Internet Engineering Task Force
Internet-Draft
Updates: 4253, 4419, 4432, 4462, 5656
(if approved)
Intended status: Standards Track
Expires: March 24, 2017

M. Baushke
Juniper Networks, Inc.
September 20, 2016

Key Exchange (KEX) Method Updates and Recommendations for Secure Shell (SSH)
draft-ietf-curdle-ssh-kex-sha2-06

Abstract

This document adds recommendations for adoption of ssh-curves from the [I-D.ietf-curdle-ssh-curves] and new-modp from the [I-D.ietf-curdle-ssh-modp-dh-sha2], and deprecates some previously specified Key Exchange Method algorithm names for the Secure Shell (SSH) protocol. It also updates [RFC4253], [RFC4419], [RFC4462], and [RFC5656] by specifying the set key exchange algorithms that currently exist and which ones MUST, SHOULD, MAY, and SHOULD NOT be implemented. New key exchange methods use the SHA-2 family of hashes.

Status of This Memo

Copyright Notice

1. Overview and Rationale

Secure Shell (SSH) is a common protocol for secure communication on the Internet. In [RFC4253], SSH originally defined the Key Exchange Method Name diffie-hellman-group1-sha1 which used [RFC2409] Oakley Group 2 (a 1024-bit MODP group) and SHA-1 [RFC3174]. Due to recent security concerns with SHA-1 [RFC6194] and with MODP groups with less than 2048 bits [NIST-SP-800-131Ar1] implementer and users request support for larger MODP group sizes with data integrity verification using the SHA-2 family of secure hash algorithms as well as MODP groups providing more security.

The United States Information Assurance Directorate (IAD) at the National Security Agency (NSA) has published a FAQ [MFQ-U-OO-815099-15] suggesting that the use of Elliptic Curve Diffie-Hellman (ECDH) using the nistp256 curve and SHA-2 based hashes less than SHA2-384 are no longer sufficient for transport of Top Secret information. It is for this reason that this draft moves ecdh-sha2-nistp256 from a REQUIRED to OPTIONAL as a key exchange method. This is the same reason that the stronger MODP groups being adopted. As the MODP group14 is already present in most SSH implementations and most implementations already have a SHA2-256 implementation, so diffie-hellman-group14-sha256 is provided as an easy to implement and faster to use key exchange. Small embedded applications may find this KEX desirable to use.

The NSA Information Assurance Directorate (IAD) has also published the Commercial National Security Algorithm Suite (CNSA Suite) [CNSA-SUITE] in which the 3072-bit MODP Group 15 in RFC 3526 is explicitly mentioned as the minimum modulus to protect Top Secret communications.

It has been observed in [safe-curves] that the NIST recommended Elliptic Curve Prime Curves (P-256, P-384, and P-521) are perhaps not the best available for Elliptic Curve Cryptography (ECC) Security. For this reason, none of the [RFC5656] curves are marked as a MUST implement. However, the requirement that "every compliant SSH ECC implementation MUST implement ECDH key exchange" is now taken to mean that if ecdsa-sha2-[identifier] is implemented, then ecdh-sha2-[identifier] MUST be implemented.

Please send comments on this draft to curdle@ietf.org.

2. Requirements Language

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].

3. Key Exchange Algorithms

This memo adopts the style and conventions of [RFC4253] in specifying how the use of new data key exchange is indicated in SSH.

A new set of Elliptic Curve Diffie-Hellman ssh-curves exist. The curve25519-sha256 MUST be adopted where possible.

As a hedge against uncertainty raised by the NSA IAD FAQ publication, new MODP Diffie-Hellman based key exchanges are proposed for inclusion in the set of key exchange method names as well as the curve448-sha512 curve.

The following new key exchange algorithms are defined:

Key Exchange Method Name	Note
curve25519-sha256	MUST/REQUIRED
curve448-sha512	MAY/OPTIONAL
diffie-hellman-group14-sha256	MUST/REQUIRED
diffie-hellman-group15-sha512	MAY/OPTIONAL
diffie-hellman-group16-sha512	SHOULD/RECOMMENDED
diffie-hellman-group17-sha512	MAY/OPTIONAL
diffie-hellman-group18-sha512	MAY/OPTIONAL
gss-group14-sha256-*	SHOULD/RECOMMENDED
gss-group15-sha512-*	MAY/OPTIONAL
gss-group16-sha512-*	SHOULD/RECOMMENDED
gss-group17-sha512-*	MAY/OPTIONAL
gss-group18-sha512-*	MAY/OPTIONAL

The SHA-2 family of secure hash algorithms are defined in [FIPS-180-4].

4. IANA Considerations

This RFC augments the Key Exchange Method Names in [RFC4253]. It downgrades the use of SHA-1 hashing for key exchange methods in [RFC4419], [RFC4432], and [RFC4462]. It also moves from MUST to SHOULD the ecdh-sha2-nistp256 given in [RFC5656].

It adds a new set of named "gss-*" methods to [RFC4462] with a MAY recommendation.

It is desirable to also include the new-modp from the [I-D.ietf-curdle-ssh-modp-dh-sha2] in this list.

It is desirable to also include the ssh-curves from the [I-D.ietf-curdle-ssh-curves] in this list. The "curve25519-sha256" is currently available in some Secure Shell implementations under the name "curve25519-sha256@libssh.org" and is the best candidate for a fast, safe, and secure key exchange method.

IANA is requested to update the SSH algorithm registry to ensure that all of the listed Key Exchange Method Name and References exist in the following table. However, the Implement column is just the current recommendations of this RFC.

Key Exchange Method Name	Reference	Implement
curve25519-sha256	ssh-curves	MUST
curve448-sha512	ssh-curves	MAY
diffie-hellman-group-exchange-sha1	RFC4419	SHOULD NOT
diffie-hellman-group-exchange-sha256	RFC4419	MAY
diffie-hellman-group1-sha1	RFC4253	SHOULD NOT
diffie-hellman-group14-sha1	RFC4253	SHOULD
diffie-hellman-group14-sha256	new-modp	MUST
diffie-hellman-group15-sha512	new-modp	MAY
diffie-hellman-group16-sha512	new-modp	SHOULD
diffie-hellman-group17-sha512	new-modp	MAY
diffie-hellman-group18-sha512	new-modp	MAY
ecdh-sha2-nistp256	RFC5656	SHOULD
ecdh-sha2-nistp384	RFC5656	SHOULD
ecdh-sha2-nistp521	RFC5656	SHOULD
ecdh-sha2-*	RFC5656	MAY
ecmqv-sha2	RFC5656	SHOULD NOT
gss-gex-sha1-*	RFC4462	SHOULD NOT
gss-group1-sha1-*	RFC4462	SHOULD NOT
gss-group14-sha1-*	RFC4462	SHOULD
gss-group14-sha256-*	new-modp	SHOULD
gss-group15-sha512-*	new-modp	MAY
gss-group16-sha512-*	new-modp	SHOULD
gss-group17-sha512-*	new-modp	MAY
gss-group18-sha512-*	new-modp	MAY
gss-*	RFC4462	MAY
rsa1024-sha1	RFC4432	SHOULD NOT
rsa2048-sha256	RFC4432	MAY

The Implement column in the above table is a suggestion/ recommendation for the listed key exchange method to be implemented in the default list of key exchange methods. It is up to the end-user as to what algorithms they choose to be able to negotiate, so the KEX algorithms should be configurable by the administrator of the server as well as the user of the client. This RFC is intended to provide IANA defined names for these groups for interoperability. The Note column of the IANA table should probably continue to point to the implementation detail sections of the Reference RFCs where appropriate.

The guidance of his RFC is that the SHA-1 algorithm hashing SHOULD NOT be used. If it is used, it should only be provided for backwards compatibility, should not be used in new designs, and should be phased out of existing key exchanges as quickly as possible because of its known weaknesses. Any key exchange using SHA-1 SHOULD NOT be in a default key exchange list if at all possible. If they are needed for backward compatibility, they SHOULD be listed after all of the SHA-2 based key exchanges.

The RFC4253 REQUIRED diffie-hellman-group14-sha1 method SHOULD be retained for compatibility with older Secure Shell implementations. It is intended that this key exchange method be phased out as soon as possible.

It is believed that all current SSH implementations should be able to achieve an implementation of the "diffie-hellman-group14-sha256" method. To that end, this is one method that MUST be implemented.

If GSS-API methods are available, then the RFC4462 REQUIRED gss-group14-sha1-* method SHOULD be retained for compatibility with older Secure Shell implementations and the gss-groups14-sha256-* method SHOULD be added as for "sha1".

[TO BE REMOVED: This registration should take place at the following location: <http://www.iana.org/assignments/ssh-parameters/ssh-parameters.xhtml#ssh-parameters-16>]

5. Acknowledgements

Thanks to the following people for review and comments: Denis Bider, Peter Gutmann, Damien Miller, Niels Moeller, Matt Johnston, Iwamoto Kouichi, Simon Josefsson, Dave Dugal, Daniel Migault.

Thanks to the following people for code to implement interoperable exchanges using some of these groups as found in an this draft: Darren Tucker for OpenSSH and Matt Johnston for Dropbear. And thanks to Iwamoto Kouichi for information about RLogin, Tera Term (ttssh)

and Poderosa implementations also adopting new Diffie-Hellman groups based on this draft.

6. Security Considerations

The security considerations of [RFC4253] apply to this RFC.

The security considerations of [RFC3526] suggest that these MODP groups have security strengths given in this table. They are based on [RFC3766] Determining Strengths For Public Keys Used For Exchanging Symmetric Keys.

Group modulus security strength estimates (RFC3526)

Group	Modulus	Strength Estimate 1		Strength Estimate 2	
		in bits	exponent size	in bits	exponent size
14	2048-bit	110	220-	160	320-
15	3072-bit	130	260-	210	420-
16	4096-bit	150	300-	240	480-
17	6144-bit	170	340-	270	540-
18	8192-bit	190	380-	310	620-

Figure 1

Many users seem to be interested in the perceived safety of using larger MODP groups and hashing with SHA2-based algorithms.

7. References

7.1. Normative References

[FIPS-180-4]
National Institute of Standards and Technology, "Secure Hash Standard (SHS)", FIPS PUB 180-4, August 2015, <http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf>.

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

[RFC3526] Kivinen, T. and M. Kojo, "More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE)", RFC 3526, DOI 10.17487/RFC3526, May 2003, <http://www.rfc-editor.org/info/rfc3526>.

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

7.2. Informative References

[CNSA-SUITE] "Information Assurance by the National Security Agency", "Commercial National Security Algorithm Suite", September 2016, <https://www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm>.

[I-D.ietf-curdle-ssh-curves] Adamantiadis, A. and S. Josefsson, "Secure Shell (SSH) Key Exchange Method using Curve25519 and Curve448", draft-ietf-curdle-ssh-curves-00 (work in progress), March 2016.

[I-D.ietf-curdle-ssh-modp-dh-sha2] Baushke, M., "More Modular Exponential (MODP) Diffie-Hellman (DH) Key Exchange (KEX) Groups for Secure Shell (SSH)", draft-ietf-curdle-ssh-modp-dh-sha2-00 (work in progress), September 2016.

[MFQ-U-OO-815099-15] "National Security Agency/Central Security Service", "CNSA Suite and Quantum Computing FAQ", January 2016, <https://www.iad.gov/iad/library/ia-guidance/ia-solutions-for-classified/algorithm-guidance/cnsa-suite-and-quantum-computing-faq.cfm>.

[NIST-SP-800-131Ar1] Barker, and Roginsky, "Transitions: Recommendation for the Transitioning of the Use of Cryptographic Algorithms and Key Lengths", NIST Special Publication 800-131A Revision 1, November 2015, <http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-131Ar1.pdf>.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, DOI 10.17487/RFC2409, November 1998, <http://www.rfc-editor.org/info/rfc2409>.

[RFC3174] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, DOI 10.17487/RFC3174, September 2001, <http://www.rfc-editor.org/info/rfc3174>.

[RFC3766] Orman, H. and P. Hoffman, "Determining Strengths For Public Keys Used For Exchanging Symmetric Keys", BCP 86, RFC 3766, DOI 10.17487/RFC3766, April 2004, <http://www.rfc-editor.org/info/rfc3766>.

[RFC4419] Friedl, M., Provos, N., and W. Simpson, "Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol", RFC 4419, DOI 10.17487/RFC4419, March 2006, <http://www.rfc-editor.org/info/rfc4419>.

[RFC4432] Harris, B., "RSA Key Exchange for the Secure Shell (SSH) Transport Layer Protocol", RFC 4432, DOI 10.17487/RFC4432, March 2006, <http://www.rfc-editor.org/info/rfc4432>.

[RFC4462] Hutzelman, J., Salowey, J., Galbraith, J., and V. Welch, "Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol", RFC 4462, DOI 10.17487/RFC4462, May 2006, <http://www.rfc-editor.org/info/rfc4462>.

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

[RFC6194] Polk, T., Chen, L., Turner, S., and P. Hoffman, "Security Considerations for the SHA-0 and SHA-1 Message-Digest Algorithms", RFC 6194, DOI 10.17487/RFC6194, March 2011, <http://www.rfc-editor.org/info/rfc6194>.

[safe-curves]
Bernstein, and Lange, "SafeCurves: choosing safe curves for elliptic-curve cryptography.", February 2016, <https://safecurves.cr.yp.to/>.

Author's Address

Mark D.   Baushke
Juniper Networks, Inc.
1133 Innovation Way
Sunnyvale, CA 94089-1228
US

Phone: +1 408 745 2952
Email: mdb@juniper.net
URI: http://www.juniper.net/