Exploring the Optimum Buffer Size of an Emerging Stream Cipher Engine

Masa-aki Fukase[1], Non-member and **Tomoaki Sato**[2],

ABSTRACT

One of crucial points for further improving ubiquitous network is to enhance temporary security without relying on permanent network infrastructure. Since a practical solution to provide such ubiquitous security is a safety aware, high-performed single chip processor, a multimedia stream cipher engine has been exploited. In order to keep security, usability, speed, and power consciousness, the stream cipher engine takes a compact multicore architecture. Each core implements a double cipher scheme that covers RAC (random addressing cryptography) and data sealing. The double cipher is microarchitecture-based, software-transparent hardware cryptography that offers the protection of the whole data with negligible hardware cost and moderate performance overhead. Stream cipher engine chips have been developed by using 0.18-μm standard cell CMOS technologies. Through the rough evaluation of those chips, it has emerged that streaming buffer size is crucial for prospective specifications. In order to achieve sophisticated design strategy, this paper evaluates in detail the buffer size dependency of power dissipation, clock speed, running time, and throughput, focusing on the latest version of the stream cipher engine chip. From the tradeoff between these specificative factors, the guideline of optimum buffer size is made clear in view of safety and performance for ubiquitous computing.

Keywords: Ubiquitous Security, Hardware Cryptography, Cipher Streaming, Processor Engine, CMOS Chip

1. INTRODUCTION

Although diversity is inevitable for the usability and functionality of ubiquitous network, it also causes notorious security issues like insecurity, security threat or illegal attack such as tapping, intrusion, pretension. The worldwide diversity vs. security threat is really the two-faced characteristics of ever growing ubiquitous network [1]. On the other hand, massive quantity of multimedia information over the ubiquitous network is also crucial for the interaction between ubiquitous devices and human being. However, massive data is very awkward for regular techniques used in ubiquitous devices to satisfy overall demands for not only security but also usability, speed, and power consciousness, because they mainly depend on embedded software.

One of practical solutions to fulfill overall demands for ubiquitous environment is a single VLSI chip processor [2]. Especially, a sophisticated processor engine that intensively executes cipher streaming of multimedia data will be very effective. The point of this concept is not to explore an extremely strong cipher scheme, but to develop a real chip for ubiquitous security. It is temporary security with practically enough cipher strength and without relying on permanent network infrastructure. According to this scope, a hardware cryptography-embedded processor named RAP (random addressing-accelerated processor) was exploited. [3]. Then, the SISD (single instruction stream single data stream) mode of RAP has been improved by a stream cipher engine that executes SIMD (single instruction stream multiple data stream) mode cipher [4, 5]. This has caused the drastic improvement of throughput. The stream cipher engine has high potential for a safety aware, high-performed single chip ubiquitous processor.

The hardware cryptography run on RAP and the previous stream cipher engine has been called RAC (random addressing cryptography) [6]. RAC is based on directly connecting the output of a built-in RNG (random number generator) to the access line of data cache. The direct connection makes the transfer of data from register file to data cache at random. In spite of RACfs features for ubiquitous security, it unfortunately lacks the hidability of data itself, because RAC does not change the content of data.

In order to compliment the cryptographic weakness of RAC, the latest version of the stream cipher engine has adopted a double cipher scheme that covers RAC and data sealing [7]. The double cipher is microarchitecture-based, software-transparent hardware cryptography that offers the protection of the whole data with negligible hardware cost and moderate performance overhead. Through the rough evaluation of the latest version chip by using a 0.18-μm standard cell CMOS technology, it has emerged that buffer size is crucial for prospective specifications.

Manuscript received on August 1, 2009 ; revised on November 10, 2009.

[1] The author is with Graduate School of Science and Technology, Hirosaki University, Hirosaki 036-8561, Japan Tel.+81-172-39-3630, Fax:+81-172-39-3645, E-mail: slfuka@eit.hirosaki-u.ac.jp.

[2] The author is with C&C Systems Center, Hirosaki University, Hirosaki 036-8561, Japan Tel.+81-172-39-3723, Fax:+81-172-39-3722, E-mail: tsato@cc.hirosaki-u.ac.jp.

In order to achieve the sophisticated design strategy of the emerging stream cipher engine, this paper evaluates more in detail the buffer size dependency of power dissipation, clock speed, running time, and throughput, applying Synopsys Design Compiler to the latest version of the stream cipher engine chip. From the tradeoff between these specificative factors, the guideline of optimum buffer size is made clear in view of safety and performance for ubiquitous computing [8].

2. RELATED WORK

Stream cipher engine chips so far developed are summarized in Table 1. Design environments are basically 0.18-μm standard cell process technologies and Synopsys and Cadence tools supported by VDEC. Power consumption is a rough approximation or the maximum power derived from the summation of mean value of every gate. It does not take into account of switching condition. Clock frequency was derived from the maximum critical path. This is an address line between RNG and register file in the case of 3^{rd} version.

Table 1: *Specifications of the stream cipher engine chips*

Development version			1st	2nd	3rd
Architecture	Cipher		RAC		Double cipher
	Register file		2byte × 64word		2byte × 512word
	Data cache		2byte × 64word		2byte × 512word
Chip	Process		HITACHI 0.18μm CMOS		ROHM 0.18μm CMOS
	Area	Die	2.8-mm square		2.5mm × 5.0mm
		Core	1.38-mm square		1.62mm × 4.28mm
	Voltage		1.8V (I/O 3.3V)		
	Power		73mW	137mW	536mW
	Clock		400MHz	666MHz	200MHz
	Throughput				0.2GOPS
	Current status		Synthesis		Chip implementation

In view of architectural characteristics, the 1^{st} version of the stream cipher engine fixes the role of double core. Core 1 is distinguished for encryption and core 2 is dedicated for decryption [4]. On the other hand, the 2^{nd} version [5] and the 3^{rd} version [7] do not fix the role of each core. Focusing on the 3^{rd} version stream cipher engine chip, the amendment is as follows.

(i) Implementation of the double cipher scheme: This complements RACfs shortcoming. Hardware units of the double cipher scheme enhance the security of data information as a whole with negligible hardware cost and moderate performance overhead.

(ii) The scale up of register file, data cache, and RNG (random number generator): While the buffer size is extremely limited in the case of the 1^{st} and 2^{nd} versions in order to make sure the development, the

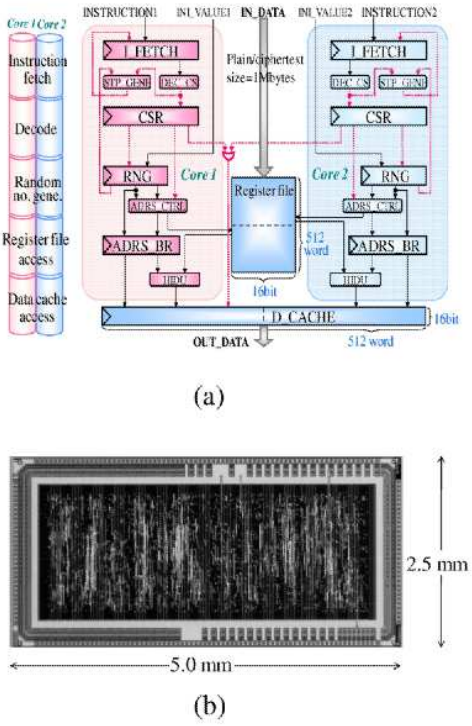

Fig.1: *The 3^{rd} version stream cipher engine chip (a) Microarchitecture (b) Die photo*

3^{rd} version stream cipher engine chip extends the size of a streaming buffer for practical use.

Fig. 1 shows the microarchitecture and die photo of the 3^{rd} version stream cipher engine chip. This is composed of two symmetric cores in order to do bidirectional communication and to achieve power conscious high-speed performance. Each core executes SIMD mode double cipher coeds to do streaming cipher. These are a random store code rsw and a random load code rlw.

Register file plays the role of streaming buffer. It buffers a block of external data that is plaintext or cipher text. The transfer of the block to the register file is assumed to be DMA (direct memory access) mode, though it is not concern in this study. Register file and data cache are logically divided. The logical separation is distinguished by simply modifying upper bits of address.

The built-in RNG is formed by LFSR (linear feedback shift register). Although this is due to restricted hardware quantity, LFSR falling into the category of M-sequence requires trivial additional chip area and power dissipation. Also, a tiny n-bit LFSR produces the huge 2^n random numbers. A 1K-, 1M-, 1G-byte length texts require only 10-, 20-, and 30-bit LFSR respectively. This is sufficient for practical use in producing cipher keys.

HIDU on data lines in Fig. 1 (a) is a hidable unit that does data sealing. This is a simple wired logic in order to keep usability, speed, and power consciousness as well as security. Double cipher pro-

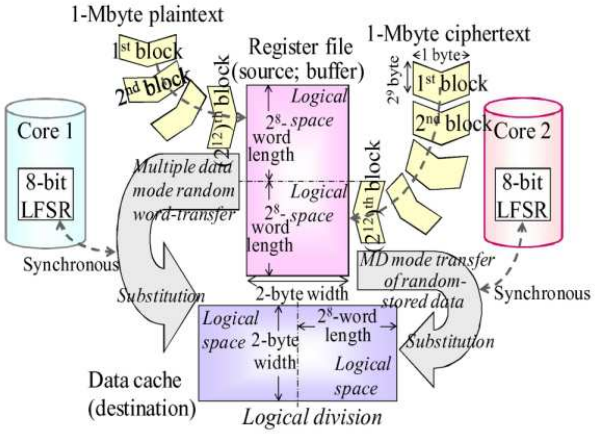

Fig.2: *Data transfer mechanism of the 3^{rd} version stream cipher engine*

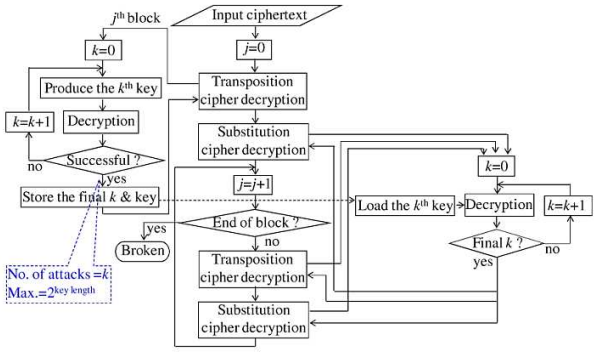

Fig.3: *Measurement of cipher strength*

ceeds according to following microarchitecture-based, software-transparent mechanism.
(i) Make RNG output integer specify a register file address.
(ii) Synchronize a data cache address with the current clock count.
(iii) Store the specified register filefs content to the synchronized data cache address. During the storage, a hidable function works for the plaintext block.
The resultant content stored in the data cache is the encryption of the register filefs content.

3. CIPHER STREAMING BUFFER SIZE VS. SPECIFICATIVE FACTORS

The buffer size dependency of power dissipation, clock speed, running time, and throughput is evaluated by varying the size of the register file and data cache of the latest 3^{rd} version stream cipher engine chip. The reason why the streaming buffer size is targeted is because it has emerged that buffer size is crucial for the safety and performance of ubiquitous computing through the rough estimation of various performance factors [7].

3.1 Cipher Strength

The buffer size dependency of cipher strength is evaluated from the analysis of a round robin attack, which repeats the attack, decipher, break, or crack of a ciphertext. Before the discussion of the round robin attack for the double cipher, let us firstly clear how the ciphertext is treated by the stream cipher engine. Fig. 2 explains the data transfer mechanism within the stream cipher engine. Especially, this focuses on the interaction between block, register file, data cache, and LFSR.

In the execution of the double cipher, plain and cipher texts are divided into blocks as is similar to AES (Advanced Encryption Standard) and DES (Data Encryption Standard). The double cipher can treat a full text if we assume an ideal buffer that immediately stores a full text. However, the practical buffer built in the stream cipher engine is register file whose space and speed are limited. So, the actual target of the double cipher by the stream cipher engine is a block. Each block is stored in register file by DMA transfer. Then, the block transfer from register file to data cache follows multiple data stream. As shown in Fig. 1 (a), the output of RNG is used for both random addressing and data sealing. The sequence like this results in the formation of a cipher in data cache.

The general relation on the block, RNG, register file, and data cache is given by

Block size $\leq$ RNG output length
= register file's logical space size
= data cache's logical space sizes (1)
No. of blocks
= plain or cipher text size / block size. (2)

Here, the size is the product of word length and width in byte. Generally, block and register file have different length and width, because block changes its form depending on communication and cipher systems. In order to correspond to flexible block, register file must be reconstructive, which is not the subject of this study. Thus, register file stores a block, and the following relation is derived from Eq. (1).

Block size = register file size = data cache size (3)

Fig. 3 shows the round robin attack against the double cipher by the 3^{rd} version stream cipher engine. j is the number of block. k is the number of attack. The strength or the degree of enduringness is reasonably measured by the time needed in the attack and the discovery of a true key. Thus, the cryptographic strength is the product of the time for decryption and the maximum number of trials. Since RNG made by n-bit LFSR outputs 2^n-length random numbers, the strength of the double cipher for a round robin attack is given by

Cipher strength
= max. time of the round robin attack
= no. of the round robin attack
$\times$ time for decryption
$= 2^n \times 2^{\text{buffer width}} \times$ time for decryption. (4)

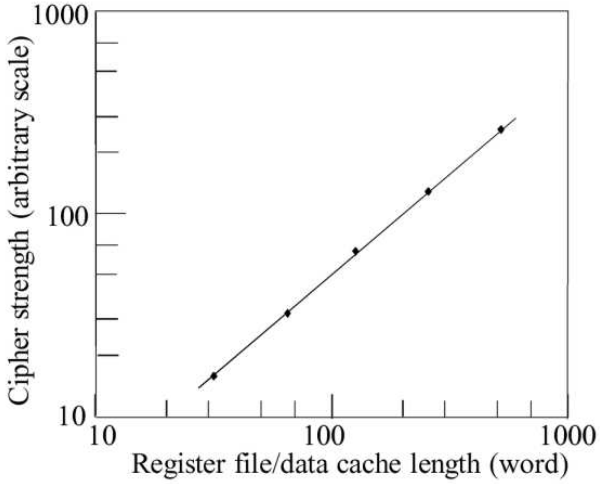

Fig.4: *Cipher strength vs. register file length of the 3^{rd} version stream cipher engine*

Although the length and width are both crucial for cipher strength from Eq. (4), the width is usually fixed to byte width because ubiquitous media takes the form of byte structured stream. Thus, the buffer length is a critical factor. Fig. 4 shows register file length dependency of the cipher strength of the 3^{rd} version stream cipher engine. It is clear from this figure that cipher strength increases as the increase of buffer length.

3.2 Power, Clock, Running Time, and Throughput

The buffer size dependency of power dissipation and clock speed is evaluated by using Synopsys Design Compiler. Fig. 5 shows power dissipation vs. register file length of the 3^{rd} version of the stream cipher engine chip. Power dissipation is a rough approximation or the maximum power derived from the summation of mean value of every gate. It does not take into account of actual switching condition in running a test bench. Thus, occu-pied area is also shown in Fig. 5. Fig. 6 shows clock speed vs. register file length of the 3^{rd} version stream cipher engine chip. Clock frequency is derived from the timing analysis of the netlist.

The buffer size dependency of running time and throughput is evaluated as shown in Fig. 7. Since the double cipher process is SIMD mode, running time and throughput are derived as follows.

Running time= $t \times m$ (5)

t = {instruction pipeline degree

+ (blockfs word length -1) }

$\times$ clock cycle time (6)

Blockfs word length =

register filefs logical space length (7)

m = no. of blocks

= full textfs size / block size. (8) Here, the time to be taken by rewriting register file is neglected as the

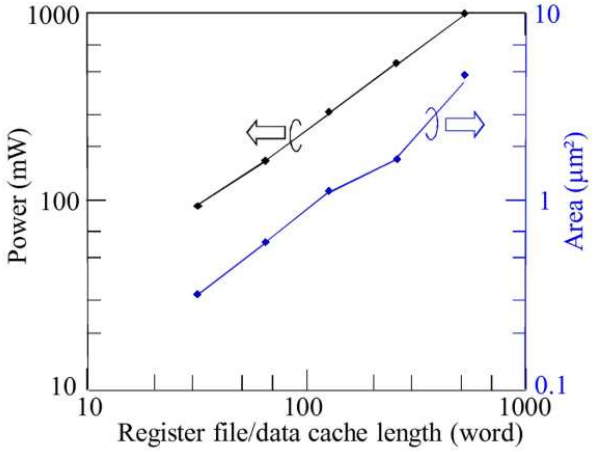

Fig.5: *Power vs. register file length of the 3^{rd} version stream cipher engine*

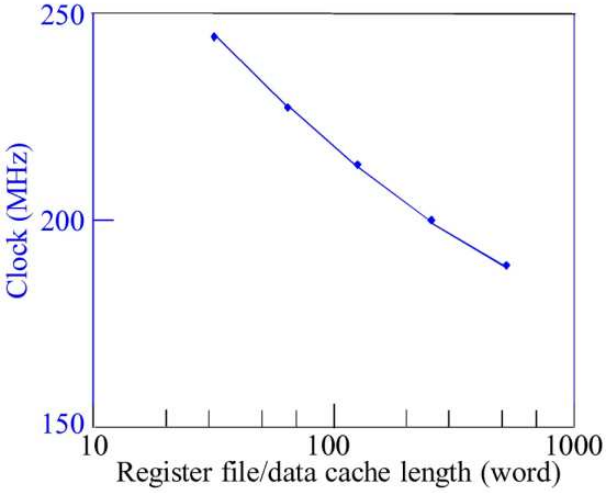

Fig.6: *Clock speed vs. register file length of the 3^{rd} version stream cipher engine*

first approximation in the evaluation of the running time. This is because DMA transfer normally taken in rewriting register file is independent on instruction execution.

Fig. 8 shows running time. It is almost independent on register file length. By using the running time, throughput is given as follows in OPS (operations per second).

Throughput =

no. of word transfers / running time. (9) Similarly to running time, the register file dependency of throughput is also weak. Such tendency as the increase of register file is due to the balance of the increase of block-processing time t and the decrease of the number of blocks, m. This makes running time almost constant.

3.3 Discussion

As is clear from Figs. 4 and 5, tradeoff exists between the cipher strength and power consumption of

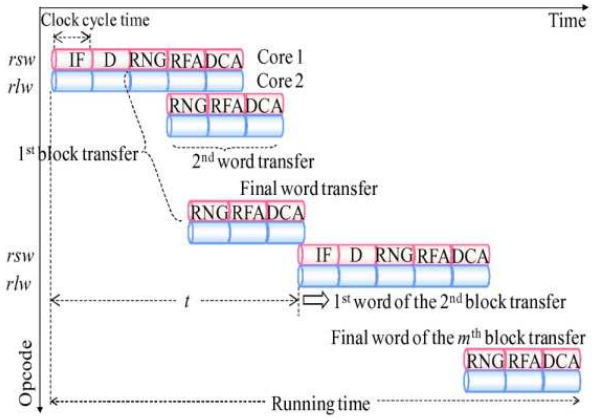

Fig.7: *Derivation of running time and throughput*

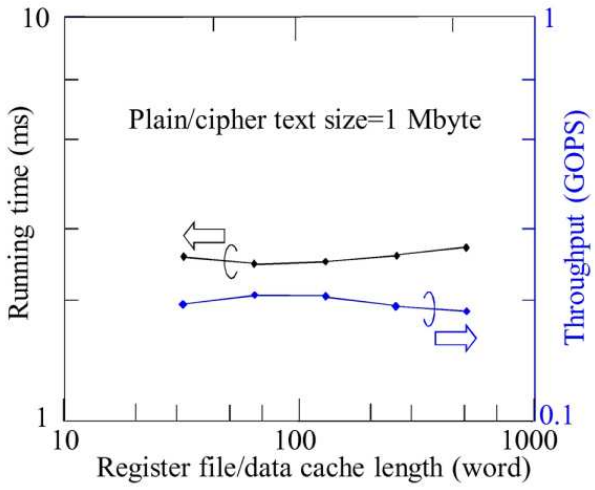

Fig.8: *Running time, throughput vs. register file length of the 3rd version stream cipher engine*

the stream cipher engine. In view of cipher strength, longer buffer size is desirable from Fig. 4. However, more power is consumed as the increase of register file from Fig. 5. Considering the power dissipation of mobile processors is usually less than 1 watt, the register file size should be at most 512 words from Fig. 5. This validates the 512-word register file and data cache of the 3^{rd} version stream cipher engine.

On the other hand, the throughput of the stream cipher engine is almost independent on register file length from Fig. 8. The throughput of 0.2-GOPS (Giga OPS) is sufficient for multimedia cipher streaming. Actually, it is allowable for video format, because running time used for cipher streaming occupies very small portion of video processing time. For example, let us take into account of the video processing of usual portable devices with QVGA (Quarter Video Graphics Array) format. Since QVGAfs resolution is

320 × 240 pixel/frame = 0.23 Mbyte/frame (10)

1-Mbyte text discussed in Fig. 8 forms 4.3 flames. This takes 4.3/30=143 ms in video processing, because video frame rate is 30 frame/s. Then, the time taken for 1-Mbyte cipher streaming is 2.6 ms from Fig. 8. It is only 1.8

Text size = 0.23 (Mbyte/frame)

× 30 (frame/s) × 60 (s)= 414 Mbytes. (11) Thus, 1-minute video takes

2.6 ms × 414 ≈ 1.1 s (12) for cipher streaming, because running time is proportional to text size from Eqs. (5)-(8).

4. CONCLUSION

This paper has explored optimum streaming buffer size in order to achieve practical safety and performance for ubiquitous computing. The tradeoff is made clear between the cipher streaming buffer length and power dissipation, clock speed, running time, and throughput. The critical factor is power dissipation. The limit of the scale up of register file and data cache is at most 512 words in view of power limitation.

The next steps of this study are as follows.

(i) Power and timing evaluation more in detail by using place and route tools after netlists and delay information.

(ii) More accurate analysis of the running time by taking into account of the time taken for block transfer to register file and.

(iii) Improvement of the double cipher strength by practical modification of hardware mechanism.

5. ACKNOWLEDGMENT

This work is supported by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and Cadence Design Systems, Inc.

References

[1] M. Satyanarayanan, “Privacy: The Achilles Heel of Pervasive Computing?h *IEEE pervasive,* Vol. 2, No. 1, pp. 2-3, Jan.-Mar. 2003.

[2] A. Jerraya, H. Tenhunen, and W. Wolf, “Multipro-cessor Systems-on-Chips,h *Computer Magazine,* Vol. 38, No. 7, 2005.

[3] M. Fukase and T. Sato, “Power Conscious Endeavor in Processors to Speed Up Random Sampling,” *Proc. of SCI 2003,* Vol. V, pp. 111-116, Jul. 2003.

[4] M. Fukase, H. Takeda, R. Tenma, K. Noda, Y. Sato, R. Sato, and T. Sato, “Development of a Multimedia Stream Cipher Engine,h *Proc. of ISPACS 2006,* pp. 562-565, Dec. 2006.

[5] M. Fukase and T. Sato, “A Stream Cipher Engine for Ad-hoc Security,h *Proc. of CISf2007,* pp. 902-906, Dec. 2007.

[6] M. Fukase, A. Fukase, Y. Sato, and T.@Sato, “Cryptographic System by a Random Addressing-Accelerated Multimedia Mobile Processor,h *Proc. of SCI 2004,* Vol. II, pp. 174-179, Jul. 2004.

[7] M. Fukase, K. Noda, and T. Sato, "Emerging Hardware Cryptography and VLSI Implementation,h *Proc. of ISPACS 2008,* pp. 445-448, Feb. 2009.

[8] M. Fukase, Y. Ohsumi, and T. Sato, "Exploring the Optimum Buffer Size of an Emerging Stream Cipher Engine,h *Proc. of ECTI-CON 2009,* pp. 607-610, May 2009.

Masa-aki Fukase received the B.S., M.S., and Dr. of Eng. Degrees in Electronics Engineering from Tohoku University in 1973, 1975, and 1978, respectively. He was Research staff member from 1978 to 1979 at The Semiconductor Research Institute of the Semiconductor Research Foundation. He was Assistant Professor from 1979 to 1991, and Associate Professor from 1991 to 1994 at the Integrated Circuits Engineering Laboratory of the Research Institute of Electrical Communication, Tohoku University. He has been Professor of computer engineering since 1995 at the Faculty of Science and Technology, Hirosaki University. He has been the Director of the Hirosaki University C&C Systems Center since 2004. He is also Representative of Hirosaki University R&DC of Next Generation IT Technologies since 2008. His current research activities are mainly concerned with the design, chip implementation, and application of power conscious highly performable VLSI processors.

Tomoaki Sato received the B.S. and M.S. degrees from Hirosaki University, Japan, in 1996 and 1998 respectively, and the Ph.D. degree from Tohoku University, Japan, in 2001. From 2001 to 2005, he was an Assistant Professor of Sapporo Gakuin University, Japan. Since 2005, he has been an Associate Professor of Hirosaki University. His research interests include VLSI Design, Computer Hardware, and Computer and Network Security.