A Generic Method to Extend Message Space of a Strong Pseudorandom Permutation

Método Genérico para Extender el Espacio del Mensaje de una Permutación Pseudo-aleatoria Fuerte

Mridul Nandi

Indian Statistical Institute
mridul.nandi@gmail.com

Article received on March 1, 2008, accepted on October 30, 2008

Abstract
Let $\mathbf{E}$ be a strong pseudorandom permutation (or SPRP) secure enciphering scheme (i.e., a length-preserving encryption scheme) which can only encrypt messages of size multiple of n, the block size of the underlying block cipher. There are several such constructions, e.g., CBC mode or cipher block chaining mode. In this paper we present how a secure enciphering scheme $\overline{\mathbf{E}}$ can be obtained which can encrypt any messages of size at least n based on $\mathbf{E}$ and some other cryptographic objects such as weak pseudorandom function (or WPRF) and a universal hash function. So $\overline{\mathbf{E}}$ can encrypt messages which might contain incomplete message blocks. Since an enciphering scheme is a length preserving encryption algorithm, one can not use a padding rule to handle the incomplete message block. In 2007, Ristenpart and Rogaway first proposed a secure method known as XLS (eXtension by Latin Squares). It needs two invocations of a block cipher e whose key is chosen independently of the key of $\mathbf{E}$. The SPRP security of XLS is based on the SPRP security of the block cipher e. Our proposed enciphering scheme is SPRP and it needs only one invocation of a WPRF and two invocations of a universal hash function. Any SPRP construction, e.g., a secure block cipher, is a WPRF. Moreover, there are other several efficient constructions for universal hash functions and WPRF which are not SPRP. Thus, we are able to replace SPRP security by two weaker security notions to extend the domain of a secure enciphering scheme.

Keywords: strong pseudorandom permutation, weak pseudorandom function, universal hash function, modes of operations.

Resumen
Sea $\mathbf{E}$ un esquema seguro de cifrado que preserva la longitud del texto en claro y que se comporta como una permutación pseudo-aleatoria fuerte (SPRP por sus siglas en inglés), el cual únicamente puede cifrar mensajes con longitudes que sean múltiplos de n, donde n es el tamaño del bloque utilizado por el esquema de cifrado. Existen varios ejemplos de construcciones de este tipo, por ejemplo, el modo de cifrado por bloque encadenado (CBC por sus siglas en inglés). En este artículo describimos cómo construir un esquema de cifrado seguro $\overline{\mathbf{E}}$, capaz de cifrar cualquier mensaje de tamaño mayor o igual que n. Mostramos que $\overline{\mathbf{E}}$ puede ser construido con $\mathbf{E}$ y algunos otros objetos criptográficos tales como una función pseudo-aleatoria débil (WPRF por sus siglas en inglés) y una función picadillo universal. El esquema $\overline{\mathbf{E}}$ así obtenido puede cifrar mensajes con longitudes que no son múltiplos de n. Un esquema de cifrado que preserva la longitud del texto en claro no puede rellenar el último bloque de mensaje cuando éste está incompleto. En 2007, Ristenpart y Rogaway fuernos los primeros en proponer un método seguro conocido como extensión de cuadrados latinos (XLS por sus siglas en inglés). XLS utiliza dos invocaciones al cifrador por bloques e, cuya llave es escogida independientemente de la llave de $\mathbf{E}$. La seguridad SPRP de XLS se basa en la seguridad SPRP del cifrador por bloques e. El esquema de cifrado propuesto aquí es SPRP y necesita únicamente una invocación de una WPRF y dos invocaciones a una función picadillo universal. Cualquier construcción SPRP, esto es, un cifrador por bloques seguro, es un WPRF. Por otro lado, existen construcciones eficientes para funciones picadillo universales y para WPRF que no son SPRP. Estas dos últimas observaciones implican que

en este artículo logramos obtener seguridad del tipo SPRP al utilizar dos nociones de seguridad más débiles, al tiempo que extendemos el dominio original del esquema de cifrado seguro.

Palabras Claves: Permutación pseudo-aleatoria fuerte, función pseudo-aleatoria débil, función picadillo universal, modos de operación.

1 Introduction

The notion of domain extension arises in many areas of cryptography e.g., collision resistant hash function, *pseudorandom function* or prf, *strong pseudorandom permutation* or SPRP (Luby and Rackoff 1988) etc. Intuitively, a domain extension extends the message space or domain of a cryptographic primitive. For example, the block cipher AES (Daemen and Rijmen 2002) or Advanced Encryption Standard is a keyed permutation family defined over the set of all 128 bits. AES can be used to encrypt of 128 bit messages only. Given any message of size multiple of 128, one may use CBC or cipher block chaining mode (Bellare, Kilian, and Rogaway 1994) based on AES to encrypt the message. A similar kind of treatment can be found in the hash function where a hash function is designed from a compression function. To encrypt a message whose size is not multiple of 128, one can use a padding rule to make the message size multiple of 128. This methods trivially can not preserve length, in particular the size of ciphertext is more than that of plaintext. In some applications like disk encryption, length-preserving encryption is desirable. We call a length-preserving encryption an *enciphering scheme*. The length-preserving property makes our task more difficult and restricted too. In this paper we mainly study how one can obtain a length-preserving encryption scheme or an enciphering scheme which can encrypt any messages of size at least n where n is the block size of the underlying block cipher (e.g., $n = 128$ in case of the AES). There are some known standard tricks like ciphertext stealing (Meyer and Matyas. 1982), applying the underlying block cipher twice to the last full blocks (applied to EME (Halevi and Rogaway 2004; Halevi 2004), TET (Halevi), HEH (Sarkar. 2007)), using counter-based prf (applied to HCTR (Wang, Feng, and Wu 2005), HCH (Chakraborty and Sarkar 2006), XCB (McGrew and Fluhrer 2004)) etc. But those approaches are not generic. There was a heuristic domain extension (D. Cook and Keromytis. 2004a; D. Cook and Keromytis. 2004b) without any security proof. The first and so far only one concrete provable secure generic domain extension is XLS (Ristenpart and Rogaway. 2007) (or eXtension by Latin Squares).

1.1 Discussion on SPRP, WPRF and universal hash function

The most popular and strong security notion for an encryption or enciphering scheme is strong pseudorandom permutation security or SPRP-security. Intuitively, an SPRP block cipher (or an enciphering scheme) should be indistinguishable from an ideal random permutation with respect to chosen ciphertext attack. In other words, any distinguisher who can make encryption or decryption queries adaptively (i.e., the queries may depend on the previous query-responses) should not be able to distinguish block cipher or enciphering scheme from an ideal random permutation. A pseudorandom function or prf is a similar security notion for a keyed function family (instead of permutation family). It is hard to distinguish a prf function family from an ideal random function family with respect to chosen plaintext attack. Note that, here distinguishers can only make forward queries. Weak pseudorandom function or WPRF is obtained by weakening the prf distinguisher. In this case, the distinguisher is not allowed to choose the plaintext. All plaintexts will be chosen at random and its corresponding ciphertexts for a keyed function family will be given to the distinguisher. If it is hard to distinguish from a randomly generated ciphertexts (or outputs of an ideal random function family) then the keyed function family is called WPRF. These randomly generate plaintext can also be generated by the distinguisher as long as it is generated independently and uniformly.

A universal hash function is a function family where the collision probability for any two chosen plaintexts is negligible. There exist several provably secure universal hash function (e.g., finite field multiplication based universal hash function). In case of SPRP or prf or WPRF, we believe (without any proof) some constructions to be secure with respect to these security notions The AES (an NIST or National Institutes of Standards and Technology standard for

a block cipher) is a possible candidate of an SPRP defined over 128 bits. It is very efficient in both hardware and software. There are very few papers on the practical constructions of a WPRF. It is easy to see that any SPRP or prf construction is also a WPRF but the converse need not be true (in fact, a WPRF does not need to be a permutation). So AES itself is also a possible candidate for WPRF. One can use keyed hash function as a WPRF since it is believed to be a prf. There are some other possible candidates of WPRF which are very efficient (Blum, Furst, Kearns, and Lipton 1993; Naor and Reingold 1999).

There are several examples of universal hash function. A universal hash function based on field multiplication is very fast in hardware since a field multiplication in $\mathbf{F}_{2^n}$ takes only one cycle by using Karatsuba-Ofman (Karatsuba and Ofman.) algorithm. In software, there are several efficient examples of universal hash function (Nevelsteen and Preneel 1999). One can also use prime field multiplication as described in (Bernstein. 2005) to make it more faster.

1.2 A comparison study of XLS and the new domain extension DE

Let $\mathbf{E}$ be an SPRP secure encryption scheme for the message space $(\{0,1\}^n)^+ = \cup_{i=1}^{\infty}\{0,1\}^{ni}$. The XLS construction needs two invocations of a block cipher, say AES, whose key is chosen independently from the key of $\mathbf{E}$. The enciphering scheme $\mathbf{E}$ may use the same block cipher but the key should be chosen independently. Thus, in hardware it is not easy to have a pipe line implementation. Moreover as it needs two different keys two key scheduling algorithms have to be performed separately.

In this paper we provide a generic alternative construction of an enciphering scheme $\overline{\mathbf{E}}$ with domain $\{0,1\}^{\geq n}$ based on a secure $\mathbf{E}$ encrypting messages from $(\{0,1\}^n)^+$, a WPRF f and a universal hash function $\mathtt{H}$. Our new construction is mainly motivated by the counter-based modes of operation. In a counter-based construction one first computes counter (something like a tag) by using a polynomial hash (an example of a universal hash function) and then the counter is used to generate a random bit sequence. In our domain extension, we use similar structure. We need one WPRF f and a universal hash function $\mathtt{H}$ to encrypt the incomplete message block. We denote it by $\overline{\mathbf{E}} := \mathrm{DE}[\mathbf{E}, f, \mathtt{H}]$. In Section 3 we prove that $\overline{\mathbf{E}}$ is SPRP (or tweakable SPRP) whenever $\mathbf{E}$ is SPRP (or tweakable SPRP respectively), f is a WPRF and $\mathtt{H}$ is a universal hash function. In a nutshell, we are able to replace two invocations of SPRP by one invocation of a WPRF and two invocations of a universal hash function to encrypt an incomplete message block securely.

WPRF is much weaker security notion than prf or SPRP (Maurer and Sjdin 2007). Potentially one can have efficient implementation of WPRF. In the worst case one can use an SPRP secure block cipher as a WPRF since any SPRP is WPRF. So even if we use the AES, we can have faster implementation than XLS if we have an implementation of a universal hash function which is twice as efficient as the AES. Moreover, an SPRP-weakness of AES would not immediately threaten our construction. In Table 1, we have a comparison study.

Table 1. A comparison table of XLS and our domain extension DE. Here k_{BC} is the key size of the underlying block cipher key, k_{WPRF} is the key size of a WPRF and k_{hash} is the key size of a universal hash function

	XLS	DE
Key size	k_{BC}	$k_{\mathrm{WPRF}} + k_{\mathrm{hash}}$
Universal Hash	0	2
SPRP	2	0
WPRF	0	1

Organization of the paper. We first provide some definitions and notations about the security notion in section 2. Then in Section 3, we describe our new domain extension and discuss some important issues. We also provide complete security analysis of the new construction in the same section. Finally we conclude in section 4.

2 Preliminaries

We denote $x[s]$ to represent the first s bits of $x \in \{0,1\}^n$ where $s \leq n$. We write $|x| = i$ whenever $x \in \{0,1\}^i$. Given any x, $0 \leq |x| < n$, we define $\overline{x} = x10^i$ where $i = n - 1 - |x|$. Any n bit element is called a block and $\mathbf{0} = 0^n$ is called the zero block. A bit string x is said to be an incomplete block if $|x| < n$. It is easy to see that, for an incomplete block x, $\overline{x}$ is a nonzero block and whenever $x \neq x'$, we must have $\overline{x} \neq \overline{x}'$.

We identify $\{0,1\}^n$ as $\mathbf{F}_{2^n}$ with the field addition $\oplus$ (bitwise addition) and a field multiplication '$*$'. In this paper, we fix an irreducible polynomial and hence we have a fixed multiplication operation on $\{0,1\}^n$.

In cryptography, usually a message space can be $\{0,1\}^*$, $\{0,1\}^{\geq n} := \cup_{i \geq n}\{0,1\}^i$ or $\{0,1\}^{n^+} := \cup_{i \geq 1}\{0,1\}^{ni}$. Note that, all these sets can be written as $\cup_{i \in L}\{0,1\}^i$ for some set (known as length set) $L \subseteq \mathbf{N} := \{0,1,2,\cdots\}$.

Definition 2.1 *A set $\mathcal{M} \subseteq \{0,1\}^*$ is said to be* complete *if there exists a set $L \subseteq \mathbf{N}$ such that $\mathcal{M} = \cup_{i \in L}\{0,1\}^i$. In this case, we also denote $\mathcal{M} = \mathcal{M}_L$. The set L is called the length-set for $\mathcal{M}$.*

Let $\mathcal{M}_L$ be a complete set. A function (permutation) $F : \mathcal{M}_L \to \mathcal{M}_L$ is called length-preserving *(or l.p.) if $|F(x)| = |x|$ for all $x \in \mathcal{M}_L$ (equivalently, $F_i := F|_{\{0,1\}^i}$, the function F restricted on $\{0,1\}^i$, is a function (permutation) from $\{0,1\}^i$ to $\{0,1\}^i$ for all $i \in L$).*

In this paper, we mainly consider the length-sets $L = \{n\}$, or $[n,\infty] = \{n, n+1, \cdots\}$, or $n^+ := \{n, 2n, 3n, \cdots,\}$. We denote the corresponding complete sets as $\mathcal{M}_n = \{0,1\}^n$, $\mathcal{M}_{\geq n} = \cup_{i \geq n}\{0,1\}^i$, $\mathcal{M}_{n^+} = \cup_{i \geq 1}\{0,1\}^{ni}$ respectively. Given a l.p. function (permutation) F defined over a complete set $\mathcal{M}_L$, we can equivalently characterize F by a sequence of functions $\langle F_i \rangle_{i \in L}$, where F_i is the restricted function (permutation) on $\{0,1\}^i$ (as mentioned in the above definition). If F is a l.p. permutation then the inverse l.p. permutation F^{-1} can be similarly characterized by the sequence $\langle F_i^{-1} \rangle_{i \in L}$.

Definition 2.2 *A random function from A to B is a random variable f taking values on $\mathrm{Func}(A,B)$, the set of all functions from A to B where A and B are finite sets. We say a random function is a random permutation on A if it has support on $\mathrm{Perm}(A)$, the set of all permutations on A. In other words, a random permutation takes values from the set of all permutations on A with probability one. A* **length preserving random function** *over a length set L is a sequence of random functions $F = \langle F_i \rangle_{i \in L}$ where F_i is a random function from $\{0,1\}^i$ to $\{0,1\}^i$. We say that F is a length preserving random permutation if F_i is a random permutation on $\{0,1\}^i$ for all $i \in L$.*

In cryptography, one can find several examples of random functions and random permutations. Let e_K be a block cipher over the domain $\{0,1\}^n$ and key space $\{0,1\}^k$. If the key K is chosen uniformly from $\{0,1\}^k$ then the block cipher is a random permutation (not necessarily the ideal random permutation or uniform random random permutation which is going to be defined next). Similarly an enciphering scheme is nothing but a length preserving random permutation. We use the word "uniform" to represent the ideal candidates of random functions. In fact, all cryptographic ideal candidates of random functions have uniform distributions on a certain space of functions. Now we define the following ideal random functions which will be considered later defining the cryptographic security notions.

1. Let $\mathtt{R}_i$ denote the *uniform random function* from $\{0,1\}^i$ to $\{0,1\}^i$, i.e., $\mathtt{R}_i$ has uniform probability distribution on $\mathrm{Func}(\{0,1\}^i, \{0,1\}^i)$. Given a length-set L, we denote $\mathtt{R}_L$ for the tuple $\langle \mathtt{R}_i \rangle_{i \in L}$ of random functions where $\mathtt{R}_i$'s are independently distributed (more precisely, for any finite collections of i, R_i's are independently distributed). We call it a *length-preserving uniform random function* on $\mathcal{M}_L$. Note that it is not a random function according to our original definition of random function. It is rather a sequence of independent random functions. In this paper we are interested in length-preserving uniform random functions $\mathtt{R}_{\geq n}$ and $\mathtt{R}_{n^+}$ over domains $\{0,1\}^{\geq n}$ and $(\{0,1\}^n)^+$ respectively.

2. Let $\mathtt{P}_i$ denote the uniform random permutation on $\{0,1\}^i$, i.e., the uniform distribution on $\mathrm{Perm}(\{0,1\}^i)$. Note that the inverse random permutation, $\mathtt{P}_i^{-1}$, is also a uniform random permutation. We similarly define $\mathtt{P}_L$ on

$\mathcal{M}_L$ and its inverse $\mathtt{P}_L^{-1} = \langle \mathtt{P}_i^{-1} \rangle_{i \in L}$, called length-preserving uniform random permutation on $\mathcal{M}_L$. Similar to uniform random function, in this paper we also consider uniform length-preserving random permutations $\mathtt{P}_{\geq n}$ or $\mathtt{P}_{n^+}$ over domains $\{0,1\}^{\geq n}$ and $(\{0,1\}^n)^+$ respectively.

2.1 SPRP Security Notion

Let $\mathcal{A}$ be an oracle algorithm which has access to two oracles $\mathcal{O}_1$ (first oracle) and $\mathcal{O}_2$ (second oracle). For example, let F_L be a length-preserving random permutation then we write $\mathcal{A}^{\mathtt{F}_L, \mathtt{F}_L^{-1}}$ to denote that the first oracle of $\mathcal{A}$ is $\mathtt{F}_L$ and $\mathtt{F}_L^{-1}$ is the second oracle. The algorithm $\mathcal{A}$ makes queries from the set $\mathcal{M}_L$ for both oracles. We define SPRP-advantage of $\mathcal{A}$ for a length-preserving random permutation $\mathtt{F}_L$ by

$$\mathbf{Adv}_{\mathtt{F}_L}^{\mathrm{SPRP}}(\mathcal{A}) = \big|\Pr[\mathcal{A}^{\mathtt{F}_L, \mathtt{F}_L^{-1}} = 1] - \Pr[\mathcal{A}^{\mathtt{P}_L, \mathtt{P}_L^{-1}} = 1]\big|.$$

Here oracles are considered as a sequence of random functions. When we consider the algorithm $\mathcal{A}^{F_L, F_L^{-1}}$, for each $\mathcal{O}_1$-query (or $\mathtt{F}_L$-query) $x \in \{0,1\}^\ell$, $\ell \in L$, $\mathtt{F}_L$ responses $\mathtt{F}_\ell(x)$. Similarly, for the inverse query the oracle responses $\mathtt{F}_L^{-1}(x)$. Here is the behavior of an oracle algorithm $\mathcal{A}^{\mathtt{F}_L, \mathtt{F}_L^{-1}}$.

1. $\mathcal{A}$ makes i^{th} query $x_i \in \{0,1\}^{\ell_i}$, which is a function of $(x_1, y_1, \cdots, x_{i-1}, y_{i-1})$, to either $\mathtt{F}_L$ or $\mathtt{F}_L^{-1}$. Here ℓ_i denotes the size of the i^{th} query. If it makes $\mathtt{F}_L$-query then the response follows the probability distribution $y_i = \mathtt{F}_{\ell_i}(x_i)$, otherwise it follows $\mathtt{F}_{\ell_i}^{-1}(x_i)$.

2. After making q queries, $\mathcal{A}$ returns 0 or 1 based on all query-responses $((x_1, y_1, \delta_1), \cdots, (x_q, y_q, \delta_q))$ where δ_i is either 1 or -1 depending on whether i^{th} query is $\mathtt{F}_L$ or $\mathtt{F}_L^{-1}$-query.

In general, we can define advantage for two pairs of tuples of length-preserving random functions $(\mathtt{F}_L, \mathtt{F}'_L)$ and $(\mathtt{G}_L, \mathtt{G}'_L)$ as

$$\mathbf{Adv}_{\mathcal{A}}((\mathtt{F}_L, \mathtt{F}'_L), (\mathtt{G}_L, \mathtt{G}'_L)) = \big|\Pr[\mathcal{A}^{\mathtt{F}_L, \mathtt{F}'_L} = 1] - \Pr[\mathcal{A}^{\mathtt{G}_L, \mathtt{G}'_L} = 1]\big|.$$

In this notation, we have $\mathbf{Adv}_{\mathtt{F}_L}^{\mathrm{SPRP}}(\mathcal{A}) = \mathbf{Adv}_{\mathcal{A}}((\mathtt{F}_L, \mathtt{F}_L^{-1}), (\mathtt{P}_L, \mathtt{P}_L^{-1}))$. In this paper, we are mainly interested on the oracle algorithms which make bounded number of queries (say the total number of queries are bounded by Q). Note that only information about oracles can be obtained from queries and responses. If $\mathcal{A}$ interacts with a length-preserving random permutation and its inverse then we can assume following :

1. $\mathcal{A}$ is not making any repetition query. Let x_i denote the i^{th} query then $x_i \neq x_j$ whenever i^{th} and j^{th} queries are both either $\mathcal{O}_1$-queries or $\mathcal{O}_2$-queries.

2. If x_i is $\mathcal{O}_1$-query and y_i is its response then there is no $\mathcal{O}_2$-query x_j with $x_j = y_i$ for some $j > i$. Similarly if x_i is $\mathcal{O}_2$-query and y_i is its response then there is no $\mathcal{O}_1$-query x_j with $x_j = y_i$ for some $j > i$.

The responses of the queries which are not of this type, are completely determined from the previous query responses. A set of queries are called **pointless queries** if the above is not true. We say an adversary satisfying the above conditions as an *allowed adversary*. In this paper we only consider allowed adversaries (not making pointless queries). Now we define the insecurity of a random permutation $\mathtt{F}_L$ as the maximum advantage over all allowed adversaries. More precisely,

$$\mathbf{Insec}_{\mathtt{F}_L}^{\mathrm{SPRP}}(Q) = \max_{\mathcal{A}} \mathbf{Adv}_{\mathtt{F}_L}^{\mathrm{SPRP}}(\mathcal{A})$$

where maximum is taken over all allowed adversaries $\mathcal{A}$ which make at most Q queries. Now we state a result which are commonly used in analyzing SPRP.

Theorem 2.1 *(Halevi and Rogaway 2003) Let L be a length set with $m = \min\{\ell : \ell \in L\}$. Let $\mathtt{R}_L$ and $\mathtt{R}'_L$ be independently chosen length-preserving uniform random functions and let $\mathtt{P}_L$ be length-preserving uniform random permutation. Then for any allowed adversary $\mathcal{A}$ which makes at most Q queries, we have,*

$$\mathbf{Adv}_{\mathcal{A}}((\mathtt{P}_L, {\mathtt{P}_L}^{-1}), (\mathtt{R}_L, \mathtt{R}'_L)) \leq \frac{Q(Q-1)}{2^{m+1}}.$$

The above result says that a uniform length-preserving random permutation is very close to a uniform length-preserving random function. If we want to prove that an enciphering scheme is SPRP-secure then it would be enough to bound the distinguishing advantage from uniform random function.

$$\begin{aligned}\mathbf{Adv}_{\mathcal{A}}((\mathtt{F}_L, \mathtt{F}_L^{-1}), (\mathtt{P}_L, {\mathtt{P}_L}^{-1})) &\leq \mathbf{Adv}_{\mathcal{A}}((\mathtt{F}_L, \mathtt{F}_L^{-1}), (\mathtt{R}_L, \mathtt{R}'_L)) + \mathbf{Adv}_{\mathcal{A}}((\mathtt{P}_L, {\mathtt{P}_L}^{-1}), (\mathtt{R}_L, \mathtt{R}'_L)) \\ &\leq \mathbf{Adv}_{\mathcal{A}}((\mathtt{F}_L, \mathtt{F}_L^{-1}), (\mathtt{R}_L, \mathtt{R}'_L)) + \frac{Q(Q-1)}{2^{m+1}}\end{aligned}$$

The first inequality is true by using simple replacement argument and the second inequality is obtained by using the Theorem 2.1. So if we can obtain an upper bound of $\mathbf{Adv}_{\mathcal{A}}((\mathtt{F}_L, \mathtt{F}_L^{-1}), (\mathtt{R}_L, \mathtt{R}'_L))$ then we can also obtain an upper bound of $\mathbf{Adv}_{\mathcal{A}}((\mathtt{F}_L, \mathtt{F}_L^{-1}), (\mathtt{P}_L, {\mathtt{P}_L}^{-1}))$. In particular,

$$\mathbf{Insec}^{\mathrm{SPRP}}_{\mathtt{F}_L}(Q) \leq \max_{\mathcal{A}} \mathbf{Adv}_{\mathcal{A}}((\mathtt{F}_L, \mathtt{F}_L^{-1}), (\mathtt{R}_L, \mathtt{R}'_L)) + \frac{Q(Q-1)}{2^{n+1}} \tag{1}$$

where maximum is taken over all allowed adversaries $\mathcal{A}$ which make at most Q queries. We consider the message space with length set $[n, \infty)$ and hence $m = n$. We use the above equation to prove our main theorem.

2.2 WPRF or weak pseudorandom function

We can similarly define an adversary which interacts with one oracle. The prf-advantage of an adversary $\mathcal{A}$ for a random function $\mathtt{f}$ from $\{0,1\}^n$ to $\{0,1\}^n$ is defined as

$$\mathbf{Adv}^{\mathrm{prf}}_{\mathtt{f}}(\mathcal{A}) = \left|\Pr[\mathcal{A}^{\mathtt{f}} = 1] - \Pr[\mathcal{A}^{\mathtt{R}_n} = 1]\right|$$

and prf-insecurity of the random function $\mathtt{f}$ is defined as

$$\mathbf{Insec}^{\mathrm{prf}}_{\mathtt{f}}(Q) = \max_{\mathcal{A}} \mathbf{Adv}^{\mathrm{prf}}_{\mathtt{f}}(\mathcal{A})$$

where maximum is taken over all adversary $\mathcal{A}$ which makes at most Q queries. Recall that $\mathtt{R}_n$ is a uniform random function defined over the set of all n bits to itself. Thus, on any distinct inputs it outputs from $\{0,1\}^n$ which are uniformly and independently distributed. Weak pseudorandom function or WPRF is a similar to prf with respect to known plaintext attack. In particular, the plaintexts are chosen at random and given to the attacker. One can equivalently define WPRF where attacker is choosing the queries uniformly and independently of previous query responses. Since the query distribution is independent of the previous query-responses, it really does not matter by whom queries have been selected. We define **weak-prf** insecurity as

$$\mathbf{Insec}^{\mathrm{WPRF}}_{\mathtt{f}}(Q) = \max_{\mathcal{A}} \mathbf{Adv}^{\mathrm{prf}}_{\mathtt{f}}(\mathcal{A})$$

where maximum is taken over all adversary $\mathcal{A}$ which makes at most Q queries and all queries are uniformly and independently distributed over $\{0,1\}^n$. Thus, only difference between prf and WPRF is the nature of queries of the distinguisher. In case of prf distinguisher, the query can be made adaptively and hence it is not necessarily have uniform and independent distribution. In fact if it is adaptive in nature then the queries are actually not independent. Clearly, any prf or SPRP-secure construction is weak-prf but the converse need not be true (Maurer and Sjdin 2007). So potentially we can have an efficient implementation of WPRF. In fact, achieving weak-prf may be easier than to achieve prf or SPRP security (Maurer and Sjdin 2007). For example, let $f : \{0,1\}^{b+n} \to \{0,1\}^n$ be a good compression function. We can assume $f(K, \cdot) : \{0,1\}^n \to \{0,1\}^n$ as a WPRF where $K \in \{0,1\}^b$. Since a SPRP is WPRF we can also consider the AES as a possible candidate of WPRF too.

2.3 Universal Hash Function

Now we define another important object known as a universal hash function.

Definition 2.3 *A random function* $\mathtt{H}$ *from* $\{0,1\}^{2n}$ *to* $\{0,1\}^n$ *is* ϵ*-universal if*

$$\textit{for all } (x,y) \neq (x',y'), \quad \Pr[\mathtt{H}(x,y) = \mathtt{H}(x',y')] \leq \epsilon.$$

In other words, a keyed function family $H : \mathcal{K} \times \{0,1\}^{2n} \to \{0,1\}^n$ *is* ϵ*-universal if* $\Pr[H_K(x,y) = H_K(x',y')] \leq \epsilon$ *for all* $(x,y) \neq (x',y')$ *where probability is computed w.r.t. the uniform probability distribution of* K *over* $\mathcal{K}$*.*

A simple example is based on field multiplication. Let $*$ denote the field multiplication over $\{0,1\}^n$. Let K be chosen at random from $\{0,1\}^n$. Define $\mathtt{H}_K(x,y) = K * x \oplus y$. Now it is easy to see that $\mathtt{H}_K$ is $\frac{1}{2^n}$-universal hash function. There are several other examples of universal hash function which are much efficient in software (Nevelsteen and Preneel 1999; Bernstein. 2005). Note that this universal hash function has the following property. Given key K, the value of y is uniquely determined from $\mathtt{H}_K(x,y)$ and x. More precisely, $y = \mathtt{H}_K(x,y) \oplus (K * x)$.

3 The new domain extension $\mathrm{DE}[\mathbf{E}, \boldsymbol{f}, \mathtt{h}]$

An enciphering scheme $\mathbf{E}$ over a complete message space $\mathcal{M}$ is a keyed permutation family $\mathbf{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ where for each key $K \in \mathcal{K}$, $\mathbf{E}(K,\cdot)$ is a length-preserving permutation on $\mathcal{M}$. The complete message space $\mathcal{M}$ is called the domain of the enciphering scheme. Note that if we choose the key $\mathtt{K}$ uniformly from the keyspace $\mathcal{K}$ then we obtain a length-preserving random permutation $\mathbf{E}(\mathtt{K},\cdot)$. We also denote $\mathbf{E}_\mathtt{K}$ for $\mathbf{E}(\mathtt{K},\cdot)$. Now we propose a generic method to extend the domain of an enciphering scheme. More precisely, if we have a secure enciphering scheme $\mathbf{E}$ with domain $\mathcal{M}_{n^+}$ then we can construct a secure enciphering scheme $\overline{\mathbf{E}}$ with domain $\mathcal{M}_{\geq n}$. Let $x \xleftarrow{\$} S$ represents that x is chosen uniformly from the set S. Recall that $\oplus$ denote the bitwise xor over the set of all n-bits.

Algorithm $\overline{\mathbf{E}} = \mathrm{DE}[\mathbf{E}, \boldsymbol{f}, \mathtt{H}]$

1. Building blocks :

1. Let $\mathbf{E} : \mathcal{K}_1 \times (\{0,1\}^n)^+ \to (\{0,1\}^n)^+$ be a keyed family of length-preserving permutations. Thus for each key $K_1 \in \mathcal{K}_1$ the function $\mathbf{E}_{K_1} := \mathbf{E}(K_1,\cdot) : (\{0,1\}^n)^+ \to (\{0,1\}^n)^+$ is a length-preserving permutation.
2. Let $f : \mathcal{K}_2 \times \{0,1\}^n \to \{0,1\}^n$ be a keyed family of function. We denote $f_{K_2}(\cdot)$ for $f(K_2,\cdot)$.
3. Let $\mathtt{H} : \mathcal{K}_3 \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a keyed family of hash function (a universal hash function). We also use the notation $\mathtt{H}_{K_3}(\cdot)$ for $\mathtt{H}(K_3,\cdot)$. We also need to assume that H has invertibility property. That is, the value of y is uniquely determined from $\mathtt{H}_K(x,y)$ and x. We write $\mathtt{H}_K^{-1}(x,y') = y$ where $y' = \mathtt{H}_K(x,y)$.

2. Key generation : $\mathtt{K}_1 \xleftarrow{\$} \mathcal{K}_1$, $\mathtt{K}_2 \xleftarrow{\$} \mathcal{K}_2$ and $\mathtt{K}_3 \xleftarrow{\$} \mathcal{K}_3$ are chosen uniformly and independently. The triple $(\mathtt{K}_1, \mathtt{K}_2, \mathtt{K}_3)$ is the secret key of $\overline{\mathbf{E}}$. For each such triple we define a length-preserving permutation $\overline{\mathbf{E}}_{\mathtt{K}_1,\mathtt{K}_2,\mathtt{K}_3}$ as given in below.

3. Encryption : Plaintext : $(M_1, \cdots, M_\ell, x) \in \{0,1\}^{\geq n}$ where $|M_i| = n$, $1 \leq i \leq \ell$ and $0 \leq |x| := s < n$.
The corresponding ciphertext $\overline{\mathbf{E}}_{\mathtt{K}_1,\mathtt{K}_2,\mathtt{K}_3}(M_1, \cdots, M_\ell, x)$ is computed as follows.

step-1 $\quad M'_\ell = \mathtt{H}_{K_3}(\overline{x}, M_\ell)$;

step-2 $\quad (C_1, \cdots, C_{\ell-1}, C'_\ell) = \mathbf{E}_{\mathtt{K}_1}(M_1, \cdots, M_{\ell-1}, M'_\ell)$;

step-3 $y = f_{\mathtt{K}_2}(M'_\ell \oplus C'_\ell)[s] \oplus x$;

step-4 $C_\ell = \mathtt{H}_{K_3}(\overline{y}, C'_\ell)$;

step-5 return $(C_1, \cdots, C_\ell, y)$;

4. Decryption : Ciphertext : $(C_1, \cdots, C_\ell, y) \in \{0,1\}^{\geq n}$ where $|C_i| = n$, $1 \leq i \leq \ell$ and $0 \leq |y| := s < n$. The corresponding plaintext $\overline{\mathbf{E}}^{-1}_{\mathtt{K}_1,\mathtt{K}_2,\mathtt{K}_3}(C_1, \cdots, C_\ell, y)$ is computed as follows.

step-1 $C'_\ell = \mathtt{H}^{-1}_{K_3}(\overline{y}, C_\ell)$;

step-2 $(M_1, \cdots, M_{\ell-1}, M'_\ell) = \mathbf{E}^{-1}_{\mathtt{K}_1}(C_1, \cdots, C_{\ell-1}, C'_\ell)$;

step-3 $y = f_{\mathtt{K}_2}(M'_\ell \oplus C'_\ell)[s] \oplus y$;

step-4 $M_\ell = \mathtt{H}^{-1}_{K_3}(\overline{x}, M'_\ell)$;

step-5 return $(M_1, \cdots, M_\ell, x)$;

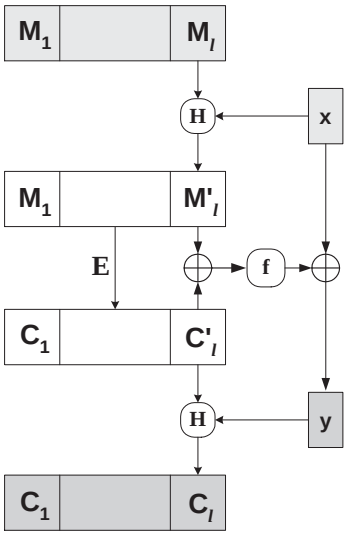

Fig 1. Domain Extension $\mathrm{DE}[\mathbf{E}, f, H]$ where $\mathbf{E}$ is an enciphering scheme with domain $\mathcal{M}_{n^+}$, $f : \{0,1\}^n \to \{0,1\}^n$ is a WPRF and H is a universal hash function

3.1 Discussion

Our construction is mainly motivated by the counter modes SPRP. In the counter mode enciphering scheme a polynomial hash is evaluated over the message M to obtain the counter say S. The ciphertext is obtained by xoring the plaintext with a pseudorandom bit sequence which is obtained from the counter. In this construction pseudorandom bit sequence is obtained from the WPRF. It is also a generic construction. In other words, this method can be applied to any enciphering scheme $\mathbf{E}$ which can encrypt messages of sizes multiple of n. In the next section, we show that $\overline{\mathbf{E}}$ is SPRP-secure whenever $\mathbf{E}$ is SPRP-secure, f_{K_2} is a WPRF and $\mathtt{H}_{K_3}$ is an ϵ-universal hash function for negligible ϵ. A weak prf and universal hash function are both strictly weaker notions than strong pseudorandom permutation.

In the efficiency point of view, it needs one invocation of $\mathbf{E}$, two invocations of universal hash function $\mathtt{H}$ and one WPRF invocation f. The previous generic construction XLS needs one invocations of $\mathbf{E}$ and two invocations of n-bit SPRP. Since an SPRP is always WPRF we can always implement AES as a candidate of WPRF. If we have an implementation of a universal hash function which is twice as efficient as AES then our domain extension is more efficient than XLS. Moreover, there is a possibility to have a more efficient implementation of WPRF than a block cipher, e.g., a keyed hash function.

3.2 Security analysis

Now we provide a complete, simple and more straightforward security analysis of our domain extension. By abuse of notation we use f and $\mathtt{H}$ to mean f_{K_2} and $\mathtt{H}_{K_3}$ where K_2 and K_3 are chosen independently and uniformly from their key spaces.

Theorem 3.1 *Let $\mathbf{E}$ be a keyed family of length-preserving random permutation defined over $(\{0,1\}^n)^+$. Let f be a keyed family of functions defined from $\{0,1\}^n$ to $\{0,1\}^n$ and $\mathtt{H}$ is an ϵ-universal hash function. Then we have*

$$\mathbf{Insec}^{\mathrm{SPRP}}_{\overline{\mathbf{E}}}(Q) \leq \mathbf{Insec}^{\mathrm{SPRP}}_{\mathbf{E}}(Q) + \mathbf{Insec}^{\mathrm{WPRF}}_{f}(Q) + (2\epsilon + 1/2^n) \times \frac{Q(Q-1)}{2}.$$

Proof. Let $\mathtt{P}_{\geq n}$ and $\mathtt{P}_{n^+}$ denote the uniform length preserving random permutation on $\{0,1\}^{\geq n}$ and $(\{0,1\}^n)^+$ respectively. We denote our proposed length-preserving random permutation as $\overline{\mathbf{E}} = \mathrm{DE}[\mathbf{E}, f, \mathtt{H}]$. Now we define some intermediate length-preserving random functions between $(\mathtt{G}_0, \mathtt{G}'_0) = (\overline{\mathbf{E}}, \overline{\mathbf{E}}^{-1})$ and $(\mathtt{G}_5, \mathtt{G}'_5) = (\mathtt{P}_{\geq n}, \mathtt{P}^{-1}_{\geq n})$. These are namely,

1. $\mathtt{G}_1 = \mathrm{DE}[\mathtt{P}_{n^+}, f, \mathtt{H}]$ and $\mathtt{G}'_1 = \mathtt{G}_1^{-1}$. These two random permutations are obtained by replacing $\mathbf{E}$ by an ideal length-preserving random permutation.

2. $\mathtt{G}_2 = \mathrm{DE}[\mathtt{R}'_{n^+}, f, \mathtt{H}]$ and $\mathtt{G}'_2 = \mathrm{DE}[\mathtt{R}''_{n^+}, f, \mathtt{H}]$, where $\mathtt{R}'_{n^+}$ and $\mathtt{R}''_{n^+}$ are independently distributed length-preserving uniform random function on n^+. Thus we replace uniform random permutation and its inverse by two independent uniform random functions. Since we only consider those adversary which make no pointless queries, there is no loss in considering two independent uniform random functions (see Theorem 2.1).

3. Now, we replace f by another n-bit independent uniform random function $\mathtt{R}_n$. Thus, $\mathtt{G}_3 = \mathrm{DE}[\mathtt{R}'_{n^+}, \mathtt{R}_n, \mathtt{H}]$ and $\mathtt{G}'_3 = \mathrm{DE}[\mathtt{R}''_{n^+}, \mathtt{R}_n, \mathtt{H}]$.

4. Finally we consider $\mathtt{G}_4 = \mathtt{R}'_{\geq n}$ and $\mathtt{G}'_4 = \mathtt{R}''_{\geq n}$. These are independently distributed uniform length-preserving random function defined over $\{0,1\}^{\geq n}$.

Now we compute advantage of a distinguisher (making pointless queries only) at distinguishing $(\mathtt{G}_i, \mathtt{G}'_i)$ from $(\mathtt{G}_{i+1}, \mathtt{G}'_{i+1})$, $0 \leq i \leq 4$. Then we can apply the triangle inequality for advantages to obtain our main result.

- The maximum advantage distinguishing $(\mathtt{G}_1, \mathtt{G}'_1)$ from $(\mathtt{G}_0, \mathtt{G}'_0)$ is bounded by $\mathbf{Insec}^{\mathrm{SPRP}}_{\mathbf{E}}(Q)$.

  $$\mathbf{Adv}_{\mathcal{A}}((\mathtt{G}_0, \mathtt{G}'_0), (\mathtt{G}_1, \mathtt{G}'_1)) \leq \mathbf{Insec}^{\mathrm{SPRP}}_{\mathbf{E}}(Q).$$

  This follows from a straightforward replacement argument. More precisely, given an adversary $\mathcal{A}$ which can distinguish $(\mathtt{G}_0, \mathtt{G}'_0)$ and $(\mathtt{G}_1, \mathtt{G}'_1)$ with probability p, there is a distinguisher $\mathcal{A}'$ which distinguishes $(\mathbf{E}, \mathbf{E}^{-1})$ and $(\mathtt{P}_{n^+}, \mathtt{P}^{-1}_{n^+})$ with probability at least p. $\mathcal{A}'$ first run the distinguisher $\mathcal{A}$ and the responses of $(\mathtt{G}_1, \mathtt{G}'_1)$ or $(\mathtt{G}_0, \mathtt{G}'_0)$ can be computed based on the responses of $(\mathtt{P}_{n^+}, \mathtt{P}^{-1}_{n^+})$ or $(\mathbf{E}, \mathbf{E})$ respectively.

- The maximum advantage distinguishing $(\mathtt{G}_1, \mathtt{G}'_1)$ from $(\mathtt{G}_2, \mathtt{G}'_2)$ is bounded by $\frac{Q(Q-1)}{2^{n+1}}$. This is true since the distinguishing advantage between a length preserving uniform random permutation and and uniform length-preserving random function is bounded by $\frac{Q(Q-1)}{2^{n+1}}$ where the minimum bit size of any query is at least n (by using Theorem 2.1).

- A similar argument (distinguishing $(\mathtt{G}_1, \mathtt{G}'_1)$ from $(\mathtt{G}_0, \mathtt{G}'_0)$) can be used to prove that

$$\mathbf{Adv}_{\mathcal{A}}((\mathtt{G}_2, \mathtt{G}'_2), (\mathtt{G}_3, \mathtt{G}'_3)) \leq \mathbf{Insec}_f^{\mathrm{WPRF}}(Q).$$

  Note that here we use the fact that inputs of f are uniformly and independently distributed since input of f is nothing but the last block of $(M_1, \cdots, M_{\ell-1}, M'_\ell) \oplus \mathtt{R}'_{n^+}(M_1, \cdots, M_{\ell-1}, M'_\ell)$ or $(M_1, \cdots, M_{\ell-1}, M'_\ell) \oplus \mathtt{R}''_{n^+}(M_1, \cdots, M_{\ell-1}, M'_\ell)$. Thus, either the inputs are equal or these are independently distributed. This property is true for both f and $\mathtt{R}_n$ and hence the above bound of advantage is true.

- When $\mathcal{A}$ is interacting with $(\mathtt{G}_3, \mathtt{G}'_3)$ the probability that there is a collision among all inputs of $\mathtt{R}'_{n^+}$ (in case of $\mathtt{G}_3$ queries) or all inputs of $\mathtt{R}''_{n^+}$ (in case of $\mathtt{G}'_3$ queries) is bounded by $\epsilon \times Q(Q-1)/2$. This is true since the function $\mathtt{H}$ is ϵ universal hash function and we need to compare at most $Q(Q-1)/2$ pairs. Given that all inputs of $\mathtt{R}'_{n^+}$ and $\mathtt{R}''_{n^+}$ are distinct the probability that there is a collision among all inputs of $\mathtt{R}_n$, is also at most $\epsilon \times Q(Q-1)/2$. Since $\mathtt{R}_n$ is independently distributed from $\mathtt{R}'_{n^+}$ and $\mathtt{R}''_{n^+}$, the complete responses will behave as uniformly and independently distributed strings unless any two of the above event occurs. Thus, we have

$$\mathbf{Adv}_{\mathcal{A}}((\mathtt{G}_3, \mathtt{G}'_3), (\mathtt{G}_4, \mathtt{G}'_4)) \leq \epsilon Q(Q-1).$$

Now we use triangle inequalities for advantages and Theorem 2.1 to obtain the result.

3.3 Tweakable SPRP security analysis

Strong pseudorandom permutation (Luby and Rackoff 1988)is one of the desired security notions for symmetric key encryptions. Later, Liskov et al. (Liskov, Rivest, and Wagner 2002) followed by Halevi-Rogaway (Halevi and Rogaway 2003) considered tweakable version of length-preserving SPRP, which allows us to process associated data or tweak as a part of the messages. Disk-encryption is one of the important application for the length-preserving tweakable SPRP as mentioned in (Halevi and Rogaway 2003). Motivated by disc-encryption algorithms, there are several tweakable SPRP proposals.

Here we briefly describe tweakable enciphering scheme or TES over domain $\mathcal{M}_L$ for some length set L. A tweakable enciphering scheme is a function $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M}_L \to \mathcal{M}_L$, where $\mathcal{K} \neq \emptyset$ and $\mathcal{T} \neq \emptyset$ are the key space and the tweak space respectively. We shall write $\mathbf{E}_K^T(.)$ instead of $\mathbf{E}(K, T, .)$. The inverse of an enciphering scheme is $\mathbf{D} = \mathbf{E}^{-1}$ where $X = \mathbf{D}_K^T(Y)$ if and only if $\mathbf{E}_K^T(X) = Y$.

Let $\mathrm{Perm}^{\mathcal{T}}(\mathcal{M}_L)$ denote the set of all functions $\pi : \mathcal{T} \times \mathcal{M}_L \to \mathcal{M}_L$ where $\pi(\mathcal{T}, .)$ is a length preserving permutation. Such a $\pi \in \mathrm{Perm}^{\mathcal{T}}(\mathcal{M}_L)$ is called a tweak indexed permutation. For a tweakable enciphering scheme $\mathbf{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M}_L \to \mathcal{M}_L$, we define the advantage of an adversary $\mathcal{A}$ has in distinguishing $\mathbf{E}$ and its inverse from a random tweak indexed permutation and its inverse in the following manner.

$$\mathbf{Adv}_{\mathbf{E}}^{\mathrm{tSPRP}}(\mathcal{A}) = \left| \Pr\left[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathbf{E}_K(.,.), \mathbf{E}_K^{-1}(.,.)} \Rightarrow 1\right] - \Pr\left[\pi \xleftarrow{\$} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M}_L) : \mathcal{A}^{\pi(.,.), \pi^{-1}(.,.)} \Rightarrow 1\right] \right|. \quad (2)$$

Here, $\pi \xleftarrow{\$} \mathrm{Perm}^{\mathcal{T}}(\mathcal{M}_L)$ means that for each $\ell \in L$ and $T \in \mathcal{T}$ we choose a tweakable random permutation π^T from $\mathrm{Perm}(\ell)$ independently. We define $\mathbf{Insec}_{\mathbf{E}}^{\mathrm{tSPRP}}(q, \sigma)$ by $\max_{\mathcal{A}} \mathbf{Adv}_{\mathbf{E}}^{\mathrm{tSPRP}}(\mathcal{A})$ where maximum is taken over all allowed adversaries which makes at most q queries having at most σ many blocks. Now we define tweakable version of SPRP security. We skip the proof as it is very similar to Theorem 3.1.

Theorem 3.2 *Let* $\mathbf{E}$ *be a keyed family of tweakable length-preserving random permutation defined over* $(\{0,1\}^n)^+$. *Let* f *be a keyed family of functions defined from* $\{0,1\}^n$ *to* $\{0,1\}^n$ *and* $\mathtt{H}$ *is an* ϵ*-universal hash function. Then we have*

$$\mathbf{Insec}^{\mathrm{tSPRP}}_{\mathbf{E}}(q,\sigma) \leq \mathbf{Insec}^{\mathrm{tSPRP}}_{\mathbf{E}}(q,\sigma) + \mathbf{Insec}^{\mathrm{WPRF}}_{f}(q) + (2\epsilon + 1/2^n) \times \frac{q(q-1)}{2}.$$

4 Conclusion

This paper presents a generic method to construct an encryption algorithm defined over arbitrary messages of size at least n out of an encryption algorithm which only can encrypt messages of size multiple of n. This method is potentially more efficient than recently proposed generic construction XLS. This approach has similarity with the approaches used in counter modes SPRP. But, those approaches are specific for counter modes SPRP and it is not clear how it can be used for other non-counter type constructions such as HEH, TET, EME etc. It is true that this generic approach may not give more efficient construction for variable length encryption (e.g., EME^* is efficient compared with our method applied to EME). But, most of the cases it provides a similar performance as the original variants for the specific constructions (for example, HEH and all counter based modes of operations) except the fact that it uses more keys. It would be interesting to have a generic secure domain extension without using any extra key. As of a theoretical interest, this result would carry a significance contribution and provides some idea how one extend domain for a given security notion in an efficient manner based on security notions which are as much as possible weaker security notions.

References

Bellare, M., J. Kilian, and P. Rogaway (1994). The security of cipher block chaining. In Y. Desmedt (Ed.), *CRYPTO*, Volume 839 of *Lecture Notes in Computer Science*, pp. 341–358. Springer.

Bernstein., D. J. (2005). The poly1305-aes message-authentication code. In H. Gilbert and H. Handschuh (Eds.), *FSE*, Volume 3557 of *Lecture Notes in Computer Science*, pp. 32–49. Springer.

Blum, A., M. L. Furst, M. J. Kearns, and R. J. Lipton (1993). Cryptographic primitives based on hard learning problems. In D. R. Stinson (Ed.), *CRYPTO*, Volume 773 of *Lecture Notes in Computer Science*, pp. 278–291. Springer.

Chakraborty, D. and P. Sarkar (2006). HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In R. Barua and T. Lange (Eds.), *INDOCRYPT*, Volume 4329 of *Lecture Notes in Computer Science*, pp. 287–302. Springer.

D. Cook, M. Y. and A. Keromytis. (2004a). Elastic **aes**. Cryptology ePrint Archive, Report 2004/141. `http://eprint.iacr.org/`.

D. Cook, M. Y. and A. Keromytis. (2004b). Elastic block ciphers. Cryptology ePrint Archive, Report 2004/128. `http://eprint.iacr.org/`.

Daemen, J. and V. Rijmen (2002). AES the advanced encryption standard. Springer 2002. `http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael-ammended.pdf`.

Halevi, S. Invertible universal hashing and the tet encryption mode.

Halevi, S. (2004). EME^*: Extending EME to handle arbitrary-length messages with associated data. In A. Canteaut and K. Viswanathan (Eds.), *INDOCRYPT*, Volume 3348 of *Lecture Notes in Computer Science*, pp. 315–327. Springer.

Halevi, S. and P. Rogaway (2003). A tweakable enciphering mode. In D. Boneh (Ed.), *CRYPTO*, Volume 2729 of *Lecture Notes in Computer Science*, pp. 482–499. Springer.

Halevi, S. and P. Rogaway (2004). A parallelizable enciphering mode. In T. Okamoto (Ed.), *CT-RSA*, Volume 2964 of *Lecture Notes in Computer Science*, pp. 292–304. Springer.

Karatsuba, A. and Y. Ofman. Multiplication of multidigit numbers by automata. Soviet Physics-Doklady, 7:595596, 1963.

Liskov, M., R. L. Rivest, and D. Wagner (2002). Tweakable block ciphers. In M. Yung (Ed.), *CRYPTO*, Volume 2442 of *Lecture Notes in Computer Science*, pp. 31–46. Springer.

Luby, M. and C. Rackoff (1988). How to construct pseudorandom permutations from pseudorandom functions. *SIAM J. Comput. 17*(2), 373–386.

Maurer, U. and J. Sjdin (2007). A fast and key-efficient reduction of chosen-ciphertext to known-plaintext security. In *EUROCRYPT*, Volume 4515 of *Lecture Notes in Computer Science*, pp. 498–516. Springer.

McGrew, D. A. and S. R. Fluhrer (2004). The extended codebook (XCB) mode of operation. Cryptology ePrint Archive, Report 2004/278. `http://eprint.iacr.org/`.

Meyer, C. and M. Matyas. (1982). *Cryptography: A New Dimension in Data Security.* John Wiley & 23 Sons, New York.

Naor, M. and O. Reingold (1999). Synthesizers and their application to the parallel construction of pseudo-random functions. *J. Comput. Syst. Sci. 58*(2), 336–375.

Nevelsteen, W. and B. Preneel (1999). Software performance of universal hash functions. In *EUROCRYPT*, Volume 1592 of *Lecture Notes in Computer Science*, pp. 24–41. Springer.

Ristenpart, T. and P. Rogaway. (2007). How to enrich the message space of a cipher. In *FSE*, Volume 4593 of *Lecture Notes in Computer Science*, pp. 101–118. Springer.

Sarkar., P. (2007). Improving upon the tet mode of operation. In *ICISC*, Volume 4817 of *Lecture Notes in Computer Science*, pp. 180–192. Springer.

Wang, P., D. Feng, and W. Wu (2005). HCTR: A variable-input-length enciphering mode. In D. Feng, D. Lin, and M. Yung (Eds.), *CISC*, Volume 3822 of *Lecture Notes in Computer Science*, pp. 175–188. Springer.

Mridul Nandi *received his bachelors, master and Phd degree from Indian Statistical Institute in 1999,2001 and 2005 respectively. The bachelor and master degree were in Statistics and Ph.D. degree was in Computer Science. Currently he is with the Computer Security Division of National Institute of Standards and Technology.*