PAPER Special Section on Cryptography and Information Security

An AES Based 256-bit Hash Function for Lightweight Applications: Lesamnta-LW*

Shoichi HIROSE[†], **Kota IDEGUCHI**[††], **Hidenori KUWAKADO**[†††], ***Members***, **Toru OWADA**[††], **Bart PRENEEL**[††††], ***Nonmembers***, ***and*** **Hirotaka YOSHIDA**[††,††††a)], ***Member***

SUMMARY This paper proposes a new lightweight 256-bit hash function Lesamnta-LW. The security of Lesamnta-LW is reduced to that of the underlying AES-based block cipher and it is theoretically analyzed for an important application, namely the key-prefix mode. While most of recently proposed lightweight primitives are hardware-oriented with very small footprints, our main target with Lesamnta-LW is to achieve compact and fast hashing for lightweight application on a wider variety of environments ranging from inexpensive devices to high-end severs at the 2^{120} security level. As for performance, our primary target CPUs are 8-bit and it is shown that, for short message hashing, Lesamnta-LW offers better tradeoffs between speed and cost on an 8-bit CPU than SHA-256.

key words: *hash functions, lightweight cryptography, security reduction proofs*

1. Introduction

Systems and solutions using small portable electronic devices employing low-cost 8-bit CPUs have gained increasing attention from both companies and end users. About 55% of all CPUs sold in the world are 8-bit microcontrollers and microprocessors and over 4 billion 8-bit controllers were sold in 2006 [37], [46]. These devices include low-end smart cards and RFID (Radio frequency identification) tags. Based on the report in [40] stating that the passive RFID tag market is expected to hit $486M in 2013, it is expected that, in the near future, we will see a wide variety of applications for mobile phones and wireless sensor networks, etc.

Security and privacy in such devices have recently opened up an active research area called lightweight cryptography. The main challenge in this area is to design cryptographic primitives or protocols that meet the system requirements which are often very severe in the sense that the available resources are quite limited for implementing these cryptographic components. Lightweight ciphers such as PRESENT [9] and KATAN [12] have been proposed. On the other hand, it is pointed out [18] that in RFID security community, it is commonly assumed that hash functions are the better choice than block ciphers from an implementation perspective, even though RFID tags supporting AES are already available [17]. In this sense, lightweight hash functions such as H-PRESENT [10], MAME [43], SQUASH [38] and QUARK [2] hold promise for implementation. However, these hash functions mentioned above are hardware-oriented with very small footprints. Hardware-oriented schemes do not necessarily provide good performance on 8-bit CPUs. We also notice that there are not large RAM/ROM available on small portable electronic devices.

This paper proposes a 256-bit hash function, *Lesamnta-LW*, that provides good performance on memory-constrained devices employing 8-bit CPUs. Its domain extension is the strengthened Merkle-Damgård construction and its underlying component is an AES-based block cipher taking a 256-bit plaintext and a 128-bit key. As for choice of algorithms, block cipher technology appears to be more mature than hash function technology due to the AES competition organized by NIST. Note that Lesamnta-LW is a lightweight variant of Lesamnta [22] that was submitted to the SHA-3 competition. The design goals of Lesamnta-LW are summarized below.

1. Compact and fast, optimized for lightweight applications on a wider variety of environments ranging from cheap devices to high-end severs:
   Our primary target CPUs are 8-bit and it is shown that, for short message hashing, Lesamnta-LW offers better tradeoffs between speed and cost on an 8-bit CPU than SHA-256. Our software implementation of Lesamnta-LW requires only 50 bytes of RAM. On high-end processors where AES instruction set can be utilized, Lesamnta-LW is reasonably fast.
   A provably secure key-prefix (KP) mode (required in PPP Challenge Handshake Authentication Protocol [39]) of Lesamnta-LW gains significant advantage over the standard method HMAC-SHA-256.
2. 2^{120} security level achieved with a high security margin:
   The compression function is a new mode of a block cipher, called the *LW1 mode*, which enables us to provide proofs reducing the security of Lesamnta-LW to that of

Manuscript received March 31, 2011.
Manuscript revised July 1, 2011.

[†]The author is with the Graduate School of Engineering, University of Fukui, Fukui-shi, 910-8507 Japan.
[††]The authors are with Systems Development Laboratory, Hitachi, Ltd. Yokohama-shi, 244-0817 Japan.
[†††]The author is with the Graduate School of Engineering, Kobe University, Kobe-shi, 657-8501 Japan.
[††††]The authors are with Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium.

*An earlier version of this paper is appeared in the preproceedings of the ICISC 2010 conference.

a) E-mail: hirotaka.yoshida.qv@hitachi.com

DOI: 10.1587/transfun.E95.A.89

the underlying block cipher. The block cipher is based on AES in order to gain confidence in its security.

For the security levels, an ideal 256-bit hash function would provide the 2^{256} security level against preimage attacks. However, the 2^{120} security level is sufficient for most applications, especially on small devices. We give preference to cost over preimage resistance in the design of Lesamnta-LW. (There is always a tradeoff between security and cost. The security and the cost do not go together generally.)

The outline of this paper is as follows. In Sect. 2, we explain our design strategy. In Sect. 3, we give the specification of the Lesamnta-LW hash function. In Sect. 4, we discuss the security reduction of Lesamnta-LW. In Sect. 5, we evaluate the security of Lesamnta-LW against all relevant attacks. Section 6 presents implementation results. Section 7 concludes the paper.

2. Design Principle

While recent symmetric-key primitive proposals provide a relatively low security levels such as 64-bit and 80-bit levels, we argue that there is an increasing demand for lightweight hash functions providing a high security level. A reasonable application would be code signing for small but highly sensitive devices which can be targeted at medical applications or car electronics. Our main design goal to satisfy these application requirements is to develop a secure 256-bit hash function which achieves small hardware/software implementations. More specifically, the most important aspects are to have security reductions, to have a small hardware footprint, and to have a low working memory (RAM) requirement for software. Our next target is to achieve fairly fast speed when taking into account the context, including the length of the input message and the modes of operation. This is because the required efficiency could include good performance for very short messages such as IDs or for the pseudorandom function derived from the hash function with constructions such as HMAC or Key-Prefix (KP) mode as discussed in this paper. A speedup can be obtained with the KP mode, compared to the standard solution HMAC with SHA-256.

2.1 Padding Method

For the padding method of Lesamnta-LW, the last block does not contain any part of the message input. It only contains the length of the message input. This property is required to guarantee preimage resistance of Lesamnta-LW.

2.2 LW1 Mode

Sophisticated designs and attacks on block ciphers were presented in the AES competition. Knowledge on block ciphers is useful in designing secure hash functions. This is why Lesamnta-LW is designed as a block-cipher-based hash function. A few reasons for choosing the LW1 mode are also listed below. First, from the viewpoint of attacks on a block cipher, recent collision attacks use the fact that an attacker can directly control the key of a block cipher. In contrast, the LW1 mode does not allow attackers to control the key of the block cipher directly. Second, the LW1 mode is theoretically analyzed. It enables us to reduce the security of Lesamnta-LW to that of the underlying block cipher to a greater extent than the popular Davies-Meyer mode [30] used by the SHA family.

2.3 Block Cipher

The block cipher is designed to meet the following requirements:

- The security analysis should be simple to have confidence in the design.
- It should be compact in software/hardware.
- It should offer a reasonable speed on high-end/low-end CPUs.

For this purpose, the block cipher is an AES-based design such that Lesamnta-LW can gain clear advantages over known block-cipher based designs such as SHA-256 and MAME. The key scheduling function ensures a strong nonlinearity and an excellent diffusion property by re-using the 32-bit permutation of the mixing function; this reduces the hardware complexity since a part of the hardware can be reused. The round constants sequentially generated from a linear feedback shift register introduce randomness and asymmetry into the key scheduling function.

3. Specification

3.1 Message Padding

The first step of the hash computation is the padding of the message. The purpose of the padding is to ensure that the input consists of a multiple of 128 bits. Suppose that the length of a message M is l bits. Append the bit "1" to the end of the message, followed by $k + 63$ zero bits, where k is the smallest non-negative integer such that $l + k \equiv 0 \pmod{128}$. Then, append a 64-bit block equal to the number l as expressed in binary representation. Thus, the maximum length of the message is $2^{64} - 1$.

3.2 Compression Function and Domain Extension

Lesamnta-LW is a Merkle-Damgård iterated hash function [15], [31] using the following compression function on 128-bit words $H_0^{(i-1)}$, $H_1^{(i-1)}$, and $M^{(i)}$:

$$h(H^{(i-1)}, M^{(i)}) = E_{H_0^{(i-1)}}(M^{(i)} \| H_1^{(i-1)}),$$

where $H^{(i-1)} = H_0^{(i-1)} \| H_1^{(i-1)}$ and E_K is a 256-bit block cipher with a 128-bit key K. We call this method to construct a compression function the LW1 mode. For a padded message input $M = M^{(1)} \| \cdots \| M^{(N)}$, Lesamnta-LW works as follows:

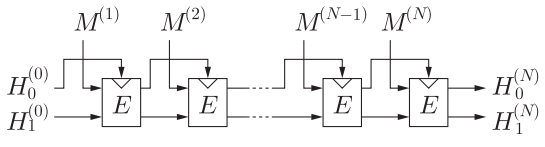


Fig. 1 The structure of Lesamnta-LW.

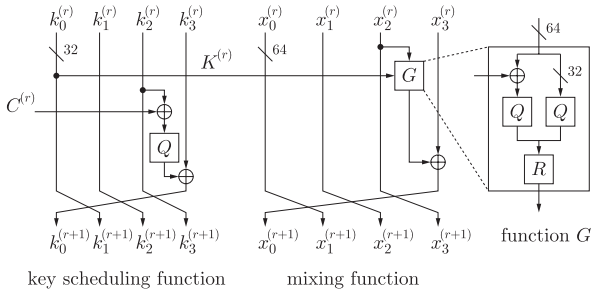


Fig. 2 The round function.

```
ConstantGenerator(word C[64])
begin
  word c;
  c = ffffffff; /*in hexadecimal*/
  for i = 0 to (64 * 3) - 1
    /* Galois LFSR */
    if c & 00000001 == 00000001
      c = (c >> 1) ^ dbcdcc80;
    else
      c = c >> 1;
    end if
    if i mod 3 == 0
      C[i/3] = c;
    end if
  end for
end
```

Fig. 3 The algorithm for generating the round constants.

$H^{(i)} = h(H^{(i-1)}, M^{(i)})$ for $1 \le i \le N$, where $H^{(0)}$ is a fixed initial value and $H^{(N)}$ is the output. It is illustrated in Fig. 1. This structure is referred to as $\mathrm{LW}^E(H^{(0)}, M)$ later in Sect. 4.

3.3 Block Cipher

Lesamnta-LW uses a 64-round block cipher E that takes as input a 128-bit key and a 256-bit plaintext. The block cipher consists of two parts: the key scheduling function mapping the key to the round keys and the mixing function taking as input a plaintext and the round keys to produce a ciphertext. Both of them use a type-1 4-branch generalized Feistel network (GFN) (cf. Zheng et al. [48]). One round of the block cipher is illustrated in Fig. 2. The input variables to round r for the mixing function and the key scheduling function are denoted by $(x_0^{(r)}, x_1^{(r)}, x_2^{(r)}, x_3^{(r)})$ and $(k_0^{(r)}, k_1^{(r)}, k_2^{(r)}, k_3^{(r)})$ respectively. Each $x_i^{(r)}$ is a 64-bit word and each $k_i^{(r)}$ is a 32-bit word.

The mixing function consists of XORs, a wordwise permutation, and a non-linear function G. Taking as input a 32-bit round key $K^{(r)}$, the mixing function updates its intermediate state in the following manner:

$$x_0^{(r+1)} = x_3^{(r)} \oplus G(x_2^{(r)}, K^{(r)}), \quad x_1^{(r+1)} = x_0^{(r)},$$
$$x_2^{(r+1)} = x_1^{(r)}, \quad x_3^{(r+1)} = x_2^{(r)}.$$

The function G consists of XOR operations, a 32-bit non-linear permutation Q, and a function R. For a 64-bit input $y = y_0 \parallel y_1$ and a 32-bit round key $K^{(r)}$, $G(y, K^{(r)})$ is defined as follows:

$$G(y, K^{(r)}) = R(Q(y_0 \oplus K^{(r)}) \parallel Q(y_1)).$$

Using the AES components [14], the function Q is defined as follows:

$$Q = \textbf{MixColumns} \circ \textbf{SubBytes}.$$

The **SubBytes** transformation is a non-linear byte substitution that takes 4 bytes s_0, s_1, s_2, s_3 as input and operates independently on each byte by using the AES S-box. It proceeds as follows:

$$s'_i = \text{S-box}(s_i) \quad \text{for } 0 \le i < 4.$$

The **MixColumns** step is a bytewise operation that takes 4 bytes s_0, s_1, s_2, s_3 as input. The **MixColumns** step is given by the AES MDS matrix multiplication defined over $\mathrm{GF}(2^8)$ as follows:

$$\begin{bmatrix} s'_0 \\ s'_1 \\ s'_2 \\ s'_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix}.$$

For a 64-bit input $s = s_0 \parallel s_1 \parallel s_2 \parallel s_3 \parallel s_4 \parallel s_5 \parallel s_6 \parallel s_7$, the function $R(s)$ is defined as follows: $R(s) = s_4 \parallel s_5 \parallel s_2 \parallel s_3 \parallel s_0 \parallel s_1 \parallel s_6 \parallel s_7$.

One round of the key scheduling function consists of the following two steps:

Firstly, it generates the r-th round-key $K^{(r)} = k_0^{(r)}$.

Secondly, it updates the intermediate state in the following manner:

$$k_0^{(r+1)} = k_3^{(r)} \oplus Q(C^{(r)} \oplus k_2^{(r)}), \quad k_1^{(r+1)} = k_0^{(r)},$$
$$k_2^{(r+1)} = k_1^{(r)}, \quad k_3^{(r+1)} = k_2^{(r)},$$

where the 32-bit round constants $C^{(r)}$ are generated using the algorithm presented in Fig. 3. The algorithm is based on the linear feedback shift register (LFSR) of the following primitive polynomial:

$$g(x) = x^{32} + x^{31} + x^{29} + x^{28} + x^{26} + x^{25} + x^{24} + x^{23} + x^{20} + x^{19} + x^{17} + x^{16} + x^{15} + x^{12} + x^{11} + x^8 + 1.$$

4. Security Reduction

In this section, it is assumed that E is a block cipher with key

length $n/2$ and block length n for even n. For Lesamnta-LW, $n = 256$.

4.1 Collision Resistance

The collision resistance of LW^E can be proved in the ideal cipher model using the technique by Black et al. in [8].

Let $\mathcal{BC}(\kappa, \nu)$ be the set of all block ciphers with key size κ and block size ν. Let H^E be a hash function using a block cipher E. Let A be an adversary trying to find a collision for H^E. The col-advantage of A against H^E, $\mathrm{Adv}^{\mathrm{col}}_{H^E}(A)$, is given by

$$\Pr\left[A^E = (M, M') \wedge M \neq M' \wedge H^E(M) = H^E(M')\right],$$

where E is chosen uniformly at random from $\mathcal{BC}(\kappa, \nu)$.

The following theorem gives an upper bound on the probability of finding a collision of LW^E in the ideal cipher model. It implies that Lesamnta-LW has a claimed security level of at least 2^{120} block-cipher operations against collision attacks.

Theorem 1: For any collision-finding adversary A against LW^E asking at most q queries to E,

$$\mathrm{Adv}^{\mathrm{col}}_{\mathrm{LW}^E}(A) \leq \frac{(\gamma(n) + 3)q}{2^{n/2} - 1}$$

in the ideal cipher model, where $\gamma(n) = (\mathrm{e}/2)n/(\log_2 n - \log_2 \log_2 \mathrm{e} - 1)$.

The following lemma is used for the analysis of multicollision, which should be taken into consideration to evaluate the success probability of meet-in-the-middle attacks.

Lemma 1 (Theorem 3.1 in [32]): Suppose that there are t balls and t bins and that each ball is placed in a bin chosen independently and uniformly at random. Then, with probability at least $1 - 1/t$, no bin has more than $\mathrm{e} \ln t / \ln \ln t$ balls in it.

Proof of Theorem 1: For $1 \leq i \leq q$, let $(t_i, k_i, w_i \| x_i, y_i \| z_i)$ be a tuple such that $E(k_i, w_i \| x_i) = y_i \| z_i$ and $t_i \in \{\mathsf{e}, \mathsf{d}\}$ obtained by the i-th query. t_i represents the type of the i-th query: encryption (e) or decryption (d). Let $G_1, G_2, \ldots, G_q$ be a sequence of directed graphs such that $G_i = (V_i, L_i)$, where

- $V_1 = \{k_1 \| x_1, y_1 \| z_1\}$, $L_1 = \{(k_1 \| x_1, y_1 \| z_1)\}$, and
- $V_i = V_{i-1} \cup \{k_i \| x_i, y_i \| z_i\}$, $L_i = L_{i-1} \cup \{(k_i \| x_i, y_i \| z_i)\}$ for $2 \leq i \leq q$.

Each edge $(k_i \| x_i, y_i \| z_i)$ is labeled by (t_i, w_i). Notice that $y_i \| z_i = h(k_i \| x_i, w_i)$, where h is the compression funcuion of LW^E.

Suppose that the adversary A finds a collision of LW^E with the i-th query for the first time. Then, there must be a path in G_i from the initial value IV to some colliding output, which does not exist in $G_1, \ldots, G_{i-1}$. This path also contains the nodes $k_i \| x_i$ and $y_i \| z_i$, and the edge (t_i, w_i).

If $t_i = \mathsf{e}$, that is, the i-th query is an encryption query, then there must be an event such that $y_i \| z_i \in \{y_j \| z_j \mid 1 \leq j \leq i-1\} \cup \{k_j \| x_j \mid 1 \leq j \leq i-1\} \cup \{IV\}$. If $t_i = \mathsf{d}$, then there must be an event such that $k_i \| x_i \in \{y_j \| z_j \mid 1 \leq j \leq i-1\} \cup \{IV\}$.

For the case where $t_i = \mathsf{d}$ and $k_i \| x_i \in \{y_j \| z_j \mid 1 \leq j \leq i-1\}$, let us look into the new path in G_i mentioned above. Let $IV \xrightarrow{(t_{j_1}, M_{j_1})} v_{j_1} \xrightarrow{(t_{j_2}, M_{j_2})} \cdots \xrightarrow{(t_{j_{l-1}}, M_{j_{l-1}})} v_{j_{l-1}} \xrightarrow{(t_{j_l}, M_{j_l})} v_{j_l}$ be the prefix of the path, where $v_{j_{l-1}} = k_i \| x_i$, $(t_{j_l}, M_{j_l}) = (\mathsf{d}, w_i)$ and $v_{j_l} = y_i \| z_i$. We start from v_{j_l} and go back toward IV until we first find an edge (e, M_{j_k}) or reach the node IV without finding such an edge. Suppose that we reach IV. Then, it implies that there is an event such that $t_{i'} = \mathsf{d}$ and $k_{i'} \| x_{i'} = IV$ for some i' such that $1 \leq i' < i$. On the other hand, suppose that we find an edge (e, M_{j_k}). Then, it implies that there is an event such that $t_{i'} = \mathsf{e}$ and $y_{i'} \| z_{i'} \in \{k_j \| x_j \mid 1 \leq j < i'\}$ for some i' such that $1 < i' < i$, or an event such that $t_{i'} = \mathsf{d}$ and $k_{i'} \| x_{i'} \in \{y_j \| z_j \mid 1 \leq j < i' \wedge t_j = \mathsf{e}\}$ for some i' such that $1 < i' \leq i$.

From the discussions above, if A finds a collision with at most q queries, then it implies that there must be at least one of the following events for some i such that $1 \leq i \leq q$:

$\mathtt{A}_i$ $t_i = \mathsf{e}$ and $y_i \| z_i = IV$,
$\mathtt{B}_i$ $t_i = \mathsf{e}$ and $y_i \| z_i \in \{y_j \| z_j \mid 1 \leq j \leq i-1\} \cup \{k_j \| x_j \mid 1 \leq j \leq i-1\}$,
$\mathtt{C}_i$ $t_i = \mathsf{d}$ and $k_i \| x_i = IV$,
$\mathtt{D}_i$ $t_i = \mathsf{d}$ and $k_i \| x_i \in \{y_j \| z_j \mid 1 \leq j < i \wedge t_j = \mathsf{e}\}$.

It is easy to see that

$$\Pr[\mathtt{A}_i] \leq 1/(2^n - (i-1)),$$
$$\Pr[\mathtt{B}_i] \leq 2(i-1)/(2^n - (i-1)),$$
$$\Pr[\mathtt{C}_i] \leq 2^{n/2}/(2^n - (i-1)).$$

For $\mathtt{D}_i$, the probability of multicollision on y_j should be taken into consideration. From Lemma 1, for $1 \leq q \leq 2^n$,

$$\Pr[\mathtt{D}_i] \leq \gamma(n) 2^{n/2}/(2^n - (i-1)) + 1/2^{n/2}.$$

Precisely speaking, the distribution of $y_j \| z_j$ is not uniform on $\{0,1\}^n$ since E is a keyed permutation. However, since $\Pr[y_j \in \{y_1, \ldots, y_{j-1}\}] \leq \Pr[y_j \notin \{y_1, \ldots, y_{j-1}\}]$, the probability of multicollision is smaller in this case.

Thus, for $1 \leq q \leq 2^n$,

$$\begin{aligned}\mathrm{Adv}^{\mathrm{col}}_{\text{kp-LW}^E}(A) &\leq \sum_{i=1}^{q} (\Pr[\mathtt{A}_i] + \Pr[\mathtt{B}_i] + \Pr[\mathtt{C}_i] + \Pr[\mathtt{D}_i]) \\ &\leq (\gamma(n) + 3)q/(2^{n/2} - 1).\end{aligned}$$

The upper bound exceeds 1 for $q > 2^{n/2}$. □

4.2 (Second-)Preimage Resistance

The preimage resistance of LW^E can also be proved in the ideal cipher model using the same technique. It is at the same level as its collision resistance. It implies that Lesamnta-LW also has a claimed security level of at least 2^{120} block-cipher operations against (second-)preimage attacks. Lesamnta-LW cannot provide security larger than 2^{128} since its compression function is invertible.

4.3 Keyed Hashing Mode

4.3.1 Keyed-via-IV (KIV) Mode

The KIV mode is a method to construct a PRF from a given hash function. It simply replaces the initial value IV with a secret key.

The KIV mode of Lesamnta-LW with the first half of the output chopped off resists any distinguishing attack that requires much fewer than 2^{128} queries if the underlying block cipher is a pseudorandom permutation (PRP).

Let $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ be a set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. Let $F : \mathcal{K} \times \mathcal{X} \to \mathcal{Y}$ be a keyed function from $\mathcal{X}$ to $\mathcal{Y}$, where $\mathcal{K}$ is its key space. Let A be an adversary which has oracle access to functions from $\mathcal{X}$ to $\mathcal{Y}$ and outputs 0 or 1. The prf-advantage of A against F, $\mathrm{Adv}_F^{m\text{-prf}}(A)$, is given by

$$\left|\Pr[A^{F_{K_1},\ldots,F_{K_m}} = 1] - \Pr[A^{\rho_1,\ldots,\rho_m} = 1]\right|,$$

where K_j's and ρ_j's are chosen uniformly and independently at random from $\mathcal{K}$ and $\mathcal{F}(\mathcal{X}, \mathcal{Y})$, respectively. $\mathrm{Adv}_F^{1\text{-prf}}(A)$ is simply denoted by $\mathrm{Adv}_F^{\mathrm{prf}}(A)$. F is called a PRF if $\mathrm{Adv}_F^{\mathrm{prf}}(A)$ is negligible for any efficient A.

Let $\mathcal{P}(\mathcal{X})$ be a set of all permutations on $\mathcal{X}$. If F is a keyed permutation on $\mathcal{X}$ and A has oracle access to permutations in $\mathcal{P}(\mathcal{X})$, then the advantage of A is called prp-advantage and denoted by $\mathrm{Adv}_F^{m\text{-prp}}(A)$.

In the remaining part, the KIV mode of LW^E with the first half of the output chopped off is denoted by $\mathsf{kiv}\text{-}\mathrm{LW}^E$.

Theorem 2: Let A be a prf-adversary against $\mathsf{kiv}\text{-}\mathrm{LW}^E$. Suppose that A runs in time at most t, and makes at most q queries, and each query has at most ℓ message blocks. Then, there exists a prp-adversary B against E such that

$$\mathrm{Adv}^{\mathrm{prf}}_{\mathsf{kiv}\text{-}\mathrm{LW}^E}(A) \le \ell q \cdot \mathrm{Adv}^{\mathrm{prp}}_E(B) + \frac{\ell q(q-1)}{2^{2n+1}}.$$

B makes at most q queries and runs in time at most $t + O(\ell q T_E)$, where T_E represents the time required to compute E.

Theorem 2 follows from Lemmas 2 and 3 shown below. Proofs are given in Appendix B and Appendix C, respectively.

Lemma 2: Let A be a prf-adversary against $\mathsf{kiv}\text{-}\mathrm{LW}^E$. Suppose that A runs in time at most t, and makes at most q queries, and each query has at most ℓ message blocks. Then, there exists a prf-adversary B against E with access to q oracles such that

$$\mathrm{Adv}^{\mathrm{prf}}_{\mathsf{kiv}\text{-}\mathrm{LW}^E}(A) = \ell \cdot \mathrm{Adv}^{q\text{-prf}}_E(B).$$

B makes at most q queries and runs in time at most $t + O(\ell q T_E)$, where T_E represents the time required to compute E.

Lemma 3: Let A be a prf-adversary against E with m oracles. Suppose that A runs in time at most t, and makes at most q queries. Then, there exists a prp-adversary B against E such that

$$\mathrm{Adv}_E^{m\text{-prf}}(A) \le m \cdot \mathrm{Adv}_E^{\mathrm{prp}}(B) + \frac{q(q-1)}{2^{2n+1}}.$$

B makes at most q queries and runs in time at most $t + O(q\,T_E)$, where T_E represents the time required to compute E.

4.3.2 Key-Prefix (KP) Mode

The KP mode is a method to construct a PRF from a given hash function [42]. It simply feeds $K\|M$ to the hash function as an input, where K is a secret key and M is a given message. This mode uses a hash function as a black box. In this sense, it is similar to HMAC [34].

The KP mode of Lesamnta-LW with the first half of the output chopped off resists any distinguishing attack that requires much fewer than 2^{128} queries if the underlying block cipher is a pseudorandom permutation (PRP) and it also has a mild security property given later.

Let h be the compression function of LW^E and $\mathcal{B} = \{0,1\}^{n/2}$. Let $G_1^E : \mathcal{B} \times \mathcal{B} \to \mathcal{B}^2$ be a keyed function such that $G_1^E(K, M) = h(h(IV, K), M)$, where $K \in \mathcal{B}$ and $M \in \mathcal{B}$. Let $G_2^E : \mathcal{B}^2 \times \mathcal{B} \to \mathcal{B}^2$ be a keyed function such that $G_2^E(K', M) = h(K', M)$, where $K' \in \mathcal{B}^2$ and $M \in \mathcal{B}$.

In the remaining part, the KP mode of LW^E with the first half of the output chopped off is denoted by $\mathsf{kp}\text{-}\mathrm{LW}^E$.

Theorem 3: Let A be a prf-adversary against $\mathsf{kp}\text{-}\mathrm{LW}^E$. Suppose that A runs in time at most t, and makes at most q queries, and each query has at most ℓ message blocks. Then, there exist an adversary B against G_1^E such that

$$\mathrm{Adv}^{\mathrm{prf}}_{\mathsf{kp}\text{-}\mathrm{LW}^E}(A) \le \mathrm{Adv}^{\mathrm{prf}}_{\mathsf{kiv}\text{-}\mathrm{LW}^E}(A) + \mathrm{Adv}^{G_2^E}_{G_1^E}(B),$$

where

$$\mathrm{Adv}^{G_2^E}_{G_1^E}(B) = \left|\Pr[B^{G_1^E(K,\cdot)} = 1] - \Pr[B^{G_2^E(K',\cdot)} = 1]\right|$$

and K and K' are random variables uniformly distributed over $\mathcal{B}$ and $\mathcal{B}^2$, respectively. B makes at most q queries and runs in time at most $t + O(\ell q T_E)$, where T_E represents the time required to compute E.

Proof : Let ρ be a random function uniformly distributed over $\mathcal{F}(\mathcal{B}^{\le\ell}, \mathcal{B})$, where $\mathcal{B}^{\le\ell} = \bigcup_{i=0}^{\ell} \mathcal{B}^i$. Let K and K' be random variables uniformly distributed over $\mathcal{B}$ and $\mathcal{B}^2$, respectively. Then,

$$\begin{aligned}\mathrm{Adv}^{\mathrm{prf}}_{\mathsf{kp}\text{-}\mathrm{LW}^E}(A) &= \left|\Pr[A^{\mathsf{kp}\text{-}\mathrm{LW}^E_K} = 1] - \Pr[A^{\rho} = 1]\right| \\ &\le \left|\Pr[A^{\mathsf{kp}\text{-}\mathrm{LW}^E_K} = 1] - \Pr[A^{\mathsf{kiv}\text{-}\mathrm{LW}^E_{K'}} = 1]\right| \\ &\quad + \left|\Pr[A^{\mathsf{kiv}\text{-}\mathrm{LW}^E_{K'}} = 1] - \Pr[A^{\rho} = 1]\right|.\end{aligned}$$

It is easy to see that

$$\left|\Pr[A^{\mathsf{kiv}\text{-}\mathrm{LW}^E_{K'}} = 1] - \Pr[A^{\rho} = 1]\right| = \mathrm{Adv}^{\mathrm{prf}}_{\mathsf{kiv}\text{-}\mathrm{LW}^E}(A).$$

Let us consider the following adversary B against G_1^E. B first runs A. For each query $M = M_1 \| M_{\text{tail}}$ from A, B asks the first block M_1 to its oracle and receives the reply H. Then, B returns the second half of H if M_{tail} is empty and $\text{LW}^E(H, M_{\text{tail}})$ otherwise. Finally, B outputs A's output. Then,

$$\begin{aligned}\text{Adv}_{G_1^E}^{G_2^E}(B) &= \left|\Pr[B^{G_1^E(K,\cdot)} = 1] - \Pr[B^{G_2^E(K',\cdot)} = 1]\right| \\ &= \left|\Pr[A^{\mathsf{kp}\text{-}\mathrm{LW}_K^E} = 1] - \Pr[A^{\mathsf{kiv}\text{-}\mathrm{LW}_{K'}^E} = 1]\right|.\end{aligned}$$

This completes the proof. □

G_2^E is a PRF if E is a PRP. Thus, $\text{Adv}_{G_1^E}^{G_2^E}(B)$ is negligible for any efficient B if E is a PRP and G_1^E is a PRF.

5. Preliminary Analysis

In our preliminary analysis, we evaluate the security of Lesamnta-LW and the underlying block cipher against all relevant attacks. In the analysis of the block cipher, the attacker can have at most 2^{128} complexity because of the key length (128 bits) of the cipher rather than the plaintext length (256 bits).

5.1 Differential and Linear Attacks

We examined resistance of the block cipher against differential [6] and linear attacks [29] which are two of the most powerful tools in block cipher cryptanalysis. Hereafter, we only explain our method of evaluating the security against differential cryptanalysis as we can apply a similar method regarding linear cryptanalysis because of its duality to differential cryptanalysis [13]. For this purpose, we compute upper bounds on the probabilities of differential and linear characteristics. Our method is as follows:

- Make abstraction of the exact differences used in these characteristics and then just consider patterns of active S-boxes.
- Perform experiments with the Viterbi algorithm to compute lower bounds on the minimum number of the active S-boxes. These experiments consider the MDS matrix property whose branch number is 5.

With this method, we can observe that the minimum number of the active S-boxes for 24 rounds is 24. Therefore the probabilities of differential characteristics of 24 rounds of Lesamnta-LW are upperbounded by 2^{-144} because the maximum differential probability of the AES S-box is 2^{-6}. As a result, it is very unlikely that differential/linear attacks can be applied successfully to the full Lesamnta-LW.

5.2 Higher Order Differential and Interpolation Attack

In the higher order differential attacks [27], the attacker constructs Boolean polynomial expressions for a cipher. The idea of the attack is that if the bits in the intermediate state are expressed by Boolean polynomials of degree at most d, the $(d+1)$-th order differential in polynomial sense of the Boolean polynomial would be 0. Therefore if the value d is reasonably small, the attack can be mounted. In the case of Lesamnta-LW, we found that every output bit of the S-box can be expressed as a Boolean polynomial of degree 7 in terms of input bits. Our experiments confirmed that the degree of such polynomials for Lesamnta-LW with 19 rounds reaches to the required degree 256. Therefore, we expect that the full Lesamnta-LW is secure against higher order differential attacks.

In the interpolation attack [24], an attacker constructs a polynomial expression for a cipher over some field using cipher input/output pairs and then he aims to determine its key-dependent coefficients. If the number of terms in the polynomial expression is reasonably small, the attack can be mounted. Lesamnta-LW uses the AES S-box which can be expressed as a polynomial of degree 254 over GF(2^8). Our experiments have confirmed that after the 16th round, each byte in the intermediate state of the mixing function depends on all the 32 variables while this is not the case just after the 15th round. We expect that the number of coefficients grows fast after the 16th round due to the high degree of the S-box and deduce that the full Lesamnta-LW is secure against interpolation attacks.

5.3 Impossible Differential Attack

In the impossible differential Attack [5], an attacker exploits differences that are impossible at some intermediate state of the cipher. The best impossible difference we have found is the difference of the form $(0, \Delta, 0, 0)$ at input $\rightarrow$ $(?, ?, ?, 0)$ after the 11th rounds. Note that the symbol ? denotes an arbitrary difference and Δ denotes non-zero difference. However, we expect that it is unlikely that impossible differential attacks can be successful against the full Lesamnta-LW.

5.4 Related-Key Attacks

In the related-key setting model, the attacker chooses the relation between the keys, which typically is a difference between the keys. We can show that the maximum differential characteristic probabilities for 24 rounds of the key scheduling function are less than 2^{-128} in the same way we did in Sect. 5.1. Hence, we expect that it is unlikely that related-key attacks can be successful against Lesamnta-LW because it is very difficult to find a high probability related-key differentials.

5.5 Collision Attacks Using Message Modification

Wang et al. [44], [45] showed methods for finding collisions for widely used hash functions such as SHA-1. Their approach is based on the differential cryptanalysis and the message modification technique which can be used to reduce the attack complexity by exploiting degrees of freedom in the input message.

For differential collision attacks on Lesamnta-LW, the attacker has to use messages of at least two blocks because the message block is shorter than the chaining variable. Using multiple block message, he has some control over 384 bits of the input to the compression function. However, out of these 384 bits, the only input bits over which he can have control in a deterministic way are 128 bits, which correspond to the message block input. He can have control over the remaining 256 bits corresponding to the chaining variable input only in a probabilistic way. On the other hand, we can show that the maximum differential characteristic probabilities for 44 rounds of the mixing function and for 24 rounds of the key scheduling function are less than 2^{-256} and 2^{-128} in the same way we did in Sect. 5.1. Their methods for finding collisions require a differential characteristic with a large probability and a large degree of freedom in the message block space. Thus, we expect that it is very unlikely that differential attacks with message modification are effective against Lesamnta-LW.

5.6 Attacks on the Lesamnta Compression Function Using Self-Duality

Recently, attacks [11] on the compression function of the SHA-3 Round-1 candidate Lesamnta [22] have been reported. The main idea is to find some structure in round constants. The block cipher of Lesamnta exhibits a correlation between keys, ciphertexts and plaintexts. This correlation is caused by the self-duality of the key schedule and the mixing function. Using the correlation, the block cipher of Lesamnta is easily distinguished from an ideal cipher, and a pseudo-collision for Lesamnta can be found with less complexity than expected. Note that the concept of self-duality was given as the property of the AES round function [28].

Since Lesamnta-LW has been designed in such a way that it does not have the self-duality property, similar attacks are not applicable to Lesamnta-LW. It is easy to destroy the self-duality, that is, it is sufficient that the round key looks like random. In the case of Lesamnta, since the 32-bit difference of a 64-bit round constant is periodically constant, it is easy to find a key such that the round key satisfies special conditions. In the case of Lesamnta-LW, round constants are generated with the linear feedback shift register that is based on the primitive polynomial with degree 32. Since the primitive polynomial has 17 non-zero coefficients, almost half of bits of the internal state may be changed by one operation. In addition, the mixing function of Lesamnta-LW was designed in such a way that the size of a round key is a half the size of G. This guarantees that the mixing function does not have the self-duality property independent of the value of a round key.

6. Implementation Results

We present software and hardware implementation results to show the flexibility of Lesamnta-LW for lightweight applications.

Table 1 Our ASIC implementation estimates of Lesamnta-LW, MAME, and SHA-256 with known results on other hash functions. The digest size of SHA-3 candidates is omitted.

Algorithm	Logic Process	Area (kGates)	Throughput (Mbit/s)	Clock (MHz)
BLAKE [41]	0.35 μm	25.57	15.4	31.25
Grϕstl[41]	0.35 μm	14.62	145.9	55.87
Skein [41]	0.35 μm	12.89	19.8	80
SHA-256 [18]	0.35 μm	10.9	22.5	50
Lesamnta-LW	90 nm	8.24	125.55	188.3
MAME	90 nm	12.95	1164.48	436.68
SHA-256	90 nm	14.6	1766	220.8

Table 2 Our estimates of RAM/ROM requirements on low-cost 8-bit CPUs

Algorithm	RAM(bytes)	ROM(CONST.)(bytes)
BLAKE[1]	96	172
Grøstl[21]	128	288
JH[47]	128	144
Keccak[4]	200	144
Skein[19]	96	46
SHA-256[33]	128	288
MAME[43]	64	40
Lesamnta-LW	50	768

6.1 Low-Area ASIC Implementation Results

We have estimated speed and gate count of a hardware architecture of Lesamnta-LW, MAME, and SHA-256. In Table 1, our results are compared to known results on the SHA-3 final round candidates such as BLAKE-32 [1], Grϕstl-224/256 [21], and Skein [19]. It is clear that Lesamnta-LW achieves a very small implementation and it is substantially smaller than most of them.

6.2 Software Implementation Results

For software, Lesamnta-LW is targeted at RAM requirement on an 8-bit CPU employed in smart devices. In low-cost 8-bit CPU applications, hash functions should require limited resources, memory and computation time. We argue that the most important constraint for hash functions is the limited RAM which could be critical in many cases.

6.2.1 8-bit CPU

We have estimated RAM/ROM requirements of SHA-3 candidates, SHA-256, and Lesamnta-LW. Our results are shown in Table 2. As for RAM requirement, it is clear that Lesamnta-LW achieves a very small implementation that is substantially smaller than most SHA-3 final round candidates. As for ROM requirement, we estimate the size of constants such as initial vectors, lookup tables, and round constants. Lesamnta-LW is larger than the other algorithms shown in Table 2. However, it is typical on 8-bit CPUs that the ROM size is large enough for symmetric-key algorithm implementations. We expect that the ROM requirement of Lesamnta-LW is reasonable.

Table 3 Our software implementation estimates on an 8-bit CPU Renesas® H8®. Three type values are shown depending on the implementation policy, namely ROM-optimized, RAM-optimized, and balanced.

Algorithm	Bulk Speed (cycles/ byte)	Short Message (cycles/ message)	ROM (CONST. +CODE) (byte)	RAM (byte)
SHA-256	1033.3	66434	32 + 37034	330
	1046.9	67308	288 + 5046	330
	1281.1	82296	288 + 948	330
Lesamnta-LW	1650.9	52828	512 + 20006	50
	1736.5	55568	768 + 1346	50
	2055.0	65760	768 + 370	54

Table 4 Our software implementation estimates on the Intel® Core i5™ processor where, for our estimate of the speed of SHA-256, we use the code used in OpenSSH.

Algorithm	Language	cycles/byte (32-bit mode)	cycles/byte (64-bit mode)
SHA-256	ANSI C	26.9	30.4
Lesamnta-LW	assembly	43.4	39.2

We have estimated speed and ROM/RAM size of Lesamnta-LW and SHA-256 on an 8-bit CPU Renesas® H8®in assembly language. The performance results are shown in Table 3 where by short message we mean a message whose length is less than 128 bits.

As for RAM requirement, it is clear that Lesamnta-LW gains advantages over SHA-256 with respect to speed on short messages and RAM/ROM requirements.

6.2.2 32-bit CPU

We have estimated the speed of Lesamnta-LW and SHA-256 on the Intel Core i5 processor which offers instructions for fast encryption of AES. Our results are shown in Table 4. Lesamnta-LW is reasonably fast on this platform.

7. Conclusion

A new lightweight 256-bit hash function Lesamnta-LW has been proposed. We claim that its distinct features over the existing lightweight primitives are compactness, high-speed, and a very good tradeoff between speed and cost on 8-bit CPUs as well as the high security levels with security reductions. We expect that Lesamnta-LW will open up a new set of lightweight applications.

Although we believe that the underlying block cipher of Lesamnta-LW withstands a number of recently proposed attacks because of our conservative design, more extensive analysis such as evaluation of security against rebound attacks would be needed.

Acknowledgments

We would like to mention the people who gave us valuable feedback and important comments on this work: Yasuko Fukuzawa, Kazuo Ota, and Kazuo Sakiyama.

References

[1] J.P. Aumasson, L. Henzen, W. Meier, and R.C.-W. Phan, "SHA-3 proposal BLAKE," http://131002.net/blake/

[2] J.P. Aumasson, L. Henzen, W. Meier, and M. Naya-Plasencia, "QUARK: A lightweight hash," Cryptographic Hardwareand Embedded Systems CHES 2010, LNCS, vol.6225, pp.1–15, 2010.

[3] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede, "Low-cost elliptic curve cryptography for wireless sensor networks," Security and Privacy in Ad-Hoc and Sensor Networks, ESAS 2006, LNCS, vol.4357, pp.6–17, 2007.

[4] G. Bertoni, J. Daemen, M. Peeters, and G.V. Assche, "Keccak specifications," http://keccak.noekeon.org/

[5] E. Biham, A. Biryukov, and A. Shamir, "Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials," Advances in Cryptology — EUROCRYPT'99, LNCS, vol.1807, pp.12–23, 1999.

[6] E. Biham and A. Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer, 1993.

[7] A. Biryukov and D. Wagner, "Advanced slide attacks," Advances in Cryptology — EUROCRYPT 2000, LNCS, vol.1807, pp.589–606, 2000.

[8] J. Black, P. Rogaway, and T. Shrimpton, "Black-box analysis of the block-cipher-based hash-function constructions from PGV," Advances in Cryptology — CRYPTO 2002, LNCS, vol.2442, pp.320–335, 2002.

[9] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, and C. Vikkelsoe, "PRESENT: An ultra-lightweight block cipher," Cryptographic Hardware and Embedded Systems CHES 2007, LNCS, vol.4727, pp.450–466, 2007.

[10] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M.J. B. Robshaw, and Y. Seurin, "Hash functions and RFID tags: Mind the gap," Cryptographic Hardware and Embedded Systems CHES 2008, LNCS, vol.5154, pp.283–299, 2008.

[11] C. Bouillaguet, O. Dunkelman, G. Leurent, and P.A. Fouque, "Another look at complementation properties," Fast Software Encryption 2010 Workshop FSE 2010, LNCS, vol.6147, pp.347–364, 2010.

[12] C.D. Canniére, O. Dunkelman, and M. Knezevic, "KATAN and KTANTAN a family of small and efficient hardware-oriented block ciphers," Cryptographic Hardware and Embedded Systems CHES 2009, LNCS, vol.5747, pp.272–288, 2009.

[13] F. Chabaud and S. Vaudenay, "Links between differential and linear cryptanalysis," Advances in Cryptology — EUROCRYPT'94, LNCS, vol.950, pp.356–365, 1995.

[14] J. Daemen and V. Rijmen, The Design of Rijndael: AES — Advanced Encryption Standard, Springer-Verlag, 2002.

[15] I.B. Damgård, "A design principle for hash functions," Advances in Cryptology — CRYPTO'89, LNCS, vol.435, pp.416–427, 1990.

[16] E. Fleischmann, C. Forler, and M. Gorski, "Classification of the SHA-3 candidates," eprint archive: http://eprint.iacr.org/2008/511

[17] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," Proc. Workshop of Cryptographic Hardware and Embedded Systems — CHES 2004, LNCS, vol.3156, pp.357–370, Springer, 2004.

[18] M. Feldhofer and C. Rechberger, "A case against currently used hash functions in RFID protocols," Proc. On the Move to Meaningful Internet Systems 2006, OTM 2006 Workshops, LNCS, vol.4227, pp.372–381, 2006.

[19] N. Ferguson, S. Lucks, B. Schneier, D. Whiting, M. Bellare, T. Kohno, J. Callas, and J. Walker, "The skein hash function family," http://www.schneier.com/skein.html

[20] G. Gaubatz, J.P. Kaps, E. Ozturk, and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks," Workshop on Pervasive Computing and Communication Security PerSec 2005.

[21] P. Gauravaram, L.R. Knudsen, K. Matusiewicz, F. Mendel, C. Rechberger, M. Schläffer, and S.S. Thomsen, "Grøstl — A SHA-

3 candidate," http://www.groestl.info/
[22] S. Hirose, H. Kuwakado, and H. Yoshida, "SHA-3 proposal: Lesamnta," http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/documents/Lesamnta.zip, Oct. 2008. latest version: http://www.sdl.hitachi.co.jp/crypto/lesamnta/
[23] S. Hirose, H. Kuwakado, and H. Yoshida, "Security analysis of the compression function of Lesamnta and its impact," http://csrc.nist.gov/groups/ST/hash/sha-3/Round1/documents/LESAMNTA_Comments.pdf, June 2009.
[24] T. Jakobsen and L.R. Knudsen, "The interpolation attack on block ciphers," Fast Software Encryption, FSE'97, LNCS, vol.1267, pp.28–40, 1997.
[25] A. Joux, "Multicollisions in iterated hash functions. Application to cascaded construction," Advances in Cryptology — CRYPTO 2004, Lect. Notes Comput. Sci., vol.3152, pp.306–316, 2004.
[26] J. Kelsey and B. Schneier, "Second preimages on n-bit hash functions for much less than 2^n work," Advances in Cryptology — EUROCRYPT 2005, LNCS, vol.3494, pp.474–490, 2005.
[27] L.R. Knudsen, "Truncated and higher order differentials," Fast Software Encryption, FSE'94, LNCS, pp.196–211, 1995.
[28] T.V. Le, R. Sparr, R. Wernsdorf, and Y. Desmedt, "Complementation-like and cyclic properties of AES round functions," Advanced Encryption Standard — AES, 4th International Conference, AES 2004, Lect. Notes Comput. Sci., vol.3373, pp.128–141, 2005.
[29] M. Matsui, "Linear cryptanalysis method for DES cipher," Advances in Cryptology — EUROCRYPT'93, LNCS, vol.765, pp.386–397, 1994.
[30] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, HANDBOOK of APPLIED CRYPTOGRAPHY, CRC Press, 1996.
[31] R.C. Merkle, "One way hash functions and DES," CRYPTO'89, LNCS, vol.435, pp.428–446, 1990.
[32] R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.
[33] National Institute of Standards and Technology, "Secure hash standard," Federal Information Processing Standards Publication 180-2, Aug. 2002. http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf
[34] National Institute of Standards and Technology, "The keyed-hash message authentication code (HMAC)," Federal Information Processing Standards Publication 198, March 2002.
[35] 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/, Nov. 2007.
[36] R. Rivest, "The MD5 message-digest algorithm," Request for Comments, no.1321, April 1992. ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt
[37] http://www.semico.com
[38] A. Shamir, "SQUASH — A new MAC with provable security properties for highly constrained devices such as RFID tags," Fast Software Encryption, FSE 2008, Lect. Notes Comput. Sci., vol.5086, pp.144–157, 2008.
[39] W. Simpson, "PPP challenge handshake authentication protocol (CHAP)," Request for Comments, no.1994, 1996. http://www.ietf.org/rfc/rfc1994.txt
[40] M.L. Songini, "Passive RFID tag market to hit $486M in 2013," InfoWorld, http://www.infoworld.com/t/networking/passive-rfid-tag-market-hit-486m-in-2013-102
[41] S. Tillich, M. Feldhofer, W. Issovits, T. Kern, H. Kureck, M. Muhlberghuber, G. Neubauer, A. Reiter, A. Kofler, and M. Mayrhofer, "Compact hardware implementations of the SHA-3 candidates ARIRANG, BLAKE, Grøstl, and Skein," eprint archive: http://eprint.iacr.org/2009/349.pdf
[42] G. Tsudik, "Message authentication with one-way hash functions," ACM Computer Communications Review, vol.22, no.5, pp.29–38, 1992.
[43] H. Yoshida, D. Watanabe, K. Okeya, J. Kitahara, H. Wu, Ö. Küçük, and B. Preneel, "MAME: A compression function with reduced hardware requirements," Cryptographic Hardware and Embedded Systems CHES 2009, LNCS, vol.4727, pp.148–165, 2007.
[44] X. Wang, X. Lai, D. Feng, H. Chen, and X. Yu, "Cryptanalysis of the hash functions MD4 and RIPEMD," Advances in Cryptology — EUROCRYPT 2005, LNCS, vol.3494, pp.1–18, 2005.
[45] X. Wang, Y.L. Yin, and H. Yu, "Finding collisions in the full SHA-1," Advances in Cryptology — CRYPTO 2005, LNCS, vol.3621, pp.17–36, 2005.
[46] Wikipedia, "Microprocessor," ch. Market statistics, http://en.wikipedia.org/wiki/Microprocessor
[47] H. Wu, "The hash function JH," http://www3.ntu.edu.sg/home/wuhj/research/jh/
[48] Y. Zheng, T. Matsumoto, and H. Imai, "On the construction of block ciphers provably secure and not relying on any unproved hypotheses," Advances in Cryptology — CRYPTO'89, LNCS, vol.435, pp.461–480, 1990.

Appendix A: Lesamnta-LW Example

The 32-bit round constants $C^{(r)}$ are

```
a432337f 945e1f8f 92539a11 24b90062
6971c64c d6e3f449 2c2f0da9 33769295
eb506df2 708cebfe b83ab7bf 97df0f17
9223b802 7fa29140 0ff45228 01fe8a45
ed016ee8 1da02ddd ee8aba1b 46c4c223
53cd0d24 d1b46d24 c1fb4124 c3f2a4a4
c3b39814 c3bbbf82 759191b0 0eb23236
b7fd6c86 a0d48750 141a90ea 6f65b45d
e0d2092b 470fd445 e5df4528 1cbbe8a5
eea9c2b4 c618f4d6 aee8345a 783be0cb
5412e979 3c712e0f 87567c21 2619bca4
df0efb14 c02c13e2 75e3643c d571a007
9a766de0 134ecdbc d9a41537 9becdb46
a556b1a8 14aad635 efabe566 abde566c
ceb6064d f4e87f69 286e7ccd e8337039
2bf51d27 85a6fa44 cb7913c8 196f2279
```

For Lesamnta-LW, the initial hash value $H^{(0)}$ is $H_0^{(0)}||H_1^{(0)}||H_2^{(0)}||H_3^{(0)}||H_4^{(0)}||H_5^{(0)}||H_6^{(0)}||H_7^{(0)}$, where each $H_i^{(0)}$ is a 32-bit word `00000256` in hex.

Let the message M be the 24-bit ($l = 24$) ASCII string "**abc**", which is equivalent to the following binary string: `01100001 01100010 01100011`. Then the resulting 256-bit message digest is

```
2558c1d3 7f9f307b e3cddad4 a23c8654
518f6079 7eb491e7 3758727d fc83de65.
```

Appendix B: Proof of Lemma 2

Let $\mathcal{B} = \{0,1\}^{n/2}$. Let $\mathcal{B}^{\le i} = \bigcup_{d=0}^{i} \mathcal{B}^d$. Let $M_{[1,l]} = M_1||M_2||\cdots||M_l$ for $1 \le l \le \ell$. For $i \in \{0,1,\ldots,\ell\}$ ($\ell \ge 1$), let $I_i : \mathcal{B}^{\le \ell} \to \mathcal{B}$ be a random function such that $I_i(M_{[1,l]})$ equals

$$\begin{cases} \alpha_1(M_{[1,l]}) & \text{if } l \le i, \\ \mathsf{kiv\text{-}LW}^E(\alpha_0(M_{[1,i]})||\alpha_1(M_{[1,i]}), M_{[i+1,l]}) & \text{otherwise,} \end{cases}$$

where α_0 and α_1 are random functions uniformly distributed over $\mathcal{F}(\mathcal{B}^i, \mathcal{B})$ and $\mathcal{F}(\mathcal{B}^{\leq i}, \mathcal{B})$, respectively. Notice that α_0 and α_1 are just random elements from $\mathcal{B}$ if $i = 0$. Let $P_i = \Pr[A^{I_i} = 1]$. Then,

$$\mathrm{Adv}^{\mathrm{prf}}_{\mathsf{kiv\text{-}LW}^E}(A) = \left|P_0 - P_\ell\right|.$$

Let us consider the following prf-adversary B with q oracles $u_1, \ldots, u_q$ using A as a subroutine.

B first selects i from $\{1, \ldots, \ell\}$ uniformly at random. Then, B runs A. B simulates a random function β uniformly distributed over $\mathcal{F}(\mathcal{B}^{\leq i-1}, \mathcal{B})$ via lazy sampling. When B receives the k-th query $M^{(k)} = M^{(k)}_{[1,l]}$ of A, B returns

$$\begin{cases} \beta(M^{(k)}_{[1,l]}) & \text{if } l \leq i-1, \\ \omega(u_{\mathsf{idx}(k)}(M^{(k)}_i \| \beta(M^{(k)}_{[1,i-1]}))) & \text{if } l = i, \\ \mathsf{kiv\text{-}LW}^E(\upsilon(M^{(k)}_{[1,i]}), M^{(k)}_{[i+1,l]}) & \text{if } l \geq i+1, \end{cases}$$

where ω is a function which outputs the second half of its input, and $\upsilon(M^{(k)}_{[1,i]}) = u_{\mathsf{idx}(k)}(M^{(k)}_i \| \beta(M^{(k)}_{[1,i-1]}))$. $\mathsf{idx}(k)$ is a unique integer in $\{1, \ldots, q\}$. If there is a previous query $M^{(p)}$ $(p < k)$ such that $M^{(p)}_{[1,i-1]} = M^{(k)}_{[1,i-1]}$, then $\mathsf{idx}(k) = \mathsf{idx}(p)$. Otherwise, $\mathsf{idx}(k) = k$.

Now, suppose that B has oracle access to $E_{K_1}, E_{K_2}, \ldots, E_{K_q}$, where K_j's are independent random variables uniformly distributed over $\mathcal{B}$. Then, in response to $M^{(k)}_{[1,l]}$, B returns

$$\begin{cases} \beta(M^{(k)}_{[1,l]}) & \text{if } l \leq i-1, \\ \mathsf{kiv\text{-}LW}^E(K_{\mathsf{idx}(k)} \| \beta(M^{(k)}_{[1,i-1]}), M^{(k)}_{[i,l]}) & \text{if } l \geq i. \end{cases}$$

Since $K_{\mathsf{idx}(k)}$ can be regarded as a random function of $M^{(k)}_{[1,i-1]}$, we can say that A has oracle access to I_{i-1}. Therefore,

$$\Pr[B^{E_{K_1}, \ldots, E_{K_q}} = 1] = \frac{1}{\ell} \sum_{i=1}^{\ell} P_{i-1}.$$

Next, suppose that B has oracle access to $\rho_1, \ldots, \rho_q$, where ρ_j's are independent random functions uniformly distributed over $\mathcal{F}(\mathcal{B}^2, \mathcal{B}^2)$. Since the first half and the second half of $\rho_{\mathsf{idx}(k)}(M^{(k)}_i \| \beta(M^{(k)}_{[1,i-1]}))$ are independent random functions of $M^{(k)}_{[1,i]}$, we can say that A has oracle access to I_i. Therefore,

$$\Pr[B^{\rho_1, \ldots, \rho_q} = 1] = \frac{1}{\ell} \sum_{i=1}^{\ell} P_i.$$

From the discussions above,

$$\begin{aligned} \mathrm{Adv}^{q\text{-prf}}_E(B) &= \left|\Pr[B^{E_{K_1}, \ldots, E_{K_q}} = 1] - \Pr[B^{\rho_1, \ldots, \rho_q} = 1]\right| \\ &= \frac{1}{\ell} \mathrm{Adv}^{\mathrm{prf}}_{\mathsf{kiv\text{-}LW}^E}(A). \end{aligned}$$

B makes at most q queries and runs in time at most $t + O(\ell q T_E)$.

Appendix C: Proof of Lemma 3

Let $\mathcal{B} = \{0,1\}^{n/2}$. Let $K_1, \ldots, K_m$ be independent random variables uniformly distributed over $\mathcal{B}$. Let $\rho_1, \ldots, \rho_m$ be independent random functions uniformly distributed over $\mathcal{F}(\mathcal{B}^2, \mathcal{B}^2)$. Let $\varpi_1, \ldots, \varpi_m$ be independent random permutations uniformly distributed over $\mathcal{P}(\mathcal{B}^2)$. Then,

$$\begin{aligned} &\mathrm{Adv}^{m\text{-prf}}_E(A) \\ &= \left|\Pr[A^{E_{K_1}, \ldots, E_{K_m}} = 1] - \Pr[A^{\rho_1, \ldots, \rho_m} = 1]\right| \\ &\leq \left|\Pr[A^{E_{K_1}, \ldots, E_{K_m}} = 1] - \Pr[A^{\varpi_1, \ldots, \varpi_m} = 1]\right| \\ &\quad + \left|\Pr[A^{\varpi_1, \ldots, \varpi_m} = 1] - \Pr[A^{\rho_1, \ldots, \rho_m} = 1]\right|. \end{aligned}$$

For $0 \leq i \leq m$, let O_i be m oracles such that $E_{K_1}, \ldots, E_{K_i}, \varpi_{i+1}, \ldots, \varpi_m$.

A prp-adversary B is constructed using A as a subroutine. B has an oracle u, which is either E_K or ϖ, where K is a random variable uniformly distributed over $\mathcal{B}$ and ϖ is a random permutation uniformly distributed over $\mathcal{P}(\mathcal{B}^2)$. B first selects i uniformly at random from $\{1, 2, \ldots, m\}$. Then, B runs A with oracles $E_{K_1}, \ldots, E_{K_{i-1}}$, u, $\varpi_{i+1}, \ldots, \varpi_m$, and outputs A's output. Then,

$$\begin{aligned} \mathrm{Adv}^{\mathrm{prp}}_E(B) &= \left|\Pr[B^{E_K} = 1] - \Pr[B^{\varpi} = 1]\right| \\ &= \frac{1}{m}\left|\Pr[A^{O_m} = 1] - \Pr[A^{O_0} = 1]\right|. \end{aligned}$$

B makes at most q queries and runs in time at most $t + O(q\,T_E)$.

It is possible to distinguish $\varpi_1, \ldots, \varpi_m$ and $\rho_1, \ldots, \rho_m$ only by the fact that there may be a collision for ρ_i's. Thus, since A makes at most q queries,

$$\left|\Pr[A^{\varpi_1, \ldots, \varpi_m} = 1] - \Pr[A^{\rho_1, \ldots, \rho_m} = 1]\right| \leq \frac{q(q-1)}{2^{2n+1}}.$$

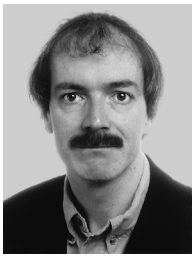
Trademarks

Shoichi Hirose received the B.E., M.E. and D.E. degrees in information science from Kyoto University, (Japan) in 1988, 1990 and 1995, respectively. From 1990 to 1998, he was a research associate at Faculty of Engineering, Kyoto University. From 1998 to 2005, he was a lecturer at Graduate School of Informatics, Kyoto University. From 2005 to 2009, he was an associate professor at Faculty of Engineering, University of Fukui. From 2009, he is a professor at Graduate School of Engineering, University of Fukui. His current interests include cryptography and information security. He received Young Engineer Award from IEICE in 1997, and KDDI Foundation Research Award in 2008. He is a member of ACM, IEEE, IACR and IPSJ.

Kota Ideguchi received the B.S., M.S., and Ph.D. degrees from the University of Tokyo (Japan) in 2001, 2003, and 2006, respectively. He is a researcher of Hitachi, Ltd. His current research focuses on symmetric-key cryptography.

Hidenori Kuwakado received the B.E., M.E. and D.E. degrees from Kobe University (Japan) in 1990, 1992, and 1999 respectively. He worked for Nippon Telegraph and Telephone Corporation from 1992 to 1996. From 1996 to 2002 he was a Research Associate in the Faculty of Engineering, Kobe University. From 2002 to 2007, he was an Associate Professor in the Faculty of Engineering, Kobe University. Since 2007, he has been an Associate Professor in Graduate School of Engineering, Kobe University. His research interests are in cryptography and information security.

Toru Owada received the B.E., and M.E. degrees from Hokkaido University (Japan) in 1992 and 1994 respectively. He is a researcher of Hitachi, Ltd. His current interests include the hardware implementation of cryptographic algorithms.



Bart Preneel received the Electr. Eng. and Ph.D. degrees from the Katholieke Universiteit Leuven (Belgium) in 1987 and 1993. He is a full professor in the COSIC research group at the Katholieke Universiteit Leuven in Belgium. He has authored more than 400 scientific publications and is inventor of 3 patents. His main research interests are cryptography and information security (e.g., privacy and e-voting) and he frequently consults on these topics. He is president of the IACR (International Association for Cryptologic Research). He has served as program chair of 14 international conferences and he has been invited speaker at more than 70 conferences in 30 countries.

Hirotaka Yoshida received the B.S. degree from Meiji University (Japan) in 1999 and M.S. degree from Tokyo Institute of Technology (Japan) in 2001. He is a researcher of Hitachi, Ltd. He also is a research assistant in the COSIC research group at the Katholieke Universiteit Leuven in Belgium. His current research focuses on symmetric-key cryptography. He is a member of the IACR (International Association for Cryptologic Research).