Persistence Wears Down Resistance: Persistent Fault Analysis on Unprotected and Protected Block Cipher Implementations

–Extended Abstract–

Shivam Bhasin[1], Jingyu Pan[1,2], and Fan Zhang[2]

[1] Temasek Laboratories, Nanyang Technological University, Singapore
sbhasin@ntu.edu.sg
[2] Zhejiang University, China
{joeypan,fanzhang}@zju.edu.cn

Abstract. This works gives an overview of persistent fault attacks on block ciphers, a recently introduced fault analysis technique based on persistent faults. The fault typically targets stored constant of cryptographic algorithms over several encryption calls with a single injection. The underlying analysis technique statistically recovers the secret key and is capable of defeating several popular countermeasures by design.

Keywords: Fault Attacks · Modular Redundancy · Persistent fault

1 Introduction

Fault attacks [1, 2] are active physical attacks that use external means to disturb normal operations of a target device leading to security vulnerability. These attacks have been widely used for key recovery from widely used standard cryptographic schemes, such as AES, RSA, ECC etc.

Several types of faults can be exploited to mount such attacks. Commonly known fault types are *transient* and *permanent*. A transient fault, which is most commonly used, generally affects only one instance of the target function call (eg. one encryption). On the other hand, a permanent fault, normally owing to device defects, affects all calls to the target function. Based on these two fault types, several analysis techniques have been developed. The most common are differential in nature, which require a correct and faulty computation with same inputs, to exploit the difference of final correct and faulty output pair for key recovery. Common examples of such techniques are differential fault analysis (DFA) [2], algebraic fault analysis (AFA) [4], etc. Some analyses are also based on statistical methods which can exploit faulty ciphertexts only like statistical fault analysis (SFA) [5] and fault sensitivity analysis (FSA) [6].

Recently, a new fault analysis technique was proposed [8] with a *persistent* fault model. Persistent fault lies between transient and permanent faults. Unlike transient fault, it affects several calls of the target function, however, persistent fault is not permanent, and disappears with a device reset/reboot. The corresponding analysis technique is called *Persistent Fault Analysis (PFA)* [8].

2 Persistent Fault Analysis (PFA)

PFA [8] is based on persistent fault model. In the following, the fault is assumed to alter a stored constant (like one or more entries in Sbox look-up) in the target algorithm, typically stored in a ROM. The attacker observes multiple ciphertext outputs with varying plaintext (not known). The modus operandi of PFA is explained with the following example. Let us take PRESENT cipher which uses a 4×4 Sbox i.e. 16 elements of 4-bits each, where each element has an equal expectation $\mathbb{E}$ of $\frac{1}{16}$. A persistent fault alters value of element x in Sbox to another element x', it makes $\mathbb{E}(x) = 0, \mathbb{E}(x') = \frac{2}{16}$, while all other elements still have the expectation $\frac{1}{16}$. The output ciphertext is still correct if faulty element x is never accessed during the encryption else the output is faulty. This difference can be statistically observed in the final ciphertext where some values will be missing (related to x) and some occuring more often than others (due to x'), which leaks information on the key k. This is illustrated in Fig. 1 (top) with $x = 10, x' = 8$. The key can be recovered even if x, x' are not known by simple brute-forcing. The strategy for key recovery can be one of the following:

1. t_{min}: find the missing value in Sbox table. Then $k = t_{min} \oplus x$;
2. $t \neq t_{min}$: find other values t where $t \neq t_{min}$ and eliminate candidates for k;
3. t_{max}: find the value with with maximal probability $k = t_{max} \oplus x'$.

The distribution of t_{min} or t_{max} can be statistically distinguished from the rest. The minimum number of ciphertexts N follows the classical coupon collector's problem [3] where it needs $N = (2^b - 1) \times (\sum_{i=1}^{(2^b-1)} \frac{1}{i})$, where b is the bit width of x. In PRESENT ($b = 4$) $N \approx 50$, as shown in Fig. 1 (bottom).

2.1 PFA *vs.* Other Fault Analysis

Here we list the key merits and demerits of PFA against other fault analysis.

Merits

- The main advantage of PFA is that it needs only one fault injection, which reduces the injection effort to minimum. Fault targets a constant in memory which persists over several following encryptions. This also reduces the injection effort in terms of timing precision within an injection. Moreover, live detection by sensors can be bypassed by injecting before the sensitive computation starts and sensors become active.
- The attack is statistical on ciphertexts only, and thus unlike differential attacks, needs no control over plaintexts.
- The fault model remains relaxed compared to other statistical attacks which may require multiple injections (one per encryption) with a known bias or additional side-channel information.
- Unlike any other known attacks, PFA can also be applied in the multiple fault (in a single encryption) setting.

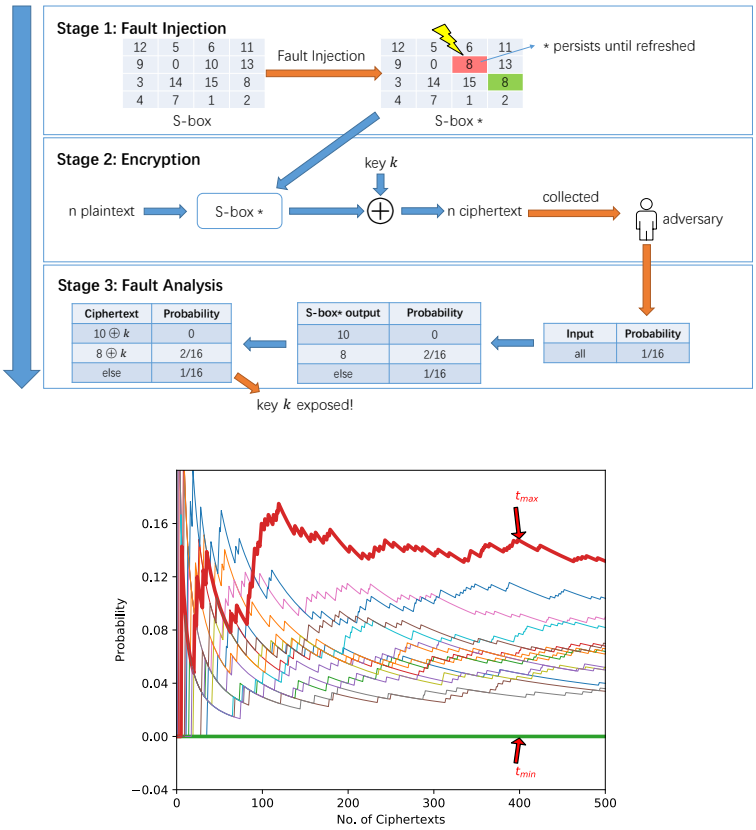

Fig. 1: Overview of Persistent Fault Attack (top), distribution of t_{min} and t_{max} against no. of ciphertexts for PRESENT leading to key recovery (bottom)

Demerits

- Being statistical in nature, it needs a higher number of ciphertexts as compared to DFA. Some known DFA can lead to key recovery with 1 or 2 correct/faulty ciphertext pair.
- Persistent faults can be detected by built-in self check mechanism.

2.2 Application of PFA on Countermeasures

PFA has natural properties which make several countermeasures vulnerable. The details on analysis of the countermeasure remain out of scope of this extended abstract due to limited space and interested readers are encouraged to refer [8]. Dual modular redundancy (DMR) is a popular fault countermeasure. The most common DMR proposes to compute twice and compare outputs. This countermeasure is naturally vulnerable to PFA if shared memories for constants are

used, which is often the case due to resource constraint environments. Other proposals use separate memories or compute forward followed by compute inverse and compare inputs. All these countermeasures output a correct ciphertext when no fault is injected. For a detected fault, the faulty output can be suppressed by no ciphertext output (NCO), zero value output (ZVO), or random ciphertext output (RCO) [8]. As PFA leaves certain probability for correct ciphertext output despite the persistent fault, it leads to key recovery using statistical method. However, more ciphertexts would be required in the analysis as some information is suppressed by the DMR countermeasure. Masking [7] on the other hand is a side channel countermeasure which is widely studied. As a persistent fault injects a bias in the underlying computation due to a biased constant, the bias can also affect the masking leading to key recovery.

3 Conclusion

Persistent fault analysis is a powerful attack technique which can make several cryptographic schemes vulnerable. With as low as one fault injection and simple statistical analysis on ciphertexts, PFA can perform key recovery. The introduced vulnerability also extends to protected implementations. We briefly discussed the impact of PFA on modular redundancy and masking based countermeasures. Existing countermeasures and other cryptographic schemes including public key cryptography must be analyzed to check their resistance against PFA. This further motivates research for dedicated countermeasures to prevent PFA.

References

1. Bar-El, H., Choukri, H., Naccache, D., Tunstall, M., Whelan, C.: The sorcerer's apprentice guide to fault attacks. Proceedings of the IEEE **94**(2), 370–382 (2006)
2. Biham, E., Shamir, A.: Differential cryptanalysis of the data encryption standard. Crystal Research & Technology **17**(1), 79–89 (2006)
3. Blom, G., Holst, L., Sandell, D.: Problems and Snapshots from the World of Probability. Springer Science & Business Media (2012)
4. Courtois, N.T., Jackson, K., Ware, D.: Fault-algebraic attacks on inner rounds of DES. In: e-Smart'10 Proceedings: The Future of Digital Security Technologies. Strategies Telecom and Multimedia (2010)
5. Fuhr, T., Jaulmes, E., Lomne, V., Thillard, A.: Fault attacks on AES with faulty ciphertexts only. In: The Workshop on Fault Diagnosis & Tolerance in Cryptography. pp. 108–118 (2013)
6. Li, Y., Sakiyama, K., Gomisawa, S., Fukunaga, T., Takahashi, J., Ohta, K.: Fault sensitivity analysis. In: CHES 2010, International Workshop, Santa Barbara, Ca, Usa, August 17-20, 2010. Proceedings. pp. 320–334 (2010)
7. Rivain, M., Prouff, E.: Provably secure higher-order masking of AES. In: CHES 2010. pp. 413–427 (2010)
8. Zhang, F., Lou, X., Zhao, X., Shivam, B., He, W., Ding, R., Qureshi, S., Ren, K.: Persistent Fault Analysis on Block Ciphers. In: IACR Transactions on Cryptographic Hardware and Embedded Systems. vol. 2018.3, pp. 150–172 (2018)