DE GRUYTER OPEN

Journal of ELECTRICAL ENGINEERING, VOL 67 (2016), NO3, 192–198

SECURITY PROBLEMS OF SCAN DESIGN AND ACCOMPANYING MEASURES

Anton Biasizzo — Franc Novak *

The paper deals with the security problems of scan design and investigates currently proposed solutions. A solution based on data encryption to protect the data in scan chains is discussed and problems related to the block-based encoding are outlined. Next, security extension for IEEE Std. 1149.1 providing a locking mechanism is analysed. The mechanism prevents unauthorised users to interfere via test bus with the system normal operation. Possible attack scenario is considered and the probabilities of successful attack within a given time interval are calculated for different lengths of the Lock register. The paper concludes with the description of current work focused on improvements the security of the locking mechanism, in particular by using simplified public key infrastructure.

K e y w o r d s: test structures, boundary-scan test, security

1 INTRODUCTION

Some time ago, a discussion aroused on the security problems of systems incorporating scan-chains that enable the access to system's internal points and consequently facilitate testing [1–3]. Since scan-chain technology is used in embedded systems in different applications ranging from smart credit cards to process control systems in critical infrastructures supporting our everyday life, it is of vital importance to be aware of potential vulnerabilities and to provide mechanisms for ensuring system safety.

In this paper we investigate currently proposed solutions. We briefly review the principle of scan-chain test approach [4, 5]. This integrated circuit (IC) test technique is often combined with the popular boundary-scan approach formalized in IEEE Std. 1149.1 [6] at the board or system level [7, 8]. Two potential vulnerability problems in scan-based systems are considered: (a) scan-chain can be used to reveal IC internal structure, so hackers or other interested third parties can steal intellectual property, or (b) hackers or other unauthorised users can brake into a system and disturb its normal operation by executing an invasive test sequence which may lead to a catastrophic event. A solution based on data encryption to protect the data in scan chains [2] is discussed and problems stemming from packet encoding are outlined. The security extension for IEEE Std. 1149.1 providing a locking mechanism [9] is described. The mechanism prevents unauthorised persons to interfere with the system via IEEE Std. 1149.1 test port. Typical attack scenario is considered and analysed. In the last part, current work on improvements of the locking mechanism is presented.

2 SCAN DESIGN

Complex digital circuits are difficult to test. Deeply embedded registers and memories store data in many internal states that are hard to control and to observe via input/output pins. In order to efficiently test complex sequential logic, different design-for-test (DFT) techniques are employed.

Ad hoc DFT techniques that rely on good design practices learned from the past experience prove to be insufficient for larger designs. More powerful techniques such as scan design and built-in self-test (BIST) are nowadays employed. These are known as structured DFT techniques [11] since extra logic is added to the circuit in order to implement test procedure.

In BIST approach, test patterns are generated within the tested circuit. Likewise, test results are evaluated by a test response evaluator internal to the tested circuit. Typically only BIST initiation signals and final test result are communicated via IC input/output pins. This kind of communication does not present potential threat for the system security.

Scan design is a popular DFT technique, which represents an efficient way of accessing internal storage elements. The technique proposed in 1973 [4], has been widely applied in practice since 1977 [5]. The circuit with implemented scan design has two modes of operation: normal functional operation and test mode. In test mode, flip-flops are chained together in one or more shift registers. The test sequence consists of the following steps:

- shift in a test pattern into the circuit storage elements,
- return the circuit for one clock cycle to its normal mode,
- shift out the resulting internal state of the storage elements.

Combinational logic is thus stimulated by the stored test pattern during one clock cycle and the result is

* Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia, anton.biasizzo@ijs.si, franc.novak@ijs.si

DOI: 10.1515/jee-2016-0027, Print (till 2015) ISSN 1335-3632, On-line ISSN 1339-309X

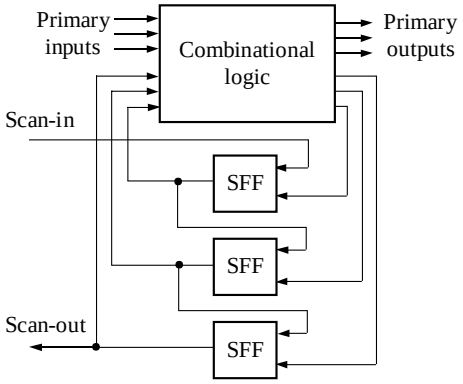

Fig. 1. Scan design

shifted out and evaluated. Test patterns for testing combinational logic can be generated automatically using ATPG (Automatic Test Pattern Generation) tools, which makes the technique very efficient and widely accepted in practice. Scan design is often combined with the test infrastructure of DFT standards IEEE Std. 1149.1, IEEE Std. 1149.4 and IEEE Std. 1500.

3 DFT STANDARDS AND SCAN DESIGN

IEEE Standard 1149.1 originated as a response to the problems of restricted access of individual leads on printed circuit boards by the traditional bed-of-nails approach due to the miniaturization and introduction of surface mounted devices. The need for an alternative access of internal test points gave the idea of building the test probes directly into the chips and to connect the probes with the external ATE (Automated Test Equipment) by simple serial line. The effort of ATE manufacturers and EDA tool suppliers organized as the Joint Test Action Group (JTAG) resulted in a boundary-scan test technique for digital circuits and systems and was approved as the IEEE Std. 1149.1 in 1990.

The principle of the boundary-scan technique is to place a shift register boundary-scan cell adjacent to each component pin and to interconnect the boundary-scan cells in order to form a chain (boundary register) around the border of the chip logic design. During the test mode, boundary-scan cells are used to control the status or read the states of the pins, while during the normal mode the cells are transparent. Components of a board that are fitted with the test structure of the IEEE Std. 1149.1 are interconnected by way of a standard interface termed "test access port" (TAP) with the 4-wire test bus providing serial input data, serial output data, test clock and test mode select line.

Addition of the boundary-scan logic has the following principal tasks: it allows normal circuit operation, it allows data to be shifted in or test results to be shifted out, and it provides a number of circuit tests. The operation of the boundary-scan infrastructure is controlled by the TAP controller which is a finite state machine driven by test clock and test mode select. TAP controller recognises communication protocol and generates internal control signals for the remaining part of boundary-scan logic. The latter consists of instruction register and data registers. Mandatory data registers are boundary register (mentioned above) and a bypass register (which is used to shortcut the boundary register and thus reduce the shift time when testing other components on a board).

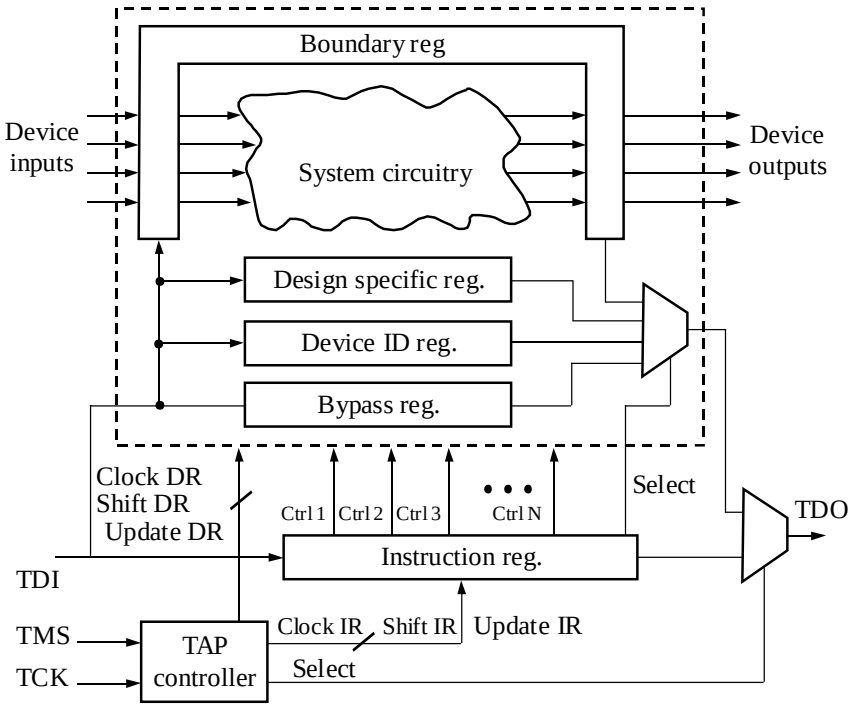

Fig. 2. IEEE Std. 1149.1 architecture

Besides, a number of optional other data registers can be included. Their description is however beyond the scope of this paper. The IEEE Std. 1149.1 architecture is shown in Fig. 2.

IEEE Std. 1149.4 [12] can be regarded as the extension of IEEE Std. 1149.1 to mixed-signal devices. The 1149.4 extensions are analog boundary modules (ABMs) on analog functional pins accessed via internal analog test bus (AB1, AB2). Digital pins have boundary cells as specified in IEEE Std. 1149.1.

Both IEEE Std. 1149.1 and IEEE Std. 1149.4 provide the IC internal test option where the serial boundary-scan chain is used to to scan in test patterns and scan out the test results. In many implementations in practice scan chains of an IC are connected to the serial boundary-scan chain providing access to the internal registers of the circuit-under-test.

IEEE Std. 1500-2005 [13] (Standard Testability Method for Embedded Core-based Integrated Circuits) was created to address test problems of systems implemented on one single die. Modern technology advances allow to integrate functions that have been traditionally implemented on one or more complex printed circuit boards into one single IC, often referred to as system-on-chip (SoC). The development of this new class of ICs is based on the design technique which integrates large reusable blocks (i.e. cores) that have been designed and verified in earlier applications in practice. Embedded cores provide a wide range of functions, like CPUs, DSPs, interfaces, controllers, memories, and others. The design of a complex system-on-chip normally requires expertise in different technology areas which is difficult to find in a single design house. Consequently, embedded-core design involves two parties: core providers and core users. In most cases, the core user (*ie*, system integrator) does not have the knowledge about the design of the building blocks (cores). It is neither the interest of core providers to reveal design and implementation details in order to protect their intellectual property. IEEE Std. 1500 facilitates SoC test considering these restrictions. It provides a standard interface and a set of rules for creation of a wrapper around a core that allows the core to be tested alone by isolating it from its environment. In this way, the core can be tested by the tests supplied by core provider. In addition, the wrapper allows the external logic surrounding the core to be tested independent from the core's state.

Similar to the boundary-scan chain of IEEE Std. 1149.1, the wrapper comprises wrapper cells for each functional input and output port of the core. IEEE Std. 1500 includes instructions which connect scan-chains of the core to the wrapper cells and thus provides the access to the internal registers of the core. In addition, the IEEE Std. 1500 compliant test infrastructure is designed to allow interface compatibility with the common IEEE Std. 1149.1 test access port (TAP) controller. In this way, loading of instructions into embedded core wrappers and scanning in and out test data of a SoC can be performed via IEEE Std. 1149.1 test access port. Beside IEEE Std. 1500, a number of other standards and applications such as IEEE-ISTO 5001^{TM}-2003, IEEE Std. 1532 and IEEE Std. 1149.6 have piggybacked on the IEEE 1149.1 standard instead of defining their own access infrastructure.

In summary, any chip that uses scan design and any system built around it (either in some ad hoc DFT solution or with test or application infrastructure defined by the above standards) provides access to the system's internal logic and may be vulnerable to hackers.

As an illustrative example of vulnerability case study of scan design consider the implementation of DES algorithm with inserted scan chain using Synopsys Test Compiler [14]. Assuming that the attacker knows the DES algorithm (it is public), and assuming that the attacker has access to the high level timing diagrams (provided by the ASIC vendor), the authors show that the attacker needs less than 42000 clock cycles to determine the scan chain structure, recover round key and discover the user key.

4 SECURITY MEASURES

High-quality testing of complex systems requires full access to internal flip-flops and scan design is in many cases the preferred solution with no real alternative. Given that scan design is needed, one has to take into account its vulnerability threats and take suitable countermeasures.

4.1 Adding decoding and encoding logic to scan chains

So far, the countermeasures have been directed mainly at preventing unauthorized access to the system internal logic and stealing intellectual property. At the International Test Conference, Charlotte, 2004, a panel discussion "Security *vs* Test quality: Can we really only have one at a time?" R. Kapur [2] proposed to employ encryption techniques to encrypt sensitive data that is made available to the user in order to perform scan test. In this case, the scan chain logic of the tested unit includes decoding logic at scan-in and decoding logic at scan-out as shown in Fig 3.

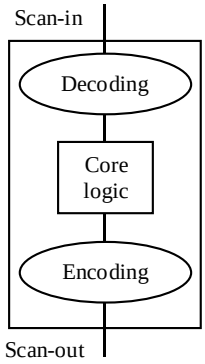

Fig. 3. Decoding logic at scan-in and encoding logic at scan-out

Table 1. FPGA resource utilization

Key length	Slice	LUT	SFF
64	154	228	221
128	284	401	412

The application of cryptographic algorithms in scan design chain is, however, not trivial. The logic implementing a cryptographic algorithm is itself a complex sequential circuit which requires some DFT solution and BIST seems to be the only possible choice in order to avoid the transfer of the internal data sensitive information to an attacker.

Typical cryptographic algorithms are block based. For example, DES is a symmetrical block cipher — an algorithm that takes a fixed-length string of plaintext bits and transforms it through a series of complicated operations into another ciphertext bitstring of the same length. In the case of DES, the block size is 64 bits. Scan chain data decoding/encoding by a DES algorithm implemented in hardware requires a number of 64-bit blocks for the subsequent stages of processing. The resulting logic may represent a non-negligible overhead especially when accompanied by a BIST. In the implementation of DES algorithm reported in [14], 198 flip-flops were used for encoding logic: 64 for the input register, 64 for the output register, 64 for data manipulation and 4 for the controller. The same amount of hardware is needed for decoding.

Another drawback is incompatibility of the length of a scan chain with the size of the block of a cryptographic algorithm, which additionally complicates control logic. Besides, special software must be provided by the ASIC vendor for proper interpretation of the scan test results (*ie*, for fault diagnosis more precise than merely pass/fail test result).

4.2 Adding a locking mechanism to boundary-scan

Theft of intellectual property is, however, not the only vulnerability threat of scan design. Test infrastructure of IEEE Std. 1149.1 is often employed for field reconfiguration, troubleshooting and system maintenance [15]. For example, making a field upgrade to the firmware stored in programmable logic devices can be performed remotely by providing access to the boundary-scan via internet. Likewise, in some implementations of system maintenance, system's boundary-scan is permanently connected to a low-cost test equipment (*ie*, a dedicated PC) for remote diagnostics. All such solutions represent a potential weakness in system's security. An attacker may crack the system and get access to the test port. Executing some pin-permission instruction (*ie*, an instruction which disconnects the component I/O pins from the system logic) during normal system operation may lead to a serious damage. Although intimate knowledge of the boundary-scan infrastructure and the boundary-scan instruction codes of the system is required to brake into the system, worst case scenarios cannot be ruled out in safety critical applications. A recent study of analysed cyber-attacks incident reports from various infrastructure control systems shows a fivefold increase from 1994 to 2004. The type of the incidents is changing from accidental and internal to external. From 2002 to 2004, 66 percent were classified as external, 22 percent were accidental and only 3 percent were internal [16, 17]. The threat of sophisticated web attack on boundary-scan based systems calls for appropriate countermeasures.

Different attack scenarios and defenses for JTAG are studied in [16]. This approach uses a keyed hash, a stream cipher, a message authentication code, and defines challenge/response protocol to prevent the attacks on JTAG. The drawback is the fuse usage for keyed hashes since once the hashes are compromised the device remains exposed. The stream ciphers are also weaker then block ciphers but they are suitable if messages are short and if continuous data stream is required.

The JTAG TAP design that enables the digital rights management is described in [17]. This solution uses hashes and challenge/response protocol to enable the access of the JTAG infrastructures. It can have different hashes for groups of JTAG instructions thus providing a hierarchy of the JTAG access. Like in the previous solution, the hashes are hardwired, which means that a successfully attacked device remains compromised. However, on the JTAG data stream it does not apply encryption, thus it is vulnerable to eavesdropping and man-in-the-middle attacks.

Another approach is to use public/private key pairs in the authentication process. Special care has to be taken for key management and exchange. Furthermore, additional hardware performing asymmetric encryption cores has to be provided. As far as we know, an intensive work in this direction is underway by other groups and their solutions are likely to be reported in the forthcoming publications.

A simple security extension of the JTAG standard was proposed in [9]. The security extension conforms to the IEEE 1149.1 standard and disables all except basic JTAG instructions, unless the proper locking key is loaded. The lock key can be modified via the JTAG interface hence the authorized user can restrict the device access at any time. In order to maintain the lock key over the power-up/power-down cycles the key must be stored in a non-volatile memory.

This solution is very simple and uses little hardware resources as presented in Table 1, however the keys are stored and exchanged in plaintext. This opens the possibility of the eavesdropping on the JTAG bus as well as retrieving the keys from the storage within the device.

A typical non-invasive attack, in which the attacker takes care to remove all tracks of intrusion, consists of the following steps.

1. Evaluation of the Lock register length. (The attacker executes the UNLOCK instruction and feeds values 1 to the input of the boundary-scan chain (TDI). By

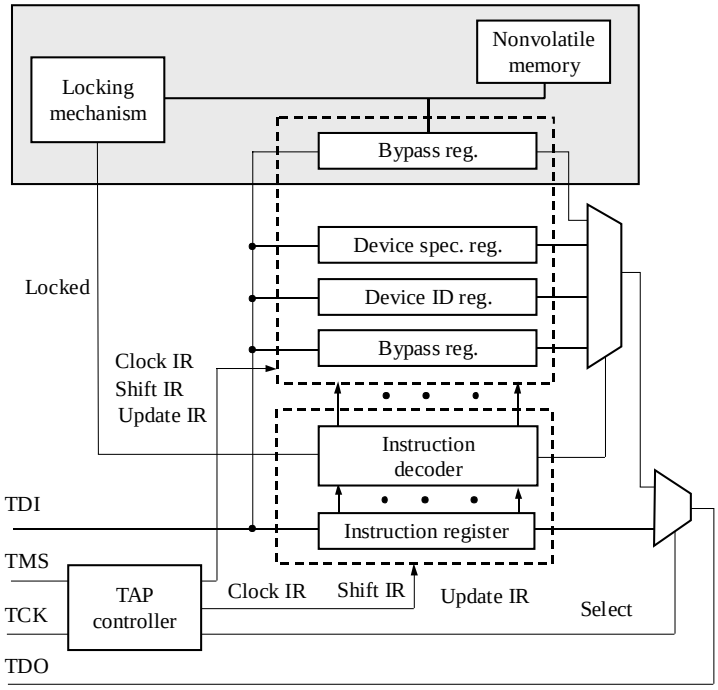

Fig. 4. Security extension for IEEE Std 1149.1

Table 2. The probabilities that the system gets compromised in one hour

M	probability
32	1
36	7.3×10^{-2}
40	4.3×10^{-3}
48	1.5×10^{-5}
56	5.4×10^{-8}
64	2.0×10^{-10}

Table 3. Estimated and exact lower bound values of the Lock register length

probability	$M_{(\mathrm{est})}$	M
1	34	33
10^{-1}	37	36
10^{-2}	40	39
10^{-3}	44	43
10^{-4}	47	46
10^{-5}	50	49
10^{-6}	54	52

counting zeros at the output, the length of the Lock register can be determined.)

2. Repeating of the following steps:

- performing UNLOCK instruction with the guessed value of the lock code,
- performing LOCK instruction and checking the length of the data path:
  - if the length of the data path is 1 then the boundary-scan test logic remains locked and step 2 is repeated with new guess value,
  - if the length of the data path is longer than 1 then the guess value is correct lock code. Use 0 as the new lock code (unlock the boundary-scan test logic) and stop the attack.

After the circuit exploitation with unlocked boundary-scan test logic the test logic can be locked with the original lock code in order to cover the track of the intrusion.

A useful measure of the security strength of the circuit can be given by the probability that the system gets compromised in the given time span, for example, in one hour.

Let us determine the number of unlock codes that the attacker can exploit in a given time interval t. Initially, in order to determine the length of the Lock register, $M + 1$ cycles are required. For each guess of the lock value $M + 2L + 20$ cycles are required, where L denotes the length of the Instruction Register, and M the length of the Lock register, respectively. During one guess two instructions (LOCK and UNLOCK) and the trial lock value (of length M) must be loaded. For this, at least 20 additional TAP state machine transitions are required. The number of exploited unlock codes is

$$N = \frac{t\,f - (M+1)}{M + 2L + 20}$$

where f denotes the frequency of the boundary-scan clock (TCK). The probability that the system gets compromised is given by

$$p = \frac{N}{2^M} = \frac{t\,f - (M+1)}{2^M(M + 2L + 20)}\,.$$

Let us assume that the length of the Instruction Register L is 8 bit and that the boundary-scan clock frequency f is 100 MHz. The probabilities that the system gets

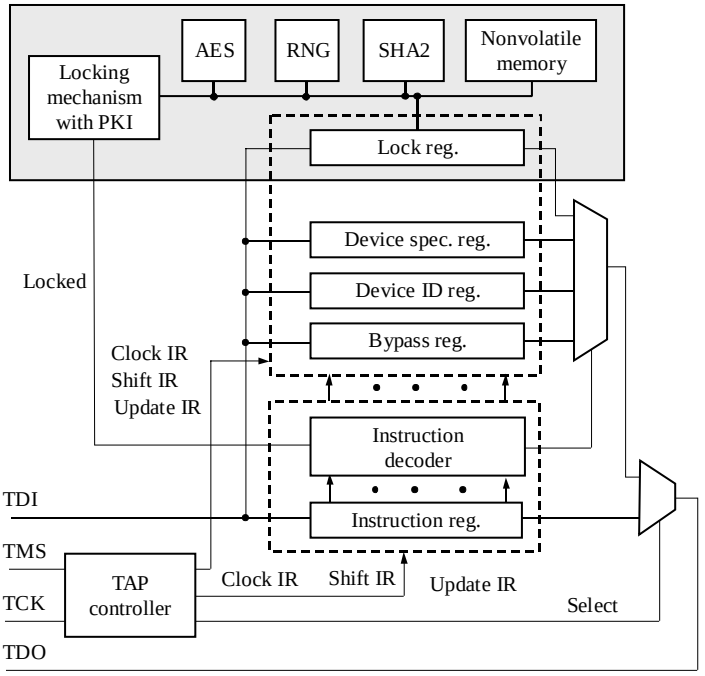

Fig. 5. JTAG locking mechanism using SHA hashes and PKI infrastructure

compromised in the time interval of one hour are given in the Table 2.

From the above equations we can determine the lower bound of the length of the lock register that would assure the required system security for a given time interval t:

$$M = \left\lceil \log_2 \frac{t\,f - M - 1}{p(2L + M + 20)} \right\rceil = \lceil \log_2(t\,f - M - 1) - \log_2 p - \log_2(2L + M + 20) \rceil .$$

This equation cannot be solved analytically yet the impact of $M + 1$ is negligible and can be omitted. In the estimation of the lock register length that assures the required system security the term $(2L + M + 20)$ can be replaced with a smaller value $(2L + 20)$. Estimated lock register length is

$$M_{(\mathrm{est})} = \lceil \log_2(t\,f) - \log_2 p - \log_2(2L + 20) \rceil .$$

In Table 3 the estimations as well as the exact lower bound of the Lock register length are given.

4 IMPROVEMENTS OF THE JTAG LOCKING MECHANISM — WORK IN PROGRESS

The lock key is stored in the memory thus there is potential threat that it can be accessed using appropriate equipment. The approach could be improved by using hashes instead of keys for the authentication. Any hash function that provides sufficient strength, is suitable like MD5, SHA1, and SHA2 hashes. They are commonly used as password hashes in operating systems and as hashes for key exchange by the SSL protocol. They are continuously stressed and their flaws reported. The hardware implementation of hash function requires additional hardware resources.

The authentication scheme can be further refined by using different keys for different JTAG instruction groups. This way a fine grained security system can be established. For instance, a set of benign JTAG instructions could be exposed to any JTAG user, more revealing JTAG instructions like EXTEST could be restricted to a usual JTAG user, while most hazardous instructions could be permitted only to a few users. Some instructions may be even locked for all users but still available for designers and IP-core testers. Manufacturing tests are an example of such restricted functionality and fuses are typically used in order to disable their use after the device is tested in sent out of production. Using the locking mechanism the functionality may stay intact but only the producer has the information to enable it.

Previously described methods are still vulnerable to a replay attacks. In order to prevent such attacks a challenge/response protocol, like [17], can be included in the authentication scheme. In order to establish a trustworthy challenge/response protocol, a source of unpredictable hashes is required for challenge generation. Usually a random number generator is used for challenge generation. To implement such protocol, the JTAG interface has to be extended to retrieve the challenge from the device and to analyze the tester response.

All methods described so far use a plaintext streams to transfer the security keys to the JTAG interface of DUT. While challenge/response protocol may obfuscate some information transferred over the JTAG stream an eavesdropping attacker may still gain important information

about the DUT and may even retrieve the keys. To overcome this drawback encryption of the keys can be used. The encryption of a whole JTAG data stream is problematic, since block cipher does not meet the JTAG standard specification while the stream cipher might not be strong enough. However, the encryption of the authentication information can be performed using block cipher given that the appropriate synchronization with the tester is established.

To maintain secure channel the encryption keys has to be exchanged in a secure way to prevent man in the middle attack. This can be achieved by using simplified public key infrastructure (PKI), which has private/public keys for both tester and JTAG device. While such a protocol provides the best protection and flexibility of the JTAG infrastructure, its hardware implementation requires a substantial amount of hardware resources. An architecture using PKI with SHA2 hash function used for authentication keys, random number generator used in challenge/response protocol, and AES stream encryption for data stream is depicted in Fig. 5.

Like in previous authentication schemes different keys can be used to achieve different authorization levels. In such implementation a key hash for each authentication group has to be kept in nonvolatile memory. The proposed locking mechanism also allows the change of authorization keys.

5 CONCLUSION

Scan design chains can potentially be used to break in a system and steal intellectual property. Even worse, critical infrastructure systems such as power plants, chemical plants, pipelines, dams, *etc* with process control systems incorporating boundary scan with TAP connected to internet in order to upload firmware upgrades or perform remote system maintenance are vulnerable: an attacker familiar with the IEEE Std. 1149.1 can brake into a system and disturb its normal operation by executing an invasive test sequence which may lead to a catastrophic event. Consequently, the risk of system brake-in should be seriously considered and appropriate countermeasures taken. In this paper we have analysed currently proposed solutions. We have pointed out to some problems of adding decoding and encoding logic to scan chains: besides non-negligible logic overhead, this solution requires special software provided by the ASIC vendor for proper interpretation of the scan test results. The security extension of the JTAG standard aims to prevent unauthorised users to break in a system and disturb its normal operation via IEEE 1149.1 test port has been briefly reviewed. For this solution, a possible attack scenario is described and probabilities that the system gets compromised in one hour are calculated, together with the lower bound values of the Lock register length that assure the required system security. In the frame of HORIZON 2020 action ECSEL-RIA: "Cyber Physical System based Proactive Collaborative Maintenance – MANTIS" we are currently exploring modifications of the locking mechanism including different PKI architectures, which would allow different authorization levels, and possibly also the change of authorization keys.

Acknowledgments

This paper reports the work performed within the activities of HORIZON 2020 action ECSEL-RIA: "Cyber Physical System based Proactive Collaborative Maintenance – MANTIS"

References

[1] MARINISSEN, E. J. (moderator): Security *vs* Test Quality: Can We Really Only Have One at a Time?, Proc. of the ITC, Charlotte, 2004, pp. 1411.

[2] KAPUR, R.: Security *vs* Test Quality: Are they mutually exclusive?, Proc. of the ITC, Charlotte, 2004, pp. 1414.

[3] GOERING, R.: EE Times On Line, Latest News, http://www.us.design-reuse.com/news/news8974.html.

[4] WILLIAMS, M. J. Y.—ANGEL, J. B.: Enhancing Testability of Large Scale Integrated Circuits via Test Points and Additional Logic, IEEE Trans. Comput. **C-22** No. 1 (1973), 46–60.

[5] EICHELBERGER, E. B.—WILLIAMS, T. W.: A Logic Design Structure for LSI Testability, Proc. 14^{th} Des. Autom. Conf., New Orleans, 1977, pp. 462–468.

[6] IEEE Standard Test Access Port and Boundary-Scan Architecture. IEEE Std1149.1-2001, Institute of Electrical and Electronics Engineers, 14-Jun-2001.

[7] BLEEKER, H.—VAN DEN EIJUDEN, P.—DE JONG, F.: Boundary-Scan Test, A Practical Approach, Kluwer Acad. Publ, 1993.

[8] PARKER, K. P.: The Boundary-Scan Handbook, Third edition, Kluwer Acad. Publ., 2003.

[9] NOVAK, F.—BIASIZZO, A.: Security Extension for IEEE Std 1149.1, Journal of Electronic Testing, Theory and Practice **22** No. 3 (June 2006), 301–303.

[9] EICHELBERGER, E. B.—LINDBLOOM, E.—WAICUKAUSKI, J. A.—WILLIAMS, T. W.: Structured Logic Testing, Prentice-Hall, 1991.

[10] IEEE Standard for a Mixed-Signal Test Bus. IEEE Std 1149.4-1999. Institute of Electrical and Electronics Engineers, 2000.

[11] IEEE Std 1500-2005. IEEE Standard Testability Method for Embedded Core-based Integrated Circuits. Institute of Electrical and Electronics Engineers, 2005.

[12] YANG, B.—WU, K.—KARRI, R.: Scan Based Side Channel Attack on Dedicated Hardware Implementations of Data Encryption Standard, Proc. of the ITC, Charlotte, 2004, pp. 339-344.

[13] BONNETT, D.: Boundary Scan Goes Underground, Test & Measurement World (Sep 2005), 49–56.

[14] MILLER, A.: Trends in Process Control System Security, IEEE Security & Privacy **3** No. 5 (2005), 57–60.

[15] US Computer Emergency Readiness Team, Control Systems Cyber Security Awareness, http://www.us-cert.gov/reading_room/Control_System_Security.pdf.

[16] ROSENFELD, K.—KARRI, R.: Attacks and Defenses for JTAG, IEEE Design and Test of Computers **27** No. 1 (2010), 36–47.

[17] CLARK, C. J.: Anti-Tamper JTAG TAP Design Enables DRM to JTAG Registers and P1687 On-Chip Instruments, Proc. HOST 2010, Anaheim, CA, USA, June 2010, pp. 19–24.

Received 2 July 2015