# Security Analysis of the Compression Function of Lesamnta and its Impact

Shoichi HIROSE[1], Hidenori KUWAKADO[2], Hirotaka YOSHIDA[3,4]

[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.

A security analysis of the compression function of Lesamnta has been reported [1]. In this document, we give a short overview of how this analysis affects the security of the full Lesamnta hash function. We divide our arguments into three categories:

- A security analysis of the Lesamnta compression function
- The impact of the security analysis on the security of the full Lesamnta
- A plan for a minor change to the specification

## 2 A Security Analysis of the Compression Function

### 2.1 Observation on Lesamnta's Block Cipher

This section describes a correlation among a key, a plaintext, and a ciphertext in Lesamnta's block cipher. The correlation was discovered by Bouillaguet *et al.* [1]. We only describe the observation on Lesamnta-256, but we can obtain similar observation on Lesamnta-512; the difference is just word size.

We follow symbols and notations of [2] and consider Lesamnta-256's block cipher $EncComp_{256}$. Let $\mathtt{C}[r][0]$ and $\mathtt{C}[r][1]$ be the left part and the right part of the $r$-th round constant in the key schedule function (see Figure 18 of [2]). For example, $\mathtt{C}[0][0] = \mathtt{00000001}$ and $\mathtt{C}[0][1] = \mathtt{00000000}$ according to p.14 of [2]. We define a difference $\Delta_r$ as

$$\Delta_r = \mathtt{C}[r][0] \oplus \mathtt{C}[r][1] \tag{1}$$

for $r = 0, 1, \ldots, 31$. According to p.14 of [2], we see that the following equations hold.

$$\Delta_0 = \Delta_4 = \Delta_8 = \Delta_{12} = \ldots = \Delta_{24} = \Delta_{28},$$

$$\begin{aligned}
&\Delta_1 = \Delta_5 = \Delta_9 = \Delta_{13} = \ldots = \Delta_{25} = \Delta_{29}, \\
&\Delta_2 = \Delta_6 = \Delta_{10} = \Delta_{14} = \ldots = \Delta_{26} = \Delta_{30}, \\
&\Delta_3 = \Delta_7 = \Delta_{11} = \Delta_{15} = \ldots = \Delta_{27}.
\end{aligned} \tag{2}$$

Precisely speaking, we also see $\Delta_0 = \Delta_1 = \Delta_2 = \Delta_3 = \mathsf{00000001}$ and $\Delta_3 = \Delta_{31}$, but these properties are unnecessary for the following discussion. We will see that the relation of Eq. (2) allows an adversary to attack Lesamnta.

A key `chain`, which corresponds to `chain`[8] in Figure 18 of [2], is denoted by $\mathtt{chain}[0] \parallel \mathtt{chain}[1] \parallel \ldots \parallel \mathtt{chain}[7]$ where $\mathtt{chain}[i] \in \{0,1\}^{32}$. The key schedule function produces 32 round keys $\mathtt{K}[0][0]||\mathtt{K}[0][1], \ldots, \mathtt{K}[31][0]||\mathtt{K}[31][1]$ from the key.

**Proposition 1.** *Let* $\mathtt{chain}_0$ *be any key* $\mathtt{chain}_0[0] \parallel \mathtt{chain}_0[1] \parallel \ldots \parallel \mathtt{chain}_0[7]$. *Suppose that another key* $\mathtt{chain}_1$ *is determined as*

$$\begin{aligned}
\mathtt{chain}_1 = &(\mathtt{chain}_0[1] \oplus \Delta_2) \parallel (\mathtt{chain}_0[0] \oplus \Delta_2) \parallel (\mathtt{chain}_0[3] \oplus \Delta_1) \parallel (\mathtt{chain}_0[2] \oplus \Delta_1) \\
&\parallel (\mathtt{chain}_0[5] \oplus \Delta_0) \parallel (\mathtt{chain}_0[4] \oplus \Delta_0) \parallel (\mathtt{chain}_0[7] \oplus \Delta_3) \parallel (\mathtt{chain}_0[6] \oplus \Delta_3).
\end{aligned}$$

*When round keys* $\mathtt{K}_0$ *generated from* $\mathtt{chain}_0$ *are denoted by*

$$\mathtt{K}_0[0][0]||\mathtt{K}_0[0][1], \mathtt{K}_0[1][0]||\mathtt{K}_0[1][1], \ldots, \mathtt{K}_0[31][0]||\mathtt{K}_0[31][1],$$

*round keys* $\mathtt{K}_1$ *generated from* $\mathtt{chain}_1$ *are given by*

$$\begin{aligned}
&\mathtt{K}_1[4i][0] = \mathtt{K}_0[4i][1] \oplus \Delta_2, && \mathtt{K}_1[4i][1] = \mathtt{K}_0[4i][0] \oplus \Delta_2, \\
&\mathtt{K}_1[4i+1][0] = \mathtt{K}_0[4i+1][1] \oplus \Delta_3, && \mathtt{K}_1[4i+1][1] = \mathtt{K}_0[4i+1][0] \oplus \Delta_3, \\
&\mathtt{K}_1[4i+2][0] = \mathtt{K}_0[4i+2][1] \oplus \Delta_0, && \mathtt{K}_1[4i+2][1] = \mathtt{K}_0[4i+2][0] \oplus \Delta_0, \\
&\mathtt{K}_1[4i+3][0] = \mathtt{K}_0[4i+3][1] \oplus \Delta_1, && \mathtt{K}_1[4i+3][1] = \mathtt{K}_0[4i+3][0] \oplus \Delta_1,
\end{aligned}$$

*for* $i = 0, 1, \ldots, 7$.

Next, consider the mixing function of the block cipher $EncComp_{256}$ (see Figure 11 of [2]). A message block `mb` is denoted by $\mathtt{mb}[0] \parallel \mathtt{mb}[1] \parallel \ldots \parallel \mathtt{mb}[7]$ where $\mathtt{mb}[i] \in \{0,1\}^{32}$. Let $\mathtt{K}[0][0]||\mathtt{K}[0][1], \ldots, \mathtt{K}[31][0]||\mathtt{K}[31][1]$ be 32 round keys generated by the key schedule function. The output `x` of $EncComp_{256}$ (i.e., the ciphertext) is denoted by $\mathtt{x}[0] \parallel \mathtt{x}[1] \parallel \ldots \parallel \mathtt{x}[7]$.

**Proposition 2.** *Let* $\mathtt{K}_0[0][0]||\mathtt{K}_0[0][1], \ldots, \mathtt{K}_0[31][0]||\mathtt{K}_0[31][1]$ *be 32 round keys* $\mathtt{K}_0$, *and let* $\mathtt{mb}_0[0] \parallel \mathtt{mb}_0[1] \parallel \ldots \parallel \mathtt{mb}_0[7]$ *denote a message block* $\mathtt{mb}_0$. *Suppose that round keys* $\mathtt{K}_1$ *and a message block* $\mathtt{mb}_1$ *satisfy the following equations.*

$$\begin{aligned}
&\mathtt{K}_1[4i][0] = \mathtt{K}_0[4i][1] \oplus \delta_0, && \mathtt{K}_1[4i][1] = \mathtt{K}_0[4i][0] \oplus \delta_0, \\
&\mathtt{K}_1[4i+1][0] = \mathtt{K}_0[4i+1][1] \oplus \delta_1, && \mathtt{K}_1[4i+1][1] = \mathtt{K}_0[4i+1][0] \oplus \delta_1, \\
&\mathtt{K}_1[4i+2][0] = \mathtt{K}_0[4i+2][1] \oplus \delta_2, && \mathtt{K}_1[4i+2][1] = \mathtt{K}_0[4i+2][0] \oplus \delta_2, \\
&\mathtt{K}_1[4i+3][0] = \mathtt{K}_0[4i+3][1] \oplus \delta_3, && \mathtt{K}_1[4i+3][1] = \mathtt{K}_0[4i+3][0] \oplus \delta_3, \\
&\mathtt{mb}_1[0] = \mathtt{mb}_0[1] \oplus \delta_2, && \mathtt{mb}_1[1] = \mathtt{mb}_0[0] \oplus \delta_2, \\
&\mathtt{mb}_1[2] = \mathtt{mb}_0[3] \oplus \delta_1, && \mathtt{mb}_1[3] = \mathtt{mb}_0[2] \oplus \delta_1, \\
&\mathtt{mb}_1[4] = \mathtt{mb}_0[5] \oplus \delta_0, && \mathtt{mb}_1[5] = \mathtt{mb}_0[4] \oplus \delta_0, \\
&\mathtt{mb}_1[6] = \mathtt{mb}_0[7] \oplus \delta_3, && \mathtt{mb}_1[7] = \mathtt{mb}_0[6] \oplus \delta_3,
\end{aligned}$$

*where* $i = 0, 1, \ldots, 7$ *and* $\delta_0, \ldots, \delta_3$ *are any 32-bit strings. Let* $\mathtt{x}_0[0] \parallel \mathtt{x}_0[1] \parallel \ldots \parallel \mathtt{x}_0[7]$ *be the output* $\mathtt{x}_0$ *of* $EncComp_{256}(\mathtt{K}_0, \mathtt{mb}_0)$*. Then, the output* $\mathtt{x}_1$ *of* $EncComp_{256}(\mathtt{K}_1, \mathtt{mb}_1)$ *is given by*

$$\begin{array}{ll} \mathtt{x}_1[0] = \mathtt{x}_0[1] \oplus \delta_2, & \mathtt{x}_1[1] = \mathtt{x}_0[0] \oplus \delta_2, \\ \mathtt{x}_1[2] = \mathtt{x}_0[3] \oplus \delta_1, & \mathtt{x}_1[3] = \mathtt{x}_0[2] \oplus \delta_1, \\ \mathtt{x}_1[4] = \mathtt{x}_0[5] \oplus \delta_0, & \mathtt{x}_1[5] = \mathtt{x}_0[4] \oplus \delta_0, \\ \mathtt{x}_1[6] = \mathtt{x}_0[7] \oplus \delta_3, & \mathtt{x}_1[7] = \mathtt{x}_0[6] \oplus \delta_3. \end{array}$$

Proposition 1 and Proposition 2 are proved by using properties of internal functions such as `SubWord256`. Assuming that

$$\delta_0 = \Delta_2,\ \delta_1 = \Delta_3,\ \delta_2 = \Delta_0,\ \delta_3 = \Delta_1,$$

we obtain the following proposition from Proposition 1 and Proposition 2.

**Proposition 3.** *Let* $\mathtt{chain}_0$ *and* $\mathtt{mb}_0$ *be a key and a message block, respectively.*

$$\begin{aligned} \mathtt{chain}_0 = \mathtt{chain}_0[0] \parallel \mathtt{chain}_0[1] \parallel \mathtt{chain}_0[2] \parallel \mathtt{chain}_0[3] \\ \parallel \mathtt{chain}_0[4] \parallel \mathtt{chain}_0[5] \parallel \mathtt{chain}_0[6] \parallel \mathtt{chain}_0[7], \\ \mathtt{mb}_0 = \mathtt{mb}_0[0] \parallel \mathtt{mb}_0[1] \parallel \mathtt{mb}_0[2] \parallel \mathtt{mb}_0[3] \\ \parallel \mathtt{mb}_0[4] \parallel \mathtt{mb}_0[5] \parallel \mathtt{mb}_0[6] \parallel \mathtt{mb}_0[7]. \end{aligned}$$

*Suppose that a key* $\mathtt{chain}_1$ *and a message block* $\mathtt{mb}_1$ *are given as*

$$\begin{aligned} \mathtt{chain}_1 = (\mathtt{chain}_0[1] \oplus \Delta_2) \parallel (\mathtt{chain}_0[0] \oplus \Delta_2) \\ \parallel (\mathtt{chain}_0[3] \oplus \Delta_1) \parallel (\mathtt{chain}_0[2] \oplus \Delta_1) \\ \parallel (\mathtt{chain}_0[5] \oplus \Delta_0) \parallel (\mathtt{chain}_0[4] \oplus \Delta_0) \\ \parallel (\mathtt{chain}_0[7] \oplus \Delta_3) \parallel (\mathtt{chain}_0[6] \oplus \Delta_3), \end{aligned} \tag{3}$$

$$\begin{aligned} \mathtt{mb}_1 = (\mathtt{mb}_0[1] \oplus \Delta_0) \parallel (\mathtt{mb}_0[0] \oplus \Delta_0) \parallel (\mathtt{mb}_0[3] \oplus \Delta_3) \parallel (\mathtt{mb}_0[2] \oplus \Delta_3) \\ \parallel (\mathtt{mb}_0[5] \oplus \Delta_2) \parallel (\mathtt{mb}_0[4] \oplus \Delta_2) \parallel (\mathtt{mb}_0[7] \oplus \Delta_1) \parallel (\mathtt{mb}_0[6] \oplus \Delta_1). \end{aligned} \tag{4}$$

*When the output* $\mathtt{x}_0$ *of* $EncComp_{256}(\mathtt{chain}_0, \mathtt{mb}_0)$ *is denoted by* $\mathtt{x}_0[0] \parallel \mathtt{x}_0[1] \parallel \ldots \parallel \mathtt{x}_0[7]$*, the output* $\mathtt{x}_1$ *of* $EncComp_{256}(\mathtt{chain}_1, \mathtt{mb}_1)$ *is given by*

$$\begin{aligned} \mathtt{x}_1 = (\mathtt{x}_0[1] \oplus \Delta_0) \parallel (\mathtt{x}_0[0] \oplus \Delta_0) \parallel (\mathtt{x}_0[3] \oplus \Delta_3) \parallel (\mathtt{x}_0[2] \oplus \Delta_3) \\ \parallel (\mathtt{x}_0[5] \oplus \Delta_2) \parallel (\mathtt{x}_0[4] \oplus \Delta_2) \parallel (\mathtt{x}_0[7] \oplus \Delta_1) \parallel (\mathtt{x}_0[6] \oplus \Delta_1). \end{aligned} \tag{5}$$

### 2.2 Distinguisher for Lesamnta's Block Cipher

Proposition 3 immediately gives an efficient related-key adversary $A$ for distinguishing between Lesamnta-256's block cipher $EncComp_{256}$ and an ideal cipher $IC$. The basic idea of this distinguisher was shown in [1].

The algorithm of the adversary $A$ is described below. Suppose that a block cipher $BC$ to which $A$ has access is promised to be either $EncComp_{256}$ or $IC$ and $A$ is allowed to have access to the related-key oracle such as Eq. (3). Namely, $A$ does not know keys $\mathtt{chain}_0, \mathtt{chain}_1$, but $A$ can have access to $BC(\mathtt{chain}_0, \cdot)$ and $BC(\mathtt{chain}_1, \cdot)$.

1. Choose a message block $\mathtt{mb}_0$ at random and determine another message block $\mathtt{mb}_1$ as Eq. (4).
2. Let $\mathtt{x}_i$ be the output of $BC(\mathtt{chain}_i, \mathtt{mb}_i)$ where $i = 0, 1$. If Eq. (5) holds, then output 1, otherwise output 0.

We evaluate the probability that $A$ outputs 1. If $BC$ is $EncComp_{256}$, then $A$ always outputs 1 because of Proposition 3. If $BC$ is $IC$, then the probability that $A$ outputs 1 is $2^{-256}$. Thus, $A$ can distinguish between Lesamnta-256's block cipher and the ideal cipher by making only two queries.

### 2.3 Pseudo-Collision of Lesamnta

The sophisticate use of Proposition 3 allows an adversary to produce a pseudo-collision of Lesamnta-256 with $O(2^{64})$ computations of the compression function. This attack was shown in [1].

Consider Lesamnta-256's compression function `Compression256` (Figure 11 of [2]). The algorithm of an adversary $A$ that finds a pseudo-collision is described below.

1. Let a set $\mathcal{U} = \emptyset$.
2. For $i = 1, 2, \ldots, 2^{64}$, do the following steps.
   2.1 Choose $\mathtt{chain}_i[j], \mathtt{mb}_i[j]$ where $j = 0, 2, 4, 6$ at random.
   2.2 Determine $\mathtt{chain}_i, \mathtt{mb}_i$ as follows:

$$\begin{aligned} \mathtt{chain}_i = {} & \mathtt{chain}_i[0] \parallel (\mathtt{chain}_i[0] \oplus \Delta_2) \\ & \parallel \mathtt{chain}_i[2] \parallel (\mathtt{chain}_i[2] \oplus \Delta_1) \\ & \parallel \mathtt{chain}_i[4] \parallel (\mathtt{chain}_i[4] \oplus \Delta_0) \\ & \parallel \mathtt{chain}_i[6] \parallel (\mathtt{chain}_i[6] \oplus \Delta_3) \end{aligned} \tag{6}$$

$$\begin{aligned} \mathtt{mb}_i = {} & \mathtt{mb}_i[0] \parallel (\mathtt{mb}_i[0] \oplus \Delta_0) \parallel \mathtt{mb}_i[2] \parallel (\mathtt{mb}_i[2] \oplus \Delta_3) \\ & \parallel \mathtt{mb}_i[4] \parallel (\mathtt{mb}_i[4] \oplus \Delta_2) \parallel \mathtt{mb}_i[6] \parallel (\mathtt{mb}_i[6] \oplus \Delta_1) \end{aligned} \tag{7}$$

   2.3 Compute $\mathtt{Compression256}(\mathtt{chain}_i, \mathtt{mb}_i)$. The output is denoted by $\mathtt{z}_i$.
   2.4 Let $\mathcal{U} \leftarrow \mathcal{U} \cup (\mathtt{chain}_i, \mathtt{mb}_i, \mathtt{z}_i)$.
3. Find $(\mathtt{chain}_\iota, \mathtt{mb}_\iota)$ and $(\mathtt{chain}_\nu, \mathtt{mb}_\nu)$ such that $\mathtt{z}_\iota = \mathtt{z}_\nu$ from $\mathcal{U}$. (i.e., a pseudo-collision).

Recall that Lesamnta's compression functions is the MMO mode. The output $\mathtt{z}_i$ of $\mathtt{Compression256}(\mathtt{chain}_i, \mathtt{mb}_i)$ always satisfies the following property due to Proposition 3.

$$\mathtt{z}_i[0] = \mathtt{z}_i[1],\ \mathtt{z}_i[2] = \mathtt{z}_i[3],\ \mathtt{z}_i[4] = \mathtt{z}_i[5],\ \mathtt{z}_i[6] = \mathtt{z}_i[7].$$

Namely, the size of the output space of $\mathtt{Compression256}(\mathtt{chain}_i, \mathtt{mb}_i)$ is $2^{128}$. Since $\mathcal{U}$ has $2^{64}$ elements, there exists a pair satisfying step 3 with probability $1 - 1/\mathrm{e}$ due to the birthday paradox.

# 3 The Impact of the Security Analysis of the Compression Function on the Full Lesamnta

In this section, we discuss the impact of the security analysis described in 2 on the security of the full Lesamnta by firstly reviewing the expected strength and security goals claimed in [2] and by secondly considering several attacking scenarios.

## 3.1 Review of What Was Claimed in [2]

In the section of "Expected Strength and Security Goals" in [2], we described as follows:

*Table 1 shows the expected strength of Lesamnta for each of the security requirements (i.e., the expected complexity of attacks). What values in Table 1 mean is explained below. The row indicated by "HMAC" lists the approximate number of queries required by any distinguishing attack against HMAC using Lesamnta. The row indicated by "PRF" lists the approximate number of queries required by any distinguishing attack against the additional PRF modes described in Sec. 13.1. The row indicated by "Randomized hashing" lists the approximate complexity to find another pair of a message and a random value for a given pair of a $2^k$-bit message and a random value. The fourth row lists the approximate complexity of any collision attack. The fifth row lists the approximate complexity of any preimage attack. The sixth row lists the approximate complexity of the Kelsey-Schneier second-preimage attack with any first preimage shorter than $2^k$ bits. The seventh row lists the approximate number of queries required by any length-extension attack against Lesamnta. A cryptanalytic attack may be a profound threat to Lesamnta if its complexity is much less than the complexity in Table 1.*

**Table 1.** Expected strength of Lesamnta

| Requirement | Lesamnta | | | |
|---|---|---|---|---|
| | 224 | 256 | 384 | 512 |
| HMAC | $2^{112}$ | $2^{128}$ | $2^{192}$ | $2^{256}$ |
| PRF | $2^{112}$ | $2^{128}$ | $2^{192}$ | $2^{256}$ |
| Randomized hashing | $2^{256-k}$ | $2^{256-k}$ | $2^{512-k}$ | $2^{512-k}$ |
| Collision resistance | $2^{112}$ | $2^{128}$ | $2^{192}$ | $2^{256}$ |
| Preimage resistance | $2^{224}$ | $2^{256}$ | $2^{384}$ | $2^{512}$ |
| Second-preimage resistance | $2^{256-k}$ | $2^{256-k}$ | $2^{512-k}$ | $2^{512-k}$ |
| Length-extension attacks | $2^{112}$ | $2^{128}$ | $2^{192}$ | $2^{256}$ |

*Table 1 includes proof-based strength and attack-based strength. The security proof of Lesamnta is given as follows:*

Proved security 1: *Lesamnta is indifferentiable from a random oracle under the assumption that block ciphers $E, L$ are independent ideal ciphers.*
*This proof partially ensures the security of randomized hashing, collision resistance, preimage resistance, second-preimage resistance, and length-extension attacks.*

Proved security 2: *Lesamnta is collision resistant under the assumption that the compression function $h$ and the output function $g$ are collision resistant.*
*This proof ensures the security of collision resistance, and in part, preimage resistance and second-preimage resistance.*

Proved security 3: *Lesamnta is a pseudorandom function under the assumption that block ciphers $E, L$ are independent pseudorandom permutations.*
*This proof ensures the security of HMAC and PRF.*

We claim that the impact of the security analysis of the compression function on the security of Lesamnta described in 2 is limited to the following:

– Each of the assumption made in Proved Security 1 and the one in Proved Security 2 no longer holds because the above attack means that Lesamnta's block cipher is a *poor* instantiation of an ideal cipher.

We claim that there is no problem regarding Proved Security 3 because their proofs only assume the pseudo-randomness of the underlying block ciphers, that is, the key is secret and chosen at random.

### 3.2 Collision Resistance, Second-preimage Resistance, and Preimage Resistance

As for collision resistance, second-preimage resistance, and preimage resistance, Lesamnta does not have proof-based strength but we still claim that, regarding each of these security requirements, Lesamnta has attack-based strength which is estimated in security analysis described in [2] together with the arguments we describe below.

As for collision resistance and second-preimage resistance, we think that it is difficult to transform the collision attack on the compression function given in Section 2 into an attack on the full Lesamnta hash function because it is not clear how to find the chaining variable $H_i$ of the specific form described in Section 2 for the full Lesamnta.

As for preimage resistance, we do not know any way to transform the pseudo-collision attack given in Section 2 into a preimage attack on the full Lesamnta.

### 3.3 Security against a Collision Attack on the Full Lesamnta

Using Proposition 3, we can find a collision of Lesamnta hash function with the same complexity of a generic attack.

Consider Lesamnta-256. The algorithm of an adversary that finds a collision is described below.

1. Let $\mathcal{U}^{(0)} = \emptyset$ and $\mathcal{U}^{(1)} = \emptyset$.
2. Choose message block blocks $\mathtt{mb}_i^{(0)}, \mathtt{mb}_i^{(1)}$ at random. If $\mathtt{mb}_i^{(1)}$ satisfies the following equations (i.e., Eq. (7)), then choose $\mathtt{mb}_i^{(1)}$ again.

$$\begin{aligned} \mathtt{mb}_i^{(1)}[1] = \mathtt{mb}_i^{(1)}[0] \oplus \Delta_0, \quad \mathtt{mb}_i^{(1)}[3] = \mathtt{mb}_i^{(1)}[2] \oplus \Delta_3, \\ \mathtt{mb}_i^{(1)}[5] = \mathtt{mb}_i^{(1)}[4] \oplus \Delta_2, \quad \mathtt{mb}_i^{(1)}[7] = \mathtt{mb}_i^{(1)}[6] \oplus \Delta_1, \end{aligned}$$

   where $\Delta_i$ is given by Eq. (1).
3. Compute

$$\begin{aligned} \mathtt{chain}_i^{(0)} &= \mathtt{Compression256}(IV, \mathtt{mb}_i^{(0)}), \\ \mathtt{chain}_i^{(1)} &= \mathtt{Compression256}(\mathtt{chain}_i^{(0)}, \mathtt{mb}_i^{(1)}), \end{aligned}$$

   where $IV$ is the standard initial value (Section 5.2.3.2 of [2]).
4. Let $\mathcal{U}^{(0)} \leftarrow \mathcal{U}^{(0)} \cup (\mathtt{chain}_i^{(0)}, \mathtt{mb}_i^{(0)})$ and $\mathcal{U}^{(1)} \leftarrow \mathcal{U}^{(1)} \cup (\mathtt{chain}_i^{(1)}, \mathtt{mb}_i^{(1)})$.
5. If all the following conditions hold, then go to the next step, otherwise go back to step 2.
   - There is an element in $\mathcal{U}^{(0)}$ satisfying Eq. (6), that is, for some $i$

$$\begin{aligned} \mathtt{chain}_i^{(0)}[1] = \mathtt{chain}_i^{(0)}[0] \oplus \Delta_2, \quad \mathtt{chain}_i^{(0)}[3] = \mathtt{chain}_i^{(0)}[2] \oplus \Delta_1, \\ \mathtt{chain}_i^{(0)}[5] = \mathtt{chain}_i^{(0)}[4] \oplus \Delta_0, \quad \mathtt{chain}_i^{(0)}[7] = \mathtt{chain}_i^{(0)}[6] \oplus \Delta_2. \end{aligned}$$

     This index $i$ is denoted by $i_0$.
   - There is an element in $\mathcal{U}^{(1)}$ such that for some $i$

$$\mathtt{chain}_i^{(1)}[j] = \mathtt{chain}_i^{(1)}[j+1]$$

     for $j = 0, 2, 4, 6$. This index $i$ is denoted by $i_1$.
6. Choose a message block $\mathtt{mb}'^{(1)}$ at random such that

$$\begin{aligned} \mathtt{mb}'^{(1)}[1] = \mathtt{mb}'^{(1)}[0] \oplus \Delta_0, \quad \mathtt{mb}'^{(1)}[3] = \mathtt{mb}'^{(1)}[2] \oplus \Delta_3, \\ \mathtt{mb}'^{(1)}[5] = \mathtt{mb}'^{(1)}[4] \oplus \Delta_2, \quad \mathtt{mb}'^{(1)}[7] = \mathtt{mb}'^{(1)}[6] \oplus \Delta_1. \end{aligned}$$

7. Compute

$$\mathtt{chain}'^{(1)} = \mathtt{Compression256}(\mathtt{chain}_{i_0}^{(0)}, \mathtt{mb}'^{(1)}).$$

8. If the following equations hold, then output $\mathtt{mb}_{i_1}^{(0)} \parallel \mathtt{mb}_{i_1}^{(1)}$ and $\mathtt{mb}_{i_0}^{(0)} \parallel \mathtt{mb}'^{(1)}$ as a collision-message pair, that is,

$$\mathtt{chain}'^{(1)}[j] = \mathtt{chain}_{i_1}^{(1)}[j],$$

   for $j = 0, 1, \ldots, 7$. Otherwise go back to step 6.

We evaluate the complexity of the above algorithm. In order to satisfy the conditions in step 5 and the condition in step 8, $O(2^{128})$ computations of the compression function are required. As a result, we conclude that the above attack is not better than the generic collision attack. This means that this attack does not pose any threat on the full Lesamnta.

## 4 A Plan for a Minor Change

We observe that the security analysis discussed here is based on some symmetry in Lesamnta. To destroy the symmetry, we plan to make a minor change to the specification of Lesamnta by changing the round constants. The important design goals for the new round constants are security and hardware efficiency.

The possible ideas for new round constants are using the following techniques: LFSR, publicly known random-looking numbers, pseudo-random generators, etc. We also consider the possibility of using the on-the-fly technique and the adaptability to the extension of Lesamnta specified in [2].

## 5 Concluding Remarks

In this paper, we have discussed the security analysis of the compression function of Lesamnta that was pointed by Bouillaguet *et al.* As the result of examining several attacking scenarios based on this analysis, we conclude that the expected strength of Lesamnta described still remains the same despite of the loss of proved security regarding preimage resistance, second preimage resistance, and collision resistance.

In order for Lesamnta to get back proved security on each of these security requirements, we will make a minor change to the specification by changing round constants.

## Acknowledgments

We would like to thank Charles Bouillaguet, Orr Dunkelman, Gaëtan Leurent, Pierre-Alain Fouque for their excellent analysis on Lesamnta. We also thank Kota Ideguchi, Yasuko Fukuzawa, and Toru Owada for fruitful discussions. This work was partially supported by the National Institute of Information and Communications Technology, Japan.

## References

1. C. Bouillaguet, O. Dunkelman, G. Leurent, and P. A. Fouque, Private communication, 2009.
2. S. Hirose, H. Kuwakado, and H. Yoshida, "SHA-3 proposal: Lesamnta," `http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/documents/Lesamnta.zip%`, October 2008. latest version: `http://www.sdl.hitachi.co.jp/crypto/lesamnta/`.
3. National Institute of Standards and Technology, "Announcing request for candidate algorithm nominations for a new cryptographic hash algorithm (SHA-3) family," `http://csrc.nist.gov/groups/ST/hash/documents/`, November 2007.