Vol. 8, No. 2, April 2026

ISI Journal of Information Systems and Informatics

Published By Asosiasi Doktor Sistem Informasi Indonesia

Experimental Evaluation of Avalanche Effect and Hash Consistency in SHA-256, SHA3-256, and BLAKE2b-256 for Smart Grid Data Integrity Verification

Muhammad Ridwan[1], Rusydi Umar[2], Muhammad Kunta Biddinika[3], Puguh Prasetyo[4]

[1,2,3]Magister Informatika, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

[4] Mathematics Education Department, Faculty of Industrial Technology, Universitas Ahmad Dahlan, Yogyakarta, Indonesia

Received:
February 20, 2026
Revised:
May 31, 2026
Accepted:
July 22, 2026
Published:
August 22, 2026

Corresponding Author:
Author Name*:
Muhammad Ridwan
Email*:
Ridwan211221@gmail.com

DOI:
10.63158/journalisi.v8i4.1685

Abstract. Ensuring the integrity of Smart Grid data is essential for preventing unauthorized data modification during transmission and storage. Cryptographic hash functions provide an efficient mechanism for integrity verification; however, their diffusion characteristics and computational performance under identical conditions require further evaluation. This study experimentally compares SHA-256, SHA3-256, and BLAKE2b-256 for Smart Grid data integrity verification using avalanche effect analysis, hash consistency evaluation, and computational performance benchmarking with a standardized 256-bit digest length. The results show that all evaluated algorithms achieved avalanche effect values close to the theoretical 50% criterion (SHA-256: 49.97%, SHA3-256: 49.96%, and BLAKE2b-256: 50.04%) and maintained a 100% hash consistency rate, indicating comparable diffusion capability and deterministic behavior. In terms of computational performance, BLAKE2b-256 achieved the shortest average execution time (11.14 ms), outperforming SHA-256 (13.24 ms) and SHA3-256 (16.87 ms). These findings indicate that all evaluated algorithms are suitable for Smart Grid data integrity verification, while BLAKE2b-256 provides the highest computational efficiency under the experimental conditions. The results are limited to the experimental evaluation of cryptographic hash function behavior and do not represent a comprehensive assessment of Smart Grid cybersecurity.

Keywords: Smart Grid, Data Integrity, Cryptographic Hash Function, SHA-256, SHA3-256, BLAKE2b-256, Avalanche Effect.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

1. INTRODUCTION

The increasing demand for sustainable energy systems has accelerated the adoption of smart grids, particularly in developing countries[1]. A smart grid integrates advanced metering, communication, and control technologies to improve efficiency, reliability, and adaptability of energy distribution[2]. However, this integration also introduces new challenges. As energy data become more digital and dynamic, the system is exposed to risks of manipulation, false data injection, and cyberattacks that may compromise stability[3],[4]. To address these issues, cryptographic techniques are essential to ensure data integrity and trust[5]. Hash functions, in particular, offer lightweight yet powerful mechanisms to detect unauthorized modifications[6],[7].

In practical Smart Grid environments, cryptographic hash functions are commonly employed as integrity verification mechanisms rather than encryption tools[8]. Hash values can be attached to smart meter readings, energy consumption records, distributed databases, or blockchain transactions to ensure that data remain unchanged during transmission and storage[9]. When a record is modified intentionally or accidentally, even by a single bit, the resulting hash value should differ significantly from the original one. Therefore, hash functions provide an efficient mechanism for detecting tampering and supporting trustworthy energy data management[10].

Their effectiveness depends on two critical properties: the avalanche effect, where a single-bit change in the input leads to significant differences in the output, and consistency, which guarantees identical outputs for identical inputs[7]. These properties are crucial for real-time energy monitoring, where even minor data manipulation can lead to misjudgments in power balancing and stability assessment[8]. The importance of energy data integrity becomes clearer when viewed through the lens of real smart grid datasets[3]. In this study, we use an augmented dataset of smart grid stability that includes parameters such as reaction time of producers and consumers, power balance, and price elasticity coefficients, alongside system stability labels (stable or unstable)[7]. These parameters reflect the dynamic behavior of energy producers and consumers in maintaining balance within the grid[9]. Any manipulation of such data could misrepresent whether the system is stable or unstable, potentially leading to failures in decision-

making for energy management[11]. Several related studies provide the theoretical foundation for this research[12].

This study focuses on integrity-related threats that target Smart Grid data, particularly false data injection, unauthorized record modification, transmission tampering, and integrity attacks against blockchain-based energy records[13]. Such attacks may alter measurements related to power balance and system stability, potentially leading to incorrect operational decisions[14]. Although cryptographic hash functions cannot prevent attacks directly, they can provide an effective mechanism for detecting unauthorized modifications and verifying data integrity[8].

Rajiman and Indrayani demonstrated the robustness of SHA-256 in securing digital documents[15], while Franck et al. focused on hardware optimizations for performance. Research on smart grids by Moze and Sinaga et al[7]. highlighted the importance of reliable data aggregation in energy systems. Furthermore, studies on blockchain emphasize that hash functions underpin distributed trust and immutability. Meanwhile, Verma and Sharma affirmed the avalanche effect as a fundamental requirement for cryptographic strength[16],.

Despite extensive research on cryptographic hash functions and Smart Grid security, experimental studies that compare SHA-256, SHA3-256, and BLAKE2b-256 under identical conditions using Smart Grid stability data remain limited[8]. Furthermore, previous studies often focus on computational performance or blockchain implementation without examining avalanche behavior and consistency as integrity-related indicators[17]. To address this gap, this study experimentally evaluates the avalanche effect, hash consistency, and computational performance of three widely used cryptographic hash algorithms using Smart Grid data. The results are intended to provide empirical evidence regarding the suitability of these algorithms for integrity verification in Smart Grid data management environments[5].

Among the available attributes in the Smart Grid Stability Augmented Dataset, the Power Balance–Energy Producer attribute was selected as the primary experimental input because it directly represents the balance between energy generation and consumption within the grid. This parameter plays a critical role in determining system stability and is

highly sensitive to manipulation. Any unauthorized alteration may affect stability assessment and operational decision-making, making it an appropriate attribute for evaluating hash-based integrity verification mechanisms.

2. METHODS

To improve the reproducibility and clarity of the experimental procedure, the overall workflow of this study is illustrated in Figure 1[18]. The workflow consists of four major stages: data preparation and input processing, cryptographic hash generation, evaluation and analysis, and result visualization and interpretation[19]. Each stage was designed to ensure that the three evaluated cryptographic hash algorithms SHA-256, SHA3-256, and BLAKE2b-256 were assessed under identical experimental conditions[20].

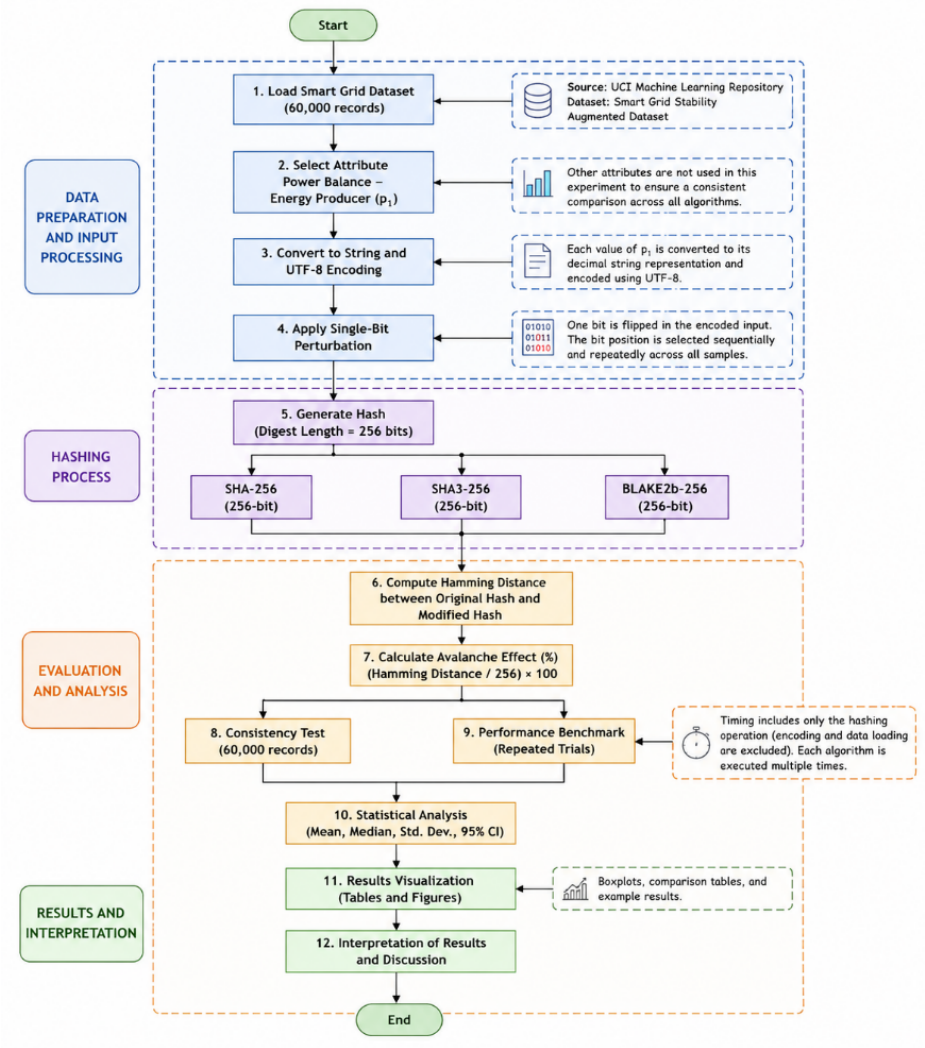

Figure 1. summarizes the complete experimental pipeline adopted in this study.

The first stage begins by loading the Smart Grid Stability Augmented Dataset, which contains 60,000 operational records[3],[17]. Although the dataset provides multiple operational attributes, only the Power Balance Energy Producer (p_1) attribute was selected as the experimental input[1]. This attribute was chosen because it represents continuously varying power-generation measurements, making it suitable for evaluating the sensitivity of cryptographic hash functions to small input modifications while maintaining a consistent comparison across all evaluated algorithms[7]. Next, each selected value is converted into its decimal string representation and encoded using UTF-8 before hashing[21]. The encoded byte sequence is then subjected to a true single-bit perturbation[5], in which one bit is flipped using a bitwise XOR operation[8].

To avoid bias toward a particular bit location, the perturbation position is varied systematically across the dataset during the avalanche-effect experiment[20],[22]. After preprocessing, the encoded inputs are processed independently using SHA-256, SHA3-256, and BLAKE2b-256, all configured to produce a 256-bit digest. Standardizing the digest length ensures that the avalanche effect and Hamming distance can be compared fairly across the three algorithms[23],[24]. The evaluation stage consists of three complementary analyses. First, the Hamming distance between the original and modified hash outputs is computed[20]. Second, the Avalanche Effect is calculated as the percentage of changed bits relative to the 256-bit digest length.[23] Third, a hash consistency test is performed using all 60,000 dataset records to verify that identical inputs consistently produce identical hash outputs[18]. In parallel, a performance benchmark is conducted through repeated hashing trials after warm-up executions, measuring only the hash computation time while excluding data loading and UTF-8 encoding overhead[25],[26]. Finally, the experimental results are analyzed statistically by reporting the mean, median, standard deviation, and 95% confidence interval[27]. The results are then presented using comparison tables, boxplots, and avalanche-trend visualizations, followed by an interpretation of the findings in terms of computational efficiency, avalanche behavior, and Smart Grid data-integrity verification[28].

2.1 Dataset Description

This study employed the Smart Grid Stability Augmented Dataset, which is publicly available through the UCI Machine Learning Repository[3]. The dataset consists of 60,000 simulated Smart Grid operational records, representing various operating conditions of

a distributed Smart Grid system[29]. Each record contains multiple numerical attributes describing reaction time, power balance, price elasticity, and system stability, allowing comprehensive evaluation under different Smart Grid operating scenarios[30]. Table 1 summarizes the attributes contained in the Smart Grid Stability Augmented Dataset[22]. These variables represent the dynamic characteristics of producers and consumers participating in Smart Grid operation, including response time, power balance, price elasticity, and overall system stability[6],[27].

Table 1. Attributes of the Smart Grid Stability Augmented Dataset

Variable	Attribute	Description
τ_1	Reaction Time – Energy Producer	Response time of the energy producer.
τ_2	Reaction Time – Consumer 1	Response time of consumer 1.
τ_3	Reaction Time – Consumer 2	Response time of consumer 2.
τ_4	Reaction Time – Consumer 3	Response time of consumer 3.
p_1	Power Balance – Energy Producer	Power balance generated by the energy producer.
p_2	Power Balance – Consumer 1	Power balance of consumer 1.
p_3	Power Balance – Consumer 2	Power balance of consumer 2.
p_4	Power Balance – Consumer 3	Power balance of consumer 3.
g_1	Price Elasticity Coefficient – Energy Producer	Price elasticity coefficient of the energy producer.
g_2	Price Elasticity Coefficient – Consumer 1	Price elasticity coefficient of consumer 1.
g_3	Price Elasticity Coefficient – Consumer 2	Price elasticity coefficient of consumer 2.
g_4	Price Elasticity Coefficient – Consumer 3	Price elasticity coefficient of consumer 3.
stab	Stability Index	Numerical indicator of system stability.
stabf	Stability Label	Binary stability classification (Stable/Unstable).

Among the available attributes, this study selected the Power Balance Energy Producer (p_1) variable as the experimental input for evaluating the cryptographic hash algorithms[22]. This attribute was chosen because it represents continuously varying

numerical measurements associated with power generation, making it suitable for assessing the sensitivity of cryptographic hash functions to small input perturbations while maintaining a consistent experimental basis across SHA-256, SHA3-256, and BLAKE2b-256[23],[31]. Prior to the hashing process, each value of the selected attribute was converted into its decimal string representation and encoded using UTF-8[20],[22]. The encoded byte sequence was subsequently used as the input for all cryptographic hash functions[32]. Before conducting the experiments, the dataset was inspected to verify data completeness and duplicate records[19]. No additional normalization or feature transformation was applied because cryptographic hash functions operate directly on encoded input data rather than on statistical properties of numerical features[33]. For the avalanche-effect evaluation, 10,000 samples were randomly selected from the Power Balance Energy Producer (p_1) attribute, whereas the hash consistency evaluation was performed using the complete dataset of 60,000 records[22].

2.2 Experimental Environment

All experiments were conducted using a personal computer equipped with a 9th Generation Intel® Core™ i5 processor, 16 GB of RAM, a 256 GB solid-state drive (SSD) for the operating system and software installation, and an additional 1 TB hard disk drive (HDD) for data storage[22]. The experimental implementation was developed in Python 3.x, utilizing the built-in hashlib library for the implementation of SHA-256, SHA3-256, and BLAKE2b-256[32]. To ensure a fair comparison among the evaluated algorithms, the BLAKE2 implementation employed the *hashlib.blake2b()* function configured with a digest size of 32 bytes (256 bits), matching the digest length produced by SHA-256 and SHA3-256[20]. All cryptographic algorithms were executed under the same software configuration and experimental conditions to eliminate implementation bias[23]. The benchmark evaluation was performed using repeated execution trials following an initial warm-up phase to minimize runtime initialization effects[30]. The reported execution time represents only the cryptographic hashing process, while data loading, UTF-8 encoding, and other preprocessing operations were excluded from the timing measurements[20],[22]. To improve the reliability of the performance evaluation, the benchmark results were analyzed using descriptive statistical measures, including the mean, median, standard deviation, and 95% confidence interval[27].

2.3 Cryptographic Hash Generation

The cryptographic hash generation process was performed using three widely adopted cryptographic hash algorithms, namely SHA-256, SHA3-256, and BLAKE2b-256[23]. To ensure a fair comparison, all algorithms were configured to produce a 256-bit digest, thereby eliminating differences caused by unequal output lengths[20],[22]. Before hash generation, each value of the selected Power Balance–Energy Producer (p_1) attribute was converted into its decimal string representation and encoded using the UTF-8 character encoding standard[19],[25]. The resulting byte sequence served as the input to all evaluated cryptographic hash functions[24]. This encoding procedure ensured that identical input data were processed consistently across all algorithms throughout the experiments[33].

The SHA-256 and SHA3-256 implementations were executed using the standard hashlib library available in Python[32]. The BLAKE2 implementation employed the *hashlib.blake2b()* function configured with a digest size of 32 bytes (256 bits), thereby producing output lengths equivalent to those of SHA-256 and SHA3-256[20]. Standardizing the digest length was necessary to ensure that the Hamming distance and avalanche-effect measurements were directly comparable across all evaluated algorithms[23],[22]. For each encoded input, a hexadecimal hash value was generated independently by the three cryptographic hash functions[30]. These hash values subsequently served as the basis for avalanche-effect evaluation, hash consistency verification, and computational performance analysis[25],[27].

2.4 Avalanche Effect Evaluation

The avalanche effect evaluation was performed to assess the sensitivity of each cryptographic hash function to minimal input modifications[8],[1]. An ideal cryptographic hash function should exhibit the avalanche property, whereby a one-bit change in the input produces changes in approximately 50% of the output digest bits[11]. This property is essential for evaluating diffusion characteristics and resistance to input tampering[34]. For the avalanche-effect experiment, 10,000 samples were randomly selected from the Power Balance–Energy Producer (p_1) attribute of the Smart Grid Stability Augmented Dataset[3],[17]. Each selected value was first converted into its decimal string representation and encoded using UTF-8[35],[21]. A true single-bit perturbation was then introduced by applying a bitwise Exclusive OR (XOR) operation to the encoded byte

array[16]. Unlike character replacement, this approach guarantees that only one binary bit is modified in the original input[12]. To avoid positional bias, the perturbed bit position was varied sequentially throughout the experiment so that different bit locations were evaluated across the dataset[36].

After perturbation, both the original and modified inputs were processed independently using SHA-256, SHA3-256, and BLAKE2b-256[7]. Since all three algorithms generated 256-bit digest outputs, the resulting hashes could be compared directly without additional normalization[9]. The similarity between the original and modified hash outputs was quantified using the Hamming Distance[35], which represents the total number of differing bits between two binary hash values[11]. The avalanche effect was subsequently calculated as the percentage of changed bits relative to the digest length of 256 bits[12]. The Hamming Distance was computed according to Equation (1).

$$HD(h1, h2) = \sum_{i=1}^{n} (b_{1i} \oplus b_{2i}) \quad (1)$$

where b_{2i} and b_{2i} denote the it bit of the original and modified digest outputs, respectively. The avalanche percentage was calculated as shown in Equation (2).

$$AE(\%) = 256 \div HD \times 100 \quad (2)$$

where AE represents the avalanche effect expressed as a percentage and HD is the corresponding Hamming Distance[35]. An avalanche value approaching 50% indicates an ideal diffusion property, whereas substantially lower or higher values suggest less desirable avalanche behavior[7]. To ensure a fair comparison, the same input samples, perturbation procedure, and bit-flip positions were applied consistently across SHA-256, SHA3-256, and BLAKE2b-256[9]. The resulting avalanche percentages were subsequently analyzed using descriptive statistical measures, including the mean, median, standard deviation, minimum, maximum, and boxplot visualization[17].

2.5 Hash Consistency Evaluation

The hash consistency evaluation was conducted to verify the deterministic behavior of the evaluated cryptographic hash functions[37]. A deterministic hash function is expected to generate the identical hash output whenever the same input is processed repeatedly

under identical computational conditions[32]. Although this property represents a fundamental characteristic of cryptographic hash functions, the experiment was included to verify the correctness and consistency of the software implementation adopted in this study[23]. Unlike the avalanche-effect experiment, which evaluates the sensitivity of hash functions to minimal input modifications, the consistency evaluation utilized the entire Smart Grid Stability Augmented Dataset consisting of 60,000 records[22]. Each value of the selected Power Balance–Energy Producer (p_1) attribute was converted into its decimal string representation and encoded using UTF-8 before being processed independently by SHA-256, SHA3-256, and BLAKE2b-256[20].

To verify consistency, every input value was hashed repeatedly under identical experimental conditions[23]. The resulting hash values generated from repeated executions were compared with the corresponding reference hash values[22]. A mismatch was recorded whenever two identical inputs produced different hash outputs[24]. The consistency rate was subsequently calculated as the ratio between the number of matched hashes and the total number of evaluated records[27]. The hash consistency was calculated using Equation (3).

$$Consistency(\%) = N_{total} \div N_{match} \times 100 \quad (3)$$

where N_{match}denotes the number of identical hash outputs obtained from repeated executions, while N_{total}represents the total number of evaluated records.

A consistency value of 100% indicates that identical inputs always produce identical hash outputs, confirming the deterministic behavior of the evaluated cryptographic hash functions under the implemented experimental environment[32]. The consistency results obtained from SHA-256, SHA3-256, and BLAKE2b-256 were subsequently compared to verify the reliability of each implementation[32]. all evaluated cryptographic hash functions achieved a consistency rate of 100%, with no mismatched hash values observed during repeated executions[25]. These results confirm that the implemented SHA-256, SHA3-256, and BLAKE2b-256 algorithms preserve the deterministic property required for reliable Smart Grid data integrity verification[38].

2.6 Performance Benchmark

The computational performance of the evaluated cryptographic hash algorithms was assessed by measuring the execution time required to generate hash values for the selected Smart Grid dataset attribute[33]. Performance evaluation was conducted to compare the computational efficiency of SHA-256, SHA3-256, and BLAKE2b-256 under identical experimental conditions[18]. Prior to benchmarking, an initial warm-up phase was performed to minimize the influence of runtime initialization and interpreter optimization. Following the warm-up stage, repeated benchmark trials were executed for each algorithm using the same experimental dataset and identical software configuration[38]. To ensure a fair comparison, all timing measurements were performed using the same implementation environment and processing pipeline[28].

The reported execution time represents only the cryptographic hashing operation. Data loading, UTF-8 encoding, dataset preprocessing, and result visualization were intentionally excluded from the benchmark because these operations are independent of the cryptographic algorithms and could introduce additional measurement bias[17],[17]. The benchmark results were summarized using descriptive statistical measures, including the mean execution time, median execution time, standard deviation, and 95% confidence interval[39]. These statistical indicators provide a more reliable representation of computational performance by reducing the influence of random runtime fluctuations observed during repeated executions[34]. The benchmark results obtained from SHA-256, SHA3-256, and BLAKE2b-256 were subsequently compared to identify differences in computational efficiency while maintaining identical experimental conditions[9]. the computational performance of the evaluated algorithms varies primarily in terms of execution time, while all algorithms maintain identical digest lengths of 256 bits[5]. These benchmark results provide the basis for comparing computational efficiency without affecting the cryptographic properties evaluated in the avalanche-effect and consistency experiments[16].

3.7 Experimental Procedure

To improve the reproducibility of the proposed methodology, the complete experimental procedure is summarized in Algorithm 1[40]. The algorithm provides a step-by-step representation of the implemented workflow, covering dataset preparation, input encoding, cryptographic hash generation, avalanche-effect evaluation, hash consistency

verification, computational performance benchmarking, and statistical analysis[7]. Unlike the conceptual workflow presented in Figure 2, Algorithm 1 describes the exact sequence of operations performed during the experiments[11].

Algorithm 1 Proposed Experimental Procedure for Cryptographic Hash Evaluation

Require: Smart Grid Stability Augmented Dataset (60,000 records)
Ensure: Avalanche effect results, hash consistency results, and benchmark statistics

```
Load dataset D                                                  // Load Smart Grid dataset
Extract attribute p_1 (Power Balance–Energy Producer)           // Attribute used in the experiment
Convert each value of p_1 to decimal string                     // Standardize to string representation
Encode each string using UTF-8 to obtain byte array             // Input representation for hashing
Randomly select 10,000 samples from D for avalanche evaluation  // Dataset for avalanche test
Use all 60,000 records of D for hash consistency evaluation     // Dataset for consistency test
Define HashSet = {SHA-256, SHA3-256, BLAKE2b-256}               // Algorithms under evaluation
Set digest length to 256 bits (32 bytes) for all algorithms     // Ensure fair comparison
// ———— Part I: Avalanche Effect Evaluation ————
for each sample s in the 10,000 selected samples do
    for each algorithm h in HashSet do
        H_orig ← Hash(h, s)                                     // Generate original hash
        pos ← SelectBitPosition(s)                              // Select one bit position to flip
        s_mod ← FlipOneBit(s, pos)                              // Apply single-bit perturbation (XOR)
        H_mod ← Hash(h, s_mod)                                  // Generate modified hash
        HD ← HammingDistance(H_orig, H_mod)                     // Compute Hamming Distance
        AE ← (HD/256) × 100                                     // Compute Avalanche Effect (%)
        Store AE                                                // Save result for analysis
    end for
end for
// ———— Part II: Hash Consistency Evaluation ————
for each algorithm h in HashSet do
    Mismatch ← 0
    for each record r in all 60,000 records do
        H_1 ← Hash(h, r)
        H_2 ← Hash(h, r)                                        // Repeated hashing
        if H_1 ≠ H_2 then
            Mismatch ← Mismatch + 1                             // Count mismatches
        end if
    end for
    Consistency(%) ← ((60,000 − Mismatch) / 60,000) × 100       // Compute consistency rate
    Store consistency result                                    // Save result for reporting
end for
// ———— Part III: Performance Benchmark ————
for each algorithm h in HashSet do                              // Reduce initialization effect
    Perform warm-up execution                                   // N = number of benchmark repetitions
    for i ← 1 to N trials do
        Start timer
        for each record r in all 60,000 records do
            Hash(h, r)                                          // Measure hashing time only
        end for
        Stop timer and record execution time T_i                // Exclude encoding/loading time
    end for
    Compute mean, median, standard deviation, and 95% confidence interval of {T_i}   // Statistical analysis
    Store benchmark statistics                                  // Save result for reporting
end for
// ———— Part IV: Result Generation ————
Generate tables and figures for avalanche, consistency, and benchmark results   // Visualization
Return all evaluation results                                   // End of procedure
```

Figure 2. Complete Experimental Procedure for Comparative Evaluation of SHA-256, SHA3-256, and BLAKE2b-256 in Smart Grid Data Integrity Verification.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

The procedure begins by loading the Smart Grid Stability Augmented Dataset, which contains 60,000 operational records, followed by the selection of the Power Balance–Energy Producer (p_1) attribute as the experimental input[41]. Each selected value is converted into its decimal string representation and encoded using the UTF-8 character encoding standard to ensure consistent input formatting across all evaluated cryptographic hash functions[5]. For the avalanche-effect evaluation, 10,000 samples are randomly selected from the dataset[42],[2]. Each sample is processed independently using SHA-256, SHA3-256, and BLAKE2b-256, all configured with a 256-bit digest length. A true single-bit perturbation is then introduced by applying a bitwise XOR operation to the encoded input, after which the modified input is hashed again using the same cryptographic algorithm[9]. The Hamming Distance between the original and modified hash outputs is subsequently computed, and the Avalanche Effect is expressed as the percentage of changed bits relative to the 256-bit digest length[5].

The second stage of the procedure performs hash consistency verification using the complete dataset of 60,000 records[15]. Each input is repeatedly processed under identical computational conditions, and the generated hash values are compared to verify deterministic behavior[9]. The number of matching and mismatching hash values is recorded to determine the overall consistency rate for each evaluated algorithm[7]. The final stage evaluates the computational performance of SHA-256, SHA3-256, and BLAKE2b-256 through repeated benchmark executions following an initial warm-up phase[11]. Only the cryptographic hashing operation is included in the execution-time measurement, whereas data loading, UTF-8 encoding, and preprocessing overhead are excluded[21]. The collected benchmark data are analyzed using descriptive statistical measures, including the mean, median, standard deviation, and 95% confidence interval, before being presented through comparison tables and graphical visualizations.[17] Overall, Algorithm 1 provides a concise representation of the complete experimental pipeline implemented in this study, ensuring that the proposed methodology can be reproduced consistently for comparative evaluation of cryptographic hash functions in Smart Grid data integrity verification[14].

3. RESULTS AND DISCUSSION

3.1 Avalanche Effect Analysis

The avalanche effect evaluation was conducted to assess the diffusion capability of the evaluated cryptographic hash algorithms by measuring the proportion of output bits that changed following a one-bit modification in the input data. According to the avalanche criterion, an ideal cryptographic hash function should produce approximately 50% changed output bits when a single input bit is altered. To quantitatively evaluate this property, the statistical results obtained from 10,000 randomly selected Smart Grid data samples are summarized in Table 2.

Table 2. Avalanche Effect Statistics of the Evaluated Cryptographic Hash Algorithms

Algorithm	Average Bits	Median Bits	Std Dev	Minimum	Maximum	Average Avalanche (%)
SHA256	127.92	128.0	8.02	93	156	49.97
SHA3	127.90	128.0	8.01	95	160	49.96
BLAKE2	128.10	128.0	8.00	98	157	50.04

Table 2 shows that all evaluated algorithms produced average avalanche effect values close to the theoretical expectation of 50%. SHA-256 achieved an average avalanche effect of 49.97%, followed by SHA3-256 with 49.96%, while BLAKE2b-256 produced 50.04%. The differences among the three algorithms are minimal, indicating that they exhibit comparable diffusion characteristics when evaluated using identical input samples and a standardized 256-bit digest length. In addition, the relatively small standard deviation indicates that the avalanche effect remained stable throughout the experiment.

To further examine the distribution of avalanche effect values, the experimental results are visualized in Figure 3. The visualization provides a comprehensive overview of the dispersion, central tendency, and variability of the avalanche effect generated by each evaluated algorithm.

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

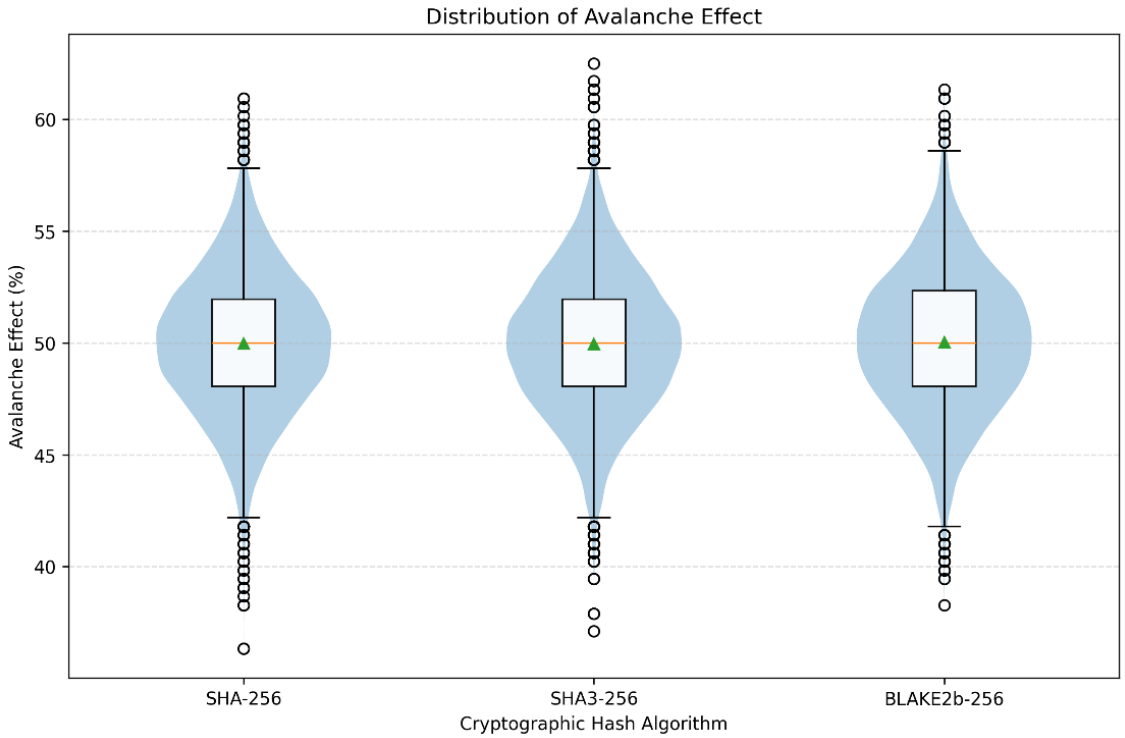

Figure 3. Distribution of Avalanche Effect for SHA-256, SHA3-256, and BLAKE2b-256

As illustrated in Figure 3, the distributions of SHA-256, SHA3-256, and BLAKE2b-256 are highly similar and centered around the 50% avalanche threshold. The median values of all three algorithms are nearly identical, while the interquartile ranges demonstrate comparable variability across the evaluated samples. Although several outliers are present, they represent only a small proportion of the observations and do not significantly influence the overall distribution. These findings further confirm that the evaluated algorithms possess equivalent diffusion capability under the experimental conditions. To illustrate the avalanche effect mechanism in greater detail, one representative example for each evaluated algorithm is presented in Figure 4. The table reports the original input, the modified input generated using a true one-bit perturbation, the corresponding hash outputs, the resulting Hamming distance, and the calculated avalanche percentage.

	Algorithm	Original Input	Modified Input	Original Hash	Modified Hash	Hamming Distance	Avalanche %
0	sha256	399799508641593.000000000000000	399799508641593.000020000000000	d517dafcba63f06625b76d1633d74899ddf037a4b31fa0...	216db3ff95920aa0a0f94db08b68b84bed3ecae875e430...	130	50.78
1	sha3	399799508641593.000000000000000	399799508641593.000020000000000	f70589ad67d46ab4a29f46c2584f9806705284d96fbf84...	609ccdd786e82a0667db9940d08723fb15815a7559ebaf...	129	50.39
2	blake2	399799508641593.000000000000000	399799508641593.000020000000000	2c57b4fffed39c0ec5d4b86255c667fbc72600e20836c5...	eba125d9b442e1835863a7e688236a87a9cfd683545699...	130	50.78

Figure 4. Representative Hash Outputs and Avalanche Effect Results

Figure 4 demonstrates that a single-bit modification in the encoded Smart Grid data produced substantial changes throughout the 256-bit hash digest. Despite the minimal alteration applied to the input, the resulting Hamming distances remained close to half of the digest length, yielding avalanche effect values between approximately 45% and 52%. This behavior confirms that the evaluated cryptographic hash algorithms satisfy the diffusion property expected from secure hash functions, where even the smallest input modification generates an unpredictable and significantly different output.

Overall, the avalanche effect evaluation demonstrates that SHA-256, SHA3-256, and BLAKE2b-256 provide nearly identical diffusion performance for Smart Grid data integrity verification. Since all algorithms consistently generated avalanche effect values close to the theoretical expectation of 50%, no meaningful difference in diffusion capability was observed. Therefore, the avalanche effect alone should not be considered the sole criterion for selecting a cryptographic hash algorithm for Smart Grid applications. Additional aspects, including computational efficiency, implementation complexity, interoperability, and deployment constraints, should also be considered during algorithm selection.

3.2 Hash Consistency Analysis

The hash consistency evaluation was performed to verify the deterministic behavior of the evaluated cryptographic hash algorithms. A deterministic hash function is expected to generate an identical output whenever the same input is processed repeatedly under identical conditions. This property is essential for data integrity verification in Smart Grid systems, where repeated integrity checks must consistently produce the same digest for unchanged data. The quantitative results of the consistency evaluation are presented in Table 3.

Table 3. Hash Consistency Evaluation Results

Algorithm	Evaluated Records	Repeated Trials	Matching Hashes	Mismatches	Consistency (%)
SHA-256	60,000	60,000	60,000	0	100.00
SHA3-256	60,000	60,000	60,000	0	100.00
BLAKE2b-256	60,000	60,000	60,000	0	100.00

ISI Journal of Information Systems and Informatics
Published By Asosiasi Doktor Sistem Informasi Indonesia

Table 3 summarizes the consistency evaluation conducted using the complete Smart Grid dataset comprising 60,000 records. For each evaluated algorithm, every input was hashed repeatedly under identical experimental conditions, and the generated hash values were compared to identify any mismatches. The results demonstrate that SHA-256, SHA3-256, and BLAKE2b-256 all achieved a 100% consistency rate, with zero mismatches observed throughout the evaluation. These findings confirm that the three algorithms consistently produced identical hash outputs for identical inputs, demonstrating correct deterministic behavior.

The absence of mismatched hash values indicates that the implemented hashing procedures remained stable throughout the experiment and that no computational inconsistencies were introduced during repeated executions. Although deterministic consistency is an inherent characteristic of cryptographic hash functions, experimentally verifying this property is important to ensure the correctness of the software implementation and the reproducibility of the generated results. Consequently, the observed consistency provides additional confidence that the subsequent avalanche effect and computational performance evaluations were performed on reliable and reproducible hash outputs.

From the perspective of Smart Grid data integrity, deterministic hash generation enables reliable verification of measurement records, transaction logs, and operational data throughout the communication process. During integrity verification, a newly generated hash can be directly compared with the previously stored reference hash. Any discrepancy immediately indicates that the corresponding data have been modified, either intentionally or unintentionally, allowing unauthorized changes to be detected efficiently without requiring the original data to be transmitted or stored multiple times. Overall, the consistency evaluation confirms that SHA-256, SHA3-256, and BLAKE2b-256 satisfy the deterministic property required for cryptographic hash functions. Since all evaluated algorithms achieved identical consistency performance, this evaluation does not distinguish one algorithm from another. Instead, the primary differences among the evaluated algorithms are expected to arise from other performance characteristics, particularly avalanche behavior and computational efficiency, which are examined in the following section.

3.3 Computational Performance Analysis

The computational performance evaluation was conducted to compare the execution efficiency of SHA-256, SHA3-256, and BLAKE2b-256 under identical experimental conditions. To obtain statistically reliable measurements, the benchmark was performed using 5,000 randomly selected Smart Grid records and repeated 20 times for each algorithm following an initial warm-up phase. The average execution time, median execution time, standard deviation, minimum and maximum execution times, together with the 95% confidence interval, were calculated to evaluate the computational efficiency of each algorithm. The benchmark results are summarized in Table 4.

Table 4. Computational Performance Benchmark

Algorithm	Mean (ms)	Median (ms)	Std Dev (ms)	Minimum (ms)	Maximum (ms)	95% CI Lower	95% CI Upper
SHA256	13.237202	13.084077	0.706406	12.294253	15.873396	12.898005	13.576399
SHA3	16.868331	15.963713	1.778667	15.431986	21.520271	16.014264	17.722398
BLAKE2	11.143167	10.786529	1.131131	7.837922	12.993327	10.600029	11.686305

Table 4 demonstrates that BLAKE2b-256 achieved the shortest average execution time of 11.14 ms, followed by SHA-256 with 13.24 ms, whereas SHA3-256 required the longest execution time of 16.87 ms. The median execution times showed a similar trend, with BLAKE2b-256 recording 10.79 ms, SHA-256 13.08 ms, and SHA3-256 15.96 ms. These results indicate that BLAKE2b-256 consistently required less processing time to generate a 256-bit hash digest under identical benchmarking conditions.

The benchmark variability remained relatively small for all evaluated algorithms. SHA-256 exhibited a standard deviation of 0.71 ms, indicating highly stable execution performance throughout the repeated measurements. BLAKE2b-256 recorded a standard deviation of 1.13 ms, while SHA3-256 showed the largest variation at 1.78 ms, although the observed variability remained within acceptable limits for repeated computational benchmarking. Furthermore, the narrow 95% confidence intervals obtained for each algorithm indicate that the measured average execution times are statistically reliable and reproducible. To provide a clearer comparison of the execution efficiency, the benchmark results are illustrated in Figure 5.

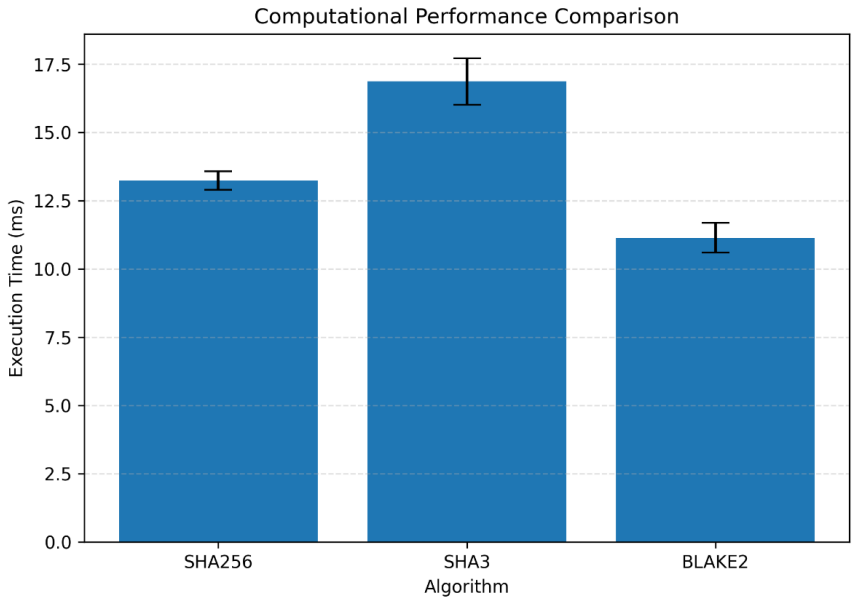

Figure 5. Computational Performance Comparison of SHA-256, SHA3-256, and BLAKE2b-256

As illustrated in Figure 5, BLAKE2b-256 consistently outperformed the other evaluated algorithms in terms of execution speed. Compared with SHA-256, BLAKE2b-256 reduced the average execution time by approximately 15.8%, while the reduction reached approximately 33.9% compared with SHA3-256. Conversely, SHA3-256 exhibited the highest computational cost, reflecting the additional processing complexity associated with its sponge-based construction. SHA-256 demonstrated intermediate performance, providing a balance between computational efficiency and broad implementation compatibility across existing software and hardware platforms.

From the perspective of Smart Grid applications, computational efficiency is an important consideration because integrity verification may be performed repeatedly on large volumes of meter readings, operational measurements, and transaction records. Lower execution time reduces processing latency and computational overhead, particularly in resource-constrained environments such as smart meters, edge gateways, and distributed monitoring devices. Consequently, a more efficient hash function enables integrity verification to be executed more frequently without significantly affecting system performance. Overall, the computational performance evaluation indicates that BLAKE2b-256 provides the highest execution efficiency under the experimental conditions employed in this study. However, this performance advantage should not be interpreted as evidence of superior cryptographic security, since the avalanche effect

and consistency evaluations demonstrated that all three evaluated algorithms exhibit comparable diffusion characteristics and deterministic behavior. Therefore, the primary advantage of BLAKE2b-256 lies in its computational efficiency, whereas the selection of an appropriate cryptographic hash algorithm for Smart Grid data integrity verification should consider both performance requirements and implementation constraints.

3.4 Discussion

The experimental results demonstrate that SHA-256, SHA3-256, and BLAKE2b-256 exhibit comparable cryptographic characteristics for Smart Grid data integrity verification[1]. The avalanche effect evaluation indicates that all three algorithms consistently produced output changes close to the theoretical expectation of approximately 50%, which is widely recognized as the ideal diffusion criterion for cryptographic hash functions[11]. This behavior confirms that a minimal modification in the input data propagates extensively throughout the generated hash value, thereby making unauthorized data modification readily detectable[16]. Since the observed differences among the evaluated algorithms were negligible, the avalanche effect results should not be interpreted as evidence that one algorithm provides stronger cryptographic security than another. Instead, the findings demonstrate that all evaluated algorithms satisfy the diffusion property required for reliable hash-based integrity verification[35]. Although the avalanche effect provides an important indication of diffusion capability, it should not be regarded as a comprehensive measure of cryptographic security[43]. A secure hash function must also satisfy additional security requirements, including collision resistance, preimage resistance, second-preimage resistance, and resilience against practical cryptographic attacks. Therefore, evaluating the avalanche effect alone is insufficient for determining the overall security of a cryptographic hash algorithm. The results obtained in this study should consequently be interpreted as evidence of effective diffusion performance rather than definitive proof of superior cryptographic strength[17].

The hash consistency evaluation further confirms the deterministic behavior of the evaluated algorithms. Identical Smart Grid data consistently produced identical hash values during repeated executions, demonstrating that the implemented hashing procedures are stable and reproducible[11]. This deterministic property is fundamental for integrity verification because newly generated hash values can be directly compared with previously stored reference digests to determine whether measurement data have

been altered. In practical Smart Grid environments, this verification mechanism can be applied to smart meter readings, operational measurements, energy transaction records, and communication logs to ensure that transmitted or stored information remains authentic throughout the data lifecycle[14].

From an operational perspective, cryptographic hash functions can be incorporated into multiple stages of the Smart Grid communication workflow[40]. After measurement data are generated by smart meters, a cryptographic hash can be computed before the data are transmitted to edge gateways or control centers[1]. Upon reception, the hash value can be regenerated and compared with the original reference digest to verify data integrity before the information is processed or stored. Similarly, hash values may be associated with audit logs and blockchain-based transaction records to detect unauthorized modifications while preserving the integrity of historical operational data[8]. These capabilities demonstrate that cryptographic hash functions provide an efficient integrity verification mechanism for distributed Smart Grid infrastructures without introducing substantial computational complexity[11].

While the evaluated algorithms demonstrated comparable avalanche behavior and identical consistency characteristics, the computational performance evaluation identified measurable differences in execution efficiency. BLAKE2b-256 consistently required less computational time than SHA-256 and SHA3-256 under identical benchmarking conditions[11]. This performance advantage is primarily attributable to the efficient internal design of the BLAKE2b compression function, whereas SHA3-256 employs a sponge-based construction that generally requires greater computational effort. Nevertheless, the superior execution efficiency of BLAKE2b-256 should not be interpreted as evidence of stronger cryptographic security[5]. Rather, the principal advantage observed in this study is reduced computational overhead, which may be particularly beneficial for Smart Grid environments that continuously process large volumes of measurement data or operate on resource-constrained devices such as smart meters and edge gateways[17].

Despite the encouraging findings, several limitations should be acknowledged. First, the evaluation was conducted using a single Smart Grid data attribute rather than complete measurement records, which may not fully represent the complexity of operational Smart

Grid datasets[41]. Second, the experimental evaluation focused on avalanche effect, deterministic consistency, and computational performance, while other important security properties, including collision resistance, preimage resistance, second-preimage resistance, and resistance to practical cyberattacks, were beyond the scope of this study. Third, the computational benchmark was performed on a desktop computing platform rather than embedded Smart Grid devices, where hardware constraints may significantly influence execution performance[21]. Future research should therefore investigate complete Smart Grid records, integrate digital signatures and message authentication mechanisms, evaluate blockchain-based audit trails, and perform experimental validation on embedded platforms under realistic Smart Grid attack scenarios, including false data injection and communication tampering[44].

4. CONCLUSION

This study presented an experimental evaluation of three cryptographic hash algorithms, namely SHA-256, SHA3-256, and BLAKE2b-256, for Smart Grid data integrity verification by analyzing avalanche effect, hash consistency, and computational performance[17]. Experimental results demonstrated that all evaluated algorithms produced avalanche percentages close to the ideal cryptographic value of approximately 50%, indicating comparable diffusion characteristics under a standardized 256-bit digest length. Hash consistency testing using 60,000 Smart Grid records further confirmed deterministic behavior, with all algorithms achieving a 100% consistency rate. Computational benchmarking showed that BLAKE2b-256 required the lowest execution time, while SHA-256 and SHA3-256 remained competitive due to their extensive standardization and widespread adoption. These findings indicate that algorithm selection should be guided primarily by application requirements rather than differences in diffusion capability, since all three algorithms exhibited comparable avalanche behavior[35]. Nevertheless, this study is limited to evaluating hash-function behavior using a single Smart Grid data attribute and does not include collision resistance, preimage resistance, digital signatures, blockchain deployment, embedded-device benchmarking, or real-world cyberattack simulations[39]. Future research should investigate complete Smart Grid communication workflows, full-record hashing, blockchain-based integrity verification, embedded implementations, and simulated attack scenarios to provide a more comprehensive evaluation of cryptographic hash deployment in Smart Grid environments[19],[14].

ACKNOWLEDGMENT

The authors would like to express their sincere gratitude to the Department of Informatics, Universitas Ahmad Dahlan, Yogyakarta, Indonesia, for providing the facilities and academic environment that supported this research. The authors also sincerely thank the anonymous reviewers for their valuable comments and constructive suggestions, which have significantly improved the quality, clarity, and scientific rigor of this manuscript.

REFERENCES

[1] D. H. Sinaga, R. Rifai, O. Sasue, and H. D. Hutahaean, "Pemanfaatan Energi Terbarukan Dengan Menerapkan Smart Grid Sebagai Jaringan Listrik Masa Depan," *zetroem*, vol. 03, pp. 11–17, 2021, doi: 10.36526/ztr.v3i1.1251.

[2] G. Dileep, "A survey on smart grid technologies and applications," *Renew. Energy*, vol. 146, pp. 2589–2625, 2020, doi: 10.1016/j.renene.2019.08.092.

[3] V. Arzamasov, K. Böhm, and P. Jochem, "Towards Concise Models of Grid Stability," no. 2, pp. 0–5.

[4] A. Shamaseen, M. Qatawneh, and B. Elshqeirat, "Smart Grid System Based on Blockchain Technology for Enhancing Trust and Preventing Counterfeiting Issues," *Energies*, vol. 18, no. 13, pp. 1–24, 2025, doi: 10.3390/en18133523.

[5] S. P. ANGGRAENI, "Penerapan Algoritma Kriptografi Sha-256 Dan Teknologi Blockchain Untuk Keamanan Citra Digital," *UNISSULA*, vol. 11, no. 1, pp. 1–14, 2025.

[6] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart grid - The new and improved power grid: A survey," *IEEE Commun. Surv. Tutorials*, vol. 14, no. 4, pp. 944–980, 2012, doi: 10.1109/SURV.2011.101911.00087.

[7] J. S. G. Sinaga, Nehemia Sitorus, and Steven Lukas Samosir, "Analisis Kinerja Algoritma Hash pada Keamanan Data: Perbandingan Antara SHA-256, SHA-3, dan Blake2," *J. QUANCOM QUANTUM Comput. J.*, vol. 2, no. 2, pp. 9–16, Dec. 2024, doi: 10.62375/jqc.v2i2.432.

[8] F. B. Fajrin, Ahmad Mifta, "Analisis Performa Algoritma BLAKE2b dan SHA-256 pada Implementasi Blockchain," *KESATRIA*, vol. 6, no. 2, pp. 451–460, 2025, doi: 10.30645/kesatria.v6i2.588.

[9] M. S. L. Muhammad Hakim Sadiq, Abdul Wahid, "Implementation of One-Time Password and SHA-3 Algorithm on the Lab Inventory Website of the Department of Informatics and Computer Engineering," *IOTA*, vol. 05, 2025, doi: 10.31763/iota.v5i2.914.

[10] H. Eldo, A. Ayuliana, D. Suryadi, G. Chrisnawati, and L. Judijanto, "Penggunaan Algoritma Support Vector Machine (SVM) Untuk Deteksi Penipuan pada Transaksi Online," *J. Minfo Polgan*, vol. 13, no. 2, pp. 1627–1632, 2024, doi: 10.33395/jmp.v13i2.14186.

[11] J. Sugier, "Reducing Power of BLAKE3 implementations with dedicated FPGA resources," *Int. J. Electron. Telecommun.*, vol. 71, no. 3, pp. 1–7, 2025, doi: 10.24425/ijet.2025.153622.

[12] Z. Abdullah Jasim and A. Kadhim Hadi, "Optimizing Blockchain Network Performance Using Blake3 Hash Function in POS Consensus Algorithm," *IEEE Access*, vol. 13, no. March, pp. 44760–44774, 2025, doi: 10.1109/ACCESS.2025.3546723.

[13] 1 Azhar Mahmooda, Abid Khanb, Adeel Anjumc, Carsten Mapled, Gwanggil Jeone, "The version presented in WRAP is the author ' s accepted manuscript and may differ from the An Efficient and Privacy-preserving Blockchain-based Secure Data Aggregation in Smart Grids," pp. 0–37, 2023.

[14] G. Aldabbagh, O. Bamasag, L. Almasari, R. Alsaidalani, A. Redwan, and A. Alsaggaf, "Blockchain for Securing Smart Grids," *Int. J. Comput. Sci. Netw. Secur.*, vol. 21, no. 4, pp. 255–263, 2021, doi: 10.22937/IJCSNS.2021.21.4.31.

[15] Ioan salomie. tudor ciora, claudia pop, Razvan Zanc, Ionut anghel ,arcel antal, "Smart Grid Management using Blockchain: Future Scenarios and Challenges Tudor," *IEEE Xplore*, vol. 2, no. 3, pp. 9–16, 2025, doi: 10.1109/RoEduNet51892.2020.9324874.

[16] S. Adilah, Rumani, Marisa, and Paryasto, "Implementasi Kriptosystem menggunakan metode Algoritma ECC dengan Fungsi Hash SHA-256 pada sistem ticketing online," *Univ. Telkom*, vol. 4, no. 3, pp. 4138–4146, 2017.

[17] A. Agung Aryasatya Daniswara and I. K. Dwi Nuryana, "Data Preprocessing Pola Pada Penilaian Mahasiswa Program Profesi Guru," *J. Informatics Comput. Sci.*, vol. 05, pp. 97–100, 2023, doi: 10.26740/jinacs.v5n01.p97-100.

[18] V. V. L. Minh Tuan Pham1*, Anh Tuan Hoang2, Anh Tuan Le1, Abdel Rahman M.Said Al-Tawaha3, Van Huong Dong2, "No Title 済無No Title No Title No Title," vol. 2, pp. 306–312, 2024.

[19] C. Chen, H. Guo, Y. Wu, B. Shen, M. Ding, and J. Liu, "A Lightweight Authentication and Key Agreement Protocol for IoT-Enabled Smart Grid System," *Sensors*, vol. 23, no. 8, pp. 1–16, 2023, doi: 10.3390/s23083991.

[20] M. Gladis Kurian and Y. Chen, "The Untapped Potential of Ascon Hash Functions: Benchmarking, Hardware Profiling, and Application Insights for Secure IoT and Blockchain Systems," *Sensors*, vol. 25, no. 19, pp. 1–26, 2025, doi: 10.3390/s25195936.

[21] O. P. Hafizh Fianto Putra, Wirawan, "Penerapan Blockchain dan Kriptografi untuk Keamanan Data pada Jaringan Smart Grid," *Jural Tek. its*, vol. 8, no. 1, 2019, doi: 10.12962/j23373539.v8i1.38525.

[22] Y. Rogi *et al.*, "Hash-Based Message Authentication Code with a Reverse Fuzzy Extractor for a CMOS Image Sensor †," *Electron.*, vol. 14, no. 10, 2025, doi: 10.3390/electronics14101971.

[23] A. Area-optimized, C. T. Ngo, J. K. Eshraghian, and J. Hong, "An Area-Optimized and Power-Efficient CBC-PRESENT and HMAC-PHOTON," pp. 1–16, 2022.

[24] jan pelzl christof paar, *Undertanding cryptography*. 2016.

[25] M. Tanveer and H. Alasmary, "LACP-SG: Lightweight Authentication Protocol for Smart Grids," *Sensors*, vol. 23, no. 4, pp. 1–18, 2023, doi: 10.3390/s23042309.

[26] W. Tushar, T. K. Saha, C. Yuen, D. Smith, and H. V. Poor, "Peer-to-Peer Trading in Electricity Networks: An Overview," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3185–3200, 2020, doi: 10.1109/TSG.2020.2969657.

[27] Q. Xie and Y. Luo, "Provably Secure and Lightweight Authentication Protocol for Smart Microgrids," *Symmetry (Basel).*, vol. 18, no. 5, pp. 1–20, 2026, doi: 10.3390/sym18050838.

[28] E. A.-C. Mohsen Hatami, Qian Qu Yu Chen, Javad Mohammadi, Erik Blasch, "ANCHOR-Grid: Authenticating Smart Grid Digital Twins Using Real-World Anchors," *MDPI*, vol. 25, pp. 0–25, 2024, doi: 10.3390/s25102969.

[29] M. R. Asghar, G. Dán, D. Miorandi, and I. Chlamtac, "Smart meter data privacy: A survey," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 4, pp. 2820–2835, 2017, doi: 10.1109/COMST.2017.2720195.

[30] A. Peivandizadeh *et al.*, "Lightweight and Efficient Protocol Based on ECDH for Securing Smart Grid Communication Infrastructure," *IEEE Access*, vol. 13, no. June, pp. 123666–123681, 2025, doi: 10.1109/ACCESS.2025.3585982.

[31] M. A. Muhammad Tanveer, Abd Ullah Khan, Habib Shah, Ahmed Alkhayyat, Shehzad Ashraf Chaudhry, "ARAP-SG : Anonymous and Reliable Authentication Protocol for Smart Grids," *IEEE Access*, vol. 9, pp. 143366–143377, 2021, doi: 10.1109/ACCESS.2021.3121291.

[32] S. A. Abdel Hakeem and H. Kim, "Centralized Threshold Key Generation Protocol Based on Shamir Secret Sharing and HMAC Authentication," *Sensors*, vol. 22, no. 1, 2022, doi: 10.3390/s22010331.

[33] J. Du, C. Dai, P. Mao, W. Dong, X. Wang, and Z. Li, "An Efficient Lightweight Authentication Scheme for Smart Meter," *Mathematics*, vol. 12, no. 8, pp. 1–17, 2024, doi: 10.3390/math12081264.

[34] I. Riadi, R. Umar, I. Busthomi, and A. Wirawan, "Block-hash of blockchain framework against man-in-the- middle attacks," *Register*, vol. 8, no. January, pp. 1–9, 2022, doi: 10.26594/register.v8i1.2190.

[35] V. N. Maulidya, "Implementasi Algoritma SHA-3 dan Mceliece dengan kode hamming untuk otentikasi dokumen berbasis Digital Signature," *UIN Malang*. 2025.

[36] T. L. Imam Riadi, Rusydi Umar, "Telematika Smart Payment Application Security Optimization from Cross-Site Scripting (XSS) Attacks Based on Blockchain Technology," *Telematika*, vol. 14, no. 2, pp. 74–85, 2021, doi: 10.35671/telematika.v14i2.1221.

[37] R. Canetti and H. Krawczyk, "Analysis of Key-Exchange Protocols," pp. 453–474, 2001.

[38] Y. Zhang, J. Chen, S. Wang, K. Ma, and S. Hu, "Lightweight Anonymous Authentication and Key Agreement Protocol for a Smart Grid," *Energies*, vol. 17, no. 18, 2024, doi: 10.3390/en17184550.

[39] R. Umar, I. Riadi, M. Ihya, and A. Elfatiha, "Security analysis of web-based academic information system using OWASP framework," vol. 4, no. 4, 2024.

[40] T. Cioara, C. Pop, R. Zanc, I. Anghel, M. Antal, and I. Salomie, "Smart grid management using blockchain: Future scenarios and challenges," *Proc. - RoEduNet IEEE Int. Conf.*, vol. 2020-Decem, 2020, doi: 10.1109/RoEduNet51892.2020.9324874.

[41] A. H. T. ZENG ZENG, MEIYA DONG, WEIWEI MIAO, MINGMING ZHANG, "A Data-Driven Approach for Blockchain-Based Smart Grid System," *IEEE Access*, vol. 9, pp. 70061–70070, 2021, doi: 10.1109/ACCESS.2021.3076746.

[42] Y. Guo, L. Li, X. Jin, C. An, C. Wang, and H. Huang, "Physical-Unclonable-Function-Based Lightweight Anonymous Authentication Protocol for Smart Grid," *Electron.*, vol. 14, no. 3, 2025, doi: 10.3390/electronics14030623.

[43] Nilda Aulia, "Prediksi Harga Ethereum Berdasarkan Informasi Blockchain Menggunakan Metode Long Short Term Memory," *Univ. Islam Indones.*, no. 9, pp. 1689–1699, 2020.

[44] Y. Guo, Z. Wan, and X. Cheng, "When blockchain meets smart grids: A comprehensive survey," *High-Confidence Comput.*, vol. 2, no. 2, pp. 0–1, 2022, doi: 10.1016/j.hcc.2022.100059.