On the Construction of Variable-Input-Length Ciphers

Mihir Bellare[1] and Phillip Rogaway[2]

[1] Dept. of Computer Science & Engineering, University of California at San Diego, 9500 Gilman Drive, La Jolla, CA 92093 USA
mihir@cs.ucsd.edu www-cse.ucsd.edu/users/mihir/

[2] Dept. of Computer Science, Eng. II Bldg., One Shields Ave., University of California at Davis, Davis, CA 95616 USA
rogaway@cs.ucdavis.edu www.cs.ucdavis.edu/~rogaway/

Abstract. Whereas a block cipher enciphers messages of some one particular length (the blocklength), a variable-input-length cipher takes messages of varying (and preferably arbitrary) lengths. Still, the length of the ciphertext must equal the length of the plaintext. This paper introduces the problem of constructing such objects, and provides a practical solution. Our VIL mode of operation makes a variable-input-length cipher from any block cipher. The method is demonstrably secure in the provable-security sense of modern cryptography: we give a quantitative security analysis relating the difficulty of breaking the constructed (variable-input-length) cipher to the difficulty of breaking the underlying block cipher.

Keywords: Ciphers, Modes of Operation, Provable Security, Symmetric Encryption.

1 Introduction

This paper introduces the question of how to construct ciphers which operate on messages of varying lengths. Such a cipher, F, maps a key K and a plaintext M in $\{0,1\}$ (or M in some other set containing strings of various lengths) into a ciphertext $C = F_K(M)$ having the same length as M. Note that the length of M is not restricted to some xed blocklength n, or even to some multiple of a blocklength. At the same time, being a cipher, F_K is a length-preserving permutation for which possession of K enables the e cient computation of both F_K and $F_K^{\otimes 1}$.

The ciphers we construct have a strong security property: we want that no ef cient adversary can distinguish an oracle for $F_K(\ )$, for a random and secret K, from an oracle for a random length-preserving permutation $(\ )$ (having the same domain as F_K). This is the (now customary) requirement for a block cipher (security in the sense of being a "pseudorandom permutation," or "PRP") originally suggested in [10,4], and so it is the property we want for any variable-input-length cipher as well.

L. Knudsen (Ed.): FSE'99, LNCS 1636, pp. 231–244, 1999.
© Springer-Verlag Berlin Heidelberg 1999

One could try to construct a variable-input-length cipher from scratch, in the confusion/di usion tradition. But that approach is specialized and error-prone. Instead, we provide constructions which assume one already has in hand some underlying block cipher. We give a "mode of operation"—VIL mode (for "variable-input-length enciphering") which enciphers strings of arbitrary length (but at least n) using an n-bit block cipher.

We prove the soundness of VIL mode in the provable-security sense of modern cryptography: if the underlying block cipher is secure then so is the variable-input-length cipher we construct from it. VIL is actually more than one particular mode of operation; it is an approach for making a variable-input-length cipher that can be realized in many di erent ways.

WHY VARIABLE-INPUT-LENGTH CIPHERS? The obvious use of variable-input-length ciphers is to encrypt (ie, provide privacy protection) without any increase in message length. Suppose we'll be encrypting messages $M_1, M_2,$ where the lengths of these message may vary. We want to create ciphertexts $C_1, C_2,$ where $|C_i| = |M_i|$ and where ciphertext C_i hides everything about M_i (with respect to e cient computation) except for the length of M_i *and which earlier plaintext, if any, equals* M_i.

It is important to understand that the last sentence embodies a weaker notion of privacy than the customary one—semantic security, and its equivalent formulations [7,3]. A semantically secure encryption computationally hides all information about M_i except for $|M_i|$—in particular, one does *not* allow to be leaked which earlier plaintext (if any) a given ciphertext corresponds to. But you pay a price for this added security—semantically secure encryption cannot possibly be length preserving. Thus length-preserving "encryption" (enciphering) embodies a tradeo : shorter ciphertexts at the cost of an inferior security guarantee (and slower encryption/decryption).

Is this tradeo a good one? If you don't know anything about how the encryption will be used, then we'd have to say no. But there are applications when the tradeo *is* a good one. Let us give an example.

In networking applications a "packet format" may have been de ned, this packet format having various elds, none of which were intended for cryptographic purposes. Now suppose a need arises to *add in* privacy features but, at the same time, it is no longer desirable (or feasible) to adjust the packet format. It cannot be lengthened by even one bit. Enciphering with a variable-input-length cipher leaves the packet size alone, and it leaves packets looking identical (after deciphering) to the way they looked before. This contributes to ease-of-acceptance, an easier migration path, and better code-reuse. These factors may outweigh the security consideration that we will be leaking which packets of a session are identical to which earlier ones.

As a second example, we may have *a priori* reason to believe that all the plaintexts $M_1, M_2,$ will be distinct. For example, each message may be known to contain a sequence number. In such a case the additional piece of information that secure encipherment leaks amounts to no information at all, and so here enciphering provides a way to achieve semantic security in a way that is both

length-minimal and oblivious to the formatting conventions of each message (eg, where the sequence number appears in each message). This obliviousness contributes to the making of robust software; when message formats change the cryptography need not be adjusted. With typical length-minimal approaches this would not have been true.

Variable-input-length ciphers may prove to be useful tools for protocol design. As an example, Rivest put forward the idea of "strongly non-separable encryption" [23], wherein an adversary with a ciphertext C who guesses an encryption key K should have to invest $\Omega(|C|)$ time before obtaining information useful to verify if C was enciphered under K. Variable-input-length enciphering provides a simple way to provably achieve Rivest's goal.

THE DIFFICULTY. It is not so clear *how* to construct a secure variable-input-length cipher from a block cipher. We are making a stringent security requirement: we expect our ciphers to approximate a family of random permutations. In addition, we want them to be length-preserving permutations. This eliminates any hope of using conventional modes of operation. Consider, for example, using DES in CBC mode with a zero initialization vector (IV). For simplicity, assume the message length is a multiple of the blocklength.[1] This does *not* give a cipher that approximates a family of random permutation: if two plaintexts agree on blocks $1, \ldots, i$ then their ciphertexts agree on blocks $1, \ldots, i$, which is almost never true of a random permutation. To get around this one might try to make the IV some sort of hash of the message—but then how could one get a length-preserving construction?

OUR METHOD. We suggest simple and efficient ways for making variable-input-length ciphers from block ciphers. Our VIL mode of operation makes two passes over the message. In our preferred instantiation, the first pass computes some sort of CBC MAC over the message M, while the second pass encrypts M (in counter mode, for example) using the pass-one MAC as the IV. However, one cannot take the ciphertext C for M to be the pass-two ciphertext (including the IV), since this would be too long. Instead, we exploit a certain feature of the CBC MAC, which we call its "parsimoniousness." This enables us to drop one block from the pass-two ciphertext and *still* be able to recover the plaintext. (This is the main idea of our construction.) There are some technical matters that complicate things; see Section 2 and Fig. 1.

Our approach can be instantiated in many further ways; it actually encompasses many modes of operation. We describe VIL mode in terms of two specialized-tools: what we call a "parsimonious" pseudorandom function (PRF) and a "parsimonious" encryption scheme. Both of these tools can be constructed from block ciphers, and we show a few ways to do this. Thinking of VIL mode in these general terms not only provides versatility in instantiation, but,

[1] The difficult issue is not in dealing with messages of length not a multiple of the blocklength; there are well-known methods for dealing with this, like stream-cipher encrypting the short block and ciphertext stealing. See [13, Chapter 2] for a description of these techniques.

equally important, our proof of correctness is made much simpler by the added generality: what is irrelevant is out of sight, and what is relevant can be singled out and separately proved, in part by invoking known results [4,21,3].

Related work. There is a quite a lot of work on constructing block ciphers of one blocklength given block ciphers of another blocklength. Luby and Racko [10] consider the question of how to turn an n-bit to n-bit pseudorandom function (PRF) into a $2n$-bit to $2n$-bit block cipher. They show that three rounds of the Feistel construction su ces for this purpose, and that four rounds su ce to obtain a "super" PRP from a PRF. The paper has spawned much work, with [12,22,19,20,25] to name a few.

Naor and Reingold [15] provide a construction which extends a block cipher on n-bits to a block cipher on $N = 2ni$ bits, for any desired i 1. A variation on their construction yields a cipher on $N = ni$ bits for any i 1 [18]. It is unclear how to use these constructions for arbitrary N (meaning not necessarily a multiple of n) and across assorted input lengths.

Lucks [11] generalizes Luby-Racko to consider a three round unbalanced Feistel network, using hash functions for round functions. This yields a block cipher on any given length N by starting with a PRF of r bits to bits and another of bits to r bits where $r + = N$. Of course this requires the availability of the latter primitives for given values of $r, $.

Anderson and Biham [1] provide two constructions for a block cipher (BEAR and LION) which use a hash function and a stream cipher. This too is an unbalanced Feistel network.

Some ciphers which are intended to operate on blocks of various lengths have been constructed from scratch. The CMEA (attacked by [24]) is an example.

A "forward-then-backwards" mode of operation is described in [8], under the names "Triple-DES Key Wrap" and "RC2 Key Wrap." While not length-preserving, a length-preserving variant is possible, and it might be a good cipher across messages of assorted lengths. See Section 5 for further discussion.

We have already mentioned Rivest's "strongly non-separable" encryption [23] and that variable-input-length enciphering provides one mechanism to achieve that goal.

The VIL mode of operation was invented in 1994 when the authors were at IBM [2]. No security analysis was provided at that time.

2 VIL Mode Example

In this section we describe one particular instantiation of VIL mode enciphering. For concreteness, let us start from DES, a map DES : $\{0,1\}^{56}$ $\{0,1\}^{n} \rightarrow \{0,1\}^{n}$ where $n = 64$. Using this map we construct the function $F : \{0,1\}^{56}$ ^{3} $\{0,1\}$ $^{64} \rightarrow \{0,1\}$ ^{64} for enciphering strings of length at least 64. (Extending to messages of length less than 64 will be discussed later.) Given a key $K = K1 \parallel K2 \parallel K3$, partitioned into three 56-bit pieces, and given a plaintext $M \in \{0,1\}$ ^{64}, form the ciphertext $C = F_K(M)$ as depicted in Fig. 1 and as speci ed here:

Algorithm $F_{K1 \,\|\, K2 \,\|\, K3}(M)$

(1) Let $M_{\text{pre x}}$ be the rst $|M| \otimes n$ bits of M. Let $M_{\text{su x}}$ be the remaining bits.
(2) Let *pad* be a "1" followed by the minimum number of "0" bits such that $|M| + |pad|$ is divisible by 64.
(3) Partition $M_{\text{pre x}} \,\|\, pad \,\|\, M_{\text{su x}}$ into 64-bit blocks $M_1 \quad M_m$.
(4) Let $C_0 = 0^n$, and let $C_i = \text{DES}_{K1}(C_{i\otimes 1} \quad M_i)$ for all $1 \quad i \quad m$.
(5) Let $= \text{DES}_{K2}(C_m)$.
(6) Let P be the rst $|M| \otimes n$ bits of

$$\text{DES}_{K3}(\ ) \,\|\, \text{DES}_{K3}(\ + 1) \,\|\, \text{DES}_{K3}(\ + 2) \quad .$$

(7) Let $C_{\text{pre x}} = P \quad M_{\text{pre x}}$.
(8) Return ciphertext $C = \ \| \, C_{\text{pre x}}$.

The computation of C can be looked at as having two stages. In the rst stage (Steps 1–5) we compute , which is some sort of CBC-MAC of M under the key $K1 \,\|\, K2$. In the second stage (Steps 6–7) we encrypt M, except for M's last 64 bits, under key $K3$. We use counter-mode encryption with an initialization vector of . The ciphertext is the MAC together with the encrypted pre x of M.

The MAC is not computed by the "basic" CBC-MAC, but some variant of it. Our constraints preclude using the CBC-MAC in its customary form. First we need to be able to properly handle messages of arbitrary length (the basic CBC-MAC is only secure on messages of some xed length, this length being a multiple of the blocklength). But in addressing this issue we must ensure that given and an $|M| \otimes 64$ bit pre x of M, we are able to reconstruct the last 64 bits of M. That this can be done can be seen in the following algorithm for computing $F^{\otimes 1}_{K1 \,\|\, K2 \,\|\, K3}(C)$. As before, $C \in \{0,1\}^{64}$ and $K1, K2, K3 \in \{0,1\}^{56}$. The existence of the following algorithm demonstrates that F is indeed a cipher.

Algorithm $F^{\otimes 1}_{K1 \,\|\, K2 \,\|\, K3}(C)$

(1) Let be the rst 64 bits of C. Let $C_{\text{pre x}}$ be the remaining bits.
(2) Let P be the rst $|C_{\text{pre x}}|$ bits of

$$\text{DES}_{K3}(\ ) \,\|\, \text{DES}_{K3}(\ + 1) \,\|\, \text{DES}_{K3}(\ + 2) \quad .$$

(3) Let $M_{\text{pre x}} = P \quad C_{\text{pre x}}$.
(4) Let *pad* be a "1" followed by the minimum number of "0" bits such that $|M_{\text{pre x}}| + |pad|$ is divisible by 64.
(5) Partition $M_{\text{pre x}} \,\|\, pad$ into 64-bit blocks $M_1 \quad M_{m\otimes 1}$.
(6) Let $C_0 = 0^n$, and let $C_i = \text{DES}_{K1}(C_{i\otimes 1} \quad M_i)$ for all $1 \quad i \quad m \otimes 1$.
(7) Let $M_m = \text{DES}^{\otimes 1}_{K1}(\text{DES}^{\otimes 1}_{K2}(\ ))) \quad C_{m\otimes 1}$.
(8) Return $M = M_{\text{pre x}} \,\|\, M_m$.

The interesting step is Step 7, where one exploits the structure of (this version of) the CBC-MAC to compute the last block of plaintext.

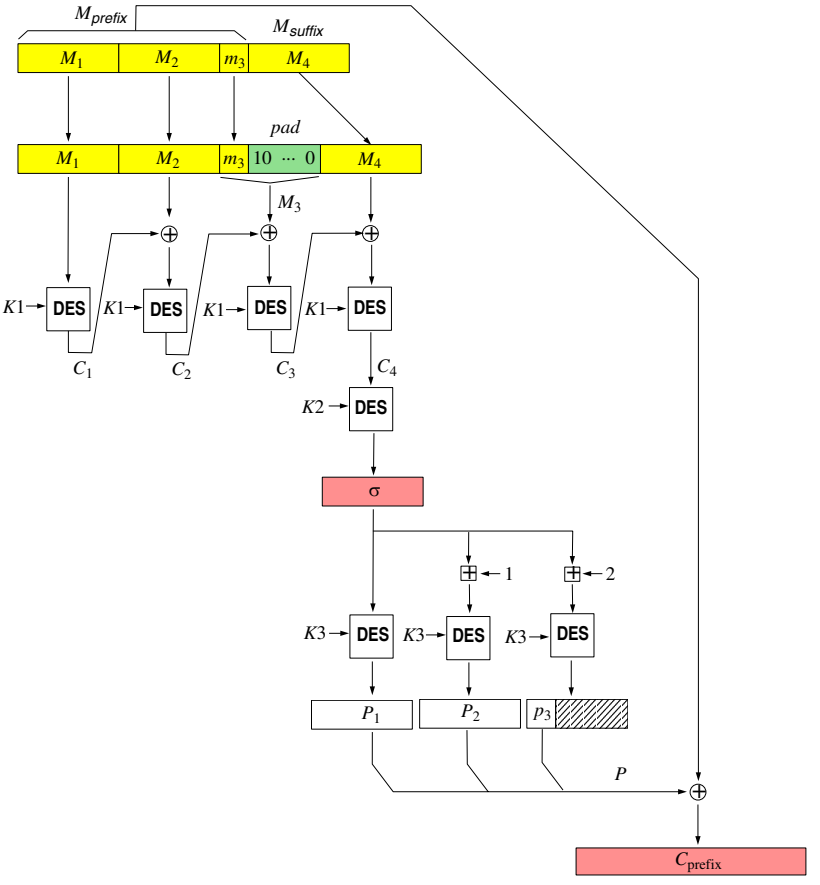

Fig. 1. *An example way to realize* VIL*-mode encipherment. Here we use the block cipher DES. In this example the message M to encipher is a few bits longer than 64×3 bits. The underlying key is $K = K1 \parallel K2 \parallel K3$. The ciphertext is $C = \sigma \parallel C_{\text{prefix}}$.*

We remark that standard methods, like setting $K\text{i} = \text{DES}_K(\text{i})[1..56]$, would allow $K1$, $K2$ and $K3$ to be derived from a single 56-bit key, in which case F would be a map $F : \{0,1\}^{56} \quad \{0,1\}^{64} \rightarrow \{0,1\}^{64}$.

We also remark that that the domain can be extended to all of $\{0,1\}$ (that is, we can encipher strings of fewer than < 64 bits) using methods which we will later discuss. However, these methods have not been proven secure with desirable security bounds.

It should be kept in mind that the above example is just one way to instantiate VIL-mode encipherment. Both stages (the computation of and the encryption of $M_{\text{pre x}}$) can be accomplished in other ways. We now move towards these generalizations.

3 The General Approach

Towards the general description of VIL and its proof of correctness we now make some de nitions.

Preliminaries. A *message space* $\mathcal{M}$ is a nonempty subset of $\{0,1\}$ for which $M \in \mathcal{M}$ implies that $M' \in \mathcal{M}$ for all M' of the same length of M. A *ciphertext space* (or *range*) $\mathcal{C}$ is a nonempty subset of $\{0,1\}$. A *key space* $\mathcal{K}$ is a nonempty set together with a probability measure on that set. A *pseudorandom function* (PRF) with key space $\mathcal{K}$, message space $\mathcal{M}$ and range $\mathcal{C}$ is a set of functions $F = \{F_K \mid K \in \mathcal{K}\}$ where each $F_K : \mathcal{M} \to \mathcal{C}$. We usually write $F : \mathcal{K} \quad \mathcal{M} \to \mathcal{C}$. We assume that $|F_K(M)|$ depends only on $|M|$. A *cipher* is a PRF $F : \mathcal{K} \quad \mathcal{M} \to \mathcal{M}$ in which each $F_K : \mathcal{M} \to \mathcal{M}$ is a bijection. A *block-cipher* is a cipher $F : \mathcal{K} \quad \{0,1\}^n \to \{0,1\}^n$. The number n is called the *blocklength*.

Let $\mathcal{M}$ be a message space and let $\square : \mathsf{N} \to \mathsf{N}$ be a function. We de ne "reference" PRFs $\mathsf{Rand}(\mathcal{M}, \square)$ and $\mathsf{Perm}(\mathcal{M})$. A random function $\mathsf{Rand}(\mathcal{M}, \square)$ is de ned as follows: for each $M \in \mathcal{M}$, let (M) is a random string in $\{0,1\}^{\square(|M|)}$. A random function $\mathsf{Perm}_{\mathcal{M}}$ is de ned as follows: for each number i such that $\mathcal{M}$ contains strings of length i, let $_i$ be a random permutation on $\{0,1\}^i$, and de ne $(M) = _i(M)$ where $i = |M|$.

We de ne security following [6], adapted to concrete security as in [4]. A *distinguisher* is a (possibly probabilistic) algorithm A with access to an oracle. Let A be a distinguisher and let $F = \{F_K \mid K \in \mathcal{K}\}$ be a PRF with key space $\mathcal{K}$ and $|F_K(M)| = \square(|M|)$. Then we let

$$\mathrm{Adv}^{\mathrm{prf}}_F(A) = \Pr[K \quad \mathcal{K} : A^{F_K(\,)} = 1] \otimes \Pr[\quad \mathsf{Rand}(\mathcal{M}, \square) : A^{(\,)} = 1] \quad \text{and}$$
$$\mathrm{Adv}^{\mathrm{prp}}_F(A) = \Pr[K \quad \mathcal{K} : A^{F_K(\,)} = 1] \otimes \Pr[\quad \mathsf{Perm}(\mathcal{M}) : A^{(\,)} = 1] \,.$$

De ne the functions $\mathrm{Adv}^{\mathrm{prf}}_F(t, q, \,) = \max_A\{\mathrm{Adv}^{\mathrm{prf}}_F(A)\}$ and $\mathrm{Adv}^{\mathrm{prp}}_F(t, q, \,) = \max_A\{\mathrm{Adv}^{\mathrm{prp}}_F(A)\}$ where the maximum is over all adversaries which run in time at most t and ask at most q oracle queries, these queries totaling at most bits. We omit the argument when $\mathcal{M}$ contains strings of just one length. Time is always understood to include the space for the description of the distinguishing algorithm. Throughout, if the distinguisher inquires as to the value of oracle f at a point $M \notin \mathcal{M}$ then the oracle responds with the distinguished point $\perp$. We assume there is a (simple) algorithm to decide membership in $\mathcal{M}$ and so we may assume adversaries do not make such queries.

Parsimonious PRF. Let $G : \{0,1\} \quad \mathcal{M} \to \{0,1\}^n$ be a PRF where $\mathcal{M}$ only includes strings of length at least n. Then G is said to be *parsimonious* if for all $K \in \mathcal{K}$ and all $M \in \mathcal{M}$, the last n bits of M are uniquely determined by the remaining bits of M, the key K, and $G_K(M)$. In other words, with a parsimonious PRF G, if you know K and receive the n-bit value $= G_K(M)$ then you don't need to receive *all* of M in order to know what it is: it is su cient to get the $|M| \otimes n$ bit pre x of M, $M_{\text{pre x}}$: from that you can recover the n missing bits by applying some function $\mathsf{Recover}_K(M_{\text{pre x}}, \,)$ associated to the PRF.

Examples. The (basic) CBC-MAC is a parsimonious PRF. Assume a block cipher $E : \mathcal{K} \quad \{0,1\}^n \to \{0,1\}^n$. Fix a constant $m \quad 1$. Consider the PRF $G : \mathcal{K} \quad \{0,1\}^{nm} \to \{0,1\}^n$ de ned by $G_K(M_1 \quad M_m) = C_m$, where $C_0 = 0^n$ and $C_i = E_K(M_i \quad C_{i\otimes 1})$ for $1 \quad i \quad m$. To recover M_m from K, $M_1 \quad M_{m\otimes 1}$, and $\quad = G_K(M_1 \quad M_m)$, compute $C_0, C_1, \ldots, C_{m\otimes 1}$ by $C_0 = 0^n$ and $C_i = E_K(C_{i\otimes 1} \quad M_i)$ and then, since $C_m = E_K(M_m \quad C_{m\otimes 1})$, recover M_m as $M_m = E_K^{\otimes 1}(\ ) \quad C_{m\otimes 1}$. Note that it is crucial that we use the "full" CBC MAC (that is, the MAC is all of C_m, not a proper pre x). In [4] it is shown that the CBC MAC is secure whenever E is, in the sense that $\mathrm{Adv}^{\mathrm{prf}}_G(t,q) \quad \mathrm{Adv}^{\mathrm{prf}}_E(t',q') + 3q^2m^2 2^{\otimes n\otimes 1}$ where $t' \quad t$ and $q' = qm$.

The computation of $\quad$ in the algorithm of Section 2 builds on the idea described above. We extend the CBC-MAC variant analyzed in [21] to domain $\{0,1\}^{\ }$, doing this in a way that retains parsimoniousness (padding the second-to-last block instead of the last one). This CBC-MAC variant is once again secure. Let $G : \mathcal{K}^2 \quad \{0,1\}^{\ } \to \{0,1\}^n$ be the PRF obtained from the block cipher E by the method illustrated in Lines 1–5 in the description of Algorithm $F_{K1 \,\|\, K2 \,\|\, K3}(M)$ in Section 2 (where we had $E = \mathrm{DES}$). Then the results of [21] can be adapted to establish that that $\mathrm{Adv}^{\mathrm{prf}}_G(t,q,\ ) \quad 2 \quad \mathrm{Adv}^{\mathrm{prf}}_E(t',q') + (\ /n + q)^2 2^{\otimes n} + 2^{\otimes n}$ where $t' \quad t$ and $q' = \ /n + q$.

Parsimonious encryption. A parsimonious encryption scheme is a triple of algorithms $\mathsf{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. Algorithm $\mathcal{K}$ returns a random element from the key space (which we likewise denote $\mathcal{K}$). Encryption algorithm $\mathcal{E} : \mathcal{K} \quad \mathcal{M}$ takes a key $K \in \mathcal{K}$ and $M \in \mathcal{M}$, chooses a random IV $\quad \{0,1\}^n$, and then encrypts the message M into a ciphertext $C = \mathrm{IV} \,\|\, C\ $, where $|C\ | = |M|$. The process is denoted $C \quad \mathcal{E}_K(M)$, or $C \quad \mathcal{E}_K(M;\ \mathrm{IV})$ when we regard IV as an explicitly given input to $\mathcal{E}$. The decryption algorithm has domain $\mathcal{K} \quad \{0,1\}^{\ }$ and, given $K \in \mathcal{K}$ and $C \in \{0,1\}^{\ }$, $\mathcal{D}_K(C) = M$ whenever $C = \mathcal{E}_K(M;\ \mathrm{IV})$ for some $M \in \mathcal{M}$ and $\mathrm{IV} \in \{0,1\}^n$.

We de ne security following [3]. The idea is that an adversary cannot distinguish the encryption of text from the encryption of an equal-length string of garbage. Let $\mathsf{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a parsimonious encryption scheme and let A be a distinguisher. Then

$$\mathrm{Adv}^{\mathrm{priv}}_A(\mathsf{S}) \ = \ \Pr\left[K \quad \mathcal{K} : A^{\mathcal{E}_K(\ )} = 1\right] \otimes \Pr\left[K \quad \mathcal{K} : A^{\mathcal{E}_K(\$^{|\ |})} = 1\right].$$

In the rst experiment the oracle, given M, returns a random encryption of M under K, while in the second experiment it returns a random encryption of a random string of length $|M|$. De ne $\mathrm{Adv}^{\mathrm{priv}}_{\mathsf{S}}(t,q,\ )$ to be $\max_A\{\mathrm{Adv}^{\mathrm{priv}}_{\mathsf{S}}(A)\}$ where the maximum is over all adversaries who run in time at most t and ask at most q oracle queries, these totaling at most $\quad$ bits.

Examples. Common methods of symmetric encryption using a block cipher are parsimonious. For example, CBC-mode encryption with a random IV is parsimonious. Its domain is $\mathcal{M} = (\{0,1\}^n)^+$, where n is the blocklength of the underlying block cipher. The domain for CBC-mode encryption is easily enlarged to $\mathcal{M} = \{0,1\}^{\ }$; for example, if the last "block" M_m of plaintext has length

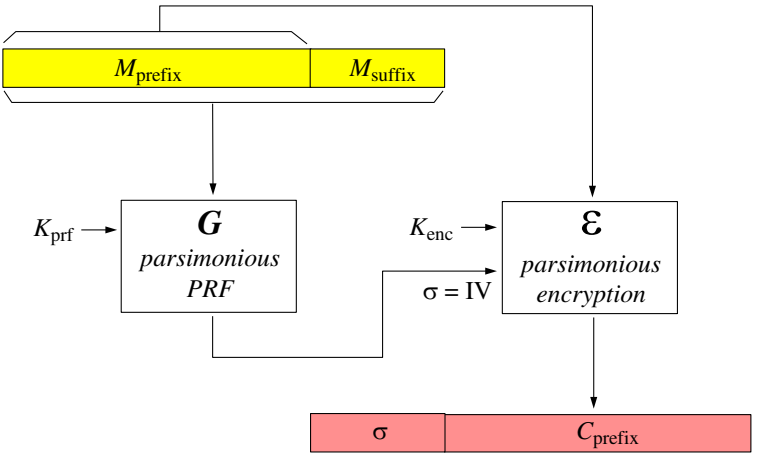

Fig. 2. *A general description of* VIL *mode. The ciphertext is* $\sigma \parallel C_{\text{prefix}}$. *The value* σ *is the output of the PRF* G, *the IV to the encryption scheme, and the first* n *bits of ciphertext.*

less than the blocklength n then encrypt it as $C_m = E_K(C_{m\otimes 1})[1..|M_m|) \quad M_m$. Alternatively, counter-mode encryption (with a random initial counter) is parsimonious and has domain $\{0,1\}$. This was the choice for Stage 2 in our example scheme of Section 2. The security of CBC-mode and counter-mode encryption are established in [3].

VIL: General scheme. We are now ready to give the general description of VIL mode. Let $\mathcal{M}'$ be a message space, let $n \quad 1$ be a number, and let $\mathcal{M} = \mathcal{M}'\{0,1\}^n$ (strings n-bits longer than strings in $\mathcal{M}$). Let $G : \mathcal{K}_{\text{prf}} \quad \mathcal{M} \to \{0,1\}^n$ be a parsimonious PRF, and let $\mathsf{Recover} : \mathcal{K}_{\text{enc}} \quad \mathcal{M}' \quad \{0,1\}^n \to \{0,1\}^n$ be its associated recovery algorithm. Let $\mathsf{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a parsimonious encryption scheme in which $\mathcal{E} : \mathcal{K}_{\text{enc}} \quad \mathcal{M}' \to \mathcal{M}$. Then we construct the cipher $F = \text{VIL}[G, \mathsf{S}]$, where $F : \mathcal{K} \quad \mathcal{M} \to \mathcal{M}$, by setting $\mathcal{K} = \mathcal{K}_{\text{prf}} \quad \mathcal{K}_{\text{enc}}$ and de ning:

Algorithm $F_{K_{\text{prf}} \parallel K_{\text{enc}}}(M)$	**Algorithm** $F^{\otimes 1}_{K_{\text{prf}} \parallel K_{\text{enc}}}(C)$
$M_{\text{pre x}} = M\,[1..\lvert M\rvert \otimes n]$	be the rst n bits of C
$\quad = G_{K_{\text{prf}}}(M)$	$M_{\text{pre x}} = \mathcal{D}_{K_{\text{enc}}}(C)$
$C_{\text{pre x}} = \mathcal{E}_{K_{\text{enc}}}(M_{\text{pre x}};\ )$	$M_{\text{su x}} = \mathsf{Recover}_{K_{\text{prf}}}(M_{\text{pre x}},\ )$
return $C = \quad \parallel C_{\text{pre x}}$	return $M = M_{\text{pre x}} \parallel M_{\text{su x}}$

For a picture of the general scheme, see the Fig. 2.

4 Analysis

The following theorem says that F as constructed above is a secure variable-input-length cipher, as long as both G and S are secure.

Theorem 1. *Let* $F = \mathrm{VIL}[G, \mathsf{S}]$ *be the cipher obtained from the parsimonious PRF* $G : \mathcal{K}_{\mathrm{prf}} \quad \mathcal{M} \to \{0,1\}^n$ *and the parsimonous encryption scheme* $\mathsf{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$*. Then*

$$\mathrm{Adv}^{\mathrm{prp}}_{F}(t, q, \ ) \quad \mathrm{Adv}^{\mathrm{prf}}_{G}(t', q, \ ) + \mathrm{Adv}^{\mathrm{priv}}_{\mathsf{S}}(t', q, \ ) + \frac{q^2}{2^n} \ ,$$

where $t' = t + O(qn + \ )$*.*

Proof. Let A be an adversary attacking F, and let t be its running time, q the number of queries it makes, and the total length of all its queries put together. We assume without loss of generality that A never repeats an oracle query. This is important to some of the claims made below. We consider various probabilities related to running A under various di erent experiments:

$$\begin{aligned}
p_1 &= \Pr[K \quad \mathcal{K} : A^{F_K(\ )} = 1] \\
p_2 &= \Pr[K_{\mathrm{enc}} \quad \mathcal{K}_{\mathrm{enc}} \ ; g \quad \mathsf{Rand}(\mathcal{M}, n) : A^{\mathcal{E}_{K_{\mathrm{enc}}}((\ )_{\mathrm{prefix}}; g(\ ))} = 1] \\
p_3 &= \Pr[K_{\mathrm{enc}} \quad \mathcal{K}_{\mathrm{enc}} \ ; A^{\mathcal{E}_{K_{\mathrm{enc}}}((\ )_{\mathrm{prefix}})} = 1] \\
p_4 &= \Pr[K_{\mathrm{enc}} \quad \mathcal{K}_{\mathrm{enc}} \ ; A^{\mathcal{E}_{K_{\mathrm{enc}}}(\$^{|(\ )_{\mathrm{prefix}}|})} = 1] \\
p_5 &= \Pr[\quad \mathsf{Perm}(\mathcal{M}) : A^{\ (\ )} = 1]
\end{aligned}$$

Let us explain the new notation. In the experiment de ning p_2, A s oracle, on query M, responds by encrypting the rst $|M| \otimes n$ bits of M using coins $\mathrm{IV} = g(M)$. In the experiment de ning p_3, A s oracle, on query M, responds by randomly encrypting the rst $|M| \otimes n$ bits of M. In the experiment de ning p_4, A s oracle, on query M, responds by randomly encrypting a string of $|M| \otimes n$ random bits.

Our goal is to upper bound $\mathrm{Adv}^{\mathrm{prp}}_{F}(A) = p_1 \otimes p_5$. We do this in steps.

Claim. $p_1 \otimes p_2 \quad \mathrm{Adv}^{\mathrm{prf}}_{G}(t', q, \ )$.

Proof. Consider the following distinguisher D for G. It has an oracle for g: $\mathcal{M} \to \{0,1\}^n$. It picks $K_{\mathrm{enc}} \quad \mathcal{K}_{\mathrm{enc}}$. It runs A, and when A makes oracle query M it returns $\mathcal{E}_{K_{\mathrm{enc}}}(M_{\mathrm{pre\ x}}; g(M))$ to A as the answer (where $M_{\mathrm{pre\ x}}$ is the rst $|M| \otimes n$ bits of M.) Finally D outputs whatever A outputs. Then

$$\begin{aligned}
\Pr[K_{\mathrm{prf}} \quad \mathcal{K}_{\mathrm{prf}} : D^{G_{K_{\mathrm{prf}}}(\ )} = 1] &= p_1 \\
\Pr[g \quad \mathsf{Rand}(\mathcal{M}, n) : D^{g(\ )} = 1] &= p_2
\end{aligned}$$

So $\mathrm{Adv}^{\mathrm{prf}}_{G}(D) = p_1 \otimes p_2$. The claim follows.

Claim. $p_2 = p_3$.

Proof. The only di erence between the experiment underlying p_2 and that underlying p_3 is that in the former, the IV used for encryption is a random function of M, while in the latter it is chosen at random by the encryption algorithm. These are the same as long as all the oracle queries are di erent, which is what we assumed about A.

Claim. $p_3 \otimes p_4 \quad \mathrm{Adv}^{\mathrm{priv}}_{\mathsf{S}}(t', q, \ )$.

Proof. Consider the following adversary B for S that is given an oracle $\mathcal{O}$. It runs A, and when A makes oracle query M it returns $\mathcal{O}(M_{\mathrm{pre}\ x})$ to A as the answer (where $M_{\mathrm{pre}\ x}$ is the rst $|M| \otimes n$ bits of M). Finally D outputs whatever A outputs. Then

$$\Pr[K_{\mathrm{enc}} \quad \mathcal{K}_{\mathrm{enc}} : B^{\mathcal{E}_{K_{\mathrm{enc}}}(\ )} = 1] = p_3$$
$$\Pr[K_{\mathrm{enc}} \quad \mathcal{K}_{\mathrm{enc}} : B^{\mathcal{E}_{K_{\mathrm{enc}}}(\$|\ |)} = 1] = p_4 \ .$$

So $\mathrm{Adv}^{\mathrm{priv}}_{B}(\mathsf{S}) = p_3 \otimes p_4$. The claim follows.

Claim. $p_4 \otimes p_5 \quad q^2/2^n$.

Proof. Let $r = \Pr[h \quad \mathsf{Rand}(\mathcal{M}) : A^{h(\ )} = 1]$. We argue that $p_4 \otimes r \quad q^2/2^{n+1}$ and also $r \otimes p_5 \quad q^2/2^{n+1}$. The claim follows by the triangle inequality. It remains to prove the two subclaims.

The second subclaim, that $r \otimes p_5 \quad q^2/2^{n+1}$, is of course clear; the statistical distance between a family of functions and a family of permutations is given by the collision probability under q queries. So consider the rst subclaim, namely $p_4 \otimes r \quad q^2/2^{n+1}$. This is true because the encryption scheme is parsimonious. The IV is chosen at random, and for each xed IV, the map $\mathcal{E}_{K_{\mathrm{enc}}}((\ )_{\mathrm{pre}\ x}; \mathrm{IV})$ is a permutation on $\mathcal{M}$. Thus, $p_4 \otimes r$ is the statistical distance between a family of permutations on $\mathcal{M}$ and a family of random functions on $\mathcal{M}$, which is again $q^2/2^{n+1}$ because all strings in $\mathcal{M}$ have length at least n.

Given these claims, we can complete the proof of the theorem by noting that

$$\mathrm{Adv}^{\mathrm{prp}}_{F}(A) = p_1 \otimes p_5 = (p_1 \otimes p_2) + (p_2 \otimes p_3) + (p_3 \otimes p_4) + (p_4 \otimes p_5) \ . \quad \blacksquare$$

5 Comments and Open Problems

Our security bound for VIL mode enciphering degrades with q^2, as do the bounds for other common modes of operation. It would be interesting to nd a method and analysis which had better quantitative security.

It would be desirable to have a good constructions for a *super* variable-input-length cipher (again, starting with a block cipher). Following [10], a super pseudorandom cipher F is one for which no reasonable adversary can do well at distinguishing a pair of oracles $(F_K(\ ), F_K^{\otimes 1}(\ ))$, for a random $K \in \mathcal{K}$, from a pair of oracles $(\ (\ ), \ ^{\otimes 1}(\ ))$, for a random permutation $(\ )$. This question has been investigated by Bleichenbacher and Desai, who point out that our VIL construction is not a super variable-input-length cipher, and they propose a construction for such a cipher [5].

We have focussed on the case in which the message length is at least the blocklength n of the underlying block cipher. For shorter messages of even length $2\square$ one can proceed as follows. First map the underlying enciphering key K into subkeys $(K_{\mathrm{enc}}, K_{\mathrm{prf}}, K_1, K_2, \ldots, K_{\square})$ using standard key-separation techniques.

Now when $|M| \quad n$, proceed according to VIL mode, using keys K_{enc} and K_{prf}. But when $|M| < n$ encipher M using an r-round Feistel network, keying the block-cipher-derived round function by $K_{|M|/2}$. We point out that while such an approach may work well in practice, the bounds one gets following [10] and its follow-on work will be very weak for our purposes, since these bounds degrade as the blocklength shrinks and we are here imagining a blocklength of just a few bits. Thus enciphering very short messages in a provably-good way remains open.

When this paper was presented at FSE 99, Mike Matyas described an alternative construction to encipher a message M: rst, CBC-encrypt M (with zero IV) to get a ciphertext N; and then, to generate the ciphertext C, CBC-encrypt N, but starting from the its last black and working back towards the rst block. A similar scheme is given in [8]. Ciphertext stealing can be used to handle inputs of length not a multiple of the blocklength. This sort of "forward-then-backwards" CBC sounds like an elegant approach, and it would be interesting to know if some version of it can be proven secure.

Acknowledgments

Many thanks to the anonymous reviewers of FSE 99, whose comments signi - cantly improved our presentation. And thanks to Stefan Lucks and Ron Rivest for their comments on an earlier version of this work.

Mihir Bellare was supported by NSF CAREER Award CCR-9624439 and a Packard Foundation Fellowship in Science and Engineering. Phillip Rogaway was supported by NSF CAREER Award CCR-962540, and MICRO grants 97-150 and 98-129, funded by RSA Data Security, Inc., and ORINCON Corporation. Much of Phill s work on this paper was carried out while on sabbatical at Chiang Mai University, Thailand, hosted by the Computer Service Center, under Prof. Krisorn Jittorntrum and Prof. Darunee Smawatakul.

References

1. R. Anderson and E. Biham, "Two practical and provably secure block ciphers: BEAR and LION." *Proceedings of the 3rd Fast Software Encryption Workshop*, Lecture Notes in Computer Science Vol. 1039, Springer-Verlag, 1996.
2. M. Bellare and P. Rogaway, "Block cipher mode of operation for secure, length-preserving encryption." US Patent #5,673,319, September 30, 1997. Filed February 6, 1995.
3. M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption." *Proceedings of the* 38th *Symposium on Foundations of Computer Science*, IEEE, 1997.
4. M. Bellare, J. Kilian and P. Rogaway, "The security of cipher block chaining." *Advances in Cryptology – Crypto* 94 *Proceedings*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.
5. D. Bleichenbacher and A. Desai, "A construction of a super-pseudorandom cipher." Manuscript, February 1999.

6. O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions." *Journal of the ACM,* Vol. 33, No. 4, 210–217, 1986.
7. S. Goldwasser and S. Micali, "Probabilistic encryption." *Journal of Computer and System Sciences* Vol. 28, 270-299, April 1984.
8. R. Housley, "Cryptographic message syntax." S/MIME Working Group of the IETF, Internet Draft `draft-ietf-smime-cms-12.txt`. March 1999.
9. ISO/IEC 9797, "Information technology – Security techniques – Data integrity mechanism using a cryptographic check function employing a block cipher algorithm," International Organization for Standardization, Geneva, Switzerland, 1994 (second edition).
10. M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions." *SIAM J. Computing,* Vol. 17, No. 2, April 1988.
11. S. Lucks, "Faster Luby-Rackoff ciphers." *Proceedings of the 3rd Fast Software Encryption Workshop*, Lecture Notes in Computer Science Vol. 1039, Springer-Verlag, 1996.
12. U. Maurer, "A simplified and generalized treatment of Luby-Rackoff pseudorandom permutation generators." *Advances in Cryptology – Eurocrypt* 92 *Proceedings*, Lecture Notes in Computer Science Vol. 658, R. Rueppel ed., Springer-Verlag, 1992, pp. 239–255.
13. C. Meyer and M. Matyas, *Cryptography: A New Dimension in Data Security*. John Wiley & Sons, New York, 1982.
14. S. Micali, C. Rackoff and R. Sloan, "The notion of security for probabilistic cryptosystems." *SIAM J. Computing,* Vol. 17, No. 2, April 1988.
15. M. Naor and O. Reingold, "On the construction of pseudorandom permutations: Luby-Rackoff revisited." *Proceedings of the* 29th *Annual Symposium on Theory of Computing*, ACM, 1997.
16. National Bureau of Standards, FIPS PUB 46, "Data encryption standard." U.S. Department of Commerce, January 1977.
17. National Bureau of Standards, FIPS PUB 81, "DES modes of operation." U.S. Department of Commerce, December 1980.
18. S. Patel, Z. Ramzan and G. Sundaram, "Towards making Luby-Rackoff ciphers optimal and practical." *Proceedings of the 6th Fast Software Encryption Workshop*, 1999.
19. J. Patarin, "Improved security bounds for pseudorandom permutations." *Proceedings of the Fourth Annual Conference on Computer and Communications Security*, ACM, 1997.
20. J. Patarin, "About Feistel schemes with six (or more) rounds." *Proceedings of the 5th Fast Software Encryption Workshop*, Lecture Notes in Computer Science Vol. 1372, Springer-Verlag, 1998.
21. E. Petrank and C. Rackoff, "CBC MAC for real-time data sources." Manuscript, available at `http://philby.ucsd.edu/cryptolib.html`, 1997.
22. J. Pieprzyk, "How to construct pseudorandom permutations from single pseudorandom functions." *Advances in Cryptology – Eurocrypt* 90 *Proceedings*, Lecture Notes in Computer Science Vol. 473, I. Damgård ed., Springer-Verlag, 1990 pp. 140–150.
23. R. Rivest, "All-or-nothing encryption and the package transform." *Proceedings of the 4th Fast Software Encryption Workshop*, Lecture Notes in Computer Science Vol. 1267, Springer-Verlag, 1997.

24. D. Wagner, B. Schneier and J. Kelsey, "Cryptanalysis of the cellular message encryption algorithm." *Advances in Cryptology – Crypto* 97 *Proceedings*, Lecture Notes in Computer Science Vol. 1294, B. Kaliski ed., Springer-Verlag, 1997, pp. 526–537.
25. Y. Zheng, T. Matsumoto and H. Imai, "Impossibility results and optimality results on constructing pseudorandom permutations." *Advances in Cryptology – Eurocrypt* 89 *Proceedings*, Lecture Notes in Computer Science Vol. 434, J-J. Quisquater, J. Vandewille ed., Springer-Verlag, 1989.