A Statistical Saturation Attack against the Block Cipher PRESENT

B. Collard[*], F.-X. Standaert[**]

UCL Crypto Group, Microelectronics Laboratory, Université catholique de Louvain,
Place du Levant 3, Louvain-la-Neuve, Belgium

baudoin.collard;fstandae@uclouvain.be

Abstract. In this paper, we present a statistical saturation attack that combines previously introduced cryptanalysis techniques against block ciphers. As the name suggests, the attack is statistical and can be seen as a particular example of partitioning cryptanalysis. It extracts information about the key by observing non-uniform distributions in the ciphertexts. It can also be seen as a dual to saturation (*aka* square, integral) attacks in the sense that it exploits the diffusion properties in block ciphers and a combination of active and passive multisets of bits in the plaintexts. The attack is chosen-plaintext in its basic version but can be easily extended to a known-plaintext scenario. As an illustration, it is applied to the block cipher PRESENT proposed by Bogdanov *et al.* at CHES 2007. We provide theoretical arguments to predict the attack efficiency and show that it improves previous (linear, differential) cryptanalysis results. We also provide experimental evidence that we can break up to 15 rounds of PRESENT with $2^{35.6}$ plaintext-ciphertext pairs. Eventually, we discuss the attack specificities and possible countermeasures. Although dedicated to PRESENT, it is an open question to determine if this technique improves the best known cryptanalysis for other ciphers.

Introduction

This paper introduces a statistical attack that is closely related to previous works in *partitioning cryptanalysis* [2, 8, 9]. Such attacks can be seen as a generalization of the linear cryptanalysis in which one exploits partitions of the plaintexts (*resp.* ciphertexts) leading to significantly non uniform distributions of the ciphertexts (*resp.* plaintexts). While arguably more powerful than linear cryptanalysis, they usually face the question of how to find good partitions for a given cipher. Hence, in practice they generally rely on some specificities that a cryptanalyst may find within the ciphers, *e.g.* in [7, 15]. Following these works, our results focus on a (relatively) generic and simple way to find partitions that can, in certain contexts, lead to efficient attacks. For this purpose, we exploit ideas from the *integral cryptanalysis* [13], originally introduced as a specialized attack against the SQUARE block cipher [6] and also known as saturation attacks [11]. Such attacks are chosen-plaintext and generally study the propagation of well chosen

[*] Work supported by the Walloon Region under the project Nanotic-Cosmos.

[**] Associate researcher of the Belgian Fund for Scientific Research (FNRS-F.R.S.).

sets of plaintexts through the cipher. In practice, they typically fix a number of plaintext bytes to a constant value and track the evolution of some other bytes having a known distribution. To some extent, the proposed statistical saturation attack can be seen as a dual of the previous saturation attacks. It also takes advantage of several plaintexts with some bits fixed while the others vary randomly. But instead of observing the evolution of the variable bits in the cipher state, we observe the diffusion of the fixed bits during the encryption process. That is, we track the evolution of a non-uniform input plaintext distribution through the cipher. The name statistical saturation attack refers both to the the way the inputs are generated and to the fact that it exploits the diffusion properties (and possibly weaknesses) of the target cipher.

As an illustration, we apply the proposed technique to the block cipher PRESENT that was presented at CHES 2007 by Bogdanov *et al.* It is a compact block cipher primarily designed for hardware constrained environments such as RFID tags and sensor networks. The name PRESENT reflects its similarity with the block cipher SERPENT [1], known for its security and hardware performances. In the specifications of the cipher [4], the authors analyze the security of PRESENT against various cryptanalytic attacks. In order to argue about the immunity against linear and differential cryptanalysis, they provide lower bounds for the number of active S-boxes in any linear/differential trail of the cipher. Resistance against structural, algebraic and related-key attacks is also analyzed. The security of PRESENT against differential cryptanalysis was further studied in [17] in which the authors present an attack against 16 rounds that requires the entire codebook and a time complexity of 2^{65} memory accesses.

PRESENT is a good target for the proposed statistical saturation attack because it exhibits a particular weakness in its diffusion layer. As a consequence, our following results improve the complexities of the best reported attacks against this cipher, both in theoretical estimations and in experimental validations. In practice, we broke up to 15 rounds of PRESENT with $2^{35.6}$ plaintext-ciphertext pairs. Additionally to these results, we discuss the specificities of the attack compared to other known cryptanalytic techniques. We show that it depends both on the diffusion and substitution layers in a block cipher. We also show that current criteria for S-box design do not properly capture the non-uniformities that are exploited in our partitions. Due to the generality of its principles, the proposed technique could be applied to other ciphers as well. However, since its effectiveness depends on the diffusion properties of the targets, it is an open question to determine if it can improve other cryptanalytic results.

The rest of this paper is divided in three parts. Section 1 presents the basic principles of our attack with theoretical arguments that support it. A comparison between theoretical predictions and experimental observations is also provided. The second section extends the basic profiling attack of Section 1 to a distinguishing attack that is more efficient, both in terms of computations and data complexity. Section 3 discusses countermeasures and the impact of the S-boxes on the attack performances. We conclude the paper and suggest further research in Section 4. The PRESENT block cipher is additionally described in Appendix.

1 A basic profiling attack

1.1 Principle of the attack

Our attack is based on a weakness in the diffusion layer of PRESENT. A closer look at the permutation shows that, *e.g.* for the S-boxes 5,6,9 and 10 (counting from S-box 0 at the right), only 8 out of 16 input bits are directed to other S-boxes. Figure 1 illustrates this observation. Note that there exists many other examples of poor diffusion in the permutation (*i.e.* with 8 bits out of 16 remaining in the same 4 S-boxes after permutation). Consequently, if we fix the 16 bits at the input of the S-boxes 5-6-9-10, then 8 bits will be known at the very same input for the next round. We can iteratively repeat this process round by round and observe a non-uniform behavior at the output of the S-boxes 5-6-9-10.

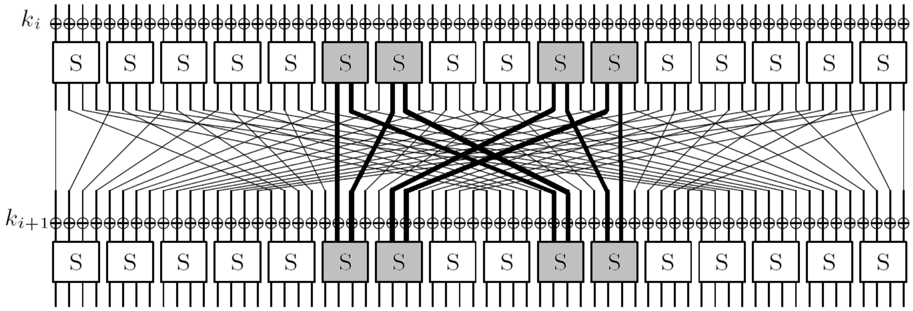

Fig. 1: Permutation layer of PRESENT: bold lines underline the poor diffusion property.

In order to exploit this weakness, we first evaluate theoretically the distribution of the 8 bits in the bold trail of Figure 1 at the output of the S-box layer, for a fixed value of the same 8 bits of plaintext. This requires to guess the 8 subkey bits involved in the trail. One also needs to assume that the bits not situated in the trail are uniformly distributed. This is a reasonable assumption as soon as the 56 remaining bits of plaintext (excluding the 8 bits in the trail) are randomly generated. Then, given the distribution of the 8-bit trail at the input of a round, it is possible to compute the 8-bit distribution at the output of the round with Algorithm 1 (given in Appendix B). By iteratively applying this algorithm, we can compute the distribution for an arbitrary number of rounds. For each key guess, the work needed to compute the theoretical distribution of the target trail after r rounds is equivalent to $r \cdot 2^{16}$ partial encryptions.

Once we have computed the theoretical distributions of the trail for each possible key guess, we can attack the cipher by simply comparing them with a practical distribution obtained by encrypting a large number of plaintexts with a secret key. The key guess minimizing the distance between theoretical and practical distributions is chosen as the correct key. As in [3], we can construct a list of key candidates sorted according to the distance between theory and practice. The better the position of the right key in the list, the better the attack.

1.2 Experimental results

We evaluated the practicability of our attack a against reduced-round version of PRESENT. In order to reduce the guess work, the key-scheduling of PRESENT was simplified in these experiments and the same subkey was used at each round. With this modification, only 8 bits of the master key have to be estimated and the correct distribution has to be found among 256 possible ones.

Comparison between theoretical and experimental distributions. Figure 2 depicts the distribution of the 8 bits in the trail after 2,4,6 and 8 rounds for a fixed 8-bit key byte. The theoretical predictions (in black) are compared with experimental results (in grey) generated with 2^{30} plaintexts-ciphertexts pairs. Note that our attack is choosen-plaintext as we have to fix 8 plaintext bits. But it can be turned into a known-plaintext attack by dividing random plaintexts in 256 classes according to the value of the 8 fixed input bits in the trail and observing the output distributions for each of the 256 cases independently.

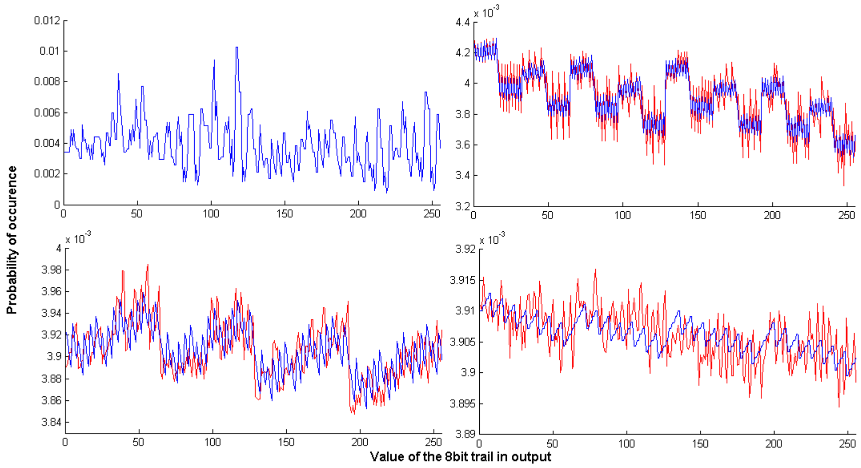

Fig. 2: Comparison between the experimental (in gray) and theoretical (in black) distributions of the target trail output for a given key byte and 2, 4, 6 and 8 rounds.

Both experimental and theoretical distributions present a significant deviation from uniform, even after 8 rounds. The deviation tends to decrease with the number of rounds however. We can observe that predictions match experiments very closely for up to 6 rounds, and then begin to distinguish. This illustrates that the sampling is not sufficient anymore to approximate the distributions.

Comparison between theoretical and experimental distances. Figure 3 shows the evolution of the distance between the distribution corresponding to a secret key byte 32 and the distributions corresponding to the 256 possible key guesses for this secret key byte. The number of rounds in the figure again varies from 2 to 8. The black (*resp.* grey) curves represent the distance between the theoretical (*resp.* experimental) distribution of the correct key byte and the theoretical distributions for each possible key guess. For up to 8 rounds, we observe that position 32 minimizes the distance between theory and practice. Note that we used both an Euclidean and a Kullback-Leibler distance in our experiments: both metrics gave similar results.

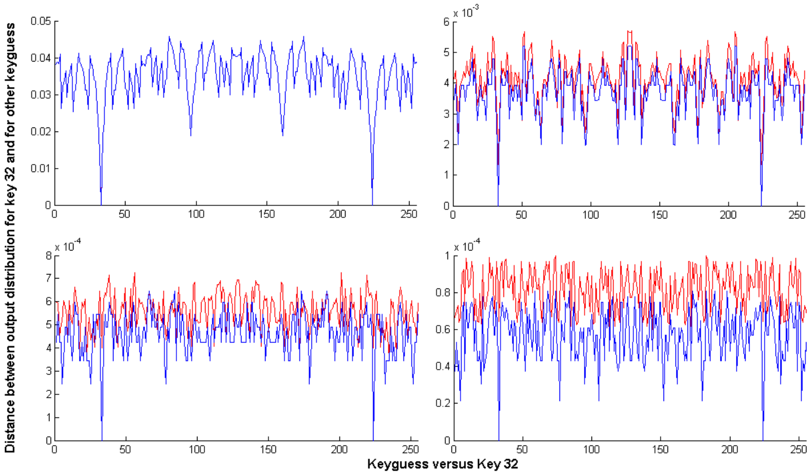

Fig. 3: Distance between the experimental (in gray) and theoretical (in black) distribution of a secret key byte and the theoretical distributions of the 256 possible key guesses. 2^{30} plaintexts were used for the experimental results.

In order to confirm the effectiveness of the proposed cryptanalysis in a key recovery context, we also computed the gain of the attack, as defined in [3]:

Gain the attack. *If an attack is used to recover an n-bit key and is expected to return the correct key after having checked on the average M candidates, then the gain of the attack, expressed in bits, is defined as:*

$$\gamma = -log_2 \frac{2 \cdot M - 1}{2^n} \tag{1}$$

Intuitively, the gain is a measure of the remaining workload (or number of key candidates to test) after a cryptanalysis has been performed. In the context of our attack, we can produce a list of key candidates sorted according to the distance between their theoretical distribution and the experimental distribution computed with the correct secret key. The gain is simply determined by the position of the secret key in this list. Figure 4 shows the gain of the attack for 1 to 14 rounds of PRESENT (still with a modified key-scheduling) in function of the data complexity. This experiment used up to 2^{30} plaintext-ciphertext pairs. The gain is bounded by 8, as we guess 8 bits of key material. It increases with the number of texts and decreases with the number of rounds.

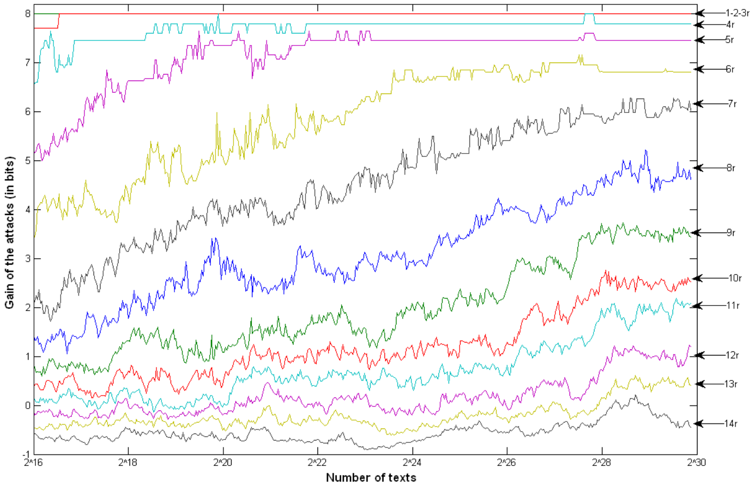

Fig. 4: Gain of the profiling attack against 1- to 14- round PRESENT.

Effect of the key-scheduling algorithm Unlike slide and related key attack, the proposed technique does not use a particular weakness in the key-scheduling. However, the number of subkey bits to guess at each round directly affects the time complexity of the attack, as one must compute the theoretical distribution for each of these key guesses. It may also increase the data complexity as distinguishing more keys generally requires more text pairs. The number of bits to guess according to the number of rounds is given in Table 1 for the complete key-scheduling algorithm. After 12 rounds, we have to guess 63 bits of the key, for a complexity equivalent to $2^{63} \times 2^{16} = 2^{79}$ encryptions. Consequently, this bounds the interest of the profiling attack to 12 rounds of the (simplified) cipher.

$\#rounds$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
$\#bits$	8	15	24	31	35	39	43	47	51	55	59	63	65	67	69	71	73	75	77	79	80

Table 1: Number of bits to guess according to the number of rounds in the attack.

2 A distinguishing attack

In order to get rid of the previous computational limits, we now present a variant of the attack. It has the advantage of not requiring the precomputation of theoretical distributions anymore. The distinguisher is based on the fact that the theoretical distribution at the output of the target trail (as computed with Algorithm 1) is significantly different from uniform, whatever subkey is used.

2.1 Principle of the attack

The attack is similar to the one presented above, yet it is simpler. We begin by generating a large number of plaintexts with 8 fixed bits. We encrypt those plaintexts using r-rounds PRESENT and record the distribution of the ciphertexts for the 16 bits at the output of the 4 active S-box in the last round. Given this experimental distribution, it is possible to compute the output distribution of the target 8-bit trail one round before by a classical partial decryption process. For one key guess, the evaluation of such an $r-1$-round distribution requires 2^{16} computations. Hence the total time complexity for all the key guesses equals $2^{16} * 2^{16} = 2^{32}$. Additionally using an FFT-based trick similar to the technique presented in [5], this complexity can be decreased to $16 \cdot 2^{16}$. For the correct key guess, the experimental 8-bit distribution in the penultimate round is expected to be more non-uniform than for any other guess. This is because decrypting with a wrong guess is expected to have the same effect as encrypting one more round. We can thus hope to distinguish the correct key from the wrong ones by computing the distance between a partially decrypted distribution and the uniform distribution. If the attack works properly, the distribution with the highest distance should correspond to the correct key.

2.2 Extensions of the attack

(ext. 1) Increase the fixed part in the plaintext. One can easily gain one round in the attack by simply fixing the 16 bits of plaintext corresponding to the 4 active input S-boxes of the trail. This way, the 8-bit trail in the second round is also fixed and the diffusion is postponed by one round. By fixing 32 bits out of 64 (corresponding to S-boxes 4-5-6-7-8-9-10-11), one can similarly extend the attack by 2 rounds. However, we are then limited in the generation of at most 2^{32} texts. This limitation may be mitigated with the following extension.

(ext. 2) Use multiple fixed plaintext values. The same analysis can be performed multiple times, using different values for the 8-bit (or 16- or 32-bit) fixed part of the plaintexts and then combining the results (*e.g.* taking the sum of the uniform *vs.* measured distances corresponding to the different fixed plaintexts). This allows exploiting more texts and moving to a known-plaintext context. The resulting attack is similar to multiple linear cryptanalysis: each fixed part of the plaintext can be seen as analogous to an additional approximation in [3, 12].

(ext. 3) Partial decryption of two rounds instead of one. In this case, 8 S-boxes are active in the last round instead of 4. Therefore, we have to keep a 32-bit distribution table in memory. Additionally, 38 bits of the key must be guessed for the partial decryption (32 bits for the last round + 16 bits for the penultimate round − 10 bits that are redundant). Using this trick, the adversary has to distinguish an $r-2$-round distribution for the correct key from an $r+2$-round distribution for the wrong candidates. The time complexity would be $(32 \cdot 2^{32}) \cdot (16 \cdot 2^{16}) = 2^{57}$ using again the results in [5].

2.3 Experimental results

We have run experiments against reduced-round versions of PRESENT with up to 15 rounds and evaluated the gain of the attack in different contexts. First, Figure 5 represents the mean result of 4 attacks using 2^{34} plaintexts where 16 input bits were fixed (*i.e.* using only **ext. 1**).

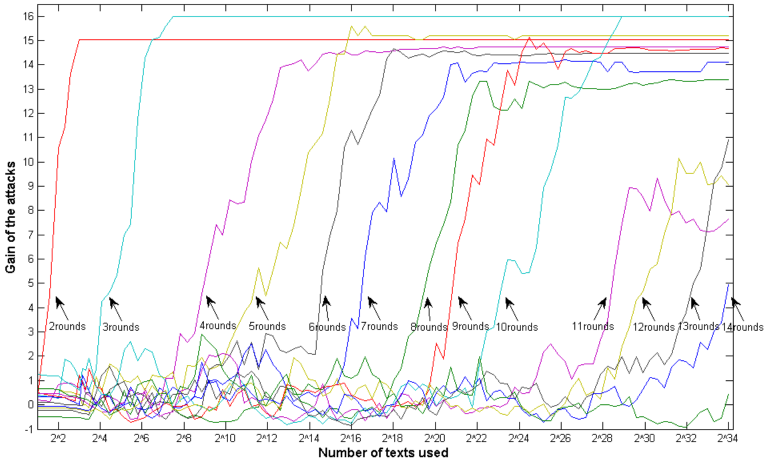

Fig. 5: Average gain of 4 attacks against 2 to 15-round PRESENT (**ext. 1**, 16 fixed bits).

To confirm the intuition that non-uniform distributions are observed for the correct key candidate, we represented the distance between the experimental distributions of the trail after partial decryption using a correct key and a uniform distribution. Figure 6 illustrates that this distance decreases with the number of rounds and stabilizes after a sufficient number of plaintexts have been reached.

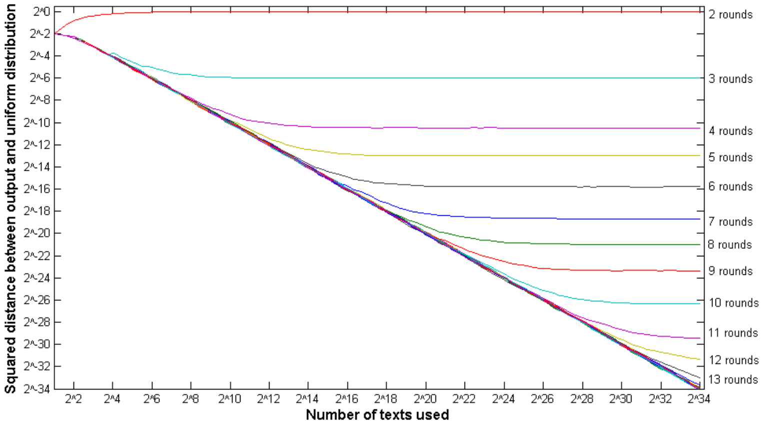

Fig. 6: Distance between uniform and output distributions after 1 to 15 rounds.

Figure 7 illustrates the results of a variant of the attack where 32 plaintext bits were fixed and consequently only 2^{32} texts were generated. As expected, the results are slightly better than in the previous experiment.

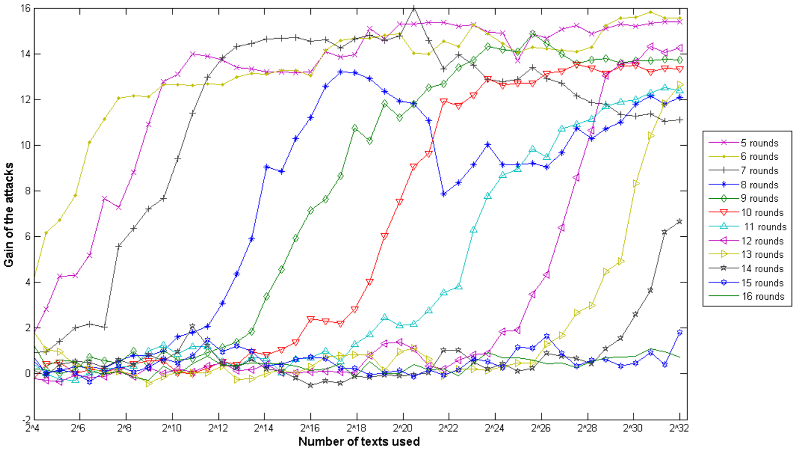

Fig. 7: Avg. gain of 12 attacks against 5 to 16-round PRESENT (**ext. 1**, 32 fixed bits).

Figure 8 finally shows an application of **ext. 2**. The graph represents the evolution of the attack gain against 1 to 32-rounds after 2^{32} plaintexts. The top and bottom curves represent the maximum and minimum gains among 12 experiments, while the two other curves represent respectively the average gain and the gain of the attack combining the 12 experiments. We clearly observe that combining the distances corresponding to the 12 experiments and computing a list of key candidates afterwards gives rise to much better result than computing a list for each experiment and then taking the average position for each key guess. Using the first method, we reach a significant gain up to 15 rounds.

For discussion purpose, Figure 13 in appendix C represents the gain of a linear cryptanalysis against 6- to 16-rounds PRESENT. The attack is based upon an iterative approximation involving one S-box with bias 2^{-2} in the first round and one S-box with bias 2^{-3} in each other round. It can recover up to 12 bits of the last subkey. Note that this example is not given for comparing the efficiencies of different attacks but to illustrate the big difference between security bounds as provided for the "best possible" linear attacks in [4] and attacks based on approximations that can be found in practice. The one that we proposed in appendix is obviously not optimal, but it is exploits the same iterativeness as our statistical saturation attack. As a matter of fact, the attacks we discuss in this paper are experimented and therefore cannot be straightforwardly compared with bounds. They more directly relate to actual attacks such as presented in [17].

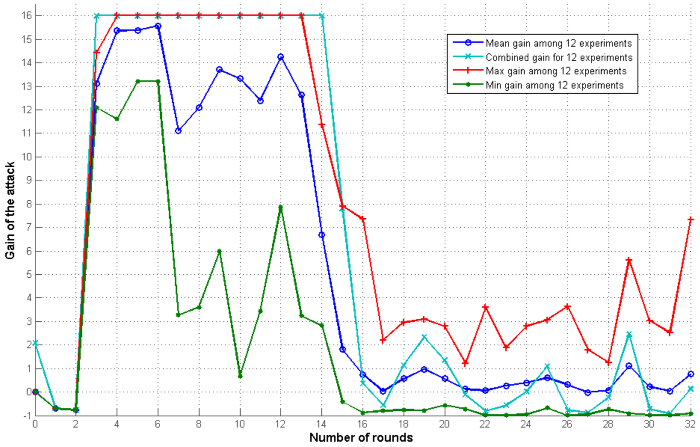

Fig. 8: Gain of attacks against 1- to 32-rounds PRESENT with $12 \cdot 2^{32} = 2^{35.6}$ texts.

3 Theoretical complexity

Intuitively, the efficiency of our distinguisher depends on the extent to which an experimental distribution after r-rounds PRESENT can be distinguished from a uniform distribution. Therefore, it can be nicely related to the theoretical analysis of Baignères *et al.* in [2] which shows that the data complexity required to distinguish two distributions is proportional to the inverse of the squared Euclidean distance between these distributions. Using Algorithm 1, we can easily compute a theoretical approximation of this Euclidean distance for PRESENT. It directly gives rise to Figure 9 in which the complexity of distinguishing the theoretical distributions at the output of PRESENT from uniform distributions is given for 1 to 16 rounds. We also illustrate the complexity of a linear cryptanalysis using the same approximation as the attack in Appendix C. Again, the difference between the effectiveness of an actual linear attack as in this figure and the security bounds in [4] should be emphasized. But even these security bounds (*e.g.* 2^{84} plaintext-ciphertext pairs to break 28-rounds PRESENT) suggest that our attack is an improvement compared to the theoretical expectations.

More interesting are the results in Table 2 that summarize the complexity of the attacks against PRESENT known so far (*i.e.* mainly [17] and the results in this paper). Note that due to the iterative nature of our trail, the time and memory complexities do not vary with the number of rounds in the trail. They only depend on the number of rounds that are partially decrypted. Most importantly, the provided data complexities are based upon the theoretical values given by the graph in Figure 9 and rely on the following assumptions:

- All attacks use **ext. 1** with 32 plaintext bits fixed.
- In 1-round (*resp.* 2-round as suggested in **ext. 3**) decryption attacks, we use a $r - 3$-round (*resp.* $r - 4$-round) distinguisher and have the time and memory complexities discussed in Sections 2.1 and 2.2.

- When the number of plaintexts needed to perform the attack exceeds 2^{32}, we use **ext. 2**. By combining multiple fixed plaintext values, we consider an attack exploiting distributions of larger dimensions, similarly to the multiple linear cryptanalysis in [3]. But it is an open problem to determine exactly the effect of this extension to the attack complexities. At least, our experiments suggest that these estimations are valid up to 16 rounds.

Note that our attack only recovers 16 key bits while [17] recovers the whole key. But as mentioned in Section 1.1, similar trails could be used to recover 32 more key bits with similar complexities. Hence, our results (including the part confirmed experimentally) anyway improve the best reported attack considerably.

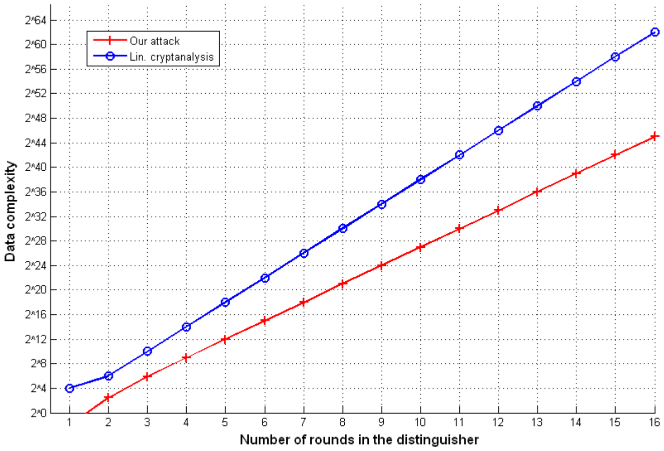

Fig. 9: Theoretical data complexity of the attack against PRESENT.

$\#rounds$	type of attack	data compl.	time compl.	memory compl.	gain	reference
16	Diff. Crypt.	2^{64}	$2^{65} MA$	$6 * 2^{32} bits$	≤ 32	[17]
8	our attack*	$c * 2^{12}$	2^{20} op.*	2^{16} counters	≤ 16	this paper
	our attack**	$c * 2^{9}$	2^{57} op.*	2^{32} counters	≤ 38	this paper
12	our attack*	$c * 2^{24}$	2^{20} op.*	2^{16} counters	≤ 16	this paper
	our attack**	$c * 2^{21}$	2^{57} op.*	2^{32} counters	≤ 38	this paper
16	our attack*	$c * 2^{36}$	2^{20} op.*	2^{16} counters	≤ 16	this paper
	our attack**	$c * 2^{33}$	2^{57} op.*	2^{32} counters	≤ 38	this paper
20	*our attack**	$c * 2^{48}$	2^{20} op.*	2^{16} counters	≤ 16	this paper
	our attack**	$c * 2^{45}$	2^{57} op.*	2^{32} counters	≤ 38	this paper
24	*our attack**	$c * 2^{60}$	2^{20} op.*	2^{16} counters	≤ 16	this paper
	our attack**	$c * 2^{57}$	2^{57} op.*	2^{32} counters	≤ 38	this paper

* 1-round decryption, ** 2-round decryption

Table 2: Summary of attacks (italic are not experimented and use **ext. 2**).

4 Countermeasures and influence of the S-box

The origin of the proposed statistical saturation attack against PRESENT mainly lies in a weakness of the diffusion layer. A straightforward countermeasure would be to modify the permutation in order to avoid poor diffusion in any subset of S-boxes. But the proposed attack relates to the overall diffusion properties of the cipher. Hence the S-boxes also have an impact with this respect that we shortly study in this section. To do so, we generated 5000 different S-boxes respecting the four conditions imposed in the generation of the PRESENT S-box (see [4], Section 4.3). According to the authors, these constraints ensure that PRESENT is resistant to differential and linear attacks. Figure 14 in Appendix D represents the evolution of the squared distance between the uniform and output distribution of the cipher according to the number of rounds. Each curve represent a different choice for the S-box used in the cipher. It is noticeable that the PRESENT S-box is among the worst possible choices to resist our attack.

To confirm this impact of a weak *vs.* strong S-box in our cryptanalysis, we finally ran new experiments against a tweaked PRESENT where the original S-boxes were replaced by the dashed S-boxes of Figure 14 (*i.e.* those corresponding to the best and worst diffusion properties among our 5000 generated S-boxes). Figure 10 gives the gain of these attacks for different number of rounds (each attack used 2^{30} chosen plaintexts). As expected, the attack against the weak version of the cipher gives the best results. The figure emphasizes that the proposed attack is not directly related to linear or differential cryptanalysis (*i.e.* it is possible to find a cipher that is immune against linear and differential cryptanalysis, but not against the proposed statistical saturation attack).

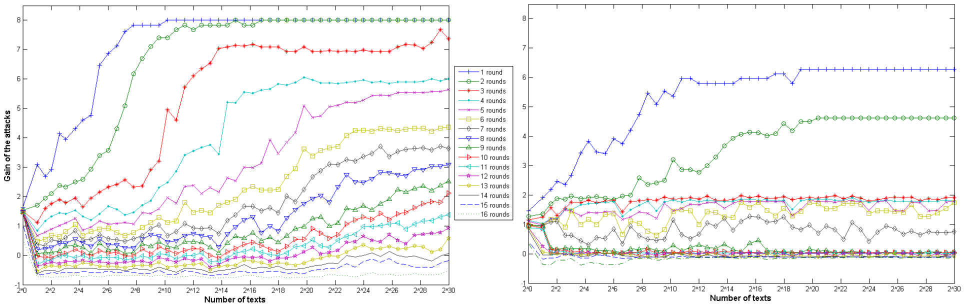

Fig. 10: Comparison between weak (left) and strong (right) S-boxes.

5 Conclusion and further works

In this paper, we presented a new attack against the block cipher PRESENT that improves previously known cryptanalyses against this cipher. Experimentally, it allows us to break 15 rounds with $2^{35.6}$ plaintext-ciphertext pairs. We also present theoretical estimations of attacks than can break more cipher rounds. Additionally, we show that the proposed cryptanalysis is not directly related to linear and differential attacks. In practice, the security of the full cipher does not seem to be compromised by our results although the proposed attack was not discussed in the algorithm specifications. However, it confirms and emphasizes that PRESENT has been designed with little security margins.

Determining if the proposed statistical saturation attacks can improve cryptanalytic results against other ciphers is an interesting open question. Since they only exploit very general principles (namely, uncomplete diffusion after some cipher rounds), they are likely to be applicable to other reduced algorithms. But on the other hand, the proposed attack was particularly efficient against PRESENT due to a weakness in its permutation. Hence, it is not clear if the proposed technique can be as effective against other ciphers, with better diffusion properties. More theoretically, a better theoretical analysis of the attack, in particular the analogy between multiple linear cryptanalysis and the use of multiple fixed plaintext bytes in our context is also worth further investigation. Eventually, it would be interesting to investigate if the multidimensional cryptanalysis presented in [10] could be used to improve our results.

Another research direction would be to use the trail of Figure 11 in which 27 bits out of 36 are redirected to only 9 S-boxes at each round. It means that the lack of diffusion could be worse than in the trail of Figure 1 (we found 4 trails of this kind). However, this weakness is compensated by a larger trail size (36 bits instead of 16) which increases the diffusion inside the trail. Applying the attack presented in this paper to this new trail is also more difficult to experiment because of the 36-bit distributions for which a 1-round decryption would have to be performed. Hence, the efficiency of this attack is an open question.

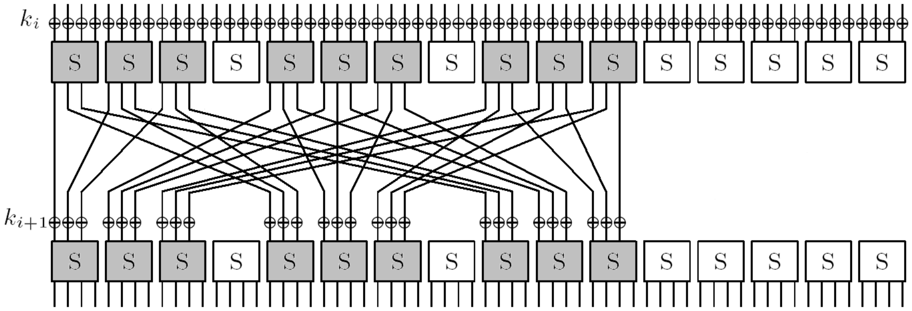

Fig. 11: Another poor diffusion trail in the permutation layer of PRESENT.

References

1. R. Anderson, E. Biham, L. Knudsen, *Serpent: A Proposal for the Advanced Encryption Standard*, in the proceedings of the First Advanced Encryption Standard (AES) Conference, Ventura, CA, August 1998.

2. T. Baignères, P. Junod, S. Vaudenay, *How Far Can We Go Beyond Linear Cryptanalysis?*, in the proceedings of ASIACRYPT 2004, Lecture Notes in Computer Science, vol 3329, pp 432-450, Jeju Island, Korea, December 2004.
3. A. Biryukov, C. De Cannière, M. Quisquater, *On Multiple Linear Approximations*, in the proceedings of CRYPTO 2004, Lecture Notes in Computer Science, vol 3152, pp 1-22, Santa Barbara, California, USA, August 2004.
4. A. Bogdanov, L. Knudsen, G. Leander, C. Paar, A. Poschmann, M. Robshaw, Y. Seurin, C. Vikkelsoe, *PRESENT: An Ultra-Lightweight Block Cipher*, in the proceedings of CHES 2007, Lecture Notes in Computer Science, vol 4727, pp 450-466, Vienna, Austria, September 2007.
5. B. Collard, F.-X. Standaert, J.-J. Quisquater, *Improving the Time Complexity of Matsui's Linear Cryptanalysis*, in the proceedings of The International Conference on Information Security and Cryptology - ICISC 2007, Lecture Notes in Computer Science, vol 4817, pp 77-88, Seoul, Korea, November 2007.
6. J. Daemen, L.R. Knudsen, V. Rijmen, *The Block Cipher Square*, in the proceedings of Fast Software Encryption 1997, Lecture Notes in Computer Science, vol 1267, pp 149-165, Haifa, Israel, January 1997.
7. H. Gilbert, H. Handschuh, A. Joux, S. Vaudenay, *A Statistical Attack on RC6*, in the proceedings of Fast Software Encryption, Lecture Notes in Computer Science, vol 1978, pp 64-74, Yokohama, Japan, April 2001.
8. C. Harpes, G. Kramer, J. Massey, *A Generalization of Linear Cryptanalysis and the Applicability of Matsui's Piling-Up Lemma*, in the proceedings of EUROCRYPT 1995, LNCS, vol 921 , pp 24-38, Saint-Malo, France, May 1995.
9. C. Harpes, J. Massey, *Partitioning Cryptanalysis*, in the proceedings of Fast Software Encryption 1997, LNCS, vol 1267, pp 13-27, Haifa, Israel, January 1997.
10. M. Hermelin, J.Y. Cho, K. Nyberg, *Multidimensional Linear Cryptanalysis of Reduced Round Serpent*, in the proceedings of ACISP 2008, Lecture Notes in Computer Science, vol 5107, pp 203-215, Wollongong, Australia, July 2008.
11. K. Hwang, W Lee, S. Lee, S. Lee, J. Lim, *Saturation Attacks on Reduced Round Skipjack*, in the proceedings of Fast Software Encryption 2002, Lecture Notes in Computer Science, vol 2365, pp 100-111, Leuven, Belgium, February 2002.
12. B.S. Kaliski, M.J.B. Robshaw, *Linear Cryptanalysis using Multiple Approximations*, in the proceedings of CRYPTO 1994, Lecture Notes in Computer Sciences, vol 839, pp 26-39, Santa Barbara, California, USA, August 1994.
13. L.R. Knudsen, D. Wagner, *Integral cryptanalysis*, in the proceedings of Fast Software Encryption 2002, Lecture Notes in Computer Science, vol 2365, pp 112-127, Leuven, Belgium, February 2002.
14. M. Matsui, *Linear cryptanalysis method for DES cipher*, in the proceedings of EUROCRYPT 1993, LNCS, vol 765, pp 386-397, Lofthus, Norway, May 1993.
15. M. Minier, H. Gilbert, *Stochastic Cryptanalysis of Crypton*, in the proceedings of Fast Software Encryption 2000, Lecture Notes in Computer Science, vol 1978, pp 121-133, New York, USA, April 2000.
16. S. Vaudenay, *An experiment on DES - Statistical Cryptanalysis*, in the third ACM Conference on Computer Security, New Dehli, India, pp 139-147, March 1996.
17. M. Wang, *Differential Cryptanalysis of Reduced-Round PRESENT*, in the proceedings of AFRICACRYPT 2008, Lecture Notes in Computer Science, vol 5023, pp 40-49, Casablanca, Morocco, June 2008.

A The block cipher PRESENT

PRESENT is a Substitution-Permutation Network with a block size of 64 bits. The recommended key size is 80 bits, which should be sufficient for the expected applications of the cipher. However a 128-bit key-schedule is also proposed. The encryption is composed of 31 rounds. Each of the 31 rounds consists of a XOR operation to introduce a round key K_i for $1 \leq i \leq 32$, where K_{32} is used for post-whitening, a linear bitwise permutation and a non-linear substitution layer. The non-linear layer uses a single 4-bit S-box S which is applied 16 times in parallel in each round. The cipher is described in pseudo-code in Figure 1.

```
generateRoundKeys()
for $i = 1$ to 31 do
    addRoundKey(STATE,$K_i$)
    sBoxLayer(STATE)
    pLayer(STATE)
end for
addRoundKey(STATE,$K_{32}$)
```

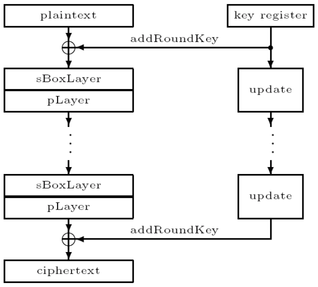

Fig. 12: Top-level algorithmic description of PRESENT according to [4].

The linear permutation is defined by Table 3 where bit i of input is moved to bit position $P(i)$. The 4-bit S-box is defined according to the table 4. The 4-bit nibble i at the input of an S-box is substituted by the 4-bit $\mathsf{S}[i]$ in output.

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$P(i)$	0	16	32	48	1	17	33	48	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
$P(i)$	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
$P(i)$	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
$P(i)$	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63

Table 3: Permutation layer for PRESENT.

i	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[i]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 4: S-box table for PRESENT (hexadecimal notation).

We don't mention the key-schedule here as we don't make explicit use of it in our attack. We refer to the original paper [4] for the details of the specifications.

B Theoretical evaluation of the target trail distribution

Algorithme 1

```
input: a 8-bit subkey guess sk and the 8-bit input distribution distrib_in[256]
output: the 8-bit output distribution distrib_out[256]

initialize distrib_out[256] to the all-zero state
for each 8-bit values text do
  for each 8-bit values rand do
    fix the 8-bit trail to text and xor with sk
    fix the 8-bit non trail to rand
    apply the sboxes
    apply the permutation
    evaluate the value of the 8 bit trail out
    update distrib_out[out]= distrib_out[out]+ distrib_in[text]/256;
  end for
end for
```

C Linear cryptanalysis using a single approximation

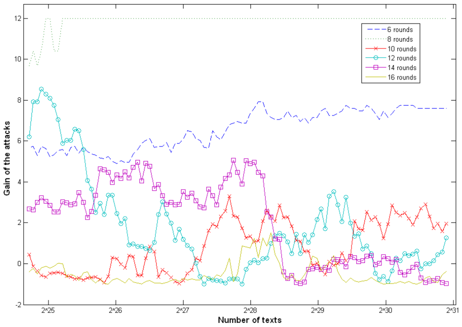

Fig. 13: Gain of a linear cryptanalysis against 6- to 16-rounds PRESENT.

D Influence of the S-box on the attack effectiveness

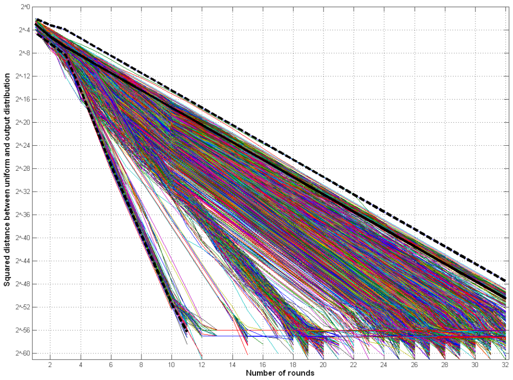

Fig. 14: Evolution of the squared distance between uniform and output distribution for 5000 different S-boxes (the PRESENT S-box is in plain black).