บทความวิจัย –วิชาการ

การประชุมวิชาการ งานวิจัย และพัฒนาเชิงประยุกต์ ครั้งที่ 6 การพัฒนาเทคโนโลยีเพื่อให้โลกมีสันติสุข

ECTI-CARD Proceedings 2014, Chiang Mai, Thailand

An implementation of AES algorithm on multicore processors for high throughput

Supachai Thongsuk[1], Prabhas Chongstitvatana, Ph.D. [2]

Department of Computer Engineering Faculty of Engineering, Chulalongkorn University Bangkok, Thailand

E-mail: Supachai.Th@student.chula.ac.th[1], prabhas@chula.ac.th[2]

Abstract

AES (Advanced Encryption Standard) algorithm is a block encryption algorithm established by the U.S. National Institute of Standards and Technology (NIST) in 2001. It has been adopted by many data security systems and now used worldwide. Most of AES implementations are for single-core processors. To achieve high performance for large data, this work proposed an AES algorithm for multi-core processors. Using parallelism inherent in large data, all cores are working concurrently to speed up the task.

***Keywords*:** cryptography; AES; Multicore processor;

1. Introduction

The information security has become an important concern today due to popular use of computers. The AES algorithm is a standard encryption algorithm with a symmetric key. This technique converts an input plain text into a cipher text by repetitions of transformation rounds with cipher key 128, 192 or 256 bits. Each round of AES algorithm performs the calculation using a complex sequence. The computation load is quite high. To achieve high throughput multi-processors are required. This work proposed an AES algorithm based on multicore processors. The experiments are carried out to compare the throughput of multicore processors with a single core processor.

2. Advance Encryption Standard (AES)

AES is a symmetric block data encryption technique that converts input plaintext into cipher text by repetitions of transformation rounds with cipher key 128, 192 or 256 bits. The standard is established by the U.S. National Institute of Standards and Technology (NIST) in 2001 [1]. Each round consists of four different transformations: Sub-byte, Shift-rows, Mix-columns and Add-Round-key procedure.

Table 1. Key length and number of rounds

Key (Bits)	Rounds (Nr)
AES-128	10
AES-192	12
AES-256	14

Block of data input is 128 bits or 4 words in 4x4 square matrix of bytes. The 4x4 matrix of bytes is called the state. The number of iterations depends on the key length. (Table 1)

AES steps

1) Key expansion -- Round keys are derived from the cipher key. Each round requires a separate key block.

2) Initial round -- Each byte of the state is bitwise XOR with the round key.

3) Rounds -- A round consisted of four transformations: SubBytes, ShiftRows, MixColumns, AddRoundKey.

4) Final round -- Three transformations: SubBytes, ShiftRows, AddRoundKey.

Each transformation is explained as follows:

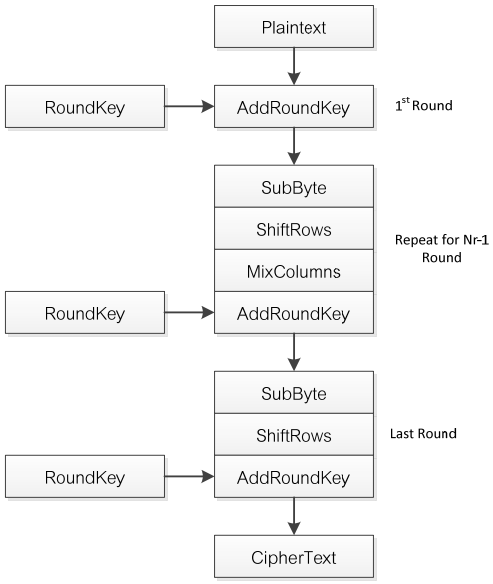

Fig. 1. AES Encryption flow

A. Sub-byte

Each byte a_{ij} in the state is replaced by S-box, $S(a_{ij})$. The S-box is derived from the multiplicative inverse over GF2

B. Shift-rows

Each byte is shifted cyclically by row. First row is shifted by 0 offset. Second row is shifted by one offset, third and fourth rows by two and three offset.

C. Mix-columns

Each column of the state is combined using an invertible linear transformation. Each column is multiplied by a fixed matrix

$$\begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

This is a multiplication modulo $x^4 + 1$ with a polynomial

$c(x) = 0x03\ x^3 + x^2 + x + 0x02$.

D. Add-Round-Key

In this step each byte of input state is combined with the round key by using XOR algorithm that round key is generated from the cipher key using a key schedule algorithm.

The detailed description of the AES algorithm can be found in [2].

3. Multicore Processor

Multicore processor architecture [3] has two or more computing units contained in single physical processor. It can run multiple instructions at the same time. There are three types of different system architecture using multiple computing units. Heterogeneous system contains different types of cores. Asymmetric Multi-Processing (AMP) contains two or more processors of the same type which either run different operating systems or separate copies of the same operating system. Symmetric Multi-Processing (SMP) has two or more CPUs of the same type like an Asymmetric Multi-Processing but all of which run under the same operating system, Fig 2. In this work, we have used S2 multicore processor to perform encryption and decryption on a shared memory. We perform the experiment using a 32-bit multicore processor simulator with a simple C-like language as a programming tool.

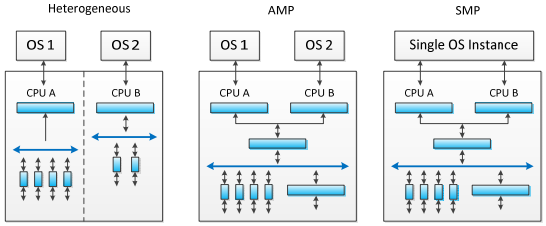

Fig. 2. Multicore processing system architectures

A. S2 multi-core processor

S2 [4] is 32-bit CPU simulator that allows to setup 2 to 8 cores simulation. The processor has three-address instruction format. The instruction set is divided into 4 different groups: arithmetic, logic, control and data. (Table 2)

Table 2. Instruction type of S2 multi-core processor

Instruction type	Operation name
Arithmetic	add sub mul div mod
Logic	and or xor eq ne lt le gt ge shl shr
Control	jmp jt jf jal ret
Data	ld st push pop

B. RZ Language

RZ [5] is a small programming language that is similar to a subset of C language. The source code of AES is developed and tested on S2 simulator. The simulator is accurate to the clock cycle. The compiler and simulator are available publicly.

4. Related work

In this section, we sum up several related work of AES implementation on CPU and GPU. Many work present GPU and hardware implementation.

Barnes [6] implemented AES algorithm on a multicore processor using fork and pthread function to improve throughput. They achieve 6637Mb/s on 32- core processor with pthread architect. Many work have used GPU processors. Manavski [7] used Nvidia GeForce 8800GTX to compute AES with CUDA technology with 32-bit processor that can perform sub-byte 4 bytes in the same instruction. It also works on add-round-key. GPU will process sub-byte and add-round-key 4 times per round by CPU and need 16 times for sub-byte and add-round-key. He achieved 8.28Gbit/s throughput on 8MB of data with 128 bits AES. In 2010, Nhat-Phuong Tran [8] presented a work that increase the size of AES block from standard 16 bytes to reduce the overhead of the data transfer in the memory. The extended block size can increase the encryption speed by 25% to 28% and 603% to 853% for GPU implementation. However it does not work well for small size data. Huang Chang Lin and Tai [9] presented 32 bits AES on Xilinx FPGA (Spartan-3 XC3S200) with throughput 647 Mbs. The current processor has a special extension of the instruction set for AES, for example Intel Core i3/i5/i7 CPUs support AES-NI instruction set extensions, throughput can be over 700 MB/s per thread. [10]

This work proposes an AES implementation for multicore processors using division of data of each core. The data is split into 16 bytes block in the main memory. Each block of data is fed to each core to run AES encryption. At the end, each ciphered block is merged with the ciphered block from another core. (See Fig. 3)

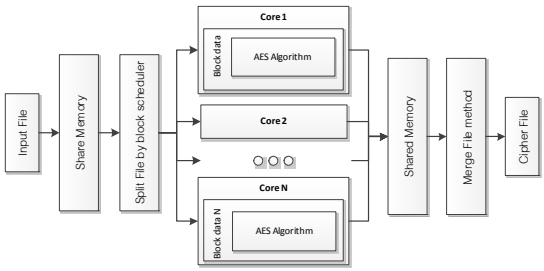

Fig. 3. AES algorithm on multicore processor

5. Experimental result and analysis

The experiments are carried out to compare AES program on single-core and multi-core processors. The flow of data collection is shown in Fig.4. All experiments are performed on the same condition using the same program (except for the number of core) and hardware. The multicore simulator can monitor the number of instructions executed in each core. Each instruction is assumed to take one clock cycle, except for the memory access which takes two cycles. When more than one core access the memory, only one core is granted the access. Other core will be stalled to wait for the first core to finish. Instruction fetch is assumed to have no conflict. The graph in Fig.5 shows the experiment with a multi-core configuration. The results show the speed of AES encryption is increasing when more cores were used. However , the number of cycle that has memory conflict also increases.. Therefore it does not have any further speed up when using 3 and 4 cores. We conclude that for this task, two-core is the best configuration. (See Table 3, 4 and Fig.5).

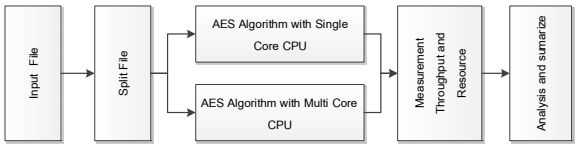

Fig. 4. Test and experiment methodology for this work

Table 3. The time in CPU cycle to complete the task

Data (Byte)	Core 1	Core 2	Core 3	Core 4
32	37189	26917	-	-
64	69901	48358	45664	49924
128	135325	91428	85669	90828
256	266173	177448	175088	172687

Table 4. The total number of cycle of memory stall (please note that this is the sum of the stall of all cores hence it may be larger than the number of cycle time of CPU)

Data (Byte)	Core 1	Core 2	Core 3	Core 4
32	10346	19459	-	-
64	19444	35475	62383	81733
128	37640	67695	119257	150909
256	74000	131995	222702	289148

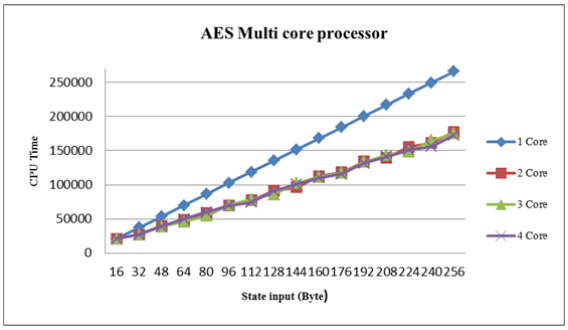

Fig. 5. The result of AES on single core and multi-core processor

6. Conclusion and future work

This paper shows that the throughput of AES on multicore process increase with the number of core used. From this result we can use multicore processors to secure large data. For a future work, GPU implementation is very promising. The cooperation between GPU and CPU to work together on AES encryption will be very important.

References

[1] NIST. Announcing the advanced encryption standard (AES), FIPS 197. Technical report, National Institute of Standards and Technology, November 2001.

[2] Daemen, Joan; Rijm Daemen, Joan; Rijmen, Vincent , "AES Proposal: Rijndael". National Institute of Standards and Technology, 2003.

[3] Atsushi Hasegawa.Renesas'Multi-Core Technology [Online]. 2013. http://www.renesas.com/products/mpumcu/multi_core/child/multicore.jsp. Retrieved 28 April 2013.

[4] http://www.cp.eng.chula.ac.th/~piak/project/s2/s23.htm

[5] http://www.cp.eng.chula.ac.th/~piak/project/s2/rz37.htm

[6] Barnes, A., Fernando, R., Mettananda, K., Ragel, R., "Improving the throughput of the AES algorithm with multicore processors," 7th IEEE International Conference on Industrial and Information Systems (ICIIS), 2012, pp.1-6.

[7] Manavski, S.A.; "CUDA Compatible GPU as an Efficient Hardware Accelerator for AES Cryptography," IEEE International Conference on Signal Processing and Communications, 2007. ICSPC 2007, pp.65-68, 24-27 Nov. 2007.

[8] Nhat-Phuong Tran; Myungho Lee; Sugwon Hong; Seung-Jae Lee, "Parallel Execution of AES-CTR Algorithm Using Extended Block Size," Computational Science and Engineering (CSE), 2011 IEEE 14th International Conference on , pp.191,198, 24-26 Aug. 2011.

[9] Gielata, A.; Russek, P.; Wiatr, K.; "AES hardware implementation in FPGA for algorithm acceleration purpose," International Conference on Signals and Electronic Systems, 2008. ICSES '08, pp.137-140, 14-17 Sept. 2008

[10] McWilliams, Grant (6 July 2011). "Hardware AES Showdown - VIA Padlock vs Intel AES-NI vs AMD Hexacore". http://grantmcwilliams.com/tech/technology/item/532-hardware-aes-showdown-via-padlock-vs-intel-aes-ni-vs-amd-hexacore. Retrieved 10 February 2014.