Cryptanalysis of a Post-Quantum Signature Scheme Based on Number-Theoretic Assumptions

Agha Aghayev[1] and Nour-eddine Rahmani[2]

[1] Azerbaijan Technical University, Azerbaijan agayevagha@gmail.com
[2] ACSA Laboratory, Faculty of Sciences, Mohammed I University, Oujda, Morocco; nour-eddine.rahmani@ump.ac.ma

Abstract. The asymmetric cryptographic constructions upon on number-theoretic hardness assumptions have become insecure, due to Shor's quantum algorithm and they will be vulnerable to large scale quantum computers. Hence, the adaption to quantum-resistant cryptosystems is a major task. Digital signatures, being a fundamental primitive in numerous applications. Recently, a new approach by Nguyen et al. [9] has claimed post-quantum security by basing the signature algorithm's security on a variant of the discrete logarithm problem. In this paper, we present a cryptanalysis of this construction and demonstrate a practical forgery attack that allows generating an unlimited number of valid signatures—without access to a signing oracle.

Keywords: Cryptanalysis, Digital signatures, Post-quantum cryptography, forgery

1 Introduction

In the internet era, there are millions of transactions happening around the world in open networks, and cryptography provides efficient protocols for ensuring trust in electronic communications. Digital signatures allow parties to send a mathematical proof of demonstrating the authenticity of a digital message. They are widely used in online banking systems, software distributions, and a modern trend of blockchain systems. Hence, we naturally look for efficient and compact signature protocols. In a signature scheme, a party have two keys, the secret key sk is used to sign the message by the signer, and another party uses signer's public key pk to verify the signature. Anyone who knows the public key pk can verify the authenticity of the signature and ensure that the corresponding message has not been modified in transit.

The security of any cryptographic scheme, including digital signatures rely on some mathematical hard problems, which are believed to unsolvable in polynomial time by classical computers, but easy to verify the correctness when given a solution. The first digital signature was introduced by Rivest et al. [10], relying on hardness of integer factorization problem. There are other approaches relying on discrete-logarithm problem, hard problems defined on elliptic curves, etc. which are also known as number-theoretic hardness assumptions. The advantages

of such schemes include small signature and key sizes, and they resist against attacks by classical computers. In 1980, Peter Shor made a major breakthrough in the quantum computing field by introducing an efficient quantum algorithm [11] for the integer factorization and sicrete logarithms problem. This improvement in quantum attacks raised concerns about the security of all cryptosystems whose security relies on such number-theoretic assumptions, which still are used today in industry. Even though such powerful large-scale quantum computers do not exist, the "harvest-now, attack-later" strategy could allow adversaries to collect encrypted information today and decrypt it in the future once quantum capabilities become available.

Post-quantum cryptography has been emerging as an important field, investigating efficient cryptographic algorithms which can resist against attacks of both classical and quantum algorithms, and can run on today's classical computers. Such schemes include, but are not limited to, lattice-based [1], code-based [2], hash-based cryptosystems [6], which often come with bigger signature and key sizes. the U.S. National Institute of Standards and Technology made a call for standardization of post-quantum cryptosystems, and current digital signature schemes include Crystal-Dilithium [4] based on $\mathsf{Module}-\mathsf{SIS}$, Falcon scheme [7] based on NTRU lattices, Hawk signature scheme [5] based on $\mathsf{Module}-\mathsf{LIP}$ problem, etc. often rely on lattice-based assumptions and are highly optimized for limited hardware. Moreover, researchers put great effort on improving the efficiency and memory requirements of digital signatures. While cryptosystems based on number-theorethical assumptions are useful for practice for latter reasons, they lack post-quantum security, hence it is important to consider the attack of both classical and quantum algorithms. A recent work by [9] have developed a signature scheme with post-quantum security claim, while basing its security on a number-theoretic hard problem which is not widely studied. In our work, we cryptanalyze the security of the proposed scheme in classical settings, and show an efficient attack on forging unlimited number of valid signatures using only the public key information. The flaw of the signature scheme is in the generation and verification of the signature, whose allow message signing with an invalid forged secret key that lies outside of the range $[0; p]$. We forge a secret key by first constructing a system of modular equations, and solving it with the Chinese remainder to find an secret key. Hence, we first define the underlying hardness assumption, briefly show the scheme, and demonstrate how to efficiently forge a fake secret key which can create correct signatures. We also provide an implementation of the attack using SageMath platform.

2 Preliminaries

We denote the set of integers as $\mathbb{Z}$. Let $p \in \mathbb{Z}$ be a prime number, we denote the ring as $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$. The multiplicative finite field $\mathbb{F}$ is the simplest field $\mathbb{F}_p = \{1, 2, 3, .., p-1\}$ over modulo p. Given a number n, $\phi(n)$ denotes the number of relatively primes elements smaller than n, for prime numbers it holds

that

$$\phi(p) = p - 1$$

.

For any $a \in \mathbb{F}_p$ and any $x \in \mathbb{Z}$, we have

$$a^x \;=\; a^{\,x \bmod (p-1)}$$

The Chinese remainder theorem is a powerful tool for many cryptographic applications and solving system of modular equations.

Theorem 1 (Chinese Remainder Theoremm, [3]). *Let $m_1, m_2, \ldots, m_k \in \mathbb{Z}_{>0}$ be pairwise coprime positive integers, and let $M = m_1 m_2 \cdots m_k$ represent their multiplication. For any integers $a_1, a_2, \ldots, a_k$, the system of congruences*

$$\{x \equiv a_i \pmod{m_i}\}_{i=1}^{k}$$

has a unique solution modulo M and there exists an algorithm that finds x in time polynomial in the input size.

3 The signature scheme

In this section, we recall the hardness assumptions used in [9] and show how they underpin the security of the signature scheme. As mentioned earlier, the security of the construction depends on a relatively new number-theoretic problems that are defined on finite fields, and they claim that current attacks on discrete-log problem do not apply to this self-exponent problem, making the cryptosystem resist against known type of attacks.

3.1 Hardness assumptions

The security of the digital signature algorithm relies on two hardness assumptions: the first hard problem is some form of discrete-logarithm problem, defined as follows:

Theorem 2. *Let p be a prime number. Given an element from the finite field $y \in \mathbb{F}_p^{\times}$, find an element $x \in \mathbb{F}_p^{\times}$ such that*

$$y \equiv x^{\,x} \pmod{p}.$$

The authors argue that this problem cannot be efficiently solved by either classical or quantum algorithms , and they use it as the basis for the key generation step of their scheme. The second problem, which plays the role in the verification process, is defined as follows:

Theorem 3. *Let p be a prime number and let $a, b \in \mathbb{F}_p^{\times}$ be positive integers. Find all $x \in \mathbb{F}_p$ such that*

$$a^x \equiv x^b \pmod{p}.$$

Both assumptions introduce computational search problems where the variable appears in the base and in the exponent, creating a self-referential structure that may resist against contemporary standard reduction techniques. These assumptions provide the hardness foundation of the scheme: the first ensures the secrecy of private keys, while the second guarantees that only valid signatures can be produced.

3.2 The proposed signature scheme

A signature scheme consist of 3 algorithms: *key generation*, *singing*, and *verification algorithms*. The key generation algorithm generates a secret and public key pairs for a user, and defined in [9] as follows:

Algorithm 1 $\mathsf{Gen}(1^\lambda)$ - Key Generation Algorithm

Require: λ
Ensure: sk, pk
1: Generate a prime p of λ bits.
2: Generate x such that $1 < x < p$ and $\gcd(x, p-1) = 1$.
3: Compute $y \leftarrow x^{-(x^{-1})} \bmod p$. If $y = 1$, then go to step 2.
4: **return** $sk = x, pk = (y, p)$

Given only the public parameters y and p, computing a valid x within the prescribed range should be computationally infeasible, thereby reducing the inversion problem to the first hardness assumption. Nevertheless, we will show that the invalidity of the signing algorithm does not prevent from forging valid signatures.

The proposed scheme is essentially a *hash-and-sign* signature scheme. The message M is first mapped by a Hash function to an integer smaller than p, that is,

$$H : \{0,1\}^* \longrightarrow \mathbb{Z}_p,$$

The algorithm computes two values r and s in a way that after some additional computation they yield the same result by canceling some powers of x, using the following algorithm:

Algorithm 2 $\mathsf{Sign}_{sk}(M)$ - Signature Algorithm

Require: $p, sk = x, M$
Ensure: $\sigma = (r, s)$
1: Generate k such that $1 < k < p-1$ and $\gcd(k, p-1) = 1$
2: Compute $h \leftarrow H(M)$
3: Compute $n \leftarrow p \cdot (p-1)$
4: Compute $r \leftarrow x^{xk} \cdot (xh + k)^{xk} \cdot (k^h x^k) \bmod p$
5: Compute $s \leftarrow r \cdot x^k \cdot k^h \bmod n$
6: **return** (r, s)

The signature algorithm is a probabilistic algorithm, since we choose a random k for signing, it yields a different signature each time. Signature verification algorithm checks the validness of the signature by computing a and b using following algorithm, and then checks equality.

Algorithm 3 $\mathsf{Vrfy}_{pk}(\sigma)$ - Verification Algorithm

Require: Prime p, public key y, message M, signature (r, s)
Ensure: 1 (valid) or 0 (invalid)
1: $h \leftarrow H(M)$
2: $a \leftarrow y^{r \cdot h} \times r^{s+r} \pmod{p}$
3: $b \leftarrow s^{r} \pmod{p}$
4: **return** **1** if $a = b$, else **0**

3.3 Security of digital signatures

In general, for the correctness of the signature schemes, it is required that $\mathsf{Vrfy}(M, \mathsf{Sign}_{sk}(M)) = 1$ for every legal message M. We define the formal security as [8]. Consider the following experiment: there is an adversary $\mathcal{A}$ and it is given the public key pk, and has access to an signing oracle $\mathsf{Sign}_{sk}(\cdot)$. The set Q contains all queries made by the adversary $\mathcal{A}$. After some queries, the adversary then outputs (M, σ). The adversary $\mathcal{A}$ is said to succeed if it outputs a forgery on a new message

$$\mathsf{Vrfy}_{pk}(M, \sigma) = 1 \quad \text{and} \quad M \notin Q.$$

We say that output of the experiment is defined to be 1. Let the proposed signature scheme be $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$. We say that it is secure under an adaptive chosen-message attack, if for all probabilistic polynomial-time adversaries $\mathcal{A}$, the probability of previous experiment is negligible (very small)

$$\Pr\left[\mathsf{Sig\text{-}forge}_{\mathcal{A},\Pi}(n) = 1\right] \leq \mathsf{negl}(n).$$

In other words, it should be infeasible to forge a digital signature even after seeing polynomially-many valid signatures signed with the secret key sk by the challenger. However, we have examined that it is possible to forge a signature without using the oracle. This property is caused by the verification algorithm, as it essentially relies on canceling $x^{-x^{-1}rh}$ from a as shown in the original work [9] for the correctness of the algorithm, and yielding the same result with s^r. However, this equality also holds for values of x outside of the range, which underpins the main idea of our attack.

4 Forgery attack

The generation of the public and secret keys allows for generating an invalid secret key out of the range $[0; p]$, while resulting in valid signatures. We generate

a fake secret key by using the Euler's formula for prime numbers and Chinese remainder theorem to solve a system of congruences . As defined in Algorithm 1, the construction generates public and secret keys by first selecting a secret key x such that $1 < x < p$ and $\gcd(x, p-1) = 1$, and then compute

$$y \leftarrow x^{-(x^{-1})} \bmod p.$$

Based on Euler's theorem, we have the following equation

$$x^{-(x^{-1})} = x^{-(x^{-1}) \bmod \phi(p)} = x^{-(x^{-1}) \bmod (p-1)} \bmod p.$$

If we assume that the exponent of the above formula equals 1 modulo $(p-1)$, then we get x to be inverse of y. Hence, we use the last formula to generate a fake secret key, by mapping it into a system of congruences:

$$\begin{cases} x_f \equiv 1 \pmod{p-1} \\ x_f \equiv y^{-1} \pmod{p} \end{cases}$$

where x_f denotes a forged solution and will be used to sign messages. We use the Chinese remainder theorem to solve this system of congruences, which yields the result

$$\begin{aligned} x_f \equiv p \cdot \left(p^{-1} \bmod (p-1)\right) \\ +(p-1) \cdot \left((p-1)^{-1} \bmod p\right) \cdot y^{-1} \pmod{n} \end{aligned}$$

We now show that the forged solution also satisfies the verification algorithm, while also simplifying the correctness of the algorithm introduced in the original paper.

4.1 Correctness of the attack

We compute the a and b as in the verification algorithm, and show that they result in same expression:

$$\begin{aligned} a &= y^{rh} \times r^{s+r} \\ &= x_f^{-x_f^{-1} rh} \times \left((x_f^{x_f^{-1} h} \times x_f^k \times k^h)^{x_f^{-k} k^{-h}} \right)^{r x_f^k k^h} \times r^r \\ &= x_f^{-x_f^{-1} rh} \times x_f^{x_f^{-1} rh} \times x_f^{kr} \times k^{hr} \times r^r \\ &= x_f^{kr} \times k^{hr} \times r^r \pmod{p} \end{aligned}$$

$$b = s^r = (r \cdot x_f^k \cdot k^h)^r = x_f^{kr} \times k^{hr} \times r^r \pmod{p}$$

Since $a \equiv b \pmod{p}$ for the forged secret key x_f, we conclude that our forgery algorithm succeeds. The verification holds because, upon substituting y with the forged secret key, the third line simplifies to

$$x_f^{-x_f^{-1}rh} \times x_f^{x_f^{-1}rh} \equiv 1 \pmod{p}.$$

This follows from the fact that the two exponential terms cancel each other out, akin to the cancellation observed in the original algorithm. Since we can forge a signature for any message, our attack succeeds in $\mathsf{Sig\text{-}forge}_{\mathcal{A},\Pi}(n)$ with overwhelming probability.

The authors also propose a second signature scheme, which differs from the first scheme only in the key generation algorithm. Instead of randomly picking a secret key x, they first generate two random prime numbers p and q such that $q|(p-1)$. Then, the private key is set to

$$x \leftarrow \alpha^{(p-1)/q} \pmod{p}$$

where $\alpha \in \mathbb{Z}_p^*$ is randomly selected. Since the public key generation, signing and verification algorithms are the same, we conclude that the same attack procedure will work, as selection of secret key does not affect the forgery algorithm.

4.2 Experiment

To demonstrate the attack in practice, we also provide experimental attacks on the signature algorithm. We have also designed an experimental evaluation using Sage math, implemented the key generation, signing, and verification algorithms, and finally the forging algorithm. We generated key pairs for both the original scheme and the second variant with $x = \alpha^{(p-1)/q} \bmod p$, signed a set of test messages, and applied the forgery algorithm. Our experiments show that the attack successfully produced valid signatures, confirming that the secret key generation method does not affect the attack. The attack succeeds in 0.000109s for 1024 bit p. Timing and reproducibility details are provided in the repository, the implementation can be found at our github repository `github.com/aghayevagha/signatureattack`.

5 Conclusion

In this work, we have provided a cryptanalysis of a new type of signature scheme with claim of quantum-resistance. We showed a way of unlimited forgery by computing a forged secret key by using only public key information, without using the signing oracle. Moreover, we could not find a way to fix the flaw of the algorithm, and it is for the future work. Moreover, the hardness assumption remain unclear if it can be solved with quantum algorithm, which is left as another open problem.

References

1. Asif, R.: Post-quantum cryptosystems for internet-of-things: A survey on lattice-based algorithms. IoT **2**(1), 71–91 (2021)
2. Baldi, M., Santini, P., Cancellieri, G.: Post-quantum cryptography based on codes: State of the art and open challenges. In: 2017 AEIT International Annual Conference. pp. 1–6. IEEE (2017)
3. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to algorithms. MIT press (2022)
4. Ducas, L., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G., Stehlé, D.: Crystals–dilithium: Digital signatures from module lattices (2018)
5. Ducas, L., Postlethwaite, E.W., Pulles, L.N., Woerden, W.v.: Hawk: Module lip makes lattice signatures fast, compact and simple. In: International Conference on the Theory and Application of Cryptology and Information Security. pp. 65–94. Springer (2022)
6. Fathalla, E., Azab, M.: Beyond classical cryptography: A systematic review of post-quantum hash-based signature schemes, security, and optimizations. IEEE Access (2024)
7. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z., et al.: Falcon: Fast-fourier lattice-based compact signatures over ntru. Submission to the NIST's post-quantum cryptography standardization process **36**(5), 1–75 (2018)
8. Katz, J., Lindell, Y.: Introduction to modern cryptography: principles and protocols. Chapman and hall/CRC (2007)
9. Nguyen, K.T., Tong, M.D., Hoang, T.H., Luu, H.D.: A new type of post-quantum digital signature scheme. Journal of Science and Technique-Section on Information and Communication Technology **14**(01) (2025)
10. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM **21**(2), 120–126 (1978)
11. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM review **41**(2), 303–332 (1999)