

# Side channel power analysis resistance evaluation of masked adders on FPGA

Yilin Zhao<sup>1</sup>, Hiroki Nishikawa<sup>2</sup>, Xiangbo Kong<sup>1</sup>, Hiroyuki Tomiyama<sup>1</sup>

<sup>1</sup>Graduate School of Science and Engineering, Ritsumeikan University, Shiga, Japan

<sup>2</sup>Graduate School of Information Science and Technology, Osaka University, Osaka, Japan

## Article Info

### Article history:

Received Oct 19, 2022

Revised Dec 12, 2022

Accepted Dec 26, 2022

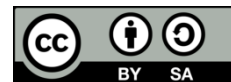
### Keywords:

Adders  
FPGA  
Masking  
Power analysis attack  
Side channel attacks  
T-test  
Xilinx

## ABSTRACT

Since many internet of things (IoT) devices are threatened by side-channel attacks, security measures are essential for their safe use. However, there are a variety of IoT devices, so the accuracy required depends on the system's application. In addition, security related to arithmetic operations has been attracting attention in recent years. Therefore, this paper presents an empirical experiment of masking for adders on field programmable gate arrays (FPGAs) and explores the trade-off between cost and security by varying the bit length of the mask. The experimental results show that masking improves power analysis attack resistance, and increasing the bit length of the random numbers used for masking increases security. In particular, the series-connected masked adder is found to be effective in improving power analysis attack resistance.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



## Corresponding Author:

Yilin Zhao

Graduate School of Science and Engineering, Ritsumeikan University

1-1-1 Nojihigashi, Kusatsu, Shiga, 525-8577, Japan

Email: irin.cho@tomiyama-lab.org

## 1. INTRODUCTION

In recent years, our lives have become more convenient due to the spread of Internet of things (IoT) devices. However, those devices are exposed to the physical environment, making them vulnerable to side-channel attacks [1], [2]. Representative side-channel attacks include power analysis attacks that analyze power consumption [3], timing attacks that use processing time [4], and electromagnetic wave analysis attacks that use electromagnetic waves [5] and more [6]. Recently, software-based remote power side-channel attacks have also been proposed, which allow attacks to be carried out without physically approaching the target system [7]-[10]. Additionally, to improve the performance of side-channel analysis, methods using deep learning [11] and conditional generative adversarial networks [12] are proposed. Moreover, security is attracting increasing attention not only for specific devices and circuits but also for arithmetic operations. Among them, power analysis attacks have attracted particular attention because the equipment used to analyze power is not expensive. According to the work [3], the power analysis attacks are a serious threat to security on field programmable gate arrays (FPGAs) as well as application-specific integrated circuits. Örs *et al.* [13] introduces a setup for power analysis attacks on FPGAs. Therefore, target devices and circuits must be designed securely so that attackers cannot extract security keys. Therefore, one of the countermeasures against side channel attacks is a technique called masking. Masking is a technique that uses a random mask to make the leakage of values handled by the implementation independent of sensitive inputs and intermediate variables [14]. Since the random mask should be unbreakable, the side-channel information from individual shares will not reveal sensitive variables. Even if an attacker acquires a leak of the side-channel information, the sensitive intermediate variables are sufficiently mysterious to

remain secret. There are various methods of masking, such as multiplicative masking [15] and masked AND operation [16]. Masoumi *et al.* [17] presents an example implementation of masking to advanced encryption standard (AES). Gravelier *et al.* [18], masking is also effective as a countermeasure against remote attacks, which have received attention in recent years. Wu and Picek [19] aims to investigate the effectiveness of a deep learning technique called autoencoder for masking as a future work. However, detailed methods of mystifying cryptographic variables by masking processes have rarely been studied [20]. Moreover, there are some issues by implementing countermeasures, such as increased circuit area, power consumption and implementation cost [21]. Therefore, a research is in progress to improve power analysis attack resistance at low cost [22]. In addition, IoT devices have a variety of sizes and applications, and each device has different acceptable cost, power and required safety accuracy. Therefore, by devising masking methods and mask itself, we can expect a trade-off between implementation cost and safety.

This paper presents an empirical study on masking for adders on FPGA, focusing on one of the most basic components of circuits, the adder. The contributions of this paper are three-fold. First, we examine tolerance towards power analysis attacks for three types of adders with basic circuit configurations. Specifically, we investigate a ripple carry adder (RCA), a carry lookahead adder (CLA), and a RCA with built-in carry chains. Second, two types of masked adders will be compared: the series-connected masked adder and the compression-based masked adder. We also evaluate the effect of the bit length of the mask on the power analysis attack resistance. For each evaluation, we first synthesize the circuit and analyze the power consumption. From the obtained power consumption, we statistically evaluate the degree of leakage of power-based side-channel information by T-test. By applying masking to adders, this research aims to design security circuits that are highly versatile, not for a specific device. Moreover, we will explore the trade-off between cost and security by changing the bit length of the mask and performing the evaluation. To the best of our knowledge, this is the first paper which studies the power side-channel leakage of different adders on FPGA.

The rest of this paper is organized as follows. In section 2, we describe the three adders used in this study. Section 3 presents the two circuits used in our study and introduces the masking methods in these circuits. Section 4 describes the experimental scenario and the results of the side-channel attack resistance. Section 5 summarizes this paper.

## 2. POWER SIDE CHANNEL LEAKAGE ANALYSIS OF NON-MASKED ADDERS

An adder is the simplest of the arithmetic circuits and is an essential component in the processing of the system. In recent years, as devices have become smaller and faster, performance aspects such as power consumption, Circuit area and delay time have become more important factors. Knowles [23] proposes a circuit with a reduced area compared to existing adders. In addition, the structure in the work Tyagi [24] is implemented considering the trade-off between circuit area and delay. However, there are few studies that provide a detailed survey of adder security. Therefore, this study evaluates the power analysis attack resistance of simple adders. This section describes the structure of three adders on the FPGA. Then we evaluate the power analysis attack tolerance for the three adders presented in subsection 2.1.

### 2.1. Adders on FPGA

This subsection describes features and structures of the three adders on a FPGA. Moreover, we explain the flow of operations in each adders with equations. A FPGA is integrated devices that allow designers to configure the logic circuit and we can change the behavior of the circuitry as often as we like. In this study, we examine a RCA, a CLA, and an RCA with fast carry chains. These three types of adders have different characteristics in terms of circuit area and arithmetic delay. In this study, Xilinx 7-series FPGA is assumed [25].

#### 2.1.1. Ripple carry adder

RCA has the most basic structure of all adders. A  $N$ -bit RCA is designed by sequentially chaining  $N$  full-adders (FAs). Figure 1 shows an organization of a 4-bit RCA. RCA has A, B and carry-in as inputs and outputs carry-out and sum. As shown in Figure 1, in RCA the carry of the previous digit is connected to the input of the next digit. Therefore, the worst-case carry propagation goes through all digits, resulting in large computation times.

Each FA calculates a sum of the digit and a carry signal to the next FA as shown in:

$$S_i = A_i \oplus B_i \oplus C_i \quad (1)$$

$$C_{i+1} = A_i \cdot B_i + B_i \cdot C_i + C_i \cdot A_i \quad (2)$$

the Xilinx FPGA consists of 6-input LUTs, and each 6-input LUT can be configured as two 5-input LUTs. Therefore, an FA is mapped to a 6-input LUT, and a  $N$ -bit RCA utilizes  $N$  LUTs.

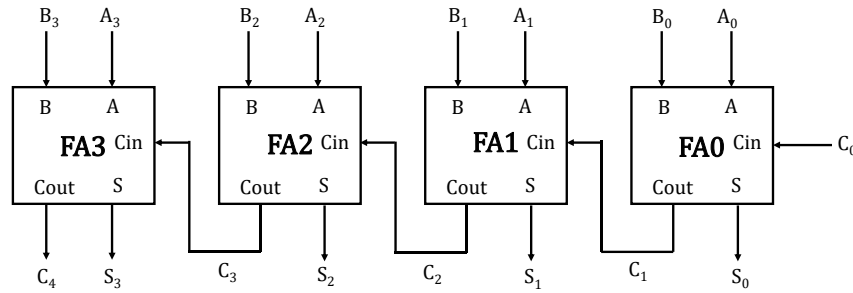


Figure 1. Circuit diagram of a 4-bit RCA

### 2.1.2. Carry lookahead adder

CLA is another common circuit. CLA reduces delay time by calculating carries first, before all calculations are completed. Figure 2 shows an organization of a 4-bit CLA. CLA calculates carry generation and carry propagation as shown in (3), (4).

$$G_i = A_i \cdot B_i \quad (3)$$

$$P_i = A_i + B_i \quad (4)$$

Then, carry to the next digit is computed by (5).

$$C_{i+1} = G_i + P_i C_i \quad (5)$$

By recursively calculating (5), we obtain (6)-(9).

$$C_1 = G_0 + P_0 C_0 \quad (6)$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0 \quad (7)$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 + P_2 P_1 P_0 C_0 \quad (8)$$

$$C_4 = G_3 + P_3 C_3 = G_3 + P_3 G_2 + P_3 P_2 G_1 + P_3 P_2 P_1 G_0 + P_3 P_2 P_1 P_0 C_0 \quad (9)$$

In this circuit, the carry signals can be computed in parallel and all digit carries are determined from the inputs  $A_i$  and  $B_i$  only, so the arithmetic time is not proportional to the number of digits as in the RCA. Therefore, CLAs compute faster than RCAs at the cost of more LUTs. According to the work Uma *et al.* [26], CLA has a shorter delay time than many adders including RCA.

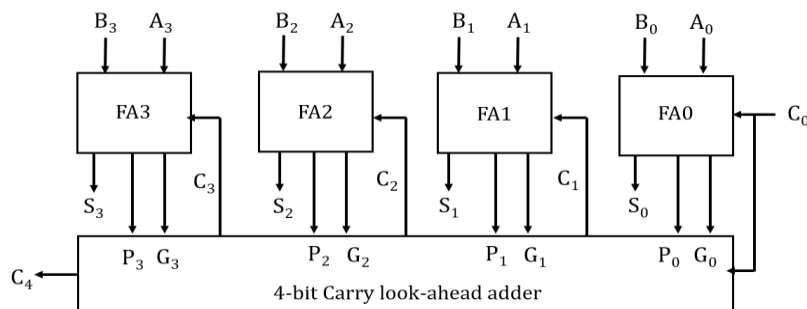


Figure 2. Circuit diagram of a 4-bit CLA

### 2.1.3. Ripple carry adder with fast carry chains

Xilinx FPGAs have special hardware blocks, named Carry4 [27], to rapidly propagate carry signals. At the output of each LUT, an XOR gate and a 2-input multiplexer are placed, and the multiplexers are chained, as shown in Figure 3. In this circuit, one LUT is used to perform 1-bit addition. The carry output Cout of the last stage can be extended to 128 bits by connecting it to the carry input Cin of the next digit.

Using the carry chains, adders can be calculated in (10)-(12).

$$P_i = A_i \oplus B_i \quad (10)$$

$$S_i = P_i \oplus C_{in} \quad (11)$$

$$C_{i+1} = G_i + P_i \cdot C_i = \bar{P}_i \cdot A_i + P_i \cdot C_i \quad (12)$$

As shown in (10), (11) and (12) are calculated by an LUT, an XOR gate, and a multiplexer, respectively. The carry propagation P is computed as the exclusive OR of A and B. The sum S is calculated by the exclusive OR of A, B, and the carry input Cin. The carry output of the next digit is represented by (12). Similar to the 32-bit RCA, a 32-bit RCA with fast carry chains uses 32 LUTs. Unlike the RCA, however, the carry-out signal of a digit does not go into the LUT of the next digit. Carry signals are propagated through the built-in fast carry chains, leading to a shorter critical-path delay than the RCA shown in sub-subsection 2.1.1.

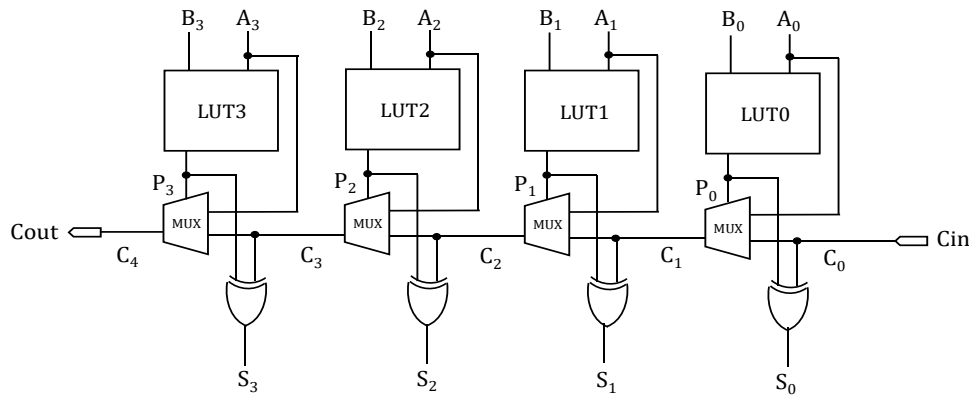


Figure 3. Circuit diagram of a 4-bit RCA with fast carry chains

## 2.2. Synthesis result

First, we synthesize the three 32-bit adders. We synthesize with AMD Xilinx Vivado 2019.2. The target device is assumed to be Artix-7 FPGA. Our synthesis has been performed with enabling a couple of optimization options to the performance. The results on hardware resources and delay are shown in Table 1. In terms of both hardware cost and performance, RCA with fast carry chains is the best among the three. Table 2 shows the number of resources and delay times for 128-bit CLA and RCA with fast carry chains. 128-bit adder is an extended circuit of the 32-bit adder described in subsection 2.1. The RCA is not evaluated for performance and tolerance at 128 bits because its delay time is larger than that of the other two adders as shown in Table 1. The results in Table 2 show that the RCA with fast carry chains outperforms the CLA in both number of resources and performance. Comparing Tables 2 and 3, the number of LUTs and the delay time are larger for the 128-bit circuit than for the 32-bit circuit for both CLA and RCA with fast carry chains.

Table 1. Resource and delay of the 32-bit adders

Adders	LUTs	Delay (ns)
RCA	64	18.991
CLA	195	7.012
RCA with fast carry chains	32	6.118

Table 2. Resource and delay of the 128-bit adders

Adders	LUTs	Delay (ns)
CLA	681	11.718
RCA with fast carry chains	128	8.470

Table 3. The number of T-values violating a security criterion ( $>|\pm 4.5|$ ) of the 32-bit adders

Adders	No. of T-values over $ \pm 4.5 $	Maximum T-value
RCA	50	29.72
CLA	43	23.86
RCA with fast carry chains	21	13.97

### 2.3. Power analysis

Next, we analyze the power consumption of the three adders based on post-synthesis simulation. We use the power analysis tool presented in the work [28] and Vivado toolkit for power analysis. This tool can observe dynamic temporal changes in power. The results are shown in Figure 4. The X-axis represents hundreds of testbenches of which each contains 2000 test vectors, and the Y-axis represents the power consumption. RCA, CLA, and RCA with fast carry chains results are shown in gray, orange and blue respectively. Only the logic and signal power are evaluated since the adders designed for this study are too small compared with the FPGA capacity. The results show the RCA with fast carry chains consumes the least power. Figure 5 shows the power consumption of the two 128-bit adders. This result shows that RCA with fast carry chains operates at lower power consumption than CLA. Comparing Figures 4 and 5, the power consumptions of both CLA and RCA with fast carry chains are higher for the 128-bit circuit than for the 32-bit circuit.

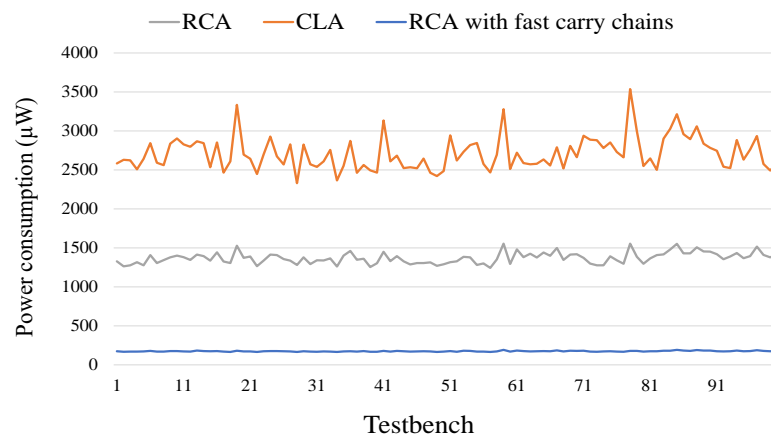


Figure 4. Power consumption of 32-bit adder

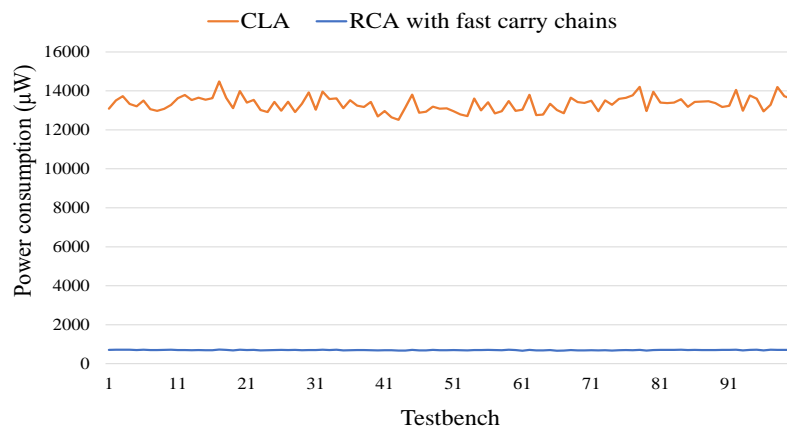


Figure 5. Power consumption of 128-bit adder

## 2.4. Power side channel leakage analysis

In order to evaluate tolerance to side-channel leakage, we conduct T-test [29]-[31] for the adder circuits. Side-channel attacks such as power analysis attacks, use information about intermediate values in the side-channel traces observed from a device. Therefore, a statistical hypothesis test can be used to detect whether sensitive intermediate values have a significantly effect on the measured data. The Welch's T-test is one of the measures of side-channel attack resistance and is a test of the difference between the means of two sets of data. This test can determine whether the power traces obtained by an attacker leak data about the secret information. The equation of the T-test is shown (13).

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{S_1^2/N_1 + S_2^2/N_2}} \quad (13)$$

Here,  $\bar{X}_1$  and  $\bar{X}_2$  represent the average power consumption for random and fixed inputs,  $S_1$  and  $S_2$  are the standard deviation, and  $N_1$  and  $N_2$  represent the number of samples. The T-value  $t$  is desire to be less than  $\pm 4.5$  to meet security criteria. Details on safety criteria and specific method of the T-test are described in the work [29].

We conduct a T-test for the power traces obtained in subsection 2.3. The obtained T-values of the 32-bit adders are shown in Figure 6. The X-axis represents hundreds of testbenches of which each contains 2000 test vectors, and the Y-axis represents the T-value. The red lines in the graph indicate the standard values of 4.5 and -4.5 for the attack tolerance evaluation. The results show that all circuits have many points with T-values above  $\pm 4.5$ , indicating that they are vulnerable to power analysis attacks. Table 3 shows the number of T-values over  $\pm 4.5$  and the maximum T-value for each circuit. Table 3 shows that RCA is the least tolerant since T-values exceeds  $\pm 4.5$  in 50 test benches. The maximum T-value of RCA is also the largest among the three circuits. Based on this result, RCA evaluation is omitted for the 128-bit adder. The RCA with fast carry chains is superior in terms of both the number of times the T-value exceeds  $\pm 4.5$  and the maximum value. However, the RCA with fast carry chains also still exceeds the security criteria for T-values in 21 test benches. Figure 7 and Table 4 show the results of side channel leakage analysis for 128-bit CLA and RCA with fast carry chains. Figure 7 indicates that T-values are frequently exceeded over 4.5 both CLA and RCA with fast carry chain. Also, the T-values of CLA are seemingly larger than that of RCA with fast carry chains. Table 4 shows that the RCA with fast carry chains has a smaller number of times the T-value exceeds  $\pm 4.5$  than the CLA, and that the maximum value is also smaller. Comparing the 32-bit and 128-bit results for CLA and RCA with fast carry chains, it can be seen that the 128-bit adder has a higher T-value exceeding  $\pm 4.5$  more times. The maximum T-value is larger for the 128-bit circuit than for the 32-bit circuit for the CLA, and slightly larger for the 32-bit circuit for the RCA with fast carry chains. This result indicates that vulnerability tends to increase as the number of bits increases. In addition, for the adders verified in this section, there are some patterns which failed to meet the safety criterion. Therefore, a simple adder without countermeasures against power analysis attacks is not completely secure in terms of security.

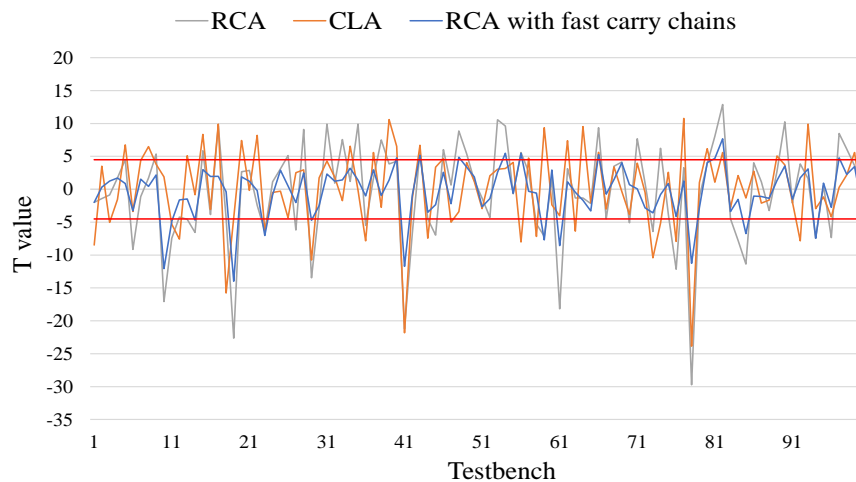


Figure 6. T-values of 32-bit adder

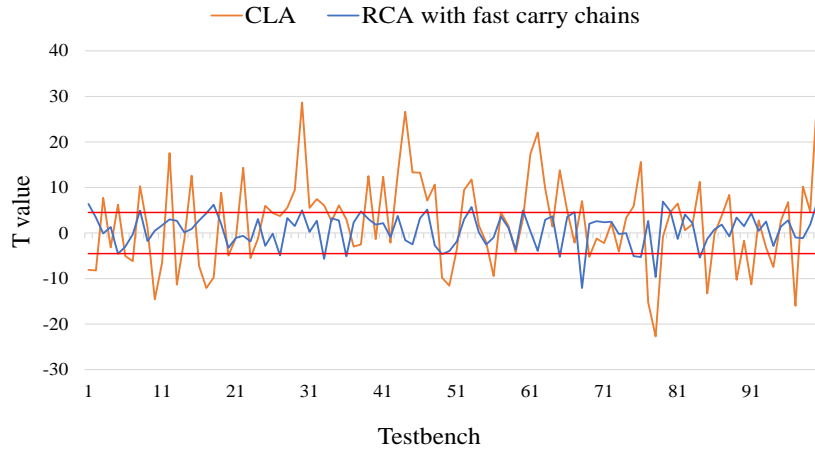


Figure 7. T-values of 128-bit adder

Table 4. The number of T-values violating a security criterion ( $>|\pm 4.5|$ ) of the 128-bit adders

Adders	No. of T-values over $ \pm 4.5 $	Maximum T-value
CLA	65	31.65
RCA with fast carry chains	23	12.10

### 3. POWER SIDE CHANNEL LEAKAGE ANALYSIS OF MASKED ADDERS

The results in section 2 show that adders without attack countermeasures are vulnerable to power analysis attacks. Therefore, we focus on a side-channel attack countermeasure called masking and apply masking to adders. In this section, we introduce two types of masked adders and evaluate their attack resistance. First, the structures of two types of 128-bit masked adders are presented. Then, the power analysis attack resistance of each masked adder is evaluated and compared to the resistance of a simple non-masked adder.

#### 3.1. Fully masked arithmetic adders

In this subsection, we describe the structure of the two masked adders used in this study. Specifically, we examine a masked adder using fast carry chain of Xilinx and a masked adder using carry save adder. In this paper, we make the following assumptions. An attacker inputs augend A to the adder and tries to identify the addend B through observing a side-channel leakage of the power consumption. Note that we assume the attacker cannot observe B and the sum C. In addition, the adders presented in this section assume Xilinx 7-series FPGA the same as section 2.

##### 3.1.1. Series-connected masked adder

Figure 8 shows the structure of a simple masked adder. It consists of two carry-propagate adders and one carry-propagate subtractor connected in series. First, the random number R generated by the pseudo-random number generator (PRNG) is added to A. Then, B is added to the output of A+R. Finally, R is subtracted to derive the final output of C ( $=A+B$ ). In this example, we assume that A is masked with adding R, and its intermediate variable mystifies the side-channel leakage of the power consumption by the subsequent calculation. The RCA with fast carry chains introduced in Section 3 is used for the additive part of this masked adder. Based on the results in section 3, the RCA with fast carry chains is the best adder in terms of number of resources, delay time, and power analysis attack resistance. In this experiment, the pseudo-random number generator is excluded from the attack resistance evaluation, and only the arithmetic unit part is evaluated. Therefore, a detailed description of the PRNG scheme is omitted.

##### 3.1.2. Compression-based masked adder

Figure 9 shows the compression-based masked adder. The series-connected masked adder previously presented has a long delay in adding the augend A, addend B, and random number R, since it is connected to two carry-propagate adders in series. Therefore, to reduce the delay, the three inputs are first compressed into two terms by carry save adder (CSA). The architecture of CSA is given in Figure 10. CSA is used to compute the sum of three or more numbers. Unlike other adders, it outputs two numbers of the same size as the input. The delay due to the carry is reduced in this structure since the carry is not propagated

through the stages [32]. According to the work [33], the CSA achieves the shortest latency among seven types of adders.

The three numbers A, B, and R are computed by CSA as partial sum and shift carry as shown in (14), (15).

$$S_i = A_i \oplus B_i \oplus R_i \quad (14)$$

$$C_i = (A_i \cdot B_i) + (A_i \cdot R_i) + (B_i \cdot R_i) \quad (15)$$

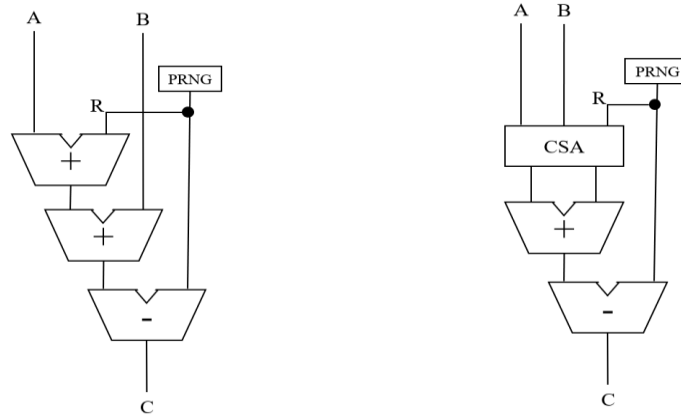


Figure 8. Series-connected masked adder      Figure 9. Compression-based masked adder

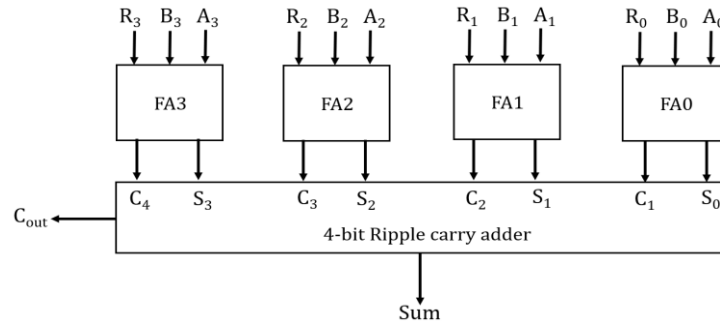


Figure 10. Circuit diagram of a 4-bit CSA

### 3.2. Synthesis result

In this subsection, we synthesize the two types of masked adders described in subsection 3.1 and compare their performance. Masked adders have 128 bits and use AMD Xilinx Vivado 2019.2 for synthesis. The target device is assumed to be an Artix-7 FPGA. We have enabled several optimization options for performance when synthesizing the circuit. Table 5 shows the number of LUTs and delay times for the non-masked circuit and the two masked circuits. However, this result does not include the area and delay of the pseudo-random number generator. Table 5 shows that the non-masked adder uses 128 LUTs and has the smallest circuit area and shortest delay time because there are no additional calculations for masking. Comparison of the two masked adders shows that the series-connected masked adder has fewer LUTs than the compression-based masked adder, and the compression-based masked adder operates with a shorter delay than the series-connected masked adder.

Table 5. The number of LUTs, delay and power consumption for the synthesized adder circuits

Masks	Adders	LUTs	Delay (ns)	Power consumption (μW)
0-bit (Non-masked)	Carry propagate adder	128	8.470	699
128-bit (Fully masked)	Series-connected masked adder	384	11.048	9675
	Compression-based masked adder	511	10.268	9708



### 3.3. Power analysis

Next, we evaluate the power consumption. We use Vivado and the tools presented in the work [28] to perform power analysis. This tool is capable of observing the dynamic time variation of power consumption and can analyze power changes in detail. Figure 11 shows the non-masked circuit and the two masked circuits power consumption. The X-axis represents hundreds of testbenches of which each contains 2000 test vectors, and the Y-axis represents the power consumption. In the results, we focus on only the logic and signal power consumption. Table 5 also shows the average power consumption for each circuit. The results show the non-masked circuit consumes the least power since due to none of additional computation for masking. The power consumption on series-connected masked adder looks almost the same as that on compression-based masked adder.

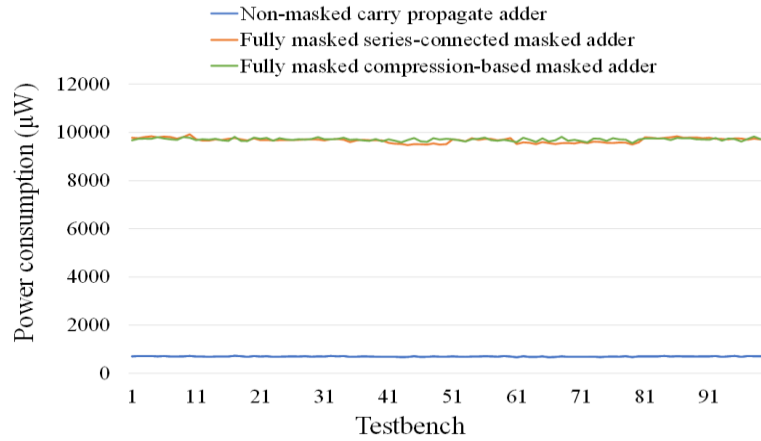


Figure 11. Analysis results of power consumption

### 3.4. Power side channel leakage analysis

Finally, we perform a T-test on the results of the power consumption analysis to evaluate the resistance against side-channel attacks. The results of the T-test for the two types of mask adders are shown in Figure 12 and Table 6. Figure 12 shows the T-values for each test bench, with the security criterion of  $\pm 4.5$  indicated by the red line. Table 6 shows the number of times T-values exceeded  $\pm 4.5$  and the maximum T-value among the 100 test cases. Figure 4 shows that without masking, the T-value exceeds the security criterion in many cases, 23 times from Table 6. On the other hand, in the masked circuit, the T-value of the series-connected masked adder stays between  $-4.5$  and  $4.5$  in any case. The T-value of the compression-based masked adder is clearly larger than that of the series-connected mask adder, and Table 6 shows that the T-value exceeds the security criterion eight times. This result shows that the series-connected masked adder is highly resistant to power analysis attacks.

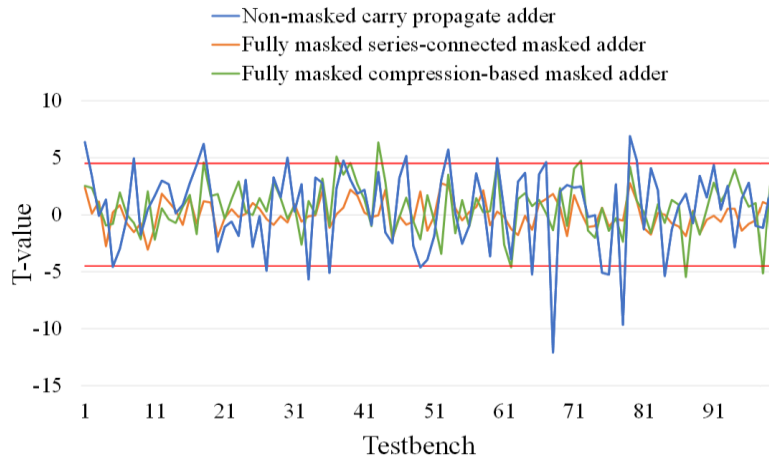


Figure 12. The results of T-test

Table 6. The number of T-values violating a security criterion ( $>|\pm 4.5|$ )

Masks	Adders	No. of T-value over $ \pm 4.5 $	Maximum T-value
0-bit (Non-masked)	Carry propagate adder	23	12.10
128-bit (Fully masked)	Series-connected masked adder	0	3.07
	Compression-based masked adder	8	6.35

#### 4. POWER SIDE CHANNEL LEAKAGE ANALYSIS OF PARTIALLY MASKED ADDERS

In the previous section, we have compared the circuits with non-masked and full-masked in 128-bit adders. In this section, unlike the previous section, we utilize the masks whose bit-widths of random numbers are different (Random number is 32, 64 or 96 bits). We present how to design masks that use 32, 64 or 96 bits random numbers in subsection 4.1. Then, we conduct the same experiments as in section 3 for the masked adders introduced in subsection 4.1 to evaluate power analysis attack resistance. The experiments aim to explore the trade-off between the resource cost, power consumption, and side-channel attack resistance.

##### 4.1. Partially masked arithmetic adders

This subsection describes two methods of creating masks when random numbers are 32, 64, or 96 bits. In this study, masks are designed in two different ways for each bit of random numbers used. The first method is called lower bit mask, in which random numbers are placed in the lower bits. The second method is called distributed bit mask, which alternates random numbers and zeros. A total of six masks are prepared by these two methods.

###### 4.1.1. Lower n-bit mask

Figure 13 shows the structure of three masks. The adder uses the same circuit as in section 4, which performs 128-bit addition. This method uses random numbers for the lower  $N$ -bits ( $N=32, 64, 96$ ) and assigns 0 to the upper  $(128-N)$  bits. We prepare the following three masks:

- Lower 32-bit mask: Upper 96 bits in 128-bit are set to 0. The rest of lower 32 bits are assigned to a 0-1 random value  $R$  as shown in Figure 13(a).
- Lower 64-bit mask: Upper 64 bits in 128-bit are set to 0. The rest of lower 64 bits are assigned to a 0-1 random value  $R$  as shown in Figure 13(b).
- Lower 96-bit mask: Upper 32 bits in 128-bit are set to 0. The rest of lower 96 bits are assigned to a 0-1 random value  $R$  as shown in Figure 13(c).

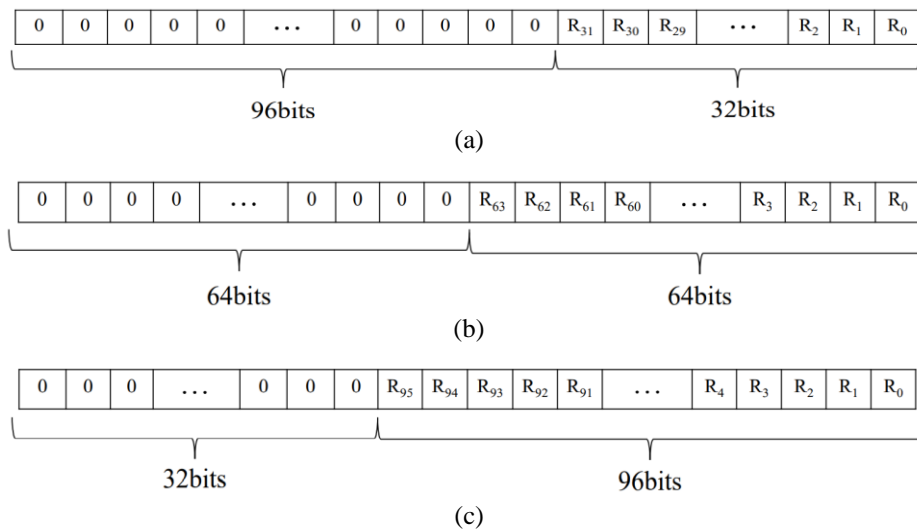


Figure 13. Lower random masks (a) lower 32-bit mask, (b) lower 64-bit mask, and (c) lower 96-bit mask

###### 4.1.2. Distributed n-bit mask

Second method is to alternately input 0 and a random number. Figure 14 shows the mask structures when random numbers are 32-bit, 64-bit and 96-bit. For each bit, a mask is created by distributing a random number and 0, starting with the least significant bit. We created the three masks:

- Distributed 32-bit mask: A 0-1 random value  $R$  is set every four bits and there is the mask for 32-bit in total as shown in Figure 14(a).
- Distributed 64-bit mask: A 0-1 random value  $R$  is set every two bits and there is the mask for 64-bit in total as shown in Figure 14(b).
- Distributed 96-bit mask: 0 is set every four bits and the others are set to 0-1 random values  $R$ . The total number of bits for the mask is 96-bit as shown in Figure 14(c).

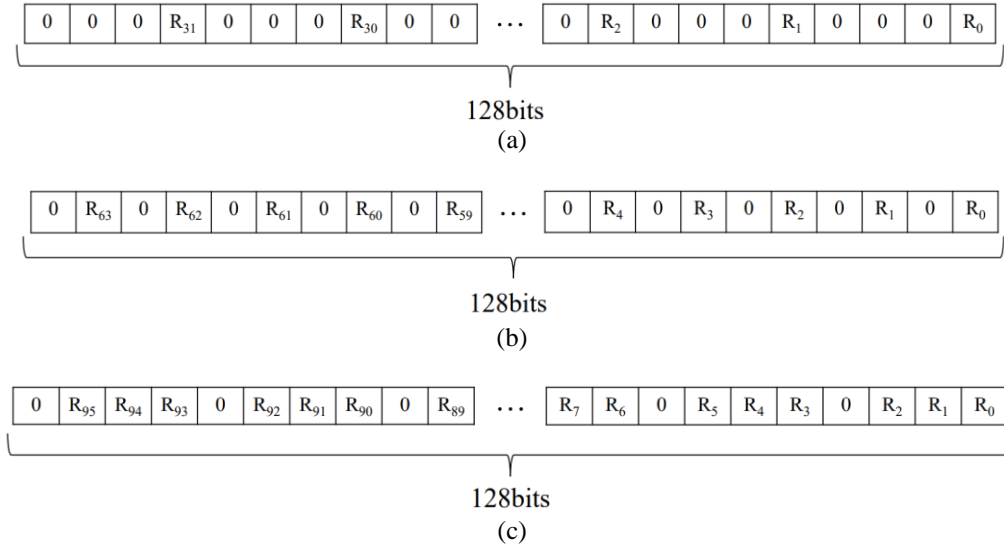


Figure 14. Distributed random masks (a) distributed 32-bit mask, (b) distributed 64-bit mask, and (c) distributed 96-bit mask

#### 4.2. Synthesis result

We synthesize masked adders the same as in the previous section. The results of the number of look-up-tables (LUTs) and delay for each circuit are shown in Table 3. Distributed  $n$ -bit masks are denoted as Dist.  $n$ -bit in the table. The best value for each bit number is shown in red. If the number of LUTs for the masked circuits are larger than that for the non-masked circuit shown Table 5 since the masked circuits require to mystify the intermediate variable by masking. In addition, in the series-connected masked adder with the lower bits masked, the number of LUTs is increased as increasing the number of bits to mask. We highlight the circuits with the smallest number of LUTs and the shortest delay. In the case of 32-bit masks, the series-connected masked adder by the lower 32-bit mask uses 288 LUTs. In terms of the delay, the compression-based masked adder by the lower 32-bit mask takes 10.268 ns. As shown in the table, it is found that the trend has been observed in any case. Overall, the series-connected masked adder uses the smallest number of LUTs, and the compression-based masked adder takes the shortest delay. We focus on the type of masks, the lower-bit masked adders require fewer resources and have shorter delay than distributed-bit masked adder.

#### 4.3. Power analysis

The average power consumption of hundred test patterns for each adder is shown in Table 7. From Table 7, the power consumption of the compression-based masked adder is larger than that of the series-connected masked adder for any number of bits. This is due to the larger circuit area of the compression-based masked adder. Compared to the masks, the lower-bit masking shows the smaller power consumption than the distributed-bit masking. Thus, masks with upper bits tend to have higher dynamic power consumption. Figure 15 shows the power consumption of the Lower  $n$ -bit masked adder and Figure 16 shows the power consumption of the distributed  $n$ -bit masked adder. The X-axis represents hundreds of testbenches, and the Y-axis represents the power consumption. Looking at lower-bit series-connected masked adders which have the lowest power consumption, the average power consumption for 32-bit is 4283  $\mu$ W, for 64-bit is 6128  $\mu$ W and for 96-bit is 7934  $\mu$ W. As the Figure 15 and 16 illustrates, this trend is also seen in other masked adders. To sum up, the power consumption increases if the number of bits for masking is large.

Table 7. Synthesis results with regard to the number of LUTs, delay and power consumption

Masks	Adders	LUTs	Delay (ns)	Power consumption ( $\mu$ W)
0 bit (Non-masked)	Carry propagate adder	128	8.470	699
Lower 32 bits	Series-connected masked adder	<b>288</b>	11.048	<b>4283</b>
	Compression-based masked adder	511	<b>10.268</b>	6722
Dist. 32 bits	Series-connected masked adder	512	11.494	4772
	Compression-based masked adder	639	10.738	6923
Lower 64 bits	Series-connected masked adder	<b>320</b>	11.048	<b>6128</b>
	Compression-based masked adder	511	<b>10.268</b>	7714
Dist. 64 bits	Series-connected masked adder	512	11.494	6814
	Compression-based masked adder	639	10.738	8337
Lower 96 bits	Series-connected masked adder	<b>352</b>	11.048	<b>7934</b>
	Compression-based masked adder	511	<b>10.268</b>	8738
Dist. 96 bits	Series-connected masked adder	512	11.511	7994
	Compression-based masked adder	639	10.738	9673
128 bits (Fully masked)	Series-connected masked adder	<b>384</b>	11.048	<b>9675</b>
	Compression-based masked adder	511	<b>10.268</b>	9708

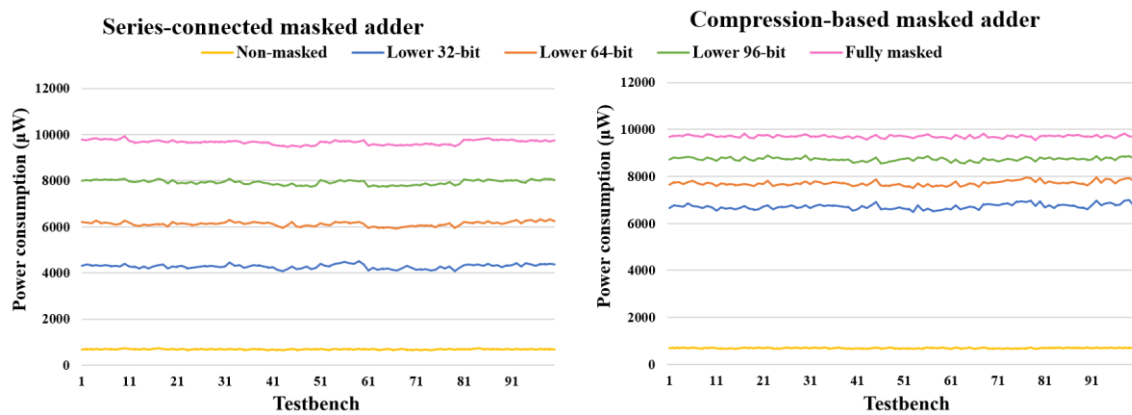


Figure 15. Power consumption of lower-bit masked adders

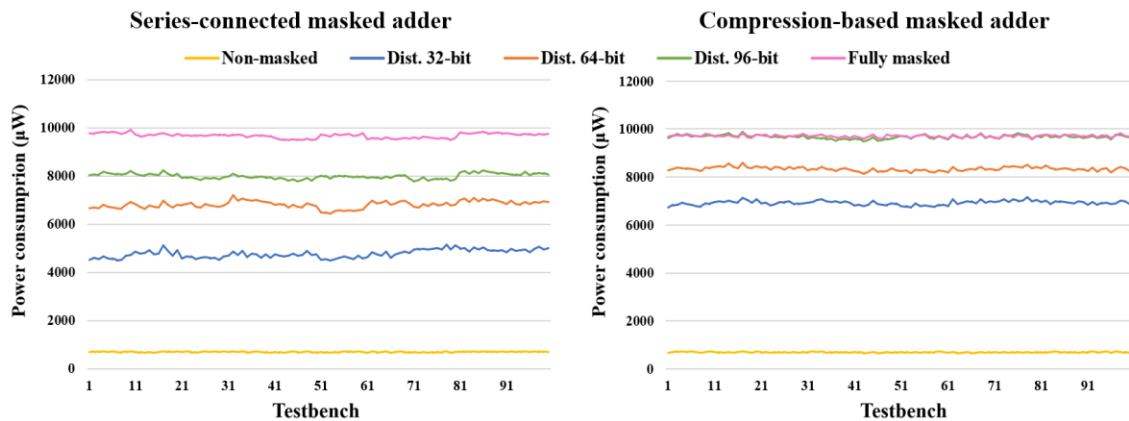


Figure 16. Power consumption of distributed-bit masked adders

#### 4.4. Power side channel leakage analysis

The T-test was performed using the results of the power analysis obtained in subsection 4.3, and the results obtained are shown in Table 8. Table 8 shows the number of times the T-value for each circuit exceeded  $\pm 4.5$  out of 100 test patterns and the maximum T-value. If the T-value is greater than  $\pm 4.5$ , the circuit is considered less tolerant to power analysis attacks.

Table 8 shows that in the series-connected mask adder, the T-value exceeds  $\pm 4.5$  32 times for the lower 32-bit mask and 28 times for the distributed 32-bit mask, more than in the unmasked circuit, which

may compromise security. We now focus on the series-connected masked adder that use distributed masks. When the mask is 32 bits, the T-value exceeds  $\pm 4.5$  28 times, but as the number of bits in the mask increases to 64, 96, and 128 bits, the number of times the T-value exceeds  $\pm 4.5$  decreases to 16, 1, and 0 times. Also, as the number of bits in the mask increases, the maximum T-value also decreases. This tendency is also seen in adders using other masks. This result indicates that using a random number of 64 bits or more for masks improves the attack resistance compared to an adder without masking. Also, the vulnerability decreases as the number of random bits used for masking increases. Figures 16 and 17 show the results of comparing the T-values of the adders for each type of mask. The X-axis shows the T-value and the Y-axis represents the number of testbenches. The red line indicates the security standard. Figures 17 and 18 show that circuits with non-mask or a mask of 32 bits greatly exceed the security standard. However, in fully masked circuits, T-value stays within  $\pm 4.5$  in many test cases. Comparing the results for each type of adder shows that the series-connected masked adder on the left in both Figures 17 and 18 tends to have a smaller T-value overall than the compression-based masked adder, indicating that it is more secure against power analysis attacks.

These results show that series-connected masked adders using 64 bits or more random numbers as masks are more attack resistant than non-masked adders. However, in the case of 64-bit and 96-bit masks, the T-value still exceeds  $\pm 4.5$  in some cases and is not completely safe. Therefore, when the highest priority is placed on safety, it is preferable to use the series-connected masked adder that uses 128 bits of random numbers and is completely masked. On the other hand, if costs in terms of power consumption and number of resources are more important than security, we use the series-connected masked adder with a smaller random number such as 96 bits or 64 bits used for masking. By doing this, we can use circuits that are smaller area and lower power than fully masked adders, and more secure than non-masked adders.

Table 8. The number of T-values violating a security criterion ( $>|\pm 4.5|$ )

Masks	Adders	No. of T-values over $ \pm 4.5 $	Maximum T-value
0-bit	Carry propagate adder	23	12.10
(Non-masked)			
Lower 32-bit	Series-connected masked adder	32	12.26
	Compression-based masked adder	44	19.04
Dist. 32-bit	Series-connected masked adder	<b>28</b>	<b>10.98</b>
	Compression-based masked adder	32	11.83
Lower 64-bit	Series-connected masked adder	<b>12</b>	<b>8.23</b>
	Compression-based masked adder	26	12.12
Dist. 64-bit	Series-connected masked adder	16	8.52
	Compression-based masked adder	27	10.92
Lower 96-bit	Series-connected masked adder	2	5.22
	Compression-based masked adder	18	8.11
Dist. 96-bit	Series-connected masked adder	<b>1</b>	<b>4.62</b>
	Compression-based masked adder	24	10.59
128-bit	Series-connected masked adder	<b>0</b>	<b>3.07</b>
(Fully masked)	Compression-based masked adder	8	6.35

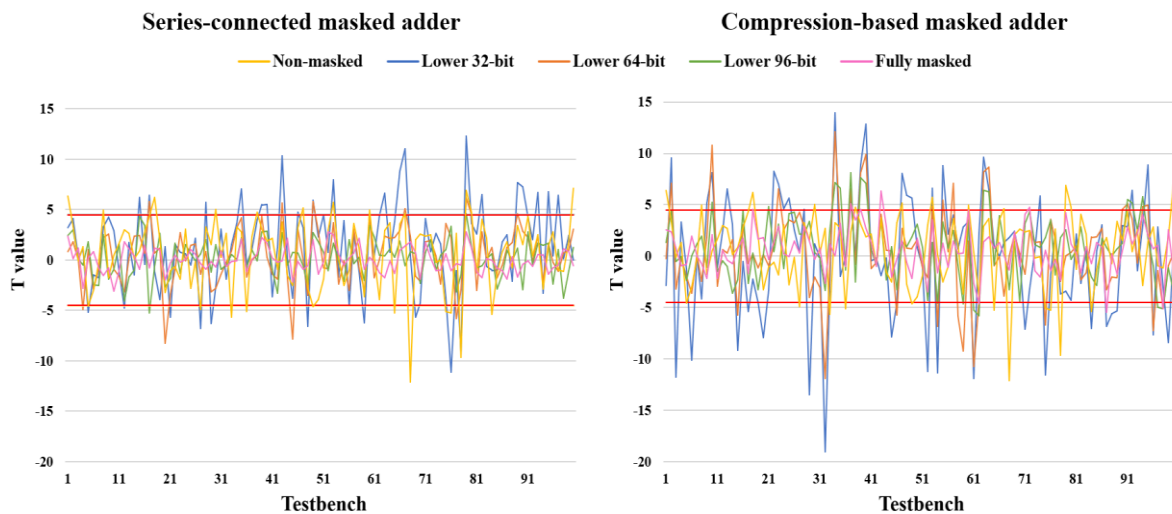


Figure 17. T-value of lower-bit masked adders

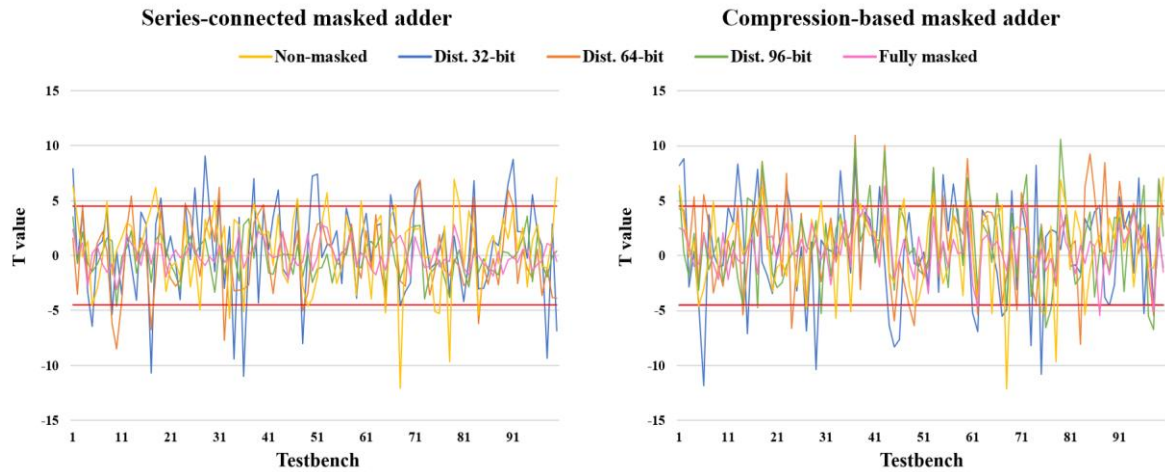


Figure 18. T-value of distributed-bit masked adders

## 5. CONCLUSION

This paper evaluates the resistance of masked adders against power analysis attacks and investigates the effectiveness of masking. The results show that circuits without masking are vulnerable to attacks even when the number of bits in the adder is small. And it was found that for masked adders, the greater the number of bits in the mask, the better the attack. Comparison of performance showed that the series-connected masked adder has a smaller circuit area and the compression-based masked adder is superior in terms of delay time. In terms of safety, the series-connected masked adder was found to be safer than the compression-based masked adder. Overall, the higher the attack resistance, the higher the power consumption. In particular, the circuit area also increases as the number of bits increases for the series-connected masked adder. Therefore, it is necessary to consider the type of adder and masking method depending on whether safety or implementation cost is a priority. Future work will include a more detailed analysis of the relationship between the internal structure of the masked circuit and its resistance to power analysis attacks. We also plan to evaluate not only the adders introduced in this paper, but also other types of circuits in the same way.

## ACKNOWLEDGEMENTS

This work is supported partly by KAKENHI 20H00590, 20K23333, 21K19776 and 22K21276.

## REFERENCES





- [1] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal, and B. Sikdar, "A survey on IoT security: application areas, security threats, and solution architectures," *IEEE Access*, vol. 7, pp. 82721–82743, 2019, doi: 10.1109/ACCESS.2019.2924045.
- [2] F.-X. Standaert, "Introduction to side-channel attacks," in *Secure integrated circuits and systems*, Boston, MA: Springer, 2010, pp. 27–42.
- [3] F.-X. Standaert, L. V. O. T. Oldenzeel, D. Samyde, and J.-J. Quisquater, "Power analysis of FPGAs: How practical is the attack?," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2778, 2003, pp. 701–710.
- [4] J. F. Dhem, F. Koeune, P. A. Leroux, P. Mestré, J. J. Quisquater, and J. L. Willems, "A practical implementation of the timing attack," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 1820, pp. 167–182, 2000, doi: 10.1007/10721064\_15.
- [5] K. Gandolfi, C. Mourtél, and F. Olivier, "Electromagnetic analysis: concrete results," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2162, 2001, pp. 251–261, doi: 10.1007/3-540-44709-1\_21.
- [6] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Annual International Conference Cryptology*, Berlin/Heidelberg: Springer-Verlag, 2012, pp. 338–397.
- [7] S. Moini, S. Tian, D. Holcomb, J. Szefer, and R. Tessier, "Remote power side-channel attacks on BNN accelerators in FPGAs," *Proceedings - Design, Automation and Test in Europe, DATE*, vol. 2021-February, pp. 1639–1644, 2021, doi: 10.23919/DATE51398.2021.9473915.
- [8] M. Zhao and G. E. Suh, "FPGA-based remote power side-channel attacks," *Proceedings - IEEE Symposium on Security and Privacy*, vol. 2018-May, pp. 229–244, 2018, doi: 10.1109/SP.2018.00049.
- [9] F. Schellenberg, D. R. E. Gnad, A. Moradi, and M. B. Tahoori, "An inside job: remote power analysis attacks on FPGAs," *IEEE Design and Test*, vol. 38, no. 3, pp. 58–66, 2021, doi: 10.1109/MDAT.2021.3063306.






- [10] M. C. Martínez-Rodríguez, I. M. Delgado-Lozano, and B. B. Brumley, "SoK: remote power analysis," in *The 16th International Conference on Availability, Reliability and Security*, Aug. 2021, pp. 1–12, doi: 10.1145/3465481.3465773.
- [11] D. Kwon, H. Kim, and S. Hong, "Improving non-profiled side-channel attacks using autoencoder based preprocessing," *Eprint* 2020-396, pp. 1–26, 2020, [Online]. Available: <https://eprint.iacr.org/2020/396>.
- [12] P. Wang *et al.*, "Enhancing the Performance of practical profiling side-channel attacks using conditional generative adversarial networks," *arxiv preprints*, Jul. 2020, [Online]. Available: <http://arxiv.org/abs/2007.05285>.
- [13] S. B. Örs, E. Oswald, and B. Preneel, "Power-analysis attacks on an FPGA - First experimental results," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2779, 2003, pp. 35–50.
- [14] L. D. Meyer, O. Reparaz, and B. Bilgin, "Multiplicative masking for aes in hardware," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2018, no. 3, pp. 431–468, 2018, doi: 10.13154/tches.v2018.i3.431-468.
- [15] J. D. Golić and C. Tymen, "Multiplicative masking and power analysis of AES," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2523, 2003, pp. 198–212, doi: 10.1007/3-540-36400-5\_16.
- [16] E. Trichina, "Combinational logic design for aes subbyte transformation on masked data," *IACR Cryptology ePrint Archive*, pp. 1–13, 2003.
- [17] M. Masoumi, P. Habibi, and M. Jadidi, "Efficient implementation of masked AES on side-channel attack standard evaluation board," in *2015 International Conference on Information Society (i-Society)*, Nov. 2015, pp. 151–156, doi: 10.1109/i-Society.2015.7366878.
- [18] J. Gravelier, J.-M. Dutertre, Y. Teglia, P. L. Moundi, and F. Olivier, "Remote side-channel attacks on heterogeneous SoC," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11833 LNCS, 2020, pp. 109–125.
- [19] L. Wu and S. Picck, "Remove some noise: on pre-processing of side-channel measurements with autoencoders," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2020, no. 4, pp. 389–415, 2020, doi: 10.13154/tches.v2020.i4.389-415.
- [20] Y. Zhao, Q. Zhang, H. Nishikawa, X. Kong, and H. Tomiyama, "Power side-channel analysis for different Adders on FPGA," in *2021 18th International SoC Design Conference (ISOCC)*, Oct. 2021, pp. 367–368, doi: 10.1109/ISOCC53507.2021.9613957.
- [21] M. Taouil, A. Aljuffri, and S. Hamdioui, "power side channel attacks: where are we standing?," in *2021 16th International Conference on Design & Technology of Integrated Systems in Nanoscale Era (DTIS)*, Jun. 2021, pp. 1–6, doi: 10.1109/DTIS53253.2021.9505075.
- [22] M. Masoumi, "Novel hybrid CMOS/Memristor implementation of the AES algorithm robust against differential power analysis attack," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 7, pp. 1314–1318, Jul. 2020, doi: 10.1109/TCSII.2019.2932337.
- [23] S. Knowles, "A family of adders," in *Proceedings 15th IEEE Symposium on Computer Arithmetic. ARITH-15 2001*, 2001, pp. 277–281, doi: 10.1109/ARITH.2001.930129.
- [24] A. Tyagi, "A reduced-area scheme for carry-select adders," *IEEE Transactions on Computers*, vol. 42, no. 10, pp. 1163–1170, 1993, doi: 10.1109/12.257703.
- [25] Xilinx Inc., "Vivado design suite 7 series FPGA and Zynq-7000 SoC libraries guide," *Xilinx Technical Documentation*, 2020, [Online]. Available: [https://www.xilinx.com/support/documentation/sw\\_manuals/xilinx2020\\_2/ug953-vivado-7series-libraries.pdf](https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_2/ug953-vivado-7series-libraries.pdf).
- [26] R. Uma, "Area, delay and power comparison of adder topologies," *International Journal of VLSI Design & Communication Systems*, vol. 3, no. 1, pp. 153–168, Feb. 2012, doi: 10.5121/vlsic.2012.3113.
- [27] Xilinx Inc., "Vivado design suite 7 series FPGAs configurable logic block," *Xilinx Technical Documentation*, p. 43, 2016.
- [28] Q. Zhang, X. Kong, and H. Tomiyama, "A toolkit for power behavior analysis of HLS-designed FPGA circuits," *Low-Power and High-Speed Chips and Systems (COOL Chips)*, 2021.
- [29] G. Goodwill, B. Jun, J. Jaffe, and P. Rohatgi, "A testing methodology for side-channel resistance validation," *NIST non-invasive attack testing workshop*, vol. 7, 2011.
- [30] T. Schneider and A. Moradi, "Leakage assessment methodology," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9293, 2015, pp. 495–513.
- [31] G. Becker, J. Cooper, E. De Mulder, G. Goodwill, J. Jaffe, and G. Kenworthy, "Test vector leakage assessment (TVLA) derived test requirements (DTR) with AES," 2015.
- [32] R. Preet, P. Singh, P. Kumar, and B. Singh, "Performance analysis of 32-bit array multiplier with a carry save adder and with a carry-look-ahead adder," *International Journal*, vol. 2, no. 6, pp. 83–86, 2009.
- [33] M. SaiKumar and P. Samundiswary, "Design and performance analysis of various adders using verilog," *International journal of computer science and mobile computing*, vol. 2, no. 9, pp. 128–138, 2013.

## BIOGRAPHIES OF AUTHORS






**Yilin Zhao**     received her BE degree in electronic and computer engineering from Ritsumeikan University in 2021. She is in the Master's degree program at Ritsumeikan University. Her research interests include design methodologies for embedded systems. She can be contacted at email: [irin.cho@tomiyama-lab.org](mailto:irin.cho@tomiyama-lab.org).






**Hiroki Nishikawa**    received his BE, ME and Ph.D. degrees from Ritsumeikan University in 2018, 2020, and 2022, respectively. In 2022, he joined the Graduate School of Information Science and Technology, Osaka University as an assistant professor. His research interests include system-level design methodologies, design methodologies for cyber-physical systems, and so on. He is a member of IEEE, IEICE, and IPSJ. He can be contacted at email: [nishikawa.hiroki@ist.osaka-u.ac.jp](mailto:nishikawa.hiroki@ist.osaka-u.ac.jp).



**Xiangbo Kong**    received B.E. degree from Nankai University in 2012 and he received M.E. and Ph.D degrees from Ritsumeikan University in 2018 and 2020, respectively. In 2020, he joined the College of Science and Engineering, Ritsumeikan University as an assistant professor. His research interests include artificial intelligence, image processing, embedded system, etc. He is a member of IEEE and IPSJ. He can be contacted at email: [kong@fc.ritsumei.ac.jp](mailto:kong@fc.ritsumei.ac.jp).



**Hiroyuki Tomiyama**    received his BE, ME, and DE degrees in computer science from Kyushu University in 1994, 1996, and 1999, respectively. He worked as a visiting researcher at UC Irvine, as a researcher at ISIT/Kyushu, and as an associate professor at Nagoya University. Since 2010, he has been a full professor with the College of Science and Engineering, Ritsumeikan University. He has served on program and organizing committees for several premier conferences including DAC, ICCAD, DATE, ASP-DAC, CODES+ISSS, CASES, ISLPED, RTCSA, FPL, and MPSoC. He has also served as an editor-in-chief for IPSJ TSLDM; an associate editor for ACM TODAES, IEEE ESL, and Springer DAEM; and a chair for the IEEE CS Kansai Chapter and IEEE CEDA Japan Chapter. His research interests include, but are not limited to, design methodologies for embedded and cyber-physical systems. He can be contacted at email: [ht@fc.ritsumei.ac.jp](mailto:ht@fc.ritsumei.ac.jp).