A Hardware Trojan for Cryptographic Countermeasure Circuits

Masaya Yoshikawa and Takaya Tsukadaira

Abstract— **A hardware Trojan is a malicious hardware virus that is incorporated into the LSI circuit by a designer as the LSI is being designed or manufactured. When the hardware Trojan trigger is not actuated, the LSI acts according to its own specifications. Therefore, a hardware Trojan is difficult to detect using general functional tests. Unlike a software Trojan, a hardware Trojan is difficult to identify from the outside and it cannot be removed since it is physically incorporated into the LSI. In previous studies that focus on hardware Trojan, a hardware Trojan is often incorporated into a cryptographic circuit. Since confidential information is generally protected using a cryptographic circuit, important information can be stolen by attacking that cryptographic circuit. In general, circuits that contain measures to protect them from illegal attacks (hereinafter referred to as countermeasure circuits) are often used as cryptographic circuits. In the future, fault attacks will be the most threatening type of illegal attacks. Measures that use back-check circuits are said to be most effective against fault attacks. Since circuits that do not contain measures against illegal attacks (hereinafter referred to as non-countermeasure circuits) were used as Trojan circuits in previous studies, Trojan circuits are difficult to directly use as countermeasure circuits in systems that use back-check circuits. In order to examine a type of hardware Trojan that will have important implications for the future security of circuits, the present study develops a new hardware Trojan for countermeasure circuits that can be used against fault attacks and verifies the validity of the new hardware Trojan.**

Index Terms—**Hardware Trojan, Cryptographic Circuit, Fault Analysis Attack, Security, Countermeasure Circuit**

I. Introduction

FOLLOWING the advancement of reverse-engineering technologies, 5% of semiconductors in the market are said to be imitations. An investigative report, published by the United States Department of Commerce Bureau of Industry and Security in January 2010, mentions that US military weapons were seriously damaged by imitations of electronic components using semiconductors. Recently, the threat of hardware Trojans in forged electronic components has become even more evident [1]-[27]. A hardware Trojan is a hardware virus. When predetermined conditions are satisfied, that malicious virus performs subversive activities, such as a system shutdown and the leaking of important information, without the LSI users even being aware of that activity.

This research was supported by Japan Science and Technology Agency (JST), Core Research for Evolutional Science and Technology (CREST).

Masaya Yoshikawa and Takaya Tsukadaira are with Department of Information engineering, Faculty of Science and Engineering, Meijo University, Nagoya, JAPAN. (corresponding author to provide e-mail: evolution_algorithm@ yahoo.co.jp).

When the hardware Trojan trigger is not actuated, the LSI acts according to its own specifications. Therefore, a hardware Trojan is difficult to detect using general functional tests. Unlike a software Trojan, a hardware Trojan is difficult to identify from the outside and it cannot be removed since it is physically incorporated into the LSI. As noted in the investigative report mentioned above, in response to the specific hardware Trojan that infected the weapons used by the United States military, in 2007 the United States initiated a program to secure the reliability of the integrated circuits centering on the Defense Advanced Research Projects Agency.

In previous studies that focus on hardware Trojan, a hardware Trojan is often incorporated into a cryptographic circuit. Since confidential information is generally protected using a cryptographic circuit, important information can be stolen by attacking that cryptographic circuit. Previous studies on hardware Trojans can be roughly classified into studies that focus (1) on methods to detect a Trojan and (2) on Trojan circuits. A typical study belonging to the former group proposed a method to measure the electricity consumption and delay time of a chip without a Trojan virus and the researchers constructed a reference model. Consequently, a Trojan virus was detected by comparing the chip and the model.

A typical study belonging to the latter group paid attention to the two functional encryption and decryption blocks that are generally embedded into cryptographic circuits, and proposed a Trojan circuit into which a path connecting the encryption and decryption blocks was incorporated. In general, the path connecting two functional blocks does not exist in cryptographic circuits. Since previous studies on Trojan circuits used non-countermeasure circuits, countermeasure circuits are difficult to use as Trojan circuits. However, in general, countermeasure circuits are often used as cryptographic circuits.

In order to examine the security of future cryptographic circuits, the present study proposes a new hardware Trojan for countermeasure circuits. The present study also verifies the validity and effect of the proposed hardware Trojan by performing evaluation tests.

II. Countermeasure Circuit against a Fault Analysis

A method using a back-check system was reported as a typical measure of the advanced encryption standard (AES) against a fault analysis[28],[29] in a paper [30]. That method uses two clocks to perform the encryption processing of a round. In this way, encryption (decryption) of a 1/2 round is performed in the first clock and decryption (encryption) of a

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

1/2 round is performed in the second clock. The method also checks whether or not the intermediate value at the time of the previous round, which occurred a 1/2 round before, can remain. By performing this process, the intermediate value is confirmed as being unchanged during any given round. Using a key at the final round, the key value is also confirmed as being unchanged.

Figure 1 shows an actual circuit diagram. In Register1 in Figure 1, a value, obtained by performing an EXOR operation for a plane text and a master key, or an intermediate value obtained by performing the encryption processing for a 1/2 round, is stored. In Register2, the value in Register1 is stored in order to compare it to the value of the encryption processing recorded at the time of the round that occurred a 1/2 round before. In Register3, a 128-bit master key that has been input is stored. In Register4, it is assumed that the 128-bit key has been stored before the final round. The processing of each clock is explained using a diagram of operation examples in countermeasure circuits. In the explanation, variables Di and DiX represent intermediate values during the encryption processing (i expresses the number of rounds). In the first clock, an EXOR operation is performed for a master key stored in Register3 and an input plane text, and the output D0 of the operation is stored in Regisetr1. The encryption processing of route (1) is performed for the D0 stored in Register1, and the result is fed back to Register1 as D1X. Simultaneously, the D0 stored in Register1 is delivered to Register2 through route (2). At this moment, D1X and D0 are not yet stored in Register1 and Register2, respectively. In the second clock, D1X and D0 are stored in Register1 and Register2, respectively. The decryption processing of route (1) is performed for the D1X stored in Register1 to obtain D0. This D0 is compared with the D0 stored in Register2 to confirm whether or not the intermediate value remains unchanged. When the intermediate value changes, back-check output 1 outputs an error. Subsequently, the encryption processing of route (3) is performed for the D1X stored in Register1, and the result is fed back to Register1 as D1. Simultaneously, the D1X stored in Register1 is delivered to Register2.

In the third clock, D1 and D1X are stored in Register1 and Register2, respectively. The decryption processing of route (3) is performed for the D1 stored in Register1 to obtain D1X. This D1X is compared with the D1X stored in Register2 to confirm whether or not the intermediate value remains unchanged. When the intermediate value changes, back-check output 1 outputs an error. Subsequently, the encryption processing of route (1) is performed for the D1 stored in Register 1, and the result is fed back to Register1 as D2X. Simultaneously, the D1 stored in Register1 is delivered to Register2.

The same procedure is repeated until the 21st clock, when the back-check of the D10X at the second half of round 10 is completed after the encryption processing at round 10 is completed. In the 21st clock, the back-check is performed for both the intermediate value of a cipher and for the secret keys.

The key at the final round, which is obtained when round 10 is completed, is compared with the key at the final round previously stored in Register4 in order to confirm whether or not these two keys are the same. When a difference between these two keys is observed, back-check output 2 outputs an error.

A cipher is output only when no error occurs in each of the back-check operations all the way up to and including the operations conducted on the 21st clock.

III. Proposed Hardware Trojan

For cryptographic circuits, secret keys are the most valuable information to be concealed. For countermeasure circuits, it is important to prevent the information of the secret keys from being leaked. When secret keys are directly output using a hardware Trojan, the possibility of detecting the output using a functional test is high.

In the present study, secret keys are not leaked due to the actuation of a Trojan; instead, secret keys are specified by annulling the countermeasures and by performing a fault analysis. In this way, the LSI acts according to its specifications in a functional test regardless of the type of input.

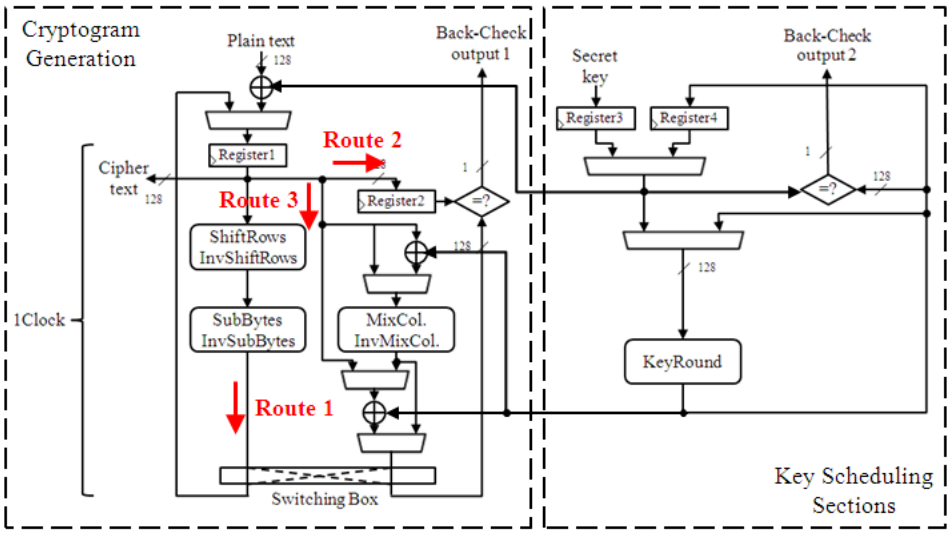

Figure 1 Circuit diagram of countermeasure circuit against a fault analysis

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Unlike the hardware Trojans that have been previously studied, the proposed Trojan is difficult to detect in the traditional LSI test environment. Therefore, the proposed Trojan causes a serious threat.

A. *Countermeasure-annulled hardware Trojan*

The present study proposes a new Trojan circuit for countermeasure circuits against the fault analysis explained in Section 2. For Trojan circuits, two factors are important to consider: the difficulty of detecting the virus and the leaking of valuable information. The difficulty in detection can be achieved by minifying the scale of a Trojan circuit and the leaking of valuable information can be achieved by leaking the information of the secret keys. Consequently, a hardware Trojan can present a serious threat.

By changing the back-check circuit section in Figure 1, a cipher is directly output even when a fault is mixed. Keys are specified by performing a fault analysis using a cipher that has been output (a cipher containing an operation error due to the mixing of a fault). Since no key information is directly output to the outside, the proposed hardware Trojan cannot be detected in general functional tests performed by persons other than attackers who know the conditions needed to actuate the trigger of the proposed hardware Trojan. The configuration and actions of the proposed hardware Trojan are explained using Figure 2.

In the proposed hardware Trojan, a coincidence judgment circuit for input plain texts is added before both back-check output 1 and back-check output 2, as shown in Figure 2. This coincidence judgment circuit checks whether or not a specific plain text (the Trojan trigger) is input. When a specific plain text is input, the results of back-check output 1 and back-check output 2 indicate that “no error occurred” regardless of the back-check results. When a specific plain text is not input (the Trojan trigger is not actuated), the results of back-check output 1 and back-check output 2 are directly output. In the case where the Trojan trigger is not actuated, when a fault is mixed during the encryption processing, the back-check output results indicate that “an error occurred.”

In other words, the LSI acts according to its specifications even when the Trojan trigger is actuated. In this important regard, the proposed hardware Trojan greatly differs from previously studied hardware Trojans.

B. *Easy-to-attack hardware Trojan*

The present study introduces both a countermeasure annulled hardware Trojan and a new easy-to-attack hardware Trojan, by which secret keys can be easily derived using fault attacks. A fault analysis based on key value estimation uses the humming distance between a cipher containing a fault and a correct cipher. The easy-to-attack hardware Trojan outputs a cipher containing a fault at the maximum humming distance from a correct cipher. Figure 3 shows a circuit block diagram of the easy-to-attack hardware Trojan. When the easy-to-attack hardware Trojan is adopted, the accuracy of a fault analysis based on key value estimation can be improved and fault attacks can be easily performed.

IV. Evaluation Experiments and Discussion

A. *Attack experiments on FPGA*

In order to evaluate the threat of the proposed hardware Trojan, attack experiments were performed. Regarding to fault generation method in the experiments, we utilized the insertion of a shortening cycle in a clock.

In the illegal clock method, the setup time constraint of a flip-flop that composes a register is violated by mixing a short clock pulse (glitch) in a clock signal during a specific round of the processing operation, resulting in induction of a malfunction (error).

By mixing a glitch in a clock signal, its cycle is shortened and a flip-flop cannot obtain correct values; consequently, incorrect values are stored in a register. In that method, the circuits are not destroyed. The present study aims to make a change-over on out-of-phase clock signals at a specific timing. With this, a glitch can be inserted into a clock at an arbitrary round. Figure 3 shows the configuration of a glitch generation circuit.

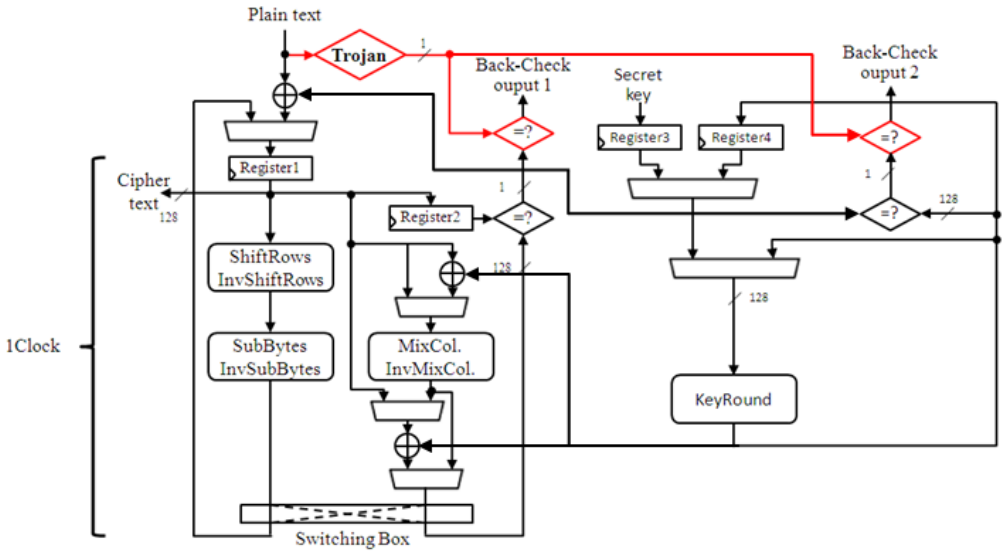

Figure 2 Proposed Hardware Trojan

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

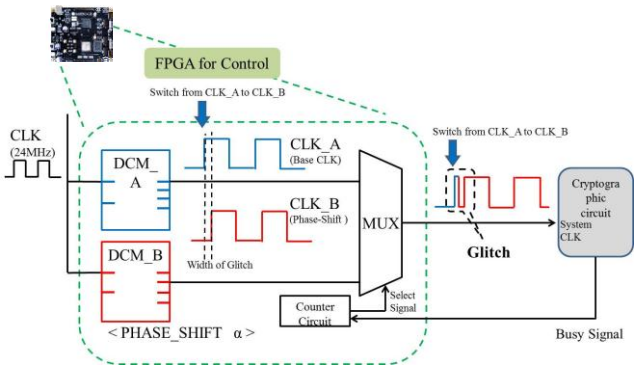

Figure 3 Configuration of a glitch generation circuit

B. *Key derivation tests*

In order to estimate keys, a method to estimate key values that was proposed in a paper [31] is used. This key value estimation method uses the following characteristic: an error due to a fault tends to lean toward a small number of bits. Even if the occurrence location of a fault and the number of faults are unknown, the key values can be estimated. Figures 4 and 5 show the estimation results. The results shown in Figure 4 were obtained using five pairs of a cipher containing a fault and a correct cipher. The results shown in Figure 5 were obtained using 500 pairs of ciphers.

These figures show the key estimation results of each byte. As shown, the horizontal axis expresses the key value (0-255) and the vertical axis expresses the humming distance. In the key value estimation method, a value possessing the maximum humming distance is judged as an estimated key. As shown in Figure 4, when five pairs were used in tests using an actual device, any keys could not be obtained. By contrast, when 500 pairs were used, 14-byte keys could be obtained from 16-byte secret keys as shown in Figure 5.

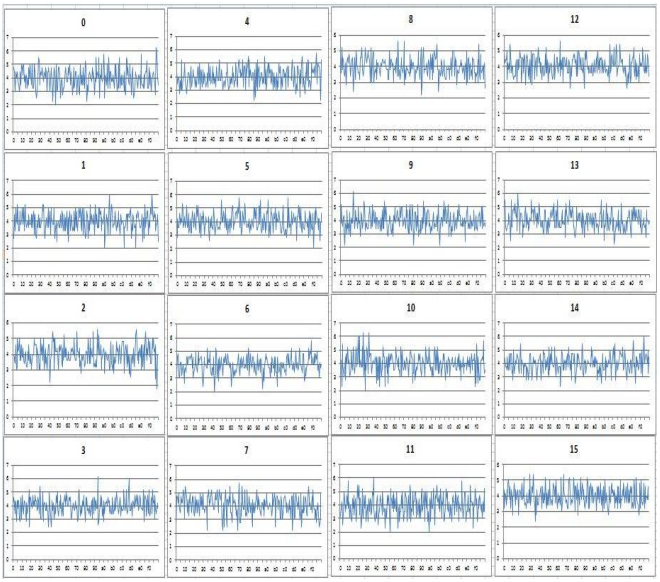

Figure 4 Key analysis result using 5 pairs

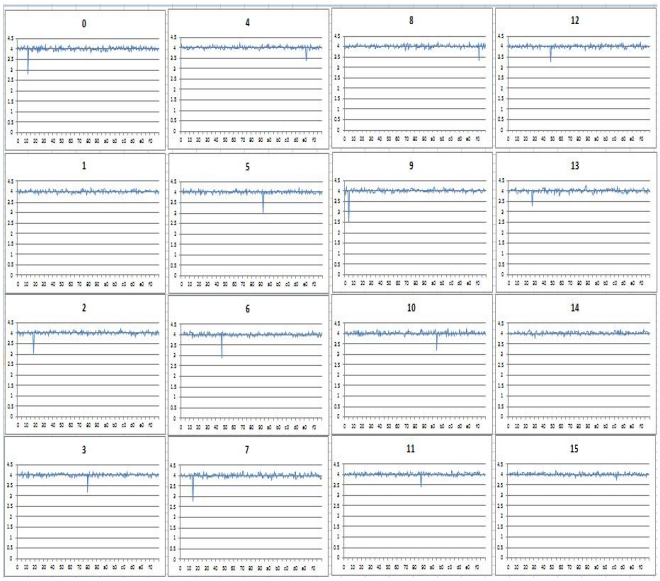

Figure 5 Key analysis result using 500 pairs

C. *Comparison tests*

Figure 6 shows the estimation results obtained using the easy-to-attack hardware Trojan. Similar to the results shown in Figure 4, Figure 6 shows the estimation results obtained using five pairs. As shown in Figure 6, when the proposed easy-to-attack hardware Trojan is adopted, all the 16-byte secret keys could be obtained using five pairs.

By using both the countermeasure-annulled and the easy-to-attack hardware Trojans, the estimation accuracy could be greatly improved.

V. Conclusion

The present study proposed a new Trojan circuit for the AES that contain measures against fault attacks.

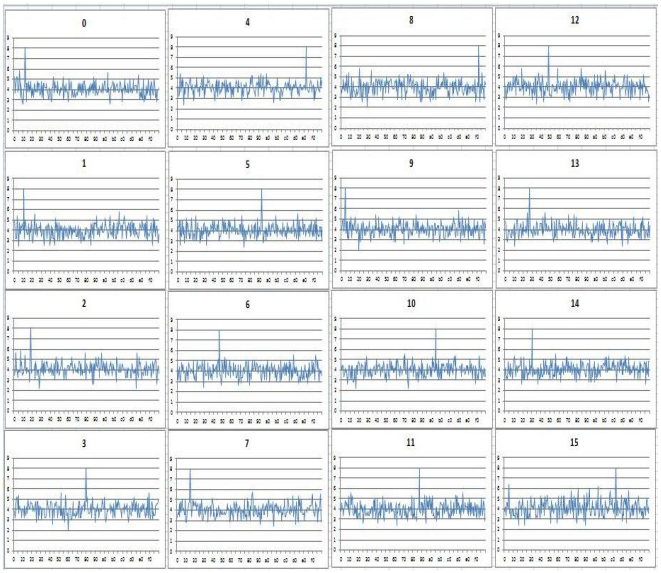

Figure 6 Key analysis result with Easy-to-attack hardware Trojan using 5 pairs

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

In the proposed hardware Trojan, the back-check results are mistakenly detected by inputting a specific plain text that could be understood only by attackers, and a cipher is output even when a fault is mixed during the encryption processing.

Even if a Trojan is actuated, a correct cipher can be output when no fault is mixed. Therefore, the proposed hardware Trojan cannot be detected in functional tests. Thus, the proposed hardware Trojan could be used for applicable countermeasure circuits against illegal attacks. Moreover, the detection of the proposed hardware Trojan was more difficult than the detection of the previously studied hardware Trojans. In the future, we will examine a method to detect a Trojan virus using information about multiple side channels.

REFERENCES

[1] R.S.Chakraborty, S.Narasimhan, S.Bhunia, "Hardware Trojan: Threats and emerging solutions", Proc. of IEEE International High Level Design Validation and Test Workshop, pp.166-171, 2009.

[2] H.Salmani, M.Tehranipoor, "Layout-Aware Switching Activity Localization to Enhance Hardware Trojan Detection", IEEE Transactions on Information Forensics and Security, vol.7, no.1, Part 1, pp.76-81, 2012.

[3] H.Salmani, M.Tehranipoor, J.Plusquellic, "A Novel Technique for Improving Hardware Trojan Detection and Reducing Trojan Activation Time", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.20, no.1, pp.112-125, 2012.

[4] T.Kumaki, Y.Mochizuki, T.Fujino, "A Study on Hardware Trojyan embedded Manchurian LSI for Cipher processing", Technical report of IEICE, vol. 111, no. 328, CPSY2011-46, pp. 21-26, 2011.

[5] Huafeng Liu, Hongwei Luo, Liwei Wang, "Design of hardware trojan horse based on counter", Proc. of International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), pp.1007-1009, 2011.

[6] Song Yun, Qingbao Li, Hongbo Gao, Zhang Ping, "Towards Hardware Trojan: Problem Analysis and Trojan Simulation", Proc. of International Conference on Information Engineering and Computer Science (ICIECS), pp.1-4, 2011.

[7] J.Clark, S.Leblanc, S.Knight, "Risks associated with USB Hardware Trojan devices used by insiders", Proc. of IEEE International Systems Conference (SysCon), pp.201-208, 2011.

[8] Liu Changlong, Zhao Yiqiang, Shi Yafeng, Gao Xingbo, "A System-On-Chip bus architecture for hardware Trojan protection in security chips ", Proc. of International Conference of Electron Devices and Solid-State Circuits (EDSSC), pp.1-2, 2011.

[9] Yier Jin, N.Kupp, Y.Makris, "DFTT: Design for Trojan Test", Proc. of 17th IEEE International Conference on Electronics, Circuits, and Systems (ICECS), pp.1168-1171, 2010.

[10] H.Salmani, M.Tehranipoor, J.Plusquellic, "New design strategy for improving hardware Trojan detection and reducing Trojan activation time", Proc. of IEEE International Workshop on Hardware-Oriented Security and Trust(HOST), pp.66-73, 2009.

[11] S.Narasimhan, Dongdong Du, R.S.Chakraborty, S.Paul, F.Wolff, C.Papachristou, K.Roy, S.Bhunia, "Multiple-parameter side-channel analysis: A non-invasive hardware Trojan detection approach", Proc. of IEEE International Workshop on Hardware-Oriented Security and Trust(HOST), pp.13-18, 2010.

[12] Xiaoxiao Wang, H.Salmani, M.Tehranipoor, J.Plusquellic, "Hardware Trojan Detection and Isolation Using Current Integration and Localized Current Analysis", Proc. of IEEE International Symposium on Defect and Fault Tolerance of VLSI Systems (DFTVS 08), pp.87-95, 2008.

[13] G.T.Becker, A.Lakshminarasimhan, S.Lang Lin, S.Srivathsa, V.B.Suresh, W.Burelson, "Implementing hardware Trojans: Experiences from a hardware Trojan challenge", Proc. of IEEE 29th International Conference on Computer Design (ICCD), pp.301-304, 2011.

[14] R.S.Chakraborty, S.Paul, S.Bhunia, "On-demand transparency for improving hardware Trojan detectability", Proc. of IEEE International Workshop on Hardware-Oriented Security and Trust(HOST), pp.48-50, 2008.

[15] Yier Jin, Y.Makris, "Hardware Trojan detection using path delay fingerprint", Proc. of IEEE International Workshop on Hardware-Oriented Security and Trust(HOST), pp.51-57, 2008.

[16] R.S.Chakraborty, S.Bhunia, "Security against hardware Trojan through a novel application of design obfuscation", IEEE/ACM International Conference on Computer-Aided Design (ICCAD) - Digest of Technical Papers, pp.113-116, 2009.

[17] S.Narasimhan, Xinmu Wang,Dongdong Du, R.S.Chakraborty, S.Bhunia, "TeSR: A robust Temporal Self-Referencing approach for Hardware Trojan detection", Proc. of IEEE International Symposium on Hardware-Oriented Security and Trust (HOST), pp.71-74, 2011.

[18] Min Li, A.Davoodi, M.Tehranipoor, "A sensor-assisted self-authentication framework for hardware trojan detection", Proc. of Design, Automation & Test in Europe Conference & Exhibition (DATE 12), pp.1331-1336, 2012.

[19] J.Clark, S.Leblanc, S.Knight, "Hardware Trojan Horse Device Based on Unintended USB Channels", Proc. of Third International Conference on Network and System Security (NSS 09), pp.1-8, 2009.

[20] F.Wolff, C.Papachristou, S.Bhunia, R.S.Chakraborty, "Towards Trojan-Free Trusted ICs: Problem Analysis and Detection Scheme", Proc. of Design, Automation and Test in Europe (DATE 08), pp.1362-1365, 2008.

[21] Li-Wei Wang, Hong-Wei Luo, "A power analysis based approach to detect Trojan circuits", Proc. of International Conference on Quality, Reliability, Risk, Maintenance, and Safety Engineering (ICQR2MSE), pp.380-384, 2011.

[22] R.Rad, J.Plusquellic, M.Tehranipoor, "A Sensitivity Analysis of Power Signal Methods for Detecting Hardware Trojans Under Real Process and Environmental Conditions", IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.18, no.12, pp.1735-1744, 2010.

[23] M.Tehranipoor, F.Koushanfar, "A Survey of Hardware Trojan Taxonomy and Detection", IEEE Design & Test of Computers, vol.27, no.1, pp.10.25, 2010.

[24] R.Karri, J.Rajendran, K.Rosenfeld, M.Tehranipoor, "Trustworthy Hardware: Identifying and Classifying Hardware Trojans", IEEE Journals & Magazines Computer, vol.43, no.10, pp.39-46, 2010.

[25] A.Adamov, A.Saprykin, D.Melnik, O.Lukashenko, "The problem of Hardware Trojans detection in System-on-Chip", Proc .of 10th International Conference - The Experience of Designing and Application of CAD Systems in Microelectronics (CADSM), pp.178-179, 2009. CADSM 2009.

[26] J.Rajendran, E,Gavas, J.Jimenez, V.Padman, R.Karri, "Towards a comprehensive and systematic classification of hardware Trojans", Proc.of 2010 IEEE International Symposium on Circuits and Systems (ISCAS), pp.1871-1874, 2010.

[27] M.Banga, M.S.Hsiao, "A Novel Sustained Vector Technique for the Detection of Hardware Trojans", Proc. of 22nd International Conference on VLSI Design, pp.327-332, 2009.

[28] Z.Wang, M.Karpovsky, A.Joshi, "Secure Multipliers Resilient to Strong Fault-Injection Attacks Using Multilinear Arithmetic Codes", IEEE Trans. on Very Large Scale Integration (VLSI) Systems, pp.1-13, 2011.

[29] S.S.Ali, D.Mukhopadhyay, "A Differential Fault Analysis on AES Key Schedule Using Single Fault", Proc. of 2011 Workshop on Fault Diagnosis and Tolerance in Cryptography (FDTC), pp.35-42, 2011.

[30] RCIS,"Standard Cryptographic LSI Specification with Side Channel Attack Counter Measures", AIST, pp.85-89,2010.

[31] M.Ono, M.Katsube, M.Shiozaki, T.Fujino, M.Yoshikawa, "Architecture aware fault analysis based on differential presumption for multiple errors and its evaluation", IEEJ Trans. EIS, Vol.132, No.12, pp.1888-1896, 2012.

ISBN: 978-988-19252-6-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)