Integrals go Statistical: Cryptanalysis of Full Skipjack Variants

Meiqin Wang[1,2], Tingting Cui[1], Huaifeng Chen[1], Ling Sun[1], Long Wen[1], Andrey Bogdanov[3]

[1] Key Laboratory of Cryptologic Technology and Information Security,
Ministry of Education, Shandong University, Jinan 250100, China
[2] State Key Laboratory of Cryptology, P.O. Box 5159, Beijing 100878, China
[3] Technical University of Denmark, Denmark

Abstract. Integral attacks form a powerful class of cryptanalytic techniques that have been widely used in the security analysis of block ciphers. The integral distinguishers are based on balanced properties holding with probability one. To obtain a distinguisher covering more rounds, an attacker will normally increase the data complexity by iterating through more plaintexts with a given structure under the strict limitation of the full codebook. On the other hand, an integral property can only be deterministically verified if the plaintexts cover all possible values of a bit selection. These circumstances have somehow restrained the applications of integral cryptanalysis.

In this paper, we aim to address these limitations and propose a novel *statistical integral distinguisher* where only a part of value sets for these input bit selections are taken into consideration instead of all possible values. This enables us to achieve significantly lower data complexities for our statistical integral distinguisher as compared to those of traditional integral distinguisher. As an illustration, we successfully attack the full-round Skipjack-BABABABA for the first time, which is the variant of NSA's Skipjack block cipher.

Keywords: Block cipher, Statistical integral, Integral attack, Skipjack-BABABABA

1 Introduction

Integral attack is an important cryptanalytic technique for symmetric-key ciphers, which was originally proposed by Knudsen as a dedicated attack against Square cipher [7]. Later, Knudsen and Wagner unified it as integral attack [11]. The integral distinguisher of this attack makes use of the *balanced property* where one fixes a part of plaintext bits and takes all possible values for the other plaintext bits such that a specific part of the corresponding ciphertext gets balanced, i.e., each possible partial value for the ciphertext occurs exactly the same number of times. If one additional linear layer after this distinguisher is considered, the property will be that the XOR of all possible values of the specific part of ciphertext becomes zero, referred to as *zero-sum property* [1] throughout this paper[4]. Being variants of the original integral distinguisher, saturation distinguisher [15] and multiset distinguisher [3] also use the same balanced property or zero-sum property with probability one as integral distinguisher.

Statistical saturation attack is different from integral attack, as proposed by Collard and Standaert in [6]. Here by choosing a plaintext set with some bits fixed while the others vary randomly, the statistical saturation distinguisher tracks the evolution of a non-uniform plaintext distribution through the cipher instead of observing the evolution of the plaintext bits in the integral distinguisher. In other words, the statistical saturation distinguisher requires the same inputs as the integral distinguisher, but uses the different property on the output side to distinguish between the right or wrong key guesses. As Leander showed that the statistical saturation distinguisher is identical to multidimensional linear distinguisher on average in [13], the statistical saturation distinguisher makes use of the advantage (bias or capacity) while the balanced property used in the integral distinguisher has no bias. The first publication of statistical saturation distinguisher came without a method to estimate its complexity. However, this complexity was demonstrated to be inverse proportional to the capacity or square of the capacity for the output under the chosen input set [4, 13]. Block ciphers such as PRESENT and PUFFIN are natural targets for such statistical saturation attacks as well as linear cryptanalysis, but the integral cryptanalysis has not been proven efficient for them [21, 22]. This highlights the difference between the integral distinguisher and statistical saturation distinguisher.

Integral attack has been widely used for many other block ciphers. In order to reduce the time complexity of integral attack, Moriai *et al.* gave a method to improve the time complexity against low degree round function for higher order differential attacks including integral attacks in [16]. Ferguson *et al.* proposed the partial-sum technique in [8]. Sasaki and Wang presented the meet-in-the-middle technique for integral attack on Feistel ciphers in [17].

So far the data complexity for a given integral has been determined by taking all values of a bit selection at the input of the balanced property. However, there are cases where it is possible or even desirable to shift the tradeoff from data towards time. Often it is the data requirements that exceeds the restriction while the time complexity budget of an attack is far from being exhausted. Therefore, in these cases, it is of paramount importance to reduce the data complexity of an attack to make it applicable. An interesting example of this behaviour is constituted by NSA's Skipjack variant Skipjack-BABABABA studied at ASIACRYPT'12 [5]. It has been attacked for 31 rounds with an integral distinguisher, whereas the data complexity prohibits the attack to apply to the full 32 rounds. In this paper, we aim to remove this restriction by proposing a novel type of integral distinguisher that features a lower data complexity with non-balanced output bits that are still distinguishable from random.

[4] Although the common sense of balanced property refers to as zero-sum property, the balanced property used in this paper is active or ALL property.

1.1 Our contributions

Integrals go statistical. We propose a new statistical integral distinguisher that consists in applying a statistical technique on top of the original integral distinguisher with the balanced property. The proposed statistical integral distinguisher requires less data than the original integral distinguisher. Although the balanced property does not strictly hold in the statistical integral distinguisher, we prove that the distribution of output values for a cipher can be distinguished from the distribution of output values which originate from a random permutation. This allows us to distinguish between the two distributions and to construct our statistical integral distinguisher. To quantify the advantage, let s be the number of input bits that take all possible values at some bits of the input while the other input bits are fixed. Furthermore, let t be the number of the output bits that are balanced. Then, for the original integral distinguisher, the data complexity is $\mathcal{O}(2^s)$. At the same time, by deploying our new statistical integral distinguisher, the data complexity is reduced to $\mathcal{O}(2^{s-\frac{t}{2}})$.

In summary, statistical integral attacks we propose have lower data complexity than traditional integral attacks. From [5, 19], the traditional integral distinguisher with the balanced property can be converted to a zero-correlation integral distinguisher, so our proposed statistical integral attacks can be regarded as chosen-plaintext multidimensional zero-correlation attacks.

Note that the statistical integral attack is different from the statistical saturation attack as they use different distinguishers and the statistical integral attack is efficient for word-wise ciphers but the statistical saturation attack seems to be valid for bitwise ciphers.

The effectiveness of our proposed statistical integral distinguisher is well presented with the key-recovery attack the full-round Skipjack-BABABABA.

Key recovery attack on full-round Skipjack's variants. Using the statistical integral cryptanalysis, we propose a first-time cryptanalysis on the full-round Skipjack-BABABABA — a variant of Skipjack suggested by Knudsen *et al.* [10, 12] to strengthen its resistance against impossible differential attacks. Skipjack-BABABABA has been shown to withstand truncated differentials (which implies that the impossible differentials are also thwarted). At ASIACRYPT'12, Bogdanov *et al.* [5] attacked 31-round Skipjack-BABABABA by utilizing a 30-round integral distinguisher. Built upon their work, we achieves the full-round attack of Skipjack-BABABABA by taking advantage of the statistical integral technique. To the best of our knowledge, this is the first full-round cryptanalysis against Skipjack-BABABABA. Moreover, we improved the previous attack on 31-round Skipjack-BABABABA in [5] with the new statistical integral distinguisher. The results are summarized in Table 1.

Outline. The new statistical integral distinguisher is established in Section 2. Section 3 presents the attack on the full-round Skipjack-BABABABA and the improved attack on 31-round Skipjack-BABABABA. Finally the paper is concluded in Section 4.

Table 1: Summary of attacks on Skipjack-BABABABA

Attack	Rounds	Data	Time	Memory	Ref.
Integral ZC	31	2^{48}CP	2^{49}	2^{33} bytes	[5]
Statistical integral	**31**	**$2^{46.8}$CP**	**2^{48}**	**$2^{26.6}$ bytes**	**Sec.3**
Statistical integral	**32**	**$2^{61.7}$CP**	**$2^{78.1}$**	**$2^{65.7}$ bytes**	**Sec.3**

CP: Chosen Plaintext.

2 Statistical integral distinguisher

2.1 Integral distinguisher

In this section, we give some notions and results about the integral distinguisher with balanced property, following the description in [5]. Assume that $H : \mathbb{F}_2^n \to \mathbb{F}_2^n$ is a part of a block cipher. To be convenient and without loss of generality, we split the inputs and outputs into two parts each.

$$H : \mathbb{F}_2^r \times \mathbb{F}_2^s \to \mathbb{F}_2^t \times \mathbb{F}_2^u,\ H(x,y) = \begin{pmatrix} H_1(x,y) \\ H_2(x,y) \end{pmatrix}.$$

Then we use T_λ to denote the function H where the first r bits of its input are fixed to the value λ and only the first t bits of the output are considered:

$$T_\lambda : \mathbb{F}_2^s \to \mathbb{F}_2^t,\ T_\lambda(y) = H_1(\lambda, y).$$

For an integral distinguisher, if y in the above notation iterates all possible values of $\mathbb{F}_2^s$, then the output value $T_\lambda(y)$ is uniformly distributed where $n > s \geq t$ to ensure the balanced property on the t-bit. However, this uniform distribution cannot be obtained if the attacker chooses some random values (other than iterating all possible values) for y. The good side is that when considerable quantity of values of y are chosen, the distribution of $T_\lambda(y)$ can be distinguished from a random variable's distribution. In this case, $T_\lambda(y)$ obeys multivariate hypergeometric distribution while t-bit value chosen randomly from an uniform distribution obeys multinomial distribution. These two distributions can be distinguishable from each other as they have different parameters for large number of input-output pairs N.

2.2 Statistical integral distinguisher

Assume that we need N different values of y to distinguish the above two distributions. A t-bit value $T_\lambda(y) \in \mathbb{F}_2^t$ is computed for each y and we allocate a counter vector $V[T_\lambda(y)], T_\lambda(y) \in \mathbb{F}_2^t$ and initialize these counters to zero. These counters are used to keep track of the number of each value $T_\lambda(y)$. Usually t is far from block size n.

It is easy to construct a simple distinguisher which can be described as follows:

- If there is one or more values of $T_\lambda[y]$ satisfying $V[T_\lambda(y)] > 2^{s-t}$, then output random permutation.
- If there is no value of $T_\lambda[y]$ satisfying $V[T_\lambda(y)] > 2^{s-t}$, then output actual cipher.

However, for a random permutation, the probability satisfying $V[T_\lambda(y)] > 2^{s-t}$ is too low to distinguish from the cipher. For example, if $s = 16$, $t = 8$ and $N = 2^{12}$ values of y are involved. For some fixed z, $0 \le z \le 2^t - 1$, the probability that $T_\lambda(y) = z$ is $p = 2^{-8}$. Then $V[z]$ follows a binomial distribution,

$$V[z] \sim B(N, p),$$

which approximately follows a normal distribution $\phi(Np, Np(1-p))$. The probability that $V[z] > 2^{s-t} = 2^8$ for some fixed z is computed as follows,

$$1 - \Phi(\frac{2^{s-t} - Np}{\sqrt{Np(1-p)}}) \approx 1 - \Phi(60.12) \approx 1.1 \times 10^{-787}.$$

As a result, the probability that any $V[z]$ is greater than 2^8 is upper bounded by $256 \times 1.1 \times 10^{-787}$, which is too low to be detected. Thus such a distinguisher only using single counter value is invalid.

Now we will construct an efficient distinguisher by investigating the distribution of the following statistic

$$C = \sum_{T_\lambda(y)=0}^{2^t-1} \frac{(V[T_\lambda(y)] - N \cdot 2^{-t})^2}{N \cdot 2^{-t}}. \tag{1}$$

This statistic is widely used in probability theory. It was also used in [20] for the χ^2 cryptanalysis on DES.

This statistic C follows different distributions determined by whether we are dealing with an actual cipher (right key guess) or a random permutation (wrong key guess).

Proposition 1. *For sufficiently large N and t, the statistic $\frac{2^s-1}{2^s-N} C_{cipher}$ (C_{cipher} is the statistic C for cipher) follows a χ^2-distribution with degree of freedom $2^t - 1$, which means that C_{cipher} approximately follows a normal distribution with mean and variance*

$$\mu_0 = Exp(C_{cipher}) = (2^t - 1)\frac{2^s - N}{2^s - 1} \text{ and } \sigma_0^2 = Var(C_{cipher}) = 2(2^t - 1)(\frac{2^s - N}{2^s - 1})^2.$$

The statistic C_{random} (C_{random} is the statistic C for randomly drawn permutation) follows a χ^2-distribution with degree of freedom $2^t - 1$, which means that C_{random} approximately follows a normal distribution with mean and variance

$$\mu_1 = Exp(C_{random}) = 2^t - 1 \text{ and } \sigma_1^2 = Var(C_{random}) = 2(2^t - 1).$$

Proof. For a randomly drawn permutation, the values of $V[T_\lambda(y)]$ are obtained by counting the occurrences of $T_\lambda(y)$ when the values are chosen uniformly at random, which follows the multinomial distribution with parameter N and $\boldsymbol{p} = (p_0, \ldots, p_{2^t-1})$, $p_i = 2^{-t}$ $(0 \leq i = T_\lambda(y) < 2^t)$.

The well-known Pearson's χ^2 statistical result is that $\sum_{i=1}^{k} \frac{(X_i - np_i)^2}{np_i}$ follows a χ^2-distribution with degree of freedom $k-1$, where the vector $X = (X_1, \ldots, X_k)$ follows a multinomial distribution with parameters n and $\boldsymbol{p}$, where $\boldsymbol{p} = (p_1, \ldots, p_k)$. We give a short proof for Pearson's χ^2 statistic in Appendix A.1 based on [9, 14].

Thus we get the statistic for the randomly drawn permutation

$$C_{random} = \sum_{i=T_\lambda(y)=0}^{2^t-1} \frac{(V[T_\lambda(y)] - Np_i)^2}{Np_i} = \sum_{i=T_\lambda(y)=0}^{2^t-1} \frac{(V[T_\lambda(y)] - N \cdot 2^{-t})^2}{N \cdot 2^{-t}},$$

which follows a χ^2-distribution with degrees of freedom $2^t - 1$. Then for sufficiently large N and t, C_{random} approximately follows a normal distribution with the expected value and variance:

$$Exp(C_{random}) = 2^t - 1 \text{ and } Var(C_{random}) = 2(2^t - 1).$$

For the cipher, the values of $V[T_\lambda(y)]$ follows a multivariate hypergeometric distribution with parameters $(\boldsymbol{K}, 2^s, N)$, where $\boldsymbol{K} = (2^{s-t}, \ldots, 2^{s-t})$.

If the vector $X = (X_1, \ldots, X_k)$ follows a multivariate hypergeometric distribution with parameters $(\boldsymbol{K}, m, n)$, where $\boldsymbol{K} = (K_1, \ldots, K_k)$ with $\sum_{i=1}^{k} K_i = m$, the statistic $\frac{m-1}{m-n} \sum_{i=1}^{k} \frac{(X_i - np_i)^2}{np_i}$ follows a χ^2-distribution with degree of freedom $k-1$, which is proved in Appendix A.2.

So the statistic for the cipher

$$\frac{2^s - 1}{2^s - N} \sum_{T_\lambda(y)=0}^{2^t-1} \frac{(V[T_\lambda(y)] - N \cdot 2^{-t})^2}{N \cdot 2^{-t}} = \frac{2^s - 1}{2^s - N} C_{cipher}$$

follows a χ^2-distribution with degrees of freedom $2^t - 1$. For sufficiently large N and t, we get C_{cipher} approximately follows a normal distribution with the expected value and variance:

$$Exp(C_{cipher}) = (2^t - 1)\frac{2^s - N}{2^s - 1} \text{ and } Var(C_{cipher}) = 2(2^t - 1)(\frac{2^s - N}{2^s - 1})^2.$$

□

To distinguish these two normal distributions with different means and variances, one can compute the data complexity required as follows, given error probabilities.

Corollary 1 (Data complexity). *Under the assumption of Proposition 1, for type-I error probability α_0 (the probability to wrongfully discard the cipher), type-II error probability α_1 (the probability to wrongfully accept a randomly chosen* *permutation as the cipher), to distinguish a cipher and a randomly chosen permutation based on t-bit outputs when fixing r-bit inputs and randomly choosing values for s-bit inputs, the data complexity can be approximated by*

$$N = \frac{(2^s - 1)(q_{1-\alpha_0} + q_{1-\alpha_1})}{\sqrt{(2^t - 1)/2} + q_{1-\alpha_0}} + 1, \tag{2}$$

where $q_{1-\alpha_0}$ and $q_{1-\alpha_1}$ are the respective quantiles of the standard normal distribution.

Note that this statistic test is based on the decision threshold $\tau = \mu_0 + \sigma_0 q_{1-\alpha_0} = \mu_1 - \sigma_1 q_{1-\alpha_1}$: if $C \leq \tau$, the test outputs 'cipher'. Otherwise, if the statistic $C > \tau$, the test outputs 'random'.

As the integral distinguisher with the balanced property is equivalent to the multidimensional zero-correlation distinguisher [5], the statistical integral attacks can be regarded as the chosen-plaintext multidimensional zero-correlation attacks which require lower data complexity than the known-plaintext multidimensional zero-correlation attacks.

2.3 Experiment results

In order to verify the theoretical model of statistical integral distinguisher, we implement a distinguishing attack on a mini variant of AES with the block size 64-bit denoted as AES* here. The round function of AES* is similar to that of AES, including four operations, *i.e.,* SB, SR, MC and AK. 64-bit block is partitioned into 16 nibbles and SB uses S-box S_0 in LBlock. SR is similar as that of AES, and the matrix used in MC is

$$M = \begin{pmatrix} 1\,1\,4\,9 \\ 9\,1\,1\,4 \\ 4\,9\,1\,1 \\ 1\,4\,9\,1 \end{pmatrix},$$

which is defined over $GF(2^4)$. For the multiplication, each nibble and value in M are considered as a polynomial over $GF(2)$ and then the nibble is multiplied modulo $x^4 + x + 1$ by the value in M. The addition is simply XOR operation. The subkeys are XORed with the nibbles in AK operation.

The distinguisher is shown in Figure 1, where $(A_1^i, A_2^i, A_3^i, A_4^i), i = 1, 2, 3, 4$ denotes that these special 16 bits are balanced in the integral. Note that the state after SB operation in round 3 takes all 2^{16} values in each row, and 2^4 values in each column. However, after SR operation the state takes all 2^{16} values in each column. We consider the distributions of the 8-bit values of the output including the first nibble in the first row and the last nibble in the second row, which are colored in red in Figure 1, so $s = 16, t = 8$ here. If we set $\alpha_0 = 0.2$ and different values for N, α_1 and τ can be computed using Equation (2), thus we proceed the experiment to compute the statistic C for AES* and random permutations. With 1000 times of experiment, we can obtain the empirical error

$$\text{R1:}\ \begin{bmatrix} A_1^1 & C & C & C \\ C & A_2^1 & C & C \\ C & C & A_3^1 & C \\ C & C & C & A_4^1 \end{bmatrix} \xrightarrow{SB} \begin{bmatrix} A_1^1 & C & C & C \\ C & A_2^1 & C & C \\ C & C & A_3^1 & C \\ C & C & C & A_4^1 \end{bmatrix} \xrightarrow{SR} \begin{bmatrix} A_1^1 & C & C & C \\ A_2^1 & C & C & C \\ A_3^1 & C & C & C \\ A_4^1 & C & C & C \end{bmatrix} \xrightarrow{AK \circ MC} \begin{bmatrix} A_1^1 & C & C & C \\ A_2^1 & C & C & C \\ A_3^1 & C & C & C \\ A_4^1 & C & C & C \end{bmatrix}$$

$$\text{R2:}\ \begin{bmatrix} A_1^1 & C & C & C \\ A_2^1 & C & C & C \\ A_3^1 & C & C & C \\ A_4^1 & C & C & C \end{bmatrix} \xrightarrow{SB} \begin{bmatrix} A_1^1 & C & C & C \\ A_2^1 & C & C & C \\ A_3^1 & C & C & C \\ A_4^1 & C & C & C \end{bmatrix} \xrightarrow{SR} \begin{bmatrix} A_1^1 & C & C & C \\ C & C & C & A_2^1 \\ C & C & A_3^1 & C \\ C & A_4^1 & C & C \end{bmatrix} \xrightarrow{AK \circ MC} \begin{bmatrix} A_1^1 & A_2^1 & A_3^1 & A_4^1 \\ A_1^2 & A_2^2 & A_3^2 & A_4^2 \\ A_1^3 & A_2^3 & A_3^3 & A_4^3 \\ A_1^4 & A_2^4 & A_3^4 & A_4^4 \end{bmatrix}$$

$$\text{R3:}\ \begin{bmatrix} A_1^1 & A_2^1 & A_3^1 & A_4^1 \\ A_1^2 & A_2^2 & A_3^2 & A_4^2 \\ A_1^3 & A_2^3 & A_3^3 & A_4^3 \\ A_1^4 & A_2^4 & A_3^4 & A_4^4 \end{bmatrix} \xrightarrow{SB} \begin{bmatrix} A_1^1 & A_2^1 & A_3^1 & A_4^1 \\ A_1^2 & A_2^2 & A_3^2 & A_4^2 \\ A_1^3 & A_2^3 & A_3^3 & A_4^3 \\ A_1^4 & A_2^4 & A_3^4 & A_4^4 \end{bmatrix} \xrightarrow{SR} \begin{bmatrix} A_1^1 & A_1^2 & A_1^3 & A_1^4 \\ A_2^1 & A_2^2 & A_2^3 & A_2^4 \\ A_3^1 & A_3^2 & A_3^3 & A_3^4 \\ A_4^1 & A_4^2 & A_4^3 & A_4^4 \end{bmatrix} \xrightarrow{AK \circ MC} \begin{bmatrix} A_1^1 & A_1^2 & A_1^3 & A_1^4 \\ A_2^1 & A_2^2 & A_2^3 & A_2^4 \\ A_3^1 & A_3^2 & A_3^3 & A_3^4 \\ A_4^1 & A_4^2 & A_4^3 & A_4^4 \end{bmatrix}$$

$$\text{R4:}\ \begin{bmatrix} A_1^1 & A_1^2 & A_1^3 & A_1^4 \\ A_2^1 & A_2^2 & A_2^3 & A_2^4 \\ A_3^1 & A_3^2 & A_3^3 & A_3^4 \\ A_4^1 & A_4^2 & A_4^3 & A_4^4 \end{bmatrix} \xrightarrow{SB} \begin{bmatrix} A_1^1 & A_1^2 & A_1^3 & A_1^4 \\ A_2^1 & A_2^2 & A_2^3 & A_2^4 \\ A_3^1 & A_3^2 & A_3^3 & A_3^4 \\ A_4^1 & A_4^2 & A_4^3 & A_4^4 \end{bmatrix} \xrightarrow{SR} \begin{bmatrix} A_1^1 & A_1^2 & A_1^3 & A_1^4 \\ A_2^2 & A_2^3 & A_2^4 & A_2^1 \\ A_3^3 & A_3^4 & A_3^1 & A_3^2 \\ A_4^4 & A_4^1 & A_4^2 & A_4^3 \end{bmatrix} \xrightarrow{AK} \begin{bmatrix} A_1^1 & A_1^2 & A_1^3 & A_1^4 \\ A_2^2 & A_2^3 & A_2^4 & A_2^1 \\ A_3^3 & A_3^4 & A_3^1 & A_3^2 \\ A_4^4 & A_4^1 & A_4^2 & A_4^3 \end{bmatrix}$$

Fig. 1: Integral property for 4-round AES* (The MC operation in the last round is omitted.)

probabilities $\hat{\alpha_0}$ and $\hat{\alpha_1}$. The experiment results for $\hat{\alpha_0}$ and $\hat{\alpha_1}$ are compared with the theoretical values α_0 and α_1 in Figure 2, which shows that the test results for the error probabilities are in good accordance with those for theoretical model.

3 Statistical integral attack on Skipjack-BABABABA

3.1 Skipjack and its variant Skipjack-BABABABA

Before SIMON and SPECK were proposed in 2013, Skipjack [18] was the only block cipher known to be designed by NSA (declassified in 1998). Skipjack is a 64-bit block cipher with 80-bit key adopting an unbalanced Feistel network with 32 rounds of two types, namely Rule A and Rule B. The 64-bit block of Skipjack is divided into four 16-bit words and each round is described in the form of a linear feedback shift register with additional non-linear keyed G permutation. The keyed G permutation $G : \mathbb{F}_2^{32} \times \mathbb{F}_2^{16} \to \mathbb{F}_2^{16}$ consists of a 4-round Feistel structure whose internal function $F : \mathbb{F}_2^8 \to \mathbb{F}_2^8$ is an 8×8 S-box. Skipjack applies eight rounds of Rule A, followed by eight rounds of Rule B and once again eight rounds of Rule A and finally eight rounds of Rule B. The key schedule of Skipjack takes 10 bytes secret key and uses four bytes at a time to key each G permutation, thus Skipjack's key schedule has a periodicity of five rounds. In this section, we use $k_0, k_1, \ldots, k_9$ to denote the ten bytes secret key. This original Skipjack is often referred to as Skipjack-AABBAABB, where A denotes 4-round Rule A and B denotes 4-round Rule B. A variant of Skipjack, namely Skipjack-BABABABA consisting of four iterations of four-round Rule

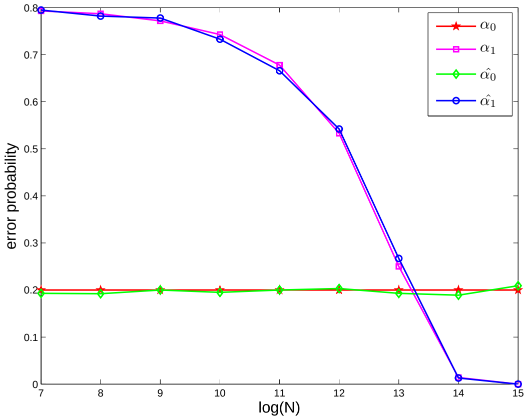

Fig. 2: Experimental results for AES* considering four input nibbles

B followed by four-round Rule A, is also discussed. This variant has the same number of rounds and key schedule as Skipjack-AABBAABB.

Since its declassification, Skipjack-AABBAABB has sparked numerous security analysis. Among which, the best known cryptanalytic result against Skipjack-AABBAABB was reported more than one decade ago by Biham *et al.* [2] at EUROCRYPT'99, where a 24-round impossible differential was revealed and with which an attack against 31-round Skipjack-AABBAABB was mounted. Besides the considerable security analysis, Skipjack's structure was also studied to discuss variants of Skipjack to improve its strength. In [10] and [12], Knudsen *et al.* suggested that putting Rule B before Rule A, for example, the earlier mentioned Skipjack-BABABABA, might facilitate the resistance to truncated differential attacks. Till now, the only security analysis against Skipjack-BABABABA was reported by Bogdanov *et al.* [5] at ASIACRYPT'12, where an integral distinguisher over 30-round Skipjack-BABABABA was utilized to attack a 31-round version.

3.2 Integral distinguisher of Skipjack-BABABABA

To attack full-round Skipjack-BABABABA, we are going to use the 30-round integral distinguisher proposed at ASIACRYPT'12 [5]. The 30-round integral distinguisher can be described as: when we take all 2^{48} possible values for the input of round 2 $(\alpha^2, \beta^2, \gamma^2, \delta^2)$ with $\delta^2 = \alpha^2$, the set of all corresponding values for the output of round 31 $\beta^{32} \oplus \gamma^{32}$ is balanced.

3.3 Key recovery attack on 32-round Skipjack-BABABABA

As the integral distinguisher starts at the input of round 2 and ends at the output of round 31, to attack full-round Skipjack-BABABABA we add one round (Rule

B) before and append one round (Rule A) after the distinguisher, illustrated in Figure 3. Note that in Figure 3, the internal details of the keyed G permutation are also illustrated. To be more clear, several 8-bit variables a, b, c, d are employed in the attack procedure, see Figure 3.

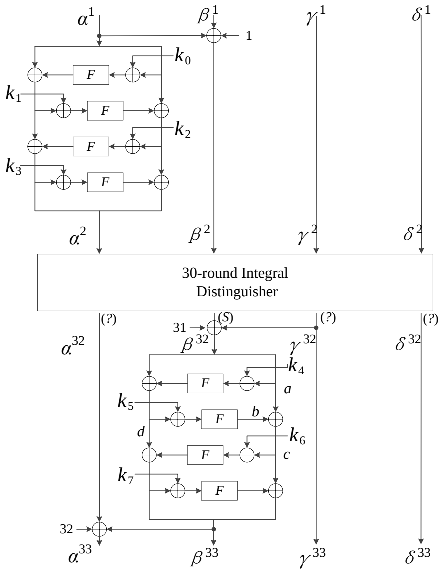

Fig. 3: Key recovery attack on full-round Skipjack-BABABABA

We consider only the integral property of the right 8 bits of $\beta^{32} \oplus \gamma^{32}$, namely $\beta_R^{32} \oplus \gamma_R^{32}$, making $t = 8$ in Equation (2). And according to the 30-round integral distinguisher, to guarantee the integral property with probability one, we need to iterate through all possible values of $(\alpha^2, \beta^2, \gamma^2, \delta^2 = \alpha^2)$. In other words, s in Equation (2) is 48. Set $\alpha_0 = 2^{-2.7}$ and $\alpha_1 = 2^{-4}$ (the values of α_0 and α_1 can be chosen appropriately to balance the data complexity, success rate and time complexity in exhaustive phase), we have $q_{1-\alpha_0} \approx 1.02$ and $q_{1-\alpha_1} \approx 1.53$. Thus we need about $2^{45.7}$ values of $(\alpha^2, \beta^2, \gamma^2, \delta^2 = \alpha^2)$ and the threshold value $\tau \approx 221.6$. We can traverse through all possible values of α^1 and β^1 and randomly choose $2^{13.7}$ values for γ^1 and guess the value of k_0, k_1, k_2, k_3 to compute $\alpha^2, \beta^2, \gamma^2$ and set $\delta^2 = \alpha^2$. In this way, $2^{45.7}$ values of $(\alpha^2, \beta^2, \gamma^2, \delta^2 = \alpha^2)$ could be produced under any key value of (k_0, k_1, k_2, k_3). The key can be recovered following Algorithm 1, where β_R^{33} and β_L^{33} denote the right 8-bit and left 8-bit of β^{33} respectively, and so as γ_R^{33}.

Complexity estimation. In Step 8 and Step 9, the time complexity is $2^{61.7} \cdot 2 = 2^{62.7}$ memory accesses which is equivalent to $2^{62.7}$ encryptions. Next, Step 15 needs about $2^{32} \cdot 2^{16} = 2^{48}$ times of G computation equivalent to $2^{48} \cdot \frac{1}{32} = 2^{43}$

Algorithm 1: Key recovery attack on full-round Skipjack-BABABABA

```
Allocate two counter vector $V_0[]$ and $V_0'[]$ with size $2^{61.7}$ and initialize them to zero.
Allocate a counter $a$ and initialize $a$ to zero.
Take $2^{13.7}$ random values of $\gamma^1$ and store them in set $S$.
for all $2^{16}$ values of $\alpha^1$ do
    for all $2^{16}$ values of $\beta^1$ do
        for all $2^{16}$ values of $\delta^1$ do
            for $2^{13.7}$ values of $\gamma^1$ in set $S$ do
                Ask the ciphertext $(\alpha^{33}, \beta^{33}, \gamma^{33}, \delta^{33})$ for the plaintext $(\alpha^1, \beta^1, \gamma^1, \delta^1)$.
                $V_0[a] = (\alpha^1, \beta^1, \gamma^1, \delta^1)$, $V_0'[a] = (\alpha^{33}, \beta^{33}, \gamma^{33}, \delta^{33})$.
                Increase $a$ by one.
Allocate a counter vector $V_1[\beta^{33}||\gamma_R^{33}]$.
for all $2^{32}$ values of $k_0, k_1, k_2, k_3$ do
    Initialize the counter vector $V_1[\beta^{33}||\gamma_R^{33}]$ to zero.
    for all $2^{16}$ values of $\alpha^1$ do
        Compute $\alpha^2$ and set $\delta^1 = \alpha^2$.
        for all $2^{16}$ values of $\beta^1$ and $2^{13.7}$ values of $\gamma^1$ in set $S$ do
            // Till here, we have $2^{45.7}$ values of $(\alpha^2, \beta^2, \gamma^2, \delta^2 = \alpha^2)$.
            Access $V_0[a]$ with $(\alpha^1, \beta^1, \gamma^1, \delta^1)$ and get the index $a$, then access $V_0'[a]$ to get the corresponding ciphertext $(\alpha^{33}, \beta^{33}, \gamma^{33}, \delta^{33})$.
            Increase the corresponding counter $V_1[\beta^{33}||\gamma_R^{33}]$ by one.
    // $\beta_R^{32} = a \oplus \gamma_R^{33} = b \oplus c \oplus \gamma_R^{33}$
    Allocate a counter vector $V_2[d||c \oplus \gamma_R^{33}]$.
    for all $2^{16}$ values of $k_7$ and $k_6$ do
        Initialize the counter vector $V_2[d||c \oplus \gamma_R^{33}]$ to zero.
        for all $2^{24}$ values of $\beta^{33}||\gamma_R^{33}$ do
            Compute $c = F(\beta_L^{33} \oplus k_7) \oplus \beta_R^{33}$, $d = F(c \oplus k_6) \oplus \beta_L^{33}$.
            Compute $c \oplus \gamma_R^{33}$, update $V_2$ by $V_2[d||c \oplus \gamma_R^{33}] += V_1[\beta^{33}||\gamma_R^{33}]$.
        Allocate a counter vector $V_3[\beta_R^{32} \oplus \gamma_R^{32}]$.
        for all $2^8$ values of $k_5$ do
            Initialize the counter vector $V_3[\beta_R^{32} \oplus \gamma_R^{32}]$ to zero.
            for all $2^{16}$ values of $d||c \oplus \gamma_R^{33}$ do
                Compute $b = F(d \oplus k_5)$ and $\beta_R^{32} \oplus \gamma_R^{32} = b \oplus c \oplus \gamma_R^{33}$.
                Update counter vector $V_3$ by $V_3[\beta_R^{32} \oplus \gamma_R^{32}] += V_2[d||c \oplus \gamma_R^{33}]$.
            Compute $C$ from $V_3$ according to Equation (1).
            if $C \leq \tau$ then
                Exhaustively search all right key candidates compatible with this key value.
```

encryptions. Suppose that one memory access to an array of size 2^{24} and of size $2^{61.7}$ are equivalent to one round encryption and full cipher encryption respectively, then Step 17 and 18 need about $2^{32} \cdot 2^{16} \cdot 2^{16} \cdot 2^{13.7} \cdot (1 + \frac{1}{32}) \approx 2^{77.7}$ encryptions. The operations done in Step 23 and Step 24 are comparable to half-round encryption, which are about $2^{32} \cdot 2^{16} \cdot 2^{24} \cdot \frac{1}{2} \cdot \frac{1}{32} = 2^{66}$ encryptions. In the same way, we regard the operations in Step 29 and Step 30 also as half-round encryption, then the time complexity of these two steps is about $2^{32} \cdot 2^{16} \cdot 2^{8} \cdot 2^{16} \cdot \frac{1}{2} \cdot \frac{1}{32} = 2^{66}$ encryptions. As we set the wrong key guess filteration ratio as $\alpha_1 = 2^{-4}$, thus in Step 33, we need to exhaustively search about $2^{80-4} = 2^{76}$ key values to find the right key. To summarize, the time complexity of our key recovery attack on full-round Skipjack-BABABABA is about $2^{62.7} + 2^{43} + 2^{77.7} + 2^{66} + 2^{66} + 2^{76} \approx 2^{78.1}$ encryptions. About the data complexity, in Step 6, all possible values of δ^1 will be iterated through. Thus our attack needs about $2^{61.7}$ chosen plaintexts. The dominant memory requirements occur to store the the plaintext/ciphertext pairs in Step 1, which needs about $2 \times 2^{61.7} \times 8 = 2^{65.7}$ bytes.

3.4 Improved integral attack on 31-round Skipjack

With the statistical integral model, we can improve the integral attack on 31-round Skipjack [5] by appending one round after the 30-round distinguisher above, too. In Figure 3, we attack from the second round to the 32nd round. In order to reduce the time complexity, we consider the statistical integral property of $\beta_R^{32} \oplus \gamma_R^{32}$ and $\beta_L^{32} \oplus \gamma_L^{32}$ respectively, so $t = 8$ in Equation (2). According to the 30-round integral distinguisher, to guarantee the integral property to hold with probability one, we should iterate through all possible values of $(\alpha^2, \beta^2, \gamma^2, \delta^2 = \alpha^2)$. In other words, s in Equation (2) is 48. Set $\alpha_0 = 2^{-3.7}$ and $\alpha_1 = 2^{-16}$, we have $q_{1-\alpha_0} \approx 1.43$ and $q_{1-\alpha_1} \approx 4.17$. Thus we need about $2^{46.8}$ values of $(\alpha^2, \beta^2, \gamma^2, \delta^2 = \alpha^2)$ and the threshold value $\tau \approx 160.84$. The key recovery attack is described in Algorithm 2.

Complexity estimation. Assume that one memory access is equivalent to one round encryption, Step 3 and 4 need about $2^{46.8} \times \frac{1}{31} \approx 2^{41.8}$ encryptions. Then the operations in Step 9 and 10 are about $2^{16} \times 2^{24} \times \frac{1}{2} \times \frac{1}{31} \approx 2^{34.0}$ encryptions. Step 15 and 16 need about $2^{16} \times 2^{8} \times 2^{16} \times \frac{1}{2} \times \frac{1}{31} \approx 2^{34.0}$ encryptions. As we set the wrong key guess filteration ratio as 2^{-16}, the numbers of remained key (k_5, k_6, k_7) are about $2^{24-16} = 2^8$ in Step 19. Until now, we exploit the integral property of $\beta_R^{32} \oplus \gamma_R^{32}$ to filter most wrong keys. Next, we use the integral property of $\beta_L^{32} \oplus \gamma_L^{32}$ to filter all wrong keys of (k_4, k_5, k_6, k_7). Step 25 needs about $2^8 \times 2^8 \times 2^{24} \times \frac{1}{31} \approx 2^{35.0}$ encryptions. Finally, by setting $\alpha_1 = 2^{-16}$ we need to exhaustively search about $2^{80-16-16} = 2^{48}$ key values in Step 28 to find the right key. In total the time complexity is about $2^{41.8} + 2^{34.0} + 2^{34.0} + 2^{35.0} + 2^{48} \approx 2^{48}$ encryptions. The dominant memory complexity is required in Step 1 which is about $2 \times 2^{24} \times 3 \approx 2^{27.6}$ bytes which happen.

4 Conclusion

In this paper, we propose the statistical integral attack where we use the statistic technique to deal with the original integral distinguisher with balanced property.

Algorithm 2: Key recovery attack on 31-round Skipjack-BABABABA

```
Allocate counter vectors $V_0[\beta^{33}||\gamma_L^{33}]$ and $V_1[\beta^{33}||\gamma_R^{33}]$, then initialize them to zero.
for $2^{46.8}$ random values of $(\alpha^2, \beta^2, \gamma^2, \delta^2 = \alpha^2)$ do
    Ask for the corresponding ciphertext $(\alpha^{33}, \beta^{33}, \gamma^{33}, \delta^{33})$.
    Increase $V_0[\beta^{33}||\gamma_L^{33}]$ and $V_1[\beta^{33}||\gamma_R^{33}]$ by one respectively.
// $\beta_R^{32} = a \oplus \gamma_R^{33} = b \oplus c \oplus \gamma_R^{33}$
Allocate a counter vector $V_2[d||c \oplus \gamma_R^{33}]$ and a list $V_4[\cdot]$.
for all $2^{16}$ values of $k_7$ and $k_6$ do
    Initialize the counter vector $V_2[d||c \oplus \gamma_R^{33}]$ to zero.
    for all $2^{24}$ values of $\beta^{33}||\gamma_R^{33}$ do
        Compute $c = F(\beta_L^{33} \oplus k_7) \oplus \beta_R^{33}$, $d = F(c \oplus k_6) \oplus \beta_L^{33}$.
        Compute $c \oplus \gamma_R^{33}$, update $V_2$ by $V_2[d||c \oplus \gamma_R^{33}] += V_1[\beta^{33}||\gamma_R^{33}]$.
    Allocate a counter vector $V_3[\beta_R^{32} \oplus \gamma_R^{32}]$.
    for all $2^8$ values of $k_5$ do
        Initialize the counter vector $V_3[\beta_R^{32} \oplus \gamma_R^{32}]$ to zero.
        for all $2^{16}$ values of $d||c \oplus \gamma_R^{33}$ do
            Compute $b = F(d \oplus k_5)$ and $\beta_R^{32} \oplus \gamma_R^{32} = b \oplus c \oplus \gamma_R^{33}$.
            Update counter vector $V_3$ by $V_3[\beta_R^{32} \oplus \gamma_R^{32}] += V_2[d||c \oplus \gamma_R^{33}]$.
        Compute $C$ from $V_3$ according to Equation (1).
        if $C \leq \tau$ then
            Store the $(k_5, k_6, k_7)$ in the list $V_4[\cdot]$.
// Since $\alpha_1 = 2^{-16}$, about $2^8$ keys in $V_4$.
Allocate a counter vector $V_5[\beta_L^{32} \oplus \gamma_L^{32}]$.
for all values of $(k_5, k_6, k_7)$ in $V_4[\cdot]$ do
    for all $2^8$ values of $k_4$ do
        Initialize the counter vector $V_5[\beta_L^{32} \oplus \gamma_L^{32}]$ to zero.
        for all $2^{24}$ values of $\beta^{33}||\gamma_L^{33}$ do
            Compute $\beta_L^{32}$, update counter vector $V_5$ by
            $V_5[\beta_L^{32} \oplus \gamma_L^{32}] += V_0[\beta^{33}||\gamma_L^{33}]$.
        Compute $C$ from $V_5$ according to Equation (1).
        if $C \leq \tau$ then
            Exhaustively search all right key candidates compatible with this key value.
```

The new integral attack has the lower data complexity than that of the original one. Our experiment for mini version of AES shows that the experimental results are in good accordance with the theoretic results. What' more, with this new distinguisher we can improve the previous integral attack on 31-round Skipjack-BABABABA and achieve the full-round attack of Skipjack-BABABABA. In the future, we will apply the statistical integral model to many other block ciphers which are vulnerable to integral attack.

Acknowledgments This work has been supported by 973 Program (No. 2013C B834205), NSFC Projects (No. 61133013, No. 61572293), Program for New Century Excellent Talents in University of China (NCET- 13-0350).

References

1. Aumasson, J. P., Meier, W.: Zero-Sum Distinguishers for Reduced Keccak-f and for the Core Functions of Luffa and Hamsi. Presented at the rump session of Cryptographic Hardware and Embedded Systems-CHES 2009, 2009.
2. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: Stern, J. (ed.) EUROCRYPT 1999, LNCS 1592, pp. 12-23. Springer, Heildelberg (1999)
3. Biryukov, A., Shamir, A.: Structural Cryptanalysis of SASAS. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 394–405. Springer, Heidelberg (2001)
4. Blondeau, C., Nyberg, K.: Links Between Truncated Differential and Multidimensional Linear Properties of Block Ciphers and Underlying Attack Complexities. In: Nguyen, P. Q., Oswqld. (eds) EUROCRYPT 2014, LNCS, vol.8441, pp.165-182. Springer, Heidelberg (2014)
5. Bogdanov, A., Leander, G., Nyberg, K., Wang, M.: Integral and Multidimensional Linear Distinguishers with Correlation Zero. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012, LNCS, vol. 7658, pp. 244-261. Springer, Heidelberg (2012)
6. Collard, B., Standaert, F. X.: A Statistical Saturation Attack Against The Block Cipher PRESENT. In: Fischlin, M. (ed.) CT-RSA 2009. LNCS, vol. 5473, pp. 195–210. Springer, Heidelberg (2009)
7. Daemen, J., Knudsen, L. R., Rijmen, V.: The Block Cipher Square. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 149-165. Springer, Heidelberg (1997)
8. Ferguson, N., Kelsey, J., Lucks, S., Schneier, B., Stay, M., Wagner, D., Whiting, D.: Improved Cryptanalysis of Rijndael. In: Schneier, B. (ed.) FSE 2000. LNCS, vol. 1978, pp. 213–230. Springer, Heidelberg (2001)
9. Fergnson, T. S.: A Course in Large Sample Theory. London: Chapman and Hall. (1996)
10. Knudsen, L. R., Robshaw, M. J. B., Wagner, D.: Truncated Differentials and Skipjack. In: Wiener, M. (ed.) CRYPTO 1999, LNCS, vol. 1666, pp. 165-180. Springer, Heidelberg (1999)
11. Knudsen, L. R., Wagner, D.: Integral Cryptanalysis. In: Daemen, J., Rijmen, V. (eds.) FSE 2002. LNCS, vol. 2365, pp. 112-127. Springer, Heidelberg (2002)
12. Knudsen, L. R., Wagner, D.: On the Structure of Skipjack. Discrete Applied Mathematics 111(1-2): pp. 103-116. Elsevier (2001)
13. Leander, G.: On Linear Hulls, Statistical Saturation Attacks, PRESENT and a Cryptanalysis of PUFFIN. In: Paternson, K.G. (ed.) EUROCRYPT 2011. LNCS, vol. 6632, pp. 303-322. Springer, Heidelberg (2011)
14. Lehmann, E. L.: Elements of Large-Sample Theory. New York: Springer-Verlag. (1999)
15. Lucks, S.: The Saturation Attack–A Bait for Twofish. In: Matsui, M. (ed.) FSE 2001. LNCS, vol. 2355, pp. 1–15. Springer, Heidelberg (2002)
16. Moriai, S., Shimoyama, T., Kaneko, T.: Higher Order Differential Attack of a CAST Cipher. In: Vaudenay, S. (Ed.) FSE 1998. LNCS, vol. 1372, pp. 17-31. Springer, Heidelberg (1998)

17. Sasaki, Y., Wang, L.: Meet-in-the-Middle Technique for Integral Attacks against Feistel Ciphers. In: Knudsen, L., Wu, H. (eds.) SAC 2012. LNCS, vol. 7707, pp. 234-251. Springer, Heidelberg (2013)
18. Skipjack and KEA Algorithm Specifications, Version 2.0, 29 May 1998. Available at the National Institute of Standards and Technology's web page, http://csrc.nist.gov/groups/ST/toolkit/documents/skipjack/skipjack.pdf
19. Sun, B., Liu, Z., Rijmen, V., Li, R., Cheng, L., Wang, Q., Alkhzaimi, H., Li, C.: Links among Impossible Differential, Integral and Zero Correlation Linear Cryptanalysis. http://eprint.iacr.org/2015/181.pdf
20. Vaudenay, S.: An Experiment on DES Statistical Cryptanalysis. In Proceedings of the 3rd ACM conference on Computer and communications security, pp. 139-147. ACM. (1996)
21. Wu, S.,Wang, M.: Integral Attacks on Reduced-Round PRESENT. In: Qing, S. et al. (eds.) ICICS 2013. LNCS, vol. 8233, pp. 331–345. Springer, Heidelberg (2013)
22. Z'aba, M. R., Raddum, H., Henricksen, M., Dawson E.: Bit-Pattern Based Integral Attack. In: Nyberg, K. (ed.) FSE 2008. LNCS, vol. 5086, pp. 363-381. Springer, Heidelberg (2008)

A Appendices

A.1 Pearson's χ^2 statistic from the multinomial distribution

In this subsection, we describe Pearson's χ^2 statistic deduced from multinomial distribution and provide a short proof based on [9,14] the asymptotic distribution of the χ^2 expression.

A fundamental result about Pearson's χ^2 statistic is that the following expression follows a χ^2-distribution with degree of freedom $k-1$

$$\sum_{i=1}^{k} \frac{(X_i - np_i)^2}{np_i},$$

where the random vector $\boldsymbol{X} = (X_1, \ldots, X_k)$ follows a multinomial distribution with parameters n and $\boldsymbol{p}$, where $\boldsymbol{p} = (p_1, \ldots, p_k)$ with $\sum_{i=1}^{k} p_i = 1$.

Now we will give a short proof based on [9, 14] in the following.

In probability theory, the multinomial distribution is a generalization of the binomial distribution. For n independent trials each of which leads to a success for exact one of k categories, with each category i $(1 \leq i \leq k)$ having a given fixed success probability p_i satisfying $\sum_{i=1}^{k} p_i = 1$. Then if the random variable X_i indicates that the number of times outcome number i is observed over the n trials, the vector $\boldsymbol{X} = (X_1, \ldots, X_k)$ follows a multinomial distribution with parameters n and $\boldsymbol{p} = (p_1, \ldots, p_k)$. Note that while the trials are independent, k outcomes are dependent because they must be summed to n.

Since the variance of X_j is $np_j(1-p_j)$ and $Cov(X_j, X_l) = -np_jp_l, j \neq l$, the random vector $\boldsymbol{X}$ with $(k-1)$ dimensions has covariance matrix

$$\varSigma = \begin{pmatrix} np_1(1-p_1) & -np_1p_2 & \dots & -np_1p_{k-1} \\ -np_1p_2 & np_2(1-p_2) & \dots & -np_2p_{k-1} \\ \vdots & \vdots & \ddots & \vdots \\ -np_1p_{k-1} & -np_2p_{k-1} & \dots & np_{k-1}(1-p_{k-1}) \end{pmatrix}.$$

So we can denote $\varSigma$ as follows,

$$\varSigma = n(D - \boldsymbol{p'p}),$$

where $\boldsymbol{p} = (p_1, p_2, \dots, p_{k-1})$ and $\boldsymbol{p'}$ is its transposition, D is a $(k-1) \times (k-1)$ diagonal matrix and

$$D = \begin{pmatrix} p_1 & & & \\ & p_2 & & \\ & & \ddots & \\ & & & p_{k-1} \end{pmatrix}.$$

Thus, one can show

$$\varSigma^{-1} = \frac{1}{n}\left(D^{-1} + \frac{D^{-1}\boldsymbol{p'p}D^{-1}}{1 - \boldsymbol{p}D^{-1}\boldsymbol{p'}}\right) = \frac{1}{n}\left(D^{-1} + \frac{E}{p_k}\right),$$

where E is a $(k-1) \times (k-1)$ matrix where all entries are equal to one.

We only consider $k-1$ dimensions here, since using all k dimensions would make the variance singular. The first $k-1$ dimensions have all of the information needed anyway, so there's no problem in doing this.

There is a fact: for any d-dimensional normal $\boldsymbol{X}$ with nonsingular covariance matrix, the statistic $(\boldsymbol{X} - \boldsymbol{\mu})'\varSigma^{-1}(\boldsymbol{X} - \boldsymbol{\mu})$ follows a χ^2-distribution with degree of freedom d.

Thus, in the above case we concern $(k-1)$-dimensional normal $\boldsymbol{X}$:

$$\begin{aligned}(\boldsymbol{X} - \boldsymbol{\mu})'\varSigma^{-1}(\boldsymbol{X} - \boldsymbol{\mu}) &= (\boldsymbol{X} - n\boldsymbol{p})'\left(\frac{1}{n}\left(D^{-1} + \frac{E}{p_k}\right)\right)(\boldsymbol{X} - n\boldsymbol{p}) \\ &= \frac{1}{n}(\boldsymbol{X} - n\boldsymbol{p})'D^{-1}(\boldsymbol{X} - n\boldsymbol{p}) + \frac{1}{np_k}(\boldsymbol{X} - n\boldsymbol{p})'E(\boldsymbol{X} - n\boldsymbol{p}) \\ &= \sum_{i=1}^{k-1}\frac{(X_i - np_i)^2}{np_i} + \frac{1}{np_k}\left(\sum_{i=1}^{k-1}(X_i - np_i)\right)^2 \\ &= \sum_{i=1}^{k}\frac{(X_i - np_i)^2}{np_i} + \frac{1}{np_k}\left((n - x_k) - n(1-p_k)\right)^2 \\ &= \sum_{i=1}^{k}\frac{(X_i - np_i)^2}{np_i}.\end{aligned}$$

That is, $\sum_{i=1}^{k}\frac{(X_i - np_i)^2}{np_i}$ has an approximation to χ^2-distribution with degree of freedom $k-1$ for large enough n.

A.2 Extend Pearson's χ^2 statistic to multivariate hypergeometric distribution

In this subsection, we will extend Pearson's χ^2 statistic to multivariate hypergeometric distribution based on the proof of the above subsection and prove that the following expression follows a χ^2-distribution with degree of freedom $k-1$

$$\frac{m-1}{m-n}\sum_{i=1}^{k}\frac{(X_i-np_i)^2}{np_i},$$

where the random vector $\boldsymbol{X} = (X_1, \ldots, X_k)$ follows a multivariate hypergeometric distribution with parameters $(\boldsymbol{K}, m, n)$ where $\boldsymbol{K} = K_1, \ldots, K_k$ with $\sum_{i=1}^{k} K_i = m$.

The multivariate hypergeometric distribution is a generalization of the hypergeometric distribution. For n dependent trials each of which leads to a success for exact one of k categories, with each category i $(1 \leq i \leq k)$ having a given fixed success probability $(p_1, p_2, \ldots, p_k)$. The multivariate hypergeometric distribution gives the probability of any particular combination of numbers of successes for the various categories.

Then if the random variables X_i indicates that the number of times outcome number i is observed over the n trials, the vector $\boldsymbol{X} = (X_1, \ldots, X_k)$ follows a multivariate hypergeometric distribution with parameters $(\boldsymbol{K}, m, n)$.

As the mean for X_j is np_j and the variance of X_j is $np_j(1-p_j)\frac{m-n}{m-1}$ and since $Cov(X_j, X_l) = -np_jp_l\frac{m-n}{m-1}, j \neq l$, the random vector $\boldsymbol{X}$ with $k-1$ dimension has covariance matrix

$$\Upsilon = n\frac{m-n}{m-1}(D - \boldsymbol{p'p})$$

and

$$\Upsilon^{-1} = \frac{1}{n}\frac{m-1}{m-n}\left(D^{-1} + \frac{D^{-1}\boldsymbol{p'p}D^{-1}}{1-\boldsymbol{p}D^{-1}\boldsymbol{p'}}\right) = \frac{1}{n}\frac{m-1}{m-n}\left(D^{-1} + \frac{E}{p_k}\right).$$

With the similar trick as the above subsection, for the $(k-1)$-dimensional normal $\boldsymbol{X}$, it is easy to show that

$$\begin{aligned}(\boldsymbol{X}-\boldsymbol{\mu})'\Upsilon^{-1}(\boldsymbol{X}-\boldsymbol{\mu}) &= (\boldsymbol{X}-n\boldsymbol{p})'\left(\frac{1}{n}\frac{m-1}{m-n}\left(D^{-1}+\frac{E}{p_k}\right)\right)(\boldsymbol{X}-n\boldsymbol{p}) \\ &= \frac{m-1}{m-n}\sum_{i=1}^{k}\frac{(X_i-np_i)^2}{np_i},\end{aligned}$$

which means that $\frac{m-1}{m-n}\sum_{i=1}^{k}\frac{(X_i-np_i)^2}{np_i}$ has an approximation to χ^2-distribution with degree of freedom $k-1$ for large enough n.