Note on Impossible Differential Attacks

Patrick Derbez
{patrick.derbez}@irisa.fr

Université Rennes 1 / IRISA

Abstract. While impossible differential cryptanalysis is a well-known and popular cryptanalytic method, errors in the analysis are often discovered and many papers in the literature present flaws. Wishing to solve that, Boura *et al.* [1] presented at ASIACRYPT'14 a generic vision of impossible differential attacks with the aim of simplifying and helping the construction and verification of this type of cryptanalysis. In particular, they gave generic complexity analysis formulas for mounting such attacks and develop new ideas for optimizing them.
In this paper we carefully study this generic formula and show impossible differential attacks for which the real time complexity is much higher than estimated by it. In particular, we show that the impossible differential attack against 25-round TWINE-128, presented at FSE'15 by Biryukov *et al.* [2], actually has a complexity higher than the natural bound of exhaustive search.
Keywords: truncated impossible differential, cryptanalysis, block cipher, TWINE, complexity

1 Introduction

Impossible differential cryptanalysis, which was independently introduced by Knudsen [3] and Biham *et al.* [4], is well-known and popular cryptanalytic method. Unlike differential attacks [5] that exploit differential characteristics of high probability, the aim of impossible differential cryptanalysis is to use differentials that have a probability of zero to occur in order to eliminate the key candidates leading to such impossible transitions. The first step to mount an impossible differential attack is to find an impossible differential covering a large number of rounds. This is a procedure that has been extensively studied and several approaches have been proposed to derive such impossible transitions efficiently [6,7,8]. Once an impossible differential has been chosen and placed, one uses it to restrict the possible values of some key bits involved in outer rounds. Indeed, if a candidate key partially encrypts/decrypts a given pair to the impossible differential, then this key is wrong. In this way, we discard as many wrong keys as possible and exhaustively search the rest of the keys. Organizing the attack is usually done with the *early abort technique* [9], introduced by Lu *et al.*

at CT-RSA 2008, originally to improve impossible differential attacks against Camellia and MISTY1. With this technique, one does not guess all the involved key material at once but step by step, discarding unwished pairs as soon as possible to reduce the time complexity of the whole procedure.

While the attack principle is rather clear, errors in the analysis are often discovered and many papers in the literature present flaws [9,10,11,12]. These flaws include errors in the computation of the time or the data complexity, in the analysis of the memory requirements or of the complexity of some intermediate steps of the attacks. Wishing to solve that, Boura *et al.* [1] presented at ASIACRYPT'14 a generic vision of impossible differential attacks with the aim of simplifying and helping the construction and verification of this type of cryptanalysis. In particular, they gave generic complexity analysis formulas for mounting such attacks and develop new ideas for optimizing them. These advances led to the improvement of previous attacks against well known ciphers such as CLEFIA-128 and Camellia, while also to new attacks against 23-round LBlock and all members of the Simon family.

Our Contribution. In this paper we carefully study the early abort technique from Lu *et al.* and the generic formula given by Boura *et al.*. In particular we build impossible differential attacks against a toy cipher for which the real time complexity is much higher than estimated by the formula. Then we describe an algorithm looking for optimal complexity of impossible differential attacks under the early abort technique. We finally apply it on an attack of Biryukov *et al.* [2] presented at FSE'15 against round-reduced TWINE-128 [13] and show that its complexity is higher than the natural bound of the exhaustive search.

Organization of the paper. In Section 2 we introduce the notations and give the formula of Boura *et al.*. In Section 3 we highlight the computational problem behind the early abort technique and provide simple examples for which the real complexity is far from the one given by the formula. Finally, in Section 4 we describe the algorithm we used to show that the complexity of the impossible differential attack against 25-round TWINE-128 from Biryukov *et al.* was underestimated and actually higher than 2^{128}.

2 Preliminaries

2.1 Impossible Differential Attacks

We first briefly remain how an impossible differential attack is constructed and introduce our notations (for sake of simplicity we use the exact same ones than in [1]).

Mounting an impossible differential attack starts by splitting the cipher E in three parts $E = E_3 \circ E_2 \circ E_1$ and by finding an impossible differential transition $(\Delta_X \nrightarrow \Delta_Y)$ through E_2. Then Δ_X (resp. Δ_Y) is propagated through E_1^{-1} (resp. E_3) with probability 1 to obtain Δ_{in} (resp. Δ_{out}). We denote by c_{in} and c_{out} the $\log_2$ of the probability of the transitions $\Delta_{in} \rightarrow \Delta_X$ and $\Delta_{out} \rightarrow \Delta_Y$ respectively. Finally we denote by k_{in} and k_{out} the key materials involved in

those transitions. All in all the attack consists in discarding the keys k for which at least one pair follows the characteristic through E_1 and E_3 and in exhausting the remaining ones.

2.2 A Generic Formula

At ASIACRYPT'14, Boura *et al.* proposed a generic vision of impossible differential attacks with the aim of simplifying and helping the construction and verification of this type of cryptanalysis. In particular, they provided a formula to compute the complexity of such an attack according to its parameters. According to notations introduced Section 2.1, their formula is:

- **data:** C_{N_α}
- **memory:** N_α
- **time:** $C_{N_\alpha} + \left(1 + 2^{|k_{in} \cup k_{out}| - c_{in} - c_{out}}\right) N_\alpha C_{E'} + 2^{|k|-\alpha}$

where N_α is such that $(1 - 2^{-c_{in}-c_{out}})^{N_\alpha} = 2^{-\alpha}$, C_{N_α} is the number of chosen plaintexts required to generate N_α pairs satisfying $(\Delta_{in}, \Delta_{out})$, $|k|$ is the key size and $C_{E'}$ is the ratio of the cost of partial encryption to the full encryption.

This formula was given without proof but authors claimed that "*it approximates really well the actual time complexity, as it can be seen in the applications, and in particular, in the tight correspondence shown between the LBlock estimation and the exact calculation from [14]*".

3 Counter-Examples

3.1 The Problem

Computing the time complexity of an impossible differential attack based on the early abort technique [9] is actually an optimization problem. Using notations introduced in Section 2.1, and introducing $k_1, k_2, \ldots, k_b$ as the key bits of the key material $k_{in} \cup k_{out}$ involved in the attack, the best complexity reached with the early abort technique is the minimal complexity of the following procedure over all the permutations of $\{1, 2, \ldots, b\}$:

0. Discard pairs which cannot follow the impossible differential.
1. Guess $k_{\sigma(1)}$
   (a) partially encrypt/decrypt pairs
   (b) discard pairs which cannot follow the impossible differential.
2. Guess $k_{\sigma(2)}$
   (a) partially encrypt/decrypt pairs
   (b) discard pairs which cannot follow the impossible differential.

   ⋮

b. Guess $k_{\sigma(b)}$
   (a) partially encrypt/decrypt pairs
   (b) discard pairs which cannot follow the impossible differential.

(c) if all pairs have been discarded then perform an exhaustive search over remaining key bits.

Let r_i^σ be the $\log_2$ of the number of pairs discarded after step i. Without taking into account the exhaustive search part, the complexity of the procedure is

$$\sum_{1 \leq i \leq b} 2^{|k_{\sigma(1)} \cup \ldots \cup k_{\sigma(i)}| - \sum_{0 \leq j < i} r_j^\sigma} \cdot N_\alpha C_{E'}.$$

As we see, computing a generic formula for such a problem is far from being trivial.

3.2 A Simple Counter-Example

To highlight the main issue of the generic formula given in [1], let consider a toy block cipher E defined as follows:

$$E = E' \circ MC \circ SR \circ SB \circ AK,$$

where E' is a 128-bit block cipher and where AK, SB, SR and MC respectively are the AddRoundKey, SubBytes, ShriftRows and MixColumns operations from the AES [15]:

- **AddRoundKey** (AK) adds a 128-bit subkey to the state.
- **SubBytes** (SB) applies the same 8-bit to 8-bit invertible Sbox S 16 times in parallel on each byte of the state,
- **ShriftRows** (SR) shifts the i-th row left by i positions,
- **MixColumns** (MC) replaces each of the four column C of the state by $M \times C$ where M is a constant 4×4 maximum distance separable matrix over $GF(2^8)$.

We remind that in the AES, the 128-bit internal state is seen as a 4×4 matrix of bytes where each byte is seen as an element of the finite field $GF(2^8)$.

Now, let us assume the existence of an impossible transition $\Delta_X \not\longrightarrow \Delta_Y$ over E' where Δ_X has only one active byte as depicted on Figure 1. We use this impossible transition to mount an impossible differential against our toy cipher E. We will show that, depending on the key schedule we choose, we are able to make the real complexity of the attack non-marginally higher than the estimated complexity obtained from the generic formula of Boura *et al.*.

Independent key bytes. As a well-known fact, the probability of the transition $\Delta_{in} \longrightarrow \Delta_X$ is 2^{-24} and exactly four key bytes are involved in the attack: k_0, k_5, k_{10} and k_{15}. For now let us assume those key bytes are independent. As a consequence, and according to the generic formula, the complexity of the impossible attack (without taking into account the pairs generation process and the exhaustive search part) is:

$$(1 + 2^{|k_{in}| - c_{in}}) \cdot N \cdot C'_E = (1 + 2^{32-24}) \cdot N \cdot C'_E = 257 \cdot N \cdot C'_E,$$

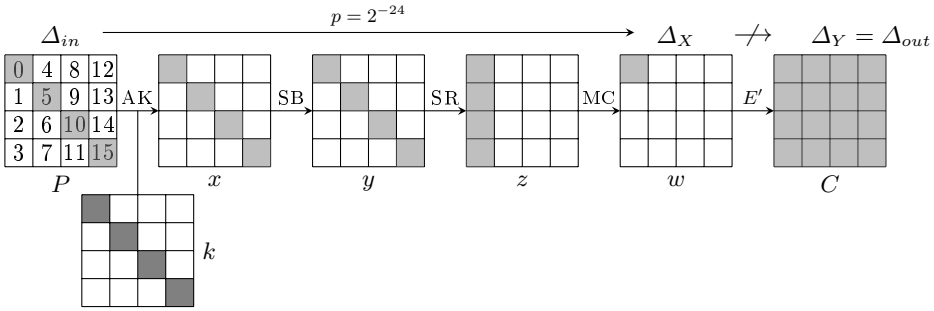

Fig. 1: Impossible differential attack against the toy cipher E.

where N is the number of pairs available and C'_E is the ratio of the cost of partial encryption to the full encryption. A common practice is to take for C'_E the ratio between the active Sboxes during a partial encryption and the total number of Sboxes (say S_E). Hence, the approximated complexity is $4 \cdot 257 \cdot N \cdot S_E^{-1}$.

Let us now compute the real complexity of the attack. Here the order in which key bytes are guessed does not impact the resulting complexity so the *best* procedure is as follows:

1. Guess k_0
   (a) partially encrypt/decrypt pairs
   (b) discard pairs which cannot follow the impossible differential.
2. Guess k_5
   (a) partially encrypt/decrypt pairs
   (b) discard pairs which cannot follow the impossible differential.
3. Guess k_{10}
   (a) partially encrypt/decrypt pairs
   (b) discard pairs which cannot follow the impossible differential.
4. Guess k_{15}
   (a) partially encrypt/decrypt pairs
   (b) discard pairs which cannot follow the impossible differential.

After performing step 1a), for each pair the differences in the three state variables y_5, y_{10} and y_{15} are known. Indeed, as the MixColumn matrix is MDS, they are fully determined by the value of Δy_0. As a consequence Δx_5, Δx_{10} and Δx_{15} can assume only 2^7 values each and thus only $2^{-3} \cdot N$ pairs remains after step 1b). Then, each of steps 2b), 3b) and 4b) decreases the number of pairs by a factor 2^7. As a result, the complexity of this procedure is:

$$(2^8 + 2^{8+8-3} + 2^{8+8+8-3-7} + 2^{8+8+8+8-3-7-7}) \cdot N \cdot S_E^{-1} = 57600 \cdot N \cdot S_E^{-1}.$$

All in all the real complexity is higher than the estimated one by a factor $57600/1028 \approx 2^{5.8}$. This factor is non-negligible, especially when compared to involved complexities.

Related key bytes. Let now study cases where k_0, k_5, k_{10} and k_{15} are related by one linear equation, so they can assume only 2^{24} values instead of 2^{32}. In that case the generic formula estimates the complexity to $(1+2^{24-24}) \cdot N \cdot S_E^{-1} = 2 \cdot N \cdot S_E^{-1}$, independently of the linear relation.

We first consider the case where the equation is $k_0 = k_5$. Thanks to the symmetry in the problem we only have six orders to try: $[k_0, k_5, k_{10}, k_{15}]$, $[k_0, k_{10}, k_5, k_{15}]$, $[k_0, k_{10}, k_{15}, k_5]$, $[k_{10}, k_0, k_5, k_{15}]$, $[k_{10}, k_0, k_{15}, k_5]$ and $[k_{10}, k_{15}, k_0, k_5]$. The corresponding complexities are respectively:

- $(2^8 + 2^{8-3} + 2^{8+8-3-7} + 2^{8+8+8-3-7-7}) \cdot N \cdot S_E^{-1} \approx 2^{8.9} \cdot N \cdot S_E^{-1}$
- $(2^8 + 2^{8+8-3} + 2^{8+8-3-7} + 2^{8+8+8-3-7-7}) \cdot N \cdot S_E^{-1} \approx 2^{13.1} \cdot N \cdot S_E^{-1}$
- $(2^8 + 2^{8+8-3} + 2^{8+8+8-3-7} + 2^{8+8+8-3-7-7}) \cdot N \cdot S_E^{-1} \approx 2^{14.6} \cdot N \cdot S_E^{-1}$
- $(2^8 + 2^{8+8-3} + 2^{8+8-3-7} + 2^{8+8+8-3-7-7}) \cdot N \cdot S_E^{-1} \approx 2^{13.1} \cdot N \cdot S_E^{-1}$
- $(2^8 + 2^{8+8-3} + 2^{8+8+8-3-7} + 2^{8+8+8-3-7-7}) \cdot N \cdot S_E^{-1} \approx 2^{14.6} \cdot N \cdot S_E^{-1}$
- $(2^8 + 2^{8+8-3} + 2^{8+8+8-3-7} + 2^{8+8+8-3-7-7}) \cdot N \cdot S_E^{-1} \approx 2^{14.6} \cdot N \cdot S_E^{-1}$

As we can see the first order is much better than the other ones, as it leads to a much smaller complexity. Thus the real complexity of the attack is $2^{8.9} \cdot N \cdot S_E^{-1}$, higher than the estimated one by a factor $2^{7.9}$. We note that the deviation from the expected complexity is bigger than in the *independent subkey bytes* case.

We now consider the case where the equation is $k_0 \oplus k_5 \oplus k_{10} \oplus k_{15} = 0$, or more generally, the case where the knowledge of three key bytes leads to the knowledge of the fourth one but where there is no relation involving only three key bytes. The real complexity of the attack becomes:

$$(2^8 + 2^{8+8-3} + 2^{8+8+8-3-7} + 2^{8+8+8-3-7-7}) \cdot N \cdot S_E^{-1} \approx 2^{14.6} \cdot N \cdot S_E^{-1},$$

which is higher than for the equation $k_0 = k_5$ by a factor $2^{5.7}$, increasing again the deviation from the expected complexity.

A trick. One may note that after performing step 1b), we could directly retrieve for each pair the $2 \times 2 \times 2 = 8$ values of (k_5, k_{10}, k_{15}) for which it follows the impossible differential. This would be done at the low cost of 3 memory accesses to a precomputed table. But only the values of (k_5, k_{10}, k_{15}) for which no pair follows the impossible differential matter. Thus we would have to make the list of the 2^{24} possible values of (k_5, k_{10}, k_{15}) before to discard reached values. As a consequence, the resulting complexity of this procedure is:

$$(2^8 \cdot N + 2^8 \cdot 2^{24} + 8 \cdot 2^{8-3} \cdot 2^{|k_0 \cup k_5 \cup k_{10} \cup k_{15}| - 32} \cdot N) \cdot S_E^{-1}.$$

As the number of pairs N should be at least close to 2^{24}, this procedure is better than the basic early abort technique. If there is no equation between the four key bytes then the complexity is very close to the one given by Boura *et al*'s formula. On the other hand, if there is at least one equation then the complexity is higher than expected due to the two first terms of the above formula.

3.3 Remarks

Those results highlight some issues with the generic formula of Boura *et al.*. Firstly, there exist impossible differential attacks for which the estimated time complexity is too optimistic and thus attacks with estimated time complexity close to the natural bound may actually not be faster than exhaustive search. Secondly, the formula only takes into account the number of equations between involved key bits while we showed that different equations may lead to different time complexities. In particular, the correct sequence of guesses has to take into account the *fastest* filtering first. It seems Boura *et al* make the assumption that the order of key guesses/filtering does not matter as all key bits are equally filtering. But this is far from being correct, especially in the context of ARX constructions.

4 Application to TWINE

At FSE'15, Biryukov *et al.* [2] used Boura *et al.* formula to compute the complexity of their impossible differential attack against 25-round TWINE-128 [13]. The attack involves 52 key nibbles which can assume only 2^{124} values instead of 2^{208} thanks to the key schedule and the resulting time complexity is $2^{124.5}$ encryptions, very close to the natural bound of the exhaustive search. As a consequence, and according to remarks of the previous section, it seems probable for the actual time complexity of this attack to be higher than 2^{128}, making it a non-valid attack.

4.1 Description of TWINE

This block cipher uses 16 branches of 4-bits and has a very simple round function: the Feistel function consists in a xor of a sub-key and a call to a unique Sbox based on the inverse function in $GF(2^4)$. Then, the branches are shuffled using a sophisticated nibble permutation ensuring faster diffusion than a simple shift [16]. One version of TWINE uses an 80 bits key, another uses a 128 bits key and we denote these versions as TWINE-80 and TWINE-128. They only differ by their key-schedule and both have 36 rounds. Both key schedules are sparse GFN's using only 2 Sbox calls per round for TWINE-80 and 3 for TWINE-128. At each round, some fixed nibbles of the key-state are used as round keys for the block cipher. One round of TWINE is depicted on Figure 2.

Keyschedule. The keyschedule produces the 36 round keys from the master key K. It is a variant of GFN with few application of the Sbox used in the round function of TWINE. Two key lengths are available: 80 and 128 bits. In both cases, the subkey WK_0 is first initialized to K and then next subkeys are generated using round constants and the same round function: $WK_{i+1} = F(WK_i, CON^i)$, for $0 \leq i \leq 35$. Finally the round key RK_i is obtained by extracting 8 nibbles from WK_i. The function F used for 128-bit keys is depicted on Figure 3. We refer the reader to [13] for the 80-bit version of the keyschedule.

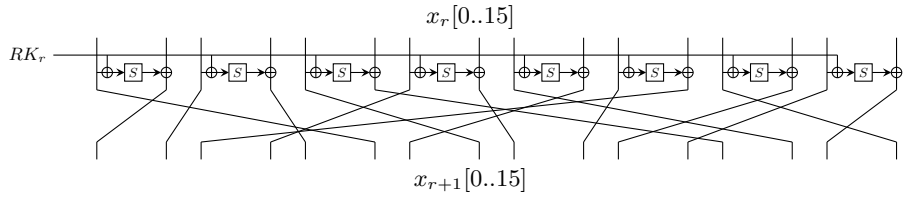

Fig. 2: The round function of TWINE.

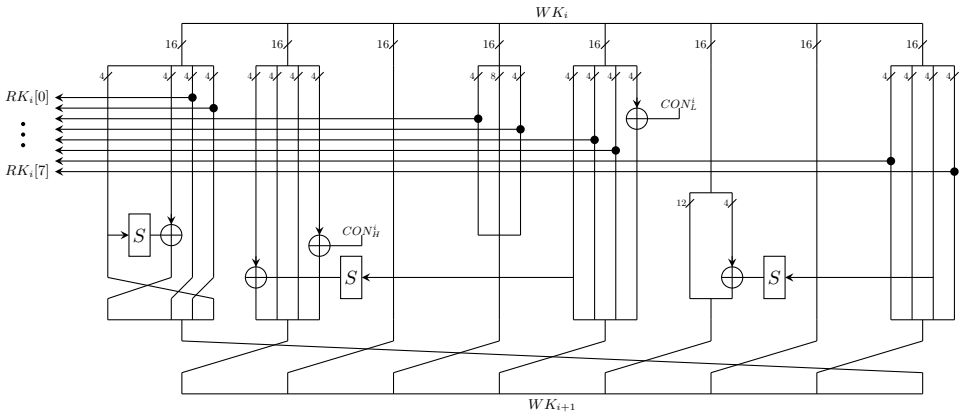

Fig. 3: Keyschedule of TWINE-128.

4.2 Biryukov *et al.* impossible differential attack

Biryukov *et al.* found a truncated impossible characteristic through 13 rounds of TWINE that they extended by 4 rounds at the start and by 8 rounds at the end in order to attack 25 rounds of the cipher. Their attack is depicted on Figure 4.

The difference in the plaintexts has to be zero in 11 nibbles such that $c_{in} + c_{out} = 16+60 = 76$. The key material $k_{in} \cup k_{out}$ is composed of $7+45 = 52$ round-key nibbles which can assume only 2^{124} thanks to the keyschedule of TWINE-128 as they all can be computed from the whole subkey WK_{24} except nibble 1.

As a consequence, and according to formula of Boura *et al.*, the complexity of their attack is $D = \alpha \cdot 2^{75.5-39} \cdot 2^{20} = \alpha \cdot 2^{56.5}$, $M = \alpha \cdot 2^{75.5}$ and $T \approx \alpha \cdot 2^{123.5} \cdot C_{E'} + 2^{128-\alpha}$, complexity parametrized by α. As they estimate the ratio $C_{E'}$ to $52/200 \approx 2^{-1.9}$, the value of α minimizing the overall complexity is 5.87.

4.3 Real Complexity of the Attack

Computing the real complexity of Biryukov *et al.* attack seems impossible due to the huge number of involved key nibbles. Indeed, there are 52 key nibbles leading

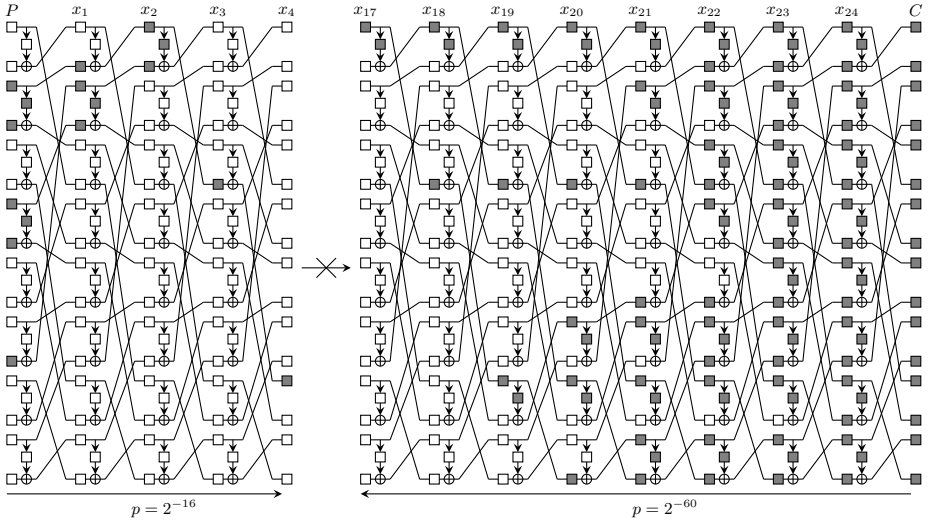

Fig. 4: Impossible differential attack on 25 rounds. No difference in white nibbles.

to $52! \approx 2^{225}$ orders for the early abort technique. Thus a naive approach would fail and a clever one has to be used.

Pruning strategy. We note that for the early abort technique, if between two guesses no pairs are discarded then the order in which they are guessed does not matter. Thus key nibbles can be grouped so that at each step pairs are discarded. So now the question becomes when do pairs are discarded? As saw with our simple example this is related to knowing differences before and/or after an Sbox. Since TWINE is a Feistel network things are a bit different and only one case has to be considered. Equations involved to describe round funtion of TWINE all have the following shape:

$$y \oplus z = S(x \oplus k),$$

where x, y and z are state variables while k is a round-key variable. We are interested in the case where both Δx and Δy are known (obtained by partially encrypting plaintexts (resp. decrypting ciphertexts)) and such that $\Delta z = 0$. In that case half of the pairs are discarded since the transition $\Delta x \longrightarrow \Delta y$ is possible with probability 2^{-1}. Then if the actual value of x is obtained by partially encrypting/decrypting plaintexts or ciphertexts then guessing k will allow to reduce the number of pairs by a factor 2^3. So we only have to consider groups of round key nibbles required to compute Δx and Δy, and the ones required to compute $x \oplus k$. Finally, as we are only looking for the fastest attack we can adopt a branch-and-bound strategy to accelerate the search.

Practice. For the considered attack there are 19 tuples (x, y, z) as expected. Determining the corresponding groups of round key nibbles is an easy task.

However, computing the number of values those groups (and their unions) can assume is more complicated while essential to the computation of the complexity. To solve this we used the same approach Derbez *et al.* [17] used to exhaust a particular kind of meet-in-the-middle attacks against the AES in a paper presented at FSE'13. Indeed, they provided a tool which takes as input a system of equations E in variable X and a subset $Y \subseteq X$ and gives as output a list of optimal algorithms enumerating all the possible values of Y under constraint of E with predictable time and memory complexities. The system of equations has to be composed of equations with the following shape:

$$\sum \alpha_i x_i \oplus \sum \beta_j S(x_j) \oplus \gamma = 0,$$

where a_i's, β_j's and γ are constant from a finite field $GF(2^q)$ and S is an q-bit Sbox. As the key schedule of TWINE is naturally described by such equations we were able to use this tool. Note that the output of their tool is a list because the number of possible values of Y enumerated by considered algorithms is not necessary constant and if an algorithm is slower than an other but finds less possible values for Y than it then they had to study both of them. But in our case we only care about the fastest algorithm, even if it enumerates more solutions.

Our algorithm was able to find the optimal permutation (see Appendix A) for the early abort technique in about 1h on a personal computer. As a result we found that for all permutation σ:

$$\sum_{1 \leq i \leq 38} 2^{|k_{\sigma(1)} \cup \ldots \cup k_{\sigma(i)}| - \sum_{0 \leq j < i} r_j^\sigma} \cdot N_\alpha C_{E'} > 2^{54} \cdot N_\alpha C_{E'}.$$

As $N_\alpha = \alpha \cdot 2^{75.5}$, the time complexity of the whole attack is higher than:

$$C_{N_\alpha} + \alpha \cdot 2^{127.6} + 2^{128-\alpha},$$

where $2^{128-\alpha}$ corresponds to time complexity of performing an exhaustive search on the remaining keys. Hence, if only based on the early abort technique, the attack is actually slower than an exhaustive search for all value $\alpha > 0$.

5 Conclusion

In this paper we have shown that the generic complexity analysis formula presented by Boura *et al.* at ASIACRYPT'14 does not always give a right estimation of the time complexity of impossible differential attacks. As proof we constructed simple counter-examples for which the real complexity is much higher than expected, one reaching a deviation of $2^{13.6}$ from the formula. As a consequence the formula is to use with caution, in particular when time complexity is close to the natural bound of the exhaustive search.

While we searched for, we were unable to find an impossible differential attack for which the real time complexity would be lower than the estimated

one. Finding such an attack or proving that the formula provides a lower bound on the complexity would be an interesting future work.

Finally we also showed that, if using only the early abort technique, the time complexity of the impossible differential attack against 25-round TWINE-128, presented at FSE'15 by Biryulov *et al.*, is higher than expected, and in particular, higher than 2^{128}.

References

1. Boura, C., Naya-Plasencia, M., Suder, V.: Scrutinizing and improving impossible differential attacks: Applications to clefia, camellia, lblock and simon. In: Advances in Cryptology - ASIACRYPT 2014 - 20th International Conference on the Theory and Application of Cryptology and Information Security, Kaoshiung, Taiwan, R.O.C., December 7-11, 2014. Proceedings, Part I. (2014) 179–199
2. Biryukov, A., Derbez, P., Perrin, L.: Differential analysis and meet-in-the-middle attack against round-reduced TWINE. In: Fast Software Encryption - 22nd International Workshop, FSE 2015, Istanbul, Turkey, March 8-11, 2015, Revised Selected Papers. (2015) 3–27
3. Knudsen, L.R.: Deal – a 128-bit block cipher. Technical Report Department of Informatics (1998)
4. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In: Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2-6, 1999, Proceeding. (1999) 12–23
5. Biham, E., Shamir, A.: Differential cryptanalysis of des-like cryptosystems. In: CRYPTO'91. (1991)
6. Kim, J., Hong, S., Sung, J., Lee, C., Lee, S.: Impossible differential cryptanalysis for block cipher structures. In: Progress in Cryptology - INDOCRYPT 2003, 4th International Conference on Cryptology in India, New Delhi, India, December 8-10, 2003, Proceedings. (2003) 82–96
7. Luo, Y., Lai, X., Wu, Z., Gong, G.: A unified method for finding impossible differentials of block cipher structures. Inf. Sci. **263** (2014) 211–220
8. Wu, S., Wang, M.: Automatic search of truncated impossible differentials for word-oriented block ciphers. In: Progress in Cryptology - INDOCRYPT 2012, 13th International Conference on Cryptology in India, Kolkata, India, December 9-12, 2012. Proceedings. (2012) 283–302
9. Lu, J., Kim, J., Keller, N., Dunkelman, O.: Improving the efficiency of impossible differential cryptanalysis of reduced camellia and MISTY1. In: Topics in Cryptology - CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008, San Francisco, CA, USA, April 8-11, 2008. Proceedings. (2008) 370–386
10. Minier, M., Naya-Plasencia, M.: A related key impossible differential attack against 22 rounds of the lightweight block cipher lblock. Inf. Process. Lett. **112**(16) (2012) 624–629
11. Wu, W., Zhang, L., Zhang, W.: Improved impossible differential cryptanalysis of reduced-round camellia. In: Selected Areas in Cryptography, 15th International Workshop, SAC 2008, Sackville, New Brunswick, Canada, August 14-15, Revised Selected Papers. (2008) 442–456
12. Zhang, W., Han, J.: Impossible differential analysis of reduced round CLEFIA. In: Information Security and Cryptology, 4th International Conference, Inscrypt 2008, Beijing, China, December 14-17, 2008, Revised Selected Papers. (2008) 181–191

13. Suzaki, T., Minematsu, K., Morioka, S., Kobayashi, E.: TWINE: A lightweight block cipher for multiple platforms. In: Selected Areas in Cryptography, 19th International Conference, SAC 2012, Windsor, ON, Canada, August 15-16, 2012, Revised Selected Papers. (2012) 339–354
14. Boura, C., Minier, M., Naya-Plasencia, M., Suder, V.: Improved impossible differential attacks against round-reduced lblock. IACR Cryptology ePrint Archive **2014** (2014) 279
15. NIST: Advanced Encryption Standard (AES), FIPS 197. Technical report, NIST (November 2001)
16. Suzaki, T., Minematsu, K.: Improving the generalized feistel. In: Fast Software Encryption, 17th International Workshop, FSE 2010, Seoul, Korea, February 7-10, 2010, Revised Selected Papers. (2010) 19–39
17. Derbez, P., Fouque, P.: Exhausting demirci-selçuk meet-in-the-middle attacks against reduced-round AES. In: Fast Software Encryption - 20th International Workshop, FSE 2013, Singapore, March 11-13, 2013. Revised Selected Papers. (2013) 541–560

A Optimal sequence

We found the following permutation to be optimal for the early abort technique applied to the 25-round impossible differential attack:

1. $\Delta x_0[2]$, $\Delta x_0[3]$
2. $\Delta x_0[6]$, $\Delta x_0[7]$
3. $\Delta x_1[2]$, $\Delta x_1[3]$
4. $\Delta x_2[0]$, $\Delta x_2[1]$
5. $y_0[2]$
6. $\Delta x_{23}[12]$, $\Delta x_{24}[10]$
7. $y_{23}[12]$
8. $\Delta x_{22}[12]$, $\Delta x_{23}[10]$
9. $y_{22}[12]$
10. $\Delta x_{22}[6]$, $\Delta x_{23}[8]$
11. $y_{22}[6]$
12. $\Delta x_{22}[2]$, $\Delta x_{23}[4]$
13. $y_{22}[2]$
14. $y_0[6]$
15. $y_1[2]$
16. $\Delta x_{21}[10]$, $\Delta x_{22}[2]$
17. $y_{21}[10]$
18. $\Delta x_{20}[10]$, $\Delta x_{21}[2]$
19. $y_{20}[10]$
20. $\Delta x_{21}[2]$, $\Delta x_{22}[4]$
21. $y_{21}[2]$
22. $\Delta x_{21}[0]$, $\Delta x_{22}[0]$
23. $\Delta x_{20}[0]$, $\Delta x_{21}[0]$
24. $y_{20}[0]$
25. $\Delta x_{19}[0]$, $\Delta x_{20}[0]$
26. $\Delta x_{21}[12]$, $\Delta x_{22}[10]$
27. $y_{19}[0]$
28. $y_2[0]$
29. $\Delta x_{19}[12]$, $\Delta x_{20}[10]$
30. $y_{19}[12]$
31. $y_{21}[12]$
32. $y_{21}[0]$
33. $\Delta x_{18}[0]$, $\Delta x_{19}[0]$
34. $\Delta x_{20}[12]$, $\Delta x_{21}[10]$
35. $\Delta x_{17}[0]$, $\Delta x_{18}[0]$
36. $y_{20}[12]$
37. $y_{18}[0]$
38. $y_{17}[0]$

Each item v has to be understood as *guess the key material required to compute v from the plaintexts/ciphertexts* and $y_r[2i] = x_r[2i] \oplus k_r[i]$.