UDC 004.31, 004.056.55, 003.26

Karpinskyy M., *Korkishko L.
Department of Electrical Engineering
Faculty of Mechanical Engineering and Computer Sciences
University of Bielsko-Biala
Willowa Str. 2, Bielsko-Biala, 43-309, Poland
*Department of Security of Informational Technologies
Faculty of Computer and Informational Technologies
Ternopil State Economic University
Lvivska Str. 11, Ternopil, 46004, Ukraine
E-mail: mkarpinski@ath.bielsko.pl, lesykkor@yahoo.com

ARCHITECTURE OF CRYPTOGRAPHIC DEVICES RESISTANT TO SIDE-CHANNEL ATTACKS

Processed by a computational device cryptographic data can be leaked through various side channels, such as power consumption, electromagnetic radiation, execution time etc. We suggest general architecture and framework for development of computational device that counter side channel attacks by processing randomized data.

Keywords – information security, cryptography, side-channel attack, differential power analysis, computer architecture, data masking.

1. Introduction

A number of custom and standardized algorithms for data protections are used in computer security application. While many implementations are depended on software realization of cryptographic functions on powerful universal processors, there is a big amount of mobile applications with high security requirements. Consequently, the requirements and design criteria for cryptographic functions vary considerably.

Small footprint, low power consumption, high throughput of data processing – these are small subset of criteria for implementation of cryptographic functions designated for smart cards and related embedded devices. From the time of side-channel attacks advent, one of major concerns is resistance to such kind of attacks.

Processed by a computational device cryptographic data can be leaked through various side channels, such as power consumption [1], electromagnetic radiation [2], execution time [3] etc. Therefore, the most general method to counter side channel attacks is to randomize processed data. The problem is to guarantee that an attacker can receive random data from a side channel and cannot obtain any useful actual knowledge about initial and/or intermediate data involved in computations [4].

To protect against side channel attack, a computational device that can transform all initial/intermediate results into random form should be provided. The problem is that different operations on data should take into account a fact that data actually are randomized. Therefore, it is not always straightforward how to modify data processing flow of original algorithm's specification/implementation.

The main contribution of this paper is that we suggest architecture of computational device that is protected against side channel attacks. Such architecture creates a framework for development of number of specialized computational devices. These became possible, since we provide high-level architecture without concrete assumptions about the character of executed operations. As a basic method for data randomization we enforce data masking with exclusive or operation.

2. Computational device for cryptographic operations

Cryptographic operations and functions consist of broad range operations on data and secret key, e.g. encryption with symmetric algorithms (block ciphers, stream ciphers etc), public key algorithms (RSA, ECC, etc), message authentication codes (MACs) [5]. We can view a computational device for cryptographic operations as a "black box" with inputs for data and key, and output for processed data. However, if we consider modern implementations of the cryptographic computational devices, we can describe its architecture using memory-processor unit approach (Fig. 1).

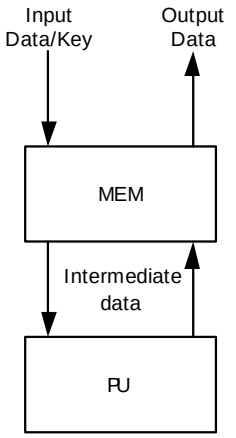

Fig. 1. High-level representation of cryptographic computational device.

Cryptographic computational device consists of memory and processor unit. Here, for generality, we assume multiport memory and multi data flow processor unit that executes a cryptographic algorithm. As partial cases of the cryptographic computational device we name combination of random-access memory and programmable processor, specialized hardware-oriented processors with different architectures (iterative, pipeline, combined) with memory for input, output, and intermediate results [6].

Memory stores input data and key. Then processor unit performs computations on input data and key according to the cryptographic algorithm specification. During computations, some intermediate results might be produced. Then the processor unit stores those intermediate results in memory. Having processed data and key according to cryptographic algorithm, final results are placed into memory. Final results become available on output data output port of the memory.

3. Protection of computational device against side channel attacks using data masking

Side-channel attacks are sound because there is a correlation between the physical measurements taken during computations (e.g. power consumption [1], electro-magnetic field radiation [2], time of computations [3]) and the internal state of the cryptographic computational device, with itself is related to a secret key.

Among many attacks, the Differential Power Analysis (DPA) is the most dangerous [7, 8]. It uses statistical analysis to extract information from a collection of power consumption curves obtained by running an algorithm many times with different inputs. Then an analysis of a probability distribution of points on the curves is carried on. The DPA attack uses correlations between power consumption patterns and specific key-dependent bits which appear at known steps of the cryptographic computations. For example, a selected bit b at the output of one S-box of the first round of AES [9] will depend on the known input message and 8 unknown bits of the key. The correlation between power consumption and b can be computed for all 256 values of 8 unknown bits of the key. Then an attack can be repeated for the remaining S-boxes.

There are many strategies to combat side-channel attacks [10, 11]:

- among software countermeasures against SPA/DPA are such techniques as introducing dummy instructions and/or random wait states, balancing of Hamming weight of internal data, randomization of instruction executing sequence etc.;
- among hardware countermeasures against SPA/DPA are such techniques as introducing random noise into power consumption patterns, using internal power source or switching power sources, randomization of order of operation execution, randomizing registers renaming etc.

The most powerful software and hardware countermeasure appears to be so-called bit-splitting [7, 10, 11, 12], which in case when each bit is split into two shares can be reduced to masking data with random values. However, more shares offer better protection against higher-order DPA attacks.

The idea how to apply data masking to cryptographic computations is as following: input data, as well as the key, are masked with some random masks at the beginning of computations, and thereafter computations are done on masked values. From architectural point of view, we transform our initial architecture from Fig. 1 to another one. Proposed architecture of cryptographic computational device (Fig. 2) reflects computations on masked data.

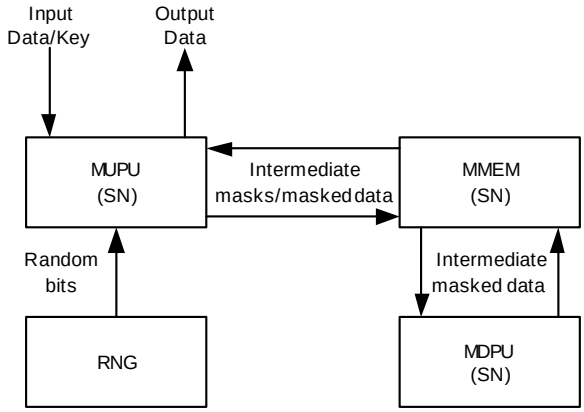

Fig. 2. Architecture of side-channel resistant cryptographic computational device.

Side-channel resistant cryptographic computational device consists of physical source of random data – random number generator (RNG), mask-update processor unit (MUPU), memory for masked data and masks (MMEM), and masked data processor unit (MDPU). General idea is to follow bit-splitting countermeasure against side-channel attacks. Before initiation of computation on input data and key, that information is entered into cryptographic computational device. Note that input data and key are entered in plaintext form since it is assumed that first user of the device is legitimate. Entered information is immediately masked and stored in masked form.

MUPU splits Input Data or Key in plaintext into set of n randomized shares using random bit strings provide by RNG. Then shares are stored in MMEM to allow MPDU to process masked shares according to a cryptographic algorithm. Note that for most cases the cryptographic algorithm remains the same as for plaintext data, but some additional computations have to be done on masks. Those additional computations are called “mask correction” [13]. Mask correction process is required since operations on masked data require respective modifications of the masks themselves. Without mask correction process it will be impossible to retrieve original (not masked) results of the computations.

MPDU stores intermediate results of computations on masked data into MMEM. Later on, information from MMEM is updated by MUPU to reflect changes into masks according to respective cryptographic algorithm. There changes include mask update procedure that is performed on intermediate masks. For this purpose new random bits from RNG might be used. There are several strategies for new random bits usage in masking computations [13]:

- using initial mask values used to mask information before initial processing of that information (i.e. single-mask approach);
- using mask update for some (or all) intermediate results (i.e. multiple-mask approach).

Both strategies have their strong points – first one offers affordable performance of RNG, while second – can offer potentially more secure solution, as new masks are re-applied to some (or all) intermediate data. Selection of the strategy depends on design criteria for particular cryptographic device.

Having processed all intermediate results and received final result of computations, MDPU stores results into MMEM. Thereafter MUPU executes unmasking operation by removing masks from actual final data and providing final data in clear text to Output Data.

Described above set and sequence of operations are done over masked data. Since real implementation of side-channel resistant cryptographic computational device might leak side-channel information, at attacker receives power curves that contain randomized information about real key was used for computations. To combat higher-order side-channel attacks, a security parameter SN can be used. This security parameter controls the number of shares that Input Data/Key is split to. Minimal value of SN=2 that allows protection from first-order differential attacks.

If the architecture utilizes higher number of SN, an implementation of such side-channel resistant cryptographic computational device will take more cost in terms of spent resources and performance of RNG. Therefore, designer of cryptographic system should analyze possibilities of an attacker and select appropriate SN to make the device resistant against selected order of side channel attack.

4. Conclusion

In this contribution we propose architecture for side-channel resistant cryptographic computational device. We use masking technique to combat side-channel attacks on such cryptographic computational device. Since an

attacker might has different capabilities for attacking the device, we introduce a security parameter that controls the number of shares that input data/key is split to. Minimal number of shares is two that prevents first-order differential side-channel attacks. Other values of the security parameter provide changes in architecture that allows one to offer protection from higher-order differential side-channel attacks. However, increasing value of security parameter results in growing complexity of device implemetation.

The device consists of two processor units – one for masks processing (mask correction) and another – for masked data processing. All intermediate results are processed and stored in randomized form that prevents at attacker from successful using leaked side-channel information to perform side-channel attacks.

Such architecture provides a framework for designers of side-channel resistant cryptographic computational devices. Among a number of possible application fields for such the devices are smart cards, mobile phones, and communication equipment. In future work we will detail the architecture for basic operations used in cryptographic primitives and typical cryptographic functions (primitives) like symmetric, asymmetric ciphers, and MACs. Thus we will provide systematic approach for designing of cryptographic devices immune to side-channel attacks.

References

[1] P. Kocher, J. Jaffe, B. Jun, “Differential power analysis”, Advances in Cryptology – CRYPTO’99, vol. 1666, Lecture Notes in Computer Science, Springer-Verlag, 1999. pp. 388 – 397.

[2] J. J. Quisquater, D. Samide, “Electromagnetic analysis (ema): measures and countermeasures for smart card”, Proceedings “Smartcard Programming and Security”, vol. 2140, Lecture Notes in Computer Science, Springer-Verlag, 2001, pp. 200 – 210.

[3] P. Kocher, “Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems”, Proceedings “Advances in Cryptology – CRYPTO’96”, vol. 1109, Lecture Notes in Computer Science, Springer-Verlag, 1996, pp. 104 – 113.

[4] P. Kocher, J. Jaffe, B. Jun, “Using unpredictable information to minimize leakage from smartcards and other cryptosystems”, USA Patent, International Publication number WO 99/63696, December 9, 1999.

[5] A. Memezes, P. Van Oorschot, S. Vanstone, “Handbook of Applied Cryptography”, CRC Press, Boca Raton, FL, USA, 2001.

[6] Коркішко Т. Алгоритми та процесори симетричного блокового шифрування: Захист інформації в комп'ютерних та телекомунікаційних мережах/ Т. Коркішко, А. Мельник, В. Мельник; Наук. ред. О. Коссак. – Львів: БаК, 2003. – 168 с.

[7] T. S. Messerges, E. A. Dabbish, R. H. Sloan, “Examining smart-card security under the threat of power analysis”, IEEE Trans. Computers, vol. 51, no. 5, May 2002, pp. 541 – 552.

[8] Коркішко Л.М., Васильцов І.В., Статистична модель операції додавання за модулем 2N для проведення інженерно-криптографічних атак за побічними каналами витоку інформації // Правове, нормативне та метрологічне забезпечення системи захисту інформації в Україні. – 2004. – №8. – С. 115 – 121.

[9] J. Daemen, V. Rijmen, “The design of Rijndael: AES – The Advanced Encryption Standard”, Berlin, Heidelberg, Springer-Verlag, 2002.

[10] S. Chari, C. Jutla, J. Rao, P. Rohatgi, “Towards sound approaches to counteract power-analysis attacks”, Proceedings “Advances in Cryptology – CRYPTO’99”, vol. 1666, Lecture Notes in Computer Science, Springer-Verlag, 1999, pp. 398 – 412.

[11] М.Карпінський, Л.Коркішко, Т.Коркішко Інженерно-криптографічна атака за аналізом споживаної потужності на прорамно-апаратні реалізації криптографічного перетворення за чинним стандартом // Вісник Тернопільського державного технічного університету. – 2005. №3. – С. 127-135.

[12] L. Goublin, J. Patarin, “DES and differential power analysis”, Proceedings “Cryptographic Hardware and Embedded Systems: CHES’99”, vol. 1717, Lecture Notes in Computer Science, Springer-Verlag, 1999, pp. 158 – 172.

[13] E. Trichina, L. Korkishko, “Secure and efficient AES software implementation for smart cards”, Proceedings “5th International Workshop on Information Security Applications – WISA 2004”, vol. 3325, Lecture Notes in Computer Science, Sprinder-Verlag, 2004, pp. 425 – 439.