Workgroup: Network Working Group
Internet-Draft:
draft-rransom-secsh-client-ntru-kex-00
Published: 6 December 2022
Intended Status: Informational
Expires: 9 June 2023
Authors: R. Ransom

Client-Side NTRU Key Exchange for Secure Shell (SSH)

Abstract

This document describes the specification for using NTRU keypairs generated by the client for key exchange in the Secure Shell (SSH) protocol.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 9 June 2023.

Table of Contents

1. Introduction

The Secure Shell (SSH) transport layer protocol specified in [RFC4253] provides for extension to support new key exchange methods. This document specifies key exchange methods which provide post-quantum security based on the NTRU KEM [NTRU].

For ease of implementation in existing SSH software, this key exchange method uses an ephemeral NTRU keypair generated by the client, retains the same structure and pattern of messages as the existing Diffie-Hellman and ECDH [RFC5656][RFC8731] key exchange methods, and relies on a signature keypair to authenticate the server.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Notation

In this document, the concatenation of two strings a and b will be denoted a || b.

3. Key exchange methods

The key exchange procedure follows the same general pattern as the ECDH key exchange specified in section 4 of [RFC5656], and uses the same message numbers; however, the contents differ, and the key exchange is not symmetric as in ECDH.

Each key exchange method name specifies both an NTRU parameter set and a hash function. NTRU operations Key_Pair, Encapsulate, and Decapsulate are performed as in [NTRU] for the given parameter set, except that its Hash is replaced with the hash function specified for the key exchange method. The kem_public_key_bytes and kem_ciphertext_bytes constants are also as specified in [NTRU] for the given parameter set.

Let hash_bytes denote the length of the hash function's output.

The client generates a private key priv and a public key pub by applying Key_Pair to the output of a random number generator. This key may be stored by the client and used for more than one connection. Each priv and pub **MUST** only be used with a single hash function.

For each connection, the client generates a new random string nonce of length hash_bytes. nonce **MUST NOT** be reused.

The client sends the following:

byte	SSH_MSG_KEX_ECDH_INIT
string	pub || nonce

Table 1

Both pub and nonce have fixed length for each key exchange method, so the string pub || nonce can be uniquely parsed into pub and nonce by the server. The server applies Encapsulate to pub, to produce a shared secret key sk and a ciphertext ct.

The server responds with:

byte	SSH_MSG_KEX_ECDH_REPLY
string	ct

Table 2

The client applies Decapsulate to priv and ct, to recover sk.

Let pad denote the string containing the single byte 1. Both parties compute K' as pad || Hash(sk || nonce), and compute K by interpreting K' as a big-endian integer. Equivalently, the mpint K specified by section 7.2 of the SSH transport layer protocol [RFC4253] as the secret output of a key exchange method can be replaced with the string K'.

The exchange hash H is computed as in section 4 of [RFC5656], with Q_C = pub || nonce and Q_S = ct.

4. Security considerations

The exchange hash H is computed using the hash algorithm specified by the key exchange method. This limits the security of authentication in both directions to the second-preimage resistance of the hash function specified by the weakest KEX accepted by both parties.

Reuse of an NTRU keypair for more than one Decapsulate operation is intended and believed to be safe, and the nonce sent by the client is used to prevent a replay of the server's ciphertext from producing the same exchange hash H or shared secret K. However, reusing a keypair discloses that multiple connections originated from the same client. Clients which support reuse of NTRU keypairs **MUST** document this key reuse, and **SHOULD** provide a way to disable it.

Section 7.2 of [RFC4253] specifies that the shared secret K is to be encoded as an mpint, in which bytes must be removed or added at the beginning to ensure certain conditions on the leading byte. As section 4 of [RFC8731] points out, this is likely to introduce a side-channel attack. This key exchange method prepends a fixed non-zero padding byte, to eliminate that side-channel risk without requiring extensive reworking of implementations which internally handle K as an mpint.

5. IANA considerations

This document specifies the following names to be added to the "Key Exchange Method Names" registry for SSH [IANA-KEX], as follows:

Key exchange method name	Hash function	NTRU parameter set
client-ntruhps4096821-sha3-512	SHA3-512	ntruhps4096821
client-ntruhps4096821-sha256	SHA-256	ntruhps4096821

Table 3: Key exchange method names

6. References

[RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, January 2006, <https://www.rfc-editor.org/info/rfc4253>.

[NTRU] Chen, C., Danba, O., Hoffstein, J., Hülsing, A., Rijneveld, J., Schanck, J. M., Schwabe, P., Whyte, W., and Z. Zhang, "NTRU - Algorithm Specifications and Supporting Documentation", NIST Post-Quantum Cryptography project submission document, 30 March 2019.

[RFC5656] Stebila, D. and J. Green, "Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer", RFC 5656, December 2009, <https://www.rfc-editor.org/info/rfc5656>.

[RFC8731] Adamantiadis, A., Josefsson, S., and M. Baushke, "Secure Shell (SSH) Key Exchange Method Using Curve25519 and Curve448", RFC 8731, February 2020, <https://www.rfc-editor.org/info/rfc8731>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[IANA-KEX] IANA, "Secure Shell (SSH) Protocol Parameters: Key Exchange Method Names", <https://www.iana.org/assignments/ssh-parameters/>.

Author's Address

Robert Ransom