Does Encryption with Redundancy Provide Authenticity?

Jee Hea An and Mihir Bellare

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

Abstract. A popular paradigm for achieving privacy plus authenticity is to append some "redundancy" to the data before encrypting. We investigate the security of this paradigm at both a general and a specific level. We consider various possible notions of privacy for the base encryption scheme, and for each such notion we provide a condition on the redundancy function that is necessary and sufficient to ensure authenticity of the encryption-with-redundancy scheme. We then consider the case where the base encryption scheme is a variant of CBC called NCBC, and find sufficient conditions on the redundancy functions for NCBC encryption-with-redundancy to provide authenticity. Our results highlight an important distinction between public redundancy functions, meaning those that the adversary can compute, and secret ones, meaning those that depend on the shared key between the legitimate parties.

1 Introduction

The idea that authenticity can be easily obtained as a consequence of the privacy conferred by encryption has long attracted designers. Encryption-with-redundancy is the most popular paradigm to this end. Say that parties sharing key K are encrypting data via some encryption function $\mathcal{E}$. (Typically this is some block cipher mode of operation.) To obtain authenticity, the sender computes some function h of the data M to get a "checksum" $\tau = h(M)$. [1] It then computes a ciphertext $C \leftarrow \mathcal{E}_K(M\|\tau)$ and sends C to the receiver. The latter decrypts to get $M\|\tau$ and then checks whether $\tau = h(M)$. If not, it rejects the ciphertext as unauthentic.

The attraction of the paradigm is clear: the added cost of providing authenticity is small, amounting to computation of the checksum function plus perhaps one or two extra block cipher invocations in order to encrypt the now longer message. (Designers attempt to use simple and fast checksum functions.) However, the paradigm has a poor security record. For example, using CBC encryption with the checksum being the XOR of the message blocks (called CBCC) was proposed by the U.S. National Bureau of Standards, and was subsequently found

[1] Other names for the checksum include MDC —Manipulation Detection Code— and "redundancy," whence the name of the paradigm.

B. Pfitzmann (Ed.): EUROCRYPT 2001, LNCS 2045, pp. 512–528, 2001.

to not provide authenticity, as discussed in [23,16]. If the encryption algorithm is an additive stream cipher (e.g. CTR-mode encryption) where the adversary knows the plaintext, a forgery attacks by [15,16] apply. An attack attributed to Wagner on a large class of CBC-mode encryption-with-redundancy schemes is described in [24].

1.1 General Results

The many and continuing efforts to achieve authenticity via the encryption-with-redundancy paradigm point to the existence of some intuition that leads designers to think that it should work. The intuition appears to be that the privacy conveyed by the encryption makes attacks on the integrity harder. The first goal of our work is to assess the correctness of this intuition, and the security of the paradigm, at a general level. We are not concerned so much with the security of specific schemes as with trying to understand how the authenticity of the encryption-with-redundancy scheme relates to the security properties of the underlying primitives and to what extent the paradigm can be validated at a general level.

We denote the base encryption scheme by $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$. (It is specified by its key-generation, encryption, and decryption algorithms.) We are general with regard to the form of the redundancy computation method, allowing it to be key-based. A choice of method is given by a *redundancy code* $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ where $\mathcal{K}_r$ is an algorithm responsible for generating a key K_r while $\mathcal{H}$ takes K_r and the text M to return the redundancy or checksum $\tau = \mathcal{H}_{K_r}(M)$. Associated to $\mathcal{SE}$ and $\mathcal{RC}$ is the encryption-with-redundancy scheme $\mathcal{ER}$ in which one encrypts message M via $C \leftarrow \mathcal{E}_{K_e}(M \| \mathcal{H}_{K_r}(M))$. Upon receipt of ciphertext C, the receiver applies $\mathcal{D}_K$ to get back $M \| \tau$ and accepts iff $\tau = \mathcal{H}_{K_r}(M)$. Here K_e is the (secret) encryption key for $\mathcal{SE}$.

We distinguish *public redundancy* and *secret redundancy*. In the first case, K_r is public information. ($\mathcal{H}_{K_r}(\cdot)$ might be a public hash function like SHA-1, or simply return the XOR of the message blocks.) In this case, K_r is known to the adversary, who is thus capable of computing the redundancy function. In the case of secret redundancy, K_r is part of the secret key shared between the parties. (It might for example be a key for a universal hash function [11] or a message authentication code.) In this case the key K_r is not given to the adversary.

The desired authenticity property of the encryption-with-redundancy scheme $\mathcal{ER}$ is integrity of ciphertexts [7,19,8]: it should be computationally infeasible for an adversary to produce a ciphertext that is valid but different from any created by the sender.

We allow the assumed privacy attribute of the base encryption scheme to range across the various well-established notions of privacy used in the literature: IND-CPA, NM-CPA, IND-CCA. (Indistinguishability under chosen-plaintext attack [13,4], non-malleability under chosen-plaintext attack [12], and indistinguishability under chosen-ciphertext attack, respectively. Recall that non-malle-

Type of base encryption	Condition on redundancy code	
	For public redundancy	For secret redundancy
IND-CPA	None	None
NM-CPA	None	UF-NMA
IND-CCA	None	UF-NMA

Fig. 1. For each possible privacy attribute SSS-AAA of the base encryption scheme, we indicate a condition on the redundancy code that is *necessary and sufficient* for it to be integrity-providing with respect to SSS-AAA. We distinguish the cases where the redundancy is public (anyone can compute it) and secret (depends on the shared secret key). "None" means that the corresponding class of redundancy codes is empty: *No* redundancy code is integrity-providing.

ability under chosen-ciphertext attack is equivalent to IND-CCA [5,18] so we don't need to consider it separately.)

We say that a redundancy code $\mathcal{RC}$ is *integrity-providing with respect to security notion SSS-AAA* if for *all* base encryption schemes $\mathcal{SE}$ that are SSS-AAA secure, the encryption-with-redundancy scheme $\mathcal{ER}$ obtained from $\mathcal{SE}$ and $\mathcal{RC}$ is secure in the sense of integrity of ciphertexts. (This property of a redundancy code is attractive from the design viewpoint, since a redundancy code having this property may be used in conjunction with *any* SSS-AAA-secure base encryption scheme, and authenticity of the resulting encryption-with-redundancy scheme is guaranteed.) The question we ask is the following. Given a notion of security SSS-AAA, what security attribute of the redundancy code $\mathcal{RC}$ will ensure that $\mathcal{RC}$ is integrity-providing with respect to security notion SSS-AAA?

We find that an important distinction to be made in answering this question is whether or not the redundancy computation is secret-key based. Figure 1 summarizes the results we expand on below.

Encryption with public redundancy. We show that there is *no* choice of public redundancy code $\mathcal{RC}$ which is integrity-providing with respect to notions of security IND-CPA, NM-CPA or IND-CCA. This is a powerful indication that the intuition that privacy helps provide integrity via encryption-with-redundancy is wrong in the case where the adversary can compute the redundancy function.

This conclusion is not surprising when the base encryption scheme meets only a weak notion of privacy like IND-CPA. But one might have thought that there are redundancy codes for which a condition like NM-CPA on the base encryption scheme would suffice to prove integrity of ciphertexts for the resulting encryption-with-redundancy scheme. Not only is this false, but it stays false when the base encryption scheme has even a stronger privacy attribute like IND-CCA.

Note that the most popular methods for providing redundancy are public, typically involving computing a keyless checksum of the message, and our result applies to these.

The result is proved by giving an example of a base encryption scheme meeting the notion of privacy in question such that for any redundancy code the corresponding encryption with public redundancy scheme can be attacked. (This assumes there exists some base encryption scheme meeting the notion of privacy in question, else the issue is moot.)

Encryption with secret redundancy. As Figure 1 indicates, allowing the computation of the redundancy to depend on a secret key does not help if the base encryption scheme meets only a weak notion of privacy like IND-CPA— *no* secret redundancy code is integrity-providing with respect to IND-CPA.

However secret redundancy does help if the base encryption scheme has stronger privacy attributes. We characterize the requirement on the redundancy code in this case. We say that it is UF-NMA (UnForgeable under No-Message Attack) if it is a MAC for which forgery is infeasible for an adversary that is not allowed to see the MACs of any messages before it must output its forgery. Our result is that this condition on the redundancy code is necessary and sufficient to ensure that it is integrity-providing with respect to NM-CPA and IND-CCA.

We stress that UF-NMA is a very weak security requirement, so the implication is that allowing the redundancy computation to depend on a secret key greatly increases security as long as the base encryption scheme is strong enough. We also stress that our condition on the redundancy code is both necessary and sufficient. Still in practice, the implication is largely negative because standard modes of operation do not meet notions like NM-CPA or IND-CCA.

Perspective. The above results do not rule out obtaining secure schemes from the encryption-with-redundancy paradigm. The results refer to the ability to prove authenticity of the encryption-with-redundancy scheme *in general*, meaning based *solely* on assumed privacy attributes of the base encryption scheme and attributes of the redundancy code.

One might consider encryption with some specific redundancy code using as base encryption scheme a block cipher based mode of operation that is only IND-CPA secure, and yet be able to prove authenticity by analyzing the encryption-with-redundancy scheme directly based on the assumption that the block cipher is a pseudorandom permutation. This would not contradict the above results. What the above results do is show that the intuition that privacy helps integrity is flawed. Encryption-with-redundancy might work, but not for that reason. If a specific scheme such as the example we just mentioned works, it is not because of the privacy provided by the encryption, but, say, because of the pseudorandomness of the block cipher. In practice this tell us that to get secure encryption-with-redundancy schemes we must look at specific constructions and analyze them directly. This is what we do next.

1.2 Encryption with NCBC

We consider a variant of (random-IV) CBC mode encryption in which the enciphering corresponding to the last message block is done under a key different from that used for the other blocks. We call this mode NCBC. Here we are able to obtain positive results for both public and secret redundancy functions.

We show that if secret redundancy is used, quite simple and efficient redundancy codes suffice for the NCBC with redundancy scheme to provide authenticity. The redundancy code should satisfy the property called *AXU (Almost Xor Universal)* in [20,25]. (Any Universal-2 function [27] has this property and there are other efficient constructs as well [14,10,2].) On the other hand we show that if the redundancy is public, then authenticity of the NCBC with redundancy scheme is guaranteed if the redundancy code is *XOR-collision-resistant*. (The latter, a cryptographic property we define, can be viewed either as a variant of the standard collision-resistance property, or as an extension of the AXU property to the case where the key underlying the function is public.) These results assume the underlying block cipher is a strong pseudorandom permutation in the sense of [22].

These results should be contrasted with what we know about encryption with redundancy using the standard CBC mode as the base encryption scheme. Wagner's attack, pointed out in [24], implies that *no* public redundancy code will, in conjunction with CBC encryption, yield an encryption-with-redundancy scheme possessing integrity of ciphertexts. In the case where the redundancy is secret, Krawczyk [21] shows that it suffices for the redundancy code to be a MAC secure against chosen-message attack, but this is a strong condition on the redundancy code compared to the AXU property that suffices for NCBC. Thus, the simple modification consisting of enciphering under a different key for the last block substantially enhances CBC with regard to its ability to provide authenticity under the encryption-with-redundancy paradigm.

1.3 Related Work

Preneel gives an overview of existing authentication methods [24] that includes much relevant background. A comprehensive treatment of authenticated encryption —the goal of joint privacy and authenticity— is provided in [7]. They relate different notions of privacy and authenticity to compare their relative strengths.

Encryption-with-redundancy is one of many approaches to the design of authenticated encryption schemes. Another general approach is "generic composition:" combine an encryption scheme with a MAC in some way. This is analyzed in [7], who consider the following generic composition methods: *Encrypt-and-mac, Mac-then-encrypt, Encrypt-then-mac.* For each of these methods they consider two notions of integrity, namely integrity of ciphertexts and a weaker notion of integrity of plaintexts, and then, assuming the base encryption scheme is IND-CPA and the MAC is secure against chosen-message attack, indicate whether or not the method has the integrity property in question. Krawczyk's recent work [21] considers the same methods from the point of view of building "secure channels" over insecure networks. The drawback of the generic composition approach compared to the encryption-with-redundancy approach is that some MACs might be less efficient than redundancy codes, and that public redundancy avoids the additional independent key that is required for MACs.

Another general paradigm is "encode then encipher" [8] —add randomness and redundancy and then encipher rather than encrypt. Encode then encipher requires a variable-input length strong pseudorandom permutation, which can be relatively expensive to construct.

Let $SNCBC[F, \mathcal{RC}]$ denote NCBC encryption with block cipher F and secret redundancy provided by an efficient AXU redundancy code $\mathcal{RC}$. We compare this to other authenticated encryption schemes such as RPC mode [19], IACBC [17], and OCB [26]. RPC is computation and space inefficient compared to all the other methods. IACBC and OCB have cost comparable to that of $SNCBC[F, \mathcal{RC}]$, but OCB is parallelizable.

Encryption-with-redundancy is one of many approaches to simultaneously achieving privacy and authenticity. Our goal was to analyze and better understand this approach. We do not suggest it is superior to other approaches.

2 Definitions

A string is a member of $\{0,1\}^*$. We denote by "$\|$" an operation that combines several strings into one in such a way that the constituent strings are uniquely recoverable from the final one. (If lengths of all strings are fixed and known, concatenation will serve the purpose.) The empty string is denoted ε.

Extended encryption schemes. The usual syntax of a symmetric encryption scheme (cf. [4]) is that encryption and decryption depend on a key shared between sender and receiver but not given to the adversary. We wish to consider a setting where operations depend, in addition to the shared key, on some public information, such as a hash function. The latter may be key based. (Think of the key as having been chosen at random at design time and embedded in the hash function.) All parties including the adversary have access to this key, which we call the *common key*. We need to model it explicitly because security depends on the random choice of this key even though it is public. This requires a change in encryption scheme syntax. Accordingly we define an *extended encryption scheme* which extends the usual symmetric encryption scheme by addition of another key generation algorithm. Specifically an extended encryption scheme $\mathcal{EE} = (\mathcal{K}_c, \mathcal{K}_s, \mathcal{E}, \mathcal{D})$ consists of four algorithms as follows. The randomized *common key generation* algorithm $\mathcal{K}_c$ takes input a security parameter $k \in \mathsf{N}$ and in time poly(k) returns a key K_c; we write $K_c \xleftarrow{R} \mathcal{K}_c(k)$. The randomized *secret key generation* algorithm $\mathcal{K}_s$ also takes input $k \in \mathsf{N}$ and in time poly(k) returns a key K_s; we write $K_s \xleftarrow{R} \mathcal{K}_s(k)$. We let $K = (K_c, K_s)$. The *encryption* algorithm $\mathcal{E}$ is either randomized or stateful. It takes K and a *plaintext* M and in time poly$(k, |M|)$ returns a *ciphertext* $C = \mathcal{E}_K(M)$; we write $C \xleftarrow{R} \mathcal{E}_K(M)$. (If randomized, it flips coins, anew upon each invocation. If stateful, it maintains a state which it updates upon each invocation.) The deterministic and stateless *decryption* algorithm $\mathcal{D}$ takes the key K and a string C and in time poly$(k, |C|)$ returns either the corresponding plaintext M or the distinguished symbol $\perp$; we write $x \leftarrow \mathcal{D}_K(C)$. We require that $\mathcal{D}_K(\mathcal{E}_K(M)) = M$ for all $M \in \{0,1\}^*$.

Notice that it is not apparent from the syntax why there are two keys because they are treated identically. The difference will surface when we consider security: we will view the legitimate users as possessing K_s while both they and the adversary have K_c. (It also surfaces in something we don't consider explicitly here, which is a multi-user setting. In that case, although K_s will be generated anew for each pair of users, K_c may be the same across the whole system.)

A standard symmetric encryption scheme, namely one where there is no common key, can be recovered as the special case where the common key generation algorithm $\mathcal{K}_c$ returns the empty string. Formally, we say that $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is a (symmetric) encryption scheme if $\mathcal{EE} = (\mathcal{K}_c, \mathcal{K}, \mathcal{E}, \mathcal{D})$ is an extended encryption scheme where $\mathcal{K}_c$ is the algorithm which on any input returns the empty string. When the common key K_c is the empty string we may also omit it in the input given to the adversary.

Notions of security. Notions of security for symmetric encryption schemes are easily adapted to extended encryption schemes by giving the adversary the common key as input. Via the formal definitions shown below and this discussion we will summarize the definitions we need.

We let $\mathcal{EE} = (\mathcal{K}_c, \mathcal{K}_s, \mathcal{E}, \mathcal{D})$ be the extended encryption scheme whose security we are defining. The formalizations, given in Definition 1 and Definition 2, associate to each notion of security and each adversary an *experiment*, and based on that, an *advantage*. The latter is a function of the security parameter that measures the success probability of the adversary. Asymptotic notions of security result by asking this function to be negligible for adversaries of time complexity polynomial in the security parameter. Concrete security assessments can be made by associating to the scheme another advantage function that for each value of the security parameter and given resources for an adversary returns the maximum, over all adversaries limited to the given resources, of the advantage of the adversary.

Note that these definitions apply to standard symmetric encryption schemes too, since as per our conventions the latter are simply the special case of extended encryption schemes in which the common key generation algorithm returns the empty string.

Privacy. The basic and weakest natural notion of privacy is IND-CPA. We use one of the formalizations of [4] which adapts that of [13] to the symmetric setting. A challenge bit b is chosen, the adversary is given K_c, and can query, adaptively and as often as it likes, the left-or-right encryption oracle. The adversary wins if it can guess b. For IND-CCA the adversary gets in addition a decryption oracle but loses if it queries it on any ciphertext returned by the left-or-right encryption oracle.

Non-malleability captures, intuitively, the inability of an adversary to change a ciphertext into another one such that the underlying plaintexts are meaningfully related [12]. We do not formalize it directly as per [12,5] but rather via the equivalent indistinguishability under parallel chosen-ciphertext attack characterization of [9,18]. (This facilitates our proofs.) The adversary gets the left-or-right encryption oracle and must then decide on a vector of ciphertexts $\mathbf{c}$. (It loses if

they contain an output of the left-or-right encryption oracle.) It is given their corresponding decryptions $\mathbf{p}$ and then wins if it guesses the challenge bit.

The formal definition of privacy is below with the associated experiments.

Definition 1. [**Privacy**] Let $\mathcal{EE} = (\mathcal{K}_c, \mathcal{K}_s, \mathcal{E}, \mathcal{D})$ be an extended encryption scheme, $b \in \{0,1\}$ a challenge bit and $k \in \mathsf{N}$ the security parameter. Let A be an adversary that outputs a bit d. The *left-or-right* encryption oracle $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$, given to the adversary A, takes input a pair (x_0, x_1) of equal-length messages, computes ciphertext $X \leftarrow \mathcal{E}_K(x_b)$, and returns X to the adversary. (It flips coins, or updates state for the encryption function, as necessary. If the input messages are not of equal length it returns the empty string.) Now consider the following experiments each of which returns a bit.

Experiment $\mathbf{Exp}^{\text{ind-cpa-}b}_{\mathcal{EE},A}(k)$	Experiment $\mathbf{Exp}^{\text{ind-cca-}b}_{\mathcal{EE},A}(k)$
$K_c \xleftarrow{R} \mathcal{K}_c(k)\,;\ K_s \xleftarrow{R} \mathcal{K}_s(k)$	$K_c \xleftarrow{R} \mathcal{K}_c(k)\,;\ K_s \xleftarrow{R} \mathcal{K}_s(k)\,;\ K \leftarrow (K_c, K_s)$
$K \leftarrow (K_c, K_s)$	$d \leftarrow A^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b)),\mathcal{D}_K(\cdot)}(k, K_c)$
$d \leftarrow A^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))}(k, K_c)$	If $\mathcal{D}_K(\cdot)$ was never queried on an output of
return d	$\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ then return d else return 0

Experiment $\mathbf{Exp}^{\text{nm-cpa-}b}_{\mathcal{EE},A}(k)$

$K_c \xleftarrow{R} \mathcal{K}_c(k)\,;\ K_s \xleftarrow{R} \mathcal{K}_s(k)\,;\ K \leftarrow (K_c, K_s)$

$(\mathbf{c}, s) \leftarrow A_1^{\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))}(k, K_c)\,;\ \mathbf{p} \leftarrow (\mathcal{D}_K(c_1), \cdots, \mathcal{D}_K(c_n))\,;\ d \leftarrow A_2(\mathbf{p}, \mathbf{c}, s)$

If $\mathbf{c}$ contains no ciphertext output by $\mathcal{E}_K(\mathcal{LR}(\cdot,\cdot,b))$ then return d else return 0

For each notion of privacy sss-aaa $\in \{\text{ind-cpa}, \text{ind-cca}, \text{nm-cpa}\}$ we associate to the adversary A a corresponding advantage defined via

$$\mathbf{Adv}^{\text{sss-aaa}}_{\mathcal{EE},A}(k) \;=\; \Pr\left[\,\mathbf{Exp}^{\text{sss-aaa-1}}_{\mathcal{EE},A}(k) = 1\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{sss-aaa-0}}_{\mathcal{EE},A}(k) = 1\,\right] .$$

For each security notion SSS-AAA $\in \{\text{IND-CPA}, \text{IND-CCA}, \text{NM-CPA}\}$, the scheme $\mathcal{EE}$ is said to be *SSS-AAA secure* if the corresponding advantage function, $\mathbf{Adv}^{\text{sss-aaa}}_{\mathcal{EE},F}(\cdot)$ of any adversary F whose time-complexity is polynomial in k, is negligible. ∎

Integrity. The formalization of integrity follows [7]. The adversary is allowed to mount a chosen-message attack on the scheme, modeled by giving it access to an encryption oracle. Success is measured by its ability to output a "new" ciphertext that makes the decryption algorithm output a plaintext rather than reject by outputting $\perp$. Here the "new" ciphertext means that the ciphertext was never output by the encryption oracle as a response to the adversary's queries. The formal definition of integrity is below with the associated experiment.

Definition 2. [**Integrity**] Let $\mathcal{EE} = (\mathcal{K}_c, \mathcal{K}_s, \mathcal{E}, \mathcal{D})$ be an extended encryption scheme, and $k \in \mathsf{N}$ the security parameter. Let B be an adversary that has access to the encryption oracle and outputs a ciphertext. Now consider the following experiment.

Experiment $\mathbf{Exp}^{\text{int-ctxt}}_{\mathcal{SE},B}(k)$
$K_c \xleftarrow{R} \mathcal{K}_c$; $K_s \xleftarrow{R} \mathcal{K}_s$; $K \leftarrow (K_c, K_s)$; $C \leftarrow B^{\mathcal{E}_K(\cdot)}(k, K_c)$
If $\mathcal{D}_K(C) \neq \perp$ and C was never a response of $\mathcal{E}_K(\cdot)$ then return 1 else return 0

We associate to the adversary B a corresponding advantage defined via

$$\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},B}(k) \;=\; \Pr\left[\,\mathbf{Exp}^{\text{int-ctxt}}_{\mathcal{SE},B}(k) = 1\,\right] .$$

The scheme $\mathcal{SE}$ is said to be *INT-CTXT secure* if the advantage function $\mathbf{Adv}^{\text{int-ctxt}}_{\mathcal{SE},F}(\cdot)$ of any adversary F whose time-complexity is polynomial in k, is negligible. ∎

3 The Encryption-with-Redundancy Paradigm

We describe the paradigm in a general setting, as a transform that associates to any given symmetric encryption scheme and any given "redundancy code" an extended encryption scheme. We first define the syntax for redundancy codes, then detail the constructions, separating the cases of public and secret redundancy, and conclude by observing that the transform always preserves privacy. This leaves later sections to investigate the difficult issue, namely the integrity of the extended encryption scheme with redundancy.

Redundancy codes. A *redundancy code* $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ consists of two algorithms $\mathcal{K}_r$ and $\mathcal{H}$. The randomized key generation algorithm $\mathcal{K}_r$ takes a security parameter k and in time $\text{poly}(k)$ returns a key K_r; we write $K_r \xleftarrow{R} \mathcal{K}_r(k)$. The deterministic redundancy computation algorithm $\mathcal{H}$ takes K_r and a string $M \in \{0,1\}^*$ and in time $\text{poly}(k, |M|)$ returns a string τ; we write $\tau \leftarrow \mathcal{H}_{K_r}(M)$. Usually the length of τ is $\ell(k)$ where $\ell(\cdot)$, an integer valued function that depends only on the security parameter, is called the *output length* of the redundancy code. We say that the redundancy is *public* if the key K_r is public and known to the adversary. We say the redundancy is *secret* if K_r is part of the shared secret key.

Extended encryption schemes with redundancy. Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be a given (symmetric) encryption scheme, which we will call the *base* encryption scheme. Let $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ be a given redundancy code as above. We define an associated *extended encryption scheme with public redundancy* and an associated *extended encryption scheme with secret redundancy.*

Construction 1. The extended encryption scheme with public redundancy $\mathcal{EPR} = (\mathcal{K}_c, \mathcal{K}_s, \overline{\mathcal{E}}, \overline{\mathcal{D}})$, associated to base encryption scheme $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ and redundancy code $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$, is defined as follows:

Algorithm $\mathcal{K}_c(k)$	Algorithm $\mathcal{K}_s(k)$	Algorithm $\overline{\mathcal{E}}_{\langle K_e, K_r\rangle}(M)$	Algorithm $\overline{\mathcal{D}}_{\langle K_e, K_r\rangle}(C)$
$K_r \xleftarrow{R} \mathcal{K}_r(k)$	$K_e \xleftarrow{R} \mathcal{K}_e(k)$	$\tau \leftarrow \mathcal{H}_{K_r}(M)$	$P \leftarrow \mathcal{D}_{K_e}(C)$
return K_r	return K_e	$C \xleftarrow{R} \mathcal{E}_{K_e}(M\|\tau)$	Parse P as $M\|\tau$
		return C	if $\tau \neq \mathcal{H}_{K_r}(M)$
			then return $\perp$
			else return M

Note that the common-key generation algorithm returns the key for the redundancy function, which is thus available to the adversary. That is why we say the redundancy is public. ∎

Construction 2. The extended encryption scheme with secret redundancy $\mathcal{ESR} = (\mathcal{K}_c, \mathcal{K}_s, \overline{\mathcal{E}}, \overline{\mathcal{D}})$, associated to base encryption scheme $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ and redundancy code $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$, is defined as follows:

Algorithm $\mathcal{K}_c(k)$	Algorithm $\mathcal{K}_s(k)$	Algorithm $\overline{\mathcal{E}}_{\langle K_e, K_r\rangle}(M)$	Algorithm $\overline{\mathcal{D}}_{\langle K_e, K_r\rangle}(C)$
return ε	$K_e \xleftarrow{R} \mathcal{K}_e(k)$	$\tau \leftarrow \mathcal{H}_{K_r}(M)$	$N \leftarrow \mathcal{D}_{K_e}(C)$
	$K_r \xleftarrow{R} \mathcal{K}_r(k)$	$C \xleftarrow{R} \mathcal{E}_{K_e}(M\|\tau)$	Parse N as $M\|\tau$
	return $\langle K_e, K_r\rangle$	return C	if $\tau \neq \mathcal{H}_{K_r}(M)$
			then return $\perp$
			else return M

Note that the common key generation algorithm $\mathcal{K}_c$ returns the empty string ε. We may omit the algorithm $\mathcal{K}_c$ and write $\mathcal{ESR} = (\mathcal{K}_s, \overline{\mathcal{E}}, \overline{\mathcal{D}})$. The key for the redundancy function is part of the secret key not available to the adversary. ∎

The symbol $\perp$ is a distinct symbol that indicates that the ciphertext is not valid. When we refer to an extended encryption scheme with redundancy in general we mean either of the above, and denote it by $\mathcal{ER}$.

Privacy is preserved. We now present a theorem regarding the privacy of an extended encryption scheme with redundancy. It applies both to the case of public and to the case of secret redundancy. The theorem below says that the encryption scheme with redundancy inherits the privacy of the base symmetric encryption scheme regardless of the redundancy code being used. This means that privacy depends only on the underlying encryption scheme, not on the redundancy code. The proof is straightforward and can be found in the full version of this paper [1].

Theorem 1. [Privacy of an extended encryption scheme with redundancy] *Let $\mathcal{SE} = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme and let $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ be a redundancy code. Let $\mathcal{ER} = (\mathcal{K}_c, \mathcal{K}_s, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ be an associated extended encryption scheme with redundancy, either public or secret. Then if $\mathcal{SE}$ is IND-CPA (resp. IND-CCA, NM-CPA) secure, so is $\mathcal{ER}$.* ∎

For simplicity we have stated the theorem with reference to asymptotic notions of security but we remark that the reduction in the proof is tight, and a concrete security statement reflecting this can be derived from the proof.

4 Encryption with Public Redundancy

Here we will show that in general the encryption with public redundancy paradigm fails in a strong way, meaning there is a base encryption scheme such that for *all* choices of public redundancy code, the associated extended encryption

scheme with public redundancy scheme (cf. Construction 1) fails to provide integrity. This is true regardless of the security property of the base encryption scheme (i.e. IND-CPA, NM-CCA, or IND-CCA).

The result follows the paradigm of similar negative results in [4,7]. We must make the minimal assumption that some encryption scheme $\mathcal{SE}'$ secure in the given sense exists, else the question is moot. We then modify the given encryption scheme to a new scheme $\mathcal{SE}$ so that when $\mathcal{SE}$ becomes the base encryption scheme of the extended encryption scheme with public redundancy, we can provide an attack on the integrity of the latter. The proof of the following theorem can be found in the full version of this paper [1].

Theorem 2. [Encryption with public redundancy] *Suppose there exists a symmetric encryption scheme $\mathcal{SE}'$ which is IND-CCA (resp. IND-CPA, NM-CPA) secure. Then there exists a symmetric encryption scheme $\mathcal{SE}$ which is also IND-CCA (resp. IND-CPA, NM-CPA) secure but, for any redundancy code $\mathcal{RC}$, the extended encryption scheme with public redundancy $\mathcal{EPR}$ associated to $\mathcal{SE}$ and $\mathcal{RC}$ is not INT-CTXT secure.* ∎

5 Encryption with Secret Redundancy

In this section, we examine encryption schemes with secret redundancy in general so as to whether or not they provide integrity.

The following theorem states the negative result where the base encryption scheme is IND-CPA secure. The proof can be found in the full version of this paper [1].

Theorem 3. [IND-CPA encryption with secret redundancy] *Suppose there exists a symmetric encryption scheme $\mathcal{SE}'$ which is IND-CPA secure. Then there exists a symmetric encryption scheme $\mathcal{SE}$ which is also IND-CPA secure but, for any redundancy code $\mathcal{RC}$, the extended encryption scheme with secret redundancy $\mathcal{ESR}$ associated to $\mathcal{SE}$ and $\mathcal{RC}$ is not INT-CTXT secure.* ∎

For the positive result, we define below the (necessary and sufficient) security property required of the redundancy code.

We define a notion of *unforgeability under no message attack (UF-NMA)*, which is the weakest form of security required of a MAC (message authentication code) —roughly, the adversary wins if it outputs a valid message and tag pair without seeing any legitimately produced message and tag pairs. Since a MAC and a redundancy code are syntactically identical, we adopt the weakest security notion of a MAC as the security notion of a redundancy code. The formal definition is given below. Note that, in the attack model, the key to the redundancy code is not given to the adversary, indicating that the redundancy is secret.

Definition 3. [Unforgeability under no message attack (UF-NMA)] Let $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ be a redundancy code. Let $k \in \mathsf{N}$. Let F be an adversary. Consider the following experiment:

Algorithm $\mathcal{K}_e(k)$
$a_1 \xleftarrow{R} \{0,1\}^\kappa$
$a_2 \xleftarrow{R} \{0,1\}^\kappa$
Return $(a_1 \| a_2)$

Algorithm $\mathcal{E}_{a_1\|a_2}(X)$
Parse X as $x_1 \cdots x_{n+1}$
$y_0 \xleftarrow{R} \{0,1\}^l$
For $i = 1, \cdots, n$ do
$y_i \leftarrow F_{a_1}(y_{i-1} \oplus x_i)$
$y_{n+1} \leftarrow F_{a_2}(y_n \oplus x_{n+1})$
Return $y_0 y_1 \cdots y_{n+1}$

Algorithm $\mathcal{D}_{a_1\|a_2}(Y)$
Parse Y as $y_0 y_1 \cdots y_{n+1}$
For $i = 1, \cdots, n$ do
$x_i \leftarrow F_{a_1}^{-1}(y_i) \oplus y_{i-1}$
$x_{n+1} \leftarrow F_{a_2}^{-1}(y_{n+1}) \oplus y_n$
$X \leftarrow x_1 \cdots x_{n+1}$
Return X

Fig. 2. Nested CBC encryption scheme $NCBC[F] = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$.

Experiment $\mathbf{Exp}^{\text{uf-nma}}_{\mathcal{RC},F}(k)$
$K_r \xleftarrow{R} \mathcal{K}_r(k)$; $(M, \tau) \leftarrow F(k)$
If $\tau = \mathcal{H}_{K_r}(M)$ then return 1 else return 0

We define the *advantage* of the adversary via,

$$\mathbf{Adv}^{\text{uf-nma}}_{\mathcal{RC},F}(k) = \Pr\left[\, \mathbf{Exp}^{\text{uf-nma}}_{\mathcal{RC},F}(k) = 1 \,\right]$$

The redundancy code $\mathcal{RC}$ is said to be *UF-NMA secure* if the function $\mathbf{Adv}^{\text{uf-nma}}_{\mathcal{RC},F}(\cdot)$ is negligible for any adversary F whose time complexity is polynomial in k. ∎

The following theorem states the positive results. The proof can be found in the full version of this paper [1].

Theorem 4. [NM-CPA or IND-CCA encryption with secret redundancy] *Let $\mathcal{SE}$ be a symmetric encryption scheme which is NM-CPA or IND-CCA secure and let $\mathcal{RC}$ be a redundancy code. Then the extended encryption scheme with secret redundancy $\mathcal{ESR}$ associated to $\mathcal{SE}$ and $\mathcal{RC}$ is INT-CTXT secure if and only if the redundancy code $\mathcal{RC}$ is UF-NMA secure.* ∎

6 Nested CBC (NCBC) with Redundancy

In this section, we will consider a "natural" variant of CBC encryption, called "Nested CBC (NCBC)", designed to eliminate length-based attacks. The detailed description of NCBC is given below.

Let F: $\{0,1\}^\kappa \times \{0,1\}^l \to \{0,1\}^l$ be a family of permutations (i.e. a block cipher). We let $F_a(\cdot) = F(a, \cdot)$ and we let F_a^{-1} denote the inverse of F_a, for any key $a \in \{0,1\}^\kappa$. Our variant of CBC encryption involves the use of two keys instead of just one. The additional key is used for the last iteration of the block cipher. We call this variant of CBC the *Nested CBC (NCBC)* and denote it by $NCBC[F] = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$. The algorithms for the NCBC encryption scheme are shown in Figure 2. We assume that the messages have length a multiple of the block length l.

Given the NCBC encryption scheme, we examine what kinds of security properties for the redundancy code will provide integrity of ciphertexts for the encryption scheme with redundancy. We examine this for both public redundancy

and secret redundancy. In order to facilitate the practical security analyses, we will make concrete security assessments for the schemes examined in this section.

Since the security of the NCBC scheme is based on the security of the underlying block cipher (as well as that of the redundancy code), we first define the security property of the underlying block cipher on which our security analysis will be based.

Block ciphers are usually modeled as "pseudorandom permutations" (sometimes even as "pseudorandom functions") [4]. However, we use a stronger notion called *strong pseudorandom permutation (SPRP)* [22], where the adversary gets access to both forward and inverse permutation oracles in the attack model.

Definition 4. [**Strong pseudorandom permutation (SPRP)**]
Let $F\colon \{0,1\}^\kappa \times \{0,1\}^l \to \{0,1\}^l$ be a block cipher with key-length κ and block-length l. Let P^l be the family of all permutations on l-bits. Let $k \in \mathsf{N}$ and $b \in \{0,1\}$. Let D be an adversary that has access to oracles $g(\cdot)$ and $g^{-1}(\cdot)$. Consider the following experiment:

Experiment $\mathbf{Exp}^{\text{sprp-b}}_{F,D}(k)$
If $b = 0$ then $g \xleftarrow{R} P^l$ else $K \xleftarrow{R} \{0,1\}^l$; $g \leftarrow F_K$
$d \leftarrow D^{g(\cdot),g^{-1}(\cdot)}(k)$; return d

We define the *advantage* and the *advantage function* of the adversary as follows. For any integers $t, q \geq 0$,

$$\mathbf{Adv}^{\text{sprp}}_{F,D}(k) = \Pr\left[\,\mathbf{Exp}^{\text{sprp-1}}_{F,D}(k) = 1\,\right] - \Pr\left[\,\mathbf{Exp}^{\text{sprp-0}}_{F,D}(k) = 1\,\right]$$

$$\mathbf{Adv}^{\text{sprp}}_{F}(k,t,q) = \max_{D}\left\{\mathbf{Adv}^{\text{sprp}}_{F,D}(k)\right\}$$

where the maximum is over all D with time complexity t, making at most q queries to the oracles $g(\cdot)$ and $g^{-1}(\cdot)$. The block cipher F is said to be *SPRP secure* if the function $\mathbf{Adv}^{\text{sprp}}_{F,D}(k)$ is negligible for any adversary D whose time complexity is polynomial in k. ∎

The "time-complexity" refers to that of the entire experiment. Here, the choice of a random permutation g is not made all at once, but rather g is simulated in the natural way.

6.1 NCBC with Secret Redundancy

Here we examine what kind of property on the redundancy code suffices to make the NCBC with secret redundancy provide integrity. We denote by $SNCBC[F, \mathcal{RC}] = (\mathcal{K}_s, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ the extended encryption scheme with secret redundancy associated to the NCBC encryption scheme $NCBC[F] = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ and a redundancy code $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$.

It turns out that the NCBC scheme with secret redundancy provides integrity if the underlying secret redundancy meets the notion of *almost XOR universal* (AXU) introduced in [20,25].

Definition 5. [Almost XOR Universal (AXU)] Let $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ be a redundancy code whose output length is $\ell(k)$, where $k \in \mathsf{N}$. We define the *advantage function* of the redundancy code $\mathcal{RC}$ as follows.

$$\mathbf{Adv}^{\mathrm{axu}}_{\mathcal{RC}}(k, \mu)$$
$$= \max_{x,x' \in \{0,1\}^*, r \in \{0,1\}^{\ell(k)}} \left\{ \Pr\left[\mathcal{H}_{K_r}(x) \oplus \mathcal{H}_{K_r}(x') = r \ : \ K_r \xleftarrow{R} \mathcal{K}_r(k) \right] \right\}$$

where maximum is taken over all *distinct* x, x' of length at most μ each, and all $r \in \{0,1\}^{\ell(k)}$. ∎

We now state the theorem concerning the security of NCBC scheme with secret redundancy. The proof can be found in the full version of this paper [1].

Theorem 5. [Integrity of NCBC with secret redundancy] *Let $\mathcal{RC}$ be a redundancy code whose output length is l-bits. Let $F\colon \{0,1\}^\kappa \times \{0,1\}^l \to \{0,1\}^l$ be a block cipher, and let $NCBC[F]$ be the NCBC encryption scheme based on F. Let $SNCBC[F, \mathcal{RC}]$ be the extended encryption scheme with secret redundancy associated to $NCBC[F]$ and $\mathcal{RC}$. Let $k \in \mathsf{N}$. Then*

$$\mathbf{Adv}^{\text{int-ctxt}}_{SNCBC[F,\mathcal{RC}]}(k, t, q, \mu)$$
$$\leq \left(\frac{q(q-1)}{2} + 1 \right) \cdot \mathbf{Adv}^{\mathrm{axu}}_{\mathcal{SRC}}(k, \mu) + \frac{1}{2^l - q} + \mathbf{Adv}^{\mathrm{sprp}}_{F}(k, t, q + \mu/l) \quad ∎$$

6.2 NCBC with Public Redundancy

The NCBC with *public* redundancy scheme also provides authenticity if a certain condition on the underlying redundancy code is satisfied. We denote by $PNCBC[F, \mathcal{RC}] = (\mathcal{K}_c, \mathcal{K}_s, \overline{\mathcal{E}}, \overline{\mathcal{D}})$ the extended encryption scheme with public redundancy associated to the NCBC encryption scheme $NCBC[F] = (\mathcal{K}_e, \mathcal{E}, \mathcal{D})$ and a redundancy code $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$.

We want to examine what kind of security property for the underlying public redundancy suffices to make the NCBC scheme with public redundancy provide integrity. It turns out that, for the redundancy code, a cryptographic property called "XOR-collision-resistance" suffices to provide integrity for the NCBC scheme with public redundancy. XOR-collision-resistance is slightly stronger than "collision-resistance". Roughly, a redundancy code $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ is said to be *XOR-collision-resistant (XCR)* if it is "hard" to find strings x, x' where $x \neq x'$ such that $\mathcal{H}_{K_r}(x) \oplus \mathcal{H}_{K_r}(x') = r$ for any committed value r and any given key K_r. We define XOR-collision-resistance (XCR) more formally as follows.

Definition 6. [XOR-Collision-Resistance (XCR)] Let $\mathcal{RC} = (\mathcal{K}_r, \mathcal{H})$ be a redundancy code whose output length is $\ell(k)$, where $k \in \mathsf{N}$. Let $B = (B_1, B_2)$ be an adversary. Consider the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{xcr}}_{\mathcal{PRC},B}(k)$
$(r, s) \leftarrow B_1(k)$; $K_r \xleftarrow{R} \mathcal{K}_r(k)$; $(x, x') \leftarrow B_2(K_r, r, s)$
if $\mathcal{H}_{K_r}(x) \oplus \mathcal{H}_{K_r}(x') = r$ and $x \neq x'$ then return 1 else return 0

Above, the variable s denotes the state information. We define the *advantage* and the *advantage function* of the adversary via,

$$\mathbf{Adv}^{\mathrm{xcr}}_{\mathcal{PRC},B}(k) = \Pr\left[\,\mathbf{Exp}^{\mathrm{xcr}}_{\mathcal{PRC},B}(k) = 1\,\right]$$

$$\mathbf{Adv}^{\mathrm{xcr}}_{\mathcal{PRC}}(k,t) = \max_{B}\left\{\mathbf{Adv}^{\mathrm{xcr}}_{\mathcal{PRC},B}(k)\right\}$$

where the maximum is over all B with time complexity t. The scheme $\mathcal{PRC}$ is said to be *XCR secure* if the function $\mathbf{Adv}^{\mathrm{xcr}}_{\mathcal{PRC},A}(k)$ is negligible for any adversary A whose time complexity is polynomial in k. ∎

XOR-collision-resistance (XCR) as defined above is a new notion that has not been explicitly studied in the literature. In XCR, the adversary first outputs a string r and then obtains the key to the function. The adversary's goal is to find a pair of strings x, x' (called an "XOR-collision" pair) such that the XOR of their images equals r.

Given the definitions for the security properties of the underlying primitives, we now state the theorem regarding the security of the *PNCBC* scheme. Following that we will further discuss XCR redundancy codes. The proof can be found in the full version of this paper [1].

Theorem 6. [Integrity of NCBC with public redundancy] *Let $\mathcal{RC}$ be a redundancy code whose output length is l-bits. Let $F\colon \{0,1\}^{\kappa} \times \{0,1\}^{l} \to \{0,1\}^{l}$ be a block cipher, and let $NCBC[F]$ be the NCBC encryption scheme based on F. Let $PNCBC[F,\mathcal{RC}]$ be the extended encryption scheme with public redundancy associated to $NCBC[F]$ and $\mathcal{RC}$. Let $k \in \mathsf{N}$. Then*

$$\mathbf{Adv}^{\text{int-ctxt}}_{PNCBC[F,\mathcal{RC}]}(k,t,q,\mu)$$

$$\leq mq \cdot \mathbf{Adv}^{\mathrm{xcr}}_{\mathcal{RC}}(k,t') + \frac{2}{2^l - m} + \frac{m^2}{2(2^l - m)} + 2 \cdot \mathbf{Adv}^{\mathrm{sprp}}_{F}(k,t,q+m)$$

where $m = \mu/l$. ∎

We now further discuss XCR redundancy codes. Note that the XCR property can be thought of as a cryptographic counterpart of the AXU property described in the previous section. The combinatorial property of AXU (for secret redundancy) is weaker, and therefore, easier to implement than the cryptographic property of XCR (for public redundancy). This tells us that by adding the power of secrecy to the redundancy code, one can achieve the same security (i.e. integrity) for the NCBC with redundancy scheme under a weaker security assumption on the underlying redundancy code.

What are candidates for XCR redundancy codes? Note that an unkeyed hash function like SHA-1 does not yield an XCR redundancy code. Indeed, an adversary can choose any distinct x, x', and let $r = \text{SHA-1}(x) \oplus \text{SHA-1}(x')$. It can output r in its first stage, and x, x' in its second, and win the game. An XCR redundancy code must be keyed. A keyed hash function is a good candidate. Specifically, we suggest that HMAC [3] is a candidate for a XCR redundancy

code. In the full version of this paper [1] we discuss other constructions including a general way to transform any collision-resistant function into an XCR redundancy code.

Acknowledgments

We thank Hugo Krawczyk for helpful comments on a previous version of this paper. We thank Daniele Micciancio for helpful discussions. The authors are supported in part by Bellare's 1996 Packard Foundation Fellowship in Science and Engineering and NSF CAREER Award CCR-9624439. The first author was also supported in part by an NSF graduate fellowship.

References

1. J. An and M. Bellare, "Does encryption with redundancy provide authenticity?" Full version of this paper, available via `http://www-cse.ucsd.edu/users/mihir`.
2. M. Atici and D. Stinson, "Universal Hashing and Multiple Authentication," *Advances in Cryptology – CRYPTO '96*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.
3. M. Bellare, R. Canetti and H. Krawczyk, "Keying hash functions for message authentication," *Advances in Cryptology – CRYPTO '96*, Lecture Notes in Computer Science Vol. 1109, N. Koblitz ed., Springer-Verlag, 1996.
4. M. Bellare, A. Desai, E. Jokipii and P. Rogaway, "A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation," *Proc. of the 38th* IEEE FOCS, IEEE, 1997.
5. M. Bellare, A. Desai, D. Pointcheval and P. Rogaway, "Relations among notions of security for public-key encryption schemes," *Advances in Cryptology – CRYPTO '98*, Lecture Notes in Computer Science Vol. 1462, H. Krawczyk ed., Springer-Verlag, 1998.
6. M. Bellare, J. Kilian and P. Rogaway, "The Security of the Cipher Block Chaining Message Authentication Code," *Journal of Computer and System Sciences*, Vol. 61, No. 3, December 2000, pp. 362–399.
7. M. Bellare and C. Namprempre, "Authenticated Encryption: Relations among notions and analysis of the generic composition paradigm," *Advances in Cryptology – ASIACRYPT '00*, Lecture Notes in Computer Science Vol. 1976, T. Okamoto ed., Springer-Verlag, 2000.
8. M. Bellare and P. Rogaway, "Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography," *Advances in Cryptology – ASIACRYPT '00*, Lecture Notes in Computer Science Vol. 1976, T. Okamoto ed., Springer-Verlag, 2000.
9. M. Bellare and A. Sahai, "Non-Malleable Encryption: Equivalence between Two Notions, and an Indistinguishability-Based Characterization," *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.
10. J. Black, S. Halevi, H. Krawczyk, T. Krovetz and P. Rogaway, "UMAC: Fast and secure message authentication," *Advances in Cryptology – CRYPTO '99*, Lecture Notes in Computer Science Vol. 1666, M. Wiener ed., Springer-Verlag, 1999.

11. L. Carter and M. Wegman, "Universal Classes of Hash Functions," *Journal of Computer and System Sciences*, Vol. 18, 1979, pp. 143–154.
12. D. Dolev, C. Dwork and M. Naor, "Non-malleable cryptography," *Proc. of the 23rd* ACM STOC, ACM, 1991.
13. S. Goldwasser and S. Micali, "Probabilistic encryption," *Journal of Computer and System Sciences*, Vol. 28, 1984, pp. 270–299.
14. S. Halevi and H. Krawczyk, "MMH: Software Message Authentication in the Gbit/Second Rates," *Fast Software Encryption — 4th International Workshop, FSE'97 Proceedings*, Lecture Notes in Computer Science, vol. 1267, E. Biham ed., Springer, 1997.
15. R. Jueneman, "A high speed manipulation detection code," *Advances in Cryptology – CRYPTO '86*, Lecture Notes in Computer Science Vol. 263, A. Odlyzko ed., Springer-Verlag, 1986.
16. R. Jueneman, C. Meyer and S. Matyas, "Message Authentication with Manipulation Detection Codes," in Proceedings of the 1983 IEEE Symposium on Security and Privacy, IEEE Computer Society Press, 1984, pp.33-54.
17. C. Jutla, "Encryption modes with almost free message integrity," Report 2000/039, *Cryptology ePrint Archive,* `http://eprint.iacr.org/`, August 2000.
18. J. Katz and M. Yung, "Complete characterization of security notions for probabilistic private-key encryption," *Proc. of the 32nd* ACM STOC, ACM, 2000.
19. J. Katz and M. Yung, "Unforgeable Encryption and Adaptively Secure Modes of Operation," *Fast Software Encryption '00*, Lecture Notes in Computer Science, B. Schneier ed., Springer-Verlag, 2000.
20. H. Krawczyk, "LFSR-based Hashing and Authentication," *Advances in Cryptology – CRYPTO '94*, Lecture Notes in Computer Science Vol. 839, Y. Desmedt ed., Springer-Verlag, 1994.
21. H. Krawczyk, "The order of encryption and authentication for protecting communications (Or: how secure is SSL?)," Manuscript, 2001.
22. M. Luby and C. Rackoff, "How to Construct Pseudorandom Permutations from Pseudorandom Functions," *SIAM Journal of Computing,* Vol. 17, No. 2, pp. 373–386, April 1988.
23. A. Menezes, P. van Oorshot and S. Vanstone, "Handbook of applied cryptography," CRC Press LLC, 1997.
24. B. Preneel, "Cryptographic Primitives for Information Authentication — State of the Art," *State of the Art in Applied Cryptography*, COSIC'97, LNCS 1528, B. Preneel and V. Rijmen eds., Springer-Verlag, pp. 49-104, 1998.
25. P. Rogaway, "Bucket Hashing and its Application to Fast Message Authentication," *Advances in Cryptology – CRYPTO '95*, Lecture Notes in Computer Science Vol. 963, D. Coppersmith ed., Springer-Verlag, 1995.
26. P. Rogaway, "OCB mode: Parallelizable authenticated encryption," Presented in *NIST's workshop on modes of operations,* October, 2000. See `http://csrc.nist.gov/encryption/modes/workshop1/`
27. M. Wegman and L. Carter, "New hash functions and their use in authentication and set equality," *Journal of Computer and System Sciences*, Vol. 22, 1981, pp. 265–279.