International Journal of Software and Informatics, ISSN 1673-7288
http://www.ijsi.org, ijsi@iscas.ac.cn, +86-10-62661048
IJSI, 2025, 15(4): 419–433, doi: 10.21655/ijsi.1673-7288.00360

International Journal of Software and Informatics

Research Article

SM3-OTS: Compact Post Quantum One-time Signature Scheme Based on Chinese Cryptographic Algorithm SM3

Yatao Yang (杨亚涛)[1,2], Fangrui Yin (殷方锐)[1], Liangyu Chen (陈亮宇)[1], Deng Pan (潘登)[1]

[1] (Department of Electronic and Communication Engineering, Beijing Electronic Science & Technology Institute, Beijing 100070, China)
[2] (School of Telecommunications Engineering, Xidian University, Xi'an 710071, China)
Corresponding author: Yatao Yang, yy2008@163.com

Abstract SPHINCS$^+$ is a stateless digital signature scheme designed using hash functions and has been proven resistant to quantum computing attacks. However, its wide practical application is constrained by the large size of the generated signature values. To address the issue of the lengthy signature value generated by the WOTS+ one-time signature scheme within SPHINCS$^+$, a compact one-time signature scheme, SM3-OTS, based on Chinese cryptographic algorithm SM3, is proposed in this paper. The proposed scheme utilizes the binary and hexadecimal information of the message digest as the indices for node positions in the first 32 hash chains and the last 16 hash chains, respectively. This approach effectively reduces the key length and the signature value length compared to traditional one-time signature schemes based on hash functions. Compared to WOTS+ in SPHINCS$^+$, Balanced WOTS+ in SPHINCS-α, and WOTS+C in SPHINCS$^+$C, the proposed SM3-OTS shortens the signature value length by about 29%, 27%, and 26%, respectively, with a significant improvement in signing performance. In addition, by adopting the SM3 algorithm, SM3-OTS exhibits strong resistance to quantum attacks while maintaining well-balanced overall performance.

Keywords hash function; SPHINCS$^+$; digital signature; one-time signature (OTS); post-quantum cryptography (PQC)

Citation Yang YT, Yin FR, Chen LY, Pan D. SM3-OTS: Compact post quantum one-time signature scheme based on Chinese cryptographic algorithm SM3, *International Journal of Software and Informatics*, 2025, 15(4): 419–433. http://www.ijsi.org/1673-7288/360.htm

1 Introduction

With the rapid development of quantum computing technology[1, 2], traditional cryptographic schemes based on elliptic curves, discrete logarithms, large integer decomposition, and other difficult mathematical problems will face quantum computing attacks[3]. In 1994,

This is the English version of the Chinese article "SM3-OTS: 基于国密算法 SM3 的紧凑型后量子一次签名方案. 软件学报, 2025, 36(10): 4417–4429. doi: 10.13328/j.cnki.jos.007392"

Funding items: Beijing Natural Science Foundation (4232034); Special Fund for Basic Research Business Expenses of Central Universities (3282023017, 3282024058, 3282024052)
Received 2024-07-01; Revised 2024-09-05; Accepted 2024-12-30; IJSI published online 2025-12-30

the Shor algorithm proposed by Shor[4] is a quantum algorithm that can effectively solve the problem of large integer decomposition. In 1996, Grover[5] proposed the quantum search Grover algorithm, which reduces the time complexity of cracking the key of the symmetric cryptographic algorithm from $\mathrm{O}(2^n)$ to $\mathrm{O}(2^{n/2})$ by violent cracking. By adopting the Shor algorithm and Grover algorithm, quantum computers pose serious threats to traditional public key cryptographic algorithms and symmetric cryptographic algorithms. It is urgent to conduct research and optimization on post-quantum cryptography (PQC)[6] with the ability to resist quantum attacks to deal with these threats. In December 2016, the National Institute of Standards and Technology (NIST) formally collected post-quantum cryptography with the ability to resist quantum computer attacks from all over the world to gradually replace the traditional public key cryptography algorithms in the quantum age.

In July 2022, NIST initially selected four algorithms as the standardized algorithms from the schemes submitted in the third round of quantum cryptography algorithms, including a public key encryption algorithm and three digital signature algorithms. The algorithms include the public key encryption algorithm CRYSTALS-Kyber[7], digital signature algorithms CRYSTALS-Dilithium[8], FALCON[9], and SPHINCS$^+$[10]. Specifically, Kyber, Dilithium, and FALCON are constructed based on lattice theory. In 2022, Yang *et al*.[11] designed a VPN software system resisting quantum computing attacks based on the CRYSTALS-Kyber cryptographic scheme, which showed the engineering application of post-quantum cryptographic schemes in network security. Different from the above three cryptographic schemes, SPHINCS$^+$ is the only scheme developed based on hash functions, and NIST will collect and standardize other cryptographic algorithms with better security and performance.

At present, post-quantum cryptographic schemes can be roughly divided into cryptographic schemes[12] based on coding, lattice, multiple variables, and hash functions[13]. The digital signature scheme designed based on hash functions has the shortest signature generation and verification time and more reliable security. Therefore, the digital signature scheme based on hash functions is considered as a post-quantum cryptographic scheme with great research significance in the quantum age. SPHINCS$^+$[10] is a stateless digital signature scheme designed based on hash functions, and it is built by employing the one-time signature (OTS), few-time signature (FTS), and Merkle tree[14]. OTS is the core of the SPHINCS$^+$ scheme. Generally, OTS based on hash functions has the problem of lengthy generated signature values, which makes it difficult for stateless digital signature schemes based on hash functions such as SPHINCS$^+$ to be applied in practice.

The contributions of this paper are as follows:

(1) SM3-OTS, a new post-quantum OTS scheme based on the Chinese cryptographic algorithm SM3, is designed. This scheme employs the binary and hexadecimal information of the message digest as the indices for node positions in the first 32 hash chains and the last 16 hash chains, respectively, thus effectively shortening the key length and generated signature value length compared to the traditional OTS schemes based on hash functions.

(2) The performance of SM3-OTS, a new OTS scheme based on the Chinese cryptographic algorithm SM3, is tested. By conducting experimental tests, compared with WOTS+ in SPHINCS$^+$[10], Balanced WOTS+ in SPHINCS-α[15], and WOTS+C in SPHINCS$^+$C[16], SM3-OTS reduces the generated signature value length by 29%, 27%, and 26%, respectively, with significantly improved signing performance.

2 Research Status

Adopting the preimage attack characteristic (one-way characteristic) for the construction of digital signature schemes is a research idea that many cryptographers pay attention to and recognize. In 2020 and 2021 respectively, Suhail *et al*.[17] and Kumar *et al*.[18] analyzed the importance of digital signature schemes based on hash functions in the security of the Internet of Things (IoT) devices after the maturity of quantum computing technology, various factors to be considered in the deployment, and various challenges, and provided corresponding suggestions and prospects for post-quantum cryptography technology applied to the network environment of IoT devices.

In 1979, Lamport[19] put forward the first OTS scheme, Lamport-OTS (LOTS), which signs the message bit by bit. This scheme has two defects: firstly, each pair of keys in the scheme can only be employed once; otherwise the adversary can forge the signature by adopting the signature value of the previous message. Thus, each pair of keys in the LOTS scheme can only be employed once to avoid forgery attacks. Secondly, the length of the public and private keys utilized in the scheme and the length of the generated signature value are very large. If the security parameter selected during signing is n, the lengths of the private key and the public key are both $2 \times n \times n$, and the length of the final generated signature value of the scheme is $n \times n$. The excessively large lengths of the public and private keys and signature value of the LOTS scheme make it unsuitable for practical application. In the same year, Merkle[20, 21] proposed Winternitz-OTS (WOTS) based on the Lamport-OTS scheme. In the WOTS scheme, a hash chain structure is employed to construct the public and private keys of the signature scheme. A hash chain is a chain generated by conducting repeated hash operations on the data, with the last node in the hash chain as the hash value of the previous node. Similar to the LOTS scheme, each pair of keys in the WOTS scheme can only sign one message. Compared with the hash values of n bytes corresponding to each message digest bit in the LOTS scheme, the hash values of n bytes corresponding to 2, 4, or 16 message digest bits are employed in the WOTS scheme, thus greatly shortening the length of signature values generated by the LOTS scheme.

Since each pair of keys in the OTS scheme can only be employed once, Merkle[20, 21] proposed the Merkle signature scheme (MSS). This scheme extends the OTS scheme to the FTS scheme by adopting the structure of the Merkle tree, which is essentially a binary hash tree. In the MSS scheme, the leaf nodes of the Merkle tree are utilized to manage the public key of the OTS scheme, with the root node of the Merkle tree employed as the public key of the MSS scheme to extend the OTS scheme to an FTS scheme. If the height of a Merkle tree in the scheme is h, the Merkle tree has 2^h leaf nodes, which are used to store the hash values of the public keys of the OTS scheme. This means a Merkle tree with a height of h can manage 2^h OTS keys and sign 2^h messages. The signature value generated by the Merkle tree signature scheme mainly includes two parts: the signature value generated by the OTS scheme corresponding to the Merkle tree leaf node and the authentication path of this leaf node. These two parts of data together constitute the signature value of the Merkle tree signature scheme. The OTS scheme is the core of all hash-based signatures, and it can be extended to multiple signature schemes by employing hash trees.

The WOTS+OTS scheme and the FORS FTS scheme are employed in SPHINCS$^+$, and are extended to stateless digital signature schemes by extending the Merkle tree. In 2022, Zhang *et al*.[15] improved the OTS scheme WOTS+ used in SPHINCS$^+$ in the SPHINCS-α scheme and put forward Balanced WOTS+, thus shortening the signature value generated by the WOTS+ OTS scheme and improving the security. In the same year, Kudinov *et al*.[16] improved the OTS scheme in the SPHINCS$^+$C scheme and proposed WOTS+C. It canceled the original three check chains, and instead, before signing, the message digest value is concatenated with a random number of n bytes. Meanwhile, the hash function is adopted to conduct hash operations on the message digest value concatenated with the random number to generate a hash value, and the latter part of the hash value is the index of the node position of the hash chain.

By adopting the same security parameter $n = 32$, the comparison of the signature value lengths (bytes) generated by WOTS+, Balanced WOTS+, WOTS+C, and LMOTS is shown in Table 1.

Table 1 Comparison of signature value lengths generated by the WOTS+ scheme and several variants (bytes)

Scheme	Generated signature length
WOTS+	2,144
Balanced WOTS+	2,112
WOTS+C	2,080
LMOTS	2,144

Sun *et al*.[22] employed the Chinese cryptographic hash algorithm SM3 to replace the hash function in the SPHINCS$^+$ digital signature scheme, and provided the relevant experimental results after instantiating SPHINCS$^+$ with SM3. The results show that it is feasible for this algorithm to realize the internationally standardized hash-based digital signature schemes. Sun *et al*.[23] also adopted the domestic hash function SM3 to replace the hash function in the Leighton-Micali signature system (LMS) scheme, and provided preliminary experimental results. Additionally, Li *et al*.[24] systematically summarized the research progress of hash-based signature components, and deeply analyzed the overall design ideas of different types of hash-based signature schemes.

The optimization of the WOTS+ scheme conducted in Refs. [10, 14–16] promotes the standardization and applicability of post-quantum signature schemes based on hash functions, and also provides ideas for the optimization of hash-based signature schemes. However, as the length of signature values generated by the optimized scheme is still large, the research and optimization of hash-based digital signature schemes need to be further advanced.

3 Basic Knowledge

This section will introduce the basic knowledge and symbols of the SM3-OTS signature scheme, and related concepts of hash functions.

3.1 Symbol description

The symbol description is shown in Table 2.

3.2 Hash functions

Hash functions can map an input of any length to an output of fixed length, which is generally called the hash value or digest value[25]. This mapping is a kind of contraction mapping, as the space of the hash value is much smaller than that of an input, and different inputs may be mapped to the same output. Thus, it is impossible to determine the unique input value from the hash value. An ideal hash function should have the following properties.

(1) Determinism. The same input information always produces the same hash value by the same hash function.

(2) Irreversibility. It is computationally infeasible to calculate the input information from the hash value.

(3) No collision. It is computationally infeasible to find two inputs with the same hash value.

3.3 Hash function security

The security of hash functions is the most basic security requirement of hash-based digital signature schemes. An ideal hash function has the following three properties[25]:

Table 2 Symbol description

Symbol	Description
M	Plaintext message to be signed
H	Chinese cryptographic algorithm SM3 in the signature scheme
m	Digest value of the message to be signed, and $m = H(M)$
n	Security parameters employed by the signature scheme
$PRNF$	Pseudo-random number generation function for generating random numbers
$base\_8$	Converting the hash value into a decimal number in groups of eight bits
$Seed$	Secret seed, a seed for generating random numbers used in $PRNF$
m_bin	Binary information of message digests
m_hex	Hexadecimal information of message digests
KeyGen()	Key generation algorithm
Sign()	Signature generation algorithm
Verify()	Signature verification algorithm
sk	Private key of the signature scheme for generating signature values for the messages
pk	Public key of the signature scheme for validity verification of the message signature values
pk'	Public key verification. If it is equal to pk, the signature value is valid.
σ	Signature value generated by the signature scheme
ADV	Attacker aiming to forge digital signatures
sk_size	Private key size (bytes)
pk_size	Public key size (bytes)
σ_size	Signature value size (bytes)

(1) Anti-first preimage attacks. Anti-first preimage attacks mean that it is difficult to find any original input x within a reasonable time for a given hash value h, thus making $H(x) = h$ difficult. Thus, for a given h, it is computationally infeasible to find any x to make $H(x) = h$[25].

(2) Anti-second preimage attacks. Anti-second image attacks mean that for a fixed input x, it is difficult to find a different input y within reasonable time to make $H(x) = H(y)$. Therefore, for a fixed x and any $y \neq x$, it is computationally infeasible to find $H(x) = H(y)$[25].

(3) Anti-collision attacks. Anti-collision attacks mean that it is difficult to find two different inputs x and y and make their hash values the same within a reasonable time, which is $H(x) = H(y)$. Thus, for all $x \neq y$, it is computationally infeasible to find $H(x) = H(y)$[25].

4 SM3-OTS Scheme

The SM3-OTS scheme adopts a hash chain structure similar to that in the WOTS+ scheme to sign messages, and the length of signature values generated by the SM3-OTS scheme is about 29% shorter than that of the WOTS+ scheme. This optimization significantly improves the signature storage and transmission efficiency, which makes the SM3-OTS scheme have greater application potential than the WOTS+ scheme in an environment with limited storage resources.

The SM3-OTS scheme mainly includes three algorithms: the key generation algorithm KeyGen(), the signature generation algorithm Sign(), and signature verification algorithm Verify(). The key generation algorithm accepts a security parameter n as the input, and the signature key pair (sk, pk) is output by the corresponding operation. The private key sk can only be owned by those who sign and is employed to sign the messages, while the public key can be obtained by anyone to verify the validity of the signature of a message. The signature generation algorithm accepts the message M to be signed and the private key sk as the two inputs, and generates the signature value σ corresponding to the message M generated by the algorithm. The signature verification algorithm accepts the message M, the signature value σ corresponding to M, and the public key pk as the inputs. The signature verification algorithm can verify the validity of the signature value. If the signature value is valid, “true” is output, and “false” is output if the signature value is invalid.

4.1 Key generation of SM3-OTS scheme

Under the security parameter of $n = 256$ bits, the secret seed $Seed$ is employed as the seed data of the pseudo-random number generator $PRNG$ to generate 48 pseudo-random numbers: $sk_0, sk_1, \ldots, sk_{47}$. The pseudo-random hexadecimal data with each pseudo-random number of 32 bytes is called a private key block, and the private key sk of the SM3-OTS signature scheme contains 48 private key blocks denoted as $sk = sk_0, sk_1, \ldots, sk_{47}$.

The public key of the SM3-OTS signature scheme is generated by the corresponding private key, and the above 48 private key blocks are hashed 255 times by adopting the Chinese cryptographic algorithm SM3, which is $pk_i = H^{255}(sk_i)$. Finally, 48 public key blocks are generated: $pk_0, pk_1, \ldots, pk_{47}$. The public key pk of the SM3-OTS signature scheme contains 48 public key blocks, which are denoted as $pk = pk_0, pk_1, \ldots, pk_{47}$. 48 hash chains are obtained from the process of generating public keys from private keys, which are $L_0, L_1, \ldots, L_{47}$, respectively, with each hash chain containing 256 chain nodes. The last node of two adjacent chain nodes in each hash chain is the hash value of the previous node, the head node of each hash chain is the private key block corresponding to the hash chain, and the tail node of each hash chain is the public key block corresponding to the hash chain. The key generation is shown in Algorithm 1.

Algorithm 1. Key generation algorithm KeyGen.

Input: Security parameter n;
Output: Private key set $sk = sk_0, sk_1, \ldots, sk_{47}$; Public key set $pk = pk_0, pk_1, \ldots, pk_{47}$.

1. $Seed = PRNG$ `// Generate random seeds via a pseudo-random number generator PRNG`
2. **for** $0 \le i \le 47$ **do** `// Generate 48 private key blocks with seeds`
3. $sk_i = PRNF(Seed, i)$
4. **end for**
5. **for** $0 \le i \le 47$ **do** `// Employ private key blocks to generate corresponding public key blocks`
6. $pk_i = H^{255}(sk_i)$ `// Hash each private key block 255 times to obtain the corresponding public key block`
7. **end for**
8. **return** $sk = sk_0, sk_1, \ldots, sk_{47}$; $pk = pk_0, pk_1, \ldots, pk_{47}$ `// Output key pairs`

As shown in Algorithm 1, under the preset security parameter $n = 256$ bits, the secret seed $Seed$ and a pseudo-random number generator $PRNG$ are employed to randomly generate 48 random numbers with n bytes as private keys, and 48 hash values with n bytes are generated as public keys by 255 hash operations of 48 private key blocks using the Chinese cryptographic algorithm SM3. The key generation is shown in Figure 1, and the hash chain structure is shown in Figure 2.

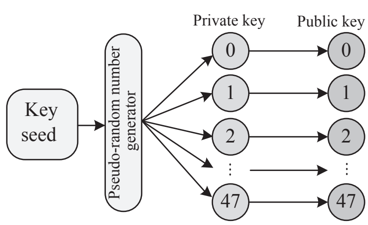


Figure 1 Key generation of SM3-OTS

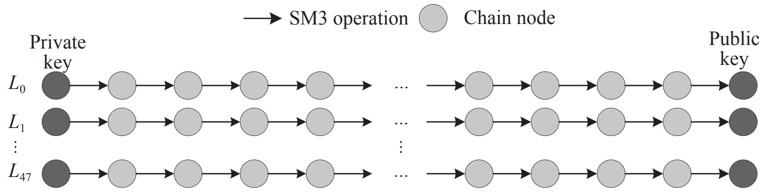
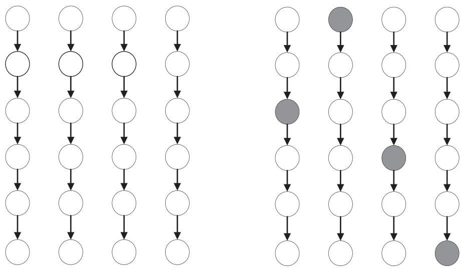


Figure 2 Hash chain structure

4.2 SM3-OTS signature generation

SM3-OTS signature generation is the process of mapping the message digest information to the hash chain nodes. As shown in Figure 3, Figure 3(a) is the hash chain of the signature scheme, and the message digest information to be signed is processed and mapped to the corresponding dark chain nodes as shown in Figure 3(b), with the mapped chain nodes recorded as signature blocks in the final generated signature value. This section will introduce the specific mapping methods in the SM3-OTS signature process in detail.



(a) Hash chain (b) Signature value

Figure 3 Hash chain mapping

Firstly, the message M to be signed is hashed ($m = H(M)$) by adopting the Chinese cryptographic algorithm SM3 to obtain the message digest value of 32 bytes. Meanwhile, the binary information of the message digest is recorded by m_bin, and the hexadecimal information of the message digest is recorded by m_hex. The binary form of the 32-byte message digest can be converted into 32 groups of decimal digits between 0 and 255, one group every eight bits. The decimal digits converted from the message digest are employed as the indexes of the node positions of the first 32 hash chains L_0–L_{31}, and the indexed chain nodes serve as the first 32 signature blocks of the signature values of SM3-OTS, denoted as $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_{47}$. For example, the message digest value of "Hello World!" calculated by the SM3 algorithm is 0AC0A9FEF0D212AA76A3C431F793853CE145659 CA1D14B114E96C1215CF26582, where the first byte is 0A and converted to 00001010 in binary form, and its corresponding decimal number is 10. The first private key block sk_0 is hashed for ten times to obtain the first signature block σ_0. The second byte in the digest is C0 and converted to 11000000 in binary form, and its corresponding decimal number is 192. The second private key block sk_1 is hashed for 192 times to obtain the second signature block σ_1. The third byte in the digest is A9 and converted to 10101001 in binary form, and its corresponding decimal number is 169. Additionally, the third private key block sk_2 is hashed for 169 times to obtain the third signature block σ_2. Similarly, the signature blocks corresponding to hash chains L_0–L_{31} are calculated as shown in Table 3.

Table 3 Process parameters and signatures (partial content omitted)

Hash chain	Information in hash values	Corresponding value	index	Signature block
0	0A	10	10	$H^{10}(sk_0)$
1	C0	192	192	$H^{192}(sk_1)$
⋮	⋮	⋮	⋮	⋮
31	82	130	130	$H^{130}(sk_{31})$
32	0	15	15	$H^{15}(sk_{32})$
33	1	362	107	$H^{107}(sk_{33})$
⋮	⋮	⋮	⋮	⋮
40	8	92	92	$H^{92}(sk_{40})$
41	9	123	123	$H^{123}(sk_{41})$
42	A	98	98	$H^{98}(sk_{42})$
43	B	46	46	$H^{46}(sk_{43})$
⋮	⋮	⋮	⋮	⋮
47	F	100	100	$H^{100}(sk_{47})$
$\sigma = H^{10}(sk_0), \ldots, H^{130}(sk_{31}), H^{15}(sk_{32}), \ldots, H^{123}(sk_{41}), H^{98}(sk_{42}), \ldots, H^{100}(sk_{47})$				

The hexadecimal form of the 32-byte message digest contains 64 data elements of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F. The position of the first element in the message digest in hexadecimal form is defined as 1, and the position of the last element is defined as 64. Meanwhile, the position of each element of 0–F is calculated. The modulo 255 operation is performed on the position information of each element, and the position information of element 0 is $1 + 4 + 10 = 15$. Then, the modulo 255 operation is conducted on the calculated position information to get 15, and the private key block sk_{32} is hashed 15 times to get the 33rd signature block σ_{32}. The position information of element 1 is calculated as 362, and the modulo 255 operation is performed on 362 to get 107. Next, the private key block sk_{33} is hashed 107 times to get the 34th signature block σ_{33}. Similarly, the data obtained after conducting the modulo 255 operation on the position information of each element is adopted as the index of the last 16 hash chains L_{32}–L_{47}. Finally, the indexed chain nodes serve as the last 16 signature blocks of the signature values of SM3-OTS, denoted as $\sigma_{32}, \sigma_{33}, \ldots, \sigma_{47}$ (see Table 3).

Therefore, the signature value generated by the SM3-OTS scheme is $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_{47}$. The signature generation algorithm is shown in Algorithm 2.

4.3 SM3-OTS signature verification

The process of SM3-OTS signature verification is the process of restoring the public key from the signature value. The public key of SM3-OTS is generated by calculating the corresponding private key, and the intermediate node of each hash chain is the process node for generating the public key block corresponding to the private key block. Thus, the public key value can be obtained by calculating the private key and also by calculating the intermediate node of the hash chain. For example, as the corresponding signature block generated by the chain 4 in Figure 4 is obtained by hashing the corresponding private key block twice, the corresponding public key block can be obtained by hashing the signature block three times.

The process of message processing by the signature verifier is similar to that of signature generation. The verifier first calculates the binary and hexadecimal forms of the message digest, 32 groups of decimal digits with every eight bits in binary form of the message digest, and the position information of elements 0–F in hexadecimal form of the message digest. Meanwhile, 255 is employed to make the difference with the above information to verify whether the signature block is valid. If the calculated value is the same as the public key value, the signature value is valid, otherwise the signature is invalid. The specific process of signature verification is shown in Algorithm 3. The signature verification information by taking “Hello World!” as an example is demonstrated in Table 4.

Algorithm 2. Signature generation algorithm Sign.

Input: Message M to be signed and private key set $sk = sk_0, sk_1, \ldots, sk_{47}$;
Output: Signature $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_{47}$.

$m = H(M)$ `// Calculate the digest value of the message` M
$m\_bin = m$ `// Convert the digest value into binary form and store it in` m_bin
$a = base\_8(m\_bin)$ `// Convert every eight bits into decimal data and store them in` a
for $0 \leq i \leq 31$ **do**
  $step[i] = a[i]$
  $\sigma_i = H^{step[i]}(sk_i)$ `// Perform` $step[i]$ `hash operations on the private key blocks to obtain corresponding signature blocks`
end for
$hex\_symbols = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{F}$
$m\_hex = m$
for $0 \leq i \leq 15$ **do**
  $sum[i] = 0$
  for $1 \leq j \leq 64$ **do**
    if $m\_hex[j] = hex\_symbols[i]$ **then**
      $sum[i] = sum[i] + j$ `// Calculate the position information of each character`
    end if
  end for
  $step[i+32] = sum[i] (\bmod\ 255)$ `// Perform the modulo 255 operation on the position information of each character`
  $\sigma_{i+32} = H^{step[i+32]}(sk_{i+32})$ `// Perform` $step[i+32]$ `operations on private key blocks to obtain corresponding signature blocks`
end for
return $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_{47}$ `// Return the signature value`

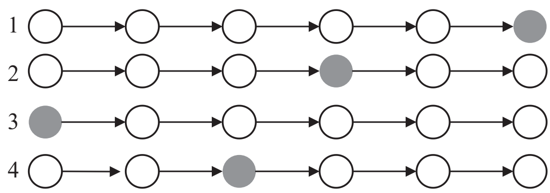


Figure 4 Signature chain

5 Security Proof

To prove that there is unforgeability under the SM3-OTS adaptive chosen-message attack (CMA), firstly it is necessary to understand the influence of the one-way and collision resistance properties of hash functions on the security of the signature scheme. The one-way property of hash functions ensures that the given hash value cannot effectively be used to deduce the original input, while collision resistance guarantees that different inputs will not produce the same hash value. In this paper, the security of SM3-OTS is reduced to the properties of these hash functions, which means that if an attacker can successfully forge a signature under CMA, then the attacker can use this forgery process to crack the one-way or collision resistance properties of the hash function. For example, if an attacker can forge a signature, the attacker may find the collision or reverse input of the hash function by reverse engineering the signature generation process. This violates the basic properties of hash functions. Since this reduction process can prove that the security of SM3-OTS really depends on the one-way and collision resistance properties of hash functions, it is crucial to choose a secure hash function to ensure the overall security of the signature scheme.

Algorithm 3. Signature verification algorithm Verify.

Input: Message M; signature value $\sigma = \sigma_0, \sigma_1, \ldots, \sigma_{47}$; public key set $pk = pk_0, pk_1, \ldots, pk_{47}$;

Output: Valid signature value (true); invalid signature value (false).

$m = H(M)$ `// Calculate the digest value of message` M

$m\_bin = m$ `// Convert the digest value into binary form and store it in` m_bin

$a = base\_8(m\_bin)$ `// Convert every eight bits in` m_bin `into decimal data and store them in` a

for $0 \leq i \leq 31$ **do**

  $step[i] = a[i]$

  $pk'_i = H^{255-step[i]}(\sigma_i)$ `// Perform` $(255 - step[i])$ `operations on the signature blocks to obtain corresponding verified public key blocks`

end for

$hex\_symbols = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \mathrm{A}, \mathrm{B}, \mathrm{C}, \mathrm{D}, \mathrm{E}, \mathrm{F}$

$m\_hex = m$

for $0 \leq i \leq 15$ **do**

  $sum[i] = 0$

  for $1 \leq j \leq 64$ **do**

    if $m\_hex[j] = hex\_symbols[i]$ **then**

      $sum[i] = sum[i] + j$ `// Calculate the position information of each character`

    end if

  end for

  $step[i + 32] = sum[i] \pmod{255}$ `// Perform the modulo 255 operation on the position information of each character`

  $pk'_{i+32} = H^{255-step[i+32]}(\sigma_{i+32})$ `// Perform` $(255 - step[i + 32])$ `operations on the signature block to obtain the verified public key block`

end for

$pk' = pk'_0, pk'_1, \ldots, pk'_{47}$

if $pk' = pk$ **then**

  return true `// If the signature value is valid, "true" is returned.`

else

  return false `// If the signature value is valid, "false" is returned.`

end if

In the challenge of the first preimage attack, the opponent ADV's challenge is to calculate the corresponding input under a given output.

$$\Pr[y = f_h(x); x' \longleftarrow ADV(y) : x = x'] \leq \varepsilon_{\text{pre-image}}$$

In the challenge of the second preimage attack, the opponent ADV knows the input-output pair (x, y), and its challenge is to find another input x' different from the input x, but whose corresponding output is y.

$$\Pr[y = f_h(x); x' \longleftarrow ADV(x, y) : x \neq x' \wedge y = f_h(x')] \leq \varepsilon_{\text{second-pre-image}}$$

The ability of the SM3-OTS cryptographic scheme to resist different types of attacks is usually called the security level provided by the protocol. The classical and quantum security levels provided by hash functions depend on the digest length of the hash functions. Due to the influence of the Grover search algorithm, the post-quantum security level of hash functions is relatively lower than that in the traditional attack mode. The hash function with d bit output

Table 4 Process parameters and signature verification (partial content omitted)

Hash chain	Information in the digest value	Corresponding value	index	Verified public key block
0	0A	10	10	$H^{255-10}(\sigma_0)$
1	C0	192	192	$H^{255-192}(\sigma_1)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
31	82	130	130	$H^{255-130}(\sigma_{31})$
32	0	15	15	$H^{255-15}(\sigma_{32})$
33	1	362	107	$H^{255-107}(\sigma_{33})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
40	8	92	92	$H^{255-92}(\sigma_{40})$
41	9	123	123	$H^{255-123}(\sigma_{41})$
42	A	98	98	$H^{255-98}(\sigma_{42})$
43	B	46	46	$H^{255-46}(\sigma_{43})$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
47	F	100	100	$H^{255-100}(\sigma_{47})$
$pk' = H^{255-10}(\sigma_0), \ldots, H^{255-130}(\sigma_{31}), H^{255-15}(\sigma_{32}), \ldots, H^{255-123}(\sigma_{41}), H^{255-98}(\sigma_{42}), \ldots, H^{255-100}(\sigma_{47})$				

length can provide the classical security level of the first and second anti-preimage attacks and the post-quantum security level of the *d*/2 bits. However, collision resistance attack is a complex security requirement, and the hash function with general bit output length d can provide the classical *d*/2 security level and post-quantum *d*/3 security level to resist collision attack.

Anti-first preimage attack requires that the attacker cannot deduce the corresponding input x from the known hash output y, that is $f(x) = y$. Assuming that the attacker *ADV* can forge the SM3-OTS signature σ', then *ADV* must find a valid signature block σ'_i corresponding to the public key block pk_i, which is $f^{255-step'_i}(\sigma'_i) = pk_i$. $step'_i$ is the forged hash value of the message by the attacker. This means that the attacker needs to deduce a signature block σ'_i from the public key block pk_i. If the attacker can find this, the attacker has actually cracked the anti-first preimage property of hash functions.

Assuming that the attacker successfully replaces the signature element, the attacker needs to find two different signature blocks $\sigma_i \neq \sigma'_i$, but they get the same public key block after a certain number of hash operations, which is $f^{255-step_i}(\sigma_i) = f^{255-step'_i}(\sigma'_i)$. This means the second preimage attack resistance of the hash function has been cracked, because the attacker finds two different inputs σ_i and σ' and the two inputs get the same output pk_i after the same hash function operation.

The probability of an attacker successfully forging a signature can be calculated by the security of the hash function against the first preimage attack and the second preimage attack: $P_{\text{success}} \leq \varepsilon_{\text{pre-image}} + \varepsilon_{\text{second-pre-image}}$. $\varepsilon_{\text{pre-image}}$ is the failure probability of the hash function against the first preimage attack, and $\varepsilon_{\text{second-pre-image}}$ is the failure probability of the hash function against the second preimage attack. If the success probability P_{success} of the attacker *ADV*'s attack on the hash function employed in the signature scheme is negligible, it means that the hash function in the SM3-OTS scheme is secure, indicating that the SM3-OTS scheme is a secure digital signature scheme.

The core of the SM3-OTS scheme is based on hash functions, which are proven to have strong attack resistance in the face of quantum computing. Unlike signature schemes based on number theory (such as RSA and ECC), these number theory schemes can be cracked by quantum computers in polynomial time by adopting the Shor algorithm, but the resistance of hash functions will only be affected by the Grover algorithm, which is a quantum search algorithm.

It can speed up the search process at the speed of square roots, but it cannot completely crack the hash function. Quantum computers can reduce the security of the hash function by half via employing the Grover algorithm. For example, the Chinese cryptographic algorithm SM3 can provide 128-bit classical security, but on quantum computers, this security will be reduced to about 85 bits. This means that quantum computers can reduce the computation amount when cracking the hash function, but they cannot destroy the security of hash functions. Although quantum computers can reduce the partial security of the SM3 algorithm by adopting the Grover algorithm, this can be made up for by increasing the output length of the SM3 algorithm. Thus, SM3-OTS is still safe and reliable in the era of quantum computing.

6 Key and Signature Size

This section will explain the key size and generated signature size of the SM3-OTS signature scheme, and compare it with other mainstream one-time digital signature schemes.

6.1 Key and signature size

The private key of the SM3-OTS signature scheme is generated by the iterative operation of hash functions for secret seed *Seed* and serial number i, while the public key is generated by hashing the corresponding private key with hash functions. The size of the private key of the SM3-OTS signature scheme is $sk\_size = i \times 32 = 48 \times 32 = 1536$ (Bytes). Similarly, the size of the public key of the SM3-OTS scheme is $pk\_size = i \times 32 = 48 \times 32 = 1536$ (Bytes).

As the signature blocks of the SM3-OTS signature scheme are the chain nodes on the corresponding hash chains, the number of signature blocks corresponds to the number of hash chains, which is the number of private key blocks of the corresponding signature scheme. The signature size of the SM3-OTS scheme is $\sigma\_size = i \times 32 = 48 \times 32 = 1536$ (bytes).

SM3-OTS is a post-quantum one-time digital signature scheme based on hash function design, featuring smaller private keys and signature size and stronger security. For the WOTS+ scheme in SPHINCS$^+$, the Balanced WOTS+ scheme in SPHINCS-α, and the WOTS+C scheme in SPHINCS$^+$C, the length of the signature value generated by these mainstream one-time digital signature schemes is shortened by about 29%, 27% and 26%, respectively, thus significantly improving signing performance. This reduction not only helps to reduce the storage and bandwidth requirements but also improves the signature and verification efficiency. Therefore, SM3-OTS shows great potential in application scenarios that require efficient signature and verification. Table 5 provides the parameter size comparison between SM3-OTS and other hash-based one-time signature schemes.

Table 5 Comparison of parameters of several signature schemes

Signature scheme	Security parameter	Key size	Signature value size
WOTS	32	2,144	2,144
WOTS+	32	2,144	2,144
Balanced WOTS+	32	2,112	2,112
WOTS+C	32	2,080	2,080
LMOTS	32	2,144	2,144
SM3-OTS	32	1,536	1,536

6.2 Efficiency analysis of SM3-OTS

In this paper, SM3-OTS is compared with WOTS, WOTS+, Balanced WOTS+, and WOTS+C schemes. This test is run on AMD Ryzen 7840S CPU (3.3 GHz) equipped with 4 GB RAM, with the test environment of Ubuntu 22.04 LTS. Figures 5–7 show the time required for key generation, signature generation, and signature verification of the signature schemes, respectively. According to the experiment, the running time of the SM3-OTS digital signature scheme has sound performance. Compared with the WOTS+ scheme in the NIST standardization algorithm SPHINCS$^+$, the time required for key generation, signature generation, and signature verification of SM3-OTS is reduced by 27.2%, 18.7% and 25.3%, respectively, and the running efficiency of SM3-OTS is less different from that of Balanced WOTS+ in WOTS and SPHINCS-α and WOTS+C in SPHINCS$^+$C. Considering the signature values generated by the several signature schemes in Table 5, the SM3-OTS scheme has very sound application prospect.

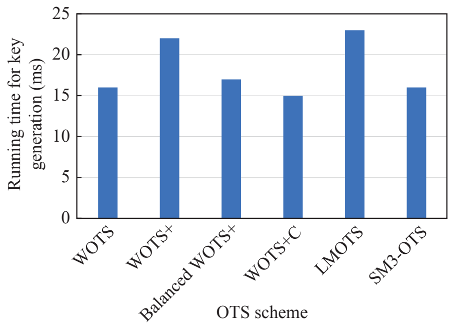


Figure 5 Comparison of key generation time of several signature schemes

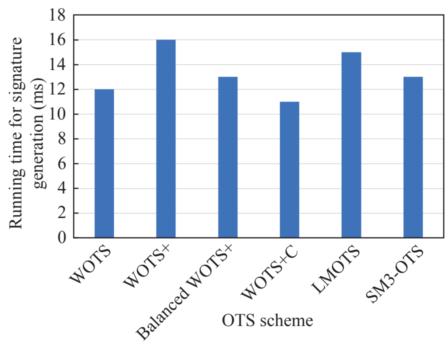


Figure 6 Comparison of signature generation time of several signature schemes

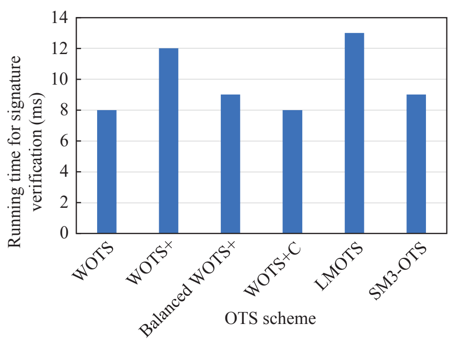


Figure 7 Comparison of signature verification time of several signature schemes

7 Conclusion

In view of the problem that the generated signature value of the existing one-time digital signature schemes based on hash functions is too long, in this paper we proposed SM3-OTS, a new one-time digital signature scheme based on hash functions. This scheme provides higher security than the WOTS and WOTS+ schemes, and the key generation time, signature generation time, and signature verification time of the SM3-OTS scheme are shorter than those of the above two schemes. The length of the key pair and signature value of the SM3-OTS signature scheme is 1,536 bytes, thus greatly shortening the length of the key pair and signature value of OTS schemes based on hash functions. Therefore, SM3-OTS provides better performance and higher security. In the future, SM3-OTS can be extended to a stateless digital signature scheme by expanding binary hash trees to improve the security and signature performance of post-quantum digital signature schemes based on hash functions.

References

[1] O'Brien JL. Optical quantum computing. Science, 2007, 318(5856): 1567–1570. [doi: 10.1126/science.1142892]

[2] Gruska J. Quantum Computing. London: McGraw-Hill, 1999.

[3] Cui FX, Wang B, Liu Y, Li Y. Research status and prospect of quantum attacks in public-key cryptography. Cyber Security and Data Governance, 2022, 41(9): 3–12 (in Chinese with English abstract). [doi: 10.19358/j.issn.2097-1788.2022.03.001]

[4] Shor PW. Algorithms for quantum computation: Discrete logarithms and factoring. Proc. of the 35th Annual Symp. on Foundations of Computer Science. IEEE, 1994. 124–134.

[5] Grover LK. A fast quantum mechanical algorithm for database search. Proc. of the 28th Annual ACM Symp. on Theory of Computing. 1996. 212–219.

[6] Bernstein DJ, Lange T. Post-quantum cryptography. Nature, 2017, 549(7671): 188–194. [doi: 10.1038/nature23461]

[7] Bos J, Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schanck JM, Schwabe P, Seiler G, Stehle D. CRYSTALS-Kyber: A CCA-secure module-lattice-based KEM. Proc. of the 2018 IEEE European Symp. on Security and Privacy (EuroS&P). London: IEEE, 2018. 353–367. [doi: 10.1109/EuroSP.2018.00032]

[8] Ducas L, Kiltz E, Lepoint T, Lyubashevsky V, Schwabe P, Seiler G, Stehlé D. Crystals-Dilithium: A lattice-based digital signature scheme. IACR Trans. on Cryptographic Hardware and Embedded Systems, 2018, 2018(1): 238–268. [doi: 10.13154/tches.v2018.i1.238-268]

[9] Soni D, Basu K, Nabeel M, Aaraj N, Manzano M, Karri R. Hardware Architectures for Post-quantum Digital Signature Schemes. Cham: Springer, 2021. 31–41. [doi: 10.1007/978-3-030-57682-0]

[10] Bernstein DJ, Hülsing A, Kölbl S, Niederhagen R, Rijneveld J, Schwabe P. The SPHINCS$^+$ signature framework. Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security (CCS19). London: ACM, 2019. 2129–2146. [doi: 10.1145/3319535.3363229]

[11] Yang YT, Zhao RY, Chang X, Guo C, Xiao S. PQVPN: Design of software VPN against quantum computing attack. Journal of Cyber Security, 2022, 7(5): 108–119 (in Chinese with English abstract). [doi: 10.19363/J.cnki.cn10-1380/tn.2022.09.09]

[12] Yang YT, Chang X, Shi HP, Wang W, Wang K. CDBS: Blind signature scheme based on CRYSTALS-Dilithium algorithm. Journal on Communications, 2024, 45(7): 184–195 (in Chinese with English abstract). [doi: 10.11959/j.issn.1000-436x.2024129]

[13] National Institute of Standards and Technology. Recommendation for Stateful Hash-based Signature Schemes: NIST SP 800-208. Gaithersburg: U.S. Department of Commerce, 2020.

[14] Hülsing A, Butin D, Gazdag S, Rijneveld J, Mohaisen A. XMSS: Extended Merkle signature scheme. RFC 8391, 2018.

[15] Zhang KY, Cui HR, Yu Y. SPHINCS-α: A compact stateless hash-based signature scheme. Cryptology ePrint Archive, 2023/850, 2022.

[16] Kudinov M, Hülsing A, Ronen E, Yogev E. SPHINCS$^+$C: Compressing SPHINCS$^+$ with (almost) no cost. Cryptology ePrint Archive, 2022/778, 2022.

[17] Suhail S, Hussain R, Khan A, Hong CS. On the role of hash-based signatures in quantum-safe Internet of Things: Current solutions and future directions. IEEE Internet of Things Journal, 2021, 8(1): 1–17. [doi: 10.1109/JIOT.2020.3013019]

[18] Kumar A, Ottaviani C, Gill SS, Buyya R. Securing the future Internet of Things with post-quantum cryptography. Security and Privacy, 2022, 5(2): e200. [doi: 10.1002/spy2.200]

[19] Lamport L. Constructing digital signatures from a one way function. Technical Report, CSL-98, SRI Int'l Computer Science Laboratory, 1979.

[20] Merkle RC. Secrecy, authentication, and public key systems [Ph.D. Thesis]. Stanford: Stanford University, 1979.

[21] Merkle RC. A certified digital signature. Proc. of the 1989 Conf. on the Theory and Application of Cryptology. New York: Springer, 1989. 218–238. [doi: 10.1007/0-387-34805-0_21]

[22] Sun SW, Liu TY, Guan Z, He YF, Hu L, Jing JW, Zhang LT, Yan HL. SPHINCS$^+$-SM3: SM3-based stateless digital signature scheme. Journal of Cryptologic Research, 2023, 10(6): 1266–1278 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000658]

[23] Sun SW, Liu TY, Guan Z, He YF, Jing JW, Hu L, Zhang ZF, Yan HL. SM3-based post-quantum digital signature schemes. Journal of Cryptologic Research, 2023, 10(1): 46–60 (in Chinese with English abstract). [doi: 10.13868/j.cnki.jcr.000578]

[24] Li LY, Lu XH, Wang KP. Hash-based signature revisited. Cybersecurity, 2022, 5(1): 13. [doi: 10.1186/s42400-022-00117-w]

[25] Wang JD, Zhang T, Song JK, Sebe N, Shen HT. A survey on learning to hash. IEEE Trans. on Pattern Analysis and Machine Intelligence, 2018, 40(4): 769–790. [doi: 10.1109/TPAMI.2017.2699960]

Yatao Yang, Ph.D., professor, doctoral supervisor. His research interests include cryptography and information security, post-quantum cryptography, homomorphic encryption, cryptographic protocols and algorithms.

Liangyu Chen, master's degree candidate. His research interests include cryptography and information security, post-quantum cryptography.

Fangrui Yin, master's degree candidate. His research interests include cryptography and information security, and post-quantum cryptography.

Deng Pan, Ph.D. candidate. His research interests include cryptography and information security, and lattice-based cryptography.