Improved Fault Analysis of Signature Schemes

Christophe Giraud, Erik W. Knudsen, Michael Tunstall

Improved Fault Analysis of Signature Schemes

Christophe Giraud[1], Erik W. Knudsen[2], and Michael Tunstall[3]

[1] Oberthur Technologies,
4, allée du doyen Georges Brus, 33 600, Pessac, France.
c.giraud@oberthur.com
[2] Alm. Brand,
Midtermolen 7, 2100 København Ø, Denmark.
aberkn@almbrand.dk
[3] Department of Computer Science, University of Bristol,
Merchant Venturers Building, Woodland Road,
Bristol BS8 1UB, United Kingdom.
tunstall@cs.bris.ac.uk

Abstract. At ACISP 2004, Giraud and Knudsen presented the first fault analysis of DSA, ECDSA, XTR-DSA, Schnorr and ElGamal signatures schemes that considered faults affecting one byte. They showed that 2304 faulty signatures would be expected to reduce the number of possible keys to 2^{40}, allowing a 160-bit private key to be recovered. In this paper we show that Giraud and Knudsen's fault attack is much more efficient than originally claimed. We prove that 34.3% less faulty signatures are required to recover a private key using the same fault model. We also show that their original way of expressing the fault model under a system of equations can be improved. A more precise expression allows us to obtain another improvement of up to 47.1%, depending on the values of the key byte affected.

Keywords: Fault analysis, Signature schemes, Smart card.

1 Introduction

Since their introduction in 1996 by Boneh, DeMillo and Lipton [3], fault attacks have been widely studied from both practical and theoretical points of view. This type of attack poses a serious threat that needs to be considered when implementing cryptographic algorithms on embedded devices [2]. Once a suitable mechanism for injecting a fault is found, fault attacks would allow an attacker to break any cryptographic implementation faster than any other kind of attacks. The following examples speak for themselves: a DES secret key can be revealed by using two faulty ciphertexts [4, 10], an 128-bit AES secret key can be recovered by using one faulty ciphertext [16] and CRT RSA private parameters can be obtained by using one faulty signature [13, 11]. In the particular case of DSA [8], ECDSA [8], XTR-DSA [14], Schnorr [17] and ElGamal [7] signature schemes, fault attacks are not as efficient. Indeed, if an attacker is able to induce an error on a single bit, the most efficient attacks on such schemes can recover the

corresponding private keys by using one faulty signature per bit of the private key [1, 6]. However, injecting a fault that would only affect one bit would be very difficult to put into practice. Giraud and Knudsen extended previous attacks to use a more realistic model where faults where considered that would affect one byte [9]. By using this fault model, they claimed that one would be able to recover a 160-bit private key by using 2304 faulty signatures and then conducting an exhaustive search amongst 2^{40} possible keys.

In this paper, we analyse the attack proposed by Giraud and Knudsen [9] in more detail, and we show that their attack is much more efficient than originally claimed. Our improved analysis is two-fold. Firstly, under the same assumptions used in [9], we prove that the average number of trials required to conduct an exhaustive search is 2^{30} and not 2^{39} as implicitly indicated in [9]. Consequently, under the same attack model used in [9] we show that an attacker needs 34.3% less faulty signatures to recover a 160-bit private key. Secondly, we improve the detail of the system of equations representing the fault model. This more precise definition allows us to significantly improve the results given in [9], especially if the key is composed of bytes close to 0 or 255. In such a case, the improvement can be up to 47.1%.

The rest of this paper is organised as follows. In Section 2, we recall the previous fault attacks which have been published on signature schemes. In Section 3, we compute the average number of guesses to conduct an exhaustive search if performing a fault attack by using the fault model presented in [9]. In Section 4, we present a more accurate system of equations representing the fault model used in [9]. This allows us to improve the original result and to show that the efficiency of this attack is dependant on the key value. Finally, Section 5 investigates if there is an optimal method to perform the exhaustive search.

2 Previous Work

The first fault attack on signature schemes was proposed in [5] on the RSA using the Chinese Remainder Theorem (CRT). It was shown that a CRT RSA private key could be derived if one fault was injected during the generation of a signature. This attack requires an attacker to obtain two signatures for the same message M, where one signature is correct and the other one is faulty. By computing the gcd of the modulus N and the difference between the correct and the faulty signature, the attacker would obtain one private factor of the modulus. This attack was extended in [13, 11] by using only one signature generation.

A fault attack on RSA, where the CRT is not used, is defined in [1]. Again the attack compares the result of generating signatures from the same message with, and without faults. While a signature is being generated an attacker generates a fault causing one bit of the private exponent d to be changed, resulting in a faulty signature S'. Assuming that the i-th bit of d is complemented, then, as described in [1], we have:

$$\frac{S'}{S} \equiv M^{d'-d} \equiv \begin{cases} M^{2^i} \pmod N & \text{if the } i\text{-th bit of } d = 0, \\ \frac{1}{M^{2^i}} \pmod N & \text{if the } i\text{-th bit of } d = 1. \end{cases} \quad (1)$$

where S is the correct signature. In [1] it was also shown that this attack can be applied to other signature schemes, such as ElGamal, Schnorr and DSA.

In [12] it was shown that this attack can be improved by raising the formula to the power of the public exponent v, giving:

$$\frac{S'^v}{M} \equiv \begin{cases} (M^v)^{2^i} \pmod N & \text{if the } i\text{-th bit of } d = 0, \\ \frac{1}{(M^v)^{2^i}} \pmod N & \text{if the } i\text{-th bit of } d = 1. \end{cases} \tag{2}$$

In this case it is not necessary to know the correct signature S.

An attacker can compute $\frac{S'^v}{M} \bmod N$ for all the possible one-bit fault errors in d. This requires a total of $\log_2 d$ trials to completely derive d if an attacker is able to dictate which bit of d is complemented.

This was extended by Giraud and Knudsen in [9] to consider faults that affect one byte of an exponent. Let us now describe this extension where, to avoid any ambiguity, the fault needs to be treated as an integer difference between a correct exponent byte and a corrupted exponent byte. As above, we define the private key as d and d' as the corresponding corrupt private key. We define d_j and d'_j as the j^{th} byte of d and d' respectively, a fault on the i^{th} byte will therefore produce:

$$\begin{cases} d'_j = d_j, & \forall j \neq i \\ d'_j = d_j + e, \text{ if } j = i \end{cases} \tag{3}$$

with d_j, $d'_j \in \{0, \ldots, 255\}$ and $e \in \{-255, \ldots, 255\}$ where $e = 0$ corresponds to no error. For each fault on one byte of d, the attacker will deduce the position i of the corrupted byte since it is the only value for j which satisfies $d_j \neq d'_j$. He will then compute the corresponding value of e by subtracting d'_i with d_i.
By definition, the possible values for e are:

$$e \in \{-d_i, \ldots, 255 - d_i\} \ . \tag{4}$$

If two faults are observed, e and $\widehat{e}$, the possible values of d_i can be restricted to n values where:

$$e < \widehat{e} : \widehat{e} - e = 256 - n \, , \tag{5}$$

which means that:

$$-e \leq d_i \leq -e + (n-1) \ . \tag{6}$$

In [9] it is shown that the number of faulty signatures required to observe two faults with a significant enough difference to reduce d_i to a certain number of hypotheses can be defined. The average number of faults required to reduce the possible values to x values, for $x \in \{1, \ldots, 8\}$, are shown in Table 1.

In the next section, we present another way of analysing the efficiency of the attack described in [9]. Indeed, whatever the number of candidates left, the most important information is the average number of guesses the attacker needs to do to recover the value of the secret key.

Table 1. Expected number of faulty signatures required to reduce the number of possible values for one key byte [9].

Maximum[4] number of candidates left for d_i	Expected number of faulty signatures required
1	384.0
2	213.3
3	149.3
4	115.2
5	93.9
6	79.2
7	68.6
8	60.4

3 A More Practical Analysis

The analysis in [9] defines the average number of faulty signatures that are required to reduce the number of possible values for d_i to a given number of hypotheses. It is then suggested that the private key can be found using an exhaustive search. However, the expected number of guesses an attacker would have to conduct is not defined in [9]. This information is of uttermost importance when analysing the efficiency of the attack. For example, it is shown in [9] that a 2^{160} key space is reduced to at most 2^{40} by using 2304 faulty signatures. Therefore, one could assume that expected number of guesses in the exhaustive search is 2^{39}, but this is not the case as it will be seen in this section. In the following, we derive the expected number of guesses for a given number of induced faults under the same assumptions defined in [9].

3.1 Notation

After inducing t faults γ_i for $i \in \{1, \cdots, t\}$, we define the Min statistic by:

$$\min = \mathrm{Min}(\gamma_1, \ldots, \gamma_t) \ , \tag{7}$$

and the Max statistic by:

$$\max = \mathrm{Max}(\gamma_1, \ldots, \gamma_t) \ . \tag{8}$$

3.2 Expected Search Time

By denoting by N the number of possible values for the variable we disturb (in our case $N = 256$ since we disturb one byte), we summarise the attack described in [9] as follows:

[4] In [9], it is written "Average number of candidates left" but their formula gives the average number of faulty signatures required to reduced the number of candidates to *at most* n. Therefore we need to replace "Average" by "Maximum".

> "Induce t faults on a variable $K \in \{0, \cdots, N-1\}$ which results in observed values min and max with difference $f = \max - \min$. Then guess between the $N - f$ remaining candidates at random."

Theorem 1 gives the probability of guessing the correct key in the g^{th} guess using this algorithm:

Theorem 1. *Let a variable $K \in \{0, \cdots, N-1\}$ with value a be given. The probability of the number of guesses X being equal to g using the ACISP Algorithm with t faulty signatures uniformly distributed is given by:*

$$\begin{aligned} \Pr(X = g | K = a) &= \Pr(max = a - g + 1 | K = a) \\ &= \left(\tfrac{N-g+1}{N}\right)^t - \left(\tfrac{N-g}{N}\right)^t \end{aligned} \tag{9}$$

Using Theorem 1, we can compute the expected waiting time as:

Corollary 1. *The expected size of an exhaustive search by using the algorithm described in [9] is:*

$$\sum_{s=1}^{N} \left(\frac{s}{N}\right)^t \quad .$$

The proofs of Theorem 1 and Corollary 1 are given in Appendix A[5].

By using Corollary 1 with $N = 256$, we can compute the expected number of faulty signatures to reduce the number of guesses to x, for $x \in \{2, \ldots, 8\}$, as shown in Table 2.

Table 2. Expected number of faulty signatures required to reduce the expected number of guesses to x, for $x \in \{2, \ldots, 8\}$.

Expected number of guesses for d_i	Number of faulty signatures performed
2	176.5
3	102.8
4	72.7
5	56.2
6	45.7
7	38.5
8	33.2

We can, therefore, demonstrate using Corollary 1 that using 2304 faulty signatures to attack a 160-bit key (i.e. 115.2 faulty signatures per key byte) implies that the expected number of guesses required to conduct an exhaustive search is $\left(\sum_{s=1}^{256} \left(\frac{s}{256}\right)^{115.2}\right)^{20} \approx 2^{30}$ and not 2^{39} as implicitly indicated in [9].

[5] In Appendix A.1, we present a simpler proof of [9, Theorem 1] than the one in the aforementioned article.

Moreover, under the same attack model as in [9] (i.e. an attacker being able to inject faults on one byte and being able to perform 2^{39} signatures during the exhaustive search to find the correct key), we compute by using Corollary 1 that only 1513.3 faulty signatures are required[6] to recover a 160-bit private key. It is therefore an improvement of 34.3% compared to what was announced in [9].

The analysis presented in [9] and in this section are based on Relations (4) and (6). However, we show in the next section that these Relations can be refined, leading to an improvement of the corresponding analyses.

4 Improving the Analysis

In this section we will discuss some improvements to the analysis given in [9]. This is based on a more precise definition of the fault model and subsequent interpretation.

4.1 Reducing the number of possible values for the error

In this section, we present two remarks that allow us to improve the number of faulty signatures required.

Firstly, as described above, we have:

$$d_i + e = d'_i \tag{10}$$

with d_i, $d'_i \in \{0, \ldots, 255\}$ and $e \in \{-255, \ldots, 255\}$. In [9], Relation (10) leads them to the following relation:

$$-e \leq d_i \leq 255 - e \tag{11}$$

However, we can reduce this interval of possible values for d_i by taking into account that $d_i \in \{0, \ldots, 255\}$. This leads us to the following relation:

$$\mathrm{Max}(0, -e) \leq d_i \leq \mathrm{Min}(255, 255 - e)\,, \tag{12}$$

One may note that this restriction only significantly improves the analysis when the key byte value is close to 0 or to 255.

Another way to slightly improve the analysis of [9] is to notice that they take into account the case $e = 0$ when computing their probabilities. As this case corresponds to not inflicting any error, the resulting signature will be correct and can then be excluded. We can, therefore, define the error e as:

$$e \in \{-d_i, \ldots, 255 - d_i\} \setminus \{0\}\,, \tag{13}$$

which further allows us to reduce the number of faulty signatures required.

In the next section, we redo the analysis done in [9] and in Section (3) by using Relations (12) and (13) instead of Relations (6) and (4) respectively.

[6] To obtain $\left(\sum_{s=1}^{256} \left(\frac{s}{256}\right)^t\right)^{20} \approx 2^{39}$, we need to use $t = 75.66$ faulty signatures per byte.

4.2 The Expected Number of Faults Required

To compute the statistically expected number of faults that will need to be observed we can model the system using a series of geometric distributions (see Appendix B). In this section we assume that an attacker wants to reduce the number of possible hypotheses for a given byte to less than α. That is, the results of this analysis can be directly compared to those presented in [9].

The simplest case will be where the key byte is equal to zero or 255. Assuming that the effect of a fault is uniformly distributed, the probability of a given event allowing this is $\alpha/255$. So, we can say that attacker would expect to reduce the hypotheses to less than α after $255/\alpha$ observed faults.

This is no longer the case if we consider other key values as observations can occur that are improved upon with subsequent observations. An attacker will be interested in observing a fault such that $e \in \{-k, \ldots, -k+\alpha\}$ or $e \in \{255-k-\alpha, \ldots, 255-k\}$, for some key byte k. This is because these values can be combined to make a list of less than, or equal to, α hypotheses. For any event an attacker would expect to observe a value form these groups with a probability of $\frac{2\,\alpha}{255}$, and therefore would expect this to occur after $\frac{255}{2\,\alpha}$ observed faults.

For each of the $2\,\alpha$ possible faults there will be a certain number of faults that will be of interest to an attacker. For each of the possible faults the number of faults that improve the amount of information can be noted. The probability of observing a further fault of interest can be computed and another expectation derived from this. This can be continued for each possible combination of observations for a given α and the average expected number of observations can be computed.

For key values close to zero or 255 the initial two groups $\{-k, \ldots, -k+\alpha\}$ or $\{255-k-\alpha, \ldots, 255-k\}$ will be modified since an attacker will know the value when no fault is applied, i.e. $e = 0$. This means that the average expectation across all the possible key values will lead to the expected number of faults.

If, for example, we consider the case where $\alpha = 2$. For simplicity, we will also assume that the key byte value is $\in \{2, \ldots, 253\}$, i.e. the value of the key byte will not interfere with the analysis. Using the above method the expected number of observations X_1 is computed as

$$\begin{aligned} E(X_1) &= \frac{255}{4} + \frac{1}{4} \cdot \frac{255}{2} + \frac{1}{4}\left(\frac{255}{3} + \frac{2}{3} \cdot \frac{255}{2}\right) + \frac{1}{4} \cdot \frac{255}{2} + \frac{1}{4}\left(\frac{255}{3} + \frac{2}{3} \cdot \frac{255}{2}\right) \\ &= 212.5\,, \end{aligned}$$

where E is a function that returns the statistical expectation.

If the key byte value is $\in \{1, 254\}$, then the known value for no fault being injected will be part of the process. Using the above method the expected number of observations X_2 is computed as

$$\begin{aligned} E(X_2) &= \frac{255}{3} + \frac{1}{3} \cdot \frac{255}{2} + \frac{1}{3} \cdot \frac{255}{2} \\ &= 170\ . \end{aligned}$$

If the key byte value is $\in \{0, 255\}$, then we only need one observation of interest to reduce the number of possible values to the required amount. Using the above method the expected number of observations X_3 is computed as

$$E(X_3) = \frac{255}{2} = 127.5 \ .$$

The expectation for a uniformly distributed key $\in \{0, \ldots, 255\}$ is a weighted mean of the above expectations. The expected number of faults required X therefore becomes

$$E(X) = \frac{252\,E(X_1) + 2\,E(X_2) + 2\,E(X_3)}{256} = 211.5 \ .$$

The results of computing the above for $\alpha \in \{1, \ldots, 8\}$ is shown in Table 3 (all rounded to one decimal place).

Table 3. Expected number of guesses for one key byte depending on the number of faulty signatures.

Maximum number of hypotheses for d_i	Expected number of faulty signatures
1	381.5
2	211.5
3	147.8
4	113.8
5	92.5
6	77.9
7	67.3
8	59.2

One can note that the improvement in Table 3 is minimal compared to Table 1 (1.3% on average). The improvements discussed in Section 4.1 only have a significant impact when a the value of a key byte is close to 0 or to 255. Figure 1 shows the key value dependency of this improvement when using 114 faulty signatures. In such a case, the improvement is of 39.2% if the key byte is equal to 0 or 255. Moreover, the smallest the number of faulty signatures we use, the biggest the improvement is. For instance, we obtain an improvement of 49.3% when using 10 faulty signatures if the key byte is equal to 0 or 255.

4.3 The Expected Size of an Exhaustive Search

The analysis described in the previous section is limited in the same way as the analysis described in [9]. The number of faults required is that needed to reduce the number of faults to less than a given value. In this section we define the expected number of faults required to produce an exhaustive search of a given size.

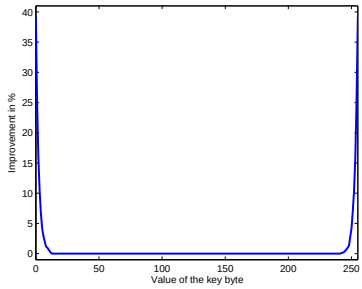

Fig. 1.a. Global view

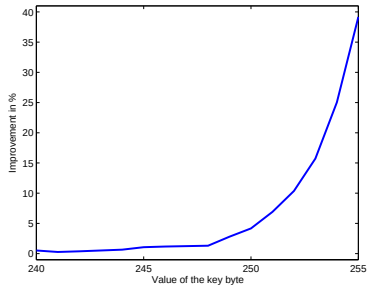

Fig. 1.b. Zoom on Fig. 1.a

Fig. 1. Improvement of the analysis on the number of remaining candidates depending of the key byte value when using 114 faulty signatures.

We first define the probability that a set of t observations will span X different values, for $X \in \{1, \ldots, N-1\}$ (in our case $N = 256$). Trivially, we can say that

$$\Pr(X = 1) = \frac{N-1}{(N-1)^t} \tag{14}$$

since the probability of all t observations having the same value is $\frac{1}{(N-1)^t}$ for each possible value of X.

For $X = 2$,

$$\Pr(X = 2) = \frac{2\,S(t,2)\,(N-2)}{(N-1)^t} \tag{15}$$

where we define $S(n, i)$ as a function that returns the Stirling numbers of the second kind. That is, the number of ways of partitioning n elements into i non-empty sets. Given that our sets are unique this needs to be multiplied by 2. Furthermore, there are $(N-2)$ possible sets where $X = 2$.

Computing the expectation becomes more complex for $X > 2$ since the sets representing the minimum and maximum possible values need to be non-empty, but the values in between can be empty. In order to compute the number of possible combinations that will span x different values we consider that the values fall into two sets. The first set represents the minimum and maximum values (there will be a minimum of two observations in this set). The second set represents the values in between the minimum and maximum observation and can contain empty sets.

We consider t faults that span x different values, where $t > 2$ and $x > 2$. These faults will be distributed between the two sets defined above. The set representing the minimum and maximum values contains i observations, where i is $\in \{2, \ldots, t\}$, then the number of combinations that are possible for a given i will be $2\,S(i, 2)$, as described above. The remaining observations will be in the other set where the number of combinations will be $(x-2)^{t-i}$. We can note that this number of combinations will only take into account one particular set of i

observations from t faults. This means that there will be $\binom{t}{i}$ possible ways of choosing i observations. This leads to

$$\Pr(X = x \,|\, x > 2) = \frac{(N-x)\sum_{i=2}^{t} 2\,S(i,2)\,(x-2)^{t-i}\binom{t}{i}}{(N-1)^t}\,, \tag{16}$$

where we note that there are $(N-x)$ possible combinations where $X = x$.

The list of key hypotheses that can be excluded will also depend on the value of the byte affected by the fault being injected. The probability of a set of observations having a particular difference between the maximum and minimum observation x for a specific key value k is

$$\Pr(X = x \,|\, k = z) = \frac{\Pr(X = x)}{256}\,, \tag{17}$$

where we assume that the z is $\in \{0, \ldots, 255\}$, i.e. 256 possible values.

However, the expectation cannot be computed directly from this probability. This is because the signature where no fault is injected is known, i.e. an observation where the fault induced is equal to zero. For a minimum observation M_1 and a maximum observation M_2, spanning x values, we can define:

$$\begin{aligned} &\mathrm{Min}(M_1, 0) \\ &\mathrm{Max}(0, M_2) \end{aligned}$$

For a given M_1 and M_2 the number of values that are actually spanned will be equal to $\mathrm{Max}(0, M_2) - \mathrm{Min}(M_1, 0)$, and the number of remaining hypotheses is equal to $N-1-\mathrm{Max}(0, M_2)+\mathrm{Min}(M_1, 0)$. Therefore, the probability given above for $\Pr(X = x \,|\, k = z)$ needs to be divided by $N - x$ to produce the probability for specific values of M_1 and M_2.

The expected number of remaining hypotheses $E(Y)$ can be computed as

$$E(Y) = \sum_{z=0}^{N-1} \sum_{x=1}^{N-1} \sum_{\substack{m \,\in\, \Phi \\ m+x-1 \,\in\, \Phi}} (N-\mathrm{Max}(0, m+x-1)+\mathrm{Min}(m, 0)) \frac{\Pr(X = x \,|\, k = z)}{N-x}\,,$$

where we use the notation defined above (cf. (14), (15), (16) and (17)). In addition we denote m as the minimum observation in a set of observations that span x values. The values that m and $m + x - 1$ can take are in Φ, where Φ is the set $\{-z, \ldots, N - z - x\} \setminus \{0\}$.

If we assume the actual key value is uniformly distributed between the remaining hypotheses h then the expected number of tests required to find a key byte is computed as $(h + 1)/2$. For a fault in one by these values are shown in Table 4, rounded to one decimal place.

Table 4. Expected number of hypotheses and guesses required to determine a given key byte.

t	E(# of Remaining Hypotheses)	E(# of Tests)
1	170.3	85.7
2	127.7	64.3
3	102.1	51.6
4	85.1	43.0
5	72.9	37.0
6	63.7	32.4
7	56.7	28.8
8	51.0	26.0
9	46.4	23.7
10	42.5	21.8

In order to allow the results of the improvement to be directly compared to the results described in Section 3, we used the formula for $E(Y)$ to compute the expected number of faulty signatures required to reduce the number of guesses between 2 and 8, cf. Table 5.

Table 5. Expected number of faulty signatures required to reduce the expected number of guesses to x, for $x \in \{1, \ldots, 8\}$.

Expected number of guesses for d_i	Number of faulty signatures required
2	174.8
3	101.4
4	71.4
5	55.0
6	44.5
7	37.4
8	32.1

By observing Table 5, one can see that the improvement is minimal compared to Table 2 (2.1% on average). It only has a significant impact when the value of a key byte is close to 0 or 255. Figure 2 shows this improvement when using 73 faulty signatures. In such a case, the improvement is of 37.6% if the key byte is equal to 0 or 255. Moreover, the smallest the number of faulty signatures we use, the biggest the improvement is. For instance, we obtain an improvement of 48.1% when using 10 faulty signatures.

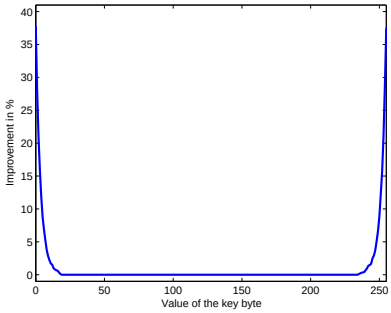

Fig. 2.a. Global view

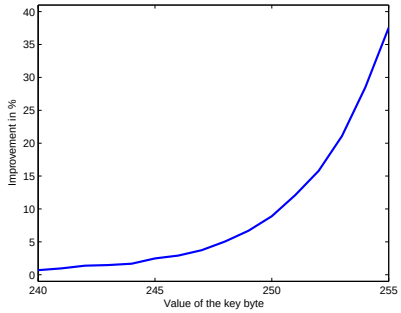

Fig. 2.b. Zoom on Fig. 2.a

Fig. 2. Improvement of the analysis on the number of guesses depending of the key byte value when using 73 faulty signatures.

The analysis conducted in this section shows that using the same attack model used in [9][7] and considering a random 160-bit key (resp. a 160-bit key that consists of bytes equal to 0 or 255), we would need 1488[8] (resp. 800[9]) faulty ciphertexts to recover the entire key. In the latter case, we have an improvement of 47.1% comparing to the result presented in Section 3.

5 The Exhaustive Search

In [9], no detail is given on how to obtain the correct value of the key amongst the possible values that are defined by the analysis. If, for example, there are 4^{20} (2^{40}) possible values for a private key, we can say that one would expect to conduct 2^{39} trials, on average, to identify the correct value amongst the 4^{20} possible values. In this section we demonstrate that there is no best method of searching through the hypotheses that are produced by the analyses described in this paper.

We define a series of t faults γ_i for $i \in \{1, \ldots, t\}$, which we assume are drawn from a distribution that is uniform over the integers in $\{\eta, \ldots, 255 - \eta\} \setminus \{0\}$, where η is an unknown integer $\in \{0, \ldots, 255\}$. In order to rank a given value of η, we divide the interval into s intervals. For simplicity, we assume that the intervals are evenly spaced. Let x_i denote the number of the observed γ which fall within the i-th interval, so we have

$$x_1 + \cdots + x_s = t \ . \tag{18}$$

[7] That is, by using byte-fault model and a expected number guesses of 2^{39}.

[8] To obtain $E(Y)^{20} \approx 2^{39}$ with $N = 256$, we need to use $t = 74.4$ faulty signatures per key byte.

[9] To obtain $E(Y)^{20} \approx 2^{39}$ with $N = 256$, $z \in \{0, 255\}$ and by dividing $\Pr(X = x)$ by 2 instead of 256 to obtain $\Pr(X = x \,|\, k = z)$, we need to use $t = 40$ faulty signatures per key byte.

Given that the faults induced are uniformly distributed, then

$$E(x_1) = \cdots = E(x_s) = \frac{t}{s} \ , \tag{19}$$

where E is a function that returns the statistical expectation.

This can be used to conduct a frequency test. That is, we compute a statistic λ, where

$$\lambda = \sum_{i=1}^{s} \frac{(O(x_i) - E(x_i))^2}{E(x_i)} \ , \tag{20}$$

and O is a function that returns the number of observed faults in a particular interval. Then $\lambda \sim \chi^2(s-1)$ and can be used to test the null hypothesis that η is a given value for each possible value $\in \{0, \ldots, 255\}$. For a given λ the P-value can be computed which is a value between 0 and 1, where the larger the value the more evidence is provided against a hypothesis for η (typically values above 0.95 or 0.99 are used as a boundary, above which it can be stated that there is no evidence to support a given null hypothesis).

A set of 32 simulated observations were generated where η was, arbitrarily, set to 140. The above statistical test was then conducted for $s \in \{3, 5, 15\}$, i.e. small divisors of 255. In Figure 3 we show the resulting P-values for all the possible hypotheses for η. The dashed lines represent the boundaries that can be deduced by observing the maximum and minimum possible values (see Equation (12)).

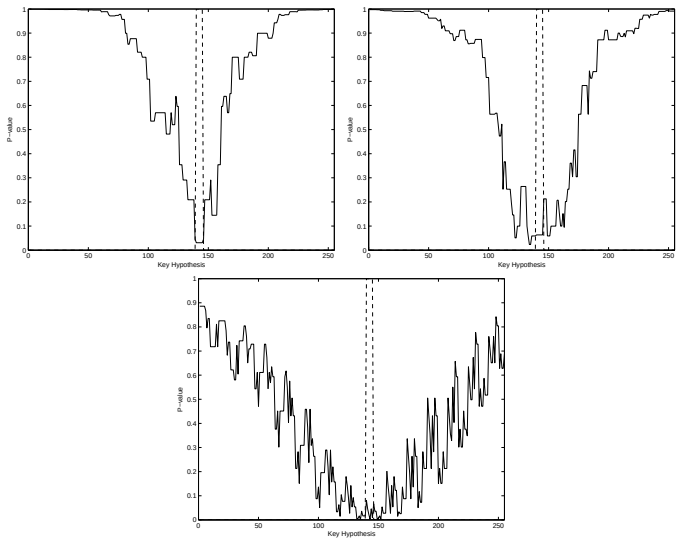

Fig. 3. Returned P-values plotted for all possible values for one byte for $s = 3$, 5, 15 (top left, top right and bottom middle respectively). The dashed lines represent the boundaries identified by the minimum and maximum observed fault.

One could assume that the P-value between the two lines could be used to determine which values of η are most likely and use this to optimise the exhaustive search required once the possible values for a private key have been reduced to a certain amount. However, this is not the case.

The test described above where the number of faults was set to values $\in \{1, \ldots, 64\}$, where 10 000 tests were conducted for each value. In each case the distribution of the number of tests required in an exhaustive search was identical to that produced by simply starting with the least (or most) significant hypothesis and incrementing (or decrementing) the hypothesis for each test.

This is because the number of observed faults is not enough to give a good indication of the distribution to which they belong. If enough faults were observed to give a strong indication of the correct value of η, it would be expected that the minimum and maximum observation would also identify η without requiring any further computation.

6 Conclusion

In this paper, we show that the fault attack described by Giraud and Knudsen at ACISP 2004 is much more efficient than originally claimed. We proved that, using the same attack model, we need 34.3% less faulty signatures to recover a 160-bit private key. Furthermore, by improving the fault model expression, we show that for some key values we obtain another improvement of up to 47.1%. Finally, we show that there is no optimal way of performing the exhaustive search in order to reduce the computation complexity of this step.

In summary, Giraud and Knudsen claim that they need 2304 faulty signatures to recover a 160-bit private key, but in this paper we prove that one would only need between 800 and 1488 faulty signatures to recover such a key, using the same model and expected size of the required exhaustive search.

Acknowledgments

The authors would like to thank Frédéric Amiel for initiating this work and Emmanuel Prouff for his helpful comments on the preliminary version of this paper. The work described in this paper has been supported in part by the European Commission IST Programme under Contract IST-2002-507932 ECRYPT and EPSRC grant EP/F039638/1.

References

1. F. Bao, R. Deng, Y. Han, A. Jeng, A. D. Narasimhalu, and T.-H. Ngair. Breaking Public Key Cryptosystems an Tamper Resistance Devices in the Presence of Transient Fault. In *5th Security Protocols Workshop*, volume 1361 of *LNCS*, pages 115–124. Springer, 1997.
2. H. Bar-El, H. Choukri, D. Naccache, M. Tunstall, and C. Whelan. The Sorcerer's Apprentice Guide to Fault Attacks. *IEEE*, 94(2):370–382, 2006.

3. Bellcore. New Threat Model Breaks Crypto Codes. Press Release, Sept. 1996.
4. E. Biham and A. Shamir. Differential Fault Analysis of Secret Key Cryptosystem. In B. Kalisky Jr., editor, *Advances in Cryptology – CRYPTO '97*, volume 1294 of *LNCS*, pages 513–525. Springer, 1997.
5. D. Boneh, R. DeMillo, and R. Lipton. On the Importance of Checking Cryptographic Protocols for Faults. In W. Fumy, editor, *Advances in Cryptology – EUROCRYPT '97*, volume 1233 of *LNCS*, pages 37–51. Springer, 1997.
6. E. Dottax. Fault Attacks on NESSIE Signature and Identification Schemes. Technical report, NESSIE, Oct. 2002.
7. T. ElGamal. A Public-Key Cryptosystems and a Signature Scheme based on Discret Logarithms. *IEEE Transaction on Information Theory*, 31(4):469–172, 1985.
8. FIPS PUB 186-3. *Digital Signature Standard*. National Institute of Standards and Technology, Mar. 2006. Draft.
9. C. Giraud and E. Knudsen. Fault Attacks on Signature Schemes. In H. Wang, J. Pieprzyk, and V. Varadharajan, editors, *Information Security and Privacy - 9th Australasian Conference – ACISP 2004*, volume 3108 of *LNCS*, pages 478–491. Springer, 2004.
10. C. Giraud and H. Thiebeauld. A Survey on Fault Attacks. In J.-J. Quisquater, P. Paradinas, Y. Deswarte, and A. E. Kalam, editors, *Smart Card Research and Advanced Applications VI – CARDIS 2004*, pages 159–176. Kluwer Academic Publishers, 2004.
11. M. Joye, A. Lenstra, and J.-J. Quisquater. Chinese Remaindering Based Cryptosystems in the Presence of Faults. *Journal of Cryptology*, 12(4):241–245, 1999.
12. M. Joye, J.-J. Quisquater, F. Bao, and R. Deng. RSA-type Signatures in the Presence of Transient Faults. In M. Darnell, editor, *Cryptography and Coding*, volume 1355 of *LNCS*, pages 155–160. Springer, 1997.
13. A. Lenstra. Memo on RSA Signature Generation in the Presence of Faults. Manuscript, 1996.
14. A. Lenstra and E. Verheul. An Overview of the XTR Public Key System. In K. Alster, J. Urbanowicz, and H. Williams, editors, *Public Key Cryptography and Computational Number Theory*, pages 151–180. de Gruyter, 2000.
15. D. Naccache, P. Nguyen, M. Tunstall, and C. Whelan. Experimenting with Faults, Lattices and the DSA. In S. Vaudenay, editor, *Public Key Cryptography – PKC 2005*, volume 3386 of *LNCS*, pages 16–28. Springer, 2005.
16. D. Saha, D. Mukhopadhyay, and D. RoyChowdhury. A Diagonal Fault Attack on the Advanced Encryption Standard. Cryptology ePrint Archive, Report 2009/581, 2009. `http://eprint.iacr.org/`.
17. C. Schnorr. Efficient Identification and Signatures for Smart Cards. In G. Brassard, editor, *Advances in Cryptology – CRYPTO '89*, volume 435 of *LNCS*, pages 239–251. Springer, 1989.

A Proof of Theorem 1 and Corollary 1

A.1 Distributions

Theorem 2. *Let a random variable X be given, which is uniformly discrete on the interval* 1 *to* N*:*

$$\forall 1 \leq x \leq N : P(X \leq x) = \frac{x}{N}$$

and define the function G as:

$$\forall 1 \leq x \leq N : G(x) := P(X_1, \cdots, X_t \leq x) = P(X \leq x)^t = \left(\frac{x}{N}\right)^t.$$

The following hold:

i. For $a - (N-1) \leq m \leq a$:

$$P(max = m|K = a) = \left(\frac{N + m - a}{N}\right)^t + \left(\frac{N - 1 + m - a}{N}\right)^t$$
$$P(min = m|K = a) = \left(\frac{a + 1 - m}{N}\right)^t + \left(\frac{a - m}{N}\right)^t$$

ii. For $d \geq 2$ and $a - (N-1) \leq m \leq a - d$:

$$\begin{aligned} P(min = m, max = m + d|K = a) = & P(max = a - (N-1) + d|K = a) \\ & - P(max = a - N + d|K = a) \end{aligned}$$

iii. For $d \geq 2$:

$$P(max - min = d) = (N - d)(P(max = d + 1) - P(max = d))$$
$$P(d \leq max - min) = 1 - (N - d + 1)G(d) + (N - d)G(d - 1)$$

The latter result is the same as the result from [9]:

$$P(T_n \leq t) = P(d \leq max - min) = 1 - (N - d + 1)G(d) + (N - d)G(d - 1)$$

Proof (Theorem 2.i).

$$\begin{aligned} P(max \leq a) &= P(X_1, \cdots, X_t \leq a) \\ &= \prod_{i=1}^{t} P(X_i \leq a) \\ &= P(X \leq x)^t \\ &= G(a) \\ P(min \leq a) &= 1 - P(min > a) \\ &= 1 - P(a + 1 \leq X_1, \cdots, X_t) \\ &= 1 - G(N - a) \\ P(max = a) &= P(max \leq a) - P(max \leq a - 1) \\ &= G(a) - G(a - 1) \\ P(min = a) &= P(min \leq a) - P(min \leq a - 1) \\ &= G(N - a + 1) - G(N - a) \end{aligned}$$

□

Proof (Theorem 2.ii). For $d \geq 0$ and $1 \leq a \leq N - d$:

$$\begin{aligned} P(min \geq a, max \leq a + d) &= P(a \leq X_1, \cdots, X_t \leq a + d) \\ &= P(1 \leq X_1, \cdots, X_t \leq d + 1) \\ &= G(d + 1) \end{aligned}$$

For $d \geq 2$ and $1 \leq a \leq N - d$:

$$\begin{aligned} P(min = a, max = a + d) &= P(a \leq X_1, \cdots, X_t \leq a + d) \\ &\quad + P(a + 1 \leq X_1, \cdots, X_t \leq a + d - 1) \\ &\quad - P(a \leq X_1, \cdots, X_t \leq a + d - 1) \\ &\quad - P(a + 1 \leq X_1, \cdots, X_t \leq a + d) \\ &= G(d + 1) - G(d) - (G(d) - G(d - 1)) \\ &= P(max = d + 1) - P(max = d) \end{aligned}$$

□

Proof (Theorem 2.iii).

$$\begin{aligned} P(max - min = d) &= \sum_{a=1}^{N-d} P(min = a, max = d + a) \\ &= \sum_{a=1}^{N-d} (P(min = d + 1) - P(max = d)) \\ &= (N - d)(P(max = d + 1) - P(max = d)) \end{aligned}$$

$$\begin{aligned} P(d \leq max - min) &= P(d \leq max - min \leq N - 1) = \sum_{a=d}^{N-1} P(max - min = a) \\ &= \sum_{a=d}^{N-1} (N - a)(P(max = a + 1) - P(max = a)) \\ &= \sum_{a=d+1}^{N} (N - a + 1)P(max = a) - \sum_{a=d}^{N-1} (N - a)P(max = a) \\ &= \sum_{a=d+1}^{N-1} P(max = a) + P(max = N) - (N - d)P(max = d) \\ &= P(d + 1 \leq max) - (N - d)P(max = d) \\ &= 1 - G(d) - (N - d)(G(d) - G(d - 1)) \\ &= 1 - (N - d + 1)G(d) + (N - d)G(d - 1) \end{aligned}$$

If we substitute $N = 256$ and $d = N - n = 256 - n$ we get the result from [9, Theorem 1]:

$$\begin{aligned} P(T_n \leq t) &= P(256 - n \leq max - min \leq 255) \\ &= 1 - (n + 1)G(256 - n) + nG(255 - n) \\ &= 1 - (n + 1)\left(\frac{256 - n}{256}\right)^t + n\left(\frac{255 - n}{256}\right)^t \end{aligned}$$

but the derivation here is much simpler than the one in the aforementioned article. □

A.2 Generic Formula for Making the Correct Guess

Lemma 1. *With the notation:*

$$\begin{aligned} A &= \text{"}NumGuess = n\text{"} \\ B_{m,d} &= \text{"}min = m, max = m + d\text{"} \\ C &= \text{"}K = a\text{"} \end{aligned}$$

The probability of guessing the correct key in the g^{th} guess can be expressed as

$$\begin{aligned} P(NumbGuess = g|K = a) = &\textstyle\sum_{d=0}^{N-g}(P(max = d) - P(max = d-1)) \\ &\times \textstyle\sum_{m=a-(N-1)}^{a-d} P(A|B_{m,d} \cap C) \end{aligned}$$

Proof. Notice that

$$P(A \cap B|C) = \frac{P(A \cap B \cap C)}{P(C)} = \frac{P(A|B \cap C)P(B \cap C)}{P(C)} = P(A|B \cap C)P(B|C)$$

and

$$P(A|C) = P(\cup_{m,d}(A \cap B_{m,d})|C) = \sum_{m,d} P((A \cap B_{m,d})|C) = \sum_{m,d} P(A|B_{m,d} \cap C)P(B_{m,d}|C)$$

Now we can write:

$$P(NumbGuess = g|K = a) = \sum_{d=0}^{N-g} \sum_{m=a-(N-1)}^{a-d} P(A|B_{m,d} \cap C)P(B_{m,d}|C)$$

and we know from Theorem 2.ii that

$$P(B_{m,d}|C) = P(max = d) - P(max = d - 1)$$

Insertion concludes the proof. □

A.3 Proofs of Theorem 1 and Corollary 1

Proof (Theorem 1). Having observed $min = m$ and $max = m + d$, [9] is content with noting that the number of remaining candidates equals $N - d$ and therefore

$$P(A|B_{m,d} \cap C) = \frac{1}{N - d}$$

The claim follows by insertion in the formula of Lemma 1. □

Proof (Corollary 1). The expected is given by the formula

$$\sum_{s=1}^{N} sP(NumbGuess = g|K = a) = \sum_{s=1}^{N} s \quad \left(\frac{N - s + 1}{N}\right)^t - \left(\frac{N - s}{N}\right)^t \Bigg)$$

which reduces to the claimed formula. □

B The Geometric Distribution

A geometric distribution is produced when we consider a system

$$\Pr(X = x) = (1-p)^{x-1}\, p$$

That is, a geometric distribution is produced when there will be $x-1$ failures before a successful event (that occurs with probability p) and the system stops.

We note that a geometric series, for $-1 < r < 1$ gives

$$g(x) = \sum_{k=0}^{\infty} a\, r^k = \frac{a}{1-r}\,,$$

and the first differential is

$$g'(x) = \sum_{k=1}^{\infty} a\, k\, r^{k-1} = \frac{a}{(1-r)^2}\ .$$

If X has a geometric distribution and $0 < p < 1$, then the expectation of X is given by

$$E(X) = \sum_{x=1}^{\infty} x\, q^{x-1}\, p = \frac{p}{(1-q)^2} = \frac{1}{p}\,,$$

using the above formula for $g'(x)$ with $a = p$ and $r = q$.