Length-doubling Ciphers and Tweakable Ciphers

Haibin Zhang

Dept. of Computer Science, University of California, Davis, USA
hbzhang@cs.ucdavis.edu

Abstract. We motivate and describe a mode of operation HEM (resp., THEM) that turns a n-bit blockcipher into a variable-input-length cipher (resp., tweakable cipher) that acts on strings of $[n..2n-1]$ bits. Both HEM and THEM are simple and intuitive and use only *two* blockcipher calls, while prior work at least takes *three*. We prove them secure in the sense of strong PRP and tweakable strong PRP, assuming the underlying blockcipher is a strong PRP.

Keywords: ciphers, tweakable ciphers, deterministic encryption, enciphering scheme, symmetric encryption, universal hash function.

1 Introduction

Designing modes of operation from a blockcipher (e.g., AES) is one of the central tasks in shared-key cryptography. They allow the repeated use of a blockcipher to achieve confidentiality or authentication for variable-input-length (VIL) messages. Many confidentiality applications like disk-sector encryption require that the encryption be *length-preserving*. The requirement entails the usage of *ciphers* [19]. A cipher is a family of keyed length-preserving permutations. Such a primitive is also called an *enciphering scheme*, or *deterministic encryption*. The conventional security notions for a cipher are "pseudorandom permutation" (PRP) and "strong pseudorandom permutation" (SPRP) [17]. Tweakable cipher [18] is an extension of conventional cipher which takes a "tweak" (or "associated data", that does not have to be encrypted) as an additional input. Correspondingly, the security notions are "tweakable PRP" and "tweakable SPRP".

Compared to many other cryptographic primitives like signatures, MACs, and encryption schemes, it is not easy to build a VIL cipher from a fixed-input-length (FIL) cipher (e.g., blockcipher), where techniques such as "padding" and "tainting" fail to work. Indeed, to this end, a very large number of wide blocksize ciphers [1, 3, 4, 6, 10–13, 27, 28] are proposed (though not all of them can handle settings where the messages need not be a multiple of n bits).

This work mainly considers a special case of this problem—on how to turn a blockcipher of size n into a VIL cipher and a VIL tweakable cipher with the message space $\bigcup_{i=n}^{2n-1} \{0,1\}^i$, which basically "doubles" the domain of a blockcipher in the VIL sense. First, this length-doubling problem is of *historical* interest. Luby and Rackoff's Feistel construction [17] can be viewed as the first attempt to double the domain of a cipher for fixed-input-length (FIL) messages, while our work deals with the problem in the VIL setting and aims to improve its efficiency. Besides its theoretical value, it is particularly applicable to many useful settings. For instance, short headers in the Internet are usually of the targeted lengths that we study. If messages to be encrypted are of such short lengths, one should be careful about the cipher used since otherwise it can influence the efficiency *by percentage*. And, perhaps more importantly, length-doubling ciphers are useful tools to build high-level protocols. In particular, Rogaway and Zhang show how to turn such a VIL length-doubling tweakable cipher into an arbitrary-length-input online cipher [24]. Last, length-doubling cipher seems to be the "right" method of dealing with the incomplete final block for IEEE P1619 standard [21]. This standard is applicable to cases like disk-sector encryption, which cannot afford the extra hardware or latency required by a two-pass wide blocksize encryption mode. Current standard XTS-AES makes use of *ciphertext stealing* [19] to handle the issue. Though P1619 does not provide a formal definition for security property (which is somewhat unfortunate), an intuitive one may be that each block should be enciphered by an independent SPRP (indexed by a tweak), and the "long" final block (i.e., the second last complete block and the partial final block) should be enciphered by an independent length-doubling tweakable SPRP secure in the VIL setting.[1] It is easy to verify that ciphertext stealing construction described in [21] and its possible variants (see, e.g., [25]) do *not* satisfy such a definition of security even in the PRP sense.[2] We believe that it is necessary to reconsider the problem and find efficient alternatives.

On the other hand, the above-mentioned general wide blocksize ciphers, if being restricted to our targeted domain, do not give very efficient length-doubling schemes. For instance, EME2 cipher of Halevi [10] uses five blockcipher calls to achieve a VIL length-doubling tweakable SPRP. It takes at least four blockcipher calls to use unbalanced Feistel networks [26]. The currently best solution is obtained from the XLS construction by Ristenpart and Rogaway [23] which essentially uses three blockcipher calls and little extra work. In fact, *none* of them are designed specifically for the length-doubling problem—a problem that we aim to address in this paper.

Our method. At the heart of our motivation is how to construct *efficient* VIL (tweakable) ciphers using *only two* blockcipher calls. The question itself has both theoretical and practical interest. The blockcipher implementation is still the most expensive one in most of the platforms. Trimming one blockcipher call would be highly likely to result in a considerable improvement in efficiency.[3] But this goal is not as easy as it looks. (From a purely theoretical view—without considering efficiency, the problem can be well solved with good provable-security.)

[1] Another possible definition is to ask the last block to be "short", but this will not give a tight security reduction for very short messages.

[2] We note that Liskov and Minematsu [16] provide a "proof" to justify the security of ciphertext stealing in XTS-AES. However, one can barely argue anything from such a proof since there is no security notion.

[3] Other examples of this kind include [9, 15].

We extend the idea of Naor and Reingold [20] to construct an efficient VIL length-doubling cipher and tweakable cipher. The overhead for the VIL cipher construction is about two blockcipher calls and two AXU hash function calls and little additional work. We name the VIL cipher construction "HEM" to indicate that we use hash function, blockcipher encryption, and mixing function [23] as components.

We begin by describing a mode FHEM (fixed-length HEM) for a fixed length $n+s$ where n is the blocksize and $s \in [1..n-1]$. The mode is depicted in Figure 1. Given an input M of length $n+s$, it first parses M as M_1 and M_2 such that $|M_1| = n$ and $|M_2| = s$. The algorithm takes four "rounds". The first and last rounds use AXU hash functions, and the second and third rounds use regular blockciphers. The overall structure can be viewed as following the framework of Feistel networks, but is neither exactly like Feistel nor unbalanced Feistel networks. The input and output of the hash function are both simply from $\{0,1\}^n$. This is *crucial* for our construction—the efficiency of the hash functions usually decreases rapidly if the input size gets larger, while the security would lose if the output size gets smaller. Furthermore, we use a third tool, a mixing function [23], to "fix" the consequence of not exactly following the four rounds Feistel. This makes our construction a little less elegant but does not essentially hurt efficiency. Besides, different from Naor-Reingold construction, the security assumption needed for the underlying blockcipher is SPRP rather than PRP. Moreover, the round two and three functions must be permutations. Clearly, these two requirements are insignificant to the implementation since one usually chooses to use AES anyway.

It may still seem hard to make a VIL cipher, since intuitively it needs an AXU hash function for VIL messages. We circumvent the problem by applying the same AXU hash function (with an independent key) to the length of M_2. See Figure 3 for our construction. We stress that we can pre-compute all values for the additional hash functions since there are only n of them where n is the underlying blocksize (e.g., 128). Concretely, the additional operations needed compared to FHEM are just two xors. Thus, we can practically make the VIL cipher from the FIL construction with "no" extra work. We comment that this basically uses an idea similar to that used in [22] yet in a more efficient way.

We go on to present two constructions of VIL length-doubling tweakable ciphers. One of them gets better provable security, while the other is more concise.

To instantiate the modes, we can use many ready solutions for AXU hash functions. One notable construction is the GF(2^n) multiply. The efficiency of AXU hash functions may vary due to software and hardware support and other factors, and thus we do not give a specific recommendation. For the mixing function, we recommend using the more efficient one in [23] that takes only three xors and a single one-bit circular rotation. Another immediate consequence of our results is that any progress in the area of AXU hash functions will result in improved VIL length-doubling (tweakable) ciphers.

Further related work. Luby and Rackoff [17] showed the classical result that four rounds of Feistel suffice to construct a length-doubling SPRP in the

FIL sense. Naor and Reingold [20] revisited four rounds Feistel construction and showed that the first and fourth layer blockciphers can be well replaced with two pairwise independent permutations. (In particular, they showed that a weaker almost XOR universal (AXU) hash function is sufficient.) This change improves the efficiency and enables a simpler proof. Patel, Ramzan, and Sundaram [22] constructed a VIL cipher achieving SPRP security for a larger domain $\bigcup_{i\geq 2n}\{0,1\}^i$, by combining unbalanced Feistel networks [26] and pairwise independent permutations. It is not clear how to extend their idea to design even a FIL cipher for our target domain, say, a cipher of length $3n/2$, with a *tight* reduction. Cook, Yung, and Keromytis [5] designed "from scratch" the elastic blockcipher to solve the same length-doubling problem. Their construction is not designed from the perspective of provably secure mode of operation. The XLS mode of operation by Ristenpart and Rogaway [23] in essence solved a more general problem that turns a m-bit-size cipher and a n-bit-size cipher to a cipher that acts on strings of $[m..m+n-1]$ bits. Goldenberg *et al.* addressed the question on how to directly incorporate a tweak on Luby-Rackoff blockciphers [7].

Discussion. Intel released AES-New Instructions (AES-NI) [8], starting from Westmere, in order to more efficiently implement AES. The gap about the efficiency between a well-chosen and carefully-implemented AXU hash function and AES becomes less obvious. However, this change only appears for the recent Intel and AMD architectures. For other platforms, especially for specified hardware-based ones, our scheme outperforms other schemes notably.

Our results also answer an interesting question regarding how to construct *efficient* length-doubling (tweakable) ciphers in the VIL sense using *only two* blockcipher calls, which may be a more important contribution. In fact, the problems studied in our work can be understood from a broader perspective: How do we achieve an *efficient* VIL cipher for messages with the domain $\bigcup_{i\geq n}\{0,1\}^i$ using the *least* blockcipher calls? Of course, this question only makes sense if there exists a lower bound for the number of blockcipher calls for an *efficient* construction. We conjecture on this informal question that it needs at least $\lceil i/n \rceil$ blockcipher calls.

2 Preliminaries

Notation. A *string* is a member of $\{0,1\}^*$. If $A, B \in \{0,1\}^*$ then $A \,\|\, B$ or $A\,B$ denotes their concatenation. If X is a string then $|X|$ denotes its length. The *empty string* is denoted ε. Throughout this work, we fix a number n called the *blocksize*.

Ciphers, blockciphers and tweakable ciphers. A map $f\colon \mathcal{X} \to \mathcal{X}$ for $\mathcal{X} \subseteq \{0,1\}^*$ is a *length-preserving function* if $|f(x)| = |x|$ for all $x \in \{0,1\}^*$. It is a length-preserving *permutation* if it is also a permutation. A *cipher* is a map $\mathcal{E}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ where $\mathcal{K}$ is a nonempty set, $\mathcal{M} \subseteq \{0,1\}^*$ is a nonempty set, and $\mathcal{E}_K = \mathcal{E}(K,\cdot)$ is a length-preserving permutation for all $K \in \mathcal{K}$. The set $\mathcal{K}$ is called the *key space* and $\mathcal{M}$ is called the *message space*. If $\mathcal{E}\colon \mathcal{K} \times$

$\mathcal{M} \to \mathcal{M}$ is a cipher then its *inverse* is the cipher $\mathcal{E}^{-1}: \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ defined by $\mathcal{E}^{-1}(K, Y) = \mathcal{E}_K^{-1}(Y)$ being the unique point X such that $\mathcal{E}_K(X) = Y$. A *blockcipher* is a map $E: \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ where $\mathcal{K}$ is a finite nonempty set and $E_K(\cdot) = E(K, \cdot)$ is a permutation on $\{0,1\}^n$ for every $K \in \mathcal{K}$. Equivalently, a blockcipher is a cipher with message space $\mathcal{M} = \{0,1\}^n$. A *tweakable cipher* is a map $\widetilde{\mathcal{E}}: \mathcal{K} \times \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $\mathcal{K}$ is a finite nonempty set and $\mathcal{T}$ is a nonempty set (the *tweak space*) and $\mathcal{M}$ is a nonempty set (the message space) and $\widetilde{\mathcal{E}}_K^T(\cdot) = \widetilde{\mathcal{E}}(K, T, \cdot)$ is a permutation on $\mathcal{M}$ for every $K \in \mathcal{K}$, $T \in \mathcal{T}$.

Let $\mathrm{Perm}(n)$ be the set of all permutations on n bits, $\mathrm{Perm}(\mathcal{M})$ be the set of all length-preserving permutations on the finite set $\mathcal{M} \subseteq \{0,1\}^*$, and $\mathrm{Perm}(\mathcal{T}, \mathcal{M})$ be the set of all functions $\pi: \mathcal{T} \times \mathcal{M} \to \mathcal{M}$ where $\pi_T(\cdot) = \pi(T, \cdot)$ is a permutation for each $T \in \mathcal{T}$. We may regard $\mathrm{Perm}(n)$, $\mathrm{Perm}(\mathcal{M})$, and $\mathrm{Perm}(\mathcal{T}, \mathcal{M})$ as blockciphers, ciphers, and tweakable ciphers, respectively; they are the ideal blockcipher on n bits, the ideal cipher on $\mathcal{M}$, and the ideal tweakable cipher on message space $\mathcal{M}$ and tweak space $\mathcal{T}$. When an adversary $\mathcal{A}$ is run with an oracle $\mathcal{O}$ we let $\mathcal{A}^{\mathcal{O}} \Rightarrow 1$ denote the event that $\mathcal{A}$ outputs 1. Define the $\pm$prp (i.e., SPRP) and $\pm\widetilde{\mathrm{prp}}$ (i.e., tweakable SPRP) *advantage* of $\mathcal{A}$ against E, $\mathcal{E}$ or $\widetilde{\mathcal{E}}$ by:

$$\mathbf{Adv}_E^{\pm\mathrm{prp}}(\mathcal{A}) = \Pr[\, K \xleftarrow{\$} \mathcal{K} :\ \mathcal{A}^{E_K, E_K^{-1}} \Rightarrow 1\,] - \Pr[\, \pi \xleftarrow{\$} \mathrm{Perm}(n) :\ \mathcal{A}^{\pi, \pi^{-1}} \Rightarrow 1\,]$$
$$\mathbf{Adv}_{\mathcal{E}}^{\pm\mathrm{prp}}(\mathcal{A}) = \Pr[\, K \xleftarrow{\$} \mathcal{K} :\ \mathcal{A}^{\mathcal{E}_K, \mathcal{E}_K^{-1}} \Rightarrow 1\,] - \Pr[\, \pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{M}) :\ \mathcal{A}^{\pi, \pi^{-1}} \Rightarrow 1\,]$$
$$\mathbf{Adv}_{\widetilde{\mathcal{E}}}^{\pm\widetilde{\mathrm{prp}}}(\mathcal{A}) = \Pr[\, K \xleftarrow{\$} \mathcal{K}:\ \mathcal{A}^{\widetilde{\mathcal{E}}_K,\, \widetilde{\mathcal{E}}_K^{-1}} \Rightarrow 1\,] - \Pr[\, \pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{T}, \mathcal{M}):\ \mathcal{A}^{\pi,\, \pi^{-1}} \Rightarrow 1\,]$$

Almost XOR universal hash function. We recall the definition of ϵ-*almost XOR universal* (ϵ-AXU) *hash function* [14]. A hash function $H: \mathcal{K} \times \mathcal{X} \to \{0,1\}^n$ is called ϵ-AXU, if for all distinct $X, X' \in \mathcal{X}$ and all $C \in \{0,1\}^n$, we have that $\Pr[K \xleftarrow{\$} \mathcal{K} :\ H_K(X) \oplus H_K(X') = C] \leq \epsilon$. There are many efficient AXU hash functions candidates. For concreteness, we review one such function for $\mathcal{X} = \{0,1\}^n$—multiplication in *Galois Field* $\mathrm{GF}(2^n)$ (i.e., $H_K(X) = K \cdot X$ where $K, X \in \{0,1\}^n$), which achieves 2^{-n} for ϵ—the minimum value one can hope for. Assume that a, b are strings of $\{0,1\}^n$ where $a = a_{n-1} \cdots a_1 a_0$ and $b = b_{n-1} \cdots b_1 b_0$. The Galois Field addition is defined as their bitwise xor. To multiply $a, b \in \mathrm{GF}(2^n)$, write them as polynomials $a(\mathtt{x}) = a_{n-1}\mathtt{x}^{n-1} + \cdots + a_1\mathtt{x} + a_0$ and $b(\mathtt{x}) = b_{n-1}\mathtt{x}^{n-1} + \cdots + b_1\mathtt{x} + b_0$, compute $c(\mathtt{x}) = a(\mathtt{x}) \cdot b(\mathtt{x}) \bmod p(\mathtt{x})$ where $p(\mathtt{x})$ is a fixed irreducible polynomial over $\mathrm{GF}(2^n)$, and then return the binary representation of $c(\mathtt{x})$ as output. For the AXU hash function input string $X \in \{0,1\}^*$ and $|X| < n$, we let $\mathsf{pad}(X)$ be the string $X \| 0^{n-|X|}$. (Namely, the minimal number of zero-bits are padded on the right such that $\mathsf{pad}(X)$ is a complete block.) Note that for example $\mathsf{pad}(Y) = \mathsf{pad}(Y\|0)$ for some string Y where $|Y| < n$. This all zero padding method can be applied to other AXU hash functions.

Mixing Function. We review the definition of the mixing function formally defined and studied by Ristenpart and Rogaway [23]. We define a mixing function $\mathsf{mix}: \mathcal{S}^2 \to \mathcal{S}^2$ ($\mathcal{S} \supseteq \bigcup_{i=1}^{n-1} \{0,1\}^i$) such that $\mathsf{mix}_{\mathsf{L}}(\cdot,\cdot)$ and $\mathsf{mix}_{\mathsf{R}}(\cdot,\cdot)$ are the left projection and right projection of mix function. Ideally, we want such a primitive to have the property of a *multipermutation*: namely, for any $A \in \mathcal{S}$, $\mathsf{mix_L}(A,\cdot)$, $\mathsf{mix_L}(\cdot,A)$, $\mathsf{mix_R}(A,\cdot)$, and $\mathsf{mix_R}(\cdot,A)$ are all permutations. Like the construction in [23], a relaxed notion of mixing function can work almost as well for our schemes. We say that mix is an $\epsilon(s)$-*good mixing function*, if for all s such that $\{0,1\}^s \in \mathcal{S}$ and all A,B,$C \in \{0,1\}^s$, we have both $\mathsf{mix_L}(A,\cdot)$ and $\mathsf{mix_R}(\cdot,B)$ are permutations, and $\Pr[\,R \xleftarrow{\$} \{0,1\}^s :\ C = \mathsf{mix_L}(R,B)\,]$ and $\Pr[\,R \xleftarrow{\$} \{0,1\}^s :\ C = \mathsf{mix_R}(A,R)\,]$ are both less than $\epsilon(s)$. In their work, two efficient mixing functions are given. The more efficient one with $\epsilon(s) = 2^{1-s}$ only takes three xors and a one-bit circular rotation.

3 A Fix-Input-Length Cipher

In this section, we provide a cipher for a fixed length $n+s$ where n is the blocksize and $s \in [1..n-1]$. In other words, the cipher we shall describe is secure against adversaries who are only allowed to ask queries of length $n+s$. The construction only works in the FIL setting, but it serves as the basis for constructing VIL (tweakable) ciphers.

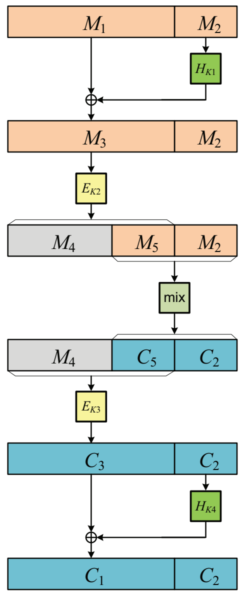
Let $E\colon \mathcal{K}_1 \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and let $H\colon \mathcal{K}_2 \times \{0,1\}^n \to \{0,1\}^n$ be a 2^{-n}-AXU hash function family. Define an $\epsilon(s)$-good mixing function $\mathsf{mix}\colon \mathcal{S}^2 \to \mathcal{S}^2$ where $\mathcal{S} \supseteq \bigcup_{i=1}^{n-1}\{0,1\}^i$. We define a cipher $\mathcal{E} = \mathrm{FHEM}[H,E,\mathsf{mix}]$ with key space $\mathcal{K}_1^2 \times \mathcal{K}_2^2$. See Figure 1 for the construction. We claim that this cipher is $\pm$prp-secure for fixed-input-length $n+s$.

The intuition of the proof is as follows. Like the Naor-Reingold construction, by using AXU-hash function, for any two different messages M^i and M^j of the same length, the probability that M_3^i and M_3^j "collide" is negligible. After applying a random function, the output $M_4||M_5$ is now uniformly distributed. This perfectly hides the complete block M_1, but the partial block M_2 remains unprotected. A mixing function is used to force the output of $M_5||M_2$ to inherit the distribution of M_5. An independent random function is then employed to further hide part of the mixing function output. The overall construction should be made "symmetrizing" in order to achieve strong PRP security. Also note that unlike Naor-Reingold and subsequent work, our constructions (i.e., one in this section and the following extensions) ask the underlying blockcipher to be reversible and the complexity assumption for it is SPRP.

The following theorem establishes the security of FHEM.

Theorem 1. *Let* $\mathcal{E} = \mathrm{FHEM}[H,\mathrm{Perm}(n),\mathsf{mix}]$ *with message space* $\{0,1\}^{n+s}$. *If* $\mathcal{A}$ *asks at most* q *queries then* $\mathbf{Adv}_{\mathcal{E}}^{\pm\mathrm{prp}}(\mathcal{A}) \le 1.5\,q^2/2^n + 0.5\,q^2\frac{\epsilon(s)}{2^{n-s}} + 0.5\,q^2/2^{n+s}$, *and if we use a mixing function with* $\epsilon(s) = 2^{1-s}$ *then we have that* $\mathbf{Adv}_{\mathcal{E}}^{\pm\mathrm{prp}}(\mathcal{A}) \le 3\,q^2/2^n$. ∎

Proof. We assume without loss of generality that $\mathcal{A}$ is deterministic and makes q queries from $\{0,1\}^{n+s}$. We further assume that it does not ask "pointless" queries: it never repeats an encipher query, never repeats a decipher query, never asks a decipher query of a value that it earlier received from an encipher query,



algorithm $\mathcal{E}_K(M)$ **where** $K = K_1||K_2||K_3||K_4$

if $|M| \neq n + s$ **then return** $\perp$

$M_1||M_2 \leftarrow M$ **where** $|M_1| = n$ and $|M_2| = s$

$M_3 \leftarrow M_1 \oplus H_{K_1}(\mathsf{pad}(M_2))$

$M_4||M_5 \leftarrow E_{K_2}(M_3)$ **where** $|M_4| = n - s$ and $|M_5| = s$

$C_5||C_2 \leftarrow \mathsf{mix}(M_5||M_2)$ **where** $|C_5| = |C_2| = s$

$C_3 \leftarrow E_{K_3}(M_4||C_5)$

$C_1 \leftarrow C_3 \oplus H_{K_4}(\mathsf{pad}(C_2))$

$C \leftarrow C_1||C_2$

return C

Fig. 1. Mode FHEM. Each input M is parsed as a complete block M_1 and a partial block M_2. We should pad M_2 to a complete block before applying the AXU hash function. Similar operations should be carried out for the deciphering algorithm.

and never makes an encipher query of a value that it earlier received from a decipher query.

We use the code-based games [2] in Figure 2. Variable *bad* is initialized to false. A functions π is initialized to everywhere undefined. Its current domain and range are denoted domain(π) and range(π), while their complements relative to $\{0,1\}^n$ are denoted codomain(π) and corange(π).

We begin with game G_1, which precisely describes the FHEM construction with the ideal blockcipher π_1 and π_2. Game G_6 always outputs random values, simulating a pair of random functions. Let p denote the probability that $\mathcal{A}$ outputs 1 in the game simulating a random permutation and its inverse. The dif-

procedure $E(M)$	**procedure** $D(C)$
$j \leftarrow j+1;\ \ M^j \leftarrow M$	$j \leftarrow j+1;\ \ C^j \leftarrow C$
$M_1^j \| M_2^j \leftarrow M^j$	$C_1^j \| C_2^j \leftarrow C^j$
$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$	$C_3^j \leftarrow C_1^j \oplus H_{K_1}(\mathsf{pad}(C_2^j))$
$M_4^j \| M_5^j \leftarrow \pi_1(M_3^j)$	$M_4^j \| C_5^j \leftarrow \pi_2^{-1}(C_3^j)$
$C_5^j \| C_2^j \leftarrow \mathsf{mix}(M_5^j \| M_2^j)$	$M_5^j \| M_2^j \leftarrow \mathsf{mix}(C_5^j \| C_2^j)$
$C_3^j \leftarrow \pi_2(M_4^j \| C_5^j)$	$M_3^j \leftarrow \pi_1^{-1}(M_4^j \| M_5^j)$
$C_1^j \leftarrow C_3^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$	$M_1^j \leftarrow M_3^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
$C^j \leftarrow C_1^j \| C_2^j$	$M^j \leftarrow M_1^j \| M_2^j$
return C^j	**return** M^j Game G_1

procedure $E(M)$	**procedure** $D(C)$
$j \leftarrow j+1;\ \ M^j \leftarrow M$	$j \leftarrow j+1;\ \ C^j \leftarrow C$
$M_1^j \| M_2^j \leftarrow M^j$	$C_1^j \| C_2^j \leftarrow C^j$
$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$	$C_3^j \leftarrow C_1^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$
if $M_3^j \notin \mathrm{domain}(\pi_1)$ **then**	**if** $C_3^j \notin \mathrm{range}(\pi_2)$ **then**
$Y_1 \xleftarrow{\$} \{0,1\}^n$	$X_2 \xleftarrow{\$} \{0,1\}^n$
if $Y_1 \in \mathrm{range}(\pi_1)$ **then**	**if** $X_2 \in \mathrm{domain}(\pi_2)$ **then**
$bad \leftarrow \mathsf{true};\ \left[\, Y_1 \xleftarrow{\$} \mathrm{corange}(\pi_1) \,\right]$	$bad \leftarrow \mathsf{true};\ \left[\, X_2 \xleftarrow{\$} \mathrm{codomain}(\pi_2) \,\right]$
$\pi_1(M_3^j) \leftarrow Y_1$	$\pi_2^{-1}(C_3^j) \leftarrow X_2$
$M_4^j \| M_5^j \leftarrow \pi_1(M_3^j)$	$M_4^j \| C_5^j \leftarrow \pi_2^{-1}(C_3^j)$
$C_5^j \| C_2^j \leftarrow \mathsf{mix}(M_5^j \| M_2^j)$	$M_5^j \| M_2^j \leftarrow \mathsf{mix}(C_5^j \| C_2^j)$;
if $M_4^j \| C_5^j \notin \mathrm{domain}(\pi_2)$ **then**	**if** $M_4^j \| M_5^j \notin \mathrm{range}(\pi_1)$ **then**
$Y_2 \xleftarrow{\$} \{0,1\}^n$	$X_1 \xleftarrow{\$} \{0,1\}^n$
if $Y_2 \in \mathrm{range}(\pi_2)$ **then**	**if** $X_1 \in \mathrm{domain}(\pi_1)$ **then**
$bad \leftarrow \mathsf{true};\ \left[\, Y_2 \xleftarrow{\$} \mathrm{corange}(\pi_2) \,\right]$	$bad \leftarrow \mathsf{true};\ \left[\, X_1 \xleftarrow{\$} \mathrm{codomain}(\pi_1) \,\right]$
$\pi_2(M_4^j \| C_5^j) \leftarrow Y_2$	$\pi_1^{-1}(M_4^j \| M_5^j) \leftarrow X_1$
$C_3^j \leftarrow \pi_2(M_4^j \| C_5^j)$	$M_3^j \leftarrow \pi_1^{-1}(M_4^j \| M_5^j)$
$C_1^j \leftarrow C_3^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$	$M_1^j \leftarrow M_3^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
$C^j \leftarrow C_1^j \| C_2^j$	$M^j \leftarrow M_1^j \| M_2^j$ [Game G_2]
return C^j	**return** M^j Game G_3

procedure $E(M)$	**procedure** $D(C)$
$j \leftarrow j+1;\ \ M^j \leftarrow M$	$j \leftarrow j+1;\ \ C^j \leftarrow C$
$M_1^j \| M_2^j \leftarrow M^j$	$C_1^j \| C_2^j \leftarrow C^j$
$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$	$C_3^j \leftarrow C_1^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$
$Y_1 \xleftarrow{\$} \{0,1\}^n$	$X_2 \xleftarrow{\$} \{0,1\}^n$
if $M_3^j \in \mathrm{domain}(\pi_1)$ **then**	**if** $C_3^j \in \mathrm{range}(\pi_2)$ **then**
$bad \leftarrow \mathsf{true};\ \left[\, Y_1 \leftarrow \pi_1(M_3^j) \,\right]$ **else**	$bad \leftarrow \mathsf{true};\ \left[\, X_2 \leftarrow \pi_2^{-1}(C_3^j) \,\right]$ **else**
$\pi_1(M_3^j) \leftarrow Y_1$	$\pi_2^{-1}(C_3^j) \leftarrow X_2$
$M_4^j \| M_5^j \leftarrow \pi_1(M_3^j)$	$M_4^j \| C_5^j \leftarrow \pi_2^{-1}(C_3^j)$
$C_5^j \| C_2^j \leftarrow \mathsf{mix}(M_5^j \| M_2^j)$	$M_5^j \| M_2^j \leftarrow \mathsf{mix}(C_5^j \| C_2^j)$
$Y_2 \xleftarrow{\$} \{0,1\}^n$	$X_1 \xleftarrow{\$} \{0,1\}^n$
if $M_4^j \| C_5^j \in \mathrm{domain}(\pi_2)$ **then**	**if** $M_4^j \| M_5^j \in \mathrm{range}(\pi_1)$ **then**
$bad \leftarrow \mathsf{true};\ \left[\, Y_2 \leftarrow \pi_2(M_4^j \| C_5^j) \,\right]$ **else**	$bad \leftarrow \mathsf{true};\ \left[\, X_1 \leftarrow \pi_1^{-1}(M_4^j \| C_5^j) \,\right]$ **else**
$\pi_2(M_4^j \| C_5^j) \leftarrow Y_2$	$\pi_1^{-1}(M_4^j \| C_5^j) \leftarrow X_1$
$C_3^j \leftarrow \pi_2(M_4^j \| C_5^j)$	$M_3^j \leftarrow \pi_1^{-1}(M_4^j \| M_5^j)$
$C_1^j \leftarrow C_3^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$	$M_1^j \leftarrow M_3^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
$C^j \leftarrow C_1^j \| C_2^j$	$M^j \leftarrow M_1^j \| M_2^j$ [Game G_4]
return C^j	**return** M^j Game G_5

procedure $E(M)$	**procedure** $D(C)$ Game G_6
$j \leftarrow j+1;\ \ M^j \leftarrow M$	$j \leftarrow j+1;\ \ C^j \leftarrow C$
$C^j \xleftarrow{\$} \{0,1\}^{n+s}$; **return** C^j	$M^j \xleftarrow{\$} \{0,1\}^{n+s}$; **return** M^j
procedure Finalize	**procedure Finalize**
$M_1^j \| M_2^j \leftarrow M^j$	$C_1^j \| C_2^j \leftarrow C^j$
$C_1^j \| C_2^j \leftarrow C^j$	$M_1^j \| M_2^j \leftarrow M^j$
$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$	$C_3^j \leftarrow C_1^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$
$M_4^j \xleftarrow{\$} \{0,1\}^{n-s}$	$M_4^j \xleftarrow{\$} \{0,1\}^{n-s}$
$M_5^j \leftarrow \mathsf{mix}_{\mathsf{R}}^{-1}(C_2^j \| M_2^j)$	$C_5^j \leftarrow \mathsf{mix}_{\mathsf{R}}^{-1}(M_2^j \| C_2^j)$
$C_5^j \leftarrow \mathsf{mix}_{\mathsf{L}}(M_5^j \| M_2^j)$	$M_5^j \leftarrow \mathsf{mix}_{\mathsf{L}}(C_5^j \| C_2^j)$
$C_3^j \leftarrow C_1^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$	$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
$bad \leftarrow (M_3^j = M_3^i)$ **or**	$bad \leftarrow (C_3^j = C_3^i)$ **or**
$(M_4^j \| C_5^j = M_4^i \| C_5^i)$, for some $i < j$	$(M_4^j \| M_5^j = M_4^i \| M_5^i)$, for some $i < j$

Fig. 2. Games used in the proof of Theorem 1. Game G_2 includes the bracketed statements while game G_3 does not. Similarly, game G_4 includes the bracketed statements while game G_5 does not. In game G_6, encipher and decipher queries are answered by random values.

ference between p and $\Pr[\,G_6^{\mathcal{A}} \Rightarrow 1\,]$ is at most $0.5\,q^2/2^{n+s}$, due to the PRP/PRF switching lemma. We must bound $\Pr[\,G_1^{\mathcal{A}} \Rightarrow 1\,] - \Pr[\,G_6^{\mathcal{A}} \Rightarrow 1\,]$.

Game G_2 rewrites G_1 using lazy sampling [2] and these two games are adversarially indistinguishable. The probability that *bad* gets set to true in G_3 is bounded by the PRP/PRF switching lemma; $\Pr[\,G_2^{\mathcal{A}} \Rightarrow 1\,] - \Pr[\,G_3^{\mathcal{A}} \Rightarrow 1\,] \leq 2 \times 0.5\,q^2/2^n$. Game G_4 makes several trivial modifications compared to G_3; they are adversarially indistinguishable. $\Pr[\,G_4^{\mathcal{A}} \Rightarrow 1\,] - \Pr[\,G_5^{\mathcal{A}} \Rightarrow 1\,]$ is at most the probability that $\mathcal{A}$ manages to set *bad* in game G_5. Game G_6 simply changes the order of many random choices; it is adversarially indistinguishable from game G_5. In game G_6, the encipher and decipher queries are answered by random values over $\{0,1\}^{n+s}$. It remains to bound the probability that *bad* gets set to true in this game.

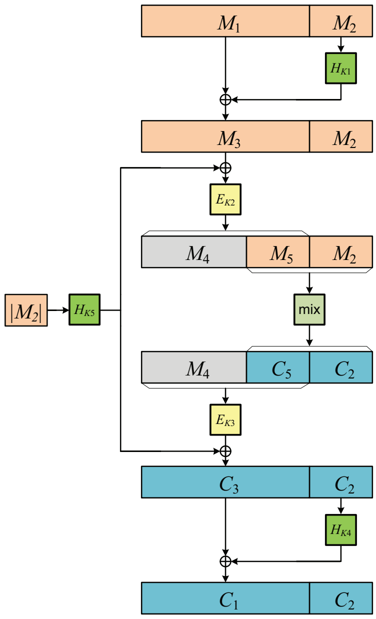
In game G_6, we let $\mathsf{mix}_{\mathsf{R}}^{-1}(\cdot, B)$ denote the inverse of $\mathsf{mix}_{\mathsf{R}}(\cdot, B)$. We first analyze the circumstance where the j-th query is an encipher query. If the j-th and i-th queries M^j and M^i are both encipher queries and $i < j$, then $M^j \neq M^i$ since encipher queries are not repeated. If $M_2^j \neq M_2^i$ then $\mathsf{pad}(M_2^j) \neq \mathsf{pad}(M_2^i)$. By the definition of AXU hash function the probability that $M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j)) = M_1^i \oplus H_{K_1}(\mathsf{pad}(M_2^i))$ is at most 2^{-n}. Otherwise, we have that $M_2^j = M_2^i$ and $M_1^j \neq M_1^i$, and the probability that $M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j)) = M_1^i \oplus H_{K_1}(\mathsf{pad}(M_2^i))$ is zero. Thus we have that the probability that $M_3^j = M_3^i$ is at most 2^{-n}. If the j-th query is an encipher query and the i-th query is a decipher query, then we still have $M^j \neq M^i$ because $\mathcal{A}$ never makes an encipher query of a value that it earlier received from a decipher query. Similarly, in this case, the probability that $M_3^j = M_3^i$ is at most 2^{-n}.

On the other hand, we claim that the probability that $M_4^j||C_5^j = M_4^i||C_5^i$ (for some $i < j$) is at most $\frac{\epsilon(s)}{2^{n-s}}$. This can be justified as follows. First, M_4^j is freshly chosen at random and thus the probability that $M_4^j = M_4^i$ is 2^{s-n}. Second, by $M_5^j = \mathsf{mix}_{\mathsf{R}}^{-1}(C_2^j||M_2^j)$, we have that M_5^j is uniformly distributed since C_2^j is independently chosen at random. By the definition of $\epsilon(s)$-good mixing function and by $C_5^j = \mathsf{mix}_{\mathsf{L}}(M_5^j||M_2^j)$, we have that the probability that $C_5^j = C_5^i$ is at most $\epsilon(s)$. Therefore, the probability that $M_4^j||C_5^j$ equals $M_4^i||C_5^i$ (for some $i < j$) is at most $\frac{\epsilon(s)}{2^{n-s}}$.

The same probability results hold for the case where the j-th query is a decipher query with a proof symmetric to the above one. Since there are at most $q^2/2$ possible collisions at both Line 620/670 and Line 621/671, the probability that $\mathcal{A}$ manages to set *bad* in this game is at most $0.5\,q^2/2^n + 0.5\,q^2\frac{\epsilon(s)}{2^{n-s}}$. This completes the proof of the claim. ▮

It is easy to pass from the information-theoretic setting to complexity-theoretic one.

Insecurity of FHEM against VIL adversaries. FHEM is designed against FIL adversaries. It is not a PRP secure cipher with respect to VIL attacks. A simple attack is illustrated as follows. The adversary simply makes two encipher queries 0^{n+1} and 0^{n+2}, and gets two replies from the oracle $C_1||C_2$ and $C_1'||C_2'$



algorithm $\mathcal{E}_K(M)$ **where** $K = K_1||K_2||K_3||K_4||K_5$
if $M \notin \bigcup_{i=n+1}^{2n-1} \{0,1\}^i$ **then return** $\perp$
$M_1||M_2 \leftarrow M$ **where** $|M_1| = n$ and $|M_2| = s$
$M_3 \leftarrow M_1 \oplus H_{K_1}(\mathsf{pad}(M_2))$
$M_4||M_5 \leftarrow E_{K_2}(M_3 \oplus H_{K_5}(\mathsf{pad}(|M_2|)))$ **where** $|M_4| = n - s$ and $|M_5| = s$
$C_5||C_2 \leftarrow \mathsf{mix}(M_5||M_2)$ **where** $|C_5| = |C_2| = s$
$C_3 \leftarrow E_{K_3}(M_4||C_5) \oplus H_{K_5}(\mathsf{pad}(|M_2|))$
$C_1 \leftarrow C_3 \oplus H_{K_4}(\mathsf{pad}(C_2))$
$C \leftarrow C_1||C_2$
return C

Fig. 3. Mode HEM. The input for the AXU hash functions H_{K_1}, H_{K_4}, and H_{K_5} should be all padded to a complete block. In particular, the first $\log n$ bits of input for H_{K_5} is the length encoding of the partial block M_2, while the remaining are $n - \log n$ zero-bits.

where $|C_1| = |C'_1| = n$. If $C_1 = C'_1$, the adversary returns 1; otherwise, it returns 0. If the adversary is given a FHEM oracle, one can check that the probability that $C_1 = C'_1$ is quite high; otherwise, it is about 2^{-n}. Such an adversary can thus attack the PRP security of FHEM.

4 A Length-Doubling VIL Cipher

We now show how to make a VIL cipher based on the FIL one in the above section. The basic idea is to replace the AXU hash function with a length related AXU hash function. Namely, we want a primitive that enjoys the AXU hash function property even for variable length input. We do not design such a primitive from scratch. Instead, this can be achieved by applying the same AXU hash function (with an independently and uniformly chosen key) to the length of incomplete block of the input.

Let $E\colon \mathcal{K}_1 \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and let $H\colon \mathcal{K}_2 \times \{0,1\}^n \to \{0,1\}^n$ be a 2^{-n}-AXU hash function family. Let $\mathsf{mix}\colon \mathcal{S}^2 \to \mathcal{S}^2$ ($\mathcal{S} \supseteq \bigcup_{i=1}^{n-1}\{0,1\}^i$) be an $\epsilon(s)$-good mixing function. From these building blocks we define a cipher $\mathcal{E} = \mathrm{HEM}[H, E, \mathsf{mix}]$ with key space $\mathcal{K}_1^2 \times \mathcal{K}_2^3$ and message space $\mathcal{M} = \bigcup_{i=n+1}^{2n-1}\{0,1\}^i$. See Figure 3. The construction is $\pm$prp-secure for VIL adversaries. The AXU-hash function H_{K_5} can be replaced with a blockcipher E_{K_5} and the security remains.

We emphasize that the AXU hash function H_{K_5} taking as input the length of incomplete block can be precomputed. It only needs n (typically, 128) invocations of hash function calls (or blockcipher calls). This thus yields a highly efficient implementation of HEM with a few preprocessed operations and a little additional storage.

To make this cipher also secure for queries of length n, one can choose an independent blockcipher to encipher all n-bit messages. The complexity assumption used is SPRP. We give the security analysis for the scheme with message space $\bigcup_{i=n+1}^{2n-1}\{0,1\}^i$, using AXU hash function H_{K_5}.

Theorem 2. *Let* $\mathcal{E} = \mathrm{HEM}[H, \mathrm{Perm}(n), \mathsf{mix}]$. *If* $\mathcal{A}$ *asks at most* q *queries then* $\mathbf{Adv}^{\pm\mathrm{prp}}_{\mathcal{E}}(\mathcal{A}) \le 2\,q^2/2^n + 0.5\,q^2\,\frac{\epsilon(s)}{2^{n-s}}$, *and if we use a* 2^{1-s}*-good mixing function then we have that* $\mathbf{Adv}^{\pm\mathrm{prp}}_{\mathcal{E}}(\mathcal{A}) \le 3\,q^2/2^n$. ∎

Proof. The proof follows an analogous line to the one of Theorem 1. We begin with game G_1, which precisely describes the HEM construction with the ideal blockcipher π_1 and π_2. Game G_6 simulates a pair of random functions. We let p' denote the probability that $\mathcal{A}$ outputs 1 in the game simulating a random permutation and its inverse. The difference between p' and $\Pr[\,G_6^{\mathcal{A}} \Rightarrow 1\,]$ is at most $0.5\,q^2/2^n$, again due to the PRP/PRF switching lemma. We have to bound $\Pr[\,G_1^{\mathcal{A}} \Rightarrow 1\,] - \Pr[\,G_6^{\mathcal{A}} \Rightarrow 1\,]$.

Game G_2 modifies G_1 using lazy sampling and they are adversarially indistinguishable. By the PRP/PRF switching lemma, $\Pr[\,G_2^{\mathcal{A}} \Rightarrow 1\,] - \Pr[\,G_3^{\mathcal{A}} \Rightarrow 1\,] \le 2 \times 0.5\,q^2/2^n$. Game G_4 and G_3 are easily seen to be adversarially indistinguishable. $\Pr[\,G_4^{\mathcal{A}} \Rightarrow 1\,] - \Pr[\,G_5^{\mathcal{A}} \Rightarrow 1\,]$ is at most the probability that $\mathcal{A}$ can set *bad* in game G_5. We delay the calculation of the probability in an adversarially indistinguishable game G_6 where the encipher and decipher queries are answered by random values from $\bigcup_{i=n+1}^{2n-1}\{0,1\}^i$.

It remains to bound the probability that $\mathcal{A}$ can set *bad* in game G_6. We only give the analysis for the circumstance where the j-th query is an encipher query. (The case where j-th query is a decipher query is symmetric.)

Consider the j-th and i-th queries M^j and M^i (where $i < j$). We have that $M^j \neq M^i$ since encipher queries are not repeated and $\mathcal{A}$ never makes an encipher

procedure $E(M)$
$j \leftarrow j+1;\ M^j \leftarrow M$
$M_1^j \| M_2^j \leftarrow M^j$
$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(|M_2^j|))$
$M_4^j \| M_5^j \leftarrow \pi_1(M_3^j \oplus H_{K_5}(\mathsf{pad}|M_2^j|))$
$C_5^j \| C_2^j \leftarrow \mathsf{mix}(M_5^j \| M_2^j)$
$C_3^j \leftarrow \pi_2(M_4^j \| C_5^j) \oplus H_{K_5}(\mathsf{pad}(|M_2^j|))$
$C_1^j \leftarrow C_3^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$
$C^j \leftarrow C_1^j \| C_2^j$
return C^j

procedure $D(C)$
$j \leftarrow j+1;\ C^j \leftarrow C$
$C_1^j \| C_2^j \leftarrow C^j$
$C_3^j \leftarrow C_1^j \oplus H_{K_1}(\mathsf{pad}(C_2^j))$
$M_4^j \| C_5^j \leftarrow \pi_2^{-1}(C_3^j \oplus H_{K_5}(\mathsf{pad}(|C_2^j|)))$
$M_5^j \| M_2^j \leftarrow \mathsf{mix}(C_5^j \| C_2^j)$
$M_3^j \leftarrow \pi_1^{-1}(M_4^j \| M_5^j) \oplus H_{K_5}(\mathsf{pad}(|C_2^j|))$
$M_1^j \leftarrow M_3^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
$M^j \leftarrow M_1^j \| M_2^j$
return M^j

Game G_1

---

procedure $E(M)$
$j \leftarrow j+1;\ M^j \leftarrow M$
$M_1^j \| M_2^j \leftarrow M^j$
$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
if $M_3^j \oplus H_{K_5}(\mathsf{pad}(|M_2^j|)) \notin \mathrm{domain}(\pi_1)$ **then**
  $Y_1 \xleftarrow{\$} \{0,1\}^n$
  if $Y_1 \in \mathrm{range}(\pi_1)$ **then**
    $bad \leftarrow \mathsf{true};\ \left[\, Y_1 \xleftarrow{\$} \mathrm{corange}(\pi_1) \,\right]$
  $\pi_1(M_3^j \oplus H_{K_5}(\mathsf{pad}(|M_2^j|))) \leftarrow Y_1$
$M_4^j \| M_5^j \leftarrow \pi_1(M_3^j \oplus H_{K_5}(\mathsf{pad}(|M_2^j|)))$
$C_5^j \| C_2^j \leftarrow \mathsf{mix}(M_5^j \| M_2^j)$
if $M_4^j \| C_5^j \notin \mathrm{domain}(\pi_2)$ **then**
  $Y_2 \xleftarrow{\$} \{0,1\}^n$
  if $Y_2 \in \mathrm{range}(\pi_2)$ **then**
    $bad \leftarrow \mathsf{true};\ \left[\, Y_2 \xleftarrow{\$} \mathrm{corange}(\pi_2) \,\right]$
  $\pi_2(M_4^j \| C_5^j) \leftarrow Y_2$
$C_3^j \leftarrow \pi_2(M_4^j \| C_5^j) \oplus H_{K_5}(\mathsf{pad}(|M_2^j|))$
$C_1^j \leftarrow C_3^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$
$C^j \leftarrow C_1^j \| C_2^j$
return C^j

procedure $D(C)$
$j \leftarrow j+1;\ C^j \leftarrow C$
$C_1^j \| C_2^j \leftarrow C^j$
$C_3^j \leftarrow C_1^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$
if $C_3^j \oplus H_{K_5}(\mathsf{pad}(|C_2^j|)) \notin \mathrm{range}(\pi_2)$ **then**
  $X_2 \xleftarrow{\$} \{0,1\}^n$
  if $X_2 \in \mathrm{domain}(\pi_2)$ **then**
    $bad \leftarrow \mathsf{true};\ \left[\, X_2 \xleftarrow{\$} \mathrm{codomain}(\pi_2) \,\right]$
  $\pi_2^{-1}(C_3^j \oplus H_{K_5}(\mathsf{pad}(|C_2^j|))) \leftarrow X_2$
$M_4^j \| C_5^j \leftarrow \pi_2^{-1}(C_3^j \oplus H_{K_5}(\mathsf{pad}(|C_2^j|)))$
$M_5^j \| M_2^j \leftarrow \mathsf{mix}(C_5^j \| C_2^j)$
if $M_4^j \| M_5^j \notin \mathrm{range}(\pi_1)$ **then**
  $X_1 \xleftarrow{\$} \{0,1\}^n$
  if $X_1 \in \mathrm{domain}(\pi_1)$ **then**
    $bad \leftarrow \mathsf{true};\ \left[\, X_1 \xleftarrow{\$} \mathrm{codomain}(\pi_1) \,\right]$
  $\pi_1^{-1}(M_4^j \| M_5^j) \leftarrow X_1$
$M_3^j \leftarrow \pi_1^{-1}(M_4^j \| M_5^j) \oplus H_{K_5}(\mathsf{pad}(|C_2^j|))$
$M_1^j \leftarrow M_3^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
$M^j \leftarrow M_1^j \| M_2^j$
return M^j

[Game G_2]
Game G_3

---

procedure $E(M)$
$j \leftarrow j+1;\ M^j \leftarrow M$
$M_1^j \| M_2^j \leftarrow M^j$
$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
$Y_1 \xleftarrow{\$} \{0,1\}^n$
if $M_3^j \oplus H_{K_5}(\mathsf{pad}(|M_2^j|)) \in \mathrm{domain}(\pi_1)$ **then**
  $bad \leftarrow \mathsf{true};$
  $\left[\, Y_1 \leftarrow \pi_1(M_3^j \oplus H_{K_5}(\mathsf{pad}(|M_2^j|))) \,\right]$ **else**
$\pi_1(M_3^j \oplus H_{K_5}(\mathsf{pad}(|M_2^j|))) \leftarrow Y_1$
$M_4^j \| M_5^j \leftarrow \pi_1(M_3^j)$
$C_5^j \| C_2^j \leftarrow \mathsf{mix}(M_5^j \| M_2^j)$
$Y_2 \xleftarrow{\$} \{0,1\}^n$
if $M_4^j \| C_5^j \in \mathrm{domain}(\pi_2)$ **then**
  $bad \leftarrow \mathsf{true};\ \left[\, Y_2 \leftarrow \pi_2(M_4^j \| C_5^j) \,\right]$ **else**
$\pi_2(M_4^j \| C_5^j) \leftarrow Y_2$
$C_3^j \leftarrow \pi_2(M_4^j \| C_5^j) \oplus H_{K_5}(\mathsf{pad}(|M_2^j|))$
$C_1^j \leftarrow C_3^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$
$C^j \leftarrow C_1^j \| C_2^j$
return C^j

procedure $D(C)$
$j \leftarrow j+1;\ C^j \leftarrow C$
$C_1^j \| C_2^j \leftarrow C^j$
$C_3^j \leftarrow C_1^j \oplus H_{K_4}(\mathsf{pad}(C_2^j))$
$X_2 \xleftarrow{\$} \{0,1\}^n$
if $C_3^j \oplus H_{K_5}(\mathsf{pad}(|C_2^j|)) \in \mathrm{range}(\pi_2)$ **then**
  $bad \leftarrow \mathsf{true};$
  $\left[\, X_2 \leftarrow \pi_2^{-1}(C_3^j \oplus H_{K_5}(\mathsf{pad}(|C_2^j|))) \,\right]$ **else**
$\pi_2^{-1}(C_3^j \oplus H_{K_5}(\mathsf{pad}(|C_2^j|))) \leftarrow X_2$
$M_4^j \| C_5^j \leftarrow \pi_2^{-1}(C_3^j)$
$M_5^j \| M_2^j \leftarrow \mathsf{mix}(C_5^j \| C_2^j)$
$X_1 \xleftarrow{\$} \{0,1\}^n$
if $M_4^j \| M_5^j \in \mathrm{range}(\pi_1)$ **then**
  $bad \leftarrow \mathsf{true};\ \left[\, X_1 \leftarrow \pi_1^{-1}(M_4^j \| M_5^j) \,\right]$ **else**
$\pi_1^{-1}(M_4^j \| M_5^j) \leftarrow X_1$
$M_3^j \leftarrow \pi_1^{-1}(M_4^j \| M_5^j) \oplus H_{K_5}(\mathsf{pad}(|C_2^j|))$
$M_1^j \leftarrow M_3^j \oplus H_{K_1}(\mathsf{pad}(M_2^j))$
$M^j \leftarrow M_1^j \| M_2^j$
return M^j

[Game G_4]
Game G_5

---

procedure $E(M)$
$j \leftarrow j+1;\ M^j \leftarrow M$
$C^j \xleftarrow{\$} \{0,1\}^{n+s}$; **return** C^j

procedure Finalize
$M_1^j \| M_2^j \leftarrow M^j$
$C_1^j \| C_2^j \leftarrow C^j$
$X^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j)) \oplus H_{K_5}(\mathsf{pad}(|M_2^j|))$
$M_4^j \xleftarrow{\$} \{0,1\}^{n-s}$
$M_5^j \leftarrow \mathsf{mix}_{\mathsf{R}}^{-1}(C_2^j \| M_2^j)$
$C_5^j \leftarrow \mathsf{mix}_{\mathsf{L}}(M_5^j \| M_2^j)$
$C_3^j \leftarrow C_1^j \oplus H_{K_4}(\mathsf{pad}(C_2^j)) \oplus H_{K_5}(\mathsf{pad}(|M_2^j|))$
$bad \leftarrow (X^j = X^i)$ **or**
  $(M_4^j \| C_5^j = M_4^i \| C_5^i)$, for some $i < j$

procedure $D(C)$
$j \leftarrow j+1;\ C^j \leftarrow C$
$M^j \xleftarrow{\$} \{0,1\}^{n+s}$; **return** M^j

procedure Finalize
$C_1^j \| C_2^j \leftarrow C^j$
$M_1^j \| M_2^j \leftarrow M^j$
$Y_j \leftarrow C_1^j \oplus H_{K_4}(\mathsf{pad}(C_2^j)) \oplus H_{K_5}(\mathsf{pad}(|C_2^j|))$
$M_4^j \xleftarrow{\$} \{0,1\}^{n-s}$
$C_5^j \leftarrow \mathsf{mix}_{\mathsf{R}}^{-1}(M_2^j \| C_2^j)$
$M_5^j \leftarrow \mathsf{mix}_{\mathsf{L}}(C_5^j \| C_2^j)$
$M_3^j \leftarrow M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j)) \oplus H_{K_5}(\mathsf{pad}(|C_2^j|))$
$bad \leftarrow (Y^j = Y^i)$ **or**
  $(M_4^j \| M_5^j = M_4^i \| M_5^i)$, for some $i < j$

Game G_6

Fig. 4. Games used in the proof of Theorem 2. In game G_6, encipher and decipher queries are answered by random values.

query of a value that it earlier received from a decipher query. We have several cases to consider:

- If $|M_2^j| \neq |M_2^i|$ then by the definition of AXU hash function (for H_{K_5}) the probability that $M_1^j \oplus H_{K_1}(\mathsf{pad}(M_2^j)) \oplus H_{K_5}(\mathsf{pad}(|M_2^j|)) = M_1^i \oplus H_{K_1}(\mathsf{pad}(M_2^i)) \oplus H_{K_5}(\mathsf{pad}(|M_2^i|))$ is at most 2^{-n}. In other words, we have that $\Pr[\, X^j = X^i \,] \leq 2^{-n}$.
- If $|M_2^j| = |M_2^i|$ and $M_2^j \neq M_2^i$ then again by the definition of AXU hash functions (for H_{K_1}) we have $\Pr[\, X^j = X^i \,] \leq 2^{-n}$.
- If $|M_2^j| = |M_2^i|$ and $|M_2^j| = |M_2^i|$ then we immediately have that $M_1^j \neq M_1^i$. The probability that X^j equals X^i is zero.

In any case, we have $\Pr[\, X^j = X^i \,] \leq 2^{-n}$.

We now bound the probability that $M_4^j || C_5^j = M_4^i || C_5^i$ (for some $i < j$). M_4^j is freshly chosen at random, and the probability that $M_4^j = M_4^i$ is 2^{s-n}. Moreover, we have $M_5^j = \mathsf{mix}_\mathsf{R}^{-1}(C_2^j || M_2^j)$, and thus M_5^j is uniformly distributed since C_2^j is independently chosen at random. We also have $C_5^j = \mathsf{mix}_\mathsf{L}(M_5^j || M_2^j)$; by the definition of mixing function we have that the probability that $C_5^j = C_5^i$ is at most $\epsilon(s)$. The probability that $M_4^j || C_5^j = M_4^i || C_5^i$ (for some $i < j$) is at most $\frac{\epsilon(s)}{2^{n-s}}$.

There are at most $q^2/2$ pairs of possible collisions at both Line 620/670 and Line 621/671, the probability that $\mathcal{A}$ manages to set *bad* in this game is at most $0.5\, q^2/2^n + 0.5\, q^2 \frac{\epsilon(s)}{2^{n-s}}$. The theorem now follows. ▮

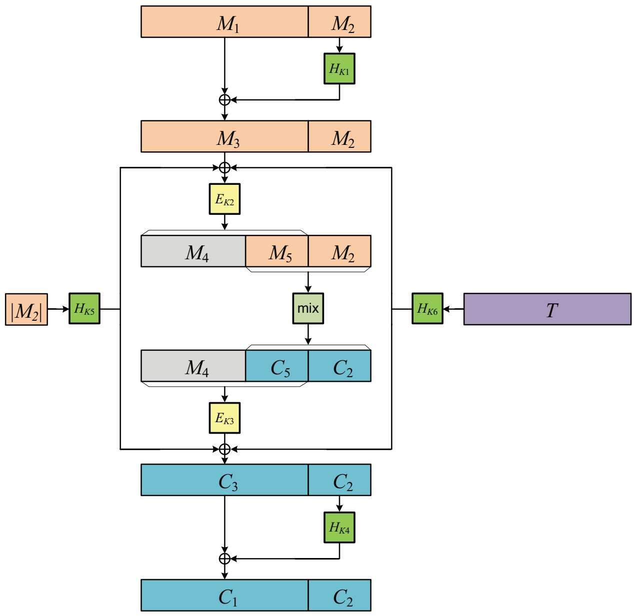
It is straightforward to pass from the information-theoretic setting to complexity-theoretic one. For completeness, we show it as follows.

Corollary 1. *Let E be a blockcipher, let H be a 2^{-n}-AXU hash function family, and let mix be an $\epsilon(s)$-good mixing function. Let $\mathcal{E} = \mathrm{HEM}[H, E, \mathsf{mix}]$ and let $\mathcal{A}$ be an adversary that asks at most q queries. Then there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that $\mathbf{Adv}_{\mathcal{E}}^{\pm\mathrm{prp}}(\mathcal{A}) \leq \mathbf{Adv}_{E}^{\pm\mathrm{prp}}(\mathcal{B}) + \mathbf{Adv}_{E}^{\pm\mathrm{prp}}(\mathcal{C}) + 2\, q^2/2^n + 0.5\, q^2 \frac{\epsilon(s)}{2^{n-s}}$ for any $s \in [n-1]$. Specifically, if we use a mixing function with $\epsilon = 2^{1-s}$ then we have $\mathbf{Adv}_{\mathcal{E}}^{\pm\mathrm{prp}}(\mathcal{A}) \leq \mathbf{Adv}_{E}^{\pm\mathrm{prp}}(\mathcal{B}) + \mathbf{Adv}_{E}^{\pm\mathrm{prp}}(\mathcal{C}) + 3\, q^2/2^n$.* ▮

5 Length-Doubling VIL Tweakable Ciphers

In this section, we present a VIL tweakable cipher over $\bigcup_{i=n+1}^{2n-1} \{0,1\}^i$ with tweak space $\{0,1\}^n$. It is easy to modify the scheme to support larger tweak space. We also give a variant with a slightly more succinct structure.

Let $E \colon \mathcal{K}_1 \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher and let $H \colon \mathcal{K}_2 \times \{0,1\}^n \to \{0,1\}^n$ be a 2^{-n}-AXU hash function family. Let $\mathsf{mix} \colon \mathcal{S}^2 \to \mathcal{S}^2$ ($\mathcal{S} \supseteq \bigcup_{i=1}^{n-1} \{0,1\}^i$) be an $\epsilon(s)$-good mixing function. Define from the above primitives a VIL tweakable cipher $\widetilde{\mathcal{E}} = \mathrm{THEM}[H, E, \mathsf{mix}]$ with key space $\mathcal{K}_1^2 \times \mathcal{K}_2^4$ and tweak space $\mathcal{T} = \{0,1\}^n$ and message space $\mathcal{M} = \bigcup_{i=n+1}^{2n-1} \{0,1\}^i$. See Figure 5. The construction is $\pm\widetilde{\mathrm{prp}}$-secure for VIL adversaries. To extend the domain of THEM to in-



$\textbf{algorithm } \widetilde{\mathcal{E}}_K^T(M) \quad \textbf{where } K = K_1||K_2||K_3||K_4||K_5||K_6$

$\textbf{if } M \notin \bigcup_{i=n+1}^{2n-1} \{0,1\}^i \textbf{ then return } \perp$

$M_1||M_2 \leftarrow M \quad \textbf{where } |M_1| = n \text{ and } |M_2| = s$

$M_3 \leftarrow M_1 \oplus H_{K_1}(\mathsf{pad}(M_2))$

$M_4||M_5 \leftarrow E_{K_2}(M_3 \oplus H_{K_5}(\mathsf{pad}(|M_2|)) \oplus H_{K_6}(T)) \quad \textbf{where } |M_4| = n-s,\ |M_5| = s$

$C_5||C_2 \leftarrow \mathsf{mix}(M_5||M_2) \quad \textbf{where } |C_5| = |C_2| = s$

$C_3 \leftarrow E_{K_3}(M_4||C_5) \oplus H_{K_5}(\mathsf{pad}|M_2|) \oplus H_{K_6}(T)$

$C_1 \leftarrow C_3 \oplus H_{K_4}(\mathsf{pad}(C_2))$

$C \leftarrow C_1||C_2$

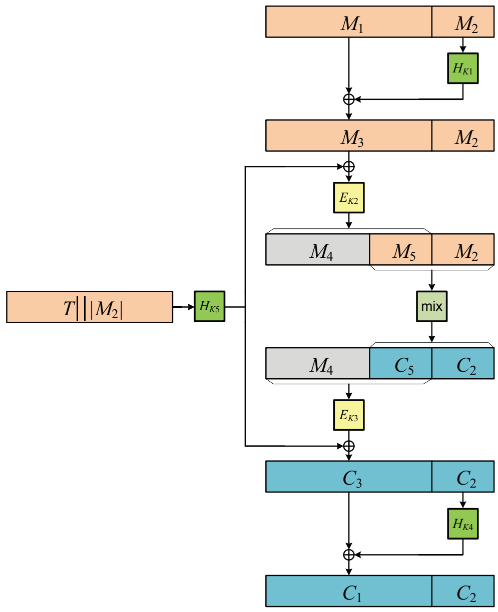
$\textbf{return } C$

Fig. 5. Mode THEM. Compared to HEM mode, FHEM takes an additional tweak T as input. For simplicity, we can assume that the tweak space is $\{0,1\}^n$. Of course, it is trivial to handle larger tweak space by selecting an AXU hash function that supports longer input.

clude $\{0,1\}^n$, we can choose an independent tweakable blockcipher to encipher the n-bit messages. The following theorem establishes the security of THEM.

Theorem 3. *Let $\widetilde{\mathcal{E}} = \mathrm{THEM}[H, \mathrm{Perm}(n), \mathsf{mix}]$. If $\mathcal{A}$ asks at most q queries then $\mathbf{Adv}_{\widetilde{\mathcal{E}}}^{\pm\widetilde{\mathrm{prp}}}(\mathcal{A}) \le 2\,q^2/2^n + 0.5\,q^2 \frac{\epsilon(s)}{2^{n-s}}$, and if we use a 2^{1-s}-good mixing function then we have that $\mathbf{Adv}_{\widetilde{\mathcal{E}}}^{\pm\widetilde{\mathrm{prp}}}(\mathcal{A}) \le 3\,q^2/2^n$.* ∎

The proof of the above theorem largely resembles the previous ones, and is thus omitted.



algorithm $\widetilde{\mathcal{E}}_K^T(M)$ **where** $K = K_1||K_2||K_3||K_4||K_5$
  if $M \notin \bigcup_{i=n+1}^{2n-1} \{0,1\}^i$ **then return** $\perp$
  $M_1||M_2 \leftarrow M$ **where** $|M_1| = n$ and $|M_2| = s$
  $M_3 \leftarrow M_1 \oplus H_{K_1}(\mathsf{pad}(M_2))$
  $M_4||M_5 \leftarrow E_{K_2}(M_3 \oplus H_{K_5}(T\ |M_2|))$ **where** $|M_4| = n - s$ and $|M_5| = s$
  $C_5||C_2 \leftarrow \mathsf{mix}(M_5||M_2)$ **where** $|C_5| = |C_2| = s$
  $C_3 \leftarrow E_{K_3}(M_4||C_5) \oplus H_{K_5}(T\ |M_2|)$
  $C_1 \leftarrow C_3 \oplus H_{K_4}(\mathsf{pad}(C_2))$
  $C \leftarrow C_1||C_2$
  return C

Fig. 6. An Alternative Mode—"Tweak Stealing". This mode is specified to support tweak space $\mathcal{T} = \{0,1\}^{n-\log n}$. The input for AXU hash function H_{K_5} is a tweak $T \in \mathcal{T}$ concatenating $\log n$ bits encoding of the length of partial input M_2.

An Alternative Design–Tweak Stealing. A more compact variant using the idea of "tweak stealing" is depicted in Figure 6. This algorithm causes a small decrease in tweak space to $\{0,1\}^{n-\log n}$ (if we insist using a AXU hash function from $\{0,1\}^n$ to $\{0,1\}^n$ for the tweak input), and leads to a slight security loss. However, this does not necessarily restrict its usage. For instance, it suffices for constructing arbitrary-input-length online ciphers [24]: the *stolen* tweak does not impair the encipher and decipher algorithms. We comment that in spite of its structural simplicity, the variant does not seem to give a notable improvement of efficiency if we consider pre-computation.

Acknowledgments

The author would like to gratefully acknowledge the support of NSF grant CNS 0831547, CNS 0904380, and CNS 1228828.

References

1. M. Bellare, and P. Rogaway. On the construction of variable-input-length ciphers. *Fast Software Encryption 1999*, LNCS vol. 1636, Springer, pp. 231–244, 1999.
2. M. Bellare and P. Rogaway. Code-based game-playing proofs and the security of triple encryption. *EUROCRYPT 2006*, LNCS vol. 4004, Springer, pp. 409–426, 2006.
3. D. Chakraborty and P. Sarkar. HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. *INDOCRYPT'06*, LNCS vol. 4329, Springer, pp. 287–302, 2006.
4. D. Chakraborty and P. Sarkar. A new mode of encryption providing a tweakable strong pseudo-random permutation. *FSE'06*, LNCS vol. 4047, Springer, pp. 293–309, 2006.
5. D. Cook, M. Yung, A. Keromytis. Elastic block ciphers: method, security and instantiations. *Int. J. Inf. Sec.* 8(3): 211–231 (2009)
6. D. McGrew and S. Fluhrer. The security of the Extended Codebook (XCB) mode of operation. *Selected Areas in Cryptography 2007*, LNCS vol. 4876, Springer, pp. 311-327, 2007.
7. D. Goldenberg, S. Hohenberger, M. Liskov, E. Schwartz and H. Seyalioglu. On tweaking Luby-Rackoff blockciphers. *ASIACRYPT 2007*, LNCS vol. 4833, Springer, pp. 342-356, 2007.
8. S. Gueron. Intel's New AES instructions for enhanced performance and security. *FSE 2009, LNCS* vol. 5665, Springer, pp. 51-66, 2009.
9. S. Halevi. An observation regarding Jutla's modes of operation. Cryptology ePrint report 2001/015. April 2, 2001.
10. S. Halevi. EME*: extending EME to handle arbitrary-length messages with associated data. *INDOCRYPT 2004*, LNCS vol. 3348, Springer, pp. 315–327, 2004.
11. S. Halevi. Invertible universal hashing and the TET encryption Mode. *CRYPTO 07*, LNCS vol. 4622, Springer, pp. 412-429, 2007.
12. S. Halevi and P. Rogaway. A parallelizable enciphering mode. *CT-RSA 04*, LNCS vol. 2964, Springer, pp. 292–304, 2004.
13. S. Halevi and P. Rogaway. A tweakable enciphering mode. *CRYPTO 03*, LNCS vol. 2729, pp. 482–499, 2004.
14. H. Krawczyk. LFSR-based hashing and authentication. *CRYPTO 1994*, LNCS vol. 839, Springer, pp. 129–139, 1994.
15. T. Krovetz and P. Rogaway. The software performance of authenticated-encryption modes. *FSE 2011*, LNCS vol. 6733, Springer, pp. 306–327, 2011.
16. M. Liskov and K. Minematsu. Comments on XTS-AES. Sept 2008. See http://csrc.nist.gov/
17. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal of Computing*, vol. 17, no. 2, pp. 373–386, 1988.
18. M. Liskov, R. Rivest, and D. Wagner. Tweakable block ciphers. *CRYPTO 2002*, LNCS vol. 2442, Springer, pp. 31–46, 2002.

19. C. Meyer and M. Matyas. Cryptography: A new dimension in data security. *John Wiley & Sons*, New York, 1982.
20. M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, vol. 12, no. 1, pp. 29-66, 1999.
21. IEEE P 1619. IEEE Standard for Cryptographic Protection of Data on Block-Oriented Storage Devices.
22. S. Patel, Z. Ramzan, and G. Sundaram. Efficient constructions of variable-input-length block ciphers. *SAC 2004*, LNCS vol. 3357, pp. 326–340, 2005.
23. T. Ristenpart and P. Rogaway. How to enrich the message space of a cipher. *FSE 2007*, LNCS vol. 4593, pp. 101–118, 2007.
24. P. Rogaway and H. Zhang. Online ciphers from tweakable blockciphers. *CT-RSA 2011*, LNCS vol. 6558, Springer, pp. 237-249, 2011.
25. P. Rogaway, M. Wooding, and H. Zhang. The security of ciphertext stealing. *FSE 2012*, LNCS, Springer, 2012.
26. B. Schneier and J. Kelsey. Unbalanced Feistel networks and block cipher design. *Fast Software Encryption 1996*, LNCS vol. 1039, Springer, pp. 121–144, 1996.
27. P. Wang, D. Feng, and W. Wu. HCTR: A variable-input-length enciphering mode. *CISC 2005*, LNCS vol. 3822, Springer, pp. 175–188, 2005.
28. Y. Zheng, T. Matsumoto, and H. Imai. On the construction of block ciphers provably secure and not relying on any unproved hypotheses. *CRYPTO 1989*, LNCS vol. 435, Springer, pp. 461–480, 1990.