IACR Transactions on Symmetric Cryptology
ISSN 2519-173X, Vol. 2019, No. 4, pp. 369–406. DOI:10.13154/tosc.v2019.i4.369-406

# Multiple Linear Cryptanalysis Using Linear Statistics

Jung-Keun Lee and Woo-Hwan Kim

The Affiliated Institute of Electronics and Telecommunications Research Institute (ETRI),
Daejeon, Republic of Korea
{jklee,whkim5}@nsr.re.kr

**Abstract.** We propose an improved and extended approach of the multiple linear cryptanalysis presented by A. Biryukov et al. at CRYPTO 2004 that exploits dominant and statistically independent linear trails. While they presented only rank based attacks with success probability 1, we present threshold based attacks as well as rank based ones using newly introduced statistic that is a linear combination of the component statistics for the trails and is an approximation of the LLR statistic. The rank based Algorithm 1 style attack yields the same estimate for the gain with Biryukov et al.'s Algorithm 1 style attack. For each of the threshold based Algorithm 1 style and Algorithm 2 style attacks, we provide a formula for its advantage in terms of the correlations of the trails, the data complexity, and the success probability in case the aimed success probability is not 1. Combining the threshold based attacks with the rank based ones, we get attacks each of which has better estimates for the advantage compared to the threshold based one in case the aimed success probability is close to 1. We then extend the methods to get a new framework of multiple linear attacks exploiting close-to-dominant linear trails that may not be statistically independent. We apply the methods to full DES and get linear attacks using 4 linear trails with about the same or better complexity compared to those presented at ASIACRYPT 2017 that use 4 additional trails. With data complexity less than $2^{41}$, the attack has better complexity than existing attacks on DES.

**Keywords:** multiple linear approximation · linear cryptanalysis · success probability · false alarm · attack complexity · multivariate normal distribution · statistical model · DES

## 1 Introduction

Since first introduced by Matsui [Mat93], the linear attack has been regarded as one of the most important attacks against block ciphers. The attack has evolved into many variants and extensions. The original linear attack was extended to an attack using linear hulls containing many nondominant linear trails [Nyb94, Ohk09].

B. Kaliski Jr. and M. Robshaw presented an Algorithm 1 style attack using multiple linear approximations [JR94]. But it has a severe limitation that it is applicable only when all the parity bits of the approximations are equal.

A. Biryukov et al.[BCQ04] presented Algorithm 1 style and Algorithm 2 style attacks that use independent and dominant linear trails, not imposing such condition on the parity bits. Their attack is based on a maximum likelihood approach and they provided a formula for the gain or advantage of the attack in terms of the sum of the squared correlations of the linear approximations and the data complexity. It is essentially an exhaustive search based on the rank of a statistic so that the success probability of the attack is 1. But

Received: 2019-06-01, Revised: 2019-09-01, Accepted: 2019-11-01, Published: 2020-01-31

they could not provide ways to get an estimate for the advantage in terms of the success probability and the data complexity for success probability not 1.

M. Hermelin et al. introduced the multidimensional linear attack [CHN08, HCN09, HVLN15, HCN19], exploiting the linear span of possibly statistically dependent multiple linear approximations in an attempt to improve the approach in [BCQ04]. They used LLR statistics or $\chi^2$ statistics to provide an explicit formula for the success probability in terms of the advantage and presented examples for which such estimates can be regarded reasonable. The multidimensional linear attack has proven to be powerful against some ciphers including PRESENT [Cho10].

But, in its original form, it is not more efficient than the attack in [BCQ04] when using several dominant linearly independent linear trails and, mainly from this reason, arguably the most efficient multiple linear attack on DES is not a multidimensional one as of now [BV17]. A recent attack on DES employed the method of multidimensional linear cryptanalysis but used a new statistic that is separable [FS18]. But the computational complexity of the attack does not seem to be fully established since the complexity of some additional computation is not theoretically analyzed and it does not seem to be experimentally verified.

A. Bogdanov and P. Vejre [BV17] presented a multiple linear attack on DES using 8 linearly dependent approximations, 4 of which are linearly independent. They introduced a new right key/wrong key model for the joint distribution of correlations for multiple approximations and proposed a new classifier. They were able to get an attack on DES with the smallest combined computational and data complexity at the time of the publication. Their estimates are valid under the assumption that the right key/wrong key distribution they approximated are accurate, and they presented supporting experimental results.

E. Biham and S. Perle proposed the conditional linear cryptanalysis and applied it to DES yielding an attack having the smallest complexity on the cipher as of now [BP18]. They presented an attack with both data and time complexity below $2^{42}$ while maintaining the success probability of 0.85.

In this work, we propose an improved and extended method of the multiple linear cryptanalysis presented at CRYPTO 2004. We present three versions of Algorithm 1 style attacks and Algorithm 2 style attacks. They are threshold based one, rank based one, and one combining the two. By using suitable linear combination of trail statistics, for each version, we can provide an explicit formula for the advantage in terms of the correlations of the trails, the data complexity, and the success probability. We also extend the methods and present a framework of multiple linear cryptanalysis using trails that are possibly not dominant or not statistically independent. The framework is very different from the one for the multidimensional linear attacks. Applying the Algorithm 2 style attacks to DES in a straightforward way, we get attacks that are comparable with the most efficient ones. A comprehensive list of attacks on DES can be found in [BP18].

**Our Contribution.**

- We propose an improved and extended approach of the multiple linear cryptanalysis presented by A. Biryukov et al. at CRYPTO 2004 that exploits dominant and statistically independent linear trails. We introduce a new statistic that is a linear combination of the component statistics for the trails and apply it in three versions of Algorithm 1 style and Algorithm 2 style attacks. It is an approximation of the LLR statistic and enables us to get an explicit formula for the estimate of the advantage for each attack in terms of the correlations of the trails, the data size and the success probability for each of the versions.

- We develop a new framework of multiple linear cryptanalysis that can exploit multiple linear trails that are close-to-dominant.

**Table 1:** Comparison of Recent Attacks on DES

| Attack | Data | Time | $p_S$ | Reference |
|---|---|---|---|---|
| Multiple | $2^{42.78}$ | $2^{38.86}$ | 0.85 | [BV17] |
| LC | $2^{41.00}$ | $2^{49.76}$ | 0.80 | |
| MultiDim. | $2^{41.81}$ | $2^{41.81} + O(2^{41.81})$ | 0.83 | [FS18] |
| LC | $2^{41.85}$ | $2^{41.85} + O(2^{41.85})$ | 0.85 | |
| Conditional | $2^{42.00}$ | $2^{41.00}$ | 0.82 | [BP18] |
| LC | $2^{41.90}$ | $2^{41.90}$ | 0.85 | |
| | $2^{41.00}$ | $2^{50.00}$ | 0.92 | |
| | $2^{40.00}$ | $2^{52.00}$ | 0.82 | |
| Multiple | $2^{42.75}$ | $2^{38.87}$ | 0.85 | **This Work** |
| LC | $2^{42.00}$ | $2^{42.35}$ | 0.80 | |
| | $2^{41.90}$ | $2^{43.77}$ | 0.85 | |
| | $\mathbf{2^{41.00}}$ | $\mathbf{2^{48.17}}$ | **0.80** | |
| | $\mathbf{2^{41.00}}$ | $\mathbf{2^{49.23}}$ | **0.95** | |
| | $\mathbf{2^{40.00}}$ | $\mathbf{2^{51.14}}$ | **0.80** | |
| | $\mathbf{2^{40.00}}$ | $\mathbf{2^{51.89}}$ | **0.95** | |

- We present a linear attack on the full DES that uses just 4 independent linear trails. The advantage of the attack is about the same or larger compared to that of the attack by A. Bogdanov and P. Vejre presented at ASIACRYPT 2017 that uses 4 more linear trails. Moreover, with relatively small data complexity, the attack has smaller complexity than the best currently known attack on DES [BP18] as in Table 1. We verify by experiments that the success probability and the attack complexity are correctly predicted.

**Organization of the Paper.** In Sect. 2 we present the terminology and notations used in the paper together with an overview of previous related works. In Sect. 3 we describe the new method of multiple linear attack using dominant and statistically independent trails. In Sect. 4 we reformulate classical Matsui's algorithms using a single dominant trail in the framework of Sect. 3. In Sect. 5 we present an extended framework of the multiple linear attack that deals with close-to-dominant trails that may not be statistically independent. In Sect. 6 we present a new attack on full DES applying the method and confirm that the theoretical estimates for the success probabilities and attack complexities are close to the experimental ones. In Sect. 7 we discuss additional issues and details regarding the presented framework. We conclude in Sect. 8.

# 2 Preliminaries

## 2.1 Terminology and Notations

The field with 2 elements, the ring of integers, and the field of real numbers are denoted by $\mathbb{F}_2$, $\mathbb{Z}$, and $\mathbb{R}$, respectively. For integers $i$, $j$ with $i \leq j$, the set of integers $x$ such that $i \leq x \leq j$ is denoted by $[i..j]$. The Boolean inner product of a $w$-bit mask $\gamma$ and a $w$-bit value $x$ is defined to be $\oplus_{i=0}^{w-1} \gamma[i] \cdot x[i]$ and is denoted by $\langle \gamma, x \rangle$, where $\oplus$ and $\cdot$ denote the XOR and AND operation, respectively.

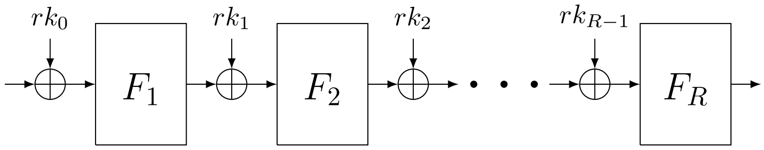

**Figure 1:** a long-key cipher

For a Boolean function $G : \mathbb{F}_2^l \to \mathbb{F}_2$, the correlation of $G$ is defined to be the imbalance

$$\frac{1}{2^l}\sum\nolimits_{x\in\mathbb{F}_2^l}(-1)^{G(x)} = \frac{1}{2^l}\left(|\{x : G(x) = 0\}| - |\{x : G(x) = 1\}|\right).$$

For a vectorial Boolean function $F : \mathbb{F}_2^l \to \mathbb{F}_2^m$, an $l$-bit mask $\gamma$, and an $m$-bit mask $\gamma'$, the (linear) correlation of $F$ with respect to the mask pair $(\gamma, \gamma')$ is defined to be the correlation of the Boolean function $G$ given by $G(x) = \langle\gamma, x\rangle \oplus \langle\gamma', F(x)\rangle$ and is denoted by $\varepsilon(\gamma, \gamma'; F)$. Thus

$$\varepsilon(\gamma, \gamma'; F) = \frac{1}{2^l}\sum\nolimits_{x\in\mathbb{F}_2^l}(-1)^{\langle\gamma,x\rangle\oplus\langle\gamma',F(x)\rangle}$$

For real numbers $\mu$ and $\sigma > 0$, the normal distribution with the mean $\mu$ and the standard deviation $\sigma$ is denoted by $\mathcal{N}(\mu, \sigma^2)$. The probability density function for $\mathcal{N}(\mu, \sigma^2)$ is denoted by $\phi(\mu, \sigma; \cdot)$ so that

$$\phi(\mu, \sigma; x) = \frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}.$$

The cumulative distribution function of the standard normal distribution is denoted by $\Phi$.

The (central) chi square distribution with degree of freedom $m$, denoted by $\chi_m^2$, is the probability distribution of $Z_1^2 + \cdots + Z_m^2$ where $Z_1, \cdots, Z_m$ are independent, standard normal random variables. The noncentral chi square distribution with degree of freedom $m$ and noncentrality parameter $\lambda$, denoted by $\chi_m'^2(\lambda)$, is the probability distribution of $Z_1^2 + \cdots + Z_m^2$ where $Z_1, \cdots, Z_m$ are independent, normal random variables with variance 1 and $\sum_{j=1}^m \mathrm{E}(Z_j)^2 = \lambda$.

For a vector $\boldsymbol{\mu}$ and a matrix $\boldsymbol{\Sigma}$, the multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$ is denoted by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Thus the probability density function for $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is

$$\frac{1}{(2\pi)^{m/2}|\det(\boldsymbol{\Sigma})|^{1/2}}e^{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}}.$$

For a vector $\boldsymbol{v} = (v_1, \cdots, v_m) \in \mathbb{R}^m$, the diagonal $m \times m$ matrix $M$ with $M(i, i) = v_i$ for each $i$ is denoted by $\mathbf{diag}(\boldsymbol{v})$. The $m \times m$ identity matrix is denoted by $\boldsymbol{I}_m$. By abuse of notation, for real vectors $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^m$, the real-valued inner product of $\boldsymbol{v}$ and $\boldsymbol{w}$ is also denoted by $\langle\boldsymbol{v}, \boldsymbol{w}\rangle$. For a real vector $\boldsymbol{v}$, the length of $\boldsymbol{v}$ is denoted by $|\boldsymbol{v}|$. Thus $|\boldsymbol{v}| = \langle\boldsymbol{v}, \boldsymbol{v}\rangle^{1/2}$. The determinant of a square matrix $M$ is denoted by $\det(M)$. The concatenation of bit strings is denoted by $\|$.

## 2.2 Linear Trails and Linear Hulls

Let $E$ be a key-alternating iterative block cipher obtained from the long key cipher $\tilde{E}$ and the key scheduling function $\psi$. Let $k$, $n$, and $R$ be the key size, the block size, and the number of rounds of $E$, respectively. So $\tilde{E}$ is a function $\mathbb{F}_2^{Rn} \times \mathbb{F}_2^n \to \mathbb{F}_2^n$ defined by

$$\tilde{E}(rk_0\|rk_1\|\cdots\|rk_{R-1}, x) = F_R(rk_{R-1} \oplus \cdots F_2(rk_1 \oplus F_1(rk_0 \oplus x))\cdots)$$

as in Fig. 1, where each $F_i$ is a fixed $n$-bit permutation, $\psi$ is a function $\mathbb{F}_2^k \to \mathbb{F}_2^{Rn}$, and $E(K, x) = \tilde{E}(\psi(K), x)$ for $(K, x) \in \mathbb{F}_2^k \times \mathbb{F}_2^n$. Now let $\gamma$ and $\gamma'$ be $n$-bit masks. We denote $\varepsilon(\gamma, \gamma'; \tilde{E}(\boldsymbol{rk}, \cdot))$ by $\varepsilon(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk})$ for each long key $\boldsymbol{rk}$. Note that $\varepsilon(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk})$ is the correlation of the cipher $\tilde{E}$ with the initial mask $\gamma$ and the final mask $\gamma'$ for the long key $\boldsymbol{rk}$. Thus

$$\varepsilon(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk}) = \frac{1}{2^n} \sum\nolimits_{x \in \mathbb{F}_2^n} (-1)^{\langle \gamma, x \rangle \oplus \langle \gamma', \tilde{E}(\boldsymbol{rk}, x) \rangle}.$$

Here, the pair $(\gamma, \gamma')$ of masks is called a linear approximation of $\tilde{E}$.

Let $\Gamma = [\Gamma_0, \cdots, \Gamma_R]$ be a linear trail of $\tilde{E}$. We define $\varepsilon^{\pi}(\Gamma; \tilde{E}, \boldsymbol{rk})$ to be

$$(-1)^{\bigoplus_{i=0}^{R-1} \langle \Gamma_i, rk_i \rangle} \varepsilon(\Gamma_0, \Gamma_R; \tilde{E}, \boldsymbol{rk})$$

and call it the *parity-adjusted correlation* of the linear hull for the long key $\boldsymbol{rk}$ and the linear trail $\Gamma$.

For each linear trail $\Gamma$, we call the set of linear trails that share the same initial mask and final mask with $\Gamma$ the linear hull of $\Gamma$. Note that for two linear trails $\Gamma$ and $\Gamma'$ in the same linear hull, $|\varepsilon^{\pi}(\Gamma; \tilde{E}, \boldsymbol{rk})| = |\varepsilon^{\pi}(\Gamma'; \tilde{E}, \boldsymbol{rk})|$.

Let $D$ be the data of size $N$ that is obtained from $\tilde{E}(\boldsymbol{rk}, \cdot)$. So $D$ consists of $N$ pairs $(P, \tilde{E}(\boldsymbol{rk}, P))$ of plaintexts and ciphertexts. We define the *undersampled correlation* of $\tilde{E}(\boldsymbol{rk}, \cdot)$ for the mask pair $(\gamma, \gamma')$ and data $D$ as

$$\frac{1}{N} \sum\nolimits_{(P,C) \in D} (-1)^{\langle \gamma, P \rangle \oplus \langle \gamma', C \rangle}$$

and denote it by $\hat{\varepsilon}(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk}, D)$.

We then define the parity-adjusted undersampled correlation $\hat{\varepsilon}^{\pi}(\Gamma; \tilde{E}, \boldsymbol{rk}, D)$ of the linear hull for the long key, the data, and the linear trail $\Gamma$ as

$$(-1)^{\bigoplus_{i=0}^{R-1} \langle \Gamma_i, rk_i \rangle} \hat{\varepsilon}(\Gamma_0, \Gamma_R; \tilde{E}, \boldsymbol{rk}, D).$$

So $|\hat{\varepsilon}^{\pi}(\Gamma; \tilde{E}, \boldsymbol{rk}, D)| = |\hat{\varepsilon}^{\pi}(\Gamma'; \tilde{E}, \boldsymbol{rk}, D)|$ for two linear trails $\Gamma$ and $\Gamma'$ in the same linear hull.

When $\boldsymbol{rk}$ is the long key corresponding to $K$ for the block cipher $E$, we also denote $\hat{\varepsilon}(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk}, D)$, $\varepsilon(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk})$, $\hat{\varepsilon}^{\pi}(\Gamma; \tilde{E}, \boldsymbol{rk}, D)$, and $\varepsilon^{\pi}(\Gamma; \tilde{E}, \boldsymbol{rk})$ by $\hat{\varepsilon}(\gamma, \gamma'; E, K, D)$, $\varepsilon(\gamma, \gamma'; E, K)$, $\hat{\varepsilon}^{\pi}(\Gamma; E, K, D)$, and $\varepsilon^{\pi}(\Gamma; E, K)$, respectively.

We also define $C(\Gamma; \tilde{E})$ to be

$$\prod_{i=0}^{R-1} \varepsilon(\Gamma_i, \Gamma_{i+1}; F_{i+1})$$

and call it the (key independent) correlation of the trail $\Gamma$. For simplicity, we omit $\tilde{E}$ or $E$ in the notation when no ambiguity is caused. For example, $\hat{\varepsilon}(\gamma, \gamma'; E, K, D)$ and $C(\Gamma; \tilde{E})$ are abbreviated as $\hat{\varepsilon}(\gamma, \gamma'; K, D)$ and $C(\Gamma)$, respectively.

For a linear trail $\Gamma$, we denote the linear hull of $\Gamma$ by $\mathcal{H}(\Gamma)$ and

$$\Big( \sum_{\Lambda \in \mathcal{H}(\Gamma)} C(\Lambda)^2 \Big)^{1/2}$$

by $C_H(\Gamma)$. $\mathcal{H}(\Gamma)$ and $C_H(\Gamma)$ are also denoted by $\mathcal{H}(\gamma, \gamma')$ and $C_H(\gamma, \gamma')$, respectively, where $\gamma$ is the initial mask and $\gamma'$ is the final mask of $\Gamma$. We have the following basic theorem regarding the key-dependent correlation of the linear hull [DR02]:

**Theorem 1.** *The linear correlation of a linear hull is the sum of the parity-adjusted correlation of the linear trails in it. That is, for masks $\gamma$, $\gamma'$ and a long key $\boldsymbol{rk}$ for $\tilde{E}$, we have*

$$\varepsilon(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk}) = \sum_{\Gamma \in \mathcal{H}(\gamma, \gamma')} (-1)^{\bigoplus_{i=0}^{R-1} \langle \Gamma_i, rk_i \rangle} C(\Gamma; \tilde{E}).$$

**Table 2:** Notations for correlations

| Notation [Abbrev.] | Definition | Meaning |
|---|---|---|
| $\varepsilon(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk})$<br>$\varepsilon(\gamma, \gamma'; \boldsymbol{rk})$ | $\frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} (-1)^{\langle \gamma, x \rangle \oplus \langle \gamma', \tilde{E}(\boldsymbol{rk}, x) \rangle}$ | correlation<br>of the linear hull |
| $\hat{\varepsilon}(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk}, D)$<br>$\hat{\varepsilon}(\gamma, \gamma'; \boldsymbol{rk}, D)$ | $\frac{1}{N} \sum_{(P,C) \in D} (-1)^{\langle \gamma, P \rangle \oplus \langle \gamma', C \rangle}$ | undersampled correlation<br>of the linear hull |
| $\varepsilon^{\pi}(\Gamma; \tilde{E}, \boldsymbol{rk})$<br>$\varepsilon^{\pi}(\Gamma; \boldsymbol{rk})$ | $(-1)^{\bigoplus_{i=0}^{R-1} \langle \Gamma_i, rk_i \rangle} \varepsilon(\Gamma_0, \Gamma_R; \tilde{E}, \boldsymbol{rk})$ | parity-adjusted<br>correlation |
| $\hat{\varepsilon}^{\pi}(\Gamma; \tilde{E}, \boldsymbol{rk}, D)$<br>$\hat{\varepsilon}^{\pi}(\Gamma; \boldsymbol{rk}, D)$ | $(-1)^{\bigoplus_{i=0}^{R-1} \langle \Gamma_i, rk_i \rangle} \hat{\varepsilon}(\Gamma_0, \Gamma_R; \tilde{E}, \boldsymbol{rk}, D)$ | parity-adjusted<br>undersampled correlation |
| $C(\Gamma; \tilde{E})$<br>$C(\Gamma)$ | $\prod_{i=0}^{R-1} \varepsilon(\Gamma_i, \Gamma_{i+1}; F_{i+1})$ | correlation<br>of the linear trail |

Theorem 2 provides an interpretation of $C_H(\gamma, \gamma')$ whose square is also called the expected linear potential of the linear hull [Nyb94].

**Theorem 2.** *The average of the squares of the correlations of a linear hull over the long keys is the sum of the squares of the correlations of the linear trails in the linear hull. That is,*

$$\mathrm{E}_{\boldsymbol{rk}}(\varepsilon(\gamma, \gamma'; \tilde{E}, \boldsymbol{rk})^2) = C_H(\gamma, \gamma')^2$$

## 2.3 Basic Statistics for Linear Cryptanalysis Using a Single Linear Trail

### 2.3.1 Statistic for Algorithm 1 Style Attack

In Algorithm 1 style attack, given a linear trail $\Gamma = [\Gamma_0, \cdots, \Gamma_R]$ for the full cipher $E$ and a key $K^*$, and a data $D = \{(P_i, E_{K^*}(P_i)) : i = 1, \ldots, N\}$ of size $N$, we consider the statistic

$$\tau^I(\Gamma, K^*, D) := N\hat{\varepsilon}(\Gamma_0, \Gamma_R; E, K^*, D)$$

and try to restore the parity bit $\beta^* = \bigoplus_{i=0}^{R-1} \langle \Gamma_i, rk_i^* \rangle$ before trial encryption, where $rk_i^*$ is the $i$-th round key derived from $K^*$ for $i = 0, \cdots, R-1$.

### 2.3.2 Statistic for Algorithm 2 Style Attack

In Algorithm 2 style attack using a single trail, we try to add more rounds to the trail before and after it. Such rounds that are added to the trail will be called outer rounds. Suppose that we have an $r$-round linear trail $\Gamma = [\Gamma_s, \cdots, \Gamma_{s+r}]$ with the correlation $\epsilon = C(\Gamma)$ for an $R$-round block cipher $E$ with $R \geq s + r$. We need to compute $\langle \Gamma_s, X_s^i \rangle \oplus \langle \Gamma_{s+r}, X_{s+r}^i \rangle$ for each data entry $(P_i, C_i)$ with $C_i = E(K^*, P_i)$, where $X_s^i$ and $X_{s+r}^i$ are the intermediate states for the start of the $s$-th round and the end of the $(s + r - 1)$-th round, respectively, that is $X_s^i = E|_0^{s-1}(K, P_i)$ and $E|_{s+r}^{R-1}(K, X_{s+r}^i) = C_i$ for each $K$. (For integers $r_1, r_2$ with $0 \leq r_1 \leq r_2 < R$, $E|_{r_1}^{r_2}$ denotes the subcipher of $E$ spanning from the start of the $r_1$-th round to the end of the $r_2$-th round. So, for example, $E|_0^{R-1} = E$.) In the computation of $\langle \Gamma_s, X_s^i \rangle \oplus \langle \Gamma_{s+r}, X_{s+r}^i \rangle$, part of the round key bits for the outer rounds are involved. We call such bits of outer round keys as outer key bits and denote the concatenation of the outer key bits by $\kappa$. Thus $\langle \gamma_s, X_s^i \rangle \oplus \langle \gamma_{s+r}, X_{s+r}^i \rangle$ can be expressed as $g(\kappa, P_i, C_i)$ for some function $g$. Then we define

$$\tau(\Gamma, K^*, \kappa, D) := \sum_i (-1)^{g(\kappa, P_i, C_i)}.$$

Sometimes, $D = \{P_i : i = 1, ..., N\}$ is data of plaintexts only and in this case $\tau(\Gamma, K^*, \kappa, D)$ is defined as

$$\sum_i (-1)^{g(\kappa, P_i, E_{K^*}(P_i))}.$$

In most of Algorithm 2 style attacks, we need to calculate $\tau(\Gamma, K^*, \kappa, D)$ for all $\kappa$'s. In this step that is called the analysis phase, we usually perform the *data compression* first to reduce the computational cost. The data compression in linear attack is a process that collapses the data into a new data with multiplicity considering the outer round computations. So the "compression function" $H_c : \mathbb{F}_2^{2n} \to \mathbb{F}_2^d$ with $2^d \ll N$ we need to get for the data compression is one such that the computation of $g(\kappa, P, C)$ can be carried out using $\kappa$ and $H_c(P, C)$ or such that there is a function $h$ such that $g(\kappa, P, C) = h(\kappa, H_c(P, C))$ for any $(P, C)$. Once we have a compression function, we apply it to the data to get the compressed data

$$\{(v, n_v) \in \mathbb{F}_2^d \times \mathbb{Z} : n_v = |\{i : H_c(P_i, C_i) = v\}|\}.$$

Then we can compute $\tau(\Gamma, K^*, \kappa, D)$ for each $\kappa$ as

$$\sum_{v:h(\kappa,v)=0} n_v - \sum_{v:h(\kappa,v)=1} n_v.$$

If $h(\kappa, v)$ can be expressed as $h'(\kappa \oplus v)$, we can use the Fast Walsh-Hadamard Transform (FWHT) to reduce the computational complexity as described in [CSQ07].

In some of Algorithm 2 style attacks, we also try to recover the parity bit

$$\beta^* = \bigoplus_{i=s}^{s+r-1} \langle \Gamma_i, rk_i^* \rangle$$

additionally before trial encryption.

## 2.4 Previous Works

### 2.4.1 Linear attacks using a single approximation

Matsui's attack on DES [Mat93] used a dominant linear trail and implicitly assumed the following hypotheses:

**Hypothesis 1.** *For a dominant trail $\Gamma$ of $E$, $\varepsilon^\pi(\Gamma; K^*)$ is very close to $C(\Gamma)$ regardless of the key $K^*$.*

**Hypothesis 2.** *In Algorithm 2 style attack using a dominant trail $\Gamma$, $\tau(\Gamma, K^*, \kappa, D)/2^n$ is very close to 0 for any wrong outer key $\kappa$ with the full codebook $D$ for $K^*$.[BCQ04]*

In general, we can regard $\varepsilon^\pi(\Gamma; K^*)$ as a random variable, letting the key $K^*$ vary. Also, letting the key $K^*$ and the data $D$ of size $N$ vary, we can regard $\hat{\varepsilon}^\pi(\Gamma; K^*, D)$ as a random variable. In Algorithm 2 style attack, under Hypothesis 1 and 2, it can be presumed that

$$(-1)^{\beta^*}\tau(\Gamma, K^*, \kappa, D)/N \sim \mathcal{N}(C(\Gamma), 1/N)$$

for the correct outer key $\kappa$ and

$$\tau(\Gamma, K^*, \kappa, D)/N \sim \mathcal{N}(0, 1/N)$$

for wrong $\kappa$'s where $\beta^*$ is the parity bit.

A. Selçuk presented a rank based Algorithm 2 style attack using a dominant trail assuming that the wrong key statistics are independent and that the right key statistics and the order statistics for the wrong key statistics are independent [Sel08]. He considered the rank of the outer key candidates $\kappa$ with respect to the value $|\tau(\Gamma, K^*, \kappa, D)|$ for given data $D$ as in [JV03] and derived an attack with the success probability

$$\Phi\left(\sqrt{N}|C(\Gamma)| - \Phi^{-1}(1 - 2^{-a-1})\right), \tag{1}$$

where $a$ is the advantage of the attack defined as follows:

**Definition 1** (Advantage)**.** A linear attack has advantage $a$ with success probability $p$ if the search for the right candidate key stops after trying $2^{l-a-1}$ candidate keys on average when the number of candidate keys is $2^l$ and the search is successful with probability $p$.

Thus a threshold based linear attack has advantage $a$ with success probability $p$ if the right candidate key satisfies the threshold condition with probability $p$ and there are $2^{l-a}$ wrong candidate keys on average satisfying the same threshold condition. In a rank based attack, the advantage can be estimated differently depending on the attack method.

- In a rank based attack using the order statistic (e.g. [Sel08, JV03, HCN19]), one considers the probability that the right candidate key is found among the $2^{l-a}$ highest ranked candidates as the success probability for each preassigned advantage $a$. When the search is performed among the $2^{l-a}$ highest ranked candidates, it is expected to stop after trying $2^{l-a-1}$ ones on average if successful.
- In our rank based attack, we analyze the average proportion of candidate keys that are ranked higher than the right key candidate similarly as in [BCQ04].
- In our combined attack, we analyze the average proportion of candidate keys that are ranked higher than the right key candidate and satisfies the threshold condition after preassigning a threshold parameter corresponding to the aimed success probability.

We say that a linear trail $\Gamma$ is significant in its hull if $C_H(\Gamma) \neq 0$ and $|C(\Gamma)|/|C_H(\Gamma)|$ is considerably large. Regarding significant trails, Hypotheses 3 and 4 are postulated [BN16, BN17].

**Hypothesis 3** (Right key randomization)**.** *Let $\Gamma$ be a significant trail such that its linear hull does not have other significant trails. Then $\varepsilon^\pi(\Gamma; K^*)$ has the normal distribution with*

$$\textit{mean } C(\Gamma) \textit{ and variance } C_H(\Gamma)^2 - C(\Gamma)^2$$

*as the key $K^*$ varies.*

*Also, $\hat{\varepsilon}^\pi(\Gamma; K^*, D)$ has the normal distribution with*

$$\textit{mean } C(\Gamma) \textit{ and variance } C_H(\Gamma)^2 - C(\Gamma)^2 + B/N$$

*as the key $K$ and the data $D$ of size $N$ vary, where $B = (2^n - N)/(2^n - 1)$ or 1 depending on whether the data is sampled with or without replacement.*

Note that in many recent works, the distribution of $\varepsilon(\Gamma_0, \Gamma_R, ; K^*)$ is considered and it is presumed to have the normal distribution with mean 0 and variance $C_H(\Gamma)^2$(e.g. [DR07]). If the linear hull of a significant trail $\Gamma$ contains other significant ones, $\varepsilon^\pi(\Gamma; K^*)$ does not have the normal distribution in general and Algorithm 1 style attacks described in [RN13, AR16] can be applicable in this case.

**Hypothesis 4** (Wrong key randomization)**.** *In Algorithm 2 style attack using a significant trail $\Gamma$ whose linear hull does not have other significant trails, $\tau(\Gamma, K^*, \kappa, \bar{D})/2^n$ has the normal distribution with*

$$\textit{mean } 0 \textit{ and variance } 2^{-n}$$

*as the wrong outer key $\kappa$ varies when $\bar{D}$ is the full codebook for $K^*$.*

*Also, $\tau(\Gamma, K^*, \kappa, D)/N$ has the normal distribution with*

$$\textit{mean } 0 \textit{ and variance } 2^{-n} + B/N$$

*as the wrong outer key $\kappa$ and the data $D$ of size $N$ vary, where $B$ is the same as in the preceding hypothesis.*

Hypothesis 1 and 2 can be considered to be in accordance with Hypothesis 3 and 4 only when $N \ll 2^n$.

### 2.4.2 Linear attacks using multiple approximations

**Multiple Linear Cryptanalysis.** B. Kaliski Jr. and M. Robshaw [JR94] presented an Algorithm 1 style attack exploiting $m$ linear trails $\Gamma^1, \cdots, \Gamma^m$ using the statistic

$$\sum_j^m \epsilon_j \tau^I(\Gamma^j, D)/(N \sum_j |\epsilon_j|)$$

varying data $D$ of size $N$, where $\epsilon_j = C(\Gamma^j)$ for each $j$. (For the definition of the component statistic $\tau^I(\Gamma^j, D)$ for each $j$, see Sect. 3.1.) But the attack is valid only when the $m$ parity bits are always the same regardless of $K^*$. They assumed the independence of the statistics $\tau^I(\Gamma^1, D), \cdots, \tau^I(\Gamma^m, D)$ as $D$ varies.

A. Biryukov et al. [BCQ04] used the "quadratic" statistic

$$\sum_j \left((-1)^{\beta_j}\epsilon_j - \tau^I(\Gamma^j, D)/N\right)^2 = \epsilon^2 - 2\sum_j (-1)^{\beta_j}\epsilon_j \tau^I(\Gamma^j, D)/N + \sum_j \tau^I(\Gamma^j, D)^2/N^2 \tag{2}$$

in their Algorithm 1 style attack exploiting $m$ dominant and statistically independent trails, where $\beta_1, \cdots, \beta_m$ are binary variables. Letting $\boldsymbol{\beta}^* = (\beta_1^*, \cdots, \beta_m^*)$ be the correct parity bit vector as described in Sect. 3.1, they assumed that the statistic vector

$$((-1)^{\beta_1^*}\tau^I(\Gamma^1, D)/N, \cdots, (-1)^{\beta_m^*}\tau^I(\Gamma^m, D)/N)$$

approximately has the distribution of $m$-variate normal distribution with mean

$$(\epsilon_1, \cdots, \epsilon_m)$$

and covariance $(1/N)\boldsymbol{I}_m$. They derived an explicit formula for the gain of the attack with rank based approach, where the gain of an attack is defined as follows [BCQ04]:

**Definition 2** (Gain)**.** The gain of a linear attack is

$$-\log_2 \frac{2L-1}{2^k},$$

with $L$ being the average number of keys that are checked until the correct $k$-bit key for the block cipher is found.

Note that for a linear attack with success probability 1, its gain is almost equal to its advantage. The gain of their attack was obtained as

$$-\log_2 \left(2^{-m+1} \sum_{\boldsymbol{\beta} \neq \boldsymbol{\beta}^*} \Phi\left(-\frac{\sqrt{N}}{2}|\boldsymbol{c}_{\boldsymbol{\beta}} - \boldsymbol{c}_{\boldsymbol{\beta}^*}|\right) + 2^{-m}\right),$$

where

$$\boldsymbol{c}_{\boldsymbol{\beta}} = ((-1)^{\beta_1}\epsilon_1, \cdots, (-1)^{\beta_m}\epsilon_m)$$

based on the estimate of the average number of false alarms $\boldsymbol{\beta}$ obtained as

$$\sum_{\boldsymbol{\beta} \neq \boldsymbol{\beta}^*} \Phi\left(-\frac{\sqrt{N}}{2}|\boldsymbol{c}_{\boldsymbol{\beta}} - \boldsymbol{c}_{\boldsymbol{\beta}^*}|\right). \tag{3}$$

The success probability of their attack is 1 since it is an efficient exhaustive search based on the rank of the parity bit vectors $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_m)$ with respect to the statistic (2). They used similar statistic to get an Algorithm 2 style attack, where they assumed in addition that $\tau(\Gamma^j, \kappa, D)/N$ has a normal distribution with mean 0 regardless of wrong outer key $\kappa$. But it is not clear whether they incorporated the distribution of the statistics for wrong outer keys in their analysis of the Algorithm 2 style attack.

**Multidimensional Cryptanalysis.** M. Hermelin et al. presented the framework of multidimensional linear cryptanalysis [CHN08, HCN09, HCN19] based on prior works [BJV04, BV08]. They presented Algorithm 1 and Algorithm 2 style attacks using various statistics, e.g., LLR(log-likelihood ratio) and $\chi^2$ statistic with different statistical models.

**Definition 3** (LLR statistic)**.** Let $\mathcal{D}^0$ and $\mathcal{D}^1$ be two probability distributions on $\{0,1\}^m$ with the pdf $p^0$ and $p^1$, respectively. Let $S$ be a multiset with cardinality $N$ consisting of elements of $\{0,1\}^m$. Let $\hat{p}^S : \{0,1\}^m \to \mathbb{R}$ be the pdf of the empirical probability distribution obtained from $S$ so that

$$\hat{p}^S(\boldsymbol{x}) = |\{\boldsymbol{y} \in S : \boldsymbol{y} = \boldsymbol{x}\}|/N$$

for each $\boldsymbol{x} \in \{0,1\}^m$. Then the LLR statistic $LLR(S, \mathcal{D}^1, \mathcal{D}^0)$ is defined as

$$LLR(S, \mathcal{D}^1, \mathcal{D}^0) = N \sum\nolimits_{\boldsymbol{x}} \hat{p}_S(\boldsymbol{x}) \frac{\log p^1(\boldsymbol{x})}{\log p^0(\boldsymbol{x})}.$$

In the rank based method using the LLR statistic or $\chi^2$ statistic, they assume that the wrong key statistics are independent and that the right key statistics and the order statistics for the wrong key statistics are independent similarly as in [Sel08]. The LLR method is more powerful but can be regarded to be valid only when an accurate knowledge of the key-dependent behavior of certain probability distribution is accompanied. So $\chi^2$ method is usually preferred(e.g. [Cho10]). In the $\chi^2$ method, the distribution of the $m$-bit value

$$(\langle \gamma^1, x\rangle \oplus \langle \gamma'^1, E(K,x)\rangle, \cdots, \langle \gamma^m, x\rangle \oplus \langle \gamma'^m, E(K,x)\rangle) \tag{4}$$

is considered varying $x \in \mathbb{F}_2^n$ where each $(\gamma^j, \gamma'^j)$ is a pair of input mask and output mask for a linear trail $\Gamma^j$ such that $\Gamma^1, \cdots, \Gamma^m$ are linearly independent.

**Definition 4** ($\chi^2$ statistic)**.** The $\chi^2$ statistic $\chi^2(K)$ is

$$2^m N \sum_{i=0}^{2^m-1} \frac{(\eta_i^K - 2^{-m})^2}{2^{-m}},$$

where $\eta_i^K$ is the probability that the above $m$-bit value (4) takes the value $i$ for each $i = 0, \cdots, 2^{m-1}$.

In the $\chi^2$ method, they used the fact that the key-dependent capacity defined as

$$\sum_{(\gamma,\gamma'):\text{linear combination of } (\gamma^j,\gamma'^j)'s} \varepsilon(\gamma, \gamma'; K)^2$$

is equal to $\chi^2(K)/(2^m N)$.

So in the $\chi^2$ method, the distribution of the statistic $\sum_{(\gamma,\gamma')\in\boldsymbol{\Lambda}} \varepsilon(\gamma, \gamma; K)^2$ and its undersampled counterpart can be considered for some set $\boldsymbol{\Lambda}$ of approximations instead. The multidimensional cryptanalysis has the advantage that the size of the data needed for the attack is inversely proportional to the capacity. But as observed in [HCN19], increasing the number of linear approximations may decrease the advantage of the attack using $\chi^2$ statistics. Excluding linear approximations with very small squared correlation can increase the advantage. Recently, K. Nyberg [Nyb19] introduced the affine linear cryptanalysis that is expected to improve the complexity of the previous $\chi^2$ based multidimensional attack by discarding trivial approximations.

The key-dependent behavior of the capacity for the multidimensional cryptanalysis was studied in [HVLN15]. Some statistical models for right keys and wrong keys in the key-dependent setting appeared in [BN17].

**Multivariate Linear Cryptanalysis.** A. Bogdanov et al. [BTV18] presented the multivariate linear cryptanalysis that considers the distribution of

$$(\tau(\Gamma^1, K, \kappa_1, D)/N, \cdots, \tau(\Gamma^m, K, \kappa_m, D)/N).$$

They presumed Hypothesis 5 and 6 regarding multiple linear approximations $(\gamma^j, \gamma'^j)$ $(j = 1, \cdots, m)$ of a block cipher with linearly independent masks.

**Hypothesis 5.** *The correlation vector* $(\varepsilon(\gamma^1, \gamma'^1; K), \cdots, \varepsilon(\gamma^m, \gamma'^m; K))$ *follows some* $m$*-variate probability distribution* $\mathcal{D}_m$ *as the key* $K$ *varies.*

**Hypothesis 6.** *In Algorithm 2 style attack, for a wrong outer key, the correlation vector follows the* $m$*-variate normal distribution* $\mathcal{N}(\mathbf{0}, 2^{-n}\boldsymbol{I}_m)$*.*

Under the hypotheses, they derived the following result:

**Theorem 3.** *Assume that the correlation of any combination of two approximations among* $(\gamma^j, \gamma'^j)$*'s is 0. Then the vector of undersampled correlations measured with data of size* $N$ *has distribution* $\mathcal{D}_m + \mathcal{N}(\mathbf{0}, 2^{-n}\boldsymbol{I}_m)$ *as the key and the data varies. For the wrong outer keys, the undersampled correlation has distribution* $\mathcal{N}(\mathbf{0}, (2^{-n} + 1/N)\boldsymbol{I}_m)$*.*

They also considered the $\chi^2$ statistic

$$\mathcal{T}(\boldsymbol{\kappa}) = \frac{1}{N}\sum_{j=1}^{m} \tau(\Gamma^j, K^*, \kappa_j, D)^2, \tag{5}$$

where $\kappa_j$'s are candidate outer keys involved in the outer computation for computation of the undersampled correlation for $\Gamma^j$ and $\boldsymbol{\kappa}$ is obtained by combining $\kappa_j$'s removing redundancy. The $\chi^2$ method does not require accurate knowledge on the key-dependent distribution of correlation vectors and the $\chi^2$ statistic is separable in that it is the sum of quantities that depend on part of the outer key.

### 2.4.3 Issues with linear attacks on full DES using multiple approximations

A. Biryukov et al. applied their method to full DES but were not able to get an efficient attack [BCQ04]. One of the main reason is that they didn't know how to get the tradeoff between the success probability and the advantage for the attack. For example, their Algorithm 2 style attack with success probability 1 does not give satisfactory advantage when using the linear trails in [BV17]. The LLR based multidimensional attack has not been so successful since no efficient ways to perform the analysis phase of computing the statistic for all candidate keys have been known when using several trails with large squared correlation: when the number of the guessed outer key bits is $k_O$, the analysis phase has been regarded to require at least $O(2^{k_O})$ computation. (In fact, LLR statistic is separable when using dominant and statistically independent trails though it seems that this fact has not been known before at least in the cryptographic context.[1]) Note that the analysis phase was performed rather efficiently in the attack on PRESENT [Cho10]. The approach in [FS18] tried to resolve this issue regarding DES by introducing a new separable statistic. When we apply the multivariate linear attack [BTV18] using some of the trails presented in [BV17], the advantage of the attack is not large enough to yield an efficient attack as discussed in Sect. 7. When using a small number of dominant and statistically independent linear trails in the $\chi^2$ based multidimensional attack, the linear approximations obtained by linear combinations of the trails (other than those obtained from themselves) do not help to decrease the attack complexity due to having very small squared correlations. So $\chi^2$ based multidimensional attack is not more efficient than the

[1] The separability was noticed by K. Nyberg during communication with us.

multivariate linear attack. The linear statistic we use enables us to get large advantage when used with a small number of dominant trails in an attack combining the threshold based method and the rank based one. Also, it is separable so that we can perform the analysis phase rather efficiently running a small number of FWHT(fast Walsh-Hadamard Transform)'s [CSQ07] on relatively small-sized domains. Thus we can get simple and arguably the most efficient multiple linear attacks on full DES using 4 of the linear trails in [BV17].

# 3 New Methods of Multiple Linear Cryptanalysis

In this section, we present three Algorithm 1 style attacks and three Algorithm 2 style attacks that use dominant and independent linear trails. The three Algorithm 1 style attacks are Algorithm 1MT, Algorithm 1MR, and Algorithm 1MC. Algorithm 1MT and Algorithm 1MR are threshold based and rank based, respectively, and Algorithm 1MC combines the two methods. Similarly, we have three variants Algorithm 2MT, Algorithm 2MR, and Algorithm 2MC of Algorithm 2 style attacks. Algorithm 1MR yields the same advantage as the Algorithm 1 style attack given in [BCQ04] though different statistics are used. Algorithm 2MR yields similar but more precise estimation of the Algorithm 2 style attack with more reasonable statistical model for wrong key statistics. All the attack methods are quite new. First, we define a new linear statistic both for Algorithm 1 and Algorithm 2 style attacks. Also the analysis using this linear statistic with the multivariate normal distribution considering the wrong key types is novel. *Throughout this section, we assume that the unknown $k$-bit key $K^*$ and its corresponding long key* rk$^*$ *as well as the long key cipher $\tilde{E}$ and the block cipher $E$ with $n$-bit blocks are fixed.* We also assume that we have fixed $m$ linear trails $\Gamma^1, \cdots, \Gamma^m$ spanning from the start of the $s$-th round to the end of the $(s + r - 1)$-th round such that each trail is dominant in its linear hull or $|C_H(\Gamma^j)| \approx |C(\Gamma^j)|$ for each $j$ and the trails are statistically independent. An extended method that can exploit close-to-dominant linear trails that may not be statistically independent will be presented in Sect. 5. We assume further that the size $N$ of the data is very small compared to $2^n$ so that $1/N + 2^{-n} \approx 1/N$. We let $\epsilon_j = C(\Gamma^j)$ for each $j$ and let $\epsilon = (\sum_j \epsilon_j^2)^{1/2}$.

## 3.1 Description of Algorithm 1 Style Attacks

In this attack, we use full round trails and try to recover parity bits. So we assume that we have $m$ dominant linear trails $\Gamma^j$'s for the full cipher. Let $\beta_j^*$ be the parity bit for the $j$-th trail for each $j = 1, \cdots, m$. Let $D$ be the available data of size $N$. We consider the following linear statistic

$$T^I(\boldsymbol{\beta}, D) := \sum_{j=1}^{m} (-1)^{\beta_j} \epsilon_j \tau_j^I(D)$$

for each candidate $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_m)$ of $\boldsymbol{\beta}^* = (\beta_1^*, \cdots, \beta_m^*)$, where

$$\tau_j^I(D) := \tau^I(\Gamma^j, D) = N\hat{\varepsilon}(\Gamma^j; K^*, D)$$

for each $j$. Let $t$ be the threshold parameter that will be determined according to the aimed success probability. We have three versions of the Algorithm 1 style attacks, called Algorithms 1MT, 1MR, and 1MC, that are described below. They have different estimates on the success probability and the attack complexity.

### 3.1.1 Algorithm 1MT: Threshold based

We first compute $\tau_j^I(D)$ for each $j$ and then compute $T^I(\boldsymbol{\beta}, D)$ for each $\boldsymbol{\beta}$. We determine $\boldsymbol{\beta}$ to be possibly correct if it satisfies the threshold condition

$$T^I(\boldsymbol{\beta}, D) \geq tN\epsilon^2.$$

For each possibly correct $\boldsymbol{\beta}$'s, check whether it is indeed the correct one by trial encryption. The attack fails if $\boldsymbol{\beta}^*$ does not satisfy the threshold condition.

### 3.1.2 Algorithm 1MR: Ranking based

After computing $T^I(\boldsymbol{\beta}, D)$ for each $\boldsymbol{\beta}$ as in Algorithm 1MT, sort the list of all $\boldsymbol{\beta}$'s according to the value $T^I(\boldsymbol{\beta}, D)$ in the descending order. Then check $\boldsymbol{\beta}$'s one by one considering the order in the list by trial encryption.

### 3.1.3 Algorithm 1MC: Combined

After computing $T^I(\boldsymbol{\beta}, D)$ for each $\boldsymbol{\beta}$, pick out $\boldsymbol{\beta}$'s for which $T^I(\boldsymbol{\beta}, D) \geq tN\epsilon^2$. Sort the list of selected $\boldsymbol{\beta}$'s according to the statistic $T^I(\boldsymbol{\beta}, D)$ in the descending order. Then check all the $\boldsymbol{\beta}$'s in the sorted list in the order until we find the correct one by trial encryption. The attack fails if $\boldsymbol{\beta}^*$ is not in the list.

### 3.1.4 False Alarm Probability

We need to clarify the meaning of the false alarm probability for each of the versions. In Algorithm 1MT, the false alarm probability is the average proportion of the $\boldsymbol{\beta}$'s that satisfy the threshold conditions to $2^m$. In Algorithm 1MR, it is the average proportion of the $\boldsymbol{\beta}$'s that are ranked higher than $\boldsymbol{\beta}^*$ in terms of the statistic to $2^m$. In Algorithm 1MC, it is the average proportion of the $\boldsymbol{\beta}$'s that satisfy the threshold condition and are ranked higher than $\boldsymbol{\beta}^*$ in terms of the statistic to $2^m$. False alarm probabilities can be also defined similarly for the attacks to follow in this work.

## 3.2 Description of Algorithm 2 Style Attacks

We denote by $\kappa_j$ the variable representing the candidate outer key for the trail $\Gamma^j$ as described in Sect. 2.3. For simplicity, we assume that bits of $\kappa_1 \| \cdots \| \kappa_m$ are identical regardless of $K$ or are independent. Combining $\kappa_j$'s by taking bits from $\kappa_j$'s and removing redundancy, we get the candidate outer key $\boldsymbol{\kappa}$. We denote by $k_O^j$ the number of bits of $\kappa_j$ for each $j$. We denote by $k_O$ the number of bits of $\boldsymbol{\kappa}$ so that $2^{k_O}$ is the number of candidate outer keys. As described in Sect. 2.3, we can get the statistic

$$\tau_j(\kappa_j, D) := \tau(\Gamma^j, \kappa_j, D)$$

for each $j$ by outer round computation. Let $t$ be the threshold parameter and let $D$ be the available data of size $N$. We also have three versions of the Algorithm 2 style attacks, called Algorithms 2MT, 2MR, and 2MC, that are described below. In each of the versions, the first step is to compute the following linear statistic

$$T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D) := \sum_j (-1)^{\beta_j} \epsilon_j \tau_j(\kappa_j, D) \qquad (6)$$

for each $(\boldsymbol{\kappa}, \boldsymbol{\beta})$, where $\boldsymbol{\beta}$ is a candidate for the binary vector $\boldsymbol{\beta}^*$ consisting of the correct parity bits.

**Alg. 1** Algorithm 2MT

[Step 1] For each $j = 1, \cdots, m$, get a list $[((-1)^{\beta_j}\epsilon_j\tau_j(\kappa_j, D), (\kappa_j, \beta_j)) : \kappa_j \in \mathbb{F}_2^{k_O^j}, \beta_j \in \mathbb{F}_2]$ of size $2^{k_O^j+1}$.
[Step 2] Compute $T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D)$ for *some* $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s and get the list of all $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s for which $T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D) \geq tN\epsilon^2$.
[Step 3] For each $(\boldsymbol{\kappa}, \boldsymbol{\beta})$ in the list, try to recover all the key bits by trial encryption.

### 3.2.1 Algorithm 2MT

Algorithm 2MT is performed as in Alg. 1. After Step 1, we pick out $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s satisfying the threshold condition

$$T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D) \geq tN\epsilon^2.$$

Then we check each selected one by trial encryption. In general we do not need to compute $T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D)$ for all $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s in Step 2 since we might be able to discard many $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s for free if, for example, we sort the $m$ lists obtained in Step 1 before Step 2.

### 3.2.2 Algorithm 2MR

After Step 1 of Algorithm 2MT, sort the list of all $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s according to the statistic $T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D)$ in the descending order. Then check $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s one by one in the order by trial encryption.

### 3.2.3 Algorithm 2MC

After Step 1 and 2 of Algorithm 2MT, sort the resulting list of $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s according to the value of $T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D)$ in the descending order. Then we try all the $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s in the sorted list in the order until we find the correct one.

## 3.3 Statistical Model, Success Probability, and Advantage

*We fix the cipher $E$ and the unknown key $K^*$* in this section and thus we will mostly drop $E, K^*$ from the notations regarding the statistics.

### 3.3.1 Algorithm 1 Style Attacks

Letting $D$ vary, we can regard

$$((-1)^{\beta_1^*}\epsilon_1\tau_1^I(D), \cdots, (-1)^{\beta_m^*}\epsilon_m\tau_m^I(D))$$

as a vector-valued random variable. For simplicity, we assume that the parity bits are independent. To estimate the success probability and the false alarm probability of Algorithm 1M, we presume the following hypothesis that extends Hypothesis 3 considering the independence of the distribution of the undersampled correlations of the trails together with the assumption that $N \ll 2^n$.

**Hypothesis 7.** *The random variable $((-1)^{\beta_1^*}\epsilon_1\tau_1^I(D), \cdots, (-1)^{\beta_m^*}\epsilon_m\tau_m^I(D))$ has the m-variate normal distribution with mean*

$$\boldsymbol{\mu} = (N\epsilon_1^2, \cdots, N\epsilon_m^2)$$

*and covariance matrix*

$$\boldsymbol{\Sigma} = \mathbf{diag}(N\epsilon_1^2, \cdots, N\epsilon_m^2).$$

**Algorithm 1MT.** Under Hypothesis 7, the success probability $p_S(t)$ is

$$\mathrm{Pr}_D(T^I(\boldsymbol{\beta}^*, D) \geq tN\epsilon^2) = \mathrm{Pr}_{(X_1,\cdots,X_m)\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\sum_j X_j \geq tN\epsilon^2),$$

which equals $\Phi((1-t)\sqrt{N}\epsilon)$ by Proposition 1.

**Proposition 1.** *Let $\boldsymbol{X} = (X_1, \cdots, X_m)$ be a random variable having the $m$-variate normal distribution with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma} = \boldsymbol{\sigma}\boldsymbol{\sigma}^T$, where $\boldsymbol{\sigma}$ is an invertible matrix. Let $\boldsymbol{a} \neq \boldsymbol{0}$ be an $m$-dimensional vector and $b$ a real number. Then*

$$\mathrm{Pr}_{\boldsymbol{X}}(\langle \boldsymbol{a}, \boldsymbol{X}\rangle + b \geq 0) = \Phi\left((\langle \boldsymbol{a}, \boldsymbol{\mu}\rangle + b)/|\boldsymbol{\sigma}^T \boldsymbol{a}|\right).$$

We will use Proposition 1 extensively in this work. Its proof will be provided in the Appendix. To consider the false alarm probability, we classify the wrong keys as follows: Let $J$ be a proper subset of $[1..m]$. $\boldsymbol{\beta}$ is called the *wrong key of type $J$* if

- $\beta_j = \beta_j^*$ for $j \in J$, and
- $\beta_j \neq \beta_j^*$ for $j \notin J$.

Note that when $\boldsymbol{\beta}$ is the wrong key of type $J$, the probability that it is a false alarm is

$$\begin{aligned}&\mathrm{Pr}_D\left(T^I(\boldsymbol{\beta}, D) \geq tN\epsilon^2\right)\\ &= \mathrm{Pr}_{(X_1,\cdots,X_m)\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\textstyle\sum_{j\in J} X_j - \sum_{j\notin J} X_j \geq tN\epsilon^2)\end{aligned}$$

which equals

$$\Phi\left(\frac{\sqrt{N}}{\epsilon}(\sum_{j\in J}\epsilon_j^2 - \sum_{j\notin J}\epsilon_j^2 - t\epsilon^2)\right)$$

by Proposition 1. Thus, considering all the wrong key types, we have

**Theorem 4.** *Under Hypothesis 7, the false alarm probability $p_{\mathrm{fa}}(t)$ of Algorithm 1MT is*

$$\frac{1}{2^m}\sum_{J\subsetneq[1..m]}\Phi\left(\frac{\sqrt{N}}{\epsilon}(\sum_{j\in J}\epsilon_j^2 - \sum_{j\notin J}\epsilon_j^2 - t\epsilon^2)\right).$$

**Algorithm 1MR.** The success probability is 1 since we try all $\boldsymbol{\beta}$'s. We consider the same vector-valued random variable $((-1)^{\beta_1^*}\epsilon_1\tau_1^I(D), \cdots, (-1)^{\beta_m^*}\epsilon_m\tau_m^I(D))$ as in the case of Algorithm 1MT. For a wrong key $\boldsymbol{\beta}$ of type $J$, the probability that $\boldsymbol{\beta}$ is a false alarm is

$$\begin{aligned}&\mathrm{Pr}_D\left(T^I(\boldsymbol{\beta}, D) \geq T^I(\boldsymbol{\beta}^*, D)\right)\\ &= \mathrm{Pr}_{(X_1,\cdots,X_m)\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left(\textstyle\sum_{j\notin J} X_j \leq 0\right)\end{aligned}$$

which equals $\Phi\left(-\sqrt{N\sum_{j\notin J}\epsilon_j^2}\right)$ by Proposition 1. Considering all the wrong key types, we have

**Theorem 5.** *Under Hypothesis 7, the false alarm probability of Algorithm 1MR is*

$$\frac{1}{2^m}\sum_{J\subsetneq[1..m]}\Phi\left(-\sqrt{N\textstyle\sum_{j\notin J}\epsilon_j^2}\right).$$

This is the same result as Theorem 1 of [BCQ04] in view of (3) as expected from the arguments in Sect. 7.1.

**Algorithm 1MC.** The success probability is $\Phi((1-t)\sqrt{N}\epsilon)$ as in Algorithm 1MT. To consider the false alarm probability, we again consider the same vector-valued random variable as in the case of Algorithm 1MT and Algorithm 1MR. For a wrong key $\boldsymbol{\beta}$ of type $J$, the probability that $\boldsymbol{\beta}$ is a false alarm is

$$\begin{aligned}
&\Pr_D\left(T^I(\boldsymbol{\beta}, D) \geq T^I(\boldsymbol{\beta}^*, D),\ T^I(\boldsymbol{\beta}, D) \geq tN\epsilon^2\right)\\
&= \Pr_{(X_1,\cdots,X_m)\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left(\textstyle\sum_{j\notin J} X_j \leq 0,\ \sum_{j\in J} X_j - \sum_{j\notin J} X_j \geq tN\epsilon^2\right)\\
&= \Pr_{(Y_1,\cdots,Y_m)\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_m)}\Big(\textstyle\sum_{j\notin J} \epsilon_j Y_j \leq -\sqrt{N}(\sum_{j\notin J}\epsilon_j^2),\\
&\qquad\qquad \textstyle\sum_{j\in J}\epsilon_j Y_j - \sum_{j\notin J}\epsilon_j Y_j \geq \sqrt{N}(t\epsilon^2 - \sum_{j\in J}\epsilon_j^2 + \sum_{j\notin J}\epsilon_j^2\Big)
\end{aligned}$$

under Hypothesis 7. In the last expression $\sum_{j\in J}\epsilon_j Y_j$ and $\sum_{j\notin J}\epsilon_j Y_j$ are two independent random variables that jointly have a 2-variate normal distribution by Proposition 2 which is a rephrasement of Proposition 1.

**Proposition 2.** *Let $\boldsymbol{X} = (X_1,\cdots,X_m)$ be the random variable as in Proposition 1. Let $\boldsymbol{a} \neq \mathbf{0}$ be an $m$-dimensional vector. Then $\langle \boldsymbol{a}, \boldsymbol{X}\rangle$ is a random variable having the normal distribution with mean $\langle \boldsymbol{a}, \boldsymbol{\mu}\rangle$ and standard deviation $|\boldsymbol{\sigma}^T\boldsymbol{a}|$.*

So the false probability for wrong keys of each type is the probability that a vector-valued random variable having a 2-variate normal distribution takes a value in the intersection of two half spaces, one coming from the threshold condition and the other coming from the rank condition. Such value can be easily computed numerically or by simulation.

### 3.3.2 Algorithm 2 Style Attacks

To estimate the false alarm probabilities for Algorithm 2 style attacks, we classify the wrong keys as follows: Let $J_O$ and $J_I$ be subsets of $[1..m]$. For $J_O \subsetneq [1..m]$, a candidate outer key $\boldsymbol{\kappa}$ is called a *wrong key of type* $J_O$ if

- $\kappa_j = \kappa_j^*$ for $j \in J_O$, and
- $\kappa_j \neq \kappa_j^*$ for $j \notin J_O$.

For $(J_O, J_I) \neq ([1..m],[1..m])$, a pair $(\boldsymbol{\kappa},\boldsymbol{\beta})$ of a candidate outer key and an $m$-bit value is called a *wrong key of type* $(J_O, J_I)$ if

- $\kappa_j = \kappa_j^*$ for $j \in J_O$,
- $\kappa_j \neq \kappa_j^*$ for $j \notin J_O$,
- $\beta_j = \beta_j^*$ for $j \in J_I$, and
- $\beta_j \neq \beta_j^*$ for $j \notin J_I$.

We denote the set of all wrong keys of type $J_O$ and the set of all wrong keys of type $(J_O, J_I)$ by

$$\mathcal{W}(J_O) \text{ and } \mathcal{W}(J_O, J_I),$$

respectively. We let $\mathcal{W}([1..m]) := \{\boldsymbol{\kappa}^*\}$ and $\mathcal{W}([1..m],[1..m]) := \{(\boldsymbol{\kappa}^*,\boldsymbol{\beta}^*)\}$ for completeness. We remind that we let

$$T(\boldsymbol{\kappa},\boldsymbol{\beta},D) = \sum_{j=1}^{m}(-1)^{\beta_j}\epsilon_j\tau_j(\kappa_j, D)$$

for each data $D$ of size $N$, outer key $\boldsymbol{\kappa}$, $m$-bit value $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_m)$. Let $D$ be the available data of size $N$. For each type $J_O$, letting $(\boldsymbol{\kappa}, D)$ vary with $\boldsymbol{\kappa} \in \mathcal{W}(J_O)$, we can regard

$$((-1)^{\beta_1^*}\epsilon_1\tau_1(\kappa_1, D), \cdots, (-1)^{\beta_m^*}\epsilon_m\tau_m(\kappa_m, D))$$

as vector-valued random variable that we denote by $\mathbf{X}_{J_O}$. For Algorithm 2 style attacks, we presume an extension of both Hypothesis 3 and 4, also considering the independence of right key statistics and wrong key statistics for different trails, noting that $N \ll 2^n$. Specifically, we presume

**Hypothesis 8.** *For each $J_O$, $\mathbf{X}_{J_O}$ has the m-variate normal distribution with mean*

$$(\mu_1, \cdots, \mu_m)$$

*and covariance matrix*

$$\mathbf{diag}(N\epsilon_1^2, \cdots, N\epsilon_m^2),$$

*where $\mu_j = N\epsilon_j^2$ for $j \in J_O$ and $\mu_j = 0$ for $j \notin J_O$.*

We also presume a stronger hypothesis which further assumes the independence of wrong key statistics and right key statistics for each trail. For each $J_O$, let $j_1, \cdots, j_u$ be the elements in $[1..m] \setminus J_O$. Varying $(\boldsymbol{\kappa}, D)$ with $\boldsymbol{\kappa} \in \mathcal{W}(J_O)$, we can regard

$$\left((-1)^{\beta_1^*}\epsilon_1\tau_1(\kappa_1^*, D), \cdots, (-1)^{\beta_m^*}\epsilon_m\tau_m(\kappa_m^*, D), \epsilon_{j_1}\tau_{j_1}(\kappa_{j_1}, D), \cdots, \epsilon_{j_u}\tau_{j_u}(\kappa_{j_u}, D)\right)$$

as vector-valued random variable that we denote by $\tilde{\mathbf{X}}_{J_O}$. Thus $\tilde{\mathbf{X}}_{[1..m]} = \mathbf{X}_{[1..m]}$. The stronger hypothesis we presume is

**Hypothesis 9.** *For $J_O = \{j_1, \cdots, j_u\}$ or $J_O = \emptyset$ (in which case $u = 0$), $\tilde{\mathbf{X}}_{J_O}$ has the $(m+u)$-variate multivariate normal distribution with mean*

$$(\mu_1, \cdots, \mu_{m+u})$$

*and covariance matrix*

$$\mathbf{diag}(\sigma_1^2, \cdots, \sigma_{m+u}^2),$$

*where $(\mu_j, \sigma_j^2) = (N\epsilon_j^2, N\epsilon_j^2)$ for $j \in [1..m]$ and $(\mu_{m+l}, \sigma_{m+l}^2) = (0, N\epsilon_{j_l}^2)$ for $j = 1, \cdots, u$.*

We will denote the distribution of $\mathbf{X}_{J_O}$ and $\tilde{\mathbf{X}}_{J_O}$ by $\mathcal{D}_{J_O}$ and $\tilde{\mathcal{D}}_{J_O}$, respectively, for each $J_O$.

**Algorithm 2MT.** The success probability of the attack is

$$\mathrm{Pr}_{(X_1, \cdots, X_m) \sim \mathcal{D}_{[1..m]}}\left(\sum\nolimits_j X_j \geq tN\epsilon^2\right)$$

which is equal to $\Phi((1-t)\sqrt{N}\epsilon)$ under Hypothesis 8 by Proposition 1. To consider the false alarm probability, we consider the false alarm probability for each wrong key type. Let $J_O = \{j_1, \cdots, j_u\}$ or $J_O = \emptyset$ (in which case $u = 0$). Under Hypothesis 9, the false alarm probability $p_{\mathrm{fa}}^{T,(J_O,J_I)}(t)$ for wrong keys $(\boldsymbol{\kappa}, \boldsymbol{\beta})$ of type $(J_O, J_I)$ is

$$\mathrm{Pr}_{(X_1, \cdots, X_{m+u}) \sim \tilde{\mathcal{D}}_{J_O}}\left(\sum_{j \in J_O \cap J_I} X_j - \sum_{j \in J_O \setminus J_I} X_j + \sum_{l=1}^{u}(-1)^{\beta_{j_l}} X_{m+l} \geq tN\epsilon^2\right),$$

which equals

$$\Phi\left(\frac{\sqrt{N}}{\epsilon}\left(\sum_{j \in J_O \cap J_I} \epsilon_j^2 - \sum_{j \in J_O \setminus J_I} \epsilon_j^2 - t\epsilon^2\right)\right)$$

by Proposition 1. Thus we have

**Theorem 6.** *Under Hypothesis 9, the false alarm probability $p_{\mathrm{fa}}^T(t)$ of Algorithm 2MT is*

$$\sum_{(J_O,J_I):\text{wrong}} \frac{|\mathcal{W}(J_O)|}{2^{k_O+m}} \Phi\left(\frac{\sqrt{N}}{\epsilon}\left(\sum_{j\in J_O\cap J_I} \epsilon_j^2 - \sum_{j\in J_O\setminus J_I} \epsilon_j^2 - t\epsilon^2\right)\right).$$

We call the wrong key type $(J_O, J_I)$ with $J_O = \emptyset$ the *major* (wrong) key types. All the component outer keys of the wrong keys of the major types are wrong. Note that wrong keys of the major types occupy the great majority of the set of the false alarms and $p_{\mathrm{fa}}^T(t)$ is approximated by the summation of the terms over $(J_O, J_I)$ with $J_O = \emptyset$, which is equal to

$$|\mathcal{W}(J_\emptyset)|\Phi(-t\sqrt{N}\epsilon)/2^{k_O} \approx \Phi(-t\sqrt{N}\epsilon)$$

in many cases. This is mostly the case when each $k_O^j$ is not very small or $N\epsilon^2$ is not very large.

**Algorithm 2MR.** The success probability is 1. To estimate the false alarm probability, we consider the false alarm probability for each wrong key type. Let $(J_O, J_I)$ be a wrong key type. Let $j_1, \cdots, j_u$ be elements of $[0..m] \setminus J_O$. The false alarm probability $p_{\mathrm{fa}}^{R,(J_O,J_I)}$ for wrong keys $(\boldsymbol{\kappa}, \boldsymbol{\beta})$ of type $(J_O, J_I)$ is

$$\begin{aligned}&\Pr_{D,\boldsymbol{\kappa}\in\mathcal{W}(J_O)}\left(T(\boldsymbol{\kappa},\boldsymbol{\beta},D) \geq T(\boldsymbol{\kappa}^*,\boldsymbol{\beta}^*,D)\right)\\ &= \Pr_{\tilde{\mathbf{X}}_{J_O}}\left(-2\textstyle\sum_{j\in J_O\setminus J_I}(-1)^{\beta_j^*}X_j - \sum_{j:j\leq m, j\notin J_O} X_j + \sum_{l=1}^{u}(-1)^{\beta_{j_l}^*}X_{m+l} \geq 0\right)\end{aligned}$$

which is equal to

$$\Phi\left(-\left(N\left(\textstyle\sum_{j\in J_O\setminus J_I}\epsilon_j^2 + \frac{1}{2}\sum_{j\in[1..m]\setminus J_O}\epsilon_j^2\right)\right)^{1/2}\right)$$

by Proposition 1 under Hypothesis 9. Considering the false alarm probabilities for all wrong key types, we have the following result:

**Theorem 7.** *Under Hypothesis 9, the false alarm probability $p_{\mathrm{fa}}^R$ of Algorithm 2MR is*

$$\sum_{(J_O,J_I):\text{wrong}} \frac{|\mathcal{W}(J_O)|}{2^{k_O+m}} \Phi\left(-\left(N\left(\textstyle\sum_{j\in J_O\setminus J_I}\epsilon_j^2 + \frac{1}{2}\sum_{j\in[1..m]\setminus J_O}\epsilon_j^2\right)\right)^{1/2}\right). \quad (7)$$

The false alarm probability $p_{\mathrm{fa}}^R$ is approximated by the summation of the terms over $(J_O, J_I)$ with $J_O = \emptyset$, which is equal to

$$|\mathcal{W}(\emptyset)|\Phi\left(-\sqrt{\frac{N}{2}}\epsilon\right)/2^{k_O} \approx \Phi\left(-\sqrt{\frac{N}{2}}\epsilon\right) \quad (8)$$

in many cases. But the difference between the estimate of the advantage obtained from (7) and the approximated advantage obtained from (8) ignoring the wrong keys other than of the major types can be large in some cases. For example, assume that we proceed with Algorithm 2MR using $m = 4$ linear trails. Assume also that $\kappa_j$'s does not have any bits in common with each other, $k_O^j = 6$ for each $j$, and $|\epsilon_1| = |\epsilon_2| = |\epsilon_3| = |\epsilon_4|$. Then the advantage $-\log_2(p_{\mathrm{fa}}^B) - 1$ and the approximate gain $-\log_2(\Phi(-\sqrt{\frac{N}{2}}\epsilon)) - 1$ are 23.68 and 26.04, respectively when $N\epsilon^2 = 64$. The difference gets larger as $N$ increases.

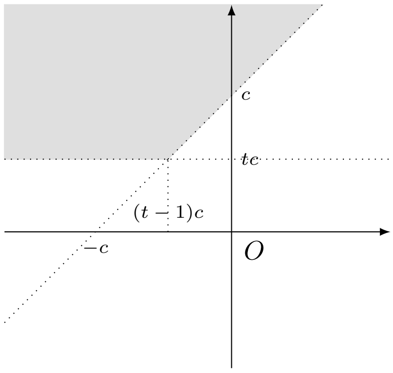

**Figure 2:** Region $\mathcal{R}_{c,t}$ when $c > 0$, $0 < t < 1$

**Algorithm 2MC.** The success probability of the attack is $\Phi((1-t)\sqrt{N}\epsilon)$ as for Algorithm 2MT. To estimate the false alarm probability, for each wrong key type $(J_O, J_I)$, we need to compute the false alarm probability $p_{\text{fa}}^{C,(J_O,J_I)}(t)$ that is equal to

$$\Pr_{D,\boldsymbol{\kappa}\in\mathcal{W}(J_O)}\left(T(\boldsymbol{\kappa},\boldsymbol{\beta},D) \geq T(\boldsymbol{\kappa}^*,\boldsymbol{\beta}^*,D),\ T(\boldsymbol{\kappa},\boldsymbol{\beta},D) \geq tN\epsilon^2\right).$$

We can compute $p_{\text{fa}}^{C,(J_O,J_I)}(t)$ numerically or by simulation for each $(J_O, J_I)$ since it is the probability that the random variable $\tilde{\mathbf{X}}_{J_O}$ having a multivariate normal distribution lies in the intersection of two half spaces, one coming from the threshold condition and the other from the ranking condition. Actually, we can get the probability as the probability that a vector-valued random variable having a 4-variate normal distribution takes a value in the intersection of two half spaces: Let

$$\begin{aligned} U &= \textstyle\sum_{j\in J_O\cap J_I} X_j, \\ V &= \textstyle\sum_{j\in J_O\setminus J_I} X_j, \\ W &= \textstyle\sum_{l=1}^{u} (-1)^{\beta^*_{j_l}} X_{m+l}, \\ Z &= \textstyle\sum_{j\in[1..m]\setminus J_O} X_j. \end{aligned}$$

Then $U, V, W, Z$ are independent, $U \sim \mathcal{N}(\mu_U, \sigma_U^2)$, $V \sim \mathcal{N}(\mu_V, \sigma_V^2)$, $W \sim \mathcal{N}(\mu_W, \sigma_W^2)$, $Z \sim \mathcal{N}(\mu_Z, \sigma_Z^2)$, for some $\mu_U, \sigma_V, \cdots, \mu_Z, \sigma_Z$, that are easily computable in terms of $J_O$, $J_I$, $N$ and $\epsilon_j$'s by Proposition 2. Then

$$p_{\text{fa}}^{C,(J_O,J_I)}(t) = \Pr_{(U,V,W,Z)}(-2V + W - Z \geq 0, U - V + W \geq tN\epsilon^2).$$

For the wrong key types with $J_O = \emptyset$, we can get a simple expression for $p_{\text{fa}}^{C,(J_O,J_I)}(t)$ : In this case

$$\begin{aligned} &\Pr_{D,\boldsymbol{\kappa}\in\mathcal{W}(J_O)}\left(T(\boldsymbol{\kappa},\boldsymbol{\beta},D) \geq T(\boldsymbol{\kappa}^*,\boldsymbol{\beta}^*,D),\ T(\boldsymbol{\kappa},\boldsymbol{\beta},D) \geq tN\epsilon^2\right) \\ &= \Pr_{\tilde{\mathbf{X}}_{J_O}}\left(\textstyle\sum_j (-1)^{\beta_j} X_{m+j} \geq \sum_j X_j,\ \sum_j (-1)^{\beta_j} X_{m+j} \geq tN\epsilon^2\right). \end{aligned}$$

Letting $U = \sum_j (-1)^{\beta_j} X_{m+j}$ and $V = \sum_j X_j$, the probability becomes

$$\Pr_{(U,V)\sim\mathcal{N}((0,N\epsilon^2),(N\epsilon^2)\boldsymbol{I}_2)}(U \geq V, U \geq tN\epsilon^2) = \int_{\mathcal{R}_{\sqrt{N}\epsilon,t}} \phi(0,1;x)\phi(0,1;y)\,dx\,dy,$$

where $\mathcal{R}_{c,t}$ is the region $\{(x,y) \in \mathbb{R}^2 : y \geq x + c, y \geq tc\}$(cf. Fig. 2). So we have the following result:

**Theorem 8.** *Under Hypothesis 9, the false alarm probability $p^C_{\mathrm{fa}}(t)$ of Algorithm 2MC satisfies*

$$p^C(t) < p^C_{\mathrm{fa}}(t) < p^C(t) + \sum_{\substack{(J_O,J_I):\text{wrong}\\ \text{with } J_O\neq\emptyset}} \frac{|\mathcal{W}(J_O)|}{2^{k_O+m}} \min\left(p^{T,(J_O,J_I)}_{\mathrm{fa}}(t), p^{R,(J_O,J_I)}_{\mathrm{fa}}\right),$$

*where* $p^C(t) := \frac{|\mathcal{W}(\emptyset)|}{2^{k_O}} \int_{\mathcal{R}_{\sqrt{N}\epsilon},t} \phi(0,1;x)\phi(0,1;y)\,dx\,dy$.

Note that $p^C_{\mathrm{fa}}(t) \approx p^C(t) \approx \Pr_{\boldsymbol{X}\sim\mathcal{N}(\boldsymbol{0},\boldsymbol{I}_2)}(\boldsymbol{X} \in \mathcal{R}_{\sqrt{N}\epsilon,t})$ in many cases.

**Relation between the advantage and the false alarm probability.** According to Definition 1, the advantage of Algorithm 2MT with success probability $p_S(t) = \Phi((1-t)\sqrt{N}\epsilon)$ is $-\log_2(p^T_{\mathrm{fa}}(t))$. But the advantage of Algorithm 2MR with the success probability of 1 is $-\log_2(p^R_{\mathrm{fa}}) - 1$ and that of Algorithm 2MC with the success probability $p_S(t)$ is $-\log_2(p^C_{\mathrm{fa}}(t)) - 1$.

## 3.4 The Linear Statistic

### 3.4.1 Geometric Description

To illustrate the linear statistic defined as (6), we consider the Algorithm 2 style attack with $m = 2$. The distribution of the statistic vectors $(\tau_1(\kappa_1, D)/N, \tau_2(\kappa_2, D)/N)$ for varying $\boldsymbol{\kappa}$, $D$ are as follows under Hypothesis 8:

- When $\boldsymbol{\kappa}$ is fixed at the correct outer key $\boldsymbol{\kappa}^*$ and $D$ varies, the distribution of the statistic vector has a 2-variate normal distribution with mean $C_{0,0} = ((-1)^{\beta^*_1}\epsilon_1, (-1)^{\beta^*_2}\epsilon_2)$.
- When $(\boldsymbol{\kappa}, D)$ varies with $\kappa_1 = \kappa^*_1$ and $\kappa_2 \neq \kappa^*_2$, the distribution has mean $C_{0,1} = ((-1)^{\beta^*_1}\epsilon_1, 0)$.
- When $(\boldsymbol{\kappa}, D)$ varies with $\kappa_1 \neq \kappa^*_1$ and $\kappa_2 = \kappa^*_2$, the distribution has mean $C_{1,0} = (0, (-1)^{\beta^*_2}\epsilon_2)$.
- When $(\boldsymbol{\kappa}, D)$ varies with $\kappa_1 \neq \kappa^*_1$ and $\kappa_2 \neq \kappa^*_2$, the distribution has mean $C_{1,1} = (0, 0)$.

Each of the 4 distributions has the same covariance matrix $\sqrt{1/N}\boldsymbol{I}_2$. In previous works, the second and third distributions were ignored. For the time being, let us assume that $(-1)^{\beta^*_1}\epsilon_1 > 0$ and $(-1)^{\beta^*_2}\epsilon_2 > 0$. (Other cases can be considered in the same way.)

The red line in the left subfigure of Fig. 3 depicts the set of vectors $(x_1, x_2) \in \mathbb{R}^2$ satisfying

$$(-1)^{\beta^*_1}\epsilon_1 x_1 + (-1)^{\beta^*_2}\epsilon_2 x_2 = t\epsilon^2.$$

Each of the black lines depict the set of vectors satisfying

$$(-1)^{\beta_1}\epsilon_1 x_1 + (-1)^{\beta_2}\epsilon_2 x_2 = t\epsilon^2$$

for some $\boldsymbol{\beta} = (\beta_1, \beta_2)$ with $\beta_1 \neq \beta^*_1$ or $\beta_2 \neq \beta^*_2$. So for the correct $\boldsymbol{\beta}$, we can detect the correct outer key using the threshold condition probabilistically. For the smaller $t$, the success probability, or the probability of detection is larger. The probability that $\boldsymbol{\kappa}$ with $\kappa_1 \neq \kappa^*_1$ and $\kappa_2 \neq \kappa^*_2$ satisfies the threshold condition is the same regardless of $\boldsymbol{\beta}$ due to the symmetry of the distribution of the statistic vectors for such $\boldsymbol{\kappa}$'s with varying $D$. Such probability gets smaller as $t$ gets larger. The blue lines in the right subfigure depicts the threshold lines for different threshold parameters $t$.

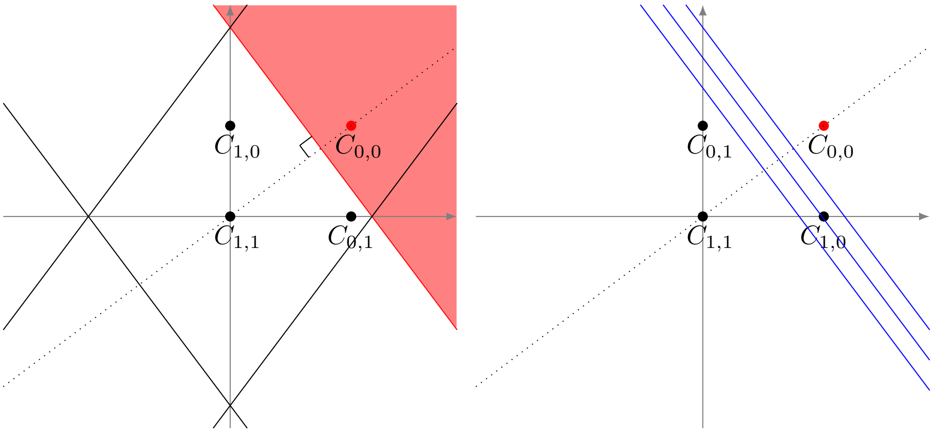

**Figure 3:** Centers of statistic vectors and the threshold lines

### 3.4.2 Relation with the LLR statistic

In this section we will show that when we use dominant and statistically independent trails, our statistic is very close to the LLR statistic up to a constant when the data size $N$ is $O(\epsilon^{-2})$. This implies that we can get similar attacks with the LLR statistic as with our statistic such that the success probabilities and the advantages are almost identical. For real numbers $c_1, \cdots, c_m$ such that $-1 \le c_j \le 1$, let $\boldsymbol{B}(c_1, \cdots, c_m)$ denote the probability distribution on $\{0,1\}^m$ with the pdf $p$ defined by

$$p(x_1, \ldots, x_m) = \prod_{j=1}^{m} \frac{1 + (-1)^{x_j} c_j}{2}$$

for each $(x_1, \ldots, x_m) \in \{0,1\}^m$. So $\boldsymbol{B}(c_1, \cdots, c_m)$ is the joint distribution of $m$ independent binary random variables $\boldsymbol{X}_j$'s such that $\Pr(\boldsymbol{X}_j = 0) = (1 + c_j)/2$ for each $j = 1, \cdots, m$. We consider the Algorithm 2 style attacks described in Sect. 3.2. Let $D$ be a data of size $N$. Let $S_{\boldsymbol{\kappa},D}$ be the multiset that consists of elements of $\{0,1\}^m$ obtained by the outer round computations for the trails using $D$ and $\boldsymbol{\kappa}$. Let $\overline{S}_{\boldsymbol{\kappa}}$ be such a multiset for the full codebook and let $\overline{\mathcal{D}}_{\boldsymbol{\kappa}}$ be the empirical probability distribution on $\{0,1\}^m$ obtained from $\overline{S}_{\boldsymbol{\kappa}}$ for each $\boldsymbol{\kappa}$. For each $j$, let $S^j_{\boldsymbol{\kappa},D}$ be the multiset that consists of elements of $\{0,1\}$ obtained by the outer round computations for the $j$-th trail using $D$ and $\boldsymbol{\kappa}$. By the dominance and the independence of the trails, $\overline{\mathcal{D}}_{\boldsymbol{\kappa}}$ is assumed to be the joint distribution of independent binary random variables for each $\boldsymbol{\kappa}$ and, in particular, $\overline{\mathcal{D}}_{\boldsymbol{\kappa}^*}$ is assumed to be $\boldsymbol{B}((-1)^{\beta_1^*}\epsilon_1, \cdots, (-1)^{\beta_m^*}\epsilon_m)$. Now the LLR statistic described in [HCN19] is

$$\Lambda(\boldsymbol{\kappa}, \boldsymbol{\beta}, D) := LLR(S_{\boldsymbol{\kappa},D}, \boldsymbol{B}((-1)^{\beta_1}\epsilon_1, \cdots, (-1)^{\beta_m}\epsilon_m), \boldsymbol{B}(0, \cdots, 0)).$$

Since LLR statistic is separable as shown in Appendix, Sect. D,

$$\begin{aligned}
&\Lambda(\boldsymbol{\kappa}, \boldsymbol{\beta}, D) = \sum\nolimits_j LLR(S^j_{\boldsymbol{\kappa},D}, \boldsymbol{B}((-1)^{\beta_j}\epsilon_j), \boldsymbol{B}(0)) \\
&= N\left(\sum\nolimits_j \left(\tfrac{1+\tau_j(\kappa_j,D)/N}{2}\log(1 + (-1)^{\beta_j}\epsilon_j) + \tfrac{1-\tau_j(\kappa_j,D)/N}{2}\log(1 - (-1)^{\beta_j}\epsilon_j\right)\right) \\
&= N\sum\nolimits_j \left((-1)^{\beta_j}\epsilon_j \tfrac{\tau_j(\kappa_j,D)}{N} - \tfrac{\epsilon_j^2}{2} + O(\epsilon_j^3)\right) \\
&= T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D) - \tfrac{N\epsilon^2}{2} + O(|\epsilon|)
\end{aligned}$$

So the threshold condition $T(\boldsymbol{\kappa}, \boldsymbol{\beta}, D) \ge tN\epsilon^2$ is about the same condition as $\Lambda(\boldsymbol{\kappa}, \boldsymbol{\beta}, D) \ge (t - \frac{1}{2})N\epsilon^2$, for example. Note that the Algorithm 2 style attack using LLR static as described in [HCN19] provides approximate estimates for the advantage in a different setting with the assumption that all the wrong key distributions are the same.

# 4 Matsui's Algorithms Revisited

In this section, we briefly reformulate classical Matsui's Algorithms that use a single linear trail. All the results in this section can be regarded as the restatement of the results in the preceding section, with the number $m$ of used trails set to 1. But we present the results in this separate section since the estimates of advantage and success probability are simple and comparable with that of existing formulations summarized in [BT13]. We also provide a self-contained proof in the Appendix. *Throughout this section, we assume that the $k$-bit key $K^*$ to recover and its corresponding long key* $\mathrm{rk}^*$ *as well as the long key cipher $\tilde{E}$ and the block cipher $E$ with $n$-bit blocks are fixed.* So we also drop the cipher $E$, the key $K^*$ and the trail $\Gamma$ from the notations. We also assume that each linear trail $\Gamma$ used in the attack is dominant in its linear hull, or $|C(\Gamma)| \approx |C_H(\Gamma)|$. We assume further that the size $N$ of the data is very small compared to $2^n$ as in Sect. 3.

## 4.1 Matsui's Algorithm 1

### 4.1.1 Description

Assume that we have a dominant linear trail $\Gamma$ for the full cipher $E$. We try to restore the parity bit $\sum_{i=0}^{R-1} \langle \Gamma_i, rk_i^* \rangle$ using a data $D = \{(P_i, C_i) : i = 1, \ldots, N\}$ of size $N$. For that we compute $\tau^I(\Gamma, D)$ and then determine the parity bit to be 0 or 1 according as $\epsilon\tau^I(\Gamma, D) \geq 0$ or not.

### 4.1.2 Success Probability

Let us consider the attack that uses a data $D$ of size $N$. If we let $D$ vary, we can regard $(-1)^{\beta^*}\epsilon\tau^I(D)$ as a random variable. For the attack, we presume Hypothesis 10 that follows from Hypothesis 1 that is regarded to be valid with $N \ll 2^n$.

**Hypothesis 10** (Right Key Hypothesis for Algorithm 1)**.** *The random variable* $(-1)^{\beta^*}\epsilon\tau^I(D)$ *has the normal distribution with mean $N\epsilon^2$ and variance $N\epsilon^2$.*

Under this hypothesis, the success probability of the attack is $\Phi(\sqrt{N}|\epsilon|)$ (cf. Lemma 2 in [Mat93]).

## 4.2 Matsui's Algorithm 2

Assume that we have an $r$-round linear trail $\Gamma = [\Gamma_s, \cdots, \Gamma_{s+r}]$ with the correlation $\epsilon = C(\Gamma)$ for a $R$-round block cipher $E$ with $R \geq r + s$. We try to add outer rounds to $\Gamma$ and recover some of the outer round key bits together with the parity bit using $D = \{(P_i, C_i) : i = 1, \ldots, N\}$ of size $N$ with $C_i = E(K^*, P_i)$ for each $i$. Let $\beta^*$ be the parity bit and let $\beta$ denote the candidate value for $\beta^*$. Let $\kappa$ denote the candidate for the correct outer key $\kappa^*$. We write the statistic $(-1)^\beta\epsilon\tau(\kappa, D)$ as $T(\kappa, \beta, D)$ for each $(\kappa, \beta, D)$. Note that $\kappa^*$ and $\beta^*$ are fixed since $K^*$ is. Let $D$ be the available data of size $N$.

### 4.2.1 Description

**Algorithm 2T.** Compute the statistic $T(\kappa, \beta, D)$ for each $(\kappa, \beta)$. Then check only $(\kappa, \beta)$'s that satisfy $T(\kappa, \beta, D) \geq tN\epsilon^2$ in the trial encryption.

**Algorithm 2R.** After getting the list $[T(\kappa,\beta,D),(\kappa,\beta)]$, sort the list of $(\kappa,\beta)$'s according to the statistic in the descending order. Then try all the $(\kappa,\beta)$'s according to the rank until we find the correct one.

**Algorithm 2C.** After computing the statistic $T(\kappa,\beta,D)$ for each $(\kappa,\beta)$, pick out $(\kappa,\beta)$'s that satisfy the condition $T(\kappa,\beta,D)\geq tN\epsilon^2$. Sort the list of the selected $(\kappa,\beta)$'s according to the statistic in the descending order. Then try all the $(\kappa,\beta)$'s in the sorted list according to the order until we find the correct one.

#### 4.2.2 Success Probability and Advantage

Letting $D$ vary, we can regard $(-1)^{\beta^*}\epsilon\tau(\kappa^*,D)$ as a random variable. Also, letting $(\kappa,D)$ vary with $\kappa\neq\kappa^*$. we can regard $\epsilon\tau(\kappa,D)$ as a random variable. Furthermore, letting $(\kappa,D)$ vary with $\kappa\neq\kappa^*$. we can regard $((-1)^{\beta^*}\epsilon\tau(\kappa^*,D),\epsilon\tau(\kappa,D))$ as a vector-valued random variable. We presume Hypotheses 11 and 12 that follow from Hypotheses 1 and 2.

**Hypothesis 11** (Right Key Hypothesis for Algorithm 2)**.** *$(-1)^{\beta^*}\epsilon\tau(\kappa^*,D)$ has the normal distribution with mean $N\epsilon^2$ and variance $N\epsilon^2$.*

**Hypothesis 12** (Wrong Key Hypothesis for Algorithm 2T)**.** *$\epsilon\tau(\kappa,D)$ has the normal distribution with mean $0$ and variance $N\epsilon^2$ as $D$ and the wrong key $\kappa$ varies.*

We just need to presume Hypotheses 11 and 12 for Algorithm 2T. But Algorithm 2R and 2C require the following stronger hypothesis which additionally implies the independence of the distributions of the undersampled correlations for wrong keys and right keys:

**Hypothesis 13.** *The random variable $((-1)^{\beta^*}\epsilon\tau(\kappa^*,D),\epsilon\tau(\kappa,D))$ has the 2-variate normal distribution with mean $(N\epsilon^2,0)$ and covariance matrix $(N\epsilon^2)\boldsymbol{I}_2$.*

We let $\mathcal{K}$ be the set of the candidate outer keys and let $k_O=\log_2|\mathcal{K}|$.

**Theorem 9.** *Under Hypothesis 11, the success probability of Algorithm 2T with the threshold parameter $t$, data size $N$ is $\Phi((1-t)\sqrt{N}|\epsilon|)$. Under Hypothesis 12, the false alarm probability is*

$$\frac{2^{k_O}-1}{2^{k_O}}\Phi(-t\sqrt{N}|\epsilon|)+\frac{1}{2^{k_O+1}}\Phi((-1-t)\sqrt{N}|\epsilon|).$$

Note that the false alarm probability is very close to and not larger than $\Phi(-t\sqrt{N}|\epsilon|)$ for not too small $k_O$. With this estimate, rather interestingly, the advantage of the attack is the same as the one for the rank based attack by A. Selçuk given as (1).

**Theorem 10.** *The success probability of Algorithm 2R is 1. Under Hypothesis 13, its false alarm probability is*

$$\frac{2^{k_O}-1}{2^{k_O}}\Phi\left(-\sqrt{\frac{N}{2}}|\epsilon|\right)+\frac{\Phi(-\sqrt{N}|\epsilon|)}{2^{k_O+1}}.$$

Note that $p_{\mathrm{fa}}\approx\Phi(-\sqrt{N/2}|\epsilon|)$ when $k_O$ is not too small.

**Theorem 11.** *Under Hypothesis 13, the success probability of Algorithm 2C with the threshold parameter $t$, data size $N$ is $\Phi((1-t)\sqrt{N}|\epsilon|)$ and its false alarm probability is*

$$\frac{2^{k_O}-1}{2^{k_O}}\int_{\mathcal{R}_{\sqrt{N}|\epsilon|,t}}\phi(0,1;x)\phi(0,1;y)\,dx\,dy+\frac{1}{2^{k_O+1}}q(\sqrt{N}|\epsilon|,t),$$

*where $q$ is a function on $\mathbb{R}^2$ defined by $q(x,t)=\Phi(-x)$ for $t\leq 0$ and $q(x,t)=\Phi(-(1+t)x)$ for $t\geq 0$.*

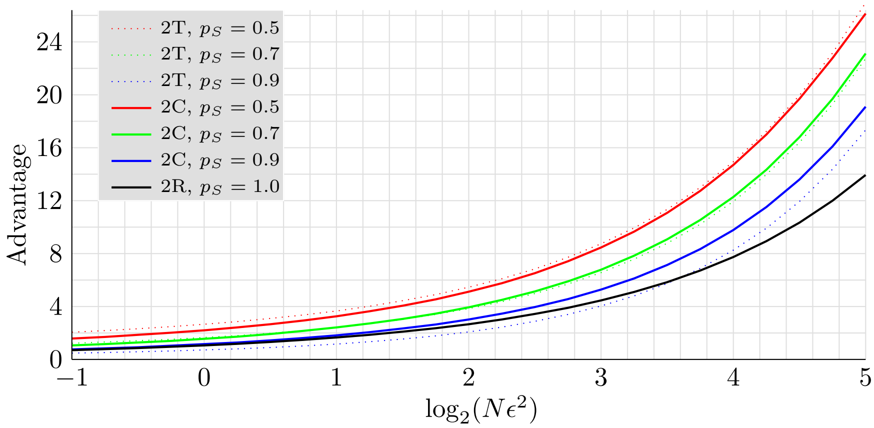

**Figure 4:** Theoretical advantages of Algorithms 2T, 2R, and 2C

Note that $q(\sqrt{N}|\epsilon|, t) \leq \Phi(-\sqrt{N}|\epsilon|)$ so that $p_{\text{fa}}(t) \approx \Pr_{X\sim\mathcal{N}(\mathbf{0},\boldsymbol{I}_2)}(X \in \mathcal{R}_{\sqrt{N}|\epsilon|,t})$ in many cases. The advantages of the attack for various values of $\sqrt{N}|\epsilon|$ and success probabilities are as in Fig. 4 in case $k_O$ is not too small. Note that when the aimed success probability is close to 1, the advantage of Algorithm 2C over Algorithm 2T, which has the same advantage as that of the linear attack in [Sel08], is quite visible, especially when $\sqrt{N}|\epsilon|$ is large. For example, when $\sqrt{N}|\epsilon| = 4$ and $p_S = 0.9$, the advantages of Algorithm 2T and Algorithm 2C are 8.25 and 9.78, respectively.

## 5 Generalization

In this section, we provide a method of linear attack that uses several linear trails that are not necessarily dominant. With certain constraints on the linear trails, we will presume that vector-valued random variables consisting of component statistics for wrong keys and right keys have multivariate normal distributions as in recent works [BV17, BTV18]. Under such a hypothesis, we provide a method of linear attack with the estimates on success probability and false alarm probability in Sect. 3 and Sect. 5.3. Let $E$ be an $R$-round iterated cipher and $0 \leq s < s + r \leq R$. Assume that we have $m \geq 1$ linear trails $\Gamma^j$'s spanning from the start of the $s$-th round to the end of the $(s + r - 1)$-th round. We impose the following constraints on the linear trails:

- The number $m$ of used trails is small.

- Each trail is close-to-dominant. More specifically, for each $j$, $|C(\Gamma^j)| \geq C_H(\Gamma^j)/2$ and $|C(\Lambda)| \ll |C(\Gamma^j)|$ for all $\Lambda \in \mathcal{H}(\Gamma^j)$ such that $\Lambda \neq \Gamma^j$.

Let $\epsilon_j = C(\Gamma^j)$ for each $j$ and let $\epsilon = (\sum_j \epsilon_j^2)^{1/2}$. For simplicity we assume that the parity bits are independent. We do not fix $K^*$ so that $\kappa^*$ varies depending on $K^*$ in the Algorithm 2 style attacks. But we fix the vector $\boldsymbol{\beta}^*$ of correct parity bits so that we let $K^*$ vary in such a way that its vector of parity bits is equal to $\boldsymbol{\beta}^*$. We let $\mathcal{K}^*$ be the set of all $K^*$'s whose vector of parity bits are equal to $\boldsymbol{\beta}^*$.

Let $\mathbf{1}_m$ be the vector in $\mathbb{R}^m$ such that each of its component is 1. For $J \subset [1..m]$, let $J^c = [1..m] \setminus J$ and let $\boldsymbol{e}^J \in \mathbb{R}^m$ be such that $\boldsymbol{e}^J_j = 1$ for $j \in J$ and $\boldsymbol{e}^J_j = 0$ for $j \notin J$.

## 5.1 Algorithm 1 Style Attacks

### 5.1.1 Prerequisites

Letting the set $D$ of plaintexts of size $N$ and $K \in \mathcal{K}^*$ vary , we can regard

$$\left((-1)^{\beta_1^*}\epsilon_1\tau_1^I(K,D), \cdots, (-1)^{\beta_m^*}\epsilon_m\tau_m^I(K,D)\right)$$

as a vector-valued random variable. For the generalized Algorithm 1 style attacks, we presume

**Hypothesis 14.** *The above vector-valued random variable has the m-variate normal distribution with mean $\boldsymbol{\mu} = (N\epsilon_1^2, \cdots, N\epsilon_j^2)$ and covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{\sigma}\boldsymbol{\sigma}^T$.*

For the attack, we need to know $\boldsymbol{\Sigma}$ in advance.

### 5.1.2 Description

We also have three versions of Algorithm 1 style attacks that are carried out in the same way as the Algorithms 1MT, 1MR, and 1MC described in Sect. 3.1 using the statistic $T^I(K^*, \boldsymbol{\beta}, D) := \sum_j (-1)^{\beta_j}\epsilon_j\tau_j^I(K^*, D)$.

### 5.1.3 Success Probability and Advantage

Let $t$ be the threshold parameter.

**Generalized 1MT.** We use the threshold condition $T^I(K^*, \boldsymbol{\beta}, D) \geq tN\epsilon^2$. The success probability is

$$\frac{1}{(\sqrt{2\pi})^m|\det(\boldsymbol{\sigma})|}\int_{\boldsymbol{x}:\langle \mathbf{1}_m, \boldsymbol{x}\rangle - tN\epsilon^2 \geq 0} e^{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}} d\boldsymbol{x},$$

under Hypothesis 14 which is equal to $\Phi((1-t)N\epsilon^2/|\boldsymbol{\sigma}^T\mathbf{1}_m|)$ by Proposition 1. The false alarm probability for wrong keys of type $J$ is

$$\frac{1}{(\sqrt{2\pi})^m|\det(\boldsymbol{\sigma})|}\int_{\boldsymbol{x}:\langle \boldsymbol{e}^J - \boldsymbol{e}^{J^c}, \boldsymbol{x}\rangle - tN\epsilon^2 \geq 0} e^{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}} d\boldsymbol{x},$$

which is equal to

$$\Phi\left(\frac{N(\sum_{j\in J}\epsilon_j^2 - \sum_{j\notin J}\epsilon_j^2 - t\epsilon^2)}{|\boldsymbol{\sigma}^T(\boldsymbol{e}^J - \boldsymbol{e}^{J^c})|}\right)$$

under the same hypothesis, by Proposition 1. Thus we have

**Theorem 12.** *Under Hypothesis 14, the false alarm probability of the Generalized 1MT is*

$$\frac{1}{2^m}\sum_{J\subsetneq[1..m]}\Phi\left(\frac{N(\sum_{j\in J}\epsilon_j^2 - \sum_{j\notin J}\epsilon_j^2 - t\epsilon^2)}{|\boldsymbol{\sigma}^T(\boldsymbol{e}^J - \boldsymbol{e}^{J^c})|}\right).$$

**Generalized 1MR.** The success probability is 1. Under Hypothesis 14, the false alarm probability for wrong keys of type $J$ is equal to

$$\begin{aligned}
&\Pr_{(X_1,\cdots,X_m)\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\textstyle\sum_{j\in J}X_j - \sum_{j\notin J}X_j \geq \sum_j X_j)\\
&= \Pr_{(X_1,\cdots,X_m)\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\textstyle\sum_{j\notin J}X_j \leq 0)\\
&= \tfrac{|\det(\boldsymbol{\sigma})|}{(\sqrt{2\pi})^m}\textstyle\int_{\boldsymbol{x}:\langle \boldsymbol{e}^{J^c}, \boldsymbol{x}\rangle \leq 0} e^{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}} d\boldsymbol{x},
\end{aligned}$$

which is equal to $\Phi\left(\frac{-N(\sum_{j\notin J}\epsilon_j^2)}{|\boldsymbol{\sigma}^T\boldsymbol{e}^{J^c}|}\right)$ by Proposition 1. Thus we have

**Theorem 13.** *Under Hypothesis 14, the false alarm probability of the Generalized 1MR is*

$$\frac{1}{2^m}\sum_{J\subsetneq[1..m]}\Phi\left(\frac{-N(\sum_{j\notin J}\epsilon_j^2)}{|\boldsymbol{\sigma}^T\boldsymbol{e}^{J^c}|}\right).$$

**Generalized 1MC.** Under Hypothesis 14, the success probability is

$$\Phi((1-t)N\epsilon^2/|\boldsymbol{\sigma}^T\mathbf{1}_m|)$$

with threshold parameter $t$ as for Generalized 1MT. Under the same hypothesis, the false alarm probability of Generalized 1MC for wrong keys of type $J$ is

$$p_{\text{fa}}^J(t)=\Pr_{(X_1,\cdots,X_m)\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}(\sum_{j\notin J}X_j\leq 0,\sum_{j\in J}X_j-\sum_{j\notin J}X_j\geq tN\epsilon^2).$$

So the false alarm probability is

$$\frac{1}{2^m}\sum_{J\subsetneq[1..m]}p_{\text{fa}}^J(t).$$

## 5.2 Algorithm 2 Style Attacks

### 5.2.1 Prerequisites

To generalize the Algorithm 2 style attack presented in Sect. 3, we consider wrong key types again. We remind that we have fixed $\boldsymbol{\beta}^*$, but $K^*$ takes values in $\mathcal{K}^*$ and is not fixed and neither is $\kappa^*$. For each $j$, letting $D$, $K$ and $\kappa_j$ vary with $\kappa_j$ being a wrong key for $K$ regarding $\Gamma^j$, we can regard $\epsilon_j\tau_j(K,\kappa_j,D)$ as a real-valued random variable.

**Hypothesis 15.** *For each $j$, the random variable $\epsilon_j\tau_j(K,\kappa_j,D)$ has a normal distribution $\mathcal{N}(0,\sigma_j^2)$ for some $\sigma_j\approx\sqrt{N}|\epsilon_j|$.*

Let $J_O\subset[1..m]$ be a key type. Let $K^*$ be a key and $\kappa^*$ be the correct outer key for $K^*$. Then $\boldsymbol{\kappa}$ is called an outer key of type $J_O$ for $K^*$ if $\kappa_j^*=\kappa_j$ exactly for $j\in J$. We denote by $\mathcal{W}(J_O,K)$ the set of all outer keys of type $J_O$ for $K$. Letting $(K,D,\boldsymbol{\kappa})$ vary, where $D$ is a set of plaintexts of size $N$, $K$ is a key in $\mathcal{K}^*$ and $\boldsymbol{\kappa}\in\mathcal{W}(J_O,K)$, we can regard

$$\left((-1)^{\beta_1^*}\epsilon_1\tau_1(K,\kappa_1,D),\cdots,(-1)^{\beta_m^*}\epsilon_m\tau_m(K,\kappa_m,D),\epsilon_{j_1}\tau_{j_1}(K,\kappa_{j_1},D),\cdots,\epsilon_{j_u}\tau_{j_u}(K,\kappa_{j_u},D)\right)$$

as a vector-valued random variable that we denote by $\tilde{\mathbf{Y}}_{J_O}$. Let $j_1,\cdots,j_u$ be the elements of $[1..m]\setminus J_O$. The hypothesis we presume is

**Hypothesis 16.** *For each $J_O$, the vector-valued random variable $\tilde{\mathbf{Y}}_{J_O}$ has the $(m+u)$-variate normal distribution with mean $\boldsymbol{\mu}=(\mu_1,\cdots,\mu_{m+u})$ and covariance $\boldsymbol{\Sigma}$, where $\mu_j=N\epsilon_j^2$ for $j\leq m$, $\mu_j=0$ for $j>m$ and*

$$\boldsymbol{\Sigma}=\begin{pmatrix}\boldsymbol{\Sigma}_R & \mathbf{0}^{m\times l}\\ \mathbf{0}^{l\times m} & \boldsymbol{\Sigma}_{W^{J_O}}\end{pmatrix},$$

*where $\boldsymbol{\Sigma}_R=\boldsymbol{\sigma}_R\boldsymbol{\sigma}_R^T$ for an invertible $m\times m$ matrix $\boldsymbol{\sigma}_R$ and $\boldsymbol{\Sigma}_{W^{J_O}}=\mathbf{diag}(\sigma_{j_1}^2,\cdots,\sigma_{j_u}^2)$.*

Here, $\boldsymbol{\Sigma}_R$ is the joint distribution of statistics for the right keys and it does not depend on $J_O$. For the attack, we need to know $\boldsymbol{\Sigma}_R$ and $\sigma_j$'s in advance.

### 5.2.2 Description

Let $t$ be the threshold parameter and let $D$ be the available data of size $N$. We also have three versions of the generalized Algorithm 2 style attacks, called Generalized 2MT, 2MR, and 2MC, that are carried out exactly in the same way as Algorithm 2MT, 2MR, and 2MC, respectively, using the statistic $T(K^*, \boldsymbol{\kappa}, \boldsymbol{\beta}, D) = \sum_j (-1)^{\beta_j} \epsilon_j \tau_j(K^*, \boldsymbol{\kappa}, D)$.

### 5.2.3 Success Probability and Advantage

Let $t$ be the threshold parameter.

**Generalized 2MT.** Under Hypothesis 15, the success probability is $\Phi((1-t)N\epsilon^2/|\boldsymbol{\sigma}_R^T \mathbf{1}_m|)$ with threshold parameter $t$. Let $(J_O, J_I)$ be a wrong key type. Then, under Hypothesis 16, the false alarm probability $p_{\text{fa}}^{T,(J_O,J_I)}(t)$ for wrong keys of type $(J_O, J_I)$ is

$$\Pr_{\tilde{\mathbf{Y}}_{J_O}} \left( \sum_{j \in J_O \cap J_I} Y_j - \sum_{j \in J_O \setminus J_I} Y_j + \sum_l (-1)^{\beta_{j_l}} Y_{m+l} \geq tN\epsilon^2 \right),$$

which is equal to

$$\Phi \left( \frac{N(\sum_{j \in J_O \cap J_I} \epsilon_j^2 - \sum_{j \in J_O \setminus J_I} \epsilon_j^2 - t\epsilon^2)}{\left(|\boldsymbol{\sigma}_R^T(\boldsymbol{e}^{J_O \cap J_I} - \boldsymbol{e}^{J_O \setminus J_I})|^2 + \sum_l \sigma_{j_l}^2\right)^{1/2}} \right)$$

by Proposition 1. The false alarm probability $p_{\text{fa}}^T(t)$ is $\sum_{(J_O,J_I):wrong} \frac{|\mathcal{W}(J_O)|}{2^{k_O+m}} p_{\text{fa}}^{T,(J_O,J_I)}(t)$, where $|\mathcal{W}(J_O)| = |\mathcal{W}(J_O, K)|$ for any $K$.

**Generalized 2MR.** Success probability is 1. Under Hypothesis 16, for a wrong key type $(J_O, J_I)$, the false alarm probability $p_{\text{fa}}^{R,(J_O,J_I)}$ for wrong keys of type $(J_O, J_I)$ is

$$\begin{aligned} &\Pr_{\tilde{\mathbf{Y}}_{J_O}} \left( \textstyle\sum_{j \in J_O \cap J_I} Y_j - \sum_{j \in J_O \setminus J_I} Y_j + \sum_{l=1}^{u} (-1)^{\beta_{j_l}} Y_{m+l} \geq \sum_{j \leq m} Y_j \right) \\ &= \Pr_{\tilde{\mathbf{Y}}_{J_O}} \left( -2 \textstyle\sum_{j \in J_O \setminus J_I} Y_j + \sum_{l=1}^{u} (-1)^{\beta_{j_l}} Y_{m+l} - \sum_{j \leq m, j \notin J_O} Y_j \geq 0 \right), \end{aligned}$$

which is equal to

$$\Phi \left( \frac{-N(2 \sum_{j \in J_O \setminus J_I} \epsilon_j^2 + \sum_{j \leq m, j \notin J} \epsilon_j^2)}{\left(|\boldsymbol{\sigma}_R^T(-2\boldsymbol{e}^{J_O \setminus J_I} - \boldsymbol{e}^{J_O^c})|^2 + (\sum_l \sigma_{j_l})^2\right)^{1/2}} \right)$$

by Proposition 1. The false alarm probability $p_{\text{fa}}^R$ is $\sum_{(J_O,J_I):wrong} \frac{|\mathcal{W}(J_O)|}{2^{k_O+m}} p_{\text{fa}}^{R,(J_O,J_I)}$.

**Generalized 2MC.** Under Hypothesis 15, the success probability is $\Phi((1-t)N\epsilon^2/|\boldsymbol{\sigma}_R^T \mathbf{1}_m|)$ just as in Generalized 2MT. The false alarm probability $p_{\text{fa}}^{C,(J_O,J_I)}(t)$ is

$$\begin{aligned} \Pr_{\tilde{\mathbf{Y}}_{J_O}} \Big( &\textstyle\sum_{j \in J_O \cap J_I} Y_j - \sum_{j \in J_O \setminus J_I} Y_j + \sum_l (-1)^{\beta_{j_l}} Y_{m+l} \geq tN\epsilon^2, \\ &-2 \textstyle\sum_{j \in J_O \setminus J_I} Y_j + (-1)^{\beta_{j_l}} Y_{m+l} - \sum_{j \leq m, j \notin J_O} Y_j \geq 0 \Big). \end{aligned}$$

Under Hypothesis 16, the false alarm probability $p_{\text{fa}}^C(t)$ is

$$\sum_{(J_O,J_I):wrong} \frac{|\mathcal{W}(J_O)|}{2^{k_O+m}} p_{\text{fa}}^{C,(J_O,J_I)}(t)$$

which can by computed by simulation.

### 5.3 Attacks Using Close-to-dominant and Independent Trails

Suppose that we use $m$ significant trails $\Gamma^j$'s that are statistically independent, but not dominant. Assume that we sample data with replacement and that the data size $N$ is negligible compared to $2^n$. For each $j$, let $C_H(\Gamma^j)^2 = C(\Gamma^j)^2 + \rho_j^2$. Then we can postulate Hypothesis 14 and 16, where $\mathbf{\Sigma}_R = \mathbf{diag}((N^2\rho_1^2 + N)\epsilon_1^2, \cdots, (N^2\rho_m^2 + N)\epsilon_m^2)$ in view of Hypothesis 3 and from the statistical independence of the trails. Also we may assume that $\mathbf{\Sigma}_{W^{[1..m]}} = \mathbf{diag}(N\epsilon_1^2, \cdots, N\epsilon_m^2)$. From this, we can get an estimate for the advantage of Algorithm 1 or Algorithm 2 style attack as provided in Sect. 5.1 and 5.2.

## 6 Application to DES

In this section, we apply the attack methods in Sect. 3 to the full DES without the initial and final permutations.

### 6.1 Description of the Attack

We use 4 of the 8 14-round trails used in [BV17]. With the notations in [Mat93, BV17], the trails $\Gamma_j$'are as follows:
- $\Gamma_1 = \gamma_1$ represented as "-DCA-ACD-DCA-A", $\epsilon_1 = C(\Gamma_1) = -2^{-19.75}$
- $\Gamma_2 = \gamma_4$ represented as "-ACD-DCA-ACD-E", $\epsilon_2 = C(\Gamma_2) = -2^{-20.07}$
- $\Gamma_3 = \delta_3$ represented as "A-ACD-DCA-ACD-", $\epsilon_3 = C(\Gamma_3) = -2^{-19.75}$
- $\Gamma_4 = \delta_2$ represented as "E-DCA-ACD-DCA-", $\epsilon_4 = C(\Gamma_4) = -2^{-20.07}$

The trails are considered to be dominant and statistically independent. We try to prepend 1 round and append 1 round simultaneously to the trails. The number of bits for $\kappa_j$'s are 12, 18, 12, and 18, respectively. $\kappa_1$ and $\kappa_2$ have 6 bits in common and so are $\kappa_3$ and $\kappa_4$. $\kappa_1\|\kappa_2$ and $\kappa_3\|\kappa_4$ do not have any bits in common. Thus $\kappa$ has 48 bits. The 4 parity bits are independent. Let $\epsilon = (\sum_{j=1}^4 \epsilon_j^2)^{1/2} = 2^{-18.89}$.
Let $N$ be the size of the available data $D$. We set the aimed success probability $p_S$ and let $t$ be the threshold parameter such that $\Phi((1-t)\sqrt{N}\epsilon) = p_S$. We first compress the data for each trail and get 4 compressed sets of size $2^{13}$, $2^{19}$, $2^{13}$, and $2^{19}$, respectively. We apply FWHT to each of the compressed set to get 4 lists $L_j$'s such that

$$L_j = [((-1)^{\beta_j}\epsilon_j\tau_j(\kappa_j, D), (\kappa_j, \beta_j)) : \kappa_j \in \mathbb{F}_2^{k_O^j}, \beta_j \in \mathbb{F}_2]$$

for each $j = 1, 2, 3, 4$. The numbers of entries in $L_j$ are $2^{13}$, $2^{19}$, $2^{13}$, and $2^{19}$ for $j = 1, 2, 3, 4$, respectively. Since $\kappa_1$ and $\kappa_2$ have 6 bits in common, we combine $L_1$ and $L_2$ to get a list $L_{1,2}$ of size $2^{26}$: $L_{1,2}$ contains

$$(T^{1,2}(\kappa_{1,2}, \beta_1, \beta_2), (\kappa_{1,2}, \beta_1, \beta_2))$$

where $\kappa_{1,2}$ is the 24-bit value obtained from combining $\kappa_1$ with $\kappa_2$ and

$$T^{1,2}(\kappa_{1,2}, \beta_1, \beta_2) = (-1)^{\beta_1}\epsilon_1\tau_1(\kappa_1, \beta_1, D) + (-1)^{\beta_2}\epsilon_2\tau_2(\kappa_2, \beta_2, D)$$

for each $(\kappa_{1,2}, \beta_1, \beta_2)$. After combining the lists, we sort the resulting list $L_{1,2}$ according to the value $T^{1,2}(\kappa_{1,2}, \beta_1, \beta_2)$ to get a new list $L'_{1,2}$. We can apply the efficient counting sort algorithm by suitably rescaling the values $T^{1,2}(\kappa_{1,2}, \beta_1, \beta_2)$. In the same way, we also get a list $L_{3,4}$ and a sorted list $L'_{3,4}$ of size $2^{26}$ from $L_3$ and $L_4$. Using the sorted lists, we get a new list

$$\mathcal{L} = [(T(\kappa, \boldsymbol{\beta}), (\kappa, \boldsymbol{\beta})) : T(\kappa, \boldsymbol{\beta}) \geq tN\epsilon^2]$$

noting that

$$T(\boldsymbol{\kappa}, \boldsymbol{\beta}) = T^{1,2}(\kappa_{1,2}, \beta_1, \beta_2) + T^{3,4}(\kappa_{3,4}, \beta_3, \beta_4).$$

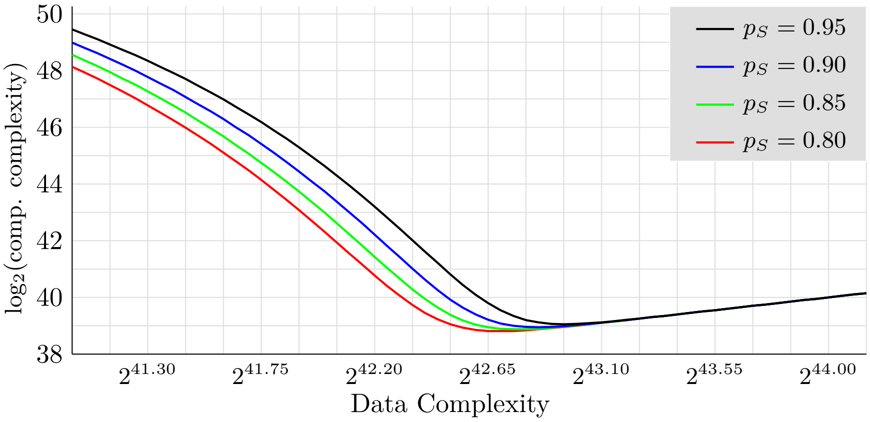

**Figure 5:** Theoretical Complexity for Algorithm 2MC on DES

We have about $2^{52}p^T_{\text{fa}}(t)$ of $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s satisfying $T(\boldsymbol{\kappa}, \boldsymbol{\beta}) \geq tN\epsilon^2$. After getting $\mathcal{L}$, we sort it and try all the $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s in the sorted list by guessing 4 additional key bits. We do not need the final sorting in Algorithm 2MT. Also we can do without the final sorting in Algorithm 2MC with little decrease in its advantage as explained in Sect. E of the Appendix.

## 6.2 Complexity of the Attacks

We consider the average complexity of Algorithm 2MT and 2MC with the aimed success probability $p_S$ and the size of the available data $N$. We assume that $2^{40.00} \leq N \leq 2^{44.05}$. Let $t$ be the threshold parameter such that $\Phi((1-t)\sqrt{N}\epsilon) = p_S$. Let $C_1 = 1/16$ be the computational complexity of 1 encryption round (including the key schedule). We estimate the complexity of the compression step to be $C_1N$ similarly as in [BV17]. Considering the FWHT applied 4 times, we estimate the complexity of Step 1 as $C_1(N+2\cdot 13\cdot 2^{13+1.6}+2\cdot 19\cdot 2^{19+1.6})$. The complexity of getting two lists $L_{1,2}$ and $L_{3,4}$ and sorting them is bounded by $C_1(2\cdot 2^{26}+2\cdot 2^{26})$. Getting and sorting the list $\mathcal{L}$ costs $C_1(2^{52+1}p^T_{\text{fa}}(t)+26\cdot 2^{26})$ by Lemma 1 in the Appendix since there are about $2^{52}p^T_{\text{fa}}(t)$ of $(\kappa, \boldsymbol{\beta})$'s satisfying $T(\kappa, \boldsymbol{\beta}) \geq tN\epsilon^2$. Thus, ignoring the negligible terms, we estimate the complexity of Algorithm 2MC as

$$C_1(N + 2^{52+1}p^T_{\text{fa}}(t)) + 2^{52+4}p^C_{\text{fa}}(t), \tag{9}$$

where $t = 1 - \Phi^{-1}(p_S)/(\sqrt{N}\epsilon)$, and the amount of required memory is $2^{52}p^T_{\text{fa}}(t)$. But if we do not perform the final sorting, then the complexity is

$$C_1N + 2^{52+4+\alpha}p^C_{\text{fa}}(t)$$

with required memory $O(2^{32})$, where $\alpha$ is negligible. The complexity of Algorithm 2MT is

$$C_1N + 2^{52+4-1}p^T_{\text{fa}}(t) \tag{10}$$

with required memory less than $2^{30}$. We have presented the complexities given by (9) and (10) in Fig. 5 varying $N$ and $p_S$.

The computational complexity presented in Fig. 5 seems to be very close to those presented in Fig. 13 of [BV17] at first sight. For $N$ near $2^{42.6}$, they are close indeed. But there is large difference for $N$ less than $2^{42.2}$ for large $p_S$. For example, for $N = 2^{41.75}$, the computational complexities are $2^{45.47}$ and $2^{46.28}$, for $p_S = 0.9$ and $p_S = 0.95$, respectively. But those presented in [BV17] are about $2^{48}$ and $2^{52}$, respectively. Our estimates are slightly better than those presented in [BP18] for $N$ less than $2^{41}$ as shown in Table 1.

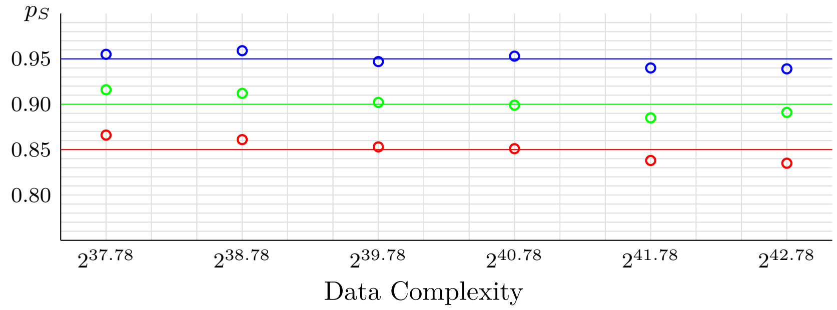

**Figure 6:** Experimental Success Probability

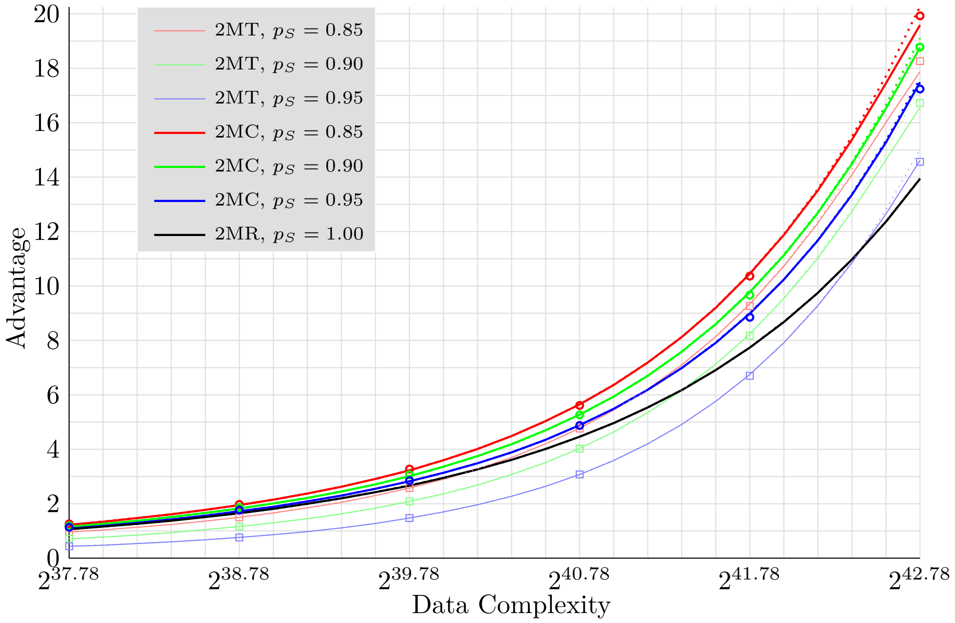

**Figure 7:** Theoretical and Experimental Advantages for Attacks on DES

## 6.3 Experimental Verification

We set the data size $N$ to be $2^{37.78+l}$ with $l = 0, 1, 2, 3, 4,$ or 5. We performed the experiments using 1,000 keys. In each experiment, we fixed $N$, generated 1,000 keys, and then generated data of size $N$ with each key. For each $(N, K)$ and aimed success probability $0.85, 0.9$ and 0.95, we proceeded with Algorithm 2MT/2MC until we get the information necessary to estimate the success probability and the advantage. That is, we stop when we know whether the attack is successful or not and we know the number of the false alarms and the number of false alarms $(\boldsymbol{\kappa}, \boldsymbol{\beta})$ ranked higher in the list than the correct key $(\boldsymbol{\kappa}^*, \boldsymbol{\beta}^*)$. The theoretical estimates of advantages for Algorithm 2MT, 2MR, and 2MC are as in Fig. 7 by Theorems 6, 7 and 8. For Algorithm 2MC, the lower bounds for the advantage coming from the upper bounds on $p_{\text{fa}}^C(t)$ are shown by thick curves. The dotted curves show the approximate advantages obtained from only major wrong key types for Algorithm 2MT, 2MR, and 2MC. For Algorithm 2MC, the difference between the lower bound and upper bound for the estimates on $p_{\text{fa}}^C(t)$ are visible for larger $N$. Note that the advantages for 2MR are those for the Algorithm 2 style attacks in [BCQ04]. The small rectangles and circles correspond to the experimental advantages for Algorithm 2MT and

Algorithm 2MC . This result confirms the validity of our estimates of the advantage for each attack.
In case $N = 2^{42.78}$, we used a single PC equipped with 1 Core i7 CPU and 2 GTX 1080 GPUs and the 1,000 experiments took about 3 months. Data generation requiring $N$ DES encryptions and data compression were performed on the GPUs and the other steps were carried out on the CPU. In each of the test with one pair of $(N, K)$, most of the time was spent on the data generation and the CPU time for each test was just a few minutes or less. The experimental success probabilities are depicted in Fig. 6 which confirms the correctness of our estimates.

# 7 Discussion

## 7.1 Comparison with the Statistic Used in [BCQ04]

A. Biryukov et al.[BCQ04] used the "quadratic" statistic

$$\sum_j \left((-1)^{\beta_j}\epsilon_j - \tau_j^I(D)/N\right)^2 = \epsilon^2 - 2\sum_j (-1)^{\beta_j}\epsilon_j\tau_j^I(D)/N + \sum_j \tau_j^I(D)^2/N^2$$

in their Algorithm 1 style attack. Note that it equals $\epsilon^2 - 2T^I(\boldsymbol{\beta}, D)/N + \sum_j \tau_j^I(D)^2/N^2$ where $T^I(\boldsymbol{\beta}, D)$ is the statistic we have used. Since $\epsilon^2$ is a constant and $\sum_j \tau_j^I(D)^2/N^2$ is also a constant for given data $D$, the rank of $\boldsymbol{\beta}$ with respect to their statistic is exactly the same as its rank with respect to ours. It seems that the existence of the added quadratic term $\sum_j \tau_j^I(D)^2/N^2$ makes it hard to get a threshold based variant using their statistic.

## 7.2 The Validity of the Statistical Models

In [BTV18], it is shown that the distribution of the correlation vector may not be normal taking an example of SMALLPRESENT, where they use many nondominant trails. But we use a small number of linear trails that are close-to-dominant so we think that our models do not contradict their observations. We do not claim that the right key statistics and the wrong key statistics are always independent. But we think that such assumption is reasonable in many cases considering similar hypothesis adopted in some of the rank based Algorithm 2 style attacks [Sel08, HCN19]. We think that the validity of the statistical models when using a small number of dominant and statistically independent trails has been confirmed by our experiments with full DES. But the validity of the models presented in Sect. 5 together with the efficiency of the attack methods using them needs to be checked in future works.

## 7.3 Experimental Verification with full DES

We think that we have provided the most convincing experimental results for the claims on the complexity of the attack on the full DES using multiple linear approximations. We have performed key recovery experiments with 1,000 keys for data size up to $2^{42.78}$ and provided the details of the used data. In [BV17], no experimental results with key recovery are provided. Instead it is claimed that corroborating experimental results can be obtained since the approximated distributions for right keys and wrong keys are accurate. In [FS18], an experimental result with just 1 key is provided so that the claims therein do not seem to be verified experimentally. In [BP18], experiments with 79 keys are claimed to have been performed, but the experimental success probabilities and advantages were not presented.

**Table 3:** Comparison of Advantages for the $\chi^2$ method, Algorithm 2MT, and 2MC

| $N$ | $\chi^2$ method | Algorithm 2MT | Algorithm 2MC |
|---|---|---|---|
| $2^{40.78}$ | 2.16 | 4.60 | 5.63 |
| $2^{41.78}$ | 5.48 | 9.08 | 10.46 |
| $2^{42.78}$ | 13.66 | 18.23 | 19.98 |

## 7.4 Necessity of Considering Wrong Key Types

To the best of our knowledge, all the previous works regarding multiple linear attacks neglect the wrong key types other than the major ones. The consideration of such wrong key types does not lead to improved attacks. On the contrary, it may decrease the advantage of the attack. That is, there are cases we need to consider the other wrong key types to prevent overestimation of the advantage as noted in Sect. 3.3.2 regarding Algorithm 2MR. We can also observe difference between the advantage and the approximated advantage in Algorithm 2MT and Algorithm 2MC. It seems straightforward to incorporate the consideration of wrong key types into multivariate linear attacks using the $\chi^2$ statistic since it is separable. It is not clear whether one can get a refined version of Biryukov et al's Algorithm 2 style attack that incorporates the wrong key types using their quadratic statistic.

## 7.5 The Coefficients in the Linear Statistic

We have used the statistic $T(K^*, \boldsymbol{\kappa}, \boldsymbol{\beta}) = \sum_j (-1)^{\beta_j} \epsilon_j \tau_j(K^*, \kappa_j, D)$ throughout the work. Considering each $\tau_j(K^*, \kappa_j, D)$ as the component statistics, we have used the coefficients $(-1)^{\beta_1}\epsilon_1, \cdots, (-1)^{\beta_m}\epsilon_m$ in forming the linear combination. This yields good tradeoff between the success probability and the advantage for the attack described in Sect. 3 using dominant and statistically independent linear trails since it makes the linear combination an approximation of the LLR statistic. In the generalized attack described in Sect. 5, we can get estimates for success probability and advantage with different coefficients by modifying the provided arguments. There will be better combination of coefficients according to the distribution of right key statistics, but we leave the issue of getting the optimal combination as future works.

## 7.6 Comparison with $\chi^2$ Method in [BTV18]

Though the multivariate linear cryptanalysis [BTV18] depends less on the knowledge of the distribution of the statistic vectors, it has small advantage compared to Algorithm 2MC (and even 2MT) when using dominant and statistically independent linear trails. When we use $m$ independent dominant trails with correlations $\epsilon_1, \cdots, \epsilon_m$, with data size $N \ll 2^n$, the success probability and the false alarm probability of the attack are $\Pr_{X \sim \chi'^2_m(N\epsilon^2)}(X \geq \theta)$ and $\Pr_{X \sim \chi^2_m}(X \geq \theta)$, respectively, with the threshold condition

$$\mathcal{T}(\boldsymbol{\kappa}) \geq \theta,$$

where $\mathcal{T}$ is the $\chi^2$ statistic (5) and $\epsilon^2 = \sum_j \epsilon_j^2$. So, if we use the 4 linear trails described in Sect. 6.1, we get the advantages given in Table 3 with success probability 0.85.

## 7.7 Other Considerations

Throughout the work, we implicitly have assumed that the concatenation of the parity bits and the outer key bits is uniformly distributed as $K^*$ varies. So the parity bits and the outer key bits don't have any relations. But we may adapt the methods in this work to deal with cases when they are related in many cases.

Dominant trails are not common to observe for modern block ciphers, but the methods in Sect. 3 seem to be relevant to tweakable blockciphers since it can be argued that each linear trail behaves like a dominant one when random data is used [LKK18].

# 8 Conclusive Remarks

We have presented a new method of the multiple linear cryptanalysis exploiting dominant and statistically independent trails. We have also presented a generalized framework of multiple linear attack making use of close-to-dominant trails that may not be statistically independent. Applying the method to DES, we get a strongly verified attack that can be regarded as one of the most efficient attacks on DES as of now. As further works, one might try to improve the attack on DES by finding suitable trails with large squared correlations that can be used together with the 4 trails already used in this work. The generalized method in this work needs further analysis to confirm its validity and to measure its efficiency.

# Acknowledgments

We are grateful to the anonymous reviewers and K. Nyberg for their help in improving the quality of the paper considerably. This work was supported by Institute for Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korean government(MSIT) (No.2017-0-00267).

# References

[AR16] Tomer Ashur and Vincent Rijmen. On linear hulls and trails. In Orr Dunkelman and Somitra Kumar Sanadhya, editors, *INDOCRYPT 2016*, volume 10095 of *LNCS*, pages 269–286. Springer, 2016.

[BCQ04] Alex Biryukov, Christophe De Cannière, and Michaël Quisquater. On multiple linear approximations. In Matthew K. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 1–22. Springer, 2004.

[BJV04] Thomas Baignères, Pascal Junod, and Serge Vaudenay. How far can we go beyond linear cryptanalysis? In Pil Joong Lee, editor, *Advances in Cryptology - ASIACRYPT 2004, 10th International Conference on the Theory and Application of Cryptology and Information Security, Jeju Island, Korea, December 5-9, 2004, Proceedings*, volume 3329 of *Lecture Notes in Computer Science*, pages 432–450. Springer, 2004.

[BN16] Céline Blondeau and Kaisa Nyberg. Improved parameter estimates for correlation and capacity deviates in linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2016(2):162–191, 2016.

[BN17] Céline Blondeau and Kaisa Nyberg. Joint data and key distribution of simple, multiple, and multidimensional linear cryptanalysis test statistic and its impact to data complexity. *Des. Codes Cryptography*, 82(1-2):319–349, 2017.

[BP18] Eli Biham and Stav Perle. Conditional linear cryptanalysis - cryptanalysis of DES with less than 242 complexity. *IACR Trans. Symmetric Cryptol.*, 2018(3):215–264, 2018.

[BT13] Andrey Bogdanov and Elmar Tischhauser. On the wrong key randomisation and key equivalence hypotheses in Matsui's algorithm 2. In Shiho Moriai, editor, *FSE 2013, Revised Selected Papers*, volume 8424 of *LNCS*, pages 19–38. Springer, 2013.

[BTV18] Andrey Bogdanov, Elmar Tischhauser, and Philip S. Vejre. Multivariate profiling of hulls for linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2018(1):101–125, 2018.

[BV08] Thomas Baignères and Serge Vaudenay. The complexity of distinguishing distributions (invited talk). In Reihaneh Safavi-Naini, editor, *Information Theoretic Security, Third International Conference, ICITS 2008, Calgary, Canada, August 10-13, 2008, Proceedings*, volume 5155 of *Lecture Notes in Computer Science*, pages 210–222. Springer, 2008.

[BV17] Andrey Bogdanov and Philip S. Vejre. Linear cryptanalysis of DES with asymmetries. In Tsuyoshi Takagi and Thomas Peyrin, editors, *ASIACRYPT 2017, Part I*, volume 10624 of *LNCS*, pages 187–216. Springer, 2017.

[CHN08] Joo Yeon Cho, Miia Hermelin, and Kaisa Nyberg. A new technique for multidimensional linear cryptanalysis with applications on reduced round Serpent. In Pil Joong Lee and Jung Hee Cheon, editors, *ICISC 2008, Revised Selected Papers*, volume 5461 of *Lecture Notes in Computer Science*, pages 383–398. Springer, 2008.

[Cho10] Joo Yeon Cho. Linear cryptanalysis of reduced-round PRESENT. In Josef Pieprzyk, editor, *CT-RSA 2010*, volume 5985 of *Lecture Notes in Computer Science*, pages 302–317. Springer, 2010.

[CSQ07] Baudoin Collard, François-Xavier Standaert, and Jean-Jacques Quisquater. Improving the time complexity of Matsui's linear cryptanalysis. In Kil-Hyun Nam and Gwangsoo Rhee, editors, *ICISC 2007*, volume 4817 of *LNCS*, pages 77–88. Springer, 2007.

[DR02] Joan Daemen and Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard.* Information Security and Cryptography. Springer, 2002.

[DR07] Joan Daemen and Vincent Rijmen. Probability distributions of correlation and differentials in block ciphers. *J. Mathematical Cryptology*, 1(3):221–242, 2007.

[FS18] Stian Fauskanger and Igor Semaev. Separable statistics and multidimensional linear cryptanalysis. *IACR Trans. Symmetric Cryptol.*, 2018(2):79–110, 2018.

[HCN09] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. Multidimensional extension of Matsui's algorithm 2. In Orr Dunkelman, editor, *FSE 2009*, volume 5665 of *LNCS*, pages 209–227. Springer, 2009.

[HCN19] Miia Hermelin, Joo Yeon Cho, and Kaisa Nyberg. Multidimensional linear cryptanalysis. *J. Cryptology*, 32(1):1–34, 2019.

[HVLN15] Jialin Huang, Serge Vaudenay, Xuejia Lai, and Kaisa Nyberg. Capacity and data complexity in multidimensional linear attack. In Rosario Gennaro and Matthew Robshaw, editors, *CRYPTO 2015, Part I*, volume 9215 of *LNCS*, pages 141–160. Springer, 2015.

[JR94] Burton S. Kaliski Jr. and Matthew J. B. Robshaw. Linear cryptanalysis using multiple approximations. In Yvo Desmedt, editor, *CRYPTO '94*, volume 839 of *LNCS*, pages 26–39. Springer, 1994.

[JV03] Pascal Junod and Serge Vaudenay. Optimal key ranking procedures in a statistical cryptanalysis. In Thomas Johansson, editor, *Fast Software Encryption, 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003, Revised Papers*, volume 2887 of *Lecture Notes in Computer Science*, pages 235–246. Springer, 2003.

[LKK18] Jung-Keun Lee, Bonwook Koo, and Woo-Hwan Kim. A general framework for the related-key linear attack against block ciphers with linear key schedules. Cryptology ePrint Archive, Report 2018/152, 2018.

[Mat93] Mitsuru Matsui. Linear cryptanalysis method for DES cipher. In Tor Helleseth, editor, *EUROCRYPT '93*, volume 765 of *LNCS*, pages 386–397. Springer, 1993.

[Nyb94] Kaisa Nyberg. Linear approximation of block ciphers. In Alfredo De Santis, editor, *EUROCRYPT '94*, volume 950 of *LNCS*, pages 439–444. Springer, 1994.

[Nyb19] Kaisa Nyberg. Affine linear cryptanalysis. *Cryptography and Communications*, 11(3):367–377, 2019.

[Ohk09] Kenji Ohkuma. Weak keys of reduced-round PRESENT for linear cryptanalysis. In Michael J. Jacobson Jr., Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *SAC 2009, Revised Selected Papers*, volume 5867 of *Lecture Notes in Computer Science*, pages 249–265. Springer, 2009.

[RN13] Andrea Röck and Kaisa Nyberg. Generalization of Matsui's algorithm 1 to linear hull for key-alternating block ciphers. *Des. Codes Cryptography*, 66(1-3):175–193, 2013.

[Sel08] Ali Aydin Selçuk. On probability of success in linear and differential cryptanalysis. *J. Cryptology*, 21(1):131–147, 2008.

# A Proof of Proposition 1

Though Proposition 1 seems to be well-known, we provide a proof since we could not find one in the literature:

$$\Pr_{\boldsymbol{X}}(\langle \boldsymbol{a}, \boldsymbol{X}\rangle + b \geq 0) = \frac{1}{(\sqrt{2\pi})^m |\det(\boldsymbol{\sigma})|} \int_{\boldsymbol{x}:\langle \boldsymbol{a},\boldsymbol{x}\rangle + b \geq 0} e^{-\frac{(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}{2}} d\boldsymbol{x},$$

which is equal to

$$\frac{1}{(\sqrt{2\pi})^m} \int_{\boldsymbol{y}:\langle \boldsymbol{\sigma}^T\boldsymbol{a},\boldsymbol{y}\rangle + \langle \boldsymbol{a},\boldsymbol{\mu}\rangle + b \geq 0} e^{-\frac{\boldsymbol{y}^T\boldsymbol{y}}{2}} d\boldsymbol{y} \tag{11}$$

by change of variables $\boldsymbol{y} = \boldsymbol{\sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$. Let $\Theta$ be any rigid motion of $\mathbb{R}^m$ such that $\Theta(\boldsymbol{0}) = \boldsymbol{0}$ and $\Theta(\boldsymbol{\sigma}^T\boldsymbol{a}) = (|\boldsymbol{\sigma}^T\boldsymbol{a}|, 0, \cdots, 0)$. By change of variables $\boldsymbol{z} = \Theta\boldsymbol{y}$, the half space $\{\boldsymbol{y} : \langle \boldsymbol{\sigma}^T\boldsymbol{a}, \boldsymbol{y}\rangle + \langle \boldsymbol{a}, \boldsymbol{\mu}\rangle + b \geq 0\}$ is sent onto the "vertical" half space $\{\boldsymbol{z} : |\boldsymbol{\sigma}^T\boldsymbol{a}|z_1 + \langle \boldsymbol{a}, \boldsymbol{\mu}\rangle + b \geq 0\}$. So (11) is equal to

$$\frac{1}{(\sqrt{2\pi})^m} \int_{\boldsymbol{z}:|\boldsymbol{\sigma}^T\boldsymbol{a}|z_1 + \langle \boldsymbol{a},\boldsymbol{\mu}\rangle + b \geq 0} e^{-\frac{\boldsymbol{z}^T\boldsymbol{z}}{2}} d\boldsymbol{z},$$

which is again equal to $\Phi\left((\langle \boldsymbol{a}, \boldsymbol{\mu}\rangle + b)/|\boldsymbol{\sigma}^T\boldsymbol{a}|\right)$. □

# B Proof of Statements in Sect. 4

**Proof of Theorem 9.** The success probability $p_S(t)$ of the attack is $\Pr_D(T(\kappa^*, \beta^*, D) \geq tN\epsilon^2)$ which is seen to be $\Phi((1-t)\sqrt{N}|\epsilon|)$ by Proposition 1 under Hypothesis 11. To consider the false alarm probability, we classify the false alarms into two types: Those of the first type are $(\kappa, \beta)$'s such that $\kappa \neq \kappa^*$; The others are $(\kappa, \beta)$'s such that $\kappa = \kappa^*$ and $\beta \neq \beta^*$. Using Proposition 1, we also have

$$\Pr_{\kappa \neq \kappa^*, D}((-1)^\beta \epsilon T(\kappa, \beta, D) \geq tN\epsilon^2) = \Phi(-t\sqrt{N}|\epsilon|)$$

under Hypothesis 13 for each $\beta$. For $\beta \neq \beta^*$,

$$\Pr_D((-1)^\beta \epsilon T(\kappa^*, \beta, D) \geq tN\epsilon^2) = \Phi((-1-t)\sqrt{N}|\epsilon|)$$

under Hypothesis 11. So the false alarm probability is

$$\begin{aligned}
&\Pr(\mathbf{cond}, (\kappa, \beta) \neq (\kappa^*, \beta^*)) = \Pr(\mathbf{cond}, \kappa \neq \kappa^*) + \Pr(\mathbf{cond}, \kappa = \kappa^*, \beta \neq \beta^*) \\
&= \Pr(\mathbf{cond} \mid \kappa \neq \kappa^*)\Pr(\kappa \neq \kappa^*) + \Pr(\mathbf{cond} \mid \kappa = \kappa^*, \beta \neq \beta^*)\Pr(\kappa = \kappa^*, \beta \neq \beta^*) \\
&= (2^{k_O} - 1)\Phi(-t\sqrt{N}|\epsilon|)/2^{k_O} + \Phi((-1-t)\sqrt{N}|\epsilon|)/2^{k_O+1},
\end{aligned}$$

where **cond** is short for the statement $T(\kappa^*, \beta, D)\epsilon \geq tN\epsilon^2$. □

**Proof of Theorem 10.** The success probability is 1 for this attack since we try all the candidate values for $\kappa^*$. To consider the false alarm probability $p_{\rm fa}$, we classify the false alarms into two types again. Then the false alarm probability of the first type is

$$\Pr_{D,\kappa}(\kappa \neq \kappa^*)\Pr_{D,\kappa \neq \kappa^*}\left((-1)^\beta \epsilon\tau(\kappa, D) \geq (-1)^{\beta^*}\epsilon\tau(\kappa^*, D)\right). \tag{12}$$

Note that $\Pr_{D,\kappa}(\kappa \neq \kappa^*) = (2^{k_O} - 1)/2^{k_O}$. Let $(U, V)$ be the vector-valued random variable having the 2-variate normal distribution with mean $(N\epsilon^2, 0)$ and covariance matrix $(N\epsilon^2)\boldsymbol{I}_2$. Then, under Hypothesis 13, for each $\beta \in \{0, 1\}$

$$\begin{aligned}
&\Pr_{D,\kappa \neq \kappa^*}\left((-1)^\beta \epsilon\tau(\kappa, D) \geq (-1)^{\beta^*}\epsilon\tau(\kappa^*, D)\right) \\
&= \Pr_{(U,V)}(V \geq U),
\end{aligned}$$

which is equal to $\Phi(-\sqrt{N/2}|\epsilon|)$ by Proposition 1. So (12) is equal to

$$\frac{2^{k_O} - 1}{2^{k_O}}\Phi\left(-\sqrt{\frac{N}{2}}|\epsilon|\right).$$

The false alarm probability of the second type is

$$\Pr_D\left((-1)^{\beta^*}\tau(\kappa^*, D) \leq 0\right)/2^{k_O+1} = \Phi(-\sqrt{N}|\epsilon|)/2^{k_O+1}.$$

Summing up the false alarm probabilities over all the types, we get the result. □

**Proof of Theorem 11.** The success probability of the attack is $\Phi((1-t)\sqrt{N}|\epsilon|)$ just as in Algorithm 2T. The expected false alarm probability $p_{\rm fa}(t)$ can be estimated similarly as in Algorithm 2R. We consider two types of false alarms in this case, too, but we further classify the attacks according as the attack is successful or not. When the attack is successful, the wrong keys with the statistic larger than that of the correct key are false alarms. Otherwise, all the wrong keys with the statistic larger than the threshold value are false alarms. Thus $p_{\rm fa}(t)$ is

$$\begin{aligned}
&\tfrac{2^{k_O}-1}{2^{k_O}}\Pr_{D,\kappa \neq \kappa^*}(T(\kappa, \beta, D) \geq T(\kappa^*, \beta^*, D) \geq tN\epsilon^2) \\
&+\tfrac{2^{k_O}-1}{2^{k_O}}\Pr_{D,\kappa \neq \kappa^*}(T(\kappa, \beta, D) \geq tN\epsilon^2, T(\kappa^*, \beta^*, D) \leq tN\epsilon^2) \\
&+\tfrac{1}{2^{k_O+1}}\Pr_D(T(\kappa^*, \beta^*, D) \leq \min(0, -tN\epsilon^2))
\end{aligned}$$

Note that

$$\begin{aligned}&\Pr_{D,\kappa\neq\kappa^*}(T(\kappa,\beta,D)\geq T(\kappa^*,\beta^*,D)\geq tN\epsilon^2)\\&=\int_{t\sqrt{N}\epsilon}^{\infty}\int_{x}^{\infty}\phi(\epsilon,1/\sqrt{N};x)\phi(0,1/N;y)dy\,dx,\end{aligned}$$

and

$$\begin{aligned}&\Pr_{D,\kappa\neq\kappa^*}(T(\kappa,\beta,D)\geq tN\epsilon^2,T(\kappa^*,\beta^*,D)\leq tN\epsilon^2)\\&=\int_{-\infty}^{t\sqrt{N}\epsilon}\int_{t\sqrt{N}\epsilon}^{\infty}\phi(\epsilon,1/\sqrt{N};x)\phi(0,1/N;y)dy\,dx.\end{aligned}$$

under Hypothesis 13. Note also that

$$\Pr_D\left(T(\kappa^*,\beta^*,D)\leq\min(0,-tN\epsilon^2)\right)$$

is $\Phi(-\sqrt{N}|\epsilon|)$ for $t\leq 0$ and $\Phi(-(1+t)\sqrt{N}|\epsilon|)$ for $t\geq 0$. □

# C An Auxiliary Lemma and Its Proof

**Lemma 1.** *Suppose that we are given two lists $[(a_i,i)\in\mathbb{R}\times\mathbb{Z}:i=0,\cdots,n_1]$ and $[(b_j,j)\in\mathbb{R}\times\mathbb{Z}:i=0,\cdots,n_2]$ that are sorted according to the values of $a_i$ and $b_j$ in the descending order. Let $\theta$ be a real number. Assume that there are $N_1$ of $(i,j)$'s for which $a_i+b_j\geq\theta$. Then the complexity of getting the list $[(a_i+b_j,i,j):a_i+b_j\geq\theta]$ is bounded by $O(\min(n_2\log_2(n_1),n_1\log_2(n_2))+N_1)$.*

*Proof.* We may assume that $n_2\geq n_1$. For each $i$, find $j_i$ such that $a_i+b_j\geq\theta$ exactly for $j\leq j_i$. Finding such $j_i$ costs $\log_2(n_2)$ additions and comparisons for each $i$ by binary search. For each $i$, compute $a_i+b_j$ and put $(a_i+b_j,(i,j))$ into the list for each $j\leq j_i$. The complexity of this algorithm is $O(n_1\log_2(n_2)+N_1)$. □

# D Separability of the LLR statistic for independent random variables

We will show that the LLR statistic is separable with respect to independent distributions:

**Theorem 14.** *Let $\mathcal{D}_j^0$ and $\mathcal{D}_j^1$ be the probability distributions on $\{0,1\}^{d_j}$ for each $j=1,\cdots,m$ such that, for each $b=0,1$, $\mathcal{D}_1^b,\cdots,\mathcal{D}_m^b$ are independent. For each $b=0,1$, let $\mathcal{D}^b$ be the probability distribution of the $(d_1+\cdots+d_m)$-bit valued random variable $(\boldsymbol{X}_1^b,\cdots,\boldsymbol{X}_m^b)$ such that each $\boldsymbol{X}_j^b$ has the probability distribution $\mathcal{D}_j^b$. Let $S$ be a multiset consisting of elements of $\{0,1\}^{d_1+\cdots+d_m}$. For each $j$, let $S^j$ be the multiset consisting of elements of $\{0,1\}^{d_j}$ obtained by taking the $j$-th projection for each element of $S$. Then*

$$LLR(S,\mathcal{D}^1,\mathcal{D}^0)=\sum\nolimits_j LLR(S^j,\mathcal{D}_j^1,\mathcal{D}_j^0).$$

*Proof.* For each $j$ and $b$, let $p_j^b$ be the pdf of $\mathcal{D}_j^b$. Also, for each $b$,let $p^b$ be the pdf of $\mathcal{D}^b$. By the independence of the distributions,

$$p^b(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m)=\prod\nolimits_j p_j^b(\boldsymbol{x}_j)$$

for each $(\boldsymbol{x}_1,\cdots,\boldsymbol{x}_m)\in\{0,1\}^{d_1+\cdots+d_m}$. Let $\hat{p}^S$ be the pdf of the empirical probability distribution on $\{0,1\}^{d_1+\cdots+d_m}$ obtained from $S$. Also, for each $j$, let $\hat{p}_j^S$ be the pdf of the empirical probability distribution on $\{0,1\}^{d_j}$ obtained from $S^j$. Thus

$$\hat{p}_j^S(\boldsymbol{x}_j)=\sum\nolimits_{\boldsymbol{y}:\boldsymbol{y}_j=\boldsymbol{x}_j}\hat{p}^S(\boldsymbol{y})$$

for each $j$ and each $\boldsymbol{x}_j$. Now, letting $N = |S|$, $LLR(S, \mathcal{D}^1, \mathcal{D}^0)$ is, by definition,

$$N \sum\nolimits_{(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)} \hat{p}^S(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m) \log \frac{p^1(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)}{p^0(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)},$$

which is equal to

$$\begin{aligned}
&N \sum\nolimits_{(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)} \hat{p}^S(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m) \left( \sum\nolimits_j \log \tfrac{p_j^1(\boldsymbol{x}_j)}{p_j^0(\boldsymbol{x}_j)} \right) \\
&= N \sum\nolimits_j \left( \sum\nolimits_{(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m)} \hat{p}^S(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_m) \log \tfrac{p_j^1(\boldsymbol{x}_j)}{p_j^0(\boldsymbol{x}_j)} \right) \\
&= N \sum\nolimits_j \left( \sum\nolimits_{\boldsymbol{x}_j} \log \tfrac{p_j^1(\boldsymbol{x}_j)}{p_j^0(\boldsymbol{x}_j)} \left( \sum\nolimits_{\boldsymbol{y}:\boldsymbol{y}_j = \boldsymbol{x}_j} \hat{p}^S(\boldsymbol{y}) \right) \right) \\
&= N \sum\nolimits_j \left( \sum\nolimits_{\boldsymbol{x}_j} \hat{p}_j^S(\boldsymbol{x}_j) \log \tfrac{p_j^1(\boldsymbol{x}_j)}{p_j^0(\boldsymbol{x}_j)} \right) \\
&= \sum\nolimits_j LLR(S^j, \mathcal{D}_j^1, \mathcal{D}_j^0)).
\end{aligned}$$

□

## E Doing without the final sorting in the last step of Algorithm 2MC with full DES

Divide the sorted lists $\mathcal{L}'_{1,2}$ and $\mathcal{L}'_{3,4}$ into $2^b$ sublists $\mathcal{L}'_{(1,2),i}$'s and $\mathcal{L}'{}_{(3,4),j}$'s of the same size $2^{26-b}$ (where $10 \leq b \leq 16$ is chosen suitably) also considering the order of the statistics. That is, the values $T^{1,2}(\kappa_{1,2}, \beta_1, \beta_2)$ are not smaller for entries in $\mathcal{L}'_{(1,2),i_1}$ than for entries in $\mathcal{L}'_{(1,2),i_2}$, whenever $i_1 < i_2$, and similarly for $\mathcal{L}'{}_{(3,4),j}$'s. Then we can sort $(i,j)$'s according to the value of $\mathcal{L}'_{(1,2),i}[0] + \mathcal{L}'{}_{(3,4),j}[0]$, where $\mathcal{L}'_{(1,2),i}[0]$ and $\mathcal{L}'{}_{(3,4),j}[0]$ are the maximum of the statistics for the entries in $\mathcal{L}'_{(1,2),i}$ and $\mathcal{L}'{}_{(3,4),j}$, respectively. Then in the descending order, for each $(i,j)$, we try all $(\boldsymbol{\kappa}, \boldsymbol{\beta})$ that are obtained from combining entries in the list $\mathcal{L}'_{(1,2),i}$ and $\mathcal{L}'_{(3,4),j}$ if $\mathcal{L}'_{(1,2),i}[0] + \mathcal{L}'{}_{(3,4),j}[0] \geq tN\epsilon^2$. This may decrease the advantage since we do not try the $(\boldsymbol{\kappa}, \boldsymbol{\beta})$'s in the optimal order, but the decrease is negligible as we have checked in the experiments setting $b$ suitably for each data size $N$ in the range we are considering.

## F The keys and data used in the experiments with DES

To accelerate the data generation with GPUs, we used the following (keys and) plaintexts instead of randomly generated ones. For each fixed data size $N$, the 56-bit key $K^l = (K^l[0], |K^l[1])$ for the $l$-th test was set as follows ($l = 0, \ldots, 999$):
  - $K^l[0]$ = `0x012345` + $l$ (mod $2^{28}$)
  - $K^l[1]$ = `0x6789ab` + $l$ (mod $2^{28}$)

For each key $K^l$, the 64-bit plaintext $P_i^l$ was set to be
`0x123456789abcdef`×$l$ + `0xfedcba987654321`×$i$ (mod $2^{64}$) ($i = 0, \ldots, N-1$).