compared to the XOR/XNOR method.

The power overhead of the proposed technique has also been evaluated, and the results are reported in Table 2. The results reported in the table are the average power consumed over 2000 test vectors applied to the set of benchmarks. Hspice was used to implement the circuits at the transistor level. PTM 45nm BSIM4 [13] has been used to model each transistor in simulated circuits. The power overhead for the MUX-TF approach is also reported in the table for comparison.

As the proposed inverter-based MUX structure has fewer transistors compared to the MUX-TF method (and also compared to other methods), lower power overhead is expected for the proposed approach. While this can be seen in major simulated benchmarks in the table, s1423 and s5378 benchmarks show different results. The proposed technique has resulted in more power overhead for these two benchmarks. To investigate the issue, a deeper analysis of the extracted power results was done. Based on our observations, more leakage power is seen in s1423 and s5378 benchmarks compared to the MUX-TF approach, which is the main cause of higher power overhead.

We believe that the higher leakage power is related to the weak performance of NMOS and PMOS transistors in passing 1 and 0 logics, respectively. To explain the issue, consider the circuit shown in Fig. 7. Based on the figure, a key-gate is used to select between the output nets of two cells G1 and G2. The key-gate is connected to at least two transistors (Q1 and Q2) at the next level of the circuit. The load effects of the two transistor gates, together with the capacitance of the connecting wire, can be modeled with the load capacitance C at the output of the key-gate.

Now, consider the following scenario: The net from G1 cell is the correct net to be selected, and the logic value on this net is 0. As G1 is connected to pin P of the key-gate, the correct key bit to apply to the key-gate is 0. Applying 0 to the key-gate would make PMOS transistor ON, discharging C from high to low (assuming C is already charged). Since PMOS is weak in passing logic 0, it cannot completely discharge C, and it goes OFF after the charge on C is less than the threshold voltage. Since the leakage current of a transistor is exponentially proportional to its sub-threshold gate voltage, the remained voltage on the capacitance load C will make a considerable amount of leakage current to flow from the transistors at the next level, resulting in higher power consumption. A similar discussion can be made when capacitive load C is to be charged using NMOS transistor.

The issue with leakage power mentioned above will not arise if G1 is connected to pin N as NMOS transistors perform well in passing logic 0. This instructs the designers to connect key-gates' inputs to the corresponding nets according to the logic levels mostly occurred on the signal nets. In other words, nets with a high probability of being 0 should be connected to pin N, and nets with a high probability of being 1 should be connected to pin P of key-gates.

https://doi.org/10.22362/ijcert/2020/v7/i07/v7i0706

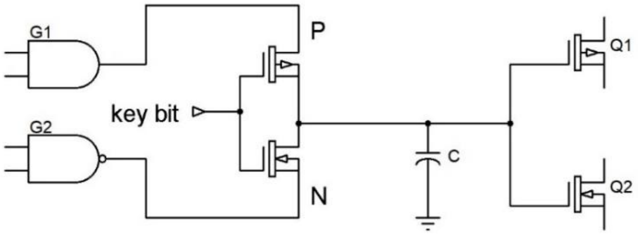


Figure 5. Modeling the capacitive effects of a key-gate's output using a capacitive load C

TABLE III. POWER OVERHEAD RESULTS FOR THE ENHANCED KEY-GATE EMBEDDING STRATEGY

	MUX-TF	Inverter-based MUX	Improved Inverter-based MUX
s1423	2.08%	3.21%	0.77%
s5378	0.48%	1.67%	0.03%

To investigate the effectiveness of the proposed solution, the inputs to all key-gates in s1423 and s5378 benchmark circuits were swapped, and the results for power overhead are reported in Table 3.

As can be seen in the last column of the table, a considerable improvement in power performance is achieved by wisely connecting signal wires to the inputs of the key-gates. This means that the results reported in Table 2 are not necessarily the best possible results, and still lower power overhead can be expected to achieve by connecting signals to Inverter-based MUXs according to their signal probabilities.

In terms of timing overhead, the signals should be propagated through one transistor level of key-gates in the proposed approach. The key-gates in other techniques, however, contain more than one level of transistors and incur more signal delay. It is, therefore, obvious that the proposed technique incurs less timing overhead compared to other approaches.

5. Conclusion

In this paper, a low overhead logic encryption technique was presented. In the proposed method, typical CMOS inverter cells are embedded in the design as key-gates. By connecting their pull-up and pull-down networks to internal signal nets instead of VDD and GND, these inverter cells act as a MUX that can select between their input signals, hence providing the required logic locking functionality. The proposed method does not require any specialized technology library and can be fully implemented through typical CMOS cells. The performance evaluation of the proposed method on a set of ISCAS'89 benchmarks reveals the efficiency of the method over other encryption techniques.

References

[1] U. Guin, D. DiMase, M. Tehranipoor, "Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead," Journal of Electronic Testing, Vol. 30, Issue 1, pp. 9-23, 2014.

[2] S. Bhunia, M.S. Hsiao, M. Banga, S. Narasimhan, "Hardware Trojan Attacks: Threat Analysis and Countermeasures," In Proc. IEEE, Vol. 102, Issue 8, 2014.

[3] Y. Alkabani, F. Koushanfar, M. Potkonjak, "Remote Activation of ICs for Piracy Prevention and Digital Right Management," In Proc. ICCAD, pp. 674-677, 2007.

[4] F. Koushanfar, "Provably Secure Active IC Metering Techniques for Piracy Avoidance and Digital Rights Management," IEEE Transactions on Information Forensics and Security, Vol. 7. Issue 1, pp. 51-63, 2012.

[5] J.A. Roy, F. Koushanfar, I.L. Markov, "EPIC: Ending Piracy of Integrated Circuits," In Proc. DATE, pp. 1069-1074, 2008.

[6] J. Rajendran, H. Zhang, C. Zhang, G.S. Rose, Y. Pino, O. Sinanoglu, R. Karri, "Fault Analysis-based Logic Encryption," IEEE Transactions on Computers, Vol. 64, Issue 2, pp. 410-424, 2013.

[7] J.A. Roy, F. Koushanfar, I.L. Markov, "Protecting Bus-based Hardware IP by Secret Sharing," In Proc. DAC, pp. 846-851, 2008.

[8] S. Zamanzadeh, A. Jahanian, "Automatic Netlist Scrambling Methodology in ASIC Design Flow to Hinder the Reverse Engineering," In Proc. VLSI-SoC, pp. 52-53, 2013.

[9] J. Zhang, "A Practical Logic Obfuscation Technique for Hardware Security," IEEE Transactions on VLSI, Vol. 24, Issue 3, pp. 1193-1197, 2016.

[10] A. Baumgarten, A. Tyagi, J. Zambreno, "Preventing IC Piracy Using Reconfigurable Logic Barriers," IEEE Design & Test of Computers, Vol. 27, Issue 1, pp. 66-75, 2010.

[11] K. Juretues, I. Savidis, "Reduced Overhead Gate Level Logic Encryption," In Proc. GLSVLSI, 2016.

[12] A. Morgenshtein, A. Fish, I.A. Wagner, "Gate-Diffusion Input (GDI): A Power-Efficient Method for Digital Combinatorial Circuits," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, Vol. 10, No. 5, 2002.

[13] Predictive Technology Model (PTM), Available online at: http://ptm.asu.edu.

Authors Profile

Ghobad Zarrinchian received his Ph.D. degree in Computer Engineering from Amirkabir University of Technology in 2017. He joined Islamic Azad University of Yadegar-e-Imam Khomeini (rah) Shahr-e-Rey in 2016. He is currently an assistant professor at the Department of Computer Engineering. His research interests include electronic systems design and design for hardware security.