American Journal of Industrial and Production Engineering
1(1):1-6, August 2019.

HARDWARE ATTACK MITIGATION TECHNIQUES ANALYSIS

Samer Moein and T. Aaron Gulliver

ABSTRACT

The goal of a hardware attack is to physically access a digital system to obtain secret information or modify the system behaviorl. These attacks can be classified as covert or overt based on the awareness of the attack. Each hardware attack has capabilities as well as objectives. Some employ hardware trojans, which are inserted during, manufacture, while others monitor system emissions. Once a hardware attack has been identified, mitigation techniques should be employed to protect the system. There are now a wide variety of techniques, which can be used against hardware attacks. In this paper, a comprehensive survey of hardware attack mitigation techniques is presented. These techniques are matched to the hardware attacks and attack criteria they can counter, which helps security personnel choose appropriate mitigation techniques to protect their systems against hardware attacks. An example is presented to illustrate the choice of appropriate countermeasures.

KEYWORDS Hardware Attack, Hardware Attack Mitigation, Hardware Security, Covert Attack & Overt Attack

1.INTRODUCTION

The use of semiconductor devices in military, financial, economic, and other critical infrastructure has raised significant concerns regarding hardware security. A victim is unaware of the occurrence of a covert attack but may have knowledge of an overt attack. Overt hardware attacks [1] such as deprocessing and reverse engineering are employed to reveal device functionality in order to steal information and copy devices. Further, some overt attacks introduce hardware trojans [2] by modifying Integrated Circuits (ICs) to create abnormal system behaviour, while others monitor system emissions to obtain information. The increasing sophistication of hardware attacks as well as the growing chip complexity makes hardware security a major challenge for the semiconductor industry [3–6]. The design and manufacture of an IC involves multiple processes. These provide numerous opportunities for attacks, and mitigation techniques must be developed to counter them. Figure. 1 shows the approaches to both attacking and defending a chip. Overt attacks, i.e. reverse engineering, deprocessing, and microprobing, allow an attacker to examine the internal structure of a chip. This information can be used to identify chip vulnerabilities for covert at-tacks, i.e. power, timing, and electromagnetic, or to copy the chip. An attacker can also insert a hardware trojan into a chip to allow an attack to be initiated. A defender can use destructive techniques to check if malicious modifications have been made to a chip. However, this approach is time consuming and requires significant resources, so it is not practical to examine a large number of chips. In practice, defenders rely on non-destructive techniques to determine if a chip is working probably [7–9]. These techniques can be employed during testing and/or chip operation. Hardware attack mitigation techniques are used to protect a chip during both chip design and operation. These techniques can be used to produce a secure chip when it is being designed. Further, if unexpected behaviour is detected during chip operation, they can be employed to counter any attacks. The knowledge, skill and resources that modern attackers possess enable them to introduce modifications into the design during the IC life cycle. Many of these modifications are not detected during the testing and deployment phases [4, 10, 11]. Developing mitigation techniques against these malicious attacks begins with their identification and classification. Hardware attacks can be classified as covert or overt [1, 2]. They can also be classified based on the accessibility, resources, and time required for implementation [50]. The classification can be used to determine the system requirements to defend against attacks [12, 13].

Hardware attack mitigation techniques can be divided into two categories, those designed to counter multiple attacks and those developed for single attacks. A number of approaches have been used to counter multiple attacks. Hiding techniques are based on reducing the signal strength or increasing the noise level [53]. Masking techniques make it difficult for an attacker to determine the relationship between chip emissions and the corresponding data or operations [16– 19]. Random noise can be employed to decrease the Signal-to-Noise Ratio (SNR) of IC signals [54], and make emissions more independent of the chip operations [53, 55]. Chip emissions can also be masked using asynchronous logic gates [56, 57], or reduced by using low power design techniques [53]. Further, emissions between chip regions can be lowered via design partitioning [20, 21]. Restricting chip access using anti-tampering techniques can prevent an attacker from collecting chip data [22, 23]. Moreover, emission filtering can be used to reduce data leakage [67]. These techniques can be used to counter most covert attacks, as these attacks typically monitor chip emotions. Sensors can also be deployed around a chip to detect anomalies and counter overt attacks [35]. Numerous countermeasures have been proposed for specific hardware attacks. Algorithmic resistance, restricting physical access, randomized computation time [33], and duplicate encryption [26] have been used to counter fault attacks. Time/branch equalization, random delays, and constant time hardware [27] have been used to counter timing attacks. Keyed hash functions, message authentication codes, public key infrastructure, and stream ciphers have been employed to increase the security of JTAG devices and encryption circuitry [29, 30].

Shielding has been used to counter acoustic attacks [15]. Cycling memory with random data can be used to mitigate data remanence attacks [25]. Cache partitioning has been shown to pre-vent information leakage [44], and sensitive cache lines can be placed in a secure partition [45] to counter cache attacks. Further, a non-deterministic processor can be used to run instructions in random order [47].

2. HARDWARE ATTACKS

Hardware attacks aim at physically accessing a system to obtain stored information, determine the internal structure of the hardware, or inject a fault. Several approaches have been pro-posed to classify hardware attacks based on security levels [69–71], algebraic properties [68], accessibility [66], and resources [1, 2]. In order to evaluate security, tamper protection levels were introduced by IBM [69]. Their classification has six security levels from zero corresponding to a system without any security protection to high for a virtually unbreakable system. U.S. and Canadian federal government agencies are required to use cryptographic products that have been validated using Federal Information Processing Standards (FIPS) [70] or Common Criteria (CC) [71]. Most CC protection profiles rely on FIPS validation for cryptographic security. FIPS 140-2 or 140-1 validations have four security levels from level 1 which indicates basic security requirements for a cryptographic module to level 4, which indicates physical security, i.e. an envelope of protection around the cryptographic module to detect device penetration. This classification focuses on cryptographic applications and/or devices.

A flexible methodology was proposed in [68] to categorize hardware attacks based on their properties. Weights can be assigned based on attack criteria so that detailed comparisons can be made, and as technology changes these weights can be adjusted according to attack and/or defence capabilities. A defender can use this methodology to determine the possible approaches an attacker may use to launch an attack. Variations of the same attack can also be considered. For example, two Deprocessing (DEP) attacks DEP-1 and DEP-2 were considered in [68] where DEP-1 assumes that the attacker uses in-house resources, while DEP-2 assumes the attacker employs outsourcing and so requires fewer resources. In general, the classification of a hardware attack is based on the capabilities and techniques used by the attacker and de-fender. This information can be used by security designers to identify system vulnerabilities and develop countermeasures. A classification based on attack accessibility was proposed in [66]. This classification divides attacks into three groups: non-invasive, invasive, and semi-invasive. Non-invasive attacks do not require any initial preparation or direct connection to the device. Invasive attacks require direct access to the internal components of the device. Semi-invasive attacks introduced in [73] lie in the gap between non-invasive and invasive attacks. These attacks require a moderate level

of accessibility to gain access to the chip surface, but do not require internal physical contact. In [1, 2], a classification was proposed based on the resources and awareness needed for an attack to succeed. Attacks were classified based on four criteria: Accessibility (A), Re-sources (R), Time (T), and Awareness (W). The awareness criterion (W) divides hardware at-tacks based on the evidence left of an attack on a system, so there are two categories, covert [2] and overt [1].

3. COVERT HARDWARE ATTACK MITIGATION TECHNIQUES

Many mitigation techniques have been proposed to counter covert attacks. Most focus on making chip emissions independent of the operations. Some of these techniques have been designed for a specific attack while others can counter multiple attacks. The covert attack mitigation techniques are described below.

3.1. Hiding

Hiding is a powerful technique that can be used against an attacker attempting to gain information from chip emissions [52, 53]. The following techniques can be used to hide chip emissions.

3.2. Shielding

Shielding is an effective method to hide chip emissions. This can be achieved via physical shielding or filtering of chip emissions. Metal layers on the outside of a chip can be used to shield EM emissions. For FBA, a sensor mesh can monitor the chip operations for interruptions or short circuits and raise an alarm if one of these events occurs. Glass shielding, opaque material or black taping can be used to guard against optical attacks [14]. For ACA, acoustic shielding such as foam can be employed [15]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, ACA, OPLP, OEA, AIT, and Cache.

3.3. Masking

(Blinding) Masking or blinding is a technique used to make it difficult for an attacker to determine the relationship between chip data and emissions. This can be accomplished on a per-gate basis using masking logic, or a per-block basis by randomizing the input data and reversing this operation to obtain the results [16–19]. The input data can also be masked with random data before any operations and the results obtained by removing the mask [27]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, TA, ACA, OPLP, OEA, and AIT.

3.4. Design

Partitioning Design partitioning prevents information leakage between chip regions. For example, regions that operate on plaintext can be separated from those that operate on ciphertext [20, 21]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, TA, ACA, OPLP, and OEA.

3.5. Anti-tampering

(Physical Security) Anti-tampering or physical security is used to limit access by creating a secure zone around a chip. This also reduces the amount of emission data that can be collected [22, 23]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, DPA, TA, ACA, OPLP, OEA, DRA, C-JTAG, FBA, AIT, and Cache. It can also be used to mitigate the overt attacks: O-JTAG, FIT, and FAT.

3.6. Emission Filtering

Hardware and/or software emission filters can be used to reduce the amount of data that is leaked [67]. This technique can be used to mitigate the covert attacks: SPA, SEMA, DEMA, FBA, DPA, ACA, OPLP, and OEA.

3.7. Restricting

Physical Access Restricting access to a device is a simple countermeasure against fault attacks. Encapsulating a device in a tamper-resistant case is an effective means of restricting access [33] , which has been successfully implemented [31]. This technique can be used to mitigate the covert attacks: DRA, C-JTAG, and AIT. It can also be used to mitigate the overt attacks: FIT, O-JTAG, and FAT.

3.8. Randomized

Computation Time Randomizing the computation time of chip operations provides protection against fault at-tacks [33]. This technique can be used to mitigate the covert attack: TA, and the overt attacks: FIT and FAT.

3.9. Deep Sub-micron Technology

Data can be protected using storage devices covered with a top metal layer or constructed with deep sub-micron technology, which makes it difficult for an attacker to access the transistor level or recover data that has been erased [25]. This technique can be used to mitigate the covert attacks: DRA and AIT.

4.OVERT HARDWARE ATTACK MITIGATION TECHNIQUES

Overt attack mitigation techniques are primarily used to prevent an attacker from analyzing the inner structure of a chip. Often an attacker uses an overt attack to understand the chip structure and then use this information in a covert attack. This information can also be used to copy a chip. The overt attack mitigation techniques are described below.

4.1. Error Detection

Error detection codes are used to generate check bits for input data and operation results. If the check bits at the output are incorrect, a fault is detected and the output data is discarded [33]. This technique can be used to mitigate the overt attacks: FIT and FAT.

4.2. Duplicate Operations

Chip operations can be executed multiple times and the outputs considered valid only when they are identical [32]. If the results differ, an alarm is raised. This is not the best solution to defend against fault-based attacks since a fault may still go undetected. It increases the system complexity, but also the resources and time required by an attacker to obtain sufficient data [26], so while implementation is simple, the overhead is high. This technique can be used to mitigate the overt attacks: FIT and FAT.

4.3. Top Layer

Sensor Meshes Sensor meshes are mainly used to protect against microprobing attacks. They are placed above the circuit to detect interruptions and short circuits. If procedures such as selective etching or laser cutting are sensed, an alarm can be raised and countermeasures taken such as erasing nonvolatile memory [35]. These meshes can also protect against under-voltage or over-voltage analysis attacks. This technique can be used to mitigate the overt attacks: FIT, Micro, RE, and DEP.

4.4. Clock Frequency

Sensor Robust low frequency sensors are used to detect tampering which slows the clock frequency [35]. If a sensor raises an alarm, countermeasures such as processor reset and bus line and register grounding can be taken. This technique can be used to mitigate the overt attacks: FIT, Micro, RE, and DEP.

5. DISCUSSION

Algorithms that can be used to assess the security of a system against hardware attacks were presented in [68]. These algorithms were developed based on the criteria, relationships, and/or occurrences of hardware attacks. The criteria considered are Accessibility (A), Resources (R), Time (T), and Awareness (W). Each criterion can be divided into sub-levels depending on the application [12, 13] and target system [50]. For example, consider a defender that has discovered a system is vulnerable to a Timing Attack (TA). There can be several variations of this attack, i.e. TA-1 which requires {covert, limited access, limited

6. CONCLUSION

Determining the possible hardware attacks against a system is a critical step in developing a defence strategy. Once an attack has been identified, an appropriate mitigation technique should be employed to protect the system. Many hardware attacks are covert, in which case a de-fender will not be aware of the attack. Therefore, it is critical to develop mitigation techniques to counter these attacks. Several overt attacks have been developed to gain information about a system, which can later be used in a covert attack, or to make a copy (counterfeit), which is a major concern. Some mitigation techniques can counter multiple attacks, while others have been developed to counter single attacks. Physical security creates a secure zone around a chip to limit the data an attacker can collect from emissions, and is an effective technique against many covert attacks.

REFERENCES

[1] S. Moein and F. Gebali. Quantifying overt hardware attacks: Using ART schema. In Computer Science and its Application, Lecture Notes in Electrical Engineering, vol. 330, Springer, pp. 511–516, 2015.

[2] S. Moein, F. Gebali, and I. Traore. Analysis of covert hardware attacks. In J. Convergence, vol. 5, no. 3, pp. 26–30, 2014.

[3] M. Banga and M. Hsiao. A region based approach for the identification of hardware trojans. In Proc. IEEE Int. Workshop on Hardware-Oriented Security and Trust, pp. 40–47, 2008.

[4] M. Tehranipoor and F. Koushanfar. A survey of hardware trojan taxonomy and detection. In IEEE Design and Test of Computers, vol. 27, no. 1, pp. 10–25, 2010.

[5] M. Rostami, F. Koushanfar, and R. Karri. A primer on hardware security: models, methods, and metrics. In Proceedings of the IEEE, Vol. 102, Issue. 8, pp. 1283–1295, 2014.

[6] S. Moein, S. Khan, T. A. Gulliver, F. Gebali, and M. W. El-Kharashi. An attribute based classification of hardware trojans. in Proc. Int. Conf. on Computer Eng. and Sys., pp. 351–356, 2015.

[7] R. S. Chakraborty, F. Wolff, S. Paul, C. Papachristou, and S. Bhunia. MERO: A statistical approach for hardware trojan detection. In Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science, vol. 5747, Springer-Verlag, pp. 396–410, 2009.

[8] S. Saha, R. S. Chakraborty, S. S. Nuthakki, Anshul, and D. Mukhopadhyay. Improved test pattern generation for hardware trojan detection using genetic algorithm and boolean satisfiability. In Cryptographic Hardware and Embedded Systems, Lecture Notes in Computer Science, vol. 9293, Springer, pp. 577–596, 2015.

[9] S. Moein, J. Subramnian, T. A. Gulliver, and F. Gebali, and M. W. El-Kharashi, Classification of hardware trojan detection techniques. In Proc. Int. Conf. on Computer Engineering and Sys., pp. 357–362, 2015.

[10] S. Adee. The hunt for the kill switch. In IEEE Spectrum, vol. 45, no. 5, pp. 34–39, 2008.

[11] K. M. Goertzel and B. A. Hamilton. Integrated circuit security threats and hardware assurance countermeasures. In Crosstalk - Real Time Inform. Assurance, pp. 33–38. 2013.

[12] S. Moein and F. Gebali. Quantifying covert hardware attacks: Using ART schema. In Proc. Adv. in Inform. Science and Computer Eng., pp. 85–90, 2015.

[13] S. Moein and F. Gebali. A formal methodology for quantifying overt hardware attacks. In Proc. Adv. in Inform. Science and Computer Eng., pp. 63–69, 2015.

[14] J. Loughry and D. Umphress. Information Leakage from Optical Emanations. In ACM Trans. Inform. and Sys. Security, vol. 5, no. 3, pp. 262–289, 2002.

[15] D. Genkin, A. Shamir, and E. Tromer. RSA key extraction via low-bandwidth acoustic cryptanal-ysis. In Advances in Cryptology, Lecture Notes in Computer Science, vol. 8616, Springer, pp. 444–461, 2014.