Review of Proposed JTAG Security Methods

ECE 548: Cyber System Security

By: Payton Murdoch, V00904677

Date: April 5th, 2024

Table of Contents

Section I: Abstract

With the implementation of JTAG to allow for embedded testing on the chips themselves, attackers were able to violate the integrity of the hardware root of trust. An exposed JTAG can enable debugging commands to run and leak information to the attacker. As such, what kind of security measures have been implemented on compliant devices to help re-establish the root of trust? Luckily, JTAG is not a new system; thus, many articles and proposals describe different processes that could be implemented. Therefore, we will review a subset to establish a generalized understanding of methods and consider which implements the highest security. The main methods described are authentication and encryption procedures, anomaly detection, and access restriction. With each of the processes mentioned above examined, based on general understanding and the articles, we stated that encryption processes provide the most thorough security. This is given its general susceptibility, provided that the process is implemented most securely.

Section II: Introduction

Historically, the hardware of a cyber system was denoted as the most secure aspect of computers. This established a hardware root of trust as the most common exploitation methods of computing systems came from software attacking Operating Systems and applications. Over time, as the hardware supply chain became more and more complex, outsourcing the construction of integrated circuitry has allowed hardware to become more and more susceptible to exploitation methods from adversaries along the chain. From the fabless design house to the validation

procedures, at each step, an attacker could implement trojans, counterfeit ICs and IPs, falsify validation results and more to undermine IC producers and the functionality of their products.

To assist IC construction, groups have implemented procedures for universal standardization practices to ensure that circuitry can be adequately validated to ensure this formerly established root of trust. We have IEEE 1149.1, also called JTAG, among the standardization mentioned above methods. While the scanning system implemented following this standard is an excellent solution for testing and validating IC functionality with an embedded system, there was a massive security vulnerability upon its inception. JTAG Test Access Ports initially lacked any security. Thus, on the IC, any party has direct access to the Input and Outputs for test data. This has proven to be a source for numerous nefarious operations that can violate the integrity of the ICs and their respective systems.

With this in mind, researchers and companies have invested time in developing methods for securing JTAG Test Access Ports. Methodologies such as the encryption of test data, the implementation of authentication mechanisms, the elimination of the testing ports after production, and even the implementation of time constraints for testing have been developed over time. The strategies mentioned earlier all aim to limit the potential attack vectors on ICs and PCBs. However, when considering JTAG security methodologies, one must question which proposed and implemented methodologies are the most effective in protecting the system's integrity.

Section III: Methodology

To examine the protocols proposed to assess the most effective security for the JTAG, we must review a series of articles that provide multiple applications of the method to clarify our understanding of how it works and its susceptibility. Given that JTAG has been established as a standard for many years, there is a vast amount of reliable research dating back to the concept's inception. While researching and evaluating the articles, we decided to change the paper’s topic. Attempting to conduct a thorough analysis of each of the methodologies led to complications, as each paper would require many paragraphs discussing how the papers formatted the testing parameters and evaluating the results. Therefore, we shifted the focus towards briefly reviewing the topics, parsing specific examples, and noting some of their susceptibilities and benefits.

Section IV: Background

It is possible to proceed with this report by first describing the JTAG/IEEE 1149.1 standard, as not all readers will have a preconceived notion or understanding of it. Furthermore, once we have developed our knowledge of JTAG, we must note the vulnerabilities established along the supply chain that target this feature and the potential threat it poses to IC security. With this in mind, let us proceed to discuss JTAG. As a brief overview, with the advancements in IC and PCB complexity, developers encountered problems concerning the traditional In-Circuit testing methods for chips. These standard testing methods require validators to have access to an extensive series of physical pins to measure the electrical signals displayed through the inputs and outputs to determine if the system has the appropriate functionality. However, ICs could become more compacted with new packaging methods, thus requiring a series of physical

connections, which became a significant hindrance. With this need for more efficient testing methods, a group known as the Joint Task Action Group brought forth the Boundary Scan Cell system dubbed standard IEEE 1149.1, which the same name, JTAG, can denote. [1]

JTAG-embedded devices have a peripheral system encapsulating the IC cores; these systems consist of various scanning cells, which can be utilized to test the functionality of the core's logic at multiple steps following the instructions. While this system aims to limit the requirement for physical access points for testing, pins still need to be probed to test functionality. This is determined at the Test Access Ports. At the TAP, the device has pins for Test Data Inputs and Test Data Outputs, where data is fed through and evaluated based on the active testing mode. The testing mode is implemented through the Test Mode Select and Test Clock pins, which manipulate the instruction register that controls the logic of the boundary scanning cells, determining which function is evaluated in the systems. With JTAG implemented on a printed circuit board, the system can test the functionality of the system configurations, memory, interconnectedness, infrastructure, and more. These scan cells and boundary scan systems are joined into a series of additional boundary scan systems, which in turn is denoted as a scan chain where the Test Data Outputs can be connected to the Test Data Inputs of the subsequent system. Being a part of this scan chain allows for specific circuitry logic along the chain to be tested. In contrast, others get skipped through the bypass register so that the final outputs represent the exact scan-cell data being tested. Since this system was developed in the 1990s, this versatile system has been expanded and utilized in multiple standards as an effective method for embedded system testing. Therefore, new IC and PCB designs that hope to utilize this system's vulnerabilities must be considered.[1][2]

Moving on from JTAG functionality, we must discuss some attack vectors for this system. Without additional security on top of these embedded systems, the TAP is wholly exposed to malicious attackers along the supply chain so that numerous hardware vulnerabilities can be exploited. Given that the JTAG functionality allows for testing and verification of the inner logic of the chip, attackers and untrusted sources can utilize it to reverse engineer the chip and its functionality, thus understanding the inner workings of the devices and allowing for replication or other nefarious methods.[2] While considering other vulnerabilities, we can discuss general side-channel attacks that can be conducted throughout the JTAG scan chain. If attackers could access the JTAG ports using malicious operations and commands, they would be able to cause the scan chain to dump out pertinent information. Therefore, an attacker can extract even the most confidential information, such as cryptographic secret key information embedded within the circuitry, thus violating the system's integrity.[3] In another sense, it is denoted that JTAG has additional hidden procedures for general debugging. However, once these hidden commands were discovered, attackers could exploit their functionality and allow for the insertion of trojans into systems. In a general sense, these commands have allowed for the insertion of code to destroy the system, extract memory information, or even install a backdoor for malicious users to have unfettered access to the computer system.[4] Furthermore, with this debugging functionality, attackers have found the system susceptible to fault injection attacks. Test data injected with a specific instruction set can allow malicious code to run on the machine. This code can even conduct privilege escalation attacks, violating system integrity with the abovementioned backdoors.[5] It is challenging to state every possible exploitation method to

which an unsecured JTAG is vulnerable. However, this is a foundation for emphasizing additional JTAG security.

Section V: Body

JTAG security methods or countermeasures can generally be categorized into four groups. First, we have authentication methods encapsulating passwords, challenge-response pair procedures, and more. Next, we have encryption procedures depicting test data encryption to limit the possibility of man-in-the-middle or side-channel attacks. Additionally, we have the most straightforward method, which involves physically modifying the JTAG for access restriction. Furthermore, we have anomaly detection, a dynamic group that triggers one of the abovementioned methods based on detecting abnormal JTAG usage.[6][7] The following paragraphs will explore the general methods using literature that propose specific countermeasure cases based on the articles [6] and [7]. We will explore the countermeasure concept as a whole, and by evaluating the proposed solutions, we hope to analyze the advantages and disadvantages of the procedure. Unfortunately, this review can not cover every sample case of every proposed solution, and thus, it is subject to generalizations based on the few cases reviewed.

Section VI: Authentication

First, let us discuss authentication methods. Concerning password-based authentication, the proposed method in the articles [8] and [9] discusses adding unlock and locking instruction sets

to the TAP. This is so that users without authorized access will have the JTAG and the systems derived from it, like IEEE 1500, locked in the bypass state so they do not interfere with system operations. Of course, this depends upon some confidential key or information shared between the system and the authorized entity. The system can manipulate a code when entered and compare it to the stored password to validate it. In [8], while the specific generation of the key code is not mentioned, the proposed method was implemented and tested on Xilinx Spartan3 3s200ft256-4, a subsidiary of AMD and thus a valid security measure [10]. In another sense, the patent [11] represents password-based authentication by stating that a debugging entity with authorization to JTAG will probe the IC with the secret key input, which is then compared within the hardware key stored in long-term storage for validation.

Moving on to another implementation, [9] discusses IEEE 1500, which introduces a wrapper around the initial JTAG scan system for more complex testing purposes. The authentication method proposed within the document adds security embedded within the wrapper, emphasizing the same lock and unlocking modes that restrict functionality. This scheme introduces the concept of a golden key, a 256-bit key generated by the chip using a linear feedback shift register that is only known by the company that implements the chip's testing procedures. As an extension of the aforementioned golden key, in the paper [6], the authors propose another key-generation method dependent on the Physically Unclonable Functions, thus allowing for more obfuscation of the password creation process.

With password authentication methods, the system's security depends on the algorithm's and password's complexity. For example, the 256-bit key has 2^{256} possible values and is thus

unfeasible to determine exhaustively. However, for the debugging entity to gain access to the system, there must be some key exchange between them. Attackers can easily circumvent all security precautions if the key is improperly encrypted or leaked. Additionally, this method is not immune to side-channel attacks. If an adversary has unfettered access to the system, they can read electrical signals to determine the secret key bits in use.[7]

When considering the authentication of JTAG based on the exchange of information between the system and the debugging entity, we can take into consideration the Challenge-Response protocols; similar to how PUF-based authentication works, the trade is conducted between three entities, the JTAG with the authentication protocol, the system and the certified authority which has the credentials. The article [12] covers a variation of CRP authentication where there is an exchange from the JTAG to the Server using the host as an intermediary. This exchange contains the Device ID, user information, and Challenge to a certified authority, which returns the Challenge-Response Pair and user verification information so that the secure JTAG can verify the data and begin its operations.

With PUF-based authentication in mind, the paper [13] discusses implementing a secure JTAG wrapper for IEEE 1500 devices utilizing the CRP protocol for PUF verification, whose security depends entirely on the PUF in operation and its unpredictability. Of course, this document also states that the manufacturer of the PUF is assumed to be trusted. Instead of PUFs, the paper [14] proposes utilizing CRP with cryptographic hashing and a challenge composed of a true random number generator based on ring oscillators. This Challenge-Response method limits the requirements for the debugging entity to know the information or secret for authentication, thus

reducing the risk of an unauthorized user gaining credentials. Instead, it involves the exchanges between the Certified Authority or manufacturer and the system itself. Of course, this authentication method has additional complexity compared to simple passwords.

However, there are more opportunities for man-in-the-middle as data transmission between the Certified Authority and the system can be intercepted. Of course, this is determined by encrypting the data in transit. Furthermore, cited in [12], which utilizes simple credential CRP protocols and thus, by extension, can be applied to the other CRP methods, is immune to Replay Attacks, Brute Force Attacks, Cloning attacks, Guessing Attacks, Sniffing Attacks and Spoofing attacks. Therefore, even if a man-in-the-middle can access the information, there is a low probability of the attacker reproducing the exact states to violate the device's integrity.

Section VII: Encryption

Moving on from authentication methods, we must consider the ulterior security procedures for JTAG. This leads us to discuss the encryption of test data along the scan chain so that eavesdropping adversaries cannot gather information from the outputs. The information gathered can allow malicious users to reverse engineer the topology and functionality of the hardware, thus allowing them to conduct other attacks and target specific portions of the system. Of course, to successfully implement these procedures, as noted in [15], the secret key must be stored off the scan chain so that adversaries cannot get any information about it. Of course, common encryption types can be employed for these processes, such as block ciphers like RSA or AES, which are computationally NP-hard to solve without knowledge of the secret keys or the plaintexts.

AES is a symmetric encryption algorithm that splits the data bits into blocks. The difference is computed with the blocks compared to a round key derived from a master key based on a scheduling algorithm using the exclusive-or operator. The blocks are then put into a substitution scheme to change the block values, and finally, they are put through a method that changes the positions of each bit to add diffusion. These methods are repeated over multiple rounds utilizing the same substitution network and position changes but with new round keys at each iteration.[16] Alternatively, RSA is an asymmetric cryptosystem that depends on a publicly known encryption key and a specially computed private key. The public encryption key can encrypt the data. However, only the private key can decrypt it. Thus, users with the public key can only encrypt the data and nothing else.[17]

Alternatively, in the document [18], the concept of stream ciphers is proposed, such that the keys are generated using random initial variables and random number generators. The article proposes utilizing 80-bit length stream keys and initializing vectors so that it is hard to distinguish from a random stream of bits and shift with every input so that they cannot be susceptible to attacks given that the length causes the system to be unfeasible to decrypt without knowledge of the key or initializing vectors. Encryption security is entirely dependent on the secure storage of the keys. Therefore, side-channel attacks listening to processes dependent on secret keys could circumvent the encryption process. However, if the information remains confidential, all the methods above have been secure. However, block ciphers provide weaker encryption than stream ciphers, thus making it easier for attackers to gain knowledge from ciphertext similarities.[7]

Section VIII: Access Restriction

Alternatively, the option is to modify the circuitry to eliminate the vulnerabilities. As depicted in [19], a developer can take a few routes for these modifications. One could change the voltage required to run JTAG so that an attacker without proper knowledge would damage the system. One could obfuscate the pins so they do not map to common locations or remove them entirely. Regarding functionality elimination, other options are to hardwire the JTAG port into the reset state or turn off programmable aspects of the system so that no debugging process can occur after initial verification. Unfortunately, when it comes to these physical modifications, as denoted in [20], permanently disabling JTAG is not the most feasible choice as it is required in debugging further tests and validation.

Section IX: Anomaly Detection

The final method to discuss is anomaly detection, which supervises the JTAG functionality and determines whether a malicious user attacks the system based on predefined rulesets or machine learning methods.[7] Predefined rulesets can be like that proposed in [21], which adds a filter to the JTAG at the interface such that only a predefined set of sequences can be allowed, thus restricting access. Sequences which violate the filter are locked out of the system for a designated time frame. As described in [2], manufacturers could implement classifier-based machine learning models, training the model on predefined operations utilized for routine testing or debugging and thus allowing for the detection of abnormal activity that does not fit into nominal testing methods and is therefore labelled as an attack. The paper above examines using the Random Forest and SVM classification models, and while these machine learning methods

can be effective, they are only trained on previously defined attack vectors. Whenever a new attack vector is discovered, the system must be retrained to be used effectively. Of course, like the ruleset detection, anomalous activity will yield detections, and thus, the user will be locked out of the JTAG usage.[7]

Section X: Discussion

Given that the JTAG standard has been implemented in devices for many decades, there has been immense research and development concerning methods for securing the system. After evaluating the methods cited in the prior sections using the papers mentioned, this section discusses which general techniques are the most beneficial to implement on a system. Of course, this is subject to bias as we are not experts in the field and not electrical engineers; thus, we require a more comprehensive understanding of certain restricting factors before commenting on them. This is mainly pertinent to articles where additional logic and physical mechanisms are necessary to perform the designated tasks, thus increasing the size of the embedded systems, which may be unfeasible in some cases. Therefore, we will solely evaluate based on the general understanding of the methods, their complexity and susceptibility to attacks.

We will first eliminate the access restriction methods as viable options. Removing or destroying the JTAG functionality can effectively protect the system. However, this eliminates additional testing and debugging functionalities, which can render the system entirely useless if hardware faults occur, as there would be no way to determine where it happened or how to fix it. Secondly, authentication methods have their denoted benefits, adding complexity to the JTAG debugging

protocols and more. However, based on the method implemented, attackers can circumvent such an attack in multiple ways, whether side-channel attacks to listen for critical bits or man-in-the-middle attacks intercepting the information communicated between the system and the authority. Therefore, it will be removed from consideration. Finally, we want to eliminate anomaly detection as the most secure method. While this method is the most dynamic as it depends on rulesets or trained systems, it unfortunately only protects against known attacks initially and requires expensive hands-on training whenever new attacks have been detected; thus, zero-day vulnerabilities can completely circumvent any security.[2][6][7]

With all this in mind, we denote encryption of JTAG data as the most secure method, even though it is entirely dependent on the safe storage of the key. Suppose this key-storing mechanism is altogether hidden and enforces adequate encryption procedures. In that case, there is a low probability of an attacker discerning the data as it would appear to be an entirely random sequence of bits. Of course, this is not immune to side-channel attacks, provided the attacker located the mechanisms responsible for key storage and distribution. Additionally, if there is poor implementation of the cryptographic processes, in that case, mathematical approaches to breaking the encryptions exist, such as linear or differential cryptanalysis for AES, twin primes attack against RSA or two-time pad attack against stream ciphers.[7][15][18]

Section XI: Conclusion

This document establishes that JTAG and other System Chip testing methods were required due to the increasing complexity of modern chips as limited space was devoted to external pins,

which were utilized in traditional tests. While its initial development was revolutionary, attackers found that its ability to input and output test data allowed for the leakage of system architecture, memory and more. Therefore, there was a requirement to produce methods that secured the JTAG ports and the accompanying standards that built upon this system. The most generalized methods are authentication, encryption, access restriction, and anomaly detection protocols. From a pure observation view, encryption methods provide the most security as a standalone protection process. However, this paper has yet to consider applications that can implement numerous methodologies to offer higher security.

Section XII: References

[1] Be Van Ngo, P. Law and A. Sparks, "Use of JTAG boundary-scan for testing electronic circuit boards and systems," 2008 IEEE AUTOTESTCON, Salt Lake City, UT, USA, 2008, pp. 17-22, doi: 10.1109/AUTEST.2008.4662576.

[2] X. Ren, F. P. Torres, R. D. Blanton and V. G. Tavares, "IC Protection Against JTAG-Based Attacks," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 1, pp. 149-162, Jan. 2019, doi: 10.1109/TCAD.2018.2802866.

[3] Bo Yang, Kaijie Wu and Ramesh Karri, "Scan based side channel attack on dedicated hardware implementations of Data Encryption Standard," 2004 International Conference on Test, Charlotte, NC, USA, 2004, pp. 339-344, doi: 10.1109/TEST.2004.1386969.

[4] S. Skorobogatov and C. Woods, “Breakthrough silicon scanning discovers backdoor in military chip,” Cryptographic Hardware and Embedded Systems – CHES 2012, pp. 23–40, 2012. doi:10.1007/978-3-642-33027-8_2

[5] F. Majéric, B. Gonzalvo and L. Bossuet, "JTAG Fault Injection Attack," in IEEE Embedded Systems Letters, vol. 10, no. 3, pp. 65-68, Sept. 2018, doi: 10.1109/LES.2017.2771206.

[6] K. -J. Lee, Z. -Y. Lu and S. -C. Yeh, "A Secure JTAG Wrapper for SoC Testing and Debugging," in IEEE Access, vol. 10, pp. 37603-37612, 2022, doi: 10.1109/ACCESS.2022.3164712.

[7] E. Valea, M. Da Silva, G. Di Natale, M. -L. Flottes and B. Rouzeyre, "A Survey on Security Threats and Countermeasures in IEEE Test Standards," in IEEE Design & Test, vol. 36, no. 3, pp. 95-116, June 2019, doi: 10.1109/MDAT.2019.2899064.

[8] F. Novak and A. Biasizzo, “Security extension for IEEE Std 1149.1,” Journal of Electronic Testing, vol. 22, no. 3, pp. 301–303, Jun. 2006. doi:10.1007/s10836-006-7720-x

[9] G. -M. Chiu and J. C. -M. Li, "A Secure Test Wrapper Design Against Internal and Boundary Scan Attacks for Embedded Cores," in IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 20, no. 1, pp. 126-134, Jan. 2012, doi: 10.1109/TVLSI.2010.2089071.

[10] “Spartan 3 FPGA family,” AMD, https://www.xilinx.com/products/silicon-devices/fpga/spartan-3.html (accessed Apr. 1, 2024).

[11] W. C. Moyer and M. D. Fitzsimmons, “INTEGRATED CIRCUIT SECURITY AND METHOD THEREFOR,” Sep. 4, 2007.

[12] K. Park, S. G. Yoo, T. Kim, and J. Kim, “JTAG security system based on credentials,” Journal of Electronic Testing, vol. 26, no. 5, pp. 549–557, Sep. 2010. doi:10.1007/s10836-010-5170-y

[13] A. Das, Ü. Kocabaş, A. -R. Sadeghi and I. Verbauwhede, "PUF-based secure test wrapper design for cryptographic SoC testing," 2012 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2012, pp. 866-869, doi: 10.1109/DATE.2012.6176618.

[14] C. Clark, "Anti-tamper JTAG TAP design enables DRM to JTAG registers and P1687 on-chip instruments," 2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), Anaheim, CA, USA, 2010, pp. 19-24, doi: 10.1109/HST.2010.5513119.

[15] M. Da Silva, M. -L. Flottes, G. Di Natale and B. Rouzeyre, "Preventing Scan Attacks on Secure Circuits Through Scan Chain Encryption," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 38, no. 3, pp. 538-550, March 2019, doi: 10.1109/TCAD.2018.2818722.

[16] V. Saicheur and K. Piromsopa, "An implementation of AES-128 and AES-512 on Apple mobile processor," 2017 14th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON), Phuket, Thailand, 2017, pp. 389-392, doi: 10.1109/ECTICon.2017.8096255.

[17] S. A. Nagar and S. Alshamma, "High speed implementation of RSA algorithm with modified keys exchange," 2012 6th International Conference on Sciences of Electronics, Technologies of Information and Telecommunications (SETIT), Sousse, Tunisia, 2012, pp. 639-642, doi: 10.1109/SETIT.2012.6481987.

[18] M. Da Silva, E. Valea, M. -L. Flottes, S. Dupuis, G. Di Natale and B. Rouzeyre, "A New Secure Stream Cipher for Scan Chain Encryption," 2018 IEEE 3rd International Verification and Security Workshop (IVSW), Costa Brava, Spain, 2018, pp. 68-73, doi: 10.1109/IVSW.2018.8494852.

[19] K. Lee, Y. Lee, H. Lee and K. Yim, "A Brief Review on JTAG Security," 2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), Fukuoka, Japan, 2016, pp. 486-490, doi: 10.1109/IMIS.2016.102.

[20] A. Sguigna, "Mitigating JTAG as an Attack Surface," 2019 IEEE AUTOTESTCON, National Harbor, MD, USA, 2019, pp. 1-7, doi: 10.1109/AUTOTESTCON43700.2019.8961076.

[21] R. Baranowski, M. A. Kochte, and H.-J. Wunderlich, “Access Port Protection for reconfigurable Scan Networks,” Journal of Electronic Testing, vol. 30, no. 6, pp. 711–723, Oct. 2014. doi:10.1007/s10836-014-5484-2.