FPGA-based High Throughput XTS-AES Encryption/Decryption for Storage Area Network

Yi Wang*†, Akash Kumar* and Yajun Ha†
*School of Electrical & Computer Engineering, National University of Singapore, Singapore
Email: estelle.ywang@gmail.com, akash@nus.edu.sg
†Institute for Infocomm Research (I²R), A*STAR, Singapore
Email: {wangy,ha-y}@i2r.a-star.edu.sg

***Abstract*—The key issue to improve the performance for secure large-scale Storage Area Network (SAN) applications lies in the speed of its encryption/decryption module. Software-based encryption/decryption cannot meet throughput requirements. To solve this problem, we propose a FPGA-based XTS-AES encryption/decryption to suit the needs for secure SAN applications with high throughput requirements. Besides throughput, area optimization is also considered in this proposed design. First, we reuse the same AES encryption to produce the tweak value and unify the operations of AES encryption/decryption in XTS-AES encryption/decryption. Second, we transfer the computations of AES encryption/decryption from $GF(2^8)$ to $GF(2^4)^2$, which enables us move the map and the inverse map functions outside the AES round. Third, we propose to support the SubBytes and the inverse SubBytes by the same hardware component. Finally, pipelined registers have been inserted into the proposed unrolled architecture for XTS-AES encryption/decryption. The experiments show that the proposed design achieves 36.2 Gbits/s throughput using 6784 slices on XC6VLX240T FPGA.**

I. Introduction

Secure Storage Area Network approaches not only provide high performance, flexible, open standards based storage infrastructure but also secure solutions for protecting personal sensitive data. Therefore, "data-at-rest" in SAN needs to be protected in order to reduce the risk of information leakage. The simplest way to deal with such a situation is to encrypt the personal data before it is stored in a data center and to decrypt the encrypted personal data before it is requested by the users. The challenge is to develop a high throughput data encryption/decryption implementation to meet the throughput requirements. For example, in a real secure SAN platform, 32Gbits/s throughput is required as the connected PCIe interfaces run at 250 MHz (ML605 board supports the 8 × Gen1.1 PCIe interface).

Data transformations in a SAN system usually need real-time high throughput processing regardless of area overheads. P1619 [1] is a standard for cryptographic protection of data on block-oriented storage devices. It deals with a special case of protecting "data-at-rest" when storage media is sector-addressable. A tweakable encryption mode, called XTS-AES, has been standardized in [1], in which the tweak value is computed through the combination of the sector address and index of the block within a sector. XTS-AES includes the encryption and decryption of the data on the block-oriented storage. Hatzidimitriou presented a survey on the implementations of XTS-AES [2]. They also proposed to improve the throughput of XTS-AES. However, the maximum throughput of their implementation on Virtex-5 FPGA is only 6.4 Gbits/s, which is too slow for high throughput secure SAN applications. Furthermore, when they ported their designs to ASIC platform (0.13 μm CMOS technology), 38.077 Gbits/s is achieved. However, this is only targeted at ASIC implementation.

The main issue to achieving a faster XTS-AES is to improve the performance of the AES encryption/decryption part. There have been some existing works on AES encryption and decryption [3]. Rouvroy et al. [3] proposed a compact AES encryption and decryption design with 208 Mbps throughput, which fits the small FPGA. Gaj and Chodowiec [5] proposed a pipelined structure for the AES on Virtex XCV-1000 FPGA and achieved 12 Gbits/s. Standaert et al. [6] presented the design trade-off for the further optimization of the AES implementation on FPGA platforms. Unrolling, tiling, and pipelining structures for the AES were discussed in [7]. McLoone and McCanny's method achieved a throughput of 12Gbits/s using Look Up Table (LUT) based SubBytes [8]. Another approach [9] aimed at on-the-fly generation of SubBytes was first proposed by Rijmen, one of the creators of the AES. Hodjat and Verbauwhede presented a fully pipelined SubBytes architecture achieving a throughput of 21.54 Gbits/s [10].

In order to suit the throughput requirement of a secure SAN application, we propose an efficient high throughput XTS-AES encryption/decryption based on FPGA with the goal of area optimization. We unify the main computational part of XTS-AES, AES encryption and decryption, and reuse the AES encryption to produce the tweak value. In order to shorten the critical path and optimize the area resource, we transfer the main computations of AES encryption/decryption from $GF(2^8)$ to $GF(2^4)^2$. We also propose to support the SubBytes and the inverse SubBytes by the same hardware component. In order to meet throughput requirement, we insert proper pipelined registers into the proposed design. The experimental results show that our proposed design achieved 36.2 Gbits/s using 6784 slices on XC6VLX240T FPGA.

The rest of this paper is organized as follows: Section II gives the introduction of XTS-AES encryption and decryption. Section III proposes the high throughput unified XTS-AES encryption and decryption, and presents the detailed design methodologies about the SubByte and the inverse SubBytes

978-1-4799-6245-7/14/$31.00 ©2014 IEEE

and the MixColumns and the inverse MixColumns, and further optimizes the proposed design by inserting pipelined registers. Section IV shows the experimental results. Section V draws the conclusion.

II. Background of XTS-AES Algorithm in SANs

The architecture of the proposed secure SAN system had been discussed in our previous work [11]. in In order to meet security requirements specified by FIPS-140 [12], we use block data encryption/decryption standard, XTS-AES, to perform data encryption/decryption in SEE through PCIe interface. The detailed definition of XTS-AES can be found in [1]. This standard specifically deals with encrypting/decrypting the data stream, which is divided into consecutive equal-size data unit. This data stream must be encrypted and stored on the storage device. Algorithm 1 shows the procedure of XTS-AES encryption and its corresponding diagram is shown in Fig. 1. XTS-AES decryption differs from XTS-AES encryption at the second computational part, in which AES decryption substitutes AES encryption.

Algorithm 1 XTS AES Encryption Procedure

Input: Key=$\text{Key}_1 \mid \text{Key}_2$,
(Key is the 256 or 512 bit XTS-AES key),
Plaintext (a block of 128bit),
i (the value of the 128-bit tweak),
j (the sequential number of the 128-bit block inside the data unit)

Output: Ciphertext (the block of 128 bits of ciphertext resulting from the operation)

1: $T \leftarrow AES - Enc(Key_2, i) \otimes \alpha^j$
2: $PP \leftarrow Paintext \oplus T$
3: $CC \leftarrow AES - Enc(Key_1, PP)$
4: $Ciphertext \leftarrow CC \oplus T$

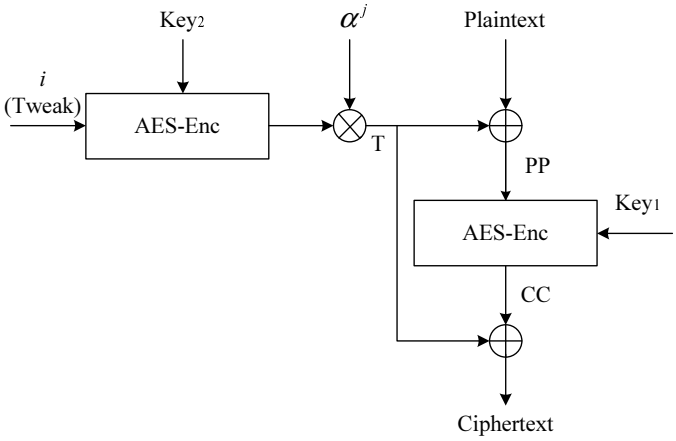

Fig. 1. XTS-AES encryption procedure

III. Proposed High Throughput XTS-AES on FPGA

In this section, we will present the proposed architecture for XTS-AES encryption and decryption. Then, we focus on the key part of this architecture, AES encryption and decryption, in which we extend the AES encryption to support the AES decryption. This is realized by a unified S-box to support the SubBytes and the inverse SubBytes. Finally, In order to improve the throughput, proper number of pipelined stages has been inserted to the proposed design.

In a secure SAN system, it needs up to 32 Gbits/s throughput, which is limited by the PCIe interface (runs at 250 MHz). This requirement is quite high, that the current existing XTS-AES design cannot meet this demand. On the other hand, in order to support additional secure features, such as key generation and hashing function, we must reserve the enough area resources for future extension. Therefore, area optimization for the proposed XTS-AES design is another challenging.

There already existed works about the unified AES encryption and decryption as discussed above. The most efficient one is proposed by Mathew et al. [13]. But, their design focused on ASIC platform which cannot directly be ported to FPGA platform. Furthermore, the architecture in their work was not the fully pipelined structure. In order to reduce the occupied area, we extend the AES encryption to support AES decryption by extending the SubBytes to support the inverse SubBytes and extending the MixColumns to support the inverse MixColumns. As shown in our previous work [14] [15] [16], We also transform the SubBytes computations from $GF(2^8)$ to $GF(2^4)^2$. This enables us to move the map and the inverse map function outside the AES round, which shortens the original critical path by around 20%. Most computations of AES encryption/decryption are carried on $GF(2^4)^2$. The left side of Fig. 2 shows the normal AES encryption and decryption, in which most operations are computed over $GF(2^8)$. The right side of Fig. 2 shows the proposed AES encryption/decryption, which moves the map and the inverse map functions outside AES round. The most computations of the proposed design are calculated over $GF(2^4)^2$. It is obvious that the main computation flow of AES encryption and AES decryption is similar, which motivates us to unify the SubBytes and the inverse SubBytes operations.

The adjustment of the SubBytes and the inverse SubBytes depends on the selected map function. This function can map a data over $GF(2^8)$ to another data over $GF(2^4)^2$. Its inverse function can map a data over $GF(2^4)^2$ to another data over $GF(2^8)$. The original polynomial to perform the modular inverse of the SubBytes over $GF(2^8)$ is $z^8 + z^4 + z^3 + z + 1$. The composite field arithmetic over $GF(2^{2\times4})$ can be transferred to the arithmetic over the combination of the appropriate ground field $GF(2^4)^2$ and the composite field $GF(2^4)$. They can be represented as $x^4 + a_3x^3 + a_2x^2 + a_1x + a_0(a_3, a_2, a_1, a_0 \in \{0, 1\})$ and $y^2 + \alpha y + \beta(\alpha, \beta \in \{GF(2^4)\})$, respectively. In this paper, ground field $x^4 + x + 1$ and composite field $y^2 + y + e$ are selected.

Usually, throughput can be significantly improved by inserting pipeline registers. For each AES encryption and de-

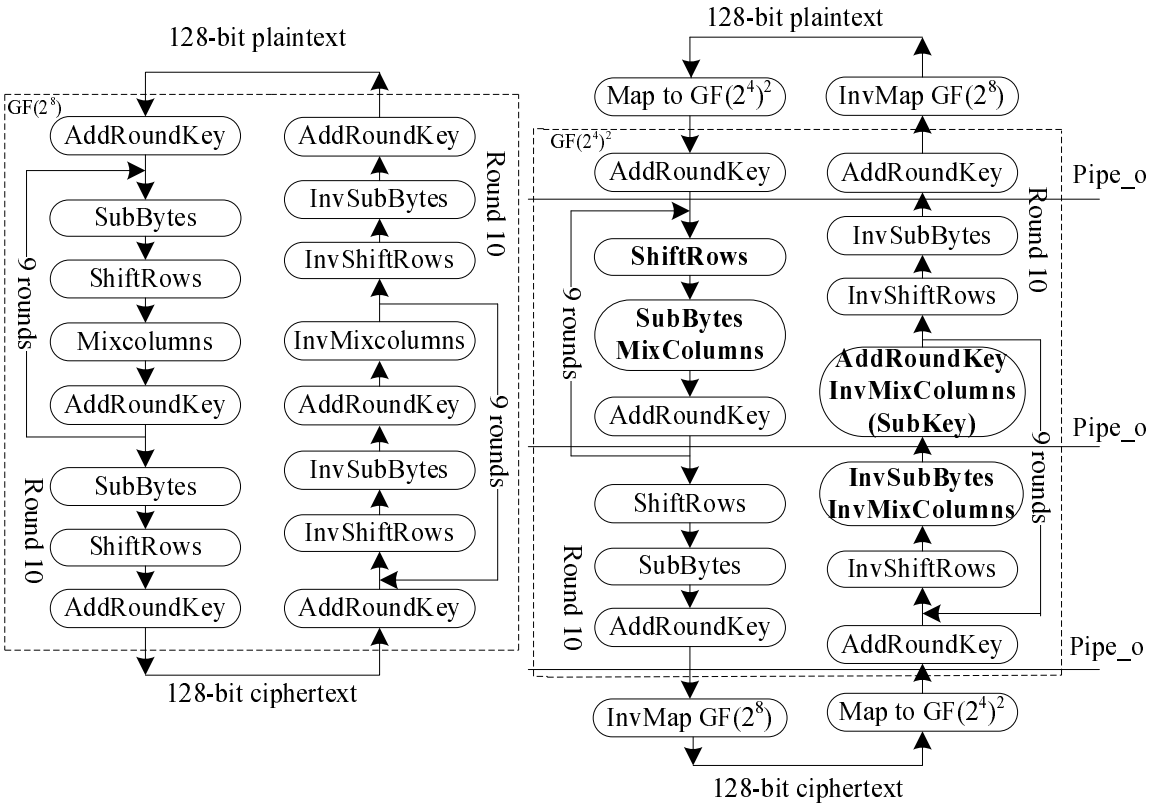

Fig. 2. Transformation of AES Encryption/Decryption over $GF(2^8)$ to $GF(2^4)^2$

cryptions round, we insert 4-stage pipelines to enhance the throughput. We insert 1 stage pipeline to each round of the AES encryption/decryption, called Pipe_o, as shown in Fig. 2. We insert three stage pipelines to the SubBytes and the inverse BubBytes. Note that, the maximum pipelined stages of our proposed design are 4. In order to be compatible with the encryption/decryption procedure, we also insert 4-stage pipelines to the key expansion in order not to affect the critical path of the main computation. Compared with the existing designs over $GF(2^8)$. Our proposed design shortens the critical path and reduces the number of the required pipeline stages, which outcomes a shorter latency. Throughput will be improved with the inserted pipelined registers, which enables us to meet the throughput requirement.

IV. Experimental Results

In this section, we have implemented the proposed design with a hardware description language (VHDL), synthesized our design using Xilinx ISE 13.3 and ported the design to Virtex-6 FPGA ML605 platform [18]. In the evaluation platform, we need make sure that the performance of our design is compliable with PCIe interface, which runs at 250 MHz (ML605 board supports the 8 $\times$ Gen1.1 PCIe interface).

Table I shows the comparisons between our proposed design and the existing designs. It is hard to compare the results with the existing works as different targeted platforms and methods. In order to have a fair comparison, we synthesize our design on the similar platforms used in the existing works. Hodjat and Ingrid's work [10] mainly focused on AES encryption procedure, whose throughput was 21.64 Gbits/s on XC2VP207 FPGA. However, current Xilinx ISE 13.4 toll does not support this FPGA. The throughput of the other work on AES encryption/decryption proposed by Lu et al. [4] was only 0.609 Gbits/s, because the application requirement of their design was not for high throughput. Recently, Kakarountas et al. presented a survey of XTS-AES implementation, which focused on the overall architecture of XTS-AES. Compared to this work, the throughput of our proposed design is around 3.5 times larger than Kakarountas's design on Virtex-5 FPGA. The Mbps/Slice of our proposed design is 13.5 larger than Kakarountas's design on Virtex-5 FPGA. When we ported our proposed design to Virtex-6 FPGA, the frequency is 26.3% faster than the one on Virtex-5 FPGA, which is 283.2 MHz. This is 13.3% faster than the required frequency by PCIe interface (250MHz). The proposed AES encryption/decryption reduces around 42.9% compared to Hodjat's design. We achieve 5.3 Mbps/Slice for the proposed XTS-AES design.

V. Conclusions

We proposed a FPGA based XTS-AES encryption/decryption with the aims of high throughput and smaller area for a secure San application. This enables us to integrate other cryptographic functions (hash and key generations etc.) into this design in the future. They

TABLE I
RESULTS OF THE XTS-AES ENCRYPTION AND DECRYPTION WITH THE EXISTING DESIGNS

	Techniques	Latency Cycles	Speed (MHz)	Area (Slices/LUTs)	Throughput (Gbits/s)	Mbps/Slice	Platforms
Hodjat [10]	AES encyrption	77	169.1*	9446/-	21.64	2.3	XC2VP207
Lu [4]	AES encyrption/decryption	21	100.0	31957 gates	0.609	-	TSMC 0.25 μm
Kakarountas [2]	XTS-AES encryptiondecryption (dual core 32*128bit)	-	209.0	1729/-	6.4	3.7*	Virtex-5
		-	1661.0	90k gates	37.73	-	0.13 μ CMOS
Our Work	AES encryption/decyrption alone	44	224.3	6129/18542	28.7	4.9	XC5VLX330T
		44	283.2	6129/18483	36.2	5.9	XC6VLX240T
	XTS-AES encryption/decryption (one core 32*128bit)	55+11	224.3	6784/20006	28.7	4.2	XC5VLX330T
		55+11	283.2	6784/19874	36.2	5.3	XC6VLX240T

*: The authors calculate based on the original paper; -: not supported

are accessible by a software program on a PC through a PCIe interface (250 MHz), which is attached in a ML605 evaluation board. We use three methods to speed up the proposed design. First, we transfer the computations of AES encryption/decryption from $GF(2^8)$ to $GF(2^4)^2$ and move the map and the inverse map functions outside the AES round. All the computations are calculated over $GF(2^4)^2$. Second, we unify the SubBytes and the inverse SubBytes operations. Finally, we inserted 5 pipelined stages into the proposed design, which includes one stage for tweak function and four stages for AES encryption/decryption. The experimental results show that the throughput of our design is around 3.5 faster compared to the existing design on Virtex-5 FPGA. When ported to Virtex-6 FPGA, the throughput of our proposed design is 36.2Gbits/s which is 13.1% larger than the required throughput by PCIe interface (32Gbits/s).

ACKNOWLEDGMENT

The author would like to thank A*STAR Singapore for the funding support (Project No. 1122804006).

REFERENCES

[1] "IEEE P1619^{TM}/D16 Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices," http://grouper.ieee.org/groups/1619/email/pdf00086.pdf

[2] A.P. Kakarountas, E. Hatzidimitriou and A. Milidonis, "A Survey on Throughput-Efficient Architectures for IEEE P1619 for Shared Storage Media," IEEE Symposium on Computers and Communications (ISCC), 2011, pp. 758-763.

[3] G. Rouvroy, F.-X. Standaert, J.-J. Quisquater and J. Legat, "Compact and efficient encryption/decryption module for FPGA implementation of the AES Rijndael very well suited for small embedded applications," International Conference on Information Technology: Coding and Computing, 2004, pp. 583-587.

[4] C-C. Lu and S-Y. Tseng, "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter," The IEEE International Conference on Application-Specific Systems, Architectures and Processors, 2002, pp. 277-185.

[5] K. Gaj and P. Chodowiec, "Fast implementation and fair comparison of the final candidates for Advanced Encryption Standard using Field Programmable Gate Arrays," CT-RSA 2001, LNCS 2020, USA, 2001, pp. 84-99.

[6] F. X. Standaert, G. Rouvroy, J. J. Quisquater and J. D. Legat, "Efficient implementation of Rijndael encryption in reconfigurable hardware: improvements and design tradeoffs," CHES 2003, LNCS 2779, German, 2003, pp. 334-350.

[7] G. P. Saggese, A. Mazzeo, N. Mazzocca and A. G. M. Strollo, "An FPGA-based performance analysis of the unrolling, tiling, and pipelining of the AES algorithm," FPL 2003, LNCS 2778, Portugal, 2003, pp. 292-302.

[8] M. McLoone and J. V. McCanny, "Rijndael FPGA implementations utilizing Look-Up tables," IEEE Workshop on Signal Processing Systems, Belgium, 2001, pp. 249-360.

[9] V. Rijmen, "Efficient implementation of the Rijndael S-Box, "Available: http://www.iaik.tu-graz.ac.at/research/krypto/AES/old/ rijmen/rijndael/sbox.pdf. 2006.

[10] A. Hodjat and V. Ingrid, "A 21.54 Gbits/s fully pipelined processor on FPGA," 12th Annual IEEE Symposium on Field-Programmable Custom Computing Machines, USA, 2004, pp. 308-309.

[11] Y. Wang and Y. Ha, "FPGA Based ReKeying for Cryptographic Key Management in Storage Area Network", International Conference on Field Programmable Logic and Applications (FPL), 2013, pp 1-6.

[12] National Institute of Standards and Technolog, "Security Requirment for Cryptographic Modules," FIPS-140-2, 2002, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[13] S. K. Mathew, F. Sheikh, M. Kounavis, S. Gueron, A. Agarwal, S. K. Hsu, H. Kaul, M. A. Anders, R. K. Krishnamurthy, "53 Gbps native $GF(2^4)^2$ composite-field AES-encrypt/decrypt accelerator for content-protection in 45 nm high-performance microprocessors," IEEE Journal of Solid-State Circuits, vol. 46, no. 4, pp. 767-776, Feb. 2011.

[14] Y. Zheng, Y. Wang, J. Li, R. Li and W. Zhao, "FPGA based optimization for masked AES implementation," IEEE International Midwest Symposium on Circuits and Systems (MWCAS), 2011, pp. 1-4.

[15] Y. Wang and Y. Ha, "FPGA-based 40.9-Gbits/s masked AES with area optimization for storage area network," IEEE Transactions on Circuit and System II, vol. 60, no. 1, pp. 36-40, January 2013.

[16] Y. Wang and Y. Ha, "A performance and area efficient ASIP for higher-order DPA-resistant AES," IEEE Journal on Emerging and Selected Topics in Circuits and Systems, Volume 4, Issue 2, pp. 190-202, April 2014.

[17] National Institute of Standards and Technology. Advanced Encryption Standard (AES), FIPS-197, 2001.

[18] Xilinx, "Virtex-6 FPGA ML605 Evaluation Kit," http://www.xilinx.com/products/boards-and-kits/EK-V6-ML605-G.htm.