Countering power analysis attacks by exploiting characteristics of multicore processors

Jianwei Yang¹, Fan Dai¹, Jielin Wang¹, Jianmin Zeng²,
Zhang Zhang², Jun Han^{1a)}, and Xiaoyang Zeng¹

¹ State Key Laboratory of ASIC and System, Fudan University,
825 Zhangheng Rd, Shanghai 201203, China

² School of Electronic Science and Applied Physics, Hefei University of Technology,
193 Tunxi Rd., Hefei 230009, China

a) junhan@fudan.edu.cn

Abstract: Power analysis attacks are major concerns among all kinds of side channel attacks. This paper proposes three kinds of countermeasures for Differential Power Analysis (DPA) attacks by fully exploiting characteristics of a many-core processor: (1) Data-level parallelism, (2) Random Dynamic Task Scheduling (RDTS), and (3) Random Dynamic Adjustment (RDA), which combines RDTS, Random Dynamic Frequency Scaling (RDFS) and Random Dynamic Phase Adjustment (RDPA). For the first time these techniques are applied to a multicore processor from the perspective of secure system design. AES algorithm with mentioned countermeasures is implemented on an 8-core processor. Simulation results show that these countermeasures considerably improve the system security with little performance overhead.

Keywords: DPA, CPA, multicore processor

Classification: Integrated circuits

References

- [1] P. Kocher, *et al.*: “Differential power analysis,” CRYPTO (1999) 388 (DOI: [10.1007/3-540-48405-1_25](https://doi.org/10.1007/3-540-48405-1_25)).
- [2] J.-S. Coron and I. Kizhvatov: “Analysis and improvement of the random delay countermeasure of CHES 2009,” CHES (2010) 95 (DOI: [10.1007/978-3-642-15031-9_7](https://doi.org/10.1007/978-3-642-15031-9_7)).
- [3] N. Veyrat-Charvillon, *et al.*: “Shuffling against side-channel attacks: A comprehensive study with cautionary note,” Proc. of the 18th International Conference on The Theory and Application of Cryptology and Information Security (2012) 740 (DOI: [10.1007/978-3-642-34961-4_44](https://doi.org/10.1007/978-3-642-34961-4_44)).
- [4] C. Clavier, *et al.*: “Differential power analysis in the presence of hardware countermeasures,” CHES (2000) 252 (DOI: [10.1007/3-540-44499-8_20](https://doi.org/10.1007/3-540-44499-8_20)).
- [5] P. C. Kocher, *et al.*: U.S. Patent 7587044 (2009).
- [6] T. Güneysu and A. Moradi: “Generic side-channel countermeasures for reconfigurable devices,” CHES (2011) 33 (DOI: [10.1007/978-3-642-23951-9_3](https://doi.org/10.1007/978-3-642-23951-9_3)).

- [7] S. Mangard: “Hardware countermeasures against DPA - A statistical analysis of their effectiveness,” *Topics in Cryptology - CT-RSA* (2004) 222 (DOI: [10.1007/978-3-540-24660-2_18](https://doi.org/10.1007/978-3-540-24660-2_18)).
- [8] S. Nagashima, *et al.*: “DPA using phase-based waveform matching against random-delay countermeasure,” *ISCAS* (2007) 1807 (DOI: [10.1109/ISCAS.2007.378024](https://doi.org/10.1109/ISCAS.2007.378024)).
- [9] S. Mangard, *et al.*: *Power Analysis Attacks: Revealing the Secrets of Smart Cards* (Springer, US, 2010) 209.
- [10] Q. Tian and S. A. Huss: “A general approach to power trace alignment for the assessment of side-channel resistance of hardened cryptosystems,” *Eighth International Conference on Intelligent Information Hiding and Multimedia Signal Processing* (2012) 465 (DOI: [10.1109/IIH-MSP.2012.119](https://doi.org/10.1109/IIH-MSP.2012.119)).
- [11] J. G. J. van Woudenberg, *et al.*: “Improving differential power analysis by elastic alignment,” *Topics in Cryptology - CT-RSA* (2011) 104 (DOI: [10.1007/978-3-642-19074-2_8](https://doi.org/10.1007/978-3-642-19074-2_8)).
- [12] L. Benini, *et al.*: “Energy-aware design techniques for differential power analysis protection,” *Proc. of the 40th Annual Design Automation Conference* (2003) 36 (DOI: [10.1145/775832.775845](https://doi.org/10.1145/775832.775845)).
- [13] K. Tiri, *et al.*: “A dynamic and differential CMOS logic with signal independent power consumption to withstand differential power analysis on smart cards,” *ESSCIRC* (2002) 403.
- [14] M. Kar, *et al.*: “Improved power-side-channel-attack resistance of an AES-128 core via a security-aware integrated buck voltage regulator,” *ISSCC Dig. Tech. Papers* (2017) 142 (DOI: [10.1109/ISSCC.2017.7870301](https://doi.org/10.1109/ISSCC.2017.7870301)).
- [15] S. Yang, *et al.*: “Power attack resistant cryptosystem design: A dynamic voltage and frequency switching approach,” *Proc. of the Conference on Design, Automation and Test in Europe* (2005) 64 (DOI: [10.1109/DATE.2005.241](https://doi.org/10.1109/DATE.2005.241)).
- [16] K. Baddam and M. Zwolinski: “Evaluation of dynamic voltage and frequency scaling as a differential power analysis countermeasure,” *20th International Conference on VLSI Design Held Jointly with 6th International Conference on Embedded Systems* (2007) 854 (DOI: [10.1109/VLSID.2007.79](https://doi.org/10.1109/VLSID.2007.79)).
- [17] F. Gurkaynak, *et al.*: “Improving DPA security by using globally-asynchronous locally-synchronous systems,” *ESSCIRC* (2005) 407 (DOI: [10.1109/ESSCIR.2005.1541646](https://doi.org/10.1109/ESSCIR.2005.1541646)).
- [18] J. Lin, *et al.*: “A scalable and reconfigurable 2.5D integrated multicore processor on silicon interposer,” *CICC* (2015) (DOI: [10.1109/CICC.2015.7338447](https://doi.org/10.1109/CICC.2015.7338447)).

1 Introduction

With the continuous development of information technology, security is becoming more and more significant for various systems. Advanced Encryption Standard (AES) is widely used to protect the key information from being leaked. Although this algorithm is mathematically secure enough by now, attackers can still retrieve the key by using side channel attacks. Power analysis attacks are the most effective side channel attacks and mainly involve Simple Power Analysis (SPA) attack and Differential Power Analysis (DPA) attack. SPA attacks can obtain the secret information by visually inspecting power traces. DPA attacks were first described in [1] and have been proved to be an efficient way to retrieve key information of device. The main idea of DPA is to exploit the relationship between power

consumption and the value of intermediate which is related to key bits. Correlation power analysis (CPA) attacks are one type of the widely used methods of DPA. It takes Hamming weight (HW) or Hamming distance (HD) as power model to predict the power consumption of computational intermediates. The hypothetical power consumption of each key is compared with the recorded power traces by using correlation coefficient. The highest correlation shows the correct key of the device.

After DPA attacks were introduced, there have been many countermeasures proposed. One class of countermeasures is making the power traces misaligned in time dimension, such as insertion of random delays [2], shuffling [3], random process interrupts [4], clock skipping [5], or the random clocking [6]. Misalignment in the measurements sets will lead to failure in DPA attacks. However, simple misalignment in time domain is easy to deal with by integration over a range of points [7] or realignment techniques, such as phase-based waveform matching [8], static alignment with cross-correlation techniques [9], Horizontal Alignment [10], Dynamic Time Warping (DTW) [11]. Another class of countermeasures is trying to hide the encryption-related power in amplitude dimension, making the power consumption either random or equal. These approaches involve increasing noise [12], using dual-rail precharge logic [13] or adding filters [14], all of which actually try to lower the signal-to-noise ratio (SNR) of the performed operation. Increasing the number of power measurements will improve the SNR and thus successful power attacks can be conducted. A more effective way is to make the waveform distortion in both time and amplitude dimension. Alignment is hard to perform due to the changes of the profile of the waveform and the SNR of the executed operation is hard to improve due to misalignment. The countermeasure using Random Dynamic Voltage and Frequency Scaling (RDVFS) [15] is an example of this method. However, secret key still can be successfully retrieved due to the relation between frequency and voltage [16]. Another example is using Globally-Asynchronous Locally-Synchronous (GALS) design methodology as a countermeasure [17]. Even though significant security improvement can be achieved, severe performance penalty due to large amounts of data exchanges among asynchronous modules is introduced.

In recent years, multicore processor is becoming popular due to its high flexibility and excellent parallelism. It can provide excellent parallelism with relatively low power consumption. Due to its characteristics of including many duplicated components, multicore processor can provide special ability for counter-acting power attacks. This paper proposes and analyzes three kinds of countermeasures based on a multicore processor. Data-level parallelism can provide high performance while enhancing the security by around 10 times. Random dynamic task scheduling among different cores increases time uncertainty and power diversity, which increases the difficulty of attacking the system. Random dynamic frequency scaling, which is applied to several cores of the multicore processor, is a little similar to RDVFS. However, the result is much better due to the fact that it's much more difficult to inspect the frequency changes in multicore processors than that in ASIC or single-core systems. Random dynamic phase adjustment, which is beneficial to the security only on multicore processors, considerably improves the

Fig. 1. Architecture of target multicore processor.

system security with negligible overhead. We combine RDTS, RDFS and RDPA as a countermeasure named RDA, and it can significantly increase the security. The rest of this paper is organized as follows: Section 2 briefly introduces the target platform we used. In section 3, detailed analysis and implementations of proposed countermeasures are described. Section 4 presents the simulation results and comparison results. Section 5 concludes the paper.

2 Target multicore platform

An 8-core processor is used as the experimental platform [18], which is shown in Fig. 1. There are two clusters in this multicore processor and each cluster contains four 32-bit MIPS-like processor cores. Each core contains a 6-stage in-order pipeline and some specific instructions are added compared to a standard MIPS core. Multipage foreground and background register file is used for faster data pre-fetch and results write-back. In each core, there is a 750-word private instruction memory and a 512-word shared data memory. All the cores are connected with 2-D mesh NoC. The on-chip network is constructed as packet-controlled circuit-switched double-layer, which makes the on chip communication more flexible. Shared-memory inter-core communication in the multicore system can meet various requirements. The cluster-based shared memory hierarchy can improve data locality, thus reducing power consumption. Communication based on shared memory is much easier and more programmer-friendly. The multicore processor also supports GALS technique, which means different clusters can operate at different frequencies.

3 Detailed analysis and implementation

3.1 Date-level parallelism

Data-level parallelism introduces asynchronous operation on multiple cores, which means more noise and misalignment will be introduced, increasing attacking effort. In addition, it can significantly improve the performance.

The main considerations about parallelization are balance of time consumption among different tasks and communication overhead among different cores. Take AES-128 into consideration, 128-bit operations can be equally partitioned into multiple cores. 4-core mapping is a good choice, since the data to be processed in each core can be exactly 32 bits wide. Inter-core communication is inevitable due to

Fig. 2. Task partitioning on a 4-core cluster.

data dependency and it's significant to using high-efficiency inter-core communication for improving overall performance. In the 8-core processor, the shared memory mechanism is the best solution for low latency inter-core communication in a cluster. The detailed implementation of 4-core mapping is shown in Fig. 2. The cores in cluster1 is used and cores in cluster2 are stalled. A 128-bit plaintext block is depicted as a 4×4 square matrix of bytes. Each core takes one row of data (4 bytes) as input and processes its own data independently except for MixColumns stage. In MixColumns stage, data exchanges have to be performed by using shared memory. Because there are 4 banks in the shared memory within each cluster, the four cores in the same cluster can access the memory simultaneously without conflicts when the requests are for different banks. Flags are used for synchronization among these cores. After MixColumns, each core processes AddRoundKey step independently and then each core gets the new state. After another 9 rounds, we get the 128-bit cipher.

3.2 Random dynamic task scheduling

Dynamic task scheduling on a multicore platform means that each round of AES is executed on different cores. Due to different transmission time among different cores, this will make the power traces misaligned, and thus makes the DPA attacks hard to be performed. On the other hand, making AES round operation computed on a randomly select core will introduce different power characteristics since each core are not totally physically identical. Difference in power characteristics means that the power traces are different in both time dimension and amplitude dimension.

Fig. 3 presents our proposed RDTS scheme. Each core executes one round of AES and then transfers the result to another core. The selection of next core is based on our proposed random algorithm, which will be described in the following part. Detailed encryption flow is shown in Fig. 4. Core0 is responsible for key expansion, and then it transmits the round key to cluster2. Other functions are similar among the cores. When the core is waiting for round results, the pipeline of the core is stalled to reduce power consumption. There are 8 cores in the target platform, and all the cores can communicate with each other through routers. Like what we described before, this platform contains two types of switch strategies-

Fig. 3. Random dynamic task scheduling scheme.

Fig. 4. Detailed encryption flow. (a) Core0. (b) Core1–Core7.

packet and circuit. And the platform has its own defined different types of packets to support inner-communication, especially for cores in disparate clusters. Take the delay into consideration, packet switch works better than circuit switch when amount of the transferred data is small. So in the dynamic random task scheduling, packet switch is used for communication. In this strategy, two types of data packets are used. One is processor data packet, another is memory data packet. The memory data packet is used to send round key from local cluster directly to another cluster's shared memory. Once the transmission of round key is completed, all the cores can get the round key through shared memory. The processor data packet is used to send round data from source core to destination core based on the random algorithm.

It's crucial for security to make the task scheduling truly random. To realize truly random dynamic scheduling, we propose a data-oriented random number generation algorithm. The detailed algorithm is shown in Table I. A random seed is necessary for random number generation algorithm. In our proposal, the operation result of each round is chosen as the random seed. It depends on the input plaintext, round number and the secret key, even the designer doesn't know it at design time. After one round operation is finished, the round operation result and round counter will be refreshed. The round operation result consists of a 16-byte state, which is from S0 to S15. And the destination axis, both x and y, can be determined by the XOR result of the state. Apparently, the destination axis is equally distributed and

Table I. Random number generation algorithm

Finish last round operation										
S0	S1	S2	S3	S4		S11	S12	S13	S14	S15
All State (S0–S15) XOR together $\Rightarrow$ S[7:0]										
S[1:0] $\Rightarrow$ x axis (0–3); S[2] $\Rightarrow$ y axis (0–1)										
Transfer State (S0–S15) and round count to next core (or local)										
Start next round operation										

random, and the transmission route length range from 0 to 4 routers. The different transmission routes cause different start and end points in each round operation and make power traces misaligned.

3.3 Random dynamic adjustment

To further increase time misalignment and power diversity, we propose Random Dynamic Adjustment (RDA), which combines RDTS, Random Dynamic Frequency Scaling (RDFS) and Random Dynamic Phase Adjustment (RDPA), as a countermeasure for power analysis attacks. It is easy to detect the frequency changes in a single-core processor or ASIC system and thus power analysis attacks can be conducted by aligning the power traces. However, things will be totally different when apply this to a multicore processor. A multicore processor consists of several to hundreds of cores and the entire power consumption is the sum of the power consumption of each core. If some of them operate at different frequency, which changes randomly over time, and the task migrates among different cores, then the entire power traces will include all the misaligned power consumption of each core, as shown in Fig. 5(a). It is very difficult to recover the frequency at which the encryption is operating and thus makes the DPA or CPA attacks unpractical. The benefit resulted from the RDFS depends on the total number of feasible frequencies. It is possible to vary operating frequency at a wide range over time by using PLL.

The problem of random dynamic frequency scaling is performance degradation. When the frequencies of the cores are dynamically changed, it means some of cores can't operate at the highest possible frequency, which results in longer execution time for encryption. Hence we propose a solution that the clock phases of different cores are different while the frequency is identical. Thus the entire processor can operate at highest frequency. Only on multicore platform is this method feasible, because a single-core processor has only one clock while each core of a multi-core processor can operate at different clocks, with same frequency but different phases, as shown in Fig. 5(b). The phase of each clock is randomly dynamically adjusted and the entire power consumption of the processor is the sum of each core's power consumption, which makes it impractical for the attackers to recover the clock information to align the power traces. It is meaningless to perform DPA attacks or CPA attacks on totally misaligned power traces. In practical processors, phase adjustment can be implemented by using programmable delay array.

As mentioned in section 2, our platform supports GALS, thus different clusters can operate at different frequencies. The method of RDFS can introduce larger

Fig. 5. (a) Clocks of RDFS, (b) Clocks of RDPA.

uncertainty than RDPA, but suffers from much more reduction in performance. As a tradeoff between security and performance, we combine RDFS with RDPA together by changing the frequency and clock phase of cluster1 in turn in our simulation. A counter whose max value is 200,000 is used to control the alteration of RDFS and RDPA. When the counter is less than 150000, the clock phase of cluster1 is dynamically changed with frequency keeping constant, otherwise the frequency of cluster1 is dynamically changed. Clock frequency scaling range is from 50% to 100% of cluster0's frequency and max phase adjustment value of RDPA is 50% of the clock cycle of cluster0.

4 Experimental results and comparison

To verify the proposed countermeasures, logical synthesis (using Design Compiler) and physical synthesis (using IC Compiler) of the target multi-core platform are performed with GlobalFoundries 65 nm technology. And to simplify the whole flow, an auto power simulation platform is set up. This platform contains compilation of c code by MIPS SDE, post-simulation by NC-Verilog and dynamic power simulation by using PTPX. MATLAB is used as the tool for power analysis. In real scenario, multiple different tasks will be deployed on the multicore processor, which significantly increases the noise of power traces. In our experiment, only encryption task is deployed (The other cores are stalled) to reduce the number of required power traces for successful attacks, since the power simulation of a multicore is really time-consuming. Much more power traces will be required for realistic CPA attacks. Furthermore, even though the multicore processor can operate at 500 MHz under 1.2 V supply, a lower frequency of 37 MHz is used for simulation. The reason is that lower frequency can reduce the overlap between the power values at two adjacent clock edges, which also makes our power analysis attack easier to reduce simulation time. First, classical CPA attacks are conducted and the results are shown in Fig. 6. Each line represents a key hypothesis. For the purpose of comparison, AES encryption on a single core of the multicore processor is conducted and almost 160 power traces are enough for correct key extraction. Data-level parallelism increases security by almost 10 times. Because of misalignment of power traces, even with 100 times more traces classical CPA does not work well for RDFS and RDPA. Actually, it's meaningless to use more traces since the signal to noise ratio (SNR) will not increase.

Fig. 6. Results of CPA attacks. (a) Single core. (b) data-level parallelism. (c) RDTs. (d) RDA.

Fig. 7. (a) The CPA attack result on RDTs after alignment. (b) DTW alignment result between trace x and trace y. (c) The CPA attack result on RDA after alignment.

Then, we perform static alignment [9] for RDTs. The power traces of RDTs can be aligned by static alignment since the noise which is introduced by these idle cores are relatively low and stable. And the CPA attack on the aligned power traces is successful as shown in Fig. 7(a). DTW [11] is used for RDA, since DTW can align two traces which are “warped” non-linearly. The alignment result between power trace x and power trace y is shown in Fig. 7(b). However, the CPA attack on the aligned power traces is failed as shown in Fig. 7(c). We analyze the detailed power traces and find out the reason is that there are many extra peak values in the power traces of RDA as shown in Fig. 8(a). This is because the frequency and phase of cluster1 is dynamically changed. Some of the peak values represent cluster0’s power values, the others are cluster1’s power values, and the encryption task is randomly dynamically scheduled among the cores of these two clusters. This method introduces confusion not only in time dimension, but also in amplitude dimension. Fig. 8(b) shows total power waveform and each core’s power waveform during some period of encryption. Core0 is the core that is executing the encryption and other cores are idle. Note that even though the idle core’s pipeline is stalled, other parts of the core, such as memory module, routers and control logic still

Fig. 8. Power traces of RDA.

consume some power. At time0, the peak power of core0 (279.5 mW) is larger than that (131.8 mW) at time1. However, due to the misalignment of other cores' peak power value (resulting from RDFS and RDPA), the total peak power (606.0 mW) at time0 is smaller than that (695.4 mW) at time1. This means the power pattern of encryption has been completely changed due to phase adjustment. Most of alignment methods, such as static alignment and DTW, are based on the profile of the power traces, thus the power traces can't be accurately aligned. For other alignment methods which are based on signal processing, RDA is also effective, since the signal and noises (the power of idle cores) are both randomly dynamically changed in time dimension and the noises are larger than the signal.

Table II shows the performance comparison of different countermeasures. Compared with single core, data-level parallelism can significantly improve the performance, while RDTs decreases the performance by 25.6%. It mainly results from the time of execution of random number generation algorithm and the time of transmission of different tasks. The performance reduction of RDA is 37.4%, which is relatively low compared with its security improvement.

Table II. Comparison of different countermeasures

Countermeasure	Cycles*	Decrease in speed
Single Core	6270	-
Data-level parallelism	2502	-60.1%
RDTs	7878	25.6%
RDA	8613	37.4%

*Cycles for 128-bit data encryption. Key expansion is not included, since it is executed only for the first time and all the round keys are saved in shared memory.

5 Conclusion

In this paper, we propose and analyze three kinds of countermeasures for DPA Attacks. Experimental results show that data-level parallelism can improve security by 10 times while significantly increase encryption speed. RDTs alone is not secure enough while RDA can tremendously prevent power attacks with little performance overhead.

Acknowledgments

This work is supported by National Natural Science Foundation of China (61404043, 61674049). We would like to express our sincere thanks to the teachers and students in State Key Laboratory of ASIC & System, Fudan University, for their help.