Algebraic Techniques in Differential Cryptanalysis Revisited*

Meiqin Wang[1,2,3,**], Yue Sun[1], Nicky Mouha[2,3,***], and Bart Preneel[2,3]

[1] School of Mathematics, Shandong University, Jinan 250100, China
[2] Department of Electrical Engineering ESAT/SCD-COSIC,
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium.
[3] Interdisciplinary Institute for BroadBand Technology (IBBT), Belgium.
mqwang@sdu.edu.cn

Abstract. At FSE 2009, Albrecht *et al.* proposed a new cryptanalytic method that combines algebraic and differential cryptanalysis. They introduced three new attacks, namely Attack A, Attack B and Attack C. For Attack A, they explain that the time complexity is difficult to determine. The goal of Attacks B and C is to filter out wrong pairs and then recover the key. In this paper, we show that Attack C does not provide an advantage over differential cryptanalysis for typical block ciphers, because it cannot be used to filter out any wrong pairs that satisfy the ciphertext differences. Furthermore, we explain why Attack B provides no advantage over differential cryptanalysis for PRESENT. We verify our results for PRESENT experimentally, using both PolyBoRi and MiniSat. Our work helps to understand which equations are important in the differential-algebraic attack. Based on our findings, we present two new differential-algebraic attacks. Using the first method, our attack on 15-round PRESENT-80 requires 2^{59} chosen plaintexts and has a worst-case time complexity of $2^{73.79}$ equivalent encryptions. Our new attack on 14-round PRESENT-128 requires 2^{55} chosen plaintexts and has a worst-case time complexity of $2^{112.83}$ equivalent encryptions. Although these attacks have a higher time complexity than the differential attacks, their data complexity is lower.

Keywords: Differential-Algebraic Attack, Block Cipher, PRESENT

1 Introduction

Differential cryptanalysis [6, 7] is one of classic cryptanalytic methods for block ciphers. Resistance against differential cryptanalysis is a typical design crite-

[*] This work was supported in part by the Research Council K.U.Leuven: GOA TENSE, the IAP Program P6/26 BCRYPT of the Belgian State (Belgian Science Policy), and in part by the European Commission through the ICT program under contract ICT-2007-216676 ECRYPT II.

[**] This author is supported by 973 Project (No.2007CB807902), NSFC Projects (No.61070244 and No.60931160442), Outstanding Young Scientists Foundation Grant of Shandong Province (No.BS2009DX030), IIFSDU Project (No. 2009TS087).

[***] This author is funded by a research grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

rion for new block ciphers. Algebraic cryptanalysis is a general method to attack ciphers. It has been widely used to cryptanalyze many primitives such as stream ciphers [13, 16], multivariate cryptosystems [19] and in particular block ciphers [14, 15, 17, 22]. The basic idea of algebraic cryptanalysis is to express the block cipher as a large multivariate polynomial system of equations. The secret key of the cipher is the solution of this system of equations. If the system is very sparse, overdefined or structured, it may be solved faster than a generic non-linear system of equations. By solving the system of equations for the block cipher, the key can be recovered with only a few plaintext-ciphertext pairs.

There are several methods to solve these systems of equations, such as computing a Gröbner basis or using a SAT solver. To compute a Gröbner basis, PolyBoRi [11] can be used. MiniSat [18] is a fast SAT solver. The advantage of computing a Gröbner basis is that useful equations can be generated, but this computation is typically slower than using a SAT solver and can more easily run out of memory.

However, the feasibility of algebraic cryptanalysis against block ciphers still remains a source of speculation. The main problem is that the size of the corresponding algebraic system is so large (thousands of variables and equations) that it seems infeasible to correctly predict the complexity of solving such polynomial systems. Therefore, algebraic cryptanalysis has so far had limited success in targeting modern block ciphers.

Recently, some works combining statistical cryptanalysis and algebraic cryptanalysis were presented [2–4, 20, 26]. Specifically, the combination of differential cryptanalysis and algebraic cryptanalysis appears to offer an advantage in reducing the data complexity. In [2, 3], Albrecht *et al.* propose new differential-algebraic cryptanalytic methods, which they refer to as Attack A, Attack B and Attack C. In order to describe them, let p denote the probability of the r-round differential characteristic for an N-round block cipher.

In Attack A, the system of equations consists of the equations of the plaintext bits, ciphertext bits, and subkey bits, the equations of the key schedule, and the linear equations resulting from the differential characteristic and the filter equations of the last $(N - r)$ rounds (i.e. the equations that must hold if the output difference after round r holds). Attack A recovers the key by solving this system of equations for each of the about $1/p$ plaintext-ciphertext pairs.

In Attack B, the same system of equations is used. The longest time to find that the system of equations is inconsistent, is measured. If this time is exceeded, a right pair is found with a high probability.

In Attack C, the system of equations only consists of the filter equations after r rounds for an r-round differential and the key schedule algorithm after r rounds. The conditions resulting from the differential characteristic and the conditions from the plaintext to the corresponding ciphertext are omitted in Attack C. The goal of Attack C in [3] is to filter out wrong pairs by solving the system of equations using tools such as PolyBoRi or MiniSat, and use the remaining right pair to recover the subkey bits.

In differential cryptanalysis, the filtering process can only filter out the wrong pairs according to the difference values of the ciphertext pairs. That is, after the filtering process, a lot of wrong pairs may still remain, which may increase the time complexity to recover the key in the differential attack. However, in Attack B and Attack C, Albrecht *et al.* claim that the right pairs can be identified with a good probability if the equations after the r-th round of the differential characteristic are inconsistent. They claim that with their technique, the time complexity will be lower than in the standard differential attack. Their work received a lot of attention in the cryptographic community [5,8,12,21,23], because it gives hope for the combination of a statistical attack and an algebraic attack.

In this paper, we will revisit the differential-algebraic attack given by Albrecht *et al.*, which they applied to PRESENT [9]. We find that Albrecht's method cannot filter out most of the wrong pairs satisfying the ciphertexts differences. However, we will show that wrong pairs that do not satisfy the ciphertext differences, can easily be filtered out without the algebraic method. Using [3,4], it is not possible to filter out more wrong pairs than using differential cryptanalysis.

Firstly, we show that Attack C typically cannot be used to filter out wrong pairs that do not satisfy the difference values of the ciphertexts to improve the differential cryptanalysis. Secondly, we verify using PolyBoRi and MiniSat2 that Attack B does not improve the current differential results for the PRESENT block cipher. The reason is that there are too few usable equations in the system of equations to derive an inconsistency for the wrong pairs or to find a solution for the right pairs. Based on our findings, we introduce two new methods that can more reliably use the right pairs to solve the right key within an acceptable time. For wrong pairs, no solution will be produced. One method is to fix certain key bits in the system of equations. This will allow an inconsistency to be derived faster. Another method is to use more than one plaintext-ciphertext pair to construct the system of equations.

We apply our attack methods to a reduced-round PRESENT block cipher. With the first method, we attack 15-round PRESENT-80 with 2^{59} chosen plaintexts and $2^{73.79}$ equivalent encryptions in the worst case. The 2R-differential attack on 15-round PRESENT-80 has a data complexity of more than 2^{59} and a time complexity of less than 2^{62} memory accesses. Therefore, the time complexity of the differential-algebraic attack for PRESENT-80 is much larger than that of the differential attack, but the data complexity is lower and the key does not have to be the same for every pair. If the number of chosen plaintext pairs that the attacker can obtain is limited, the algebraic-differential attack might be the only feasible attack. Note, however, that more rounds can be attacked in the case of PRESENT-80 using differential cryptanalysis (16 rounds instead of 15 rounds). We also provide a new attack on 14-round PRESENT-128 with a data complexity of 2^{55} chosen plaintexts and a worst-case time complexity of $2^{112.83}$ equivalent encryptions.

With our second method, the time complexity will be larger than with the first method for 15-round PRESENT-80. It is an open question whether the second method can offer an improvement for other block ciphers.

Our work also points out which equations are important in the differential-algebraic attack. With pure algebraic cryptanalysis, a 5-round PRESENT block cipher [15,22] can be attacked. Compared to this result, our differential-algebraic attack can attack more rounds, but the data complexity will be higher than that for the pure algebraic attack.

This paper is organized as follows. Section 2 describes Albrecht's differential-algebraic attack. In Sect. 3, we show why Attack C cannot filter out more wrong pairs than differential cryptanalysis for most block ciphers. We verify using PolyBoRi and MiniSat2 that Attack B cannot improve the differential cryptanalysis of the PRESENT block cipher. In Sect. 4, we present two methods that can be used to successfully solve the right key with the right pairs. Our attack methods are then applied to a reduced-round PRESENT block cipher. We conclude the paper in Sect. 5.

2 Description of Albrecht's Differential-Algebraic Attack

In [2,3], Albrecht *et al.* proposed three types of attacks that combine algebraic techniques with differential cryptanalysis. They are referred to as Attack A, Attack B and Attack C. We now describe these three types of attacks.

Attack A. For an r-round differential characteristic $\Delta = (\delta_0, \delta_1, \ldots, \delta_r)$, the probability of the differential characteristic is denoted by p. For a pair of plaintexts (P', P''), where $P' \oplus P'' = \delta_0$, and the corresponding ciphertexts (C', C''), two systems of equations F' and F'' are constructed under the same encryption key K. With the differential characteristic, the following linear equations are constructed:

$$X'_{i,j} \oplus X''_{i,j} = \Delta X_{i,j} \rightarrow \Delta Y_{i,j} = Y'_{i,j} \oplus Y''_{i,j} \ ,$$

where $X'_{i,j}$ and $X''_{i,j}$ are the j-th bit of the input to the S-box layer in round i for the systems F' and F'' respectively. The corresponding output bits are $Y'_{i,j}$ and $Y''_{i,j}$. The values resulting from the differential characteristic are $\Delta X_{i,j}$ and $\Delta Y_{i,j}$. The linear expressions corresponding to bits of active S-boxes hold with some non-negligible probability. For the non-active S-boxes, the following linear relations also hold with non-negligible probability:

$$X'_{i,j} \oplus X''_{i,j} = 0 = Y'_{i,j} \oplus Y''_{i,j} \ .$$

If the r-round differential characteristic is used to recover the key for N rounds, the differences from the $(r+1)$-th round to the N-th round can be derived from the output difference of the r-th round. Theses differences after the r-th round are described by equations. Attack A combines the two systems of equations F' and F'', the above linear relations resulting from the differential characteristic and the equations from the difference values after round r to produce the system of equations $\overline{F}$ that holds with probability p. If about $1/p$ systems corresponding to $1/p$ pairs of plaintext-ciphertext can be solved, a right pair is expected to be found which can then be used to obtain the right key. However, the time complexity to solve the system about $1/p$ times may be very high.

Attack B. Attack B uses the same system equations as Attack A to filter out the wrong pairs. In a differential attack, the ciphertext difference values are commonly used to filter out wrong pairs. However, in Attack B, by measuring the time t it maximally takes to find that the system is inconsistent, it is assumed that a right pair has been identified with high probability if a time t has elapsed without finding an inconsistency. More specifically, Attack B assumes that $\Delta Y_{1,j}$ holds with a high probability after time t has elapsed. With the remaining pairs, the subkey bits involved in the active S-boxes in the first round can be recovered. An alternative form of Attack B is to recover key bits from the last round. It is assumed that if time t passes for a given plaintext-ciphertext pair, a right pair has been found. In this case, some subkey bits in the last rounds will be fixed, and then it is checked whether time t still passes without contradiction. The time to find an inconsistency or a reduced-round PRESENT block cipher was measured in Appendix C of [3].

Attack C. In Attack C, the differential is used instead of the differential characteristic as in Attack B. If the r-round differential $\delta_0 \rightarrow \delta_r$ is used to recover the key for N rounds, the system of equations only consists of the equations resulting from the round functions from round $(r+1)$ to round N, the relations for the difference values from the $(r+1)$-th round to the N-th round, and the equations of key schedule from the $(r+1)$-th round to the N-th round. In this system of equations, there are no equations to restrict the relations between the plaintext and the corresponding ciphertext, and there are no equations for the difference values from the first round to the r-th round. By solving the system of equations and waiting for a fixed time t, a contradiction can be found in the system of equations. If one tested pair did not produce a contradiction after a fixed time, it is assumed to be a right pair satisfying the differential. Then with the right pair, the partial information for the subkey bits can be recovered. Appendix D in [3] measured the time to find an inconsistency for a reduced-round PRESENT block cipher. Based on this measured time, attacks on 16-round PRESENT-80, 17, 18 and 19 rounds of PRESENT-128 block cipher were given in [2, 3].

3 Inapplicability of Albrecht *et al.*'s Attacks

3.1 Inapplicability of Attack C

In this section, we will show that Attack C typically cannot be used to filter out the wrong pairs satisfying the difference values of the ciphertexts. Therefore, the right pairs cannot be identified and the key cannot be recovered. Moreover, Attack C can not filter out more wrong pairs than differential cryptanalysis to improve the differential cryptanalysis. As in the previous description, the system of equations in Attack C consists of the equations resulting from the round functions from round $(r+1)$ to round N, the relations resulting from the difference values from the $(r+1)$-th round to the N-th round, and the equations of key schedule from the $(r+1)$-th round to the N-th round. Let C_i' and C_i'' be the i-th bit of ciphertext pair C' and C'' respectively, and ΔC_i is the i-th bit of the difference value of ciphertext pair C' and C''. We then classify these equations into three groups, Group A, Group B and Group C.

Group A. The linear equations resulting from the difference values of ciphertexts corresponding to the non-active S-boxes in the last round are

$$\Delta C_i = C_i' \oplus C_i'' = 0 \ \ ,$$

where the i-th bit position corresponds to an output bit of any non-active S-box.

Group B. The equations resulting from the difference values of ciphertexts corresponding to the active S-boxes in the last round are

$$\begin{aligned}(\Delta C_{i_1} \parallel \Delta C_{i_2} \parallel \cdots \parallel \Delta C_{i_a}) &= \left(C_{i_1}' \parallel C_{i_2}' \parallel \cdots \parallel C_{i_a}'\right) \oplus \left(C_{i_1}'' \parallel C_{i_2}'' \parallel \cdots \parallel C_{i_a}''\right) \\ &= \delta_N, \delta_N \in \Gamma_N \ \ ,\end{aligned}$$

where $i_1, i_2, \ldots, i_a$ correspond to output bits of the active S-boxes, and Γ_N is the set of the ciphertext difference values.

Group C. The remaining equations are the equations resulting from the round functions from round $(r+1)$ to round N, the relations resulting from the difference values from the $(r+1)$-th round to the $(N-1)$-th round, and the equations of key schedule from the $(r+1)$-th round to the N-th round.

If a plaintext-ciphertext pair satisfies all the equations in Group A, Group B and Group C, it must be a right pair for the given differential. In the differential attack, the wrong pairs that do not satisfy the equations in Group A and Group B are easy to filter out using a look-up table combined with a time-memory tradeoff. Because the equations in Group C involve unknown subkey bits, they cannot easily be used to filter out the remaining wrong ciphertext pairs after the filtering process with the ciphertext differences. In Attack C, Albrecht *et al.* wish to measure the maximum time t to identify a pair as a wrong pair with all the equations in Group A, B and C. In fact, the equations in Group A and Group B can easily be used to find a contradiction because they are only related to the ciphertext difference values. For a typical block cipher, it is impossible to find contradictions for the equations in Group C. To understand why this is the case, we claim the following.

Claim 1. *If there is a wrong ciphertext pair that satisfies all the equations in Group A and Group B but does not satisfy the equations in Group C, it is* *impossible for a typical block cipher to find a contradiction for the equations in Group C.*

Proof. We consider a block cipher based on a substitution-permutation network (SPN). For other structures (Feistel, Generalized Feistel,...), a similar proof can be given. We assume that the difference value of the ciphertext pair satisfies the equations in Group A and Group B, but does not satisfy the equations in Group C. First, we will prove Claim 1 for a 1R-attack and extend the proof to an sR-attack[4] ($s = 1, 2, 3, \ldots$).

In a 1R-attack, the wrong ciphertext pair satisfies the output difference values of all non-active and active S-boxes in the last round, but does not satisfy the input difference of some active S-boxes in the last round. In most SPN block ciphers, after the S-box layer in the last round, the whitening subkeys will be XORed.

Let us introduce the shortened notation

$$X'_i \leftarrow X'_{i,j_1} || X'_{i,j_2} || \ldots || X'_{i,j_m} \ ,$$

where $X'_{i,j}$ is the j-th bit of the input to the S-box layer in round i. We can then describe the round function for the last round as follows:

$$\begin{aligned} Y'_N = S[X'_N] \ , \ C'_N = Y'_N \oplus K_N \ , \\ Y''_N = S[X''_N] \ , \ C''_N = Y''_N \oplus K_N \ , \end{aligned}$$

where X'_N and X''_N are the inputs of the S-box layer S in the last round for the system F' and F'' respectively, and Y'_N and Y''_N are the corresponding outputs. The values C'_N and C''_N are the ciphertext bits, and K'_N is the whitening subkey in the last round.

We now consider Fig. 1. Under the right key, the wrong ciphertext pair $(C' \oplus Z, C'' \oplus Z)$ will result in the output difference of the S-box Ω_e and the input difference of the S-box Ω_w, however, the right pair (C', C'') will result in the output difference and the input difference for the S-box as Ω_e and Ω_r respectively. As the subkey bits in the above equations are unknown variables, we will solve the following system of equations,

$$X'_N \oplus X''_N = \Omega_r.$$

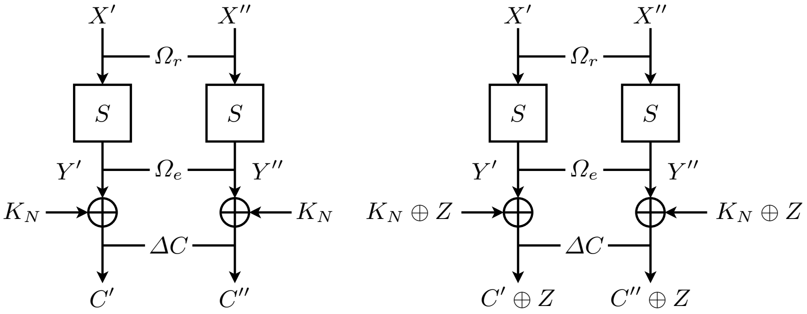
We can obtain

$$S^{-1}[Y'_N] \oplus S^{-1}[Y''_N] = \Omega_r,$$

where S^{-1} denotes the inverse S-boxes Layer. Then we have

$$S^{-1}[C'_N \oplus K_N] \oplus S^{-1}[C''_N \oplus K_N] = \Omega_r \ .$$

[4] An sR-attack means that the r-round differential is used to recover the key for $(r+s)$ rounds of the block cipher. We require in this paper that $s \ll N$, which is the case for typical differential attacks.



(a) Right pair, right key (b) Wrong pair, wrong key

Fig. 1. It is not possible to detect that $(C' \oplus Z, C'' \oplus Z)$ is a wrong pair (see Claim 1).

Because the right pair always can produce the difference from $\Omega_r \mapsto \Omega_e$ for the active S-boxes, there is at least one pair of input values (X_r', X_r'') and the corresponding output values (Y_r', Y_r'') satisfying the following equations:

$$X_r' \oplus X_r'' = \Omega_r, \;\; Y' \oplus Y'' = \Omega_e \;\; .$$

We have

$$S^{-1}[Y_r'] \oplus S^{-1}[Y_r''] = X_r' \oplus X_r'' = \Omega_r.$$

For the wrong pair $(C' \oplus Z, C'' \oplus Z)$, let the whitening subkey in the last round satisfy the following equations:

$$C_N' \oplus Z \oplus K_N = Y_r' \;\; , \;\; C_N'' \oplus Z \oplus K_N = Y_r'' \;\; .$$

The resulting wrong whitening subkey $K_N \oplus Z$ in the last round can make the wrong pair $(C' \oplus Z, C'' \oplus Z)$ produce the right input difference Ω_r, so the wrong pair $(C' \oplus Z, C'' \oplus Z)$ cannot be filtered out with the system of equations in the last round.

The proof for 1R-attack is helpful to understand the idea. The analysis of the sR-attack works in a similar way. As stated by Biham and Shamir [7] (and similarly by Selçuk [24]):

"Each surviving pair suggests several possible values for [the subkey] bits. Right pairs always suggest the correct value for [the subkey] bits (along with several wrong values), while wrong pairs suggest random values [for the subkey bits]."

This statement is true for typical block ciphers. Therefore, any remaining wrong pair must produce some solutions for the subkey satisfying the difference values in the last s-round. The solution may be the right subkey or the wrong subkey. Thus, it is impossible for most block ciphers to produce a contradiction for the sR-attack in the above s-round equations.

The equations for the key schedule may lead to a contradiction in Group C for the derived subkey value for the last s rounds, but the number of the subkey bits involved in the last s rounds is usually not large enough to produce a contradiction, assuming the key schedule is random. However, assume that the equations for the key schedule result in a contradiction for the subkey values of the last s rounds. Then, this contradiction holds for all values of the subkeys. That is, the contradiction is independent of the subkey values. The contradiction must be a contradiction on the difference of the ciphertext pair: a contradiction on the values of the ciphertext pair cannot appear because the ciphertext is calculated as $C = Y_N \oplus K_N$. Therefore, this contradiction can be included into Group A or Group B. Because the differential cryptanalysis attack uses the equations of Group A and Group B to filter the ciphertext values, an inconsistency in the key schedule does not improve the differential attack. □

In order to verify Claim 1, we tested the filtering time for different values of N and r of the PRESENT block cipher. In our tests, we constructed wrong ciphertext pairs that only satisfy the equations in Group A and Group B, but do not satisfy the equations in Group C when evaluated on the correct key. We used the source code provided by Albrecht [1] to apply Attack C with PolyBoRi-0.6 and MiniSat2. We performed a Gröbner basis computation to generate the filtering equations from the $(r+1)$-th round to the $(r+4)$-th round for the differential characteristic $(2 \leq r \leq 14)$ for PRESENT-80. These filtering equations can speed up the procedure of producing the contradiction.

However, there is no contradiction for any ciphertext pair with PolyBoRi-0.6 after six hours of computation. MiniSat2 always obtained the wrong solution for the key. In Table 1, we list these test results. For the wrong pairs under the right key, the wrong solution can be obtained within t seconds. We tested 20 wrong pairs for different values of r and N, and list one example of a wrong pair (P', P'') and the corresponding right key K. Due to space limitations, we only present the difference values for the wrong pair in the last row of Table 1 and the differential characteristic for the right pair in Table 2. In Table 2, the output difference for the wrong pair of the r-th $(r = 12)$ round is not equal to the output difference of the characteristic, but the output difference of the 13-th round is equal to the output difference of the characteristic. Therefore, this is a wrong pair.

At the same time, we construct the wrong ciphertext pairs for PRESENT-80 which do not satisfy the equations in any Group, the contradiction can be produced quickly and the filtering time is listed in Table 3. In addition, we construct some wrong ciphertext pairs that only satisfy the equation in Group A, the time to produce the contradiction is listed in Table 4. Moreover, we use a look-up table combined with a time-memory trade-off in differential cryptanalysis to filter out these pairs. As a result, our filter is more efficient than Attack C.

The computer we used is an IBM X3950 M2 with a CPU clock frequency of 2.4 GHz and 64GB RAM. From Tables 3 and 4, our test time with PolyBoRi approaches the corresponding time in Appendix D of [3], but our tested time with MiniSat2 is greater. The main reason is that our CPU is not same as Albrecht's. However, we can deduce that the wrong pairs Albrecht *et al.* used are wrong pairs that do not satisfy the equations in Group A or Group B, so they did not filter out wrong pairs that do satisfy the equations in Group A and Group B. Furthermore, even if Attack C is used as a filter for wrong pairs that do not satisfy the equations in Group A and Group B, its efficiency is much lower than the filter used in differential cryptanalysis. This shows that Attack C does not provide an advantage over differential cryptanalysis for most block ciphers.

Using Group A and Group B in a Differential Attack. We now clarify in more detail how the equations of Group A and Group B can be used in a differential attack. We consider two types of differential attacks:

(a) By generating a table of all possible ciphertext differences (corresponding to all solutions to the equations of Group A and Group B), wrong pairs can easily be filtered out. Because key counters will be used for the subkey bits corresponding to the active S-boxes, the number of output differences is less than the number of key counters required. Therefore, the table of all possible ciphertext differences provides only a relatively small overhead.

(b) In the filtering process, for each pair of ciphertexts (C', C''), a table is made of all possible input differences for the last round. This table does not depend on the value of the subkey bits in the last round. If we do not find a valid input difference for a particular pair of ciphertexts, this pair is identified as a wrong pair (i.e. it does not satisfy the equations of Group A and Group B). In this way, it is only necessary to make table of all input differences, and not all ciphertext differences. Typically, the table of all input differences should be small. For the remaining pairs, subkey bits in the last round will be guessed (instead of using key counters), to filter out pairs. For a wrong key, no pairs will remain, but the right pair will remain for the right key.

Note that (b) is in fact a time-memory trade-off applied to (a). In both (a) and (b), if output differences are invalid for some active S-boxes, they can be filtered using smaller tables. Then, the table that is described in (a) and (b) will be used to filter out the remaining pairs. In the next paragraph, we describe in detail how (a) can be used for a 2R attack on PRESENT. To construct a filter for a 3R and 4R attack on PRESENT, (b) can be used.

Relation to the Work of [4]. The equation system that Albrecht *et al.* set up in [4], is similar to the system of [3], except that the ciphertext bits (C'_i and C''_i) are variables instead of fixed values. This equation system is used to compute a Gröbner basis for PRESENT up to degree $D = 3$ using PolyBoRi. Polynomials that contain non-ciphertext variables are removed.

The resulting equations are used as a first filter for the ciphertext pairs. The probability p_1 that a random ciphertext pair passes the first filter, is estimated by Albrecht *et al.* as $p_1 \approx 2^{-50.669}$ for a 2R-attack on PRESENT-80 and PRESENT-128. Afterwards, [4] uses Attack C to filter out the remaining pairs. They estimate the total filtering probability $p_2 \approx 2^{-51.669}$ for PRESENT-80 and $p_2 \approx 2^{-51.361}$ for PRESENT-128.

For a 2R-attack on PRESENT, it is straightforward to write a fast program to compute the total number of ciphertext differences. We find that $11664 \approx 2^{13.51}$ ciphertext differences are possible, and store them in a small table. This results in the accurate filtering probability of $p_a = 2^{13.51}/2^{64} = 2^{-50.49}$ for both PRESENT-80 and PRESENT-128. When we derive the probability of p_1 ourselves, using the equations in [4, Fig. 2], we find that $p_1 = p_2 = p_a = 2^{-50.49}$. This confirms our result, and shows that the calculation of p_1 and p_2 in [4] is not correct. The accurate filtering probability p_a is slightly lower than the probability of the rough filter used by Wang [25].

By storing the output differences in a small table, we can easily filter out the wrong ciphertext pairs without using the algebraic method. Furthermore, we calculate that the reinterpretation of Attack C in [4] as a technique to filter ciphertext differences, does not result in a better filter. Therefore, Attack C does not provide an advantage over differential cryptanalysis in the case of a 2R-attack on PRESENT.

For a 3R-attack and a 4R-attack on PRESENT, we used a look-up table combined with a time-memory trade-off to filter out 1000 randomly generated wrong pairs. We note that although the filtering probability of our filter and Attack C is same, our filter is much faster than Attack C.

3.2 Inapplicability of Attack B to PRESENT

Attack B involves two other types of equations, besides the equations in Group A, Group B and Group C in Attack C. The first type of equations is the linear equations derived from the difference values from round 1 to round r, and the second type of equations is the round functions and the key schedule algorithm from round 1 to round r. In this way, the restriction from the plaintext to the corresponding ciphertext was added. Although we cannot show that Attack B does not provide an advantage over differential cryptanalysis for any block cipher, we make the following two observations for Attack B:

Observation 1. If N approaches the maximum number of rounds that can be attacked with a pure algebraic attack, the linear equations for the inner rounds and the round functions restricting the relation between the plaintext and the ciphertext are all usable to solve the system of equations. There are three possible subcases:

1. If the key size is much larger than the block size, for a wrong pair, the probability that a solution can be found for the key in the system of equations is non-negligible. In this way, there is a non-negligible probability that a contradiction for the wrong pairs cannot be produced. Attack B will likely fail.
2. If the key size is smaller than the block size, for a wrong pair, the probability that no solution can be found for the key in the system of equations is high. In this way, the contradiction for the wrong pairs can be produced and the right

solution for the right pair can be found with a high probability. Attack B is likely to succeed.

3. If the key size approaches the block size, Attack B can either succeed or fail.

Observation 2. If N is much larger than the maximum number of rounds that can be attacked with a pure algebraic attack, the linear equations for the inner rounds and the round functions and the key schedule algorithm for the inner rounds are not crucial to solve the system of equations. Only the equations for the outer rounds are relevant. We consider two subcases.

1. If there are few active S-boxes in the outer rounds, the restriction conditions are so few that a contradiction will be produced with low probability. Attack B will likely fail.
2. If there are many active S-boxes in the outer rounds, there are enough restriction conditions to derive a contradiction with high probability. Attack B is then likely to succeed.

In order to verify our observations for a small number of rounds, we apply Attack B to PRESENT-80 with for $N = 4$, $r = 3$. The block size and the key size for PRESENT-80 are 64 and 80, respectively. We have tested 10 wrong pairs satisfying the filter conditions in Group A and Group B, but not satisfying the conditions in Group C. We found that among 10 wrong pairs, only one wrong pair was filtered out within 1500 seconds. The reason is that the key size is larger than the block size.

As N and r increase, we ran several tests and list the results in Table 5. We identify different differential characteristics for the PRESENT-80 block cipher. For any value of r we tested, the characteristics have two active S-boxes from round 1 to round r. There will be two active S-boxes in round $(r+1)$ and 6, 7 or 8 active S-boxes in round $(r+2)$. Round $r+3$ has at least 12 active S-boxes and round $(r+4)$ has 16 active S-boxes. We use MiniSat2 to filter out the wrong pairs. For $N = r$, $N = r+1$ or $N = r+2$, no wrong pairs were filtered out. For $N = r+3$, very few wrong pairs were filtered out. Although for $N = r+4$, more wrong pairs were filtered out compared to $N = r+3$, lots of wrong pairs still remain. The reason is that there are more active S-boxes in round $(r+4)$ than in round $(r+3)$. This result is consistent with Table 10.8 of [2], where $N = r+4$ is used as well.

Further experiments are listed in Table 5. In Table 5, the plaintext pairs are all wrong pairs and we cannot filter them out within 1500 seconds. Even if wrong pairs can be filtered out after 1500 seconds, the time complexity of Attack B would become much higher than differential cryptanalysis. Due to space limitations, we only present the difference values for the pair in the last row of Table 5 and the characteristics for the right pair in Table 6. For the pair in Table 6, the output difference of the r-th $(r = 14)$ round is same as that of the characteristics, but the difference values from round 2 to round 10 are different from that of the characteristic. Therefore, this pair is a wrong pair. We also confirmed experimentally that Attack B cannot filter out wrong pairs that do not satisfy the output difference for the first round.

Observation 2 can be derived from the following statements:

1. SAT solvers use a tree-structured search algorithm, where branching is performed by heuristic guesses based on non-algebraic criteria. In order to reduce the search time, we must minimize both the average search depth and the dependencies of the unknown variables. In this way, those equations should be identified that tend to result in an inconsistency sooner.
2. In the system of equations in Attack B, the equations that lead to inconsistencies the soonest, are the equations related to the difference values, the round functions in the outer rounds such as the previous few rounds and the later few rounds. In contrast, the equations related to the difference values and the round functions in the inner rounds do not easily lead to inconsistencies. Therefore, the equations in the inner rounds can be removed in order to reduce the solving time.
3. Since the equations for the difference value in the outer rounds are very important for the solving process, we must obtain enough such equations to ensure there are enough restrictions for the dependent unknown subkey bits. If there are fewer active S-boxes in the outer rounds, there are not enough restrictions on the involved unknown subkey bits to obtain the right solution or filter out the wrong solutions. In other words, if there are more active S-boxes in the outer rounds, the solving process or the filtering process will be more efficient.

It is noted that if there are more active S-boxes in the outer rounds, the filtering process will be efficient, but it is not favorable to filter out the wrong ciphertext pairs directly according to the difference value of the ciphertexts. This will further increase the time complexity.

To overcome these problems, we propose the following two methods for the differential-algebraic attack. The first method is to fix certain key bits to ensure with a high probability that the right key can be recovered from the right pair. The second method has the same goal, but adds some extra equations. We will describe these two attacks in Sect. 4.

4 New Differential-Algebraic Attacks

In Sect. 3, we showed that neither Attack C nor Attack B can improve the differential cryptanalysis of the PRESENT block cipher. We also explained why Attack C does not provide an improvement for most block ciphers. The reason is that the attacks cannot filter out the wrong pairs satisfying the ciphertext difference values to identify the right pair. We present two methods that can find the right solution in acceptable time t, based on the system of equations constructed in Attack B. For the right pair, we can solve the right key within time t. If a pair cannot be filtered within time t, we discard it and consider another pair.

Attack 1 Based on Fixing Certain Key Bits. According to the key schedule algorithm and the outer rounds of the characteristic, fix the key bits related to the active S-boxes in the top rounds or the bottom rounds. In this way, inconsistencies can be found sooner. As we showed in Sect. 3.2, Attack B cannot be used to filter out most wrong pairs. Therefore, our attack fixes key bits in all tested pairs. The idea of fixing key bits was already proposed in [3]. The difference with Attack 1 is that we recover the entire key, and not only subkey bits from the last rounds.

Attack 2 Based on Multiple Pairs. Because the equations for the difference values in the outer rounds lead to inconsistencies sooner, appending more such equations will be helpful to find the inconsistency. Using multiple plaintext-ciphertext pairs to construct more equations of outer rounds will make the solving process or the filtering process more efficient. For example, if two plaintext-ciphertext pairs are used to perform the attack, the number of such equations will double. This means that if we use two right pairs to solve the system of equations, the right key can be found. However, if there is at least one wrong pair involved in the two pairs, the key cannot be found. In addition, if we use three plaintext-ciphertext pairs, the efficiency can be improved further. However, as the number of pairs increase, the number of combinations of pairs grows exponentially and the time complexity increases. So the number of pairs to construct the system of equations should not be too high.

Our experiments show that some wrong pairs can be filtered out quickly, but others cannot. However, if most of the wrong pairs cannot be filtered out, the attack becomes infeasible. So we attack the PRESENT block cipher with the above approaches and try to solve the right key with the right pairs.

4.1 Attack 1 for the PRESENT block cipher

We now apply Attack 1 to the PRESENT block cipher. The results are listed in Table 7. If we use $r = 13$ to attack $N = 15$ rounds of PRESENT-80, the probability of the characteristic is 2^{-58} (using the last 13 rounds of the 14-round characteristic of [25]). The filtering probability according to the difference value for the ciphertext pair is $2^{-50.49}$ (as calculated at the end of Sect. 3.1). The CPU clock frequency is 2.4 GHz. From Table 7, we find that it takes at most 523.16 s to find an inconsistency. The table also shows that we should guess at least 34 key bits, so the time complexity will be $2^{34} \cdot 2^{58-50.49} \cdot 2.4 \cdot 10^9 \cdot 523.16 = 2^{34} \cdot 2^{7.51} \cdot 2^{31.16} \cdot 2^{9.03} = 2^{81.70}$ CPU cycles. We assume that a single encryption costs at least 16 CPU cycles per round[5]. Therefore, the time complexity for our attack ($2^{73.79}$ equivalent encryptions) is better than exhaustive search (2^{80}).[6] The data complexity is 2^{59} chosen plaintexts. For the 2R-differential attack, the

[5] The bitsliced implementation of PRESENT by Albrecht achieves 16.5 cycles per round [2].

[6] We used 20 trials to obtain time t. Although more trials may result in a longer time t, we expect that our attack will still be much faster than exhaustive search.

data complexity must be higher than 2^{59} chosen plaintexts, because then one right plaintext-ciphertext pair is not sufficient to recover the key with a high success probability. However, the time complexity of the 15-round 2R-differential attack must be lower than 2^{62} memory accesses (the time complexity given for the 16-round differential attack in [25]). Depending on the processor, one memory access requires about 2 to 10 CPU cycles. This means the complexity of the differential-algebraic attack for PRESENT-80 is much higher than that of the differential attack, but the data complexity is lower. Depending on how many chosen plaintext-ciphertext pairs the attacker can obtain, the algebraic-differential attack might however be the only feasible attack.

For PRESENT-128, we could not identify the right pairs for $r > 12$ using the method from [2]. If we use the 12-round differential characteristic with the probability 2^{-54} to attack 14-round PRESENT-128, the time complexity will be about $2^{78+54-50.49+31.16+7.97} = 2^{120.64}$ CPU cycles, or about $2^{112.83}$ equivalent encryptions. The data complexity is 2^{55} chosen plaintexts.

4.2 Attack 2 for the PRESENT block cipher

We respectively use two pairs and three pairs to attack the PRESENT. The test results are listed in Tables 8 and 9. For the right pairs, the right key can be solved within t seconds. We ran 10 trails for different values of r and N, and one example of right pairs $\{(P'_0, P''_0), (P'_1, P''_1)\}$ or $\{(P'_0, P''_0), (P'_1, P''_1), (P'_2, P''_2)\}$ and list the corresponding right key K. As in Attack 1, we can solve the right key from the right pairs, but the wrong pairs cannot always be filtered out. So we perform the test with the right pairs to recover the right key. We obtained the following results:

1. For $N = r + 3$ or $N = r + 4$ rounds of PRESENT-80 with the r-round differential characteristic, the right key can be solved with the two right pairs. Some test results are listed in Table 8. However, because we use two right pairs, this means that if m pairs of ciphertexts remain after filtering according to the ciphertext difference, we must consider $\binom{m}{2}$ combinations of two pairs. However, the solving time for $\binom{m}{2}$ combinations of two pairs becomes unacceptable. If we attack 16-round PRESENT-80 with a 13-round differential characteristic with the probability 2^{-58}, we choose 2^{59} pairs of plaintexts and the filtering probability with the ciphertext difference is about $2^{-25.711}$, so the number of the remaining ciphertext pairs is about $2^{33.289}$ which will be combined to produce $2^{65.578}$ combinations of two pairs. The time complexity will be $2^{65.578} \cdot 2^{31.16} \cdot t > 2^{88}$. We have not identified the right pairs for $r = 13$, so we cannot test the time for t and it should be more than 100 seconds according to the test time for $r < 13$. Therefore, Attack 2 is slower than exhaustive search.
2. For $N = r + 2$ rounds of PRESENT-80, only few combinations of two right pairs can be used to solve the right key, so the success rate is too low.
3. For $N = r + 4$ rounds of PRESENT-128 with the r-round differential, only few combinations of two right pairs can be used to recover the right key and the success rate is also very low.

4. For $N = r+3$ rounds of PRESENT-80 and $N = r+4$ rounds of PRESENT-128 with the r-round differential, the right key can be solved with the three right pairs. The test results are listed in Table 9. However, because we use three pairs, this means that if m pairs of ciphertexts remain, there are $\binom{m}{3}$ combinations of three pairs. However, the solving time for $\binom{m}{3}$ combinations of three pairs becomes unacceptable.

From the above results, Attack 2 (using two pairs or three pairs for PRESENT) has no advantage over Attack 1 (fixing certain key bits). Maybe these attacks have some advantage for other ciphers. For example, if there would be more active S-boxes involved in the outer rounds in PRESENT, maybe we could obtain the right key using two right pairs with a high success probability.

5 Conclusion

The cryptanalytic method combining differential cryptanalysis and algebraic cryptanalysis has been a focus topic in the field of the cryptanalysis of symmetric ciphers. At FSE 2009, Albrecht *et al.* propose new differential-algebraic attacks, which they claim improves the results of the differential cryptanalysis. In this paper, we revisited Albrecht's cryptanalytic method and identified that the time complexity to identify the right pairs is not correct. Firstly, we showed that Attack C cannot be used to filter out the wrong pairs satisfying the difference value of the ciphertexts for most block ciphers to improve the differential cryptanalysis. We identified some important properties for Attack B and showed that Attack B does not provide an advantage over differential cryptanalysis for PRESENT. Faugère *et al.* presented a similar attack for DES, however, they could only attack 8-round DES with a 5-round differential characteristic. Their attack for DES is accordant with our Observation 1 in Sect. 3.2 because the key size for DES is smaller than the block size.

In this paper, we introduce two new methods to perform a differential-algebraic attack. The first method is to fix certain key bits to solve the system of equations and the second method is to use multiple pairs to construct the system of equations. This method is more efficient for the PRESENT block cipher and its data complexity is better than that of the differential attack, but the time complexity is worse. Although we did not significantly improve the results of the differential cryptanalysis for PRESENT, our work indicates which equations are important in the differential-algebraic attack. For the differential-algebraic attack, we obtain the following three conclusions:

1. Compared with the differential cryptanalysis, the differential-algebraic attack can reduce the data complexity, but the time complexity increases. Compared with the algebraic cryptanalysis, the differential-algebraic attack can attack more rounds because the relations resulting from the differential characteristic are very important for the solving process.

2. In order to make the solving process in the differential-algebraic attack more efficient, more active S-boxes should be involved in the outer rounds. However, more active S-boxes will reduce the filtering probability with the ciphertext difference and it will increase the time complexity. The lower bound for the number of the active S-boxes should be used to ensure the system of equations can be solved reliably. The detailed analysis of this case can be seen as future work.
3. If the methods to solve systems of equations can be improved, and if the computational power available increases, we expect that differential-algebraic attacks will gain in importance.

Acknowledgments. The authors would like thank the anonymous reviewers for their detailed comments and suggestions.

References

1. Albrecht, M. Tools for the algebraic cryptanalysis of cryptographic primitives. `http://www.ecrypt.eu.org/tools/tools-for-algebraic-cryptanalysis`.
2. Albrecht, M. *Algorithmic Algebraic Techniques and their Application to Block Cipher Cryptanalysis*. PhD thesis, Royal Holloway, University of London, 2010.
3. Albrecht, M., and Cid, C. Algebraic Techniques in Differential Cryptanalysis. In *FSE* (2009), O. Dunkelman, Ed., vol. 5665 of *LNCS*, Springer, pp. 193–208.
4. Albrecht, M., Cid, C., Dullien, T., Faugère, J.-C., and Perret, L. Algebraic precomputations in differential and integral cryptanalysis. In *INSCRYPT 2010 (to appear)* (2010), p. 18 pages.
5. Bard, G. V. *Algebraic Cryptanalysis*, vol. XXXIV of *Security and Cryptology*. Springer, 2009.
6. Biham, E., and Shamir, A. Differential Cryptanalysis of DES-like Cryptosystems. *J. Cryptology 4*, 1 (1991), 3–72.
7. Biham, E., and Shamir, A. Differential Cryptanalysis of the Full 16-Round DES. In *CRYPTO* (1992), E. F. Brickell, Ed., vol. 740 of *LNCS*, Springer, pp. 487–496.
8. Blondeau, C., and Gérard, B. Multiple Differential Cryptanalysis: Theory and Practice. In *FSE* (2011), A. Joux, Ed., vol. 6733 of *LNCS*, Springer, p. 20.
9. Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., and Vikkelsoe, C. PRESENT: An Ultra-Lightweight Block Cipher. In *CHES* (2007), P. Paillier and I. Verbauwhede, Eds., vol. 4727 of *LNCS*, Springer, pp. 450–466.
10. Boneh, D., Ed. *Advances in Cryptology - CRYPTO 2003, 23rd Annual International Cryptology Conference, Santa Barbara, California, USA, August 17-21, 2003, Proceedings* (2003), vol. 2729 of *LNCS*, Springer.
11. Brickenstein, M., and Dreyer, A. PolyBoRi: A framework for Gröbner-basis computations with Boolean polynomials. *J. Symb. Comput. 44*, 9 (2009), 1326–1345.
12. Cho, J. Y. Linear Cryptanalysis of Reduced-Round PRESENT. In *CT-RSA* (2010), J. Pieprzyk, Ed., vol. 5985 of *LNCS*, Springer, pp. 302–317.
13. Courtois, N. Fast Algebraic Attacks on Stream Ciphers with Linear Feedback. In Boneh [10], pp. 176–194.
14. Courtois, N., and Bard, G. V. Algebraic Cryptanalysis of the Data Encryption Standard. In *IMA Int. Conf.* (2007), S. D. Galbraith, Ed., vol. 4887 of *LNCS*, Springer, pp. 152–169.

15. Courtois, N., and Debraize, B. Specific S-Box Criteria in Algebraic Attacks on Block Ciphers with Several Known Plaintexts. In *WEWoRC* (2007), S. Lucks, A.-R. Sadeghi, and C. Wolf, Eds., vol. 4945 of *LNCS*, Springer, pp. 100–113.
16. Courtois, N., and Meier, W. Algebraic Attacks on Stream Ciphers with Linear Feedback. In *EUROCRYPT* (2003), E. Biham, Ed., vol. 2656 of *LNCS*, Springer, pp. 345–359.
17. Courtois, N., and Pieprzyk, J. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In *ASIACRYPT* (2002), Y. Zheng, Ed., vol. 2501 of *LNCS*, Springer, pp. 267–287.
18. Eén, N., and Sörensson, N. An Extensible SAT-solver. In *SAT* (2003), E. Giunchiglia and A. Tacchella, Eds., vol. 2919 of *LNCS*, Springer, pp. 502–518.
19. Faugère, J.-C., and Joux, A. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. In Boneh [10], pp. 44–60.
20. Faugre, J.-C., Perret, L., and Spaenlehauer, P.-J. Algebraic-Differential Cryptanalysis of DES. In *Western European Workshop on Research in Cryptology - WEWoRC 2009* (2009), pp. 1–5.
21. Gong, Z., Hartel, P. H., Nikova, S., and Zhu, B. Towards Secure and Practical MACs for Body Sensor Networks. In *INDOCRYPT* (2009), B. K. Roy and N. Sendrier, Eds., vol. 5922 of *LNCS*, Springer, pp. 182–198.
22. Nakahara, J., Sepehrdad, P., Zhang, B., and Wang, M. Linear (Hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT. In *CANS* (2009), J. A. Garay, A. Miyaji, and A. Otsuka, Eds., vol. 5888 of *LNCS*, Springer, pp. 58–75.
23. Özen, O., Varici, K., Tezcan, C., and Çelebi Kocair. Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT. In *ACISP* (2009), C. Boyd and J. M. G. Nieto, Eds., vol. 5594 of *LNCS*, Springer, pp. 90–107.
24. Selçuk, A. A. On Probability of Success in Linear and Differential Cryptanalysis. *J. Cryptology 21*, 1 (2008), 131–147.
25. Wang, M. Differential Cryptanalysis of Reduced-Round PRESENT. In *AFRICACRYPT* (2008), S. Vaudenay, Ed., vol. 5023 of *LNCS*, Springer, pp. 40–49.
26. Wang, M., Wang, X., and Hui, L. C. Differential-algebraic cryptanalysis of reduced-round of Serpent-256. *SCIENCE CHINA Information Sciences 53*, 3 (2010), 546–556.

Table 1. Attack C's Filtering Test for Wrong Pairs with MiniSat2

N	r	P'	P''	K	$t(s)$
8-10	7	$8b29917c174f21b7$	$8c29917c174f26b7$	$2b8bc6ad5d4b869101c2$	12.20-12.77
9-11	8	$d549bf122a09edfa$	$d249bf122a09eafa$	$5d05c98dce5da5894fc5$	12.26-12.92
10-12	9	$f5fc5a0d3979d9d3$	$f2fc5a0d3979ded3$	$f53e4ecaf9ce361ee6d7$	12.11-13.03
11-13	10	$50d752ee7f6017d7$	$57d752ee7f6010d7$	$afc238c99ce160d8254b$	12.22-12.73
12-14	11	$155fdec5b70e8b3a$	$125fdec5b70e8c3a$	$b544c98fce9474d53925$	12.33-12.92
13-15	12	$504ad07e763a8289$	$574ad07e763a8589$	$a7ece17b6ab73269d7e9$	12.01-12.71

N: the round number we attack; r: the round number of the differential; K: right key; (P', P''): one example of wrong pairs; t: the wrong solution obtained within t seconds.

Table 2. Difference Values for Wrong Pair and Right Pair in Attack C

R		Δ_{wrong}	Δ_{right}	R		Δ_{wrong}	Δ_{right}
I		$x_2 = 7, x_{14} = 7$	$x_2 = 1, x_{14} = 1$				
R1	S	$x_2 = 1, x_{14} = 1$	$x_2 = 1, x_{14} = 1$	R8	S	$x_0 = 9, x_2 = 9$	$x_8 = 9, x_{10} = 9$
R1	P	$x_0 = 4, x_3 = 4$	$x_0 = 4, x_3 = 4$	R8	P	$x_0 = 5, x_{12} = 5$	$x_2 = 5, x_{14} = 5$
R2	S	$x_0 = 5, x_3 = 5$	$x_0 = 5, x_3 = 5$	R9	S	$x_0 = 1, x_{12} = 1$	$x_2 = 1, x_{14} = 1$
R2	P	$x_0 = 9, x_8 = 9$	$x_0 = 9, x_8 = 9$	R9	P	$x_0 = 1, x_3 = 1$	$x_0 = 4, x_3 = 4$
R3	S	$x_0 = 4, x_8 = 4$	$x_0 = 4, x_8 = 4$	R10	S	$x_0 = 3, x_3 = 3$	$x_0 = 5, x_3 = 5$
R3	P	$x_8 = 1, x_{10} = 1$	$x_8 = 1, x_{10} = 1$	R10	P	$x_0 = 9, x_4 = 9$	$x_0 = 9, x_8 = 9$
R4	S	$x_8 = 3, x_{10} = 3$	$x_8 = 9, x_{11} = 9$	R11	S	$x_0 = 4, x_4 = 4$	$x_0 = 4, x_8 = 4$
R4	P	$x_2 = 5, x_6 = 5$	$x_2 = 5, x_{14} = 5$	R11	P	$x_8 = 1, x_9 = 1$	$x_8 = 1, x_{10} = 1$
R5	S	$x_2 = 1, x_6 = 1$	$x_2 = 1, x_{14} = 1$	R12	S	$x_8 = 9, x_9 = 9$	$x_8 = 9, x_{10} = 9$
R5	P	$x_0 = 4, x_1 = 4$	$x_0 = 4, x_3 = 4$	R12	P	$\mathbf{x_2 = 3, x_{14} = 3}$	$\mathbf{x_2 = 5, x_{14} = 5}$
R6	S	$x_0 = 5, x_1 = 5$	$x_0 = 5, x_3 = 5$	R13	S	$\mathbf{x_2 = 1, x_{14} = 1}$	$\mathbf{x_2 = 1, x_{14} = 1}$
R6	P	$x_0 = 3, x_8 = 3$	$x_0 = 9, x_8 = 9$	R13	P	$\mathbf{x_0 = 4, x_3 = 4}$	$\mathbf{x_0 = 4, x_3 = 4}$
R7	S	$x_0 = 1, x_8 = 1$	$x_0 = 4, x_8 = 4$				
R7	P	$x_0 = 1, x_2 = 1$	$x_8 = 1, x_{10} = 1$				

Rj: output difference after round j (S: after S-box layer,
P: after permutation layer); Δ_{wrong}: differential value for wrong pair;
Δ_{right}: differential value for right pair.

Table 3. Filter Time for Wrong Pairs Not Satisfying Equations in any Group

N	r	♯trails	PolyBoRi	MiniSat2	N	r	♯trails	PolyBoRi	MiniSat2
9	8	20	3.51-3.85	4.06-4.64	13	12	20	4.99-5.34	4.96-5.25
10	8	20	4.89-5.23	7.57-8.44	14	12	20	6.67-6.83	8.86-9.26
11	8	20	7.89-8.41	11.29-12.34	15	12	20	9.69-10.20	12.80-13.15
10	9	20	3.92-4.27	4.55-4.79	14	13	20	5.66-5.78	5.07-5.37
11	9	20	5.32-5.66	8.40-8.66	15	13	20	7.02-7.50	9.08-9.38
12	9	20	6.24-6.59	12.19-12.45	16	13	20	7.99-8.51	12.91-13.58
11	10	20	4.28-4.67	4.73-4.99	15	14	20	6.06-6.18	5.24-5.52
12	10	20	4.75-5.09	8.35-8.59	16	14	20	6.50-6.95	9.04-9.47
13	10	20	6.93-7.05	12.32-12.59	17	14	20	8.48-8.88	13.17-13.77
12	11	20	4.66-5.02	4.87-5.12					
13	11	20	6.09-6.42	8.69-8.97					
14	11	20	7.41-10.17	12.42-12.75					

♯trails: the number of wrong pairs we test;
PolyBoRi: the filtering time in seconds with PolyBori;
MiniSat2: the filtering time in seconds with Minisat2.

Table 4. Filter Time for Wrong Pairs Only Satisfying Equations in Group A

N	r	♯trails	PolyBoRi	MiniSat2
10	8	20	5.07-5.55	8.09-8.53
11	9	20	6.33-6.68	7.34-7.81
12	10	20	6.02-6.45	7.53-8.12

Table 5. Attack B's Filtering Test for Wrong Pairs Satisfying Ciphertext Difference Values with MiniSat2 (Timeout $t = 1500$ s)

N	r	P'	P''	K
5-7	4	$67279b1efdb93674$	$60279b1efdb93174$	$9ad864e12a6ecc872280$
6-8	5	$cdc43299824183d4$	$cac43299824184d4$	$70be32f5dd35396cdbfd$
7-9	6	$bc887a5de0597dd6$	$bb887a5de0597ad6$	$716d9698292707b0b6da$
8-10	7	$c53f11ab7329e7cf$	$c23f11ab7329e0cf$	$78bf3977acaffded898a$
9-11	8	$6d736a36a28d4f93$	$6a736a36a28d4893$	$5e7f5234d2063c5dd11d$
10-12	9	$94bd4ffd6585072e$	$93bd4ffd6585002e$	$1e00538c107f7abc4a73$
11,12,13	10	$f02f740d8d4b6d37$	$f72f740d8d4b6a37$	$df76f9fdaf4ead07d9a2$
12,13,14	11	$85f4ab19cf1dd9ac$	$82f4ab19cf1ddeac$	$5d0de0769a874e36d362$
13,14,15	12	$ca8b8755e65217af$	$cd8b8755e65210af$	$2d0d71c7a40d3084ac3a$
15,16,17	14	$934c64486fa9ed41$	$944c64486fa9ea41$	$8b1c1828ec601df09214$

Table 6. Difference Values for Wrong Pair and Right Pair in Attack B

R		Δ_{wrong}	Δ_{right}	R		Δ_{wrong}	Δ_{right}
I		$x_2 = 7, x_{14} = 7$	$x_2 = 7, x_{14} = 7$				
R1	S	$x_2 = 1, x_{14} = 1$	$x_2 = 1, x_{14} = 1$	R8	S	$x_8 = 5, x_{10} = 5$	$x_8 = 9, x_{10} = 9$
R1	P	$x_0 = 4, x_3 = 4$	$x_0 = 4, x_3 = 4$	R8	P	$x_2 = 5, x_{10} = 5$	$x_2 = 5, x_{14} = 5$
R2	S	$x_0 = 9, x_3 = 9$	$x_0 = 5, x_3 = 5$	R9	S	$x_2 = 1, x_{10} = 1$	$x_2 = 1, x_{14} = 1$
R2	P	$x_0 = 9, x_{12} = 9$	$x_0 = 9, x_8 = 9$	R9	P	$x_0 = 4, x_2 = 4$	$x_0 = 4, x_3 = 4$
R3	S	$x_0 = 4, x_{12} = 4$	$x_0 = 4, x_8 = 4$	R10	S	$x_0 = 5, x_2 = 5$	$x_0 = 5, x_3 = 5$
R3	P	$x_8 = 1, x_{11} = 1$	$x_8 = 1, x_{10} = 1$	R10	P	$x_0 = 5, x_8 = 5$	$x_0 = 9, x_8 = 9$
R4	S	$x_8 = 9, x_{11} = 9$	$x_8 = 9, x_{10} = 9$	R11	S	$x_0 = 4, x_8 = 4$	$x_0 = 4, x_8 = 4$
R4	P	$x_2 = 9, x_{14} = 9$	$x_2 = 5, x_{14} = 5$	R11	P	$x_8 = 1, x_{10} = 1$	$x_8 = 1, x_{10} = 1$
R5	S	$x_2 = 4, x_{14} = 4$	$x_2 = 1, x_{14} = 1$	R12	S	$x_8 = 9, x_{10} = 9$	$x_8 = 9, x_{10} = 9$
R5	P	$x_8 = 4, x_{11} = 4$	$x_0 = 4, x_3 = 4$	R12	P	$x_2 = 5, x_{14} = 5$	$x_2 = 5, x_{14} = 5$
R6	S	$x_8 = 5, x_{11} = 5$	$x_0 = 5, x_3 = 5$	R13	S	$x_2 = 1, x_{14} = 1$	$x_2 = 1, x_{14} = 1$
R6	P	$x_2 = 9, x_{10} = 9$	$x_0 = 9, x_8 = 9$	R13	P	$x_0 = 4, x_3 = 4$	$x_0 = 4, x_3 = 4$
R7	S	$x_2 = 4, x_{10} = 4$	$x_0 = 4, x_8 = 4$	R14	S	$x_2 = 4, x_{10} = 4$	$x_0 = 4, x_8 = 4$
R7	P	$x_8 = 4, x_{10} = 4$	$x_8 = 1, x_{10} = 1$	R14	P	$x_0 = 9, x_8 = 9$	$x_0 = 9, x_8 = 9$

Rj: output difference after round j (S: after S-box layer,
P: after permutation layer); Δ_{wrong}: differential value for wrong pair;
Δ_{right}: differential value for right pair.

Table 7. Time to Solve Right Key under Some Fixed Key Bits with MiniSat2

K_s	N	r	$\sharp$trails	N_k	$t(s)$	K_s	N	r	$\sharp$trails	N_k	$t(s)$
80	10	10	20	32	45.18-285.20	80	14-17	14	20	36	63.47-120.08
80	11	10	20	32	64.45-564.87	128	10	10	20	79	43.75-288.63
80	12	10	20	32	61.88-591.56	128	11	10	20	78	63.38-821.45
80	13	10	20	32	53.49-497.96	128	12	10	20	75	79.83-966.38
80	11	11	20	33	60.19-151.28	128	13	10	20	72	89.15-751.30
80	12	11	20	33	53.01-316.94	128	11	11	20	79	98.35-662.19
80	13	11	20	33	56.64-528.03	128	12	11	20	79	58.73-483.92
80	14	11	20	33	56.25-104.26	128	13	11	20	79	69.41-805.18
80	12	12	20	34	97.19-487.77	128	14	11	20	71	78.20-891.08
80	13	12	20	34	69.24-680.41	128	12	12	20	82	57.35-115.11
80	14	12	20	34	61.09-110.02	128	13	12	20	82	118.08-668.53
80	15	12	20	34	59.25-77.82	128	14	12	20	78	61.84-251.14
80	13-16	13	20	34	85.54-523.16	128	15	12	20	66	64.86-309.90

N_k: the number of fixed key bits.

Table 8. Time to Solve Right Key using Two Right Pairs with MiniSat2

K_s	N	r	P_0', P_1'	P_0'', P_1''	K	$t(s)$
80	12	9	$39121b2bffad3bbc$, $91f1a75a4f4d33e0$	$3e121b2bffad3cbc$, $96f1a75a4f4d34e0$	$4634342e33 \Vert$ $0d53e8cd71$	132.88-377.13
80	13	10	$67bb6eecd081767c$, $6f62c9bd561f718e$	$60bb6eecd081717c$, $6862c9bd561f768e$	$6fcaf3033d \Vert$ $39296c0f66$	122.00-849.89
80	14	11	$c2b3135aa3b8f3b4$, $8a43480c3122ab14$	$c5b3135aa3b8f4b4$, $8d43480c3122ac14$	$22c587b7b2 \Vert$ $607cddab90$	129.01-213.98
80	15	12	$c2b3135aa3b8f3b4$, $85c6576306a6a545$	$125fcb08afed6df3$, $82c6576306a6a245$	$155fcb08af \Vert$ $ed6af317f1$	133.64-141.75
80	13	9	$0c03406225bf97cd$, $0bbd25aea7c5b0c9$	$0b03406225bf90cd$, $0cbd25aea7c5b7c9$	$cca9deeb2c \Vert$ $0d98071ca6$	115.61-133.35
80	14	10	$9434381cb8083429$, $0b40a64e215244c6$	$9334381cb8083329$, $0c40a64e215243c6$	$ab7b47fdf8 \Vert$ $93fb87c9cd$	124.22-132.99
80	15	11	$8814d6bea07fd660$, $f02e367f419a412e$	$8f14d6bea07fd160$, $f72e367f419a462e$	$a7d16cda8d \Vert$ $b76ec42756$	130.48-144.89
80	16	12	$cbaef2f923614742$, $b37ee1f334c4207b$	$ccaef2f923614042$, $b47ee1f334c4277b$	$6b9b4087a6 \Vert$ $254f2bbef2$	189.26-280.49

Table 9. Time to Solve Right Key using Three Right Pairs with MiniSat2

K_s	N	r	P'_0,P'_1,P'_2	P''_0,P''_1,P''_2	K	$t(s)$
80	11	9	d9591ff50fc1df6d, f9866c0009f3bf44, 0e768137f568779d	de591ff50fc1d86d, fe866c0009f3b844, 09768137f568709d	66efab8af3 $\|$ 74afe67553	177.77-1402.2
80	12	10	3a659aa3dc72107c, 62129df1a637b88f, c566bb319010f0df	3d659aa3dc72177c, 65129df1a637bf8f, c266bb319010f7df	2dc9fceff3 $\|$ 174f9919c4	240.70-578.68
80	13	11	383663a9bc01cec5, 88042f67e3b59e95, c842b19a415d9105	3f3663a9bc01c9c5, 8f042f67e3b59995, cf42b19a415d9605	a0f5a7209b $\|$ b95180a21c	247.53-t ($t > 2500$)
80	14	12	2ddbc9427defb9ee, 2aa2624e2cb1dede, 4d19fefd126a29ee	2adbc9427defbeee, 2da2624e2cb1d9de, 4a19fefd126a2eee	3200679dd6 $\|$ 3d29ae18bc	293.21-408.40
80	12	9	3d84126858c7435e, 32a6811bd0c6a32e, cd66cbdb18c23c55	3a84126858c7445e, 35a6811bd0c6a42e, ca66cbdb18c23b55	5da70ed0b5 $\|$ 13fb14435c	216.35-239.90
80	13	10	e519cccfa40ce691, e5aa80afcfc216a3, 8a179faf87127908	e219cccfa40ce191, e2aa80afcfc211a3, 8d179faf87127e08	72ada6021d $\|$ d2667ab4e5	238.47-258.13
80	14	11	f5a33b54749b6624, b2f64b6c661d6101, 2d106b5e6d2b4e24	f2a33b54749b6124, b5f64b6c661d6601, 2a106b5e6d2b4924	8ab6e28d86 $\|$ 9ef6858a87	292.15-319.56
80	15	12	e6005b48d2abd194, 41909dfa1ac196d9, 0e43381eb485d900	e1005b48d2abd694, 46909dfa1ac191d9, 0943381eb485de00	393d660706 $\|$ 1dbe32c806	271.31-340.26
128	13	9	9d6902f268514522, 95d585a882e6e250, 2da0d2114f1805c2	9a6902f268514222, 92d585a882e6e550, 2aa0d2114f1802c2	0578224d0c9eba10 $\|$ bb0fd3b56d8b4834	235.64-265.20
128	14	10	972331fa763f86bd, 50d342a2a6dce17a, efdfd44485f1ee81	902331fa763f81bd, 57d342a2a6dce67a, e8dfd44485f1e981	d8ca446899016e69 $\|$ 17641f71e11d09f5	235.16-291.02
128	15	11	76971713b1f0d438, aed2ee07ad11dc6d, e609bfed79d4143b	71971713b1f0d338, a9d2ee07ad11db6d, e109bfed79d4133b	9e3328405c865b25 $\|$ 2201229c273fd1dd	285.00-303.82
128	16	12	eb449a907d31f33e, 84363465aaddb304, e3a2e5866f5814a9	ec449a907d31f43e, 83363465aaddb404, e4a2e5866f5813a9	73fdf364db99c472 $\|$ bb7a8e563b20a1f2	316.21-414.30