From: Sumanta Sarkar <sumanta.sarkar@gmail.com>
Sent: Thursday, April 25, 2019 3:59 PM
To: lightweight-crypto; NILANJAN DATTA; ashrujit@cs.washington.edu; debdeep; sikhar.patranabis@iitkgp.ac.in; s.picek@tudelft.nl; RAJAT SADHUKHAN
Cc: lwc-forum@list.nist.gov
Subject: Re: TRIFLE S-box has some structural weakness

Hi TRIFLE Team,

I observe that there are some structural weakness in TRIFLE S-box.
This S-box has 4 fixed points:
0 -> 0
5 -> 5
A -> A
F -> F

What is more worrisome is that these fixed points are forming a subspace (U), that is S(U) = U. Having the design where diffusion depends on the bit permutation and there are only a few places where round constants are being added, this makes a perfect stage for mounting invariant subspace attack.

Please let me know if my understanding is not correct.

Thanks
Sumanta

From: Siang Meng Sim <crypto.s.m.sim@gmail.com>
Sent: Wednesday, June 26, 2019 12:43 PM
To: lightweight-crypto
Cc: lwc-forum@list.nist.gov
Subject: OFFICIAL COMMENT: TRIFLE
Attachments: TRIFLE-BC_weakness.pdf

Dear TRIFLE team and all,

We would like to quickly highlight some weaknesses of TRIFLE-BC, the underlying block cipher of TRIFLE, namely:
1) 2 rounds partial decryption without key
2) existence of arbitrary long single active bit differential/linear trails
3) existence of iterative subspace transitions through TRIFLE-BC S-box
The draft is in the attachment, we hope that the TRIFLE team could verify them.

We believe some of these weaknesses can be exploited to mount key-recovery attack on TRIFLE-BC, we are currently working on it.

Previously, the TRIFLE team has responded to Sumanta's concern about invariant subspace attack on TRIFLE-BC, as I quote
"Thank you for your observation. Indeed, we were aware of this property of the S-Box. We believe that adding constant to the most significant bit of some nibbles at each round breaks the propagation of invariant subspace, and full round TRIFLE should resist against such attacks."
This belief is not entirely true, as we have pointed it out in Section 5 of our draft that there exists one subspace transition that is not broken by adding constant to the most significant bit of some nibbles.

--
Best Regards,
Siang Meng Sim
on behalf of Thomas Peyrin, Sumanta Sarkar, Yu Sasaki

A Study on TRIFLE-BC

Thomas Peyrin, Sumanta Sarkar, Yu Sasaki, Siang Meng Sim

1 Introduction

TRIFLE is one of the round 1 candidates in the ongoing NIST Lightweight Cryptography competition, it is a AEAD scheme which uses an SPN based block cipher TRIFLE-BC as its underlying encryption algorithm.

Although the design of TRIFLE-BC is heavily inspired by GIFT and PRESENT, the combination of its building blocks (operations in its round function) result in several potential weaknesses. In this study, we highlight the undesired cryptographic properties and the potential exploitation of these properties to launch attacks on TRIFLE.

2 Notations

Let SN, BP, AK denotes SubNibbles (using S-box S), BitPermutation (using bit permutation P), AddRoundKey plus AddRoundConst respectively, and $R = AK \circ BP \circ SN$. The round key for round r is denoted as RK^r. Let $X_i^r[j]$ denotes the bit j of nibble i in round r, where $r \in \{0, \ldots, 49\}$, $i \in \{0, \ldots, 31\}$ and $j \in \{0, \ldots, 3\}$.

$$PT = X^0 \xrightarrow[S^0]{SN} Y^0 \xrightarrow[P]{BP} Z^0 \xrightarrow[\oplus RK^0]{AK} X^1 \xrightarrow{R} \ldots \xrightarrow{R} X^{49} \xrightarrow[P,S^{49}]{BP \circ SN} Z^{49} \xrightarrow[\oplus RK^{49}]{AK} X^{50} = CT$$

3 Key-independent Decryption

The grouping of TRIFLE-BC S-boxes is as follows:

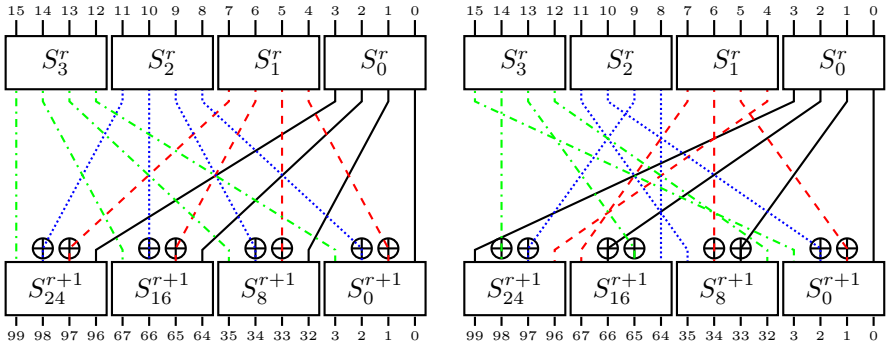
$$\begin{aligned}
\{S_0^r, S_1^r, S_2^r, S_3^r\} &\rightarrow \{S_0^{r+1}, S_8^{r+1}, S_{16}^{r+1}, S_{24}^{r+1}\} \\
\{S_4^r, S_5^r, S_6^r, S_7^r\} &\rightarrow \{S_1^{r+1}, S_9^{r+1}, S_{17}^{r+1}, S_{25}^{r+1}\} \\
\{S_8^r, S_9^r, S_{10}^r, S_{11}^r\} &\rightarrow \{S_2^{r+1}, S_{10}^{r+1}, S_{18}^{r+1}, S_{26}^{r+1}\} \\
\{S_{12}^r, S_{13}^r, S_{14}^r, S_{15}^r\} &\rightarrow \{S_3^{r+1}, S_{11}^{r+1}, S_{19}^{r+1}, S_{27}^{r+1}\} \\
\{S_{16}^r, S_{17}^r, S_{18}^r, S_{19}^r\} &\rightarrow \{S_4^{r+1}, S_{12}^{r+1}, S_{20}^{r+1}, S_{28}^{r+1}\} \\
\{S_{20}^r, S_{21}^r, S_{22}^r, S_{23}^r\} &\rightarrow \{S_5^{r+1}, S_{13}^{r+1}, S_{21}^{r+1}, S_{29}^{r+1}\} \\
\{S_{24}^r, S_{25}^r, S_{26}^r, S_{27}^r\} &\rightarrow \{S_6^{r+1}, S_{14}^{r+1}, S_{22}^{r+1}, S_{30}^{r+1}\} \\
\{S_{28}^r, S_{29}^r, S_{30}^r, S_{31}^r\} &\rightarrow \{S_7^{r+1}, S_{15}^{r+1}, S_{23}^{r+1}, S_{31}^{r+1}\},
\end{aligned}$$

where all groupings use the same 16-bit group mapping.

Following the footsteps of GIFT-128, TRIFLE-BC XORs key material to bit 1 and bit 2 or each nibble. However, the latter adopts the PRESENT group mapping rather than the GIFT group mapping, the main difference is that under the

PRESENT group mapping, bit i can be mapped to any bit j (where $i, j \in \{0, 1, 2, 3\}$) depending on the S-box position.

While in the forward direction (encryption), 2 of the 4 input bits to every S-box is masked with some secret key material, it is not the case from the backward direction (decryption).



TRIFLE-BC group mapping GIFT-128 group mapping

As one can see from the figures above, 2 of the 4 S-boxes (black and green) in the previous round of TRIFLE-BC is not masked by any key material. Such property does not exist in PRESENT because all bits are masked with some key material. Whereas for GIFT, bit i is mapped to bit i, thus 2 of the 4 output bits to every S-box is masked with some key material.

Lemma 1: Given $\{X_i^{r+1}, X_{i+8}^{r+1}, X_{i+16}^{r+1}, X_{i+24}^{r+1}\}$, the value of X_{4i}^r, X_{4i+3}^r can be fully determined and are independent of the secret key.

Corollary 1: Given the knowledge of an entire state X^r, the value of 16 nibbles of the state in the previous round, namely X_{4i}^{r-1} and X_{4i+3}^{r-1}, and the value of 8 nibbles of the state two rounds before, namely X_0^{r-2}, X_3^{r-2}, X_{12}^{r-2}, X_{15}^{r-2}, X_{16}^{r-2}, X_{19}^{r-2}, X_{28}^{r-2}, X_{31}^{r-2} are independent of the secret key and can be fully determined with probability 1.

In other words, without any guessing of key bit, one can decrypt 2 rounds of TRIFLE-BC and determine the value of a quarter of the state trivially.

4 Single Active Bit Differential/Linear Trail

Differential cryptanalysis (DC) is one of the 2 classical attacks on block ciphers that designers should always prove that their proposal to be resistant against it. PRESENT achieves that using S-box with differential branching number 3, guaranteeing that any single active bit input (resp. output) to an S-box will propagate to at least 2 active S-boxes in the next (resp. previous) round. On the other hand, GIFT proposed a paradigm called *Bad-Output must go to Good-Input* (BOGI), with a careful selection of S-box with desired BOGI property and

construct the bit permutation incoherent with the S-box properties, it ensures that there will not be consecutive *single active bit transitions* (S-box with single input and output bit active only). Surprisingly, `TRIFLE-BC` did not adopt either of these design philosophies; Thus not surprisingly, there exists infinitely long single active bit transitions.

A quick check on the `TRIFLE-BC` S-box shows that there exists 4 Hamming weight 1 differential transitions, namely

$$\begin{aligned}
\texttt{0x1} &\rightarrow \texttt{0x8} \\
\texttt{0x2} &\rightarrow \texttt{0x1} \\
\texttt{0x4} &\rightarrow \texttt{0x2} \\
\texttt{0x8} &\rightarrow \texttt{0x4}
\end{aligned}$$

This means that regardless of the position of the single active bit input, there always exists a single active bit output. Independently, the authors of [2] presented an attack on `TRIFLE` using one of such differential trails.

Each single active bit S-box transition holds with probability 2^{-3}. Thus, one can start from any single active bit and there is an unique single bit trail through r-round of `TRIFLE-BC` that holds with probability 2^{-3r}.

We can have a slightly better probability differential trail if we choose the input difference to the first round S-box such that it propagates to single active bit with probability 2^{-2}. For instance, $\texttt{0xD} \rightarrow \texttt{0x1}$, and this is always possible for any single active bit output. Similarly for the output difference, any single active bit input, there is some output that holds with proability 2^{-2}. For instance, $\texttt{0x2} \rightarrow \texttt{0x9}$.

Lemma 2: An optimal r-round differential characteristic has a maximum differential probability of 2^{3r-2}.

This formula is a simple generalisation of the bounds found by the designers in Table 4.2.

On a side note, the designers of `TRIFLE` argued that it is resistance against DC by showing there is no meaningful (probability lower than 2^{-127}) 50-round differential characteristics. However, they did not consider the fact that the several rounds could be extended before and after a differential characteristic, leaving very little security margin against DC.

The situation for the linear case seems worse, there exists multiple single bit linear trails as it has 12 Hamming weight 1 linear transitions:

$$\begin{aligned}
&\texttt{0x1} \rightarrow \texttt{0x1}\ ,\ \texttt{0x1} \rightarrow \texttt{0x4}\ ,\ \texttt{0x1} \rightarrow \texttt{0x8} \\
&\texttt{0x2} \rightarrow \texttt{0x1}\ ,\ \texttt{0x2} \rightarrow \texttt{0x2}\ ,\ \texttt{0x2} \rightarrow \texttt{0x8} \\
&\texttt{0x4} \rightarrow \texttt{0x1}\ ,\ \texttt{0x4} \rightarrow \texttt{0x2}\ ,\ \texttt{0x4} \rightarrow \texttt{0x4} \\
&\texttt{0x8} \rightarrow \texttt{0x2}\ ,\ \texttt{0x8} \rightarrow \texttt{0x4}\ ,\ \texttt{0x8} \rightarrow \texttt{0x8}
\end{aligned}$$

5 Subspace Transition

Apart from having 4 fixed points in the S-box (which not exactly a good feature to have), these 4 fixed points also forms an subspace transition, as first pointed out by Sarkar [3].

Having (affine) subspace transitions through the non-linear component of a cipher could lead to invariant subspace attacks (ISA) even though it is proven to be strong against several cryptanalysis [1]. Since the designers of `TRIFLE` did not mention about ISA, we study the (affine) subspace transitions though `TRIFLE-BC` S-box.

Notably, there are a total of 5 subspace transitions that maps to itself through the `TRIFLE-BC` S-box, as listed below:

$$\{\texttt{0x0}, \texttt{0x1}, \texttt{0xC}, \texttt{0xD}\} \rightarrow \{\texttt{0x0}, \texttt{0x1}, \texttt{0xC}, \texttt{0xD}\}$$
$$\{\texttt{0x0}, \texttt{0x2}, \texttt{0x9}, \texttt{0xB}\} \rightarrow \{\texttt{0x0}, \texttt{0x2}, \texttt{0x9}, \texttt{0xB}\}$$
$$\{\texttt{0x0}, \texttt{0x3}, \texttt{0x4}, \texttt{0x7}\} \rightarrow \{\texttt{0x0}, \texttt{0x3}, \texttt{0x4}, \texttt{0x7}\}$$
$$\{\texttt{0x0}, \texttt{0x5}, \texttt{0xA}, \texttt{0xF}\} \rightarrow \{\texttt{0x0}, \texttt{0x5}, \texttt{0xA}, \texttt{0xF}\}$$
$$\{\texttt{0x0}, \texttt{0x6}, \texttt{0x8}, \texttt{0xE}\} \rightarrow \{\texttt{0x0}, \texttt{0x6}, \texttt{0x8}, \texttt{0xE}\}$$

Among them, the most worrisome is the last subspace transition. Putting the BitPermutation (BP) aside for the time being, we consider a very simple ISA using the subspace $\mathbb{S} = \{\texttt{0x0}, \texttt{0x6}, \texttt{0x8}, \texttt{0xE}\}$ propagate through the $SubNibbles$ (SN), $AddRoundKey$ (AK) and $AddRoundConst$ (AC).

As shown above, if $X_i^r \in \mathbb{S}$, then after SN we have $Y_i^r \in \mathbb{S}$. Since AC updates the bit 3 of some nibbles with 0 or 1, it is equivalent to XORing $c \in \{\texttt{0x0}, \texttt{0x8}\}$ to each nibble. On the other hand, AK adds round key RK^r to bit 1 and 2 of each nibble. For some weak keys that have the round keys XORing $k \in \{\texttt{0x0}, \texttt{0x6}\}$ to each nibble, then the combination of AK and AC is equivalent to XORing some value $v \in \{\texttt{0x0}, \texttt{0x6}, \texttt{0x8}, \texttt{0xE}\} = \mathbb{S}$ to each nibble and the subspace $\mathbb{S}$ is preserved for arbitrary number of rounds.

Although the BP does destroy this subspace $\mathbb{S}$ above, it is still unclear if there could be other invariant subspace transition.

6 Conclusion

In this study, we highlighted 3 undesirable cryptographic properties of `TRIFLE-BC` which, to the best of our knowledge, do not exist in other block ciphers of similar structure, like `GIFT`, `PRESENT` and `RECTANGLE`.

References

1. Guo, J., Jean, J., Nikolic, I., Qiao, K., Sasaki, Y., Sim, S.M.: Invariant subspace attack against midori64 and the resistance criteria for s-box designs. IACR Transactions on Symmetric Cryptology **2016**(1) (Dec. 2016) 33–56
2. Liu, F., Isobe, T.: Iterative differential characteristic of trifle-bc. Cryptology ePrint Archive, Report 2019/727 (2019) https://eprint.iacr.org/2019/727.

3. Sarkar, S.: Nist lightweight cryptography competition. Offical comments on the Round 1 Candidate — TRIFLE (2019) https://csrc.nist.gov/Projects/Lightweight-Cryptography/Round-1-Candidates.

From: Antonio Florez gutierrez <antonio.florez-gutierrez@inria.fr>
Sent: Saturday, July 6, 2019 12:37 AM
To: lightweight-crypto
Cc: lwc-forum
Subject: OFFICIAL COMMENT: TRIFLE
Attachments: Cryptanalysis_of_TRIFLE_BC.pdf

Dear TRIFLE team and all,

We have found a simple linear attack on the TRIFLE-BC block cipher. This message contains a very short overview of the attack, please read the attached note for further details.

As was already pointed out by Peyrin, Sarkar, Sasaki and Sim, TRIFLE-BC, the primitive of TRIFLE, seems vulnerable to linear attacks because of the existence of 12 linear approximations of the S-box with Hamming weight 1 for both the input and the output masks, which could result in a very strong linear hull effect. We have confirmed this claim, and have found that the maximum achievable correlation for a linear distinguisher is larger than the designer's predictions because of the presence of a very large amount of one-bit linear trails. In particular, we have found that any approximation having one bit input and output masks can be used as a distinguisher has linear potential 2^-116 for 45 rounds. Any one of these approximations leads to a linear attack on full-round (50) TRIFLE-BC using 2^118 known plaintext-ciphertext pairs, and with a time complexity of 2^118 full-round encryptions. We are currently working on improved versions of this attack. This attack has already been communicated to the TRIFLE designers, who agree with our approach.

Since the data complexity on the full TRIFLE spec is claimed to be 2^64, this attack on its primitive should not reduce the security of the full cipher specification.

Best regards,

Antonio Flórez Gutiérrez

Cryptanalysis of TRIFLE-BC

Antonio Flórez Gutiérrez

July 2019

1 Introduction

As already pointed out by Peyrin, Sarkar, Sasaki and Sim in their note, TRIFLE-BC, the underlying primitive of the NIST lightweight competition candidate TRIFLE, seems vulnerable to linear attacks because of the existence of 12 linear approximations of the S-box with Hamming weight 1 for both the input and the output masks. In this note we show how these approximations lead to a very strong linear hull effect permitting linear distinguishers with a larger effectiveness than initially expected by the designers of TRIFLE.

We have been able to use this property to mount a simple key-recovery attack on full-spec (50 round) TRIFLE-BC using 2^{118} known plaintext-ciphertext pairs and requiring the same time complexity. Furthermore, we are working improvements on the key recovery in linear cryptanalysis which should allow more elaborate multiple/multidimensional linear attacks that need even less data and time.

TRIFLE-BC is a block cipher with block and key length 128. It consists of an SPN with 50 rounds. Each round acts on the state $X = X_{127} \dots X_0 = W_{31} \| \dots \| W_0$ (where W_i denotes each 4-bit nibble of the state) as follows:

- *SubNibbles*: A fixed S-box S is applied to each nibble W_i.
- *BitPermutation*: The bits of the state are rearranged according to a fixed permutation P.
- *AddRoundKey*: A 64-bit round subkey is extracted from K and XORed with bits $\{X_{4i+1}, X_{4i+2}\}$.
- *AddRoundConst*: A 6-bit round constant is XORed with bits $X_{23}, X_{19}, X_{15}, X_{11}, X_7, X_3$.

2 One-bit linear approximations of TRIFLE-BC

We will begin by describing usable linear distinguishers for 45-round TRIFLE-BC.

Table 1 contains the 1-bit approximations in the Linear Approximation Table of the TRIFLE-BC S-box containing the 1-bit to 1-bit approximations, that is, the linear approximations with masks of Hamming weight 1. Each cell contains

$$\#\{x \in \mathbb{F}_2^4 : \alpha \cdot x = \beta \cdot S(x)\} - 8$$

There are 12 biased approximations of the S-box with Hamming weight 1, and bias 2^{-3} (that is, they hold with probability $1/2 \pm 2^{-3}$. Furthermore, for each choice of the input mask α of weight one there

		β			
		1	**2**	**4**	**8**
α	**1**	2	0	2	-2
	2	-2	2	0	2
	4	2	-2	2	0
	8	0	2	-2	2

Table 1: 1-bit to 1-bit approximations in the LAT of the TRIFLE-BC S-box.

are exactly three biased approximations, and for each choice of the output mask β there are three approximations too. These approximations of the S-box can be combined to construct a linear trail with one active S-box in each round, and where every subkey mask has Hamming weight 1. We will call these trails *one-bit linear trails*.

Given a one-bit trail for $r-1$ rounds of TRIFLE-BC, it can always be extended to three different one-bit trails of r rounds by using three different biased approximations of the active S-box in the last round. This allows us to compute the total number of one-bit trails of TRIFLE-BC, as this number triples with every additional round. Since the number of one-bit trails for one round is $128 \cdot 3 = 2^7 \cdot 3$ because we have 128 choices for the input mask and three choices for the approximation of the active S-box, we conclude that the total number of one-bit trails for r rounds is $2^7 \cdot 3^r$. Because of the aforementioned symmetry of the LAT of the S-box, we should expect these trails to be distributed almost evenly among the 128*128 choices for the input and output masks, in other words, there should exist approximately $2^{-7} \cdot 3^r$ one bit linear trails between any input and output masks of Hamming weight 1. We have confirmed this experimentally with a trail-counting computer program. We arrive at the following result:

Lemma. *Given input and output masks α and β of Hamming weight 1, their Estimated Linear Potential for r rounds of TRIFLE-BC is*

$$ELP(\alpha, \beta) \simeq 2^{-4r} \cdot 2^{-7} \cdot 3^r = 2^{-(4-log_2(3))r-7}$$

Proof. The formula is deduced from the piling-up lemma and the definition of the ELP. The bias of each individual one bit trail is $\epsilon = 2^{r-1} \cdot (2^{-3})^r = 2^{-2r-1}$ because of the piling-up lemma (see [2]). The correlation contribution of each trail is thus $4\epsilon^2 = 2^{-4r}$. The ELP is the sum of the correlation contributions of all the linear trails in the linear hull of the approximation (see [3]). We can assume that the contribution of the linear trails which have more than one active S-box in each round is comparatively small. When considering the number of one bit trails and their correlation contribution, we obtain the ELP expression. □

Over 45 rounds of TRIFLE-BC, the Expected Linear Potential of the linear hull of each of the one-bit to one-bit approximations is $2^{-115.68}$. We have not tested whether this corresponds to the actual expected value of the square of the correlation in the case of the TRIFLE-BC key schedule, but the experimental results on the similar cipher PRESENT suggest that this is the case. A linear attack using one of these approximations will need $N = O(2^{115.68})$ known plaintext-ciphertext pairs. Using the model for linear cryptanalysis that can be found in [1], we estimate that with 2^{118} plaintext-ciphertext pairs, the achievable advantage should be larger than 10 bits with probability 0.95. In other words, an attack using any of these characteristics has a probability of 0.95 of reducing the time complexity of the exhaustive search from the 2^{128} encryptions of the brute-force attack to 2^{118}.

3 The attack on full-round TRIFLE-BC

Any of these $128 \cdot 128 = 2^{14}$ 45-round linear distinguishers can be used to mount a key recovery attack on full-round TRIFLE-BC using Matsui's Algorithm 2. We can use any of the linear approximations between the round subkey addition at the end of the third round and the round subkey addition at the end of the 48th round. The key recovery will be performed on the round subkeys for rounds 1, 2, 49 and 50. Computing the input bit of the approximation requires using 16 bits of the state immediately after round 1 (but before the first round key addition) as well as 8 bits of the first subkey and 2 bits of the second subkey. Computing the output bit of the approximation requires using 16 bits of the ciphertext as well as between 0 and 4 bits of the 49th round subkey and between 0 and 16 bits of the 50th round subkey. In total l bits of subkey need to be guessed, and $10 \leq l \leq 30$.

The attack would proceed as follows:

1. For each of the 2^{118} plaintext-ciphertext pairs, perform one round of encryption on the plaintext without the round key addition (this is possible because of the lack of whitening key at the beginning).

2. Classify the plaintexts and ciphertexts according to the 16 relevant bits of the (partially encrypted) plaintext and the 16 relevant bits of the ciphertext. This generates a table A of size $2^{16} \times 2^{16}$ containing the number of occurrences of each possibility for the relevant plaintext-ciphertext.

3. For each guess of the necessary subkey bits k, compute the counter T_k:

   (a) For each possibility for the 32 bits of plaintexts-ciphertexts, perform a partial two-round encryption on the plaintext and a partial two-round decryption on the ciphertext using the subkey guess and compute the parity of the approximation, if it is zero then increment the counter T_k by the number indicated on table A.

   (b) At the end T_k is the number of plaintext-ciphertext pairs for which the linear approximation is zero when the subkeys were guessed as k.

4. Choose the 2^{l-10} guesses for k where the counter T_k is furthest away from half of the data.

5. For each of these candidates, perform an exhaustive search over the rest of the key. The right key will be found with probability 0.95.

The cost of the first two steps is 2^{118} one-round encryptions, while the third step has complexity $O(2^{32} \cdot 2^l)$. The cost of step 4 is $O(2^l)$. The complexity of the last step is 2^{118} full-round encryptions. The total time complexity of the attack is thus 2^{118} TRIFLE-BC encryptions, as opposed to the 2^{128} encryptions of the brute-force attack.

4 Conclusion and further improvements on the attack

We are confident that the data and time complexities can be improved with the use of multiple or multidimensional linear cryptanalysis. The total capacity of the $128 \cdot 128 = 2^{14}$ one-bit to one-bit approximations is $2^{-101.68}$. An attack involving all these approximations would have worse complexity than a traditional brute-force attack, since it requires guessing two full consecutive round subkeys, which is essentially guessing the full 128 bit key. However, there is a trade-off between the number of approximations and the number of subkey bits which require guessing. As an illustrative example, we estimate that a multiple linear attack using one fourth of the 2^{14} linear approximations (for example, by choosing 64 bits on the input and 64 on the output) would only require $N = 2^{113}$ plaintext-ciphertext pairs to achieve an advantage of over 10 bits according to the model of [1], while guessing only 64 bits of the key if the approximations are well chosen.

The FFT-accelerated version of Algorithm 2 combined with other key recovery techniques could potentially allow for the use of more approximations, or perhaps even the use of a 44-round distinguisher instead of a 45-round distinguisher. We are currently working on improvements on this basic attack using a generic framework for the key recovery in linear cryptanalysis.

Finally, we would like to point out that even though our attack reduces the security of the TRIFLE-BC block cipher the full TRIFLE specification should still meet the security claims of the designers, as security is claimed with up to 2^{64} available data, which is a much lower number than the amount of data required for the attack on the primitive.

References

[1] Blondeau C., Nyberg K. (2017) *Improved Parameter Estimates for Correlation and Capacity Deviates in Linear Cryptanalysis.* In: IACR Transactions on Symmetric Cryptology, 2016 (2), pp 162-191. International Association for Cryptologic Research.

[2] Matsui M. (1994) *Linear Cryptanalysis Method for DES Cipher.* In: Advances in Cryptology - EUROCRYPT 1993. EUROCRYPT 1993. Lecture Notes in Computer Science, vol. 765, pp 386-397. Springer.

[3] Nyberg K. (1995) *Linear Approximation of Block Ciphers.* In: Advances in Cryptology — EUROCRYPT 1994. EUROCRYPT 1994. Lecture Notes in Computer Science, vol 950, pp 439-444. Springer.