The evolution of digital signatures: From classical to post-quantum

Maksim Iavich[1,*,†]

[1] Caucasus University, 1 Paata Saakadze str., 0102 Tbilisi, Georgia

Abstract

The paper investigates the vulnerability of classical digital signatures, particularly RSA, in the face of advancing quantum computing. Shor's algorithm, which is capable of efficiently factoring large numbers on a quantum computer, poses a significant threat to traditional public-key cryptography. Therefore, the paper explores post-quantum digital signature schemes based on hashing, which are known for their resilience to quantum attacks. A key focus is the introduction and application of the Verkle Tree, a novel data structure, in the design of digital signatures. Our methodology, which uses Verkle Tree enhances security and efficiency in the post-quantum era, offering a practical methodology to counter quantum threats. The paper also offers post-quantum design concepts leveraging the Verkle Tree, offering the possibility of its integration into broader cryptographic protocols. In conclusion, the paper contributes to the discussion on the future of digital signatures by addressing classical vulnerabilities, introducing post-quantum alternatives, and proposing innovative design concepts with the Verkle Tree. This work illustrates the importance of quantum-resistant cryptographic solutions and provides practical and theoretical approaches for secure digital signatures in the post-quantum landscape.

Keywords

Verkle tree, quantum attacks, digital signature, RSA

1. Introduction

Recently, the leading engineers and scientists of the world are working on the creation of quantum computers. Recognized leaders in the development of quantum computers, the Google Corporation, the Association of D-WAVE and Space Research Universities, and the federal agency NASA are already ready to make a breakthrough in the field of quantum technologies. In October 2019, Google announced the achievement of quantum supremacy, which caused serious dispute, but given that the technological giants are in a hurry to create the first quantum computers and have made significant progress in this direction, the world may be approaching the beginning of a new era. Google claims that the new design of the chip can increase memory productivity ten times, from 100 to 1000 qubits. IBM is next as it declares that by the end of 2023, it will create a quantum processor with a capacity of more than 1000 qubits with about 50 logical qubits. It has already introduced a processor with 127 cubits in 2021 and a 433-cube processor in 2022. Chinese scientists also claim that "Zuchongzhi 2"—a 66-qubit quantum processor, completed the task 1 million times faster than the Google processor. This processor was developed by the research group of the Center for Achievements in Quantum Information and Quantum Physics of the Chinese Academy of Sciences together with the Shanghai Institute of Technical Physics and the Shanghai Institute of Microsystems and Information Technologies [1–5].

Someday, quantum computers will be able to break the existing cryptographic codes used for communication and financial transactions, so the digital signature systems used today are immune to quantum computer attacks, so the world must adopt quantum-resistant cryptography at the core level. Digital signature system security today is based on the problem of calculating discrete logarithms and factoring large numbers [6–9]. Cryptosystems in use today, such as RSA with four thousand bits of keys, are useful against classical computer attacks but are completely useless against quantum computer attacks.

The Daytime RSA cryptosystem is used almost everywhere, it is used by many large organizations such as government agencies, corporations, banks, government, and not only laboratories and universities. In addition, RSA is used in commercial products, hardware, operating systems, Ethernet, network smart and cards, and also in cryptographic equipment. With a total of about 500 million users, the RSA is one of the most widely used public key cryptosystems. So hacking RSA can lead to complete chaos. Today's challenge is to create alternatives to RSA that can withstand the attacks of quantum computers. As an alternative to RSA, digital signature hashing schemes based on a cryptographic hash function can be considered. The security of this signature is based on the collision resistance of the hash function.

CQPC-2024: Classic, Quantum, and Post-Quantum Cryptography, August 6, 2024, Kyiv, Ukraine

* Corresponding author.

† These authors contributed equally.

miavich@cu.edu.ge (M. Iavich)

0000-0002-3109-7971 (M. Iavich)

2. Digital signatures

Digital signatures rely on asymmetric cryptography, which involves a pair of keys: a private key for signing and a public key for verification. The mathematical foundation often involves algorithms like RSA, DSA, or ECDSA. The private key creates the signature, and the public key verifies it [10–12].

Signing process:

1. Hashing:

- Before signing, the document or message is usually hashed. A hash function condenses the data into a fixed-size string of characters.
- The hash ensures that even a slight change in the document will result in a vastly different hash value.

2. Private Key Encryption:

- The hash is then encrypted with the sender's private key to create the digital signature.
- The private key ensures that only the sender could have created this specific signature.

Verification process:

1. Hashing:

- The recipient hashes the received document using the same hash function used by the sender.

2. Public Key Decryption:

- The recipient decrypts the digital signature using the sender's public key, revealing the original hash value.

3. Comparison:

- The decrypted hash is compared with the hash of the received document.
- If the two hashes match, it confirms that the document has not been altered, and the signature is valid.

Non-Repudiation: Digital signatures provide non-repudiation, meaning the sender cannot deny their involvement or the authenticity of the signed document. This is because only the sender possesses the private key needed to generate that specific signature.

Key Management: The security of digital signatures relies heavily on proper key management. Safeguarding private keys is crucial to prevent unauthorized access and potential forgery. Key generation, storage, and distribution should follow secure practices.

Time Stamping: To address the issue of the validity period of digital signatures, timestamping services are often used. They provide proof that the signature existed at a particular time, adding another layer of security and trust.

Use Cases: Digital signatures find applications in various fields, such as:

- Document Authentication: Ensuring the integrity and authenticity of digital documents.
- Financial Transactions: Verifying the origin and integrity of financial transactions.
- Software Distribution: Ensuring that software has not been tampered with during distribution.
- Legal Contracts: Providing a digital equivalent to handwritten signatures in legal documents.

Challenges: Despite their effectiveness, digital signatures face challenges like evolving cryptographic standards, quantum computing threats, and the need for widespread adoption to achieve their full potential.

3. RSA-based digital signatures

RSA, named after its inventors Ron Rivest, Adi Shamir, and Leonard Adleman, serves as a foundational public-key cryptosystem widely employed for digital signatures. At its core, RSA involves a pair of keys—a public key and a private key. The public key, comprising a modulus (n) and an exponent (e), is openly shared, while the private key, including(n) and another exponent (d), remains confidential. The system's security hinges on the intricate challenge of factoring large prime numbers, forming the basis of its strength [13–16].

In the generation of digital signatures using RSA, a meticulous process unfolds. Before signing, the document undergoes hashing, typically utilizing a secure hash function like SHA-256. This hashed value is then padded, an important step in ensuring a consistent size and mitigating specific vulnerabilities, often employing schemes such as PKCS#1 v1.5 or PSS. Subsequently, the padded hash is encrypted using the sender's private key, resulting in the creation of the digital signature [17, 18].

On the recipient's end, the process of verifying the digital signature unfolds. The received document is hashed, mirroring the same hash function employed by the sender. Simultaneously, the recipient decrypts the digital signature using the sender's public key. Successful matching between the decrypted signature and the hash value validates the signature, assuring the document's integrity and authenticity.

The strength of RSA-based digital signatures is based on their security features. The complexity of factoring large numbers contributes to robust security. The choice of key length plays a pivotal role, with longer keys offering heightened security albeit potentially demanding more computational resources. Despite RSA's general slowness compared to symmetric-key algorithms, its computational cost for digital signatures is typically deemed acceptable. Careful consideration of padding schemes, such as PKCS#1 v1.5 and PSS, is essential to fortify security against potential vulnerabilities.

In practical applications, RSA-based digital signatures find extensive use. They authenticate the source of messages in secure communication protocols like SSL/TLS, ensure the authenticity and integrity of digitally signed

documents, and play a crucial role in the generation of digital certificates.

However, challenges persist. The need for longer key lengths to thwart evolving threats and the looming potential of quantum computers pose considerations. Additionally, the computational overhead of RSA, especially in resource-constrained environments, remains an ongoing concern.

RSA-based digital signatures continue to be usable in classical cryptography, offering a reliable means to ensure the authenticity and integrity of digital information. Ongoing research is essential to address emerging challenges and enhance the security of RSA-based systems as technology evolves.

RSA, as a widely used public-key cryptosystem, is vulnerable to attacks by quantum computers due to its reliance on the difficulty of factoring large numbers. Quantum computers, with their potential to perform certain calculations exponentially faster than classical computers, pose a significant threat to traditional cryptographic algorithms.

The most notable algorithm for factoring large numbers efficiently on a quantum computer is Shor's algorithm. Shor's algorithm can factorize large numbers in polynomial time, rendering the security assumptions of RSA obsolete. As a result, the security of RSA and other widely used public-key cryptography systems, such as ECC (Elliptic Curve Cryptography), is compromised in the era of quantum computing.

To address the threat posed by quantum computers, the cryptographic community is actively exploring and developing quantum-resistant or post-quantum cryptographic algorithms. These algorithms aim to maintain security even in the face of quantum attacks. Some proposed post-quantum alternatives include lattice-based cryptography, hash-based cryptography, code-based cryptography, and multivariate polynomial cryptography.

Transitioning from RSA to quantum-resistant algorithms is crucial for maintaining the security of digital signatures and other cryptographic applications in the post-quantum era. Organizations and researchers are working collaboratively to standardize new cryptographic algorithms that can withstand quantum attacks.

In the context of digital signatures, the move to quantum-resistant algorithms ensures that the integrity and authenticity of signed documents remain secure even as quantum computing capabilities advance. While RSA has been a workhorse for secure communication and digital signatures, the ongoing development and adoption of quantum-resistant algorithms are essential for preparing for the future landscape of quantum computing threats.

4. Elliptic curve cryptography digital signatures

Elliptic Curve Cryptography (ECC) digital signatures are at the forefront of modern public-key cryptography, striking a delicate balance between stringent security requirements and computational efficiency.

ECC leverages the intricate properties of elliptic curves over finite fields to facilitate cryptographic operations effectively. During the key generation phase, ECC entails the creation of a key pair comprising a private key and its corresponding public key. The private key, selected at random, serves as the cornerstone for deriving the public key through elliptic curve computations [19–21].

To generate an ECC digital signature, the message undergoes hashing using a cryptographic hash function, resulting in a fixed-size hash value. A critical aspect involves the generation of a random nonce to thwart signature predictability.

Subsequently, the ECDSA algorithm computes a pair of values, typically represented as (r, s), constituting the digital signature associated with the hashed message. Verification of an ECC digital signature necessitates hashing the received message and reconstructing a point on the elliptic curve using the public key and elliptic curve operations. The computed point, combined with the received signature, undergoes mathematical scrutiny to validate its authenticity.

Successful verification establishes the genuineness and integrity of the message. ECC's robust security is achieved through shorter key lengths in comparison to conventional algorithms, rendering it computationally efficient. As a versatile cryptographic tool, ECC digital signatures find utility in various domains such as secure communication protocols, authentication mechanisms, and digital certificates.

With its resilience against quantum attacks, ECC serves as a cornerstone in the continual advancement of cryptographic solutions. Quantum computers pose a potential threat to certain public-key cryptography algorithms, including Elliptic Curve Cryptography (ECC), which underpins ECC digital signatures.

The primary quantum algorithm posing a threat to ECC is Shor's algorithm. Shor's algorithm proficiently factors large numbers and computes discrete logarithms, which are foundational challenges for breaking RSA and ECC, respectively.

5. Hash-based digital signature schemes

Hash-based digital signature schemes are a class of cryptographic algorithms that rely on the properties of hash functions to achieve secure and efficient digital signatures. These schemes are particularly interesting in the context of post-quantum cryptography due to their resilience against quantum attacks, especially when other traditional public-key cryptosystems like RSA or ECC may become vulnerable.

Lamport-Diffie's hash-based one-time signature scheme is regarded as a promising and resilient alternative for the post-quantum era, presenting a paradigm shift in cryptographic methodologies to address the vulnerabilities posed by quantum computing advancements. This scheme stands out as a beacon of security in an environment where traditional cryptographic systems, especially those reliant on integer factorization, face the imminent threat of compromise through quantum algorithms such as Shor's algorithm. With its foundation in hash functions, Lamport-Diffie introduces a robust and quantum-resistant approach to digital signatures, offering cryptographic practitioners

and researchers an avenue to fortify information security in a landscape where quantum threats loom large. This alternative not only aligns with the imperative of future-proofing cryptographic systems but also signals a proactive response to the evolving challenges ushered in by the relentless progress of quantum technologies. As the cryptographic community navigates the complex terrain of post-quantum security, Lamport-Diffie's hash-based one-time signature scheme emerges as a beacon of resilience and innovation, charting a course towards cryptographic solutions capable of withstanding the transformative impact of quantum computing on traditional cryptographic foundations. The only disadvantage is that the signature size is n^2, where the hashed grid size is n, which is quite large. The Winternitz One-Time Signature (OTS) scheme is a cryptographic construction designed to provide secure digital signatures with a focus on minimizing the impact of quantum attacks. Proposed by Ralph Winternitz in 1986, this scheme operates on the principles of hash-based cryptography, similar to the Lamport-Diffie scheme, but with distinctive features that make it particularly suitable for post-quantum cryptographic scenarios. In the Winternitz OTS scheme, the private key consists of a sequence of values derived from a hash function applied iteratively. The corresponding public key is generated from the hash function and is typically shorter than the private key. A digital signature is created by revealing a subset of the private key values, which are then used to sign a specific message. Importantly, each private key value is used only once, making it a one-time signature scheme.

The Winternitz one-time signature scheme greatly reduces the size of the signature, since here one string key signs several bits of a hashed message [22], but in this case, a problem occurs when we need to exchange a large number of keys because it uses an individual key pair for each message. To avoid using a large number of verification keys, we can use the Merkle digital signature scheme where a binary tree is used. Where we get the public key from the root of this tree [23–29].

Key generation: The length of the tree is chosen as H>=2. 2^H documents can be signed by one public key. 2^H key pairs are created, where X_i denotes the signing key and Y_i denotes the verification key, the verification keys form the leaves of the tree. Each branch is a hash value of the concatenation of the children of the tree.

$$a[1,0] = h(a[0,0] \,||\, a[0,1]) \quad (1)$$

The public key is the root of the binary tree, and it requires the calculation of 2^H pairs of unique keys to generate it.

Signature generation: The m size random is converted into a message of size n using a hash function. $h(m)$ is hash, the one-time signature is created using a random X_{arb} the one-time key, the document signature is a one-time signature, Y_{arb} verification one-time key, index arb, and the concatenation of all branches related to "authi" concerning Y_{arb}.

$$\text{Signature} = (\text{sig}||\text{arb}||\, Y_{\text{arb}}||\text{auth}_0, \dots, \text{auth}_{H-1}) \quad (2)$$

Signature verification: When verifying a Merkle cryptosystem signature, if verification of a one-time signature sig by Y_{arb} is valid, all $a[i, j]$ nodes are computed using "authi", index arb and Y_{arb}. The signature is valid, if the public key and the root of the tree are equal.

The detailed process of receiving the public key within the Merkle tree structure is visually explained in Fig. 1. This step-by-step procedure outlines how the public key is obtained from the Merkle tree, providing a comprehensive understanding of the cryptographic mechanism.

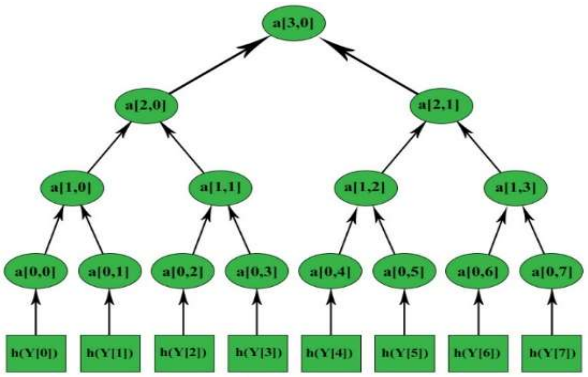

Figure 1: Merkle tree

6. Verkle and Merkle

We can tell, that Verkle trees are an improved version of upgraded Merkle trees, their construction is very similar to that of Patricia's Merkle tree [30, 31]. Werkle trees are more efficient and we can use smaller proof sizes with them.

Fig. 1 shows a Verkle tree of 9 files with a branching rate of 3. First, the files are split into subsets of size k = 3, at the next step, a commitment vector and corresponding membership proofs are calculated for each subset. After dividing the files into subsets of size k = 3, by calculating vector commitments and membership proofs for each subset, we obtain commitments VC_1, VC_2 and VC_3. As for the VC_4 commitment vector, which is computed with these last three commitments together with the p_9, p_{10} and p_{11} membership proofs for the VC_1, VC_2 and VC_3 commitments, accordingly, concerning the VC_4 commitment. The final resolution of the Verkle tree is the root commit, in this case VC_4.

The proof in a Merkle tree must contain every flattened node in the tree that shares a common ancestor with any node on the path leading to the proof node.

For this reason, the size of the signature is very large. Flattern nodes must be provided at each level since the entire set of child nodes is needed to calculate the value of a node, and this continues until the root of the tree is reached.

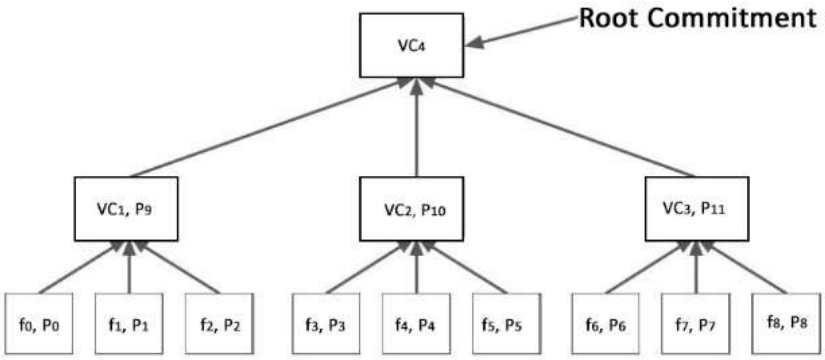

Figure 2: Verkle tree

But there is no need to provide any flattened nodes in a Verkle tree. Because here we are only pointing the way. For this reason, Verkle trees are wider than Merkle Patricia trees: a larger tree has a shorter path in both Merkle and Verkle's cases, but in a Merkle Patricia tree this effect runs down by the high cost of having to provide full-width 1 flatter node per proof branch.

Therefore, the Verkle tree is more effective. Table 1 reflects the comparison of the efficiency of Merkle and Verkle trees.

Table 1
Comparison of efficiency

Scheme	Construction	Update	Proof Size
Merkle Tree	$O(n)$	$O(\log_2 n)$	$O(\log_2 n)$
Merkle Tree (w-ary)	$O(n)$	$O(w\log_w n)$	$O(w\log_w n)$
Vector Commitment	$O(n^2)$	$O(n)$	$O(1)$
Verkle Tree	$O(wn)$	$O(w\log_w n)$	$O(\log_w n)$

7. Novel design

Verkle's core commitment is a public key (Fig. 2).

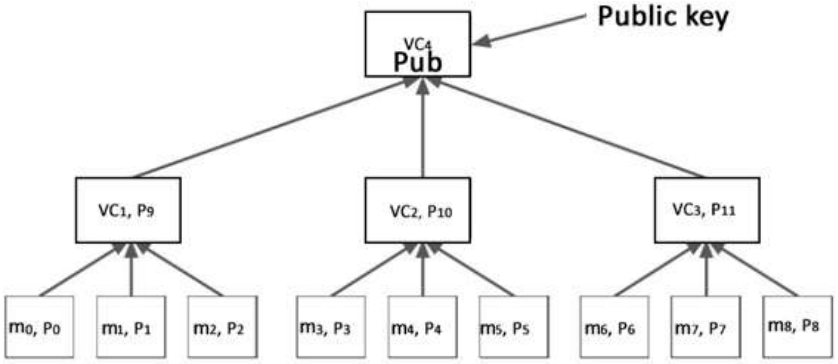

Figure 3: Verkle Signature Scheme

Key generation: H>=2 is chosen as Verkle tree length. Where, 2^H number of documents can be signed by one public key. A 2^H of key pairs is generated, assigned to the signature key Xi and the verification key Y_i, after computing $h(Y_i)$ we use it as the leaves of the tree. Each node is the hash value of the connection of its branches.

$$a[1,0] = h(a[0,0] \,||\, a[0,1]) \quad (3)$$

To generate a public key, it is necessary to calculate 2^H pairs of one-time keys.

Signature generation: A message of random size is converted to size n using a hash function. Assign to message *m*. $h(m)$ = hash and the one-time signature is created with a random X_{arb} one-time key, To sign the document we need to concatenate: one-time signature, Y_{arb} verification one-time key, proof of index arb, and root commitment.

$$\text{Signature} = (\text{sig}||\text{arb}||\, Y_{arb}||\text{proof, root commit}) \quad (4)$$

Signature verification: Digital signature verification in Verkle is done as follows, one-time signature sig must be verified with Y_{arb}, if found correct, all VC_i checks are calculated using "authi," index arb and Y_{arb}. The signature is verified If the tree root matches the commitment one.

The vector commitment must be chosen specifically, it must be resistant to the attacks of quantum computers. In the case of vector commitment is quantum resistant the final digital signature is also quantum resistant.

8. Conclusions

The Verkle scheme indeed represents a significant advancement over the traditional Merkle scheme, introducing key enhancements that contribute to both efficiency and scalability in cryptographic applications. By allowing the use of smaller keys and optimizing the verification process, the Verkle scheme showcases its potential as a powerful upgrade.

One notable feature is the reduction in the amount of verification required, approximately by a factor of 6-8, compared to the Merkle scheme. This optimization is achieved by streamlining the verification process through a single proof that validates all parent-child connections, from leaf nodes to the root. This innovation not only enhances efficiency but also paves the way for increased scalability, addressing crucial concerns in contemporary cryptographic systems.

While the Verkle scheme introduces a more complex cryptographic framework, the benefits it brings in terms of reduced verification overhead and improved scalability make it a compelling choice for applications where these factors are paramount. The utilization of SNARKs (Succinct Non-Interactive Arguments of Knowledge) for Verkle proof verification further aligns with the trend of leveraging advanced cryptographic tools for enhanced efficiency.

Looking ahead, the inevitability of quantum computing advancements necessitates a strategic shift towards STARKs (Scalable Transparent Arguments of Knowledge) proofs with hashes. This shift is driven by the recognition that the linear homomorphisms on which Verkle trees depend may no longer be secure in the face of quantum computing capabilities. While this transition introduces challenges, it also opens avenues for exploring alternative cryptographic primitives that align with post-quantum assumptions.

Moreover, the mention of SNARK-based Verkle proof verification and the potential for reverting to SNARK Merkle proofs when improved underscore the adaptability of cryptographic schemes in response to evolving technologies. This adaptability becomes crucial in ensuring the long-term security and efficiency of cryptographic systems.

In summary, the Verkle scheme represents a significant leap forward in cryptographic design, offering tangible benefits in terms of reduced verification complexity and enhanced scalability. As quantum computing looms on the horizon, the transition to STARKed proofs reflects a proactive stance in preparing cryptographic systems for the challenges posed by evolving technologies. This adaptability, coupled with ongoing advancements in cryptographic tools, positions the field for continued innovation and the development of schemes grounded in post-quantum assumptions, ensuring the robustness of cryptographic protocols in the face of emerging threats.

Acknowledgment

This work was supported by Shota Rustaveli National Science Foundation of Georgia (SRNSF) [STEM–22-1076].

References

[1] T. Ladd, et al., Quantum Computers, Nature 464 (2010) 45–53. doi: 10.1038/nature08812.

[2] D. Divincenzo, Topics in Quantum Computers, Mesoscopic Electron Transport, NATO ASI Series 345 (1997). doi: 10.1007/978-94-015-8839-3_18.

[3] B. Gardas, et al., Defects in Quantum Computers, Sci. Rep. 8 (2018) 4539. doi: 10.1038/s41598-018-22763-2.

[4] A. Lele, Quantum Computers. In: Quantum Technologies and Military Strategy, Advanced Sciences and Technologies for Security Applications (2021). doi: 10.1007/978-3-030-72721-5_3.

[5] J. Bardin, Beyond-Classical Computing Using Superconducting Quantum Processors, IEEE International Solid- State Circuits Conference (ISSCC) (2022) 422–424. doi: 10.1109/ISSCC42614.2022.9731635.

[6] A. Ilyenko, et al., Practical Aspects of Using Fully Homomorphic Encryption Systems to Protect Cloud Computing, in: Cybersecurity Providing in Information and Telecommunication Systems II, vol. 3550 (2023) 226-233.

[7] R. Chernenko, et al., Encryption Method for Systems with Limited Computing Resources, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288 (2022) 142-148.

[8] V. Sokolov, P. Skladannyi, H. Hulak, Stability Verification of Self-Organized Wireless Networks with Block Encryption, in: 5th International Workshop on Computer Modeling and Intelligent Systems, vol. 3137 (2022) 227–237.

[9] A. Bessalov, et al., Modeling CSIKE Algorithm on Non-Cyclic Edwards Curves, in: Cybersecurity Providing in Information and Telecommunication Systems, vol. 3288 (2022) 1–10.

[10] J. Katz, Digital Signatures, Volume 1, Springer (2010).

[11] Aki, Digital Signatures: A Tutorial Survey, Computer 16(2) (1983) 15–24.

[12] D. Pointcheval, J. Stern, Security Arguments for Digital Signatures and Blind Signatures, J. Cryptol. 13 (2000) 361–396.

[13] R. Haraty, A. N. El-Kassar, B. Shibaro, A Comparative Study of RSA Based Digital Signature Algorithms, J. Math. Statistics 2(1) (2006) 354–359.

[14] S. Jaju, S. Chowhan, A Modified RSA Algorithm to Enhance Security for Digital Signature, International Conference and Workshop on Computing and Communication (IEMCON), IEEE (2015).

[15] C. Fu, Z. Zhi-liang, An Efficient Implementation of RSA Digital Signature Algorithm, 4th International

Conference on Wireless Communications, Networking and Mobile Computing, IEEE (2008).
[16] U. Somani, K. Lakhani, M. Mundra, Implementing Digital Signature with RSA Encryption Algorithm to Enhance the Data Security of Cloud in Cloud Computing, First International Conference On Parallel, Distributed and Grid Computing (PDGC 2010), IEEE (2010).
[17] T. Jager, S. A. Kakvi, A. May, On the Security of the PKCS# 1 v1. 5 Signature Scheme, ACM SIGSAC Conference on Computer and Communications Security (2018).
[18] T. Jager, J. Schwenk, J. Somorovsky, On the Security of TLS 1.3 and QUIC Against Weaknesses in PKCS# 1 v1. 5 Encryption, 22nd ACM SIGSAC Conference on Computer and Communications Security (2015).
[19] M. Amara, A. Siad, Elliptic Curve Cryptography and its Applications, in: International Workshop on Systems, Signal Processing and Their Applications, WOSSPA, IEEE (2011).
[20] S. Ullah, et al., Elliptic Curve Cryptography; Applications, Challenges, Recent Advances, and Future Trends: A Comprehensive Survey, Comput. Sci. Rev. 47 (2023).
[21] G. Shankar, et al., Improved Multisignature Scheme for Authenticity of Digital Document in Digital Forensics Using Edward-Curve Digital Signature Algorithm, Secur. Commun. Netw. (2023).
[22] C. Dods, N. Smart, M. Stam, Hash Based Digital Signature Schemes, Cryptography and Coding. Cryptography and Coding 2005, LNCS 3796 (2005). doi: 10.1007/11586821_8.
[23] J. Buchmann, E. Dahmen, M. Szydlo, Hash-based Digital Signature Schemes, Post-Quantum Cryptography (2009). doi: 10.1007/978-3-540-88702-7_3.
[24] S. Rohde, et al., Fast Hash-Based Signatures on Constrained Devices, Smart Card Research and Advanced Applications, CARDIS 2008, LNCS 5189 (2008) doi: 10.1007/978-3-540-85893-5_8.
[25] M. Schneider, S.-F. Chang, A Robust Content Based Digital Signature for Image Authentication, 3rd IEEE International Conference on Image Processing 3 (1996) 227–230. doi: 10.1109/ICIP.1996.560425.
[26] M. Iavich, et al., Post-quantum Digital Signature Scheme for Personal Data Security in Communication Network Systems, International Conference of Artificial Intelligence Medical Engineering Education (2020) 303–314.
[27] M. Iavich, A. Gagnidze, G. Iashvili, Hash Based Digital Signature Scheme with Integrated TRNG, in: International Conference on Information Technologies, vol. 2145 (2018) 79–82.
[28] A. Gagnidze, M. Iavich, G. Iashvili, Some Aspects of Post-Quantum Cryptosystems, Eurasian J. Business Manag. Eurasian Publications 5(1) (2017) 16–20.
[29] H. Chen, D. Liang, Adaptive Spatio-Temporal Query Strategies in Blockchain, ISPRS Int. J. Geo-Inf. 11 (2022). doi: 10.3390/ijgi11070409.
[30] W. Wang, A. Ulichney, C. Papamanthou, BalanceProofs: Maintainable Vector Commitments with Fast Aggregation, Cryptology ePrint Archive (2022).
[31] L. de Castro, C. Peikert, Functional Commitments for All Functions, with Transparent Setup and from SIS, Annual International Conference on the Theory and Applications of Cryptographic Techniques (2023).