Journal of Theoretical and Applied Information Technology
15th July 2025. Vol.103. No.13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

POSTQUANTUM MERKLE SIGNATURE BASED ON MODIFIED LAMPORT ALGORITHM

LARISA CHERCKESOVA[1*], ELENA REVYAKINA[1], NIKITA LYASHENKO[1]
[1] Don State Technical University, Department of Cyber Security of Information Systems, Russia
E-mail: larisacherckesova@gmail.com

ABSTRACT

With the advancement of quantum computing, conventional digital signature algorithms such as RSA and El-Gamal are increasingly vulnerable to quantum attacks. This presents a significant challenge for ensuring the security and integrity of electronic signatures. As a result, there is a need to develop post-quantum signature algorithms that can withstand attacks from quantum computers while maintaining computational efficiency. This study proposes a novel modification of the Merkle post-quantum signature scheme, integrating an optimized version of Lamport's one-time signature algorithm. The core contribution of this work is the design and implementation of a new algorithm that significantly reduces signature verification time without compromising cryptographic strength. A software implementation of the modified scheme was developed using Python and the PyQt 6 library, allowing for practical testing and analysis. The performance of the modified algorithm was compared against the standard Lamport algorithm using execution time measurements and statistical analysis. The experimental results demonstrate that the modified algorithm significantly improves signature verification speed, reducing the time required by up to 44.81% compared to the standard algorithm. The study presents a more efficient post-quantum Merkle signature scheme with a modified Lamport algorithm that enhances signature verification speed while maintaining strong cryptographic security. The results suggest that the proposed scheme is particularly well-suited for environments where fast authentication of multiple digital signatures is required. Experimental results confirm the advantage of the proposed approach, offering a more efficient solution for secure, high-speed digital authentication in post-quantum environments.

Keywords: *Post-Quantum Algorithm, Digital Signature, Merkle Signature Scheme, Lamport Signature.*

1. INTRODUCTION

In today's world, electronic digital signature algorithms play an increasingly important role in ensuring information security and countering cybercrime. Electronic digital signature plays an important role in ensuring the security of electronic documents and communications. The use of an electronic digital signature allows proving the absence of unauthorized changes to a document, establishing the ownership of the signature by the owner and ensuring the unrepudiation of the authorship of the signature.

The development of computer technology has led to the emergence of quantum computers, which allow attackers to hack common electronic signature algorithms based on RSA and El-Gamal algorithms. For this reason, it is necessary to develop and implement electronic digital signature algorithms that will be resistant to quantum algorithm attacks [1]. One of the post-quantum electronic signature algorithms is the signature algorithm based on the construction of Merkle tree.

The object of the study is Merkle's post-quantum electronic digital signature scheme based on Lamport's one-time signature algorithm.

The subject of the study is the computational complexity of Merkle's signature algorithm and its modifications.

The aim of the research is to develop a modified Merkle post-quantum signature algorithm using Lamport's one-time signature algorithm, which will be resistant to modern quantum attacks and for which the signature verification will be faster than the classical Lamport algorithm.

The core concern of this research stems from the growing threat posed by quantum computing to classical digital signature schemes, such as RSA and El-Gamal, which are foundational in current information security systems. As these traditional schemes become vulnerable to quantum-based attacks, the need for quantum-resistant solutions has become critical. Among the existing post-quantum

options, Merkle signature schemes offer promising properties of security and simplicity, yet their practical deployment is limited by performance bottlenecks, particularly during the signature verification stage. Therefore, the study focuses on optimizing the Merkle signature approach by modifying the underlying Lamport algorithm to address this key issue. This focus is justified by the urgent global demand for efficient, secure, and practical post-quantum cryptographic protocols that can be integrated into existing digital infrastructure without compromising speed or reliability.

Despite extensive research on post-quantum cryptographic algorithms, including modifications of Lamport and WOTS+ schemes, existing solutions often trade off performance for security or vice versa. As reviewed, some modifications accelerate key or signature generation at the cost of slower verification or increased memory requirements. Others achieve compactness but complicate hardware implementation. This fragmentation reveals a key research gap: there is a need for a post-quantum digital signature scheme that balances strong cryptographic resistance with practical efficiency, especially in the verification stage. Therefore, the core problem this study addresses is the development of a Merkle-based post-quantum signature system that significantly improves verification speed without compromising security.

Based on the identified research gap, we hypothesize that modifying the Lamport one-time signature algorithm and integrating it within a Merkle signature framework will significantly improve the verification speed of digital signatures while maintaining post-quantum cryptographic strength.

So, the following objectives were identified in line with the aim of the study:

– To study current publications related to the research of Merkle's post-quantum signature algorithm

– Develop a modification of Lamport's one-time signature algorithm

– Perform program implementation of the classical algorithm and modifications

- Implement a software tool for data encryption using a modified Merkle's post-quantum signature algorithm.

The purpose of this article is to develop and evaluate a modified Merkle post-quantum signature algorithm based on the Lamport one-time signature scheme, with improved verification performance and preserved cryptographic strength.

2. MATERIALS AND METHODS

2.1 Description of Merkle's Signature Algorithm

Merkle's signature is a reusable electronic digital signature algorithm that was published in 1979 by Ralph Merkle in the technical report "Secrecy, authentication, and public key systems" [2]. The algorithm allows multiple messages to be signed with a single public key using a one-time digital signature algorithm. The main advantage of the algorithm is its resistance to quantum computer attacks. This means that Merkle's algorithm can be used to construct a post-quantum electronic digital signature scheme.

A Merkle tree is a binary tree whose leaves contain hash values, and the nodes of the tree contain the hash of the concatenation of two values of the children nodes of the tree [3]. Let us consider the Merkle tree algorithm for a reusable electronic digital signature scheme:

1) Generate $N = 2^k$ key pairs (X, Y), where k is a natural number and each key pair represents the private key X and public key Y of a one-time digital signature scheme.

2) for each element Y_j of the public key array Y the value $H(Y_j)$ is calculated, where H is a cryptographic hash function. Each of these values is denoted as $a_{0,j}$. These values form the null layer of the Merkle tree.

3) For each natural number i from 1 to k, we compute $j = 2^{k-i}$ tree nodes, which are denoted as a_(i,j) and computed using the formula

$$a_{i,j} = H(a_{i-1,2j} \parallel a_{i-1,2j+1}) \quad (1)$$

The value $a_{k,0}$ is the public key of Merkle's signature algorithm.

Signature Generation Algorithm:

1) Select a key pair (X, Y) that has not previously been used for signature generation.

2) Generate a one-time signature S' using the key pair selected in the previous step.

3) Compute the authentication path that is required to verify the generated signature. The authentication path consists of k nodes of the generated Merkle tree. These nodes are chosen such that given only the chosen value $a_{0,i}$ and the authentication path, it is possible to compute the value $a_{k,0}$. For each integer n from 0 to k-1, the value that is part of the authentication path is defined as:

$$auth_n = a_{x_n,y_n} \quad (2)$$

where x_n, y_n for the chosen $a_{0,i}$ are defined by recurrence relations:

$$\begin{cases} x_0 = 0 \\ x_n = x_{n-1} + 1 \end{cases} \quad (3)$$

$$\begin{cases} y_0 = i - 2*(i \bmod 2) + 1 \\ y_n = \lfloor 0{,}5 * y_{n-1} \rfloor - 2*(\lfloor 0{,}5 * y_{n-1} \bmod 2 \rfloor) + 1 \end{cases} \quad (4)$$

A digital electronic signature *sig* generated using Merkle's algorithm is of the form:

$$S = S' \| Y_i \| auth_0 \| auth_1 \| \dots \| auth_{k-1} \quad (5)$$

Verification Algorithm:

1) Verify the one-time signature S′. If the verification of the one-time signature has not been passed, then the verification of the signature S' is also not passed. If the verification of S′ is successful, proceed to step 2.

2) Calculate $A_0 = H(Y_i)$

3) For each natural number j from 1 to k, compute:

$$A_j = H(A_{j-1} \| auth_{j-1}) \quad (6)$$

4) Compare the value of A_k with pub. If A_k = pub, the verification is successful. If A_k and pub are not equal, verification failed.

2.2 Description of Lamport One-Time Signature Algorithm

Lamport signature is a public key digital signature scheme that was proposed by Lamport in 1979 [4]. This algorithm is a one-time digital signature scheme. It means that one key pair can be used to sign only one message [5].

Lamport signature scheme consists of generation, signing and verification algorithms [6]. The result of key generation algorithm is a pair of keys (public key and private key).

1) Generate 256 pairs of random numbers of length 256 bits, which are denoted as $(X_{0,0}, X_{0,1}), (X_{1,0}, X_{1,1}) \dots (X_{i,0}, X_{i,1})$. These 512 numbers represent the private key.

2) For each of the 512 numbers generated in step 1 compute $Y_{i,j}$ using the formula:

$$Y_{i,j} = H(X_{i,j}) \quad (7)$$

The 512 values calculated in step 2 form the public key.

Signature generation algorithm:

1) Perform hashing of the message

2) For each bit b_i calculated at step 1 of the hash, a number X_{i,b_i}. is taken from the corresponding pair of numbers of the private key. The selected number is denoted as A_i. The selected 256 numbers constitute an electronic digital signature and are sent along with the message. The 256 numbers from the secret key that are not selected must be deleted to avoid signature forgery.

Verification Algorithm:

The recipient of the message must perform the following steps:

1) Calculate the hash of the message

2) For each of the signature numbers $A_0, A_1 \dots A_{255}$ compute:

$$Y_i' = H(A_i) \quad (8)$$

3) For each bit b'_i of the hash computed in step 1, compare the Y'_i and Y_{i,b'_i}. Verification is successful if the equality is satisfied for all i from 0 to 255:

$$Y_i' = Y_{i,b_i'} \quad (9)$$

If at least for one i the equality is not satisfied, the verification is considered as failed.

The main advantage of the Lamport signature algorithm is the high speed of signing and verification compared to other one-time signature algorithms (e.g., Winteritz signature). The disadvantage of the algorithm is the large size of the public key and signature [4].

3. LITERATURE REVIEW

One of the main approaches to improving post-quantum electronic digital signature algorithms is modification of existing one-time signature algorithms. The paper [7] discusses the application of Lamport algorithm in Internet of Things devices. The high performance of the algorithm allowed to create an effective authentication scheme for data transfer between Internet of Things devices. In the paper [8], the authors performed a performance comparison of Lamport signature algorithm when using different cryptographic hash functions. The authors concluded that increasing the hash length has almost no effect on the signature generation speed, but significantly slows down the key generation and signature verification. In [9], a modification of the WOTS+ one-time signature algorithm was developed. This modification speeds up key

generation by 25% and signature generation by 16.7%. The disadvantage of the modification is that verification requires 3.5 times more time compared to the standard algorithm. In the paper [10], modern attacks on the WOTS+ algorithm are discussed. Paper [11] applies conversion of binary numbers into non-adjacent form of number (non-adjacentform) to reduce the number of hashing operations performed during signature generation. In [12], a modification of the WOTS+ algorithm with a smaller signature size compared to the classical algorithm is implemented.

Over the last five years, several papers have been published proposing new post-quantum algorithms for one-time electronic digital signature. In [13] a one-time post-quantum signature algorithm is developed using provably secure SWIFFT family hash functions, which are based on fast Fourier transform. In [14], a new algorithm for one-time postquantum signature using Bloom filter is proposed. The Bloom filter is a data structure that allows checking the presence of an element in some set, but if the element is not in the set, its absence is determined only with some probability less than 1.

Another actual direction of research is optimization of the Merkle signature algorithm and its most common modification - the XMSS algorithm. In [15] a hardware implementation of the modified Merkle XMSS signature algorithm using the WOTS one-time signature algorithm was performed. In [16], an efficient implementation of XMSS for GPU is presented. In [17-19] optimization of XMSS algorithm for processors using RISC-V instruction system is performed.

4. RESULTS

4.1 Modification of Lamport Signature Algorithm

Key generation algorithm

1) Generate 512 random numbers of length 256 bits each. The set of these numbers must be divided into 128 subsets, each containing 4 numbers. The numbers in the subset with index i are denoted as $A_{i,0}, A_{i,1} A_{i,2}, A_{i,3}$. The resulting numbers represent the private key.

2) For each of the 512 numbers generated in step 1 calculate $Y_{i,j}$ using the formula:

$$Y_{i,j} = H(A_{i,j}) \quad (10)$$

The 512 values calculated in step 2 form the public key.

Signature generation algorithm:

1) Perform message hashing

2) The resulting hash is split into 128 bit pairs $p_0, p_1 \ldots p_{127}$

3) For each pair of bits p_i a number $A_{i,j}$, is taken, where index j is defined as the representation of the pair of bits in decimal notation (e.g., if p=11, then j=3). The selected number is denoted by A_i. The 128 numbers A_i for i from 0 to 127, together make up the electronic digital signature and are sent with the message. The remaining 384 numbers from the secret key that have not been selected must be deleted to avoid signature forgery.

Verification Algorithm:

The recipient of the message must perform the following actions:

1) Calculate the hash of the message

2) For each number A_i, calculate Y'_i using the formula:

$$Y_i^{'} = H(A_{i,j}) \quad (11)$$

3) Split the computed hash into 128 bit pairs $p'_0, p'_1 \ldots p'_{127}$

4) For each of the 128 pairs of p'_i bits, compare Y'_i and $Y_{i,j}$,, where the index j corresponding to each i is defined as the representation of the pair of p'_i, bits, in decimal notation (similar to step 3 of signature generation). Verification is successful if equality is satisfied for all i from 0 to 127:

$$Y_i^{'} = Y_{i,j} \quad (12)$$

If at least for one i the equality is not fulfilled, the verification is not passed.

To reduce the length of the private key, the modification uses a cryptographically strong random number generator. Cryptographically resistant pseudorandom number generator is an algorithm that allows generating a sequence of numbers that obey a given distribution and have statistical properties close to a sequence of random numbers [20, 21].

The use of Merkle's PRCSG allows us to significantly reduce the amount of memory required to store the private key. Instead of the private key, it is sufficient to store only one random number r of length 256 bits. When it is necessary to sign a message, the private key is generated using a PRNGPSS, where the number r is inputted during initialization.

4.2 Proof of the Cryptographic Strength of the Modified Algorithm

Theorem 1: If a cryptographic hash function H is resistant to the first-probe finding attack, then

solving the equation H(X||Y) = A for a given A and unknowns X and Y, is infeasible in practice.

Proof. Suppose that there exists an algorithm for fast finding of X and Y. In this case, this algorithm can be used to solve the equation H(X) = M (for this purpose it is enough to choose an arbitrary string and replace the variable X=X'||Y'), which contradicts the statement about the resistance of the function H to the attack of finding the first image.

Theorem 2: If the cryptographic hash function H is resistant to the first-sample attack, then the solution of the equation H(A||X) = B for given A and B and unknown X is infeasible in practice.

Proof. Suppose that there exists an algorithm for finding X quickly. In this case, this algorithm can be used to solve the equation H(X) = M (it is enough to choose an arbitrary string A' and substitute the variable X=A'||X'), which contradicts the statement about the resistance of the function H to the attack of finding the first sample.

Theorem 3: If the cryptographic hash function H is resistant to the first-sample attack, then the solution of the equation H(X||A) = B for given A and B and unknown X is infeasible in practice.

The proof is similar to the proof of Theorem 2, but when solving the equation H(X) = M, the variable X=X'||A' is replaced.

Using Theorems 1-3, let us prove the crypto-resistance of the modified algorithm of Lamport's electronic digital signature.

Suppose an attacker has generated some message M. To generate a signature that will pass verification using the published public key Y, the attacker needs to hash the message M, split the hash into 128 pairs of bits $p_0, p_1 \dots p_{127}$ and for each pair of bits p_i find two numbers B_i and C_i for which equality is satisfied:

$$Y'_{i,j} = H(B_i \parallel C_i) \qquad (13)$$

where the index j is defined by the decimal representation of the bit pair p_i.

By Theorem 1, this problem is infeasible in practice for a hash function H that is resistant to the first-probe finding attack.

Suppose an attacker tries to replace a message M for which an electronic digital signature has been generated with some message M'. In order for the previously generated signature to be successfully verified the attacker needs to compare pairs of bits $p_0, p_1 \dots p_{127}$ in the hash partition of message M with the corresponding pairs of bits $p'_0, p'_1 \dots p'_{127}$ in the hash partition of the message M' and for each i for which $p_i \neq p'_i$ find two numbers $B_i$and C_i, for which the equality (13) is satisfied, which as proved earlier is infeasible in practice. The attacker can also use as B_i the number X_i known from the signature or use as $C_i$the number A_i. In the first case, the attacker would need to solve the equation $Y'_{i,j} = H(X_i||C_i)$, which is infeasible in practice by Theorem 2, and in the second case, to solve the equation $Y'_{i,j} = H(B_i||A_i)$,, which is infeasible in practice by Theorem 3. This means that the modification of Lamport's signature algorithm is resistant to attacks by using a cryptographic hash function that is resistant to first and second prototype finding attacks.

4.3 Software Implementation

Based on the developed modification of the Merkle signature algorithm, a post-quantum electronic signature system is realized, which allows the user to perform Merkle tree generation, file signing and signature verification.

The software implementation is made in Python 3.10.10. The graphical interface of the software tool is implemented using the PyQt 6 library. As a cryptographic hash function the user can choose one of three algorithms - SHA256, SHA3-256 and GOST 34.11-2018 (for the GOST 34.11-2018 algorithm the version with a hash length of 256 bits is used). The urandom function, which is part of the standard os library of the Python language, is used as a cryptographically resistant pseudorandom number generator.

The program consists of three modules - Merkle tree generation module, which allows to generate Merkle tree for further use of the obtained keys for signature generation, file signing module, which provides signing of files using one-time Lamport signature keys and signature verification module, which allows to perform signature verification for previously signed files using the public key of the corresponding Merkle tree.

Figure 1 shows a block diagram of the Merkle tree generation module. As parameters during generation the user specifies the parameter N, which determines the number of messages that can be signed using the public key of the tree, and the hashing algorithm to be used. The public key is saved to a separate file.

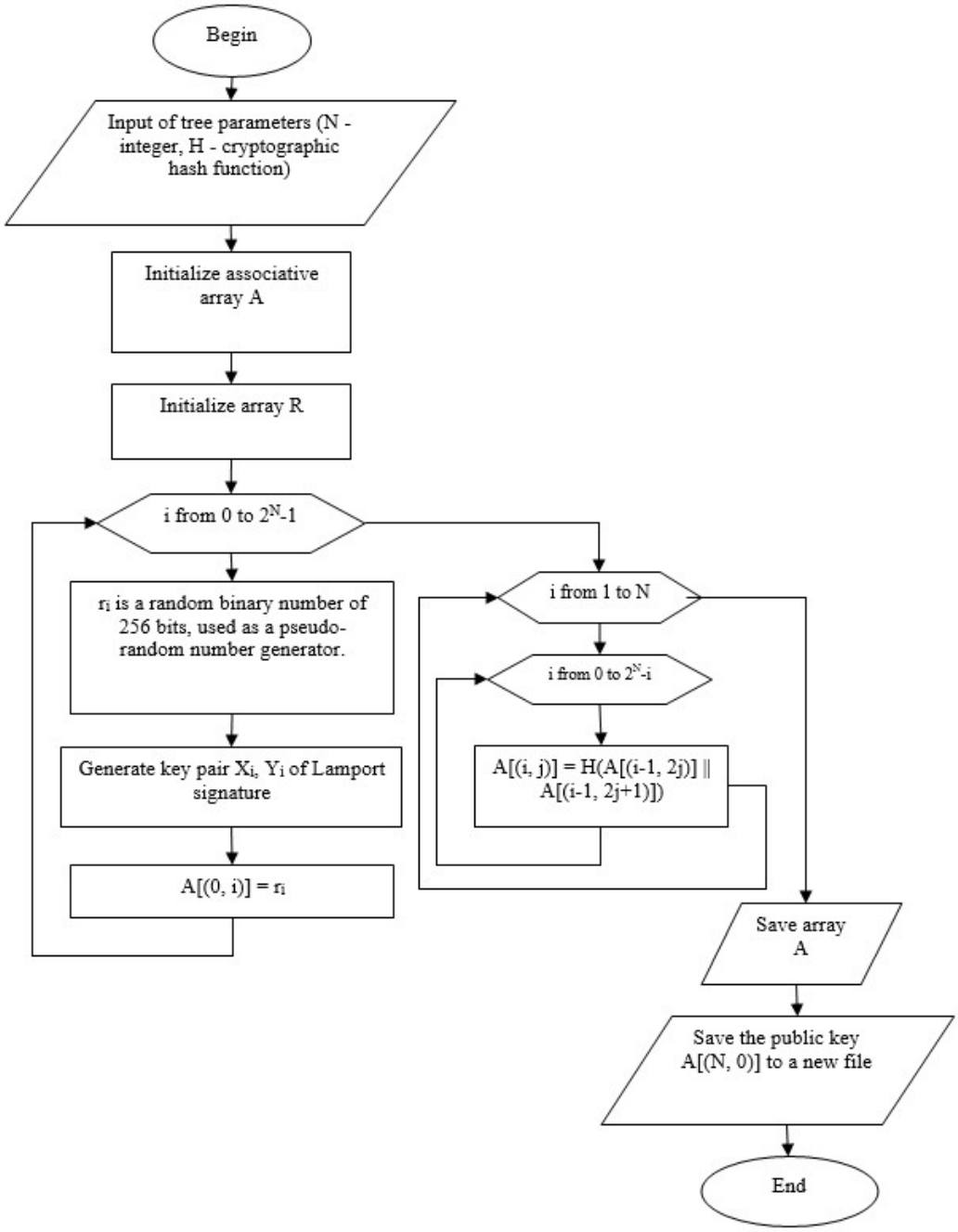

Figure 1: Flowchart of the Merkle Tree Generation Module

Figure 2 shows the block diagram of the electronic digital signature generation module. A signature is generated for each of the selected files and saved to a separate file.

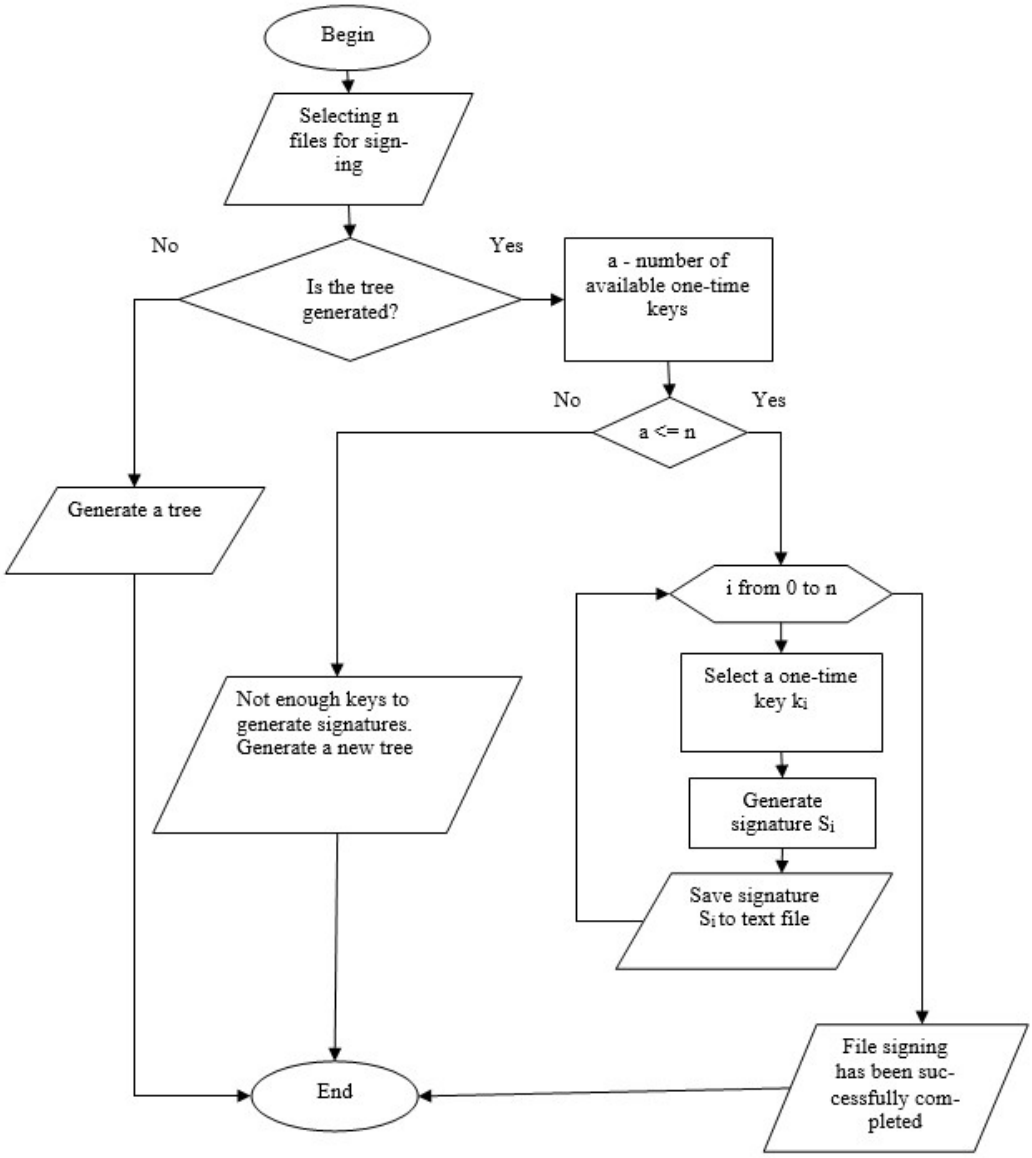

Figure 2: Flowchart of the Signature Generation Module

Figure 3 shows the schematic of the signature verification module. To perform the verification it is required to load previously signed files and a public key file.

Journal of Theoretical and Applied Information Technology
15th July 2025. Vol.103. No.13

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

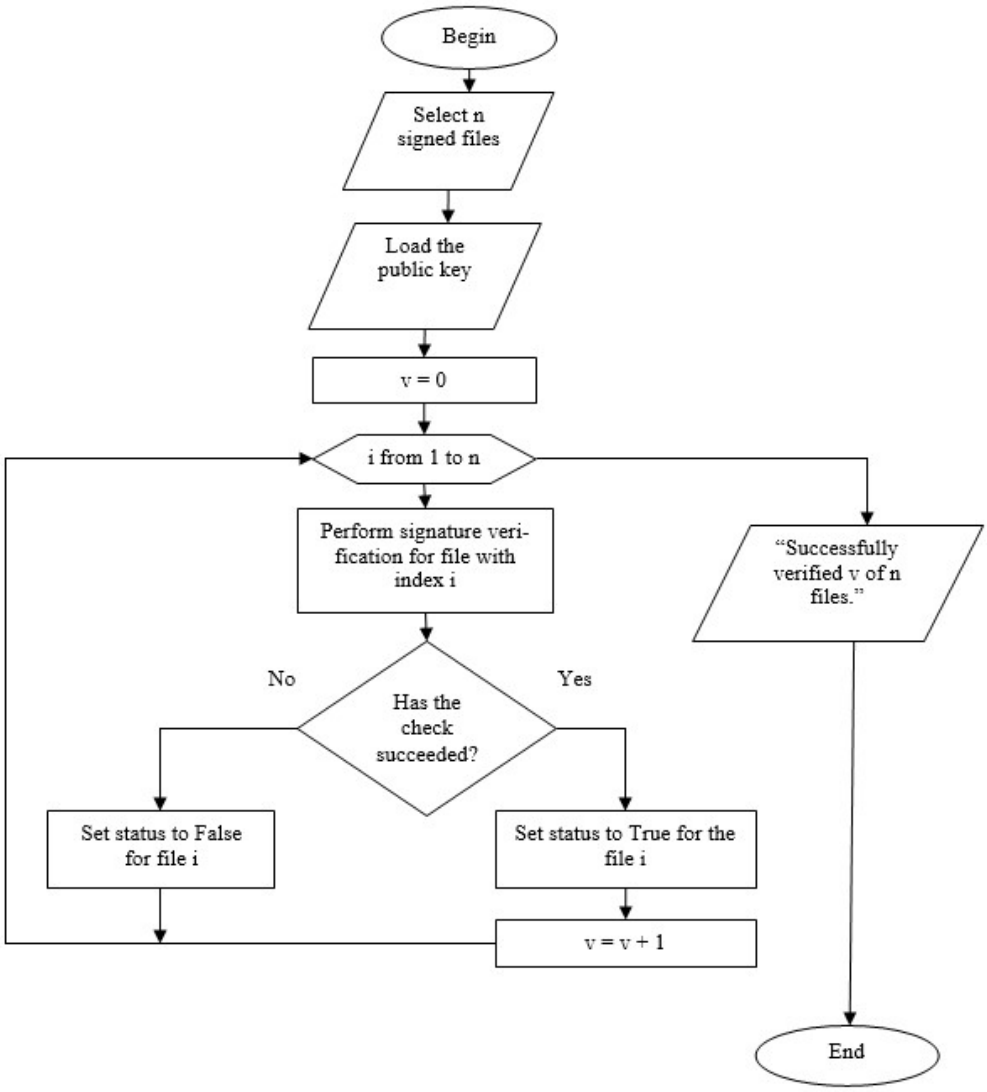

Figure 3: Flowchart of the Signature Verification Module

Figure 4 shows the main menu of the developed software tool. Each of the three main menu items corresponds to the previously described modules of the electronic digital signature generation system.

Figure 4: Main Menu of the Program

Figure 5 shows an example of successful Merkle tree generation. The screen provides information

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

about all previously generated trees, selected parameters and number of available keys for each tree.

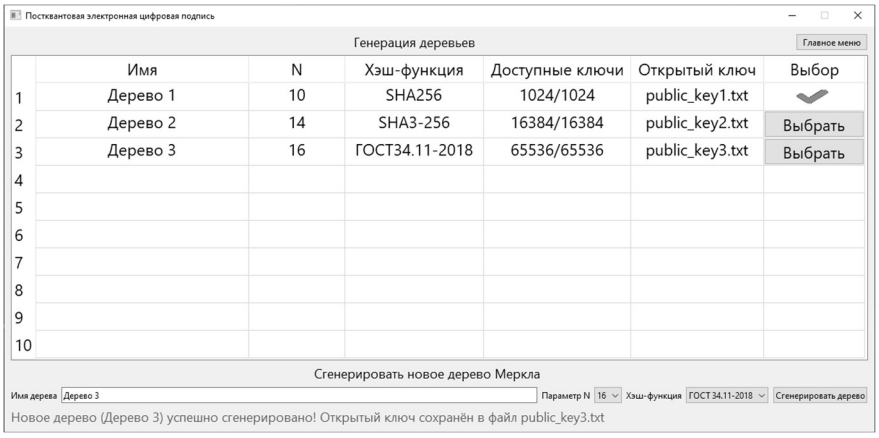

Figure 5: Example of Merkle Tree Generation

Figure 6 shows an example of successful signature generation for five user-selected files. The screen displays information about all files selected for signing and their corresponding signed files.

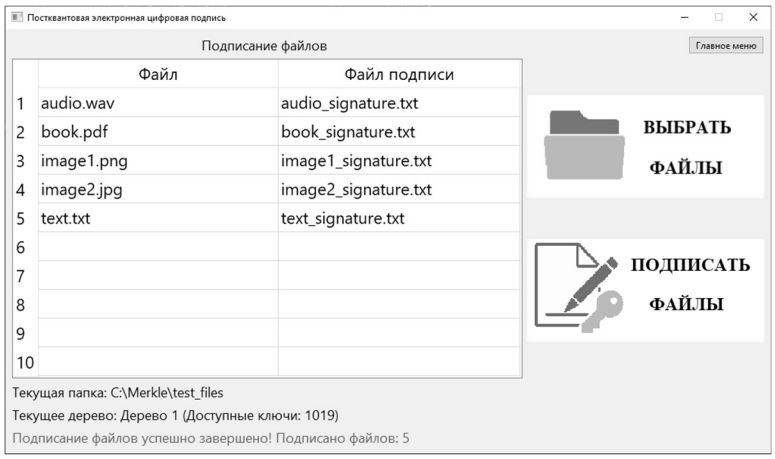

Figure 6: Example of Signature Generation

Figure 7 shows an example of successful signature verification for five previously signed files. The screen displays information about the selected files, their corresponding signature files and the result of signature verification.

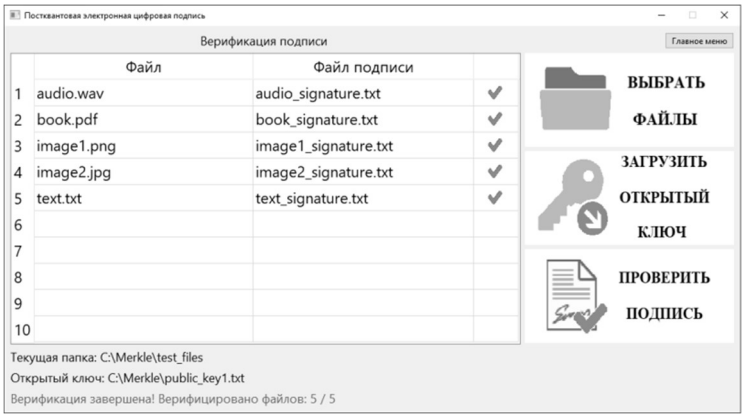

Figure 7: Example of Signature Verification

4.4 Testing of the Modified Algorithm

To compare the performance of the standard algorithm with the developed modification, a test was conducted, which consisted of calling the key generation, signing and signature verification functions 1,000,000 times for each of the two implementations of the Merkle signature algorithm. The SHA256 algorithm was used as the cryptographic hash function and the os.urandom function was used as a cryptographically robust pseudorandom number generator. The size of each message is 8 KB. The test results are presented in Table 1.

Table 1: Results of Algorithm Testing

	Execution time for the standard algorithm, s	Execution time for the modified algorithm, s	Result
Key generation	1, 7924	1,7665	Modification is faster by 1,44%
Signing	0,0439	0,0434	Modification is faster by 1,14%
Verification	0,5396	0,2978	Modification is faster by 44,81%

Testing was performed to compare the verification runtime for messages of different lengths using the standard and modified algorithms. For each possible n=16*k, where k is an integer from 16 to 1024, 10000 previously signed messages of size n kilobytes were verified using the standard and modified algorithm. Further, for each n the average time of signature verification when applying the standard and modified algorithm was calculated. The graphs shown in Figure 8 are plotted based on the test results. The algorithm testing was performed on a computer with Intel Core i5-2500K CPU 3.3 GHz.

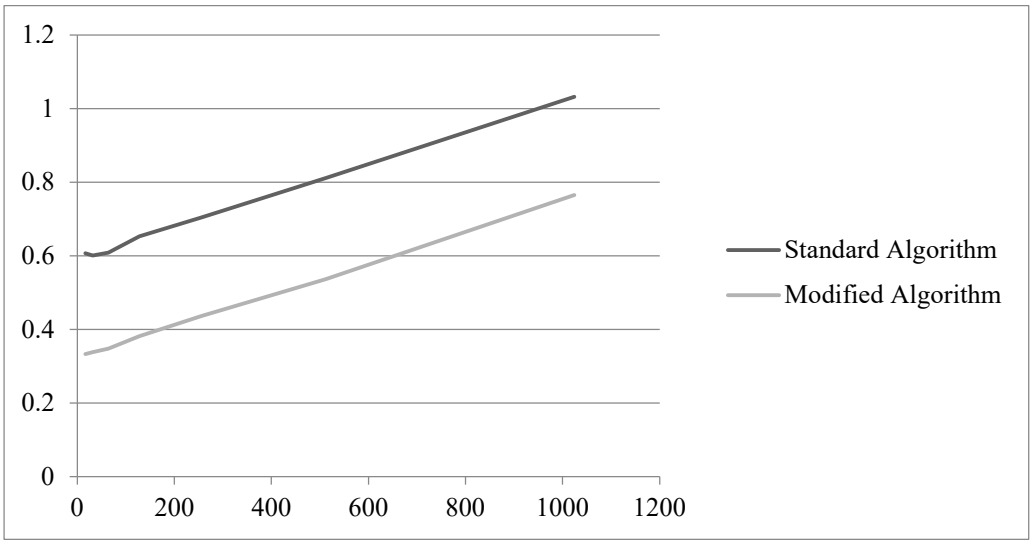

Figure 8: Perfomance Comparison of Signature Verification Between Standard and Modified Algorithm

Table 2 shows the test results of the standard and modified digital signature verification algorithm for messages of different sizes.

Table 2: Signature Verification Test Results for Messages of Different Sizes

Message size, KB	Average time of signature verification by standard algorithm, ms	Average time of signature verification by the modified algorithm, ms	Difference between the average signature verification time for the standard and modified algorithm, ms
16	0,6065	0,3335	0,273
32	0,6014	0,3379	0,2635
64	0,6087	0,3475	0,2612
128	0,6532	0,3818	0,2714
256	0,7045	0,4366	0,2679
512	0,8109	0,5361	0,2748
1024	1,032	0,7656	0,2664

The following conclusions are made on the basis of the performed tests:

1) Using the modified algorithm allows to speed up the signature verification.

2) The difference in time of signature verification between the two algorithms does not depend on the message length.

5. DISCUSSION

The results of the work are the developed modification of Merkle's post-quantum signature algorithm using Lamport's one-time signature algorithm and the system of electronic digital signature with a graphical interface realized on its basis. The advantage of the developed modification is a higher speed of signature verification. Testing has shown that the difference between the time of signature verification using the standard and modified algorithm does not depend on the message length (despite the fact that the total verification time linearly depends on the message length due to the need to calculate the message hash). It follows that it is most efficient to use the developed modification in applications where fast signature verification is required for a large number of messages of small (less than 1 MB) size.

In contrast to previous studies, such as those focusing on optimizing WOTS+ or introducing Bloom filter-based schemes [9,14], the proposed modification does not sacrifice verification time for improvements in key or signature generation. While some works [9] achieved faster signing but at the cost of significantly slower verification, our

approach ensures a substantial performance gain specifically during the verification stage, which is crucial for real-time applications like digital authentication systems and IoT devices.

The use of a cryptographically secure pseudorandom number generator also sets this work apart by allowing private keys to be generated on-demand, which reduces memory requirements. Compared to previous hardware-focused solutions like XMSS on GPUs or embedded systems [15–17], this work provides a flexible software implementation that is easier to deploy in existing systems without specialized hardware.

However, the proposed scheme also has limitations. Like all Merkle-based systems, it inherits the constraint of a limited number of signatures per generated key tree, which may require periodic key regeneration in long-running systems. Additionally, while the verification speed is improved, key generation and signing times remain only marginally faster than the classical version, which might not satisfy use cases where overall throughput is a priority.

Nonetheless, the study demonstrates a meaningful advancement in balancing cryptographic strength and operational efficiency. By focusing on verification speed the developed algorithm fills an important gap in the literature and offers a practical foundation for future enhancements and hardware acceleration.

6. CONCLUSION

The main result obtained as a result of the research is a modification of Merkle's post-quantum signature algorithm, which in comparison with the standard algorithm provides higher performance in signature verification. On the basis of the developed modification a software tool for generation and verification of electronic digital signature was realized. Performance testing was performed for two versions of the algorithm on messages of different lengths, which confirmed the effectiveness of the developed modification.

These findings directly address the initial problem posed in the study: the lack of a post-quantum signature scheme that balances strong cryptographic security with fast verification performance. The observed verification speedup of up to 44.81% across varying message sizes confirms the initial hypothesis that a carefully designed modification of Lamport's scheme within a Merkle tree structure can deliver practical performance benefits without weakening security. These results validate the proposed approach as a viable enhancement to post-quantum digital signature systems.

REFERENCES:

[1] A.V. Komarova, and A.G. Korobeynikov, "Analiz osnovnykh sushchestvuyushchikh postkvantovykh podkhodov i skhem elektronnoy podpisi [The analysis of existing post-quantum approaches and electronic signature schemes]", *Voprosy kiberbezopasnosti*, No. 2(30), 2019, pp. 58-68.

[2] Y.C. Chen, Y.P. Chou, and Y.C. Chou, "An image authentication scheme using Merkle tree mechanisms", *Future Internet*, Vol. 11, No. 7, 2019, Art. No. 149. https://doi.org/10.3390/fi11070149

[3] X. Wang, W. Lin, W. Zhang, Y. Huang, Z. Li, Q. Liu, X. Yang, Y. Yao, and C. Lv, "Integrating Merkle trees with transformer networks for secure financial computation", *Applied Sciences*, Vol. 14, No. 4, 2024, Art. No. 1386. https://doi.org/10.3390/app14041386

[4] K.N. Pankov, and Yu.B. Mironov, *Ispol'zovaniye postkvantovykh algoritmov v zadachakh zashchity informatsii v telekommunikatsionnykh sistemakh* [Usage of post-quantum algorithms in the problems of information security in telecommunication systems]. Goryachaya liniya – Telekom, Moscow, 2023.

[5] M. Iavich, T. Kuchukhidze, and R. Bocu, "A post-quantum digital signature using Verkle trees and lattices", *Symmetry*, Vol. 15, No. 12, 2023, Art. No. 2165. https://doi.org/10.3390/sym15122165

[6] T.B. Josey, and D.S. Misbha, "Man-in-the-middle attack mitigation in IoT sensors with hash based multidimensional Lamport digital signature", in R.J. Kannan, S. Geetha, S. Sashikumar, and C. Diver (Eds.), *International virtual conference on industry 4.0. IVCI 2021*, pp. 47-56. Springer, Singapore, 2023. https://doi.org/10.1007/978-981-19-9989-5_5

[7] G.M. Abdullah, Q. Mehmood, and C.B.A. Khan, "Adoption of Lamport signature scheme to implement digital signatures in IoT", in *2018 International conference on computing, mathematics and engineering technologies (iCoMET)*, Sukkur, Pakistan, March 2018, pp. 1-4. https://doi.org/10.1109/ICOMET.2018.8346359

[8] D. Zentai, "On the efficiency of the Lamport signature scheme", *Land Forces Academy Review*, Vol. 25, No. 3, 2018, pp. 275-280.

https://doi.org/10.2478/raft-2020-0033

[9] K. Zhang, H. Cui, and Y. Yu, "Revisiting the constant-sum Winternitz one-time signature with applications to SPHINCS+ and XMSS", in H. Handschuh, and A. Lysyanskaya (Eds.), *Advances in cryptology – CRYPTO 2023*, pp. 455-483. Springer, Cham, 2023. http://dx.doi.org/10.1007/978-3-031-38554-4_15

[10] M.A. Kudinov, E.O. Kiktenko, and A.K. Fedorov, "Analiz stoykosti skhemy podpisi W-OTS$^+$: Utochneniye otsenok stoykosti [Security analysis of the W-OTS$^+$ signature scheme: Updating security bounds]", *Matematicheskiye voprosy kriptografii*, Vol. 12, No. 2, 2021, pp. 129-145. https://doi.org/10.4213/mvk362

[11] D. Roh, S. Jung, and D. Kwon, "Winternitz signature scheme using nonadjacent forms", *Security and Communication Networks*, Vol. 2018, 2018, Art. No. 1452457. https://doi.org/10.1155/2018/1452457

[12] F. Shahid, A. Khan, S. Malik, and K. Choo, "WOTS-S: A quantum secure compact signature scheme for distributed ledger", *Information Sciences*, Vol. 539, 2020, pp. 229-249. https://doi.org/10.1016/j.ins.2020.05.024

[13] K. Kalach, and R. Safavi-Naini, "An efficient post-quantum one-time signature scheme", in O. Dunkelman, and L. Keliher (Eds.), *Selected areas in cryptography – SAC 2015*, pp. 331-351. Springer, Cham, 2015. https://doi.org/10.1007/978-3-319-31301-6_20

[14] M. Shafieinejad, and R. Safavi-Naini, "A post-quantum one time signature using bloom filter", in *15th Annual conference on privacy, security and trust (PST)*, Calgary, Canada, August, 2017, pp. 397-399. https://doi.org/10.1109/PST.2017.00056

[15] Y. Cao, Y. Wu, W. Wang, and X. Lu, "An efficient full hardware implementation of extended Merkle signature scheme", *IEEE Transactions on Circuits and Systems*, Vol. 69, No. 2, 2021, pp. 682-693. https://doi.org/10.1109/TCSI.2021.3115786

[16] Z. Wang, X. Dong, H. Chen, and Y. Kang, "Efficient GPU implementations of post-quantum signature XMSS", *IEEE Transactions on Parallel and Distributed Systems*, Vol. 34, No. 3, 2023, pp. 938-954. https://doi.org/10.1109/TPDS.2022.3233348

[17] W. Wang, B. Jungk, J. Walde, and S. Deng, "XMSS and embedded systems", in K. Paterson, and D. Stebila (Eds.), *Selected areas in cryptography – SAC 2019*, pp. 523-550. Springer, Cham, 2019. https://doi.org/10.1007/978-3-030-38471-5_21

[18] Yu.L. Nazarenko, "Kriptograficheskaya stoykost' generatorov sluchaynykh chisel. Algoritm Yarrou [Cryptographic stability of random number generators. Yarrow algorithm]", *European Science*, No. 10(32), 2019, pp. 24-29.

[19] E. Revyakina, L. Cherckesova, O. Safaryan, and N. lyashenko, "Improving performance, cryptographic strength of the post-quantum algorithm ntruencrypt and its resistance to chosen-ciphertext attacks", *Journal of Theoretical and Applied Information Technology*, Vol. 102, No. 1, 2024, pp. 186-194.

[20] L. Cherckesova, E. Revyakina, O. Buryakova, and A. Gazizov, "Creation of an encryption algorithm resistant to attacks through side channels of leakage", *E3S Web of Conferences*, Vol. 583, 2024, Art. No. 06011. https://doi.org/10.1051/e3sconf/202458306011

[21] L. Cherckesova, E. Revyakina, E. Roshchina, and V. Porksheyan, "The development of countermeasures against session hijacking", *E3S Web of Conferences*, Vol. 531, 2024, Art. No. 03019. https://doi.org/10.1051/e3sconf/202453103019