SM3 Hash function
draft-shen-sm3-hash-01

Abstract

This document describes a hash function which is invented by Xiaoyun Wang et al. This algorithm is published by Chinese Commercial Cryptography Administration Office ([SM3]) for the use of electronic authentication service system. This document gives IETF standard description of the algorithm.

Status of This Memo

This Internet-Draft is submitted to IETF 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 http://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 August 18, 2014.

Copyright Notice

Copyright (c) 2014 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 ........................................... 3  
2. Conventions Used in this Document .................. 3  
3. Algorithm ............................................. 3  
   3.1. Scope of SM3 ...................................... 3  
   3.2. Definitions and Terms ............................ 3  
   3.3. Symbols ........................................... 4  
      3.3.1. Iteration Procedure ......................... 4  
      3.3.2. Message Extension ......................... 4  
      3.3.3. Compression function ....................... 4  
      3.3.4. Hash Value ................................ 5  
4. IANA Considerations .................................... 5  
5. Security Considerations ............................... 5  
6. References ............................................ 5  
   6.1. Normative References ............................ 5  
   6.2. Informative References .......................... 5  
Appendix A. Appendix A. Example 1 ..................... 6  
Appendix B. Appendix A. Example 2 ..................... 8  
Appendix C. Appendix C. Acknowledgments ............... 13
1. Introduction

The algorithm described in this document is published ([SM3]) by Chinese Commercial Cryptography Administration Office. This document gives IETF standard description of these algorithms and parameters.

2. Conventions Used in this Document

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", and "MAY" in this document are to be interpreted as defined in "Key words for use in RFCs to Indicate Requirement Levels" [RFC2119].

3. Algorithm

This chapter introduced the algorithm itself. The content and structure strictly follow what is published by Chinese Commercial Cryptography Administration Office.

3.1. Scope of SM3

This document defines SM3 Hash algorithms and gives computing examples. It can be used in commercial cryptography applications like digital signature and verification, message authentication code and verification, random number generation. This document can also be technical reference of security product and standards in order to improve the trustness and interoperability.

3.2. Definitions and Terms

Bit String  String composed of 0 and 1

Big-endian  A format that describe the order in which data are stored in computer memory. It defines that the bytes with most significant value are stored at the left and bytes with least significant value are stored at the right. The high digits of a number are stored at high storage address and the low digits of a number are stored at low storage address.

Message     Bit string with arbitrary length. In this document a message is consider the input of the hash algorithm.

Hash Value  The bit string which is the output after the hash algorithm applied to a message. The length of the hash value in this document is 256 bits.
3.3. Symbols

3.3.1. Iteration Procedure

The procedures are: Divide the message \( m' \) after padding into 512 bits blocks:

\[
m' = B(0)B(1) ... B(n-1) \quad \text{where } n = (l+k+65)/512
\]

Apply iteration operation to \( m' \) as following:

\[
\text{FOR } i=0 \text{ TO } n-1 \\
\quad V(i+1) = CF(V(i);B(i)) \\
\text{ENDFOR}
\]

where \( CF \) is compression function, \( V(0) \) is a 256 bits of IVAGBP[not]

\( B(i) \) is a message block after padding, the result after iterative compression is \( V(n) \).

3.3.2. Message Extension

Divide the message block \( B(i) \) into 132 words, apply the words into the compression function:

a) divide message block \( B(i) \) into 16 words \( W_0, W_1, ..., W_{15} \).

b) \text{FOR } j=16 \text{ TO } 67 \\
\quad W_j \leftarrow P1(WAGBP(A_j-16AGBP(A)) XOR WAGBP(A_j-9AGBP(A)) XOR (WAGBP(A_j-3AGBP(A))SHIFT15)) \\
\quad \text{XOR} \ WAGBP(A_j-13AGBP(A)SHIFT7) \text{ XOR} \ WAGBP(A_j-6AGBP(A))
\text{ENDFOR}

c) \text{FOR } j=0 \text{ TO } 63 \\
\quad W_j~ = W_j \text{ XOR} \ W(j+4)
\text{ENDFOR}

3.3.3. Compression function

Let \( A,B,C,D,E,F,G,H \) be registers to store words; \( SS1, SS2, TT1 \) and \( TT2 \) be intermediate variable; compression function:

\[
V(i+1) = CF(V(i);B(i)) \quad \text{where } 0 \leq i \leq n
\]

The computation procedures are as following:
ABCDEF GH < -- V(i)
FOR j=0 TO 63
    SS1 < -- ((A SHIFT12) + E + (Tj SHIFTj))??
    SS2 < -- SS1 XOR (A SHIFT12)
    TT1 < -- FFj(A,B,C) + D + SS2 +Wj~
    TT1 < -- GGj(E,F,G) + H + SS1 +Wj
    D < -- C
    C < -- B SHIFT9
    B < -- A
    A < -- TT1
    H < -- G
    G < -- F SHIFT19
    F < -- E
    E < -- F0(TT2)
ENDFOR
V(i+1) < -- ABCDEF GH XOR V (i)

where a word is stored in memory as big-endian format.

3.3.4. Hash Value

ABCDEF GH < -- V(n)

The 256 bits of hash value is y=ABCDEFG H.

4. IANA Considerations

There is no iana related issue for this document.

5. Security Considerations

This document gives description of a cryptographic algorithm. Its security properties are under public study. There are no known feasible attacks against this algorithm by the time of publishing this document.

6. References

6.1. Normative References

6.2. Informative References

Appendix A.  Appendix A.  Example 1

The input is "abc", which has ASCII expression:

616263

The message after padding is:

61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000018

The message after extension is:

W0W1...W67
61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000 00000018 9092e200 00000000 000c0606
719c70ed 00000000 8001801f 939f7da9 00000000 2c6fa1f9 adaef14 00000000
0001801e 9a965f89 49710048 23ce86a1 b2d12f1b e1da3ee8 f8061807 055d68be
86cfd481 1f447d83 d9023dbf 185898e0 e0061807 050df55c cde0104c a5b9c955
a7df0184 6e46cd0b e3babdf8 070464a7 0353af50 a92d6ca1 5f33cfd2 e1f6f6e9
f70fe941 ca5462dc 85a90152 76af6296 c922d6b2 6837c8f5 97585344 09008723
86faee74 2ab908b0 4a64bc50 8646e608 f07e6590 325c8f78 accb8011 e11db9dd
b99c0545
W0'W1'...W63'

The intermediate values during iterative compression are:

<table>
<thead>
<tr>
<th>j</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7380166f</td>
<td>49142b89</td>
<td>172442d7</td>
<td>da8a0600</td>
<td>a96f30bc</td>
<td>163138aa</td>
<td>e38de4e4</td>
<td>b0fb0e4e</td>
</tr>
<tr>
<td>1</td>
<td>ea52428c</td>
<td>b9edc12b</td>
<td>002cdee7</td>
<td>29657292</td>
<td>ac35a23</td>
<td>b2ad29f4</td>
<td>85e54b79</td>
<td>c550b189</td>
</tr>
<tr>
<td>2</td>
<td>609f2850</td>
<td>ea52428c</td>
<td>db825773</td>
<td>002cdee7</td>
<td>d33ad5fb</td>
<td>ac35a23</td>
<td>4fa59569</td>
<td>85e54b79</td>
</tr>
<tr>
<td>3</td>
<td>35037e59</td>
<td>609f2850</td>
<td>a48519d4</td>
<td>db825773</td>
<td>b8204b5f</td>
<td>d33ad5fb</td>
<td>d11d61a9</td>
<td>4fa59569</td>
</tr>
<tr>
<td>4</td>
<td>1f995766</td>
<td>35037e59</td>
<td>3e50a0c1</td>
<td>a48519d4</td>
<td>8ad212ea</td>
<td>b8204b5f</td>
<td>afde99d6</td>
<td>d11d61a9</td>
</tr>
<tr>
<td>5</td>
<td>374a0ca7</td>
<td>1f995766</td>
<td>06fcb26a</td>
<td>3e50a0c1</td>
<td>acf0f639</td>
<td>8ad212ea</td>
<td>5afdc102</td>
<td>afde99d6</td>
</tr>
<tr>
<td>6</td>
<td>33130100</td>
<td>374a0ca7</td>
<td>32aecc3f</td>
<td>06fcb26a</td>
<td>3391ec8a</td>
<td>afcf0f639</td>
<td>97545690</td>
<td>5afdc102</td>
</tr>
</tbody>
</table>
The hash value is:

66c7f0f4 62eeedd9 d1f2d46b dc10e4e2 4167c487 5cf2f7a2 297da02b 8f4ba8e0

Appendix B. Appendix A. Example 2

A message of 512 bits:

61626364 61626364 61626364 61626364 61626364 61626364 61626364 61626364
61626364 61626364 61626364 61626364 61626364 61626364 61626364 61626364

The message after padding is:

61626364 61626364 61626364 61626364 61626364 61626364 61626364 61626364
61626364 61626364 61626364 61626364 61626364 61626364 61626364 61626364
80000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
80000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

The first message block:
The message after padding:

\[
\begin{align*}
\text{W0W1...W67} & \quad 61626364 61626364 61626364 61626364 61626364 61626364 61626364 61626364 \\
\text{a121a024 a121a024 a121a024 e0dfe567 e0dfe567 e0dfe567 e0dfe567 e0dfe567} & \quad \text{61626364 61626364 61626364 61626364} \\
\text{0e07191c 2f8a37a d528936a 89fbd8ae 0e07191c 2f8a37a d528936a 89fbd8ae} & \quad \text{6a12906 e97f7e9 3c78b9f9} \\
\text{c22b8a69 9f03f169 df45be20 9ec5beef 0a12906 e97f7e9 3c78b9f9} & \quad \text{6a12906 e97f7e9 3c78b9f9} \\
\text{6e9fa333 2ebae676 3475c386 201dcfcf 2c5f2b5c a809f3f8 bc139f34} & \quad \text{6a12906 e97f7e9 3c78b9f9} \\
\text{c47f18a7 a25ce71d 42743705 51baf619} & \quad \text{6a12906 e97f7e9 3c78b9f9} \\
\text{W0'W1'...W63'} & \quad \text{6a12906 e97f7e9 3c78b9f9}
\end{align*}
\]

The message after extension is:

\[
\begin{align*}
\text{W0W1...W67} & \quad 61626380 00000000 00000000 00000000 00000000 00000000 00000000 00000000 \\
\text{e0fba1b be6fca1b 32697922 bfa99c9f 5f29394f 03fa728b} & \quad \text{6a12906 e97f7e9 3c78b9f9} \\
\text{a97ed93 b5836157 cc44be6f 85f33eaf 3abac0d9 a2b0d71e} & \quad \text{6a12906 e97f7e9 3c78b9f9} \\
\text{c60aa36f d6fc83a9 9934c61 f92524f8 64d8a35 7645946b 7024b1c7 47fd55ef} & \quad \text{6a12906 e97f7e9 3c78b9f9} \\
\text{41e25ffe 025cd2a 9c7e5cbe 9c0e5c02 eb67e468 8e03cc41 ea7fa83d eda9692d} & \quad \text{6a12906 e97f7e9 3c78b9f9}
\end{align*}
\]

The intermediate values during iterative compression are:

\[
\begin{align*}
\text{j} & \quad \text{A} & \quad \text{B} & \quad \text{C} & \quad \text{D} & \quad \text{E} & \quad \text{F} & \quad \text{G} & \quad \text{H} \\
\text{Shen & Lee} & \quad \text{Expires August 18, 2014} & \quad [\text{Page 9}]
\end{align*}
\]
The second message block:

The message after padding:

W0W1...W67

W0'W1'...W63'

The intermediate values during iterative compression are:

<table>
<thead>
<tr>
<th>j</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5950de81</td>
<td>468646eb</td>
<td>42fd4c86</td>
<td>1e7ca00a</td>
<td>c0a5910b</td>
<td>ae9a55ea</td>
<td>1adb8d17</td>
<td>763ca222</td>
</tr>
<tr>
<td>1</td>
<td>01cc66027</td>
<td>5950de81</td>
<td>0cc9d68d</td>
<td>42fd4c86</td>
<td>24fe81a1</td>
<td>c0a5910b</td>
<td>af5574d2</td>
<td>1adb8d17</td>
</tr>
</tbody>
</table>
The hash value is:

debe9ff9 2275b8a1 38604889 c18e5a4d 6f6db70e5 387e5765 293dcba3 9c0c5732

Appendix C. Acknowledgments

TBD

Authors’ Addresses

Sean Shen (editor)
Chinese Academy of Science
No.4 South 4th Zhongguancun Street
Beijing, 100190
China

Phone: +86 10-58813038
EMail: shenshuo@cnnic.cn

Xiaodong Lee (editor)
Chinese Academy of Science
No.4 South 4th Zhongguancun Street
Beijing, 100190
China

Phone: +86 10-58813038
EMail: shenshuo@cnnic.cn