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

Mridul Nandi

Indian Statistical Institute, Kolkata
mridul.nandi@gmail.com

Abstract. In this paper we present an efficient and secure generic method which can encrypt messages of size at least n. This generic encryption algorithm needs a secure encryption algorithm for messages of multiple of n. The first generic construction, XLS, has been proposed by Ristenpart and Rogaway in FSE-07. It needs two extra invocations of an independently chosen strong pseudorandom permutation or SPRP defined over $\{0,1\}^n$ for encryption of an incomplete message block. Whereas our construction needs only one invocation of a weak pseudorandom function and two multiplications over a finite field (equivalently, two invocations of an universal hash function). We prove here that the proposed method preserves (tweakable) SPRP. This new construction is meaningful for two reasons. Firstly, it is based on weak pseudorandom function which is a weaker security notion than SPRP. Thus we are able to achieve stronger security from a weaker one. Secondly, in practice, finite field multiplication is more efficient than an invocation of SPRP. Hence our method can be more efficient than XLS.

1 Introduction

The notion of domain extension arises in many areas of cryptography such as hash function, pseudorandom function or PRF, strong pseudorandom permutation or SPRP [12] etc. Usually, we design a building block defined for a small and fixed bit size domain. Then, by applying the building block iteratively, we obtain a similar kind of function defined over arbitrary domain. For example, a block cipher defined on n bits can be used to define an encryption algorithm which can encrypt any message of size multiple of n. To define a ciphertext for a message whose size is not multiple of n, one can first use some padding rule and then can apply some encryption algorithm. This methods can not preserve length. In some applications such as disk encryption, length-preserving encryption is desirable. We call a length-preserving encryption as an enciphering scheme. The length-preserving property makes our task more difficult and restricted also. There are some standard tricks like ciphertext stealing [14], applying the underlying block cipher twice to the last full blocks (applicable for EME [8, 6], TET [7], HEH [17]) or using counter-based PRF (applicable for HCTR [18], HCH [2], XCB [13]). But these approaches are not generic. There was a heuristic domain extension by Cook, Yung and Keromytis [3, 4], without having any security proof. The first and so far only one concrete provable secure generic domain extension, called as XLS, has been proposed by Ristenpart and Rogaway [16]. It needs two extra sequential invocations of a SPRP on n-bits whose key is chosen independently from the key of the given enciphering algorithm for the domain $(\{0,1\}^n)^+ = \cup_{i=1}^{\infty}\{0,1\}^{ni}$.

NIST has made a standard for block cipher, called as AES [5]. The Rijndael block cipher has been finally accepted as a standard of block cipher. Usually, we assume AES as a good candidate of PRP or SPRP. AES is very efficient in hardware and software. In hardware, finite field multiplication is more efficient than AES. A field multiplication in $\mathbb{F}_{2^n}$ takes only one cycle by using Karatsuba-Ofman [10] algorithm, whereas AES takes at least 11 clock cycles. XLS domain extension needs

extra two invocations of a block cipher (say AES) along with some simple mixing operations. Thus, it needs at least 22 clock cycles extra to process an incomplete message block. Thus, it would be interesting question whether it is possible to replace an invocation of AES by field multiplications or some other simpler operations. In this paper, we provide a new generic construction which needs only one weak-PRF or WPRF invocation and two finite field multiplications. This result shows how we can preserve SPRP security from a much weaker security notion such as weak-PRF. We can use stream cipher or block cipher (say AES) or any other possible candidates for WPRF. So if we use AES we can have faster implementation than XLS. Moreover, a PRP-weakness of AES would not immediately threat our construction. In Table 1, we have a comparison study. In software, one can use prime field multiplication as described in [1] to make it more faster.

Table 1. The parameters are given for encryption of an incomplete message block.

	XLS [16]	DE [in this paper]
Key size	n	$2n$
Field Multiplication	0	2
PRF/PRP	2 SPRP	1 WPRF
Clock-Cycle	22	13

Our new construction is mainly motivated from counter-based modes of operation. In counter-based constructions, we first compute counter (something like a tag) based on a message and then we use the counter to generate a random bit sequence. In this domain extension, we will use similar structure. We need one weak-PRF f and an n-bit random string $\mathtt{h}$ to encrypt the incomplete message block. We denote it by $\overline{\mathtt{F}} := \mathrm{DE}[\mathtt{F}, f, \mathtt{h}]$ where $\mathtt{F}$ is any given encryption algorithm which can encrypt only messages of size multiple of n. We will prove that $\overline{\mathtt{F}}$ is SPRP (or tweakable SPRP) whenever $\mathtt{F}$ is SPRP (or tweakable SPRP respectively) and f is a weak-pseudorandom function. In a nutshell we are able to replace two invocations of SPRP by one invocation of weak-pseudorandom function and two finite field multiplications.

Organization of the paper We first provide some preliminaries about the security notion. 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.

2 Preliminaries and Notations

Let $x = x_1x_2\cdots x_n \in \{0,1\}^n$ where $x_i \in \{0,1\}$. We denote $x[s] = x_1x_2\cdots x_s$, the first s bits of x where $s \leq n$. We write $|x| = i$ whenever $x \in \{0,1\}^i$. Given any x such that $0 < |x| < n$, we define $\overline{x} = x10^i$ where $i = n - 1 - |x|$. We define $\overline{\lambda} = 0^n$ whereλ is the empty string. Thus, $|\overline{x}| = n$ and $\overline{x} \neq 0^n$ for any x. Moreover, $\overline{x} \neq \overline{x}'$ whenever $x \neq x'$. We identify $\{0,1\}^n$ as $\mathbb{F}_{2^n}$ with the field addition $\oplus$ (bitwise addition) and a field multiplication '$\cdot$'. The field multiplication can be determined by fixing an irreducible polynomial. In this paper, we fix an irreducible polynomial and hence we have a fixed multiplication operation on set of all n bits $\{0,1\}^n$.

A set $\mathcal{M} \subseteq \{0,1\}^*$ is said to be complete with respect to length if there exists a set $L \subseteq \mathbb{N} := \{0,1,2,\cdots\}$ 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 as length-set for $\mathcal{M}$. In this paper, we mainly consider the length-sets as $L = \{n\}$, or

$L = [n, \infty] = \{n, n+1, \cdots\}$, or $L = n^+ := \{n, 2n, 3n, \cdots, \}$. We denote the corresponding complete sets as $\mathcal{M}_n$, $\mathcal{M}_{\geq n}$, $\mathcal{M}_{n^+}$ respectively.

For a complete set $\mathcal{M} = \mathcal{M}_L$, a permutation $F : \mathcal{M} \to \mathcal{M}$ is called length-preserving (in short, l.p.) if $F_i := F|_{\mathcal{M}_i}$ is a permutation on $\mathcal{M}_i$ for all $i \in L$. Here, $F|_{\mathcal{M}_i}$ denotes the function F restricted on $\mathcal{M}_i$. Thus, $|F(x)| = |x|$, for all $x \in \mathcal{M}$. Given a l.p. 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 a permutation on $\{0,1\}^i$. The inverse l.p. permutation F^{-1} can be similarly characterized by the sequence $\langle F_i^{-1} \rangle_{i \in L}$.

A random function from A to B is a probability distribution on $\mathrm{Func}(A, B)$, the set of all functions from A to B, where A and B are finite set. In other words, we choose a function f from $\mathrm{Func}(A, B)$ according to the probability distribution. 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. Now we define the following ideal random functions which will be considered in the security definitions later.

1. Let $\mathtt{R}_i$ denote the uniform random function from $\{0,1\}^i$ to $\{0,1\}^i$, i.e., the uniform 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. We say it as 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. Instead it is a sequence or tuples of random functions. In this paper we are interested in length-preserving uniform random function where domain is $\{0,1\}^{\geq n}$ or $(\{0,1\}^n)^+$. We denote as $\mathtt{R}_{\geq n}$ or $\mathtt{R}_{n^+}$.
2. Let $\mathtt{P}_i$ denote the uniform random permutation on $\{0,1\}^i$, i.e., the uniform distribution on $\mathrm{Perm}(\{0,1\}^i, \{0,1\}^i)$. Note that the inverse random permutation, $\mathtt{P}_i^{-1}$, is also an uniform random permutation. We similarly define $\mathtt{P}_L$ on $\mathcal{M}_L$ and $\mathtt{P}_L^{-1} = \langle \mathtt{P}_i^{-1} \rangle_{i \in L}$ called as length-preserving uniform random permutation on $\mathcal{M}_L$. Like uniform random function we also consider uniform length-preserving random permutation defined over $\{0,1\}^{\geq n}$ or $(\{0,1\}^n)^+$ and these will be denoted as $\mathtt{P}_{\geq n}$ or $\mathtt{P}_{n^+}$. respectively.

2.1 Security notion : SPRP

Now let $\mathcal{A}$ be an oracle algorithm which has access of two oracles $\mathcal{O}_1$ and $\mathcal{O}_2$. Suppose $\mathcal{A}$ makes queries from the set $\mathcal{M}_L$ for both oracles. Now we define SPRP-advantage of $\mathcal{A}$ for a length-preserving random permutation $\mathtt{F}_L$ by

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

Here oracles are considered as a sequence of random functions. For each $\mathcal{O}_1$-query $x \in \{0,1\}^i$, $i \in L$, $\mathtt{F}_L$ (or $\mathtt{P}_L$) responses $\mathtt{F}_i(x)$ (or $\mathtt{P}_i(x)$ respectively). Similarly, for the inverse query $\mathcal{O}_2$. 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, a function of $(x_1, y_1, \cdots, x_{i-1}, y_{i-1})$, to either $\mathtt{F}_L$ or $\mathtt{F}_L^{-1}$. 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 $\mathtt{F}_{\ell_i}^{-1}(x_i)$, where $x_i \in \{0,1\}^{\ell_i}$.
2. After making q queries, $\mathcal{A}$ returns 0 or 1 depending all query-responses $((x_1, y_1, \delta_1), \cdots, (x_q, y_q, \delta_q))$ where δ_i is either 1 or 2 depending on $\mathcal{O}_1$ or $\mathcal{O}_2$-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

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

In this paper, we are mainly interested on the oracle algorithm which makes bounded number of queries (say the total number of queries are bounded by Q) but the total time computation of it can be unbounded. Thus, there is no loss in considering only deterministic algorithm. Since we know that deterministic algorithm with unbounded computational power is as powerful as a randomized algorithm. If $\mathcal{A}$ interacts with a sequence of random permutations then we can assume following :

1. $\mathcal{A}$ is not making any repetition query.
2. If x_i is $\mathtt{F}$-query and y_i is its response then there is no $\mathtt{F}^{-1}$-query $x_j = y_i$ for some $j > i$ and vice-versa.

We can assume these since the responses are determined for these types of queries. A set of queries are called as pointless queries if the above is not true. We say a deterministic adversary satisfying the above conditions as an *allowed adversary*. In this paper we will 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,

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

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

Theorem 1. *[9] 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,*

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

where $m = \min\{\ell : \ell \in L\}$.

The above result says that an uniform length-preserving random permutation is very close to an uniform length-preserving random function. Thus 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.

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

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

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

where maximum is taken over all adversary $\mathcal{A}$ which makes at most Q queries. 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 $\{0,1\}^n$. Clearly, any prf or sprp-secure construction is weak-prf but the converse need not be true. In fact, achieving weak-prf may be easier than to achieve prf or sprp security. We can use good stream cipher to obtain which would be much faster than a block cipher invocation. The block cipher is believed to be a sprp-secure candidate, for example, AES.

3 The new domain extension DE[E, f, h]

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 (i.e., $|\mathbf{E}_{K_1}(M)| = |M|$). Let $f : \mathcal{K}_2 \times \{0,1\}^n \to \{0,1\}^n$ be another keyed family of function on the set of n bits. We also use the notation $f_{K_2}(\cdot)$ for $f(K_2, \cdot)$. Now we define a length preserving enciphering scheme $\overline{\mathbf{E}}$ defined over $\{0,1\}^{\geq n}$.

Key generation : $\mathtt{K}_1 \xleftarrow{\$} \mathcal{K}_1$, $\mathtt{K}_2 \xleftarrow{\$} \mathcal{K}_2$ and $\mathtt{h} \xleftarrow{\$} \{0,1\}^n$ uniformly and independently. The triple $(\mathtt{K}_1, \mathtt{K}_2, \mathtt{h})$ is the secret key of $\overline{\mathbf{E}}$. For each such triple we define a length-preserving permutation $\overline{\mathbf{E}}_{\mathtt{K}_1,\mathtt{K}_2,\mathtt{h}}$ as given in below.

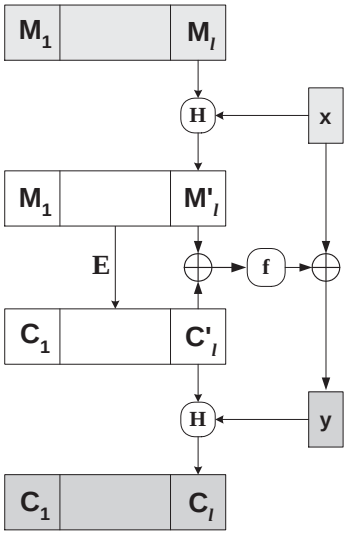

Fig. 1. Domain Extension DE[$\mathbf{E}, f, H$] where $\pi : (\{0,1\}^n)^+ \to (\{0,1\}^n)^+$ is a length preserving permutation, $f : \{0,1\}^n \to \{0,1\}^n$ is any function, $H : \cup_{i=n}^{2n-1}\{0,1\}^i \to \{0,1\}^n$ is an universal hash function such that $H(H(M,x),x) = M$.

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

step-1 $M'_\ell = \mathtt{h} * \overline{x} \oplus M_\ell$;

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

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

step-4 $C_\ell = \mathtt{h} * \overline{y} \oplus C'_\ell$;

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

Remark 1. Note that when $|x| = 0$ (in other words the message size is multiple of n) the above permutation $\overline{\mathbf{E}}_{\mathtt{K}_1,\mathtt{K}_2,\mathtt{h}}$ is same as the permutation $\mathbf{E}_{\mathtt{K}_1}$. It is also easy to see that $\overline{\mathbf{E}}$ is a keyed family of length-preserving permutation. One can compute the decryption exactly same as encryption except in step-2 we have to apply $\mathbf{E}^{-1}_{\mathtt{K}_1}$ instead of $\mathbf{E}_{\mathtt{K}_1}$. Thus, it is a length-preserving permutation on $\{0,1\}^{\geq n}$.

Remark 2. Let $H_{\mathtt{h}}(M_\ell, x) = h * \overline{x} \oplus M_\ell$. It is easy to see that the function keyed function family $H : \{0,1\}^n \times \cup_{i=n}^{2n-1}\{0,1\}^i \rightarrow \{0,1\}^n$ is an *universal hash function.* Moreover, M_ℓ is uniquely determined from x and $N = H_{\mathtt{h}}(M_\ell, x)$ and can be computed as $M_\ell = H_{\mathtt{h}}(N, x)$. A variant of encryption algorithm can be obtained by replacing step-1 by $M'_\ell = H_{\mathtt{h}}(M_\ell, x)$ and step-4 by $C_\ell = H_{\mathtt{h}}(C'_\ell, y)$ where $H_{\mathtt{h}}$ is an universal hash function such that $H_{\mathtt{h}}(H_{\mathtt{h}}(M_\ell, x), x) = M_\ell$.

3.1 Discussion

Our construction is mainly motivated from the counter modes SPRP. 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 and f is a *weak pseudorandom function.* A weak prf is a strictly weaker notion than strong pseudorandom permutation.

In the efficiency point of view, it needs one invocation of $\mathbf{E}$, two field multiplications over $\mathbb{F}_{2^n}$ and one WPRF invocation f. The previous generic construction XLS needs one invocations of $\mathbf{E}$ and two invocations of n-bit SPRP. In hardware, field multiplication is much more efficient than an invocation of AES (a possible candidate of SPRP) and hence this new construction could be an improved generic method to process incomplete message block. A field multiplication in $\mathbb{F}_{2^n}$ takes only one cycle by using Karatsuba-Ofman [10] algorithm, whereas AES takes at least 11 clock cycles. XLS domain extension needs extra two sequential invocations of a block cipher AES (say) along with some simple mixing operations. Thus, it needs at least 22 clock cycles extra to process an incomplete message block. Whereas our construction needs 13 clock cycles if we instantiate a weak prf by AES. A more efficient instantiation of weak prf can make it more faster. For example, one can use LFSR to process the incomplete message block. In Table 1, we have a comparison study. In software, one can use prime field multiplication as described in [1] to make it more faster.

To have a simple and clear presentation we skip the tweakable version. One can easily incorporate tweak if the underlying enciphering scheme $\mathbf{E}$ is tweakable SPRP. More precisely we provide tweak as an input of $\mathbf{E}$. Similar security analysis can be carried and hence we will skip the security proof for a tweakable SPRP.

3.2 Security analysis

Now we provide a complete, simple and more straightforward security analysis of our domain extension. 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 1).
3. Now, we replace f by another n-bit 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 1).

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

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

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 $Q(Q-1)/2^{n+1}$. This is true since the function $H_{\mathtt{h}}$ is $1/2^n$ 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 $Q(Q-1)/2^{n+1}$. This is the birthday collision probability of Q uniformly and independently distributed strings (the xor of last block of input and output of $\mathtt{R}'_{n^+}$ and $\mathtt{R}''_{n^+}$). Since $\mathtt{R}_n$ is independently distributed from $\mathtt{R}'_{n^+}$ and $\mathtt{R}''_{n^+}$, the complete responses will behave as an uniformly and independently distributed strings unless any two of the above event occurs. Thus, we have

$$\mathbf{Adv}_{\mathcal{A}}((\mathtt{G}_3, \mathtt{G}'_3), (\mathtt{G}_4, \mathtt{G}'_4)) \leq \frac{Q(Q-1)}{2^n}.$$

– As stated in distinguishing $(\mathtt{G}_1, \mathtt{G}'_1)$ from $(\mathtt{G}_2, \mathtt{G}'_2)$ the maximum advantage distinguishing $(\mathtt{G}_4, \mathtt{G}'_4)$ from $(\mathtt{G}_5, \mathtt{G}'_5)$ is bounded by $\frac{Q(Q-1)}{2^{n+1}}$.

Now we use triangle inequalities for advantages to obtain the following theorem.

Theorem 2. *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* $\{0,1\}^n$. *Then we have*

$$\mathbf{Insec}^{\mathrm{sprp}}_{\overline{\mathbf{E}}}(Q) \leq \mathbf{Insec}^{\mathrm{sprp}}_{\mathbf{E}}(Q) + \mathbf{Adv}^{\mathrm{wprf}}_{f}(Q) + \frac{3Q(Q-1)}{2^{n+1}}.$$

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 message of size multiple of n. This method is more efficient than recently proposed generic construction XLS. This approach has similarity with all of 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 approach may not give more efficient construction for variable length encryption (e.g., in case of 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). Moreover, as a theoretical interest, this result would carry a significance contribution and provide some idea how one extend domain for a given security notion.

References

1. D. J. Bernstein. The Poly1305-AES message-authentication code. Fast Software Encryption FSE 2005, LNCS vol. 3557/2005, Springer, pp. 32-49, 2005.
2. D. Chakraborty and P. Sarkar. HCH: A new tweakable enciphering scheme using the Hash- Encrypt-Hash approach. Advances in Cryptology INDOCRYPT 2006, LNCS vol. 4329, Springer, pp. 287302, 2006.
3. D. Cook, M. Yung, and A. Keromytis. Elastic AES. Cryptology ePrint archive, report 2004/141, LNCS? 2004.
4. D. Cook, M. Yung, and A. Keromytis. Elastic block ciphers. Cryptology ePrint archive, LNCS?, report 2004/128, 2004.
5. Joan Daemen and Vincent Rijmen The Design of Rijndael: AES The Advanced Encryption Standard. Springer 2002. http://csrc.nist.gov/CryptoToolkit/aes/rijndael/Rijndael-ammended.pdf
6. S. Halevi. EME*: Extending EME to handle arbitrary-length messages with associated data. Advances in Cryptology INDOCRYPT 2004, LNCS vol. 3348, Springer, pp. 315327, 2004.

7. S. Halevi. TET: A wide-block tweakable mode based on Naor-Reingold. Advances in Cryptology 2007. Cryptology ePrint archive, report 20007/14, CRYPTO? 2007.
8. S. Halevi and P. Rogaway. A parallelizable enciphering mode. Topics in Cryptology CT-RSA 2004, LNCS vol. 2964, Springer, pp. 292304, 2004.
9. S. Halevi and P. Rogaway. A tweakable enciphering mode. Advances in Cryptology CRYPTO 2003, LNCS vol. 2729, Springer, pp. 482499, 2003.
10. A. Karatsuba and Y. Ofman. Multiplication of multidigit numbers by automata. Soviet Physics-Doklady, 7:595596, 1963.
11. M. Liskov, R. Rivest, and D. Wagner. Tweakable block ciphers. Advances in Cryptology CRYPTO 2002, LNCS vol. 2442, Springer, pp. 3146, 2002.
12. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. SIAM Journal of Computing, vol. 17, no. 2, pp. 373386, 1988.
13. D. McGrew and S. Fluhrer. The extended codebook (XCB) mode of operation. Cryptology ePrint archive, report 2007/298, Proceedings in Selected Areas in Cryptography.
14. C. Meyer and M. Matyas. Cryptography: A New Dimension in Data Security. John Wiley & 23 Sons, New York, 1982.
15. M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. Journal of Cryptology, vol. 12, no. 1, pp. 2966, 1999.
16. Thomas Ristenpart and Phillip Rogaway. How to Enrich the Message Space of a Cipher. Fast Software Encryption - FSE 2007, Lecture Notes in Computer Science Vol. 4593/2007, pp. 101-118, Springer-Verlag, 2007.
17. Palash Sarkar. Improving Upon the TET Mode of Operation. Cryptology ePrint archive, report 2007/317.
18. P. Wang, D. Feng, and W. Wu. HCTR: a variable-input-length enciphering mode. Information Security and Cryptography, CISC 2005, LNCS vol. 3822, Springer, pp. 175188, 2005.