A Survey of the State-of-the-Art Fault Attacks

Jakub Breier and Dirmanto Jap
Physical Analysis and Cryptographic Engineering,
Temasek Laboratories@NTU
School of Physical and Mathematical Sciences, Division of Mathematical
Sciences, Nanyang Technological University, Singapore
jbreier@ntu.edu.sg, dirm0002@e.ntu.edu.sg

Abstract—Since 1996, when Boneh, DeMillo and Lipton introduced the idea of fault attacks, many theoretical and practical publications were made on this topic. These attacks belong to the class of physical cryptanalysis attacks.

In this paper we describe several methods of fault injection attacks. We provide an overview of both attacks and countermeasures on AES algorithm and on ECC.

Keywords—physical cryptanalysis, fault attacks, AES, ECC

I. INTRODUCTION

Cryptographic algorithms protect the confidentiality and the integrity of information. Each of widely used algorithms is proved to be secure from the view of the classical cryptanalysis, that means it is mathematically infeasible to decrypt a given ciphertext in a reasonable time without a key. Since late 90’s, there is increasing popularity of the physical cryptanalysis, which attacks the implementation of an algorithm instead of an algorithm itself. When attacking unprotected implementations, these attacks can be very effective.

The idea of fault attacks was introduced by Boneh, DeMillo and Lipton in 1996 [9]. These attacks exploit the possibility to insert a fault in the process of the algorithm execution in a way that could help to reveal the key. The first practical attack was implemented by Biham and Shamir [5], they implemented a successful attack on the DES algorithm, introducing a technique called the Differential Fault Attack. Since then, many types of fault attacks on different cryptosystems were proposed.

In this paper we describe the most recent techniques and methods of fault attacks. Fault injection techniques include supply voltage glitching and laser attacks as the most popular types, also the electromagnetic fault injection is getting more attention as it can be as accurate as a laser, but with the advantage of keeping the chip in its original package. Fault injection techniques for cryptographic algorithms include for example Differential Fault Analysis (DFA), Collision Fault Analysis (CFA), Ineffective Fault Analysis (IFA), Safe-Error Analysis (SEA) [13], [14].

The rest of the paper is organized as follows. Section II provides overview of attacks on symmetric cryptosystems, with emphasis on the Advanced Encryption Standard (AES) algorithm. Section III describes attacks on asymmetric algorithms, providing an overview of attacks on Elliptic Curve Cryptography (ECC) algorithms. Finally, section IV concludes this paper.

II. SECRET KEY CRYPTOSYSTEMS

This section describes fault attacks that have been proposed for the purpose of breaking symmetric algorithms. The most common attack method is the Differential Fault Analysis (DFA). The usual procedure is to invoke faults in a chosen round of the algorithm to get the desired fault propagation in the end of an encryption. The secret key can then be determined by examining the differences between a correct and a faulty ciphertext. The first attack using this technique was aimed at DES, it was proposed in 1997 by Biham and Shamir [5].

The next popular method is the Collision Fault Analysis (CFA), where an attacker invokes a fault in the beginning of the algorithm and then he tries to find a plaintext, which encrypts into the same ciphertext as the faulty ciphertext in the previous case, by using the same key.

The goal of the Ineffective Fault Analysis (IFA) method is to find such fault that does not change the intermediate result, therefore it leads into a correct ciphertext. The main problem of this method is to determine if the fault was actually invoked or not. Safe-Error Analysis (SEA) also exploits a situation when ciphertexts are equal, but it changes the intermediate result. It utilizes a state when the data is changed but it is not used.

In 2010, a method entitled Fault Sensitivity Analysis (FSA) [32] was proposed. This method is effective even for some DFA resistant implementations and does not restrict the fault model to a few bits or bytes. It exploits the side-channel information, such as sensitivity of a device to faults and uses this information to retrieve the secret key.

In 2012, a Linear Fault Analysis (LFA) [28] was proposed, which examines linear characteristics for some consecutive rounds of a block cipher. The authors successfully mounted the attack on the DFA resistant implementation of the SERPENT cipher.

A. Fault Attacks on AES

Since AES is the most popular symmetric block cipher, the majority of attacks aims on this algorithm. Table I summarizes the most important attacks on AES, in a chronological order.

B. Countermeasures

Along with attacks, works on countermeasures were usually presented as well. There are two main types of countermeasures against fault attacks [10]: sensor-based and error-detection based countermeasures. The first type checks the
TABLE I. FAULT ATTACKS ON AES

<table>
<thead>
<tr>
<th>Ref.</th>
<th>Year</th>
<th>Fault model</th>
<th># Faulty ciphertexts</th>
<th>Attack Type</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>[20]</td>
<td>2002</td>
<td>Switch 1 bit / disturb 1 byte</td>
<td>50/250</td>
<td>DFA</td>
<td></td>
</tr>
<tr>
<td>[8]</td>
<td>2002</td>
<td>Force 1 bit to 0</td>
<td>128</td>
<td>CFA/IFA</td>
<td>Practical attack on FPGA implementation shown in [26].</td>
</tr>
<tr>
<td>[30]</td>
<td>2003</td>
<td>Disturb 1 byte</td>
<td>2</td>
<td>DFA</td>
<td></td>
</tr>
<tr>
<td>[37]</td>
<td>2006</td>
<td>Disturb 1 byte</td>
<td>2^10</td>
<td>Square-DFA</td>
<td>Attacks middle rounds of the algorithm, therefore the redundancy countermeasure on the first or last rounds is ineffective.</td>
</tr>
<tr>
<td>[35]</td>
<td>2006</td>
<td>Disturb 1-4 bytes</td>
<td>6</td>
<td>DFA</td>
<td></td>
</tr>
<tr>
<td>[45]</td>
<td>2007</td>
<td>Disturb 1 column</td>
<td>2 and 4b brute-force search</td>
<td>DFA</td>
<td></td>
</tr>
<tr>
<td>[27]</td>
<td>2008</td>
<td>Disturb 3 bytes</td>
<td>2 and 32b brute-force search</td>
<td>DFA</td>
<td></td>
</tr>
<tr>
<td>[36]</td>
<td>2009</td>
<td>Disturb 1 byte</td>
<td>1 and 32b brute-force search</td>
<td>DFA</td>
<td></td>
</tr>
<tr>
<td>[40]</td>
<td>2009</td>
<td>Disturb 1-4 bytes</td>
<td>1 and 32b brute-force search</td>
<td>DFA</td>
<td></td>
</tr>
<tr>
<td>[32]</td>
<td>2010</td>
<td>Disturb 1-16 bytes</td>
<td>50</td>
<td>TSA</td>
<td>A new method - Fault Sensitivity Analysis was proposed, which does not use values of faulty ciphertexts.</td>
</tr>
<tr>
<td>[22]</td>
<td>2010</td>
<td>Disturb 1 diagonal</td>
<td>1 (AES-192), 3 (AES-256)</td>
<td>DFA</td>
<td>Extends the attack proposed in [38] to other keylengths.</td>
</tr>
<tr>
<td>[18]</td>
<td>2013</td>
<td>Disturb 1-12 bytes</td>
<td>Hundreds</td>
<td>CFA</td>
<td></td>
</tr>
<tr>
<td>[31]</td>
<td>2013</td>
<td>Disturb 1-16 bytes</td>
<td>1 and 8b brute-force search</td>
<td>CC-FSA</td>
<td>Introduces a Clockwise Collision FSA method.</td>
</tr>
<tr>
<td>[42]</td>
<td>2013</td>
<td>Disturb 1 byte</td>
<td>Square-DFA</td>
<td></td>
<td>The key can be revealed even with a large number of noisy fault injections.</td>
</tr>
</tbody>
</table>

Besides these two main types, there exist various methods, which can be either device or algorithm specific. We will briefly discuss the countermeasures on the AES presented so far.

Error-detection countermeasures can be found in [25], [24], [19]. Karpovsky et al. [25] proposed two countermeasures: one has a hardware overhead of 35%, but a low protection against small multiplicity faults, while the other one is robust, but with the overhead of 150%. Joye et al. [24] proposed a duplication scheme, with the 60% area overhead on the FPGA. It protects the implementation against the Giraud’s attack [20] using 1-4 faults, the resistance decreases rapidly with the higher number of faults. Genelle et al. [19] introduced a scheme that can be combined with the masking in order to provide a protection against the DPA as well. This protection scheme increases timing by 45% and the amount of used RAM by 448%.

The first countermeasure on the AES key schedule-based fault attack was presented by Chen and Yen [11]. They proposed three approaches for the round key protection: storing the key in the flash memory, generating the round key only once after the key update, and a parity check.

Mestiri et al. [34] used a novel scheme of protecting the S-box against faults. Instead of a normal output value from the S-box they xor the output and the input. This value is then xor-ed again with the input in order to get the correct output and the fault can be detected by using a flag, as shown in Figure 1. The other AES stages are protected by a parity check. They tested this scheme on an Virtex-5 FPGA, it can detect 99.998% of the random faults, however the area was increased by more than 1/4 and the frequency decreased by 17.03%.

Lomne et al. [33] analyzed several countermeasures and came to conclusions that most of them can be defeated by a slightly changed attack model or a multiple-fault model. They proposed enhanced countermeasures based on the randomness, that should minimize the probability of a successful attack.

Barenghi et al. [3] analyzed the efficiency of several software countermeasures based on redundant computations. Based on their findings, the instruction duplication or triplcation can provide a protection for AES against all known attacks. The overhead can be lowered by selecting only vulnerable parts of the algorithm to be duplicated or triplcated.

III. PUBLIC KEY CRYPTOSYSTEMS

The first and a well-known attack on public key cryptosystem was proposed by Boneh, DeMillo and Lipton [9], shortly improved by Lenstra [29]. The attack exploits the RSA-CRT implementation, enabling an efficient factorization of the modulus N with just a one pair of a faulty and a correct ciphertext. Since then, many fault attacks aiming at public key cryptosystems have been proposed.

A. Fault Attacks on ECC

Elliptic curve cryptosystems were proved to be vulnerable against several types of implementation attacks. The majority
of fault attacks attempt to move the computation from a secure curve to a weaker curve. There are several ways to perform this step, either by injecting faults into the curve parameters or the base point, or by attacking the scalar multiplication [1]. The first attack was proposed by Biehl et al. [4] in 2000, using the DFA technique. By disturbing 1 bit during the secret scalar multiplication, they were able to obtain the information about this scalar. This attack was further improved by Ciet and Joye [12], ECC are also vulnerable to SEA attacks, which are based on the assumption that the injected fault will change the output only if some condition of the secret data is fulfilled. Otherwise the output will stay unchanged. There are two types of these attacks: computational safe-error attacks (called C Safe-Error Attacks) [47], and M Safe-Error Attacks [46], [44]. The first type tries to induce a temporary random computational fault inside a register or a memory location.

Giraud and Knudsen [21] presented byte-fault attacks on multiple signature schemes, including the ECDSA. They extended the bit-fault model presented by Dottax in 2002 [16]. Using their method, they were able to recover the secret key with 2300 faulty signatures.

Blömer et al. [7] proposed a new method of the scalar multiplication based fault attack, called Sign Change Attacks. The main difference is that points do not leave the curve after the attack, therefore it makes the detection harder.

Schmidt and Medwed [43] suggested a new attack on ECDSA. The idea of the attack is to determine parts of the ephemeral key for several signatures. This key is different for each encryption, but it is possible to obtain the secret key by using the lattice attacks. They needed 50 faulty signatures to reveal a 160b key. Their countermeasure against this attack has an overhead of 36% in the worst case.

Sakamoto et al. [41] adjusted the Fault Sensitivity Analysis method, originally proposed by Li et al. [32] for the fault attack on the AES. They implemented an attack on the ECC implementation using the López-Dahab algorithm, which is less vulnerable to fault attacks than classical implementations.

Domínguez-Oviedo et al. [15] presented the invalid-curve attack that can be applied to the Montgomery ladder elliptic curve scalar multiplication algorithm.

Jarvinen et al. [23] extended Giraud’s attack on signature schemes [21]. They have shown that if faults are biased and the attacker can accurately estimate these biases, it can lead to a more efficient attack.

B. Countermeasures

There exist several types of countermeasures against fault attacks on ECC [17]. Point validation countermeasure verifies whether a given point lies on a curve or not. Curve integrity check can detect faults on curve parameters. Coherence check can verify the intermediate or final result with respect to a pattern. Combined curve check uses a reference curve for checking for fault occurrence and co-factor multiplication is used in order to prevent small subgroup attacks.

Baek and Vasyutsov [2] propose countermeasures against side-channel attacks and fault attacks, based on converting the definition field of elliptic curves into its random extension ring, while performing operations in the ring. It is possible to perform a validation check in a small subring, which provides a countermeasure against fault attacks.

IV. Conclusions

In this paper we provided an overview of the current state of fault attacks and their countermeasures on AES as a representative of symmetric algorithms and on ECC as a representative of public key cryptosystems.

It is worth to mention that some mathematical fault models presented in several publications can be difficult to implement in practice. For instance, bit-fault models require high-precision measuring and fault injection tools, which are costly and require an experienced operator.

REFERENCES


