New Paradigms for Constructing Symmetric Encryption Schemes Secure Against Chosen-Ciphertext Attack

Anand Desai

Department of Computer Science & Engineering,
University of California at San Diego,
9500 Gilman Drive, La Jolla, California 92093, USA.
adesai@cs.ucsd.edu

Abstract. The paradigms currently used to realize symmetric encryption schemes secure against adaptive chosen ciphertext attack (CCA) try to make it infeasible for an attacker to forge "valid" ciphertexts. This is achieved by either encoding the plaintext with some redundancy before encrypting or by appending a MAC to the ciphertext. We suggest schemes which are provably secure against CCA, and yet *every* string is a "valid" ciphertext. Consequently, our schemes have a smaller ciphertext expansion than any other scheme known to be secure against CCA. Our most efficient scheme is based on a novel use of "variable-length" pseudorandom functions and can be efficiently implemented using block ciphers. We relate the difficulty of breaking our schemes to that of breaking the underlying primitives in a precise and quantitative way.

1 Introduction

Our goal in this paper is to design efficient symmetric (ie. private-key) encryption schemes that are secure against adaptive chosen-ciphertext attack (CCA). Rather than directly applying the paradigm used in designing public-key encryption schemes secure against CCA, we develop new ones which take advantage of the peculiarities of the symmetric setting. As a result we manage to do what may not have been known to be possible: constructing encryption schemes secure against CCA wherein every string of appropriate length is a "valid" ciphertext and has a corresponding plaintext. The practical significance of this is that our schemes have a smaller ciphertext expansion than that of any other scheme known to be secure against CCA.

1.1 Privacy under Chosen-Ciphertext Attack

The most basic goal of encryption is to ensure that an adversary does not learn any useful information from the ciphertexts. The first rigorous formalizations of this goal were described for the public-key setting by Goldwasser and Micali [15]. Their goal of indistinguishability for public-key encryption has been considered

under attacks of increasing severity: chosen-plaintext attack, and two kinds of chosen-ciphertext attacks [20, 23]. The strongest of these attacks, due to Rackoff and Simon, is known as the adaptive chosen-ciphertext attack (referred to as CCA in this work). Under this attack, the adversary is given the ability to obtain plaintexts of ciphertexts of its choice (with the restriction that it not ask for the decryption of the "challenge" ciphertext itself). The combination of the goal of indistinguishability and CCA gives rise to a very strong notion of privacy, known as IND-CCA. A second goal, called non-malleability, introduced by Dolev, Dwork and Naor [13], can also be considered in this framework. This goal formalizes the inability of an adversary given a challenge ciphertext to modify it into another, in such a way that the underlying plaintexts are somehow "meaningfully related". The notion of indistinguishability under chosen-plaintext attack was adapted to the symmetric setting by Bellare, Desai, Jokipii and Rogaway [2]. Their paradigm of giving the adversary "encryption oracles" can be used to "lift" any of the notions for the public-key setting to the symmetric setting. Studies on relations among the various possible notions have established that IND-CCA implies all these other notions in the public-key setting [3, 13], as well as, in the symmetric setting [16].

Symmetric encryption schemes are widely used in practice and form the basis of many security protocols used on the Internet. The use of schemes secure in the IND-CCA sense is often mandated by the way they are to be used in these protocols. Consequently, there has been an increasing focus on designing encryption schemes that are secure in this strong sense. A commonly used privacy mechanism is to use a public-key encryption scheme to send session keys and then use these keys and a symmetric encryption scheme to actually encrypt the data. This method is attractive since symmetric encryption is significantly more efficient than its public-key counterpart. The security of such "hybrid" encryption schemes is as weak as its weakest link. In particular, if we want a hybrid encryption scheme secure in the IND-CCA sense, then we must use a symmetric encryption scheme that is also secure in the IND-CCA sense.

Barring a few exceptions, most of the recent work on encryption has concentrated on the public-key setting alone. The prevailing intuition seems to be that the ideas from the public-key setting "extend" to the symmetric setting. Indeed there are many cases where this is true, and there are often paradigms in one setting that have a counterpart in the other. However, this viewpoint ignores the important differences in the settings and we usually pay for this in terms of efficiency. We take a direct approach to the problem of designing symmetric encryption schemes secure in the IND-CCA sense. For practical reasons, we are particularly interested in block-cipher-based schemes (ie. encryption modes).

1.2 Our Paradigms

We describe two new paradigms for realizing symmetric encryption schemes secure against CCA: the Unbalanced Feistel paradigm and the Encode-then-Encipher paradigm.

UNBALANCED FEISTEL. Our first paradigm is described in terms of "variable-length" pseudorandom functions. These extend the notion of "fixed-length" pseudorandom functions (PRFs) introduced by Bellare, Kilian and Rogaway [4] so as to model block ciphers. A variable-length input pseudorandom function (VI-PRF) is a function that takes inputs of any pre-specified length or of variable length and produces an output of some fixed length. A variable-length output pseudorandom function (VO-PRF), on the other hand, is a function whose *output* can be of some pre-specified length or of variable length. The input consists of a fixed-length part and a part specifying the length of the required output.

Our paradigm is illustrated in Figure 2. It is interesting that there is a similarity between our scheme and the "simple probabilistic encoding scheme" used by Bellare and Rogaway in their OAEP scheme [7]. Their encoding scheme is defined as: $M \oplus G(r) \| r \oplus H(M \oplus G(r))$, where M is the message to be encrypted, r is a randomly chosen quantity, G is a "generator" random oracle and H is a "hash function" random oracle. They show that applying a trapdoor permutation, such as RSA, to such an encoded string constitutes asymmetric encryption secure against chosen-plaintext attack. One can view our scheme as the above "encoding scheme" with G replaced by a VO-PRF and H replaced by a VI-PRF. We show that this alone constitutes symmetric encryption secure against CCA.

Constructions for VI-PRFs and VO-PRFs could be based on one-way or trapdoor functions. For practical reasons, we are more interested in constructions that can be based on more efficient cryptographic primitives. Some efficient constructions of VI-PRFs based on PRFs are the CBC-MAC variant analyzed by Petrank and Rackoff [22] and the "three-key" variants of Black and Rogaway [10]. We give a simple and efficient construction of a VO-PRF from a PRF. See Figure 1. There could be many other ways of instantiating VO-PRFs using ideas from the constructions of VI-PRFs and "key-derivation" functions.

We give a quantitative analysis of our scheme to establish its security against CCA. Our analysis relates the difficulty of breaking the scheme to that of breaking the underlying VI-PRF and VO-PRF. We also give a quantitative security analysis of our VO-PRF example. The security of the VI-PRF examples have already been established by similar analyses, as discussed earlier.

We give a concrete example instantiating the Unbalanced Feistel paradigm using a block cipher. The encryption is done in two steps. In the first step, we encrypt the plaintext M to a string $C \| r$ using a modified form of the counter mode of encryption. Here the "counter" r is picked to be random and we have $|C| = |M|$. In the second step, we mask r by XORing it with a modified form of a CBC-MAC on C to get a string σ. The ciphertext output is $C \| \sigma$.

ENCODE-THEN-ENCIPHER. This is a rather well-known (but not particularly well-understood) method of encrypting. Recent work by Bellare and Rogaway [8] has tried to remedy this by giving a precise treatment of this idea. Encryption, in this paradigm, is a process in which the plaintext is first "encoded" and then sent through a secret-keyed length-preserving permutation in a process known as "enciphering". The privacy of the resulting encryption schemes for different security interpretations of "encoding" and "enciphering" are given in [8]. We

concentrate in this paper on one particular combination of the encoding and enciphering interpretations that was not considered in [8]. We consider "encoding" of a message to be simply the message with some randomness appended to it. We take "enciphering" to mean the application of a variable-length input super-pseudorandom permutation (VI-SPRP). We show that with these meanings, the Encode-then-Encipher paradigm yields symmetric encryption schemes that are secure against CCA. Note that a super-pseudorandom permutation (SPRP) [18] alone will not do since we need a permutation that can work with variable and arbitrary length inputs. Also, the very efficient constructions of Naor and Reingold [19] cannot be used here since they are not "full-fledged" VI-SPRPs. The problem of constructing VI-SPRPs has been explored by Bleichenbacher and Desai [11] and Patel et al. [21]. See Section 4 for more details. The encryption schemes resulting from this paradigm are quite practical, but given the current state-of-art, this approach does not match the Unbalanced Feistel paradigm for efficiency.

1.3 Related Work and Discussion

The idea behind the paradigms currently used in practice for designing encryption schemes secure in the IND-CCA sense is to make it infeasible to create a "valid" ciphertext (unless the ciphertext was created by encrypting some known plaintext). The intuition is that doing this makes the decryption access ability all but useless. There are a couple of different methods used in symmetric encryption based on this idea.

Alternate Paradigms. The most commonly used approach of getting security in the IND-CCA sense is to authenticate ciphertexts using a message authentication code (MAC). Bellare and Namprempre have shown that of the various possible ways of composing a generic MAC and a generic symmetric encryption scheme, the one consisting of first encrypting the plaintext and then appending to the result a MAC of the result, is the only one that is secure in the IND-CCA sense [5]. Another approach is to add some known redundancy to the plaintext before encrypting. The idea is that most strings of the length of the ciphertext will be "invalid" and that they will be recognized as such, since their "decryption" will not have the expected redundancy. A recently suggested encryption mode, the RPC mode of Katz and Yung [17], uses this idea. Yet another approach that uses this idea is to apply a VI-SPRP to plaintexts that are encoded with randomness *and* redundancy [8].

Comparisons. An unavoidable consequence of the paradigms used by the methods above is that the ciphertexts generated are longer than those possible using schemes that are only secure against chosen-plaintext attack. In particular, they are longer by the size of the output of the MAC or by the amount of redundancy used. To begin with, we have that any secure encryption scheme will be length-increasing. For short plaintexts these increases in the length of the ciphertext can be a significant overhead. Avoiding any overhead other than that absolutely necessary would also be useful in any environment where bandwidth

is at a premium. In our approach, we avoid the part of the overhead due to the MAC or redundancy. The ciphertext expansion due to the randomness used is unavoidable in the model we consider (ie. where the sender and receiver do not share any state other than the key).

We point out that the methods above achieve something more than privacy against CCA. They achieve privacy as well as *integrity*. There are many levels of integrity that one can consider (see [5, 17]). The strongest one exactly coincides with the idea used by the methods above. Namely, that it be infeasible to create a "valid" (new) ciphertext. This is clearly not achievable by our method or by any other where every string of appropriate length corresponds to some plaintext. A slightly weaker form of integrity requires that it be infeasible to create a (new) ciphertext such that something may be known about the underlying plaintext. Our methods can be shown to have this integrity property. Should the strongest integrity property be required, we could encode the plaintexts with some redundancy and then apply our paradigm. This would mean losing some of its advantages but it would still be a competitive alternative. These claims are substantiated in the full version of this paper [12].

2 Preliminaries

We adopt a standard notation with respect to probabilistic algorithms and sets. If $A(\cdot, \cdot, \ldots)$ is a probabilistic algorithm then $x \leftarrow A(x_1, x_2, \ldots)$ denotes the experiment of running A on inputs $x_1, x_2, \ldots$ and letting x be the outcome. Similarly, if A is a set then $x \leftarrow A$ denotes the experiment of selecting a point uniformly from A and assigning x this value.

Symmetric Encryption. A *symmetric encryption scheme*, $\Pi = (\mathcal{E}, \mathcal{D}, \mathcal{K})$, is a three-tuple of algorithms where:

- $\mathcal{K}$ is a randomized *key generation* algorithm. It returns a key a; we write $a \leftarrow \mathcal{K}$.
- $\mathcal{E}$ is a randomized or stateful *encryption* algorithm. It takes the key a and a *plaintext* x and returns a *ciphertext* y; we write $y \leftarrow \mathcal{E}_a(x)$.
- $\mathcal{D}$ is a deterministic *decryption* algorithm. It takes a key a and string y and returns either the corresponding plaintext x or the symbol $\perp$; we write $x \leftarrow \mathcal{D}_a(y)$ where $x \in \{0,1\}^* \cup \perp$.

We require that $\mathcal{D}_a(\mathcal{E}_a(x)) = x$ for all $x \in \{0,1\}^*$.

Security Against Chosen-Ciphertext Attack. The formalization we give is an adaptation of the "find-then-guess" definition of Bellare et al. [2] so as to model adaptive chosen-ciphertext attack in the sense of Rackoff and Simon [23]. In the indistinguishability of encryptions under chosen-ciphertext attack the adversary A is imagined to run in two phases. In the find phase, given adaptive access to an encryption *and* decryption oracle, A comes up with a pair of messages x_0, x_1 along with some state information s to help in the second phase. In the guess phase, given the encryption y of one of the messages and s, it must

identify which of the two messages goes with y. A may not use its decryption oracle on y in the guess phase.

Definition 1. [IND-CCA] *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme. For an adversary A and $b = 0, 1$ define the experiment*

Experiment $\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, b)$
$a \leftarrow \mathcal{K}$; $(x_0, x_1, s) \leftarrow A^{\mathcal{E}_a, \mathcal{D}_a}(\mathsf{find})$; $y \leftarrow \mathcal{E}_a(x_b)$; $d \leftarrow A^{\mathcal{E}_a, \mathcal{D}_a}(\mathsf{guess}, y, s)$;
Return d.

It is mandated that $|x_0| = |x_1|$ above and that A does not query $\mathcal{D}_a(\cdot)$ on ciphertext y in the guess *phase. Define the advantage of A and the advantage function of Π respectfully, as follows:*

$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(A) = \Pr[\,\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, 0) = 0\,] - \Pr[\,\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, 1) = 0\,]$$
$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(t, q_e, q_d, \mu, \nu) = \max_{A} \{\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(A)\,\}$$

where the maximum is over all A with "time-complexity" t, making at most q_e encryption oracle queries and at most q_d decryption oracle queries, these together totalling at most μ bits and choosing $|x_0| = |x_1| = \nu$ bits. ∎

Here the "time-complexity" is the worst case total execution time of experiment $\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, b)$ plus the size of the code of A, in some fixed RAM model of computation. This convention is used for other definitions in this paper, as well.

3 Unbalanced Feistel Encryption

We begin with a block-cipher-based instantiation of the Unbalanced Feistel paradigm. A reader interested in seeing the paradigm first may skip this example and go to Section 3.2, without any loss of understanding. A security analysis for this paradigm is given in Section 3.3.

3.1 A Concrete Example

Our starting point is a block cipher $F : \{0,1\}^k \times \{0,1\}^l \mapsto \{0,1\}^l$. The scheme $\Pi[F] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ has a key generation algorithm $\mathcal{K}$ that specifies a key $K = (K1 \| K2 \| K3 \| K4) \leftarrow \{0,1\}^{4k}$, partitioned into 4 equal pieces. We have:

Algorithm $\mathcal{E}_{K1\|K2\|K3\|K4}(M)$

(1) Let $r \leftarrow \{0,1\}^l$ be a random initial vector.
(2) Let $s = F_{K1}(r)$.
(3) Let P be the first $|M|$ bits of $F_{K2}(s+1) \| F_{K2}(s+2) \| F_{K2}(s+3) \| \cdots$.
(4) Let $C = P \oplus M$.
(5) Let $\mathsf{pad} = 10^m$ such that m is the smallest integer making $|C| + |\mathsf{pad}|$ divisible by l.

(6) Parse $C\|\mathsf{pad}$ as $C_1 \dots C_n$ such that $|C_i| = l$ for all $1 \leq i \leq n$.

(7) Let $C'_0 = 0^l$, and let $C'_i = F_{K3}(C'_{i-1} \oplus C_i)$ for all $1 \leq i \leq n-1$.

(8) Let $\sigma = r \oplus F_{K4}(C'_{n-1} \oplus C_n)$

(9) Return ciphertext $C\|\sigma$.

Algorithm $\mathcal{D}_{K1\|K2\|K3\|K4}(C'')$

(1) Parse C'' as $C\|\sigma$ such that $|\sigma| = l$.

(2) Let $\mathsf{pad} = 10^m$ such that m is the smallest integer making $|C| + |\mathsf{pad}|$ divisible by l.

(3) Parse $C\|\mathsf{pad}$ as $C_1 \dots C_n$ such that $|C_i| = l$ for all $1 \leq i \leq n$.

(4) Let $C'_0 = 0^l$, and let $C'_i = F_{K3}(C'_{i-1} \oplus C_i)$ for all $1 \leq i \leq n-1$.

(5) Let $r = \sigma \oplus F_{K4}(C'_{n-1} \oplus C_n)$

(6) Let $s = F_{K1}(r)$.

(7) Let P be the first $|C|$ bits of $F_{K2}(s+1)\|F_{K2}(s+2)\|F_{K2}(s+3)\| \cdots$.

(8) Let $M = P \oplus C$.

(9) Return plaintext M.

This example can be seen as having two stages. In the first stage we encrypt the plaintext M to a string $C\|r$ using a modified form of the counter mode. Here r is the randomness used to encrypt and $|C| = |M|$, even though $|M|$ may not be an integral multiple of the block length l. In the second stage we run C through a modified form of the CBC-MAC and XOR this value with r to get σ. The ciphertext is defined to be $C\|\sigma$. Although we do not make r a part of the ciphertext, it is still possible to retrieve it if the secret key is known. Thus the scheme is invertible. While the other counter mode variants can easily be shown to be insecure against CCA, the claim is that "masking" r in this manner, makes our mode secure against CCA.

The difference between the standard (randomized) counter mode, analyzed by Bellare et al. [2], and the variant we use here is that instead of using r directly, we use a block-cipher "encrypted" value of r. This has the effect of neutralizing simple attacks where there may be some "control" over r. We cannot use the standard "one-key" CBC-MAC in our construction, given that it is known to be secure only on fixed-length messages (with this length being an integral multiple of the block length) [4]. Instead we use a "two-key" CBC-MAC, wherein the last block of the message is processed by the block cipher with an independent key. This is a variant of a construction analyzed by Petrank and Rackoff [22] which first computes a regular CBC-MAC on the entire message and then applies a block-cipher with an independent key to the output of the CBC-MAC. We also use a standard padding method with our MAC, since $|M|$ (and hence $|C|$) may not be an integral multiple of the block length. Note that we use the padding method in a way that does not cause an increase in the length of the ciphertext.

3.2 The General Approach

We begin with a description of the primitives used to realize the general scheme and some definitions to understand the security claims.

Fixed-Length Pseudorandom Functions. A fixed-length (finite) function family is a keyed multi-set F of functions where all the functions have the same domain and range. To pick a function f from family F means to pick a key a, uniformly from the key space of F, and let $f = F_a$. A family F has input length l and output length L if each $f \in F$ maps $\{0,1\}^l$ to $\{0,1\}^L$. We let $\mathrm{Func}(l)$ denote a reference family consisting of all functions with input length l and output length l. A function $f \leftarrow \mathrm{Func}(l)$ is defined as follows: for each $M \in \{0,1\}^l$, let $f(M)$ be a random string in $\{0,1\}^l$.

A finite function family F is pseudorandom if the input-output behavior of F_a is indistinguishable from the behavior of a random function of the same domain and range. This is formalized via the notion of distinguishers [14]. Our concrete security formalization is that of [4].

Definition 2. [PRF] *Let $F : \mathcal{K} \times \{0,1\}^l \mapsto \{0,1\}^l$ be a function. For a distinguisher A and $b = 0, 1$ define the experiment*

Experiment $\mathsf{Exp}^{\mathrm{prf}}_F(A, b)$
$a \leftarrow \mathcal{K}$; $\mathcal{O}_0 \leftarrow F_a$; $\mathcal{O}_1 \leftarrow \mathrm{Func}(l)$; $d \leftarrow A^{\mathcal{O}_b}$; Return d.

Define the advantage of A and the advantage function of F respectfully, as follows:

$$\mathsf{Adv}^{\mathrm{prf}}_F(A) = \Pr[\,\mathsf{Exp}^{\mathrm{prf}}_F(A,0) = 0\,] - \Pr[\,\mathsf{Exp}^{\mathrm{prf}}_F(A,1) = 0\,]$$

$$\mathsf{Adv}^{\mathrm{prf}}_F(t,q) = \max_A \{\mathsf{Adv}^{\mathrm{prf}}_F(A)\}$$

where the maximum is over all A with time complexity t and making at most q oracle queries. ∎

Variable-Length Input Pseudorandom Functions. These functions take an input of variable and arbitrary length and produce a fixed-length output. We define a reference family VI-Func(l). A random variable-length input function $h \leftarrow \text{VI-Func}(l)$ is defined as follows: for each $M \in \{0,1\}^*$, let $h(M)$ be a random string in $\{0,1\}^l$.

Definition 3. [VI-PRF] *Let $H : \mathcal{K} \times \{0,1\}^* \mapsto \{0,1\}^l$ be a function. For a distinguisher A and $b = 0, 1$ define the experiment*

Experiment $\mathsf{Exp}^{\text{vi-prf}}_H(A, b)$
$a \leftarrow \mathcal{K}$; $\mathcal{O}_0 \leftarrow H_a$; $\mathcal{O}_1 \leftarrow \text{VI-Func}(l)$; $d \leftarrow A^{\mathcal{O}_b}$; Return d.

Define the advantage of A and the advantage function of H respectfully, as follows:

$$\mathsf{Adv}^{\text{vi-prf}}_H(A) = \Pr[\,\mathsf{Exp}^{\text{vi-prf}}_H(A,0) = 0\,] - \Pr[\,\mathsf{Exp}^{\text{vi-prf}}_H(A,1) = 0\,]$$

$$\mathsf{Adv}^{\text{vi-prf}}_H(t,q,\mu) = \max_A \{\mathsf{Adv}^{\text{vi-prf}}_H(A)\}$$

where the maximum is over all A with time complexity t and making at most q oracle queries, these totalling at most μ bits. ▌

Many of the variable-length input MAC constructions are VI-PRFs. Our "two-key" CBC-MAC, discussed earlier, is such an example. The security of this construction follows from that of the CBC-MAC variant analyzed by Petrank and Rackoff [22]. Black and Rogaway have suggested several constructions of VI-PRFs that are computationally more efficient than this one [10]. There are efficient variable-length input MAC constructions, such as the protected counter sum construction of Bernstein [9] and the cascade construction of Bellare et al. [1] that are not strictly VI-PRF due to their probabilistic nature, but which could be used in their place in our paradigm.

Variable-Length Output Pseudorandom Functions. These are functions

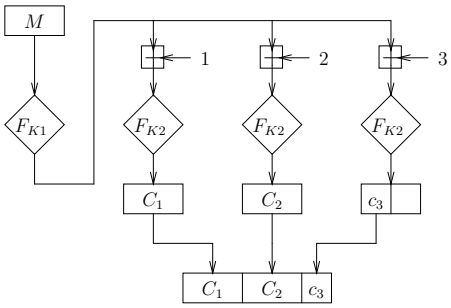

Fig. 1. *The* XORG *function.*

that can generate an output of arbitrary and variable length. We think of a function from a VO-PRF family as taking two inputs: a fixed-length binary string and a unary string, and producing an output of a size specified by the unary input. We define a reference family VO-Func(l). A random variable-length output function $g \leftarrow$ VO-Func(l) is defined as follows. For each $M \in \{0,1\}^l$ let $R_l(M)$ be a random string in $\{0,1\}^\infty$. Then for each $M \in \{0,1\}^l$ and $L \in 1^*$, let $g(M\|L)$ be the first $|L|$ bits of $R_l(M)$. One can think of a "random variable-length output function" as a process that answers a query $M\|L$ as follows: if M is "new" then return a random element $C \in \{0,1\}^{|L|}$; if M has already appeared in a past query $M\|L'$ (to which the response was C) and $|L'| \leq |L|$ then return the first $|L'|$ bits of C; and if M has already appeared in a past query $M\|L'$ (to which the response was C) and $|L'| > |L|$ then return $C\|C'$ where $C' \leftarrow \{0,1\}^{|L'|-|L|}$.

Definition 4. [VO-PRF] *Let $G : \mathcal{K} \times \{0,1\}^l \times 1^* \mapsto \{0,1\}^*$ be a function. For a distinguisher A and $b = 0, 1$ define the experiment*

Experiment $\mathsf{Exp}_G^{\text{vo-prf}}(A, b)$
$a \leftarrow \mathcal{K}$; $\mathcal{O}_0 \leftarrow G_a$; $\mathcal{O}_1 \leftarrow$ VO-Func; $d \leftarrow A^{\mathcal{O}_b}$; Return d.

Define the advantage of A and the advantage function of G respectfully, as follows:

$$\mathsf{Adv}^{\text{vo-prf}}_{G}(A) = \Pr[\,\mathsf{Exp}^{\text{vo-prf}}_{G}(A,0) = 0\,] - \Pr[\,\mathsf{Exp}^{\text{vo-prf}}_{G}(A,1) = 0\,]$$
$$\mathsf{Adv}^{\text{vo-prf}}_{G}(t,q,\mu) = \max_{A}\{\mathsf{Adv}^{\text{vo-prf}}_{G}(A)\,\}$$

where the maximum is over all A with time complexity t and making at most q queries, these totalling at most μ bits. ▮

A somewhat similar (but weaker) primitive is implicit in the counter mode of operation. Hence this is a good starting point for constructing full-fledged VO-PRFs. The result is a construction we call XORG. See Figure 1 for a picture. Let F be a block cipher and $a = (K1\|K2)$ be a key specifying permutations F_{K1} and F_{K2}. Then for any $L \in 1^*$ and $M \in \{0,1\}^l$, the output of XORG is defined as the first $|L|$ bits of $F_{K2}(M'+1)\|F_{K2}(M'+2)\|F_{K2}(M'+3)\|\cdots$, where $M' = F_{K1}(M)$. A security analysis of XORG is given in Section 3.3.

The UFE Scheme. We now describe our general scheme $\text{UFE}[G,H] = (\mathcal{K},\mathcal{E},\mathcal{D})$

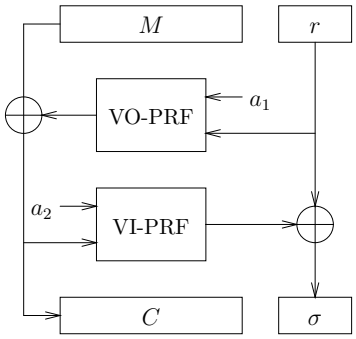

Fig. 2. *The* UFE *scheme.*

where $G\colon \mathcal{K}_{\text{vo-prf}} \times \{0,1\}^l \times 1^* \mapsto \{0,1\}^*$ is a VO-PRF and $H\colon \mathcal{K}_{\text{vi-prf}} \times \{0,1\}^* \mapsto \{0,1\}^l$ is a VI-PRF. The key generation algorithm $\mathcal{K} = \mathcal{K}_{\text{vo-prf}} \times \mathcal{K}_{\text{vi-prf}}$ specifies a key $a = a_1\|a_2$ where $a_1 \leftarrow \mathcal{K}_{\text{vo-prf}}$; $a_2 \leftarrow \mathcal{K}_{\text{vi-prf}}$. The encryption and decryption algorithms are defined as:

Algorithm $\mathcal{E}_{a_1\|a_2}(M)$	Algorithm $\mathcal{D}_{a_1\|a_2}(C')$
$r \leftarrow \{0,1\}^l$	parse C' as $C\|\sigma$ where $\|\sigma\| = l$
$C \leftarrow M \oplus G_{a_1}(r)$	$r \leftarrow \sigma \oplus H_{a_2}(C)$
$\sigma \leftarrow r \oplus H_{a_2}(C)$	$M \leftarrow C \oplus G_{a_1}(r)$
return $C\|\sigma$	return M

A picture for the UFE scheme is given in Figure 2. We analyze the security of this scheme in Section 3.3.

3.3 Analysis

We begin with an analysis of our VO-PRF example. See Figure 1. The theorem says that XORG is secure as a VO-PRF as long as the underlying PRF is secure.

Theorem 1. [Security of XORG] *Let* $G = \mathrm{XORG}[F]$ *where* $F = \mathrm{PRF}(l)$. *Then,*

$$\mathsf{Adv}^{\text{vo-prf}}_{G}(t, q, \mu) \leq 2 \cdot \mathsf{Adv}^{\text{prf}}_{F}(t', q') + \frac{2(q-1)(q+\frac{\mu}{l})}{2^l}$$

where $t' = t + \mathcal{O}(q + \mu + l)$ *and* $q' = \lceil \frac{\mu}{l} \rceil + q$.

Proof. Let A be an adversary attacking G in the VO-PRF sense, and let t, q, μ be the resources associated with $\mathsf{Exp}^{\text{vo-prf}}_{G}(A, b)$. Let $\mathcal{K}_{\text{prf}}$ be the key generation algorithm of F.

We assume without loss of generality that A does not repeat queries. (A query consists of a string $M \in \{0,1\}^l$ and a string $L \in 1^*$. Our assumption is that A picks a different M for each query). We consider various probabilities related to running A under different experiments:

$$p_1 = \Pr[\, K1, K2 \leftarrow \mathcal{K}_{\text{prf}} :\ A^{G_{K1\|K2}} = 1\,]$$
$$p_2 = \Pr[\, f \leftarrow \mathrm{Func}(l);\ K2 \leftarrow \mathcal{K}_{\text{prf}} :\ A^{G^{f}_{K2}} = 1\,]$$
$$p_3 = \Pr[\, f, h \leftarrow \mathrm{Func}(l) :\ A^{G^{f,h}} = 1\,]$$
$$p_4 = \Pr[\, g \leftarrow \text{VO-Func}(l) :\ A^{g} = 1\,]$$

The notation above is as follows: In the experiment defining p_2, A's oracle, on query M and $L \in 1^*$ responds by returning the first $|L|$ bits of $F_{K2}(M'+1)\|F_{K2}(M'+2)\|F_{K2}(M'+3)\|\cdots$, where $M' = f(M)$. In the experiment defining p_3, A's oracle, on query M and $L \in 1^*$ responds by returning the first $|L|$ bits of $h(M'+1)\|h(M'+2)\|h(M'+3)\|\cdots$, where $M' = f(M)$.

We want to upper bound $\mathsf{Adv}^{\text{vo-prf}}_{G}(A) = p_1 - p_4$. We do this in steps.

Our first claim is that $p_1 - p_2 \leq \mathsf{Adv}^{\text{prf}}_{F}(t', q)$.

Consider the following distinguisher D for F. It has an oracle $\mathcal{O} :\ \{0,1\}^l \mapsto \{0,1\}^l$. It picks $K2 \leftarrow \mathcal{K}_{\text{prf}}$ and runs A. When A makes a query M and $L \in 1^*$, it returns $G^{\mathcal{O}}_{K2}(M)$ as the answer. D outputs whatever A outputs at the end. It is clear that $\mathsf{Adv}^{\text{prf}}_{F}(D) = p_1 - p_2$. The claim follows.

Next we show that $p_2 - p_3 \leq \mathsf{Adv}^{\text{prf}}_{F}(t', q')$.

Consider the following distinguisher D for F. It has an oracle $\mathcal{O} :\ \{0,1\}^l \mapsto \{0,1\}^l$. It simulates $f \leftarrow$ Func and runs A. When A makes a query M and $L \in 1^*$, it returns $G^{f,\mathcal{O}}(M)$ as the answer. For any query $M_i\|L_i$ of A, D must make $\lceil \frac{L_i}{l} \rceil$ queries to $\mathcal{O}$. D outputs whatever A outputs at the end. It follows that $\mathsf{Adv}^{\text{prf}}_{F}(D) = p_2 - p_3$.

Finally, we show that $p_3 - p_4 \leq \frac{2(q-1)\cdot(q+\frac{\mu}{l})}{2^l}$.

We introduce some more notation to justify this. For any integer t let $[t] = \{1, \cdots, t\}$. Let $(M_1, C_1), \ldots, (M_q, C_q)$ be the transcript of A's interaction with its oracle, where for $i \in [q]$, (M_i, C_i) represents an oracle query M_i (such that $|M_i| = l$) and $L_i \in 1^*$ and the response C_i (such that $|C_i| = |L_i|$). Let $n_i = \lceil \frac{L_i}{l} \rceil$ for $i \in [q]$. Let AC ($\mathsf{Adversary\ is\ Correct}$) be the event that A correctly guesses whether the oracle is $G^{f,h}$ or g, where these are as defined in the experiments underlying p_3 and p_4. In answering the i-th query $M_i \| L_i$, the oracle computes $r_i \leftarrow f(M_i)$ and applies a random function h to the n_i strings $r_i+1, \ldots, r_i+n_i \in \{0,1\}^l$. We call these strings the i-th *sequence*, and $r_i + k$ is the k-th point in this sequence, where $k \in [n_i]$.

Let Bad be event that $(r_i + k = r_j + k')$ for some $(i,k) \neq (j,k')$, and $(i, j \in [q]) \wedge (k \in [n_i]) \wedge (k' \in [n_j])$. That is Bad is the event that there are overlapping sequences. We have

$$\begin{aligned}\Pr[\,\mathsf{AC}\,] &= \Pr[\,\mathsf{AC} \mid \overline{\mathsf{Bad}}\,] \cdot \Pr[\,\overline{\mathsf{Bad}}\,] + \Pr[\,\mathsf{AC} \mid \mathsf{Bad}\,] \cdot \Pr[\,\mathsf{Bad}\,] \\ &\leq \Pr[\,\mathsf{AC} \mid \overline{\mathsf{Bad}}\,] + \Pr[\,\mathsf{Bad}\,]\end{aligned}$$

Given the event $\overline{\mathsf{Bad}}$, we have that, in replying to $M_i \| L_i$, the output is randomly and uniformly distributed over $\{0,1\}^{|L_i|}$. It follows that $\Pr[\,\mathsf{AC} \mid \overline{\mathsf{Bad}}\,] = \frac{1}{2}$.

Next, we bound $\Pr[\,\mathsf{Bad}\,]$. For $i \in [q]$, let Bad_i be the event that r_i causes event Bad. We have

$\Pr[\,\mathsf{Bad}\,] \leq \Pr[\,\mathsf{Bad}_1\,] + \sum_{i=2}^{q} \Pr\left[\,\mathsf{Bad}_i \mid \overline{\mathsf{Bad}}_{i-1}\,\right]$. By definition, $\Pr[\,\mathsf{Bad}_1\,] = 0$.

Since we are assuming that $M_i \neq M_j$ for any $(i \neq j) \wedge (i, j \in [q])$, we have that r_i is randomly and uniformly distributed in $\{0,1\}^l$. We observe that the chance of overlapping sequences is maximized if all the $i-1$ previous queries resulted in $i-1$ sequences that were no less than $n_i - 1$ blocks apart. We have a collision if the i-th sequence begins in a block that is $n_i - 1$ blocks before any other previous query j or in a block occupied by that sequence j. We have that for $i > 1$,

$$\Pr\left[\,\mathsf{Bad}_i \mid \overline{\mathsf{Bad}}_{i-1}\,\right] \leq \frac{\sum_{j=1}^{i-1}(n_j + n_i - 1)}{2^l} \;=\; \frac{(i-1)(n_i-1) + \sum_{j=1}^{i-1} n_j}{2^l}$$

Continuing,

$$\begin{aligned}\Pr[\,\mathsf{Bad}\,] &\leq \sum_{i=2}^{q} \Pr\left[\,\mathsf{Bad}_i \mid \overline{\mathsf{Bad}}_{i-1}\,\right] \\ &\leq \sum_{i=2}^{q} \frac{(i-1)(n_i-1) + \sum_{j=1}^{i-1} n_j}{2^l} \;\leq\; \frac{(q-1)(q+\frac{\mu}{l})}{2^l} \\ p_3 - p_4 \leq 2 \cdot \Pr[\,\mathsf{AC}\,] - 1 \;&\leq\; 2 \cdot \Pr[\,\mathsf{Bad}\,] \;\leq\; \frac{2(q-1)(q+\frac{\mu}{l})}{2^l}\end{aligned}$$

Using the above bounds and that $\mathsf{Adv}^{\text{vo-prf}}_{G}(A) = p_1 - p_4 = (p_1 - p_2) + (p_2 - p_3) + (p_3 - p_4)$, we get the claimed result. ∎

We next turn to the security of our general scheme. We first establish the security of UFE assuming the underlying primitives are ideal.

Lemma 1. [Upper bound on insecurity of UFE **using random functions]** *Let* $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be the scheme* UFE$[G, H]$ *where* $G =$ VO-Func(l) *and* $H =$ VI-Func(l). *Then for any* t, μ, ν,

$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(t, q_e, q_d, \mu, \nu) \leq \delta_{\Pi} \stackrel{\text{def}}{=} \frac{(q_e + q_d)(q_e + q_d + 1)}{2^l}$$

Proof. Let A be an adversary attacking Π in the IND-CCA sense, and let t, q_e, q_d, μ, ν be the resources associated with $\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, b)$. We show that,

$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(A) \stackrel{\text{def}}{=} 2 \cdot \Pr[\,\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, b) = b\,] - 1 \;\leq\; \frac{(q_e + q_d)(q_e + q_d + 1)}{2^l}$$

We refer to the event $\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, b) = b$ as event AC ($\mathsf{Adversary\ is\ Correct}$). In the rest of the proof we will freely refer to random variables from $\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, b)$.

Let $g \leftarrow G$ and $h \leftarrow H$ be the variable-length output function and variable-length input function, respectively, specified by the key a in the experiment.

We assume without loss of generality that A does not make "redundant" decryption oracle queries. That is, we are assuming that A does not ask a decryption query v if it had already made the query v to its decryption oracle, or if it had obtained v in response to some earlier encryption oracle query. Note that since encryption is probabilistic, A may want to repeat encryption oracle queries.

Let q be the total number of distinct plaintext-ciphertext pairs resulting from A's interaction with its oracles. The following inequality holds: $q \leq q_e + q_d$. For simplicity, we assume that this is an equality. That is each query results in a unique plaintext-ciphertext pair. We are interested in an upper bound for $\Pr[\,\mathsf{AC}\,]$, and this assumption only increases this probability.

For any integer t let $[t] = \{1, \cdots, t\}$.

Let $(M_1, C_1 \| \sigma_1), \ldots, (M_k, C_k \| \sigma_k), \ldots, (M_{q+1}, C_{q+1} \| \sigma_{q+1})$ be plaintext and ciphertext pairs, such that for $(i \in [q+1]) \wedge (i \neq k)$, $(M_i, C_i \| \sigma_i)$ represents an oracle query and the corresponding response. We have that A picks plaintexts x_0, x_1 such that $|x_0| = |x_1| = \nu$ at the end of the find stage and receives $y \leftarrow \mathcal{E}_a(x_b)$ for some $b \in \{0, 1\}$. Let $M_k = x_b$ and $C_k \| \sigma_k = y$ where $|\sigma_k| = l$.

Let r_i be the l-bit IV associated to $(M_i, C_i \| \sigma_i)$, for $i \in [q+1]$.

Let Bad be event that $r_i = r_j$ for some $(i, j \in [q+1]) \wedge (i \neq j)$. We have

$$\begin{aligned} \Pr[\,\mathsf{AC}\,] &= \Pr[\,\mathsf{AC} \mid \overline{\mathsf{Bad}}\,] \cdot \Pr[\,\overline{\mathsf{Bad}}\,] + \Pr[\,\mathsf{AC} \mid \mathsf{Bad}\,] \cdot \Pr[\,\mathsf{Bad}\,] \\ &\leq \Pr[\,\mathsf{AC} \mid \overline{\mathsf{Bad}}\,] + \Pr[\,\mathsf{Bad}\,] \end{aligned}$$

Given the event $\overline{\mathsf{Bad}}$, we have that, in computing y, the output of G is randomly and uniformly distributed over $\{0, 1\}^{\nu}$. Since this value is XORed with x_b, it follows that $\Pr[\,\mathsf{AC} \mid \overline{\mathsf{Bad}}\,] = \frac{1}{2}$. Next, we turn to a bound for $\Pr[\,\mathsf{Bad}\,]$.

For $i \in [q+1]$, let Bad_i be the event that r_i causes event Bad. We have $\Pr[\,\mathsf{Bad}\,] \leq \Pr[\,\mathsf{Bad}_1\,] + \sum_{i=2}^{q+1} \Pr\left[\,\mathsf{Bad}_i \mid \overline{\mathsf{Bad}}_{i-1}\,\right]$. By definition, $\Pr[\,\mathsf{Bad}_1\,] = 0$. We will consider A's view just before it makes its i-th query, for $i \in [q+1]$. (If $i = k$, then we take the "query" to be an encryption query x_b). Let us assume that this includes the knowledge that the event $\overline{\mathsf{Bad}}_{i-1}$ holds. Now depending on the nature of A's i-th query, there are two cases we can consider: either $(M_i, C_i \| \sigma_i)$ results from an encryption query M_i or from a decryption query $C_i \| \sigma_i$.

First we consider the case of the i-th query being an encryption query. The IV r_i, in this case, will be randomly and uniformly distributed in $\{0,1\}^l$. We have that the chance of a collision is at most $\frac{i-1}{2^l}$. Next we consider the case of the i-th query being a decryption query.

For $(i \in [q+1]) \wedge (i > 1)$, consider A's view just before it makes its i-th query. We know that this includes $(M_1, C_1 \| \sigma_1), \ldots, (M_{i-1}, C_{i-1} \| \sigma_{i-1})$. However, given $\overline{\mathsf{Bad}}_{i-1}$, we claim that $h(C_j)$, for any $1 \leq j < i$, is information theoretically hidden in A's view. With $\overline{\mathsf{Bad}}_{i-1}$, the only potential ways A can learn something about $h(C_j)$ is through σ_j or r_j. However we have that r_j never becomes a part of A's view (the IV is not returned in a decryption query). And we have $\sigma_j = r_j \oplus h(C_j)$. Since r_j is unknown, we have that σ_j does not leak any information about $h(C_j)$.

There are two sub-cases we can consider (when the i-th query is a decryption query). The first sub-case is that $C_i \neq C_j$, for all $1 \leq j < i$. Since h is being invoked on a "new" string, the value $h(C_i)$ will be randomly and uniformly distributed in $\{0,1\}^l$. Consequently, r_i will also be randomly and uniformly distributed in $\{0,1\}^l$. As in the previous case, we have for $i > 1$, the chance of a collision to be at most $\frac{i-1}{2^l}$.

The other sub-case is that $C_i = C_j$, for some $1 \leq j < i$. We want to bound the probability of A picking a σ_i that causes Bad_i. We know that A cannot pick $\sigma_i = \sigma_j$, since we are assuming that it does not make redundant queries. Moreover, we know that in A's view, the value of $h(C_k)$ is information theoretically hidden, for any $1 \leq k < i$. Hence A's only strategy in picking a σ_i that causes a collision (other than choosing the value σ_j) can be to guess a value. It follows that A's chances of causing a collision are smaller in this sub-case than if it had picked a new C_i. So here too, we have for $i > 1$, the chance of a collision to be at most $\frac{i-1}{2^l}$.

Continuing,

$$\Pr[\,\mathsf{Bad}\,] \leq \sum_{i=2}^{q+1} \Pr\left[\,\mathsf{Bad}_i \mid \overline{\mathsf{Bad}}_{i-1}\,\right] \;\leq\; \sum_{i=2}^{q+1} \frac{i-1}{2^l} \;\leq\; \frac{q(q+1)}{2 \cdot 2^l}$$

Using this in the bound for $\Pr[\,\mathsf{AC}\,]$ and doing a little arithmetic we get the claimed result. ∎

The actual security of UFE is easily derived given Lemma 1.

Theorem 2. [Security of UFE] *Let* $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be the encryption scheme* UFE$[G, H]$ *where* $G = \text{VO-PRF}(l)$ *and* $H = \text{VI-PRF}(l)$*. Then,*

$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(t, q_e, q_d, \mu, \nu) \leq \mathsf{Adv}^{\text{vo-prf}}_{G}(t', q', \mu') + \mathsf{Adv}^{\text{vi-prf}}_{H}(t', q', \mu') + \delta_{\Pi}$$

where $t' = t + \mathcal{O}(\mu + \nu + lq_e + lq_d)$ *and* $q' = q_e + q_d$ *and* $\mu' = \mu + \nu$ *and* $\delta_{\Pi} \stackrel{\text{def}}{=} \frac{(q_e+q_d)(q_e+q_d+1)}{2^l}$.

Proof. Lemma 1 says that $\Pi[\text{VO-Func}, \text{VI-Func}]$ is secure. The intuition is that this implies that $\Pi[G, H]$ is secure, since otherwise it would mean that G is not secure as a VO-PRF or that H is not secure as a VI-PRF. Formally, we prove it using a contradiction argument.

Let A be an adversary attacking $\Pi[G, H]$ in the IND-CCA sense. Let t, q_e, q_d, μ, ν be the resources associated with $\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, b)$.

We will run A under different experiments. We will refer to these experiments as $\mathsf{Exp}^{\text{ind-cca}}_{\Pi_1}(A, b)$ and $\mathsf{Exp}^{\text{ind-cca}}_{\Pi_2}(A, b)$ and $\mathsf{Exp}^{\text{ind-cca}}_{\Pi_3}(A, b)$ where $\Pi_1 = \Pi[G, H]$ and $\Pi_2 = \Pi[\text{VO-Func}, H]$ and $\Pi_3 = \Pi[\text{VO-Func}, \text{VI-Func}]$. We will also refer to the corresponding advantage functions, which will follow the natural notation and interpretation, given the above. We are interested in an upper bound for $\mathsf{Adv}^{\text{ind-cca}}_{\Pi_1}(t, q_e, q_d, \mu, \nu)$. We do this in steps.

Our first claim is

$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi_1}(t, q_e, q_d, \mu, \nu) \;\leq\; \mathsf{Adv}^{\text{ind-cca}}_{\Pi_2}(t, q_e, q_d, \mu, \nu) + \mathsf{Adv}^{\text{vo-prf}}_{G}(t', q', \mu')$$

Consider the following distinguisher D for G. It has an oracle $\mathcal{O} : \{0,1\}^l \times 1^* \mapsto \{0,1\}^*$. It first picks a key for H that specifies a function h. It then runs A answering A's oracle queries as follows. If A makes an encryption oracle query M, then it picks a random $r \in \{0,1\}^l$ and makes a query $r \| 1^{|M|}$ to $\mathcal{O}$. It then takes the response P and computes $C = M \oplus P$. It returns to A as its response the string $C \| (r \oplus h(C))$. Similarly to a decryption query $C \| \sigma$, it returns the string $C \oplus \mathcal{O}(\sigma \oplus h(C))$. Note that it is important that D is able to correctly do the encryption and decryption using its oracle. It simulates the experiment defining the advantage of A. If A is successful in the end, then it guesses that $\mathcal{O}$ was from VO-Func, otherwise it guesses that it was from VO-PRF.

We get $\mathsf{Adv}^{\text{vo-prf}}_{G}(D) = \mathsf{Adv}^{\text{ind-cca}}_{\Pi_1}(A) - \mathsf{Adv}^{\text{ind-cca}}_{\Pi_2}(A)$. One can check that the number of queries q' made by D is at most $q_e + q_d$. Also the length μ' of all of D's queries is at most the sum of the length μ of all the queries of A and the length ν of the challenge that D has to prepare for A. This proves the claim.

The next claim is

$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi_2}(t, q_e, q_d, \mu, \nu) \;\leq\; \mathsf{Adv}^{\text{ind-cca}}_{\Pi_3}(t, q_e, q_d, \mu, \nu) + \mathsf{Adv}^{\text{vi-prf}}_{H}(t', q', \mu')$$

We can construct a distinguisher D for H along similar lines as above. The main difference is that D must simulate a random function from VO-Func in its simulation for A. We omit the details to prove this claim.

Combining our claims and substituting the bound for $\mathsf{Adv}^{\text{ind-cca}}_{\Pi_3}(t, q_e, q_d, \mu, \nu)$ from Lemma 1, we get the claimed result. ∎

4 Encode-then-Encipher Encryption

Bellare and Rogaway show that if messages are encoded with randomness and redundancy and then enciphered with a VI-SPRP then the resulting scheme is secure in a sense that implies security in the IND-CCA sense [8]. We show in this section that to achieve security in just the IND-CCA sense, the redundancy is unnecessary.

Variable-length Input Super-Pseudorandom Permutations. These are permutations that take an input of variable and arbitrary length and produce an output of the same length. We define a reference family VI-Perm. A random variable-length input permutation $(f, f^{-1}) \leftarrow$ VI-Perm is defined as follows: for each number i, let f_i be a random permutation on $\{0,1\}^i$, and for each $M \in \{0,1\}^*$, let $f(M) = f_i(M)$, where $i = |M|$. Let f^{-1} be the inverse of f.

Definition 5. [VI-SPRP] *Let $S: \mathcal{K} \times \{0,1\}^* \mapsto \{0,1\}^*$ be a permutation. For a distinguisher A and $b = 0, 1$ define the experiment*

Experiment $\mathsf{Exp}^{\text{vi-sprp}}_S(A, b)$
$a \leftarrow \mathcal{K}$; $\mathcal{O}_0, \mathcal{O}_0^{-1} \leftarrow S_a, S_a^{-1}$; $\mathcal{O}_1, \mathcal{O}_1^{-1} \leftarrow$ VI-Perm; $d \leftarrow A^{\mathcal{O}_b, \mathcal{O}_b^{-1}}$; Return d.

Define the advantage of A and the advantage function of S respectfully, as follows:

$$\mathsf{Adv}^{\text{vi-sprp}}_S(A) = \Pr[\,\mathsf{Exp}^{\text{vi-sprp}}_S(A,0) = 0\,] - \Pr[\,\mathsf{Exp}^{\text{vi-sprp}}_S(A,1) = 0\,]$$

$$\mathsf{Adv}^{\text{vi-sprp}}_S(t, q_e, q_d, \mu) = \max_A \{\mathsf{Adv}^{\text{vi-sprp}}_S(A)\}$$

where the maximum is over all A with time complexity t and making at most q_e queries to S_a and q_d queries to S_a^{-1}, these together totalling at most μ bits. ∎

Constructions of these "full-fledged" pseudorandom permutations are relatively rare. Naor and Reingold show how to efficiently construct an SPRP that can work with any large input-length given an SPRP (or a PRF) of some fixed smaller input-length [19]. However, their constructions cannot work with inputs of arbitrary and variable length, and it is unclear how they can be extended to do so. Bleichenbacher and Desai suggest a construction for a VI-SPRP using a block cipher (modeled as an SPRP) that has a cost of about three applications of the block cipher per message block [11]. Patel, Ramzan and Sundaram have a construction that is computationally less expensive than this but wherein the key-length varies with the message-length [21].

The EEE Scheme. We now describe the scheme $\text{EEE}[S] = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where $S: \mathcal{K}_{\text{vi-sprp}} \times \{0,1\}^* \mapsto \{0,1\}^*$ is a VI-SPRP. The key generation algorithm $\mathcal{K} = \mathcal{K}_{\text{vi-sprp}}$ specifies a key a. For any positive integer l, the encryption and decryption algorithms are defined as:

Algorithm $\mathcal{E}_a(M)$	Algorithm $\mathcal{D}_a(C)$
$r \leftarrow \{0,1\}^l$	if $\lvert C\rvert \le l$ then $M \leftarrow \perp$ else
$C \leftarrow S_a(M \Vert r)$	$(M \Vert r) \leftarrow S_a^{-1}(C)$ where $\lvert r\rvert = l$
return C	return M

We give the security of this scheme next.

Theorem 3. [Security of EEE] *Let* $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ *be the encryption scheme* EEE[S] *where* $S =$ VI-SPRP. *Then,*

$$\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(t, q_e, q_d, \mu, \nu) \leq 2 \cdot \mathsf{Adv}^{\text{vi-sprp}}_{S}(t', q'_e, q'_d, \mu') + \frac{2(q_e + q_d)}{2^l}$$

where $t' = t + \mathcal{O}(\mu + \nu + lq_e + lq_d)$ *and* $q'_e = q_e + 1$ *and* $q'_d = q_d$ *and* $\mu' = \mu + \nu$.

Proof. Let A be an adversary attacking Π in the IND-CCA sense, and let t, q_e, q_d, μ, ν be the resources associated with $\mathsf{Exp}^{\text{ind-cca}}_{\Pi}(A, b)$.

We assume without loss of generality that A does not make "redundant" queries to its decryption oracle. That is, we are assuming that A does not ask a decryption oracle query v if it had already made the query v to its decryption oracle, or if it had obtained v in response to some earlier encryption oracle query.

Our goal is to bound $\mathsf{Adv}^{\text{ind-cca}}_{\Pi}(A, b)$. To this end we introduce an algorithm D.

Algorithm D is a distinguisher for S. It is given oracles for permutations f, f^{-1} that are either from a VI-SPRP family or from the random family VI-Perm. It runs A, answering A's queries as follows: If A makes an encryption oracle query M then D picks $r \leftarrow \{0,1\}^l$ and computes $C \leftarrow f(M\|r)$. It returns C as the response to the query. If A makes a decryption oracle query C then D first checks if $|C| \leq l$. If it is then D returns "invalid" as its response. Otherwise it computes $(M\|r) \leftarrow f^{-1}(C)$ (where $|r| = l$) and returns M as the response to the query. A eventually stops (at the end its find stage) and outputs (x_0, x_1, s). D then chooses $d \leftarrow \{0,1\}$ and $r_0 \leftarrow \{0,1\}^l$ and computes $y \leftarrow f(x_d\|r_0)$. (If D has already queried on this point before or ever received this in response to some previous decryption oracle query, then it does not have to use its oracle to compute y). D then runs A with the parameters (guess, y, s), answering A's oracle queries as before. When A terminates, D checks to see if it was correct. If it was, then D guesses that its oracles were "real", otherwise it guesses that they were "random".

We develop some notation to simplify the exposition of the analysis. Let $\Pr_1[\,\cdot\,]$ denote a probability in the probability space where the oracles given to D are "real". Similarly, let $\Pr_0[\,\cdot\,]$ denote a probability in the probability space where the oracles given to D are "random". We will suppress showing explicit access to the oracles since they will be obvious from context. We have

$$\mathsf{Adv}^{\text{vi-sprp}}_{S}(D) \stackrel{\text{def}}{=} \Pr_1[\,D = 1\,] - \Pr_0[\,D = 1\,]$$

From the description of D, we see that $\Pr_1[\,D = 1\,]$ is exactly the probability of A being correct in an experiment defining the advantage in the IND-CCA sense. Thus we get,

$$\Pr_1[\,D = 1\,] \;=\; \frac{1}{2} + \frac{1}{2} \cdot \mathsf{Adv}^{\text{ind-cca}}_{\Pi}(A)$$

Next we upper bound $\mathrm{Pr}_0[\,D=1\,]$. Let Coll be the event that there is a collision of one of the nonces resulting from A's queries with the one in the challenge. More precisely, Coll is the event that $\exists i \in [q_e + q_d] :\ r_i = r_0$.

$$\begin{aligned}\mathrm{Pr}_0[\,D=1\,] &= \mathrm{Pr}_0\,[\,D=1 \mid \mathsf{Coll}\,]\cdot \mathrm{Pr}_0[\,\mathsf{Coll}\,] + \mathrm{Pr}_0\left[\,D=1 \mid \overline{\mathsf{Coll}}\,\right]\cdot \mathrm{Pr}_0[\,\overline{\mathsf{Coll}}\,] \\ &\le \mathrm{Pr}_0[\,\mathsf{Coll}\,] + \mathrm{Pr}_0\left[\,D=1 \mid \overline{\mathsf{Coll}}\,\right]\end{aligned}$$

Since the permutations underlying $\mathrm{Pr}_0[\,\cdot\,]$ are "random", we get

$$\mathrm{Pr}_0[\,\mathsf{Coll}\,] \le \frac{q_e+q_d}{2^l};\ \mathrm{Pr}_0\left[\,D=1 \mid \overline{\mathsf{Coll}}\,\right] \;=\; \frac{1}{2}$$

Using the above to lower bound the advantage of D and completing the argument in the standard way, we get the claimed result. ∎

Acknowledgements

This paper benefited a great deal from help and advice received from Mihir Bellare. Many of the ideas and motivation for the problem considered here came out of collaboration with Daniel Bleichenbacher. I would also like to thank Sara Miner, Chanathip Namprempre, Bogdan Warinschi and the CRYPTO 2000 program committee for their very helpful comments.

The author was supported in part by Mihir Bellare's 1996 Packard Foundation Fellowship in Science and Engineering and NSF CAREER Award CCR-9624439.

References

1. M. Bellare, R. Canetti and H. Krawczyk, "Pseudorandom functions revisited: The cascade construction and its concrete security," *Proceedings of the 37th Symposium on Foundations of Computer Science*, IEEE, 1996.
2. 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.
3. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology - Crypto '98*, LNCS Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
4. M. Bellare, J. Kilian and P. Rogaway, "The security of the cipher block chaining message authentication code," *Advances in Cryptology - Crypto '94*, LNCS Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.
5. M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," Report 2000/025, *Cryptology ePrint Archive*, `http://eprint.iacr.org/`, May 2000.
6. M. Bellare and P. Rogaway, "Entity authentication and key distribution," *Advances in Cryptology - Crypto '93*, LNCS Vol. 773, D. Stinson ed., Springer-Verlag, 1993.

7. M. Bellare and P. Rogaway, "Optimal asymmetric encryption: How to encrypt with RSA," *Advances in Cryptology - Eurocrypt '95*, LNCS Vol. 921, L. Guillou and J. Quisquater ed., Springer-Verlag, 1995.
8. M. Bellare and P. Rogaway, "Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography," Manuscript, December 1998, available from authors.
9. D. Bernstein, "How to stretch random functions: The security of protected counter sums," *J. of Cryptology*, Vol. 12, No. 3, 1999.
10. J. Black and P. Rogaway, "CBC MACs for Arbitrary Length Messages: The Three-Key Constructions," *Advances in Cryptology - Crypto '00*, LNCS Vol. ??, M. Bellare ed., Springer-Verlag, 2000.
11. D. Bleichenbacher and A. Desai, "A construction of a super-pseudorandom cipher," Manuscript, May 1999, available from authors.
12. A. Desai, "New paradigms for constructing symmetric encryption schemes secure against chosen-ciphertext attack," Full version of this paper, available via: `http://www-cse.ucsd.edu/users/adesai/`.
13. D. Dolev, C. Dwork and M. Naor, "Non-malleable cryptography," *SIAM J. of Computing*, to appear. Preliminary version in *Proceedings of the 23rd Annual Symposium on the Theory of Computing*, ACM, 1991.
14. O. Goldreich, S. Goldwasser and S. Micali, How to construct random functions. *Journal of the ACM*, Vol. 33, N0. 4, 1986, pp. 210-217.
15. S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Science*, Vol. 28, 1984, pp. 270–299.
16. J. Katz and M. Yung, "Complete characterization of security notions for probabilistic private-key encryption," *Proceedings of the 32nd Annual Symposium on the Theory of Computing*, ACM, 2000.
17. J. Katz and M. Yung, "Unforgeable Encryption and Adaptively Secure Modes of Operation," *Fast Software Encryption '00*, LNCS Vol. ??, B. Schneier ed., Springer-Verlag, 2000.
18. M. Luby and C. Rackoff, "How to construct pseudorandom permutations from pseudorandom functions," *SIAM J. Computing*, Vol. 17, No. 2, April 1988.
19. M. Naor and O. Reingold, "On the construction of pseudorandom permutations: Luby-Rackoff revisited," *J. of Cryptology*, Vol. 12, No. 1, 1999.
20. M. Naor and M. Yung, "Public-key cryptosystems provably secure against chosen-ciphertext attackss," *Proceedings of the 22nd Annual Symposium on the Theory of Computing*, ACM, 1990.
21. S. Patel, Z. Ramzan, and G. Sundaram, "Efficient Variable-Input-Length Cryptographic Primitives," Manuscript, 2000.
22. E. Petrank and C. Rackoff, "CBC MAC for Real-Time Data Sources," *Dimacs Technical Report*, 97-26, 1997.
23. C. Rackoff and D. Simon, "Non-interactive zero-knowledge proof of knowledge and chosen-ciphertext attack," *Advances in Cryptology - Crypto '91*, LNCS Vol. 576, J. Feigenbaum ed., Springer-Verlag, 1991.