PAPER

A Power Analysis Attack Countermeasure Based on Random Data Path Execution For CGRA

Wei GE[†a)], *Member*, Shenghua CHEN[†], Benyu LIU[†], Min ZHU[††], *and* Bo LIU[†], *Nonmembers*

SUMMARY Side-channel Attack, such as simple power analysis and differential power analysis (DPA), is an efficient method to gather the key, which challenges the security of crypto chips. Side-channel Attack logs the power trace of the crypto chip and speculates the key by statistical analysis. To reduce the threat of power analysis attack, an innovative method based on random execution and register randomization is proposed in this paper. In order to enhance ability against DPA, the method disorders the correspondence between power trace and operands by scrambling the data execution sequence randomly and dynamically and randomize the data operation path to randomize the registers that store intermediate data. Experiments and verification are done on the Sakura-G FPGA platform. The results show that the key is not revealed after even 2 million power traces by adopting the proposed method and only 7.23% slices overhead and 3.4% throughput rate cost is introduced. Compared to unprotected chip, it increases more than 4000× measure to disclosure.

key words: *side-channel attack, reconfigurable architecture, differential power analysis, power trace*

1. Introduction

With the rapid development of Internet, information and electronic technology, the cryptographic technique with high throughput and high energy efficiency has become an urgent issue. In network communication applications, due to the use of numerous security protocols (IPsec, TLS/ SSL, WTLS, SSH, S/ MIME and OpenPGP) and the actual cryptographic algorithms determined through negotiation, the implementation of cryptographic algorithms needs to meet flexible requirements. By analyzing the existing block cipher algorithms [1]–[3], it can be known that:(1) the cipher design is realized based on Feistel or SP network structure; (2) the algorithm structure of encryption and decryption is very neat, and the process is similar (including transformation of prologue, intermediate and epilogue); (3) not only in the calculation of each round function, but also in the calculation between the round functions, there is a data correlation of Read-after-Write(RAW); (4) the encryption and decryption process has a one-way, open-loop computing relationship. There are few conditional judgment branches and no data-dependent feedback relationship between computation operations; (5) the basic operation of cryptographic algorithm can be divided into four categories according to the frequency of using the process: basic arithmetic calculation operation, logic and shift operations, replacement operation and lookup table operations.

By analyzing the basic structural and computational characteristics of the block cipher algorithm, it can be found that the cipher algorithm has the potential to be realized by the hardware with configurable parameters. For example, similar calculation modules are used in hardware design to realize the operation of different round functions. Taking advantage of data RAW features, the design can adopt concurrent and pipelining techniques. At the same time, hardware operators that meet common characteristics of basic operations are designed, such as ALU with comprehensive operation types and S-box that satisfy various algorithms.

In practice, various types of cryptographic algorithms need to be combined to meet the information security requirements of confidentiality, security and availability. A coarse-grained reconfigurable cryptographic processor can achieve high throughput encryption and decryption calculations, and is a security solution to meet the needs of ever-changing cryptographic algorithms and high-performance algorithms. However, the security of cryptographic algorithms is related not only to the algorithms but also to the implementation of the algorithms.

Side-channel analysis attack such as differential power attack can gather the key at a small cost without breaking the chip physically and have high attack efficiency [4], [5]. The principle of power analysis attack is based on the different power consumption feature of CMOS under three cases: the output converts from 0 to 1, from 1 to 1 and maintains. This causes the fact that power consumption of cryptographic devices depends on the intermediate values of the cryptographic algorithms executing. While a large number of operands and power traces are acquired, the attackers can speculate the key by calculating correlation based on statistical analysis.

One of the main countermeasures is breaking the dependence between intermediate values and power consumption. Two main methods are hiding and masking. Masking achieves in algorithm level, which randomizes the intermediate values that are processed by the cryptographic devices. The masking method works quite well for specific hardware [6]. However, since masking implementation is closely related to the algorithms and needs a lot of additional operations, it is usually used for chips which execute only one specific algorithm. Besides, it also introduces a larger area overhead [7]–[10]. On the contrary, the hiding

Manuscript received November 20, 2019.
Manuscript publicized February 10, 2020.

[†] The authors are with National ASIC System Engineering Research Center, Southeast University, Nanjing, China.

[††] The author is with Wuxi Research Institute of Applied Technologies, Tsinghua University, Wuxi, China.

a) E-mail: duiker@seu.edu.cn

DOI: 10.1587/transinf.2019EDP7308

method enhances the security by breaking the link between intermediate values and power consumption. As the crypto chip is designed to process kinds of algorithms with flexibility, our work focuses on hiding method.

2. Related Work

A successful DPA attack relies on acquiring the power consumption of the time point at which the specified intermediate value data is under processing. Thus, the attacks usually need to make sure that the power trace and data was aligned. In order to complicate data alignment, many methods are proposed, such as randomized delays and dummy cycles.

Reference [11] randomizes data processing by using irregular clock cycle delays and multi-phase shifting obtained from digital clock managers (DCM) in FPGAs. The countermeasure makes use of a set of DCM, which provides different output clocks with each phased-shifted by fixed degrees. When the clock number (CN) equals 8, the minimum phase difference between two clocks is 45 degree. A clock multiplexer was used to choose output clock randomly, and it outputs a cipher clock which has a random phase in different cycles. However, in order to make sure that the crypto core meets the timing constraint, the max synthesis frequency will be CN times of the original frequency, significantly increasing the design difficulties. As a result, the experiment shows that it takes 1.77x more time to process the AES when applied with this countermeasure, compared to original implementation. Also, the throughput performance decrease by 73.5%.

Another way to hiding power leakage is adding jitter to the clock. Reference [12] uses jitter clock to randomize data processing time. The hardware generates several phase-shifted clocks (assume SP) by inserting a delay between two consecutive ones, and it uses M clock multiplexers, which selects a different shifted clock per cycle individually. The random input value, RND, is used as the select signal for multiplexers. Since the delay inserted between each consecutive clocks is δ, the maximum delay of the random clock is $SP * \delta$ compared to the input clock. Thus, the clock uncertainty needs to be $SP * \delta$ in synthesis to ensure the correct functionality of the random clocks, and it will lower the max frequency that the design can reach. The experiment on Virtex-5 FPGA platform shows that the minimum clock cycle increases by 127% when SP is set to 16 and δ is set to 0.33ns, which means the throughput decreases by 55.95%. Meanwhile, the hardware area increases by 10% because of the extra delay circuit and multiplexers.

Random dummy operations insertion was proposed in Reference [13] to enhance the security of SRCP. The hardware randomly inserts some dummy operations during the beginning and the end of the encryption, with a fixed execution time to misalign the power data. The insertion of dummy operations will slow down the throughput of the hardware. To avoid attacks eliminating traces without the same execution time, the fixed number of FN dummy operations are used. When FN is set to 5, the AES used 10.6% extra cycles while DES used 5.7% extra cycle to finish.

Reference [14], [15] uses Wave Dynamic Differential Logic(WDDL) as DPA countermeasure. The input and output signals of WDDL are complementary with dynamic logic, which makes sure that the power consumption remains constant. Thus, the WDDL has good effect against power attack. However, the difference between each path delay in WDDL may lead to power leakage. So it still has the risk of being attacked. Due to the complementary signal, the number of transistors required for WDDL is two times more than normal logic, which leads to the massive area overhead. Reference [16]–[18] proposes a method namely dynamic reconfiguration. It dynamically changes the sequences and place of algorithm execution to disturb power consumption. However, changing the sequences with right functionality has significant adverse impacts on timing, which will reduce the throughput of hardware. Meanwhile, the hardware is more complex, which increases the difficulty of implementation.

Reference [19] applied a method of changing the storage position of the intermediate value in the register every round to defend against the Hamming distance model. Taking the DES algorithm as an example, the intermediate values of the 15th and 16th rounds of the DES algorithm are stored in different registers, so that the attacker cannot establish the Hamming distance model. However, it still has obvious shortcomings. First, the register is changed to double the register space requirement of the encryption algorithm. Secondly, since the register used in each round of the DES operation is still fixed, the method can only be applied to the operations without the depth of the pipelining. For a device with a pipeline depth, the attacker can still use the two adjacent ciphertexts to calculate the corresponding 15th round of the operation result and establish a Hamming distance model for the 15th round of the operation result storage register, thereby cracking secret key.

In summary, the existing power analysis countermeasures based on misalignment and hiding have many impacts on timing, resulting in the serious cost of throughput. Random clock switching is unfriendly to the EDA tools, which increases the difficulties of design and development. Random dummy operation insertion works in the synchronizer circuit. However, the measure to disclosure (MTD) depends on the cycle of dummy operation inserted. Increasing the MTD needs more dummy cycles in each encryption, which also introduces a serious cost of throughput. The method of changing the storage position of the intermediate value in the register stores the intermediate value in a different register so that the attacker cannot establish the Hamming distance model. However, the required register space requirements are doubled and can only be applied to operations without pipeline depth.

The traditional shuffling is to disrupt the independent operation of AES [20], such as Random Permutation (RP) and Random Start Index (RSI) can change the order of S-boxes. The two types of shuffling mentioned in [21] are also implemented in the implementation calculation process of

AES. However, if different encryption algorithms are used, corresponding changes are needed. The method proposed in this paper combines the reconfigurable architecture used, without changing the hardware implementation structure of algorithms such as AES and DES. Combined with the characteristics of the reconfigurable structure, only randomizes the writing and reading of registers, and has more high universality and simplicity.

In order to meet the design requirements of low resource overhead and low throughput overhead, and applicability to multiple block cipher algorithms, this paper proposes a hiding method against DPA based on random execution and register randomization. By randomizing the execution sequence, the hardware can hide the correspondence between the processing data and the power trace and reduce the correlation between power data and intermediate data. Besides, by using redundant registers and interconnect resources in the reconfigurable architecture, the data operation path is randomized, so that the registers that store intermediate data are randomized, which fundamentally hinders the establishment of the Hamming distance model, thereby realizing effective defense against power analysis.

This article is based on the previous work [22] for further research and implementation. Based on the research content of “method based on random execution”, the research content of “The Register Data Random Module (in Sect. 4.2)” has been added, and more design implementation work has been added. Further experiments compared the effects of the “Line Random execution”, “Register Random Execution” and “Combination of Two Methods” methods against differential power analysis.

The contributions of this paper are mainly as follows:

1. Analyzed the principle of data random path countermeasure against DPA.

2. Proposed a random Data Path Execution (ARDPE) anti-power analysis attack method for CGRA.

3. Implemented ARDPE based on FPGA and comparative analyzed the resist DPA effect of this method on CGRA.

3. Motivation

The success of a DPA attack relies on the correlation between a large amount of input data and power trace information. Although the input or output data is difficult to hide, once the execution order and the transmission path in the data processing are hidden, it will significantly prevent the attacker from acquiring the actual power trace information corresponding to the specific data. This not only improves the security of the encryption chip, but also ensures the transparency of the chip to the user.

It is assumed that the plaintext to be processed is divided into groups according to every N data, wherein N is an out-of-order packet size. Assuming the input plaintext sequence X output to the encryption module in a random order, then the probability Y of plaintext sequence output order is as shown in Eq. (1). When the protected hardware was attacked, the attacker can only guess the data order, and the correct probability of which is $1/N$.

$$Px(Y) = 1/N(X < N, Y < N) \tag{1}$$

The security performance of hardware against DPA can be measured by the number of power traces that are needed to mount a successful DPA attack, which is denoted with MTD(measure to disclosed). As a rule of thumb, it can be calculated as follows [23].

$$MTD = 3 + 8 * Z_a^2 / \left(ln^2 \frac{1 + p_{ck,ct}}{1 - p_{ck,ct}} \right) \tag{2}$$

$$\lim_{p_{ck,ct}^2 \to 0} ln \left(\frac{1 + p_{ck,ct}}{1 - p_{ck,ct}} \right) = 2 * \rho_m ax + O\left(p_{ck,ct}^2\right) \tag{3}$$

Where Z_a is the quantile of the standard normal distribution, which satisfies $p(Z < Z_a) = a$. When the confidence is taken to 0.9999, it can be seen from the table that Z_a is equal to 3.7190.

The recorded traces depend on these intermediate values at the specific position which denoted as ct. $P_{ck,ct}$ represents the correlation curve corresponding to the correct key ck at the time point ct, where ct is the time point at which the power consumption has the greatest correlation with the intermediate value. When $P_{ck,ct}$ is small, the formula (3) holds, so the formula (2) can simplify to the formula (4).

$$MTD \approx 28 / \left(p_{ck,ct}^2\right) \tag{4}$$

According to the formula above, the MTD is inversely proportional to the square of the correlation between intermediate data and power consumption. Therefore, to obtain the relationship between N and MTD, it is necessary to derive the relationship between N and $P_{ck,ct}$. During the process of attacking, the data corresponding to the power trace can only be guessed randomly. Moreover, the correct possibility is $P = 1/N$. Thus, it can be assumed that only $1/N$ of the links between power traces and intermediate data is correct in the case of a large amount of statistical data. For those who have the wrong links, it can be considered that there is no correlation. In summary, only M/N of power traces is valid in the M power traces measured in the attack, and the others can be considered as noise.

Suppose there are two sets of arrays, x and y generated following the formula below:

$$y = x * 0.1 + rand * 0.9 \tag{5}$$

Moreover, the correlation coefficient between the two sets of arrays is approximately 0.1 which approach the correlation coefficient between intermediate data and power traces. The coefficient rand is the random noise, the average value of which equals to array x. Another two sets of arrays x_1 and y_1 contain M pairs of data. Some of them are copied from x and y, which are denoted with R. And other was just random numbers. To surely make the result more accurate, the average value of random numbers in x_1

and y_1 equals that in x and y. It can be seen that the ratio of valid numbers R/M and the correlation coefficient p are an obviously linear relationship. As the ratio of valid numbers R/M increases from 0 to 1, the correlation coefficient p will also increase from linearly about 0 to close 0.1 that the coefficient of x.

Thus, the relationship between MTD and N can be described as:

$$\begin{cases} MTD \sim 1/\left(P_{ck,ct}^2\right) \\ P_{ck,ct}^2 \sim 1/N \\ MTD \sim N^2 \end{cases} \tag{6}$$

Theoretically, MTD is proportional to the square of N. Based on data path grouping, we propose two schemes, random execution and register randomization.

Based on this consideration, we proposed A Random Data Path Execution (ARDPE) anti-power analysis attack method for CGRA. As shown in Fig. 1, the method consists of two parts: line data randomization and register data randomization. The line data random module is located outside the cryptographic core. After randomizing the encryption sequence, the plaintext is divided into M parts, and then the order is restored after encryption. The register data random module is located inside the encryption. The register data random module is located in the cryptographic core. By using the redundant register resources of the reconfigurable architecture, intermediate data correspondence between the round and round process can be randomly performed using additional randomization and recovery operations. Following, the ARDPE method will be described in detail in two parts.

4. Implementation

4.1 The Line Random Module

As shown in Fig. 1, a line data random module against DPA attacks based on dynamic address scrambling is implemented, which includes input and output line data buffers, the random address generator and restorer. The line data buffer can be considered to consist of several small unit buffers, each requiring $M * 128$ bits. Therefore, the total size of input or output buffer can be derived as $N * M * 128$ bits, where N is the number of units in the buffer and M is the size of each the unit buffer. In order to support parallel read and write operations at the same time, a data buffer needs to have at least two units, which means at least $N \geq 2$. The random address generator obtains the output of the random number generator(RNG) and generates a random sequence. By scrambling input buffer reading addresses and output buffer writing addresses, the random address generator makes sure that the output buffer can recover the correct data sequence.

The random address generator mainly has two functions. First of them is provided by a sequence converter which converts random number sequences to N non-repetitive randomized numbers between 0 to $N-1$ as scrambling addresses. The input buffer output data in address $Base + Bias$, where $Base$ is the base address of group and $Bias$ is the number from random address generator. The second function is to make sure that the output buffer can recover the result right sequences. The converter output will be preserved by a FIFO, and output to the buffer after X cycles which denote as Bias. X is the cycle latency of crypto core. After that, the right sequence of result can be recovered by writing data into the address ($Base + Bias$).

The converter uses classical poker algorithm. Assuming that there are N different pokers, each time one poker is removed randomly, this will make the sequence of the removed pokers a non-repetitive randomized sequence. In order to keep the randomness of sequence, the ID of the poker removed each time should be $R\%A$, where R stands for a random number from RNG and A stands for the left pokers currently. In our work, a memory is used for N pokers, and the number of left pokers is counted by a subtractor, which denoted as A. Firstly, the memory will be initialized to make sure it contains N non-repetitive number from 0 to $(N-1)$ at address space between 0 and $(N - 1)$. Moreover, the output of the module is the data in address R%A. To maintain the functionality of non-repetitive output, the data between address 0 and address $(A - 1)$ should be different. So the data in the address $(A-1)$ should be written into address $R\%A$ in each output cycle, unless $(A - 1)$ equals to $R\%A$.

As shown in Fig. 2(a), the memory module should have two output ports and one write port to support output in each cycle. However, the memory must be re-initialized

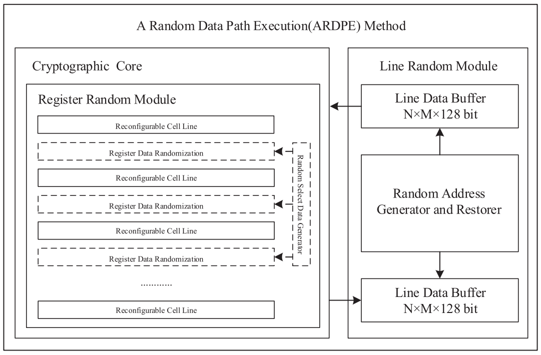
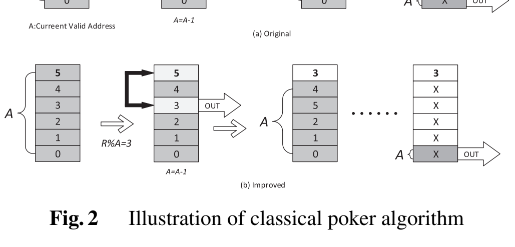


Fig. 1 Module schematic of ARDPE for CGRA



(a) Original

(b) Improved

Fig. 2 Illustration of classical poker algorithm

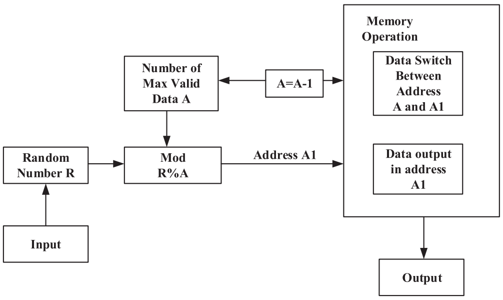

Memory Operation
Data Switch Between Address A and A1
Number of Max Valid Data A
A=A-1
Random Number R
Mod R%A
Address A1
Data output in address A1
Input
Output

Fig. 3 The implementation of the converter

every N cycles, in order to maintain the differences of the data between address 0 to address $(N-1)$. Also, the design can be improved furtherly, as shown in Fig. 2(b). To remove the re-initialize operation, another write port supporting data switching in one cycle is added to the memory module. In this case, the data in the memory can be kept different any time. Therefore, the re-initialize operation is no longer needed, which improves the efficiency of the converter.

The physical design of the converter is shown in Fig. 3, which contains a memory achieved by register file with dual write ports and dual read ports, a subtractor and a module for the modulus operator. In the beginning, the maximum valid address is N, which can be described as $A = N$. The data in memory is equal to its address. In each round, address A1 is calculated as $A1 = R\%A$. Moreover, the output is the number in address $A1$. The switching operation will be completed in the same cycle. The number of valid data, known as A, will be subtracted by 1. Then, a whole round is finished. Although some cycles are used in each round, the pipeline design makes sure the converter can output in each cycle.

4.2 The Register Data Random Module

As shown in Fig. 4, The register data random module is essentially a multiple-input and multiple-output switching network controlled by random numbers, which guarantees that the order of input and output is consistent, but the order between temporarily storing data and input data is random. The register data random module consists of three parts: a randomization unit, a temporary data storage unit, and a recovery unit. The functions of the randomization unit and the recovery unit are complementary to each other. The redundant register resources are used to randomize the data order in the execution process so that the attacker cannot establish the correspondence between the operand data and register location, which can effectively prevent the power consumption attack based on the Hamming distance model.

In our reconfigurable array, a 4-input and 4-output switching network meets most of the algorithm requirements. The 4-to-4 switching network can be combined by a 2-to-2 switching network. In order to restore the output data in the initial order, the data path should be the reverse order from the randomization unit. Therefore, the configuration data of the recovery unit should be the same as the randomization unit. Therefore, only a 6-bit random number is required to completely the register data random function between every two reconfigurable cell lines.

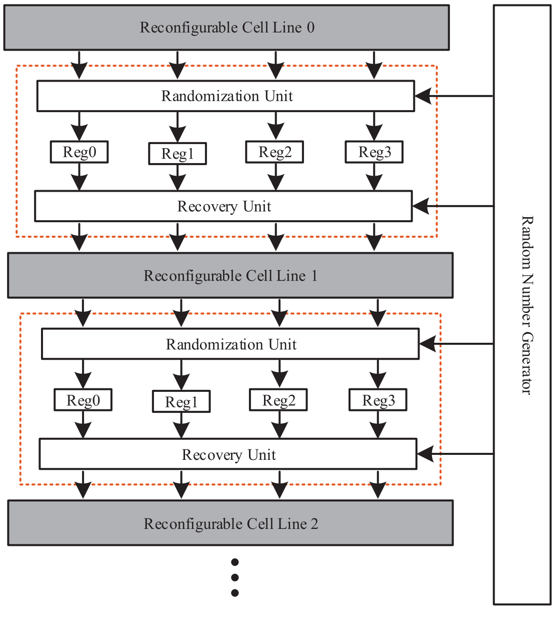


Fig. 4 The illustrate of the register data random module of ARDPE

In order to reduce the impact of register randomization on array throughput, it is necessary to improve the critical path of the data to reduce the performance impact. To meet the flexibility of the input data source for each reconfigurable cell line, there is a 4-to-1 data selector between the output and the input of every two reconfigurable cell line units. In our original idea, the register recovery module is connected in series with the 4-to-1 data selector. Therefore, the 4-to-1 data selector becomes a 4-to-4 switching network, while implementing data selection and register random functions. In the process of register data randomization, a 3-level interchange unit is used for a 4-to-4 switching network. The 3-level interchange unit allows 4 inputs to be converted into 24 output forms with 64 different data paths.

However, in a real power consumption attack, only one register is attacked. Randomizing the data path only needs to ensure that each input can be output to any one of the outputs. The probability that the two intermediate values required for the Hamming distance are stored in the same register is 1/4.

Therefore, the level 3 switching network unit can be reduced to the level 2 switching network unit. The data path of Fig. 5 is optimized, as shown in Fig. 6. Compared with the original design, the critical path delay of the optimized register randomization module is reduced from 3 Td_{su} to 2 Td_{su} (Timing-delay from D to Q through Switch units), which is reduced to 66% of the original delay. The number of switching units of the multi-input and multi-output switching network is reduced from 12 to 8, and the area re-

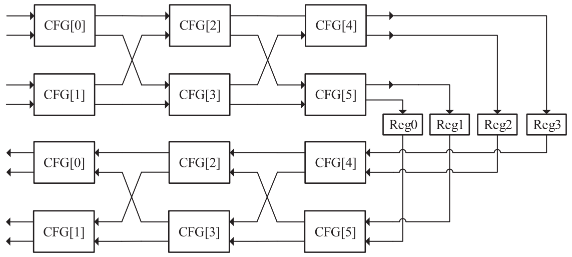


Fig. 5 Register randomization and recovery

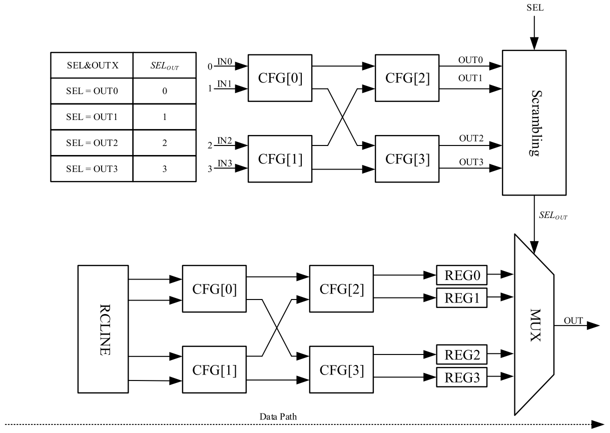


Fig. 6 Register randomization and recovery path

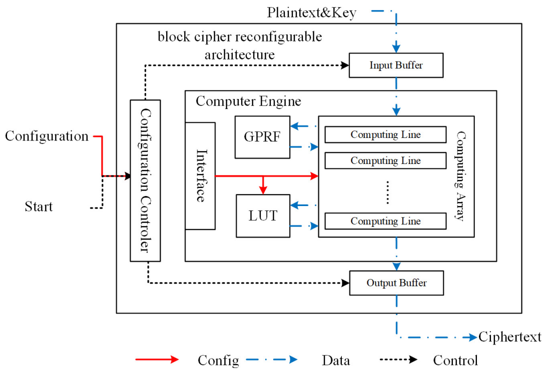


Fig. 7 Block cipher reconfigurable architecture

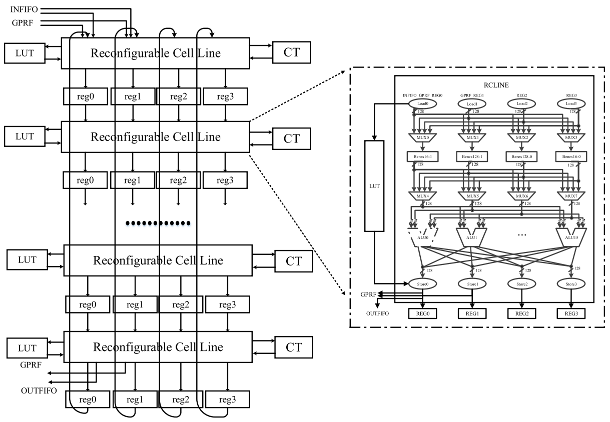


Fig. 8 Data flow of the algorithm mapping on the reconfigurable array

source of the switching unit is reduced to 66% of the original design. Especially it should be noted that, the MUX reuse of the original input selection module in the Reconfigurable Cell Line, without adding new logic resources. The random number bit width is reduced from 3bit to 2bit, which is reduced by 33% per cycle.

5. Experimental Results

5.1 Introduction of BCRA

The experiments are conducted based on a block cipher reconfigurable architecture (BCRA) to verify the ARDPE anti-power analysis attack method. The BCRA is as shown as Fig. 7. It consists of a calculation engine and a configuration controller. The calculation engine consists of a Reconfigurable Computing Array (RCA), a General-Purpose Register File (GPRF), a Reconfigurable Look-Up Table (RLUT), and a configuration interface.Based on the configuration information of the different algorithms, the configuration controller receives the configuration context and writes it to the compute array through the configuration interface. The RCL performs the corresponding logical/mathematical operations and the interconnection is dependent on the configuration context. In addition, the RLUT is used for nonlinear calculations in algorithms, while GPRF is used to store intermediate data during processing.

As shown in Fig. 8, the basic unit of RCA is Reconfigurable Cell Line (RCL), and each line corresponds to one control unit for controlling the flow of computation data. There are 4 sets of 128-bit registers between the computing lines for data temporary storage. The RCL mainly contains LOAD and STORE units, data switching network (MUX), multiple data selector (BENES) and arithmetic logic unit (ALU). The LOAD and STORE units implement the data input and output functions of each RCL according to the configuration context. The input data of each RCL comes from the register of the previous line, input buffer or GPRF, and the operation result of each RCL can be selected to be output to the register, output buffer or GPRF. In order to improve the loop efficiency of the data stream in the cryptographic algorithm, the first computing line can also use the register output of the last computing line as input data. The MUX module is used to provide more flexible input selection for BENES and ALU modules. The BENES module can select any LOAD unit as input and achieve 128-bit arbitrary replacement operation. Each RCL includes 16 8-bit ALUs. Each ALU can select the output of any BENES as input data. The ALU implements different arithmetic operations based on configuration information, including 3-input XOR, 4-input XOR, pass-through, and GF(2^8) finite field based multiplication.

The architecture is flexible and can implement many block cipher algorithms, so security issues become more prominent. Both the security of the protected system and its

Table 1 Resource distribution of the reconfigurable cryptographic processor

Module Name	Resource(slice)	Present Rate
RCA except ARDPE	14647	89.49%
ARDPE	1184	7.23%
FIFOCtrl	232	1.42%
AHBtoBuffer	159	0.97%
UARTtoAHB	88	0.54%
Other	58	0.35%

area overhead is related to a group size of buffer and group size of the register. Larger buffer group sizes require more memory space and more area occupation. Meanwhile, the larger group size requires more address bits, which causes an additional delay in decoding the address. However, the address decoding is not in the critical path and therefore does not affect the frequency of the system. To support parallel read and write operations, the buffer should have at least two groups. Although our method has no effect on throughput, it still increases the average data delay by N cycles, and N is the size of the buffer group. Once the input data is less than N, the input data needs to be padded to in order to trigger the reading of the cryptographic core. The design parameter of the line data buffer is $N * M * 128$ bits, where design parameter $N = 32$ and $M = 2$.

In this experiment, the AES encryption algorithm is used as an example to verify the efficiency of our method. At the same time, the Hamming distance (HD) model is used to attack the experimental equipment. The reconfigurable cryptographic processor implemented on FPGA includes 4 RCLs with a clock constraint of 3MHz. The experiments based on FPGA Spartan-6 LX75 show that the total resource consumption is 16368 slices, in which the RCA are 14647 slices, LUTRAM 960Byte and BRAM 252Kbyte. The addition of ARDPE protection method increased resource consumption by 1184 slices, accounting for 7.23% of the total system resource, as shown in Table 1.

5.2 Mapping Example of AES Algorithm

The following takes the AES algorithm as an example to illustrate how the block cipher algorithm maps to the reconfigurable cryptographic processor.

The AES-128 algorithm has a total of 10 iterations, and there is an initial round key addition operation before the first round of iterations. Each round of operations includes four operations of byte replacement, row shifting, column confusion and key plus. Wherein, the byte replacement can be directly operated by the RLUT module. The row shift can be implemented by the MUX module, and the key addition can be implemented by using ALU. The most complicated part of the mapping is the column obfuscation operation, which needs to be optimized and decomposed. The simplifying column confounding operation for the first line of 4 bytes data is as shown in Eq. (7) where X2 is a finite field multiplying 2 operation on GF(2^8).

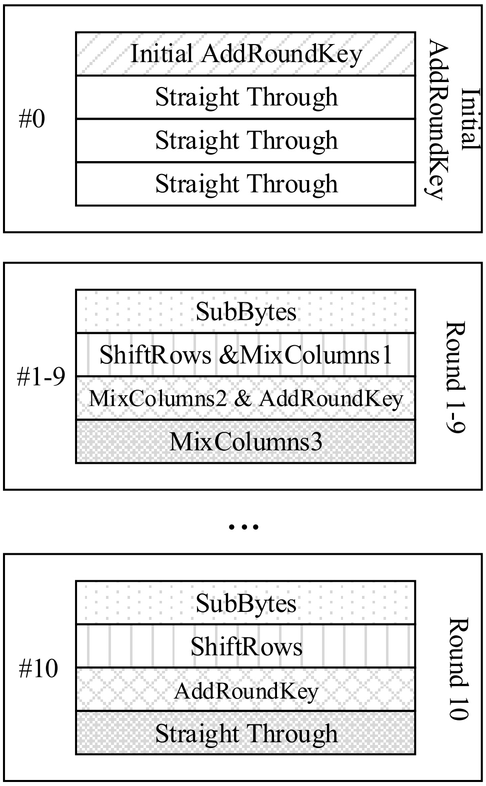


Fig. 9 The data flow diagram of the AES algorithm mapped on the reconfigurable architecture

$$\begin{aligned} S'_{0,j} &= \left(2 \bullet S_{0,j}\right) \oplus \left(3 \bullet S_{1,j}\right) \oplus S_{2,j} \oplus S_{3,j} \\ &= \left(2 \bullet S_{0,j}\right) \oplus \left((2 \oplus 1) \bullet S_{1,j}\right) \oplus S_{2,j} \oplus S_{3,j} \\ &= X2\left(S_{0,j} \oplus S_{1,j}\right) \oplus S_{1,j} \oplus S_{2,j} \oplus S_{3,j} \end{aligned} \tag{7}$$

Extending the above simplification to all 16 bytes, the final operation can be divided into three steps MC1, MC2 and MC3, as shown in Eq. (9). Where M0, M1, M2, M3 can be generated by the interchange network, as shown in Eq. (8). After the MC3 operation is completed, the result of the final column confusion operation is obtained.

$$\begin{aligned} &\mathrm{M_0{=}(S0,jS1,jS2,jS3,j)}', 0 \le \mathrm{j} \le 3 \\ &\mathrm{M_1{=}(S1,jS2,jS3,jS0,j)}', 0 \le \mathrm{j} \le 3 \\ &\mathrm{M_2{=}(S2,jS3,jS0,jS1,j)}', 0 \le \mathrm{j} \le 3 \\ &\mathrm{M_3{=}(S3,jS0,jS1,jS2,j)}', 0 \le \mathrm{j} \le 3 \end{aligned} \tag{8}$$

$$\begin{aligned} MC1 &= X2\,(M_0 \oplus M_1) \\ MC2 &= MC1 \oplus M_1 = X2\,(M_0 \oplus M_1) \oplus M_1 \\ MC3 &= MC2 \oplus M_2 \oplus M_3 \\ &= X2\,(M_0 \oplus M_1) \oplus M_1 \oplus M_2 \oplus M_3 \end{aligned} \tag{9}$$

Finally, the data flow diagram of the AES algorithm mapped on the reconfigurable architecture is shown in Fig. 9, which is divided into 11 data flow graphs. Except that the first data flow graph only performs the initial key addition operation accident, each of the other data flow graphs corresponds to the corresponding round iteration operation in the AES algorithm.

The reconfigurable cryptographic processor can support different mapping implementations of the AES algorithm by changing the data path and computational logic operations in the RCL, supporting different key lengths (128 bits, 196 bits, 256 bits). As shown in Fig. 1, the line data random module is located outside the cryptographic core. Meanwhile, the register data random module is located between the RCLs. Therefore, the ARDPE method can be applied to arbitrary algorithms mapping, supporting AES, DES, SM4 and other types of algorithms.

5.3 Experimental Analysis

The SAKURA-G development board from the SAKURA (Side Channel Attack User Reference Architecture) project is used in our experiments, which contains two FPGA boards. One is used for data and clock control and the other is for verifying our design. These boards are dedicated to side channel attacks and have dedicated ports for measuring power consumption. The LeCroy WaveRunner-MXi oscilloscope is also used to measure and record power traces. In order to highlight the signal-to-noise ratio of the power measurement, the experiment uses a lower clock frequency, which produces a very noticeable peak value per clock cycle.

A comparison of the power analysis results for the three protection methods is given in Table 2. When applying the Hamming Distance (HD) model, compared with the unprotected implementation, Random Execution improves the security of the system by 1000 times, Register Randomization by 24 times, and Combination of Two Methods increase system security by more than 4000 times. When applying the Hamming Weight (HW) model, compared with the unprotected implementation, Random Execution improves the security of the system by more than 20 times, while Register Randomization by 1.9 times, and Combination of Two Methods increase system security by more than 20 times. The Hamming Weight (HW) model reflects the correlation of the energy consumption of the circuit is weak, and its attack efficiency is generally lower than the Hamming Distance (HD) model. For unprotected implementation attacks, the HW model requires 100K energy traces, while the HD model requires only 500. Therefore, for more than 2M energy traces collected by the ARDPE implementation, the resistance improvement (4000 times) of the HD model is improved greater than the HW model (20 times).

Table 2 Comparisons between random execution, register randomization and combination of two method

Method	Model	unprotected	protected	Improvement
Line Random Execution	HD	500	500K	1000
	HW	100K	>2M	>20
Register Randomization	HD	500	12K	24
	HW	100K	190K	1.9
ARDPE	HD	500	>2M	>4000
	HW	100K	>2M	>20

The results of the unprotected and protected DPA attack for the first byte in AES are shown in Fig. 10 and Fig. 11, respectively. Specifically, the unprotected AES

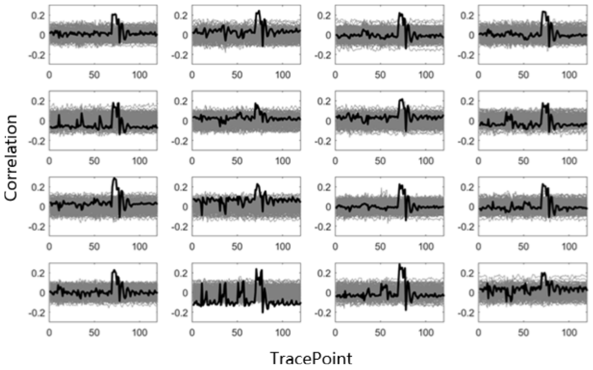


Fig. 10 Coefficient curves of AES first byte in unprotected AES (500 traces)

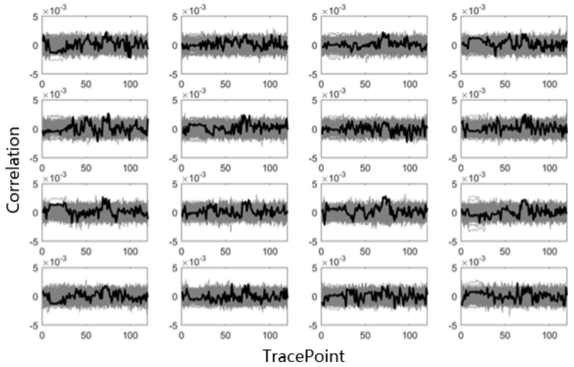


Fig. 11 Coefficient curves of AES first byte in protected AES (2 million traces)

Table 3 Comparisons between random execution, register randomization and combination of two method

Methods	Area Overhead	Throughput Rate Cost	Measure To Disclosure		
			unprotected	protected	improvement
WDDL[18]	210%	75%	N/A	>1.5M	N/A
Duplicated Complement[19]	104%	0%	N/A	>1M	N/A
Current Equalizer[20]	25%	50%	6000	>10M	>1667
SRCP[16]	Neglectable	11%	N/A	>1M	N/A
This Work	7.23%	3.4%	500(HD)	>2M	>4000
			100K(HW)	>2M	>20

refers to the architecture reconfigured without any countermeasures. In addition, the protected AES is protected by the ARDPE module. It can be seen that the highest peak at attack point discloses the right key after analyzing 500 traces, while the peak of protected AES did not identify the right key even analyzing 2 million traces.

Table 3 gives a comparison of the proposed work with other effective countermeasures. For the encryption system protected by our method, there is no indication to reveal the correct key for 2 million power traces, the system security increased by 4000 times in Hamming Distance model and by 20 times in Hamming Weight model. Compared with papers [19], [24], [25], all security levels remain the same, but the proposed method has less area overhead and throughput reduction. For the method in [26], the key is not revealed for 10 million power trace, but its significant area overhead is 25%, throughput is reduced by 50%. And our work is 7.23% and 3.4% respectively. At the same time, the proposed architecture can be reconfigured without modifying the cryptographic core hardware design, which means it has better versatility and greater flexibility, especially for high-throughput reconfigurable architectures.

6. Conclusion

In this paper, an innovative random data path execution (ARDPE) method against DPA attacks for coarse-grained reconfigurable architecture is proposed. By dividing the input data stream into groups and randomizing the data input order within the group, the correspondence between the intermediate data and the power tracking is changed, which effectively reduces the correlation. In addition, by using redundant registers and interconnect resources in the reconfigurable architecture, the data operation path and the register store location are random, which fundamentally hinders the establishment the Hamming distance model and enhances the ability against DPA attacks. The experimental results show that the method has better flexibility, the security is improved by 4000 times, with only 7.23% slice overhead and 3.4% throughput reduction. Compared to other state-of-the-art methods, the proposed method has proven to have a more positive impact on security improvements with less area overhead and throughput reduction.

Acknowledgments

We express thanks to Yang Jingjiang, Yang Jun and Liu Xinning for their useful and incisive comments. This work was supported by the National Key R&D Program of China (Grant No. 2018YFB2202102) and National Science and Technology Major Project (Grant No.2017ZX01030101).

References

[1] J. Katz, A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.

[2] A. Joseph and V. Sundaram, "Cryptography and steganography: A survey," Int. J. Comp. Tech. Appl., vol.2, no.3, pp.626–630, 2011.

[3] M. Agrawal and P. Mishra, "A comparative survey on symmetric key encryption techniques," International Journal on Computer Science and Engineering, vol.4, no.5, p.877, 2012.

[4] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," Proc. Advances in Cryptology (CRYPTO '99), ed. M. Wiener, Berlin, Heidelberg, pp.388–397, Springer Berlin Heidelberg, 1999.

[5] P. Kocher, J. Jaffe, B. Jun, and P. Rohatgi, "Introduction to differential power analysis," J. Cryptogr. Eng., vol.1, no.1, pp.5–27, April 2011.

[6] E. Oswald, S. Mangard, N. Pramstaller, and V. Rijmen, "A side-channel analysis resistant description of the AES S-box," Proc. FSE LNCS, ed. H. Gilbert and H. Handschuh, Berlin, Heidelberg, pp.413–423, Springer Berlin Heidelberg, 2005.

[7] H. Kim, S. Hong, and J. Lim, "A fast and provably secure higher-order masking of AES S-box," Proc. CHES LNCS, ed. B. Preneel and T. Takagi, Berlin, Heidelberg, pp.95–107, Springer, Berlin, Heidelberg, 2011.

[8] Y. Wang and Y. Ha, "FPGA-based 40.9-gbits/s masked AES with area optimization for storage area network," IEEE Trans. Circuits Syst. II Express Briefs, vol.60, no.1, pp.36–40, Jan. 2013.

[9] V. Grosso, E. Prouff, and F.X. Standaert, "Efficient masked S-boxes processing – a step forward –," Proc. 7th Annu. Int. Cryptol. Conf. Adv. Cryptol. (AFRICACRYPTO), ed. D. Pointcheval and D. Vergnaud, Cham, pp.251–266, Springer International Publishing, 2014.

[10] S.S. Chawla, S. Aggarwal, S. Kamal, and N. Goel, "FPGA implementation of an optimized 8-bit AES architecture: A masked S-box and pipelined approach," IEEE CONECCT, pp.1–6, July 2015.

[11] T. Güneysu and A. Moradi, "Generic side-channel countermeasures for reconfigurable devices," Proc. CHES LNCS, ed. B. Preneel and T. Takagi, Berlin, Heidelberg, pp.33–48, Springer, Berlin, Heidelberg, 2011.

[12] A.G. Bayrak, N. Velickovic, F. Regazzoni, D. Novo, P. Brisk, and P. Ienne, "An EDA-friendly protection scheme against side-channel attacks," DATE, DATE '13, San Jose, CA, USA, pp.410–415, EDA Consortium, 2013.

[13] W. Shan, L. Shi, X. Fu, X. Zhang, C. Tian, Z. Xu, J. Yang, and J. Li, "A side-channel analysis resistant reconfigurable cryptographic coprocessor supporting multiple block cipher algorithms," Proc. 51st ACM/EDAC/IEEE Design Autom. Conf. (DAC), DAC '14, New York, NY, USA, pp.176:1–176:6, ACM, 2014.

[14] K. Tiri and I. Verbauwhede, "A VLSI design flow for secure side-channel attack resistant ICs," Proc. Design Automation and Test Eur. Conf. (DATE), pp.58–63 vol.3, March 2005.

[15] K. Tiri and I. Verbauwhede, "A logic level design methodology for a secure DPA resistant ASIC or FPGA implementation," Proc. Design and Test in Europe Conf.(DATE), pp.246–251 vol.1, Feb. 2004.

[16] N. Mentens, B. Gierlichs, and I. Verbauwhede, "Power and fault analysis resistance in hardware through dynamic reconfiguration," Proc. CHES LNCS, ed. E. Oswald and P. Rohatgi, Berlin, Heidelberg, pp.346–362, Springer, Berlin, Heidelberg, 2008.

[17] D. Mesquita, B. Badrignans, L. Torres, G. Sassatelli, M. Robert, and F. Moraes, "A cryptographic coarse grain reconfigurable architecture robust against DPA," Proc. 21st Int. Parallel and Distributed Processing Symp.(IPDPS 2007), pp.1–8, March 2007.

[18] M. Stöttinger, S. Malipatlolla, and Q. Tian, "Survey of methods to improve side-channel resistance on partial reconfigurable platforms," Design Methodologies for Secure Embedded Systems, ed. A. Biedermann and H.G. Molter, Berlin, Heidelberg, pp.63–84, Springer, Berlin, Heidelberg, 2011.

[19] W. Shan, X. Fu, and Z. Xu, "A secure reconfigurable crypto ic with countermeasures against SPA, DPA, and EMA," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol.34, no.7, pp.1201–1205, July 2015.

[20] N. Veyrat-Charvillon, M. Medwed, S. Kerckhof, and F.X. Standaert, "Shuffling against side-channel attacks: A comprehensive study with cautionary note," ASIACRYPT 2012, pp.740–757, Springer,

2012.

[21] A. Moradi, O. Mischke, and C. Paar, "Practical evaluation of DPA countermeasures on reconfigurable hardware," HOST 2011, pp.154–160, IEEE, 2011.

[22] S. Chen, W. Ge, J. Yang, B. Liu, and J. Yang, "A power analysis attack countermeasure based on random execution," IEEE TrustCom, pp.1474–1479, IEEE, 2018.

[23] S. Mangard, E. Oswald, and T. Popp, Power analysis attacks: Revealing the secrets of smart cards, Springer Science & Business Media, 2008.

[24] D.D. Hwang, K. Tiri, A. Hodjat, B.C. Lai, S. Yang, P. Schaumont, and I. Verbauwhede, "AES-based security coprocessor IC in 0.18-μm CMOS with resistance to differential power analysis side-channel attacks," JSSC, vol.41, no.4, pp.781–792, 2006.

[25] M. Doulcier-Verdier, J. Dutertre, J. Fournier, J. Rigaud, B. Robisson, and A. Tria, "A side-channel and fault-attack resistant AES circuit working on duplicated complemented values," ISSCC, pp.274–276, Feb. 2011.

[26] C. Tokunaga and D. Blaauw, "Secure AES engine with a local switched-capacitor current equalizer," IEEE ISSCC Dig. Tech. Papers, pp.64–65, 65a, Feb. 2009.

Wei Ge received the B.S. degree in Electronics Engineering from Southeast University in 2006. He is now a Ph.D. candidate in Electrical Engineering Department of Southeast University. His research mainly focuses on the SoC design technology and reconfigurable computing and related VLSI design.

Shenghua Chen received the B.S. from Southwest Jiaotong University in 2015 and and the M.S. degrees from Southeast University in 2018. His current research interests include reconfigurable computing and power analysis attacks.

Benyu Liu received the B.S. degrees from Hohai University in 2017, and he is currently pursuing the M.D. degree at Southeast University. His current research interests include power analysis attacks and silicon physical unclonable function.

Min Zhu received the B.S. degree in electronic engineering from Tsinghua University, Beijing, China, in 2006, and the Ph.D. degree from the Institute of Microelectronics, Tsinghua University, in 2012. He is currently with Wuxi Research Institute of Applied Technologies, Tsinghua University. His research interests include reconfigurable computing, and cryptographic engineering.

Bo Liu received the B.S. and Ph.D. degrees in Electrical Engineering from Southeast University in 2006 and 2013 respectively. He is currently a lecturer of National ASIC system Engineering Research Center (CNASIC), Southeast University. His research interests include chip architecture design, reconfigurable computing and related VLSI designs.