Does stream cipher-based scan chains encryption really prevent scan attacks?

Mathieu da Silva, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre

Does stream cipher-based scan chains encryption really prevent scan attacks?

Mathieu Da Silva, Marie-Lise Flottes, Giorgio Di Natale, Bruno Rouzeyre
LIRMM (Université Montpellier/CNRS)
161 rue Ada, Montpellier, France
{mathieu.da-silva,flottes,dinatale,rouzeyre}@lirmm.fr

***Abstract*—Scan chains offer facilities to steal secret information embedded in a circuit. For instance on a crypto-processor, collecting data related to the round register leads to reveal the secret key used for encryption. To protect against this threat while maintaining the test and debug features, countermeasures are implemented on the test interface or on the scan chain itself. Among the countermeasures, some of them are based on stream ciphers, consisting in the encryption of both controlled and observed data in the scan chain. We show in this paper that the stream cipher protection presents vulnerabilities if the keystream generated by the stream cipher is identical after a reset.**

Keywords—Security; Scan Attacks; Stream Cipher; Differential attack

I. INTRODUCTION

To ensure quality and reliability of integrated circuits (ICs), a design approach called Design-for-Testability (DfT) is used for providing testability on the circuit under test. The common DfT solution is the insertion of scan chains, which consists in replacing original flip-flops (FFs) by so-called scan FFs organized in shift-registers during the test phase. At test time, internal states of the circuit are controlled and observed by the serial input/output of the scan chain. The access to the scan chains is ensured by test interfaces. Several standards exist: IEEE 1149.1 [1] also known as JTAG for board testing, IEEE 1500 [2] for SoC testing, and IEEE 1687 [3] also known as IJTAG for embedded instruments testing.

While full control and observation on the internal states ensure testability, this is at the expense of security. For IC containing secret information, an attacker is able to exploit scanned out data in order to retrieve secret data, in particular the secret key of crypto-processors.

For instance on the Advanced Encryption Standard (AES [4]), the scan attack consists in observing the data stored in the round register by shifting out the scan chain content after the execution of the first round. It is assumed that the attacker knows the details of the encryption algorithm, and he has full control and observation of the scan chain. Yang et al. [5] have presented the scan attack on AES.

Improved scan attacks [6][7] have also been proposed to deal with more advanced DfT structure such as partial scan, response compactor, X-masking when the round register is not entirely observable.

Several countermeasures have been proposed to cope with these scan attacks. The most common industrial practice consists in disconnecting the test accesses after manufacturing by using fuses, preventing attackers to exploit the scan chains. Manufacturing test activities are not impacted, but maintenance in the field becomes an issue. A security threat of this method comes from probing techniques by re-connecting the test accesses.

Another countermeasure is based on an alternative DfT, named Built-In Self-Test (BIST [8]). This method prevents scan attacks since the external control and observation on the scan chains are limited. However, the BIST solution compromises diagnostic and debugging.

Further solutions use a secure test access to protect against the illegal use of the test interfaces. The test interface is locked until the tester is authenticated with a password [9] or by a secure protocol [10][11]. These solutions are expensive in terms of area and test time, and require a key management to share the test-session keys to authorized users.

Another protection [12] is based on a secret obfuscating the scan chain content. The tester has to know the specific test procedure in order to scan-in and scan-out desired test data. The obfuscation approach is however not considered as strong as encryption.

The encryption of the test communication is another solution proposed to protect the test interface or the scan chains. Only authorized user with the knowledge of the secret key can read and send data to scan chains through the test interface. The scan chain content is either encrypted with lightweight block ciphers as in [13][14], or, with stream cipher in [15][16][17]. The choice of stream cipher vs block cipher is motivated by the expected smaller area overhead, and there are no issues about padding test data into block size as in the former. Stream ciphers generate a keystream from a secret key and an initial value (IV). The plaintext is XORed with the keystream in order to generate the ciphertext.

In this paper, we show how an attacker is able to carry out scan attacks even if a stream cipher encrypts the test communication. The remainder of this paper is organized as follows. Section II recalls the principle of the scan attacks on AES. Section III summarizes the state of the art on the protection based on stream ciphers. Section IV presents the differential attack on countermeasures based on stream cipher. Finally, Section V concludes the paper.

II. SCAN CHAIN ATTACK

The attack procedure in [5] consists in applying a plaintext during one clock cycle, corresponding to the first round of the AES. The attacker switches the circuit to test mode in order to scan out the partially encrypted result stored in the round register. These steps are repeated until the 128-

bits key is revealed. The applied plaintexts are closely selected by pairs with a difference of one bit, due to a property highlighted in [5]. When only one bit differs between two plaintexts, the property allows identifying a key byte uniquely if the Hamming distance between the two results obtained after one AES round is equal to 9, 12, 23 or 24. The attack strategy is thus to try pairs of plaintext until the difference between two intermediate results allows the attacker to determine a key byte. The attacker repeats the attack for every key byte to retrieve the entire AES key. On average, 32 plaintexts are required to determine a key byte. Overall, an attacker needs to apply an average of 512 plaintexts to retrieve the secret key.

III. STATE OF THE ART ON STREAM CIPHER-BASED COUNTERMEASURES

In [15], the authors propose the use of a Trivium [18] stream cipher in order to encrypt the content of the JTAG communication. This countermeasure is proposed to cope with attackers observing and controlling the test interface, and also against malicious devices in the JTAG chain. In addition to the data confidentiality ensured by encryption, the device authentication is ensured with a challenge/response protocol, and the message integrity is ensured too with the keyed-hash message authentication code (HMAC). The stream cipher is initialized by an IV hard-coded with fuses, and a secret key that is the response to a challenge sent by a user. An authorized user knows the response to any challenge. Conversely, an unauthorized user without the knowledge of the secret key (i.e. challenge/response pairs) should not be supposed to carry out scan attacks due to the encrypted bitstream. We will see in the following of the paper that a differential scan attack can be performed even if the test data are encrypted with the stream cipher.

The solution proposed in [16] is to protect cores against a malicious core inside the SoC. The countermeasure consists in encrypting test data of individual cores, preventing observation and control by the other cores. The tester generates a random key in order to encrypt parallel test data shifting through the specific core, thanks to a Trivium stream cipher. This key is shifted to the core via a parallel chain non-visible from other cores. The IV configuration is not described by the authors.

Stream ciphers are also used on IJTAG reconfigurable scan network (RSN) in [17]. A Trivium cipher encrypts the bitstream in order to protect against malicious embedded instruments. This solution aims also at protecting against external attacker wanting to use illegally embedded instruments. In order to achieve the required protection, other security features are present: (i) stub chains making difficult to unlock the Segment Insertion Bit (SIB) from an attacker without the knowledge of the RSN structure, (ii) a chain checker detecting too many shifts from an attacker trying to guess the RSN architecture. Concerning the Trivum cipher, the secret key is configured with fuses and the IV configuration is not described.

In the next section, we will present the differential attack applied on these countermeasures based on stream ciphers.

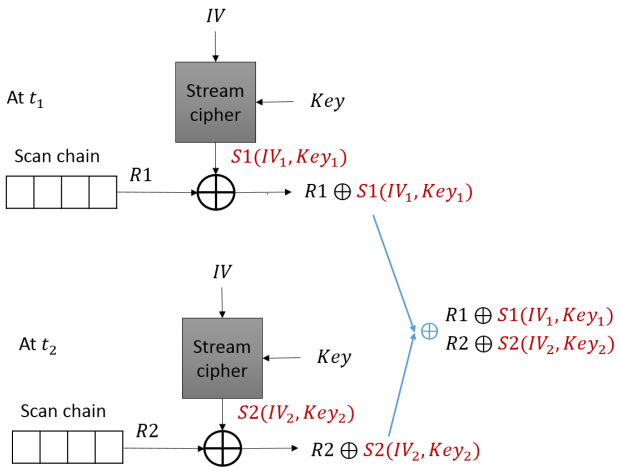

Fig. 1 Differential attack on protection based on stream cipher encryption

IV. DIFFERENTIAL ATTACK APPLIED ON STREAM CIPHER-BASED PROTECTION

For the attack, we assume that the attacker is external to the chip and is able to control and observe the test interface. The test communication is encrypted with a stream cipher whose the secret key is unknown from the attacker. After the initialization of the stream cipher, the test data are XORed with the keystream. The mode of operations is as follows: at t_i, the test response Ri, is XORed with the keystream $Si(IV_i, Key_i)$ generated from an initial value IV_i and a secret key Key_i.

The differential attack, described in the Fig. 1, consists in collecting the first encrypted response $R1 \oplus S1(IV_1, Key_1)$, then after a reset of the circuit, collecting the second encrypted response $R2 \oplus S2(IV_2, Key_2)$. The reset causes the reinitialization of the generated keystream from the stream cipher. Therefore, if the stream cipher generates the same keystream, i.e., if the key and IV do not change after a reset ($Key_1 = Key_2$ and $IV_1 = IV_2$), the difference between the two encrypted responses removes the impact of the stream cipher encryption since $S1(IV_1, Key_1) = S2(IV_2, Key_2)$. The scan attack described in [5] is thus applicable since it relies on calculating the Hamming distance between the two encrypted responses.

Thus, the differential attack circumvents the countermeasures based on stream ciphers ([15], [16], [17]).

The feasibility of the attack relies on the fact that the key and IV are kept the same after a reset. Thus, the IV or the key have to be different after a reset in order to prevent the differential attack. The main issue in changing the IV or the key is to share the different values to the authorized testers in order to decrypt the encrypted responses with the corresponding key and IV.

In [15], since the IV is hard-coded with fuses and the secret key is the response to a challenge sent by the user, the attacker needs to send the same challenge twice to carry out the differential scan attack. Even if the attacker does not know the response to the challenge sent, i.e. the secret key, the stream cipher will generate the same keystream for two different responses. The protection with the stream cipher encryption is thus insufficient against an external attacker. The solution can protect, nevertheless, against malicious

device in the JTAG daisy chain, since a malicious device can only sniff or modify encrypted test data without controlling the reset of the stream cipher. The authors propose to use in addition a HMAC signature on the test messages. This additional security feature ensures primarily the integrity of the messages, but the HMAC prevents also external control and observation from an attacker. Without the knowledge of the key used for the HMAC, the attacker cannot decrypt the hash sum, and cannot send message with the right hash sum. The countermeasure protects thus against external attacker at the expense of the hash function, implying both area cost and test time cost.

The solutions proposed in [16] for IEEE 1500 and [17] for IJTAG encrypt the test data. Since the IV configuration is not described by the authors, it can be assumed that the IV is fixed. An external attacker can therefore carry out the differential attack on encrypted test data, since the key is fixed in [17] and provided by the user in [16]. The proposed protections based on stream cipher are mainly intended to protect against an insider in the circuit, which cannot perform the differential attack due to the non-control of stream cipher reset. The solution is strictly restricted against this threat model and cannot be extended to an external attacker, unless adding extra security features like in [17].

V. CONCLUSIONS

Scan attacks are a threat against secure circuits. Some countermeasures are based on the bitstream encryption through a stream cipher. In this paper we describe a differential attack circumventing this protection. To make this attack ineffective, the key or the IV needs to change after a reset of the stream cipher. In this case, this poses an issue of sharing the changed values of IV or key to the authorized tester needing to decrypt the test responses. In the case of fixed IV and key, another countermeasure needs to be added to protect against an attacker controlling the test interface, implying additional costs.

ACKNOWLEDGMENT

This project has been funded by the French Government (BPI-OSEO) under grant FUI#20 TEEVA (Trusted Execution EVAluation).

REFERENCES

[1] Committee, I. S. (1990). IEEE Standard Test Access Port and Boundary-Scan Architecture. IEEE Std (Vol. 2001).

[2] IEEE Standard Testability Method for Embedded Core-based Integrated Circuits. (2012). IEEE Std 1500-2005.

[3] The IEEE Standards Association. (2014). IEEE Standard for Access and Control of Instrumentation Embedded within a Semiconductor Device.

[4] J. Daemen and V. Rijmen. The Design of Rijndael. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2002.

[5] Bo Yang, Kaijie Wu, and Ramesh Karri. Secure scan: a design-for-test architecture for crypto chips. In DAC, pp 135-140. ACM, 2005.

[6] Jean DaRolt, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. Scan Attacks and Countermeasures in Presence of Scan Response Compactors. In European Test Symposium, pp 19-24. IEEE Computer Society, 2011.

[7] Jean DaRolt, Giorgio Di Natale, Marie-Lise Flottes, and Bruno Rouzeyre. Are advanced DfT structures sufficient for preventing scan-attacks? In VTS, pp 246-251. IEEE, 2012.

[8] Marion Doulcier, Marie-Lise Flottes, Bruno Rouzeyre. AES-based BIST: Self-test, Test Pattern Generation and Signature Analysis. In 4th IEEE International Symposium on Electronic Design, Test & Applications, Hong-Kong, IEEE, pp.314-321, 2008.

[9] Chiu G.-M.; Li J.; C.-M. A Secure Test Wrapper Design Against Internal and Boundary Scan Attacks for Embedded Cores, IEEE Trans. on Very Large Scale Integration (VLSI) System, vol 20, issue 1, p 126-134, 2010

[10] Das, A., Da Rolt, J., Ghosh, S., Seys, S., Dupuis, S., Di Natale, G., … Verbauwhede, I. (2013). Secure JTAG implementation using schnorr protocol. Journal of Electronic Testing: Theory and Applications (JETTA), 29(2), 193–209.

[11] Pierce, L., & Tragoudas, S. (2013). Enhanced secure architecture for joint action test group systems. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 21(7), 1342–1345.

[12] Wang, X., Zhang, D., He, M., Su, D., & Tehranipoor, M. (2017). Secure Scan and Test Using Obfuscation Throughout Supply Chain. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 70(c).

[13] Da Silva, M., Flottes, M., Di Natale, G., Rouzeyre, B., Prinetto, P., & Restifo, M. (2017). Scan chain encryption for the test, diagnosis and debug of secure circuits. In 2017 22nd IEEE European Test Symposium (ETS) (pp. 1–6). IEEE.

[14] Da Silva, M., Flottes, M.-L., Di Natale, G., & Rouzeyre, B. (2017). Experimentations on scan chain encryption with PRESENT. 2017 2nd International Verification and Security Workshop, IVSW 2017, 45–50.

[15] Rosenfeld, K., & Karri, R. (2010). Attacks and defenses for JTAG. IEEE Design and Test of Computers, 27(1), 36–47.

[16] Rosenfeld, K., & Karri, R. (2011). Security-aware SoC test access mechanisms. Proceedings of the IEEE VLSI Test Symposium, 100–104.

[17] Kan, S., Dworak, J., & Dunham, J. G. (2017). Echeloned IJTAG data protection. Proceedings of the 2016 IEEE Asian Hardware Oriented Security and Trust Symposium, AsianHOST 2016.

[18] De Canniere, C., & Preneel, B. (2005). TRIVIUM Specifications. ECRYPT Stream Cipher Project, Report, 30, 2005.