Practical Analysis of Xilinx FPGAs' Bitstream Encryption

Eunchae Lee
Dept. of Electronics Engineering
Chungnam National University
Daejeon, South Korea
eclee.cas@gmail.com

Soyeon Choi
Dept. of Electronics Engineering
Chungnam National University
Daejeon, South Korea
sychoi.cas@gmail.com

Jiho Park
Dept. of Electronics Engineering
Chungnam National University
Daejeon, South Korea
jhpark.cas@gmail.com

Sungkyun Shin
Dept. of Electronics Engineering
Chungnam National University
Daejeon, South Korea
sgshin.cas@gmail.com

Hoyoung Yoo
Dept. of Electronics Engineering
Chungnam National University
Daejeon, South Korea
hyyoo@cnu.ac.kr

2023 International Conference on Electronics, Information, and Communication (ICEIC) | 979-8-3503-2021-3/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICEIC57457.2023.10049905

***Abstract*— Due to its speed and adaptability, SRAM-based Field Programmable Gate Array (FPGA) is extensively employed in various application fields. Since bitstreams stored in external memory are vulnerable to malicious attacks, most FPGA manufacturers provide bitstream encryption. In this paper, the structural differences between an unencrypted bitstream and an encrypted bitstream for Xilinx's FPGA series are investigated. First, the encryption algorithms used for each series are described, followed by the procedure for generating an encrypted bitstream using Xilinx ISE and Vivado. Lastly, the difference between the unencrypted and encrypted bitstreams for Xilinx FPGAs before 7-Series, 7-Series and after 7-Series is compared. Using the analysis, we demonstrated that it can be possible to verify if encryption is applied or not in real world utilizing encryption information extracted from the bitstream during its transfer from external memory to FPGA.**

Keywords*— *Xilinx FPGA, Bitstream encryption, Xilinx design suite.

I. INTRODUCTION

A field-programmable gate array (FPGA) is a semiconductor device with a programmable internal circuit and a programmable logic device. A HDL-implemented circuit is converted to a bitstream format, stored in an external memory, and transmitted to an SRAM-based FPGA. The bitstream transmitted to the FPGA may currently be vulnerable to attacks such as malicious cloning and reverse engineering. Consequently, the great majority of FPGA manufacturers offer bitstream encryption, where Advanced Encryption Standard (AES) is typically used as the cryptography algorithm. Xilinx, which has the largest market share among SRAM-based FPGAs, implements circuits using either ISE Design Suite (ISE) or Vivado Design Suite (Vivado), depending on the FPGA series. ISE supports FPGAs developed before 7-Series, whereas Vivado supports FPGAs developed after 7-Series, including 7-Series. Both design suites support bitstream encryption to protect circuit designs from adversaries. In this paper, we present the bitstream encryption algorithm according to different Xilinx's FPGA series and analyze the structural differences between the unencrypted bitstream and the encrypted bitstream for each series.

II. BACKGROUND

A. AES Mode

AES is a block cipher algorithm that uses symmetric key, meaning that the same secret key is used for both encryption and decryption [1]. AES operates in several modes, rather than in a single mode, to improve its cryptographic capability. In general, AES modes can be divided into two groups: those that provide only confidentiality, such as Electronic Codebook (ECB) and Cipher Block Chaining (CBC), and those that provide both confidentiality and authentication, such as CBC-MAC (CCM) and Galois/Counter Mode (GCM). Depending on the FPGA series, Xilinx uses different encryption algorithms for bitstream encryption. Classic FPGAs from 5-Series to 7-Series use AES-CBC, which guarantees only confidentiality using a 256-bit AES-symmetric key, whereas after 7-Series employs 256-bit AES-GCM, which simultaneously supports confidentiality and authentication. Note that, the FPGA series prior to 5-Series do not provide bitstream encryption. Fig. 1 depicts the operation of AES-CBC and AES-GCM modes. AES-CBC as shown in Fig. 1(a) is encrypted after the first plaintext block is XORed with the Initial Vector (IV), and then the following plaintext blocks are XORed with the previous ciphertext block. AES-GCM consists of two components, as shown in Figure 1(b): Counter

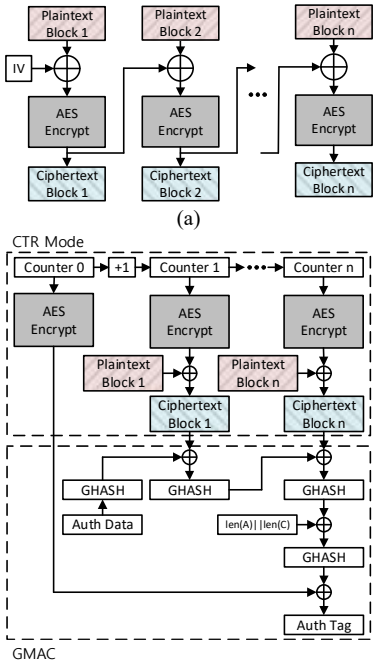
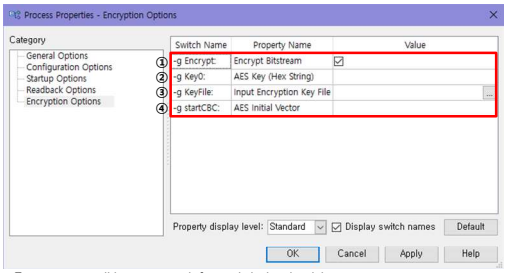
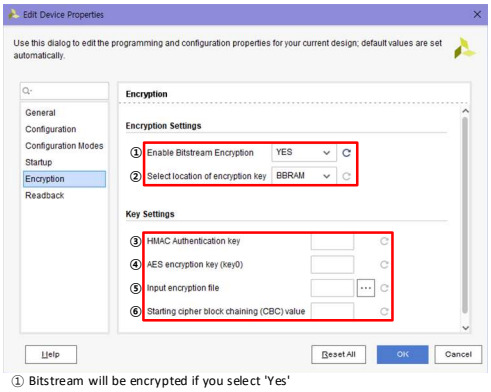


Figure 1. Algorithm of (a) AES-CBC and (b) AES-GCM.



① Bitstream will be encrypted if you click the check box
② AES key will be generated automatically if you do not enter a key value
③ You can input a key file (.nky file)
④ AES IV will be generated automatically if you do not enter a IV value

(a)



① Bitstream will be encrypted if you select 'Yes'
② You can select internal storage to store key
③ HMAC key will be generated automatically if you do not enter a key value
④ AES key will be generated automatically if you do not enter a key value
⑤ You can input a key file (.nky file)
⑥ AES IV will be generated automatically if you do not enter a IV value

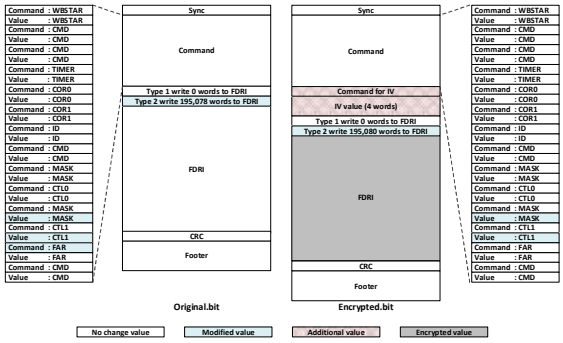
(b)

Figure 2. Bitstream encryption in Xilinx design suite (a) ISE and (b) Vivado.

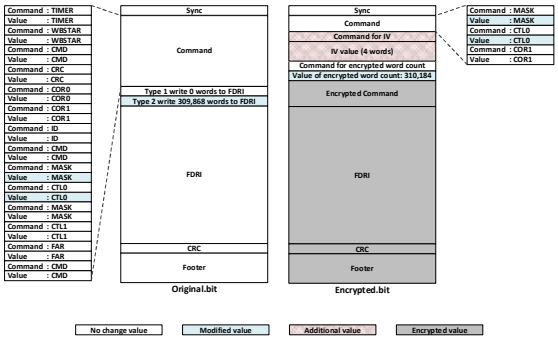
(CTR) mode and Galois Message Authentication Code (GMAC). Each plaintext block is XORed with the encrypted counter in the CTR mode part, and the ciphertext block and GHASH are XORed to generate an authenticated tag in the GMAC mode part.

B. Xilinx Bitstream Encryption

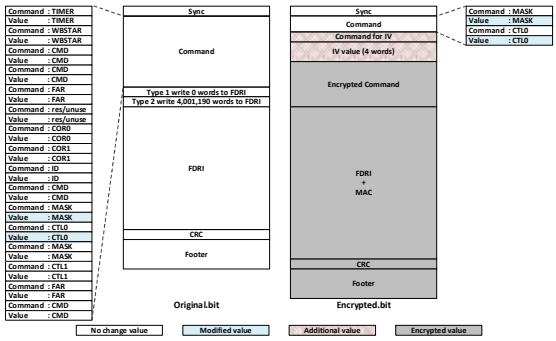
Fig. 2 shows the practical procedures for generating an encrypted bitstream using Xilinx ISE and Vivado [2]. In the case of ISE depicted in Fig. 2 (a), HDL completed the circuit implementation through synthesis and place-and-route, after which the AES key and IV value for encryption are inserted. During the configuration, AES key and IV can be set by the circuit designer or by ISE-generated random values. The AES key and IV value are stored in the key file (.nky file) while generating an encrypted bitstream. According to Xilinx official document [2], the FPGA handles the AES key and IV value differently. The key value is stored within the FPGA's internal key storage, and only JTAG is utilized to transmit the key file to the FPGA. On the other hand, the IV value is transmitted to the FPGA through JTAG or other configuration interfaces and is included in the command portion of the encrypted bitstream. Furthermore, Fig. 2 (b) depicts the use of Vivado. The procedure for configuring the AES key and IV value is identical to that of ISE in Fig. 2 (a) [3, 4]. Since FPGAs supported by Vivado provides authentication, so the authentication key should be set by the circuit designer or an arbitrary value generated by Vivado. Vivado also offers a choice for the key storage, which should determine whether the FPGA's internal key storage is battery-backed RAM (BBRAM) or eFUSE. The following programming method is similar to that of ISE.



(a)



(b)



(c)

Figure 3. Bitstream structure of (a) before 7-Series, (b) 7-Series, and (c) after 7-Series.

III. Analysis of Bitstreams

In this paper, we investigate the unencrypted bitstream *original.bit* and the encrypted bitstream *encrypted.bit* to analyze different design suite and various types of FPGA series. Virtex-5 LX20 device is used for ISE-supported FPGA bitstream analysis, while Artix-7 12T device of 7-Series and Kintex-UltraScale KU040 device of UltraScale are used for Vivado-supported FPGA bitstream analysis.

A. Classic FPGAs before 7-Series

Fig. 3 (a) compares the structure of unencrypted and encrypted bitstreams [2, 7] for the Virtex-5 LX20 device. The sync word (32'hAA995566) appears prior to the command. The command is identical for encrypted and unencrypted bitstream, including WBSTAR, CMD, TIMER, COR0, COR1, ID, MASK, CTL0 and CTL1 with their respective values. The important distinction is between IV-related commands and values, with the command (32'h30016004) representing the word count of IV and the subsequent IV value. Therefore, it appears that the length of the encrypted bitstream is longer

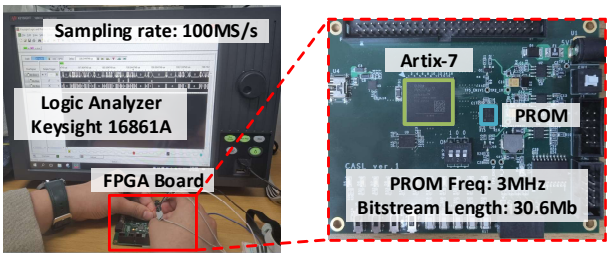


Figure 4. Experimental environment.

than the length of the unencrypted bitstream. Next, the word counts for Type 1 and Type 2 are then displayed, with Type 1 word counts of 0 for both bitstreams and Type 2 word counts of 195,078 for the unencrypted bitstream and 195,080 for the encrypted bitstream. The most important part in the bitstream is Frame Data Register Immediate (FDRI) that indicates the binary use of all FPGA hardware elements. FPGAs prior to the 7-Series use AES-CBC to encrypt FDRI when the encryption is applied, leading to a significantly different appearance. The desync is then concluded by the sequential appearance of Cyclic Redundancy Check (CRC) to determine the integrity of FDRI, footer with a command indicating the end of the bitstream. Footer and desync are the same in both bitstream structures

B. 7-Series FPGAs

Fig. 3 (b) compares the structure of the Artix-7 12T device's unencrypted and encrypted bitstreams [5, 7]. The overall comparison is similar to FPGAs prior to the 7-series. The structure of the unencrypted bitstream is identical across all series, and the encrypted bitstream of 7-Series FPGAs includes an IV command (32'h30016004) and a 4-word IV value. One important difference is that 7-Series FPGAs have same length of the unencrypted bitstream and the encrypted bitstream, despite the addition of IV. On 7-Series FPGAs, the difference between the unencrypted bitstream and the encrypted bitstream is the presence of Type 1 and Type 2 word counts. The unencrypted bitstream contains 0 Type 1 words and 309,868 Type 2 words, whereas the encrypted bitstream contains a command for encrypted word count (32'h30034001) and 310,184 encrypted words. In contrast to previous 7-Series FPGAs, AES-CBC is applied to FDRI, CRC, and footer of 7-Series FPGAs.

C. After 7-Series FPGAs

Fig. 3 (c) compares the structure of the unencrypted and encrypted bitstreams of the Kintex-UltraScale KU040 device [6]. The encrypted bitstream structure of the after 7-Series is basically equivalent to that of the 7-Series FPGAs, with the IV command (32'h30016004) and IV value (4 words) following the command. The presence or absence of Type 1 word count and Type 2 word count distinguishes the unencrypted bitstream from the encrypted bitstream. In the after 7-Series, the unencrypted bitstream reveals a Type 1 word count of 0 and a Type 2 word count of 4,001,190, whereas the encrypted bitstream reveals encrypted commands using AES-GCM after command. After 7-Series FPGAs are encrypted with the addition of MAC to FDRI, which appears after certain encrypted commands, resulting in a longer encrypted bitstream than an unencrypted bitstream. As with 7-Series FPGAs, the CRC and footer are included in encryption.

IV. EXPERIMENTAL RESULT

Based on the analysis depicted in Fig. 3, we demonstrated that encryption information extracted from the bitstream can be used to determine whether or not encryption is applied in the real world. Due to space limitations, only the Artix-7 xc7a12tcpg238 as from 7 series that supports Vivado is presented in this paper. The Artix-7 FPGA board's external memory PROM has a size of 30.6 MB and a transmission rate of 3MHz. Fig. 5 shows experimental environment to extract encrypted bitstream transmitted from the external memory to the FPGA. The bitstream is extracted from the PROM to the FPGA using the Logic Analyzer Keysight 16861A at a sampling rate of 100MS/s. According to the experiments, IV values are the same in the extracted bitstream and the generated key file (.nky file). Therefore, we checked IV-related commands and values in the extracted bitstream in order to know whether the encryption is applied or not. The same method is applicable to all Xilinx FPGAs including Virtex-5 LX20 and Kintex-UltraScale KU040.

V. CONCLUSION

This paper first analyzed structural differences between the unencrypted bitstream and the encrypted bitstream, and then described the practical analysis using the extracted bitstream. The experiments exemplified Xilinx Artix-7 and verified that it is possible to determine whether or not encryption is applied by investigating the extracted bitstream. The future research objective is to fully restore the original bitstream using secret keys and IV.

ACKNOWLEDGMENT

This research was supported by Basic Science Reserarch Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2021R1I1A 3055806), supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. 2022R1A5A8026986), and the EDA tool was supported by the IC Design Education Center (IDEC), Korea.

REFERENCES

[1] National Institute of Standards and Technology, Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, [Online]. Available: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.197.pdf

[2] Xilinx Virtex-5 FPGA Configuration User Guide *(UG191) (v3.13)*, Xilinx: San Jose, CA, USA, 2020. [online] Available: https://docs.xilinx.com/v/u/en-US/ug191

[3] Xilinx Using Encryption to Secure a 7 Series FPGA Bitstream *(v1.2)*, Xilinx: San Jose, CA, USA, 2021. [online] Available: https://docs.xilinx.com/v/u/en-US/xapp1239-fpga-bitstream-encryption

[4] Xilinx Using Encryption and Authentication to Secure an UltraScale/UltraScale+ FPGA Bitstream *(v1.5)*, Xilinx: San Jose, CA, USA, 2022. [online] Available: https://www.xilinx.com/content/dam/xilinx/support/documents/application_notes/xapp1267-encryp-efuse-program.pdf

[5] Xilinx 7 Series FPGAs Configuration User Guide *(UG470) (v1.15)*, Xilinx: San Jose, CA, USA, 2022. [online] Available: https://docs.xilinx.com/v/u/en-US/ug470_7Series_Config

[6] Xilinx UltraScale Architecture Configuration User Guide *(UG570) (v1.16)*, Xilinx: San Jose, CA, USA, 2022. [online] Available: https://docs.xilinx.com/v/u/en-US/ug570-ultrascale-configuration

[7] S. M. Trimberger, J. J. Moore, "FPGA security: Motivations, features, and applications," in Proceedings of the IEEE, vol. 102, no. 8, pp. 1248-1265, 2014