PQVPN：抗量子计算攻击的软件 VPN 设计

杨亚涛[1,2], 赵若岩[1], 常 鑫[2], 郭 超[1], 肖 嵩[1,2]

[1]北京电子科技学院电子与通信工程系, 北京 中国 100070
[2]西安电子科技大学通信工程学院, 西安 中国 710071

摘要 随着量子破译算法的不断优化和量子计算机硬件技术的快速发展, 目前传统密码算法面临越来越大的安全风险, 这使得抗量子计算成为研究热点, 目前用传统密码体制构建的 VPN, 越来越受到量子计算攻击的威胁。为了解决传统 VPN 中在身份验证和密钥协商环节不能抵抗量子计算攻击的问题, 本文基于 Microsoft PQCrypto-VPN 项目的框架, 依赖于 OpenSSL 的 Open Quantum Safe 项目分支, 设计了一套抗量子计算攻击的软件 VPN 系统。对比进入 NIST 第三轮筛选的后量子数字签名和密钥协商算法, 通过综合考量运算性能和安全性能, 系统采用后量子签名算法 Picnic 和密钥协商算法 CRYSTALS-KYBER, 以实现 VPN 通信中数据的抗量子计算攻击安全保护。同时, 本文对所使用的上述两种后量子算法进行了安全性分析, 以阐述本系统的抗量子安全性能, 并对系统进行了性能测试。在测试的带宽条件下, VPN 连接后最高上传速度可达 206Kb/s, 下载速度可达 2495Kb/s, 与通过公网直接传输和通过传统 OpenVPN 传输两种情形下的传输速度相近; 在通信延迟方面, 相比目前提出的三种后量子 VPN 系统均有明显降低, 在牺牲少量带宽的情况下实现了对数据通信的更高安全保障。

关键词 抗量子计算攻击; Picnic; CRYSTALS-KYBER; SSL VPN; OpenSSL; OpenVPN
中图法分类号 TP309.7 **DOI 号** 10.19363/J.cnki.cn10-1380/tn.2022.09.09

PQVPN: Design of Software VPN against Quantum Computing Attack

YANG Yatao[1,2], ZHAO Ruoyan[1], CHANG Xin[2], GUO Chao[1], XIAO Song[1,2]

[1]Department of Electronic and Communication Engineering, Beijing Electronics Science and Technology Institute, Beijing 100070, China
[2]School of Telecommunication Engineering, Xidian University, Xi'an 710071, China

Abstract With the continuous optimization of quantum decoding algorithm and the rapid development of quantum computer hardware technology, traditional cryptography algorithms are confronting more and more security risks, which makes post quantum computing becoming one of research hotspots. At present, Virtual Private Network (VPN) with traditional cryptographic mechanism is facing a growing security threat by quantum computing attacks in authentication and key exchange. In order to solve the issue of quantum computing attack in authentication and key exchange in traditional VPN, A software VPN system against quantum computing attacks (PQVPN) is designed in this paper based on the framework of Microsoft PQcrypto-VPN project and relied on the open quantum safe project branch of OpenSSL. The post quantum digital signature and key exchange algorithms that have been selected as the third-round candidates by National Institute of Standards and Technology (NIST) are compared in this paper with comprehensive consideration on the working performance and security of these algorithms. Picnic, a post quantum signature algorithm, and CRYSTALS-KYBER, a key agreement algorithm, are used in this system to achieve the post quantum security protection for communication data in VPN tunnel. Moreover, the security of these two post quantum algorithms is analyzed in this paper, the post quantum security in this PQVPN system is also illustrated. In addition, the working performance of this PQVPN system in the public network environment is tested. The test shows that the maximum of upload speed and download speed after VPN connection can reach 206Kb/s and 2495Kb/s under the experimental bandwidth environment, which is similar to the transmission speed under the direct transmission through public network and transmission through traditional OpenVPN. Compared with three proposed post quantum VPN systems, the communication delay is significantly reduced, higher security in data communication can be realized with a small amount of bandwidth expense in this PQVPN system.

Key words resist quantum computing attacks; picnic; CRYSTALS-KYBER; SSL VPN; OpenSSL; OpenVPN

通讯作者： 杨亚涛, 男, 1978 年生, 博士, 教授, 博导, 主要研究方向为密码学与通信安全, Email: yy2008@163.com。
本课题得到“十四五”国家密码发展基金; “通信工程”、“电子信息工程”国家级一流本科专业建设点项目资助。
收稿日期: 2021-08-28; 修改日期: 2021-11-22; 定稿日期: 2022-07-14

1 前言

抗量子密码又称后量子密码(Post Quantum Cryptography)。2015 年 8 月, 美国国家安全局公开提出美国国家安全系统目前使用的NSA Suite B将逐步向后量子密码算法升级。2016 年, 美国国家标准与技术研究院(National Institute of Standards and Technology, NIST)向全球范围内征集后量子公钥密码算法标准, 共有 25 个国家和地区的密码学家参加了此次征集活动。截至 2018 年, 共收到 82 份方案。2019 年初, NIST 宣布 26 个方案进入第二轮评估。2020 年 7 月, NIST 公布了第三轮的 7 种候选算法和 8 种备选算法。如表 1 和表 2 所示。

表 1 第三轮候选算法

Table 1 Candidate algorithms for the third round

公钥加密算法	数字签名算法
Classic McEliece	CRYSTALS-DILITHIUM
CRYSTALS-KYBER	FALCON
NTRU	Rainbow
SABER	/

表 2 第三轮备选算法

Table 2 Alternative algorithms for the third round

公钥加密算法	数字签名算法
BIKE	GeMSS
FrodoKEM	Picnic
HQC	SPHINCS+
NTRU Prime	/
SIKE	/

Open Quantum Safe 项目是美国一个研究后量子密码学的开源项目。该项目的主要工作路线为 liboqs 算法库的开发并将后量子算法集成到协议和应用程序之中, 其中包括广泛使用的 OpenSSL 库。

2019 年, 微软研究员 Douglas Stebila 等人研究了把后量子密码算法融合到 TLS(Transport Layer Security)协议和 SSL(Secure Sockets Layer)协议中的几种实现方案[1]。同年微软研究员 Christian Paquin 开发了包含后量子密码的 TLS 框架, 并测试了几种后量子密码算法对 TLS 连接建立的性能影响[2]。

2020 年, Sebastian Paul 等人针对工业网络物理系统(Cyber-Physical Systems, CPS)的后量子安全问题提出了将后量子密码集成到工业协议开放平台统一架构中, 并进行了方案实现和性能评估[3]。Loïs Huguenin-Dumittan 等人则对进入 NIST 第二轮竞赛中的几种算法提出了明文检查攻击(PCA, Plaintext Check Attack)和针对基于秩的 RQC 的 KR-PCA 攻击[4]。同年, 在研究硬件平台方面, Lukas Malina 等人评估了NIST后量子密码候选算法在特殊硬件平台上实施的适用性, 主要关注了在受限平台和快速硬件加速平台上的性能[5]。Brian Koziel 等人则研究了 NIST 第三轮中密码算法的关键运算如何在硬件中实施有效加速[6]。

2021 年, 潘彦斌等人提出两种纯密文攻击思路, 并证明在推荐参数下, NIST 竞赛中的候选算法 Compact-LWE 是不安全的[7]。Manohar Raavi 等人则针对入围的公钥签名算法比较了其安全性和相应性能, 提供了后量子密码设计的安全比较规则、性能分析以及综合分析参考标准[8]。Ward Beullens 针对 UOV 和 Rainbow 签名方案提出了适用于 UOV 和 Rainbow 的相交攻击以及仅适用于 Rainbow 的矩形 MinRank 攻击, 证明了这两种签名方案存在安全缺陷[9]。Thomas Prest 等人则对后量子密码学的知识状态系统化, 将经典和后量子密码方案转化为若干个范式[10]。杨亚涛等人在文献[11]和[12]中分别提出了一种基于 RLWE 困难问题的后量子攻击密钥协商协议和双向认证密钥协商方案 INAKA。李子臣等人基于格理论环上误差学习(Ring Learning with Errors, RLWE)问题使用 Peikert 式误差协调机制构造了一个 C/S 模式下的口令认证密钥交换协议(PAKE), 并使用 Java 在 Eclipse 平台上进行了此协议的模拟实现[13]。

虚拟专用网(Virtual Private Network, VPN)于 1990 年代中期首次创建和部署, 旨在远程节点之间建立一个加密的网络隧道, 允许应用程序流量通过该隧道跨越不安全的中介网络传输[14]。

量子计算将在未来成为现实。文献[15]中指出, 目前主流 SSL VPN 的安全性依赖于 SSL 协议中身份认证算法、密钥协商算法、信息加密算法的复杂度。随着量子计算的出现, 现在所使用的传统密码算法正面临被破解的风险。此外, Sultan Almuhammadi 等人揭示了现有 VPN 系统正面临 Web 指纹攻击的威胁[16]。

2015 年, Aymen Ghilen 等人建议在隧道两端的 OpenVPN 部署用于密钥交换和身份验证的量子协议, 以实现 VPN 隧道通信安全性能的提升[17]。

2018 年, 国盾量子与国网电力信息通信提出了一种通信方案[18], 该方案针对 IPSec VPN, 在实际电力通信网络中进行了测试, 测试结果表明, 该方法能够满足密文通信的需求, 也为公共网络环境下的

抗量子 VPN 的研究提供了参考方向。

2020 年, Quentin M. Kniep 等人提出了对 VPN 软件 WireGuard 握手协议的三个改进方式以增强其后量子安全性能, 并验证这些改进对系统性能的影响不大[19]。

2021 年, Joo Yeon Cho 和 Andrew Sergeev 对 VXLAN 上的 MAC Sec 协议进行改进, 使用了后量子的临时密钥交换协议和端到端身份验证方案, 并验证了对延迟和吞吐量的影响是最小的[20]。改进后进行的实验证实, 量子安全虚拟化链接已经可以远距离建立, 而无需更改其基础设施。

本文贡献如下:

(1) 实现了一种抗量子计算攻击的 VPN 设计框架。基于现有 SSL VPN 框架, 对 OpenSSL 库及内部通信协议进行改造, 在 VPN 通信连接过程中的身份认证和密钥协商环节使用了后量子签名算法 Picnic 和密钥协商算法 CRYSTALS-KYBER, 实现了抗量子计算安全特性。实现了公网环境下的安全隧道通信及内网访问, 能满足现阶段和量子时代的通信安全需求, 也为今后实现后量子安全通信提出了架构案例。

(2) 测试了 PQVPN 系统证书生成及验证、VPN 建立连接及连接后的隧道安全通信功能, 系统完整建立连接消耗时间为 4.93s, VPN 连接后最高上传速度可达 206Kb/s, 下载速度可达 2495Kb/s, 相同带宽下与公网正常传输速度几乎相近, 与传统 VPN 性能相差较小, 与其他后量子 VPN 系统相比性能占优, 实现了较高性能的 PQVPN 系统。

2 关键算法

2.1 Picnic 算法

Picnic 算法使用 Fiat-shamir 变换与 Unruh 变换创建了非交互式零知识证明的签名系统。为了实现签名, 签名者创建一个非交互的知识证明, 并将该证明与待签名消息绑定[21-22]。

签名过程:

输入: 签名者的密钥对 (sk, pk), 待签名的消息字节组为 M, $1 \leqslant |M| \leqslant 2^{55}$。

输出: M 的签名以字节组形式输出。

(1) 初始化以下值。$C[0\cdots T-1][0\cdots N-1]$、$Ch[0\cdots T-1]$ 和 $Cv[0\cdots T-1]$, 其中每个值的长度为 l_H 字节。$maked\_key[0\cdots T-1]$ 的输入队列 T, 每个长度为 n 位。随机队列 $tapes[0\cdots T-1][0\cdots N-1]$, 每次并行迭代, 每个长度为 $6rs+n$ 位。辅助信息比特串队列 $aux[0\cdots T-1]$, 每个长度为 $3rs$ 比特。拓展消息队列 $msgs[0\cdots T-1][0\cdots N-1]$, 长度为 $n+3rs$ 的比特串。

(2) 生成长度为 S 位的根 *seed* 和长度为 256 位的 *salt*。用 KDF 方法推导:

$$sk \| M \| pk \| S$$

其中, S 编码方式为 16 位小端整数, 要求字节长度为 $2(S/8)$(一个 *salt* 和一个 *seed*, 每个长度为 S 位)。

(3) 使用种子将根 *seed* 和 *salt* 扩展成 T 个初始 *seed*, 表示为 $iSeed[0\cdots T-1]$。

(4) 从 0 到 *T–1* 的 t 重复以下操作:

(a) 使用 $iSeed[t]$、*salt* 和整数 t, 用种子树方法, 导出 N 个种子。这些种子表示为 $seeds[0\cdots T-1]$。

(b) 使用 KDF 导出 N 个 $tapes[t][N-1]$:

$$tapes[t][i] = KDF(seeds[i] \| salt \| t \| i)$$

每个位数至少为 $6rs$。输入的 t 和 i 被编码为 16 位小端整数。

(c) 运行 Compute_aux 算法, 返回字符串 *mpcInputs*, 将在下面使用。

(d) 计算 N 个 $C[t][0\cdots N-1]$ 如下:

$$C[t][i] = H(seeds[i] \| salt \| t \| i)$$

其中, i 从 0 到 *N*–2 且

$$C[t][N-1] = H(seeds[N-1] \| aux[t] \| salt \| t \| i)$$

(e) 创建并存储私钥。

(f) 输入 $maskedKey[t]$、$tapes[t]$ 和 pk, 运行 mpc_simulate 算法, 输出为 $msgs[t][0\cdots N-1]$。

(g) 计算 $Ch[t]$ 如下:

$$Ch[t] = H(C[t][0] \| .. \| C[t][N-1])$$

(h) 计算 $Cv[t]$ 如下:

$$Cv[t] = H(maskedKey[t] \| msgs[t][0] \| .. \| msgs[t][N-1])$$

(5) 创建 T 列表的 Merkle 树, $Cv[0\cdots T-1]$ 作为叶, 让 *Cv_root* 作为根节点。

(6) 使用函数 HCP 进行计算, 输出为一个摘要 h 和两个长度为 u 的 16 位整数列表 *LC* 和 *LP*。

$$(h, LC, LP) = HCP(Ch[0], ..., Ch[T-1], Cv\_root, salt, pk, M)$$

LC 中的整数是唯一的且在 $[0, T-1]$ 范围内, *LP* 中的值在 $[0, N-1]$ 范围内。

(7) 计算 Merkle 树 *Cv* 公开信息, 公开信息表示为 *cvInfo*。

(8) 计算初始种子所需的信息 $t \notin LC$，此信息表示为 *iSeedInfo* 。

(9) 收集签名。签名是 $(h, salt, iSeedInfo, cvInfo, Z)$，其中 Z 是 5 元 u 组的列表。定义 $(t_i, P_i) = (LC[i], LP[i])$ 其中 i 从 0 到 u–1。

(10) 序列化 $(h, salt, iSeedInfo, cvInfo, Z)$，并将其作为签名输出。

2.2 CRYSTALS-KYBER 算法

Kyber 是由 Roberto Avanzi 等人设计的一种满足 IND-CCA2 安全的密钥封装机制，其安全性依赖于解决格上的模误差学习(Module Learning with Errors, MLWE)问题的难度。参数 $n, k, q, \eta_1, \eta_2, d_u$ 和 d_v 中 n 为 256, q 是 3329。表 3、表 4 和表 5 为 Kyber.CPAPKE 公钥加密方案中密钥生成、加密和解密的定义。

表 3 CRYSTALS-KYBER 算法的密钥生成过程

Table 3 Key generation in CRYSTALS-KYBER algorithm

Output: Secret key $sk \in B^{12 \cdot k \cdot n/8}$ 、Public key $pk \in B^{12 \cdot k \cdot n/8+32}$

$d \leftarrow B^{32}$

$(\rho, \sigma) := G(d)$

$N := 0$

for i from 0 to k-1 do

  for j from 0 to k-1 do

    $A[i][j] := Prase(XOF(\rho, j, i))$

  end for

end for

for i from 0 to k-1 do

  $s[i] := CBD_{\eta_1}(PRF(\sigma, N))$

  $N := N + 1$

end for

for i from 0 to k–1 do

  $e_{[i]} := CBD_{\eta_1}(PRF(\sigma, N))$

  $N := N + 1$

end for

$\hat{s} := NTT(s)$

$\hat{e} := NTT(e)$

$\hat{t} := \hat{A} \cdot \hat{s} + \hat{e}$

$pk := (Encode_{12}(\hat{t} mod^+ q) || \rho)$

$sk := Encode_{12}(\hat{s} mod^+ q)$

return (pk, sk)

表 4 CRYSTALS-KYBER 算法的加密过程

Table 4 Encryption in CRYSTALS-KYBER algorithm

Input: Public key $pk \in B^{12 \cdot k \cdot n/8+32}$ 、Message $m \in B^{32}$ 、Random coins $r \in B^{32}$ Output: Ciphertext $c \in B^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$

$N := 0$

$\hat{t} := Decode_{12}(pk)$

$\rho := pk + 12 \cdot k \cdot n/8$

for i from 0 to k-1 do

  for j from 0 to k-1 do

    $A^T[i][j] := Prase(XOF(\rho, i, j))$

  end for

end for

for i from 0 to k-1 do

  $r[i] := CBD_{\eta_1}(PRF(r, N))$

  $N := N + 1$

end for

for i from 0 to k–1 do

  $e_1[i] := CBD_{\eta_2}(PRF(r, N))$

  $N := N + 1$

end for

$e_2[i] := CBD_{\eta_2}(PRF(r, N))$

$\hat{r} := NTT(r)$

$u := NTT^{-1}(\hat{A}^T \circ \hat{r}) + e_1$

$v := NTT^{-1}(\hat{t}^T \circ \hat{r}) + e_2 + Decompress_q(Decode_1(m), 1)$

$c_1 := Encode_{d_u}(Compress_q(u, d_u))$

$c_2 := Encode_{d_v}(Compress_q(v, d_v))$

return $c = (c_1, c_2)$

图 1 描述了单向认证密钥交换协议 Kyber.UAKE, 单向认证密钥交换协议增加了静态密钥部分，其中设定 Alice 知道 Bob 的静态密钥。

其协商过程为:

设 $H: \{0,1\}^* \rightarrow \{0,1\}^{256}$ 为哈希函数。

第一步: Alice 通过 Kyber 的密钥生成算法生成临时公私钥 (pk, sk)，使用 Bob 的静态公钥 pk_2 对预设密钥 K_2 用 Kyber 加密算法加密后得到密文 c_2，之

表 5 CRYSTALS-KYBER 算法解密过程

Table 5 Decryption in CRYSTALS-KYBER algorithm

Input: Secret key $sk \in B^{12\cdot k\cdot n/8}$、Ciphertext $c \in B^{d_u\cdot k\cdot n/8+d_v\cdot n/8}$

Output: Message $m \in B^{32}$

1: $u := Decompress_q\left(Decode_{d_u}(c), d_u\right)$

2: $v := Decompress_q\left(Decode_{d_v}(c + d_u \cdot k \cdot n/8), d_v\right)$

3: $\hat{s} := Decode_{12}(sk)$

4: $m := Encode_1\left(Compress_q\left(v - NTT^{-1}\left(\hat{s}^T \circ NTT(u)\right)\right), 1\right)$

5: return m

后将临时公钥 pk 和密文 c_2 发给 Bob。

第二步: Bob 获得 Alice 的临时公钥 pk 和密文 c_2 后, 先用临时公钥 pk 对预设密钥 K 加密后得到密文 c, 然后将密文 c 发给 Alice; 同时使用自己的静态私钥 sk_2 对密文 c_2 解密以获得预设密钥 K_2'。

第三步: Alice 收到 Bob 发来的密文 c 后, 用临时私钥 sk 解密密文 c 以获得预设密钥 K'。

第四步: Alice、Bob 分别对 K' 和 K_2、K 和 K_2' 进行 Hash 运算, 并将得到的 Hash 值作为会话密钥。

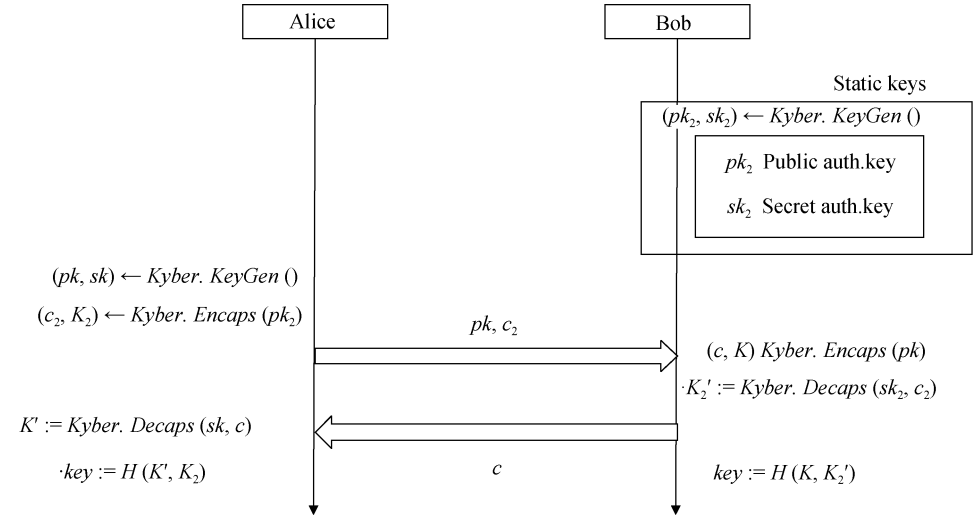

图 1 Kyber 的单向认证密钥交换协议

Figure 1 Kyber′s unidirectional authenticated key exchange protocol

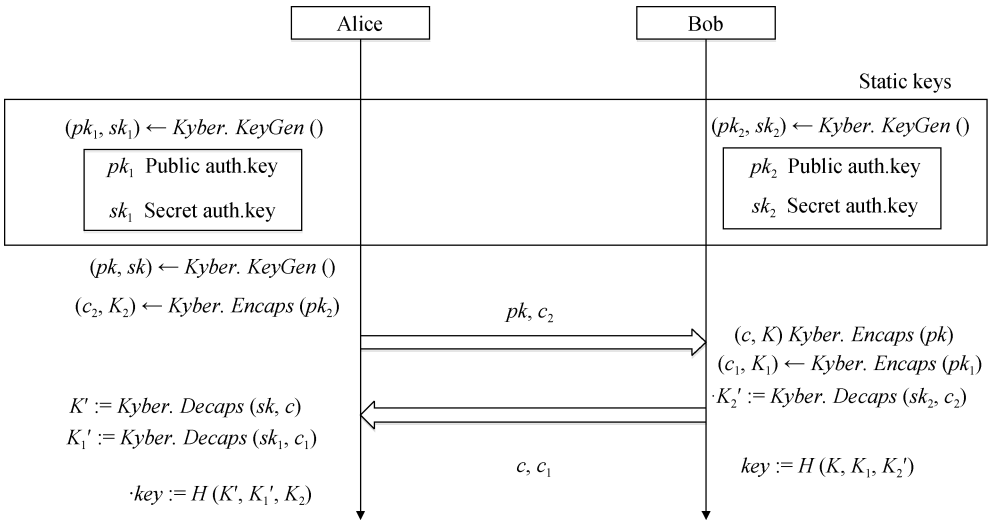

图 2 Kyber 的双向认证密钥交换协议

Figure 2 Kyber′s bi-directional authenticated key exchange protocol

图2描述了双向认证密钥交换协议Kyber.AKE, 其中设定双方都知道对方的静态密钥, 其协商过程为:

设 $H:\{0,1\}^* \to \{0,1\}^{256}$ 为哈希函数。

第一步: Alice 通过 Kyber 的密钥生成算法生成临时公私钥 (pk, sk), 使用 Bob 的静态公钥 pk_2 对预设密钥 K_2 用 Kyber 加密算法加密生成密文 c_2, 之后将公钥 pk 和密文 c_2 发给 Bob。

第二步: Bob 收到 Alice 发来的临时公钥 pk 和密

文 c_2 后, 先用公钥 pk 对预设密钥 K 加密生成密文 c, 再用 Alice 的静态公钥 pk_1 和 Kyber 加密算法对预设密钥 K_1 加密生成密文 c_1, 然后将密文 c 和 c_1 发给 Alice, 同时使用自己的静态私钥 sk_2 对密文 c_2 解密以获得预设密钥 K_2'。

第三步: Alice 收到来自 Bob 的密文 c 和 c_1 后, 用临时私钥 sk 和静态私钥 sk_1 分别解密密文 c 和 c_1, 可得到预设密钥 K' 和 K_1'。

第四步: Alice、Bob 分别对 K'、K_1' 和 K_2、K、K_1 和 K_2' 进行 Hash 运算, 并将该 Hash 值作为会话密钥。

上述协议最后导出的共享密钥不仅依赖临时密钥和密文 (pk,c), 而且还依赖静态密钥 $pk_i(i=2)$ 和相关的临时密文 $c_i(i=1,2)$。

3 系统设计

3.1 算法选择

NIST 公布了经筛选后进入第三轮评测的数字签名和密钥协商算法, 这些算法均经过了多方面的安全测试, 能够抵抗多种攻击, 且满足可证明安全性, 未来的抗量子密码算法标准大概率也将在其中产生, 因此考虑对这几种方案进行分析比较。在 NIST 看来, 这些密钥协商和数字签名算法是很有前途的具备抗量子攻击属性的潜在标准方案。

针对数字签名算法, 入选的有 CRYSTALS-DILITHIUM[23]、FALCON[24]、Rainbow[25], 其中 CRYSTALS-DILITHIUM、FALCON 均是基于格的后量子签名方案, Rainbow 是基于多变量的多项式 Oil-Vinegar 公钥签名方案的推广。这三种算法的公钥长度都比较长, CRYSTALS-DILITHIUM 的公钥长度在 864-3232Kb 之间, FALCON 的公钥长度在 157.8-1885.4Kb 之间, Rainbow 的公钥长度在 897-1793Kb 之间, 在实际应用中均高于 TLS 协议所能承载的长度, 因此, 我们考虑采用同样基于格理论困难问题且满足 TLS 协议包长度限制需求的 Picnic 算法, Picnic 算法也是进入 NIST 第三轮的候选算法之一。

针对密钥协商算法, 入选的有 Classic McEliece[26]、CRYSTALS-KYBER[27]、NTRU[28]、SABER[29] 四种。Classic McEliece 是一种基于 Goppa 纠错码的非对称密码算法; CRYSTALS-KYBER、NTRU 和 SABER 均为基于格困难问题的公钥密码体制中相较完善的密码体制, 区别在于 CRYSTALS-KYBER 是基于环上 LWE(Learning With Errors)问题, NTRU 基于大维数格中寻找最短向量的数学难题, SABER 是基于 MLWR(Module Learning With Rounding)问题而构造。在表 6 中我们对这些算法的性能进行了对比分析。

表 6 四种密钥协商算法的性能对比
(所有时间都以 CPU 周期表示)
Table 6 Comparison of working performance of four key exchange algorithms
(The computing time is presented in CPU cycles)

算法	密钥长度(bits)	密钥生成(cycles)	加密(cycles)	解密(cycles)
Classic McEliece	14120	316118712	175840	325624
CRYSTALS-KYBE	3168	307148	346648	396584
NTRU	1452	23302424	1256210	3642966
SABER	3040	205248	251248	271096

综上对比, 我们选择安全性较高、综合性能表现较好的 CRYSTALS-KYBER 算法作为 PQVPN 系统中的密钥协商算法。

3.2 PQCrypto-VPN 分析

PQCrypto-VPN 基于开源的 OpenSSL 库和传统 OpenVPN 的分支搭建而成。其设计基本思想为在现有 OpenVPN 的框架下, 修改其所使用的 OpenSSL 模块, 使其在 SSL 握手阶段中的证书签名和密钥协商部分调用抗量子 OpenSSL 分支中所支持的后量子签名算法和密钥协商算法, 而后量子算法的实现位于 liboqs 库中, 该库由所使用的 OpenSSL 分支调用。在当前版本中, 仅当流量通过客户端和服务器之间的 VPN 隧道时, 流量才受到量子计算机的保护。

OpenSSL 是一套由三个功能模块构成的 SSL 协议的开源实现: 密码算法库(Crypto Library)、SSL 协议框架以及相关工具。其中密码算法库和 SSL 协议框架是核心模块, 用于配合实现基于 SSL 协议的加密数据传输, 图 3 为 OpenSSL 整体结构。

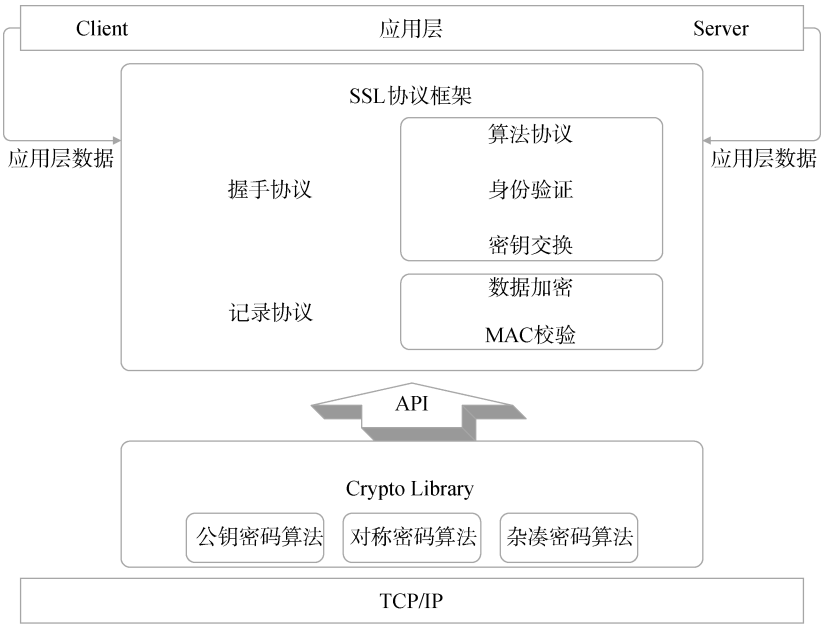

图 3　OpenSSL 整体结构

Figure 3　Overall structure of OpenSSL

密码算法库集成了 SSL 协议所必须的各种密码算法, 其主要包含以下三类算法:

(1) 公钥密码算法

OpenSSL 主要支持 4 种主流的传统公钥密码算法: Diffie-Hellman、RSA、DSA 以及 ECC, 这几种公钥密码算法主要用于实现 SSL 握手协议的密钥协商部分。

(2) 分组密码算法

OpenSSL 主要支持 7 种分组加密算法: AES、DES、Blowfish、CAST、IDEA、RC2、RC5, 同时也支持这几种算法的四种常用加密模式: ECB、CBC、CFB、OFB 模式。

(3) 杂凑密码算法

OpenSSL 的信息摘要计算中支持以下 5 种算法: MD2、MD5、MDC2、SHA(SHA1)和 RIPEMD。此外, OpenSSL 还实现了 DSS 标准中规定的两种信息摘要算法 DSS 和 DSS1。

OpenSSL 的信息传输功能实现主要依赖于 SSL 协议框架和密码算法库的支持。在运行中, OpenSSL 会调用密码算法库来完成待传输信息的加解密, SSL 协议框架会按照协议规定对信息执行封装或解包, 进而实现数据的安全传递。

本系统设计参考 2017 年 Microsoft 提出的 OpenSSL 的 Open Quantum Safe 项目分支。主要使用 OpenSSL 的 Crypto Library 密码算法库中基于 liboqs quantum-resistant 的算法和密码套件。

OpenVPN 工作于数据链路层和网络层, 允许点对点通信模式下的终端使用静态私钥、第三方证书或者用户名/口令来完成身份验证, 并允许虚拟接口用户执行访问控制策略或制定防火墙策略。为了保证用户数据传输的安全性和可靠性, OpenVPN 所有的数据加密与身份验证操作都通过 OpenSSL 库来处理。

3.3　抗量子攻击软件 VPN 系统设计方案

本系统的设计思路是在调用 Microsoft PQCrypto-VPN 项目框架的基础上, 借助其依赖的 OpenSSL 的 Open Quantum Safe 项目分支对系统原有的 OpenSSL 库进行扩充, 进而搭建 PKI(Public Key Infrastructure)系统, 并使用后量子签名算法 Picnic 为服务器和每个客户端颁发证书, 同时修改 VPN 启动配置文件实现 VPN 连接, 握手过程中密钥协商部分采用 CRYSTALS-KYBER 后量子密码算法, 在尽量不影响性能的基础上, 以实现该 PQVPN 系统的抗量子计算攻击安全性。

图 4 为抗量子计算的软件 VPN 实现框图, 为简化实现复杂度, 在测试中将 PKI 系统中的 CA(Certificate Authority)与服务器集成为一体。服务器和客户端分别生成证书请求文件, 然后向 CA 提出证书申请。CA 处理请求后分别向服务器和客户端颁发证书。每个客户端都可以和服务器端直接连接, 在完成隧道搭建前执行完身份认证、加密套件选择和密钥协商环节。与此同时, 客户端会接收到从服务端的地址池中分配来的虚拟网络地址信息, 所有地址都位于同一虚拟网段, 可实现虚拟局域网间的隧道通信。

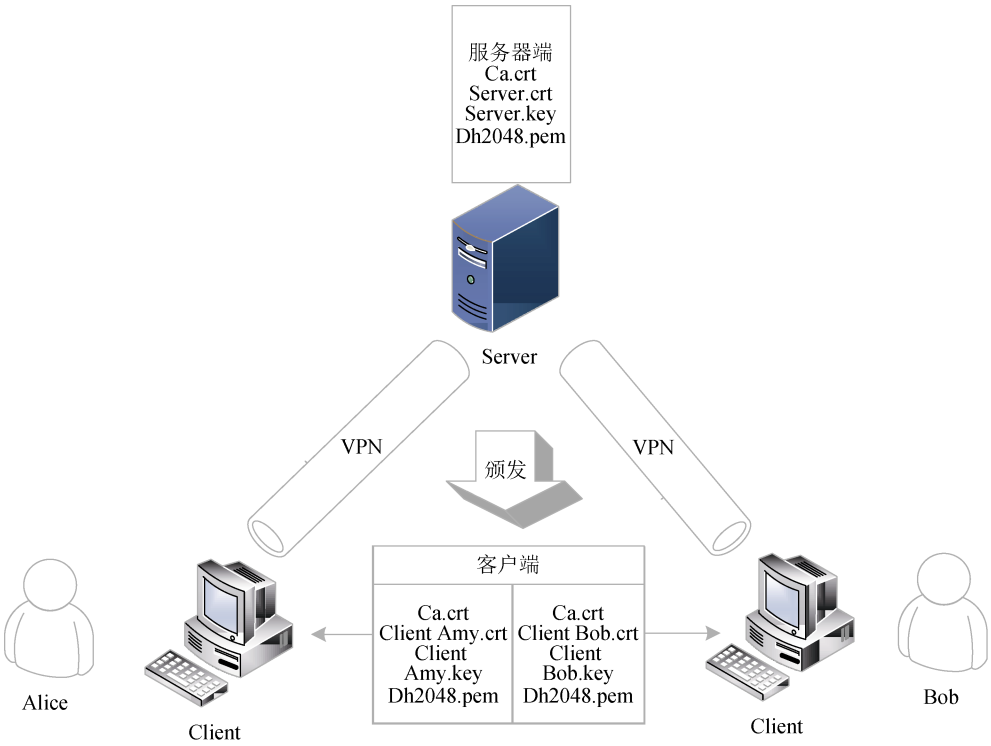

图 4　抗量子计算的软件 VPN 实现框架

Figure 4　VPN implementation framework of post quantum computing software

另外, 我们自主设计了客户端的图形界面, 集成了客户端私钥与证书申请文件的生成及连接功能。

4　运行结果与测试

本系统进行了局域网下不同网段之间的连接测试和公网下的连接测试, 局域网下的测试使用两台虚拟机来实现, 分别搭载 Ubuntu 18.04 系统, 公网下的测试使用阿里云远程服务器和 PC 机实现, 服务器采用 Ubuntu 18.04 Server 系统, PC 机为 Ubuntu 18.04 系统, 内存需要 8G 以上。

4.1　功能测试

(1) 证书验证测试

为了形成用于验证的证书链, 系统需要使用后量子签名算法 Picnic 来生成 CA 端的私钥和自签名证书, 也分别生成服务器端与客户端的私钥及证书。

如图 5 所示, 服务器端证书的签名算法为 Picnic, 签署方为 PQ-OpenVPN-Demo-CA, 证书署名为 PQ-OpenVPN-Demo-Server, 证书内包含公钥信息, 同时通过 CA 证书来证明服务器端证书的有效性。

如图 6 所示, 证书指纹的输入为证书内容, 使用 SHA-512 和 MD5 算法来确保证书的完整性; 密钥指纹的输入为公钥, 使用 SHA-512 算法确保内部公钥的完整性。

```
    Subject: CN = PQ-OpenVPN-Demo-Server
    Subject Public Key Info:
        Public Key Algorithm: picnicl1fs
            picnicl1fs Public-Key:
            pub:
                01:b5:c5:7c:33:07:5d:0c:b8:4f:f9:86:8a:40:1a:
                76:2f:1a:31:70:e1:5e:51:f6:13:d7:91:76:82:55:
                7a:06:ed
    X509v3 extensions:
        X509v3 Key Usage:
            Digital Signature
        X509v3 Extended Key Usage:
            TLS Web Server Authentication
Signature Algorithm: picnicl1fs
     96:58:68:22:81:50:01:45:90:86:92:68:a2:94:a4:8a:58:91:
     45:94:0a:64:04:26:11:65:08:41:28:a5:45:14:a0:25:16:59:
     04:59:80:69:82:48:84:54:4a:aa:50:51:89:18:15:80:45:59:
     98:ad:c9:91:d4:63:55:4f:c0:b6:65:e7:5e:b0:3d:d7:a5:5d:
     4f:96:5e:5e:92:c2:01:28:80:c2:fa:5d:07:5a:ae:7f:4c:90:
     08:71:1f:31:00:37:f0:31:89:26:1b:21:6f:87:33:c0:c3:37:
     b9:c6:76:cb:67:f5:18:2b:40:ca:f1:54:ad:14:1a:41:e4:d4:
```

图 5　Server 端证书验证

Figure 5　Server's certificate verification

(2) 服务器与客户端连接测试

服务器正常启动后, 可以在连接提示的信息流中获知系统生成的虚拟地址池及连接历史中客户端曾被分配的地址。当有客户端申请连接时, 双方会进入 SSL 握手协议过程, 在信息反馈中可以看到通过 SSL 握手协议双方进行密钥协商的过程, 如下所示:

第一部分: 建立安全能力。客户端发出 Client-hello 信息, 其中包括客户端可以支持的 SSL 协议最高版本号, 所使用的后量子密钥协商算法, 客户端可以支持的密码套件列表, 客户端可以支持的压缩

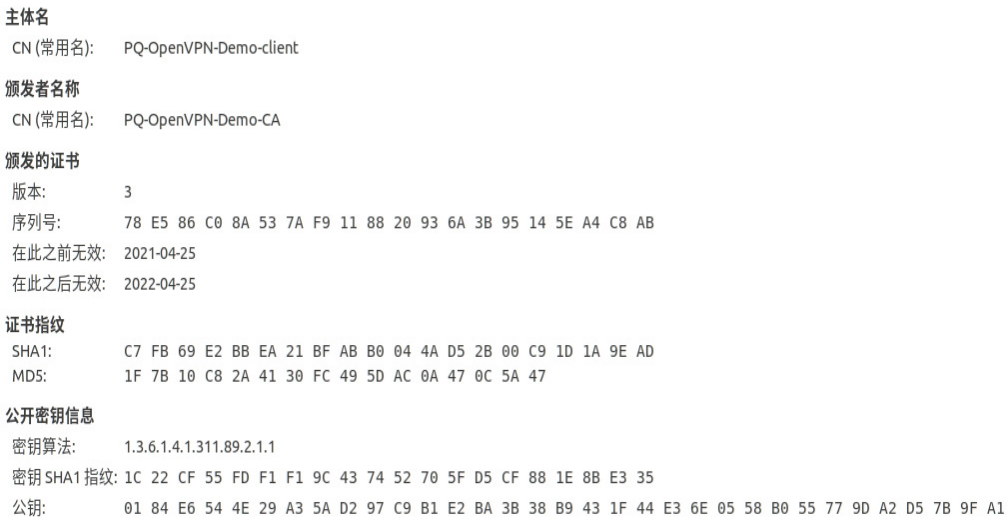

图 6 Client 端证书验证

Figure 6 Client's certificate verification

方法列表等。之后, 服务端发回 Server-hello 信息, 来确定 SSL 版本、密码套件和压缩方法。

第二部分: 服务端验证和客户端验证。服务端向客户端提供自身数字证书及完整证书链, 同时请求客户端的证书信息; 客户端验证服务端的身份信息后, 向服务端提供自身数字证书, 服务器端验证客户端身份, 进而完成彼此身份认证。

第三部分: 建立连接与客户端接受分配地址。当握手协商结束后, 客户端与服务端建立连接, 同时服务端在自身地址虚拟池内分配虚拟地址给客户端来完成通信建立。另外, 通过 Wireshark 对虚拟机进行抓包分析, 可以看到握手过程中 TLS 协议数据包、TCP 数据传输、OpenVPN 协议数据报。

(3) 客户端图形界面运行情况

用户填写申请证书所需的信息, 如用户、国家等, 即可自主选择生成目录、生成用户私钥.key 与证书请求文件.csr。之后, 用户可将文件上传至 CA 服务器, 待 CA 完成证书签署后颁发给用户。

(4) 客户端虚拟地址

当服务器端和客户端建立连接后, 通过查看客户端的 IP 信息, 可发现客户端的 tun0 端口开启, 端口的对应地址是由服务器从虚拟地址池内分配给客户端的 10.8.0.6, 从而说明通信连接及虚拟地址分配成功。

(5) ICMP 协议通信

客户端与服务器端进行 ICMP(Internet Control Message Protocol)通信, 命令中所用地址为 10.8.0.1, 是虚拟地址池内的首位地址, 其分配给服务器端, 公网上不可访问。通信时可以捕获到发送至服务器上的数据包, 来往的 ICMP 通信数据全由 OpenVPN 协议进行封装。

(6) FTP 文件传输

客户端和服务器端建立连接后即可进行通信, 可以利用 FTP(File Transfer Protocol)文件传输来测试其可行性。10.8.0.1 为服务器端虚拟地址池内的首位地址, 公网上不可访问; 客户端通过 FTP 访问此地址时, 可直接访问服务器上的 FTP 文件列表并支持文件的上传与下载。通过对数据流抓包分析, 发现通向服务器的数据流皆以 TCP 传输并且内容不可见, 说明该 PQVPN 可以实现数据的安全通信。

经测试, 证书生成及验证、VPN 连接建立及连接后的隧道安全通信等功能模块都能正常工作, 可以完成 VPN 系统下的数据安全通信。

4.2 性能测试

(1) 服务器与客户端连接

测试时间为客户端请求连接开始到握手过程结束后建立连接为止, 以 Wireshark 捕获数据包的时间差为准, 测试得到的连接时间为 4.93s。

(2) FTP 文件传输时间

在相同的网络环境下, 来传输不同类型、不同大小的文件, 测试时间从 FTP 连接开始到文件传输结束的整个过程。

性能对比结果如图 7 所示:

经测试, 排除网络流量波动的影响, 与正常的公网传输相比, PQVPN 系统牺牲了部分带宽, 但换来了直接点对点通信的优势, 同时增强了数据传输安全性。

表 7 公网正常传输

Table 7 Normal transmission in public network

文件类型	文件大小	上传时间(s)	上传速度(Kb/s)	下载速度(s)	下载速度(Kb/s)
文本文件(.docx)	2.94 Kb	1.11	13.79	0.27	12.64
文本文件(.docx)	27.8 Kb	0.49	97.65	0.49	63.34
图片文件(.png)	145 Kb	0.68	828.09	0.35	468.96
音频文件(.mp3)	4.5Mb	8.12	745.62	1.74	2581.24
视频文件(.mp4)	17.7 Mb	30.56	628.01	6.01	2899.12

表 8 VPN 隧道传输

Table 8 VPN tunnel transmission

文件类型	文件大小	上传时间(s)	上传速度(Kb/s)	下载速度(s)	下载速度(Kb/s)
文本文件(.docx)	2.94 Kb	1.85	6.48	0.55	6.84
文本文件(.docx)	27.8 Kb	2.17	79.14	1.19	51.25
图片文件(.png)	145 Kb	2.87	155.13	0.84	208.83
音频文件(.mp3)	4.5 Mb	28.8	188.33	4.28	1321.68
视频文件(.mp4)	17.7 Mb	87.76	205.56	8.01	2495.21

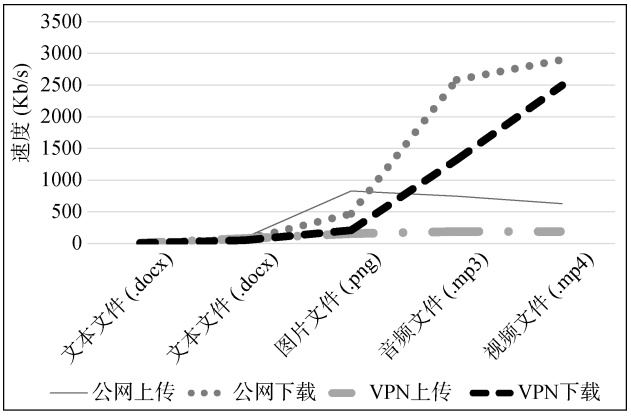

图 7 公网传输和 VPN 传输对比

Figure 7 Transmission comparison between public network and VPN

4.3 对比测试

在表 9 中, 本文选取目前已提出的三种抗量子 VPN 系统和两种常规 VPN 系统进行对比分析。文献[18]将量子安全密钥管理服务与传统 IPSec VPN 结合来实现抗量子安全隧道通信, 系统需要额外的硬件支持。文献[19]将 WG VPN 内部协议中的 KEM 环节进行后量子改造后来实现抗量子安全通信, 我们的 PQVPN 系统相比文献[19]在通信延迟方面减少了 79.3%。文献[20]对 MACsec 协议进行改进, 使用后量子临时密钥交换协议和端到端身份验证方案, 本文的 PQVPN 系统相比文献[20]在通信延迟方面减少了 71.4%。文献[30]为传统 IPSec VPN, 不具有抗量子计算攻击安全性。本文的 PQVPN 系统除了握手时长不具备优势外, 其最大吞吐量与传统 OpenVPN 性能相当, 最大并发连接数与其他 VPN 性能一致。通过综合对比分析可以看到, 本 PQVPN 系统具有较好的综合性能。

5 小结

本文介绍了抗量子计算攻击的软件 VPN 的设计

表 9 后量子 VPN 隧道通信性能对比

Table 9 Performance comparison of tunnel communication in post quantum VPN system

	VPN 类型	后量子算法/协议	通信延迟(ms)	最大吞吐量 Mb/s	握手时长	最大并发连接数
文献[18]	IPSec VPN	NewHope	—	—	—	100
文献[19]	WireGuard VPN	Kyber1024	100	—	213ms	—
文献[20]	VXLAN	PQ-MACsec	70	—	—	—
文献[30]	IPSec VPN	无	—	950	—	100
OpenVPN	SSL VPN	无	16	400	1.29s	100
本文	SSL VPN	Picnic&Kyber1024	20.7	389	3.83s	100

与实现过程。系统使用 Microsoft PQCrypto-VPN 项目框架, 依赖 OpenSSL 的 Open Quantum Safe 项目分支, 完成了基于后量子签名算法 Picnic 和密钥协商算法 CRYSTALS-KYBER 的抗量子计算攻击的软件 VPN 系统设计, 实现了 VPN 隧道通信中对数据安全的后量子安全保护。使用 QT 下的 C++实现了客户端的用户操作图形界面。

基于现有互联网协议和普通计算机终端, 本系统实现了公网环境下的抗量子计算攻击的安全隧道通信。本系统基于开源的 Open Quantum Safe 项目, 具有良好的二次拓展和可开发性。通过测试, 在 25M 带宽下, PQVPN 系统连接后最高上传速度可达 206Kb/s, 下载速度可达 2495Kb/s, 与公网正常传输和传统 OpenVPN 的传输速度几乎相近, 在通信延迟方面相比目前已提出的三种后量子 VPN 系统均有明显降低, 在牺牲少量带宽情况下实现了更安全的数据通信。

参考文献

[1] Crockett E, Paquin C, Stebila D. Prototyping Post-Quantum and Hybrid Key Exchange and Authentication in TLS and SSH[J]. *IACR Cryptol EPrint Arch* , 2019, 2019: 858.

[2] Paquin C., Stebila D., Tamvada G. Benchmarking Post-quantum Cryptography in TLS[C]. In: *11th International Conference on Post-Quantum Cryptography*, 2020, 72-91.

[3] Paul S, Scheible P. Towards Post-Quantum Security for Cyber-Physical Systems: Integrating PQC into Industrial M2M Communication[M]. Computer Security – ESORICS 2020. Cham: Springer International Publishing, 2020: 295-316.

[4] Huguenin-Dumittan L, Vaudenay S. Classical Misuse Attacks on NIST round 2 PQC[M]. Applied Cryptography and Network Security. Cham: Springer International Publishing, 2020: 208-227.

[5] Malina L, Ricci S, Dzurenda P, et al. Towards Practical Deployment of Post-Quantum Cryptography on Constrained Platforms and Hardware-Accelerated Platforms[C]. *Innovative Security Solutions for Information Technology and Communications*, 2020: 109-124.

[6] Koziel B, Kermani M M, Azarderakhsh R. Post-Quantum Cryptographic Hardware and Embedded Systems*Emerging Topics in Hardware Security*, 2021: 229-255.

[7] Li H Y, Liu R Z, Liu Z, et al. Ciphertext-only Attacks Against Compact-LWE Submitted to NIST PQC Project[J]. *Journal of Systems Science and Complexity*, 2022, 35(3): 1173-1190.

[8] Raavi M, Wuthier S, Chandramouli P, et al. Security Comparisons and Performance Analyses of Post-Quantum Signature Algorithms[C]. *Applied Cryptography and Network Security*, 2021: 424-447.

[9] Beullens W. Improved Cryptanalysis of UOV and Rainbow[C]. In: *40th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2021, 348-373.

[10] Howe J, Prest T, Apon D. SoK: How (not) to Design and Implement Post-quantum Cryptography[C]. In: *Cryptographers' Track at the RSA Conference 2021,* 2021, 444-477.

[11] Yang Y T, Han X G, Huang J R, et al. Bidirectional Authentication Key Agreement Protocol Supporting Identity's Privacy Preservation Based on RLWE[J]. *Journal on Communications*, 2019, 40(11): 180-186.
(杨亚涛, 韩新光, 黄洁润, 等. 基于 RLWE 支持身份隐私保护的双向认证密钥协商协议[J]. *通信学报*, 2019, 40(11): 180-186.)

[12] Yang Y T, Huang J R, Chen J Y, et al. INAKA: Improved Authenticated Key Agreement Protocol Based on Newhope[J]. *IEEE Access*, 8: 41764-41773.

[13] Li Z C, Xie T, Zhang J M. Post Quantum Password-Based Authentication Key Exchange Protocol Based on Ring Learning with Errors Problem[J]. *Acta Electronica Sinica*, 2021, 49(2): 260-267.
(李子臣, 谢婷, 张卷美. 基于 RLWE 问题的后量子口令认证密钥交换协议[J]. *电子学报*, 2021, 49(2): 260-267.)

[14] Garbis J, Chapman J.W. Zero Trust Security[M]. Apress, Berkeley, CA, 2021,127-134.

[15] Shu X F, Jiang N P. Analysis of the Security of Key Distribution Based on SSLVPN[J]. *Electronic Science and Technology*, 2017, 30(2): 165-168.
(舒晓飞, 蒋念平. 基于 SSLVPN 的密钥分配的安全性分析[J]. *电子科技*, 2017, 30(2): 165-168.)

[16] Kamal K M A, Almuhammadi S. Vulnerability of Virtual Private Networks to Web Fingerprinting Attack[C]. *Advances in Security*, *Networks*, *and Internet of Things*, 2021: 147-165.

[17] Ghilen A, Azizi M, Bouallegue R. Q-OpenVPN: A New Extension of OpenVPN Based on a Quantum Scheme for Authentication and Key Distribution[C]. *Cryptology and Network Security*, 2015: 238-247.

[18] Tang P Y, Li G C, Yu G, et al. VPN Enhanced Power Grid Communication Security Scheme Based on QS-KMS[J]. *Computer Engineering*, 2018, 44(12): 13-17.
(唐鹏毅, 李国春, 余刚, 等. 基于 QS-KMS 的 VPN 增强电网通信安全方案[J]. *计算机工程*, 2018, 44(12): 13-17.)

[19] Kniep Q.M, Müller W, Redlich JP. Post-Quantum Cryptography in WireGuard VPN[C]. *16th EAI International Conference on Security and Privacy in Communication Networks*, 2020, 261-267.

[20] Cho J.Y, Sergeev A. TLV-to-MUC Express: Post-quantum MACsec in VXLAN[C]. *25th Nordic Conference on Secure IT Systems,* 2021, 127-141.

[21] Albrecht M.R, Rechberger C, Schneider T, et al. Ciphers for MPC and FHE[C]. *34th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2015, 430-454.

[22] Melissa Chase, David Derler, Steven Goldfeder, et al. Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives[C]. *9th International Conference on Post -Quantum Cryptography*, 2018, 419-440.

[23] Bai S, Galbraith S D. An Improved Compression Technique for Signatures Based on Learning with Errors[M]. Topics in Cryptology – CT-RSA 2014. Cham: Springer International Publishing, 2014: 28-47.

[24] Gentry C, Peikert C, Vaikuntanathan V. Trapdoors for Hard Lat-

tices and New Cryptographic Constructions[C]. *The fortieth annual ACM symposium on Theory of computing*, 2008: 197-206.

[25] Ding J T, Schmidt D. Rainbow, a New Multivariable Polynomial Signature Scheme[M]. Applied Cryptography and Network Security. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005: 164-175.

[26] Bindel N, Hamburg M, Hövelmanns K, et al. Tighter Proofs of CCA Security in the Quantum Random Oracle Model[M]. Theory of Cryptography. Cham: Springer International Publishing, 2019: 61-90.

[27] Bos J, Ducas L, Kiltz E, et al. CRYSTALS - Kyber: A CCA-Secure Module-Lattice-Based KEM[C]. *2018 IEEE European Symposium on Security and Privacy*, 2018: 353-367.

[28] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman. NTRU: A ring-based public key cryptosystem[C]. *International Algorithmic Number Theory Symposium*, 1998, 267-288.

[29] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, et al. Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM[C]. *International Conference on Cryptology in Africa*, 2018, 282-305.

[30] Iatrou M G, Voyiatzis A G, Serpanos D N. Optimizations for High-Performance IPsec Execution[M]. Communications in Computer and Information Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011: 199-211.

杨亚涛 (1978–), 博士, 教授, 博导. 主要研究领域为密码学与通信安全、同态加密、密码协议和算法等。Email: yy2008@163.com

赵若岩 (2000–), 山西晋中人, 硕士研究生, 主要研究领域为密码学与信息安全。

常鑫 (1997–), 甘肃定西人, 硕士研究生, 主要研究领域为密码学与信息安全。

郭超 (1987–), 博士, 讲师. 主要研究领域为安全资源编排、信息安全等。

肖嵩 (1977–), 博士, 教授, 博导. 主要研究领域为多媒体通信安全, 通信与信息安全等。