SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

March 2019, Vol. 62 032107:1–032107:16
https://doi.org/10.1007/s11432-018-9489-x

A new discrete Fourier transform randomness test

Meihui CHEN[1,2], Hua CHEN[1*], Limin FAN[1], Shaofeng ZHU[1,2], Wei XI[3] & Dengguo FENG[1]

[1]*Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing* 100190*, China;*
[2]*University of Chinese Academy of Sciences, Beijing* 100049*, China;*
[3]*Southern Power Grid Science Research Institute, Guangzhou* 510080*, China*

Received 13 March 2018/Accepted 15 June 2018/Published online 25 January 2019

Abstract The randomness of random number generators (RNGs) is important for the reliability of cryptographic systems since the outputs of RNGs are usually utilized to construct cryptographic parameters. Statistical tests are employed to evaluate the randomness of the RNG outputs. The discrete Fourier transform (DFT) test is an important test item of the most popular statistical test suite NIST SP800-22. In the standard NIST DFT test and related improved studies, there exist accuracy and efficiency issues. First, the bit sequences generated by known good RNGs have a high probability to be rejected when the sequences are long or the sequence number is large, due to the deviation between the actual distribution of the test statistic values and the assumed normal distribution. Second, the long test time and high memory consumptions of the complex DFT test algorithm also affect its practicability. To solve these problems, we propose a new DFT test method for long sequences (10^6 or more bits). Different from the previous DFT test methods focusing on making the distribution of the test statistic values closer to the normal distribution, we reconstruct the statistic to follow the chi-square distribution. Our experiment result shows that our method has higher reliability in the two-level test, and could effectively reduce the test time and the memory consumptions. When applying our method on randomness test, the test efficiency has been increased to about 4 times for 10^6-bit sequences and 7 times for 10^7-bit sequences. In conclusion, our method has lower probability of making errors, and is more suitable for practical application scenarios.

Keywords statistical randomness test, discrete Fourier transform test, NIST SP800-22, two-level test, chi-square distribution

Citation Chen M H, Chen H, Fan L M, et al. A new discrete Fourier transform randomness test. Sci China Inf Sci, 2019, 62(3): 032107, https://doi.org/10.1007/s11432-018-9489-x

1 Introduction

Random number generators (RNGs) are widely used in various fields, especially in cryptographic systems. There are two types of RNG: true random number generator (TRNG) and pseudo random number generator (PRNG). TRNGs use nondeterministic source (random physical phenomenons) combining some process functions to generate bit sequences which can be used as random numbers directly or seeds for PRNGs. PRNGs utilize mathematical methods to produce deterministic sequences. Random numbers generated by RNGs are basic components in cryptographic systems which are usually utilized to construct cryptographic parameters such as encryption keys [1] and message authentication codes, so the quality of RNG outputs has a direct impact on the security of cryptographic schemes and protocols. There are various test methods to evaluate the randomness of the RNG outputs and the most widely used one is statistical randomness test.

* Corresponding author (email: chenhua@tca.iscas.ac.cn)

Statistical randomness test is an important area of cryptography. It can not only evaluate the outputs of RNGs, but also assess the randomness of cryptographic system's outputs [2]. What's more, statistical randomness test can be employed to analysis the internal states of cryptographic algorithms [3, 4], so the weakness of the cryptographic algorithms can be found. There exist a number of commonly used statistical test suites: Marsaglia's Diehard[1)], CryptXS[2)], SP800-22 [5] published by US NIST (National Institute of Standards and Technology). NIST SP800-22 is the most popular one. It is composed of 15 test items which provide comprehensive randomness evaluation for the test bit sequences.

Statistical test is based on the hypothesis testing which is a classical mathematical statistic method to detect whether the experimental data set fits with the given hypothesis. When the statistical test is used to evaluate the randomness of RNGs, the definition of null hypothesis H_0 is that the RNG under test is ideal, which means that the output sequences cannot be computationally distinguished from true random source. Assuming the output sequences are independent and identically distributed [6], the test statistic values computed on the sequences conform to a specific distribution which usually is binomial distribution or chi-square distribution. When the statistic values are in line with the assumed distribution, H_0 is accepted. Otherwise, H_0 is rejected.

To evaluate whether the distribution of the statistic values is consistent with the assumed distribution, a P-value is calculated based on the value of statistic. P-value is a real number in [0,1], which represents the probability that the true random sequence is worse than the sequence under test. The commonly used testing approaches for statistical tests are one-level and two-level test. For the one-level test, if P-value is greater than the significance level, H_0 is accepted and the sequence under the test is considered random. However the one-level test is not reliable enough. The two-level test can improve the reliability of the randomness test [7]. It focuses on the distribution of N obtained P-values and involves two estimate ways. (1) Passing proportion test: test whether the passing ratio of the N P-values can be approximated with a normal random variable. (2) Uniformity test: use a chi-square goodness-of-fit test in k bins to compare the distribution of the N P-values with the uniform distribution. The two-level test is passed only if both the passing proportion test and the uniformity test are passed. A test item is considered to be passed if its two-level test is passed. A RNG is considered ideal if its output sequences pass all the included test items of a statistical test suite.

Related work. There are some related studies to improve the accuracy and efficiency of the NIST SP800-22 test suite, which generally fall into two kinds. Some studies are designed for a specific statistic test of the test suite, while others aim to find a general improved method which is suitable for the whole test suite.

For the first kind of studies, Hamano [8, 9] corrected the overlapping template matching test and adjusted the parameters used in "test for the longest run of ones in a block". Fan et al. [10] adjusted the freedom degree of the statistical value and the expected number of different lengths runs in runs distribution test, which made the test more accurate. Chen et al. [11] studied the correlation of templates in non-overlapping template matching test, and gave an effective selection strategy based on the correlation of templates, making the number of templates reduced about 50%. Sýs et al. [12] improved the linear complexity test by proposing a new version of the Berlekamp-Massey algorithm and the test efficiency is effectively increased.

The second kind of studies aim to find a general method to improve the whole test suite. To improve the test efficiency and practicability, some studies evaluate the correlations of different test items of the whole test suite and give solutions to select the appropriate test items. Huang and Lai [13] used the conditional entropy to construct a quantitative value for comparing the tests and proposed a basic method on how to determine the tests' optimal execution order. In [14], Fan et al. gave a general hypothesis testing method to evaluate the correlation of statistical tests. And in [15], Sulak et al. defined the coverage efficiency of a test suite which is used to determine the most efficient, the least efficient, and the optimal subsuites of the NIST SP800-22 test suite. There also exists studies to improve the test accuracy and reliability by

1) Marsaglia G. The Marsaglia random number CD-ROM including the DieHard battery of test of randomness. 1995.

2) Caelli W, Dawson E, Nielsen L, et al. CRYPTCX statistical package manual, measuring the strength of stream and block ciphers. 1992.

correcting the errors in the two-level test. Pareschi et al. [16] indicated that the two-level test is sensitive to the errors produced by the approximate computation of P-values. Furthermore, Pareschi et al. [6] indicated a more accurate approximation of the cumulative distribution function (CDF) of P-values and used it to compute the reliability conditions for the two-level test. In [17], Zhu et al. presented that using the absolute value of the statistic to compute the P-value would impact the accuracy of the uniformity test. They proposed to use Q-value, which is computed by the statistic value directly, to replace P-value. And the theoretical proofs and experimental results showed that the reliability of the uniformity test is improved when using Q-value.

In this paper, we prefer the first way and focus on improving the discrete Fourier transform (DFT) test (known as the spectral test) in the NIST SP800-22 test suite. The DFT test is based on the binomial distribution, and is an important test item of NIST SP800-22. It evaluates the randomness of a sequence by detecting periodic features in the bit series. There are some related studies to improve the accuracy of the DFT test. Hamano et al. [18, 19] pointed out the dependence of the spectrums, corrected the threshold value from $\sqrt{3n}$ to $1.7308\cdots\times\sqrt{n}$ and indicated that the test statistic approximately followed the normal distribution $N(0, 0.5)$. Kim et al. [20] modified the value of the threshold and the variance, which made the distribution of statistic more closer to the standard normal distribution. In [6], Pareschi et al. further modified the value of the variance in the DFT test and the new distribution was more coherent with experimental results. In [17], Zhu et al. used the Q-value to replace P-value when conducting the uniformity test based on the DFT test, which made the DFT test is more accurate.

Our contribution. There exist several problems in the DFT test method. First, we find that the sequences with good randomness which are generated by known good RNGs have a high rate to be rejected in the standard NIST DFT test, when the sequences are long or the sequence number is large. This phenomenon is caused by the deviation between the distribution of statistic and the normal distribution, which leading to the unreliability of the two-level test. Although previous studies [6,17–20] tries to reduce this deviation to improve the accuracy of two-level test, they cannot solve this problem completely for the DFT test. Second, none of previous studies consider the actual performance issues of the DFT test. As the DFT computation is time-consuming and the time complexity is nonlinear, the test time will significantly increase along with the sequence length and the sequence number, which is unsuitable for some practical application scenarios, such as fast test of very long sequences on ordinary computers. Although the studies [13–15] for test items' correlations could reduce the test time by appropriately selecting partial test items, instead of performing the whole test items, they cannot fundamentally improve the test efficiency of any specific test item, including the DFT test. Third, the DFT algorithm requires high memory consumptions, previous DFT test methods have a high possibility to crash for long sequence test, due to the memory limitations of ordinary computers. In fact, we find that the official DFT test program of NIST SP800-22 and its corrections will always work abnormally on our experiment computer (equipped with Intel i7 CPU and 8 GB RAM) when testing long sequences whose length is 10^8 or more bits.

In this paper, we propose a new DFT test method to solve the above problems. Since there is a deviation between the distribution of statistic and the normal distribution, we jump out of the old correction ideas which focus on making the distribution of statistic closer to the standard normal distribution, and construct a new statistic following the chi-square distribution, by dividing a sequence into several subsequences. Through this correction, our two-level test is more reliable and the probability of rejecting good RNGs is reduced. The test efficiency is also effectively improved. When applying our new DFT method, the test efficiency has been increased to about 4 times for sequences with length 10^6 bits and 7 times for sequences with length 10^7 bits. Moreover, our method could reduce the memory consumptions in the DFT test, thus could test longer sequences than previous methods on devices with the same computing capabilities. So our method is more suitable for some practical application scenarios, such as fast test of very long sequences (e.g., 10^8 and 10^9 bits) on ordinary computers. Our experiment result also proves that our method could achieve these advantages without effectiveness sacrifice, i.e., our method has nearly the same capability with the standard NIST method and its corrections to detect bad RNGs. So our method has the same level ability to avoid false positive errors compared with previous studies.

In a word, the new DFT test method we proposed has higher reliability in the two-level test, lower probability of making errors, and is less time-consuming and memory-consuming.

Organization. The paper is organized as follows. In Section 2, we briefly introduce the one/two-level test and present the procedure of the official DFT test in NIST SP800-22 [5] and its corrections. In Section 3, we propose a new DFT test method for long sequences after analysing the errors in standard NIST DFT test and the reliability in two-level test based on the official DFT test. In Section 4, we conduct some experiments about our new DFT test method and give the comparisons with the previous DFT test methods. In Section 5, we draw some conclusion.

2 Preliminaries

In this section, we briefly introduce the one/two-level statistical test based on hypothesis testing method. We also present the standard DFT test in NIST SP800-22 [5] and the improved studies of it.

2.1 One-level (standard) test

In the one-level testing approach, a sequence of n bits is generated by the RNG under test and a P-value p is computed using the assumed distribution of the statistical test. Compare the P-value p with the significance level α:

• If $p > \alpha$, H_0 is accepted and the sequence generated by the RNG under test is regarded as random;

• If $p \leqslant \alpha$, H_0 is rejected and the sequence generated by the RNG under test is considered not random.

A RNG is considered perfect when its generated sequences are accepted by all test items in a statistical test suite.

However, the test is sometimes not exact. There exist two kinds of errors [16]:

• Type I error. Reject H_0 when the sequence is generated by a perfect RNG.

• Type II error. Accept H_0 when the sequence is generated by a RNG that is not random.

The probability of type I error (denoted by α) is called level of significance. The value of α is usually small. NIST suggests that $\alpha = 0.01$.

It is well known that the one-level test is not reliable enough, i.e., sometimes a PRNG which is not random enough can also pass the test. For example, if a RNG is not random, but the numbers of 0 and 1 in the sequence generated by the RNG are balanced, this non-random generator can also pass the frequency test.

2.2 Two-level (second-level) test

Because of the unreliability of one-level test, the two-level test is proposed to improve the reliability of statistical tests, which has been adopted by NIST SP800-22. In the two-level test, N sequences with length n are generated by the RNG under test. The one-level test is repeated N times and N P-values are obtained. There are two estimate ways to test the distribution of the N P-values.

(1) Passing proportion test. Focus on the passing ratio τ of the N P-values and judge whether the passing ratio can be approximated with a normal random variable. When the P-value of a sequence is equal or greater than α, the sequence is considered to have passed the test. Compute the passing ratio of the N sequences. If N is large enough, τ can be approximated to a normal random variable whose mean $\mu = 1 - \alpha$ and standard deviation $\sigma = \sqrt{\alpha(1-\alpha)/N}$. The passing proportion test is passed when the ratio τ lies in the confidence interval defined as $1 - \alpha \pm 3\sigma$. The parameter of significance level suggested by NIST is $\alpha = 0.01$.

(2) Uniformity test. Focus on the uniformity of N P-values and compare the distribution with the uniform distribution by a chi-square goodness-of-fit test in k bins. The interval [0,1] is uniformly divided into k bins. Compare the observed number O_j of P-values in each sub-interval with the expected uniform number $E_j = N/k$. A statistic is computed by performing a chi-square goodness-of-fit test $\chi^2 = \sum_{j=1}^{k} \frac{(E_j - O_j)^2}{E_j}$ and calculating a level-two P-value $p_T = \mathrm{igamc}(k/2, \chi^2/2)$. Given a significance α_T, the uniformity test is considered passed if $p_T > \alpha_T$. NIST suggests that $k = 10$ and $\alpha_T = 0.0001$.

The two-level test is passed only if both the passing proportion test and the uniformity test are passed. A test item is considered to be passed if its two-level test is passed. A RNG is considered random if its outputs pass all the included test items of a statistical test suite.

2.3 The DFT test in NIST SP800-22

The DFT test is based on the discrete Fourier transform and is used to find the deviation from the assumption of randomness by detecting periodic features in the bit series [5]. The procedure of the DFT test in NIST SP800-22 is as follows.

Let ε be the input binary sequence which is generated by a RNG. The length of ε is n. ε_i is the i-th bit of the sequence ε, where $i = 1, \ldots, n$. Convert the bits of sequence ε to the sequence X with values of -1 and $+1$, as $x_i = 2\varepsilon_i - 1$, $0 \leqslant i \leqslant n-1$. Apply the discrete Fourier transform on $\{x_i\}_{i=0}^{n-1}$ to get a sequence of complex numbers $\{f_j\}_{j=0}^{n-1}$,

$$f_j = \sum_{k=1}^{n} x_k \mathrm{e}^{2\pi \mathrm{i}(k-1)j/n}, \tag{1}$$

where $\mathrm{e}^{2\pi \mathrm{i}kj/n} = \cos(2\pi kj/n) + \mathrm{i}\sin(2\pi kj/n)$, $j = 0, \ldots, n-1$, and $\mathrm{i} \equiv \sqrt{-1}$. f_j equals to f_{n-j} because of the symmetry of the real to complex-value transform, so only half of the complex numbers $\{f_j\}_{j=0}^{n/2-1}$ are considered. Then calculate the modulus of $\{f_j\}_{j=0}^{n/2-1}$ and get a sequence of absolute values $\{|f_j|\}_{j=0}^{n/2-1}$. 95% of the values $\{|f_j|\}_{j=0}^{n/2-1}$ should be less than a threshold value $T_h = \sqrt{(\log \frac{1}{0.05})n}$. Let N_1 be the number of $|f_j|$ less than T_h. N_1 approximately conforms to a normal distribution. Let $N_0 = 0.95n/2$, which is the mean value μ of the normal distribution. And let the variance $\sigma^2 = 0.95 \cdot 0.05 \cdot n/c$ with $c = 4$. The statistic $d = (N_1 - N_0)/\sqrt{n(0.95)(0.05)/4}$. So, the P-value is computed as

$$p = 2(1 - \Phi(d)) = \mathrm{erfc}\left(\frac{|d|}{\sqrt{2}}\right). \tag{2}$$

The test parameters NIST SP800-22 suggests are: sequence length $n = 10^6$, sequence number $N = 1000$.

There are several related studies to improve the accuracy of the official NIST DFT test method. Pareschi et al. [6] modified the variance σ^2 of the assumed normal distribution and corrected the value of c from 4 to 3.8. The variance was modified from $\sigma^2 = 0.95 \cdot 0.05 \cdot n/4$ to $\sigma^2 = 0.95 \cdot 0.05 \cdot n/3.8$. The experiments indicated that the new variance was more accurate.

Zhu et al. [17] replaced the P-value with Q-value when conducting the uniformity test. P-value is computed by the absolute value of the statistic, while Q-value is computed by the statistic directly. When using the Q-value, the test was proved to be more detectable and reliable. The Q-value is calculated as

$$q = \frac{1}{2}\mathrm{erfc}\left(\frac{d}{\sqrt{2}}\right). \tag{3}$$

3 The new DFT randomness test

3.1 Error analysis of DFT test

According to the DFT test in NIST SP800-22, the number of $|f_j|$ less than the threshold value T_h (denoted as N_1) approximately conforms to a normal distribution with mean $\mu = 0.95n/2$ and variance $\sigma^2 = 0.95 \cdot 0.05 \cdot n/4$. The statistic is computed on the approximate normal distribution of N_1. However, after carrying out a large number of experiments, we found that there is a deviation between the distribution of N_1 and the normal distribution, which is consistent with previous studies [19].

We conduct DFT tests using several known good RNGs and get the distributions of N_1. The experiment results indicate that the probability distributions of N_1 are not all perfectly in line with the normal distribution. For example, we apply discrete Fourier transform on 2×10^6 sequences whose length is 10^5 bits

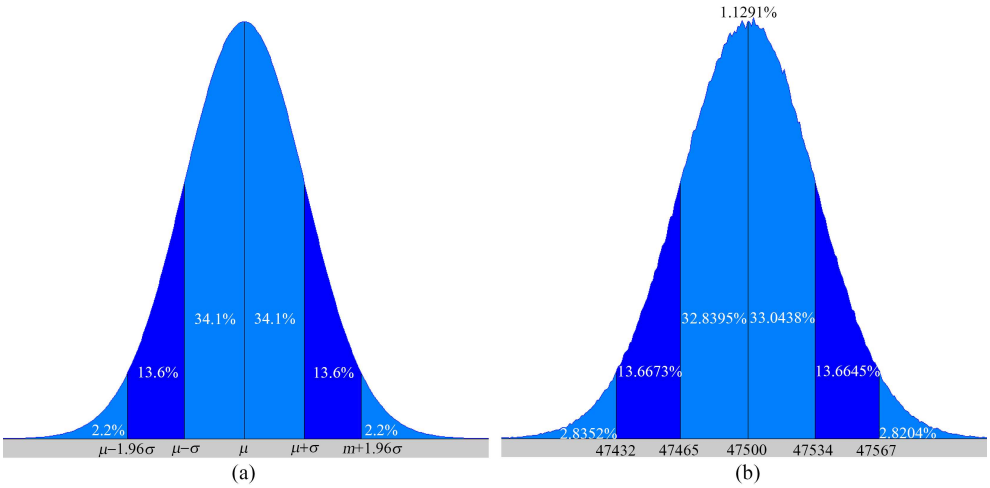

Figure 1 (Color online) Comparisons of theoretical and experimental distribution. (a) Probability distribution of normal distribution; (b) experimental probability distribution of N_1.

generated by the default good PRNG AES-256 [21] and the number of $|f_j|$ less than the threshold value T_h for each sequence is recorded. Then we get the probability distribution of N_1 (shown in Figure 1(b)), which has a little difference from the normal distribution in Figure 1(a). To show the comparisons of distribution intuitively, we let μ, $\mu \pm \sigma$ and $\mu \pm 1.96\sigma$ be the cut-off points and divide the distribution into six intervals. When $n = 10^5$, the expected distribution of N_1 should be a normal distribution with $\mu = 47500$, $\sigma = 0.95 \times 0.05 \times 100000/4 \approx 34.5$. Since the actual probability distribution of N_1 is discrete, we give the probability of $N_1 = 47500$ individually.

Figure 1 presents the deviation between the distribution of N_1 and the normal distribution. The probability distribution of N_1 is not completely symmetric as the normal distribution. And the probability of each interval for N_1 is different from the normal distribution. The standard NIST DFT test [5] assumes that the distribution of N_1 conforms to the normal distribution, which is not accurate. When the sequence under test is long or the number of sequences is large, the deviation will be more obvious. The deviation further leads to the approximation error in the computation of P-values. So some sequences generated by RNGs with good randomness cannot pass the standard NIST DFT test.

3.2 Reliability analysis of two-level test

It is well known that the two-level test is not reliable since the known good RNGs always fail in the two-level test when sequence number N is extremely large [6, 7]. This is due to the approximation error in the computation of P-values, which is caused by the deviation between the distribution of the test statistics and the normal distribution. To ensure the reliability of the two-level test, we must derive the error between the approximated P-value p and the actual one p_0. We can compute the upper threshold of N by giving a boundary of the error. The two-level test is considered reliable when N is smaller than its upper threshold.

To give the reliability condition of two-level testing approach, Pareschi et al. [6] proposed a more accurate approximation of the CDF of P-values (presented as Eq. (4)), which is fit for all tests based on binomial distribution in NIST SP800-22. He also gave the error between the level-two P-value p_T calculated by the approximated CDF of P-value and the level-two P-value p_{T_0} calculated by the more accurate CDF.

$$F'_p(x) = x + 2d(x)z(\sqrt{2\sigma^2}\mathrm{erfc}^{-1}(x)). \tag{4}$$

The definition of $d(x)$ and z are

$$d(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-(\mathrm{erfc}^{-1}(x))^2},$$

Table 1 The upper threshold values of sequence number

Length	Passing proportion test		Uniformity test	
	DFT ($c = 4$)	DFT ($c = 3.8$)	DFT ($c = 4$)	DFT ($c = 3.8$)
10000000	12312	6228	2878	4134
1000000	584	99	345	191
100000	32	13	25	35
10000	2	3	2.47	2.84
1000	0.16	0.15	0.26	0.26

$$z(\xi) = \begin{cases} \xi_{(\mathrm{mod}\ 1)}, & \xi_{(\mathrm{mod}\ 1)} < \min(\psi, 1-\psi); \\ \xi_{(\mathrm{mod}\ 1)} - \dfrac{1}{2}, & \min(\psi, 1-\psi) < \xi_{(\mathrm{mod}\ 1)} < \max(\psi, 1-\psi); \\ \xi_{(\mathrm{mod}\ 1)} - 1, & \xi_{(\mathrm{mod}\ 1)} > \max(\psi, 1-\psi); \\ \lim\limits_{x \to \xi^-} z(x), & \text{otherwise.} \end{cases}$$

For the passing proportion test of two-level test, assuming $f_{1-\alpha}(x) = \mathrm{d}F_{1-\alpha}(x)/\mathrm{d}x$, the error between p_T and p_{T_0} is given as

$$p_T - p_{T_0} \simeq -\sqrt{N} f_{1-\alpha}(F_{1-\alpha}^{-1}(1-p_{T_0}))|F_p'(\alpha) - \alpha|, \tag{5}$$

where $F_{1-\alpha}(\theta_0) = 1 - \mathrm{erfc}(\frac{|N\tau - N(1-F_p'(\alpha))|}{\sqrt{2N\alpha(1-\alpha)}})$ with $\theta_0 = \sqrt{N}|\tau - (1-F_p'(\alpha))|$.

Similarly, for the uniformity test, assuming $f_{\chi^2}(x) = \mathrm{d}F_{\chi^2}(x)/\mathrm{d}x$, the error between p_T and p_{T_0} is given in

$$p_T' - p_{T_0} \simeq -f_{\chi^2}\big(F_{\chi^2}^{-1}(1-p_{T_0})\big) N C_{\chi^2}, \tag{6}$$

where $C_{\chi^2} = k\sum_{j=1}^{k}\left(F_p'(\frac{j}{k}) - F_p'(\frac{j-1}{k}) - \frac{1}{k}\right)^2$, $F_{\chi^2}(\chi^2) = \frac{\gamma((k-1)/2;\chi^2/2)}{\Gamma((k-1)/2)}$, $\gamma(k;x)$ is the incomplete gamma function and $\Gamma(k)$ is the complete gamma function.

The equations of error given by Pareschi et al. [6] are proper for all tests based on the binomial distribution in NIST SP800-22. We can give the reliability condition of DFT test with two-level testing approach since the DFT test is based on the binomial distribution. Given a boundary of the error between p_T and p_{T_0}, we can get the upper threshold of sequence number N.

The assumptions are given as follows:

(1) For the passing proportion test, let $\alpha = 0.01$ and the maximum error of the two-level P-value $|p_T'{-}p_{T_0}| < 0.01$, where $0 \leqslant p_{T_0} \leqslant 1$.

(2) For the uniformity test, the maximum error of the two-level P-value $|p_T'{-}p_{T_0}| < 0.01$ and $0 \leqslant p_{T_0} \leqslant 1$ using $k = 10$ bins.

According to the assumptions, we calculated the upper threshold values of the sequence number N for standard NIST DFT test with $c = 4$ [5] or $c = 3.8$ [6]. The upper threshold values are shown in Table 1.

Theoretically, the two-level test is unreliable when the number of sequences under test is larger than the upper threshold values. So sequences with good randomness cannot always pass the passing proportion test and the uniformity test, which is consistent with our experiment results: PRNG AES-256's output sequences with test parameters $n = 10^7$ and $N = 30000$ and PRNG SHA-512's [22] output sequences with test parameters $n = 10^6$ and $N = 100000$ cannot pass the two-level test based on the NIST DFT test. The upper threshold values in Table 1 will be used as a reference baseline to decide the sequence number of our experiments detailed in Subsection 4.1.

3.3 New construction of DFT test statistic

Previous studies for the DFT test always focus on making the distribution of statistic closer to the normal distribution. After analysing the approximation error and the reliability of the DFT test in NIST SP800-22, we decide to jump out of the old correction idea and reconstruct the statistic to follow the chi-square distribution. To achieve this, we divide the input sequence into short sub-sequences with the same length m. We apply the discrete Fourier transform on every sub-sequence and record the number of $|f_j|$ less

than the threshold value T_h for every sub-sequence. Then we can get the distribution of N_1. The statistic is calculated by comparing the experimental distribution of N_1 with the expected distribution using a chi-square goodness-of-fit test in k bins.

To construct the expected distribution, we need first figure out $K+1$ specific ratio intervals of N_1/m. In Subsection 3.1, we compare the distribution of N_1 with normal distribution by dividing the distribution into six intervals. Similarly, we also let $\mu \pm \sigma$ and $\mu \pm 1.96\sigma$ be the cut-off points where $\mu = m/2$, $\sigma = 0.95 \times 0.05 \times m/4$. Since the distribution of N_1 is discrete and the probability of each interval cannot be too small[3)], we add $\mu \pm 0.06\sigma$ as the cut-off points. So the distribution of N_1 is divided into seven intervals: $[0, \mu - 1.96\sigma]$, $(\mu - 1.96\sigma, \mu - \sigma]$, $(\mu - \sigma, \mu - 0.06\sigma]$, $(\mu - 0.06\sigma, \mu + 0.06\sigma]$, $(\mu + 0.06\sigma, \mu + \sigma]$, $(\mu + \sigma, \mu + 1.96\sigma]$, $(\mu + 1.96\sigma, m/2]$. Given the value of m, we can calculate the $K + 1$ specific ratio intervals of N_1/m with $K = 6$. For example, let $m = 100000$, then $\mu = 47500$, $\sigma \approx 34.5$. Seven intervals of N_1/m are as follows: [0, 0.47432], [0.47433, 0.47465], [0.47466, 0.47497], [0.47498, 0.47502], [0.47503, 0.47534], [0.47535, 0.47567], [0.47568, 0.5].

Then we need to figure out the expected probability of N_1/m in each interval. For a specific subsequence length m, we use known-good RNGs to generate large numbers of m-bit sequences and get the distribution of N_1 by applying the discrete Fourier transform on every m-bit sequence. Since the RNGs are all known-good, the N_1 distribution can be used as the expected one. And the larger the sequence number is, the more accurate the expected distribution will be. Then the expected probability of N_1/m in each interval can be calculated. For example, we use PRNG AES-256 to generated 200 million sequences with length 100000 bits and apply the discrete Fourier transform on them. Then, for $m = 100000$, we can get the probability of N_1/m in each interval. We repeat this experiment with different known-good RNGs and a set of probability of N_1/m in each interval is obtained. Then we can analyse the distributions of N_1 for these RNGs and finally get an expected probability of N_1/m in each interval.

We give a length set $\{1000, 10000, 100000\}$ of the sub-sequences to satisfy the test of long sequences with different lengths. The expected probability π_r $(0 \leqslant r \leqslant K)$ of each interval for different subsequence lengths are shown in Table 2. Through our correction, the deviation between the distribution of statistic and the expected distribution is reduced, making the two-level test more reliable. What's more, our method to divide the input sequence also reduces the test time and memory consumptions. These advantages will be detailed in Section 4.

3.4 The new DFT test procedure

According to the new construction of statistic, we propose a new DFT randomness test for long sequences (10^6 or more bits). The symbols and their definitions are shown in Table 3.

The procedure of the new DFT test is as follows:

(1) For a n-bit binary sequence ε. Uniformly divide the sequence ε into M subsequences, each one's length is m bits, where $m \times M = n$, $M \geqslant 200$, $m \geqslant 1000$, and the value of m is in the range $\{1000, 10000, 100000\}$. S_i is the i-th subsequence, $0 \leqslant i \leqslant M - 1$. s_{ij} is the j-th bit of the subsequence S_i, $0 \leqslant j \leqslant m - 1$.

(2) For each subsequence S_i, $0 \leqslant i \leqslant M - 1$, convert it to subsequence Z_i, as $z_{ij} = 2s_{ij} - 1$, $0 \leqslant i \leqslant M - 1$, $0 \leqslant j \leqslant m - 1$.

(3) Apply the discrete Fourier transform on $\{Z_i\}_{i=0}^{M-1}$ and get M subsequences $\{f_i\}_{i=0}^{M-1}$ of complex numbers, f_{ij} is the j-th bit of f_i.

$$f_{ij} = \sum_{t=1}^{m} z_{it} \mathrm{e}^{2\pi \mathrm{i}(t-1)j/m}, \tag{7}$$

where $\mathrm{e}^{2\pi \mathrm{i} tj/m} = \cos(2\pi tj/m) + \mathrm{i}\sin(2\pi tj/m)$, $j = 0, \ldots, m-1$, $0 \leqslant i \leqslant M-1$ and $\mathrm{i} \equiv \sqrt{-1}$. Because of the symmetry of the real to complex-value transform, only half of the complex numbers $\{f_{ij}\}_{j=0}^{m/2-1}$ are considered in each subsequence.

3) According to the required sample size of chi-square test, the expected frequency of every interval should be equal to or larger than 5.

Table 2 The expected proportions in each intervals for sequences of different lengths

Sub-sequence length m	Interval	Probability π_r
1000	[0, 0.468]	0.034601
	[0.469, 0.471]	0.126173
	[0.472, 0.474]	0.278188
	0.475	0.112357
	[0.476, 0.478]	0.287042
	[0.479, 0.481]	0.130616
	[0.482, 0.5]	0.031023
10000	[0, 0.4728]	0.027910
	[0.4729, 0.4739]	0.145946
	[0.4740, 0.4749]	0.306825
	0.4750	0.035620
	[0.4751, 0.4760]	0.309415
	[0.4761, 0.4771]	0.147452
	[0.4772, 0.5]	0.026832
100000	[0, 0.47432]	0.028502
	[0.47433, 0.47465]	0.136399
	[0.47466, 0.47497]	0.306491
	[0.47498, 0.47502]	0.056363
	[0.47503, 0.47534]	0.307504
	[0.47535, 0.47567]	0.136647
	[0.47568, 0.5]	0.028094

Table 3 Symbols of new DFT test

Symbol	Description
n	The test sequence length, $n \geqslant 10^6$
N	The test sequence number
m	The sub-sequence length, $m \in \{1000, 10000, 100000\}$
M	The sub-sequence number, $M \geqslant 200$
S_i	The i-th subsequence, $0 \leqslant i \leqslant M-1$
s_{ij}	The j-th bit of the subsequence S_i, $0 \leqslant j \leqslant m-1$
f_i	The complex sequence after applying the discrete Fourier transform on S_i
f_{ij}	The j-th bit of sequence f_i
N_{i1}	The number of $\lvert f_{ij}\rvert$ less than T_h
v_r	The number of N_{i1}/m belonging to the i-th interval
π_r	The expected probability of each interval for different subsequence lengths

(4) For each subsequence Z_i, $0 \leqslant i \leqslant M-1$, compute the modulus of $\{f_{ij}\}_{j=0}^{m/2-1}$ and get M sequences of absolute values $|f_{ij}|$, $0 \leqslant i \leqslant M-1$, $0 \leqslant j \leqslant m/2-1$.

(5) Compare the values $|f_{ij}|$ $(0 \leqslant i \leqslant M-1,\ 0 \leqslant j \leqslant m/2-1)$ with the threshold value $T_h = \sqrt{(\log \frac{1}{0.05})m}$. For each subsequence S_i, $0 \leqslant i \leqslant M-1$, let N_{i1} be the number of $|f_{ij}|$ less than T_h.

(6) For different subsequence length m, the proportion N_{i1}/m is different. $K+1$ $(K=6)$ specific ratio intervals of N_{i1}/m for different subsequence lengths are given. Let v_r, $0 \leqslant r \leqslant K$ be the number of N_{i1}/m belonging to the r-th interval. Compute the probability of N_{i1}/m falling into the interval and compare it with the expected probability. The expected probabilities π_r, $0 \leqslant r \leqslant K$ of each interval for different subsequence lengths are shown in Table 2.

(7) Compute a test statistic:

$$X^2 = \sum_{r=0}^{K} \frac{(v_r - M\pi_r)^2}{M\pi_r}. \tag{8}$$

Under the randomness hypothesis, X^2 approximately follows the X^2-distribution with K degrees of

freedom. When using the X^2-distribution, a conservative condition is that $M \min_r \pi_r \geqslant 5$. So M should exceed 200.

(8) Compute the P-value:

$$p = \frac{\int_{X^2(\text{obs})}^{\infty} \mathrm{e}^{-u/2} u^{K/2-1} \mathrm{d}u}{\Gamma(K/2) 2^{K/2}} = \text{igamc}\left(\frac{K}{2}, \frac{X^2(\text{obs})}{2}\right). \tag{9}$$

If $p > \alpha$, then ε is accepted as random. Otherwise, ε is considered non-random. α is the significance level whose value is suggested as 0.01 by NIST SP800-22. Here we also define $\alpha = 0.01$.

The algorithm pseudocode for the new DFT test is shown in Algorithm 1.

Algorithm 1 The new DFT test procedure for long sequences

Require: A binary sequence ε; the length of the input sequence n; the length of a sub-sequence m, $m \in \{1000, 10000, 100000\}$; $K+1$ specific ratio intervals of N_{i1}/m for different subsequence lengths; the expected probabilities of each interval, π_r, $0 \leqslant r \leqslant K$.

Ensure: P-value, p.

$M = n/m$;
if $M < 200$ **then**
  return The choice value of m is too big;
else
  Let S_i be the i-th subsequence, $0 \leqslant i \leqslant M-1$ and s_{ij} is the j-th bit of the subsequence S_i, $0 \leqslant j \leqslant m-1$;
  for $i = 0$ to $M-1$ **do**
    for $j = 0$ to $m-1$ **do**
      $z_{ij} = 2s_{ij} - 1$;
      Apply the discrete Fourier transform on z_{ij}, $\mathrm{i} \equiv \sqrt{-1}$;
      $f_{ij} = \sum_{t=1}^{m} z_{it} \mathrm{e}^{2\pi \mathrm{i}(t-1)j/m}$;
    end for
    for $j = 0$ to $m/2-1$ **do**
      Compute the modulus of f_{ij};
    end for
    $T_h = \sqrt{(\log \frac{1}{0.05}) m}$;
    Let N_{i1} be the number of $|f_{ij}|$ less than T_h;
  end for
  for $r = 0$ to $M-1$ **do**
    Count the number of N_{i1}/m belonging to the r-th interval, v_r;
  end for
  Compute a test statistic: $X^2 = \sum_{r=0}^{K} \frac{(v_r - M\pi_r)^2}{M\pi_r}$;
  Compute the P-value: $p = \text{igamc}(\frac{K}{2}, \frac{X^2(\text{obs})}{2})$;
  return p;
end if

Then we conduct the two-level test based on the new DFT test procedure. First perform the algorithm for N sample sequences $\{\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_N\}$ and get N P-values. Conduct the two-level test with the N P-values, the detail test procedure is shown in Subsection 2.2. When the passing proportion test and the uniformity test of the two-level test are all passed, the RNG that generates the input sequences is concluded to be good.

4 Experiment results

In this section, we first conduct some comparison experiments to evaluate the accuracy and efficiency of our method with other three methods: the official DFT test in NIST SP800-22 ($c = 4$) [5], DFT test corrected by Pareschi et al. ($c = 3.8$) [6], and DFT test corrected by Zhu et al. [17] (using Q-value to replace P-value in the uniformity test). Moreover, as our method effectively reduces the test time and the memory consumptions, we also carry out experiments for long sequences (10^8 and 10^9) test, which usually cannot be done by previous methods [5, 6, 17] on ordinary computers, due to the limitations of computation ability and memory capacity. The experiment result shows that our method has lower probability of making type I errors, the same ability to avoid type II errors and requires less test time

Table 4 Detail information of sequences

PRNG	Length	Number
AES-256	1000000	100000
	10000000	30000
SHA-512	1000000	100000
	10000000	30000

Table 5 Comparisons of passing proportion

Sequence length	Number	Lower limit	PRNG	DFT test	Passing ratio
1000000	100000	0.98906	AES-256	NIST ($c = 4$) [5]	**0.98790**
				NIST ($c = 3.8$) [6]	0.99010
				Ours ($m = 1000$)	0.98990
			SHA-512	NIST ($c = 4$) [5]	**0.98810**
				NIST ($c = 3.8$) [6]	0.98990
				Ours ($m = 1000$)	0.98997
10000000	30000	0.98828	AES-256	NIST ($c = 4$) [5]	**0.98760**
				NIST ($c = 3.8$) [6]	0.99000
				Ours ($m = 1000$)	0.99010
				Ours ($m = 10000$)	0.98960
			SHA-512	NIST ($c = 4$) [5]	**0.98863**
				NIST ($c = 3.8$) [6]	0.99050
				Ours ($m = 1000$)	0.98990
				Ours ($m = 10000$)	0.98950

and memory consumptions. So our method has higher reliability in two-level test and is more suitable in practical application scenarios.

4.1 Comparison experiments for occurrence of type I errors

We first evaluate our method's accuracy to avoid type I errors. We use known good PRNGs (AES-256 and SHA-512) to generate binary sequences with good randomness. The detail information of the output sequences is shown in Table 4. These sequences are used to perform the two-level tests using the four DFT methods described above. Note that the sequence numbers we choose are all larger than the upper threshold values in Table 1. As explained in Susbection 3.2, the larger than the upper threshold values the sequence number is, the higher the probability of errors occurring in two-level test will be. As presented in [6], the two-level test is relatively unreliable when the sequence number is larger than 10000 bits. To highlight the reliability advantages of our method against the other three methods, we choose larger sequence numbers to perform our experiments. Similarly, the sequence length in the comparison experiments is also large (10^6 and 10^7), as the error probability will increase along with the sequence length.

4.1.1 *Comparisons of passing proportion*

For the passing proportion test, the significance level α is set to 0.01. We conduct experiments of our DFT test method and the standard NIST DFT test using $c = 4$ and $c = 3.8$ separately. The passing ratios of sequences generated by PRNG AES-256 and SHA-512 are presented in Table 5.

If the passing ratio is larger than the lower limit, the passing proportion test is passed. We highlight the passing ratio values less than the lower limit in Table 5. Since the passing ratio value of 30000 SHA-512 output 10000000-bit sequences is only a little larger than lower limit, we also highlight this value. As shown in Table 5, when the sequences are long and the sequence number is large, the sequences do not pass the standard NIST DFT test with $c = 4$ [5], but pass the DFT test with $c = 3.8$ [6] and our method. This means that in the passing proportion test, our DFT test method is more reliable than the DFT test with $c = 4$ [5], thus reducing the probability of occurrence of type I errors. Also, our

Table 6 Comparisons of uniformity ($n = 10^6, N = 10^5$)

PRNG	DFT test	P/Q	P-value p_T
AES-256	NIST ($c = 4$) [5]	P-value	**0.000000**
		Q-value [17]	**0.000000**
	NIST ($c = 3.8$) [6]	P-value	**0.000000**
		Q-value [17]	0.012646
	Ours ($m = 1000$)		0.659908
SHA-512	NIST ($c = 4$) [5]	P-value	**0.000000**
		Q-value [17]	**0.000000**
	NIST ($c = 3.8$) [6]	P-value	**0.000000**
		Q-value [17]	0.043125
	Ours ($m = 1000$)		0.250616

method has consistent accuracy with the DFT test with $c = 3.8$ [6]. However, as shown in Subsections 4.3 and 4.4, our method is more efficient, and requires less memory consumptions than the DFT test with $c = 3.8$ [6], thus is more suitable for practical application scenarios.

4.1.2 *Comparisons of uniformity*

In the uniformity test, we use sequences whose length and number is 10^6 and 10^5. We set $k = 10$ and the significance level $\alpha_T = 0.0001$ according to the suggestion of NIST. Since Zhu et al. [17] replaced the P-values with Q-values to improve the reliability of the uniformity test, our comparison experiments include five methods: the NIST DFT test ($c = 4$) using P-value, the NIST DFT test ($c = 4$) using Q-value, the NIST DFT test ($c = 3.8$) using P-value, the NIST DFT test ($c = 3.8$) using Q-value, and our method. Then we can carry out the uniformity test with the P-values and Q-values and obtain the level-two P-values p_T of the five methods. The experiment results are shown in Table 6.

Note that the larger the p_T is, the more uniformity the P-values or Q-values will have. If the p_T is smaller than the significance level α_T, the uniformity test is not passed. So in Table 6, we highlight all the level-two P-values p_T smaller than 0.0001. According to the test results, the NIST DFT test using P-values with $c = 4$ or $c = 3.8$, the NIST DFT test using Q-values with $c = 4$ all fail to pass the uniformity test, although AES-256 and SHA-512 are known good PRNGs (thus occurring type I errors), while the NIST DFT test using Q-values with $c = 3.8$ and our method both pass the uniformity test. So our method is more reliable than the first three methods in the uniformity test.

To further exhibit our method's uniformity advantage, we compare the probability of Q-values with $c = 3.8$ [6] and P-values of our method in each sub-interval ($K = 10$, $n = 10^6$), and present the comparisons in Figure 2. From Figure 2, the uniformity of Q-values with $c = 3.8$ is not good enough. We prove that the distribution of N_1 has a deviation from the normal distribution in Subsection 3.1. So no matter P-values or Q-values computed by the statistic based on the normal distribution all have approximation errors. The larger the sequence number is, the more obvious the errors will be. Compared with P-value in the original NIST method, Q-value has reduced the computation error, but it cannot ensure the accuracy of the uniformity test when the sequence number is extremely large. In contrast, P-values of our method based on chi-square distribution have better uniformity, which is more consistent with the fact that AES-256 and SHA-512 are known good PRNGs. We conclude that our method is more reliable than other DFT test methods in the uniformity test.

4.2 Comparison experiments for occurrence of type II errors

As described above, our method has a lower probability to cause type I (false negative) errors than origin NIST DFT test [5] and its corrections [6, 17] do. In these subsections, we also conduct several contrast experiments to prove our method's correctness to avoid type II (false positive) errors, i.e., to ensure that RNGs with bad randomness cannot pass our DFT test. We use three different types of non-random sequences to perform DFT tests through our method, the standard NIST method ($c = 4$) and its correction work ($c = 3.8$) [6] separately. As NIST SP800-22 suggests, the test parameters we

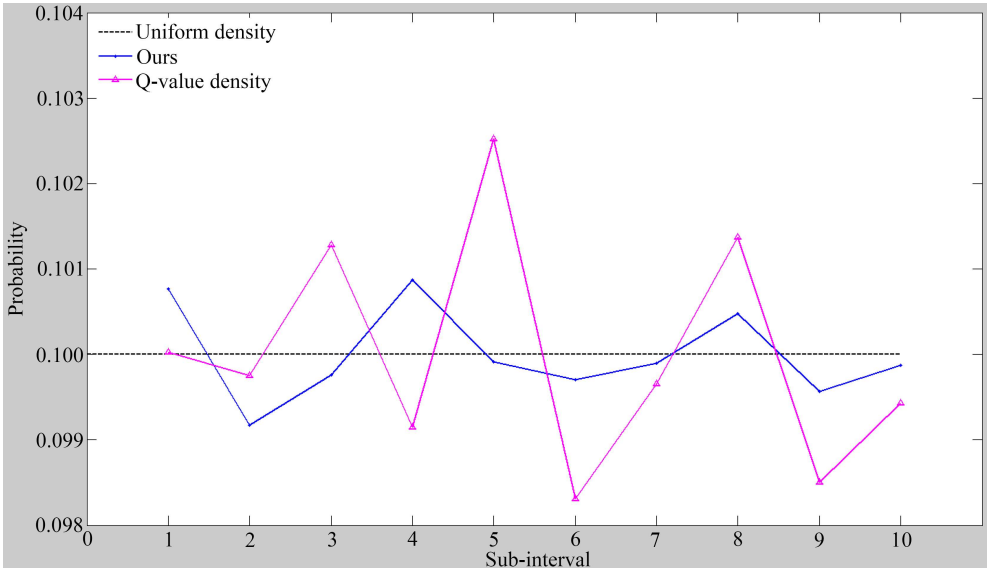

Figure 2 (Color online) Comparisons of probability in each sub-interval ($K = 10$, $n = 10^6$).

Table 7 Comparisons to detect type II errors for periodic sequences

Sequence length	Sequence period	Sequence number	Lower limit	DFT test method	Passing ratio
1000000	100000	1000	0.98056	NIST ($c = 4$)	**0.00000**
				NIST ($c = 3.8$)	**0.00000**
				Ours ($m = 1000$)	**0.00000**

used are: sequence length $n = 10^6$, sequence number $N = 1000$. The experiment results prove that our method has at least the same capability with the standard NIST method and its corrections to avoid type II errors.

Non-random sequences with periodic components. As the DFT method evaluates the randomness of a sequence by detecting periodic features in the bit series, we first evaluate the accuracy of our method using non-random sequences with periodic components. We use a loop method to generate sequences with specific length and period. For example, for length $n = 10^6$ and period $P = 10^5$, we first use one well-known good PRNG (e.g., AES-256 in our experiments) to generate a random subsequence of length 10^5, as a periodic component. Then we repeat the subsequence 10 times, link them together, and finally obtain a non-random 10^6 sequence whose period is 10^5. The comparison experiment results are given in Table 7. As the results show, the passing ratios of the three test methods are all lower than the lower limit. So all the sequences fail to pass the passing proportion test based on our method and other methods, thus proving that our method has at least the same accuracy with the standard NIST method to avoid type II errors for non-random RNGs with periodic features.

Non-random sequences based on linear congruence. We evaluate our method using the standard random function rand() in the C library. This function is based on linear congruence and is generally considered as a PRNG called LCG whose randomness is not good enough. The comparison experiment results using sequences of rand() are given in Table 8. As the results show, the passing ratios of the three test methods are all lower than the lower limit. So all the sequences fail to pass the passing proportion test of our method and the other methods, which is consistent with the fact that the randomness of LCG sequences is not good enough.

Non-random sequences based on inappropriate AES encryption. We also evaluate our method using inappropriate AES encryption as a PRNG. Due to the encryption properties of the AES algorithm, using inappropriate parameters will impact the security of the entire encryption procedure. So the randomness of the output ciphertext will not be good enough. We perform AES encryption on several normal files using insecure parameters, i.e., using a wrong S-Box whose outputs are all zero bits, an

Table 8 Comparisons to detect type II errors for linear congruence generator rand()

Sequence length	Sequence number	Lower limit	DFT test method	Passing ratio
1000000	1000	0.98056	NIST ($c = 4$)	**0.00000**
			NIST ($c = 3.8$)	**0.00000**
			Ours ($m = 1000$)	**0.00000**

Table 9 Comparisons to detect type II errors for inappropriate AES encryption

Sequence length	Sequence number	Lower limit	DFT test method	Passing ratio
1000000	1000	0.98056	NIST ($c = 4$)	**0.86600**
			NIST ($c = 3.8$)	**0.86800**
			Ours ($m = 1000$)	**0.85900**

Table 10 Comparisons of test time

Sequence length	DFT test	Test time (s)
1000000	NIST SP800-22 [5]	1.95
	Ours ($m = 1000$)	0.48
10000000	NIST SP800-22 [5]	16.88
	Ours ($m = 1000$)	2.278
	Ours ($m = 10000$)	2.424

incomplete encryption round number (2 rounds) and the ECB encryption mode. The output ciphertext sequences are used to conduct our comparison DFT tests. As the experiment results in Table 9 show, the passing ratios of the three test methods are all lower than the lower limit. So the sequences fail to pass the passing proportion test of both our method and the other methods, thus proving that our method has nearly the same accuracy with the standard NIST method to avoid type II errors for PRNG using inappropriate AES encryption.

Summary. To make our work more comprehensive and persuasive, we have conducted our comparison experiments to detect type II errors using different types of non-random sequences. The experiment results are all consistent with our conclusion, i.e., our method has nearly the same accuracy with the standard NIST method and its correction works to avoid type II errors for PRNG with poor randomness.

4.3 Comparisons of test efficiency

For sequences with length 10^6 and 10^7, we do the experiments using the standard NIST DFT test ($c = 4$) [5] and our DFT test method. Since the computation complexity of DFT test using $c = 3.8$ [6], DFT test using Q-value in the uniformity test [17] and the standard NIST DFT test ($c = 4$) [5] are almost the same, the test times of them are similar. We only record the test time of the standard NIST DFT test as a representation. The comparisons of test time for one sequence are shown in Table 10.

The test time of our new DFT test is significantly less than the other three methods. This is because the time complexity of DFT computation is non-linear. The test time will significantly increase along with the sequence length. The total time in our method to perform discrete Fourier transform to every divided subsequence separately is much smaller than performing one discrete Fourier transform to the whole sequence in other methods. For 10^6-bit sequences, the test efficiency is increased to about 4 times. For 10^7-bit sequences, the test efficiency is increased to about 7 times. So our new DFT test method could effectively improve the test efficiency of previous methods.

4.4 Experiments for long sequences (10^8 and 10^9 bits)

As the DFT algorithm requires high memory capacity, previous DFT test methods have a high possibility to crash for long sequence test, due to the memory limitations of ordinary computers. In fact, we find that the official DFT test program of NIST SP800-22 and its corrections will always work abnormally on our experiment computer (equipped with Intel i7 CPU and 8 GB RAM) when testing long sequences whose length is 10^8 or more bits. By analysing the source code of the official NIST DFT test program, when

Table 11 Experiment results for long sequences

PRNG	Sequence length	Number	Lower limit	m	Passing ratio	P-value p_T	Test time (s)
AES-256	100000000	1000	0.98056	1000	0.99400	0.026948	20.14
				10000	0.98900	0.841226	22.515
				100000	0.98800	0.189625	25.74
	1000000000	200	0.96889	1000	0.98500	0.176657	230.7
				10000	0.99000	0.554420	234.4
				100000	0.98000	0.911413	271.5
SHA-512	100000000	1000	0.98056	1000	0.99200	0.191687	20.14
				10000	0.98200	0.686955	22.515
				100000	0.98900	0.340858	25.74
	1000000000	200	0.96889	1000	0.98500	0.605916	230.7
				10000	0.99500	0.025193	234.4
				100000	0.99000	0.930026	271.5

the sequence length is 10^8, about 1.3 GB memory is required to store the sequence and the intermediate states of the DFT computations, which is much larger than the memory capacity common commercial operating systems (e.g., Windows and Linux) could offer to one single program on ordinary computers. This phenomenon will impact the practicability of previous methods, because for evaluating the quality of a RNG, the longer the sequence generated by the RNG in one single test is, the more accurate the test result will be. In contrast, our method effectively reduces the memory consumptions in one DFT test, by dividing an sequence into several subsequences. So our method could test longer sequences than previous methods on devices with the same computing capabilities.

To prove this, we conduct DFT tests of our method for sequences with length 10^8 and 10^9 bits on the same experiment computer, using different subsequence lengths. The test results are given in Table 11, which shows that long sequences can be tested normally by our method. The passing ratios are larger than the lower limit and the P-values of uniformity test are bigger than the significance level $\alpha_T = 0.0001$, which means that the sequences generated by PRNG AES-256 and SHA-512 passed the test. This result is consistent with the recognized good randomness of the PRNG AES-256 and SHA-512, so our method can ensure the accuracy of the test. Furthermore, the test times are all in the acceptable range. In conclusion, our method has stronger capability to test long sequences, hence is more suitable for practical application scenarios.

5 Conclusion

There exist accuracy issues in previous DFT random test methods, leading to a high probability of making type I (false negative) errors when the tested sequences are long or the sequence number is large. The large test time and high memory consumptions of the complex DFT test algorithm also seriously affect its practicability, which is not considered and solved in previous studies. In this paper, we propose a new DFT test method for long sequences (10^6 or more bits). Different from previous studies focusing on making the distribution of statistic closer to the normal distribution, we reconstruct the statistic to follow the chi-square distribution by dividing a long sequence into short sub-sequences. Our experiment results show that our new DFT test method has higher reliability in two-level test, lower probability of making type I errors and nearly the same accuracy with previous methods to avoid type II errors. The test time and memory consumptions are also effectively reduced, making our method more suitable for practical application scenarios, such as fast test for very long sequences on ordinary computers. In the future, we will give more choice of subsequences length, improve our construction algorithm to reduce the probability of making type II errors and improve the capability of our DFT test method.

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2018YFB-0904900), National Cryptography Development Fund (Grant Nos. MMJJ20170214, MMJJ20170211).

References

1 Sowmya S, Sathyanarayana S V. Symmetric key image encryption scheme with key sequences derived from random sequence of cyclic elliptic curve points over GF(p). In: Proceedings of International Conference on Contemporary Computing and Informatics, 2015. 1345–1350
2 Sulak F, Doğanaksoy A, Ege B, et al. Evaluation of randomness test results for short sequences. In: Proceedings of the 6th International Conference on Sequences and Their Applications, 2010
3 Hellekalek P, Wegenkittl S. Empirical evidence concerning AES. ACM Trans Model Comput Simul, 2003, 13: 322–333
4 Yin R M, Wang J, Yuan J, et al. Weak key analysis for chaotic cipher based on randomness properties. Sci China Inf Sci, 2012, 55: 1162–1171
5 Rukhin A, Soto J, Nechvatal J, et al. SP 800-22 Rev. 1a. A statistical test suite for random and pseudorandom number generators for cryptographic applications. Appl Phys Lett, 2010, 22: 1645–179
6 Pareschi F, Rovatti R, Setti G. On statistical tests for randomness included in the NIST SP800-22 test suite and based on the binomial distribution. IEEE Trans Inform Forensic Secur, 2012, 7: 491–505
7 Pareschi F, Rovatti R, Setti G. Second-level NIST randomness tests for improving test reliability. In: Proceedings of IEEE International Symposium on Circuits and Systems, 2007. 1437–1440
8 Hamano K, Kaneko T. Correction of overlapping template matching test included in NIST randomness test suite. IEICE Trans Fund Electron Commun Comput Sci, 2007, 90: 1788–1792
9 Hamano K. Correction of "test for the longest run of ones in a block" included in NIST randomness test suite. IEICE Tech Rep, 2007, 107: 17–21
10 Chen M H, Fan L M, Gao S, et al. Corrected runs distribution test for pseudorandom number generators. Electron Lett, 2016, 52: 281–283
11 Chen M H, Chen H, Fan L M, et al. Templates selection in non-overlapping template matching test. Electron Lett, 2016, 52: 1533–1535
12 Sýs M, Říha Z, Matyáš V. Algorithm 970: optimizing the NIST statistical test suite and the berlekamp-massey algorithm. ACM Trans Math Softw, 2017, 43: 27
13 Huang J L, Lai X J. Measuring random tests by conditional entropy and optimal execution order. In: Proceedings of International Conference on Trusted Systems, 2010. 148–159
14 Fan L M, Chen H, Gao S. A general method to evaluate the correlation of randomness tests. In: Proceedings of International Workshop on Information Security Applications, 2013. 52–62
15 Sulak F, Uğuz M, Koçak O, et al. On the independence of statistical randomness tests included in the NIST test suite. Turk J Electric Eng Comput Sci, 2017, 25: 3673–3683
16 Pareschi F, Rovatti R, Setti G. Second-level testing revisited and applications to NIST SP800-22. In: Proceedings of the 18th European Conference on Circuit Theory and Design, 2007. 627–630
17 Zhu S Y, Ma Y, Lin J Q, et al. More powerful and reliable second-level statistical randomness tests for NIST SP 800-22. In: Proceedings of International Conference on the Theory and Application of Cryptology and Information Security, 2016
18 Hamano K, Satoh F, Ishikawa M. Randomness Test Using Discrete Fourier Transform. Technical Report 6841, 2003
19 Hamano K. The distribution of the spectrum for the discrete Fourier transform test included in SP800-22. IEICE Trans Fund Electron Commun Comput Sci, 2005, 88: 67–73
20 Kim S J, Umeno K, Hasegawa A. Corrections of the NIST statistical test suite for randomness. 2004. https://eprint.iacr.org/2004/018.pdf
21 Daemen J, Rijmen V. The Design of Rijndael: AES – the Advanced Encryption Standard. Berlin: Springer, 2002
22 U.S. Department of Commerce. Secure Hash Standard - SHS: Federal Information Processing Standards Publication 180-4. Charlestone: CreateSpace Independent Publishing Platform, 2012