

Individual draft
Internet Draft
Intended status: <Informational>
Expires: November 22, 2013

Q. Dang
NIST
S. Turner
IECA
May 22, 2013

**Recommended Usages of SHA-512/224, SHA-512/256
draft-dang-turner-sha-512-224-256-00.txt**

Abstract

This document provides recommendations on the use of the secure hash functions SHA-512/224 and SHA-512/256 specified in FIPS 180. SHA-512/224 and SHA-512/256 are SHA-512-based and truncated to match the output size of SHA-224 and SHA-256. On 64-bit platforms, the SHA-512-truncated algorithms provide better performance than their comparably sized SHA-224 and SHA-256 variants.

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), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/lid-abstracts.html>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

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 November 22, 2013.

Copyright Notice

Copyright (c) 2013 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 (<http://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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction.....	2	
2. Conventions used in this document.....	3	
3. Usage Recommendation for Digital Signatures with SHA-512/224 and SHA-512/256.....	3	
4. SHA-512/224 and SHA-512/256 in HMAC	5	
5. Security Considerations.....	5	
6. IANA Considerations.....	5	
7. Conclusions.....	5	
8. References	5	
8.1. Normative References		5
8.2. Informative References		6
9. Acknowledgments.....	6	
10. Authors'Addresses.....	7	

[1. Introduction](#)

NIST specified two hash algorithms, SHA-512/224 and SHA-512/256, in the hash algorithms standard: FIPS 180 [[FIPS180](#)]. These two hash algorithms have the same performance characteristics of SHA-512 since the only differences between them and SHA-512 are the initial hash values (IVs) and the truncation step to reduce the 512-bit last internal hash value to become 224 or 256-bit final hash value for SHA-512/224 and SHA-512/256 respectively.

SHA-512 consumes roughly 10-45% fewer clock cycles per byte than SHA-256 as shown from performance-comparison data for SHA-256 and SHA-512 on many different 64-bit platforms by [[SHA256](#)]. This means that SHA-512 runs roughly 10-80% faster than SHA-256 and SHA-224 on these 64-bit machines, which are becoming more prevalent. Also, [512/256] provides performance comparison data for SHA-256 and SHA-512 on a specific 2010 Intel architecture, the Xeon X5670 processor. The data shows that SHA-512 consumes roughly 37% fewer clock cycles per byte than SHA-256. Put another way, SHA-512 is roughly 60% faster (more efficient) than SHA-256 on this machine.

This internet draft discusses the choices between using SHA-224 and SHA-256 verses SHA-512/224 and SHA-512/256 in digital signature applications and HMACs based on their performance advantages to each other.

2. Conventions used in this document

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. Usage Recommendation for Digital Signatures with SHA-512/224 and SHA-512/256

Obviously, SHA-512/224 and SHA-512/256 may be substituted for SHA-224 and SHA-256 respectively in protocols and applications.

One of the common uses of hash functions is in digital signature applications. There are three NIST-approved digital signature algorithms defined in [[FIPS186](#)]: RSA, DSA and ECDSA.

When a 1024 or 2048-bit RSA digital signature algorithm is used, any of the approved hash functions can be used since their biggest hash value is only 512 bits (when SHA-512 is used). Different padding methods have different required fields in the data block that is signed by the RSA private key and the RSA moduli (1024 or 2048 bits). The total size of these required fields and the hash value is not greater than 1024 bits. Therefore, RSA digital signature applications will not have any technical issues in deploying any of the approved hash algorithms including SHA-512. Therefore, SHA-512/224 and SHA-512/256 are not preferred over SHA-512 for RSA digital signature applications. However, if a RSA digital signature application in a system that is a 64-bit platform, SHA-512/224 and SHA-512/256 are preferred over SHA-224 and SHA-256 respectively due to their performance advantage over these latter two hash functions.

If communicating points in a protocol are mainly to be run on 64-bit platforms, SHA-512/224 or SHA-512/256 should be used in 2048-bit RSA digital signature application. It is important to note that 1024-bit RSA digital signature generation is disallowed by NIST after 2013, see SP 800-131A [[131A](#)] for more details.

If digital signature algorithm is negotiable in a protocol where communicating points may be run on both 64-bit and smaller (32-bit for example) platforms, RSA digital signature with either SHA-512/224 or SHA-512/256 should be an option if RSA digital signature algorithm is supported. For example, if both ends of a communication run on 64-

bit platforms, they may want to use RSA with SHA-512/224 or SHA-512/256. If both ends of the communication run on 32-(or smaller) bit platforms (constrained environments), they may prefer to use RSA with SHA-224 or SHA-256 instead. And, if one end runs on 64-bit platform and the other end runs on a 32-(or smaller) bit platform, then it depends on the situation for which what digital signature algorithm: RSA with SHA-512/224 (or SHA-512/256) or RSA with SHA-224 (or SHA-256) should be used (from negotiation). A server running on a 64-bit machine that handles a lot of computation with many clients may prefer to use RSA with SHA-512/224 or SHA-512/256, but a constrained client may prefer to use RSA with SHA-224 or SHA-256 instead.

For DSA, there are two key pair sizes, which are NIST-approved: (L=2048, N=224) and (L=3072, N=256) (the key pair size: (L = 1024, N = 160) is not NIST-allowed to generate new digital signatures after the end of 2013). In DSA digital signature generation process (see FIPS 186 for details), if the hash value of the message is greater than N (size of p), only N left-most bits of the hash value will be used in the signing operation. Therefore, there is no security reasons to deploy a hash function which produces hash output larger than N (in bits) such as SHA-512. So, when getting performance advantage from SHA-512/224 and SHA-512/256 over SHA-224 and SHA-256 on the platforms which are optimized for 64-bit operations is a good thing, SHA-512/224 and SHA-512/256 should be used for (L=2048, N=224) and (L=3072, N=256) DSA digital signature applications respectively.

If communicating points in a protocol are mainly to be run on 64-bit platforms, SHA-512/224 and SHA-512/256 should be used in (L=2048, N=224) and (L=3072, N=256) DSA digital signature applications respectively.

If digital signature algorithm is negotiable in a protocol where communicating points may be run on both 64-bit and smaller (32-bit for example) platforms, DSA with SHA-512/224 or SHA-512/256 should be an option if DSA digital signature algorithm is supported

ECDSA digital signature algorithms are specified in FIPS 186. Their NIST-approved key sizes and hash functions are described in SPs 800-57, part 1 [57] and 800-131A [131A]. After 2013, only curves with n at least 224 bits are NIST-approved for digital signature generation. In ECDSA, if the hash function produces the hash value bigger than the size of n, then only the n left-most bits of the hash value are used in computing and verifying the ECDSA digital signatures.

If communicating points in a protocol are mainly to be run on 64-bit platforms, SHA-512/224 and SHA-512/256 should be used in 224 and 256-bit ECDSA digital signature applications respectively.

If digital signature algorithm is negotiable in a protocol where communicating points may be run on both 64-bit and smaller (32-bit for example) platforms, 224 or 256-bit ECDSA with SHA-512/224 or SHA-512/256 respectively should be an option if ECDSA digital signature algorithm is supported.

4. SHA-512/224 and SHA-512/256 in HMAC

Besides being used in digital signature applications, hash functions are also used in HMAC [[RFC2104](#)]. If an exact 224-bit or 256-bit HMAC value is needed, SHA-512/224 and SHA-512/256 should be used instead of truncating SHA-512's hash output. And, HMAC with SHA-512/224 or SHA-512/256 is strongly recommended for protocols where communicating parties are mainly to be run on 64-bit platforms over HMAC with SHA-224 or SHA-256 respectively.

5. Security Considerations

Note that SHA-512/224 and SHA-512/256 provide 112 and 128 bits of collision resistance for digital signatures. See NIST SP 800-107 [[107](#)] for more discussion about security of these two hash functions.

6. IANA Considerations

None.

7. Conclusions

Will be added later.

8. References

8.1. Normative References

- [FIPS180] Federal Information Processing Standard (FIPS) 180-4, Secure Hash Standard, National Institute of Standards and Technology, March 2012.
- [FIPS186] Federal Information Processing Standard (FIPS) 186-3, Digital Signature Standard (DSS), National Institute of Standards and Technology, June 2009.

- [RFC2104] Krawczyk, H., Bellare, M. and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", [RFC 2104](#), February 1997.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.

8.2. Informative References

- [SHA256] <http://bench.cr.yp.to/xweb-hash/long-sha256.html>
- [512/256] Shay Gueron, Simon Johnson and Jesse Walker, SHA-512/256, 2011 Eighth International Conference on Information Technology: New Generat 7.
- [57] NIST Special Publication (SP) 800-57, Part 1, Recommendation for Key Management: General,(Revision 3) July 2012.
- [107] NIST SP 800-107, Revision 1, Recommendation for Applications Using Approved Hash Algorithms, August 2012.
- [131A] E. Barker and A. Roginsky, "Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths", NIST Special Publication 800-131A, January 2011.

9. Acknowledgments

Will be added later.

10. Authors' Addresses

Quynh Dang
NIST
100 Bureau Drive, Stop 8930
Gaithersburg, MD 20899-8930
USA

EMail: quynh.dang@nist.gov

Sean Turner
IECA, Inc.
3057 Nutley Street, Suite 106
Fairfax, VA 22031 USA

EMail: turners@ieca.com