A new (n, n) Blockcipher based Hash Function for Short Messages

Atsuko Miyaji
School of Information Science
Japan Advanced Institute of
Science and Technology
Nomi, Ishikawa, Japan, 923-1292
Email: miyaji@jaist.ac.jp

Mazumder Rashed
School of Information Science
Japan Advanced Institute of
Science and Technology
Nomi, Ishikawa, Japan, 923-1292
Email: rashed@jaist.ac.jp

Tsuyoshi Sawada
School of Information Science
Japan Advanced Institute of
Science and Technology
Nomi, Ishikawa, Japan, 923-1292

***Abstract*—We propose a new (n, n) double block length hash function where collision and preimage security bound is respectively $O\left(2^{tn}\right)$ and $O\left(2^{2tn}\right)$. The strategic point of this scheme is able to handle short message tn $(t < 1)$ bit, which is very significant issue for RFID tag security. It is known that the RFID tag needs to proceed short message but MDC-2, MDC-4, MJH are not properly suitable for meeting this criteria due to their constructions where these schemes can handle message size n bit $(n = 128)$. Additionally the security bound of the proposed scheme is better than other (n, n) blockcipher based hash such as MDC-2, MDC-4, MJH and as well as obtaining higher efficient rate.**

***Keywords*-Hash function, Blockcipher, SBL, DBL, Collision resistance, Preimage resistance.**

I. Introduction

Over the year, digital wireless technology being lead to many electrifying expansions including the fast intensification of mobile and ubiquitous computing. Using mobile applications and devices implanted in the surrounding environment, clients can get access of apparent computing and communication services at all times and in all places in near future. Applications of wireless technology such as smart phones, smart auto-mobiles and smart homes have already begun to popular and very much essential for us. The key point is to ensure security. Digital devices or RFID tags will play a key factor in the near future for the development of wireless computing. The RFID applications are spreading in our daily life so rapidly. So it is very much targeted for scientists to develop low-cost RFID tags for access control, inventory control, luggage tracking, library, office-appliance and product tractability etc. In 2008, it is recommended by European Commission that all kind of RFID applications need to operate in a secure manner which tends to do research for high-performance and low-cost security solutions for RFID devices [1]. So it is very interesting and challenging for the scientists to achieve a balance between cost and security issue for RFID tags.

It is founded that for RFID security protocols a wide variety of cryptographic algorithms can be used where cryptographic hash functions is being used vastly by RFID security protocol designers. A cryptographic hash function is a function which maps an input of arbitrary length to an output of fixed length, where it needs to satisfy at least collision, preimage and second-preimage resistance [2]. Current standards and state-of-the-art low-power implementation techniques favour the use of block ciphers such as Advanced Encryption Standard (AES) instead of hash functions from the SHA family. The AES module requires only a third of the chip area and half of the mean power. Smaller hash functions like SHA-1, MD5 and MD4 are also less suitable for RFID tags than the AES. Inclusively it can be said that the total power consumption of SHA-1 is about 10% higher than that of AES [3]. Recently there are several successful attacks on MD4/5 and SHA-family type functions [4], [5] so that current researchers are more interested on blockcipher based hash function.

Block-cipher based hash functions are classified into single-block-length (SBL) and double-block-length (DBL). The output length of SBL hash function is equal to the block length and DBL hash function is the twice of block length. It is well-known that due to birthday attack collision resistance of a hash function can be occurred with time complexity $O(2^{l/2})$ (l is the output length of hash function) where widely used block ciphers are 64/128 bit length, so SBL hash function is no longer secure in terms of CR. DBL hash function comes in various pretexts, depending on the number of blockcipher calls per compression function and the bit-length of the key (block-cipher) such as one call to a $2n$-bit key, two calls to a $2n$-bit key, two calls to an n-bit key. In this article, proposed construction is based on the last variant, where it is shown that, how to construct a compression function with $2n$ bit output using a component function with n-bit output (the component function is defined as blockciper). From the above table I. current research status of (n, n) and $(n, 2n)$ blockcipher hash functions have been found. Point to be noted, the actual efficiency rate of the construction of Stam and Luck is very low due to uses of full finite field multiplication [10]. In another research it is found that, (n, n) based blockcipher hash function is 40% faster than $(n, 2n)$ blockcipher hash function and as well as cost efficient because of less key size [11]. That's why currently scientists are more focused for (n, n) based

Table I
DIFFERENT $(n, 2n)$ AND (n, n) BASED BLOCKCIPHER RESULT ANALYSIS

	CF	r	E	KS	CR	PR
Weimar [6]	$3n \to 2n$	$\frac{1}{2}$	2	2	$\mathcal{O}(2^n)$	$\mathcal{O}(2^{2n})$
Hirose [15]	$3n \to 2n$	$\frac{1}{2}$	2	1	$\mathcal{O}(2^n)$	$\mathcal{O}(2^{2n})$
Abreast [16]	$3n \to 2n$	$\frac{1}{2}$	2	2	$\mathcal{O}(2^n)$	$O(2^{2n})$
Tandem [14]	$3n \to 2n$	$\frac{1}{2}$	2	2	$\mathcal{O}(2^n)$	$\mathcal{O}(2^{2n})$
Luck's [18]	$3n \to 2n$	1	1	2	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(2^n)$
Stam's [18]	$3n \to 2n$	1	1	2	$\mathcal{O}(2^n)$	$\mathcal{O}(2^n)$
MDC-2 [7]	$3n \to 2n$	$\frac{1}{2}$	2	2	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(2^n)$
MDC-4 [7]	$3n \to 2n$	$\frac{1}{4}$	4	1	$\mathcal{O}(2^{5n/8})$	$\mathcal{O}(2^{5n/4})$
MJH [11]	$3n + c \to 2n$	$\frac{1}{2}$	2	1	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(2^n)$
MSR (Prop.)	$2n + tn \to 2n$	t	2	2	$\mathcal{O}(2^{tn})$	$\mathcal{O}(2^{2tn})$

CF: Compression Function
r: Efficiency Rate
E: Number of Block Cipher Call
KS: Key Schedule
CR: Collision Resistance
PR: Preimage Resistance

Table II
AES: INFLUENCE OF THE KEY SIZE ON THE ENERGY CONSUMPTION OF A MICAZ SENSOR MOTE. [9]

	KS		**Enc.**		**Dec.**	
	(ms)	**(μJ)**	**(ms)**	**(μJ)**	**(ms)**	**(μJ)**
AES-128	2.44	62.32	1.53	39.08	3.52	89.90
AES-192	2.68	68.45	1.82	46.48	4.52	108.55
AES-256	3.01	76.88	2.11	53.89	4.98	127.19

blockcipher hash function.

Another critical issue is to measure the cost of security under RFID tags or WSN's devices. For better understanding of this cost in the aspect of WSNs security three key-points mentioned in the above table II. (encryption algorithms, modes of operation for block ciphers, and message authentication algorithms) where AES generations have been measured and compared through memory and energy consumption on the basis of MicaZ sensor motes. AES-128 is more user friendly because of less power consumption, less encryption and decryption time. Another dominating issue is short message has been used for WSN's device or RFID tags. Interestingly AES-128 based hash function such as MDC-2, MDC-4, MJH are not properly fit for this issue because of their construction, which can deal message size n bit ($n = 128$). So our one of the motivation is to develop (n, n) blockcipher hash function which can deal short message and make sure the security of RFID tags or WSN's device.

Our Contribution. In this article, a new construction of double block length hash function is being proposed with (n, n) based blockcipher. It's CR and PR security bound are respectively $O\left(2^{tn}\right)$ and $O\left(2^{2tn}\right)$. The result of this construction is better than existing other (n, n) based blockcipher hash function and also this scheme is suitable for providing security to RFID tags/ WSN's device because of capability of handling short message. Additionally it can be said that the efficient rate is higher than MDC-2, MDC-4 and MJH.

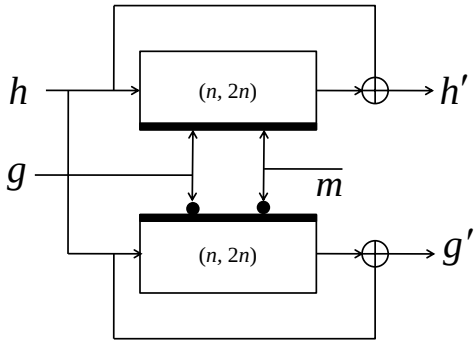

Figure 1. The Weimar-DM compression function, black-circles denote a bit complement, where key scheduling is twice.

Outline. The paper is organized as follows: Section II gives related work, Section III preliminaries and notations. In Section IV, description of new scheme is being presented. Both CR and PR security proof can be found in section V. Result will be analysed in section VI. Finally in section VII, it has been provided the limitations and future work.

II. RELATED WORK

Weimar-DM [6] double block length hash functions has been proposed by Fleischmann, Forler, Lucks and Wenzel whose CR and PR bound is respectively by $\mathcal{O}(2^n)$ and $\mathcal{O}(2^{2n})$. Another famous two schemes of $3n$-bit to $2n$-bit compression function is Abreast-DM and Tandem-DM pictured in Fig. 2, which was proposed by Lai and Massey [12]. The CR of Abreast-DM was resolved by Lee, Kwon [13] and also Tandem-DM CR was proved by Lee, Stam and Steinberger [14]. The CR of these two scheme is $\mathcal{O}(2^n)$. Later in 2011, Lee, Stam and Steinberger improved the PR security bound $\mathcal{O}(2^n)$ to $\mathcal{O}(2^{2n})$. In FSE 2006, Hirose [15] proposed another famous construction and showed that it was bound in $\mathcal{O}(2^n)$ for the CR and $\mathcal{O}(2^n)$ for the PR. Later this PR was being improved by Lee, Stam and Steinberger [16]. The construction of Hirose was further generalized by Ozen and Stam [10], who additionally discuss schemes that are only secure in the iteration. Hiroses construction (Fig. 3) is simpler than either Abreast-DM or Tandem-DM and in particular uses a single keying schedule for the top and bottom blockciphers, whereas Weimar-DM and other two have double key scheduling. Discussed above all constructions are based on $(n, 2n)$ blockcipher. Surprisingly all of these constructions have same efficiency rate which is $1/2$. Another interesting point is that accept Hirose-DM, mentioned all constructions are followed by double KS. Below it is defined how to measure efficiency of blockcipher based hash.

$$r = \frac{|M_i|}{(no.\ of\ blockcipher\ calls\ in\ F) \times n\ (block\ length)}$$

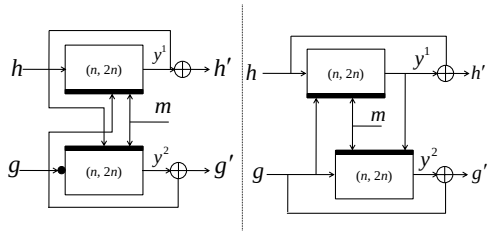

Figure 2. First figure represents the Tandem-DM compression function and second figure stand for the Abreast-DM, where black-circle in the bottom row denotes the bit inversion.

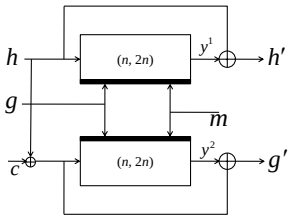

Figure 3. A compression function for Hirose-DM, classified for Cyclic-DM

$$r_{key} = \frac{1}{(no.\ of\ key\ schedule/compression\ function)}$$

The newly proposed construction is based on (n, n) blockcipher instead of $(n, 2n)$ blockcipher which has been achieved higher efficiency rate but key schedule is twice as Weimar, Tandem and Abreast-DM. The well-known $3n$-bit to $2n$-bit compression functions such as Tandem and Abreast-DM share the feature that the inputs to the top and bottom blockcipher are bi-jectively related. For example, for Abreast-DM, if the top blockcipher call is $E_{g||m}(h)$ then the bottom blockcipher call (for the same input h, g) is $E_{h||m}(\bar{g})$, where $\bar{g}$ denotes bit complementation of g; thus the inputs to the top and bottom blockciphers are related. Fleischmann [17] classified that these two constructions belong the Parallel-DM, whereas Tandem-DM belongs the Serial-DM. If it is more classified then, it can be said that Hirose and Abreast-DM followed by Cyclic-DM (subgroup of Parallel-DM): the cycle length of Hirose and Weimar-DM is 2 where Abreast-DM is more than 2. Here newly proposed scheme follows same group as Hirose and Weimar-DM.

Now try to focus some previously proposed (n, n) blockcipher based hash functions. The famous two schemes MDC-2 and MDC-4 have been bound CR and PR respectively by $\left(O\left(2^{3n/5}\right), O\left(2^{n}\right)\right)$ and $\left(O\left(2^{5n/8}\right), O\left(2^{5n/4}\right)\right)$ [18], [19]. It is noted here that the efficiency rate of MDC-2 is $1/2$ which is half of MDC-4. Another famous MJH (n, n) blockcipher based hash function's CR and PR bound is as $O\left(2^{n/2}\right)$ and $O\left(2^{n}\right)$.

III. Preliminaries

A. ideal cipher model

A *blockcipher* is a keyed function $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ (assume that, (k, n, l) be integers). For each $k \in \{0,1\}^k$, the function $E_k(\cdot) = E(k, \cdot)$ is a permutation on $\{0,1\}^n$. If E is a block cipher then E^{-1} denotes its inverse, where $E_k(x) = y$ and $E_k^{-1}(y) = x$, is called forward and backward query respectively. Assume that, Block (k, n) be the family of all block ciphers $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. A *blockcipher* based hash function is a hash function $H : \{0,1\}^* \to \{0,1\}^l$ and $E \in \mathsf{Block(k, n)}$ is the block cipher used in the round function of H. Using a block cipher $E \in \mathsf{Block(k, n)}$, an adversary is given access to two oracles E and E^{-1} which are known as forward and backward query. Hence, for the any i^{th} query-response q_i keeps the record as:

$$q_i = \begin{cases} (k_i, x_i, y_i)\, if E \\ (k_i, y_i, x_i)\, if E^{-1} \end{cases}$$

In the ideal cipher model, the complexity of an attack is measured by the total number of the optimal adversary's queries to the two oracles E and E^{-1}.

B. iterated hash function

A hash function, $H : \{0,1\}^* \to \{0,1\}^l$ generally forms of a compression function $F : \{0,1\}^l \to \{0,1\}^{l'}$ and fixed initial value $IV \in \{0,1\}^l$, where input m is divided into the l'-bit blocks such as $m_1, m_2, ...m_l$. It implies that,

$$h_i = F(h_{i-1}, m_i) \qquad \text{for, } 1 \le i \le l$$

where H is called iterated hash function. This hash function and above discussed blockcipher $E \in \mathsf{Block}(k, n)$ have been used for the round function of H. If, $l = n$, then H is called a single block length (SBL) hash function, e.g., the PGV hash functions [20]. If, $l = 2n$, then H can be called as a double block length (DBL) hash function. Ideal cipher model is the formal model for the security analysis of blockcipher-based hash functions, which is dating back to Shannon [21] and widely used in [22].

C. security definition

An adversary is a computationally unbounded but always-halting collision-finding algorithm $\mathcal{A}$ with resource-bounded access to an oracle $E \in \mathsf{Block}(k, n)$, that means in the collision resistance experiment, a computationally unbounded adversary $\mathcal{A}$ is given oracle access to a blockcipher E uniformly sampled among all blockciphers of key length n and word length n. It is allowed that, $\mathcal{A}$ can make query to a both E and E^{-1}. For the any query q to E, the query history of $\mathcal{A}$ is the set of triplets $Q = (X_i, Y_i, K_i)$ such that $E(K_i, X_i) = Y$ and $\mathcal{A}$'s i^{th} query is either $E(K_i, X_i)$ or $E^{-1}(K_i, Y_i)$.

The query history, which is denoted by $\mathcal{Q}$, is the tuple $(Q_1, Q_2, ..., Q_q)$ where $Q = (X_i, Y_i, K_i)$ is the result of the

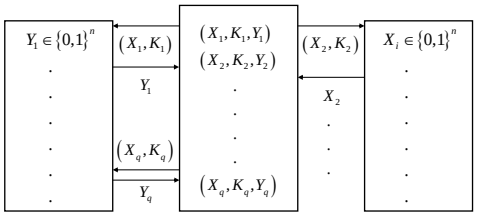

Figure 4. Adversary query to the query oracle database.

i^{th} query made by the adversary and where q is the total number of queries made by the adversary. The convention is that $\mathcal{A}$ asks at most only once on a triplet of a key K_i, a plaintext X_i and a ciphertext Y_i which are being obtained by a query and the corresponding reply.

Definition 1. Collision resistance of a hash function: The adversary $\mathcal{A}$ is given oracle access and H be blockcipher based hash function, then the advantage of $\mathcal{A}$ in finding collisions in H is:

$$Adv_H^{COLL}(\mathcal{A}) = \Pr\left[\begin{array}{l} E \leftarrow B(k,n); (M, M') \leftarrow \mathcal{A}^{E,E^{-1}} : \\ M \neq M' \wedge H^E(M) = H^E(M') \end{array}\right]$$

For $q \geq 1$, it can be expressed that $Adv_H^{COLL}(q) = \max_{\mathcal{A}}\left\{Adv_H^{COLL}\right\}$, where the maximum is taken over all adversaries which can query at best q oracle queries.

Definition 2. Collision resistance of a compression function: The adversary $\mathcal{A}$ is given oracle access and f be blockcipher based hash function, then the advantage of $\mathcal{A}$ in finding collisions in f is:

$$Adv_f^{COMP}(\mathcal{A}) = \Pr\left[\begin{array}{l} E \leftarrow B(k,n); (h,g,m), (h',g',m') \leftarrow A^{E,E^{-1}} : \\ (h,g,m) \neq (h',g',m') \wedge f^E(h,g,m) = \\ f^E(h',g',m') \vee f^E(h,g,m) = (h_0, g_0) \end{array}\right]$$

For $q \geq 1$, it can expressed that $Adv_f^{COMP}(q) = \max_{\mathcal{A}}\left\{Adv_f^{COMP}\right\}$, where the maximum is taken over all adversaries which can query at best q oracle queries.

Definition 3. Preimage resistance of a compression function: The adversary $\mathcal{A}$ is given oracle access to a block cipher $E \in B(K, X)$ and f be blockcipher based hash function. $\mathcal{A}$ arbitrary selects a value of (h', m') before making any query to oracle either E or E_1. Then the advantage of $\mathcal{A}$ in finding preimage in f is:

$$Adv_f^{pre}(\mathcal{A}) = \Pr[H(g,h,m) = (h',g')]$$

For $q \geq 1$, it can expressed that $Adv_f^{PRE}(q) = \max_{\mathcal{A}}\left\{Adv_f^{PRE}\right\}$, where the maximum is taken over all adversaries which can query at best q oracle queries.

IV. A new (n, n) double block length hash function

In this section, a new (n, n) double block length hash function hash been discussed with diagram which is defined as MSR scheme. In this scheme variable message size has been used which also varies the the value of security of hash. In figure 6 and 7 the achievement of MSR scheme has been deduced.

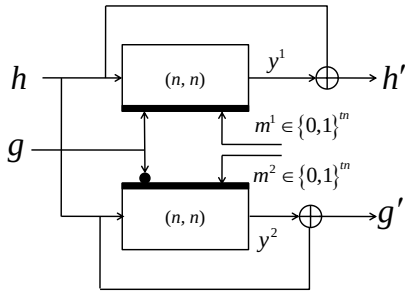

Figure 5. MSR Scheme

Definition 4. *Let $E(k, n)$ be a block cipher taking an k : n-bit key and an n-bit block size. The compression function $H^{MSR} : \{0,1\}^n \times \{0,1\}^{n-tn} \times \{0,1\}^{2tn} \rightarrow \{0,1\}^{2n}$ is defined as Def. 5 and Fig. 5.*

$$H^{MSR}(g, h, m^1, m^2) = (E_{g||m^1}(h) \oplus h, E_{\bar{g}||m^2}(h) \oplus h)$$

Definition 5. *Let $F : \{0,1\}^n \times \{0,1\}^{n-tn} \times \{0,1\}^{2tn} \rightarrow \{0,1\}^{2n}$ be a compression function such that $(g_i, h_i, m_i^1, m_i^2) = F(g_i, h_i, m_i^1, m_i^2)$ where, $h_i \in \{0,1\}^n, g_i \in \{0,1\}^{n-tn}, (m^1, m^2) \in \{0,1\}^{tn}$. F consists of $((n+m) = k, n)$ ideal block cipher E as like,*

$$h_i = F_T(h_{i-1}, g_{i-1}, m_i^1) = e(h_{i-1}, g_{i-1}||m_i^1) \oplus h_{i-1}$$
$$g_i = F_B(h_{i-1}, \bar{g}_{i-1}, m_i^2) = e(h_{i-1}, \bar{g}_{i-1}||m_i^2) \oplus h_{i-1}$$

In simplified way it can be expressed as like: (where $K_T, X_T, Z_T, K_B, X_B, Z_B$ are uniquely defined from $h_{i-1}, g_{i-1}, m_i^1, m_i^2$.)

$$\begin{cases} h_i = e_T(K_T, X_T) \oplus Z_T \\ g_i = e_B(K_B, X_B) \oplus Z_B \end{cases}$$

V. Security Analysis of the New MSR Scheme

A. collision security analysis

For any collision resistance finding experiment, a computationally unbounded adversary $\mathcal{A}$ is given oracle access to a blockcipher E uniformly sampled among all blockciphers of key length n and message length n. $\mathcal{A}$ is allowed to query both E and E^{-1}. After q queries to E, the query history of $\mathcal{A}$ is the set of triples Q= (X_i, K_i, Y_i) such that $E(K_i, X_i) = Y_i$ and $\mathcal{A}'s$ i^{th} query is either $E(K_i, X_i)$ or $E^{-1}(K_i, Y_i)$ for $1 \leq i \leq q$. Assume that $\mathcal{Q}$=$\{(X_i, K_i, Y_i)\}_{j=1}^i$ be the first i elements of the query history. Then it can be

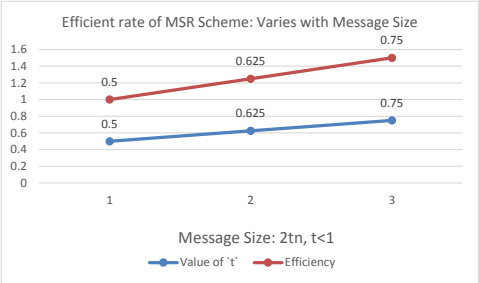

Figure 6. MSR Efficiency

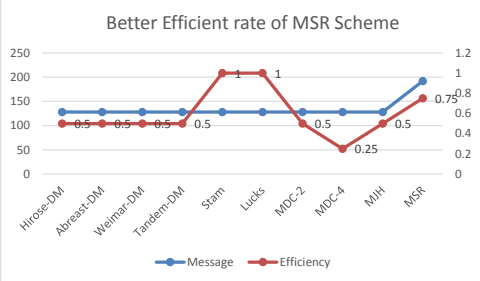

Figure 7. Efficiency Compare with Other Scheme

said that $\mathcal{A}$ succeeds or finds a collision after its first i queries if there exist distinct (h, g, m^1), $(h, \bar{g}, m^2)$ such that $H^{MSR}(h, g, m^1) = H^{MSR}(h, \bar{g}, m^2)$. As well as that $\mathcal{Q}$ contains both the queries necessary to compute $H^{MSR}(h, g, m^1)$ and $H^{MSR}(h, \bar{g}, m^2)$.

Theorem 1: Let H^{MSR} be a double-length hash function composed of compression function F specified in Def. 1. Then the advantage of an adversary in finding a collision in H^{MSR} after q queries can be upper bounded by:

$$\mathrm{Adv}_H^{COLL}(q) \leq \frac{q(q+1)}{2^{2tn}} + \frac{q}{2^{tn}}$$

Proof: It is assumed that the adversary has made any relevant query to E or E^{-1} which can occur collision in the ideal cipher model. Another issue is, the adversary never makes a query which is already available at his query database. In formal meaning, one can assume that the adversary never makes a query $E(K, X) = Y$ obtaining an answer Y and then makes the query $E^{-1}(K, Y) = X$ (which will necessarily be answered by X). At first consider, adversary $\mathcal{A}$ which is able to make an arbitrary q-query collision. Let $\mathcal{A}$ be a collision-finding algorithm of H^{MSR} with oracles E, E^{-1}. A asks q pairs of queries to E, E^{-1} in total. Since, h' and g' depends both on the plaintext and ciphertext of E/E^{-1}. One of them is fixed by a query and other is determined randomly from the query-database Q. As a result h', g' selected randomly from the query and query-oracle-database.

At first one of the important issue should be raised here, when adversary makes a query using blockcipher, the plaintext or ciphertext size are respectively n bit. Generally in all famous DBL scheme's construction n bit message has been used. But in our scheme the main innovation is using of variable message tn $(t < 1)$. That's why it is necessary to accommodate the feature of variable message in query response mechanism. Firstly it is defined, what's the problem, if traditional query response has been used in our new scheme. As for example if the collision has been occurred under the n bit then it is not problem for finding the collision under tn bit. But problem is when tn bit message hash been used in our scheme then there is probability to find out collision under the based on tn bit. Adversary $\mathcal{A}$ can't get the success under this constraints. So it is needed to implement another adversary algorithm $\mathcal{B}$ which will work based on the query response of $\mathcal{A}$.

The main trick is adversary $\mathcal{A}$ calls adversary $\mathcal{B}$ and provides access in his oracle. The scope of adversary $\mathcal{B}$ is limited and more powerful. Adversary $\mathcal{B}$ can make query on the basis on tn bit instead of n bit. So the query oracle reduces 2^n to 2^{tn}. That's why in the perspective of adversary it is more powerful due to size reduces. The collision probability naturally increased than previous scheme. Second important factor is how actually adversary $\mathcal{B}$ works or how domain has been decreased from 2^n to 2^{tn}. Adversary $\mathcal{A}$ runs the adversary $\mathcal{B}$ which has access right in the oracle of adversary $\mathcal{A}$. Each iteration $\mathcal{A}$ makes a query and adversary $\mathcal{B}$ takes the result n bit. From here it trims tn bit but this truncation is based on string-compare algorithm. That means adversary $\mathcal{B}$ tries to prune those part of tn-bit which collides with the previous result. If $\mathcal{B}$ doesn't find match then arbitrary select tn bit from n bit. So when collide being occurred $\mathcal{B}$ sends false to adversary $\mathcal{A}$ and then adversary $\mathcal{A}$ stops the query process. If not then sends true to adversary $\mathcal{A}$ and process status being will be alive. In this way actually the adversary $\mathcal{B}$ tries to find collision for the size of tn bit instead of n bit. So that the power of adversary has been increased and probability of collision finding for any event hash been increased. So it is clear that the adversary gets more advantage and the result would be more tight. Now, $\mathcal{B}$ checks in $\mathcal{Q}$. Let, (k_j, x_j, y_j^1) and (k'_j, x_j, y_j^2) be the triplets of E obtained by the j^{th} pair of queries and corresponding answers.

Case 1

For every j [where $j \leq q$], let C_j be the event that a colliding pair found for F with the j^{th} pair of queries. The event is as like $j' < j$:

$$y_j^1 \oplus x_j = y_{j'}^1 \oplus x_{j'} \quad \text{and} \quad y_j^2 \oplus x_j = y_{j'}^2 \oplus x_{j'}$$

$$y_j^1 \oplus x_j = y_{j'}^2 \oplus x_{j'} \quad \text{and} \quad y_j^2 \oplus x_j = y_{j'}^1 \oplus x_{j'}$$

It implies that,

$$\Pr[C_j] \leq \frac{2(j-1)}{(2^{tn}-(2j-1)^2} \leq \frac{2(j-1)}{2^{2tn}}$$

Let C be the event that a pair are found for F with q pairs of queries, then, $\Pr[C] = \Pr[C_2 \vee C_2 \vee ..... \vee C_q]$

$$\begin{aligned}&\leq \sum_{j=2}^{q} \Pr[C_j] = \sum_{j=2}^{q} \tfrac{2(j-1)}{2^{2tn}} = \tfrac{2}{2^{2tn}} \sum_{j=2}^{q} (j-1) \\ &= \tfrac{2}{2^{2tn}} \times \left\{ \tfrac{(q-1)(q+2)}{2} - (q+1-2) \right\} \\ &= \tfrac{q(q-1)}{2^{2tn}}\end{aligned}$$

Case 2
For $1 \leq j \leq q$, it is the event:

$$(x_j, k_j) \in (x_j \oplus y_j^1) \;\; \text{and} \;\; (x_j, k_j') \in (x_j \oplus y_j^1)$$

It implies that, $(x_j, k_j) = (x_j, k_j')$, where, the probability is: $\Pr[C] = \frac{1}{2^{tn}}$. Let C be the event that a pair are found for F with q pairs of queries. Then, $\Pr[C]$ = $\Pr[C_1 \vee C_2 \vee .....C_q]$

$$\begin{aligned}&= \sum_{j=1}^{q} \Pr[C_j] \\ &= \sum_{j=1}^{q} \tfrac{1}{2^{tn}} = \tfrac{q}{2^{tn}}\end{aligned}$$

Case 3
For $1 \leq j \leq q$, it is the event:

$$\left[\left\{(x_j, k_j) \in y_j^1\right\}, \left\{\left(x_j, k_j'\right) \in y_j^2\right\}\right] = (h_0, g_0) \;\; \text{or} \;\; (g_0, h_0)$$

where, the probability is $2/2^{2tn}$. Let C be the event that a pair are found for F with q pairs of queries. Then, $\Pr[C]$ = $\Pr[C_1 \vee C_2 \vee .....C_q]$

$$\begin{aligned}&\leq \sum_{j=1}^{q} \tfrac{2}{2^{tn}} = 2 \times \sum_{j=1}^{q} \tfrac{1}{2^{tn}} \\ &= \tfrac{2q}{2^{tn}}\end{aligned}$$

Take the result from Case 1, Case 2 and Case 3. Then finally it is shown that,

$$\frac{q(q-1)}{2^{2tn}} + \frac{2q}{2^{2tn}} + \frac{q}{2^{tn}} = \frac{q(q+1)}{2^{2tn}} + \frac{q}{2^{tn}}$$

■

B. preimage security analysis

Let $\mathcal{A}$ be an adversary that tries to find a preimage for its input σ. We follow a similar proof strategy of Armknecht and implementation strategy of Armknecht [16], when $\mathcal{A}$ selects its queries. Specifically, when $\mathcal{A}$ makes an E and E^{-1} query. Then the adversary $\mathcal{A}$ asks the conjugate queries in pairs. Now it is needed to bound the probability that i^{th} query pair leads to a preimage for σ. The definition of σ is defined as which is selected by adversary arbitrary before making any queries to queries to E or E^{-1} and it will be $(h'||g') \ni \sigma$. So findings is that to calculate the probability that in q queries the adversary finds a point (σ), such that $H^{MSR}(h, g, m^1, m^2) = \{(h'||g')\}$.

Theorem 2: Let H^{MSR} be a double-length compression function ($E \in$ block (n, n)). Then the advantage of an adversary in finding a preimage in H^{MSR} after q queries can be upper bounded by :

$$\mathrm{Adv}_h^{epre}(q) \leq \; 16q/2^{2tn}$$

Proof: According to definition of adjacent query pair [16], the adversary $\mathcal{B}$ maintains a adversary query database Q in the form of $(h, g||m^1, y^1), (h, \bar{g}||m^2, y^2)$ which has been run by adversary $\mathcal{A}$. This is called adjacent query pair. Now need to make and implement super query. It implies that the query history contains exactly $N/2$ queries with the same key, all remaining queries under this key are given for free to the adversary. Now, an adjacent query pair $(h, g||m^1, y^1), (h, \bar{g}||m^2, y^2)$ can be succeed iff,

$$h_{i-1} \oplus y_i^1 = h' \;\; \text{and} \;\; h_{i-1} \oplus y_i^2 = g'.$$

$$h_{i-1} \oplus y_i^1 = g' \;\; \text{and} \;\; h_{i-1} \oplus y_i^2 = h'.$$

Thus the adversary obtains a preimage of $\left\{(h'||g') \in \{0,1\}^{2n}\right\} \ni \sigma$, in particularly if it attains a winning adjacent query pair. It can be occurred by any of the following way such as:

- Winning pair can be the member of *Normal query* database, which is denoted by $\mathsf{NormalQueryWin}(Q)$
- Winning pair can be the member of *super query* database, which is denoted by $\mathsf{SuperQueryWin}(Q)$

From the above, we get the definition of $\mathsf{NormalQueryWin}(Q)$ and $\mathsf{SuperQueryWin}(Q)$. For the proving of Theorem 2., one's need to find out the probability of $\mathsf{NormalQuery\ Win}(Q)$ and $\mathsf{SuperQueryWin}(Q)$ where, the adversary's obtaining probability of a winning adjacent query pair respectively comes from normal query or super query database. So now sum up these two probability result for finding the preimage resistance of the new scheme.

$$\Pr[\mathsf{NormalQueryWin}(Q)] + \Pr[\mathsf{SuperQueryWin}(Q)]$$

Case 1: Probability of $\mathsf{NormalQueryWin}(Q)$
Adversary $\mathcal{B}$ which has been called by adversary $\mathcal{A}$, can make forward or backward query such as $E_{h||m^1}(g)$ or ${E^{-1}}_{h||m^2}(\bar{g})$. Under this section, the goal is to find out the $\mathsf{NormalQueryWin}(Q)$. So we need to take the definition of super query and adjacent query. Then find the set size from where the fresh value of y^1 or y^2 could be found. In that case, two cases can be happened, where following two cases are dependent on each other.

- Sub-Case 1.1 The adversary $\mathcal{B}$ can make forward or backward query. Assume adversary makes a forward $E_{g||m^1}(h)$ query, where at most $(2^{tn}/2 - 1)$ queries could be answered previously and for $E_{\bar{g}||m^2}(h)$ query,

earlier it could be answered at most $(2^{tn}/2 - 1)$ queries. If not then super query can be occurred. So the value of y^1 and y^2 comes uniformly and independently from the set size $2^{tn}/2$. So probability forms as $\binom{2}{2^{tn}/2}$.

- Sub-Case 1.2 If $h \oplus y^1 = h'$ then there is a probability for the free query $E_{\bar{g}||m^2}(h)$ (part of adjacent query pair) to return $h \oplus g'$ from the set size $(2^{tn}/2 + 1)$. So probability could be $\binom{1}{2^{tn}/2} = \frac{2}{2^{tn}}$.

So desired probability of NormalQueryWin(Q) is $^8/_{2^{2n}}$.

Case 2: Probability of SuperQueryWin*(Q)*

In this section the target is to find out the probability of Super query so that again it is needed to recall the definition and technique of super query and adjacent query pair. As for example under the key $g||m^1$ and $\bar{g}||m^2$ the value of $E_{g||m^1}(\cdot)$ and $E_{\bar{g}||m^2}(\cdot)$ already have been known on exactly $2^{tn}/2$ points. So from the definition of super query and adjacent query pair if $E_{g||m^1}(\cdot)$ is the part of super query then the corresponding $E_{\bar{g}||m^2}(\cdot)$ query must be the member of the super query domain, hence this will be considered as a paired query.

From the above discussed points, it can be said that, probability of $E_{g||m^1}(h) = h'$ is either 0 or $\frac{2}{2^{tn}}$. Now the question how it can be found. The probability will be 0 if the h' is not in the range of super query that means it is available in the domain of normal query. Just oppositely it is assumed that due to super query the result comes from the set size $2^{tn}/2$, so probability is $\frac{2}{2^{tn}}$. For the adjacent query pair following cases can be happened:

$$y^1 \oplus h \in h' \text{ and } y^2 \oplus g \in h' \text{ or}$$
$$y^1 \oplus h \in g' \text{ and } y^2 \oplus g \in g'$$

- Sub-Case 2.1 For the query of $E_{g||m^1}(h) \oplus h = h'$, this answer will come from the set size $2^{tn}/2$. So the probability would be $\frac{2}{2^{tn}}$. As well as the probability that $E_{\bar{g}||m^2}(g) \oplus g = h'$ is equal to $\frac{2}{2^{tn}}$. So the total probability of case-1 looks like, $\left(\frac{2}{2^{tn}}\right)^2$
- Sub-Case 2.2 For the same explanation, as like Case 1, the total probability of $E_{g||m^1}(h) \oplus h = g'$ and $E_{\bar{g}||m^2}(g) \oplus g = g'$ is $\left(\frac{2}{2^{tn}}\right)^2$.

Now, analysis the probability of case-1 and case-2 and point that the cost of super query occurs is $2^{tn}/2$. Another important factor is that the probability of super query occurs, which is at most $^q/_{(2^{tn}/2)}$. It implies that, $\Pr[\text{SuperQueryWin}(Q)]$:

$$\leq {}^q/_{(2^{tn}/2)} \times \left(2^{tn}/2\right) \times 2 \times \left(\frac{2}{2^{tn}}\right)^2 = \frac{8q}{2^{2tn}}$$

Taking the value of $\Pr[\text{NormalQueryWin}(Q)]$ and $\Pr[\text{SuperQueryWin}(Q)]$ and then add, which implies that,

$$\Pr[NormalQueryWin(Q)] + \Pr[SuperQueryWin(Q)] \leq 16q/2^{2tn}$$

■

C. efficiency rate

In the Related Work section, efficiency rate is defined. From that view point of definition, here for the new MSR scheme, efficiency rate has been demonstrated as:

$$r = \frac{|2tn|}{2 \times n} = t$$

From the Def. 5, it is known that the total message size $\{(m^1, m^2) \in tn\} \Rightarrow \{2tn\}$ and number of block cipher is 2, where block length is $n : n = 128$ bit. In this construction the message size option is variable $(t < 1)$. It implies the following table:

Table III
DIFFERENT EFFICIENCY RATE BASED ON VARIABLE MESSAGE

	value of t	Efficiency rate: r
$n = 128$	$1/2$	0.5
$n = 128$	$5/8$	0.625
$n = 128$	$3/4$	0.75

VI. RESULT ANALYSIS

In this section, we just mentioned our getting result and compare with previous famous schemes based on AES-128 such as MDC-2, MDC-4, MJH. In table IV we mentioned our proposed MSR scheme's result based on variable message size which is tuning by the terms t $(t < 1)$. In the following table CR stands for collision resistance, PR means preimage resistance and r indicates efficiency rate. If we carefully observe the following table, it can be easily said that our proposed scheme's result is better than previous famous schemes except for the value of $t = 1/2$.

Table IV
RESULT ANALYSIS: PROPOSED MSR SCHEME

	CR	PR	r
MDC-2	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(2^n)$	0.5
MDC-4	$\mathcal{O}(2^{5n/8})$	$\mathcal{O}(2^{5n/4})$	0.25
MJH	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(2^n)$	0.5
MSR (Proposed)	CR	PR	r
$t = 1/2$	$\mathcal{O}(2^{n/2})$	$\mathcal{O}(2^n)$	0.5
$t = 5/8$	$\mathcal{O}(2^{5n/8})$	$\mathcal{O}(2^{5n/4})$	0.625
$t = 3/4$	$\mathcal{O}(2^{3n/4})$	$\mathcal{O}(2^{3n/2})$	0.75

VII. Conclusion

In this article, a new double block length compression hash function has been introduced which is based on (n, n) bit blockcipher. The main key point of this scheme is to handle short variable size of message. Due to variable size of message, security also varies. Another key point is the security of this scheme is better bound than other famous (n, n) bit blockcipher which can be introduced from the final result in table I. It's true that, the security of $(n, 2n)$-bit blockcipher based hash function is better than our proposed scheme. But it should be noted that the (n, n)-bit blockcipher is 40% faster than $(n, 2n)$-bit blockcipher. Two other facts such as power consumption and memory utilization is better for AES-128 $[(n, n)]$ which have been already mentioned earlier. So there is open space to do work for increasing the security bound. All security proofs are based on the ideal cipher mode (ICM) but in real life AES is not ICM. So it is possible to make security proof under the weak cipher model. For the MSR scheme, it can be said that it's key schedule is twice, so there is a opportunity to make a new scheme which obtains single KS and as well as better security bound.

References

[1] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin , "Hash Functions and RFID Tags: Mind the Gap," *LNCS, CHES,* vol. 5154, pp. 283-299, 2008.

[2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography,* 5th ed, CRC Press, 2001.

[3] J. P. Kaps, B. Sunar, "Energy Comparison of AES and SHA-1 for Ubiquitous Computing," *LNCS, Emerging Directions in Embedded and Ubiquitous Computing*, vol. 4097, pp. 372-381, 2006.

[4] X. Wang, X. Lai, D. Feng, H. Chen, X. Yu, "Cryptanalysis of the Hash Functions MD4 and RIPEMD," *LNCS, EUROCRYPT*, vol. 3494, pp. 1-18, 2005.

[5] X. Wang, X. Lai, X. Yu, "Finding Collisions in the Full SHA-1.," *LNCS, CRYPTO*, vol. 3621, pp. 17-36, 2005.

[6] E. Fleischmann, C. Forler, S. Lucks, J. Wenzel, "Weimar-DM: A Highly Secure Double-Length Compression Function," *LNCS, ACISP*, vol. 7372, pp. 152-165, 2012.

[7] B. Mennink, "Optimal Collision Security in Double Block Length Hashing with Single Length Key," *LNCS, ASIACRYPT*, vol. 7658, pp. 526-543, 2012.

[8] L. Knudsen, B. Preneel, "Fast and Secure Hashing Based on Codes," *LNCS, CRYPTO*, vol. 1294, pp. 485-498, 1997.

[9] J. Lee, K. Kapitanova, S. H. Son "The price of security in wireless sensor networks," *ELSEVIER, Computer Network*, vol. 54, no. 17, pp. 2967-2978, December 2010.

[10] O. Ozen, M. Stam, "Another Glance at Double-Length Hashing," *LNCS, Cryptography and Coding*, vol. 5291, pp. 176-201, 2009.

[11] J. Lee, M. Stam, "MJH: A Faster Alternative to MDC-2," *LNCS, CT-RSA*, vol. 6558, pp. 213-236, 2011.

[12] X. Lai, X. Massey, L. J.,"Hash function based on block ciphers," *LNCS, EUROCRYPT*, vol. 658, pp. 55-70, 1993.

[13] J. Lee, D. Kwon, "The Security of Abreast-DM in the Ideal Cipher Model," *IEICE Transactions*, vol. 94-A(1), pp. 104-109, 2011.

[14] J. Lee, M. Stam, J. Steinberger, "The Collision Security of Tandem-DM in the Ideal Cipher Model," *LNCS, CRYPTO*, vol. 6841, pp. 561-577, 2011.

[15] S. Hirose, "Some Plausible Constructions of Double-Block-Length Hash Functions," *LNCS, FSE*, vol. 4047, pp. 210-225, 2006.

[16] F. Armknecht, E. Fleischmann, M. Krause, J. Lee, M. Stam, J. Steinberger, "The Preimage Security of Double-Block-Length Compression Functions," *LNCS, ASIACRYPT*, vol. 7073, pp. 233-251, 2011.

[17] E. Fleischmann, C. Forler, M. Gorski, S. Lucks, "Collision Resistant Double-Length Hashing," *LNCS, PROVSEC*, vol. 6402, pp. 102-118, 2010.

[18] B. Mennink, "Optimal Collision Security in Double Block Length Hashing with Single Length Key," *LNCS, ASIACRYPT*, vol. 7658, pp. 526-543, 2012.

[19] L. Knudsen, B. Preneel, "Fast and secure hashing based on codes," *LNCS, CRYPTO*, vol. 1294, pp. 485-498, 1997.

[20] J. A. Black, P. Rogaway, T. Shrimpton, "Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV," *LNCS, CRYPTO*, vol. 2442, pp. 320-335, 2002.

[21] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell Systems Technical Journal*, vol. 128-4, pp. 656-715, 1949.

[22] J. A. Black, P. Rogaway, T. Shrimpton, M. Stam, "An Analysis of the Blockcipher-Based Hash Functions from PGV," *LNCS, J.CRYPTOL*, vol. 23, pp. 519-545, 2010.