Efficient Parallel Simeck Encryption with GPGPU and OpenCL

Taehwan Park[1], Hwajeong Seo[2], Md. Al-Amin Khandaker[3], Yasuyuki Nogami[3], and Howon Kim[1*]
[1]Department of Electrical and Computer Engineering, Pusan National University, South Korea
E-mail: {pth5804, howonkim}@pusan.ac.kr
[2]Department of IT, Hansung University, South Korea
E-mail: hwajeong@hansung.ac.kr
[3]Graduate School of Natural Science and Technology, Okayama University, Japan
E-mail: khandaker@s.okayama-u.ac.jp, yasuyuki.nogami@okayama-u.ac.jp

***Abstract*-- Simeck family block cipher was proposed in CHES 2015. It is a kind of lightweight block cipher provide various block and key size. In this paper, we proposed efficient parallel implementation of Simeck with GPGPU by using OpenCL and present performance of Simeck parallel implementation.**

I. INTRODUCTION

In these days, there are a lot of ICT application services using Graphic Processing Unit (GPU) for processing massive data and lightweight block cipher for providing data confidentiality on Internet of Things (IoT) device environment. For using GPU, there are two programming methods such as OpenCL, and CUDA (Compute Unified Device Architecture). OpenCL is the open standard for parallel computing of heterogeneous devices such as CPU, and GPU. However, CUDA is a parallel programming for only NVIDIA GPU environment. In the case of lightweight block cipher, Simeck family block cipher was proposed in CHES 2015. It is ARX (Addition, Rotation, XOR) operation based block ciphers and provide various block and key size. In this paper, we proposed efficient parallel Simeck encryption methods on GPGPU (General Purpose Graphic Processing Unit) environment by using OpenCL. The rest of this paper is organized as follows. In section 2, we review on related works on Simeck, and OpenCL. In section 3, we proposed methods for efficient parallel Simeck encryption on GPGPU environment by using OpenCL and report performance of our implementation. Finally, we conclude the paper on section 4.

II. RELATED WORKS

A. Simeck Family Block Cipher

The Simeck family block ciphers were proposed in CHES 2015[1]. The round function of the Simeck can be written as $R_{k_i}(l_i, r_i) = (r_i \oplus f(l_i) \oplus k_i, l_i)$, $f(x) = (x \odot (x \lll 5) \oplus (x \lll 1)$ The round function with function f consists of AND($\odot$), Rotation Left($\lll$), and XOR($\oplus$). Simeck family block ciphers consist of Simeck32/64, Simeck48/96, Simeck64/128 with 32, 36, and 44 rounds, respectively.

B. Related works on Simeck

Park et al. [2] proposed efficient implementation methods of Simeck by using AVR 8-bit assembly instructions on 8-bit AVR environment, and efficient implementation methods of Simeck block ciphers on 16bit MSP430 environment [3] based on MSP430 assembly instructions are proposed.

C. OpenCL

OpenCL (Open Computing Language) is open standard and parallel processing programming framework for heterogeneous parallel processor environment. Khronos Group which made OpenGL standardize OpenCL with main processor vendors and multi-core software vendors such as AMD, Apple, IBM, Intel, NVIDIA, and etc. In 2015, they announced Vulkan API with OpenCL 2.1 specifications, the latest version of OpenCL is version 2.2 which is announced in 2016.

D. Related works on OpenCL

There are some related works on efficient implementation of cryptographic algorithms by using OpenCL. Gao, Sanshan, et al. [4] proposed a low-cost heterogeneous computing methods and performance of proposed methods for the RSA decryption on Intel Core i7-4770R processor, and Intel Iris Pro Graphics 5200 environment by using OpenCL v1.2. Velea, Radu, et al. [5] proposed performance of parallel implementation of ChaCha20 stream cipher on Intel 2600K CPU, and Radeon HD 6850 by using OpenMP and OpenCL. Heinemann, Colleen, et al. [6] proposed OpenCL and CUDA software implementation of encryption and decryption algorithms for IPsec VPNs such as AES, and TwoFish block ciphers. They measured ECB, and CTR operation mode of two block ciphers with OpenCL and CUDA on 3 platforms including NVIDIA Geforce, and AMD Radeon GPU by using Rich Multimedia(RMM) test data files. Their performances of various ciphers in various mode of operation in IPSecs and VPNs were increased from 2.9 to 8.7 times.

III. PROPOSED METHOD

A. Data Parallel Model Based Implementation

OpenCL supports data parallel model such as Figure 1. We implemented Simeck according to OpenCL data parallel model for efficient data parallel processing by using all of processing

units in GPU. For data parallel processing, we used OpenCL API *clEnqueueNDRangeKernel()* which support to enqueuer the kernel command for data parallel processing. In .cl file, we implemented Simeck block cipher as __kernel function for OpenCL data parallel processing by using C language. We used *#pragma unroll* for loop unrolling in .cl file for efficient OpenCL implementation. For reducing memory access time, we used local memory on GPGPU by using "*__local*" indicator and *get_global_id*() function for getting global work item ID and loading data from global memory to local memory. For efficient rotation operation, we made rotation operation as a function and its parameters are operand data and rotation bit parameter *r*.

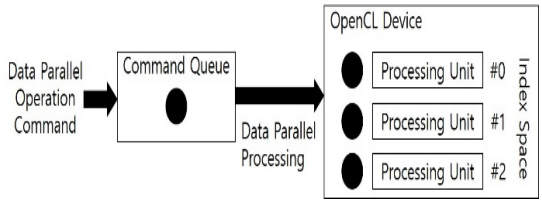

Fig. 1. OpenCL Data Parallel Processing Architecture

IV. Evaluations

A. Experiment

Experimental environment is MacBook Pro (15-inch, Mid 2015) which has Intel i7-4870HQ Processor (@2.5GHz), Intel Iris Pro Graphics 5200 (@1.2GHz) GPU, and AMD Radeon R9 M370X GPU. For evaluating performance of proposed method, we measured performance on AMD Radeon R9 M370X GPU. AMD Radeon R9 M370X GPU has 256 working group.

B. Performance Analysis

For performance Analysis, we measured average performance during 10,000 times at each working group. At each working group, each Simeck ciphers encrypt *16, 24, 32-bit*# of working group* respectively. Table 1. and Figure 2. describe performances (unit: Mbps) of Simeck block ciphers which is adapted proposed method. Performance is rapidly increased since the number of working group is 256. Because, evaluation environment is AMD Radeon R9 M370X GPU and it has 256 working group, so if the number of working group is multiple of 256, we can use all of working group on GPU.

TABLE I PERFORMANCE OF SIMECK USING OPENCL(UINIT: MBPS)

Working group / cipher	64	128	256	512	1024
Simeck32/64	2.02	3.68	8.12	13.04	30.81
Simeck48/96	1.76	3.70	6.91	13.54	27.49
Simeck64/128	1.63	3.38	6.47	13.47	25.25

V. Conclusion

In this paper, we proposed OpenCL data parallel model based efficient implementation methods of Simeck block cipher and performance according to different the number of working group on GPU environment. Performances of proposed method are increased since the number of working group is 256 or more, because target GPU has 256 working group. If proposed methods are used with multiple number of GPU's working group, it can accelerate performance of block cipher encryption.

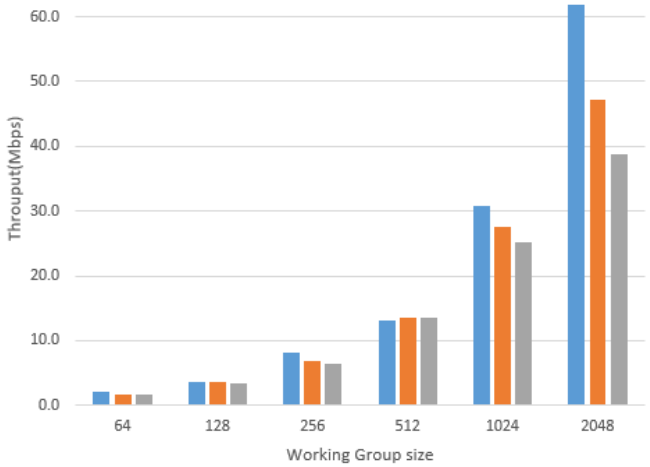

Fig. 2. Performance of Parallel Simeck encryption using OpenCL (Blue: Simeck32/64, Orange: Simeck48/96, Grey: Simeck64/128)

Acknowledgment

This work of Taehwan Park, and Howon Kim was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.017-0-01791, Development of security technology for energy platform and device). This work of Hwajeong Seo was was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (No. NRF-2017R1C1B5075742).

References

[1] Yang, Gangqiang, et al. "The simeck family of lightweight block ciphers." International Workshop on Cryptographic Hardware and Embedded Systems. Springer Berlin Heidelberg, 2015.

[2] Park, Taehwan, et al. "Efficient Implementation of Simeck Family Block Cipher on 8-Bit Processor." Journal of information and communication convergence engineering 14.3 (2016): 177-183.

[3] Park, Taehwan, et al. "Efficient implementation of simeck family block cipher on 16-bit MSP430." Ubiquitous and Future Networks (ICUFN), 2017 Ninth International Conference on. IEEE, 2017.

[4] Khronos Group, OpenCl – The open standard for parallel programming of heterogeneous systems, https://www.khronos.org/opencl/

[5] Gao, Sanshan, et al. "Cost-Efficient Parallel RSA Decryption with Integrated GPGPU and OpenCL." Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld), 2016 Intl IEEE Conferences. IEEE, 2016.

[6] Velea, Radu, et al. "Performance of parallel ChaCha20 stream cipher." Applied Computational Intelligence and Informatics (SACI), 2016 IEEE 11th International Symposium on. IEEE, 2016.

[7] Heinemann, Colleen, et al. "OpenCL and CUDA software implementations of encryption/decryption algorithms for IPsec VPNs." Electro Information Technology (EIT), 2016 IEEE International Conference on. IEEE, 2016.