Hardware Security of FPGAs

Jungmin Park
Computer Engineering
Iowa State University

Modern electronic devices expose themselves to danger from adversaries who try to extract secrete information. Security has become a major issue in the embedded system since several years. Even though the devices are protected by modern cryptographic algorithms, such as AES or RSA, the security is limited. Specially, hardware attack like differential power analysis (DPA) is very powerful method in order to obtain secrete key. Reconfigurable architectures like FPGA are selected as an efficient target for secure embedded systems. In this report, we surveyed an overview of existing attacks and countermeasures against FPGAs.

Categories and Subject Descriptors: CprE583 [**Survey**]: Hardware Security of FPGAs

Additional Key Words and Phrases: Crytpgraphy, Encryption, DPA, FPGA, Side-channel attack, Fault injection, Reverse engineer

1. INTRODUCTION

Security of electrical system, such as ATM, mobile devices, smart card and so on, is very important. If private information is revealed by any adversary, enormous damage may be caused. To protect valuable data against attacks, the data should be encrypted by cryptographic algorithms like AES or RSA. These modern cryptographic algorithms are theoretically much enough to resist cryptanalysis attacks but implementation of cryptosystem can support the opportunity for the adversary to make cryptosystem useless. For example, side channel attack is the method which uses leakage power or electromagnetic radiation to reveal secret key. It is based on the fact that power consumption depends on input data of the cryptosystem. Differential power analysis of side channel attack has been shown to be especially powerful.

Cryptographic functions in the system need high performance because they handle large volumes of data at high data rates. Thus the dedicated hardware often is used for implementation. FPGA-based design also is recommended for cryptographic hardware. It has some advantages of low cost and fast time to market compared with ASICs. It is possible to dynamically change or update the functionality of the system in order to react any attack. But there are several types of hardware attacks against FPGAs. First, one of these types is side-channel attack. By observing power consumption, timing and/or electromagnetic radiation from FPGAs, an adversary can find secret keys. The second hardware attack is to induce errors during encryption (or decryption) process in order to collect information about secret information, such as cryptographic keys. It is called fault injection. The third attack is reverse engineering. In case of SRAM-based FPGA, it should generate the bit-stream for configuration whenever it powers on. By obtaining the bit-stream, an attacker can clone the same design in other FPGAs. We describe existing countermeasures according to attack these types in the following sections.

2. SIDE-CHANNEL ATTACK

Side-channel analysis refers to the use of any information unintentionally leaked from a device while a cryptographic computation is performed. Unintentional information leakage could be in the form of timing information, acoustics, electromagnetic waves, power dissipation, etc. The power dissipation is mostly used for side-channel attack. There are two methods to use the power dissipation of side-channel attack; Simple power analysis (SPA) and Differential power analysis (DPA)[Kocher et al. 1999]. The power consumption has the direct relation with the Hamming weight of the data. The more the Hamming weight of the data is, the more the peak power operating with the data is. SPA determines the Hamming weight by measuring the pulse height of the power consumption signal at the exact cycle of the instruction that accessed the key byte. Knowledge of the Hamming weight of each key byte can be used to reduce the number of keys that need to be checked during a brute-force attack. DPA is more powerful than SPA because the attacker does not need to know the details about how the algorithm was implemented. This technique gains strength by using statistical analysis to help recover side-channel information. Kocher et al.[Kocher et al. 1999] determined the secret key used by a smart card running the DES algorithm. These techniques can also be generalized to attack other cryptographic algorithms.

FPGAs seem more vulnerable than ASICs to power analysis by different reasons. The FPGA has heavy-loaded wires made up of long lines or lines segmented by pass-transistors. Capacitance load by these long wires causes adversary to measure power consumption more easily. DFFs in each logic block of FPGA are very fast but consume high power. Standaert et al. [Standaert et al. 2004] describe successful DPA against DES implementations programmed into FPGA.

2.1 Runtime reconfiguration

One of the published countermeasures against side-channel attacks is to use runtime reconfiguration [Katashita et al. 2009]. For example, in the case of AES algorithm, differential power analysis targets sub-byte operation. Power consumption is generally correlated with data involved in the sub-byte function because this operation is always done with the same logic gates. With partial reconfiguration, sub-byte function could be realized differently at every computation, so power consumption will be different.

2.2 Masking method

Masking methods[Trichina 2003] exploit this remark; input data and secret key are masked with a random mask before calculation and after computation a mask correction is performed in order to obtain the expected value. XOR operation is used for masking. Thus the same input data with random masks could lead to different power traces.

2.3 Leak resistant arithmetic

Leak resistant arithmetic (LRA) is based on the residue number system (RNS) representation and the RNS Montgomerys modular multiplication proposed in [Mesquita et al. 2006][Mesquita et al. 2006][Ciet et al. 2003]. A residue number system relies on the Chinese Reminder Theorem (CRT). This theorem indicates that it is possible to represent a large integer using a set of smaller integers. With this representation, simple arithmetic operations, like additions, subtractions and multiplications, can be made absolutely in parallel. In addition to parallelism possibilities, LRA could also offer an algorithmic countermeasure against side-channel attacks. It is possible to compute the same calculus in different bases with LRA. Thus the same modular multiplications lead to different basic RNS operations, so the power consumption will be different if the RNS base is different. RNS bases have to be chosen randomly in order that attacker could not know the base value.

2.4 Wave dynamic differential logic

The countermeasure of side-channel attacks in logic level is to use a dual-rail four-phase protocol, that ensures both that transitions are independent of the target values (logical 0 or 1) and there are no leaks induced by consecutive data correlation. The most straightforward method built upon the dual-rail four-phase protocol is referred to as wave dynamic differential logic (WDDL) [McEvoy et al. 2009] [Amouri et al. 2010]. The WDDL logic uses a pair of wires to encode each signal in the design. For example, logic 0 is encoded as (0, 1) and logic 1 is encoded as (1, 0). For every gate in a dual-rail circuit, there exists another gate performing the complementary function. We will refer to this pair of signal paths as the direct and complementary circuits, respectively. Signals in a circuit using precharge logic exist in two states during each clock cycle. In the precharge phase, every signal in the circuit is set to the precharge value which is logic 0 normally. An evaluation phase follows, where each signal is set to its appropriate logic value. By employing the WDDL logic, a hardware designer can ensure that each pair of complementary wires in the design undergoes exactly one sequence of bit transitions during each clock period. This characteristic of the WDDL logic forms the basis of a DPA-resistant implementation, whose power consumption is constant in each clock cycle, regardless of the data being processed.

3. FAULT INJECTION

The fault injection is to induce errors during encryption (or decryption) process in order to obtain secretes. In most cases the injection of a fault is done in the last round of an algorithm. The reason is that the mark of the fault is more visible in the ciphered result. A common way to avoid fault injection attacks is to use redundancy[ACTEL 2005]. Critical parts in the design are replicated, then outputs are compared, thus errors could generally be detected. Mathematical error detection can also be used and [Ciet et al. 2003] shows how to use redundant information to detect potential errors during calculus process. This mechanism relies also on RNS representation, but an extra modulus is used to verify the correctness of the result. At the end of each modular multiplication the verification is possible, thus this type of attack can be detected. Moreover physical location of each operator can be changed dynamically during the FPGA lifetime. Therefore, accurate fault injection helped by laser or focus ion beam, are no longer possible so that attackers cannot identify operator position precisely.

4. RESERVE ENGINEERING

In FPGAs, configuration bitstream causes a potential weakness in secure applications. This information can be used to perform various types of attacks and have to be considered at the system level. For low cost SRAM FPGAs, it is very simple to retrieve the entire bitstream even without read-back mechanism. Adversary can probe the data line between FPGA and EEPROM in order to get the bitstream. This threat does not exist for non-volatile FPGAs because configuration data are stored inside the device, so only intrusive attacks could be performed. For most advanced FPGAs, bitstream encryption mechanisms [Hori et al. 2008] are available. A secret encryption key is stored inside the programmable device and an external battery is used to maintain key value for volatile FPGAs. Thus the device accepts encrypted bitstream and uses its dedicated decryption engine to get un-ciphered data. Attackers could not decrypt the bitstream without the secret key. With this feature, attackers have to discover secret key using for example intrusive attacks in order to recover bitstream data.

Remote configuration is an interesting FPGA feature that allows system upgrade, by removing potential security breach, or upgrading algorithms. But this feature must be strongly secured because it gives many possibilities to attackers. The first related threat is undesired re-configuration, the design could be remotely changed by attacker without user permission. A simple switch button could avoid this threat. The second is man on the middle attacks. The user wants to upgrade his programmable security device, but an attacker intercepts his request and replies with a fake configuration. Therefore connection must be secured with authentication and integrity checking engine. Such secured mechanisms are already suitable for most FPGAs, Actel or Xilinx application notes [ACTEL 2005] describe secured remote configuration schemes.

5. CONCLUSION

Digital private information is stored within mobile devices, network server and so on. The quantity of information is getting larger every day and exchanged or transferred information through network has also increased exponentially. Thus the importance of security should be considered significantly. Cryptographic algorithm such as AES or RSA is used in embedded system for security and FPGAs are adapted for implementation of cryptosystem due to some advantages. But because of weakness caused from technical implementation, such as leakage power consumption, security is unsure completely. In this report, we surveyed existing hardware attacks against FPGAs and classify those into three types; side-channel attack, fault injection and reverse engineering. Also, the existing countermeasure for any type attack is described. These countermeasures are not perfect against all appeared hardware attacks. New technique for attack is making by adversary and the new powerful adversary will maybe attack our secure system tomorrow. Even though it is difficult to anticipate which attack will appear, we should be ready to defend new attacks always.

REFERENCES

2001. Federal information processing standard, fips pub 197 advanced encryption standard (aes).

ACTEL. 2005. proasic3/e security. *In appplication note, actel corporation.*

Amouri, E., Marrakchi, Z., and Mehrez, H. 2010. Impact of dual placement on wddl design security in mesh-based and tree-based fpgas. In *Ph.D. Research in Microelectronics and Electronics (PRIME), 2010 Conference on.* 1 –4.

Ciet, M., Neve, M., Peeters, E., and jacques Quisquater, J. 2003. Parallel fpga implementation of rsa with residue number systems can side-channel threats be avoided. In *46 th . International Midwest Symposium on Circuits and Systems: MWSCAS 03.*

Hori, Y., Satoh, A., Sakane, H., and Toda, K. 2008. Bitstream encryption and authentication with aes-gcm in dynamically reconfigurable systems. In *Field Programmable Logic and Applications, 2008. FPL 2008. International Conference on.* 23 –28.

Katashita, T., Satoh, A., Sugawara, T., Homma, N., and Aoki, T. 2009. Development of side-channel attack standard evaluation environment. In *Circuit Theory and Design, 2009. ECCTD 2009. European Conference on.* 403 –408.

Kocher, P., Jaffe, J., and Jun, B. 1999. Differential power analysis.

McEvoy, R. P., Murphy, C. C., Marnane, W. P., and Tunstall, M. 2009. Isolated wddl: A hiding countermeasure for differential power analysis on fpgas. *ACM Transactions on Reconfigurable Technology and Systems (TRETS) 2,* 1 (April), 1–23.

Mesquita, D., Badrignan, B., Torres, L., Sassattell, G., Robert, M., Bajard, J.-C., and Moraes, F. 2006. A leak resistant architecture against side channel attacks. In *Field Programmable Logic and Applications, 2006. FPL '06. International Conference on.* 1 –4.

Mesquita, D., Badrignans, B., Torres, L., Sassatelli, G., Robert, M., and Moraes, F. 2006. A leak resistant soc to counteract side channel attacks. In *System-on-Chip, 2006. International Symposium on.* 1 –4.

Standaert, F.-X., rs, S. B., Quisquater, J.-J., and Preneel, B. 2004. Power analysis attacks against fpga implementations of the des. In *FPL'04*. 84–94.

Trichina, E. 2003. Combinational logic design for aes subbyte transformation on masked data. Tech. rep., IACR report.