A Minor Change to Lesamnta
— Change of Round Constants —

Shoichi HIROSE¹, Hidenori KUWAKADO², Hirotaka YOSHIDA³,⁴

¹ University of Fukui
hrs_shch@u-fukui.ac.jp
² Kobe University
kuwakado@kobe-u.ac.jp
³ Systems Development Laboratory, Hitachi, Ltd.,
hirotaka.yoshida.qv@hitachi.com
⁴ Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC

1 Introduction

Lesamnta is a new family of hash functions submitted to NIST for their cryptographic hash algorithm competition.

In this document, we propose to make a minor change to the specification of Lesamnta by changing round constants. We give a short overview of the minor change to the Lesamnta hash function. We divide our arguments into the following categories:

- The minor change of the Lesamnta hash function
- The motivation of the change
- The design principle for the round constants
- Security against potential attacks on Lesamnta with new round constants
- The impact of the change

Note that we mainly explain about the minor change in the case of Lesamnta-256 because the explanation could be easily adapted to the case of Lesamnta-224/384/512.

2 The Minor Change

We propose to make a minor change to the specification of Lesamnta [2]. The minor change is only the replacement of 32 round constants. No other parts of the specification is changed.
For Lesamnta-224/256, we replace the 32 round constants described in Section 5.1.1 (page 14) of [2] with the following constants.

Round constants of [2]

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0000000100000000 0000000000000000</td>
</tr>
<tr>
<td>1</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>2</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>3</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>4</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>5</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>6</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>7</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>8</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>9</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>10</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>11</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>12</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>13</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>14</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>15</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>16</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>17</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>18</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>19</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>20</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>21</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>22</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>23</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>24</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>25</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>26</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>27</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>28</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>29</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>30</td>
<td>0000000000000000 0000000000000000</td>
</tr>
<tr>
<td>31</td>
<td>0000000000000000 0000000000000000</td>
</tr>
</tbody>
</table>

New round constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>9e75470089cfedb 2db4ad503bb6f80 02db4ad503bb6f80 e1a70c522758bc4b 2a4989e511412ba9 1e95cf81bff8729e a8c416470af5c6d6 422bb32416c61cb6 4c85497227052110 04c85497227052110 fdf76aa9eba86421 f264994a0735e742</td>
</tr>
<tr>
<td>1</td>
<td>3744e7ab7dbae34 6f80451ae2875955 8b86b7ce8c169407 bda476dc1727489b 2f89be4def245d4de4 723dc79b6495eddc 966c38f97a9bf66f 2d353aa49d1db</td>
</tr>
</tbody>
</table>
New round constants

\begin{verbatim}
f6251864809494cd35cb7fa305acbe7f 78b114d45c0c00375778b114d45c0c00375757aa6c4b9d98f1bf
b508148e2c0e4608026cd2af27a24b0 ba220a9a4170d2de29fdd68d717f83f4
fa8e8475315428a0c9d29ba4c07bc9f 97fc92f852b9c3860d30da783d3f6b9d
95b68b70b22784abc19a58a8ca71e4c 48abbc03a0a7ff77422b58cddfda9ca
7c75fa0d1976cfcbfbd178c3b7e94af7 f9c7bddd46d083fdeb7b7be158c6d9d
dfc1d1492ddc08b5635cc6c7e5be34 37df3f822ec2133f52f774280cbb7e2
ed519add8adb45eaee57ed138887b7e1 eebfc9e5f470099f492d2f7871392104
159b340651e246363b85e6fe008b602c 2eb05b97b586d5603e449f66e83f515
155b3b9423a3b0eaa2f970408e7011c9 c4ac4d4d5f51d7e0cb6c807b1a503ca
c749d85c10030a936a9e3eeb3c873d5d 58d1aa49ef6a3f3a0ccfcec6d0dc475a
5f3436b7343bca903289d46dd90e26d9a a27d71f052fa6d323a61c086f06e116
17f09d2029b961fe360d4014031eb9db d7b2481063efc7686a41ae3d098b4854
514f4ad91b6c61cf873589388b8db4 b8e89afa85ebc47add66113773567d0b5
05e3eaf69155b31c8e13ac1129135b54 519d1be862b6d9876253678b1a4981a7
ac87ca0b822b705d736ec2f621c7828 2a47905563e447589bf95e6fed53f800
002a47905563e447589bf95e6fed53f8 f6e7f57d574abc562f1ea392b7fb35b
\end{verbatim}

The minor change of Lesamnta is only the above replacement. We will describe how to produce the new round constants in Sect. 4.2.

3 The Motivation for the Minor Change

The motivation why we propose to make a minor change to the specification of Lesamnta by changing round constants is to prevent the attacks described in [3] and one newly-discovered attack which will be described in Appendix A.1. These attacks are summarized in the following:

- **Distinguishing attack on Lesamnta's block cipher [3]**
  - An adversary can distinguish between Lesamnta-256's block cipher and the ideal cipher by making only two queries.

- **Pseudo-collision attack on Lesamnta [3]**
  - An adversary can produce a pseudo-collision of Lesamnta-256 with $O(2^{64})$ computations of the compression function.

- **Semi-free start collision attack on Lesamnta [4]**
  - An adversary can produce a semi-free start collision of Lesamnta-256 with $O(2^{64})$ computations of the compression function. It is a kind of pseudo-collision attack mentioned above.

- **Attack using weak messages on Lesamnta**
  - An adversary can find a kind of second preimage of Lesamnta-256 with $O(2^{128})$ computations of the compression function if the target message satisfies a certain property.

We observe that all the above attacks are based on some symmetry in the key scheduling function and the message mixing function of Lesamnta. To destroy this symmetry, we have chosen the new round constants. We expect that all the above attacks do not work any more for Lesamnta with the new round constants.
4 Design Principle for the New Round Constants

4.1 Condition for New Round Constants

We here show the condition allowing attacks described in Sect. 3. Since all the attacks use it, they fail if it does not hold. We have confirmed that the new round constants do not satisfy the condition. The relationship between conditions and attacks are described in Appendix.

Let \( C[r][0] \) and \( C[r][1] \) be the left part and the right part of the \( r \)-th round constant in the key scheduling function of \( \text{EncComp}_{256} \) or \( \text{EncComp}_{512} \) (see Figs. 18, 28 of [2]). Note that \( C[r][i] \) is a 32-bit string for \( \text{EncComp}_{256} \) and it is a 64-bit string for \( \text{EncComp}_{512} \). We define a difference \( \Delta_r \) as

\[
\Delta_r = C[r][0] \oplus C[r][1]
\]

for \( r = 0, 1, \ldots, 31 \). The condition is to satisfy all the following equations:

\[
\begin{align*}
\Delta_0 &= \Delta_4 = \Delta_8 = \Delta_{12} = \cdots = \Delta_{24} = \Delta_{28}, \\
\Delta_1 &= \Delta_5 = \Delta_9 = \Delta_{13} = \cdots = \Delta_{25} = \Delta_{29}, \\
\Delta_2 &= \Delta_6 = \Delta_{10} = \Delta_{14} = \cdots = \Delta_{26} = \Delta_{30}, \\
\Delta_3 &= \Delta_7 = \Delta_{11} = \Delta_{15} = \cdots = \Delta_{27}.
\end{align*}
\]

The condition is used for distinguishing Lesaman’s block ciphers from ideal ciphers. Furthermore, the distinguishing attack can be extended to the pseudo-collision attack and the weak-message attack.

4.2 Generators of New Round Constants

To be free of suspicion of a trapdoor, round constants must be determined in a transparent way. The new round constants for Lesaman-256 were determined by the algorithm of Fig. 1. The algorithm of Fig. 1 is based on the linear feedback shift register (LFSR) of the following primitive polynomial \( g(x) \).

\[
g(x) = x^{64} + x^{61} + x^{58} + x^{55} + x^{47} + x^{46} + x^{42} + x^{41} + x^{39} + x^{38} \\
+ x^{37} + x^{35} + x^{34} + x^{33} + x^{31} + x^{30} + x^{29} + x^{28} + x^{27} + x^{26} \\
+ x^{25} + x^{24} + x^{20} + x^{19} + x^{18} + x^{16} + x^{14} + x^{12} + x^{8} + x^{7} \\
+ x^{2} + x^{1} + 1.
\]

Due to 33 non-zero coefficients, almost half of bits of the internal state may be changed by one operation. Since there are many 33-term primitive polynomials, we adopted the first 33-term primitive polynomial obtained from the decimation of the M sequence produced by the LFSR of \( x^{64} + x^{63} + x^{61} + x^{60} + 1 \) with the all-one initial state.

Since the round constants \( C[r] \) can be considered as elements of \( \text{GF}(2^{64}) \), the above algorithm is equivalent to

\[
C[r] = C[r - 1] \ast \alpha^r \text{ over } \text{GF}(2^{64}) \text{ for } r = 1, 2, \ldots, 31
\]
\begin{verbatim}
ConstantGenerator256(word C[Nr_comp256]) /* Nr_comp256=32 */
begin
word c = fffffffffffff /* in hexadecimal */
for i = 0 to Nr_comp256*J-1 /* J = 4 */
   /* Galois-type LFSR */
   if c ∧ 0000000000000001 = 0000000000000001
      c = (c >> 1) ⊕ e18ab8ff77630124
   else
      c = c >> 1
   end if
if i mod J = 0
   C[i/J] = c
end if
end for
end
\end{verbatim}

Fig. 1. Pseudocode for generating round constants of Lesamna-224/256.

where α is a root of \( g(x) \). We chose \( J = 4 \). If \( J \leq 3 \), then there exists an initial state such that almost all \( C[\cdot] \)'s satisfy Condition 1.

The new round constants for Lesamnta-384/512 were determined in a similar manner. Specifically, the following 65-term primitive polynomial \( g(x) \) and \( J = 8 \) were used.

\[
g(x) = x^{128} + x^{124} + x^{121} + x^{120} + x^{119} + x^{117} + x^{116} + x^{114} + x^{112} + x^{111} + x^{110} + x^{107} + x^{106} + x^{105} + x^{104} + x^{103} + x^{101} + x^{100} + x^{98} + x^{97} + x^{95} + x^{94} + x^{93} + x^{92} + x^{91} + x^{90} + x^{89} + x^{87} + x^{86} + x^{84} + x^{82} + x^{81} + x^{79} + x^{78} + x^{76} + x^{74} + x^{73} + x^{70} + x^{69} + x^{66} + x^{64} + x^{63} + x^{60} + x^{58} + x^{57} + x^{54} + x^{53} + x^{51} + x^{48} + x^{39} + x^{37} + x^{36} + x^{35} + x^{32} + x^{31} + x^{30} + x^{29} + x^{28} + x^{26} + x^{23} + x^{21} + x^{18} + x^{17} + x^{15} + x^{9} + x^{8} + 1
\]

The pseudocode for generating the round constants is shown in Fig. 2. Since there are many 65-term primitive polynomials, we adopted the first 65-term primitive polynomial obtained from the decimation of the M sequence produced by the LFSR of \( x^{128} + x^{127} + x^{126} + x^{121} + 1 \) with the all-one initial state.

5 Security against Potential Attacks on Lesamnta with New Round Constants

We here consider a hash function family D-Lesamnta which is the same as Lesamnta except that the round constants \( C[\cdot] \) are replaced by some D[\cdot]. We here present two potential distinguishing attacks on this Lesamnta-like hash function family. These two attacks can not work for Lesamnta with new round constants.
ConstantGenerator512(word C|Nr_comp512 |) /* Nr_comp512=32 */
begin
word c = ffffffffffffffffffffffffffffffff
for i = 0 to Nr_comp512∗J−1 /* J = 8 */
 /* Galois-type LFSR */
if c ∧ 00...01 = 00...01
  c = (c >> 1) ⊕ 89dae79b7f66632ca34805cfa534180
else
  c = c >> 1
end if
if i mod J = 0
  C[i/J] = c
end if
end for
end

Fig. 2. Pseudocode for generating round constants of Lesamna-384/512.

**Condition 1** Let \( a = a_0∥a_1∥ ...∥a_7 \) and \( b = b_0∥b_1∥ ...∥b_7 \), where \( a_i, b_i \in \{0,1\}^8 \). Let \( \text{tp}_\ell(a) = b \) be a byte-transposition such that \( b_i = a_{i+2\ell \mod 8} \) for \( 0 \leq \ell \leq 3 \). Let \( \text{rv}(a) = b \) be a byte-transposition such that \( b_i = a_{7-i} \).

We have a distinguishing attack on the underlying block cipher of the \( D \)-Lesamnta-256 output function if \( D \) satisfies the following condition.

Let \( D[r] \) be the 64-bit \( r \)-th round constant in the key scheduling function of EncOut256. \( D[r] \) is considered as an 8-byte data, that is,

\[
D[r] = D[r](0) || D[r](1) || ... || D[r](7),
\]

where \( D[r](i) \in \{0,1\}^8 \). We define a 64-bit difference \( A_r \) as

\[
A_r = D[r] \oplus \Pi(D[r])
\]

for \( r = 0, 1, ..., 31 \), where \( \Pi \) is any composition of \( \text{tp}_\ell \) and \( \text{rv} \) but \( \Pi \neq \text{tp}_0 \).

The condition is to satisfy the following equations:

\[
A_k = A_{k+4} \quad \text{for } 0 \leq k \leq 26.
\]

Notice that round constants of EncOut256 are identical to those of EncComp256 (see Fig. 20 of [2]).

Since the new round constants for Lesamnta-256 do not satisfy the above condition, Lesamnta-256 is secure against the above attack.

**Condition 2** Let \( s = (s_{i,j}) \) and \( t = (t_{i,j}) \), where \( s_{i,j}, t_{i,j} \in \{0,1\}^8 \) and \( 0 \leq i, j \leq 3 \). Let \( \pi_\ell(s) = t \) be a byte-transposition such that \( t_{i,j} = s_{i+t \mod 4,j} \) for \( 0 \leq \ell \leq 3 \). Let \( \varpi_\ell(s) = t \) be a byte-transposition such that \( t_{i,j} = s_{i,j+t \mod 4} \) for \( 0 \leq \ell \leq 3 \).
We have a distinguishing attack on the underlying block cipher of the D-Lesamnta-512 output function if $D$ satisfies the following condition.

Let $D[r]$ be the 128-bit $r$-th round constant in the key scheduling function of $EncOut_{512}$. $D[r]$ is considered as a 16-byte data, that is,

$$D[r] = D[r](0) \parallel D[r](1) \parallel \ldots \parallel D[r](15),$$

where $D[r](i) \in \{0, 1\}^8$. We see $D[r] = (D[r]_{i,j})$, where $D[r]_{i,j} = D[r](i + 4j)$ for $0 \leq i, j \leq 3$. We define a 128-bit difference $\Xi_r$ as

$$\Xi_r = D[r] \oplus \Pi(D[r])$$

for $r = 0, 1, \ldots, 31$, where $\Pi$ is any composition of $\pi_\ell$ and $\varpi_\ell'$ but $\Pi$ is not an identity transposition. The condition is to satisfy the following equations:

$$\Xi_k = \Xi_{k+4} \quad \text{for} \quad 0 \leq k \leq 26.$$

Notice that round constants of $EncOut_{512}$ are identical to those of $EncComp_{512}$ (see Fig. 28 of [2]).

Since the new round constants for Lesamnta-512 do not satisfy the above condition, Lesamnta-512 is secure against the above attack.

6 The Impact of the Change on Lesamnta

6.1 Impact on the Security of Lesamnta

We observe that the change does not cause any impact on resistance against the known attacks on Lesamnta described in [2].

We expect that Lesamnta with the new round constants prevents the attacks described in Sect. 3 which violated the assumptions of the security proofs in the ideal cipher model in [2]. Therefore we believe that these security proofs for Lesamnta would become more meaningful if the round constants are changed to the new ones.

6.2 Impact on the Performance of Lesamnta

For speed-optimized implementations of Lesamnta, the change does not cause any impact on its performance because the round constants are stored in a table.

For area-optimized implementations of Lesamnta, the change may slightly increase the required memory size because on-the-fly generation of round constants may be less useful than storing them in a table, due to the relatively large number of terms in the feedback polynomial used in the LFSR generating them. However, based on our estimation, we expect that storing them in a table does not cause any problem in real applications.
7 Concluding Remarks

We propose to make a minor change to the specification of Lesamnta by changing round constants. The motivation of this change is to prevent the attacks described in [3] and the newly-discovered attack described in this document. We expect that all of these attacks do not work any more for Lesamnta with the new round constants. We also expect that the change does not cause any significant impact on resistance against the known attacks and on performance of Lesamnta.

Acknowledgments

We would like to thank Charles Bouillaguet, Orr Dunkelman, Gaëtan Leurent, Pierre-Alain Fouque for their excellent analysis on Lesamnta. We would like to thank Mridul Nandi for improving the analysis. We would like to mention the people who gave us feedback and important comments on this work: Kota Ideguchi, Yasuko Fukuzaawa, Toru Owada, Bart Preneel. This work was partially supported by the National Institute of Information and Communications Technology, Japan.

References


8 List of Annexes

A The Attack Methods

A.1 Attack Using Weak Messages on Lesamnta

The pseudo-collision-finding attack might be applicable to an attack using weak messages. The primary idea of this attack was shown in [1].

Consider Lesamnta-256. The algorithm of an adversary that finds a second preimage is described below.
1. Suppose that a $t$-block message and its digest are given. The $t$-block message is denoted by $mb^{(0)} \parallel mb^{(1)} \parallel \ldots \parallel mb^{(t-1)}$ where $mb^{(i)}$ is a 256-bit message block.

2. For $i = 0, 1, \ldots, t-1$, compute $chain^{(i)}$ as

$$chain^{(i)} = Compression256(chain^{(i-1)}, mb^{(i)}),$$

where $chain^{(-1)}$ is the standard initial value.

3. If there is an index $i_1$ such that

$$chain^{(i_1)}[j] = chain^{(i_1)}[j+1]$$

for $j = 0, 2, 4, 6$, go to the next step. Otherwise output fail.

4. Find $mb^{(i_1-1)}$ and $mb^{(i_1)}$ such that

$$chain^{(i_1-1)} = Compression256(chain^{(i_1-2)}, mb^{(i_1-1)}),$$
$$chain^{(i_1)} = Compression256(chain^{(i_1-1)}, mb^{(i_1)}),$$

by using the attack described in [3].

The probability that the condition in step 3 is satisfied is $t/2^{128}$. Since Lesamnta-256 accepts a $(2^{64} - 1)$-bit message at most, the probability is less than $2^{-72}$. We call a message satisfying the condition in step 3 a weak message. We notice that this attack is effective only when a given message is a weak message.