SPA-Based Adaptive Chosen-Ciphertext Attack on RSA Implementation

Roman Novak

Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia,
Roman.Novak@ijs.si

Abstract. We describe an adaptive chosen-ciphertext attack on a smart card implementation of the RSA decryption algorithm in the presence of side-channel information leakage. We studied the information leakage through power consumption variation. Simple power analysis (SPA) of the smart card that is widely used for secure Internet banking, Web access and remote access to corporate networks, revealed macro characteristics caused by improper implementation of Chinese remaindering. The findings can be used to eventually improve future implementations of fast RSA decryption.

1 Introduction

Smart card-based authentication and digital signature generation provide user identity for a broad range of business applications. The embedded microcontroller accompanied with cryptoprocessor and memory capabilities promises numerous security benefits. However, as security processor technology advances, new techniques are developed that compromise the benefits of its use. The research of new attack techniques contributes to improvement of future products while new protective measures pose new challenges to cryptoanalysts.

The implementation of a particular cryptographic algorithm often introduces new attack possibilities. Smart cards are prone to reverse engineering using chip testing equipment [1,2]. A well-known glitch attack introduces computation errors, which can be used very successfully to recover secrets by an adversary. For instance, a glitch attack against RSA [3] implementation based on the Chinese Remainder Theorem (CRT) could recover the private key using only one message and corresponding faulty signature [4,5]. Furthermore, the implementation often leaks additional side-channel information. Non-invasive attacks have been proposed based on timing information, a device's power consumption, and electromagnetic radiation [6,7]. Designers of cryptographic devices are very aware of attacks based on side-channel information leakage. Much attention has been given to preventing complex attacks while security holes remain and allow simple methods to succeed.

Simple Power Analysis (SPA) interprets a circuit's power consumption. More advanced techniques, like Differential Power Analysis (DPA) and Inferential Power Analysis (IPA), allow observation of the effects correlated to data values being manipulated [8,9]. Power analysis attacks have been known for a while

D. Naccache and P. Paillier (Eds.): PKC 2002, LNCS 2274, pp. 252–262, 2002.
© Springer-Verlag Berlin Heidelberg 2002

and effective countermeasures exist that pose difficulties, even to a well funded and knowledgeable adversary [2]. On the other hand, it is difficult to address all weaknesses in implementing a cryptographic algorithm. The intent of this paper is to show a particular weakness of a smart-card-based RSA implementation. Little side-channel information is required to break this very common implementation.

We have applied SPA to a RSA-capable smart card. In particular, we have been testing a card that is used for secure Internet banking, Web access and remote access to corporate networks world-wide. The card provider is among leaders in the integration of strong authentication and electronic certification technology. The card embeds a cryptoprocessor dedicated to security. On the card, the DES, Triple-DES and RSA algorithms are implemented. Attention has been given to the RSA decryption operation.

Protective measures against power analysis attacks have been detected. However, the card's implementation of RSA decryption is based on the Chinese Remainder Theorem and leaks the information, which allows an adaptive chosen-ciphertext type of attack to reconstruct the private key.

The rest of the paper is structured as follows. Section 2 gives a short introduction on Simple Power Analysis. Section 3 defines a power trace, describes the data acquisition equipment and test parameters. Section 4 analyses a typical power consumption pattern during RSA decryption. SPA macro-characteristics are identified. Section 5 correlates power traces with a common RSA decryption algorithm based on the Chinese Remainder Theorem. The information leakage function is defined and its properties are studied. In Sect. 6 the adaptive chosen-ciphertext attack is described that makes use of the information leakage function. An RSA private key can be reconstructed by performing $t/2$ decryptions, where t is the bitlength of public modulus n. We conclude the paper by summarising our findings in Sect. 7.

2 Simple Power Analysis

Smart cards consist of logic gates, which are basically interconnected transistors. During operation, charges are applied to or removed from transistor gates. The sum of all charges can be measured through power consumption. Power analysis techniques are based on measurements of a circuit's power consumption. A similar approach may be used on electromagnetic radiation traces.

Several variations of power analysis have been developed [8,9]. The power consumption measurements of the smart card operations are interpreted directly in Simple Power Analysis (SPA). SPA can reveal instruction sequence and it can be used to reveal hidden data in algorithms in which the execution path depends on the data being processed. Conditional branching causes large SPA and sometimes timing characteristics. The method has been known for a while and fairly simple countermeasures exist [8].

The implementers of cryptographic algorithms are usually aware of SPA based attacks. They decide not to implement appropriate countermeasures only

if they believe that a particular SPA characteristic could not threaten the overall security scheme.

3 Power Consumption Measurements

Power consumption measurements were obtained by measuring voltage variations across a resistor (25 ohm) that was inserted in series with the card ground pin. In the following, a power trace refers to a set of power consumption measurements taken across a smart card operation.

The sampling speed was set at approximately 7.15 MSamples/s. 14-bit resolution was used. Measurements were taken asynchronously with the internal processing of the card because the card uses the external clock signal only during communication with the reader. Otherwise, it uses an internally generated randomised clock, which is considered to be, in combination with some other countermeasures, one of the best countermeasures against power analysis attacks [2]. The internal clock is not available on the card's contacts.

One of the challenges of taking a measurement is to successfully trigger acquisition at the point of interest. A card actually performs a requested task after communication has taken place between a reader and the card on the dedicated line. The acquisition trigger point should be set with respect to the activities on the communication line. A pre-settable low-to-high edge counter suffices for triggering.

PC based data acquisition cards are available on the market at a price of several thousand dollars. In order to evaluate the minimum resources needed to perform simple power analysis we have decided to develop our own data acquisition equipment at minimum cost. Our hardware module has enough onboard memory to store a power trace of 100 ms duration. The total cost of the module was less than $100. However, the cost of the development was not negligible.

4 SPA Characteristics of RSA Decryption

In our case, power traces exhibit easily identifiable macro-characteristics of the RSA decryption algorithm. A 512-bit modulus is used in the example. The length of decryption varies with time. The average time needed by the card to complete the operation is approximately 70 ms. Figure 1 shows typical power consumption patterns during RSA decryption. After the first 3 ms of operation the trace remains periodic for some time. Larger deviations in power consumption occur approximately at the middle of the trace and in the last 3 ms of RSA decryption.

Large macro-features of the RSA decryption operation may be identified, since the operations performed by different parts of a card's architecture vary significantly. Selective use of the cryptoprocessor may cause such variations in power consumption. We found a more detailed view of the same trace less informative. Further investigation using more sophisticated equipment showed that the card uses an internal clock and that the frequency of the internal clock is probably intentionally randomised.

Two periodic sequences, which consume more than 90% of the time, come from two exponentiations. Usually a RSA decryption implementation makes use of the Chinese Remainder Theorem, where two modular exponentiations with smaller moduli are performed instead of one. The number of areas with increased power consumption within each periodic sequence agrees with the number of squarings and multiplications in a repeated square-and-multiply binary exponentiation algorithm [10]. The number of squarings is one less than the bitlength of the exponent while the number of multiplications is equal to one less than the number of 1's in the exponent's binary representation. We could not differentiate multiplying from squaring with our power sampling equipment.

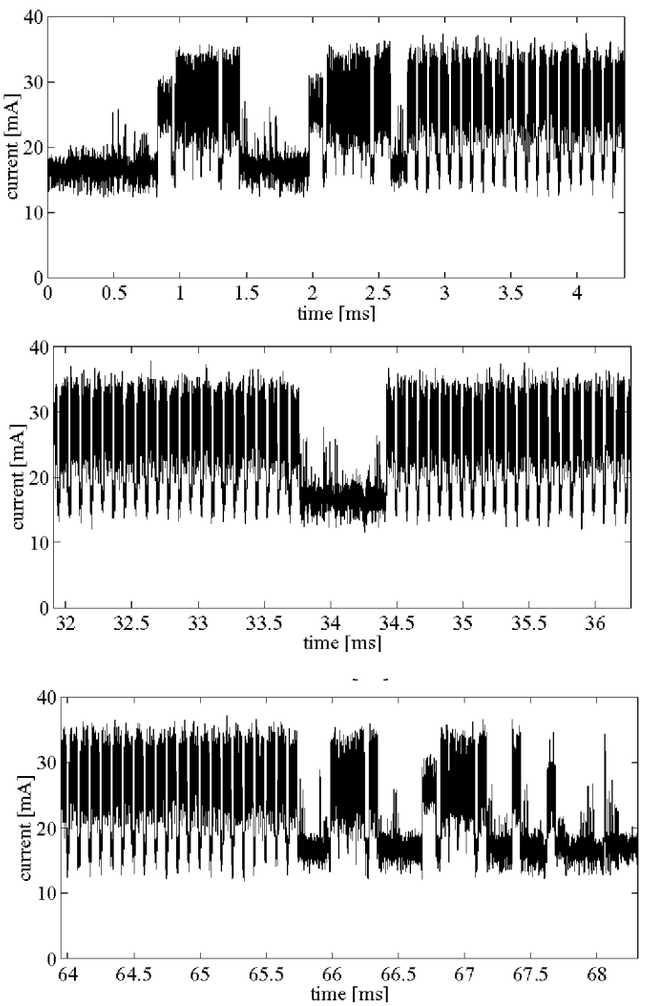

Fig. 1. Typical power consumption patterns during RSA decryption

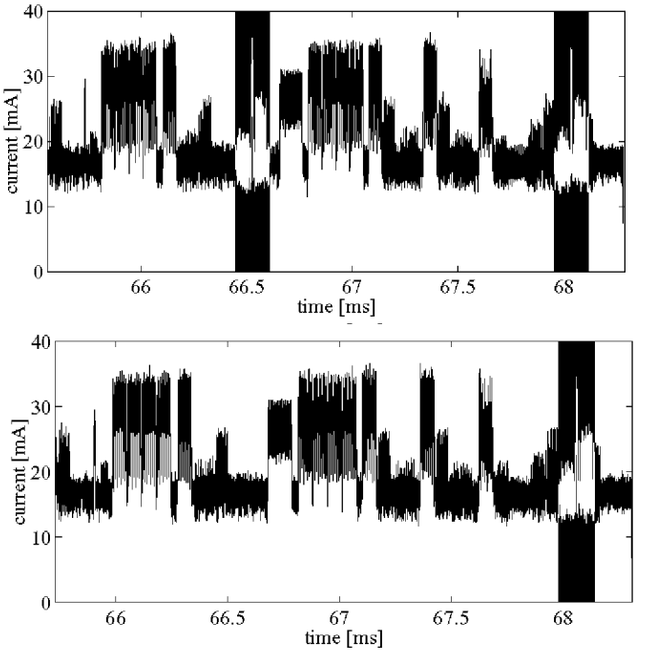

Fig. 2. Two types of power trace tails

The areas of larger consumption before and after exponentiations have to correspond to the required math steps of the algorithm outlined in the next section. The lengths of these areas appear to be randomised. We have not discovered any useful correlation with input ciphertext or output plaintext. However, the comparison between several power traces reveals a slight difference in the computation that follows both exponentiations. In Fig. 2, two similar patterns are highlighted on the upper trace while the first highlighted pattern is missing from the lower trace. We thought that a conditional branch in the RSA decryption might explain the optional pattern.

We identified these optional operations as the conditional add in Garner's version of CRT modular exponentiation. Figure 3 shows a detailed view of optional pattern. We show that an adversary can use the information about the algorithm's execution path in the adaptive chosen-ciphertext attack.

5 Leaked Information

Suppose p and q are distinct primes, and let modulus $n = pq$. Let e be an encryption exponent and d a decryption exponent, respectively. Pair (n, e) is publicly known while d is kept private. The RSA encryption computes $c = x^e \bmod n$ for some $x \in \mathbb{Z}_n$. The decryption also involves exponentiation in $\mathbb{Z}_n$ using the decryption exponent d, $x = c^d \bmod n$.

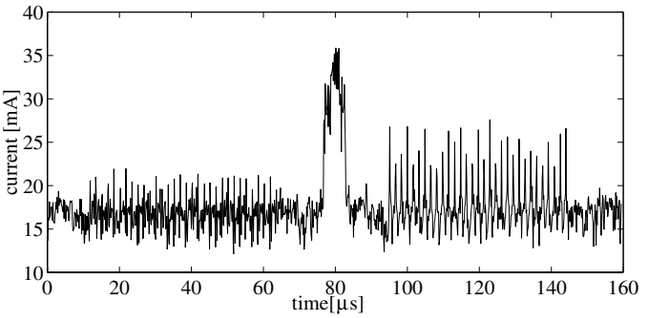

Fig. 3. Detailed view of optional pattern

When p and q are part of the private key, modular or mixed-radix representation of numbers can be used to improve the performance of RSA decryption. Each integer $x \in \mathbb{Z}_n$ can be uniquely represented by the numbers $x_p = x \bmod p$ and $x_q = x \bmod q$. RSA decryption can be performed using modular representation by two exponentiations, each considerably more efficient because the moduli are smaller. The modular representation of the value $c^d \bmod n$ can be computed as $c^d \bmod p$ and $c^d \bmod q$. In fact, as a consequence of Fermat's theorem, only $c^{d_{p-1}} \bmod p$ and $c^{d_{q-1}} \bmod q$ may be computed, where $d_{p-1} = d \bmod (p-1)$ and $d_{q-1} = d \bmod (q-1)$. The algorithm that solves simultaneous congruences in the Chinese Remainder Theorem (CRT) can do the conversion from a modular representation back to a standard radix representation. Gauss's algorithm for solving the associated system of linear congruences is not the best choice for large integers. Garner's algorithm has some computational advantages and is usually used for RSA decryption. Algorithm 1 outlines major steps of the RSA decryption algorithm using Garner's algorithm for CRT.

Algorithm 1. RSA decryption algorithm using Garner's algorithm for CRT
INPUT: ciphertext c, primes p and q, $p > q$, precomputed values $d_{p-1} = d \bmod (p-1)$, $d_{q-1} = d \bmod (q-1)$, $u = q^{-1} \bmod p$.
OUTPUT: plaintext x.

1. $x_p = c^{d_{p-1}} \bmod p$.
2. $x_q = c^{d_{q-1}} \bmod q$.
3. $t = x_p - x_q$.
4. If $t < 0$ then $t = t + p$.
5. $x = (((tu) \bmod p)\, q) + x_q$.

Step 4 needs additional explanation. The plaintext x is required to be from $\mathbb{Z}_n$. Because the difference in step 3 may be negative an additional conversion to least positive residue is required. The common way of doing that is to require prime p to be greater than prime q and to perform an addition in step 4.

Implementation of the above algorithm can produce the optional pattern in a power trace as a result of the conditional addition in step 4. The addition is

also performed as the last operation in step 5, which explains the similar pattern at the end of the power trace.

We define function *diff* as the information leakage function (1). It returns 1 if the addition in step 4 is needed and 0 otherwise. Its argument is output plaintext x. The function can be evaluated only by analysing the power trace of the RSA decryption.

$$\mathit{diff}\,(x) = \begin{cases} 1 & x \bmod p - x \bmod q < 0 \\ 0 & \text{otherwise} \end{cases}, \quad p > q \ . \tag{1}$$

The information leakage function has the following properties that can be used in adaptive chosen-ciphertext attack by an adversary:

1. *diff* changes value from 0 to 1 only at multiples of prime p, $\mathit{diff}\,(kp-1) = 0$, $\mathit{diff}\,(kp) = 1$, $0 < k < q$.
2. The value of *diff* remains 1 for l consecutive values of argument x, where $0 < l < q$. l takes each value between 0 and q exactly once.

Proof (of the first property). One should note that, on increasing x, the difference $x \bmod p - x \bmod q$ may change its value only in multiples of primes p and q. Therefore, *diff* may change its value at the same values. Furthermore, $\mathit{diff}\,(kp) = 1$, $0 < k < q$, while $\mathit{diff}\,(mq) = 0$, $0 < m < p$, due to $kp \bmod q \neq 0$ and $mq \bmod p \neq 0$. Finally, because $\mathit{diff}\,(0) = 0$ and because the relation $p > q$ ensures at least one multiple of prime q between two multiples of prime p, *diff* must always change value from 0 to 1 at multiples of prime p. □

Proof (of the second property). The sequence of 1's cannot be longer than $q - 1$ because it is started at a multiple of prime p and terminated at a multiple of prime q. Suppose a sequence is terminated at kq. The sequence length is then $kq \bmod p$, which is one of two components of kq's modular representation. Because the other component is 0 and each number has a unique modular representation, two sequences cannot have the same length, which proves the second property. □

The number of cases for which $\mathit{diff}\,(x)$ equals each of two values can be computed using the above properties. Among n possible values for x, *diff* takes 1 in $(q-1)q/2$ cases and 0 in $pq - (q-1)q/2$ cases. Therefore, the probability to randomly pick a plaintext x where $\mathit{diff}\,(x) = 1$ is $(q-1)/(2p)$. In order to illustrate the properties of the information leakage function, Tab. 1 displays *diff* for $n = 35$, $p = 7$ and $q = 5$.

6 Key Reconstruction

The reconstruction of secret key d is possible by finding prime p. An adversary may start with plaintexts x_1 and x_2, such that $\mathit{diff}\,(x_1) = 0$ and $\mathit{diff}\,(x_2) = 1$. Then, using a binary search-like algorithm and SPA information, he finds the value x, where $\mathit{diff}\,(x-1) = 0$ and $\mathit{diff}\,(x) = 1$. The value x is a multiple

Table 1. The information leakage function *diff* for $n = 35$, $p = 7$ and $q = 5$

x	x_p	x_q	$diff(x)$	x	x_p	x_q	$diff(x)$	x	x_p	x_q	$diff(x)$
0	0	0	0	14	0	4	**1**	28	0	3	**1**
1	1	1	0	15	1	0	0	29	1	4	**1**
2	2	2	0	16	2	1	0	30	2	0	0
3	3	3	0	17	3	2	0	31	3	1	0
4	4	4	0	18	4	3	0	32	4	2	0
5	5	0	0	19	5	4	0	33	5	3	0
6	6	1	0	20	6	0	0	34	6	4	0
7	0	2	**1**	21	0	1	**1**				
8	1	3	**1**	22	1	2	**1**				
9	2	4	**1**	23	2	3	**1**				
10	3	0	0	24	3	4	**1**				
11	4	1	0	25	4	0	0				
12	5	2	0	26	5	1	0				
13	6	3	0	27	6	2	0				

of prime p that can be extracted by finding the greatest common divisor of x and modulus n. The adversary can control the output x of the RSA decryption by feeding the card with $x^e \bmod n$. The attack may be classified as SPA-based adaptive chosen-ciphertext attack. Algorithm 2 gives the steps that can be taken by the adversary to recover key material.

Algorithm 2. Reconstruction of prime p
INPUT: modulus n, public exponent e.
OUTPUT: prime p such that p divides n.

1. Repeat until computed $diff(x)$ equals 1:
   (a) Pick random number x, $0 \leq x < n$.
   (b) $c = x^e \bmod n$.
   (c) Compute $diff(x)$ by analysing power trace while card decrypts c.
2. $m = x/2$, $l = 0$.
3. While $m \neq l$ do:
   (a) $c = m^e \bmod n$.
   (b) Compute $diff(m)$ by analysing power trace while card decrypts c.
   (c) If $diff(m) = 1$ then $x = m$; otherwise $l = m$.
   (d) $m = (l + x)/2$.
4. Compute $p = gcd(n, x)$.
5. Return(p).

The average number of repetitions needed in step 1 is determined by the probability $(q - 1)/(2p)$, computed in the previous section. The probability is usually high and only a few repetitions of step 1 is needed to pick x where

$\mathit{diff}(x) = 1$. At the end of step 3, variable x holds the value that is a multiple of prime p. l and x are lower and upper bounds of the search interval. On each repetition the interval is halved while $\mathit{diff}(l)$ remains 0 and $\mathit{diff}(x)$ remains 1. At $l = x - 1$ the first property of the information leakage function guarantees the upper bound x to be a multiple of prime p.

Primes p and q are in practice about the same bitlength, and sufficiently large to avoid the elliptic curve factoring algorithm [10]. For example, if a 1024-bit modulus n is to be used, then each of p and q should be about 512 bits in length. This can be used to speed up the restoration of prime p.

Suppose modulus n has a bitlength of t bits while primes p and q have a bitlength of $t/2$ bits, $2^{t/2-1} < p, q < 2^{t/2}$, and t is sufficiently large. The first sequence of 1's starts at p and ends at $2q - 1$, therefore $\mathit{diff}(2^{t/2}) = 1$. In that case, the first and fourth steps of the Alg. 2 are not needed, while the second step may be entered with $x = 2^{t/2}$. The modified algorithm requires only $t/2$ power traces.

Algorithm 3. Reconstruction of p when its bitlength is half bitlength of n
INPUT: modulus n with the bitlength t, public exponent e.
OUTPUT: prime p such that p divides n.

1. $x = 2^{t/2}$, $m = x/2$, $l = 0$.
2. While $m \neq l$ do:
   (a) $c = m^e \bmod n$.
   (b) Compute $\mathit{diff}(m)$ by analysing power trace while card decrypts c.
   (c) If $\mathit{diff}(m) = 1$ then $x = m$; otherwise $l = m$.
   (d) $m = (l + x)/2$.
3. Return(x).

Using Alg. 3 we managed to restore one of two secret primes and compute secret key d, which confirms our hypothesis about the implemented algorithm on our test card.

7 Conclusion

Power consumption is a well-known source of side-channel information leakage. Several techniques have been developed in the past to gather leaked information. SPA is the simplest among them. Efficient hardware and software based prevention measures exist. The implementers of cryptographic operations are usually aware of SPA based attacks. They decide not to implement appropriate countermeasures only if they believe that a particular SPA characteristic could not threaten the overall security scheme. This is not the practice to be followed.

We have reviewed a fast RSA decryption algorithm that is implemented on the smart card that embeds some state-of-the-art cryptographic solutions. Dedicated hardware for operations with very long integers may enable SPA to reveal macro steps of RSA decryption. When Garner's algorithm is used for conversion from modular representation back to a standard radix representation, improper

implementation could leak the information about the modular representation of the plaintext. We have shown how an adversary could use very few bits of information in an adaptive chosen-ciphertext attack. The algorithm, given in the paper, requires $t/2$ decryptions, where t is the bitlength of the public modulus n. This kind of attack is possible only on open cards, which allow to make the choice of the input. Timing attack can be mounted under the same principles, however, the internally generated randomised clock and the additional software random delays require extraction of the SPA feature.

Proper implementation of Garner's algorithm should hide SPA characteristics that make factorisation of public modulus feasible. This can be achieved by balancing conditional operations with dummy operations, or even better, by changing the algorithm to use a constant execution path. Other protective measures should be reconsidered. We have shown that the cryptoprocessor carries an additional threat to security due to the easily detectable patterns of its use.

Every side-channel leakage must be considered with great care at the implementation phase of the algorithm. Even though countermeasures against sophisticated attacks exist, designs must be carefully checked against simple methods as well.

References

1. Anderson, R., Kuhn, M.G.: Tamper Resistance - a Cautionary Note. Proceedings of the Second USENIX Workshop on Electronic Commerce, Oakland, California, November 18–21, USENIX Association (1996) 1–11
2. Kömmerling, O., Kuhn, M.G.: Design Principles for Tamper-Resistant Smartcard Processors. Proceedings of the USENIX Workshop on Smartcard Technology - Smartcard'99, Chicago, Illinois, May 10–11, USENIX Association (1999) 9–20
3. Rivest, R.L., Shamir, A., Adleman L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM **21(2)** (1978) 120–126
4. Biham, E., Shamir, A.: Differential Fault Analysis of Secret Key Cryptosystems. In: Kaliski, B.S. (ed.): Advances in Cryptology - Crypto'97. Lecture Notes in Computer Science, Vol. 1294. Springer-Verlag, Berlin Heidelberg New York (1997) 513–525
5. Bao, F., et al.: Breaking Public Key Cryptosystems in the Presence of Transient Faults. In: Christianson, B., et al. (eds.): Security Protocols. Lecture Notes in Computer Science, Vol. 1361. Springer-Verlag, Berlin Heidelberg New York (1997) 115–124
6. Kocher, P.: Timing Attacks on Implementation of Diffie-Hellman, RSA, DSS and Other Systems. In: Koblitz, N. (ed.): Advances in Cryptology - Crypto'96. Lecture Notes in Computer Science, Vol. 1109. Springer-Verlag, Berlin Heidelberg New York (1996) 104–113
7. Messerges, T.S., Dabbish, E.A., Sloan, R.H.: Investigation of Power Analysis Attacks on Smartcards. Proceedings of the USENIX Workshop on Smartcard Technology - Smartcard'99, Chicago, Illinois, May 10–11, USENIX Association (1999) 151–161

8. Kocher, P., Jaffe, J., Jun, B.: Differential Power Analysis. In: Wiener, M. (ed.): Advances in Cryptology - Crypto'99. Lecture Notes in Computer Science, Vol. 1666. Springer-Verlag, Berlin Heidelberg New York (1999) 388–397
9. Fahn, P.N., Pearson, P.K.: IPA: A New Class of Power Attacks. In: Koc, C.K., Paar, C. (eds.) Cryptographic Hardware and Embedded Systems - CHES'99. Lecture Notes in Computer Science, Vol. 1717. Springer-Verlag, Berlin Heidelberg New York (1999) 173–186
10. Menezes, A.J., van Oorschot, P.C., Vanstone, S.A.: Handbook of Applied Cryptography. CRC Press Series on Discrete Mathematics and Its Applications (1996)