Workgroup: Internet Engineering Task Force
Published: 20 January 2024
Intended Status: Informational
Expires: 23 July 2024
Authors: S. Josefsson

Hybrid X25519 and Streamlined NTRU Prime sntrup761 with SHA3-256: Chempat-X

Abstract

This memo define Chempat-X, a post-quantum/traditional hybrid key exchange method (PQ/T KEM) based on X25519 and Streamlined NTRU Prime sntrup761 with SHA3-256.

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 23 July 2024.

Copyright Notice

Copyright (c) 2024 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction

Streamlined NTRU Prime [NTRUPrime] [NTRUPrimePQCS] provides post-quantum small lattice-based key-encapsulation mechanisms. The variant sntrup761 instance has been implemented widely.

The pre-quantum elliptic-curve Diffie-Hellman X25519 function [RFC7748] has been widely implemented.

To hedge against attacks on either of sntrup761 or X25519 a hybrid construction may be used, with the intention that the hybrid would be secure if either of the involved algorithms are flawed.

This document describes how to implement key exchange based on a hybrid between Streamlined NTRU Prime sntrup761 and X25519 with SHA3-256 [NIST_FIPS_202].

This design is based on the Secure Shell protocol "sntrup761x25519-sha512", but we use a stronger combiner of the resulting shared secret. We offer this document for other protocols that desire to use a hybrid key exchange method based on established mechanisms.

2. Chempat-X Key Exchange Method

The key-agreement is done by the X25519 Diffie-Hellman protocol as described in section 6.1 (Curve25519) of [RFC7748], and the key encapsulation method described in [NTRUPrimePQCS].

Alice sends a concatenation of the 1158 byte public key output from the key generator of sntrup761 with the 32 byte K_A = X25519(a, 9) as described in [NTRUPrimePQCS] and [RFC7748]. The output value is thus 1190 bytes.

Bob sends a concatenation of the 1039 byte ciphertext output from the key encapsulation mechanism of sntrup761 with the 32 byte K_B = X25519(b, 9) as described in [NTRUPrimePQCS] and [RFC7748]. The output value is thus 1071 bytes.

Alice derive the 32 byte shared K1 based on the X25519 values as described in [RFC7748] and performs the sntrup761 key decapsulation

operation as described in [NTRUPrimePQCS] to yield the 32 byte shared secret K2. Alice derives the final hybrid shared secret key K as described below.. The output is 32 bytes.

Bob derive the 32 byte shared K1 based on the X25519 values as described in [RFC7748] and takes the 32 byte shared secret key K2 from the earlier key encapsulation method of sntrup761. Bob derives the final hybrid shared secret secret key K as described below. The output is 32 bytes.

Alice and Bob has now established a shared key.

3. Key Combiner

The final hybrid shared secret key "hybridss" is derived using SHA3-256 as follows:

```
H = SHA3-256,
hybridpk = (receiverpkECDH,receiverpkKEM),
hybridct = (senderpkECDH,senderctKEM),
hybridss = H(ssECDH,ssKEM,H(hybridct),H(hybridpk),context)
```

Figure 1

4. Acknowledgements

This work is a simple generalization of the sntrup761x25519-sha512 mechanism due to [OpenSSH] and TinySSH [TinySSH] documented in draft-josefsson-ntruprime-ssh-00, but modified to use a stronger combiner function suggested by Daniel J. Bernstein.

5. Security Considerations

The security considerations of [RFC7748], [NTRUPrimePQCS] and [NIST_FIPS_202] are inherited.

While the construct should remain secure if either X25519 or sntrup761 is found to be insecure, the security of the combined hybrid construction also depends on the security of the combiner algorithm.

6. IANA Considerations

This document has no IANA actions.

7. References

7.1. Normative References

[NIST_FIPS_202]

Dworkin, M. J. and NIST, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions", NIST Federal Information Processing Standards Publications 202, DOI 10.6028/NIST.FIPS.202, July 2015, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.202.pdf>.

[NTRUPrimePQCS] Bernstein, D.J., Brumley, B. B., Chen,, M., Chuengsatiansup, C., Lange, T., Marotzke, A., Peng, B., Tuveri, N., Vredendaal, C. V., and B. Yang, "NTRU Prime: round 3", WWW https://ntruprime.cr.yp.to/nist/ntruprime-20201007.pdf, October 2020.

[RFC7748] Langley, A., Hamburg, M., and S. Turner, "Elliptic Curves for Security", RFC 7748, DOI 10.17487/RFC7748, January 2016, <https://www.rfc-editor.org/info/rfc7748>.

7.2. Informative References

[NTRUPrime] Bernstein, D.J., Chuengsatiansup, C., Lange, T., and C. van Vredendaal, "NTRU Prime: reducing attack surface at low cost", WWW https://ntruprime.cr.yp.to/ntruprime-20170816.pdf, August 2017.

[OpenSSH] OpenSSH group of OpenBSD, "The OpenSSH Project", <https://www.openssh.com/>.

[TinySSH] TinySSH, "TinySSH - minimalistic SSH server which implements only a subset of SSHv2 features", <https://www.tinyssh.org/>.

Author's Address

Simon Josefsson

Email: simon@josefsson.org
URI: https://blog.josefsson.org/