The Problem of Half Round Key XOR

Anubhab Baksi

Nanyang Technological University, Singapore

anubhab001@e.ntu.edu.sg

Abstract. In the design of GIFT, half round key XOR is used. This leads to the undesired consequence that the security against the differential/linear attacks are overestimated. This comes from the observation that; in the usual DDT/LAT based analysis of the differential/linear attacks, the inherent assumption is the full round key is XORed at each round.

Keywords: block cipher · gift · differential attack · linear attack

1 Introduction

As race of the lightweight ciphers is underway for about a decade or so, we have seen many tricks to reduce the hardware/software footprint. This brings us neatly to `GIFT` [BPP+17], which has gained a lot of popularity.

One of the interesting traits of `GIFT` is that the full state is not XORed with the round keys, rather only half the round keys are XORed in all rounds except the first (which does not have round key XOR). This creates the problem that it underestimates the power of the differential/linear attacks (among others). Though not explicitly stated, the usual DDT/LAT based analysis inherently assumes the full round key XOR. Moreover, other attacks which work on similar principle (integral, impossible differential, higher order differential) are also underestimated.

In a loose sense, the observation can be stated as follows. Due to the round key XOR happens on 2 (out of 4 bits) per SBox. This leaves the rest two bits unchanged, thus any difference in these 2 bits propagates with probability 1. So the SBox works as if it has 4 linear structure. This is comparable to the 4-LS SBox in `DEFAULT-LAYER` [Bak21, Chapter 8] (note that `DEFAULT` uses full round key XOR, hence this problem does not arise). Curiously, the analysis done by [NDE21] is very close to uncovering the danger of half round key XOR.

The analysis in [SHW+14, ZDY18, Bak20, SWW21, SPWW22] are all correct with the assumption that the rounds are independent, which is not true due to half round key XOR[1]. In order for the rounds to be independent, the full round key XOR at each round is essential. As half of the bits at each round of the state is not XORed with any round key, these bits are passed to the next round in a deterministic way; this violates the assumption that each round is independent.

In the usual DDT/LAT based analysis, each round is treated as independent. The independence assumption, though not accurate in absolute terms, simplifies the problem a lot (also, this assumption mostly holds up in an empirical evaluation). In the analysis, only the unkeyed permutation is taken into account; this may create the feeling that the round key XOR is not important to ascertain the security against the differential/linear attacks. However, one has to note that the the round keys are crucial in making the (input,

[1] The problem we state here is not related to the independence of the round keys.

output) distribution of the SBoxes non-deterministic, and are unavoidable while figuring out the security.

Contribution

Despite receiving a large number of third-party analysis, we argue that the security claims made for `GIFT` [BPP+17] (and also in `SKINNY` [BJK+16]) are overestimated. In order to match the claimed security, the round keys at each round needs to be XORed with the full state, which is not true as per `GIFT` specification. Even if we assume all the round keys are independent, still the full state at each round is to be XORed for the inherent DDT/LAT based analysis to hold.

2 Background Information

In order to see why the round keys are crucial in determining the security of a cipher against the differential attack, we show a simple scenario (a similar argument will hold for the linear attack). Assume there is a 4-bit SBox, and a 4-bit key k_0 is XORed at the SBox input and another 4-bit key k_1 is XORed at the SBox output (i.e., first k_0 is XORed to the plaintext, then it is fed to the SBox, then k_1 is XORed to the SBox output). The attacker, Eve, does not know k_0 and k_1, though she knows the specification of the SBox. At this point, she wants to recover the key with the help of differential attack.

She first chooses an arbitrary 4-bit plaintext p_0 and observes the output c_0. Then she chooses a distinct plaintext p_1 and obtains the corresponding ciphertext c_1. Since k_0 and k_1 are unknown, she does not know which is the exact input to the SBox and which is the exact output from the SBox. However, she can compute the input difference $\delta = p_0 \oplus p_1$ and the output difference $\Delta = c_0 \oplus c_1$. Since k_1 does not impact δ or Δ, now she can use this (δ, Δ) information to retrieve k_0.

For this purpose, she first assumes k_0 can be any of the 2^4 possible values. For each of these 2^4 cases, she computes the input to the SBox, which further enables use to compute the output from the SBox. Ultimately, some of the choices of k_0 does not conform to the (δ, Δ) information, which can be discarded.

Now, if the XOR with k_0 is missing (or equivalently, k_0 is known), she can readily find out the exact input to the SBox, from which she can find out the exact output from the SBox. Notice that, when k_0 is not known, she does not know the exact input to the SBox (and the exact output to the SBox). In this case, she has to rely on the differential information.

Thus, the presence of the initial key XOR makes the huge difference – without it the full cipher is visible to the attacker, consequently there is no security[2].

If t ($t \leq 4$) bits of k_0 is XORed to the state, then the search space is 2^t (Eve knows all but t bits of the SBox input). In particular, when only 2 bits are XORed, Eve only has 4 options for the SBox. This contrasts the full XOR of k_0 (where there are 16 options) as now the search space is at the square root.

In the usual DDT based analysis, the inherent assumption is that all the SBoxes at each round is XORed with the round key. Therefore, the attacker does not know the exact input/output from any SBox. When computing the DDT of the SBox S, the number of solutions to $\Delta = S(x) \oplus S(x \oplus \delta)$ is computed and stored at the $(\delta, \Delta)^{\text{th}}$ cell, where x is considered over all possible values (i.e., 2^n values for an n-bit SBox). This works only with the assumption that Eve does not have any information about the value of x. If, on the other hand, she can reduce the search space for x; then this computation no longer holds. Now, when the full state is not updated at a certain SBox at a certain round through round key XOR, Eve knows some information about the input x of that SBox.

[2]The final key (i.e., k_1) is also important in this toy example.

Consequently, the search space for x becomes smaller than 2^n elements. Ultimately, this lowers down the security of the differential attack.

The assumption that the attacker knows only the (input difference, output difference) pair and nothing else comes from the full key XOR at the input of the SBox. Since the key (which is XORed at the input of the SBox) is not known to the attacker, the best she could do is to consider all possible values for the key, which leads to DDT. In other words, in the computation of DDT, the full key XOR at the SBox input is inherently assumed.

Role of Unkeyed Permutation

We would like to point out that, the typical papers (such as, [LIM20]) only consider the unkeyed permutation. This term is somewhat of a misnomer, as the assumption of full round key XOR at each round is inherently assumed. When it comes to analysis, unkeyed permutation does not actually mean the construction is publicly known (i.e., there is no round key XOR), it instead means:

1. The round keys are independent.
2. The full state is made non-deterministic at each round by XORing with the corresponding round key.

The security of the unkeyed permutation cipher is computed based on the assumptions. However, from what we can gather from the existing literature, the assumptions are never explicitly stated.

If truly there is no round key, then the full cipher is specified to the attacker, hence there is no question of security. In case of a truly unkeyed permutation, all the differential/linear distinguishers are trivial – always. Since there is no key, the attacker knows everything about the cipher and hence can trace out the propagation of the differential/linear trails with probability 1. Not only Eve knows about the input difference/mask for each SBox, but also the actual input(s) to each SBox; this means the full cipher is known. Lest one disagrees (i.e., the challenger thinks the truly unkeyed permutation offers some security), we invite to the following game:

1. The challenger will toss a fair coin, and based on that will choose either `CIPHER` or `RANDOM`.
2. We will give one plaintext, and the challenger return us the ciphertext (if `CIPHER`) or something random (if `RANDOM`).
3. We will tell whether the challenger has chosen `CIPHER` or `RANDOM`.

Impact of Large Round

We would like to draw attention to another potential pitfall that can be stated as follows: "*While it is true that over very small number of rounds this invalidates the assumption of independence and leads to probabilities (correlations) significantly higher than expected, this is hardly true when considering many rounds.*"

Even when considering many rounds of the cipher, the full round key XOR at each round is essential. Say, we have a half key XOR at the i^{th} round (i is sufficiently large); then the security from the i^{th} to the $(i+1)^{\text{th}}$ round will be lower than what it otherwise would be for a full round key XOR. Overall, if the full state is not XORed with the round key at any round, the security from that particular round to its next will be lower. Stated in other words, even the security bounds reach a substantially high level, we still need full XOR of the round key to the cipher state in order to make security claim for the next round onward. If full key XOR is missed at a round (no matter how large the round counter is), the security claim for that round is to be properly adjusted.

3 Observation

3.1 Basic

One notable feature in the construction of GIFT [BPP+17] is to XOR the round key to half of the state bits. This is a deviation from the usual design choice of XORing the full state with the round keys.

Despite what it may seem, XORing the full state with the round key is of paramount importance. The round keys are crucial in making the (input, output) distribution of the SBoxes non-deterministic. Note in order for the usual DDT/LAT-based analysis (including, but not limited to [ZDY18, Bak20, SWW21]) to work, this condition is to be satisfied. Otherwise, the usual DDT/LAT-based analysis would overestimate the actual bound.

For a simple thought experiment with respect to the classical differential attack, consider the trivial case of the unkeyed permutation of GIFT-128. Since there is no key, the entire cipher is known to Eve. This means, she can trace out any input difference propagating through the permutation with probability 1. Thus, the usual DDT-based modelling (which does not consider any round key XOR for simplicity) for the differential bound would be completely wrong. For the exact analysis, one has to treat as if the SBox (no matter the actual description) has 2^4 linear structures (so, the cipher has a trivial differential bound with probability 1).

Building on this, consider the next step where only the initial key XOR is performed[3] (i.e., before the first SBox layer). Let, we encrypt two plaintexts p_0 and $p_1 (= p_0 \oplus \delta)$ for a pre-determined $\delta \neq 0$. In this case, Eve knows the input difference to the SBox layer but does not know what the input pairs are. For this round, the usual DDT-based analysis will apply. For the next round onward, one has to keep in mind that the input differences to the SBoxes are all deterministic, thus the difference transition would not happen according to the DDT-based prediction – rather it would behave as if the SBox has 2^4 linear structures.

Going on this way, we can see that, if the round key XOR is missed in a round, then for that round the usual DDT-based analysis does not hold anymore. Although it is true that during the analysis we only consider the unkeyed permutation [ZDY18, Bak20, SWW21], it is inherently assumed that the full round key is XORed at each round (so that the usual DDT-based assumption holds at each round). The term of unkeyed permutation is not wrong per-se, as the round keys are not explicitly considered, rather the impact of the round key XOR is obtained by probabilistic transition in DDT/LAT.

The case for GIFT is more complicated than this, as the ciphers have 2-bit round key XOR at each SBox. Still, for the 2 bits (which do not pass through any key XOR), the difference transition is deterministic. This somewhat comparable to the situation where the SBox has is replaced by a weak one. During our study of literature, it has become apparent that this crucial fact has evaded the designers as well as the third party reviewers. It thus seems plausible that the bounds against the differential, linear and probably some other classical attacks are overestimated in the literature.

3.2 Detailed

The problem with half key XOR per SBox (as done in GIFT [BPP+17]) is not visible at the first glance. However, at a deeper look, it appears to be huge oversight. In summary, we claim that the half key XOR is a new/different paradigm and the usual classical analysis (that internally assume full key XOR) cannot be not directly used.

In order to address the potential problem that arises when only the half of the state bits are XORed, one has to look deeper than what has been considered in the literature

[3] As there is no round key XOR thereafter, it is trivially possible to retrieve the secret key by going backward; but this is beyond the point. We are only considering why it is essential to have full-state XOR with the round keys with respect to some classical attacks

so far. It is valid that when the round key is XORed to the full state (which makes the inputs to the SBoxes non-deterministic), the usual analysis based on DDT/LAT applies. This apparently does not hold when some bits of the SBoxes does not receive a round key XOR, as now those bits become deterministic in nature (with respect to the attacker, any transition relating to those bits happen in a deterministic way, i.e., with either probability 1 or 0). Loosely speaking, when 2 bits of a 4-bit SBox does not have the initial round key XOR, the SBox becomes much weak against classical attacks like differential or linear.

For the sake of simplicity and conciseness, here we only consider the traditional differential attack (it should be possible to extend this methodology with respect to other attacks in an analogous way) against `GIFT-128`. First, note that, the DDT only shows the frequency – but not the probability, which is to be calculated by dividing each entry with 2^4). In our case, we need the exact probability, so we define *Probabilistic Difference Distribution Table* (PDDT[4]) which is derived from the DDT and it shows the transition probability from each input difference (δ) to each output difference (Δ). For instance, the PDDT for the `GIFT` SBox (`1A4C6F392DB7508E`) is shown in Table 1. The entry at row δ and column Δ indicates the probability that; with the assumption of the attacker does know any information on δ (i.e., δ has full entropy), the δ (input difference) $\rightarrow \Delta$ (output difference) transition will take place. Thus, the PDDT has the property that the sum of each row (and also column) is 1. Note that (at the initial round) where Eve chooses the plaintext pair, she still does not know the input pair to the SBox thanks to the full key XOR.

Table 1: PDDT for `GIFT` SBox

δ \ Δ	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$
0010	0	0	0	0	0	$\frac{1}{4}$	$\frac{1}{4}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0
0011	0	0	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$
0100	0	0	0	$\frac{1}{8}$	0	$\frac{1}{4}$	0	$\frac{3}{8}$	0	$\frac{1}{8}$	0	0	0	$\frac{1}{8}$	0	0
0101	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$
0110	0	0	$\frac{1}{4}$	$\frac{3}{8}$	0	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	0	0	$\frac{1}{8}$	0
0111	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{8}$	0	0	0
1000	0	0	0	$\frac{1}{4}$	0	0	0	$\frac{1}{4}$	0	0	0	$\frac{1}{4}$	0	0	0	$\frac{1}{4}$
1001	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0
1010	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{4}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0
1011	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0
1100	0	0	$\frac{1}{4}$	0	$\frac{1}{4}$	0	0	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0
1101	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{4}$	0	0	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{8}$	0	$\frac{1}{8}$
1110	0	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$	0	0
1111	0	$\frac{1}{8}$	$\frac{1}{8}$	0	$\frac{1}{4}$	0	0	0	0	$\frac{1}{8}$	0	$\frac{1}{8}$	0	0	$\frac{1}{8}$	$\frac{1}{8}$

This PDDT is actually used in the usual DDT-based modelling such as [ZDY18, Bak20, SWW21, SPWW22]. It is not explicitly stated that the full round key XOR is assumed; rather it is assumed that the probability transition happens according to the PDDT – for this assumption to hold, the all the input bits to the SBoxes has to be non-deterministic – which is achieved only through XORing the full state with the round keys. In this case, the attacker only knows the input difference δ, but does not know the pair of inputs (which are δ apart). Thus, this is the maximum narrowed down search space she could have.

On the other extreme, when the δ is fully known (happens where there is no round key XOR), the form of PDDT changes. Note that the entries which are 0 do not change,

[4]This is not to be confused with the *Partial Difference Distribution Table* (pDDT) of [BV13].

but each non-zero value becomes 1. This is because, having the full knowledge of δ, the attacker can determine exactly which $\delta \to \Delta$ to follow (with probability 1) and discard the rest transitions. To describe this new PDDT in the same framework as the regular situation (i.e., when δ has full entropy), we introduce the concept of *mask*. In general, for an n-bit SBox, the mask is a 2^n-bit binary string; where $\text{mask}_i = 1$ if the i^{th} bit is not known to the attacker, 0 otherwise. Thus, for the 4-bit SBox of `GIFT`; when δ has full entropy, mask = 1111; but when δ is fully known, mask = 0000. The PDDT [mask= 0000] for the `GIFT` SBox is shown in Table 2.

Table 2: PDDT [mask = 0000] for `GIFT` SBox

δ \ Δ	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
0000	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	1	1	0	1	1	1	1	1	0	0	1
0010	0	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0
0011	0	0	0	0	0	1	1	0	1	0	0	1	1	1	1	1
0100	0	0	0	1	0	1	0	1	0	1	0	0	0	1	0	0
0101	0	0	1	0	0	1	0	0	1	0	0	0	1	1	1	1
0110	0	0	1	1	0	0	0	1	0	0	1	0	0	0	1	0
0111	0	0	1	0	0	1	0	0	1	1	1	1	1	0	0	0
1000	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
1001	0	1	0	1	0	0	1	1	1	0	1	0	1	1	0	0
1010	0	1	0	0	0	0	1	0	0	1	1	0	0	1	1	0
1011	0	1	0	1	0	0	1	1	1	1	0	0	1	0	1	0
1100	0	0	1	0	1	0	0	0	1	0	1	0	1	0	1	0
1101	0	1	1	0	1	0	0	0	0	0	1	1	0	1	0	1
1110	0	1	0	0	1	0	0	0	1	1	0	0	1	1	0	0
1111	0	1	1	0	1	0	0	0	0	1	0	1	0	0	1	1

This PDDT is useful to find the trivial case of unkeyed permutation (thus the attacker fully knows the cipher state at any round). In this case, the attacker knows the exact input pair, thus she can determine the exact output pair (with probability 1). For example, say, the input pair to a particular SBox at a particular round is (`0`, `2`), thus the input difference is `2`. As this is known to Eve, she can compute the outputs from the description of the SBox as (`0`, `4`), leading to the output difference of `e`. In this case, the difference transition $\texttt{2} \to \texttt{5}$ happens with probability 1. From the corresponding PDDT in Table 2, note that this transition indeed happens with probability 1.

The dissimilarity of the PDDT [mask = 1111] (Table 1) and the PDDT [mask = 0000] (Table 2) stems from Eve's knowledge (or, lack thereof) about the pair of inputs. In both situations, the input difference δ is known to her with probability 1; but in the former situation the input pair that generates δ is known but not with probability 1, whereas in the latter situation the input pair is known with certainty. Going back to the toy example just described, in the former situation Eve knows $\delta = \texttt{2}$ with probability 1; but it could come from any pair from $\{\{\texttt{0},\texttt{2}\},\{\texttt{1},\texttt{3}\},\{\texttt{4},\texttt{6}\},\{\texttt{5},\texttt{7}\},\{\texttt{8},\texttt{a}\},\{\texttt{9},\texttt{b}\},\{\texttt{c},\texttt{e}\},\{\texttt{d},\texttt{f}\}\}$, each with probability $\frac{1}{8}$. Thus, she obtains the following output pairs: $\{\{\texttt{1},\texttt{4}\},\{\texttt{a},\texttt{c}\},\{\texttt{6},\texttt{3}\},\{\texttt{f},\texttt{9}\},\{\texttt{2},\texttt{b}\},\{\texttt{d},\texttt{7}\},\{\texttt{5},\texttt{8}\},\{\texttt{0},\texttt{e}\}\}$, each with probability $\frac{1}{8}$. Ultimately, she is able to compute the output difference Δ as $(\texttt{5},\texttt{6},\texttt{9},\texttt{a},\texttt{d},\texttt{e})$, with respective probability as $(\frac{1}{4}, \frac{1}{4}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8})$. The PDDT [mask = 1111] reflects this scenario (match with $\delta = \texttt{2}$ in Table 1). Contrary to this, when the exact input pair (i.e., (`0`, `2`) by presumption) is known, this probability distribution is no longer respected, rather that from PDDT [mask = 0000] is respected. Here, Eve can pinpoint the output difference (which is `5`) with probability 1. Now, this PDDT can be used to get the differential bound (i.e., by replacing the DDT with it) for the unkeyed permutation.

In this way, the concept of DDT can be adjusted to take into account the attacker's knowledge about the input pair to the SBox (i.e., after round key XOR). The $\delta \to \Delta$ transition is to be interpreted as a probability conditioned by the fact that the attacker precisely knows some bits of the input pair to the SBox.

In the middle of the two scenarios lies the case of the `GIFT-128`, where the middle two bits are XORed with the round key bits. This corresponds to mask $= 0110$. Note that the half key XOR does not change the grouping: $\{0{**}0, 0{**}1, 1{**}0, 1{**}1\}$, where $*$ can be either 0 or 1; for any input to the SBox. The attacker always knows the MSB and the LSB of the input pairs which are fed to the SBox. This contrasts the usual setting where no bit of the input pair is known (only the input difference is known). It seems possible to continue further in this direction to ultimately model the entire cipher by keeping half key XOR in mind.

4 Conclusion

We show danger of deviating from the norm of cipher construction. When the usual DDT/LAT based analysis was first proposed, all the ciphers at that time used full XOR of the state with each round key. Then came `GIFT` [BPP+17] which deviated from the norm, but made the same security claim as it would be for full round key XOR.

This means that the usual analysis as in [ZDY18, Bak22, SWW21, SPWW22] are all valid[5] but those work with the inherent assumption that the full state is XORed with the round key (thereby making the input difference to each SBox fully probabilistic, i.e., using 1111 as the mask).

The simplest patch would be to opt for full round key addition instead. This would inevitably increase the device footprint.

References

[Bak20] Anubhab Baksi. New insights on differential and linear bounds using mixed integer linear programming (full version). Cryptology ePrint Archive, Report 2020/1414, 2020. https://eprint.iacr.org/2020/1414. 1, 4, 5

[Bak21] Anubhab Baksi. *Classical and Physical Security of Symmetric Key Cryptographic Algorithms*. PhD thesis, School of Computer Science & Engineering, Nanyang Technological University, Singapore, 2021. https://dr.ntu.edu.sg/handle/10356/152003. 1

[Bak22] Anubhab Baksi. *New Insights on Differential and Linear Bounds Using Mixed Integer Linear Programming*, pages 109–140. Springer Singapore, Singapore, 2022. 7

[BJK+16] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. The SKINNY family of block ciphers and its low-latency variant MANTIS. *IACR Cryptology ePrint Archive*, 2016:660, 2016. 2

[BPP+17] Subhadeep Banik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang Meng Sim, and Yosuke Todo. Gift: A small present. Cryptology ePrint Archive, Report 2017/622, 2017. https://eprint.iacr.org/2017/622. 1, 2, 4, 7

[5] The differential and linear bounds claimed in the `GIFT` paper [BPP+17] are wrong, so those do not count anyway.

[BV13] Alex Biryukov and Vesselin Velichkov. Automatic search for differential trails in arx ciphers (extended version). Cryptology ePrint Archive, Paper 2013/853, 2013. https://eprint.iacr.org/2013/853. 5

[LIM20] Fukang Liu, Takanori Isobe, and Willi Meier. Automatic verification of differential characteristics: Application to reduced gimli (full version). Cryptology ePrint Archive, Paper 2020/591, 2020. https://eprint.iacr.org/2020/591. 3

[NDE21] Marcel Nageler, Christoph Dobraunig, and Maria Eichlseder. Information-combining differential fault attacks on default. Cryptology ePrint Archive, Report 2021/1374, 2021. https://eprint.iacr.org/2021/1374. 1

[SHW+14] Siwei Sun, Lei Hu, Meiqin Wang, Peng Wang, Kexin Qiao, Xiaoshuang Ma, Danping Shi, Ling Song, and Kai Fu. Towards finding the best characteristics of some bit-oriented block ciphers and automatic enumeration of (related-key) differential and linear characteristics with predefined properties. *IACR Cryptol. ePrint Arch.*, 2014:747, 2014. 1

[SPWW22] Ling Sun, Bart Preneel, Wei Wang, and Meiqin Wang. A greater gift: Strengthening gift against statistical cryptanalysis. Cryptology ePrint Archive, Paper 2022/243, 2022. https://eprint.iacr.org/2022/243. 1, 5, 7

[SWW21] Ling Sun, Wei Wang, and Meiqin Wang. Accelerating the search of differential and linear characteristics with the sat method. Cryptology ePrint Archive, Report 2021/213, 2021. https://eprint.iacr.org/2021/213. 1, 4, 5, 7

[ZDY18] Baoyu Zhu, Xiaoyang Dong, and Hongbo Yu. MILP-based differential attack on round-reduced gift. Cryptology ePrint Archive, Report 2018/390, 2018. https://eprint.iacr.org/2018/390. 1, 4, 5, 7