A Lightweight, Plug-and-Play and Autonomous JTAG Authentication IP for Secure Device Testing

Sébastien Lapeyre, Nicolas Valette, Marc Merandat, Marie-Lise Flottes, Arnaud Virazel, Bruno Rouzeyre

A Lightweight, Plug-and-Play and Autonomous JTAG Authentication IP for Secure Device Testing

S. Lapeyre N. Valette M. Merandat
INVIA
Meyreuil, France
{firstname.lastname}@invia.fr

M.-L. Flottes B. Rouzeyre A. Virazel
LIRMM - University of Montpellier / CNRS
Montpellier, France
{firstname.lastname}@lirmm.fr

***Abstract*—As any other circuits, secure devices need to be tested to ensure their reliability. Nevertheless, test infrastructures, such as JTAG or scan chains, can maliciously be used to steal secret data stored or processed in secure devices. In this paper, we explore a lightweight solution to protect JTAG access based on a challenge-response authentication protocol. A JTAG-authentication dedicated IP is presented. Design alternatives for quick IP plug-and-play, security, area and test time optimization are presented and evaluated.**

Keywords—JTAG, security, lightweight cryptographic hash, authentication, challenge/response.

I. INTRODUCTION

In the context of constant growing IoT (Internet-of-Things) market product, providers must ensure the quality, reliability and security of their devices. Those properties cannot be treated independently [1]. For instance, it has been determined that attackers can use unsecured test infrastructures to achieve their malicious purposes [2]. Concomitantly, untested secure functions are not reliable.

One of the most efficient and commonly used test infrastructure is the JTAG, described in the IEEE Std.1149.1 [3]. It allows to run debug and test procedures on integrated circuits and printed circuits boards, through a serial communication. However, this interface can be used as very good, easy access and risky "backdoor" for hardware hacking [4][5][6].

Securing the access to JTAG facilities permits to defend circuits against those threats. One method relies on a protocol providing authentication of the Automated Test Equipment (ATE).

The first developed solution [7] was a password-based authentication, where the ATE must provide the correct password at the beginning of the test session in order to unlock critical/secure scan chains. This solution is now considered as a weak authentication solution especially when the password cannot be changed. Moreover, it does not protect against common attacks such as replay attack [8] or eavesdropping [9].

More effective solutions use a Challenge/Response-based authentication [10][11]. To enforce the solution security, the protocol relies on a cryptographic algorithm which has also a major impact in term of cost (area/timing). Solutions presented in [10] use a depreciated hash, which has been proven as non-secure to date [12]. More recent solutions [11] are based on the SHA-3 hash [13] which is the standard secure hash algorithm released by NIST. Nevertheless, the SHA-3 area footprint makes it not suitable for embedded systems.

The new cryptographic lightweight hash field is taking more and more attention in security development to deal with this issue. Cryptographic community pushed by the NIST agency are trying at this moment to standardize one algorithm [14]. Those hashes fit with security needs and low footprint of the test context. Other authentication protocols using asymmetric scheme/certificate exist [15] and offer higher security level. But to accomplish it, hardware and timing cost are not compatible with low-cost system requirements.

In this paper, we present a novel lightweight, plug and play and autonomous authentication solution in order to identify the ATE. The proposed authentication solution is built to reach the best compromise between area footprint and security level. For this purpose, a dedicated authentication IP is proposed with its protocol fully detailed. Plug-and-play facilities and alternatives for smaller area footprint are discussed, resulting into two versions of the proposed IP architecture according to the design effort to provide. Experimental data on two lightweight cryptographic hashes for secure authentication are presented. They can be used for a well informed choice in terms of security, area and test time overhead. To the best of our knowledge, it is the first authentication solution for JTAG using cryptographic lightweight hashes.

This paper is organized as follow. Section II presents the protocol used for the JTAG authentication. The proposed JTAG Authentication IP is detailed in Section III. Implementation results for different lightweight hashes are presented in Section IV. Section V concludes the paper.

II. AUTHENTICATION PROTOCOL

Considering threats against JTAG infrastructure and their potential consequences, JTAG access must be protected against illegal usage. Secure identity verification before use with authentication protocols, participates to such protection. These protocols are evaluated in terms of resilience against attacks, area footprint and execution time.

The proposed authentication protocol through JTAG is based on the SKID2 protocol [16]. This protocol protects against replay attack, eavesdropping, key recovery and brute force attack. The security level provided by SKID2 is thus correlated to the nonce generation, the cryptographic function used for the response generation and the authentication key length. This section presents main characteristics for SKID2 implementation in the context of ATE authentication.

A. Cryptographic Function

The cryptographic function aims at protecting the authentication key. Even if an attacker is eavesdropping the communication, the authentication key should not be recovered. This cryptographic function can be a block cipher, a stream cipher or a cryptographic hashes since those solutions can deliver the same amount of security requirements. This paper will focus on lightweight cryptographic hashes. Experimental results are presented in Section IV for 3 different hashes: two lightweight (ASCON and SPONGENT) and the SHA-3. The expected response is a cryptographic hash of the key concatenated with the nonce as input.

B. Communication Channel

JTAG uses a serial communication to shift-in input test patterns and shift-out test results. The proposed authentication protocol uses the same channel to exchange authentication

frames between ATE and DUT. It is a full duplex channel where when one bit is sent (i.e. shifted-in) one bit is received (i.e. shifted-out). In our case, since the ATE is the master, if the DUT needs to send data, the ATE must shift-in data too, to initiate the communication process. For this purpose, the Test Data Register (TDR) of the IEEE Std.1149.1 is used as a serial communication register.

C. Authentication Steps

Only two frames, containing the nonce and the ATE computed response respectively, are needed to securely authenticate an entity using the SKID2 protocol. In JTAG context, additional frames are added to start the authentication, to confirm the good reception or to have information on the authentication execution. Four frames, depicted in Fig. 1 (where "0*...*0" is a null frame), are mandatory to accomplish the authentication. The full duplex channel is illustrated with the double-headed arrows. The white part is the data shifted-in to the DUT and the grey one is the data shifted-out to the ATE.

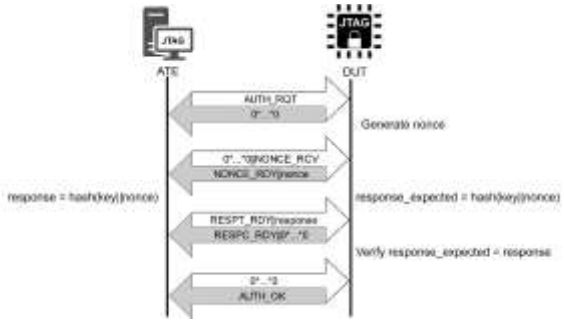

Figure 1: JTAG Authentication protocol

III. JTAG AUTHENTICATION IP

The proposed JTAG Authentication IP is in charge of implementing the authentication protocol. It has been designed to be embedded in a system on chip with secure purpose and low-area footprint, called secure device in this paper. Two architectural versions are proposed (see Fig. 2). A full plug-and-play IP including all required resources for an autonomous authentication and a lightweight IP for area improvement thanks to reusable resources in the host secure device. The main motivation for these two versions is to provide flexibility to the designer and reduce extra costs according system's resources availability.

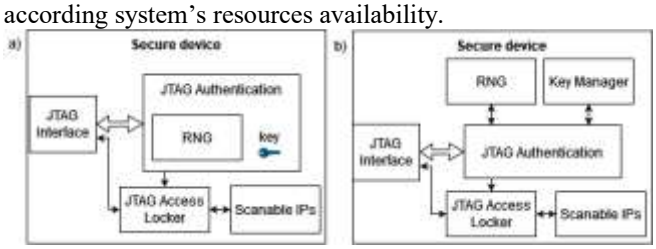

Figure 2: Architectural options with a) fully autonomous solution and b) semi-autonomous solution with borrowed security resources

The rest of this section describes all the different hardware actors used to realize the authentication protocol. Firstly, it presents the embedded secure device resources used to manage the JTAG access. Secondly, it explains which and why secure resources can be reused (random number and key(s) generation) leading to two different architecture proposals. And finally, it details all submodules of the JTAG Authentication IP.

A. Secure Device JTAG ressources used by the JTAG authentication IP

JTAG interface is composed of a JTAG ports and a Test Access Port (TAP) controller, which selects the Test Data Register (TDR) to be set by the test program under execution. The JTAG access Locker module "opens" or "closes" the TDR and scan chains access. It already exists different options to block the access such as Lock Segment Insertion BIT (LSIB) [17] and Secure SIB (SSIB) [18]. Configuration of internal scan chains for providing secure access to system's instruments is out of the scope of this paper. In any case, the proposed IP is used to trigger the JTAG Access Locker module after authentication.

B. Optional Secure Device Ressources

The JTAG Authentication IP may profit from security resources already embedded in the host secure device for secure applications. Main objectives with resource reuse is to provide flexibility to the designer and lower the implementation costs. The two resources optionally borrowed from the host secure device for the lightweight version of the Authentication IP are a Random Number Generator (RNG) and a key manager if any.

1) Random Number Generator

As explained in Section II, the chosen authentication protocol requires the generation of a new random number (i.e. a nonce) at every authentication. A RNG must be used to certify randomness (unpredictable and non-replayed numbers) of the provided nonce. We assume that a RNG is likely to be already implemented for mission mode execution when the system needs to execute secure applications. To reduce area overhead a native RNG can thus be borrowed from the host secure device and connected to the proposed lightweight version of the proposed Authentication IP (see Fig 2.b).

2) Key Manager

The authentication key required by the secure authentication protocol can be easily hardcoded in the proposed IP (Fig. 2.a). However, this solution prevents management flexibility and the possibility to use different keys for different purposes.

Using a key manager allows to manage different access right to the internal scan chains for different test/debug actors and different test environments at different times during the system's life cycle (i.e., post-manufacture, in the field…). Key manager reuse at system level allows substantial area saving if any.

RNG and Key Manager designs being out of the scope of this paper, the following sub-section details the lightweight, semi-autonomous, version of the IP architecture (Fig. 2.b).

C. JTAG Authentication IP implementation

The proposed JTAG Authentication IP runs under the external clock named Test clock (TCK). Considering that the nonce and the key generations are handled by the host secure device resources, it is composed of 3 sub-modules describes below and depicted in Fig. 3.

1) Test Data Register

The TDR allows a serial communication between the ATE and the JTAG Authentication IP as described in Section II.B.2. The data sent to the JTAG Authentication IP through the TAP are shifted-in through *scan_in_tdr* input and at the same time data sent to the ATE are shifted-out *through scan_out_tdr* output.

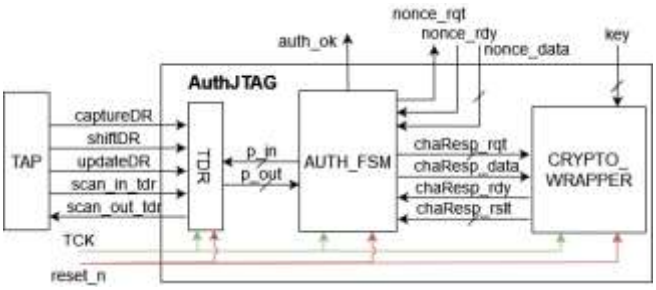

Figure 3: JTAG Authentication IP architecture

The TDR length depends on the largest frame length exchanged during the authentication protocol.

2) AUTH_FSM

The AUTH_FSM module manages the authentication protocol. Its finite state machine is depicted in Fig. 4.

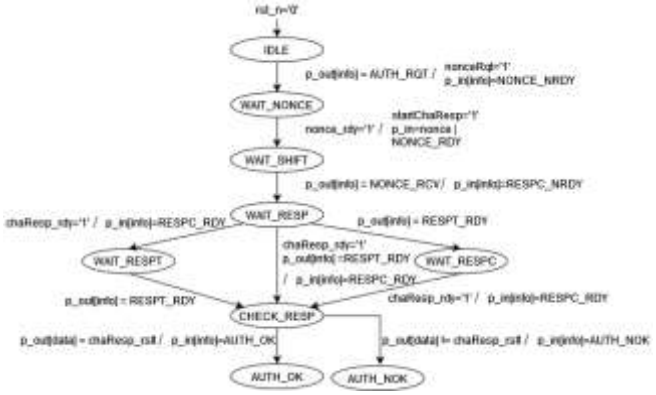

Figure 4: FSM authentication

To leave the AUTH_OK and AUTH_NOK states, a reset is needed. In case of wrong authentication, an attacker needs to throw a reset to try again.

The AUTH_FSM module also manages the frame transmission and reception through the TDR. Those frames are a combination of a status for frame authentication and data fields. Status field is used to identify frames. It is called INFO_ExtT for frame sent by the ATE (i.e. AUTH_RQT, NONCE_RCV and RESPC_RDY) and INFO_AuthIP when sent by the JTAG Authentication IP (i.e. NONCE_NRDY, NONCE_RDY, RESPC_NRDY, RESPC_RDY, AUTH_OK and AUTH_NOK). They are coded on 2 bits for INFO_ExtT and respectively 3 bits for INFO_AuthIP, as depicted in Fig. 5. Depending on the status value, data are valid or not (i.e. none).

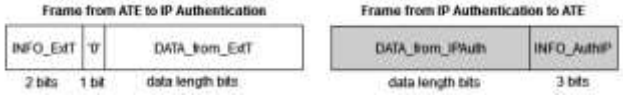

Figure 5: Frame format

3) CRYPTO_WRAPPER

The CRYPTO_WRAPPER module is in charge of the cryptographic process. This module concatenates the key and the nonce and computes the expected response with the embedded module in charge of performing the hash.

Using a dedicated submodule to compute hash contributes to the autonomous and plug and play properties of the proposed JTAG Authentication IP. On the other hand, using a hash already embedded in the host secure device could minimize the area cost but it is not straightforward to connect an existing hash to a new IP due to major design modification needed on it.

IV. Experimental Results

The lightweight semi-autonomous version of the JTAG Authentication IP (see Fig. 2.b) has been implemented in VHDL and verified with testbenches. This section details experimental results in terms of extra area and test execution time for different hashes implemented in order to compare their figures of merits and highlight the best candidate to be used in JTAG authentication context.

1) Area Result

The proposed JTAG Authentication IP must be as small as possible in term of area. Implementing lightweight cryptographic hashes is a good solution to minimize the area footprint. Several algorithms have been proposed by the cryptography community. We consider as case study the SPONGENT [19] and the ASCON [20] algorithms. The SPONGENT is considered because it has the smaller area footprint, a good throughput, and good cryptographic properties [21]. The ASCON hash has been created from the ASCON cipher, which won the European CEASAR competition [22]. It is one of the 10 finalists for the NIST lightweight cryptography project [14]. To the best of our knowledge, both solutions are not subject to attacks/vulnerabilities. The proposed JTAG Authentication IP was also implemented with the SHA-3 [13], which is the reference of cryptographic hash solutions.

SPONGENT, ASCON and SHA-3 are all sponge construction based [23]. Hash sponge construction based has 4 parameters that establish the security level and the number of state bits (equal to the capacity plus the rate): the output length, noted *n*, the capacity, noted *c*, the rate, noted *r,* and the number of rounds. To specify the type of hash, the used notation is HashName-n-c-r. SHA-3 and ASCON parameters are specified in the NIST documentation. The SPONGENT publication proposes different parameters. For a fair comparison between considered hashes, we have set the SPONGENT parameters to reach the same security level than the ASCON.

Experimental results were obtained using security parameters detailed in TABLE I. These parameters aim at reaching a high security level. In most usual cases, they can be lowered in order to reduce area cost.

TABLE I. Security Parameters

Parameter	Length
Key	128
Nonce	128
Response	256

TABLE II. Implementation Results

Hash used	Security properties (in bits)		# Gate (CMOS 55nm)
	Collision	*Pre & Second Image*	
SPONGENT 256/256/128 - 195 rounds	128	128	20038
SHA3 256/512/1088 - 24 rounds	128	256	41814
ASCON 256/256/64 - 12 rounds	128	128	20388

TABLE II. gives semi-autonomous JTAG Authentication IP (Fig. 2.b) area with the three hashes and their security level. The security level contains collision and pre and second image resistance. N-bits of security means 2^n operations are needed to find a collision or a pre/second image collision. As shown, ASCON and SPONGENT have twice less security bits for pre/second image collision, but the security level is still enough for our use case. Synthesis results, given in number of equivalent gates, have been obtained with Design Compiler from Synopsys. The CMOS 55nm technology node has been

used with a 10MHz frequency constraint. The JTAG Authentication IP has a similar low area footprint if it uses SPONGENT or ASCON while using SHA-3 doubles the gate count. These results confirm that using lightweight hashes decreases significantly the area needed for an authentication.

2) Timing Results

Apart for the area, the testing time is also an important parameter. The JTAG Authentication IP has been thought out within this problematic.

First, the ATE needs to authenticate just at the beginning of a test session. The *auth_ok* signal (Fig. 3) stays enable until a reset is used. Secondly, the design is optimized in order to minimize the necessary delay for the authentication protocol. The status field in frames has been placed deliberately at the beginning of frames sent by the ATE and at the end of frames received by the ATE. Consequently, if the ATE or DUT needs to communicate the code frame without any valuable data, they just need to shift-in 2 bits (i.e. from the ATE to the DUT) or shift-out 3 bits (i.e. from the DUT to the ATE) to get the information.

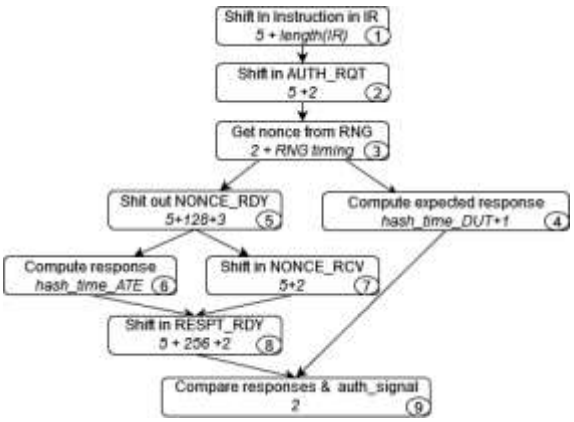

Figure 6: Authentication steps with timing

Fig. 6 depicts the different authentication steps with their corresponding timing in terms of clock cycles.

To resume, the time necessary to process an authentication is expressed by the following formula:

T_{auth} = 16 + length (IR) + RNG timing + max(*hash_time_DUT* + 1; 399 + max(*hash_time_ATE*; 7)

TABLE III. reports the time necessary to compute the expected response. The SPONGENT *hash_time_DUT* is significantly longer than the two others. Thereby, it is not as well suited as ASCON and SHA-3 to realize a fast JTAG authentication. Due to results presented in TABLE III. and the fact that *hash_time_ATE* time is longer than 7 cycles, with SPONGENT or SHA-3, the formula can be resumed to:

T_{auth} = 415+ length(IR) + RNG timing + *hash_time_ATE*

This result shows that the authentication timing is not impacted by the hardware hash implementation choice between ASCON and SHA-3 with security parameters detailed in TABLE I. Note that, lowering security parameters can reduce the authentication time.

TABLE III. COMPUTE EXPECTED RESPONSE TIMING

Hash used	***hash_time_DUT* (in clock cycles)**
SPONGENT 256/256/128 195 round	798
SHA3 256/512/1088 24 round	48
ASCON 256/256/64 12 round	108

As a conclusion, synthesis results confirm that using SPONGENT and ASCON lightweight hashes allows dividing the cost area by two, compared to a SHA-3 based solution. Moreover, timing result shows that the ASCON is x8 times faster than the SPONGENT in our use case. Consequently, ASCON appears to be the most effective hash to embed in the proposed JTAG Authentication IP. It is the best solution to minimize the area cost and authentication time.

V. CONCLUSION

In this paper, we proposed a lightweight JTAG authentication IP to authenticate an ATE by a DUT before test. Protocol principle, IP architecture, design alternatives are explored to provide two (full- and semi-) autonomous IP organization for quick plug-and-play or area saving thanks to system's security functions reuse. Experimental results in terms of extra area and authentication time demonstrated the interest of the proposed IP when using ASCON as lightweight cryptographic hash algorithm.

REFERENCES

[1] M. Wolf, et al., "Safety and Security of Cyber-Physical and Internet of Things Systems," Proc. of the IEEE, vol. 105, no. 6, pp. 983-984, 2017.

[2] G. Tshagharyan, et al., "Securing test infrastructure of system-on-chips," IEEE East-West Design & Test Symposium, pp. 1-4, 2016.

[3] "IEEE Standard 1149.1: Standard Test Access Port and BoundaryScan".

[4] K. Rosenfeld and R. Karri, "Attacks and Defenses for JTAG" IEEE Design & Test of Computers, vol. 27, no. 1, pp. 36-47, 2010.

[5] J. Da Rolt, et al., "New security threats against chips containing scan chain structures," Symp. on Hardware-Oriented Security and Trust, 2011.

[6] Free60. Xbox360 Hardware: JTAG/SMC Hack, http://free60.org/wiki/SMC_Hack, 2009.

[7] F. Novak and A. Biasizzo, "Security extension for IEEE Std 1149.1," JETTA, 22(3), 301–303, 2006.

[8] P. Syverson, "A taxonomy of replay attacks," Proceedings The Computer Security Foundations Workshop VII, pp. 187-191, 1994.

[9] Matsumoto, T and Imai, H. "Human identification through insecure channel", EUROCRYPT, pp. 409-421, 1991.

[10] C. Clark, "Anti-tamper JTAG TAP design enables DRM to JTAG registers and P1687 on-chip instruments," IEEE Int. Symp. on Hardware-Oriented Security and Trust, pp. 19-24, 2010.

[11] R. Baranowski, et al., "Fine-Grained Access Management in Reconfigurable Scan Networks," IEEE Trans. on Computer-Aided Design of ICs and Systems, vol. 34, no. 6, pp. 937-946, 2015.

[12] NIST Brief Comments on Recent Cryptanalytic Attacks on Secure Hashing Functions and the Continued Security Provided by SHA-1, http://csrc.nist.gov/news.highlights/NIST-brief-Comments-on-SHA1-attack.pdf.

[13] M.J. Dworkin, "SHA-3 standard: Permutation-based hash and extendable-output functions," Federal Inf. Process. Stds. , 2015.

[14] NIST Lightweight Cryptography Project. https://csrc.nist.gov/projects/lightweight-cryptography.

[15] Authenticated Debug Access Control Specification, ARM , 2020. [Online] Available https://developer.arm.com/documentation/den0101

[16] D. Basin, and C. Cas, "Evaluation of ISO/IEC 9798 protocols," Technical report, CRYPTREC, Version 2.0, 2011.

[17] J. Dworak, et al., "Don't Forget to Lock your SIB Hiding Instruments using P1687," Proceedings of the IEEE International Test Conference, 2013.

[18] R. Baranowski, et al., "Fine-Grained Access Management in Reconfigurable Scan Networks," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 6, pp. 937-946, 2004.

[19] A. Bogdanov, et al., "SPONGENT: A lightweight hash function," Int. workshop on cryptographic hardware and embedded system,, 2011.

[20] C. Dobraunig, and al., "Ascon v1. 2. Submission to NIST" 2019.

[21] https://www.cryptolux.org/index.php/Lightweight_Hash_Functions

[22] CAESAR: Competition for Authenticated Encryption: Security, Applicability, and Robustness, http://competitions.cr.yp.to, 2013.

[23] G. Bertoni, et al., "Sponge functions. In: Ecrypt Hash Workshop," May 2007. http://sponge.noekeon.org/SpongeFunctions.pdf.