北京大学学报(自然科学版) 第 52 卷 第 2 期 2016 年 3 月
Acta Scientiarum Naturalium Universitatis Pekinensis, Vol. 52, No. 2 (Mar. 2016)
doi: 10.13209/j.0479-8023.2015.126

# A Non-invasive Fault Attack on FPGA-based Cryptographic Applications

LIAO Nan[1], CUI Xiaoxin[1,†], LIAO Kai[1], WANG Tian[1], YU Dunshan[1], CHENG Yufang[2]

1. Institute of Microelectronics, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871; 2. Nationz technologies Inc, Shenzhen 518057; † Corresponding author, E-mail: cuixx@pku.edu.cn

**Abstract** A non-invasive, high-efficient and low-cost fault attack is realized on FPGA-based cryptographic applications. Based on the setup failures in critical paths, faults are injected into the FPGA devices by lowering the supply voltage. Then the encryption key can be retrieved efficiently with an appropriate fault model. In the attack experiments, the full 128-bit key of AES is retrieved correctly with only 8 pairs of correct and faulty ciphertexts within a few minutes, by using a power supply and a personal computer, based on the FPGA platform

**Key words** fault attack; FPGA; AES; setup failure

# 一种针对 FPGA 密码模块的非侵入式故障攻击

廖楠[1] 崔小欣[1,†] 廖凯[1] 王田[1] 于敦山[1] 程玉芳[2]

1. 北京大学信息科学技术学院微电子学研究院，北京 100871; 2. 国民技术股份有限公司，深圳 518057;
† 通信作者，E-mail: cuixx@pku.edu.cn

**摘要** 在 FPGA 平台上，利用降低电源电压的方法使电路关键路径上的数据建立失败，从而达到注入故障的目的。基于合适的故障模型，攻击者可以有效地获取密钥信息，实现了针对密码模块的高效率、低成本的非侵入式故障攻击方法。攻击实验利用一台电压源和一台个人电脑，通过 8 组正确和错误密文对，成功地恢复出一个 FPGA 中 AES 密码模块的 128bit 完整密钥。

**关键词** 故障攻击; FPGA; AES; 建立失败
**中图分类号** TN492

Nowadays FPGA has become a significant device in finance, communication and military fields due to its flexibility and performance advantages. Since the data security is one of the most important issues in these sensitive fields, an increasing attention has been paid to the FPGA security research[1]. In the modern FPGA design, strict cryptographic algorithms are applied to provide closed reliable computing environments for the sensitive information. However, as the development of physical attacks, mathematical algorithms alone cannot guarantee the security of FPGA device. Under the physical attacks, sensitive information which is correlated to the secret key will leak. Thus the attacker can recover the secret key and break into the cryptosystem. Attacks that target straightly the physical implementations of FPGA-based cryptographic applications have become the main threat in recent years.

Now, side channel attacks (SCA) are the most popular attacks aiming directly at the FPGA device[2].

国家自然科学基金(61306040)、北京市自然科学基金(4152020)和深圳市战略新兴产业发展专项资金创新环境建设计划(ZDSY20130402095348589)资助

收稿日期: 2014–12–18; 修回日期: 2015–03–19; 网络出版日期: 2016–02–11

Power consumption, timing information and electromagnetic leaks are usually utilized as side channel information to disclose the encryption keys. However more and more countermeasures based on software and hardware have been proposed to resist SCA. Thus some new methods like fault attack and template attack are introduced. Among these methods, fault attack, which was first presented in 1997, has become a powerful and popular attack against cryptographic systems[3].

Fault attack works by inducing faults to alter either the control flow or the internal-state data of the cryptographic algorithm implemented on the target device. Based on the fault model, the attacker can acquire sufficient information from the faulty ciphertexts to retrieve the secret key. Fault attacks have been successfully employed to attack all kinds of cryptographic algorithm like the AES, DES, ECC and RSA respectively, even if the cryptographic system is mathematically secure and endowed with countermeasures against SCA.

Several techniques have been proposed to inject faults into the hardware implementation, i.e. the clock frequency variation, the temperature variation and the irradiation by a laser beam. However these methods are either hard to control the variable precisely or too complicated and high-cost. Moreover some methods will cause unrecoverable damage to the device. In this paper, a non-invasive, high-efficient and low-cost technique is proposed to inject faults into the FPGA device by lowering its supply voltage. This method is successfully employed on a FPGA board designed with countermeasures against physical attacks. Using a power supply and a personal computer, the full 128-bit key of AES is retrieved correctly by 8 pairs of correct and faulty ciphertexts within a few minutes.

# 1 Fault Injection Technique

For a logic circuit, it needs a setup time for the logic gates to switch into a stable state. Raising the clock frequency or lowering the supply voltage may make some logic paths fail to setup properly[4]. Based on this principle, the proposed technique injects faults by lowering the supply voltage while keeping the clock frequency fixed. By gradually lowering the supply voltage, the longest critical path will fail first, thus the device will output faulty results. More setup failures will appear if the voltage continues to decline. However if the voltage drops down to a certain point, the setup failures will be substituted by functional errors, which may make the device stop working. Thus the exploitable voltage range starts from the point the longest critical path fails the first time to the point functional errors appear. It can clearly been seen that finding out the critical path of cryptographic algorithm implementation is critical to the attack under the fault injection technique.

AES is chosen as the target cryptographic algorithm to implement the attack in this paper. AES is an iterated block cipher that can encrypt 128-bit wide plaintext blocks using a 128-bit, 192-bit or 256-bit key[5]. Without loss of generality, this paper focuses on the 128-bit key size version AES. The AES cipher is based on a ten-round iteration which is applied to the 16 bytes data block to be encrypted. A fixed sequence of operations is consisted in each round: SubBytes, ShiftRows, MixColumns and AddRoundKey. The only exception to the repetition is the last round of encryption where the MixColumns is missing and an extra AddRoundKey performed before the first round.

Obviously, the four operations are the most important part of the encryption process, which compose the critical path of the AES. When the supply voltage is lowered in the exploitable voltage range, setup failures will appear in the encryption round, influencing the 16 bytes internal-state data. Since the implementation has an 8-bit data path, the fault type will be “byte-flip”. Besides, the stress caused by the insufficient supply voltage remains gentle, dysfunctions will not appear suddenly and thus single faults that most fault models are based on will appear at a high probability. As for the fault location, because the critical path is highly data-dependent and each round which the same operations, it can be predicted that the faults will appear randomly in each round. Therefore

the attacker can pick out the exploitable ones according to the fault model.

## 2 Fault Model

According to the fault injection technique and the AES algorithm, single byte model which aims at the 16 bytes inner-state data will be considered. By comparing multifarious models in literature, the fault model proposed by Dusart et al.[6] seems to be the most suitable one in this research. Dusart et al. proposed the differential fault analysis on AES, which is based on the single byte fault injection between the MixColumns step in the eighth round and that in the ninth round, as shown in Fig. 1. Since the last round doesn't have MixColumns step, the fault only influences 4 of the 16 bytes output ciphertext. Furthermore, it is worth nothing that the 4 bytes' location is decided by the fault position. If the single-byte fault position is in the first column of the state $w$ like Fig. 1, the faulty bytes in the ciphertext will be $\{\tilde{S}10_{0,0}, \tilde{S}10_{1,3}, \tilde{S}10_{2,2}, \tilde{S}10_{3,1}\}$. For the fault position in columns 2, 3 and 4, the corresponding faulty bytes will be $\{\tilde{S}10_{0,1}, \tilde{S}10_{1,0}, \tilde{S}10_{2,3}, \tilde{S}10_{3,2}\}$, $\{\tilde{S}10_{0,2}, \tilde{S}10_{1,1}, \tilde{S}10_{2,0}, \tilde{S}10_{3,3}\}$ and $\{\tilde{S}10_{0,3}, \tilde{S}10_{1,2}, \tilde{S}10_{2,1}, \tilde{S}10_{3,0}\}$, respectively. Thus it is easy for the attacker to select the exploitable faults.

Analyzing the fault propagation in Fig. 1, the operation of ShiftRows, MixColumns and SubBytes are fixed, and only the last round key affects the output ciphertext, thus the attacker can make a hypothesis on the 4 bytes round key in the analysis. Define the state before the MixColumns in round 9 as $w$. Since the attack only care about the difference between the correct $w$ and the faulty one, and the operations from $w$ to S10 are all linear except the SubBytes in round 10. Therefore the effect of AddRoundKey in round 9 can be bypassed. When a pair of correct and faulty ciphertexts is obtained, the attacker inverts the ciphertexts to the states $w$ and $\tilde{w}$ ($\tilde{w}$ stands for the faulty one). The difference $\delta = w \oplus \tilde{w}$ can be easily obtained since the AddRoundKey in round 9 can be ignored as shown in Eq. (1).

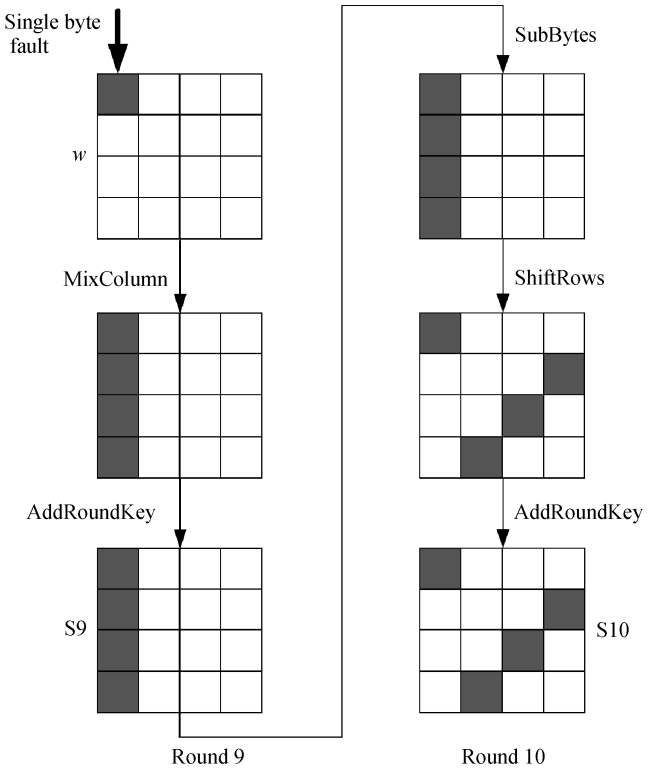


**Fig. 1　Single byte fault propagation in AES**

$$\begin{aligned}\delta &= w \oplus \overline{w} \\ &= \text{InvMixcolumn}(\text{K9} \oplus \text{S9}) \oplus \text{InvMixcolumn}(\text{K9} \oplus \tilde{\text{S}}9), \\ &= \text{InvMixcolumn}(\text{S9} \oplus \tilde{\text{S}}9), \end{aligned} \tag{11}$$

$\text{S9} = \text{InvSubBytes}(\text{InvShiftRows}(\text{K10} \oplus \text{S10}))$,

$\tilde{\text{S}}9 = \text{InvSubBytes}(\text{InvShiftRows}(\text{K10} \oplus \tilde{\text{S}}10))$。

Based on the analysis in the last section, the fault injected is most likely a single byte fault. Therefore the attacker can check if the obtained difference $\delta$ is composed of a single byte. Depending on whether the difference is only a single byte or not, the attacker can decide if the 4 bytes round key hypothesis is a valid one or not. For one pair of correct and faulty ciphertexts, there can be hundreds of valid candidates, however with two pairs which have the same fault locations, the corresponding 4 bytes round key can be uniquely determined with a reasonably high probability. Thus the complete last round key can be recovered with 8 pairs of correct and faulty ciphertexts.

# 3 Attack Experiments

## 3.1 Target platform

The attack target is a FPGA board specifically designed to secure the cryptographic applications against physical attacks, which contains a cryptographic FPGA Virtex-XC5VLX110 and a control FPGA Virtex2Pro-XC2V1000. The standard AES algorithm is implemented in the cryptographic FPGA and the control FPGA is responsible for sending and receiving data. Thus the low voltage attack will be carried out on the cryptographic FPGA. The cryptographic FPGA operates correctly at a frequency of 50 MHz with a nominal supply voltage of 1 V. In the experiment, the cryptographic FPGA is fed by a power supply with appropriate precision and the voltage measures are taken with a multimeter with a precision of 1 mV. The data transmission is realized by a USB cable connecting the FPGA and the computer. All the data processing is realized on the PC. The experiment platform is shown in Fig. 2.

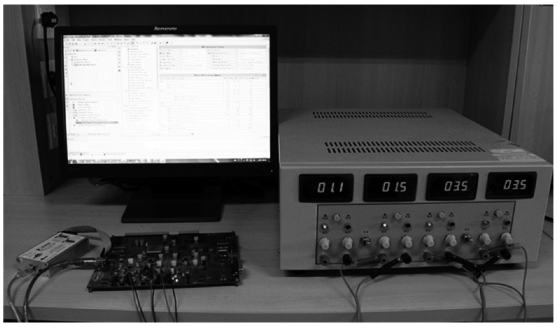


**Fig. 2 Experiment platform**

## 3.2 Experiment results

The exploitable voltage range should be found out first. Slowly lowering the supply voltage, it can be observed that the circuit starts to output faulty ciphertexts for a voltage of about 835 mV and beneath. By continually lowering the voltage to about 785 mV, the target circuit stops working, which indicates that functional errors appear. Therefore the exploitable voltage ranges from the 785 mV to 835 mV.

To further investigate the characteristics of faults, we chose several intermediate voltages within the exploitable voltage range to implement the attack. For the effectivity and consistency of comparison, twenty thousand different plaintexts and one key are prepared. For each voltage level, these plaintexts are encrypted by the same key, and the twenty thousand ciphertexts are recorded. By comparing the ciphertext with the corresponding correct one, we can know whether the result is faulty. Since the fault model is based on the single fault between the MixColumns step in the eighth round and the MixColumns step in the ninth round, we can further find out the exploitable faults as presented in last section. The number of the total faults and the exploitable ones at each voltage are depicted in Fig. 3. As shown in the figure, the FPGA moves from an error-free state to a fully faulty behavior within about 25 mV. However the exploitable faults do not show the same trend, its distribution is like a bell shape. From the voltage of 829 mV, the number of exploitable faults has a sharp increase as the voltage decrease and reaches the maximum of 296 at about 823 mV. Continue lowering the voltage, the exploitable faults also reduce quickly, for instance, the number at 814 mV is only 15. When

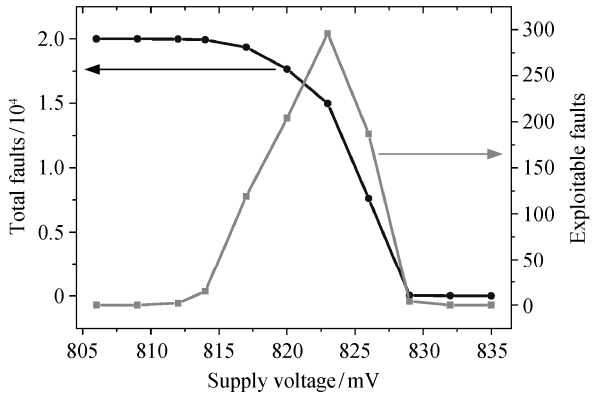


**Fig. 3　Number of total faults and exploitable faults**

the voltage is lowered beneath 812 mV, the exploitable faults do not appear any more. The phenomenon may be owing to the fact that multiple critical paths are violated by faults when the voltage is beneath 812 mV. Therefore the appropriate supply voltage should be set between 814 mV and 823 mV when implementing the attack.

Based on the analysis above, the supply voltage is set at 826 mV. At this voltage, 7622 total faults including 187 exploitable faults are obtained. For further classification, all the fault situations, namely $\{\tilde{S}10_{0,0}, \tilde{S}10_{1,3}, \tilde{S}10_{2,2}, \tilde{S}10_{3,1}\}$, $\{\tilde{S}10_{0,1}, \tilde{S}10_{1,0}, \tilde{S}10_{2,3}, \tilde{S}10_{3,2}\}$, $\{\tilde{S}10_{0,2}, \tilde{S}10_{1,1}, \tilde{S}10_{2,0}, \tilde{S}10_{3,3}\}$ and $\{\tilde{S}10_{0,3}, \tilde{S}10_{1,2}, \tilde{S}10_{2,1}, \tilde{S}10_{3,0}\}$, are covered by the exploitable faults. For each fault situation, two pairs of correct and faulty ciphertexts are selected to do the differential fault analysis. For example, we successfully recover the last round key $\{K10_{0,0}, K10_{1,3}, K10_{2.2}, K10_{3,1}\} = \{3F, E7, F1, 42\}$ with two pairs of correct and faulty ciphertexts which are shown in Fig.4. Using this method for other situations, the complete last round key are correctly recovered as 3FF86D98C44CCD429AF9F14CEAE7 538D. Thus it proves that the non-invasive low voltage fault attack technique is effective for the FPGA-based cryptographic applications, and is quiet a high-efficient and low-cost attack technique since the whole process from collecting data to completing analysis only costs a few minutes with a power supply and a personal computer.

| | | | |
|---|---|---|---|
| 00 | 24 | AE | C8 |
| 72 | 1D | AE | 3F |
| 7D | 66 | 1C | EF |
| A1 | 50 | 86 | 38 |

| | | | |
|---|---|---|---|
| 66 | 24 | AE | C8 |
| 72 | 1D | AE | 30 |
| 7D | 66 | EB | EF |
| A1 | E7 | 86 | 38 |

(a) Pair one: correct (left) and faulty (right) ciphertexts

| | | | |
|---|---|---|---|
| 26 | 28 | 4F | 28 |
| 22 | E0 | 0D | 08 |
| D1 | B9 | CE | 0C |
| BD | 03 | 87 | FE |

| | | | |
|---|---|---|---|
| D3 | 28 | 4F | 28 |
| 22 | E0 | 0D | EC |
| D1 | B9 | FF | 0C |
| BD | 25 | 87 | FE |

(b) Pair two: correct (left) and faulty (right) ciphertexts

**Fig. 4　Two pairs of correct and faulty ciphertexts**

For further consideration, the distribution of single faults in each round is investigated. It is done by inverting the faulty ciphertexts with the known key and calculating the difference between each state of the faulty and the correct ones. The round at which the single bye difference appears is considered to be the round injected by single fault. The data at the supply voltage of 826 mV is selected. Fig. 5 depicts the result of distribution. 2046 single faults of 7622 total faults spread uniformly at all the ten rounds, which can prove that single byte fault models based on other specific rounds apply to this injection technique too. It can be predicted that combining different fault models can improve the attack efficiency, which will be our direction of further research.

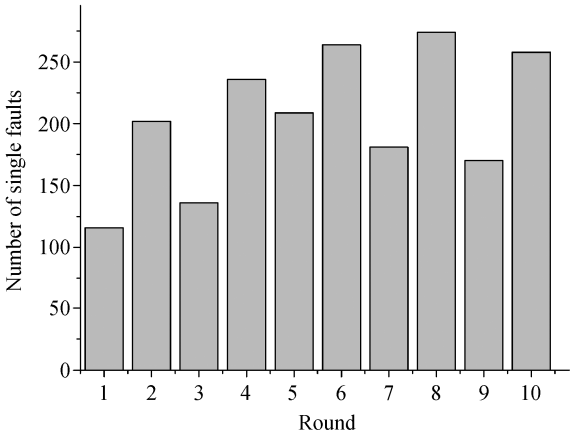


**Fig. 5　Distribution of single faults in each round**

## 4 Conclusion

A non-invasive, high-efficient and low-cost fault attack on FPGA-based cryptographic applications is proposed in this paper. By lowering the supply voltage of FPGA device, the setup failures appear in the critical path of implementation, which leads to faulty ciphertexts. Based on appropriate fault model, the encryption key can be retrieved easily. Attack experiments show that the full 128-bit key of AES is retrieved correctly by only 8 pairs of correct and faulty ciphertexts within a few minutes with the help of a power supply and a personal computer.

### References

[1] Wollinger T, Paar C. Security aspects of FPGAs in cryptographic applications // new algorithms, architectures and applications for reconfigurable computing. Cambridge, MA: Springer, 2005: 265–278

[2] Hori Y, Katashita T, Sasaki A, et al. Sasebo-giii: a hardware security evaluation board equipped with a 28-nm FPGA // Consumer Electronics (GCCE), 2012 IEEE 1st Global Conference on. Tsukub, 2012: 657–660

[3] Boneh D, DeMillo R A, Lipton R J. On the importance of checking cryptographic protocols for faults // Advances in Cryptology — EUROCRYPT'97. Berlin: Springer, 1997: 37–51

[4] Barenghi A, Bertoni G, Parrinello E, et al. Low voltage fault attacks on the RSA cryptosystem // Fault Diagnosis and Tolerance in Cryptography (FDTC), 2009 Workshop on. Lausanne, 2009: 23–31

[5] Daemen J, Rijmen V. The design of Rijndael: AES-the advanced encryption standard. Berlin: Springer, 2002

[6] Dusart P, Letourneux G, Vivolo O. Differential fault analysis on AES // applied cryptography and network security. Berlin: Springer, 2003: 293–306