Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 23 April 2026

S. Josefsson

20 October 2025

Secure Shell Key Exchange Method Using Chempat Hybrid of Classic McEliece and X25519 with SHA-512: mceliece6688128x25519-sha512 draft-josefsson-ssh-mceliece-02

Abstract

This document specify a hybrid key exchange method in the Secure Shell (SSH) protocol based on Classic McEliece (mceliece6688128) and X25519 with SHA-512 using Chempat as the combiner.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at https://datatracker.ietf.org/doc/draft-josefsson-ssh-mceliece/.

Source for this draft and an issue tracker can be found at https://gitlab.com/jas/ietf-ssh-mceliece.

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 April 2026.

Copyright Notice

Table of Contents

1. Introduction

Secure Shell (SSH) [RFC4251] is a secure remote login protocol. The key exchange protocol described in [RFC4253] supports an extensible set of methods. [RFC5656] defines how elliptic curves are integrated into this extensible SSH framework, and [RFC8731] specify "curve25519-sha256" to support the pre-quantum elliptic-curve Diffie-Hellman X25519 function [RFC7748]. In [I-D.josefsson-ntruprime-ssh] it is described how the post-quantum lattice-based Streamlined NTRU Prime is combined with X25519 for SSH, and we base our protocol and document on it but replace sntrup761 with mceliece6688128 and use Chempat [I-D.josefsson-chempat] for the combiner.

Classic McEliece [I-D.josefsson-mceliece] [CM-spec] provides a code-based Key Encapsulation Method (KEM) designed to be safe even against quantum computers. The variant "mceliece6688128" offers a balance between performance and output sizes.

To hedge against attacks on either of mceliece6688128 or X25519 a hybrid construction Chempat is used, with the intention that the hybrid would be secure if either of the involved algorithms are flawed.

This document specify how to implement key exchange based on a Chempat hybrid between Classic McEliece mceliece6688128 and X25519 [RFC6234] in SSH.

The SHA-512 in the name of this method refers to the HASH used in Section 7.2 (Output from Key Exchange) of [RFC4253], not that of the hybrid KEM combiner.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Key Exchange Method: mceliece6688128x25519-sha512

The key-agreement is done by the X25519 Diffie-Hellman protocol as described in Section 3 (Key Exchange Methods) of [RFC8731], and the mceliece6688128 key encapsulation method described in [I-D.josefsson-mceliece] [CM-spec].

The key exchange procedure reuse the Elliptic Curve Diffie-Hellman (ECDH) key exchange defined in Sections 4 (ECDH Key Exchange) and 7.1 (ECDH Message Numbers) of [RFC5656]. The protocol flow and the SSH_MSG_KEX_ECDH_INIT and SSH_MSG_KEX_ECDH_REPLY messages are identical, except that we use different ephemeral public values Q_C and Q_S and shared secret K as described below.

The SSH_MSG_KEX_ECDH_INIT value Q_C that holds the client’s ephemeral public key MUST be constructed by concatenating the 1044992 byte public key output from the key generator of mceliece6688128 (or mceliece6688128f, see Section 4) with the 32 byte K_A = X25519(a, 9) as described in [I-D.josefsson-mceliece] [CM-spec] and [RFC8731]. The Q_C value is thus 1045024 bytes.

The SSH_MSG_KEX_ECDH_REPLY value Q_S that holds the server’s ephemeral public key MUST be constructed by concatenating the 208 byte ciphertext output from the key encapsulation mechanism of mceliece6688128 (or mceliece6688128f, see Section 4) with the 32 byte K_B = X25519(b, 9) as described in [I-D.josefsson-mceliece] [CM-spec] and [RFC8731]. The Q_S value is thus 240 bytes.

Clients and servers MUST abort if the length of the received public keys Q_C or Q_S are not the expected lengths. An abort for these purposes is defined as a disconnect (SSH_MSG_DISCONNECT) of the session and SHOULD use the SSH_DISCONNECT_KEY_EXCHANGE_FAILED reason

for the message, see Section 11.1 (Disconnection Message) of [RFC4253]. No further validation is required beyond what is described in [RFC7748], [RFC8731] and [I-D.josefsson-mceliece] [CM-spec].

The SSH_MSG_KEX_ECDH_REPLY signature value is computed as described in [RFC5656] with the following changes. Instead of encoding the shared secret K as 'mpint', it MUST be encoded as 'string'. The shared secret K value MUST be the 32-byte output octet string computed by Chempat-X25519-mceliece6688128 [I-D.josefsson-chempat].

4. mceliece6688128f

The f and non-f versions are interoperable. The f versions have faster key generation, while the non-f versions have simpler key generation. For example, a key generated with mceliece6688128f can decapsulate ciphertexts that were encapsulated with mceliece6688128, and vice versa. The secret-key sizes (and formats) are the same, the encapsulation functions are the same, and the decapsulation functions are the same.

Implementations of this protocol can chose between mceliece6688128 or mceliece6688128f, however the name of this protocol is "mceliece6688128x25519-sha512" even for implementations that use mceliece6688128f internally.

Choosing mceliece6688128 generally reduce code size and complexity (at the expense of performance), and choosing mceliece6688128f generally improve performance (at the expense of code size and complexity).

5. Acknowledgments

The protocol and document is based on [I-D.josefsson-ntruprime-ssh]. The authors would like to thank Daniel J. Bernstein for discussion and suggesting the mceliece6688128 variant.

6. Implementation Status

An earlier implementation of this protocol is available as a patch [OpenSSH-McEliece-patch] for OpenSSH [OpenSSH], released under a BSD-style license.

7. Security Considerations

The security considerations of [RFC4251], [RFC5656], [RFC7748], [RFC8731], [I-D.josefsson-chempat] and [I-D.josefsson-mceliece] [CM-spec] [CM-security] [CM-impl] are inherited.

Classic McEliece is a KEM designed for IND-CCA2 security at a very high security level, even against quantum computers. The algorithm has been studied by researchers for many years, and there are implementations in the public domain for a wide range of architectures. Chempat is a conservatively designed way to combine a classical and post-quantum method. However new cryptographic primitives should be introduced and trusted conservatively, and new research findings may be published at any time that may warrant implementation reconsiderations.

The increase in communication size and computational requirements may be a concern for limited computational devices, which would then not be able to take advantage of the improved security properties offered by this work.

As discussed in the security considerations of Curve25519-sha256 [RFC8731], the X25519 shared secret K is used bignum-encoded in that document, and this raise a potential for a hash-processing time side-channel that could leak one bit of the secret due to different length of the bignum sign pad. This document resolve that problem by using string-encoding instead of bignum-encoding.

8. IANA Considerations

IANA is requested to add a new "Method Name" of "mceliece6688128x25519-sha512" to the "Key Exchange Method Names" registry for Secure Shell (SSH) Protocol Parameters [IANA-KEX] with a "reference" field to this RFC and the "OK to implement" field of "MUST".

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC4251] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Architecture", RFC 4251, DOI 10.17487/RFC4251, January 2006, <https://www.rfc-editor.org/info/rfc4251>.

[RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <https://www.rfc-editor.org/info/rfc4253>.

[RFC5656] Stebila, D. and J. Green, "Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer", RFC 5656, DOI 10.17487/RFC5656, December 2009, <https://www.rfc-editor.org/info/rfc5656>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8731] Adamantiadis, A., Josefsson, S., and M. Baushke, "Secure Shell (SSH) Key Exchange Method Using Curve25519 and Curve448", RFC 8731, DOI 10.17487/RFC8731, February 2020, <https://www.rfc-editor.org/info/rfc8731>.

9.2. Informative References

[CM-impl] Classic McEliece Team, "Classic McEliece: conservative code-based cryptography: guide for implementors", October 2022, <https://classic.mceliece.org/mceliece-impl-20221023.pdf>.

[CM-security] Classic McEliece Team, "Classic McEliece: conservative code-based cryptography: guide for security reviewers", October 2022, <https://classic.mceliece.org/mceliece-security-20221023.pdf>.

[CM-spec] Classic McEliece Team, "Classic McEliece: conservative code-based cryptography: cryptosystem specification", October 2022, <https://classic.mceliece.org/mceliece-spec-20221023.pdf>.

[I-D.josefsson-chempat] Josefsson, S., "Chempat: Generic Instantiated PQ/T Hybrid Key Encapsulation Mechanisms", Work in Progress, Internet-Draft, draft-josefsson-chempat-04, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-josefsson-chempat-04>.

[I-D.josefsson-mceliece] Josefsson, S., "Classic McEliece", Work in Progress, Internet-Draft, draft-josefsson-mceliece-03, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-josefsson-mceliece-03>.

[I-D.josefsson-ntruprime-ssh] Friedl, M., Mojzis, J., and S. Josefsson, "Secure Shell (SSH) Key Exchange Method Using Hybrid Streamlined NTRU

Prime sntrup761 and X25519 with SHA-512: sntrup761x25519-sha512", Work in Progress, Internet-Draft, draft-josefsson-ntruprime-ssh-03, 17 August 2024, <https://datatracker.ietf.org/doc/html/draft-josefsson-ntruprime-ssh-03>.

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

[OpenSSH] OpenSSH team, "OpenSSH", n.d., <https://www.openssh.com/>.

[OpenSSH-McEliece-patch] OpenSSH team, Simon Josefsson, "GitLab branch of OpenSSH with McEliece support", n.d., <https://gitlab.com/jas/openssh-portable/-/tree/jas/mceliece>.

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

[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>.

Author’s Address

Simon Josefsson
Email: simon@josefsson.org