Combining Algebraic and Side-Channel Cryptanalysis against Block Ciphers

Mathieu Renauld
UCL Crypto Group
Louvain-la-Neuve, Belgium
mathieu.renauld@uclouvain.be

François-Xavier Standaert
UCL Crypto Group
Louvain-la-Neuve, Belgium
fstandae@uclouvain.be

Abstract

This paper introduces a new type of cryptanalysis against block ciphers, denoted as algebraic side-channel attacks. In these attacks, we first write the target block cipher as a system of low degree equations. But since directly solving this system is generally hard, we additionally provide it with physical information. As a consequence, the algebraic cryptanalysis that was previously conjectured can be experimented and turns out to be very efficient to break block ciphers in practice. The proposed attacks differ from most previously known side-channel attacks in a number of interesting aspects. Namely they have a significantly reduced data complexity, the possibility to exploit the information of all the cipher rounds in an unknown plaintext/ciphertext scenario and different requirements for countermeasures. As an illustration, we apply them to the implementations of two block ciphers *using a single leakage trace* and discuss their specificities.

1 Introduction

Classical cryptanalysis generally considers adversaries getting black box access to the cryptographic primitives they target, *e.g.* the inputs and outputs of a block cipher. However, this ideal case is not always realistic because the actual implementations of a cryptosystem can leak physical information (like variations in the power consumption, electromagnetic emissions, ...) that can be measured by an adversary. These physical leakages generally contain indications on the successive states of a device and provide insights on the data and operations processed. Attacks exploiting them, *e.g.* in order to recover the secret key of a block cipher, are called side-channel attacks.

Side-channel attacks like the Differential Power Analysis (DPA) can lead to very efficient key recoveries, as demonstrated by Kocher *et al.* in 1999 [14]. In practice, they usually target the first (or last) rounds of a cryptosystem (where the diffusion is low) and try to recover the cipher key with a divide-and-conquer strategy. As these physical attacks directly target key bits, they are self-sufficient: once enough key bits are recovered with the side-channels, a simple exhaustive search can conclude the attack. The question we tackle in this work is whether it is possible to significantly reduce the amount of measurements to perform a key recovery by combining side-channel attacks with advanced cryptanalysis techniques. We answer this question positively and show that the algebraic cryptanalysis introduced by Courtois and Pieprzyk in 2002 [8] can exploit any kind of physical information leakages in a very natural and efficient manner.

Our attack is motivated by the idea of adapting the target of a side-channel attack to the leakage model of the device it exploits. On the one hand, the *target* of a physical attack is the information that we try to recover at the end of this attack. It can be very precise, like the secret key for standard DPA - or less precise, like the Hamming weights of some intermediate values in the case of our algebraic side-channel attacks. On the other hand, the *leakage model* is the type of dependencies that an adversary expects to observe in his physical measurements - hence, it is highly device dependent.

For example, with a Hamming weight leakage model (*e.g.* used in [3]) we assume that the information leakages will be correlated with the Hamming weight of the data processed in a device. Intuitively, the target directly determines the informativeness of a side-channel attack, *i.e.* the quantity of information gained thanks to the leakages. And of course, an informative target is always desirable. Our observation is that less informative targets may lead to more robust attacks if they closely correspond to the leakage model. That is, they can be recovered with high confidence using less measurements. As a consequence, if this information is still sufficient to perform a key recovery, *e.g.* using advanced cryptanalysis, it can be used to design attacks with reduced data complexity. The algebraic attacks introduced in this paper exactly exploit this intuition. They trade a less informative online measurement phase with a more powerful offline cryptanalysis phase. In practice, the question is then to determine if a high success rate can be reached both for the online and the offline phases (or to find the smallest amount of information that is required for the offline phase to be feasible).

In order to prove the validity of our approach, we performed exemplary attacks against the block ciphers PRESENT [4] and AES Rijndael. These experiments highlight a number of interesting features. First, our attacks potentially exploit the leakage of all the cipher rounds - classical DPA generally exploits the first or last rounds only. Second, they can succeed in an unknown plaintext/ciphertext adversarial context - classical DPA usually requires the knowledge of either the plaintexts or the ciphertexts. In common implementation contexts (*e.g.* assuming a Hamming weight leakage model), they recover the block cipher keys after the observation of a single encryption. Eventually, they can deal with block ciphers protected with countermeasures such as boolean masking. We mention that our experiments exploit a Hamming weight leakage model and target but in theory, any type of physical dependencies could be exploited. Of course, these attacks also have drawbacks. Mainly and as discussed later in the paper, they assume a strong adversarial context and require a precise profiling of the leakages.

Related works. Algebraic side-channel attacks can be related to several axes of research. First, their online phase exploits template attacks (*e.g.* [9, 18]). Second, their offline phase is directly derived from the algebraic cryptanalysis proposed by Courtois and Pieprzyk in [8]. Inspired from [2, 6], we use a SAT solver in order to process the system of equations and recover block cipher keys. Eventually, several other recent attacks combine side-channel information with classical cryptanalysis. We mention collision-based side-channel attacks, *e.g.* [15, 19, 20], techniques based on square attacks [7] and differential cryptanalysis [13]. These attacks have objectives similar to ours. They usually try to exploit the information leakages for more than the first block cipher rounds with advanced cryptanalysis. The goal is to break implementations for which only those rounds would be protected against side-channel attacks or to reduce the number of measurements required to perform a key recovery. We finally mention the recent and very efficient collision-based attacks of [5] that also use algebraic techniques in a more specific context and therefore closely connect to our proposal.

2 Description of the attack

Algebraic side-channel attacks are made of three separate steps. Due to space constraints, this paper only provides a high level description of these different phases. A more detailed technical description will follow in subsequent contributions.

2.1 Offline phase 1: algebraic description of the cryptosystem

The idea behind algebraic attacks is to describe a cryptosystem as a big system of low degree boolean equations. In this system, the main unknowns are the key bits so that solving the system is equivalent to finding the secret key. The system of equations can

be huge (thousands of equations and variables) but is often very sparse: the number of monomials in each equation is low. As an illustration, the AES Rijndael with 128-bit plaintext and key results in a system of approximately 18 000 equations in 10 000 variables (27 000 monomials). And the system for 31-round PRESENT corresponds to approximately 40 000 equations in 7000 variables (50 000 monomials). Note that the translation into a system of equations is not unique and the way we convert the cryptosystem into equations can greatly influence the efficiency of its solving.

2.2 Online measurement phase

The goal of this phase is to extract the side-channel information that is needed in order to solve the system of equations built during the previous phase. In our experiments, we considered implementations of PRESENT and the AES Rijndael on an 8-bit PIC microcontroller. The physical leakages of this device are typically correlated to the Hamming weight of the data transiting on the device bus, as shown in figure 1 (the bold traces represent the mean traces for the different Hamming weight values, and the grey traces are single measurements). As a consequence and as an illustration, we decided to use these Hamming weights as targets of our online phase.

In practice, a high success rate for the measurement phase is crucial for our attack to succeed. Indeed, one drawback of the algebraic cryptanalysis is its intolerance to erroneous information. That is, introducing even one error in our system of equations will generally cause the following offline solving phase to fail, without being able to point out the incorrect data. So we need to prevent such errors in the system.

To reach a high success rate, we first used a strong profiling of the leakages. This profiling consists in measuring several power traces for known plaintexts and keys on a device of the same type that the one we want to attack. With enough sample traces and with a good knowledge of the physical implementation, we can build a classification model using a maximum likelihood criteria. In other words, we can build templates in the sense of [9, 18]. And in order to perfectly adapt the target and the model in our attacks, these templates were built for every Hamming weight of a data transiting on the PIC bus. That is, we built 9 templates corresponding to 9 Hamming weight values. This strong profiling allowed us to recover the Hamming weight of a data transiting on the PIC bus with very high confidence. The success rate in recovering a single Hamming weight is close to 100%. Unfortunately, a single Hamming weight is not a sufficient information for our algebraic side-channel attacks to succeed. In fact, what we need is to recover the Hamming weights of as many intermediate data computed during a block cipher encryption as possible. But even if the probability of correctly assigning one Hamming weight is very high, the probability of correctly assigning several of them drops very rapidly, as shown in the right part of figure 1. Hence, we need solutions to further improve the success rate of the online phase and ensure that we can recover a sufficient number of Hamming weights with high success rate.

One solution is to simply drop the leakages with the highest probability of mistake, hoping that the algebraic side-channel attack can succeed with a fraction of the total leakages available. This is what is denoted as Likelihood Rating (LR) in the right part of Figure 1. It can be combined with simple Error Detection (ED) techniques, rejecting the Hamming weights that obviously lead to incompatible patterns in the equations. Another simple solution is to increase the number of measurements q (Figure 1 considers $q = 1, 2$). Clearly, such simple solutions are sufficient to reach high success rates in recovering the Hamming weights of multiple intermediate bytes during an encryption process. Finally, a more elaborated idea is to exploit less informative targets in case of dubious leakages. For example, if we cannot state the correct value of some weight $W_H(x_i)$ with enough probability, we can still use its corresponding leakage and recover a simpler target. For example, if $\Pr[W_H(x_i) = 3] \approx 93\%$ and $\Pr[W_H(x_i) = 4] \approx 7\%$, we can add to the system the information: $\Pr[W_H(x_i) = 3 \textbf{ or } 4] \approx 100\%$.

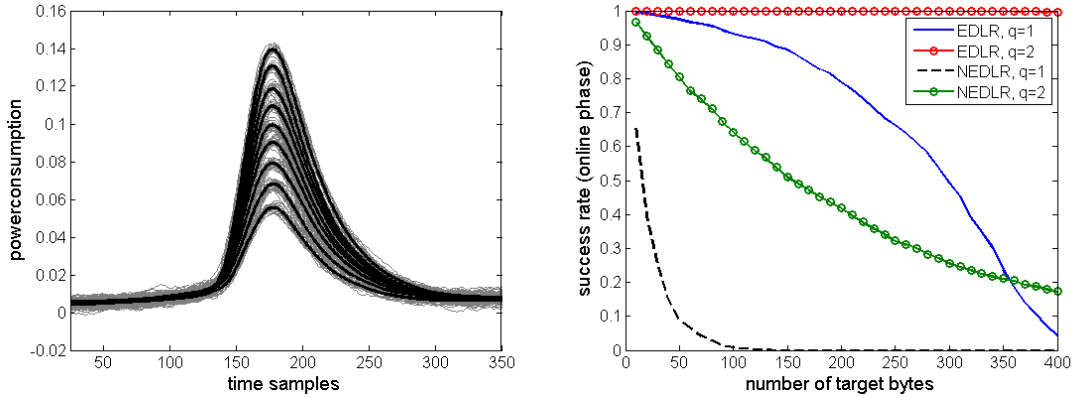

Figure 1: Leakage traces, mean leakage traces and multiple byte success rate.

2.3 Offline phase 2: SAT solving

Eventually, in the third phase of the attack, we first add the side-channel information recovered in Section 2.2 to the system of equations of Section 2.1. This is quite simple since any target (*e.g.* the Hamming weights in our context) can be viewed as a surjective function of the intermediate values in a block cipher encryption of which the output is known. Then, we try solving the resulting system. This solving can be done in many different ways, *e.g.* using the XSL technique [8], Gröbner basis-based methods [1] or using a SAT solver as in [6]. We selected this last solution.

For this purpose, we need to transform our system of equations into a satisfiability problem that can be stated as follows: given a boolean formula containing literals (x), negations of literals ($\bar{x}$), ANDs ($\wedge$) and ORs ($\vee$), is it possible to find an assignment of the literals so that the whole formula is equal to **true**. The satisfiability problem is the standard NP-complete problem, and is thus widely studied. Modern SAT solvers are more and more powerful (see [11] for a survey) and are able to efficiently solve a wide range of problems, even if they are not specifically designed for these problems. Methods for carrying out the translation of our problem into a SAT instance are described in [2]. Again, the conversion of a system of equations into a SAT instance is not unique. As shown in [12], the minimal conversion (*i.e.* the one using the shortest formula) is not always the best one in terms of solving time. A SAT solver often works better with useful redundancy in the formula. By useful redundancy, we mean the explicit description of internal properties that could be derived from the rest of the formula.

The success rate of the offline computation phase depends heavily on the quantity of information we can provide to the SAT solver. If we do not provide enough side-channel information, the solver will have difficulties to find a solution in a tractable time. By contrast, the solving time can be very short if we have enough information. There is thus a tradeoff between the complexity of the online measurement phase and the offline computation phase. A very demanding measurement phase (trying to gather very precise side-channel information) increases the number of measurements to perform, but in general, it facilitates the computation phase. In theory, the optimal target for our measurement phase should be the simplest target which still provides enough information to the solver. But of course, this notion of "enough information" essentially depends on how long one wants the SAT solver to run before considering that the experiment has failed (because it has not find a valid assignment, or detected an internal conflict that proves the formula unsatisfiable). Hence, we need to impose a time limit to the SAT solver. In our experiments, we assumed all the attacks that take over 1 hour of computation to be failures. This limit allowed us to evaluate the success rate of the offline computation phase as detailed in the next section.

3 Experimental results

We implemented various algebraic side-channel attacks on the block cipher PRESENT and the AES Rijndael with a 128-bit key. In both cases, we considered an implementation on an 8-bit microcontroller from which the Hamming weights of (possibly) all the intermediate computations could be recovered. Then, using a solving time limit of one hour, we computed the success rate of the offline phase in function of the amount of correct Hamming weights recovered by the adversary. As an illustration, an implementation of the AES in an 8-bit device usually exploits table-based substitution boxes and a MixColumn layer composed of several small operations, as described in [10]. For one round of the AES, we can consequently target a maximum of:

- 16 table lookups for the substitution layer,
- 36 XOR operations and 16 table lookups for the MixColumn layer,
- 16 XOR operations for the round key addition.

This amounts to a total of 84 Hamming weights leaked from one round, and 804 Hamming weights leaked from the whole 10-round AES*. Using this maximum amount of leakages (and similarly constructed bounds for PRESENT), we first performed experimental attacks in a known plaintext/ciphertext scenario, using *only one* power trace. For this purpose, we considered three possible contexts:

1. Knowledge of consecutive Hamming weights only, *i.e.* we assume that all the Hamming weights of consecutive rounds are known to the adversary, starting from the middle rounds (*e.g.* 3 rounds of W_H information mean that we know all the weights from rounds 4 to 7 in the AES). This is the most artificial situation since it prevents using the likelihood rating technique described in Section 2.2.
2. Knowledge of randomly distributed Hamming weights only, *i.e.* a more realistic context in which the Hamming weights obtained by the adversary correspond to intermediate values with randomly distributed positions in the cipher rounds.
3. Knowledge of randomly distributed Hamming weights + pairs of weights for the rest of the bytes, *i.e.* the same as (2), but we additionally assume the knowledge of a pair of weights including the correct one for the unknown weights.

Assuming the knowledge of all the Hamming weights, the success rate is 100% for a 32-round PRESENT, with an average solving time of 2 seconds and 97% for a 10-round AES, with an average solving time of 344 seconds. This clearly emphasizes the simpler algebraic structure of PRESENT compared to the AES Rijndael. It also shows that a higher algebraic complexity is not sufficient to prevent an algebraic cryptanalysis exploiting side-channel leakages. More interesting are the contexts with only partial information provided to the adversary for which the success rates are given in Figure 2. As expected, the success rate increases with the number of weights recovered by the adversary. It is worth noticing that a small number of consecutive weights in the central rounds (*e.g.* 3 rounds of Hamming weights for the AES, 4 for PRESENT) is sufficient to perform very effective attacks. Also, the success rate drops more rapidly when random weights are inserted in the system (compared to consecutive ones). Eventually, the additional knowledge of pairs of Hamming weights (*i.e.* context (3), experimented for the AES only) significantly improves the success rate. It highlights that any (even small) piece of information that can be learned with high confidence from the physics can be exploited - a useful observation since this scenario is frequent in practice.

*The last round has no MixColumn layer but one more round key addition.

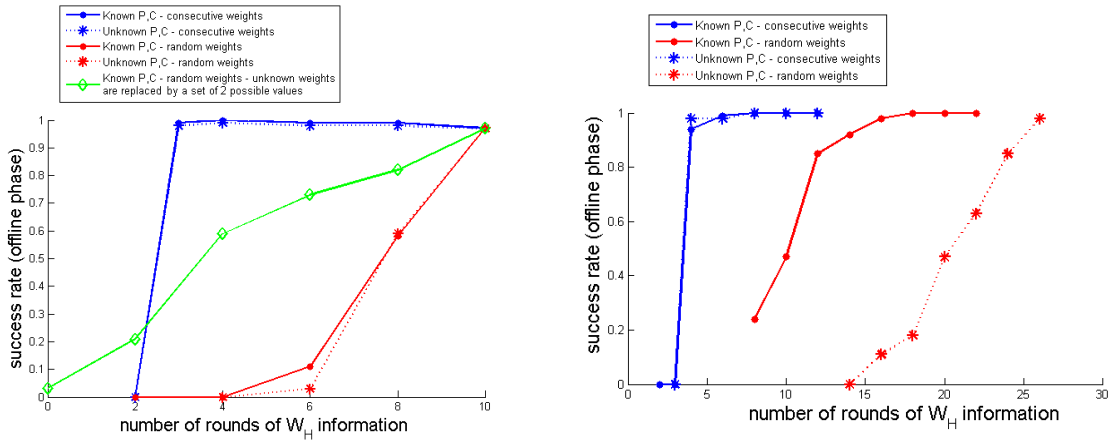

Figure 2: Experimental results: left: 10-round AES-128, right: 32-round PRESENT.

Unknown plaintext/ciphertext. The high success rate when exploiting consecutive information in the central rounds of a cipher suggests that an unknown plaintext/ciphertext scenario should not significantly affect the effectiveness of an algebraic cryptanalysis in this context. This is confirmed by Figure 2. By contrast, it has a significant impact on the success rate when attacking PRESENT with randomly distributed side-channel information. This difference between the two ciphers in this case is assumably due to the fact that for the AES, the information needed by the solver mainly comes from the MixColumn operation: provided we get enough leakages from it, the resolution succeeds regardless of the knowledge of the plaintext or ciphertext. By contrast, PRESENT has no such diffusion layer and its global algebraic structure is simpler: this allow successful resolutions with a smaller proportion of leakages, but the impact of the knowledge of P and C on the success rate is also stronger.

4 Countermeasures

Usual countermeasures against side-channel attacks intend to decrease the amount of information provided by the leakages. The goal is (as far as possible) to have the physical measurements independent of the data processed in a cryptographic device. Various ideas can be used for this purpose, including the addition of noise, the use of time and data randomizations, the design of logic styles with data-independent power consumption, ... Of course, countermeasures against side-channel attacks do not come for free and they generally imply a significant performance penalty, either in code size (or gate count), or in throughput. A central consequence of algebraic side-channel attacks is that they can exploit the leakage of any cycle in an implementation. Hence, more clock cycles generally imply more information leakages. This means that a countermeasure requiring a significant increase of clock cycles for performing an encryption could in fact decrease the security against algebraic cryptanalysis. This is in strong contrast with classical DPA attacks that only target the first/last round of a block cipher. In this section, we discuss this intuition in the context of masking and suggest generic countermeasures against algebraic side-channel attacks.

Masking is a well-known countermeasure designed to prevent certain types of side-channel attacks. It aims at de-correlating the power consumption of a device from the intermediate values it computes. For this purpose, any intermediate value that appears in a device is masked with a random value (unknown to the adversary). Overall, instead of computing the encryption of a plaintext p, one computes the encryption of a randomized plaintext $p \oplus m$ and tracks the evolution of the mask m through the block cipher execution. Hence, it requires a more complex implementation in order to propagate

the mask through the cryptosystem so that the mask can eventually be removed from the ciphertext. In practice, there exists numerous masking schemes published in the open literature, providing various tradeoffs between performance penalty and expected security improvement. We investigated two popular ones for the AES. The first one, presented in [16], uses the same 48-bit mask at each round to mask the data transiting on the bus. Hence, it can be implemented quite efficiently in an 8-bit controller, with only a reasonable increase of the number of clock cycles per encryption. The second masking scheme we considered was developed in [17]. It aims to increase the security of the countermeasure by increasing the number of random mask bits. In order to keep the memory requirements of the implementation reasonable, this solution implies the computation of numerous small operations (more than 10 XORs and 10 table lookups for each S-box). Therefore, while this additional performance penalty is supposed to increase the security against classical side-channel attacks, adding more masking bits actually decreases the security against algebraic ones. In fact, the implementation of [17] even turned out to be easier to break than an unprotected one.

As a consequence, while countermeasures against classical side-channel attacks mainly focus on reducing the amount of information leakage per clock cycle, preventing algebraic cryptanalysis additionally requires to limit the number of leaking cycles as much as possible. In addition to this new guideline, generic countermeasures include the use of block ciphers with high algebraic complexity and the use of implementations of which the leakages have high algebraic complexity (*e.g.* using large data buses like in FPGAs, time randomizations or noise addition, typically).

5 Conclusion and perspectives

This paper introduced algebraic side-channel attacks that trade a strong profiling phase for an optimized data complexity in the online extraction of physical information from cryptographic devices. We experimented first successful key recoveries against the block cipher PRESENT and the AES Rijndael, using the leakage corresponding to a single encrypted plaintext. These experiments imply a new understanding of certain countermeasures against side-channel attacks, like masking. They also highlight that the security against physical attacks cannot be understood independently of classical cryptanalysis issues. Eventually, this work raises several open problems. Determining the best tradeoff between the robustness of a leakage model and the informativeness of a target in a side-channel attack would allow to optimize the overall complexities of the combined online and offline phases in an algebraic cryptanalysis. The power of these attacks is also dependent on the quality of the SAT solver (or any other tool) to solve large systems of equations. Hence, finding the best solution with this respect would allow determining the limits of what a powerful side-channel adversary can achieve.

Acknowledgements. Work supported in part by the Walloon Region research project SCEPTIC. François-Xavier Standaert is an Associate researcher of the Belgian Fund for Scientific Research (FNRS - F.R.S.). The authors would like to thank Nicolas Veyrat-Charvillon for his useful advices about SAT solvers.

References

[1] G. Ars, J.-C. Faugre, H. Imai, M. Kawazoe, M. Sugita, *Comparison Between XL and Gröbner Basis Algorithms*, in the proceedings of ASIACRYPT 2004, LNCS, vol 3329, pp 338-353, Jeju Island, Korea, December 2004.

[2] G. Bard, N. Courtois, C. Jefferson, *Efficient Methods for Conversion and Solution of Sparse Systems of Low-Degree Multivariate Polynomials over GF(2) via SAT-Solvers*, Cryptology ePrint Archive, Report 2007/024.

[3] E. Brier, C. Clavier, F. Olivier, *Correlation Power Analysis with a Leakage Model*, in the proceedings of CHES 2004, Lecture Notes in Computer Science, vol 3156, pp 16-29, Boston, Massachussets, USA, August 2004.

[4] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, *PRESENT: An Ultra-Lightweight Block Cipher*, in the proceedings of CHES 2007, LNCS, vol 4727, pp 450-466, Vienna, Austria, September 2007.

[5] A. Bogdanov, I. Kizhvatov, A. Pyshkin, *Algebraic Methods in Side-Channel Collision Attacks and Practical Collision Detection*, in the proceedings of Indocrypt 2008, LNCS, vol 5365, pp 251-265, Kharagpur, India, December 2008.

[6] N. Courtois, G. Bard, *Algebraic Cryptanalysis of the Data Encryption Standard*, in the proceedings of 11th IMA International Conference, Lecture Notes in Computer Science, vol 4887, pp 274-289, Cirencester, UK, December 2007.

[7] V. Carlier, H. Chabanne, E. Dottax, H. Pelletier, *Generalizing Square Attack using Side-Channels of an AES Implementation on an FPGA*, in the proceedings of FPL 2005, pp 433-437, Tampere, Finland, August 2005.

[8] N. Courtois, J. Pieprzyk, *Cryptanalysis of Block Ciphers with Overdefined Systems of Equations*, in the proceedings of ASIACRYPT 2002, LNCS, vol 2501, pp 267-287, Queenstown, New Zealand, December 2002.

[9] S. Chari, J. Rao, P. Rohatgi, *Template Attacks*, in the proceedings of CHES 2002, LNCS, vol 2523, pp 13-28, California, USA, August 2002.

[10] J. Daemen, V. Rijmen, *The Design of Rijndael*, Springer 2002.

[11] J. Gu, P. W. Purdom, J. Franco, B. W. Wah, *Algorithms for the Satisfiability (SAT) Problem: A Survey*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pp 19-152, 1997.

[12] M. Heule, *Solving Edge-Matching Problems with Satisfiability Solvers*, in the proceedings of the Second International Workshop on Logic and Search (LaSh 2008), pp 88-102, Leuven, Belgium, November 2008.

[13] H. Handschuh, B. Preneel, *Blind Differential Cryptanalysis for Enhanced Power Attacks*, in the proceedings of SAC 2006, Lecture Notes in Computer Science, vol 4356, pp 163-173, Montreal, Canada, August 2006.

[14] P. Kocher, J. Jaffe, B. Jun, *Differential Power Analysis*, in the proceedings of Crypto 1999, LNCS, vol. 1666, pp 398-412, Santa-Barbara, CA, USA, August 1999.

[15] H. Ledig, F. Muller, F. Valette, *Enhancing Collision Attacks*, in the proceedings of CHES 2004, LNCS, vol 3156, pp 176-190, Cambridge, MA, USA, August 2004.

[16] S. Mangard, E. Oswald, T. Popp, *Power analysis attacks:*, Springer 2007.

[17] E. Oswald, K. Schramm, *An Efficient Masking Scheme for AES Software Implementations*, in the proceedings of WISA 2005, Lecture Notes in Computer Science, vol 3786, pp 292-305, Jeju Island, Korea, August 2005.

[18] F.-X. Standaert, C. Archambeau, *Using Subspace-Based Template Attacks to Compare and Combine Power and Electromagnetic Information Leakages*, in the proceedings of CHES 2008, LNCS, vol 5154, Washington DC, USA, August 2008.

[19] K. Schramm, G. Leander, P. Felke, C. Paar, *A Collision-Attack on AES: Combining Side Channel and Differential Attack*, in the proceedings of CHES 2004, LNCS, vol 3156, pp 163-175, Cambridge, MA, USA, August 2004.

[20] K. Schramm, T.J. Wollinger, C. Paar, *A New Class of Collision Attacks and Its Application to DES*, in the proceedings of FSE 2003, Lecture Notes in Computer Science, vol 2887, pp 206-222, Lund, Sweden, February 2003.