BITSTREAM ENCRYPTION AND AUTHENTICATION WITH AES-GCM IN DYNAMICALLY RECONFIGURABLE SYSTEMS

Yohei Hori †, *Akashi Satoh*†, *Hirofumi Sakane* †, *and Kenji Toda* †

† National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba-shi, Ibaraki 305-8568, Japan
email: {hori.y, akashi.satoh, hirofumi.sakane, k-toda}@aist.go.jp

ABSTRACT

A high-speed and secure dynamic partial reconfiguration (DPR) system is realized with AES-GCM that guarantees both confidentiality and authenticity of FPGA bitstreams. In DPR systems, bitstream authentication is essential for avoiding fatal damage caused by unintended bitstreams. An encryption-only system can prevent bitstream cloning and reverse engineering, but cannot prevent erroneous or malicious bitstreams from being configured. Authenticated encryption is a relatively new concept that provides both message encryption and authentication, and AES-GCM is one of the latest authenticated encryption algorithms suitable for hardware implementation. We implemented the AES-GCM-based DPR system targeting the Virtex-5 device on an off-the-shelf board, and evaluated its throughput and hardware resource utilization. For comparison, we also implemented AES-CBC and SHA-256 modules on the same device. The experimental results showed that the AES-GCM-based system achieved higher throughput with less resource utilization than the AES/SHA-based system. The AES-GCM module achieved more than 1 Gbps throughput and the entire system achieved about 800 Mbps throughput with reasonable resource utilization. This paper clarifies the advantage of using AES-GCM for protecting DPR systems.

1. INTRODUCTION

Some recent Field-Programmable Gate Arrays (FPGAs) provide the ability of *dynamic partial reconfiguration (DPR)*, where a portion of the circuit is replaced with another module while the rest of the circuit remains fully operational. By using DPR, the functionality of the system is reactively altered by replacing a hardware module according to, for example, a user request, performance requirement, or environmental change. The flexibility of DPR is expected to make hardware systems multifunctional, cost efficient and area efficient. DPR also achieves short configuration time and consequently makes reconfigurable computing more practical and operational. The application of DPR is studied in the fields of content distribution [1], image processing [2], automotives [3], fault-tolerant systems [4], and software defined radio [5] among others.

In a system where intellectual property (IP) cores are replaced using DPR, the security of the IP core bitstreams is of primary concern. To guarantee confidentiality of proprietary IP cores, bitstream protection using a cryptographic algorithm such as Advanced Encryption Standard (AES) [6] is quite effective and is widely applied in practical systems. The encryption prevents bitstream cloning and reverse engineering. Several FPGA families have an embedded decryptor and can be configured from an encrypted bitstream. However, such an embedded decryptor is available only for the entire configuration and not for DPR.

In addition to bitstream encryption, bitstream authentication is extremely significant in the protection of DPR systems [7]. An encryption-only system is insufficiently secure because the system cannot prevent erroneous or malicious bitstreams from being configured. Since hardware architecture itself is changed in DPR systems, an unauthorized bitstream can cause fatal, unrecoverable damage to the system. In the encryption-only system, the malicious bitstream will be jumbled by the decryptor to generate meaningless data. However, there still remains a possibility that the erroneous bitstream will damage the FPGA by indiscriminately setting the internal logic, I/O, interconnect and so on. To guarantee the authenticity of a message, Secure Hash Algorithm (SHA) [8] is widely used.

Since both confidentiality and authenticity of bitstreams must be guaranteed, authenticated encryption (AE) [9] must be effectively applied to DPR systems. AE is a relatively new concept in cryptographic technology, providing both message encryption and authentication. AE is expected to lead to area-efficient implementation when compared with the use of separate encryption and authentication algorithms. It will also enable high-speed implementation eliminating overheads of data synchronization between two separate algorithms.

To protect a DPR system against erroneous or malicious bitstreams, we implement the latest AE algorithm Galois / Counter Mode (GCM) of operation [10,11] with AES. GCM is based on the counter mode of operation (CTR) and uses universal hashing in the finite field $GF(2^w)$ [12]. GCM is pipelinable and parallelizable, and thus suitable for hardware implementation. As is explained in [13], other AE algorithms are not necessarily suitable for hardware implementation because they are not parallelizable or pipelinable. Additionally, other algorithms have weakness against bit-flipping attacks. Therefore, the use of AES-GCM is currently the best solution for protecting bitstreams with both encryption and authentication.

This paper presents the architecture, implementation results and performance evaluation of an AES-GCM-based DPR system. The system is implemented targeting Virtex-

5 on an off-the-shelf board, and we verify that its mechanism of bitstream encryption and authentication successfully works. For comparison, an AES-CBC and SHA-256-based DPR system is also implemented on the same device. To compare resource utilization with past studies, both systems are also implemented on Virtex-II Pro.

The rest of this paper is organized as follows. Section 2 introduces past studies on DPR security. Section 3 explains the partial reconfiguration of Xilinx FPGA. Section 4 briefly explains the cryptographic algorithms related to our implementation. Section 5 describes the system architecture of our AES-GCM-based and AES-CBC/SHA-256-based DPR systems. Section 6 describes the implementation results and evaluation of the developed DPR systems, and finally Section 7 summarizes this paper.

2. RELATED WORK

Xilinx Virtex series devices support configuration with an encrypted bitstream. Virtex devices have a built-in bitstream decryptor. Virtex-II and Virtex-II Pro support Triple Data Encryption Standard (Triple-DES) [14] with a 56-bit key, while Virtex-4 and Virtex-5 support AES with a 256-bit key. The secret key is stored in the dedicated volatile memory inside the FPGA. Therefore, the storage must always be supplied with power through an external battery. Unfortunately, the functionality of the configuration with an encrypted bitstream is unavailable when using DPR. If the device is configured using the built-in bitstream decryptor, DPR function is disabled. Therefore, in DPR systems, a partial bitstream must be decrypted with user logic.

Bossuet et al. proposed a secure configuration method in DPR systems [15]. In their system, an arbitrary cryptographic algorithm can be employed because the bitstream decryptor itself is implemented as a reconfigurable module. Their method uses bitstream encryption but does not consider its authenticity.

Zeineddini and Gaj developed a DPR system that used separate encryption/authentication algorithms for bitstream protection [16]. AES was used for bitstream encryption and SHA-1 for authentication. AES and SHA-1 were implemented as C programs and run on two types of embedded micro processors: PowerPC and MicroBlaze. The total processing times of authentication, decryption and configuration of a 14-KB bitstream with PowerPC and MicroBlaze were approximately 400 ms and 2.3 sec, respectively. These performances, however, would be insufficient for practical DPR systems.

Parelkar used AE to protect FPGA bitstreams [17] and implemented various AE algorithms: Offset CodeBook (OCB) [18], Counter with CBC-MAC (CCM) [19] and EAX [20] modes of operation with AES. To compare the performance of the AE method with a separate encryption and authentication method, SHA-1 and SHA-512 are also implemented with AES-ECB (Electronic CodeBook).

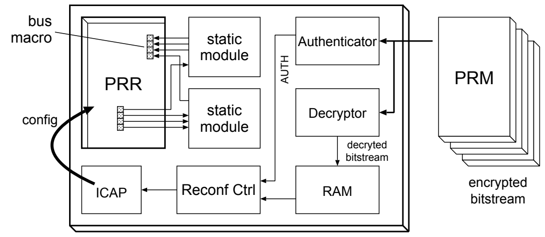

Fig. 1. Structure of a partially reconfigurable circuit with a Xilinx FPGA.

3. PARTIAL RECONFIGURATION OF FPGAS

3.1. Partial Reconfiguration Overview

In Xilinx FPGAs, a module to be dynamically replaced is called a *Partially Reconfigurable Module (PRM)*, and an area where PRM is placed is called a *Partially Reconfigurable Region (PRR)*. PRM can be an arbitrary size of rectangular. Figure 1 shows an example structure of the partially reconfigurable design.

The smallest unit of a bitstream that can be accessed is called a *frame*. In Virtex-5 devices, a frame is 1312-bit configuration information corresponding to the height of 20 configurable logic blocks. A bitstream of PRM is a collection of frames. Each device family has different frame structures, but this paper does not focus on other devices.

3.2. Bus Macro

All signals between a PRM and a fixed module must pass through *bus macros* to lock the wiring. In Virtex-5 devices, the bus macro is a 4-bit-wide pre-routed macro composed of four 6-input Lookup Tables (LUTs). The bus macro must be placed inside a PRM. The bus macros of the older device families are 8-bit-wide pre-routed macros composed of sixteen 4-input LUTs, and placed on the PRM boundary.

3.3. Internal Configuration Access Port

Virtex-II and newer Virtex series devices support *self DPR* with the *Internal Configuration Access Port* (*ICAP*). ICAP basically works in the same manner as the SelectMAP configuration interface. Since user logic can access configuration memory through ICAP, partial reconfiguration of an FPGA can be controlled by internal user logic. In Virtex-5 devices, the data width of ICAP can be selected from 8, 16 and 32 bits.

4. CRYPTOGRAPHIC ALGORITHM

4.1. Advanced Encryption Standards

AES is a symmetric key block cipher algorithm standardized by the U.S. National Institute of Standard and Technologies (NIST) [6]. While the previous DES [21] has a Feistel network architecture, AES employs a substitution-permutation network (SPN) architecture. The block length of AES is 128 bits, and the key length is selected from 128, 196 and 256 bits.

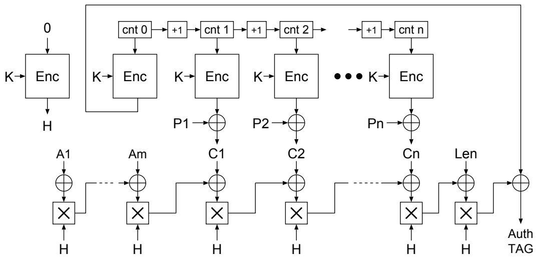

Fig. 2. Example operation of Galois/Counter Mode (GCM).

4.2. Galois/Counter Mode of Operation

A block cipher algorithm can be applied to various modes of operation. The GCM [10] is one of the latest modes of operation standardized by NIST [11]. Figure 2 shows an example operation of GCM.

The encryption and decryption scheme of GCM is based on CTR mode of operation [22]. Thus, GCM can be highly parallelized and pipelined and is therefore suitable for hardware implementation, achieving a wide variety of performances from compact to high speed [23, 24]. Some other AE algorithms are not necessarily suitable for hardware implementation because they are unable to be parallelized or pipelined [13].

AES-GCM is one of the AE algorithms providing both message confidentiality and authenticity. GCM uses universal hashing in the finite field $GF(2^w)$ for generating a message authentication code (MAC). The additional merit of using $GF(2^w)$ is that the computation cost of multiplication under $GF(2^w)$ is less than integer multiplication.

AES-GCM provides high security suitable for hardware implementation. Therefore, the use of AES-GCM is the best solution for protecting FPGA bitstreams in DPR systems.

4.3. Secure Hash Algorithm

SHA is widely used to guarantee the authenticity of a message. SHA is one of the cryptographic hash functions that generates a particular length of a message digest. Currently, five algorithms, namely, SHA-1, SHA-224, SHA-256, SHA-386 and SHA-512, are defined denoting the length of the output message digest (the output length of SHA-1 is 160 bits). The latter four algorithms are collectively referred to as SHA-2. Since SHA-1 has been reported to have security vulnerability [25], SHA-2 should be used instead for message authentication.

5. SYSTEM ARCHITECTURE

5.1. DPR system with AES-GCM

In DPR systems, both the confidentiality and the authenticity of PRM bitstreams should be guaranteed. As mentioned in Section 4.2, AES-GCM is one of the most promising algorithms to achieve this purpose. Figure 3 shows a block diagram of the DPR system with bitstream encryption and authentication using AES-GCM. In the system, the length of the AES key and an initial vector are set to 128 bits and 96 bits, respectively.

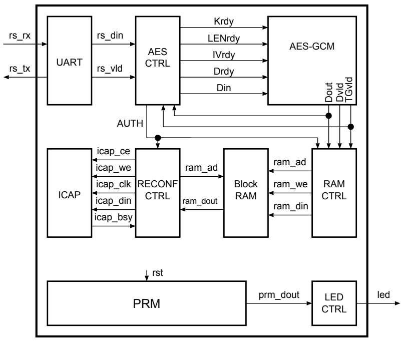

Fig. 3. Overview of the system with AES-GCM.

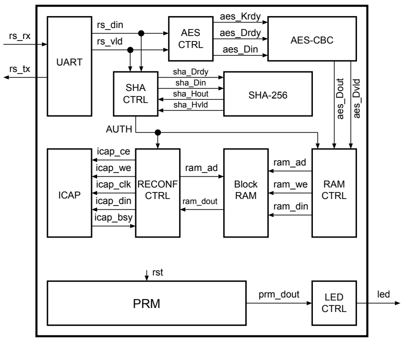

Fig. 4. Overview of the system with AES-CBC/SHA-256.

The downloaded bitstream of PRM is decrypted by the AES-GCM module and its authenticity is simultaneously verified. The secret key of AES is embedded in the system. The decrypted bitstream is stored to the 128x2048 bits (32KB) internal memory (Block RAM). A bitstream is checked for its authenticity every 32KB, so a large bitstream is divided into several 32KB blocks. Reconfiguration of PRM starts after the bitstream is authenticated by AES-GCM. Authentication and reconfiguration cannot be parallelized or processed in a fine-grained pipeline because the encrypted bitstream can not be input to ICAP before its authenticity is inspected.

5.1.1. AES-GCM

GCM is based on the CTR mode and uses universal hashing in the finite field $GF(2^w)$. The S-box of AES is implemented as a table using Block RAM. In AES-GCM, a 128-bit block is decrypted in 12 clock cycles. The last block of the message requires 12 and an additional 10 clock cycles to calculate the authentication tag.

Suppose that the size of the bitstream is N [byte] and the clock frequency of the system is f [MHz]. When N is sufficiently large, the additional 10 cycles for the tag calculation is safely ignored. Thus the maximum throughput of the AES-GCM module P_{gcm} is

$$P_{gcm} = \frac{128}{12} \times f = \frac{32}{3} f \text{ [Mbps]}. \tag{1}$$

5.1.2. Reconfiguration of PRM

Unlike other DPR systems, our system does not use an embedded processor to control partial reconfiguration. The input data and control signals of ICAP are directly connected to and controlled by the user logic. Thus, our system is free from the delay of processor buses. In the system, the width of the ICAP data port is set to 32 bits. When the frequency of input data to ICAP is f [MHz], the maximum throughput of the reconfiguration process P_{icap} is

$$P_{icap} = 32f \text{ [Mbps]}. \tag{2}$$

In Virtex-5, the maximum frequency of the ICAP is limited to 100 MHz, thus the ideal throughput of the reconfiguration process is 3,200 Mbps.

As the main purpose of this study is to clarify the feasibility of AES-GCM for bitstream encryption and authentication, rather simple function blocks, e.g., 28-bit adder and 28-bit subtractor, are used as PRM. PRM is connected to the static modules with two bus macros. The most significant 4 bits of the adder or the subtractor are output from PRM and connected to LEDs on the board. The PRR contains 80 slices, 640 LUTs and 320 registers. The size of the PRM bitstream is about 11KB.

5.2. DPR System with AES-CBC + SHA-256

To compare the performance of the AE method and the separate encryption/authentication method, we also implemented the AES-CBC and SHA-256 modules for bitstream encryption and authentication. Figure 4 shows a block diagram of the DPR system with AES-CBC and SHA-256. Bitstream encryption using AES-CBC and authentication using SHA-256 are processed in parallel. As the same as AES-GCM, the decrypted bitstream is stored to a 128x2048-bit Block RAM. Reconfiguration of PRM starts after the authenticity of the bitstream is verified.

5.2.1. AES-CBC

In our system, one of the most major confidentiality modes AES-CBC is used. The simplest mode of operation (AES-ECB) is not employed because it is not sufficiently secure for practical use [22]. The CBC mode can be used for generating a message authentication code (MAC) [26], but it is not employed because the CBC-MAC algorithm reportedly has security deficiencies [27]. Therefore, our system employs the different authentication algorithm, SHA-256, for bitstream integrity check.

Similar to the AES-GCM system, the S-box is implemented as a table using Block RAM. In AES-CBC, a 128-bit block is decrypted in 11 clock cycles. When the operating frequency of the system is f [MHz], the maximum throughput of the AES-CBC module P_{cbc} is

$$P_{cbc} = \frac{128}{11} f \text{ [Mbps]}. \tag{3}$$

5.2.2. SHA-256 Module

Since SHA-1 reportedly has security vulnerability [25], SHA-256 is selected for the authentication algorithm. The input block size of SHA-256 is 512 bits and output message digest is 256 bits. The data bus of the SHA-256 module is 32-bit wide, and thus data input and output take 16 and 8 cycles, respectively. A 512-bit block of message is calculated in 49 cycles. When the operating frequency of the system is f [MHz], the maximum throughput of the SHA-256 module P_{sha} is

$$P_{sha} = \frac{512}{(16 + 49 + 8)} \times f = \frac{512}{73} f \text{ [Mbps]}. \tag{4}$$

SHA-256 processing takes longer cycles than AES; therefore, the throughput of the overall bitstream processing is restricted by SHA-256. While the SHA-256 algorithm is relatively simple and straightforward, it is difficult to process in parallel or pipeline. Thus, the performance of the SHA-256 module is difficult to improve.

5.2.3. Reconfiguration of PRM

Reconfiguration of PRM in the AES-CBC/SHA-256 system is performed in the same manner as that in the AES-GCM system after the authenticity of the bitstream is verified by the SHA-256 module. When the operating frequency is f [MHz], the throughput of ICAP is $32f$ [Mbps]. The ideal throughput of reconfiguration is 3,200 Mbps as explained in Section 5.1.2.

6. IMPLEMENTATION

This section describes the implementation results of the AES-GCM-based system (hereinafter PR-AES-GCM) and AES-CBC/SHA-256-based system (hereinafter PR-AES-SHA). PR-AES-GCM and PR-AES-SHA are implemented targeting Virtex-5 (XC5VLX50T-FFG1136) on an ML505 board [28], and we verified that DPR successfully works on the systems. The systems are designed using Xilinx Early Access Partial Reconfiguration (EA PR) flow [29] and implemented with ISE 9.1.02i_PR10 and PlanAhead 9.2.7 [30].

6.1. Hardware Resource Utilization

Table 1 and Table 2 show the hardware utilization of PR-AES-GCM and PR-AES-SHA implemented on Virtex-5, respectively. The item “Overall” shows the total amount of hardware resource used by all modules except PRM. Table 1 and Table 2 also describe the hardware utilization of each module of stand-alone implementation.

The hardware architecture of Virtex-5 is vastly different from that of earlier devices such as Virtex-II Pro and Virtex-4. The slice of Virtex-5 contains four 6-input LUTs, whereas that of earlier devices contains two 4-input LUTs. Thus, the number of used slices becomes smaller in Virtex-5 implementation. To make a fair comparison with other studies, we also implemented the systems on Virtex-II Pro (XC2VP30-FF896). The hardware utilization of PR-AES-GCM and PR-AES-SHA on Virtex-II Pro are given in Table 3.

Table 1. Hardware utilization of the static module of PR-AES-GCM on Virtex-5 (XC5VLX50T).

Module	Register	(%)	LUT	(%)	Slice	(%)	BRAM	(%)
Overall	2,166	7%	3,040	10%	1,390	19%	13	21%
AES-GCM	1,394	4%	2,298	7%	1,039	14%	5	8%
AES_CTRL	410	1%	429	1%	187	2%	0	0%
RECONF_CTRL	70	1%	166	1%	73	1%	0	0%
RAM_CTRL	147	1%	176	1%	86	1%	0	0%
CONFIG_RAM	0	0%	0	0%	0	0%	8	13%

Table 2. Hardware utilization of the static module of PR-AES-SHA on Virtex-5 (XC5VLX50T).

Module	Register	(%)	LUT	(%)	Slice	(%)	BRAM	(%)
Overall	2,327	8%	3,661	12%	1,592	22%	13	21%
AES-CBC	545	1%	1,176	4%	590	8%	5	8%
AES_CTRL	157	1%	240	1%	85	2%	0	0%
SHA-256	500	1%	1,274	4%	325	4%	0	0%
SHA_CTRL	588	2%	542	1%	242	3%	0	0%
RECONF_CTRL	165	1%	231	1%	90	1%	0	0%
RAM_CTRL	145	1%	173	1%	86	1%	0	0%
CONFIG_RAM	0	0%	0	0%	0	0%	8	13%

Table 3. Hardware utilization of the static module of PR-AES-GCM on Virtex-II Pro (XC2VP30).

	Register	(%)	LUT	(%)	Slice	(%)	BRAM	(%)
PR-AES-GCM (V2P)	2,179	7%	4,237	15%	2,687	19%	25	18%
PR-AES-SHA (V2P)	2,302	8%	4,064	14%	2,730	19%	25	18%

Table 4. Performance comparison of secure PR systems (14,112 bytes PRM).

System	Device	Slice	Authentication	Decryption	Configuration	Overall	Ratio
PR-AES-GCM	XC5VLX50T	2,687*	106.43 μs 1067 Mbps		35.3 μs 3200 Mbps	141.73 μs 797 Mbps	1
PR-AES-SHA256	XC5VLX50T	2,730*	160.97 μs 701 Mbps	97.14 μs 1164 Mbps	35.3 μs 3200 Mbps	196.27 μs 575 Mbps	1.28
PowerPC [16]	XC2VP30	1,334**	139 ms 812 kbps	208 ms 543 kbps	56 ms 2016 kbps	403 ms 280 kbps	2843
MicroBlaze [16]	XC2VP30	1,706**	776 ms 145 kbps	1472 ms 77 kbps	32 ms 3528 kbps	2280 ms 50 kbps	16087
AES-OCB [17]	XC4VLX60	2,964	601 Mbps		-	-	
AES-CCM [17]	XC4VLX60	2,799	255 Mbps		-	-	
AES-EAX [17]	XC4VLX60	2,993	287 Mbps		-	-	

* For a fair comparison, slice utilization of Virtex-II Pro is shown.
** Includes only reconfiguration controllers.

6.2. Performance Evaluation

The clock frequencies of PR-AES-GCM and PR-AES-SHA are both 100 MHz. To enable comparison with [16], the computation time required to configure a 14,112-byte PRM is described in Table 4. In PR-AES-GCM, the overall processing time for the PRM configuration is simply

$$106.43 + 35.3 = 141.73\ [\mu s]. \qquad (5)$$

In PR-AES-SHA, authentication and decryption are processed in parallel. Therefore, the overall processing time is

$$\max(160.97, 97.14) + 35.3 = 196.27\ [\mu s]. \qquad (6)$$

In PowerPC and MicroBlaze systems, authentication, decryption and reconfiguration are sequentially performed. As such, the overall processing time is simply the sum of each processing time. Using the equation (1) through (4), the performance of the systems are calculated, as shown in Table 4. Table 4 also gives the throughput of other AE algorithms reported in [17].

6.3. Analysis of the Results

As Tables 1 and 2 show, PR-AES-GCM utilizes less registers, LUTs and slices than PR-AES-SHA for the implementations on Virtex-5. The results indicate that AES-GCM is more area efficient than separate algorithms of AES-CBC and SHA-256. Implementing on Virtex-II Pro, PR-AES-GCM utilizes less registers and slices than PR-AES-SHA, though utilizes slightly more LUTs.

As shown in Table 4, PR-AES-GCM achieved the highest overall throughput of about 800 Mbps with only 19% slice utilization. The AES-GCM module achieves a throughput of more than 1 Gbps, which is faster than those of other AE methods of OCB, CCM and EAX. Furthermore, PR-AES-GCM uses less slices than other AE methods. Note that PR-AES-GCM includes additional modules such as a reconfiguration controller and an LED controller. The results shows that high-speed and area-efficient implementation is achieved by PR-AES-GCM.

Since AES-GCM can be processed in parallel and pipeline, AES-GCM can obtain much higher throughput using more hardware resources. AES-GCM provides very flexible architecture from compact to high speed.

In the PR-AES-SHA system, the AES module achieved the highest throughput of 1164 Mbps, while the overall throughput is relatively low (575 Mbps). This is because the throughput of the SHA-256 module is relatively low (701 Mbps). Since the SHA-256 algorithm is quite straightforward and hardly parallelized or pipelined, improving the SHA-256 throughput is difficult. Although the various hardware architectures of AES can achieve a wide variety of performances, the SHA-256 module will restrict the overall performance of the system. This is the disadvantage of using SHA-256 for bitstream authentication.

The DPR systems with the PowerPC and MicroBlaze systems require the overall computation time from several hundred milliseconds to several seconds. This will not be acceptable for practical DPR systems. Therefore, authentication, decryption and reconfiguration should be processed using dedicated hardware to achieve practical DPR systems. Comparing to the software AE systems, our approach attained extremely high performances. PR-AES-GCM achieved 2843 times higher throughput than the PowerPC system, and 16087 times higher throughput than the MicroBlaze system.

7. CONCLUSIONS

We developed a secure dynamic partial reconfiguration (DPR) system with AES-GCM that guarantees both confidentiality and authenticity of FPGA bitstreams. AES-GCM is one of the latest authenticated encryption (AE) algorithms. Implementing on Virtex-5 (XC5VLX50T), AES-GCM achieved more than 1 Gbps throughput and the entire system achieved about 800 Mbps throughput sufficient for practical DPR use, utilizing less than 20% slices.

For comparison, we also implemented AES-CBC and SHA-256 on the same device. The implementation results show that the AES-GCM-based system achieves higher throughput and is more area efficient than the AES/SHA-based system. Although AES can achieve a wide variety of performances from compact to high speed, SHA is a straightforward algorithm which is hardly parallelized or pipelined. Therefore, the performance of the AES/SHA-based system is restricted by the SHA module. The performance of the AES-GCM is also compared with other AE algorithms. The AES-GCM achieved higher throughput than other modes of operation such as OCB, CCM and EAX.

Considering the experimental results, it is concluded that the use of AES-GCM is currently one of the most promising approaches for protecting FPGA bitstreams and achieving high-speed and area-efficient DPR systems.

8. REFERENCES

[1] Y. Hori, H. Yokoyama, H. Sakane, and K. Toda, “A secure content delivery system based on a partially reconfigurable FPGA,” *IEICE Trans. Inf.&Syst.*, vol. E91-D, no. 5, pp. 1398–1407, May 2008.

[2] C. Claus, J. Zeppenfeld, F. Muller, and W. Stechele, “Using partial-run-time reconfigurable hardware to accelerate video processing in driver assistance system,” in *DATE’07*, 2007, pp. 498–503.

[3] J. Becker, M. Hubner, G. Hettich, R. Constapel, J. Eisenmann, and J. Luka, “Dynamic and partial FPGA exploitation,” *Proc. IEEE*, vol. 95, no. 2, pp. 438–452, 2007.

[4] J. Emmert, C. Stroud, B. Skaggs, and M. Abramovici, “Dynamic fault tolerance in FPGAs via partial reconfiguration,” in *FCCM 2000*, 2000, pp. 165–174.

[5] J. P. Delahaye, G. Gogniat, C. Roland, and P. Bomel, “Software radio and dynamic reconfiguration on a DSP/FPGA platform,” *J. Frequenz*, vol. 58, no. 5-6, pp. 152–159, 2004.

[6] National Institute of Standards and Technology, “Announcing the advanced encryption standard (AES),” FIPS PUB 197, Nov. 2001.

[7] S. Drimer, “Authentication of FPGA bitstreams: Why and how,” in *ARC’07*, vol. LNCS 4419, 2007, pp. 73–84.

[8] National Institute of Standards and Technology, “Secure hash standard,” FIPS 180-2, Aug. 2002.

[9] M. Bellare and C. Namprempre, “Authenticated encryption: Relations among notions and analysis of the generic composition paradigm,” in *ASIACRYPT 2000*, 2000, pp. 531–545.

[10] D. A. McGrew and J. Viega, “The Galois/counter mode of operation (GCM),” May 2005, http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html.

[11] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, SP 800-38D ed., National Institute of Standards and Technology, Nov. 2007.

[12] R. Lidl and H. Niederreiter, *Finite Fields*, 2nd ed., ser. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1997.

[13] D. A. McGrew and J. Viega, “The security and performance of the Galois/counter mode (GCM) of operation,” in *INDOCRYPT 2004*, 2004, pp. 343–355.

[14] National Institute of Standards and Technology, “Recommendation for the triple data encryption algorithm (TDEA) block cipher,” May 2004.

[15] L. Bossuet and G. Gogniat, “Dynamically configurable security for SRAM FPGA bitstreams,” *Int. J. Embedded Systems*, vol. 2, no. 1/2, pp. 73–85, 2006.

[16] A. S. Zeineddini and K. Gaj, “Secure partial reconfiguration of FPGAs,” in *ICFPT’05*, 2005, pp. 155–162.

[17] M. M. Parelkar, “Authenticated encryption in hardware,” Master’s thesis, George Mason University, 2005.

[18] P. Rogaway, M. Bellare, and B. John, “OCB: A block-cipher mode of operation for efficient authenticated encryption,” *ACM Trans. Information and System Security*, vol. 6, no. 3, pp. 365–403, Aug. 2003.

[19] D. Whiting, R. Housley, and N. Ferguson, “Counter with CBC-MAC (CCM),” RFC3610, Sept. 2003.

[20] M. Bellare, P. Rogaway, and D. Wagner, “A conventional authenticated-encryption mode,” http://www-08.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/eax/eax-spec.pdf, 2003.

[21] *Data Encryption Standard (DES)*, FIPS PUB 46-3 ed., U.S. Department of Commerce/National Institute of Standards and Technology, 1999.

[22] M. Dworkin, *Recommendation for Block Cipher Modes of Operation*, SP 800-38A ed., National Institute of Standards and Technology, Dec. 2001.

[23] A. Satoh, “High-speed parallel hardware architecture for Galois counter mode,” in *ISCAS’07*, 2007, pp. 1863–1866.

[24] A. Satoh, T. Sugawara, and T. Aoki, “High-speed pipelined hardware architecture for Galois counter mode,” in *ISC’07*, 2007, pp. 118–129.

[25] W. Xiaoyun, Y. Yiqun, and Y. Hongbo, “Finding collisions in the full SHA-1,” in *CRYPTO 2005*, 2005, pp. 17–36.

[26] M. Dworkin, *Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication*, SP 800-38B ed., National Institute of Standards and Technology, May 2005.

[27] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. CRC Press, Inc., 1996.

[28] *ML505/ML506 Evaluation Platform*, UG347(v2.4) ed., Xilinx, Inc., Oct. 2007.

[29] *Early Access Partial Reconfiguration User Guide For ISE 8.1.01i*, Xilinx, Inc., 2006.

[30] B. Jackson, *Partial Reconfiguration Design with PlanAhead 9.2*, Xilinx, Inc., Aug. 2007.