Quantitative Analysis of the Avalanche Effect of Different Hash Functions Based on Chi-Squared and T-Tests

Liyuan Zhao
Applied Statistic, Anhui University, Hefei, Anhui, China

Abstract: As the demand for information security continues to escalate in modern society, the avalanche effect of hash functions has emerged as a critical metric for evaluating their resistance to cryptographic attacks. Nevertheless, conventional analyses of the avalanche which lacking systematic quantitative assessment and robust statistical underpinning effect predominantly rely on qualitative observations, those impedes objective and reproducible comparisons across different algorithms.This paper constructs a quantitative analysis framework for the avalanche effect of hash functions, employing chi-squared tests and paired sample t-tests. This framework comprehensively assesses hash algorithm performance from two perspectives: output distribution uniformity and output variability. Employing a control variate methodology,This experiment constructs statistical samples by applying two independent, random-position bit flips to various types of input strings, thereby generating perturbed outputs and calculating the Hamming distance between the two perturbed inputs.The effectiveness of the framework is validated by statistically testing classical hash algorithms such as MD5, SHA-256, and SHA-1. The experimental results were found to align highly with the known security characteristics of these algorithms.In summary,this framework for avalanche effect presented in this paper holds significant theoretical value and practical implications for the design and security comparison of cryptographic algorithms.

Keywords: Hash Function; Avalanche Effect; Chi-Squared Test; t-Test

1. Introduction

In the realms of cryptography and information security, hash functions, acting as deterministic information digest functions, are pivotal. They possess the capability to map input data of arbitrary length to a fixed-length output value, commonly referred to as the hash digest. In addition,the primary utility of hash functions lies in their provision of efficient identifiers for data storage and retrieval, thereby substantially enhancing system processing efficiency and safeguarding data integrity [1].This characteristic leads to its widespread application in core security modules across various applications, including data integrity checks, digital signatures, and authentication mechanisms. There is a fundamental requirement for cryptographically strong hash functions that their outputs should exhibit ideal random characteristics. However, due to the mathematical impossibility of rigorously proving the absolute randomness of finite bit sequences, researchers resort to statistical testing methods as a traditional means of evaluating cryptographic primitives for quantitative evaluation [2]. Diverse statistical tests are employed to detect various potential defects within sample sequences. This capability, in turn, indirectly assesses the security performance of hash functions [3]. Among the numerous security properties of hash functions, the avalanche effect serves as a crucial indicator for measuring an algorithm’s obfuscation and diffusion characteristics [4]. The avalanche effect refers to the phenomenon where a minor change in the input, such as a single bit flip, which leads to a significant and unpredictable alteration in the output hash value. Ideally, each bit in the output should change with a probability close to 50% [5]. The more pronounced the avalanche effect of a hash algorithm, the greater its inherent randomness [6]. However, traditional methods for analyzing avalanche effect are often confined to qualitative descriptions or intuitive comparisons, which impede objective, cross-algorithmic comparative analysis. Consequently, to facilitate the selection of more suitable hash functions, it is imperative to refine the existing avalanche effect evaluation

system for a more comprehensive and precise measurement of each algorithm's performance in this regard.

Motivated by the need to overcome the limitations of traditional avalanche effect evaluation methods, this study proposes a quantitative analysis and assessment framework for avalanche effect that synergistically integrates chi-squared tests and t-tests. Specifically, it employs chi-squared tests to evaluate output uniformity and paired sample t-tests to assess output variability. The effectiveness of this framework is then validated through a controlled experimental design. This paper not only provides a more systematic theoretical basis and experimental support for the security analysis of hash functions, but also enables statistical testing methods to complementarily characterize the avalanche effect features of hash algorithms.

This study possesses the following research significance. Firstly, we have constructed a quantitative evaluation framework for the avalanche effect of hash functions overcomes the limitations of traditional qualitative analysis and provides a reproducible and comparable assessment paradigm. Secondly, a multi-dimensional controlled variable experimental design enhances the reliability and generalizability of the results, enabling a more refined evaluation of their avalanche characteristics. Thirdly, we not merely employed the chi-squared test to verify the uniformity of bit distribution in hash outputs, but also innovatively utilized the paired samples t-test to quantify the output stability of algorithms. This research deeply integrates statistical methods with cryptography and expands the theoretical toolkit for cryptographic security analysis. Fourthly, through experimental comparisons of several classical hash algorithms, we have demonstrated that our framework can effectively differentiate algorithm performance. Moreover, it offers a more nuanced quantitative understanding of the security status of these classic algorithms.

In conclusion, this study provides a scientific and objective basis for the quantitative analysis of hash function security. It carries significant theoretical value and practical reference importance for the design, improvement, and standardization of cryptographic algorithms.

2. Related Work

2.1 Limitations of Traditional Avalanche Effect Analysis Methods

Existing research on the avalanche effect of hash functions exhibits notable limitations. While studies, such as those by Wang Gan and Zhang Wenying [7], have employed statistical measures like variance to validate the avalanche effect of algorithms like SHE-3, their analyses remain largely descriptive. Crucially, these approaches fail to conduct rigorous hypothesis testing to confirm the consistency between observed and theoretical distributions.Li Bo and Liu Ping [3] conducted χ^2 tests on SHA-256 to assess its randomness and avalanche effect. Their findings revealed anomalous fluctuations in specific rounds of the algorithm. Notwithstanding, their conclusions were limited to a binary "pass/fail" outcome, lacking in-depth analysis of the causes of these outliers and their significance levels. Zhou Lu and Chen Qin et al [8]. developed a statistical performance testing method for block ciphers, centered on avalanche effect testing. Nevertheless, this method exhibits insufficient adaptability for hash functions, and their application of the χ^2 test is confined to qualitative binomial distribution fitting, failing to provide quantitative analysis for continuous data.

Markku-Juhani O. Saarinen [9] highlighted that traditional avalanche effect testing inadequately captures the pseudorandomness of hash functions, proposing a d-uninominal test based on algebraic normal forms as a supplement. Nonetheless, his methodology is characterized by high computational complexity, hindering its rapid deployment and standardized evaluation in engineering practice. Subsequently, R. Damasevicius et al [10]. compared the power consumption and quality of 17 hash functions, employing the test to assess distribution uniformity and utilizing Hamming distance to compute statistics for measuring diffusion performance. Be that as it may, their analysis remained confined to performance ranking, failing to delve into the multifaceted influences on avalanche effect stability. Furthermore, their work lacked a quantitative examination of the variability in results from multiple independent experiments.

In summary, traditional methods for analyzing avalanche effect suffer from two primary limitations. Firstly, the evaluative dimensions are restricted; while many studies solely assess

whether the average number of changed bits approaches the ideal value, they neglect the variability and distributional uniformity of these changes. Secondly, most existing research remains at the level of descriptive statistics, lacking rigorous statistical inference to ascertain the degree of concordance between observed results and theoretical expectations. Secondly, the quantitative methods employed are often insufficiently rigorous. While some studies incorporate the chi-squared (χ^2) test, their analysis frequently remains at a binary "pass/fail" conclusion. This approach fails to provide a stringent interpretation of abnormal fluctuations in data or perform significant analyses, thereby hindering the exploration of underlying causal factors and the effective differentiation of performance disparities among various hash functions.

2.2 Limitations of Statistical Methods in Cryptography

In contrast, some research has begun to incorporate statistical inference tools. However, most existing methods focus on assessing overall randomness rather than providing precise measurements of the avalanche effect itself. For example, Sofi et al. [11], in the context of Bitcoin's Proof-of-Work (PoW) consensus mechanism, compared the performance of various hash functions. They employed average Hamming distance to validate the avalanche effect and utilized the Mann-Whitney U test to analyze differences in mining times. Nevertheless, their core validation remained at the level of computing the average number of flipped bits. Upadhyay et al. [6] conducted a study on the avalanche effect of 16 hash functions and their applications. They introduced Multi-Criteria Decision Making (MCDM) and the NIST Test Suite to evaluate the randomness of outputs. Notwithstanding, the core determination of the avalanche effect still relied on descriptive statistics and an intuitive comparison with the theoretical 50% change rate, lacking rigorous probabilistic distributional inference.

While other studies have incorporated statistical testing frameworks, they primarily serve to rank algorithm performance or are geared towards other cryptographic components such as block ciphers. Palukha and Kharin [12] developed testing methods for random number generators based on Renyi and Tsallis entropy statistics, deriving asymptotic distributions and analyzing test power. In the spite of this, their focus was on the overall entropy characteristics of generated sequences, rather than performing specialized statistical inference on output differences under single-bit perturbations.The CryptoStat test suite, designed by UKaminsky et al. [13], employs the Bayes factor from Bayesian model selection to aggregate the results of multiple independent tests, yielding an overall assessment of randomness. yet this approach circumvents the complexities associated with p-value interpretation, and the core of the Bayes factor test lies in model selection and does not quantitatively characterize output bit changes resulting from minute input perturbations.

In summation, while statistical methods have permeated various levels of cryptographic assessment, existing work still presents two discernible shortcomings. Firstly, the experimental design is often unidimensional, typically employing homogeneous inputs of fixed length. This lack of systematic control and comparative analysis across variables hinders a comprehensive evaluation of the avalanche effect's stability under diverse application scenarios. Secondly, current methodologies lack a holistic framework for assessing hash function characteristics, thereby failing to provide a precise quantitative description of avalanche effect performance.

2.3 Uniformity of Hash Functions

One of the fundamental security properties of hash functions is pseudorandomness, which implies that the output of a hash function, given an unknown input, should be indistinguishable from that of a pseudorandom function [2]. This characteristic indicates that any statistical test capable of detecting an uneven output distribution can serve as a distinguisher, thus demonstrating the presence of security vulnerabilities in the hash function. Therefore, the requirement for pseudorandomness is essentially equivalent to the demand for uniformity in output, and it is easy to find out that an ideally random function in the output space is uniformly distributed [14]. When a hash function reveals statistical regularities, cryptanalysts can extract exploitable information, therewith undermining its security in applications such as message authentication and digital signatures.

2.3.1 Hamming distance

Hamming distance [15] is a metric that quantifies the differences between two strings of equal length, defined as the number of positions at which the corresponding characters differ. In the analysis of the avalanche effect of hash functions, Hamming distance is widely employed to measure the degree of difference between the hash values corresponding to the original input and the perturbed input [16]. Specifically, for the original input x and the perturbed input x′, with their respective hash values denoted as H(x) and H(x′), the Hamming distanced dH reflects the extent to which input perturbations influence changes at the bit level of the output.

In an ideal scenario, a hash function should exhibit a strong avalanche effect, whereby a slight perturbation in the input should result in an output bit change with approximately 50% probability. From a statistical perspective, this process can be abstracted as an observation of the Hamming distance as a random variable. Each perturbation experiment can be viewed as a sampling event, with the resulting sequence of Hamming distances essentially forming a set of samples from this random process. Ideally, for a hash function with an output length of n bits, the Hamming distance should follow a binomial distribution B(n,1/2), with theoretical mean and variance equal to n/2 and n/4, respectively [17].

2.3.2 Uniformity of hash functions in the context of the avalanche effect

In contemporary cryptographic hash function design, the output length n is generally fixed. According to the De Moivre-Laplace central limit theorem from probability theory [18], when the number of trials n in a binomial distribution is sufficiently large and the success probability p=0.5 is not near the boundaries, the discrete binomial distribution B(n,0.5) will closely resemble the continuous normal distribution N(n/2,n/4). For widely used hash functions such as MD5, SHA-1, or SHA-256, the output length n satisfies the conditions necessary for the central limit theorem to apply to large samples. This approximation is a key mathematical premise for subsequent statistical inference: it suggests that for a large number of repeated avalanche experiments, the sampling distribution of the observed mean Hamming distance X bar will also conform to a normal distribution. Therefore, Hamming distance is not only useful for characterizing the degree of variation between individual input-output pairs but also serves as a foundational variable for statistical testing, allowing for the evaluation of output stability and variability of the same hash algorithm when comparing original and perturbed inputs.

Jaiswal et al. [19] proposed a robust output encoding method based on Hamming distance, which optimizes target label encoding by maximizing the Hamming distance between confusable categories and constructs a statistical testing framework to evaluate output stability under different inputs. However, their application scenario pertains to classification tasks in machine learning, which differs from the core issues addressed in the analysis of the avalanche effect of hash algorithms. Based on the aforementioned research and analysis, the evaluation of the avalanche effect of hash functions should delve into a quantitative analysis of the distribution characteristics of output data using Hamming distance.

2.4 The Principle of Chi-Square Test and Its Applicability in Uniformity Analysis

The chi-square distribution arises from the sum of the squares of several independent random variables that follow a standard normal distribution. It is widely employed for the statistical analysis of contingency table data. The chi-square test for contingency tables primarily serves to assess the correlation between two or more discrete random variables. This is typically achieved by evaluating the validity of the independence hypothesis concerning the variables, thereby demonstrating the existence of a certain correlation among them. In conducting this test, the constructed statistic is derived by comparing the observed frequencies with the expected frequencies. The asymptotic distribution of this statistic conforms to the chi-square distribution [20].

Based on the aforementioned principles, the chi-squared test is applicable for assessing the uniformity of the output distribution of hash functions. In the experiments, extensive avalanche effect testing must be conducted using a substantial number of random inputs. The observed frequency distribution of Hamming distances is categorized into several intervals, thereby constructing an observed frequency distribution table, which is then compared with the theoretical expected frequencies. In the context of avalanche effect analysis for hash functions, the null hypothesis (H0) for the

chi-squared test posits that the Hamming distances resulting from single-bit perturbations of the hash function inputs follow a uniform distribution. This implies that there are no significant differences between the observed frequencies and the theoretical expected frequencies under uniform distribution. The test statistic is calculated as follows:

$$\chi^2=\sum_{k=0}^{n}\frac{(O_k-E_k)^2}{E_k} \quad (1)$$

In this formula (1), represents the observed frequency, while denotes the expected frequency.

If the calculated $\chi 2$ value is less than the critical value corresponding to the specified significance level, or if the associated p-value exceeds 0.05, the null hypothesis cannot be rejected. This outcome indicates that the distribution of Hamming distances in the hash function outputs is uniform, suggesting that the avalanche effect demonstrates satisfactory uniformity. Conversely, if the p-value is less than 0.05, the null hypothesis is rejected, signifying a significant deviation between the observed distribution of Hamming distances and the theoretical uniform distribution. This implies that the changes in output bits are uneven, indicating inadequate uniformity in the avalanche effect.

2.5 The Principle of Paired Sample t-Test and Its Applicability in Stability Analysis

Paired samples can be utilized to describe two characteristic aspects of the same entity or to reflect the different attributes of the same entity under two distinct conditions. The paired sample t-test is based on the inference of whether there is a significant difference between the means of two populations derived from paired samples, with the assumption that the differences between the samples follow a normal distribution or an approximately normal distribution [21]. In the analysis of the avalanche effect in hash functions, the differences in hash values resulting from single-bit perturbations at different positions of the same original input form paired samples. Hence, this test is particularly suitable for such analyses.

By examining the significance of the differences in Hamming distances resulting from single-bit perturbations at different positions within a hash function, we can quantify the consistency of the hash function's output in response to input disturbances.

$$t=\frac{\bar{d}}{s_d/\sqrt{m}} \quad (2)$$

The test statistic is calculated as follows: where d represents the mean of the paired differences, denotes the standard deviation of the paired differences, and n is the number of paired samples according to the formula (2).

The null hypothesis (H0) posits that the mean of the differences in Hamming distances, after perturbing the same original input at different positions with single-bit changes, is zero. This indicates that the differences in perturbation locations have no significant effect on the output. If the p-value is less than 0.05, we reject the null hypothesis, suggesting that the differences in perturbation locations significantly affect the Hamming distances, indicating inconsistent responses from the hash function and poor stability in the avalanche effect. Conversely, if the p-value exceeds 0.05, we cannot reject the null hypothesis, implying that there are no significant differences in Hamming distances across different perturbation locations, thus indicating good stability of the avalanche effect.

2.6 Summary

Section 2.1 of this chapter highlights the limitations of existing methods for evaluating the avalanche effect, including a singular evaluation dimension, insufficient rigor in quantification methods, and incomplete experimental designs, which make it difficult to meet the demands for detailed analysis. Consequently, Section 2.2 introduces statistical tests to fill the theoretical gap in the assessment of the avalanche effect. Building on this foundation, Section 2.3 elaborates on the importance of hash function uniformity as a core security attribute, noting its direct impact on pseudorandomness and resistance to attacks, while further exploring the specific characterization of uniformity under the avalanche effect. Hamming distance is introduced to represent the stability and variability of outputs under perturbed inputs. The theoretical applicability of the chi-squared test and the paired sample t-test in avalanche effect analysis is also discussed: Section 2.4 explains how the chi-squared test evaluates distribution uniformity by comparing observed frequencies of Hamming distances with theoretical expected frequencies; Section 2.5 details how the paired sample t-test measures output stability and variability by analyzing the

significance of mean differences in Hamming distances. In summary, this chapter lays a solid theoretical foundation and statistical basis for subsequent experimental design.

3. Research Methodology

3.1 Experimental Design

To systematically evaluate the statistical properties of different hash functions under avalanche effect conditions, this study devised the following experimental design. The entire experimental process can be divided into three phases: the preparation and data generation phase, the statistical testing phase, and the result integration and validation phase, as detailed in the flowchart presented in Figure 1.

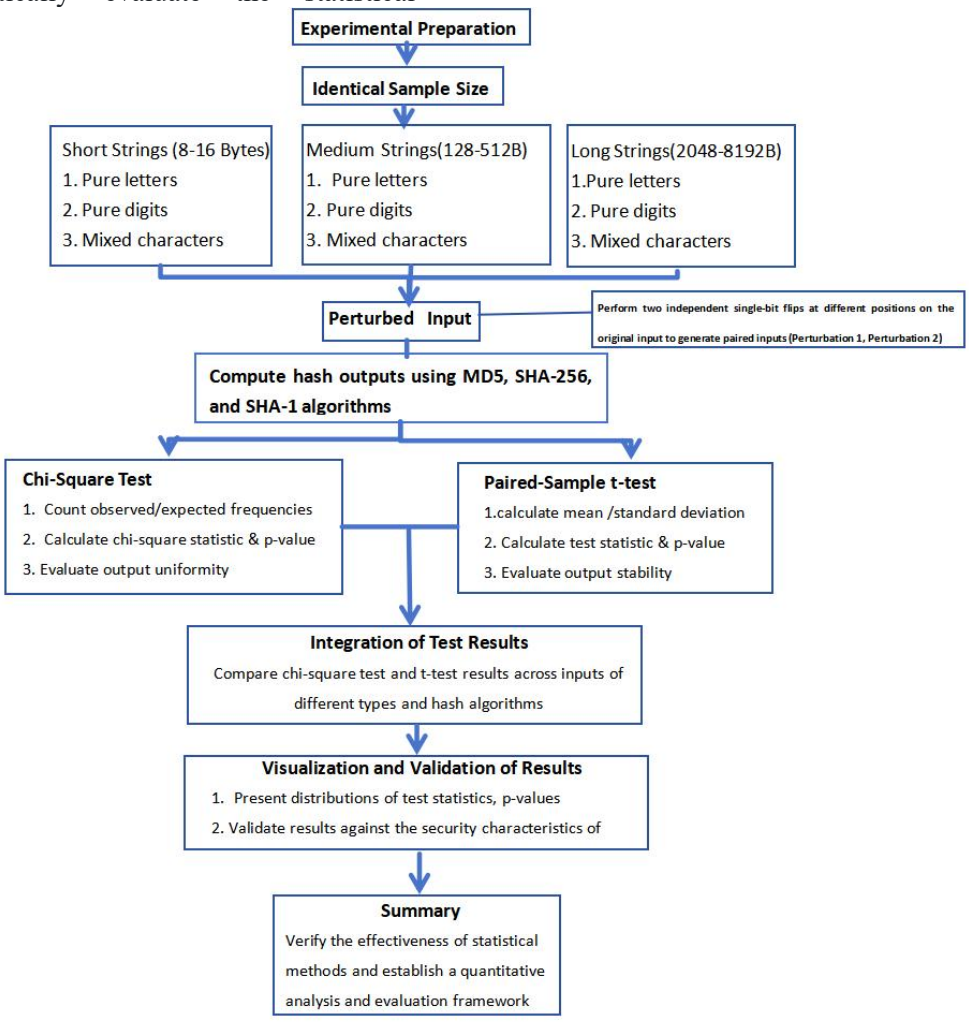

Figure 1. Experimental Flowchart for Quantitatively Analyzing Hash Algorithm Avalanche Effect Using Chi-Square and t-Tests

3.2 Selection of Hash Algorithms

3.2.1 Choice of research subjects

This experiment selects MD5, SHA-1, and SHA-256 as the research subjects. MD5 (1992), SHA-1 (1995), and SHA-256 (2001) represent the technological evolution of hash functions from widespread adoption to gradual deprecation and current standardization. Architecturally, all three are based on the Merkle-Damgård construction[1]. The subjects of study possess a basis for comparability. Although this architecture is susceptible to length extension attacks, its homologous structure ensures that when evaluating output uniformity under the avalanche effect in experiments, interference from underlying architectural differences can be mitigated. This allows statistical tests to focus on a fine-grained comparison of the internal diffusion logic among algorithms. Furthermore, the disparities in their output lengths naturally facilitate an examination of how output space

size influences the avalanche effect.

3.2.2 Known security

As widely studied standard hash functions in the field of cryptography, MD5, SHA-1, and SHA-256 have undergone extensive theoretical analysis and practical validation regarding their security properties. MD5 was once one of the most extensively used hash functions, with its output standardized to a 128-bit (16-byte) hash value. However, cryptographic researchers have identified algorithmic flaws, specifically "weak collision" vulnerabilities, which fundamentally undermine MD5's security. MD5's randomness is also considered to be slightly inferior to that of SHA-1 [2], SHA-1, drawing inspiration from the design principles of MD4 and MD5, generates a 160-bit (20-byte) hash value. However, cryptanalysts have identified fundamental weaknesses in its collision resistance. In contrast, SHA-256 remains widely regarded as secure and is extensively employed to underpin the security requirements of modern cryptographic systems [1]. As a representative algorithm of the SHA-2 family, SHA-256 produces a 256-bit hash value. Following the experimental workflow illustrated in Figure 1, the known security characteristics of these three algorithms are validated. The established security profiles of these three algorithms provide a clear reference gradient from "insecure to relatively secure to secure" for our experimental design. A strong correlation between the experimental outcomes and their known security properties would substantially corroborate the efficacy and reliability of our proposed quantitative analysis framework.

3.3 Controlled Variable Experimental Design

3.3.1 Sample size determination

Given that distinct input types, such as alphabetic characters, numerical digits, mixed alphanumeric strings, and special characters, can exert differential influences on the output characteristics of hash algorithms, a controlled experimental approach is adopted [22]. To ensure the validity of subsequent statistical tests, this study constructs input samples of three distinct types using a random number generator: purely alphabetic strings (e.g., "jxkqz"); purely numerical strings (e.g., "12345"); and mixed strings incorporating alphanumeric and special characters (e.g., "a1b2c3!@"). The sample size for each input type is set to be greater than 30. This sampling quantity is specifically chosen to satisfy the requirements of the Central Limit Theorem, which posits that the sampling distribution of the test statistic will approximate a normal distribution for sufficiently large sample sizes, thereby ensuring the robustness of the t-test.

3.3.2 Input length setting

To investigate the response characteristics of hash functions to inputs of varying lengths, three distinct length categories were established: short strings (8-16 bytes), medium strings (128-512 bytes), and long strings (2048-8192 bytes). This was performed while maintaining consistency in input types. Compared to existing research which often suffers from limitations such as singular input types, insufficient sample sizes, and arbitrary perturbation methods, the selection of these three length levels aims to balance generality with the consideration of how different hash algorithms' internal structures react to input length variations. This approach provides a unified experimental foundation for subsequent comparative analysis. Adhering to the principle of controlled variables, each "type-length" combination was processed to ensure diversity in input types, adequacy of sample sizes, and controllability of perturbation methods. This meticulous approach guarantees the rigor of statistical analysis and the reproducibility of the results.

The three input length settings are designed to systematically investigate the influence of input length on avalanche effect performance. Short strings are employed to examine the hash function's behavior when input length approaches the lower bound of hash output length. Medium strings correspond to typical input lengths in conventional application scenarios, reflecting the hash function's avalanche performance under common usage conditions. Long strings are utilized to evaluate the hash function's diffusion capability and stability when faced with large-scale data inputs.

3.4 Generation of Perturbed Inputs

For each set of original input samples under identical "string length-string type" configurations, two independent random single-bit flips at distinct positions are performed to generate two sets of perturbed inputs. Specifically: from the same original input, a random bit position is first selected for flipping, yielding the first perturbed input (Perturbation 1). Subsequently, without flipping the already modified bit position, a second distinct random

bit position is chosen for another flip, resulting in the second perturbed input (Perturbation 2). The perturbed inputs are both generated from the same original input, differing only by a single bit flip at a distinct position. For instance, if the original input is “abcde”, flipping the 3rd bit in its binary representation yields perturbed input 1 (“abdde”), and subsequently flipping the 4th bit produces perturbed input 2 (“abcle”). This methodology ensures the independence and pairwise nature of the perturbed inputs, thereby satisfying the sample requirements for subsequent statistical tests. This design effectively captures and quantifies the output variability of the same original input under different perturbation conditions. Each original input sample, along with its two corresponding perturbed inputs (perturbed 1 and perturbed 2), is fed into the MD5, SHA-1, and SHA-256 hash algorithms, respectively, to compute their respective hash outputs. For any given original input sample, three sets of hash outputs are generated: the original hash value (H0), the perturbed 1 hash value (H1), and the perturbed 2 hash value (H2). To meet the structural requirements for subsequent statistical analyses, H1 and H2 were paired to conduct a paired sample t-test, aimed at evaluating the stability of hash outputs derived from the same original input under varying perturbation positions. Additionally, the Hamming distance distributions for either H1 or H2 were utilized as observed samples for the chi-square test, which assesses the degree of deviation of the perturbed outputs from the theoretical uniform distribution.

4. Experimental Results

4.1 Data Collection

In order to quantitatively assess the uniformity of the avalanche effect distribution and the output stability of different hash algorithms, this study conducted multiple sets of independent repeated experiments based on the experimental design outlined in Chapter 3. By executing two independent single-bit flips at different positions on input samples of various types and lengths, perturbed samples were generated, and the corresponding Hamming distances of the hash outputs were calculated, thereby constructing a dataset that meets the requirements for statistical testing.Building upon this foundation, the experimental data were analyzed using both chi-square tests and paired sample t-tests, resulting in the computation of test statistics and p-values for each algorithm. This data serves as a basis for the subsequent quantitative comparative analysis of the avalanche effect. This section will present the core experimental data through three tables: Table 1: p-value data from the chi-square test for three hashing algorithms, Table 2: p-value data from the t-test for different hashing algorithms, and Table 3: t-value data for various hashing algorithms. These tables will provide a clear representation of the quantifiable performance of each algorithm in terms of uniformity and stability.

Table 1. p-value Data from the Chi-Square Test for Three Hashing Algorithms

Hash Algorithm	String Type	Length Type	Chi-square Test p-value	Average p-value	Chi-square Test Conclusion	Security Level	Uniformity Performance
MD5	Pure digits	Short	0.4844	0.2804	Not significant	Low	Average
	Pure digits	Medium	0.339	0.2804	Not significant		
	Pure digits	Long	0.2577	0.2804	Not significant		
	Pure letters	Short	0.6923	0.2804	Not significant		
	Pure letters	Medium	0.1571	0.2804	Not significant		
	Pure letters	Long	0.0658	0.2804	Not significant		
	Mixed chars	Short	0.2099	0.2804	Not significant		
	Mixed chars	Medium	0.1606	0.2804	Not significant		
	Mixed chars	Long	0.1564	0.2804	Not significant		
SHA-1	Pure digits	Short	0.7728	0.6244	Not significant	Medium	Good
	Pure digits	Medium	0.2452	0.6244	Not significant		
	Pure digits	Long	0.6584	0.6244	Not significant		
	Pure letters	Short	0.5902	0.6244	Not significant		
	Pure letters	Medium	0.777	0.6244	Not significant		
	Pure letters	Long	1	0.6244	Not significant		
	Mixed chars	Short	0.3105	0.6244	Not significant		

	Mixed chars	Medium	1	0.6244	Not significant		
	Mixed chars	Long	0.2651	0.6244	Not significant		
	Pure digits	Short	1	0.66	Not significant		
SHA-256	Pure digits	Medium	0.9661	0.66	Not significant		
	Pure digits	Long	0.9977	0.66	Not significant		
	Pure letters	Short	1	0.66	Not significant		
	Pure letters	Medium	0.1483	0.66	Not significant	Good	Optimal
	Pure letters	Long	1	0.66	Not significant		
	Mixed chars	Short	0.0866	0.66	Not significant		
	Mixed chars	Medium	0.066	0.66	Not significant		
	Mixed chars	Long	0.6754	0.66	Not significant		

Table 2. p-value Data from the t-Test for Different Hash Algorithms

Hash Algorithm	String Type	Length Type	T-test p-value	Average p-value	Standard Deviation	Stability Rating	T-Test Conclusion
MD5	Pure digits	Short	0.0156	0.491	0.287	Poor	Significant
	Pure digits	Medium	0.8866				Not significant
	Pure digits	Long	0.4543				Not significant
	Pure letters	Short	0.5586				Not significant
	Pure letters	Medium	0.4122				Not significant
	Pure letters	Long	0.6351				Not significant
	Mixed chars	Short	0.5839				Not significant
	Mixed chars	Medium	0.1629				Not significant
	Mixed chars	Long	0.9162				Not significant
SHA-1	Pure digits	Short	0.7639	0.532	0.201	MediuM	Not significant
	Pure digits	Medium	0.5611				Not significant
	Pure digits	Long	0.1969				Not significant
	Pure letters	Short	0.4924				Not significant
	Pure letters	Medium	0.3684				Not significant
	Pure letters	Long	0.4309				Not significant
	Mixed chars	Short	0.5544				Not significant
	Mixed chars	Medium	0.8823				Not significant
	Mixed chars	Long	0.5608				Not significant
SHA-256	Pure digits	Short	0.586	0.631	0.273	Good	Not significant
	Pure digits	Medium	0.1472				Not significant
	Pure digits	Long	0.7449				Not significant
	Pure letters	Short	0.9634				Not significant
	Pure letters	Medium	0.6239				Not significant
	Pure letters	Long	0.4309				Not significant
	Mixed chars	Short	0.1199				Not significant
	Mixed chars	Medium	0.8435				Not significant
	Mixed chars	Long	0.9857				Not significant

Table 3. t-value Data for Various Hash Algorithms

Hash Algorithm	String Type	Length Type	T-test t-value	Average Value	Standard Deviation	Security Rating
MD5	Pure digits	Short	-2.44	0.8078	0.6816	Poor
	Pure digits	Medium	-0.143			
	Pure digits	Long	0.75			
	Pure letters	Short	0.586			
	Pure letters	Medium	0.822			
	Pure letters	Long	-0.475			
	Mixed chars	Short	-0.549			
	Mixed chars	Medium	1.4			

	Mixed chars	Long	0.105			
SHA-1	Pure digits	Short	0.301	0.6533	0.3138	Good
	Pure digits	Medium	-0.582			
	Pure digits	Long	1.295			
	Pure letters	Short	0.688			
	Pure letters	Medium	0.902			
	Pure letters	Long	0.789			
	Mixed chars	Short	0.592			
	Mixed chars	Medium	0.148			
	Mixed chars	Long	-0.583			
SHA-256	Pure digits	Short	0.546	0.6034	0.5364	Medium
	Pure digits	Medium	-1.455			
	Pure digits	Long	0.326			
	Pure letters	Short	-0.046			
	Pure letters	Medium	-0.491			
	Pure letters	Long	0.789			
	Mixed chars	Short	1.562			
	Mixed chars	Medium	-0.198			
	Mixed chars	Long	0.018			

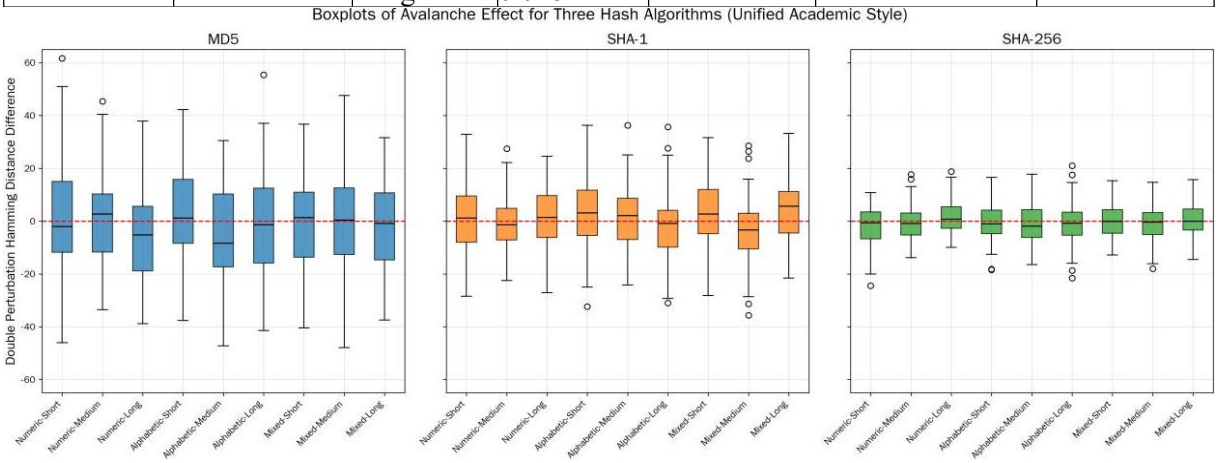

Figure 2: Box Plot of the Difference Distribution of Hamming Distances under Dual Perturbations for Three Hash Algorithms

4.2 Visualization of the Difference Distribution of Hamming Distances under Dual Perturbations for Different Hash Algorithms

Figure 2 illustrates the distribution of differences in Hamming distances resulting from dual perturbations across nine sets of input feature combinations (string types and lengths) for various hash algorithms. The box plot clearly indicates that the distribution for SHA-256 is the most concentrated, demonstrating the highest level of stability. SHA-1 follows, exhibiting a certain degree of dispersion while still maintaining a relatively stable structure. In contrast, MD5 displays the most pronounced dispersion characteristics, reflecting a significant instability in its avalanche effect. The observed gradient in these distributions correlates positively with the security levels of the three algorithms; that is, algorithms with higher security levels exhibit superior stability in their avalanche effects. This finding effectively validates the discriminative validity of the quantitative assessment framework established in this study.

Specifically, the MD5 algorithm exhibits a pronounced dispersion in the distribution of Hamming distance differences under dual perturbation conditions, owing to design flaws in its internal compression function and its known collision vulnerabilities. This characteristic leads to a frequent occurrence of extreme outliers, which directly reveals the non-random nature of its diffusion mechanism and its structural weaknesses. In contrast, the boxplot representation of SHA-256 demonstrates significant convergence, indicating that its

compression function possesses robust confusion and diffusion properties. This observation aligns with the widespread adoption of SHA-256 in security-critical domains such as digital signatures and blockchain technology. The distribution characteristics of SHA-1 fall between those of MD5 and SHA-256, a finding that corresponds with the theoretically established limitations of this algorithm and the ongoing trend of its gradual deprecation in the industry.

In summary, Figure 2 confirms that the experimental observations are consistent with theoretical expectations, providing strong empirical support for the effectiveness and reliability of this evaluation framework in comparing the security of cryptographic algorithms.

4.3 Visualization of Chi-Square Test p-Value Distributions for Three Hashing Algorithms

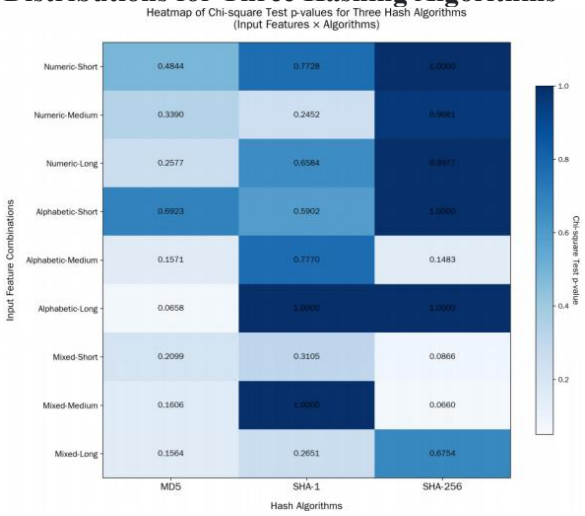

Figure 3.Heatmap of Chi-Square Test p-Value Distributions for Three Hashing Algorithms

Figure 3 illustrates the distribution characteristics of the chi-square test p-values for three hashing algorithms under nine combinations of input features (string length and type). A closer examination of the color gradient in the heatmap reveals that all p-values from the chi-square tests exceed 0.05. In contrast, the overall p-value for MD5 is notably lower, approaching the significance threshold in the pure alphabet-long input scenario. This finding corroborates the theoretical conclusion proposed by Wang et al [23]. regarding the failure of MD5's collision resistance, thereby providing new empirical evidence for the algorithm's obsolescence in the cryptographic community from a uniformity perspective.

SHA-256 exhibits the highest overall p-values with the most stable distribution, although slight fluctuations are observed in extremely complex input scenarios. Conversely, SHA-1 demonstrates robust performance, achieving an ideal p-value of 1.0000 for both pure alphabet-long strings and mixed character-medium length inputs, indicating its good adaptability to medium and long inputs. This aligns with the industry trend of gradually replacing SHA-1 with more advanced algorithms.

The heatmap effectively integrates and visually represents multidimensional information through color coding, addressing the research gap in the evaluation of hashing algorithm uniformity concerning the lack of horizontal comparison and pattern recognition tools. It offers a framework that balances intuitiveness with rigor, further validating the effectiveness and practical value of the quantitative assessment framework presented in this study.

4.4 Visualization of Absolute t-Value Comparisons for Different Hashing Algorithms

Figure 4 presents a grouped bar chart illustrating the comparative distribution of the absolute values of t-values across different hash algorithms and different input feature combinations.. The quantitative results indicate that the overall variability mean for MD5 is 0.8078, for SHA-1 it is 0.6533, and for SHA-256 it is 0.6034. This demonstrates a decreasing trend in variability as the security level increases, suggesting that algorithms with higher security exhibit more stable output responses. Specifically, MD5 shows the highest absolute t-value under the “pure numeric-short” input condition, along with the greatest fluctuation (standard deviation of 0.6816), highlighting its significant output instability. Notably, the shorter length of MD5 limits the dispersion of its output space, making short inputs, particularly pure numeric strings, more susceptible to substantial statistical variation. In contrast, SHA-256 from the SHA series generally exhibits lower absolute t-values with limited fluctuation, aligning with the strong avalanche characteristics expected of high-security hashing algorithms. Although SHA-1 performs better overall than MD5, it still shows observable fluctuations, reflecting the theoretical limitations of this algorithm and the rationale for its gradual replacement in practical

applications.

The grouped bar chart employed in this study provides a clear visualization of the hierarchical differences in output variability among different algorithms and the interaction effects of input features. This further substantiates the sensitivity and effectiveness of the statistical evaluation framework proposed in this paper in capturing the behavioral characteristics of hashing algorithms.

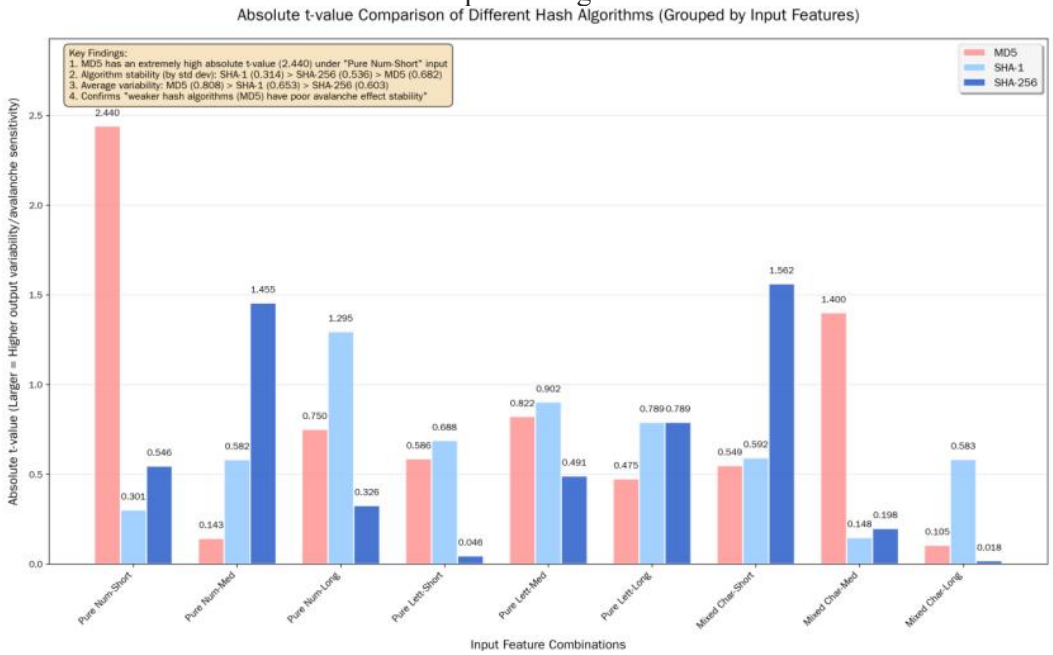

Figure 4. Comparison of Absolute t-values for Different Hashing Algorithms

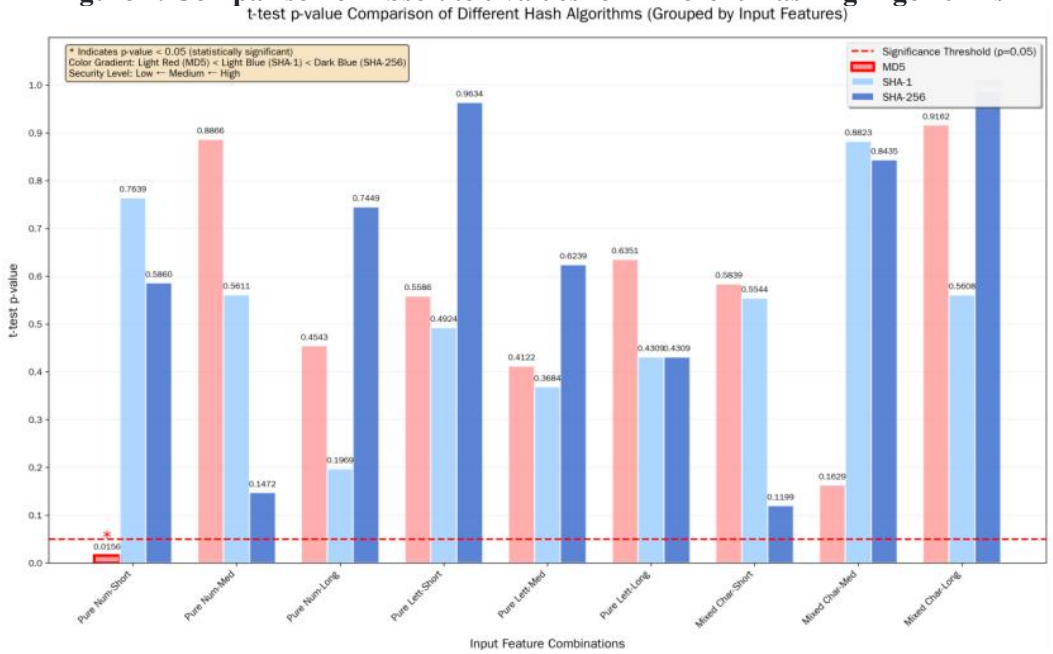

Figure 5. Bar Chart Displaying p-value Comparisons from Paired Sample t-tests for Different Hashing Algorithms

4.5 Visualization of p-Value Comparisons from t-tests for Different Hashing Algorithms

Figure 5 depicts a grouped bar chart that visualizes the comparative distribution of p-values from paired-sample t-tests conducted on different hash algorithms under various input feature combinations.. To facilitate correlation analysis, the chart employs a gradient color scheme corresponding to security levels and includes a red line indicating the 0.05 significance threshold.

The results indicate that only the MD5 algorithm yields a p-value of 0.0156 under the "pure numeric-short string" input condition, which is below the significance threshold. This finding

suggests that, in this scenario, the output variability is statistically significant, reflecting the algorithm's heightened sensitivity to specific types of short inputs and its unstable avalanche effect. In contrast, all other algorithms exhibit p-values exceeding 0.05 across all input combinations, indicating no significant fluctuations. Although SHA-1 demonstrates stronger security than MD5, it still presents theoretical limitations, as its output shows some inconsistency under certain input conditions. SHA-256 consistently displays good statistical stability across all testing conditions, confirming its reliability in practical applications.

The findings of this study align with the known security characteristics of the three algorithms, providing further quantitative justification for prioritizing the use of modern hashing algorithms such as SHA-256 in practical applications, particularly when handling short strings or structured data, while advising against the use of MD5.

5. Conclusion and Outlook

This study successfully established a quantitative assessment framework for the avalanche effect based on chi-square tests and t-tests, effectively addressing the limitations of traditional methods in terms of statistical support and systematicity. The results indicate that this framework can clearly differentiate the performance of various algorithms concerning the avalanche effect, thereby validating the effectiveness and reliability of the proposed method.

5.1 Limitations of the Study

The study has inherent limitations in its design and methodology. Firstly, while the experimental subjects-MD5, SHA-1, and SHA-256-are representative, the framework does not encompass all modern hashing algorithms, and its generalizability remains to be verified. Secondly, the types and lengths of input strings used in the experiments do not fully simulate all input scenarios encountered in practical applications, which may affect the comprehensive evaluation of the avalanche effect. Lastly, this framework evaluates the avalanche effect solely from the dimensions of distribution uniformity and output variability, without incorporating other critical metrics such as completeness of the avalanche effect and bit independence, resulting in a relatively limited assessment scope.

5.2 Outlook

Building on the findings and limitations of this study, future research can further optimize and enhance the framework. Firstly, the selection of hashing algorithms can be expanded to include more algorithms recommended by modern cryptographic standards (such as SHA-3, BLAKE2/3, etc.), thereby improving the generalizability of the assessment framework for hash functions. Secondly, the introduction of information-theoretic metrics and non-parametric testing methods could be explored to construct a multi-metric integrated evaluation system, allowing for a more comprehensive capture of the performance characteristics of hash functions regarding the avalanche effect. Lastly, applying this research framework to the evaluation of practical cryptographic protocols or lightweight cryptographic primitives could help explore the intrinsic relationships between the avalanche effect and other security attributes (such as collision resistance and second pre-image resistance), providing a more comprehensive theoretical basis and practical guidance for algorithm design and standard formulation.

References

[1] Sadeghi-Nasab, A., & Rafe, V. (2023). A comprehensive review of the security flaws of hashing algorithms. Journal of Computer Virology and Hacking Techniques, 19(2), 287-302.

[2] Saarinen, M. J. O. (2009). Cryptanalysis of Dedicated Cryptographic Hash Functions (Doctoral dissertation, Ph. D. dissertation, University of London).

[3] Li, Bo, Liu, Ping, & Wang, Zhangyi. (2007). Study on the Randomness of SHA-256 Output Sequences. Computer Engineering and Applications, (09), 142-144+156.

[4] Liu, Yang. (2015). Research on Chaotic Pseudo-Random Sequence Algorithms and Image Encryption Technologies [Doctoral dissertation, Harbin Institute of Technology].

[5] Chi, L., & Zhu, X. (2017). Hashing techniques: A survey and taxonomy. ACM Computing Surveys (Csur), 50(1), 1-36.

[6] Upadhyay, D., Gaikwad, N., Zaman, M., & Sampalli, S. (2022). Investigating the avalanche effect of various cryptographically secure hash functions and hash-based applications. IEEE Access, 10,

112472-112486.
[7] Wang, Gan & Zhang, Wenying. (2016). Security Analysis of SHA-3. Computer Applications Research, 33(03), 851-854+865.
[8] Zhou, Lü & Chen, Qin. (2002). Methods and Algorithms for Statistical Performance Testing of Block Ciphers. Journal of Hangzhou Electronic Engineering Institute, (06), 81-84. DOI: 10.13954/j.cnki.hdu.2002.06.018.
[9] Saarinen, M. J. O. (2009). Cryptanalysis of Dedicated Cryptographic Hash Functions (Doctoral dissertation, Ph. D. dissertation, University of London).
[10] Damasevicius, R., Ziberkas, G., Stuikys, V., & Toldinas, J. (2012). Energy consumption of hash functions. Elektronika ir elektrotechnika., 18(10), 81-84.
[11] Sofi, A. A., Mir, A. H., & Jabeen, Z. (2025). Effect of hash functions on speed and security within Bitcoin's proof-of-work (PoW). Cluster Computing, 28(11), 724.
[12] Palukha, U., & Kharin, Y. (2019, June). Performance analysis for statistical testing of random and pseudorandom generators by entropy statistics. In 2019 International Conference on Information and Digital Technologies (IDT) (pp. 358-364). IEEE.
[13] Kaminsky, A. (2019). Testing the randomness of cryptographic function mappings. Cryptology ePrint Archive.
[14] Damasevicius, R., Ziberkas, G., Stuikys, V., & Toldinas, J. (2012). Energy consumption of hash functions. Elektronika ir elektrotechnika., 18(10), 81-84.
[15] Bookstein, A., Kulyukin, V. A., & Raita, T. (2002). Generalized hamming distance. Information Retrieval, 5(4), 353-375.
[16] Mitsuya, S., Nakashima, Y., Inenaga, S., Bannai, H., & Takeda, M. (2021). Compressed communication complexity of hamming distance. Algorithms, 14(4), 116.
[17] Zhu, Mingfu, Zhang, Baodong, & Lü, Shuwang. (2002). A Statistical Analysis of the Diffusion Characteristics of Block Cipher Algorithms. Journal of Communications, (10), 122-128.
[18] De Moivre, A. (2020). The doctrine of chances: A method of calculating the probabilities of events in play. Routledge.
[19] JAISWAL, M. S., Cho, M., & Kang, B. (2022). U.S. Patent No. 11,410,043. Washington, DC: U.S. Patent and Trademark Office.
[20] Fang, Xiangzhong. (2022). Chi-Squared Distribution and Chi-Squared Test. China Statistics, (05), 29-31.
[21] Wang, Lili. (2018). Discussion on Testing Issues in Hypothesis Testing. Science & Technology Vision, (22), 82-84. https://doi.org/10.19694/j.cnki.issn2095-2457.2018.22.038.
[22] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2021). Handbook of Applied Cryptography. Instructor, 202101.
[23] Wang, X., Yin, Y. L., & Yu, H. (2005, August). Finding collisions in the full SHA-1. In Annual international cryptology conference (pp. 17-36). Berlin, Heidelberg: Springer Berlin Heidelberg.

