

Side-Channel Leakage Tolerant Architectures

Kris Tiri
EE Department
UC Los Angeles, USA

Patrick Schaumont
ECE Department
Virginia Tech, USA

Ingrid Verbauwhede
Dept. ESAT/SCD-COSIC
K.U.Leuven, Belgium

Abstract

Side-channel attacks compare side-channel leakage predictions or estimations with side-channel leakage measurements. The estimations are based on the exact value of a few select state bits. If it is impossible to calculate the value of the state bits, it is also impossible to estimate the side-channel leakage and a side-channel attack cannot be mounted. The leakage can be measured, but cannot be estimated. In this paper, we first give an analysis of the size of the search space related to the smallest subset of state bits. Then we propose some techniques to make side-channel leakage tolerant architectures. These are architectures of which it is more difficult to calculate the value of the state bits. The architectures still leak side-channel data. Yet, the information content is less easy to exploit.

1. Introduction

The approach of a security attack is to break down the assumptions that are being made by the security information model. The enclosure that in traditional cryptanalysis surrounds the encryption and decryption activities holds a vital assumption. It should shield the activities and it should prevent anybody else to learn about the secret key and the plaintext message. This is an optimistic assumption, as there can be side-channels left through which an attacker can observe the information processing.

Side-channel attacks (SCAs) are non-intrusive attacks, which use information leaked through the side-channels to find the secret key [8]. SCAs are non-invasive and are directed at observing the device in normal mode of operation. They analyze information ranging from time delay [6] and power consumption [12] to electromagnetic radiation [1] and often apply advanced statistical techniques to reveal the secret information. SCAs are a real threat for any device in which the security IC is easily observable, such as smart cards and embedded devices [10],[16]. SCAs have been effective in extracting the key of microprocessor-, DSP-, FPGA- and ASIC-based encryption systems [13],[7],[16],[19].

SCAs are not a new practice. A classical example of a SCA is a safecracker who uses his fingers to feel the tumblers impacting each other while turning the dial. By observing when the lock's tumblers fall into place, he can crack the combination lock quickly and much faster than anyone who attempts to open the safe by trying every possible combination. In electronic circuits, the variations in power consumption can be used as the equivalent of the falling tumblers in a lock. The effect on the circuit's secure operation is devastating.

For example, a 109-bit symmetric key should be able to guarantee the confidentiality of data encrypted with such a key until the year 2050 [9]. With a power attack, however, it is possible to find the 128-bit key of an unprotected ASIC AES implementation in less than three minutes [19]. It shows that the implementation of a strong encryption algorithm does not suffice to guarantee strong security.

In short, side-channel attacks verify an assumption with a measurement. For instance, in a power attack, measured power traces are compared with estimations on the power consumption. Only if measurements and estimations are 'alike' has the correct key been used in the power model. The basic idea of all countermeasures is to degrade the signal-to-noise ratio of the measurements sufficiently such that the comparison and thus the attack become impossible.

For instance, to thwart power attacks, it has been proposed to add random power consuming modules [5], to randomize the execution [5], or to alter the power transfer [17]. Yet, these makeshift measures have all been labeled unsuccessful. Random power does not change the power consumption profile sufficiently [2]. The execution sequence can be resynchronized using integration techniques [3]. Active power filtering is likely to lag behind the fast power fluctuations and passive filtering, battery on chip and detachable power supplies are limited by physical dimensions [17].

Even the prevailing algorithmic countermeasures and hardware techniques only influence the measurements. Algorithmic countermeasures modify the algorithm to decorrelate the power consumption and the data being processed. An example of this is masking [2],[15]. In this approach, a random mask is added to the data and removed without

changing the encryption result. Hardware techniques change the behavior of the operations invoked by function calls. For instance at the gate level, specialized circuit styles and a place & route approach exist such that each individual gate has a quasi data-independent power dissipation [19].

These countermeasures only have an effect on the creation of side-channel information. In contrast, we propose to burden the power estimation. It is hard to make a power estimation if it is difficult to model the circuit, or in other words if it is difficult to predict a sub-state of the circuit. The main contribution of this paper is that we first show how state information and the size of the state space are used to build an estimation model. Then in the second part we point out how to build side-channel attack resistant devices not by implementing devices that do not create side-channel information but by using architectures that are side-channel leakage tolerant.

The remainder of this paper is organized as follows. The next section presents first the concepts behind a side-channel attack and subsequently the details of a differential power analysis attack on the AES algorithm. In section 3, side-channel leakage tolerant architectures are presented. We first derive a design for which it is ‘hard’ to estimate the power consumption in the sense that the search space has been increased. Based on the idea, we present subsequently a system for which it is impossible to estimate the power consumption. Finally, a conclusion will be formulated.

2. Side-channel attack

Side-channel attacks are based on the fact that CMOS logic and/or application specific details cause logic operations to have electrical characteristics that depend on the input data. They extract the secret key from the variations in the characteristics.

A side-channel attack makes an assumption on the information content of the side-channel leakage, or in other words on the observed variations. The attack estimates a part of the side-channel leakage with a model that requires a guess of a secret key subset. If the secret key hypothesis is correct, the estimated and the measured side-channel leakage are alike. Statistical techniques minimize the influence of measurement errors and modeling approximations during the comparison.

In a SCA, many measurements are compared with many estimations. After a sufficient number of measurements, a signal emerges from the noise, as the signal to noise ratio ideally increases with the square root of the number of measurements [13]. The side-channel attack resistance is quantified with the number of measurements to disclosure (MTD). This number expresses how many meas-

urements are on average necessary to correctly distinguish the correct secret key from all the other wrong key guesses.

Note that there is a class of attacks that use a single observation, such as for instance the simple power analysis. Yet, the attacks are not of great concern as they can rather easily be prevented with good design practices.

2.1 Differential power analysis

In a differential power analysis (DPA), the power consumption of a component of the encryption module is estimated through a behavioral model. The model calculates one or more state bits from known plaintext or ciphertext data and from a guess on a subset of the secret key. If the guess is correct, the outcome is always equal to the actual state bits and is therefore correlated with the power consumption of the logic operations that are affected by the state bits. Measurement errors and the power consumption of the other logic operations are uncorrelated.

The original DPA uses the distance-of-mean test to perform the comparison between the estimations and the measurements [11]. Another common DPA uses the correlation test [4]. In this case, a specific characteristic of the measurements, such as the mean or the maximum supply current in a clock cycle, is correlated with the power estimation.

2.2 DPA on AES

Figure 1 depicts a block diagram of the encryption datapath of an implementation of the 128-bit key, 128-bit data version of the AES algorithm. The AES core performs an encryption in 11 cycles, with one round of the algorithm executed per clock cycle. The key scheduling routine is not shown.

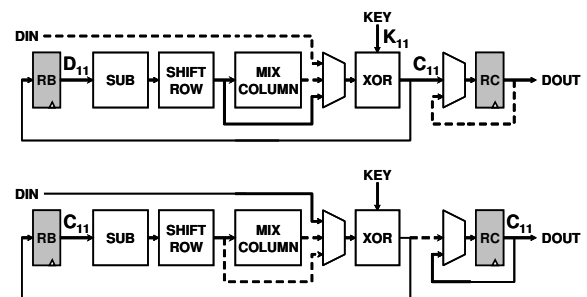


Fig. 1. Encryption datapath of AES: round 11 (top); and round 11 + 1 (bottom)

In a known ciphertext attack, the influence of register RB on the power consumption of the AES core is estimated. Register RB provides the input to and stores the output of each encryption round. The power estimation (P_{model}) is done by calculating the number of changing bits in register RB. The Hamming distance is calculated between the state of register RB in round eleven, which is the

final encryption round, and the state of RB in the round after that, in which the encrypted data is known. As shown in figure 1, RB in round eleven (D_{11}) can be found by tracing back the signal obtained after xor-ing the final ciphertext (C_{11}) and a key guess (K_{11}) through both the shift row operation and the substitution box. RB in the next round, during which we perform the supply current measurement ($P_{\text{measurement}}$), is the final ciphertext (C_{11}). The correct key is found by evaluating:

$$\max_{K_{11}} f_{\text{cost}}(K_{11}) = \text{corr}(P_{\text{measurement}}, P_{\text{model}}) \quad (1)$$

where $P_{\text{measurement}} = \max(I_{\text{supply}, 11+1})$
 $P_{\text{model}} = \text{HamDist}(D_{11}, C_{11})$
 $D_{11} = \text{sub}^{-1}(\text{shiftrow}^{-1}(K_{11} \otimes C_{11}))$

The cost function compares the estimations and the measurements with the correlation test. The correct key guess is the one that results in the highest correlation coefficient between the vector of Hamming distances and the vector of representative measurements.

AES has been designed with the limited resources of a typical 8-bit processor of a smart card in mind and most operations are byte oriented. Therefore, 8 state bits can be calculated using a guess on one key byte. By limiting the power estimation to the influence of a single byte of register RB, the secret key can be cracked byte per byte. Figure 2 displays the resulting simplified one byte encryption datapath in round eleven: C_{txt} is combined with K_{rnd} to predict P_{txt} . C_{txt} is one byte of the ciphertext C_{11} . K_{rnd} is a guess on one byte of the round key K_{11} . P_{txt} is one byte of the state D_{11} of register RB.

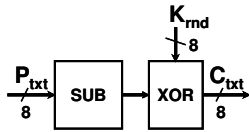


Fig. 2. Simplified one byte encryption datapath in round 11

Note that the DPA is still an exhaustive search. Yet a divide-and-conquer approach reduces the search space drastically. For example, a brute force attack on the AES algorithm requires 2^{128} key guesses to try all the possibilities of the 128-bit key. The DPA, working byte per byte, only requires $16 \cdot 2^8$ key guesses.

3. Side-channel leakage tolerant architectures

The power estimation results from a fairly simple model of the device. In principle, the attacker does not know the details of the implementation and performs a black box attack. As a result, it is next to impossible to incorporate much more accuracy into the model than the Hamming weight of a few select state bits or the Hamming distance between two successive values of a few select state

bits. Even if more information is available, such as the micro-architecture or even a detailed gate level netlist, is the exact value of a few select state bits still required to estimate the power consumption.

Note that it is perfectly possible to estimate the power consumption of a component without knowledge of the exact data that is being processed. This can for instance be done based on the component's silicon area. Such a power estimation may even be a very good average estimation. Yet it is useless to perform a side-channel power attack. The estimation is a constant, independent of the data and thus uncorrelated with the instantaneous variations in the measurements. Furthermore, a secret key cannot be cracked without a key hypothesis, which is not required in such a power estimation.

A side-channel attack relies on a power estimation that uses the exact value of a few select state bits. Consequently, if it is impossible to calculate the value of the state bits, it is also impossible to estimate the side-channel leakage. As a result, the side-channel attack cannot be mounted. The leakage can be measured, but cannot be estimated. Side-channel leakage tolerant architectures are architectures of which it is less easy to calculate the value of the state bits. The architectures create side-channel leakage. Yet, the information content is less easy to exploit.

3.1 Obstructing the power estimation

The attack in section 2.2 is a known ciphertext attack. The state P_{txt} of the circuit in figure 2 can be calculated given a guess on K_{rnd} because the ciphertext C_{txt} is known. Keeping C_{txt} secret makes P_{txt} incomputable. It is impossible to trace C_{txt} back to P_{txt} without knowing C_{txt} . Yet, the goal of the encryption operation is exactly to freely distribute the ciphertext, or in other words to make C_{txt} public.

A permutation achieves the desired effect of increasing the complexity of the state calculation. The permutation P is inserted between the output of the key addition C'_{txt} and the ciphertext C_{txt} (see figure 3). If the exact mapping of the permutation is unknown to the attacker, she/he cannot trace C_{txt} back to C'_{txt} . She/he can only make a guess on the permutation. As a result, the calculation of the exact state of the circuit becomes harder. Instead of a search space of 2^8 , the search space has ideally increased to $2^8 \cdot 8!$ ($\approx 2^{23}$). In addition to all the possible values of K_{rnd} , it is also necessary to traverse all the possible permutations P from C'_{txt} to C_{txt} and this for each value of K_{rnd} . The power estimation is not impossible but is obstructed.

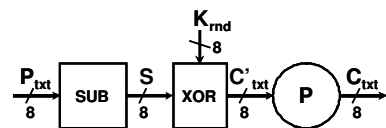


Fig. 3. Simplified one byte encryption datapath in round 11 with permutation P

Keeping the permutation secret, however, violates Kerckhoffs' principle that one cannot achieve security through obscurity. We will later see how to address this issue.

Note that there exists a K'_{rnd} equal to $P(K_{\text{rnd}})$ for which the key addition with C_{txt} results in S' . This signal S' is equal to $P(S)$, the permutation of the substitution box output $\text{sub}(P_{\text{txt}})$. S' has the same Hamming weight as S . The substitution, however, is not a linear operation. The Hamming weight at the output only provides limited information regarding the Hamming weight at the input and thus regarding the side-channel leakage caused by P_{txt} .

A power attack estimating the input S of the key addition will result in K'_{rnd} . Such an attack, however, is error prone and very noise sensitive as the diffusion of the substitution box is not used to increase the resolution of the correlation plot. Key hypotheses that differ only in one bit produce input signals that differ only in one bit. As a result, the Hamming weights of these inputs, which are used as the power estimations, are similar. Over time for the different switching events, the power estimations closely track each other. The correlation coefficients for the key hypotheses do not differ much and there is no large distinctive peak for the correct key guess.

Identifying K'_{rnd} simplifies the side-channel attack. With a power attack that estimates the input of the substitution box P_{txt} , the permutation can now be disclosed in $8!$ ($\approx 2^{15}$) hypotheses. Given that K'_{rnd} can be found, the search space only consists of all the permutations of S' in $\text{sub}^{-1}(S') = \text{sub}^{-1}(P(K'_{\text{rnd}} \otimes C_{\text{txt}}))$ to calculate P_{txt} .

Certain values of the ciphertext C_{txt} , i.e., the all zero and all one bytes, are not transformed through the permutation. In such a case, C_{txt} equals C'_{txt} . An attack on P_{txt} , which only selects these events, can ignore the permutation P . It would have a search space of 2^8 key guesses, but would require $2^8/2$ times more measurements if all events occur uniformly. Such an attack, however, does not work. The correlation coefficient is calculated using many measurements and estimations of only 2 events. Many key guesses result in the same pair of 2 power estimations. Given the fact that a DC offset does not change a correlation, the 256 key hypotheses result in only 11 different power estimation pairs. No key guess has an exclusive pair of power estimations.

Based on the preceding paradigm of the simplified one byte encryption datapath, a side-channel leakage tolerant architecture of the AES algorithm is shown in figure 4. The secret permutation P makes a known-ciphertext side-channel attack more computational complex. The permutation does not alter the encryption algorithm. The two operations are independent. The permutation has been cascaded with the encryption operation. It does not perform any encryption operation. The strength of the overall system comes from the AES algorithm.

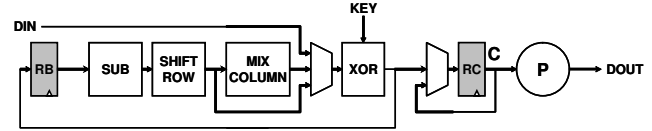


Fig. 4. Side-channel leakage tolerant AES datapath

The computational complexity of a side-channel attack is increased from $O(2^{12})$, i.e., $16 \cdot 2^8$ key guesses, to $O(2^{55})$. Suppose that the permuted 128-bit round key can be cracked through a power attack on the input S of the key addition. In that case, the permuted 128-bit output of the substitution box is also known. The exact permutation can now be found byte per byte through a power attack that estimates the input of the substitution boxes. From the 128-bit signal, one byte is chosen as the output of the first substitution box. The substitution box is not a linear operation and the 8 bits of the byte must be selected in the right order. As a result, the search space is $128!/(128-8)!$, which is the number of permutations of size 8 taken from 128 objects.

For completeness, figure 5 shows the side-channel leakage tolerant architectures for known plaintext; known ciphertext; and known plaintext and known ciphertext.

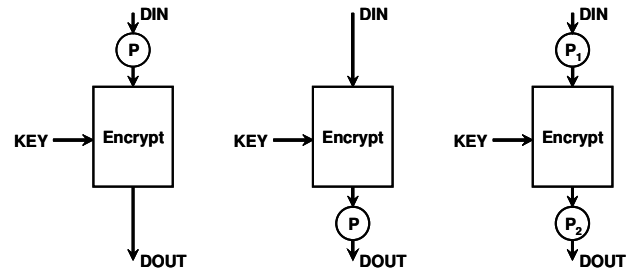


Fig. 5. Side-channel leakage tolerant architectures with permutation for: known plaintext (left); known ciphertext (middle); and known plaintext and known ciphertext (right)

The advantage of a fixed permutation is that for an ASIC implementation, it can be hardwired. It has no overhead. Once the permutation is known, however, the side-channel attack resistance of the design is compromised. The mapping of the permutation can be seen as a master key embedded into the design. Keeping the mapping secret is not an easy task. For instance, an attacker, which at one moment for a certain key has access to plaintext-data, ciphertext-data and the key, can easily find the permutation and can crack any other secret key on any other device with a side-channel attack.

A direct remedy is to make the permutation dependent on a permutation key (see figure 6). The operation P can be tailored in function of the desired computational complexity of a side-channel attack.

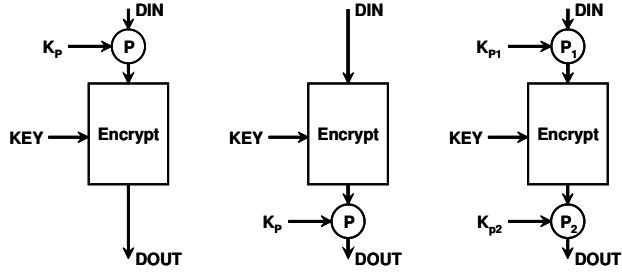


Fig. 6. Side-channel leakage tolerant architectures with key dependent permutation for: known plaintext (left); known ciphertext (middle); and known plaintext and known ciphertext (right)

To make the attack ‘hard’, it must be difficult to calculate the exact output of each substitution box. One way to implement a compact operation with this feature is shown in figure 7. P_i is a permutation from the i th byte of C_{txt} to the i th bit of each byte of C'_{txt} . A Waksman network of P_i [20], which is the permutation network comprised of the least number of switches, requires a mere 17 logic gates.

Suppose that the permuted 128-bit round key can be cracked through a power attack on the input of the key addition. In that case, the permuted 128-bit output of the substitution box is also known. The exact permutation can now be found byte per byte through a power attack that estimates the input of the substitution boxes. From the 128-bit signal, one byte is chosen as the output of the first substitution box. The substitution box is not a linear operation and the 8 bits of the byte must be selected in the right order. As a result the search space is $8^8 (=2^{24})$, which is the number of ways to chose the 8 bits from the 8 bytes.

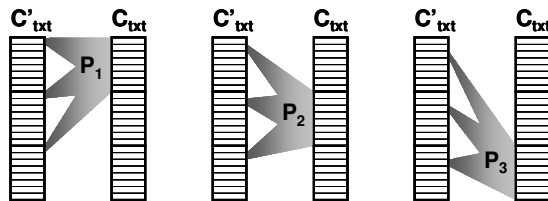


Fig. 7. Compact permutation network for power estimation obstruction

The architectures proposed in this section have no influence on the MTD. The number of measurements required to perform a successful attack has not changed. The information content in the power measurements is still the same. The search space during the power estimation, however, has been enlarged and the attack has become harder. The side-channel attack resistance has increased without increasing the MTD.

3.2 Preventing the power estimation

The information content cannot be exploited if the side-channel leakage cannot at all be estimated. Although

plaintext and/or ciphertext are known, it should be impossible to calculate a state of even a small component of the circuit without knowledge of the full key. There exists a well known mode of operation that has exactly this characteristic.

An encryption algorithm in Cipher Block Chaining (CBC) mode is secure against a known plaintext side-channel attack (see figure 8 top left). The value of the initial value IV is irrelevant. Suppose that IV is known. In that case, the state of the circuit can be calculated for the first encryption operation. It is, however, impossible to calculate the state of the circuit for the subsequent encryption operations. Starting from the second operation, the input to the central encryption unit is first xor-ed with the preceding output of the circuit. Since the attacker does not know the secret key, this output is an unknown. Only with a guess on the full key can the state of the circuit be calculated. As a result a side-channel attack has become as hard as a classical attack.

Figure 8 (top right and bottom) also shows the side-channel leakage tolerant architectures for known ciphertext; and known plaintext and known ciphertext.

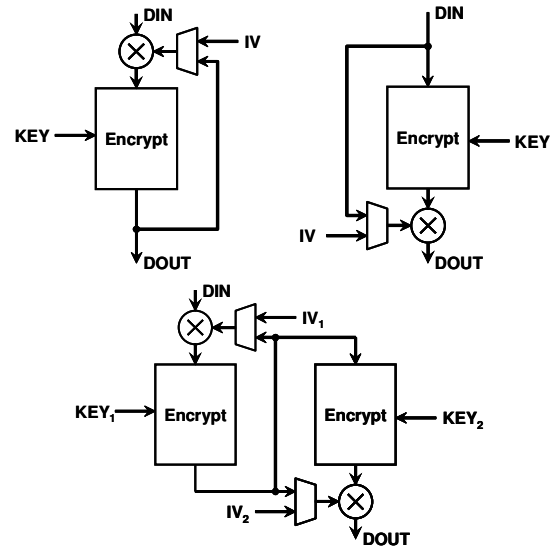


Fig. 8. Side-channel leakage tolerant architectures for: known plaintext (top left); known ciphertext (top right); and known plaintext and known ciphertext (bottom)

The state of a circuit in figure 8 depends on the initial conditions of the circuit. Without knowledge of each and every boundary condition, or in other words without knowledge of the secret key, can only a single state of the circuit be calculated. Increasing the number of measurements does not have any influence on the resolution of the correlation coefficient or the signal to noise ratio of the side-channel information as only a single comparison can be made with a side-channel leakage estimation. The attacker can make many measurements of the side-channel leakage but only one estimation.

If both ciphertext and plaintext are known, the side-channel tolerant architecture consists of a concatenation of the architectures for known plaintext and known ciphertext. It is not possible to implement an encryption operation with a combination of a single encryption unit, a feedback loop, and a feedforward loop that makes each component of the circuit only dependent on the initial conditions.

There is one constraint on the design in order to maintain the side-channel leakage tolerant feature. A session key cannot be used in more than MTD 'new' encryptions. MTD refers here to the resistance of the central encryption unit. A new encryption refers here to the case that the circuit is initialized before the encryption starts. If an attacker can initialize the circuit at will and perform each time a single encryption with the same key but a new input, she/he can collect many measurements while the state of the circuit can be estimated. Either the key must be updated regularly or the circuit must prohibit abundant initializations.

Another note of caution concerns the signal statistics of the ciphertext for the circuits of figure 8 (top left) and the signal statistics of the plaintext for the circuit of figure 8 (top right). The signals must be random enough such that the value cannot be predicted with great confidence. For instance, if it is known that the plaintext consists almost completely out of zeros, a side-channel attack can be mounted on the side-channel tolerant architecture for known ciphertext (figure 8 top right) by assuming the signal consist of only zeros. The few errors induced into the power estimation when tracing the ciphertext back through the xor-operation can be seen as measurement errors.

4. Conclusions

Side-channel attacks compare side-channel leakage estimations and measurements. Side-channel leakage tolerant architectures are architectures of which it is 'hard' or even impossible to calculate the value of the state bits and thus estimate the side-channel leakage. The architectures create side-channel leakage. Yet, the information content is less easy to exploit because of the large state space. We have shown that two different wrappers, a permutation and a feedback/feedforward mechanism, help to hide the state of the encryption unit. For sensitive, high security applications, protocols and systems can thus be developed such that the encryption algorithms are only used in modes of operation that are side-channel leakage tolerant instead of counting on the encryption unit itself to be side-channel attack resistant.

5. Acknowledgements.

This work was supported in part by the National Science Foundation (CCR-0098361) and UC Micro.

6. References

- [1] Agrawal, D., et al., "The EM Side-Channel(s)", CHES 2002, LNCS 2523, pp. 29-45, August 2002.
- [2] Chari, S., Jutla, C., Rao, J., and Rohatgi, P., "Towards Sound Approaches to Counteract Power-Analysis Attacks", CRYPTO'99, LNCS 1666, pp. 398-412, August 1999.
- [3] Clavier, C., Coron, J., and Dabbous, N., "Differential Power Analysis in the Presence of Hardware Countermeasures", CHES 2000, LNCS 1965, pp. 252-263, August 2000.
- [4] Coron, J., Kocher, P., and Naccache, D., "Statistics and Secret Leakage", FC2000, LNCS 1962, pp. 157-173, Feb 2000.
- [5] Daemen, J., and Rijmen, V., "Resistance Against Implementation Attacks: A Comparative Study of the AES Proposals", 2nd AES Candidate Conference, 1999.
- [6] Dhem, J., et al., "A practical implementation of the timing attack" CARDIS 1998, LNCS 1820, pp. 167-182, 1998.
- [7] Gebotys, C., "Design of secure cryptography against the threat of power-attacks in DSP-embedded processors", ACM TECS 3.1, pp. 92-113, February 2004.
- [8] Hess, E., et al., "Information Leakage Attacks Against Smart Card Implementations of Cryptographic Algorithms and Countermeasures - a Survey", Eurosmart, pp. 55-64, 2000.
- [9] Lenstra, A., and Verheul, E., "Selecting cryptographic key sizes," PKC, LNCS 1751, pp. 446-465, 2000.
- [10] Kocher P., et al., "Security as a New Dimension in Embedded System Design", DAC 2004, pp. 753-760, June 2004.
- [11] Kocher, P., Jaffe, J. and Jun, B., "Differential Power Analysis", CRYPTO 1999, LNCS 1666, pp. 388-397, Aug. 1999.
- [12] Mangard S., "A Simple Power-Analysis (SPA) Attack on Implementations of the AES Key Expansion" ICISC 2002, LNCS 2587, pp. 343-358, November 2002.
- [13] Messerges, T., Dabbish, E., and Sloan, R., "Examining smart-card security under the threat of power analysis attacks", IEEE TC 51.5, pp. 541-552, May 2002.
- [14] Oswald, E., Mangard, S., and Pramstaller, N., "Secure and Efficient Masking of AES - A Mission Impossible?", Report 2004/134 in IACR Cryptology ePrint Archive, June 2004.
- [15] Pramstaller, N. et al., "Towards an AES Crypto-chip Resistant to Differential Power Analysis", ESSCIRC2004, pp. 307-310.
- [16] Ravi, S. et al., "Tamper Resistance Mechanisms for Secure, Embedded Systems", VLSID 2004, pp. 605-610, Jan. 2004.
- [17] Shamir, A. "Protecting Smart Cards from Passive Power Analysis with Detached Power Supplies," CHES 2000, pp. 71-77.
- [18] Standaert, F., et al., "Power Analysis Attacks Against FPGA Implementations of the DES", FPL 2004, pp. 84-94, Sept. 2004.
- [19] Tiri, K., et al., "Prototype IC with WDDL and Differential Routing - DPA Resistance Assessment", CHES2005 pp. 354-365.
- [20] Waksman, A., "A Permutation Network", Journal of the ACM (JACM), vol. 15.1, pp. 159-163, January 1968.