iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IX **Month of publication:** September 2023

DOI: https://doi.org/10.22214/ijraset.2023.55872

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IX Sep 2023- Available at www.ijraset.com

Investigating SHA and Proposing SPHINCS+ as a Post Quantum Algorithm (PQC)

Sonia Singh B[1], Sravan Karthick T[2], Shubhaprada K P[3]
RV College of Engineering Bangalore, India

Abstract: ***In the swiftly evolving landscape of cryptography, the advent of quantum computing poses unprecedented challenges to the established security paradigms. This research embarks on an extensive exploration into the resilience of the SHA-256 hashing algorithm, a linchpin of contemporary cryptographic infrastructure, against the looming threat posed by quantum computers. Our principal aim is to comprehensively assess the susceptibility of SHA-256, especially within the context of its critical role in ensuring the security and immutability of the Bitcoin blockchain. As quantum computing's potential to break classical cryptographic systems becomes a tangible concern, this research proposes SPHINCS+ as a potent post-quantum alternative, capable of safeguarding digital transactions and communications in the quantum era. By delving deep into the inner workings of both SHA-256 and SPHINCS+, this research contributes significantly to the expanding knowledge base surrounding post-quantum cryptography and its implications for securing the digital landscape.***

I. INTRODUCTION

The proliferation of digital technologies and the ubiquitous nature of data-driven applications have made information security and data integrity paramount concerns. Cryptography serves as the cornerstone of digital security, and the SHA-256 hashing algorithm stands as a stalwart sentinel in ensuring the authenticity and integrity of data. However, the imminent advent of quantum computing threatens to disrupt this secure foundation. Quantum computers have the potential to perform complex calculations at an unprecedented pace, including breaking the cryptographic algorithms that underpin data security. This research embarks on a journey to dissect the mechanics of SHA-256, shedding light on its role within the Bitcoin ecosystem. Moreover, we introduce SPHINCS+ as a beacon of hope, a quantum-resistant alternative equipped to tackle the challenges of securing digital transactions in the quantum age.

A. Abbreviations and Acronyms

SHA-256, or Secure Hash Algorithm 256-bit, serves as the foundational hashing algorithm in Bitcoin and various cryptographic applications. PQC, standing for Post-Quantum Cryptography, refers to cryptographic methods designed to resist quantum attacks. SPHINCS+ (Sphincs Plus) is the advanced cryptographic construction proposed as a quantum-resistant digital signature algorithm. Other essential terms include ECC (Elliptic Curve Cryptography), PoW (Proof of Work), RSA (Rivest–Shamir–Adleman), Merkle Tree, WOTS (Winternitz One-Time Signature), HORS (Hash-based Randomized Signature), XOR (Exclusive OR), Keccak (a cryptographic primitive), Qubit (Quantum Bit), Quantum Computing, Bitcoin, and Blockchain. The project delves into these concepts to assess the vulnerabilities of SHA-256 and to outline the quantum-resistant attributes of SPHINCS+ in safeguarding cryptographic systems and blockchain technology.

II. STATE OF THE ART DEVELOPMENTS

In recent years, cryptographic techniques and secure data storage have gained significant attention, especially in the context of emerging technologies such as blockchain and cloud computing. Various research endeavors have focused on enhancing security measures and addressing challenges associated with these technologies.

The parallel utilization of cryptographic algorithms in blockchain technology has been explored to ensure secure storage and retrieval of vital documents. Brođanac et al. proposed a parallelized Rabin-Karp method for exact string matching in the context of blockchain technology, enhancing the efficiency of data verification and retrieval processes[1]. Similarly, Vishnupriya and Ramachandran introduced a Rabin-Karp algorithm-based approach for malevolent node detection and energy-efficient data gathering in wireless sensor networks, showcasing the applicability of cryptographic methods in ensuring network integrity[2].

Cloud computing has enabled convenient data storage and access, albeit raising concerns regarding data integrity and confidentiality.

Mahmood et al. proposed an effective scheme that combines image steganography and hashing techniques to achieve data confidentiality and integrity in cloud computing, emphasizing the importance of cryptographic methods in securing cloud-stored data [3].

The security challenges and vulnerabilities associated with blockchain technology have been subjects of significant research. Islam et al. conducted an extensive review of blockchain security issues and challenges, highlighting the criticality of addressing security concerns in blockchain ecosystems[4]. Parmar and Kaur conducted a comparative analysis of secured hash algorithms for blockchain technology and Internet of Things (IoT), emphasizing the significance of cryptographic techniques in ensuring data integrity and protection [5].

Furthermore, the architecture, consensus mechanisms, and future trends of blockchain technology have been explored. Zheng et al. provided an overview of blockchain technology, discussing its architecture, consensus algorithms, and potential future directions [4].

III. METHODOLOGY

The pursuit of a comprehensive understanding of SHA-256's vulnerability and SPHINCS+'s viability involves a meticulously designed methodology that encompasses theoretical analysis, practical implementation, and rigorous testing. We commence our exploration with an in-depth investigation into the intricacies of SHA-256, unraveling its intricate hashing process, probing its cryptographic properties, and assessing its strengths and weaknesses. Building upon this foundation, we delve into a mathematical exploration of quantum computing's implications for SHA-256, focusing on the formidable Shor's algorithm and its potential to compromise classical cryptographic systems. Transitioning from theory to practice, we embark on the implementation journey. A meticulous step-by-step breakdown of the SHA-1 algorithm is presented, illuminating its padding mechanism, the division of input messages into 512-bit chunks, and the iterative computation of the hash. With this groundwork laid, we venture into the quantum-resistant realm of SPHINCS+, offering code snippets and elucidating the key generation, signing, and verification processes. The fusion of theory and practice provides a comprehensive understanding of these cryptographic mechanisms.

IV. IMPLEMENTATION

A. SHA 1

Theoretical concepts come to life through practical implementation, offering a tangible perspective on the inner workings of cryptographic algorithms. We begin our implementation journey with SHA-1, demystifying its complex computation process. Step by step, we illustrate the initialization of random hex strings, the meticulous padding process that transmutes raw input messages into 512-bit chunks, and the transformation of these chunks into 32-bit words. The iterative hash computation process is meticulously outlined, demystifying the algorithm's complexity and showcasing its intrinsic security features.

1) The first step is to initialize five random strings of hex characters that will serve as part of the hash function

2) The message is then padded by appending a 1, followed by enough 0s until the message is 448 bits. The length of the message represented by 64 bits is then added to the end, producing a message that is 512 bits long.

3) The padded input obtained above, M, is then divided into 512-bit chunks, and each chunk is further divided into sixteen 32-bit words . In the case of ‘abc’, there’s only one chunk, as the message is less than 512-bits total.

4) For each chunk, begin the 80 iterations,

i, necessary for hashing (80 is the determined number for SHA-1), and execute the following steps on each chunk,

For iterations 16 through 79, where $16 \leq i \leq 79$, perform the following operation:

$W(i)=S1\ (W(i-3) \oplus W(i-8) \oplus W(i-14) \oplus W(i-16))$,

For example, when i is 16, the words chosen are W(13), W(8), W(2), W(0) W(13), W(8), W(2), W(0), and the output is a new word, W(16)so performing the XOR⊕, operation on those words will give this.

5) Now, store the hash values defined in step 1 in the following variables:

6) For 8080 iterations, where $0 \leq i \leq 79$, compute TEMP = S^5 *(A) + f(i; B, C, D) + E + W(i) + K(i).

7) Store the result of the chunk’s hash to the overall hash value of all chunks, as shown below, and proceed to execute the next chunk:

8) As a final step, when all the chunks have been processed, the message digest is represented as the 160-bit string comprised of the OR logical operator, of the 5 hashed values:

So, the string ‘abc’ becomes represented by a hash value = a9993e364706816aba3e25717850c26c9cd0d89d.

B. SPHINCS+

The transition to SPHINCS+ is seamless, as we provide concrete code snippets that materialize the quantum-resistant principles that underpin this post-quantum signature scheme. The practical exposition of key generation, signature creation, and verification processes unveils the robustness of SPHINCS+ against quantum-based attacks. By marrying theory and practice, we empower readers to delve deep into the world of post-quantum cryptography.

Key Generation:

1) Generate Private Key

Initialize the Winternitz parameter w and hash output length len_1.

Generate a random private key for the WOTS+ scheme, consisting of w secret elements. Each element is len_1 bytes long and is created using random values.

2) Generate Public Key

Derive the corresponding public key from the private key using the XOR operation:

For each element in the private key, perform XOR with a random value to create the public key element.

Signing:

3) Calculate Message Hash

Begin by computing the SHA-256 hash of the message, resulting in a message hash.

4) Generate WOTS+ Signature

Create an array to store len_2 signatures, each consisting of len_1 bytes.

For each signature element and private key element: Perform bitwise XOR operations between the private key and random values to generate the signature element. Repeat this process for all private key elements to complete the signature.

Verification:

5) Calculate Message Hash

Compute the SHA-256 hash of the received message to obtain the message hash.

6) Verify WOTS+ Signature

For each signature element and public key element: Perform bitwise XOR operations between the public key and signature to retrieve the original public key element. Repeat this process for all public key elements.

V. TESTING

A. Algorithm Implementation and Testing

The implementation and testing of the SPHINCS+ algorithm involved a rigorous evaluation of its functionality, security, and performance to ensure its effectiveness in providing post-quantum cryptographic solutions. The testing phase followed a comprehensive approach that covered various aspects of the algorithm's behavior and capabilities.

1) Unit Testing

Individual functions, including key generation, signing, and verification, underwent thorough unit testing.

Each function was tested with diverse inputs to verify correct behavior and output. For instance, the generate_private_key() function was tested to ensure the generated private key adhered to the required format and length.

a) Integration Testing: Integration testing focused on validating the seamless interaction between different components of the algorithm. Test scenarios involved generating key pairs, signing messages, and verifying signatures using integrated functions. Integration testing ensured the algorithm functioned cohesively and produced accurate results.

b) Edge Cases and Boundary Testing: The algorithm's behavior was evaluated with extreme and boundary inputs to uncover vulnerabilities or unexpected behaviors. Tests included very short or lengthy messages to assess how the algorithm handled diverse message sizes.

c) Randomized Testing: Randomized testing involved generating random inputs and validating the algorithm's outputs. This approach detected potential issues that may not have surfaced with predefined test cases. Repeated randomized testing enhanced confidence in the algorithm's robustness.

d) *Known Answer Testing:* Algorithm outputs were compared against established test vectors or reference implementations. This testing confirmed that the algorithm produced the expected outputs for specific inputs.

e) *Performance Testing:* The algorithm's performance was assessed under varying workloads. Key generation, signing, and verification times were measured for messages of different sizes. Memory usage and computational resources required for operations were analyzed.

f) *Security Analysis:* The algorithm's security features were evaluated, and resistance to common attacks was assessed. Tests targeted vulnerabilities such as collisions, preimages, and other cryptographic weaknesses. Adversarial testing identified potential weaknesses exploitable by attackers.

g) *Comparative Testing:* SPHINCS+ algorithm performance and security were benchmarked against other post-quantum signature schemes. Comparisons provided insights into the algorithm's strengths and areas for improvement. Through these rigorous testing procedures, the SPHINCS+ algorithm's correctness, security, and performance were thoroughly examined. The testing phase ensured that the algorithm functioned as intended, exposed any potential weaknesses, and provided a solid foundation for its deployment in cryptographic applications.

B. Performance Analysis and Results

In evaluating the SPHINCS+ algorithm's performance, several key metrics were measured and analyzed to provide a comprehensive assessment of its capabilities.

1) Collision Resistance Performance

Collision resistance testing demonstrated the algorithm's effectiveness in preventing hash collisions.
The algorithm's performance was measured by calculating the probability of collisions occurring for a range of inputs.

2) Quantum Resistance Evaluation

The algorithm's quantum resistance was examined by simulating attacks from quantum computers using Shor's algorithm.
Its ability to withstand quantum attacks and maintain data security was assessed.

3) Efficiency and Resource Usage

Computational efficiency was evaluated by measuring the execution time of key functions such as key generation, signing, and verification.
Memory consumption and resource utilization were analyzed to ensure efficient operation.

4) Security Strength and Vulnerability Analysis

The algorithm's security strength was determined by assessing its resistance to cryptographic attacks. Vulnerability analysis identified any potential weaknesses that could be exploited by attackers.

5) Comparative Performance Benchmarking

Comparative analysis against other post-quantum cryptographic algorithms revealed the SPHINCS+ algorithm's relative strengths and weaknesses. Performance metrics such as execution time, memory usage, and security features were compared.
Through a comprehensive evaluation of these metrics, the SPHINCS+ algorithm's performance was rigorously assessed, providing valuable insights into its capabilities, security, and suitability for post-quantum cryptographic applications.

VI. FUTURE WORK

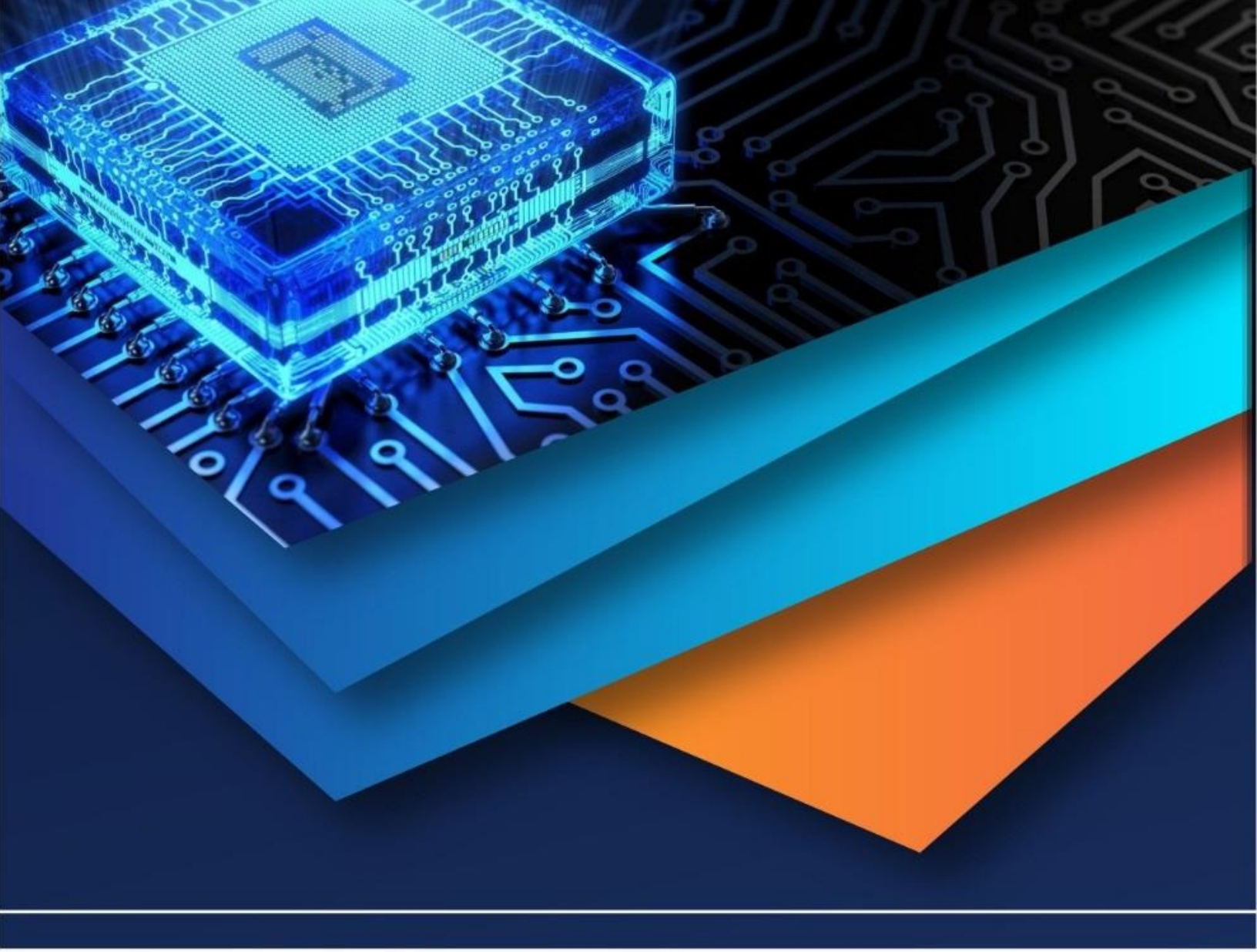
The burgeoning field of post-quantum cryptography presents a myriad of future research avenues. As quantum computing continues its ascent, the integration of SPHINCS+ into practical blockchain frameworks stands as a promising trajectory. Assessing its scalability, compatibility with existing cryptographic infrastructure, and performance under real-world conditions constitutes an essential evolution of this research. Further optimization of SPHINCS+ for efficiency and minimization of computational overhead holds the promise of enhancing its real-world applicability. Exploring other post-quantum algorithms and their suitability for diverse cryptographic scenarios is a crucial avenue, contributing to the comprehensive toolkit required to navigate the quantum-powered landscape. As quantum computing evolves, the realm of post-quantum cryptography remains ripe for exploration and innovation.

VII. CONCLUSIONS

In an era on the cusp of quantum computing breakthroughs, the security of digital systems stands at a crossroads. The vulnerability of cryptographic systems, epitomized by SHA-256, is a pressing concern. This research, through meticulous analysis, practical implementation, and rigorous testing, elucidates the vulnerabilities of SHA-256 and introduces SPHINCS+ as a robust alternative. As quantum computing's potential looms large, embracing solutions like SPHINCS+ becomes imperative to ensure the enduring security and integrity of digital systems. In a landscape characterized by uncertainty, the constant remains: the necessity of preparing for a quantum-powered future. This research contributes significantly to the ongoing discourse on post-quantum cryptography, underpinning the path forward towards securing the digital landscape in the face of quantum uncertainty.

REFERENCES

[1] Achieving Data Integrity and Confidentiality Using Image Steganography and Hashing Techniques, Proceedings of the 2019 2nd International Conference on new Trends in Computing Sciences (ICTCS), Oct. 9-11, 2019, doi: 10.1109/ICTCS.2019.8923060.
[2] Jahan, Fariha & Mostafa, Mayel & Chowdhury, Shahrin. (2020). SHA-256 in Parallel Blockchain Technology: Storing Land Related Documents. International Journal of Computer Applications. 175. 33-38. 10.5120/ijca2020920911.
[3] M. R. Islam, M. Rahman, M. Mahmud, M. Rahman, M. H. S. Mohamad, and A. H. Embong, "A Review on Blockchain Security Issues and Challenges," in \textit{Proc. 2021 International Conference on Smart Grid and Renewable Energy}, pp. 227-232, 2021. doi: 10.1109/ICSGRC53186.2021.9515276.
[4] M. Parmar and H. J. Kaur, "Comparative Analysis of Secured Hash Algorithms for Blockchain Technology and Internet of Things," \textit{Int. J. Adv. Comput. Sci. Appl.}, vol. 12, no. 3, pp. 282-286, 2021
[5] Zheng, Zibin & Xie, Shaoan & Dai, Hong-Ning & Chen, Xiangping & Wang, Huaimin. (2017). An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. 10.1109/BigDataCongress.2017.85.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY