软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, [doi: 10.13328/j.cnki.jos.007392]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

SM3-OTS:基于国密算法 SM3 的紧凑型后量子一次签名方案 *

杨亚涛[1,2], 殷方锐[1], 陈亮宇[1], 潘 登[1]

[1](北京电子科技学院 电子与通信工程系,中国 北京 100070)

[2](西安电子科技大学 通信工程学院,陕西 西安 710071)

通讯作者: 杨亚涛, E-mail: yy2008@163.com

摘 要: SPHINCS$^+$是一种基于哈希函数设计的无状态数字签名方案,其被证明能够抵抗量子计算攻击.然而,由于其生成的签名值较大,限制了 SPHINCS$^+$在实际中的广泛应用.为解决 SPHINCS$^+$签名方案中 WOTS+一次签名方案生成的签名值长度较大问题,本文设计了一种新的基于国密算法 SM3 的紧凑型一次签名方案 SM3-OTS.该签名方案利用消息摘要值的二进制信息和十六进制信息分别作为前32条哈希链和后16条哈希链节点位置的索引,从而有效缩短了传统基于哈希函数一次签名方案的密钥长度和生成签名值长度.SM3-OTS 相较于 SPHINCS$^+$中使用的 WOTS+、SPHINCS-α 中使用的 Balanced WOTS+以及 SPHINCS$^+$C 中使用的 WOTS+C 所生成的签名值长度大约缩短了 29%、27%、26%,签名性能得到明显提升.同时,通过采用国密 SM3 算法,使得 SM3-OTS 具备良好的抗量子攻击能力,并保持了较好的综合性能.

关键词: 哈希函数;SPHINCS$^+$;数字签名;一次签名;后量子密码

中图法分类号: TN918.4

中文引用格式: 杨亚涛, 殷方锐, 陈亮宇, 潘登. 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. http://www.jos.org.cn/1000-9825/7392.htm

英文引用格式: Yang YT, Yin FR, Chen LY , Pan D. SM3-OTS: A compact post quantum one time signature scheme over SM3 algorithm. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7392.htm

SM3-OTS:A compact post quantum one time signature scheme over SM3 algorithm

YANG Ya-Tao[1,2], YIN Fang-Rui[1], CHEN Liang-Yu[1], PAN Deng[1]

[1](Department of Electronic and Communication Engineering, Beijing Electronics Science and Technology Institute, Beijing 100070, China)

[2](School of Telecommunication Engineering, Xidian University, Xian 710071, China)

Abstract: SPHINCS$^+$ is a stateless digital signature scheme based on hash function, which has been proven to be resistant to quantum computing attacks. However, due to the large number of signatures generated, SPHINCS$^+$ is limited to its wide application in practice. In order to solve the problem of large length of signature value generated by WOTS+ one-time signature scheme in SPHINCS$^+$ signature scheme, a new compact one-time signature scheme SM3-OTS based on SM3 algorithm is designed in this paper. The signature scheme uses the binary and hexadecimal information of the message digest value as the index of the node positions of the first 32 hash chains and the last 16 hash chains, respectively, which effectively shortens the key length and the length of the generated signature value of the traditional one-time signature scheme based on hash function. Compared with WOTS+ in SPHINCS$^+$, Balanced WOTS+ in SPHINCS-α and WOTS+C in SPHINCS$^+$C, SM3-OTS shortens the length of signature values by about 29%, 27% and 26% respectively, and the signature performance is significantly improved. At the same time, by adopting the SM3 algorithm, the SM3-OTS has good anti-quantum attack ability and maintains good comprehensive performance.

Key words: hash function; SPHINCS$^+$; digital signature; one time signature; post quantum cryptographic

* 基金项目:北京市自然科学基金 (4232034), 中央高校基本科研业务经费(3282023017, 3282024058, 3282024052)
收稿时间: 2024-07-01; 修改时间: 2024-09-05; 采用时间: 2024-12-30; jos 在线出版时间: 2025-01-20

1 引言

随着量子计算技术[1][2]的飞速发展,传统基于椭圆曲线、离散对数、大整数分解等数学困难问题的密码方案将面临量子计算攻击[3].1994 年, Peter Shor 提出的 Shor 算法[4]是一种能够高效地解决大整数分解问题的量子算法;1996 年,Lov Grover 提出了量子搜索 Grover 算法[5]通过暴力破解的方式将破解对称密码算法密钥的时间复杂度从 $O(2^n)$ 降低到 $O(2^{n/2})$,量子计算机通过使用 Shor 算法和 Grover 算法,对传统的公钥密码算法和对称密码算法构成了严重威胁,为应对这些威胁,具有抵抗量子攻击能力的后量子密码算法(Post Quantum Cryptography, PQC)[6]的研究与优化工作迫在眉睫.2016 年 12 月,美国国家标准与技术研究院(National Institute of Standards and Technology, NIST)正式面向全世界征集具备抵抗量子计算机攻击能力的后量子密码算法,以在量子时代逐步取代传统公钥密码算法.

2022 年 7 月,NIST 从第三轮后量子密码算法提交的方案中,初步选定四种算法作为标准化算法,包括一种公钥加密算法和三种数字签名算法,分别是公钥加密算法 CRYSTALS-KYBER[7] 数字签名算法 CRYSTALS-DILITHIUM[8]、FALCON[9]以及 SPHINCS$^+$[10].其中 KYBER、DILITHIUM、FALCON 是基于格理论问题构建的,2022 年,Yang 等人基于 CRYSTALS-KYBER 密码方案设计了一套抗量子计算攻击的 VPN 软件系统[11],展现了后量子密码方案在网络安全中的工程应用.区别于上述三种密码方案, SPHINCS$^+$是其中唯一基于哈希函数构建的密码方案.且后续 NIST 也会陆续征集并标准化其他安全性及性能更加优秀的密码算法.

目前后量子密码方案大致可以分为基于编码、基于格、基于多变量以及基于哈希函数[12]等类型的密码方案[13].基于哈希函数设计的数字签名方案的签名生成和验证时间最短,且安全性更可靠,所以基于哈希函数的数字签名方案被认为是量子时代十分具有研究价值的后量子密码方案.SPHINCS$^+$[10]是一种基于哈希函数设计的无状态数字签名方案,SPHINCS$^+$是利用一次签名方案(One Time Signature, OTS)、少次签名方案(Few Times Signature, FTS)和 Merkle Tree[14]构建的.一次签名方案为 SPHINCS$^+$方案的核心,一般基于哈希函数设计的一次签名方案存在生成的签名值长度较大问题,造成了 SPHINCS$^+$等基于哈希函数的无状态数字签名方案在实际应用中的困难.

本文贡献:

(1) 设计了一种新的基于国密算法 SM3 的后量子一次签名方案 SM3-OTS:本方案通过利用消息摘要值的二进制信息(Binary)和十六进制信息(Hexadecimal)分别作为前 32 条哈希链和后 16 条哈希链节点位置的索引,从而有效地缩短了传统基于哈希函数的一次签名方案的密钥长度和生成的签名值长度.

(2) 测试了新的基于国密算法 SM3 的一次签名方案 SM3-OTS 性能:通过实验测试,SM3-OTS 相较于 SPHINCS$^+$[10]中使用的 WOTS+、SPHINCS-α [15]中使用的 Balanced WOTS+以及 SPHINCS$^+$C[16]中使用的 WOTS+C,SM3-OTS 方案所生成的签名值长度大约缩短了 29%、27%、26%,签名性能得到明显提升.

2 国内外研究现状

利用哈希函数的抗原像攻击特性(单向性)构造数字签名方案是一个众多密码研究者关注并认可的研究思路.2020 年 Suhail 团队[17]、2021 年 Kumar 团队[18]针对量子计算技术成熟后的基于哈希函数的数字签名方案在物联网设备安全中的重要性、部署时需要考虑的各种因素和各种挑战进行了分析,并对物联网设备网络环境中应用后量子密码技术给出了相应建议和展望.

1979 年 Leslie Lamport 提出了第一种基于哈希函数的一次签名方案 Lamport-OTS(LOTS)[19],LOTS 方案签名时对消息逐位进行签名,LOTS 方案有两个缺陷,其一是方案的每对密钥只能使用一次,否则敌手可以利用前一条消息的签名值伪造签名,为避免伪造攻击,LOTS 方案的每对密钥只能使用一次;其二是方案使用的公私钥长度以及生成的签名值长度非常大,若签名时选择的安全参数为 n ,则私钥的长度为 $2\times n\times n$,公钥的长度同样为 $2\times n\times n$,方案最后生成的签名值长度为 $n\times n$ .LOTS 方案公私钥及签名值的长度过大造成其不适合在实际中应用.同年,Ralph C. Merkle 在 Lamport-OTS 方案的基础上提出了 Winternitz 一次签名方案(Winternitz-OTS,

WOTS)[20][21],WOTS 方案中使用了哈希链(hash chain)结构来构造签名方案的公钥及私钥,哈希链是对一个数据重复进行哈希运算生成的链,哈希链中后一个节点为前一个结点的哈希值.与 LOTS 方案类似,WOTS 方案的每对密钥同样只能签署一条消息.对比 LOTS 方案中每个消息摘要位对应 n 字节的哈希值,在 WOTS 方案中使用 2、4 或 16 个消息摘要位对应 n 字节的哈希值,极大的缩短了 LOTS 方案生成的签名值长度.

由于 OTS 方案的每对密钥只能使用一次,Ralph C. Merkle 提出了 Merkel Tree 签名方案(Merkle Signature Scheme, MSS)[20][21][22],该方案利用 Merkle Tree 的结构将一次签名方案扩展为少次签名方案(Few-Time Signatures, FTS),Merkle Tree 本质上是一种二叉哈希树.在 MSS 方案中使用 Merkle Tree 的叶子节点管理一次签名方案的公钥,将 Merkle Tree 的根节点作为 MSS 方案的公钥,实现将一次签名方案扩展为少次签名方案.若方案中 Merkel Tree 的高度为 h ,则一棵 Merkle Tree 有 2^h 个叶子节点,这些叶子节点用于存储 OTS 方案公钥的哈希值,即一棵高度为 h 的 Merkel Tree 可以管理 2^h 个 OTS 密钥,可以签署 2^h 条消息.Merkle Tree 签名方案生成的签名值主要包括两个部分:Merkle Tree 叶子节点对应的一次签名方案生成的签名值以及对此叶子节点的身份验证路径.这两部分数据共同构成了 Merkle Tree 签名方案的签名值.OTS 方案是所有哈希基签名的核心,使用哈希树可以将 OTS 方案扩展为多次签名方案.

SPHINCS$^+$中使用了 WOTS+一次签名方案以及 FORS 少次签名方案,并利用扩展 Merkle Tree 的方式将其扩展为无状态的数字签名方案.2022 年上海交通大学团队在 SPHINCS-α 方案[15]中对 SPHINCS$^+$中使用的一次签名方案 WOTS+进行了改进提出了 Balanced WOTS+,缩短了 WOTS+一次签名方案生成的签名值并提升了安全性.同年 Andreas Hülsing、Mikhail Kudinov 团队在 SPHINCS$^+$C 方案[16]中对 SPHINCS$^+$中使用的一次签名方案 WOTS+进行了改进提出了 WOTS+C, WOTS+C 取消了原本的三条校验链,而是在签名前将消息摘要值后串联 n 字节的随机数,使用哈希函数对串联随机数后的消息摘要值再作哈希运算生成哈希值,使用后一部分哈希值作为哈希链节点位置的索引.

使用相同的安全参数 n =32,则 WOTS+、Balanced WOTS+、WOTS+C、LMOTS 生成的签名值长度(Bytes)对比如表 1 所示.

表 1 WOTS+方案与几种变体生成的签名值长度对比

方案	生成签名的值长度(Bytes)
WOTS+	2144
Balanced WOTS+	2112
WOTS+C	2080
LMOTS	2144

中国科学院大学团队使用国密哈希算法 SM3 替代了 SPHINCS$^+$数字签名方案所使用的哈希函数,并给出了 SM3 实例化 SPHINCS$^+$后的相关实验结果[23],表明国密 SM3 算法实现国际标准化的哈希基数字签名方案是切实可行的.同时,该团队利用国产杂凑函数 SM3 替代 LMS(Leighton-Micali Signature system)方案中所使用的哈希函数,并给出了初步的实验结果[24].另外,该校团队系统总结了基于哈希的签名组件的研究进展,深入分析了不同类型的基于哈希的签名方案整体设计思路等[25].

郁昱教授以及 Andreas Hülsing 等团队对 WOTS+方案的优化推动了基于哈希函数的后量子签名方案标准化及可应用化,也为哈希基签名方案的优化提供了思路.但优化后的方案生成的签名值长度还是较大,所以对哈希基数字签名方案的研究及优化工作还需要进一步推进.

3 基础知识

本节将介绍 SM3-OTS 签名方案的基础知识、表示符号以及哈希函数相关概念.

3.1 符号说明

符号说明如表 2 所示.

表 2 符号说明

符号	说明
M	待签名的明文消息
H	签名方案所使用的国密算法 SM3
m	待签名消息的摘要值, $m = H(M)$
n	签名方案使用的安全参数
$PRNF$	伪随机数生成函数, 用于生成随机数
$base\_8$	将哈希值转为每 8 个比特为一组的十进制数
$Seed$	秘密种子, 为 $PRNF$ 生成随机数使用的种子
m_bin	消息摘要的二进制信息
m_hex	消息摘要的十六进制信息
KeyGen( )	密钥生成算法
Sign( )	签名生成算法
Verify( )	签名验证算法
sk	签名方案私钥,用于对消息生成签名值
pk	签名方案公钥,用于对消息签名值有效性验证
pk'	验证公钥,若与 pk 相等,则签名值有效
σ	签名方案生成的签名值
ADV	攻击者,意图伪造数字签名
sk_size	私钥大小(Bytes)
pk_size	公钥大小(Bytes)
σ_size	签名值大小(Bytes)

3.2 哈希函数

哈希函数可以将任意长度的输入映射为固定长度的输出,该输出值一般称为哈希值、摘要值或散列值[26].这种映射是一种压缩映射,因为散列值的空间远小于输入的空间,不同的输入可能会映射成相同的输出,所以不可能从散列值来确定唯一的输入值.一个理想的哈希函数应具有以下性质:

（1）确定性:即相同的输入信息经同一个哈希函数始终产生相同的散列值.

（2）不可逆性:从散列值计算输入信息在计算上是不可行的.

（3）无碰撞性:找到相同散列值的两个输入在计算上是不可行的.

3.3 哈希函数安全性

哈希函数的安全性是哈希基数字签名方案最基本的安全要求.一个理想的哈希函数具有如下三个性质[26].

（1）抗第一原像攻击:抗第一原象攻击性是指对于给定的哈希值 h ,在合理的时间内找到任何原始输入 x ,使得 $H(x) = h$ 是困难的.即,对于给定的 h ,找到任意 x 使得 $H(x) = h$ 在计算上是不可行的[26].

（2）抗第二原像攻击:抗第二原象攻击性指的是对于一个固定的输入 x ,在合理的时间内很难找到一个不同的输入 y ,使得 $H(x) = H(y)$.即,对于固定的 x 和任意的 $y \neq x$,找到 $H(x) = H(y)$ 在计算上是不可行的[26].

（3）抗碰撞攻击:抗碰撞攻击性指的是在合理的时间内很难找到两个不同的输入 x 和 y 使得它们的哈希值相同,即 $H(x) = H(y)$.即,对于所有 $x \neq y$,找到 $H(x) = H(y)$ 在计算上是不可行的[26].

4 SM3-OTS 方案

SM3-OTS 方案采用了类似 WOTS+方案中哈希链结构对消息进行签名,SM3-OTS 方案生成的签名值长度较 WOTS+方案缩短了大约 29%,这种优化显著提升了签名的存储和传输效率,使得 SM3-OTS 方案在存储资源受限的环境中较 WOTS+方案具有更大的应用潜力.

SM3-OTS 方案主要包含三个算法:密钥生成算法 KeyGen( )、签名生成算法 Sign( )、签名验证算法 Verify( ). 密钥生成算法接受一个安全参数 n 作为输入,经过相应运算输出签名密钥对 (sk, pk),私钥 sk 为只有签名者拥有,用于对消息进行签名;公钥 pk 可以被任何人获得,用于对消息的签名作有效性验证.签名生成算法接受待签名的消息 M,私钥 sk 两个输入,经过算法后生成消息 M 对应的签名值 σ.签名验证算法接受消息 M, M 对应的签名值 σ 以及公钥 pk 三个输入,签名验证算法可验证签名值的有效性,若签名值有效则输出 true,若签名值无效则输出 false.

4.1 SM3-OTS方案密钥生成

在 n =256bits 的安全参数下,将秘密种子 *Seed* 作为伪随机数发生器 PRNG 的种子数据生成 48 个伪随机数: $sk_0, sk_1, \cdots, sk_{47}$,每个伪随机数为 32 字节的伪随机十六进制数据称为一个私钥块,SM3-OTS 签名方案私钥 sk 包含 48 个私钥块,记为 $sk = sk_0, sk_1, \cdots, sk_{47}$.

SM3-OTS 签名方案公钥由对应的私钥生成,使用国密算法 SM3 将上述 48 个私钥块 sk_i 分别作 255 次哈希运算,即 $pk_i = H^{255}(sk_i)$,生成 48 个公钥块: $pk_0, pk_1, \cdots, pk_{47}$,SM3-OTS 签名方案公钥 pk 包含 48 个公钥块,记为 $pk = pk_0, pk_1, \cdots, pk_{47}$.从私钥生成公钥的过程得到 48 条哈希链,分别为 $L_0, L_1, \cdots, L_{47}$,每条哈希链中包含 256 个链节点,每条哈希链中相邻两个链节点中后一个节点为前一个节点的哈希值,每条哈希链的首节点为该哈希链对应私钥块,每条哈希链的尾节点为该哈希链对应的公钥块,密钥生成如算法 1 所示.

算法 1 密钥生成 KeyGen 算法

输入: 安全参数 n.

输出: 私钥集合 $sk = sk_0, sk_1, \cdots, sk_{47}$; 公钥集合 $pk = pk_0, pk_1, \cdots, pk_{47}$.

1. $Seed \Leftarrow PRNG$ //通过伪随机数发生器 PRNG 生成随机种子
2. for $i = 0, \cdots, 47$ //利用种子生成 48 个私钥块
3. $sk_i = PRNF(Seed, i)$
4. end for
5. for $i = 0, \cdots, 47$ //利用私钥块生成对应公钥块
6. $pk_i = H^{255}(sk_i)$ //对每个私钥块作 255 次哈希运算得到对应公钥块
7. end for
8. return $sk = sk_0, sk_1, \cdots, sk_{47}$; $pk = pk_0, pk_1, \cdots, pk_{47}$ //输出密钥对

如算法 1 所示,在预设的安全参数 n =256bits 下,使用秘密种子 *Seed* 和伪随机数发生器 PRNG 随机生成 48 个 n 字节的随机数作为私钥,分别将 48 个私钥块使用国密算法 SM3 做 255 次哈希运算生成 48 个 n 字节的哈希值作为公钥,密钥生成如图 1 所示,哈希链结构如图 2 所示.

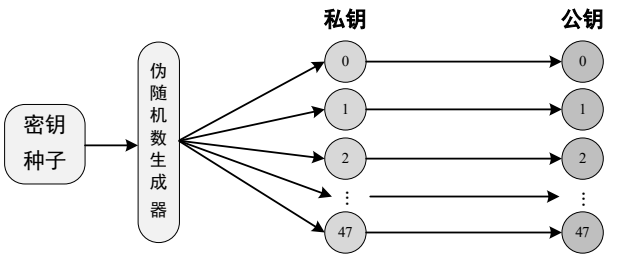

图1 SM3-OTS 密钥生成

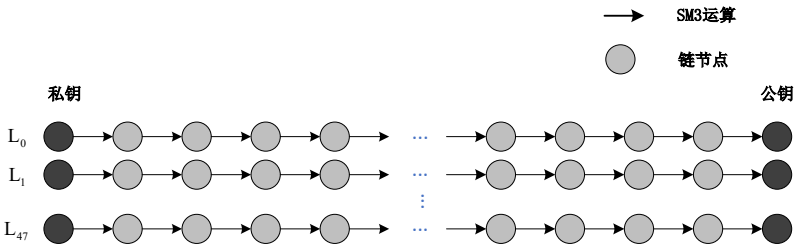

图2 哈希链结构

4.2 SM3-OTS签名生成

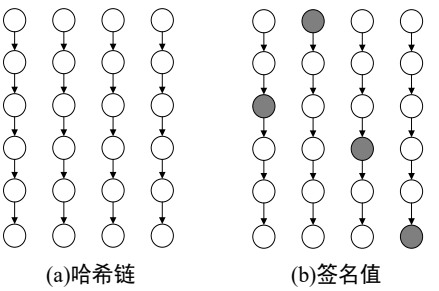

图 3 哈希链映射

SM3-OTS 签名生成是将消息摘要的信息映射到哈希链节点的过程.如图 3 所示,图 3-(a)为签名方案的哈希链,将待签名的消息摘要信息处理后映射到对应的链节点如图 3-(b)中深色节点,被映射到的链节点将作为签名块记录到最后生成的签名值中.本节将详细介绍 SM3-OTS 签名过程中的具体映射方式.

表 3 过程参数与签名(部分省略)

哈希链	哈希值中的信息	对应值	index	签名块
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	$H^{100}(sk_{47})$
$\sigma = H^{10}(sk_0),\cdots,H^{130}(sk_{31}),H^{15}(sk_{32}),\cdots,H^{123}(sk_{41}),H^{98}(sk_{42}),\cdots,H^{100}(sk_{47})$				

首先使用国密算法 SM3 对待签名的消息 M 作哈希运算: $m=H(M)$,得到 32 字节的消息摘要值,使用 m_bin 记录下消息摘要的二进制信息,使用 m_hex 记录下消息摘要的十六进制信息.32 字节消息摘要的二进制形式每 8 位比特可以转为 32 组 0-255 之间的十进制数字.使用消息摘要转换后的十进制数字分别作为前 32 条哈希链 L_0 - L_{31} 中链节点位置的索引,将索引到的链节点作为 SM3-OTS 签名值中的前 32 个签名块,记为 $\sigma_0,\sigma_1,\cdots,\sigma_{31}$.例如“Hello World!”经过 SM3 算法计算出的消息摘要值为 0AC0A9FEF0D212AA76A3C431F793853CE145659CA1D14B114E96C1215CF26582.其中第一个字节为 0A,转换为二进制为 00001010,则其对应的十进制数字为 10,将第一个私钥块 sk_0 作 10 次哈希运算得到第一个签名块 σ_0.摘要中第二个字节为 C0,转换为二进制的结果为 11000000,则其对应的十进制数字为 192,将第二个私钥块 sk_1 作 192 次哈希运算得到第二个签名块 σ_1,摘要中第三个字节为 A9,转换为二进制的结果为 10101001,则其对应的十进制数字为 169,将第三个私钥块 sk_2 作 169 次哈希运算得到第二个签名块 σ_2,以此类推计算出 0 到 31 号哈希链对应的签名块,如表 3 所示.

32 字节消息摘要的十六进制形式包含 64 个 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F 中的数据元素,以十六进制形式的消息摘要下第一个元素的位置定义为 1,最后一个元素的位置定义为 64,计算出 0-F 的每个元素所在位置的和作为每个元素的位置信息,将每个元素的位置信息模除 255,元素 0 的位置信息为 1+4+10=15,将计算出的位置信息模除 255 得到 15,将私钥块 sk_{32} 作 15 次哈希运算得到第 33 个签名块 σ_{32}.元素 1 的位置信息经计算为 362,将 362 模处 255 后得 107,将私钥块 sk_{33} 作 107 次哈希运算得到第 34 个签名块 σ_{33}.以此类推,使用每个元素位置信息模除 255 后的数据作为后 16 条哈希链 L_{32} - L_{47} 中链节点位置的索引,将索引到的链节点作为 SM3-OTS 签名值中的后 16 个签名块,记为 $\sigma_{32},\sigma_{33},\cdots,\sigma_{47}$,如表 3 所示.

所以 SM3-OTS 方案生成的签名值为 $\sigma=\sigma_0,\sigma_1,\cdots,\sigma_{47}$.签名生成算法如算法 2 所示.

算法 2 签名生成 Sign 算法

输入: 待签名的消息 M,私钥集合 $sk=sk_0,sk_1,\cdots,sk_{47}$

输出: 签名 $\sigma=\sigma_0,\sigma_1,\cdots,\sigma_{47}$

1. $m=H(M)$ //计算消息 M 的摘要值
2. $m\_bin \Leftarrow m$ //将摘要值转为二进制形式存储到 m_bin 中
3. $a=base\_8(m\_bin)$ //将 m_bin 中每 8 个比特转为一个十进制数据存储到 a 中
4. for $i=0,\cdots,31$ do
5. $step[i]=a[i]$
6. $\sigma_i=H^{step[i]}(sk_i)$ //将私钥块分别作 $step[i]$ 次哈希运算得到对应签名块
7. end for
8. $hex\_symbols=0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F$
9. $m\_hex \Leftarrow m$
10. for $i=0,\cdots,15$ do
11. $sum[i]=0$
12. for $j=1,\cdots,64$ do
13. if $m\_hex[j]=hex\_symbols[i]$ //计算每个字符的位置信息
14. $sum[i]=sum[i]+j$
15. end if
16. end for
17. $step[i+32]=sum[i](\mathrm{mod}\,255)$ //将每个字符的位置信息值模 255
18. $\sigma_{i+32}=H^{step[i+32]}(sk_{i+32})$ //将私钥块分别作 $step[i+32]$ 次运算得到对应签名块

19. end for

20. return $\sigma=\sigma_0,\sigma_1,\cdots,\sigma_{47}$ //返回签名值

4.3 SM3-OTS签名验证

SM3-OTS 签名验证的过程是从签名值中还原公钥的过程.SM3-OTS 的公钥是由对应私钥经过计算生成的,每条哈希链中间节点为对应私钥块生成公钥块的过程节点,所以公钥值可以由私钥计算得到的,同样可由哈希链中间节点经相应的计算获得.例如图 4 中 4 号链生成的对应签名块为对应私钥块作 2 次哈希运算得到,所以对该签名块作 3 次哈希运算即可得到对应的公钥块.

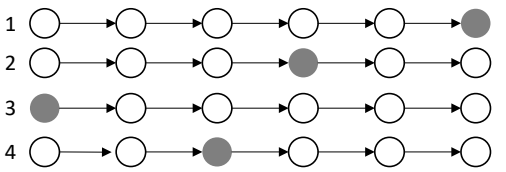

图 4 签名链

签名验证者对消息的处理过程同签名生成部分类似,验证者首先计算消息摘要的二进制和十六进制两种形式,计算消息摘要二进制形式下每 8 位比特为一组的 32 组十进制数字,计算出消息摘要十六进制形式下元素 0-F 的位置信息,使用 255 与上述信息做差分别验证签名块是否有效.若计算的值同公钥值相同则签名值有效,反之则签名无效.签名验证具体过程如算法 3 所示.以“Hello World!”为例的签名验证信息如表 4 所示.

算法 3 签名验证 Verify 算法

输入: 消息 M ; 签名值 $\sigma=\sigma_0,\sigma_1,\cdots,\sigma_{47}$;公钥集合 $pk=pk_0,pk_1,\cdots,pk_{47}$

输出: 签名值有效 true; 签名值无效 false

1. $m=H(M)$ //计算消息 M 的摘要值
2. $m\_bin \Leftarrow m$ //将摘要值转为二进制形式存储到 m_bin 中
3. $a=base\_8(m\_bin)$ //将 m_bin 中每 8 个比特转为一个十进制数据存储到 a 中
4. for $i=0,\cdots,31$ do
5. $step[i]=a[i]$
6. $pk_i'=H^{255-step[i]}(\sigma_i)$ //将签名块分别作 $(255-step[i])$ 次运算得到对应验证公钥块
7. end for
8. $hex\_symbols=0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F$
9. for $i=0,\cdots,15$
10. $sum[i]=0$
11. for $j=1,\cdots,64$ do
12. if $m\_hex[j]=hex\_symbols[i]$ //计算每个字符的位置信息
13. $sum[i]=sum[i]+j$
14. end if
15. end for
16. $step[i+32]=sum[i](\mathrm{mod}\,255)$ //将每个字符的位置信息值模 255
17. $pk'_{i+32}=H^{255-step[i+32]}(\sigma_{i+32})$ //将签名块分别作 $(255-step[i+32])$ 次运算得到验证公钥块

18. $pk' = pk'_0, pk'_1, \cdots, pk'_{47}$
19 if $pk' = pk$
20 return true //若签名值有效则返回 true
21. else
22. return false //若签名值有效则返回 false

表 4 过程参数与签名验证(部分省略)

哈希链	摘要值中的信息	对应值	index	验证公钥块
0	0A	10	10	$H^{255-10}(\sigma_0)$
1	C0	192	192	$H^{255-192}(\sigma_1)$
……				
31	82	130	130	$H^{255-130}(\sigma_{31})$
32	0	15	15	$H^{255-15}(\sigma_{32})$
33	1	362	107	$H^{255-107}(\sigma_{33})$
……				
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	$H^{255-46}(\sigma_{43})$
……				
47	F	100	100	$H^{255-100}(\sigma_{47})$
$pk' = H^{255-10}(\sigma_0), \cdots, H^{255-130}(\sigma_{31}), H^{255-15}(\sigma_{32}), \cdots, H^{255-123}(\sigma_{41}), H^{255-98}(\sigma_{42}), \cdots, H^{255-100}(\sigma_{47})$				

5 安全性证明

在证明 SM3-OTS 自适应选择消息攻击(Chosen-Message Attack, CMA)下存在不可伪造性时,首先需要理解哈希函数的单向性和抗碰撞性对签名方案安全性的影响.哈希函数的单向性保证了给定的哈希值无法有效反推出原始输入,而抗碰撞性确保不同输入不会产生相同的哈希值.本文将 SM3-OTS 的安全性归约到这些哈希函数特性上,即如果攻击者能够在 CMA 下成功伪造签名,那么攻击者可以利用这个伪造过程破解哈希函数的单向性或抗碰撞性.例如,如果攻击者能够伪造签名,攻击者可能通过逆向工程签名生成过程来找到哈希函数的碰撞或逆向输入,这违反了哈希函数的基本特性.通过这种归约过程,可以证明 SM3-OTS 的安全性确实依赖于哈希函数的单向性和抗碰撞性,因此选择一个安全的哈希函数对确保签名方案的整体安全性至关重要.

在第一原像攻击的挑战中,对手 ADV 的挑战是在给定输出计算出对应的输入.

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

在第二原像攻击的挑战中,对手 ADV 知道输入-输出对 (x, y), ADV 的挑战是需要找到另一个与输入 x 不同的输入 x',但输出为 y.

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

SM3-OTS 密码方案对不同类型攻击的抵抗能力通常被称为该协议提供的安全级别.哈希函数提供的经典和量子安全级别取决于哈希函数的摘要长度.由于 Grover 搜索算法的影响,哈希函数的后量子安全级别相对小

于传统攻击模式下的安全级别. d 位输出长度的哈希函数能够提供第一和第二抗原像攻击提供 d 位经典安全级别以及 $d/2$ 位的后量子安全性.但是,抗碰撞攻击相对是一个复杂的安全要求,一般 d 位输出长度的哈希函数可以供 $d/2$ 位经典安全级别以及 $d/3$ 位后量子安全级别对抗碰撞攻击.

抗第一原像攻击要求攻击者无法从已知的哈希输出 y 推导出相应的输入 x,即 $f(x)=y$.假设攻击者 ADV 能够伪造 SM3-OTS 签名 σ',那么 ADV 必须找到与公钥块 pk_i 对应的有效签名块 σ_i',即 $f^{255-step_i'}(\sigma_i')=pk_i$, $step_i'$ 是攻击者对消息的伪造哈希值.这意味着攻击者需要从公钥块 pk_i 反推出签名值中的某个签名块 σ_i'.如果攻击者能够找到这一点,如果攻击者能够找到这样的 σ_i',那么攻击者实际上就破解了哈希函数的抗第一原像性.

假设攻击者成功替换签名元素,攻击者需要找到两个不同的签名块 $\sigma_i \neq \sigma_i'$,但他们早经过一定次数的哈希操作后得到了相同的公钥块 pk_i,即 $f^{255-setp_i}(\sigma_i)=f^{255-setp_i'}(\sigma_i')$,这相当于破解了哈希函数的二次预映射抗性,因为攻击者找到了两个不同的输入 σ_i 和 σ_i',并且 σ_i 和 σ_i' 经相同的哈希函数运算得到了相同的输出 pk_i.

攻击者成功伪造签名的概率可以通过哈希函数的抗第一原像攻击和抗第二原像攻击特性的安全性计算:

$$P_{success} \le \varepsilon_{\text{pre-image}} + \varepsilon_{\text{second-pre-image}}.$$

其中 $\varepsilon_{\text{pre-image}}$ 为哈希函数抗第一原像攻击的失败概率, $\varepsilon_{\text{second-pre-image}}$ 为哈希函数抗第二原像攻击的失败概率.若攻击者 ADV 对签名方案所使用的哈希函数的攻击的成功概率 $P_{success}$ 可忽略,则说明 SM3-OTS 方案所使用的哈希函数是一个安全的哈希函数,即说明 SM3-OTS 方案是安全的数字签名方案.

SM3-OTS 方案的核心基于哈希函数构建,而哈希函数被证明在面对量子计算具备较强的抗攻击能力.与基于数论的签名方案(如 RSA 和 ECC)不同,这些数论方案可以被量子计算机应用 Shor 算法在多项式时间内破解,但哈希函数的抗性只会受到 Grover 算法的影响.Grover 算法是一种量子搜索算法,它能够以平方根速度加快搜索过程,但不能完全破解哈希函数.量子计算机通过 Grover 算法能够将哈希函数的安全性降低一半.举例来说,国密 SM3 算法能够提供 128 位的经典安全性,但在量子计算机上,这种安全性会降低到约 85 位.这意味着量子计算机在破解哈希函数时可以减少运算量,但并不能彻底破坏哈希函数的安全性.虽然量子计算机虽然可以通过 Grover 算法降低国密 SM3 算法的部分安全性,但可以通过增长 SM3 算法的输出长度来弥补,所以 SM3-OTS 在量子计算时代依然安全可靠.

6 密钥与签名尺寸

本节将阐述 SM3-OTS 签名方案的密钥尺寸和生成的签名值尺寸,并与其它主流一次数字签名方案密钥及生成签名值尺寸对比.

6.1 密钥和签名尺寸

SM3-OTS 签名方案的私钥为对秘密种子 $Seed$ 和序号 i 使用哈希函数迭代运算生成,而公钥则是使用哈希函数对相应私钥作哈希运算生成的.SM3-OTS 方案私钥尺寸为: $sk\_size = i \times 32 = 48 \times 32 = 1536$ (Bytes).

同理,SM3-OTS 方案公钥尺寸为: $pk\_size = i \times 32 = 48 \times 32 = 1536$ (Bytes).

SM3-OTS 签名方案的签名块为对应哈希链上的链节点.故签名块的数量对应哈希链的数量,即为对应签名方案私钥块的数量,SM3-OTS 方案签名尺寸为: $\sigma\_size = i \times 32 = 48 \times 32 = 1536$ (Bytes).

SM3-OTS 是基于哈希函数设计的一种后量子一次数字签名方案,具有更小的密钥及签名尺寸以及更强的安全性.相较于 SPHINCS$^+$中使用的 WOTS+方案、SPHINCS-α 中使用的 Balanced WOTS+方案以及 SPHINCS$^+$C 中使用的 WOTS+C 方案这几种主流的一次数字签名方案所生成的签名值长度大约缩短了 29%、27%、26%,签名性能得到明显提升.这种减少不仅有助于降低存储和带宽的需求,还提高了签名和验证的效率.因此,SM3-OTS 在需要高效签名和验证的应用场景中展现出了巨大的潜力.表 5 为 SM3-OTS 与其它几种哈希

基一次签名方案的参数尺寸对比.

表 5 几种签名方案参数对比

签名方案	安全参数(Bytes)	密钥尺寸(Bytes)	签名值尺寸(Bytes)
WOTS	32	2144	2144
WOTS+	32	2144	2144
Balanced WOTS+	32	2112	2112
WOTS+C	32	2080	2080
LMOTS	32	2144	2144
SM3-OTS	32	1536	1536

6.2 SM3-OTS效率分析

本文将 SM3-OTS 与 WOTS、WOTS+、Balanced WOTS+、WOTS+C 等方案进行对比.本次测试在配备 4GB RAM 的 AMD Ryzen 7840S CPU(3.3GHz)的硬件设备上运行,测试环境为 Ubuntu 22.04 LTS.图 5-图 7 分别为签名方案密钥生成、签名方案签名生成、签名方案签名验证所需时间.由实验看出,SM3-OTS 数字签名方案的运行时间表现良好,与 NIST 标准化算法 SPHINCS$^+$中使用的 WOTS+方案相比,SM3-OTS 在密钥生成、签名生成、签名验证所需的时间分别减少了 27.2%、18.7%、25.3%, 与 WOTS、SPHINCS-α 中使用的 Balanced WOTS+以及 SPHINCS+C 中使用的 WOTS+C 运行效率相差较小,结合表 5 中几种签名方案生成的签名值大小来看,SM3-OTS 方案具有非常良好的应用前景.

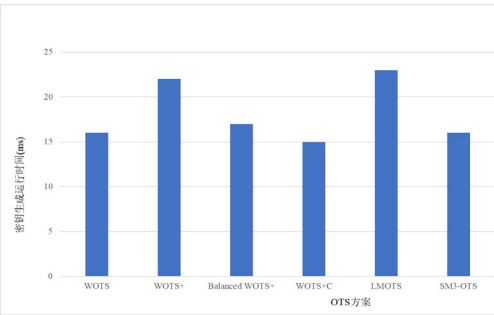

图 5 几种签名方案密钥生成时间对比

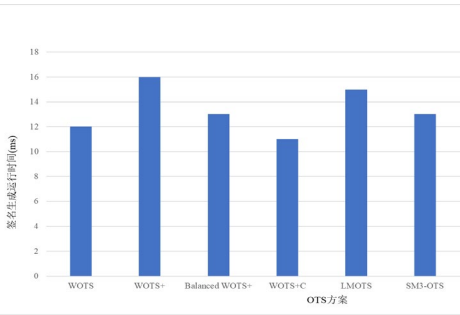

图 6 几种签名方案签名生成时间对比

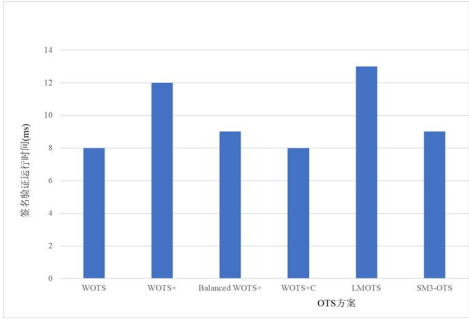

图 7 几种签名方案签名验证时间对比

7 总 结

针对现有基于哈希函数的一次数字签名方案存在生成的签名值长度过长问题,本文提出了一种新的基于哈希函数的一次数字签名方案 SM3-OTS,SM3-OTS 签名方案提供了比 WOTS 及 WOTS+方案更高的安全性,同时 SM3-OTS 方案的密钥生成时间、签名生成时间、签名验证时间相较于上述两方案更短,SM3-OTS 签名方案的密钥对及签名值长度为 1536 字节,极大的缩短了一般基于哈希函数的一次签名方案的密钥长度及签名值长度.因此,SM3-OTS 提供了更优秀的性能及更高的安全性.未来可通过扩展二叉哈希树的形式将 SM3-OTS 扩展为无状态数字签名方案,以提高基于哈希函数的后量子数字签名方案的安全性和签名性能.

References:

[1] O'brien J L. Optical quantum computing[J]. Science, 2007, 318(5856): 1567-1570.DOI:10.1126/science.1142892.

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

[3] CUI Fx,WANG B,LIU Y,et al. Research Status and Prospects on Quantum Attacks on Public Key Ciphers [J]. Cybersecurity and data governance,2022,41(09):3-12. DOI:10.19358/j.issn.2097-1788.2022.03.001.

[4] Monz T, Nigg D, Martinez E A, et al. Realization of a scalable Shor algorithm[J]. Science, 2016, 351(6277):1068-1070.DOI:10.1126/science.aad9480.

[5] Long G L. Grover algorithm with zero theoretical failure rate[J]. Physical Review A, 2001, 64(2): 022307, 1-4. DOI: https://doi.org/10.1103/PhysRevA.64.022307.

[6] Bernstein D J, Lange T. Post-quantum cryptography[J]. Nature, 2017, 549(7671): 188-194. DOI: https://doi.org/10.1038/nature23461.

[7] Bos J, Ducas L, Kiltz E, et al. CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM[C]. In: 2018 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2018: 353-367. DOI:10.1109/EuroSP.2018.00032.

[8] Ducas L, Kiltz E, Lepoint T, et al. Crystals-dilithium: A lattice-based digital signature scheme[J]. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2018: 238-268. DOI:https://doi.org/10.13154/tches.v2018.i1.238-268.

[9] Soni D, Basu K, Nabeel M, et al. Falcon[J]. Hardware Architectures for Post-Quantum Digital Signature Schemes, 2021: 31-41.

[10] Bernstein D J, Hülsing A, Kölbl S, et al. The SPHINCS+ signature framework[C]. In: 2019 ACM SIGSAC conference on computer and communications security (CCS19). ACM, 2019: 2129-2146. DOI:https://doi.org/10.1145/3319535.3363229.

[11] Yang YT, Zhao R Y, Chang X, et al. PQVPN: Software VPN design that is resistant to quantum computing attacks[J]. Chinese Journal of Information Security, 2022,7(05):108-119. DOI:10.19363/J.cnki.cn10-1380/tn.2022.09.09.

[12] NIST. Recommendation for Stateful Hash-Based Signature Schemes[S]. NIST SP 800-208, 2020.

[13] YANG YT, CHANG X, SHI HP, et al.CDBS:Blind signature scheme based on CRYSTALS-Dilithium algorithm[J].Journal on Communications,2024,45(07):184-195.DOI:10.11959/j.issn.1000-436x.2024129.

[14] HÜLSING A, BUTIN D, GAZDAG S-L, et al. XMSS: eXtended Merkle Signature Scheme[S]. RFC 8391, 2018.

[15] Zhang K, Cui H, Yu Y. SPHINCS-α: A Compact Stateless Hash-Based Signature Scheme[J]. Cryptology ePrint Archive, 2022.

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

[17] SUHAIL S,HUSSAIN R,KHAN A,et al.On the role of hash-based signatures in quantum-safe Internet of Things: Current solutions and future directions[J]. IEEE Internet of Things Journal,2021,8(1):1-17. DOI: 10.1109/JIOT.2020.3013019.

[18] KUMAR A,OTTAVIANI C,GILL S S,et al.Securing the future Internet of Things with post-quantum cryptography[J]. Security and Privacy,2022,5(2):1-11. DOI:https://doi.org/10.1002/spy2.200.

[19] Lamport L. Constructing digital signatures from a one way function[R]. Technical Report CSL-98, October1979.

[20] Merkle R C. Secrecy, authentication, and public key systems[D]. Stanford university, 1979.DOI:https://doi.org/10.1007/0-387-34805-0_21.

[21] Merkle R C. A certified digital signature[C]. In: International Conference on the Theory and Application of Cryptology (CRYPTO 1989).LNCS, volume 435. New York, NY: Springer New York, 1989: 218-238.DOI: https://doi.org/10.1007/0-387-34805-0_21.

[22] Niaz M S, Saake G. Merkle hash tree based techniques for data integrity of outsourced data[C]. In: 27th GI-Workshop on Foundations of Databases (Grundlagen von Datenbanken), Magdeburg, Germany, 2015: 66-71.

[23] SUN S W, LIU T Y, GUAN Z, et al.SPHINCS$^+$-SM3: SM3-based stateless digital signature scheme[J]. Journal of Cryptologic Research, 2023,10(6): 1266–1278.DOI:10.13868/j.cnki.jcr.000658.

[24] SUN S W, LIU T Y, et al. YAN H L. SM3-based post-quantum digital signature schemes[J]. Journal of Cryptologic Research, 2023, 10(1): 46–60.

[25] Li L, Lu X, Wang K. Hash-based signature revisited[J]. Cybersecurity, 2022, 5(1): 1-26.DOI: https://doi.org/10.1186/s42400-022-00117-w.

[26] Wang J, Zhang T, Sebe N, et al. A survey on learning to hash[J]. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(4): 769-790.DOI:10.13868/j.cnki.jcr.000578.

附中文参考文献:

[3] 崔富鑫,王辈,刘焱,等.公钥密码的量子攻击研究现状与展望[J].网络安全与数据治理,2022,41(09):3-12.DOI:10.19358/j.issn.2097-1788.2022.03.001.

[11] 杨亚涛,赵若岩,常鑫,等.PQVPN:抗量子计算攻击的软件 VPN 设计[J].信息安全学报,2022,7(05):108-119.DOI:10.19363/J.cnki.cn10-1380/tn.2022.09.09.

[13] 杨亚涛,常鑫,史浩鹏,等.CDBS：基于 CRYSTALS-Dilithium 算法的盲签名方案[J].通信学报,2024,45(07):184-195.DOI:10.11959/j.issn.1000-436x.2024129.

[23] 孙思维,刘田雨,关志,等.SPHINCS+-SM3:基于 SM3 的无状态数字签名算法[J].密码学报,2023,10(06):1266-1278.DOI:10.13868/j.cnki.jcr.000658.

[24] 孙思维,刘田雨,关志,等.基于杂凑函数 SM3 的后量子数字签名[J].密码学报,2023,10(01):46-60.DOI:10.13868/j.cnki.jcr.000578.