Side-Channel Attack on Substitution Blocks

Roman Novak

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

Abstract. [1] We describe a side-channel attack on a substitution block, which is usually implemented as a table lookup operation. In particular, we have investigated smartcard implementations. The attack is based on the identifying equal intermediate results from power measurements while the actual values of these intermediates remain unknown. A powerful attack on substitution blocks can be mounted if the same table is used in multiple iterations and if cross-iteration comparisons are possible. Adversaries can use the method as a part of reverse engineering tools on secret algorithms. In addition to the described method, other methods have to be employed to completely restore the algorithm and its accompanying secret key. We have successfully used the method in a demonstration attack on a secret authentication and session-key generation algorithm implemented on SIM cards in GSM networks. The findings provide guidance for designing smartcard solutions that are secure against this kind of attack.

1 Introduction

Modelling cryptographic algorithms as mathematical objects cannot address weaknesses in the implementation of these algorithms in real-world cryptographic devices. Any real cryptographic device provides more information to a determined adversary than just the input plaintext and output ciphertext. This side-channel information is available as the timing of operations [1], power consumption of the devices [2], electromagnetic emanations [3], etc. Very little side-channel information is required to break many common ciphers. Non-invasive attacks and accompanying countermeasures have been studied extensively over the past few years.

Systems that rely on smartcards are of particular concern. Examples of such systems are secure Internet banking, remote access to corporate networks worldwide, existing GSM phone networks, pay-TV systems, electronic wallets etc. 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. Research on new attack techniques contributes to the improvement of future products.

[1] J. Zhou, M. Yung, Y. Han (Eds.): ACNS 2003, LNCS 2846, pp. 307-318, 2003.

Many supposedly secure implementations remain vulnerable. In this paper, a side-channel attack on substitution blocks is presented. Power measurements were used as the source of side-channel information. The substitution block is one of the fundamental primitives in cryptography for introducing non-linearity. In software, a substitution block is usually implemented as a table lookup operation. Obtaining resistance to side-channel attack appears to be a difficult task. The technique to be described can be used to attack implementations with weak side-channel countermeasures, i.e., inadequate implementation of masking techniques [4, 5], or the implementation of secret algorithms. However, other methods should be employed to completely restore the secret algorithm and its accompanying key.

The side-channel attack on a substitution block is based on identifying equal intermediate results while the actual values of these intermediates remain unknown. An adversary can partially or fully restore the content of the lookup table and thus break an unknown substitution block. A demonstration attack was performed on the secret authentication and session-key generation algorithm implemented on SIM cards that are used in GSM networks. Specification of the secret algorithm was not available. All that was known was that the algorithm was not COMP128-1. Correctness of the restored algorithm was checked by means of plaintext/ciphertext pairs. A simplified example is given in order to keep the algorithm secret and be clear enough.

The purpose of this paper is to draw designers' attention to weak protective measures and to show that secret algorithms offer very little protection. The rest of the paper is structured as follows. Section 2 gives a short introduction to power analysis techniques. In Sect. 3 the problem of protecting substitution blocks is detailed. Some common errors in implementing the cardinal principle are given that can lead to the success of the proposed method. The environment in which the side-channel attack has been validated is described. The attack is presented in Sect. 4. The techniques that make search for lookup table feasible are given in Sect. 5 with examples. The actual attack would require the identification of relevant measurements. There are no general rules for identifying them reliably. Section 6 gives some guidelines that can be followed. Countermeasures against the proposed side-channel attack are discussed in Sec. 7. We conclude by summarising our findings in Sect. 8.

2 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, on which power analysis techniques are based. A similar approach may be used on electromagnetic radiation traces.

Several variations of power analysis have been developed [2, 6]. The power consumption measurements of smart card operations are interpreted directly in Simple Power Analysis (SPA). SPA can reveal hidden data in algorithms in

which the execution path depends on the data being processed. More advanced techniques, like Differential Power Analysis (DPA) and Inferential Power Analysis (IPA), allow observation of the effects correlated to the data values being manipulated.

Power analysis attacks have been known for a while and effective countermeasures exist that pose difficulties, even to a well-funded and knowledgeable adversary [7]. On the other hand, it is difficult to address all the weaknesses in implementing a cryptographic algorithm. Frequently, the developers decide not to implement appropriate countermeasures if they believe that a particular power characteristic could not threaten the overall security scheme. This is not a practice to be followed.

Some of the countermeasures, like table mask operation [4, 5], deal directly with the protection of substitution blocks; others protect the algorithm as a whole, i.e., timing randomness at the clock-cycle level [7]. Many countermeasures may be bypassed or compensated for [8]; the implementation of many others has security vulnerabilities. The side-channel attack on a substitution block may effectively remove such incomplete countermeasures. Furthermore, it can be used to reverse-engineer an unknown substitution block. As such, it is another argument against establishing secrecy by keeping cryptographic algorithms undisclosed.

3 Problem Description

Substitution has been known from traditional cryptography. For instance, one of the oldest known substitution ciphers is the Caesar cipher [9]. In this type of cipher each letter or group of letters is replaced by another letter or group of letters in order to disguise it. Modern cryptography still uses substitution as a building block in more complex compositions. By combining simple transformations it is possible to obtain strong ciphers. Substitution blocks are considered to provide a high security level because they contribute effectively to data diffusion.

Substitution block is a fundamental primitive used by many cryptographic algorithms such as DES, AES and COMP128. It is often implemented as table lookup. However, any implementation that directly looks up a table is vulnerable to differential side-channel attacks since the side-channel signals at the time of table lookup will correlate with each bit of the index accessed and with each bit of the value retrieved. To remove this correlation, protective measures have been proposed that are more or less close to the following cardinal principle [10]:

Definition 1 (cardinal principle). *Relevant bits of all intermediate cycles and their values should be statistically independent of the inputs, outputs and sensitive information if differential attacks are to be completely eliminated.*

For example, the table mask operation is proposed in [5]. For each instance of a cryptographic operation requiring one or more lookups of a table, a fresh random looking masked table is computed. This is done by selecting two permutations *ip* and *op* uniformly at random. The table indexes are then permuted

using the permutation ip. Likewise, the table elements are permuted using the permutation op. A simple example would be to choose two random values x and y, and then compute $T'(i \oplus x) = T(i) \oplus y$ for all values of index i, where T is the original table and T' is the masked table.

Implementation of such a protective measure, especially on devices with resource and cost limitations, is a challenging task. Many implementations fail to perform securely. The main problem with the table mask operation as described above is that it requires the table T' to be in RAM and the size of T' is the same as that of T. Moreover, the same table is used more than once in cryptographic algorithms. The use of the same masked table throughout the algorithm poses additional security threats, as an adversary may be able to restore some masked values using a method similar to the one described here.

Time limits on cryptographic algorithms prevent computation of a unique table for each use. For example, our secret authentication and session-key generation algorithm in GSM phone networks requires a lookup of two tables. Throughout the algorithm each table is used 1280 times. Therefore, 2560 table mask operations should be performed. In that case, the table mask operation itself may be the subject of an attack.

Some implementations use fixed masked tables in permanent memory that are specific to the card vendor or cryptographic device. For example, the size of the memory required to store the S-boxes in a masked version of the DES algorithm in [4] is too big for smartcards. The solution proposed is based on a secret bijective function which may be implemented as a fixed secret table. Other implementations may recalculate masked tables only during card initialisation, in which case different inputs may be fed to the algorithm with the same tables. Those solutions give little or no protection at all.

A fundamental rule of cryptography is that one must assume that the cryptanalyst knows the general method of encryption used. Experts have learned over the years that the only way to assure security is to follow an open design process, encouraging public review to identify flaws while they can still be fixed. Secrecy comes from having a strong public algorithm and a long key. However, many cryptographic algorithms are still kept secret. For instance, the GSM network operators use an updated version of COMP128-1, designated as COMP128-2, but the algorithm remains unpublished. Some network operators even develop a proprietary algorithm in secrecy. In either case, the algorithm used has not been publicly reviewed. A side-channel attack on the substitution block may be used as part of the reverse engineering tools against such secret algorithms.

The side-channel attack on substitution blocks described below has been validated on the SIM (Subscriber Identity Module) card deployed on several international networks. The contents of the lookup tables of the implemented authentication and session-key generation algorithm (A3A8) were not given to us. On the other hand, we knew the algorithm architecture, computations involved and the secret key. This information was restored using other reverse engineering tools, which are not covered in this paper. Some of the methods involved are new and will be published elsewhere. Protective measures against

power analysis attacks were detected and compensated for. The card uses fixed tables; therefore, multiple repetitions at the same input can be used to average out the signal noise. The experiment was part of a larger ongoing project in which the significance of the side-channel information is evaluated in various reverse engineering techniques.

As described, the method requires from the cryptanalyst the ability to encrypt pieces of plaintext of his own choosing; however, it can be adapted to the situation where plaintext is known but not the ability to choose it.

4 Breaking a Substitution Block

Let $f(p)$ be a function that incorporates a lookup table T and some further transformations of the value read from the table. The parameter p represents plaintext input and may be extended to a sequence of parameters without significant change of the method.

$$r = f(p) \tag{1}$$

The problem that has to be solved by an adversary is to find the content of the unknown or modified lookup table T just by observing the side-channel information that is present in power variations. The value of parameter p is known to the attacker, while the result r is unknown, since it is further modified during algorithm execution.

The basic idea behind the attack on a substitution block is based on the fact that the same value of the parameter p gives the same intermediate result r, while different values of parameter p do not necessarily give different intermediate results r as soon as f is an injective function. By identifying equal intermediate results one can partially or fully restore the content of the lookup table and thus break an unknown substitution block. The method differs from template attack, where profiling of the experimental device is required before attack [11].

First, identification of the relevant measurements is needed to enable comparison of the results within algorithm execution by side-channel information alone. We define an equality function that is based on the relevant measurements.

Definition 2 (equality function). *Let N be the number of repetitions of the algorithm for a given value of parameter p that can be achieved within real-time constraints. Suppose $\overline{m}_{p,t}$ is the average of measurements of supply current $m_{p,t}$ at time t for a given value p,*

$$\overline{m}_{p,t} = \frac{1}{N} \sum_{j=1}^{N} m_{p,t} \ . \tag{2}$$

The equality function $ef(p_1, p_2, r)$ is defined such that $ef(p_1, p_2, r)$ is true if and only if $f(p_1)$ is equal to $f(p_2)$, where r are time indices $t_1, t_2, ..., t_n$ of the relevant measurements.

We use the following equality function in our experiments

$$ef(p_1, p_2, r) = \sum_{i=1}^{n} |\overline{m}_{p_1,t_i} - \overline{m}_{p_2,t_i}| < T \ , \tag{3}$$

where T is a threshold value. The function (3) does not necessarily exist for given smart card implementation. One can define different equality function for the same purpose of distinguishing values $f(p_1)$ and $f(p_2)$.

In the case of (3), the main problem for an adversary is to identify relevant measurements and to select suitable threshold value. Different techniques may be used for that purpose; several of them are mentioned in the following subsection. At this point we suppose that the equality function is known and can be evaluated effectively using a side-channel information leakage.

An equation can be written for each pair of parameters p_1 and p_2 for which the equality function holds:

$$f(p_1) = f(p_2), \ p_1 \neq p_2 \ . \tag{4}$$

It is not necessary to find all similarities and, hence, all equations. The set of equations is then solved for the table values with some degree-of-freedom, DF. The number of tables that solve the set of equations grows fast with DF. When the table contains a permutation, the number of solutions is bounded by the number of variations of 2^n elements choose DF, where each table entry is composed of n-bits. In case of a compression table, the upper bound on the number of solutions is 2^{nDF}. Nevertheless, only one solution to the set of equations is actually present in the implementation. An adversary can find the correct table by doing an exhaustive search. Therefore, it is important for DF to be very small, in practice not larger than 2. A large DF makes exhaustive search of the solution space infeasible. The right table can be identified using DPA on the computation results after the lookup operation.

5 Making the Search for Lookup Table Feasible

The requirement for small DF may prevent breaking a substitution block when only a single lookup operation is observed. However, the same table is frequently used several times within an algorithm because the same substitution block is used several times to form the product cipher. Several lookups at different positions can be observed and the resulting sets of equations combined together. Furthermore, when the same code is used for all these lookups, cross-comparisons of the results are possible. The set of equations may be extended further.

In Fig. 1 a computation is shown that includes four lookup operations. The computation can be found in the GSM authentication algorithm that has been deployed by different service providers in several types of SIM cards. Only a small fraction of the sixteen similar iterations is shown.

The algorithm is a keyed hash function. It takes a 16-byte key (128 bits) and 16-byte of data (128 bits) to output a 12-byte (96 bits) hash. The key k_0-k_{15}, as

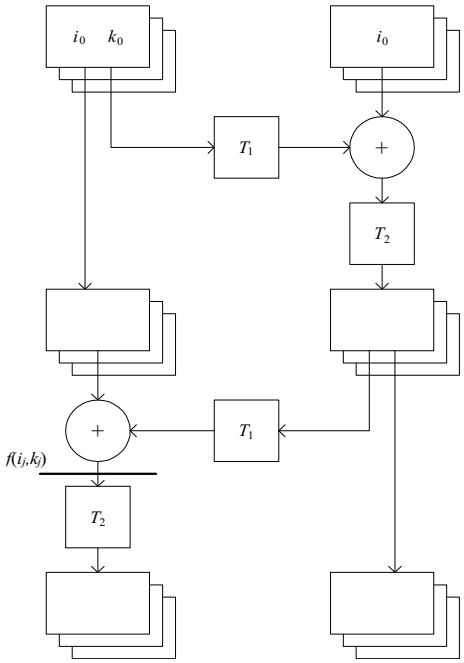

Fig. 1. Computation in the GSM authentication algorithm

used in the GSM protocol, is unique to each subscriber and is stored in the SIM card. The input data i_0-i_{15} is a random challenge supplied by the base station. The first 32 bits of the hash are used as a response to the challenge and sent back to the base station. The remaining 64 bits are used as a session key for voice encryption using the A5 algorithm.

In the example, the content of table T_1 is known while that of table T_2 is unknown. The relevant measurements have been identified such that (3) distinguishes different values $f(i_j, k_j)$, where $f(i_j, k_j)$ is defined as

$$f(i_j, k_j) = T_1(T_2(T_1(i_j \oplus k_j) \oplus i_j)) \oplus i_j \oplus k_j,\ j = 0...15\ . \tag{5}$$

The input value i_j may be chosen freely while the value k_j is part of the key, constant during investigation, and known to us. j is the iteration index. In the case of a single iteration, the resulting set of equations is highly undetermined, i.e. DF equals 157. Clearly, the restoration of table T_2 is an impossible task. However, the set of equations can be extended with equations from other iterations. Furthermore, cross-iteration comparisons are possible that reduce the number of iterations needed for a small DF. In order to demonstrate these two techniques, the 3-bit to 3-bit tables are used (Tab. 1), i.e., the tables consist of 2^3 elements of size 3-bits each.

The equality function can detect the following relations in the first iteration, where $k_0 = 5$:

Table 1. Lookup tables T_1 and T_2

index	0	1	2	3	4	5	6	7
T_1	2	5	3	6	1	7	4	0
T_2	7	1	3	0	6	2	5	4

$$\begin{gathered} f(0,5) = f(1,5) = f(4,5) = 4 \\ f(2,5) = f(5,5) = 1 \ . \end{gathered} \tag{6}$$

The actual function values are shown just to make the example more illustrative. They are unknown to an adversary during a real attack. The function values for inputs 3, 6 and 7 differ from each other and are not shown.

The next step is to write a set of equations based on the similarities. In this case, the degree-of-freedom equals 5, since there are 8 variables $T_2(x)$, and 4 of them are related to each other as follows:

$$\begin{gathered} T_1(T_2(7)) \oplus 5 = T_1(T_2(0)) \oplus 4 = T_1(T_2(1)) \oplus 1 \\ T_1(T_2(2)) \oplus 7 = T_1(T_2(7)) \ . \end{gathered} \tag{7}$$

Analysis of the second iteration with $k_1 = 7$ gives the relations

$$\begin{gathered} f(6,7) = f(7,7) = 3 \\ f(1,7) = f(4,7) = f(5,7) = 5 \end{gathered} \tag{8}$$

and a set of equations

$$\begin{gathered} T_1(T_2(3)) \oplus 1 = T_1(T_2(5)) \\ T_1(T_2(5)) \oplus 6 = T_1(T_2(2)) \oplus 3 = T_1(T_2(6)) \oplus 2 \ . \end{gathered} \tag{9}$$

Again, DF equals 5; however, when (7) and (9) are merged, the resulting DF equals 2. The process can be continued with other iterations until a DF of 1 is achieved, which is the lower bound on DF in this example. We applied this technique on our A3A8 algorithm that uses an 8-bit to 8-bit table T_2. The equations from 8 iterations out of 16 had to be merged in order to reach DF of 1. The actual number of iterations may vary with key values.

When the same code performs computations within iterations with minor differences in power consumption patterns, the possibility of finding relevant measurements exists such that cross-iteration comparisons can be carried out. In that case, the definition of the equality function has to be changed slightly in order to compare measurements with different time indices. In our simplified example, cross-iteration comparisons would produce the following relations, in addition to the relations stated so far:

$$\begin{gathered} f(3,5) = f(0,7) = 7 \\ f(6,5) = f(6,7) = f(7,7) = 3 \\ f(7,5) = f(2,7) = 6 \ . \end{gathered} \tag{10}$$

The final set of equations would contain (7), (9), and the following equations:

$$\begin{gathered} T_1(T_2(7)) \oplus 6 = T_1(T_2(0)) \oplus 7 \\ T_1(T_2(0)) \oplus 3 = T_1(T_2(3)) \oplus 1 = T_1(T_2(5)) \\ T_1(T_2(4)) \oplus 2 = T_1(T_2(5)) \oplus 5 \ . \end{gathered} \tag{11}$$

It can be shown that this set of equations has DF of 1. One must just pick a value for arbitrary table location and compute the rest of the table. 2^3 different values may be selected; the right one should be confirmed by an alternative method, for instance, a DPA on intermediate results after the lookup operation. We managed to identify relevant measurements in power traces for our SIM card that allowed cross-iteration comparisons. A power trace refers to a set of power consumption measurements taken across a card operation. Only 4 iterations were needed to achieve DF of 1 in the real world example.

6 Identification of Relevant Measurements

Clearly, the actual attack would be highly dependent on the algorithm being implemented and the architecture being used, and would require some guesswork on the part of the attacker as to the relevant measurements and the types of software countermeasures being used.

Identification of relevant measurements is a prerequisite for the success of the method. Relevant measurements are strongly correlated to the value $f(p)$. Such measurements can be taken during the processing of the value $f(p)$ and its transforms. For instance, the value $f(p)$ may be used as an index into the next lookup table T. As long as the table implements permutation, it preserves the equality relation, $T(f(p_1)) = T(f(p_2)) \Leftrightarrow f(p_1) = f(p_2)$. The measurements that are taken while $T(f(p))$ is read from or written to the memory may contribute to the existence of the equality function, and can, as such, be considered as relevant measurements.

Many measurements may contain effects correlated to the $f(p)$; not all of them are relevant measurements. For instance, Hamming weight information, which can be leaked during a bus transfer of parameter p [12], is usually weakly correlated to the $f(p)$. However, the averages $\overline{m}_{p_1,t_i}$ and $\overline{m}_{p_2,t_i}$, where $f(p_1) = f(p_2)$, may differ too much to include time index t_i in (3).

Some features of a power trace can be helpful in identifying relevant measurements. Such a feature can be a power pattern during memory access. Table lookups are usually implemented as memory read operations. They can be identified easily unless proper countermeasures are present. In Fig. 2 a power trace is shown for the first two lookups on our test SIM card. Power consumption measurements were obtained by measuring voltage variations across a resistor (25 ohm) that was inserted in series with the card ground pin. The sampling speed was set at approximately 7.15 MSamples/s. 14-bit resolution was used.

Underlined patterns clearly differ from other controller activities. Firstly, supply current increases; we suppose that this is due to the memory addressing phase. After that, a delay is introduced with lower power consumption before

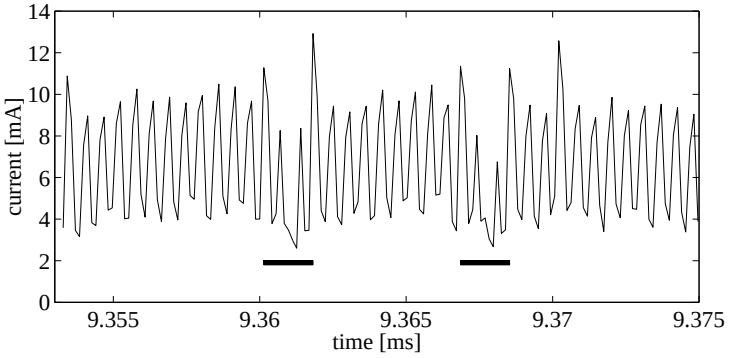

Fig. 2. Power trace of two lookup operations on a test SIM card

actual data retrieval. The measurements in the second group are the relevant measurements that can be used for restoring the content of the first lookup table. The measurements at the beginning of the selected interval contain effects correlated to the value $f(p)$, while the measurements at the end correlate to the value $T_2(f(p))$.

Other methods may be used to identify relevant measurements when they are not so obvious. The methods are based mainly on various correlation techniques. However, there is no general rule for identifying them reliably. The only confirmation that a proper equality function has been selected is the success of the method. Although the feasibility of the attack on real cards seams low due to probability of errors in the equality test, the existence of multiple relevant measurements and the possibility to average out noise usually lead to the success of the method.

7 Countermeasures

Many authors provide guidance for designing smart card solutions against power analysis attacks. Techniques for preventing side-channel attack on substitution blocks, as described in this paper, fall roughly into two categories.

A first approach is to prevent information leakage, using the general techniques that protect the algorithm as a whole. Well-known techniques are signal size reduction, introducing noise into power consumption, randomised execution timing and order, balancing state transitions, key use counters, physically shielding the device, blinding, multithreaded applications, etc [13, 2, 7, 10]. All these techniques make identification of relevant measurements difficult or even impossible.

On the other hand, many techniques can be used to bypass or compensate for these countermeasures. We suggest the use of as many redundant countermeasures as the available resources permit, because many of the countermeasures can be compensated for if they are implemented alone [8]. In our experience, a

subset of countermeasures offers a higher level of protection. For instance, attacks on substitution blocks, and many other power analysis techniques, require the attacker to predict the time at which a certain instruction is executed. As a protection, the designers insert random-time delays between the observable reactions that may be the subject of the attack. This countermeasure can be overcome rather easily using correlation techniques. The presence of other countermeasures, like feeding registers and busses with random values, can significantly strengthen the protection. The next example of synergic countermeasures is noise introduction in combination with key use counters. Noise can usually be averaged out, but in that case, a counter can prevent the attacker from gathering large number of samples.

A second category of countermeasures against side-channel attack on substitution blocks involves measures that are related to the content of the lookup table. The best way to eliminate the attack is adherence to the cardinal principle [10]. The table mask operation [5] is such a solution when different tables are used for each access. Great care must be taken when masking is performed in order not to create new security vulnerabilities.

8 Conclusion

The level of tamper resistance offered by any particular product can be measured by the time and cost penalty that the protective mechanisms impose on the attacker. The realities of a physical implementation can be extremely difficult to control. When the method is known, a side-channel attack on moderately protected smartcard typically requires a few minutes to several hours to complete, while the cost of the sampling equipment falls in the range from several hundred to several thousand dollars.

We have introduced a side-channel attack on substitution blocks, a fundamental primitive used by many cryptographic algorithms. We have shown how side-channel information could be used effectively on implementations that have been equipped with ad hoc and inadequate countermeasures. Although those countermeasures may resist some side-channel attacks, they fail to adhere to the cardinal principle. The method can be used as a part of reverse-engineering tools in the attacks on unknown algorithms.

The unknown intermediate results within the execution of the algorithm are compared. The comparisons form the basis for a set of equations, which is solved for the table values. An even more powerful attack on substitution blocks can be mounted if the table is used in multiple iterations and when cross-iteration comparisons are possible. The identification of relevant measurements is a prerequisite for the success of the method. We have shown how some features of a power trace help in identifying relevant measurements.

Proper countermeasures make identification of relevant measurements difficult or even impossible. The use of redundant countermeasures is suggested, since many protective measures can be bypassed or compensated for. The best way to eliminate attack is strict adherence to the cardinal principle. The designers must

not rely on secrecy of the algorithm, as the algorithm may be reverse-engineered in the presence of side-channel information leakage.

References

1. 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
2. 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
3. Agrawal D., Archambeault B., Rao J.R., Rohatgi P.: The EM Side-Channel(s): Attacks and Assessment Methodologies. In: Cryptographic Hardware and Embedded Systems - CHES'2002
4. Goubin L., Patarin J.: DES and Differential Power Analysis. In. Koc C.K., Paar C. (ed.): Cryptographic Hardware and Embedded Systems - CHES'1999. Lecture Notes in Computer Science, Vol. 1717. Springer-Verlag, Berlin Heidelberg New York (1999) 158–172
5. Rao J.R., Rohatgi P., Scherzer H., Tinguely S.: Partitioning Attacks: Or How to Rapidly Clone Some GSM Cards. Proceedings of the IEEE Symposium on Security and Privacy, Berkeley, California, May 12–15, IEEE Computer Society (2002) 31–44
6. 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
7. 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
8. Clavier C., Coron J.S., Dabbous N.: Differential Power Analysis in the Presence of Hardware Countermeasures. In. Koc C.K., Paar C. (ed.): Cryptographic Hardware and Embedded Systems - CHES'2000. Lecture Notes in Computer Science, Vol. 1965. Springer-Verlag, Berlin Heidelberg New York (2000) 252–263
9. Tanenbaum A.S.: Computer Networks. Prentice Hall PTR (2002)
10. Chari S., Jutla C.S., Rao J.R., Rohatgi P.: Towards Sound Countermeasures to Counteract Power-Analysis Attacks. In: Wiener, M. (ed.): Advances in Cryptology - Crypto'99. Lecture Notes in Computer Science, Vol. 1666. Springer-Verlag, Berlin Heidelberg New York (1999) 398–412
11. Chari S., Rao J.R., Rohatgi P.: Template Attacks. In: Cryptographic Hardware and Embedded Systems - CHES'2002
12. Messerges T.S., Dabbish E.A., Sloan R.H.: Examining Smart-Card Security under the Threat of Power Analysis Attacks. IEEE Transactions on Computers, **51(5)** (2002) 541–552
13. Anderson, R., Kuhn, M.: Low Cost Attacks on Tamper Resistant Devices. In: Lomas, M. et al. (ed.): Security Protocols. Lecture Notes in Computer Science, Vol. 1361. Springer-Verlag, Berlin Heidelberg New York (1997) 125–136