Low cost Solutions for Secure Remote Reconfiguration of FPGAs

Karim Moussa Ali Abdellatif, Roselyne Chotin-Avot, Habib Mehrez

Low Cost Solutions for Secure Remote Reconfiguration of FPGAs

Karim M. Abdellatif, Roselyne Chotin-Avot and Habib Mehrez

LIP6-SoC Laboratory, University of Paris VI,
LIP6-SoC, 4, Place Jussieu, 75252 Paris Cedex 05, France
Fax: (+33)(0)-44 27 72 80
E-mail: karim.abdellatif@lip6.fr
E-mail: Roselyne.Chotin-avot@lip6.fr
E-mail: Habib.Mehrez@lip6.fr

Abstract: Reconfiguration of FPGAs is becoming increasingly popular particularly in networking applications. In order to protect FPGA designs against attacks, secure reconfiguration must be performed. This paper presents efficient ASIC implementations of Authenticated Encryption (AE) algorithms, AES-CCM and AES-GCM, which are used in the static part of the FPGA in order to secure the reconfiguration process. Our focus on state of the art algorithms for efficient implementations leads to propose compact architectures to be used efficiently for FPGA bitstream security. Presented ASIC architectures were evaluated by using 90 and 130 nm technologies. Our comparison to previous work reveals that our architectures are more area-efficient.

Keywords: FPGAs; Secure Reconfiguration; Authenticated Encryption.

Biographical notes: Karim M. Abdellatif received his B.Sc (Honors) and M.Sc in Electrical Engineering from Minia University, Egypt, in 2008 and 2010, respectively. From July 2011 to November 2011, he was a visitor at Centre for High Performance Embedded Systems (CHiPES), Nanyang Technological University, Singapore. In 2012, he has joined Laboratoire d'informatique de Paris-6 (LIP6) at University of Paris VI, Paris, France, as a PhD student. His research concerns implementing cryptographic algorithms on hardware (ASIC and FPGAs).

Roselyne Chotin-Avot received the M.Sc. degree in Integrated System Architecture and Micro-Electronics from Paris VI university in 1999. She received the Ph.D. degree in Computer Science from Paris VI university in 2003. Her thesis addressed hardware architectures to control and estimate rounding errors. She is currently an associate professor at Paris VI university. Her research interests are on architectures that are specific to numerical algorithms for signal and image processing. She focuses on the design of the arithmetic datapaths for both CAD tools and arithmetic operator architectures.

Habib Mehrez is a Professor at Paris VI University and he is the team leader of CIAN (Analog and digital Integrated Circuits) of the SOC Department/LIP6 laboratory. For over 30 years, he has investigated research-oriented architectures of VLSI digital signal processing, arithmetic architectures, synthesis tools, reconfigurable architectures, testability and self-checking. He has supervised over 25 doctoral theses in VLSI. He co-authored over 90 publications in international journals and conferences.

1 Introduction

It has become obvious that embedded systems are integral parts of our every day needs. Field Programmable Gate Arrays (FPGAs) become integral parts of embedded systems. **Programmability** allows the functionality of a device to be modified outside of the founder. Adding this property to gate arrays gives us Field Programmable Gate Arrays (FPGAs). The use of FPGAs has been expanding from its traditional role in prototyping to mainstream production.

In order to redefine the functionality of the FPGA, a *bitstream* configuration file is uploaded into the FPGA. The reconfiguration includes downloading this bitstream file which contains the new design on the FPGA. The bitstream is processed by the static logic, a part of the FPGA that is not programmable. The user logic is the FPGAs reconfigurable part and where the user-defined application operates.

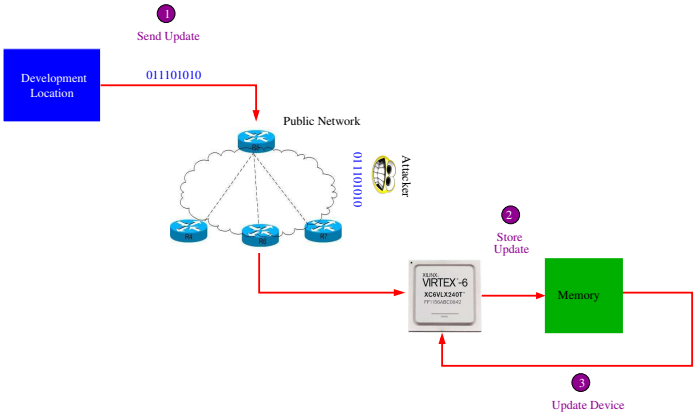

Figure 1 Remote reconfiguration

Reconfiguration of FPGAs is becoming increasingly popular particularly in networking applications and it is vital to provide security against malicious parties interfering with equipment functionality through this mechanism. Remote reconfiguration is attractive as it is used in such systems to offer new multimedia features or to repair eventual security issues. It requires transmitting the bitstream file which contains the hardware Intellectual Property (IP) over insecure communication channels and thus introduces new security issues as shown in Fig. 1.

The developer is faced with several problems resulting from sending the bitstream file through insecure network. An adversary attacker can detect the hardware IP to sell illegal copies of it or leak it to the public domain. There are several types of attacking could be occurred to the bitstream file:

- **Cloning attack:** The attacker takes a copy of the bitstream file to sell the IP illegally.
- **Reverse engineering attack:** It means that the attacker converts the bitstream file into a netlist file. Therefore, he detects the circuit layout and then he will be able to attack the device physically.
- **Tampering attack:** In this type of attacking, the attacker changes the design and sends it to the FPGA.

Our Contribution: (1) to give an overview of security issues used in reconfiguration of FPGAs and analyze how well they are suited to secure the reconfiguration process; (2) to analyze how well encryption and authentication are very important for trusted designs on FPGAs; (3) to propose compact hardware solutions which can be used efficiently for securing the reconfiguration of FPGAs.

A background about authenticated encryption is presented Section 2. Previous work of bitstream security is discussed in Section 3. Section 4 proposes compact architectures for AES-CCM and AES-GCM. Implementation details and performance comparison are discussed in Section 5. Section 6 discusses using the proposed architectures for bitstream security. Section 7 concludes this work.

2 Authenticated Encryption (AE)

Previously, confidentiality and authentication services have been implemented separately, by using different algorithms. Encryption algorithms are used to ensure confidentiality while Message Authentication Codes (MACs) can be used to provide authentication. When two separate algorithms are used to provide independent security services, it is considered cryptographically secure to use separate keys for each algorithm. Recently, techniques have been invented to combine encryption and authentication into a single algorithm which is called Authenticated Encryption(AE). Combining these two security services in hardware might support the following advantages:

- Area requirement for a single algorithm could be smaller compared to two separate algorithms.
- Designs with a smaller implementation area consume less power than larger designs.
- A slight advantage regarding key management and key storage issues because combined algorithm needs only a single key for both encryption and authentication.

2.1 ***AES-CCM***

Counter with Cipher Block Chaining-Message Authentication Code (CCM) [Dwo04] can be used in conjunction with any approved 128-bit block cipher like AES. It is designed for packet environment, where all the necessary data is available in storage before CCM processing. This implies that it is not online. CCM has been specified in the draft IEEE 802.11i standard for wireless networks. It has also been specified in IEEE 802.15 (Wireless Personal Area Networks) and 802.16 (Broadband Wireless Metropolitan Area Networks) standards.

Fig. 2 shows the block diagram of CCM. Firstly, the plaintext P is stored in a memory. Secondly, Y is generated using CBC mode, this value is used for authentication. Finally, CTR mode is used to generate ciphered text C. CCM is not suitable for on line applications as all data must be stored in memory before CCM processing.

Another useful feature of CCM mode of operation is that it can handle associated data (i.e. data which must be authenticated but not encrypted. This might be particularly useful in networking applications where data blocks like packet headers are usually sent in the clear, but the receiver must be able to ascertain their source).

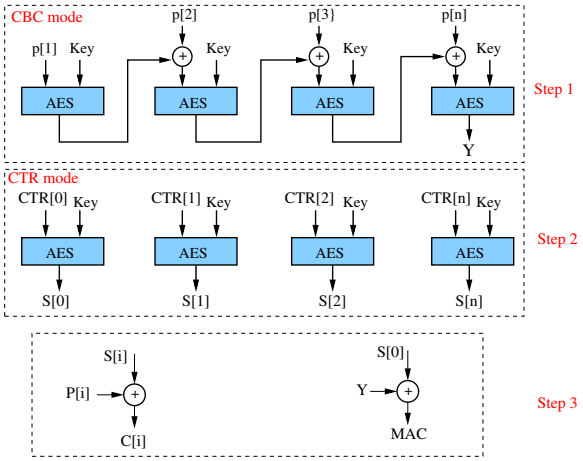

Figure 2 CCM mode of operation

2.2 ***AES-GCM***

Recently, Galois Counter Mode (GCM)[MV05] was considered as a new mode of operation of Advanced Encryption Standard (AES). GCM simultaneously provides confidentiality, integrity and authenticity assurances on the data. It supports not only high speed authenticated encryption but also protection against bit-flipping attacks. It can be implemented in hardware to achieve high speeds with low cost and low latency. Software implementations can achieve excellent performance by using table-driven field operations. GCM was designed to meet the need for an authenticated encryption mode that can efficiently achieve speeds of 10 Gbps and higher in hardware. It contains an AES engine in CTR mode and a Galois Hash (GHASH) module as presented in Fig. 3.

The authentication mechanism within GCM uses GHASH, that features multiplication by the hash subkey, within a binary Galois field. The hash subkey, denoted H, is generated by applying the block cipher to the zero block. GHASH is based on GF(2^{128}) multiplier with irreducible polynomial $F(x) = x^{128} + x^7 + x^2 + x + 1$ as described in [MV05].

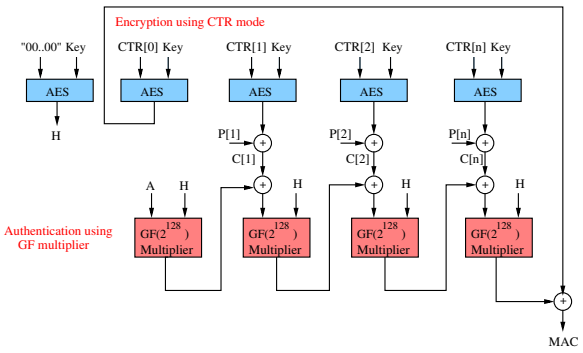

Figure 3 GCM mode of operation

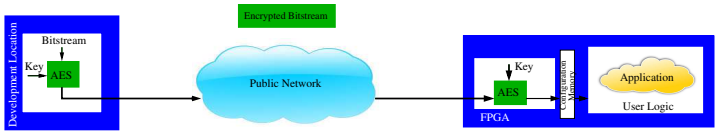

Figure 4 Bitstream encryption

3 Bitstream Security

This section describes the existing methods for securing the bitstream file during the transmission through insecure channel.

3.1 ***Bitstream confidentiality***

In order to secure the bitstream file and prevent attacking, encryption is used. Encryption provides data confidentiality and privacy. FPGAs include hardwired mechanisms that ensure bitstream confidentiality [Les07].

Bitstream encryption, first introduced by Xilinx at the production level with Virtex II FPGAs to prevent device cloning and to protect the confidentiality of the design data. Each Virtex-4, Virtex-5, and Virtex-6 device have an on-chip Advanced Encryption Standard (AES) [Pub01] decryption engine to support encrypted bitstreams. The bitstream is encrypted with a symmetric key K shared between the FPGA circuit and the developer. Key setup is performed in a secure area by the developer before the system is shipped. The encrypted bitstream is decrypted using the static logic as shown in Fig. 4. This mechanism allows for protection of the system designer's IP against cloning as well as reverse engineering.

This behavior is not enough to prevent attackers from destroying the FPGA remotely using malicious bit-stream combinations. Therefore, the FPGA should accept only bitstreams from an authenticated source.

3.2 ***Bitstream integrity***

Tampering attack is based on the modification of the bitstream. Therefore, the FPGA must be smart enough to detect the concept of **Who is the sender?**, to accept the correct bitstream sent by the trusted sender. Some FPGA vendors implement Cyclic Redundancy Checks (CRC) [Tse05]. However, the purpose of CRC is to detect transmission errors, not to check the integrity of data in the cryptographic sense. This is why Xilinx [Xilb] suggested using Message Authentication Code (MAC) function to ensure the integrity of the bitstream. Virtex-6 FPGAs are the first (and only) programmable devices to offer cryptographically strong bitstream authentication. An on-chip bitstream keyed-MAC algorithm implemented in hardware provides additional security beyond that of using AES bitstream encryption alone [Xilb]. Fig. 5 shows the architecture

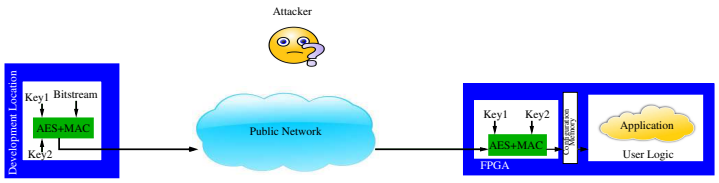

Figure 5 Bitstream encryption and authentication in Virtex6

used in Virtex-6. This concept allows for protection of the system designers IP against tampering attack.

Parelkar [Par05] noted that generic composition of authentication and encryption (AES+MAC) required more circuit area than authenticated encryption (AE) algorithms. The advantages of using one algorithm for both encryption and authentication are: smaller area, less power, and one key is used for encryption and authentication. Therefore, Parelkar [Par05] recommended counter with cipher block chaining-message (CCM) mode for achieving both authentication and encryption. The presented architecture of CCM in [Par05] used iterative design for AES where 16 s-boxes and 4 MixColumns were used.

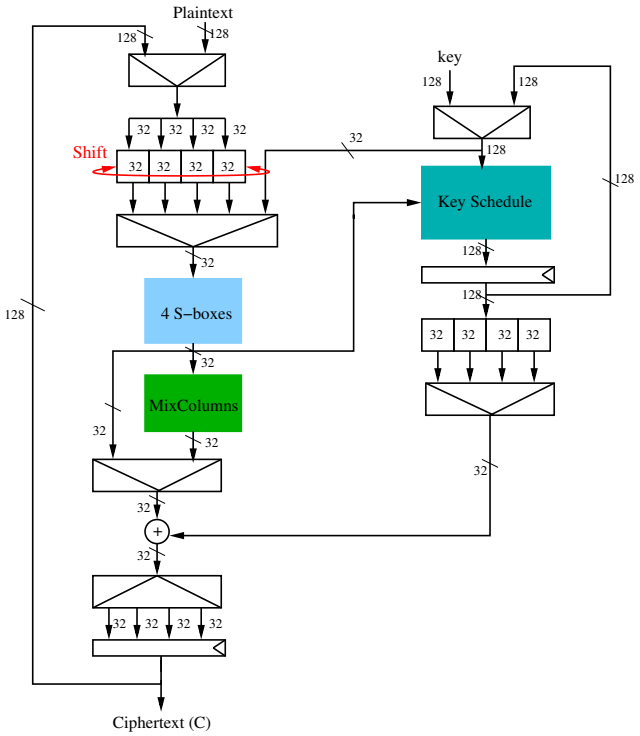

Figure 6 32-bit AES

4 Proposed Architectures of Authenticated Encryption

Our goal of this study is to design a compact solution to be used for encryption and authentication of bitstream. The reason of compact solution is to reduce the used area of the static part which performs the security task. Efficient hardware implementations for AE will be presented and compared with previous work. Presented architectures include AES-CCM and AES-GCM which can be used efficiently for secure reconfiguration of FPGAs.

4.1 Proposed CCM

In the proposed CCM, we use 32-bit AES (1/4 round) that has an advantage of reducing the consumed area with a suitable throughput that is able to support applications lower than 1Gbps.

Fig. 6 shows the architecture of 32-bit AES. The key schedule shares the s-boxes stage with the data bus. As a result, there are only four s-boxes used. Therefore, we can avoid the long data path resulting from using composite field approach by implementing a ROM to store the values of s-boxes. Moreover, only one MixColumn stage is used.

Our CCM architecture uses one 32-bit AES (1/4 round) for both encryption and authentication as shown in Fig. 7. All data must be stored in a memory. Firstly, authentication process is accomplished using CBC mode. Secondly, encryption process is performed using CTR mode.

Compared to Parelkar [Par05], the proposed architecture consumes only four s-boxes and one MixColumn instead of sixteen s-boxes and four MixColumns.

A 128-bit frame takes 55 clock cycles to be encrypted or added to MAC queue. The achieved throughput of our presented CCM is calculated as follows:

$$Throughput(Mbps) = \frac{128 \times F_{max(MHz)}}{55 \times 2} \quad (1)$$

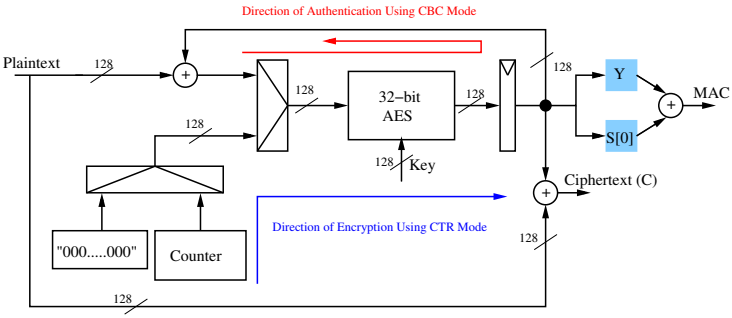

Figure 7 Proposed CCM

Table 1 Hardware comparison

Design	Architecture	Technology	Area mm^2	Frequency MHz	Throughput Mbps
This work	AES-CCM	130 nm	0.1407	285	331.6
This work	AES-GCM	130 nm	0.1615	285	663.2
This work	AES-CCM	90 nm	0.045	350	407.2
This work	AES-GCM	90 nm	0.064	344	800.5
Satoh et al. [SMTM01]	AES	110 nm	0.099	222.2	526.74
Parelkar et al.[Par05]	AES-CCM	90 nm	0.057	148	434
Parelkar et al.[Par05]	AES+HMAC	90 nm	0.183	101.2	1293

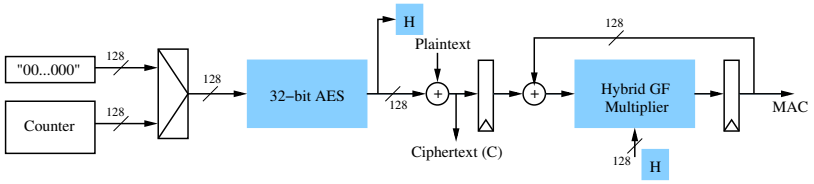

Figure 8 Proposed GCM architecture

Algorithm 1: GF(2^{128}) multiplier
Input A, H $\in$ GF(2^{128}), F(x) Field Polynomial.
Output C
C=0
for $i = 0$ to 127 do
if $A_i = 1$ then
$C \longleftarrow X \oplus H$
end if
if $H_{127} = 0$ then
$H \longleftarrow rightshift(H)$
else
$H \longleftarrow rightshift(H) \oplus F(x)$
end if
end for
return C

4.2 ***Proposed GCM***

AES-GCM uses two components: AES and GF(2^{128}) multiplier. The target must be directed to optimize the overall architecture which includes the encryption part (AES) and authentication part (GF(2^{128})).

Our proposed architecture uses 32-bit AES for area optimization. Previous architectures of GF(2^{128}) [ZMH07, SSA07] were used for high speed applications. Hence, it is important to design a multiplier which can be used efficiently with 32-bit AES. Serial GF(2^{128}) multiplier is described in **Algorithm 1** [MV05], where *A,H* are inputs to the multiplier and *F(x)* is the field polynomial, $F(x) = x^{128} + x^7 + x^2 + x + 1$. The output C needs 128 clock cycles to be ready in case of using serial multiplier.
As shown in **Algorithm 1**, it is possible to design one round for doing the multiplication in 128 clock cycles. Four rounds are used together to reduce the number of clock cycles needed to perform the multiplication from 128 to 32 (128/4) clock cycles in order to be suitable for 32-bit AES architecture.

Our proposed GCM architecture shown in Fig. 8 uses 32-bit AES with the hybrid GF(2^{128}) multiplier to accomplish the task of encryption and authentication respectively. First, the input to 32-bit AES block is zero's to perform H for the GF multiplier. Second, AES changes its mode to be in CTR mode for encryption while GF multiplier performs authentication task. The throughput of the proposed GCM is as follows:

$$Throughput(Mbps) = \frac{128 \times F_{max(MHz)}}{55} \quad (2)$$

5 Hardware Comparison

This section compares our presented architectures with the previous work. Presented architectures have been implemented using 90 and 130 nm CMOS standard cell library and its performances are compared with the prior art in Table 1.

Most of previous publications used FPGAs for implementation [AI07, DGP10, CG03]. As a result, we selected only ASIC implementations for comparison (compact architectures).

In terms of using 90nm technology, the proposed CCM occupies 0.045 mm^2 with 350 MHz as a maximum frequency and GCM needs 0.063 mm^2 with operating frequency 344 MHz. Although Parelkar et al. [Par05] presented one architecture of 128-bit AES for both encryption and authentication, the performance (area) of our CCM is better because we used 32-bit AES which has only four s-boxes. Also, [Par05] presented AES with HMAC but this method consumes more area compared to our CCM.

In terms of GCM mode, our proposed architecture presents better area and speed compared to AES with HMAC of [Par05].

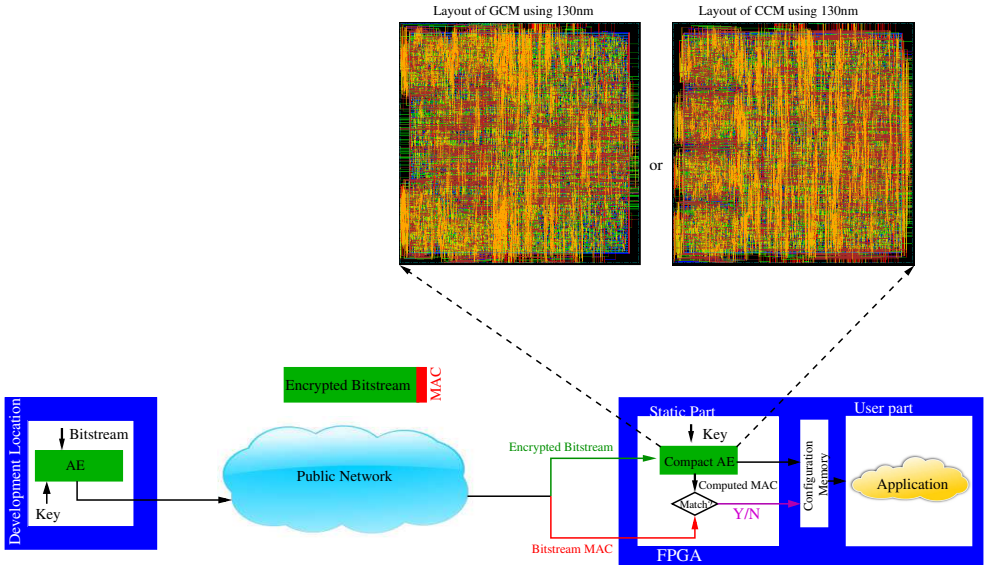

Figure 9 Bitstream security using AE

6 Proposed architectures for Bitstream Security

This section describes how our compact architectures can be used for bitstream security. AE is used in the static part of the FPGA as shown in Fig. 9. The encrypted bitstream is decrypted using AE. Also, AE is used to compute the MAC and compare it with the bitstream's MAC. If they are equal, the FPGA will continue to the startup sequence. Otherwise, configuration will abort and the cells be cleared.

The adopted solutions meet the current configuration throughput and not adversely affect total configuration times. Table 2 shows the maximum throughput of the largest family members of recent FPGAs.

Unlike current FPGAs [whi10],[Xilc] which support only encryption for bitstream security, our compact solutions add encryption and authentication in order to enhance the security of the configuration process.

Table 2 Configuration throughput of some FPGA family members

FPGA	device	Technology	Throughput
Virtex-5[Xila]	LX330T	65-nm	800 Mbits/s
Spartan-3 [Xild]	5000	90-nm	400 Mbits/s

7 Conclusion

This paper proposes low cost solutions for bitstream security. This is achieved by compact architectures for authenticated encryption (AES-CCM and AES-GCM). In order to minimize the hardware size of CCM mode, one 32-bit AES is used for both encryption and authentication. Also, GCM mode uses 32-bit AES for encryption and GF(2^{128}) hybrid multiplier for authentication. Presented architectures were evaluated through ASIC implementation. Future work deals with securing the proposed solutions against side channel attack.

References

[AI07] A. Aziz and N. Ikram. An FPGA-based AES-CCM crypto core for IEEE 802.11 i Architecture. *International Journal of Networks Security*, 5(2):224–232, 2007.

[CG03] P. Chodowiec and K. Gaj. Very Compact FPGA Implementation of the AES Algorithm. *Cryptographic Hardware and Embedded Systems-CHES*, pages 319–333, 2003.

[DGP10] S. Drimer, T. Güneysu, and C. Paar. DSPs, BRAMs, and a pinch of logic: Extended recipes for AES on FPGAs. *ACM Transactions on Reconfigurable Technology and Systems (TRETS)*, 3(1):3, 2010.

[Dwo04] Morris J Dworkin. SP 800-38C. Recommendation for Block Cipher Modes of Operation: the CCM Mode for Authentication and Confidentiality. 2004.

[Les07] A. Lesea. IP Security in FPGAs. *Xilinx http://direct. xilinx. com/bvdocs/whitepapers/wp261. pdf*, 2007.

[MV05] D. McGrew and J. Viega. The Security and Performance of The Galois/Counter Mode (GCM) of Operation. *Progress in Cryptology-INDOCRYPT 2004*, pages 377–413, 2005.

[Par05] M.M. Parelkar. *Authenticated Encryption in Hardware*. PhD thesis, George Mason University, 2005.

[Pub01] N.F. Pub. 197: Advanced Encryption Standard (AES). *Federal Information Processing Standards Publication*, 197:441–0311, 2001.

[SMTM01] A. Satoh, S. Morioka, K. Takano, and S. Munetoh. A Compact Rijndael Hardware Architecture with S-box Optimization. *Advances in CryptologyASIACRYPT 2001*, pages 239–254, 2001.

[SSA07] A. Satoh, T. Sugawara, and T. Aoki. High-Speed Pipelined Hardware Architecture for Galois Counter Mode. *Information Security*, pages 118–129, 2007.

[Tse05] C.W. Tseng. Lock Your Designs with the Virtex-4 Security Solution. *XCell Journal, XILINX, Spring*, 2005.

[whi10] Altera whitepaper. Design Security in Stratix III Devices. *available at: http://www.altera.com/literature/wp/wp-01010.pdf*, 2010.

[Xila] Xilinx. Virtex-5 FPGA Data Sheet:DC and Switching Characteristics.

[Xilb] Xilinx. Virtex-6 FPGA Configuration User Guide.

[Xilc] Xilinx Commercial Brochure. Lock your designs with the virtex-4 security solution.

[Xild] Xilinx1. Spartan-3 FPGA family:Complete data sheet.

[ZMH07] G. Zhou, H. Michalik, and L. Hinsenkamp. Efficient and High-Throughput Implementations of AES-GCM on FPGAs. In *International Conference on Field-Programmable Technology*, pages 185–192. IEEE, 2007.