International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
ISSN: 0976-1353 Volume 14 Issue 1 –APRIL 2015.

MASKING WITH NO-FILL MECHANISM DURING AES AGAINST CACHE BASED ATTACKS

[1]S.PRADEEP AND [2]DR.NOOR MUHAMMAD SK
1PG Scholar, Department of computer Science and Engineering, Sona college of Technology
2Assistant Professor, Indian Institute of Information Technology, Design and Manufacturing, Chennai

ABSTRACT

AES is the most widely used type of encryption technique till now. Several types of software side channel vulnerabilities also developed to hack the plain text and key used in the AES. This paper demonstrates an efficient algorithmic countermeasure to secure the AES encryption against different side channel attacks. The side channel attacks we mentioned here is timing attack and power analysis attack which is relevant to the cache attacks. Our idea combines two countermeasures which are disabling cache and masking mechanism. By doing this, the encryption process of AES can be implemented successfully without any vulnerabilities and thwarts the intermediate values against side channel attacks.

INTRODUCTION

Advanced Encryption Standard (AES) is the widely used encryption scheme all over the world [1]. Using this technique, we can capable of protecting sensitive information by encrypting the plain text with the key. AES is a symmetric block cipher which has a block size of 128 bits and key size of 128,192 and 256 bits. It contains different rounds depends upon the key size where each round consists of four operations such as subBytes, shiftRows, mixColumns and addRoundKey which are used to build a cipher from the plain text. In software implementation, the above mentioned four operations are minimized by using lookup tables to increase the performance of AES encryption. There are 8 lookup tables used for AES, each consists of 256 4-byte word.

However AES encryption provides security, it is vulnerable for its memory access pattern. Due to this bad publicity AES is bothering of several attacks, where attacks that do not exploit inherent weakness of an algorithm. But rather a characteristic of the implementation that gets useful information about the secret key or the plain text. This leakage of information is termed as side channel information and the attacks exploiting side channel information is called as side channel attacks [2].

Some of the side channel attacks are power analysis, timing attacks, electro-magnetic radiations, faults and so on. We mainly focus on power analysis and timing attacks because power profile and time taken for encryption of the system gives away cache behavior. In Power analysis, the attacker studies the power consumption of the hardware device and extracts information from it [4]. During the power traces the cache memory access pattern is almost visible to the attacker and provides some useful information about the encryption. Block ciphers like AES leaks some timing information during cache misses [3]. This is demonstrates as a cache-based timing attacks by the attackers to hack the plain text or key used in the encryption.

Cache timing attack depends on the time for accessing the data present in the memory [5]. If the requested data is present in the cache (cache hit) then the accessing time is very fast. Or else the data which is requested by the processor is not present in the cache memory (cache miss) then the processor wants to fetch that data from the main memory and put it into cache (for future use) after that should be used by the processor. In this case the accessing time will be slower, so the attacker can understand that the cache miss occurred and can easily identify the data that fetch from the main memory.

In this paper, we propose a countermeasure that to preload lookup table into cache memory for

reducing the cache misses. This is not possible in all situations so we mask the data in lookup table present in the main memory. Whenever cache miss occur the processor fetch the data from main memory and do not fill in the cache for reducing vulnerabilities. The fetched data which is masked can then be unmasked by the processor and used for computation of AES encryption. This results in making difficult to the attacker to hack the plain text and the secret key.

RELATED WORK

In order to avoid or reduce the attack we need to analyze certain countermeasures from various research papers. By the research we come across an idea, that is, to reduce the attack we need to bluff the power profile and the timing analysis of original result. So the countermeasure we want to take should be either, reducing the time variance so there is no difference between cache hits and misses or increasing the time difference so all are same in measure. Researchers have proposed many algorithms to countermeasure the cache based timing attack. Some of them are given below:

Herath et al. have presented a software implementation level countermeasure against cache timing attacks based on masking timing information by "constant-time-encryption" [10]. Here the author compares two algorithms such as fixed number of clock cycles and average number of clock cycles. In fixed number method the number of clock cycles for encryption is more than 2 times when compare with the normal encryption of AES. Average number method the number of clock cycles used for encryption is also greater but comparatively less than the fixed number method. The number of missing bytes is very close in both the above mentioned approach where increase in missing bytes only improves the protection of AES implementation.

Christof Paar et al. have proposed a countermeasure as masking of higher order DPA to secure AES implementation against cache based timing attacks [9]. Many approaches discussed on AES S-box and lower order DPA where this paper concentrates on higher order DPA for further protection. Likewise Coron also described an algorithm which is based on masking of lookup tables in higher order [6]. In this approach there is a lag in AES S-box structure when compare to other approaches. But the common things in these papers are, both represents masking in higher order which slows down the performance during S-box recomputation.

Jayasinghe et al. have performed a research on cache timing attacks and implemented a constant time encryption countermeasure [8]. They analyzed that many of the countermeasure had been implemented but still there is some vulnerabilities in statistical analysis. So they focus mainly in statistical analysis and proposed a technique that reschedules the instructions of AES algorithm then the cache hit and miss should consume constant time. Their implementation results that the encryption takes same amount of clock cycles which are independent from the cache hits and misses. By this approach they are sure that the vulnerability in statistical analysis is eliminated.

Fournier et al. proposed a work which describes the cache attacks against software implementation of AES in case of smart cards [7]. This is based on the power analysis which leaks side channel information during AES encryption. The countermeasures described in this paper are random delay which uses dummy codes to prevent side channel attacks, random order determines no possible of relationship between cache hits and misses and xtime operation which computed without penalty when compares with look-up table implementation. ByteSub and xtime function attacks are valid in masking types of countermeasure against DPA like attacks.

PROPOSED WORK

In this paper, we propose an idea which combines masking mechanism with disabled cache describes in [12]. Our aim is to prevent the plain text and key during encryption against side channel attack by hiding the natural cache timing pattern. The side channel attacks we mentioned in this paper are cache timing attack and power analysis attack. The given idea will overcome these attacks in different perspectives by means of security in AES cryptosystem.

In introduction we talk about the look-up tables which are used for encryption should be preload into the cache memory. If the cache holds the table data for entire rounds of encryption then no

problem will occur for the plain text and key against attacks. But there is no guarantee that those values will remain in cache, because other processes is inevitably compete for cache and removes table data. So we want to do some additional mechanisms to thwart from attacks by using masking.

Masking can be of many types which blind the original data for attackers and correct it for CPU usage. Though masking provides security for data it concerns more time for implementation. So we use a technique which is "Perfectly Masked" S-Box with minimum operation takes place [11]. By this we can easily protect data present in the main memory which is used by the processor by doing mask corrections.

Masking can prevent data from attacks but still there is some vulnerability to attack intermediate values of encryption. Those attacks are EVICT+TIME which evicts an entire cache set and forcing the encryption to fetch data from main memory and PRIME+PROBE which identifies the cache line that evicts the existing data by observing the time. Due to these attacks we want to activate "no-fill" mode. If cache miss occurs the data is taken from main memory and fill into cache which evicts existing data and create hole to PRIME+PROBE attack. So we activates "no-fill" mode that do not fill data into cache from main memory rather directly takes data to processor. By this the time for evicting and filling data in cache will be reduced, which improves performance of encryption. This can be done by using the formula given below:

VirtualProtect(ptr, length, PAGE_NOCACHE, &oldFlags)

Where VirtualProtect() is the function contains arguments such as ptr refers to the starting position, length contains number of bytes to avoid caching, PAGE_NOCACHE is &H200 default value and &oldFlags is used to refer any special bits. These operations are done before encryption starts, so there is no delay in AES mechanism. Then encryption process starts and fetches intermediate values from cache memory. Also fetch some masked values from main memory which is unmasked by the processor. Meanwhile there are no evictions and filling data in cache to prevent from side channel attacks. After the encryption completes the process then deactivate the "no-fill" mode.

CONCLUSION

Side channel attacks do not attack the AES encryption system rather it attacks the intermediate results through cache behavior. Cache behavior can be vulnerable to timing attack and power analysis attack. This vulnerability is due to the difference between cache hits and cache misses. Many countermeasures had been implemented so far, but still there is a need of good one which overcomes previous techniques. This paper proposes an idea which combines two countermeasures. These countermeasures are implemented previously as an individual but there are some loop holes in that process. By combining this we can achieve better performance in security against side channel attacks.

REFERENCES

[1] Hoang, Trang. "An efficient FPGA implementation of the Advanced Encryption Standard algorithm." *Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF), 2012 IEEE RIVF International Conference on*. IEEE, 2012.

[2] Lawson, Nate. "Side-channel attacks on cryptographic software." *Security & Privacy, IEEE* 7.6 (2009): 65-68.

[3] Rebeiro, Chester, Mainack Mondal, and Debdeep Mukhopadhyay. "Pinpointing cache timing attacks on AES." *VLSI Design, 2010. VLSID'10. 23rd International Conference on*. IEEE, 2010.

[4] Meritt, Kevin. "Differential Power Analysis attacks on AES." (2012).

[5] Acıiçmez, Onur, and Çetin Kaya Koç. "Trace-driven cache attacks on AES (short paper)." *Information and Communications Security*. Springer Berlin Heidelberg, 2006. 112-121.

[6] Coron, Jean-Sébastien. "Higher order masking of look-up tables." *Advances in Cryptology–EUROCRYPT 2014*. Springer Berlin Heidelberg, 2014. 441-458.

[7] Fournier, Jacques, and Michael Tunstall. "Cache based power analysis attacks on AES." *Information Security and Privacy*. Springer Berlin Heidelberg, 2006.

[8] Jayasinghe, Darshana, Roshan G. Ragel, and Dhammika Elkaduwe. "Constant time encryption as a countermeasure against remote cache timing attacks."*arXiv preprint arXiv:1403.7293* (2014).

[9] Schramm, Kai, and Christof Paar. "Higher order masking of the AES." *Topics in Cryptology–CT-RSA 2006*. Springer Berlin Heidelberg, 2006. 208-225.

[10] Herath, Udyani, Janaka Alawatugoda, and Roshan G. Ragel. "Software implementation level countermeasures against the cache timing attack on advanced encryption standard." *arXiv preprint arXiv:1403.1322* (2014).

[11] Canright, David, and Lejla Batina. "A very compact "perfectly masked" S-box for AES." *Applied Cryptography and Network Security*. Springer Berlin Heidelberg, 2008.

[12] Tromer, Eran, Dag Arne Osvik, and Adi Shamir. "Efficient cache attacks on AES, and countermeasures." *Journal of Cryptology* 23.1 (2010): 37-71.