Estimating the Probabilities of Low-Weight Differential and Linear Approximations on PRESENT-Like Ciphers

Mohamed Ahmed Abdelraheem

Department of Mathematics
Technical University of Denmark, Lyngby, Denmark

Abstract. We use large but sparse correlation and transition-difference-probability submatrices to find the best linear and differential approximations respectively on PRESENT-like ciphers. This outperforms the branch and bound algorithm when the number of low-weight differential and linear characteristics grows exponentially which is the case in PRESENT-like ciphers. We found linear distinguishers on 23 rounds of the SPONGENT permutation. We also found better linear approximations on PRESENT using trails covering at most 4 active Sboxes which give us 24-round statistical saturation distinguishers which could be used to break 26 rounds of PRESENT.

Keywords: block cipher, differential, difference matrix, linear hull, correlation matrix, statistical saturation attack, PRESENT, SPONGENT.

1 Introduction

In recent years, the need for lightweight encryption systems has been increasing as many applications use RFID and sensor networks which have a very low computational power and thus incapable of performing standard cryptographic operations. In response to this problem, the cryptographic community designed a number of lightweight cryptographic primitives that varies from stream ciphers such as Grain [15], Trivium [8], and block ciphers such as PRESENT [5], KATAN/KTANTAN [7] and recently to hash functions such as QUARK [1], PHOTON [14] and SPONGENT [4].

Out of these many lightweight primitives, the block cipher PRESENT gets a lot of attention from the cryptographic community and it has been recently adopted as an ISO standard (ISO/IEC 29192) [18]. In this paper, we focus on the differential and linear cryptanalysis of the following two ciphers: PRESENT and SPONGENT. We mainly discuss how to estimate the probabilities of low-weight differential and linear approximations on these kind of ciphers.

Our Contribution: We estimate the probability of low-weight linear and differential approximations in PRESENT-like ciphers. By using large but sparse correlation and difference submatrices, we overcome the memory and time problems

T. Kwon, M.-K. Lee, and D. Kwon (Eds.): ICISC 2012, LNCS 7839, pp. 368–382, 2013.

that appears in the branch and bound algorithm [22] when the number of good linear and differential trails grows exponentially. For instance, there would be memory and time problems in the branch and bound method when the number of differential and linear trails grows exponentially but using sparse correlation and difference submatrices we can handle any number of trails and investigate many differential and linear approximations with a negligible cost. For instance, all the linear approximations of PRESENT at Table 4 described in the Appendix come from a number of good and bad trails exceeding 2^{89} which clearly shows a convincing advantage of using sparse submatrices over the branch and bound algorithm when the number of trails grows exponentially.

Using sparse correlation and difference submatrices of PRESENT and SPONGENT: We report the first improved analysis on SPONGENT [4], specifically we improve the linear cryptanalysis on the SPONGENT permutation presented by the designers by one more round. We present better linear approximations on PRESENT and present a 24-round statistical saturation distinguisher that uses better input and output subspaces compared to the original attack paper [10]. These approximations also show that the assumption, made by all the previous analyses on PRESENT [9, 21, 25], that the linear approximations consisting of trails with one active Sbox at each round, yield the highest bias is not valid. We also found many 16-round differential approximations activating at most 4 Sboxes per round with probability larger than 2^{-64}, which could be used to mount a differential attack on 18-round PRESENT similar to the ones in [2, 29].

Outline of the paper: In Section 2, we give a brief description of PRESENT and SPONGENT. Section 3 defines the basic concepts about linear and differential cryptanalysis. Section 4 describes our sparse matrices approach for finding tight linear and differential approximations and its time complexity. Section 5 shows the linear and differential approximations in PRESENT and SPONGENT that were found using our method. Finally, we conclude on Section 6.

2 A Short Description of PRESENT and SPONGENT

PRESENT is a 64-bit iterated block cipher. It consists of 31 rounds and supports 80-bit and 128-bit key lengths. It was mainly designed for hardware constrained devices [5]. It has a very simple design as it consists of only three layers: the key addition layer, the 4-bit Sbox layer (SboxLayer) and the bitwise permutation layer (PLayer). This simple linear layer consisting of the bitwise permutation only allows the existence of low-weight differential and linear characteristics.

SPONGENT is a new lightweight hash function [4], its core permutation is inspired by PRESENT as it inherits its three layers. There are many variants of SPONGENT here we are concerned with the permutation of SPONGENT-88 which runs for 45 rounds. The SPONGENT permutation can be seen as a cipher using identical round keys (the key is almost the zero key xored with few bits at the leftmost and the rightmost ends generated from a counter that is meant to prevent sliding properties and invariant subspaces). In other words, the core

permutation of SPONGENT can be seen as a PRESENT-like cipher which is a definition we borrowed from [6]. For more details about the description of PRESENT and SPONGENT, we refer to [4, 5].

3 Preliminaries

Suppose we have a symmetric cipher defined by the permutation F, under a key $K \in \mathbb{F}_2^n$, $F_K : \mathbb{F}_2^n \to \mathbb{F}_2^n$.

Differential cryptanalysis exploits the difference distribution under some algebraic group operation (usually $\oplus$) of a pair of plaintexts and their corresponding ciphertexts, i.e. attacker finds a difference α in the plaintext pairs and a difference β in their corresponding ciphertexts such that $\Pr(F_K(X \oplus \alpha) \oplus F_K(X) = \beta)$ is higher than 2^{1-n}.

Linear cryptanalysis finds a linear relation between some plaintext bits and ciphertext bits (and also some secret key bits in the case of block ciphers) and then exploits the bias or the correlation of this linear relation, i.e., attacker finds an input mask α and an output mask β that yield a higher absolute *bias* $\epsilon_F(\alpha, \beta) \in [-\frac{1}{2}, \frac{1}{2}]$. In other words $\Pr(\langle \alpha, X \rangle + \langle \beta, F_K(X) \rangle = \langle \gamma, K \rangle) = \frac{1}{2} + \epsilon_F(\alpha, \beta)$ is deviated from half, where $\langle \cdot, \cdot \rangle$ denotes an inner product. The correlation of a linear approximation is defined as $C_F(\alpha, \beta) := 2\epsilon_F(\alpha, \beta)$.

Linear and differential attacks are based on the so-called linear and differential characteristics (aka trails or paths) respectively, each characteristic is a sequence of intermediate linear or difference relations for all rounds where the probability of each element in this sequence determine the probability of a differential characteristic and the bias of a linear characteristic. The collection of all the r-round differential characteristics with input $\alpha = \alpha_0$ and output $\beta = \alpha_r$ is called the *differential* of the difference approximation (α_0, α_r), each r-round differential characteristic can be seen as a sequence $(\alpha_0, \alpha_{1i}, \cdots, \alpha_{(r-1)i}, \alpha_r)$, where i defines the i-th differential trail between α_0 to α_r. Similarly, the collection of all the linear characteristics with input mask $\alpha = \alpha_0$ and output mask $\beta = \alpha_r$ is often called the *linear hull* of the linear approximation (α_0, α_r), also each r-round linear characteristic could be seen as a sequence $(\alpha_0, \alpha_{1i}, \cdots, \alpha_{(r-1)i}, \alpha_r)$, where i defines the i-th linear trail between α_0 to α_r.

One class of these iterated ciphers defined in [20], is called Markov ciphers. For such ciphers, under well established independence assumptions, the probability of a differential and the correlation of a linear relation can be computed using a difference transition matrix and a correlation matrix respectively. In the following, we briefly describe how to construct and use these matrices.

Difference Transition Matrix [20]: Given an r round Markov cipher and assuming independent and uniformly random round keys. We estimate the probability of an r-round differential (α_0, α_r) by considering the probability of each differential characteristic between α_0 and α_r. Thus, the probability of the i-th differential characteristic $(\alpha_0 = \alpha_{0i}, \alpha_{1i}, \cdots, \alpha_{(r-1)i}, \alpha_r = \alpha_{ri})$ is $p_i = \prod_{j=1}^{r} \Pr(F_K(X) \oplus F_K(X') = \alpha_{ji} | X \oplus X' = \alpha_{(j-1)i})$. Consequently the probability of an r-round

differential (α_0, α_r) is the sum of all the probabilities of all the possible differential characteristics between (α_0, α_r), that is $\sum_{i=1}^{N_d} p_i$, where N_d is the number of all the possible differential characteristics between (α_0, α_r). Let D denote the transition difference-probability matrix of an n-bit Markov cipher. D has size $(2^n - 1) \times (2^n - 1)$, the (i, j) entry in D corresponds to the probability of an output difference, say β_j, when we have an input difference, say β_i, i.e., $\Pr(\Delta(F_K(X)) = \beta_j | \Delta(X) = \beta_i)$, where F_K is the round function of the Markov cipher. Now, for any r, the (i, j) entry of the matrix D^r, $p_{ij}^{(r)}$ is equivalent to the probability of the r-round differential (β_i, β_j).

Correlation Matrix [11, 12]: Given a composite function $F : \mathbb{F}_2^n \to \mathbb{F}_2^n$ such that $F = F_r \circ \cdots \circ F_2 \circ F_1$. We estimate the correlation of an r-round linear approximation (α_0, α_r) by considering the correlation of each linear characteristic between α_0 and α_r, the correlation of i-th linear characteristic $(\alpha_0 = \alpha_{0i}, \alpha_{1i}, \cdots, \alpha_{(r-1)i}, \alpha_r = \alpha_{ri})$ is $C_i = \prod_{j=1}^{r} C_{F_j}(\alpha_{(j-1)i}, \alpha_{ji})$. It is well known, see e.g., [12], that the correlation of a linear approximation is the sum of all correlations of linear trails starting with the same mask α and ending with the same mask β, i.e., $C_F(\alpha_0, \alpha_r) = \sum_{i=1}^{N_l} C_i$, where N_l is the number of all the possible linear characteristics between (α_0, α_r).

The sign of the correlation of a linear trail depends on the round keys. In [12] the following formulas were proven under the assumption that we have a key-alternating cipher[1]: $C_i = (-1)^{s_i} \prod_{j=1}^{r} C_{F_j}(\alpha_{(j-1)i}, \alpha_{ji})$, where $s_i \in \mathbb{F}_2$ depends on the i-th linear characteristic and the round keys. Therefore, the correlation of the linear hull (α, β) is $C_F(\alpha, \beta) = \sum_{i=1 | \alpha_i = (\alpha = \alpha_{0i}, \cdots, \alpha_{(r-1)i}, \beta = \alpha_{ri})}^{N_l} (-1)^{s_i \oplus d_i} |C_i|$, where $d_i \in \mathbb{F}_2$ refers to the sign of the correlation, C_i.

Let C denote the correlation matrix of an n-bit key-alternating cipher. C has size $(2^n - 1) \times (2^n - 1)$, the (i, j) entry in C corresponds to the correlation of an input mask, say β_i, and output mask, say β_j, i.e. $C_F(\beta_i, \beta_j) = 2 \Pr(\langle \beta_i, x \rangle = \langle \beta_j, F(x) \rangle) - 1$, where F is the un-keyed composite function of the key-alternating cipher and 'x' is its input. Now the correlation matrix for the keyed round function is obtained by changing the signs of each row in C according to the round subkey bits or the round constant bits involved.

Statistical Saturation Attacks: They are the first attacks proposed on the block cipher PRESENT. Briefly the idea behind these attacks is to fix some input bits to a certain value and study the distribution of some output bits, for more details we refer to [10]. In [21] it was shown that it is closely related to the linear multidimensional attack and especially the one on PRESENT [9]. The following proposition formulated at [21] estimates the capacity of statistical saturation attacks which is used to estimate the data complexity required to mount the attack.

Proposition 1. *Let $F : \mathbb{F}_2^r \times \mathbb{F}_2^s \to \mathbb{F}_2^t \times \mathbb{F}_2^u$ be an n-bit encryption function where $r + s = t + u = n$, F is restricted by fixing s bits in the input and*

[1] Key-alternating ciphers are a subclass of Markov ciphers that alternate key addition with key-independent rounds.

only t bits of the output are considered. Let $U = \mathbb{F}_2^s \subseteq \mathbb{F}_2^n$ *and* $V = \mathbb{F}_2^t \subseteq \mathbb{F}_2^n$ *be the two subspaces corresponding to the input and output masks respectively. Then the average capacity over all the possible s-bit fixations is estimated by* $\overline{C_F} = \sum_{u \in U, v \in V} (C_F(u,v))^2$.

The data complexity of statistical saturation attacks is determined by the squared Euclidean distance which is equivalent to $2^t \overline{C_F}$ where t is the number of the output bits considered in the distribution. Statistical saturation attacks perform well when we identify subspaces U and V that make the sum $\sum_{u \in U, v \in V} (C_F(u,v))^2$ big.

4 Description of Our Estimation Approach

Assuming that PRESENT-like ciphers are Markov ciphers [20, 24], we make use of submatrices of the correlation and the transition probability matrices of the target ciphers to find the best linear and differential approximations. We focus only on describing how to find better linear approximations.

4.1 Large Sparse Correlation and Difference Matrices

In [3, 4], a submatrix of the correlation matrix of size $4n_s \times 4n_s$ was used to estimate the correlation of a linear approximation, where n_s is the number of the 4-bit Sboxes used in the permutation where only input and output masks of Hamming weight one are considered. We extend this approach by adding input and output masks with Hamming weight ≤ 4. This results in having a large correlation submatrix whose entries activate at most 4 active Sboxes. Suppose that we use a correlation submatrix with input and output masks with Hamming weight less than or equal to m. Then the size of the submatrix will be $\sum_{i=1}^{m} \binom{n}{i} \times \sum_{i=1}^{m} \binom{n}{i}$, where n is the block size of the cipher in bits. The submatrix size is large but most of its entries are zeros. For instance, we see that for any input element activating 's' Sboxes ($1 \leq s \leq m$), all the other output elements corresponding to the other Sboxes yield a zero correlation. Thus, there are more than $\sum_{i=1}^{m} \binom{n}{i} - 15^s$ zero output elements for any input activating 's' Sboxes. Thus, this submatrix has few non zero elements and therefore it can easily fit in memory using a sparse matrix storage format (See Section 5).

The construction of the correlation submatrix is straightforward. For instance to fill the submatrix entries from an input with Hamming weight 5, we proceed as follows: for each possible input, we determine the number of activated Sboxes which is in this case at least 2. Suppose we have i active Sboxes, then all the possible ordered solutions of the inequality $x_1 + x_2 + \cdots + x_i \leq m$ determine the Hamming weight of the output bits of each of the i active Sboxes. Then we fill the submatrix entries corresponding to the specified input by considering all the possible output bits of the specified Hamming weight. To estimate the cost of filling these entries, we consider the simple case where we have two active Sboxes. The time cost for filling the corresponding submatrix entries is $\sum_{2 \leq i+j \leq 5} \binom{4}{i}\binom{4}{j}$

and the number of all the possible inputs with Hamming weight 5 activating 2 Sboxes is $N_2 = \sum_{w_1+w_2=5} \binom{\frac{n}{4}}{2}\binom{4}{w_1}\binom{4}{w_2}$. Now by symmetry, the cost of filling the corresponding entries that have outputs with Hamming weight 5 activating 2 Sboxes is similar but we only exclude the duplicated cases where $i+j=5$, so the cost is $\sum_{2\leq i+j<5}\binom{4}{i}\binom{4}{j}$. Therefore, the total construction time of input and output with Hamming weight 5 activating 2 Sboxes is $2N_2\sum_{2\leq i+j\leq 5}\binom{4}{i}\binom{4}{j} - N_2\sum_{i+j=5}\binom{4}{i}\binom{4}{j}$. One can see that the construction time can be generalized as follows.

Proposition 2. *The time cost for computing the correlations corresponding to inputs and outputs of Hamming weight 'w', $1 \leq w \leq m$ is in the order of:* $2(N_1\sum_{1\leq i\leq w}\binom{4}{i} + N_2\sum_{2\leq i+j\leq w}\binom{4}{i}\binom{4}{j} + \cdots + N_{w-1}\sum_{w-1\leq i+\cdots+z\leq w}\binom{4}{i}\cdots\binom{4}{z}) + N_w 4^w - N_2\sum_{i+j=w}\binom{4}{i}\binom{4}{j} - \cdots - N_{w-1}\sum_{i+\cdots+z=w}\binom{4}{i}\cdots\binom{4}{z}$, *where* $N_1 + \cdots + N_w = \binom{n}{w}$ *and* $N_i = \sum_{w_1+\cdots+w_i=w}\binom{\frac{n}{4}}{i}\binom{4}{w_1}\cdots\binom{4}{w_i}$ *is the number of elements with Hamming weight 'w' activating 'i' number of Sboxes.*

Note that $N_1 = 0$ when $w \geq 5$ as in this case we have at least two active Sboxes. The total construction time is in the order of the sum of construction times of all the possible input and output weights (w), i.e. $1 \leq w \leq m$, where the dominant term is when $w = m$.

After constructing the correlation submatrix, C. The correlation approximations after r rounds is computed by $C^r = \prod_{i=1}^{r} M_i$, where M_i is the correlation submatrix at round i formed by changing the signs of C according to the round key and the round constant used in the cipher. The maximum correlation after r rounds is thus given by $c^r_{max} := \max|C^r_{ij}|$. This works in the case of SPONGENT since we know that it uses an almost zero key at each round and thus we can compute the actual correlation of each approximation but in the case of PRESENT, we need to compute the average squared correlation (aka potential linear approximation [23] or expected linear probability [13]) of a linear approximation (the sum of the squares of the correlations of all trails) in order to compute the capacity of the statistical saturation attack.

As the size of the correlation submatrix gets bigger when considering masks with Hamming weight equal to 4, the matrix-matrix multiplications might not be always possible for high number of rounds especially when there are many trails for most of the approximations as these make the resulted submatrix C^r very dense and consequently we might run out of memory. Thus, instead we use successive matrix-vector multiplications as described by Algorithm 1 in the Appendix.

Note that before Step 3 in Algorithm 1 when we are computing the maximum absolute correlation (for example in SPONGENT), we have to change the signs at some entries of the correlation submatrix M at each round according to the corresponding round constant. The time complexity of Algorithm 1 is the order of $l \times (r-1)$ matrix-vector multiplications where l is the number of rows or columns of the submatrix. If l is a large number, then the most convenient way is to consider correlation matrices with Hamming weight up to 2 or

3 bits depending on the size of the block cipher. Then, try to perform matrix-matrix multiplications and find the active input and output Sboxes that yield the maximum absolute value as they would probably be the Sboxes that yield the maximum value when considering matrices using Hamming weight more than 3. For instance, experiments on the PRESENT correlation submatrix with Hamming weight up to 3, where we are able to perform matrix-matrix multiplication and thus determine the correlations of all the approximations, showed us that the best linear approximations often come from an input mask activating only one Sbox and also an output mask activating only one Sbox (which get permuted afterwards).

5 Improved Linear and Differential Approximations

We use the approach described in section 4.1 and report the best linear and differential approximations we found in PRESENT and SPONGENT. Using the time complexity formula, given at section 4.1, we show the times taken in constructing the sparse matrices in PRESENT and SPONGENT.

Table 1. $n \equiv$ Cipher's block size, $m \equiv$ maximum Hamming weight used, $size \equiv$ submatrix size, $nnzC \equiv$ non zero elements of the correlation submatrix, $nnzD \equiv$ non zero elements of the difference submatrix, Time complexity of correlation or difference submatrix construction $\equiv O(t)$, $- \equiv$ bounded by t since each step in t fills an entry in the correlation or difference submatrix. The time complexity unit is simple arithmetic operations.

Cipher	n	m	$\log_2(size)$	$\log_2(nnzC)$	$\log_2(nnzD)$	$\log_2(t)$
PRESENT	64	4	19.37×19.37	23.26	18.41	27.83
PRESENT	64	5	22.99×22.99	-	-	33.85
PRESENT	64	6	26.31×26.31	-	-	39.61
SPONGENT	88	4	21.22×21.22	23.63	19.18	29.58
SPONGENT	88	5	25.31×25.31	-	-	36.04

As shown in Table 1, the number of elements in the correlation and differences submatrices of both PRESENT and SPONGENT is huge. A standard matrix representation would cost $2^{41.74}$ and $2^{45.44}$ bytes for the difference matrix of PRESENT and SPONGENT respectively. This is more than 1 TB. Therefore, we need to avoid running out of memory by using a sparse matrix representation which reduces memory by only allocating space for the nonzero elements. This will also speed the matrix-vector or matrix-matrix multiplications which we perform to find the best linear and difference approximations.

Table 1 shows us that our submatrices are very sparse, for instance the first table entry indicates that the density ($=\frac{nnz}{Size \times Size}$) of the difference transition submatrix of PRESENT with input and output differences of Hamming weight up to 4 is 7.56×10^{-7}. This confirms that these large submatrices are considerably

sparse. Therefore, using a sparse matrix storage format where we only allocate storage for the nonzero elements, our large correlation submatrix could easily fit in memory. The very general and simple format for storing sparse matrices is called Compressed Column Storage (CCS). Using this format, the storage cost of a sparse matrix depends on the number of its nonzero elements (nnz) and its column size ($ncol$). More specifically, the cost of a real-valued sparse matrix in CCS format is equivalent to the cost of nnz real-valued numbers and (nnz+$ncol$+1) integers [27]. Thus, on a 64-bit machine, where we have 8 bytes for both real and integer numbers, the total memory cost would be $8nnz + 8(nnz + ncol + 1)$ bytes. Using the numbers on Table 1, we see that the total memory cost for a CCS sparse representation of PRESENT and SPONGENT difference submatrices is 11014024 ($\approx 2^{23.4}$) and 29085768 ($\approx 2^{24.8}$) bytes respectively. Also the memory cost for a CCS representation of PRESENT and SPONGENT correlation submatrices is 165990920 ($\approx 2^{27.3}$) and 226974888 ($\approx 2^{27.8}$) bytes respectively. Each of these submatrices costs less than 1 GB and thus would easily fit in memory.

5.1 Application on SPONGENT

Here we find linear approximations that can be used to distinguish 23 rounds of the SPONGENT-88 permutation using the whole code book and this is one more round than what has been provided in [4]. We also give the maximum differential characteristic probability we found on 16-round SPONGENT-88.

Differential Approximations: We constructed a difference transition submatrix for SPONGENT-88 with input and output differences having Hamming weight at most 4 bits. The maximum differential probability obtained by powering the transition submatrix[2] is a 16-round differential and it has probability $2^{-77.83}$. One of the differentials having this probability is $e_1 \oplus e_4 \oplus e_{17} \oplus e_{20} \rightarrow e_9 \oplus e_{33} \oplus e_{75} \oplus e_{77}$ and it consists of only one differential trail. This is one round less than the best differential provided by the designers as their differential include characteristics with differences having Hamming weight more than 4. It would be interesting to see whether input and output differences with Hamming weight at most 5 bits would yield better estimations. However, as noted in Table 1 the time complexity is 2^{36} arithmetic operations which have not tried due to the lack of computing resources.

Linear Approximations: The SPONGENT Sbox was chosen carefully to avoid the many linear trails with one active Sbox in each round existing on PRESENT [25]. For instance in SPONGENT-88, there is only one trail that have one active Sbox at each round, which makes a linear distinguisher possible for not more than 22 rounds. Now we use a correlation submatrix with input and output masks of Hamming weight up to 4 to activate at most 4 Sboxes. As a result,

[2] This is possible for the difference submatrices of PRESENT and SPONGENT but not for their correlation matrices as they are dense.

we found many linear approximations with correlations larger than 2^{-44} for 23-round of SPONGENT-88. Thus, we improved the linear distinguishers provided by the designers one more round. Table 2 shows the correlations obtained along with the corresponding number of trails written between parentheses. The table shows that correlations obtained from using correlation matrices with masks of Hamming weight at most 2 bits, 3 bits and 4 bits do not vary significantly and this might indicate that linear characteristics covering more than 4 active Sboxes per round do not have a significant effect in the total correlation. The table also shows that it is difficult to accurately estimate the total correlation of some linear approximations. For instance for 22 rounds, $|C^{\leq 2}(e_{70} \oplus e_{71}, e_{56} \oplus e_{78})|$ is smaller than $|C^{\leq 3}(e_{70} \oplus e_{71}, e_{56} \oplus e_{78})|$ but bigger than $|C^{\leq 4}(e_{70} \oplus e_{71}, e_{56} \oplus e_{78})|$. This suggests that the characteristics with Hamming weight 4 bits contributed negatively to the total correlation.

Table 2. $r \equiv$ number of rounds, $\alpha \equiv$ input mask, $\beta \equiv$ output mask, $|C^{\leq i}(\alpha, \beta)| \equiv$ correlation using a submatrix with input and output masks with Hamming weight at most i bits, $- \equiv$ not applicable. $e_i \equiv$ the unit vector with single 1 at position i whose length is 88 (SPONGENT-88's block size). The values between parentheses represent the $\log_2$ of the corresponding number of trails which are easily calculated by replacing each nonzero entry with 1 in the correlation submatrix and then powering it to r.

r	α	β	$\log_2(\lvert C^{\leq 2}(\alpha,\beta)\rvert)$	$\log_2(\lvert C^{\leq 3}(\alpha,\beta)\rvert)$	$\log_2(\lvert C^{\leq 4}(\alpha,\beta)\rvert)$
22	$e_6 \oplus e_7$	$e_3 \oplus e_{25} \oplus e_{47}$	-	-43.82 (20.52)	-43.83 (36.04)
23	$e_6 \oplus e_7$	$e_3 \oplus e_{25} \oplus e_{47}$	-	-43.81 (21.91)	-43.74 (38.44)
22	$e_6 \oplus e_7$	$e_3 \oplus e_{25} \oplus e_{47} \oplus e_{69}$	-	-	-43.83 (36.17)
23	$e_6 \oplus e_7$	$e_3 \oplus e_{25} \oplus e_{47} \oplus e_{69}$	-	-	-43.75 (38.56)
22	$e_{70} \oplus e_{71}$	$e_7 \oplus e_{51}$	-42.06 (7.67)	-42.05 (22.75)	-42.05 (38.25)
23	$e_{70} \oplus e_{71}$	$e_7 \oplus e_{51}$	-44.02 (7.95)	-44.01 (24.13)	-43.95 (40.65)
22	$e_{70} \oplus e_{71}$	$e_{56} \oplus e_{78}$	-42.03 (7.12)	-42.03 (22.51)	-42.04 (38.29)
23	$e_{70} \oplus e_{71}$	$e_{56} \oplus e_{78}$	-43.99 (6.94)	-43.99 (23.89)	-43.96 (40.69)
22	$e_{70} \oplus e_{71}$	$e_7 \oplus e_{29} \oplus e_{51} \oplus e_{73}$	-	-	-42.04 (37.76)
23	$e_{70} \oplus e_{71}$	$e_7 \oplus e_{29} \oplus e_{51} \oplus e_{73}$	-	-	-43.94 (40.16)
22	$e_9 \oplus e_{10} \oplus e_{11}$	$e_3 \oplus e_{25} \oplus e_{47}$	-	-43.99 (19.73)	-43.93 (35.35)
23	$e_9 \oplus e_{10} \oplus e_{11}$	$e_3 \oplus e_{25} \oplus e_{47}$	-	-43.97 (21.14)	-43.88 (37.75)
22	$e_{46} \oplus e_{47} \oplus e_{48}$	$e_{34} \oplus e_{56} \oplus e_{78}$	-	-42.72 (22.49)	-42.69 (39.23)
23	$e_{46} \oplus e_{47} \oplus e_{48}$	$e_{34} \oplus e_{56} \oplus e_{78}$	-	-43.95 (23.86)	-43.89 (41.62)

5.2 Application on PRESENT

The PRESENT block cipher has been analyzed in several publications. In [29], a 16-round differential attack was mounted. Later a statistical saturation attack was mounted and claimed to break 24 rounds [10]. In [25], the author showed the existence of 32% of weak keys which have a higher bias that makes the cipher using those weak keys distinguishable for up to 24 rounds. In [9], the multidimensional attack was used to break 25 rounds and also 26 rounds where the latter use the whole code book. Recently, a multiple differential attack was mounted on 18-round of PRESENT [2]. Our focus here is to use the approach described above to find better linear and differential approximations.

Differential Approximations: We use a difference transition submatrix whose input and output differences have Hamming weight less than or equal to four. Now, in order to estimate the differential probability after r rounds, we raise our transition submatrix to r and extract the maximum entry. The time and memory costs of this are negligible. We found that the 2-round iterative characteristic in [5] has probability 2^{-74} for 15-round PRESENT but the differential containing this characteristic has a higher probability equivalent to $2^{-63.50}$ for a 15-round PRESENT. We also found many differentials with probability larger than 2^{-64} for 16 rounds PRESENT where the maximum one occurs with probability $2^{-62.58}$. Moreover, the maximum differential probability we found for 25 rounds is equal to $2^{-97.38}$. This is larger than the 2^{-100} differential characteristic bound for 25 rounds given in [5]. Here we note that the analysis provided in [5] is sound as the authors gave a bound for the differential characteristic and not for the differential which is hard to bound. Nevertheless, this shows that our approach can be useful in bounding the probability of a differential.

Linear Approximations and Statistical Saturation Attacks: All the previous linear attacks on PRESENT used linear trails activating only one Sbox at each round. To find better linear approximations, we considered trails activating at most 4 Sboxes. Thus, we constructed a correlation submatrix using input and output masks of Hamming weight at most 4 bits. By searching for the best approximations among input masks and output masks in one Sbox. We found that there are many approximations whose squared correlation is larger than 2^{-64} when ≤ 24 rounds of PRESENT are used. As noted in [25], these approximations follow the normal distribution with mean zero and variance equal to their squared correlation. Thus, the squared correlation is higher for 32% of keys compared to the whole key space for some approximations where each approximation has a different path. Thus when using multiple linear approximations, each key is more likely to yield a high correlation with respect to some input and output masks [17]. Therefore statistical saturation distinguishers based on linear approximations whose squared correlations are larger than 2^{-64} work exactly as predicted for almost all the keys.

Table 4 described in the Appendix lists the 10 approximations spanned from U_{11} and V_1 and also the 10 approximations spanned from U_{11} and V_3. All these approximations have a squared correlation larger than 2^{-64} and they give us two 24-round statistical saturation distinguishers. Using the input subspace $U_{11} = span\{e_{41}, e_{42}, e_{43}, e_{44}\}$ which corresponds to fixing the 4 bits entering the 11-th Sbox (counting from left to right) and the output subspace $V_1 = span\{e_1, e_{17}, e_{33}, e_{49}\}$ which corresponds to the 4 bits resulted after applying the permutation on the output of the first Sbox, we get a statistical saturation distinguisher on 24 rounds with an average capacity equal to $2^{-60.53}$. Using the same input subspace with another different output subspace $V_3 = span\{e_3, e_{19}, e_{35}, e_{51}\}$, we also get a statistical saturation distinguisher with an average capacity $2^{-60.53}$. Using another input subspace $U_{10} = \{e_{37}, e_{38}, e_{39}, e_{40}\}$

with each of the above two output subspaces we get distinguishers with the same capacities. Now all these 24-round statistical saturation distinguishers can be used to mount a key recovery attack for 16 bits of the last round key on 25 rounds of PRESENT using the whole code book.

Moreover, these 24-round distinguishers could be used to mount a 26-round key recovery attack similar to [9] to recover 16 bits from the 1st round subkey (4 bits from each of the 9th, 10th, 11th and 12th Sbox) and also 16 bits from last round subkey (4 bits from each of the 1st, 5th, 9th and 13th Sbox) but still estimating the success probability and data complexity is difficult. However, the statistical framework developed in [16] in order to estimate the success probability and data complexity of the multidimensional attack could also be used to estimate the success probability and data complexity of this attack, should we assume the independence of the linear approximations used which is not true. This is in fact what has been done in [9] as it has been noted in [17] that the linear approximations used in the 26-round multidimensional attack of PRESENT can not be statistically independent as several approximations share the same input mask.

Therefore, rather than giving the success probability and data complexity, we list in Table 3 the estimated squared Euclidean distance of the statistical saturation distinguisher with input subspace U_{11} and output subspace V_1 for various number of rounds along with the experimental Euclidean distance using 100 random master keys.

Table 3. The table shows the estimated Euclidean distance D together with the experimental Euclidean distance D' averaged over 100 random keys with various amount of plaintexts, namely 2^{10} plaintexts are used for $r = 2, 3$, 2^{12} for $r = 4$, 2^{17} for $r = 5, 6$, and 2^{20} for $r = 7, 8$. $D'_* \equiv$ the Euclidean distance for a wrong key guess.

r	2	3	4	5	6	7	8	23	24
$\log_2(D)$	$-\infty$	-8.00	-9.99	-12.81	-15.23	-17.79	-20.37	-55.52	-64.53
$\log_2(D')$	-10.07	-7.70	-9.67	-12.74	-14.32	-17.09	-19.06	-	-
$\log_2(D'_*)$	-7.69	-9.03	-11.38	-14.34	-16.19	-19.49	-19.92	-	-

Table 3 the Euclidean distance obtained via a wrong key guess which was simulated by encrypting one more round under the right key. The table also shows clearly that the experimental Euclidean distances are close to the estimated capacities and the more plaintext we use the closer our experimental distances get to the expected distances. Thus, using the above mentioned four statistical distinguishers we could find 16 key bits from each of the first and last round keys using the whole code book. We note that these statistical saturation distinguishers are better than the distinguisher reported in the original attack [10] whose input and output subspaces are $U = V = span\{e_{22}, e_{23}, e_{26}, e_{27}, e_{38}, e_{39}, e_{42}, e_{43}\}$. This is because all the linear approximations spanned from U and V do not have a single linear approximation with a squared correlation larger than 2^{-64} even when considering input and output masks with Hamming weight at most 4 bits.

6 Conclusion and Future Work

In this paper, we used sparse difference and correlation submatrices to estimate the probabilities of low-weight differential and linear approximations respectively in PRESENT-like ciphers. This estimation approach can also be used in any cipher allowing low-weight differential and linear characteristics. Using these sparse matrices, we found linear distinguishers for 23-round of SPONGENT-88. While this is far from distinguishing the full 45 rounds of SPONGENT-88, it is the best currently known result against SPONGENT. We also presented four 24-round statistical saturation distinguishers which break 26-round of PRESENT and that is more than the rounds attacked by the original statistical saturation attack [10].

It would be interesting to investigate whether using large difference and correlation submatrices for PRESENT and SPONGENT-88 with entries having Hamming weight at most 5 would make some improvements over this work. Looking at Table 1 we see that the time complexities for constructing these submatrices take around 2^{34} and 2^{36} arithmetic operations for PRESENT and SPONGENT-88 respectively which could be feasible using parallel computing.

Acknowledgements. For His uncountable blessings, unlimited thanks are to ALLAH that are suitable for His majesty and His perfect attributes. Many thanks go to Lars Knudsen, Gregor Leander and the anonymous reviewers for many useful comments.

References

1. Aumasson, J.-P., Henzen, L., Meier, W., Naya-Plasencia, M.: QUARK: A lightweight hash. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 1–15. Springer, Heidelberg (2010)
2. Blondeau, C., Gérard, B.: Multiple differential cryptanalysis: Theory and practice. In: Joux (ed.) [19], pp. 35–54
3. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., Verbauwhede, I.: Spongent: The design space of lightweight cryptographic hashing. IEEE Transactions on Computers PP(99), 1 (2012)
4. Bogdanov, A., Knežević, M., Leander, G., Toz, D., Varıcı, K., Verbauwhede, I.: SPONGENT: A lightweight hash function. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 312–325. Springer, Heidelberg (2011)
5. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
6. Borghoff, J., Knudsen, L.R., Leander, G., Thomsen, S.S.: Cryptanalysis of present-like ciphers with secret s-boxes. In: Joux (ed.) [19], pp. 270–289
7. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — A family of small and efficient hardware-oriented block ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
8. De Cannière, C., Preneel, B.: Trivium. In: Robshaw, Billet (eds.) [26], pp. 244–266

9. Cho, J.Y.: Linear cryptanalysis of reduced-round present. In: Pieprzyk, J. (ed.) CT-RSA 2010. LNCS, vol. 5985, pp. 302–317. Springer, Heidelberg (2010)
10. Collard, B., Standaert, F.-X.: A statistical saturation attack against the block cipher PRESENT. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 195–210. Springer, Heidelberg (2009)
11. Daemen, J., Govaerts, R., Vandewalle, J.: Correlation matrices. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 275–285. Springer, Heidelberg (1995)
12. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
13. Daemen, J., Rijmen, V.: Probability distributions of correlation and differentials in block ciphers. IACR Cryptology ePrint Archive, 2005:212 (2005)
14. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON family of lightweight hash functions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 222–239. Springer, Heidelberg (2011)
15. Hell, M., Johansson, T., Maximov, A., Meier, W.: The grain family of stream ciphers. In: Robshaw, Billet (eds.) [26], pp. 179–190
16. Hermelin, M., Cho, J.Y., Nyberg, K.: Multidimensional extension of matsui's algorithm 2. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 209–227. Springer, Heidelberg (2009)
17. Hermelin, M., Nyberg, K.: Linear cryptanalysis using multiple linear approximations. Cryptology ePrint Archive, Report 2011/093 (2011)
18. ISO/IEC 29192-2:2012. Information technology Security techniques Lightweight cryptography. Part 2: Block ciphers (2012)
19. Joux, A. (ed.): FSE 2011. LNCS, vol. 6733. Springer, Heidelberg (2011)
20. Lai, X., Massey, J.L.: Markov ciphers and differential cryptanalysis. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 17–38. Springer, Heidelberg (1991)
21. Leander, G.: On linear hulls, statistical saturation attacks, present and a cryptanalysis of puffin. In: Paterson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 303–322. Springer, Heidelberg (2011)
22. Matsui, M.: On correlation between the order of s-boxes and the strength of des. In: Santis (ed.) [28], pp. 366–375
23. Nyberg, K.: Linear approximation of block ciphers. In: Santis (ed.) [28], pp. 439–444
24. O'Connor, L., Golić, J.D.: A unified markov approach to differential and linear cryptanalysis. In: Pieprzyk, J., Safavi-Naini, R. (eds.) ASIACRYPT 1994. LNCS, vol. 917, pp. 387–397. Springer, Heidelberg (1995)
25. Ohkuma, K.: Weak keys of reduced-round PRESENT for linear cryptanalysis. In: Jacobson Jr., M.J., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 249–265. Springer, Heidelberg (2009)
26. Robshaw, M., Billet, O. (eds.): New Stream Cipher Designs. LNCS, vol. 4986. Springer, Heidelberg (2008)
27. Saad, Y.: SPARSKIT: A basic tool kit for sparse matrix computation. Research Institute for Advanced Computer Science, NASA Ames Research Center (1990)
28. De Santis, A. (ed.): EUROCRYPT 1994. LNCS, vol. 950. Springer, Heidelberg (1995)
29. Wang, M.: Differential cryptanalysis of reduced-round PRESENT. In: Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 40–49. Springer, Heidelberg (2008)

A Appendix

Algorithm 1. Finding the best the average squared correlation or absolute correlation

Require: Submatrix M of size $l \times l$, M is a submatrix of the average squared correlation matrix (or of the correlation matrix).

Require: Two temporary vectors of length "l", tempCorr and tempIndex.

Ensure: Finds the best average squared correlation (absolute correlation) with its corresponding input mask a and output mask b.

1: *counter* $= 0$.
2: **for** $j = 1 \rightarrow l$ **do**
3: Extract the jth Column C_j from M.
4: **repeat**
5: $C_j = M \times C_j$
6: Increment *counter*
7: **until** *counter* equals $r - 1$.
8: tempCorr(j)=max($|C_j|$).
9: tempIndex(j) = The index of max($|C_j|$) gives us the corresponding input mask.
10: **end for**
11: **return** max(tempCorr) which yields the maximum average squared correlation (absolute correlation) and its index yields the corresponding output mask b. Then the corresponding input mask a = max(tempIndex(b)).

Table 4. $(C^{\leq 4}(\alpha,\beta))^2 \equiv$ squared correlation of a 24-round PRESENT linear approximation with input mask α and output mask β computed via a correlation submatrix with Hamming weight at most 4. The first 10 approximations correspond to the output subspace V_1 while the second 10 approximations correspond to the output subspace V_3. $e_i \equiv$ the unit vector with single 1 at position i whose length is 64 (PRESENT's block size). The values between parentheses represent the $\log_2$ of the corresponding number of trails.

α	β	$\log_2((C^{\leq 4}(\alpha,\beta))^2)$
$e_{41} \oplus e_{43}$	$e_1 \oplus e_{17} \oplus e_{33}$	-63.98 (91.67)
$e_{41} \oplus e_{43}$	$e_1 \oplus e_{33} \oplus e_{49}$	-63.77 (90.62)
$e_{41} \oplus e_{43}$	$e_1 \oplus e_{17} \oplus e_{33} + e_{49}$	-63.97 (91.48)
$e_{41} \oplus e_{42} \oplus e_{43}$	$e_1 \oplus e_{17} \oplus e_{33}$	-63.80 (91.00)
$e_{41} \oplus e_{42} \oplus e_{43}$	$e_1 \oplus e_{17} \oplus e_{49}$	-63.97 (91.12)
$e_{41} \oplus e_{42} \oplus e_{44}$	$e_1 \oplus e_{17} \oplus e_{33}$	-63.97 (91.52)
$e_{41} \oplus e_{42} \oplus e_{43}$	$e_1 \oplus e_{33} \oplus e_{49}$	-63.60 (89.95)
$e_{41} \oplus e_{42} \oplus e_{44}$	$e_1 \oplus e_{33} \oplus e_{49}$	-63.77 (90.47)
$e_{41} \oplus e_{42} \oplus e_{43}$	$e_1 \oplus e_{17} \oplus e_{33} \oplus e_{49}$	-63.80 (91.48)
$e_{41} \oplus e_{42} \oplus e_{44}$	$e_1 \oplus e_{17} \oplus e_{33} \oplus e_{49}$	-63.96 (91.33)
$e_{41} \oplus e_{43}$	$e_3 \oplus e_{19} \oplus e_{35}$	-63.98 (91.66)
$e_{41} \oplus e_{43}$	$e_3 \oplus e_{35} \oplus e_{51}$	-63.78 (90.62)
$e_{41} \oplus e_{43}$	$e_3 \oplus e_{19} \oplus e_{35} \oplus e_{51}$	-63.97 (91.48)
$e_{41} \oplus e_{42} \oplus e_{43}$	$e_3 \oplus e_{19} \oplus e_{35}$	-63.81 (91.00)
$e_{41} \oplus e_{42} \oplus e_{43}$	$e_3 \oplus e_{19} \oplus e_{51}$	-63.97 (91.11)
$e_{41} \oplus e_{42} \oplus e_{44}$	$e_3 \oplus e_{19} \oplus e_{35}$	-63.97 (91.51)
$e_{41} \oplus e_{42} \oplus e_{43}$	$e_3 \oplus e_{35} \oplus e_{51}$	-63.60 (89.95)
$e_{41} \oplus e_{42} \oplus e_{44}$	$e_3 \oplus e_{35} \oplus e_{51}$	-63.77 (90.47)
$e_{41} \oplus e_{42} \oplus e_{43}$	$e_3 \oplus e_{19} \oplus e_{35} \oplus e_{51}$	-63.80 (90.81)
$e_{41} \oplus e_{42} \oplus e_{44}$	$e_3 \oplus e_{19} \oplus e_{35} \oplus e_{51}$	-63.97 (91.33)