Nonce-Based Symmetric Encryption

Phillip Rogaway

Dept. of Computer Science, University of California, Davis, CA 95616, USA, and Dept. of Computer Science, Fac. of Science, Chiang Mai University, Chiang Mai, Thailand 50200. rogaway@cs.ucdavis.edu, www.cs.ucdavis.edu/∼rogaway

Abstract. Symmetric encryption schemes are usually formalized so as to make the encryption operation a probabilistic or state-dependent function $\mathcal{E}$ of the message M and the key K: the user supplies M and K and the encryption process does the rest, flipping coins or modifying internal state in order to produce a ciphertext C. Here we investigate an alternative syntax for an encryption scheme, where the encryption process $\mathcal{E}$ is a deterministic function that surfaces an *initialization vector* (IV). The user supplies a message M, key K, and initialization vector N, getting back the (one and only) associated ciphertext $C = \mathcal{E}_K^N(M)$. We concentrate on the case where the IV is guaranteed to be a *nonce*—something that takes on a new value with every message one encrypts. We explore definitions, constructions, and properties for nonce-based encryption. Symmetric encryption with a surfaced IV more directly captures real-word constructions like CBC mode, and encryption schemes constructed to be secure under nonce-based security notions may be less prone to misuse.

1 Introduction

Ever since Goldwasser and Micali's landmark paper [7], formalizations of encryption schemes have usually made the encryption algorithm probabilistic or stateful. In this paper we investigate a different formalization for symmetric encryption: the encryption algorithm is made to be a deterministic function, but one of its argument is a user-supplied *initialization vector* (IV). Effectively, the user and not the encryption algorithm is made responsible for flipping coins or maintaining state. We are mostly interested in security properties that can be guaranteed as long as the IV is a *nonce*—a value, like a counter, used at most once within a session. Our formalization leads to what is effectively a stronger notion of privacy than the conventional formalization, and a stronger notion of authenticity as well. As a consequence, encryption schemes created so as to satisfy the given notions would seem to be less likely to be misused.

Comparing CBC and CBC$ encryption. Popular modes of operation for encryption have always surfaced an IV. For example, CBC using block cipher $E\colon \mathsf{Key} \times \{0,1\}^n \to \{0,1\}^n$ requires an initialization vector $N \in \{0,1\}^n$ to encrypt a message M (or decrypt a message C) under key $K \in \mathsf{Key}$. See Fig. 1 and note that the initialization vector N is an argument to both CBC.Encrypt

Algorithm CBC.Encrypt$^N_K(M)$
if $|M| \notin \{n, 2n, 3n, \ldots\}$ **then return** $\star$
Parse M into $M_1 \cdots M_m$ where $|M_i| = n$
$C_0 \leftarrow N$
for $i \leftarrow 1$ **to** m **do**
    $C_i \leftarrow E_K(C_{i-1} \oplus M_i)$
return $C_1 \cdots C_m$

Algorithm CBC.Decrypt$^N_K(C)$
if $|C| \notin \{n, 2n, 3n, \ldots\}$ **then return** $\star$
Parse C into $C_1 \cdots C_m$ where $|M_i| = n$
$C_0 \leftarrow N$
for $i \in [1 \mathinner{..} m]$ **do**
    $M_i \leftarrow C_{i-1} \oplus E_K^{-1}(C_i)$
return $M_1 \cdots M_m$

Fig. 1. Scheme CBC. Encryption and decryption depend on a block cipher $E\colon \mathrm{Key} \times \{0,1\}^n \to \{0,1\}^n$. The key space for the encryption scheme is the same as the key space for the block cipher. The IV N is a string in $\{0,1\}^n$. The encryption scheme is the pair (CBC.Encrypt, CBC.Decrypt).

Algorithm CBC\$.Encrypt$_K(M)$
if $|M| \notin \{n, 2n, 3n, \ldots\}$ **then return** $\star$
Parse M into $M_1 \cdots M_m$ where $|M_i| = n$
$C_0 \xleftarrow{\$} \{0,1\}^n$
for $i \leftarrow 1$ **to** m **do**
    $C_i \leftarrow E_K(C_{i-1} \oplus M_i)$
return $C_0 C_1 \cdots C_m$

Algorithm CBC\$.Decrypt$_K(C)$
if $|C| \notin \{2n, 3n, 4n, \ldots\}$ **then return** $\star$
Parse C into $C_0 C_1 \cdots C_m$ where $|M_i| = n$

for $i \in [1 \mathinner{..} m]$ **do**
    $M_i \leftarrow C_{i-1} \oplus E_K^{-1}(C_i)$
return $M_1 \cdots M_m$

Fig. 2. Scheme CBC\$. The mechanism depends on a block cipher $E\colon \mathrm{Key} \times \{0,1\}^n \to \{0,1\}^n$. Scheme CBC\$ is a conventional probabilistic encryption scheme. It is just like scheme CBC except that the encryption routine chooses the IV $N = C_0$ internally and at random. The user cannot influence it. The value is now returned as part of the ciphertext.

and CBC.Decrypt. Given that the IV is manifestly present in the description of CBC mode, this would seem to be quite natural. Nonetheless, the approach is at odds with the customary formalization of symmetric encryption going back to [1, 7]. There one explicitly models some particular manner of generating the IV and folds this into the definition of the scheme. For example, one considers the scheme CBC\$ (i.e., CBC with a random IV) as defined in Fig. 2.

Contributions. It is the purpose of this note to treat symmetric encryption schemes in a way that explicitly surfaces the IV. The approach was first taken in our earlier work on authenticated encryption [9, 10], where we adopted nonce-based definitions without significant comment. Here we more systematically investigate the explicit-IV notion of encryption, giving definitions, schemes, and basic results. We are mostly interested in the case when the IV is a nonce: a value used at most once within the scope of a given session.

This note aims to call attention to the explicit-IV approach and to nudge future work on practical encryption schemes into adopting the nonce-based framework.

STANDARDS. We believe that a nonce-based formalization is especially desirable when constructing an encryption scheme for a cryptographic standard: not knowing how the scheme will be used, standards would do well to achieve the strongest practical notion of security relative to the interface that they export. The viewpoint, then, is that conventional encryption modes like CBC, as defined in Fig. 1, are "deficient" insofar as they do not achieve a strong notion of security unless one assumes something very strong about their IVs. One would prefer an encryption mode that achieves a strong notion of security when one assumes very little about the IV. It is thus our view that, in the future, standards for privacy-only encryption would do well to achieve privacy in the ind\$-sense that we will define in Section 3, while standards for authenticated encryption would do well to achieve, in addition, authenticity in the auth-sense that we define in Section 6.

FURTHER REASONS TO SURFACE THE IV. Another motivation for explicitly surfacing the IV in the definition of an encryption scheme is that books and systems often get wrong what it may or may not be. Books will say, for example, that it is fine for the IV in CBC encryption to be a counter, or the last block of encrypted ciphertext. Both statements are wrong, assuming that one intends to achieve a strong notion of privacy. Having definitions that expose the IV across the encryption and decryption interface facilitates answering *what* the IV may or may not be in order to achieve a given notion of security.

Yet another motivation for surfacing the IV is that it allows a particularly simple and strong notion of privacy: indistinguishability from random bits with respect to an adaptive chosen-plaintext-and-IV attack (ind\$, to be defined later). This attack allows the adversary to select not only plaintexts but also the IVs that will be used to encrypt each of them, subject only to the constraint that no IV is reused. The model captures the possibility that the IVs may be chosen in an unfortunate way by the sender, possibly even influenced by the adversary, when we do not mandate any requirement on an IV beyond its non-reuse.

A SMALL WARNING. Nothing in this paper should be construed to suggest that the overall encryption process should become deterministic and stateless (and therefore not semantically secure). We are simply drawing the abstraction boundary a little differently, so that what is "inside" the scheme is deterministic, the coins or state being pushed "outside" of the scheme's formalization.

2 Syntax

DEFINITIONS. We begin by specifying the syntax for an encryption schemes that surfaces an IV. An *IV-based encryption scheme* is a pair of algorithms $\Pi = (\mathcal{E}, \mathcal{D})$ where $\mathcal{E}\colon \mathsf{Key} \times \mathsf{IV} \times \mathsf{Plaintext} \to \mathsf{Ciphertext}$ and $\mathcal{D}\colon \mathsf{Key} \times \mathsf{IV} \times \mathsf{Ciphertext} \to \mathsf{Plaintext} \cup \{\star\}$ are deterministic functions. These functions are called the *encryption function* and the *decryption function*, respectively. Here Key, IV, $\mathsf{Plaintext}$, and $\mathsf{Ciphertext}$ are nonempty sets of strings, the first of which is finite or is otherwise endowed with a distribution (the understood distribution on a finite set being the uniform one). These sets are called the *key space*, the *IV space*, the *message space*, and the *ciphertext space*. We insist that $\mathsf{Plaintext}$ has the structure that if it contains a string M then it contains all string M' having the same length of M. We often write $\mathcal{E}_K^N(M)$ in place of $\mathcal{E}(K,N,M)$ and $\mathcal{D}_K^N(C)$ in place of $\mathcal{D}(K,N,C)$. We require that $\mathcal{D}_K^N(\mathcal{E}_K^N(M)) = M$ for any $K \in \mathsf{Key}$ and $N \in \mathsf{IV}$ and $M \in \mathsf{Plaintext}$. For simplicity, we assume that $|\mathcal{E}_K^N(M)|$ depends only on $|M|$ and, in particular, that $|\mathcal{E}_K^N(M)| = |M| + \tau$ for some constant τ associated to the encryption scheme. We call τ the *stretch* of the encryption scheme.

Comments. (1) We will often use the word *nonce* instead of *IV* and write Nonce, the *nonce space*, instead of IV. We do this when we are thinking in terms of our nonce-based definitions for privacy (to follow). In such cases we call an IV-based encryption scheme a *nonce-based encryption scheme*. (2) We emphasize that $\mathcal{E}$ and $\mathcal{D}$ are deterministic and stateless functions: they may not flip coins or preserve state. (3) What we call the ciphertext $C = \mathcal{E}_K^N(M)$ is not expected to encode the IV, even though the IV is needed to decrypt. The IV may be communicated "out of band" to the receiver, maintained as shared state, or it may be manifest within the context of use, as when the IV is the sector index on a disk. (4) The encryption function $\mathcal{E}$ may be length-preserving, meaning that $|\mathcal{E}_K^N(M)| = |M|$ for all K, N, M. Indeed we will see that encryption schemes can achieve a strong notion of privacy yet have zero stretch. (5) We have allowed for the possibility that the decryption of a string returns the distinguished value $\star$, which is used to indicate that the ciphertext is invalid. While this possibility is not needed for basic notions of privacy, it is needed for defining authenticity. (6) We have not said that the sender or receiver are stateless and without benefit of coins, only that $\mathcal{E}$ and $\mathcal{D}$ are. For example, the sender might maintain a counter to use as the IV. It is simply that this state is outside of the functionality of $\mathcal{E}$.

3 Privacy

Indistinguishability from random bits. Our preferred notion of privacy is "indistinguishability from random bits under an adaptive chosen-plaintext-and-IV attack". To formalize this, let adversary A be an algorithm with access to an oracle and let $\Pi = (\mathcal{E}, \mathcal{D})$ be an IV-based encryption scheme with key space Key and IV space Nonce and stretch τ. We define

$$\mathbf{Adv}_{\Pi}^{\mathrm{ind\$}}(A) = \Pr\left[K \stackrel{\$}{\leftarrow} \mathsf{Key} :\ A^{\mathcal{E}_K(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[A^{\$(\cdot,\cdot)} \Rightarrow 1\right]$$

The superscript ind\$ may alternatively be written as ind\$-cpa. The oracle $\mathcal{E}_K(\cdot,\cdot)$, on input (N, M), returns $\mathcal{E}_K^N(M)$. We sometimes refer to this as the *real* encryption oracle. The oracle $\$(\cdot,\cdot)$, on input (N, M), returns $|M| + \tau$ random bits. We sometimes refer to this as the *random-bits* oracle. Both oracles return $\star$ if $N \notin \mathsf{Nonce}$ or $M \notin \mathsf{Plaintext}$. When we write $A^{\mathcal{O}} \Rightarrow 1$ we are referring to the event that adversary A, running with its oracle $\mathcal{O}$, outputs the bit 1. We call an adversary A *nonce-respecting* if it never repeats a nonce: if A asks (N, M) it never subsequently asks (N, M') for any M'. This must hold regardless of A's coins and regardless of oracle responses. We assume that any ind\$-adversary is nonce-respecting.

Conventional indistinguishability. It is more customary to focus on a different kind of indistinguishability. Once again, let $\Pi = (\mathcal{E}, \mathcal{D})$ be a nonce-based encryption scheme with key space Key and nonce space Nonce. Let A be an adversary. Then define

$$\mathbf{Adv}_{\Pi}^{\mathrm{ind}}(A) = \Pr\left[K \xleftarrow{\$} \mathsf{Key}:\ A^{\mathcal{E}_K(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[K \xleftarrow{\$} \mathsf{Key}:\ A^{\mathcal{E}_K(\cdot,0^{|\cdot|})} \Rightarrow 1\right]$$

The superscript ind\$ may alternatively be written as ind-cpa. The first oracle is a *real* encryption oracle, as before. The second oracle, on input (N, M), returns $\mathcal{E}_K(N, 0^{|M|})$. We call this a *fake* encryption oracle. Both oracles return $\star$ if $N \notin \mathsf{Nonce}$ or $M \notin \mathsf{Plaintext}$, and A is always assumed to be nonce-respecting.

Resource-parameterized definitions. If Π is a scheme and A is an adversary and $\mathbf{Adv}_{\Pi}^{\mathrm{xxx}}(A)$ is a measure of adversarial advantage already defined, then we write $\mathbf{Adv}_{\Pi}^{\mathrm{xxx}}(\mathcal{R})$ to mean the maximal value of $\mathbf{Adv}_{\Pi}^{\mathrm{xxx}}(A)$ over all adversaries A that use resources bounded by $\mathcal{R}$. Here $\mathcal{R}$ is a list of variables specifying the resources of interest for the adversary in question. Adversarial resources to which we pay attention are: t—the running time of the adversary; q—the number of queries asked by the adversary; and σ—the aggregate length of these queries, plus the length of the adversary's output, measured in n-bit blocks, for some understood value n. When an adversary's query or output is a tuple of strings we count in σ the sum of the lengths of each component. Fractional blocks and the emptystring contribute 1. By convention, the running time of an algorithm includes the description size of that algorithm, relative to some standard encoding.

Discussion: favoring ind\$ over ind. It is easy to verify that the ind\$-notion of security implies the ind-notion, and by a tight reduction, while ind does not imply ind\$ at all. (The same is true if one speaks of indistinguishability and indistinguishability from random bits in the context of conventional, probabilistic encryptions schemes.) Despite this, typical encryption schemes seem to achieve ind\$ if they achieve ind (again, the IV is *not* considered part of the ciphertext). Furthermore, it usually seems to be no extra trouble—indeed often it is slightly simpler—to directly demonstrate that some scheme achieves ind\$-security. Doing so is useful because an encryption scheme that satisfies ind\$ makes a more versatile tool: it can be used to directly provide a pseudorandom generator or a pseudorandom function. Finally, we find ind\$ seems to us conceptually simpler and easier to work with. For all of these reasons, we like ind\$ as the basic notion of security for building practical IV-based symmetric encryption schemes. (The counter-argument is that being indistinguishable from random bits is irrelevant

Algorithm CBC1.Encrypt$_K^N(M)$
if $|M| \notin \{n, 2n, 3n, \ldots\}$ **then return** $\star$
Parse M into $M_1 \cdots M_m$ where $|M_i| = n$
$C_0 \leftarrow E_K(N)$
for $i \leftarrow 1$ **to** m **do**
    $C_i \leftarrow E_K(C_{i-1} \oplus M_i)$
return $C_1 \cdots C_m$

Algorithm CBC1.Decrypt$_K^N(C)$
if $|C| \notin \{n, 2n, 3n, \ldots\}$ **then return** $\star$
Parse C into $C_1 \cdots C_m$ where $|M_i| = n$
$C_0 \leftarrow E_K(N)$
for $i \in [1 \mathinner{..} m]$ **do**
    $M_i \leftarrow C_{i-1} \oplus E_K^{-1}(C_i)$
return $M_1 \cdots M_m$

Fig. 3. Scheme CBC1. The scheme is not secure.

to the goal of encryption—one would argue that it goes beyond the intuition about what secure encryption needs to provide. This is true, and yet it has often proven desirable to use definitions that reach beyond the minimal notions that satisfy one's intuition.)

4 Insecure Schemes

One can see right away that CBC encryption, as formalized in Fig. 1, is not ind-secure (and therefore it is not ind\$-secure, either). Here is the attack. The adversary is trying to distinguish a real encryption oracle from a fake encryption oracle. Let us write $\mathbf{0}$ for 0^n and $\mathbf{1}$ for $0^{n-1}1$. The adversary asks a first oracle query of $(N_1, M_1) = (\mathbf{0}, \mathbf{0})$, getting back a ciphertext C_1. Let it then ask a second oracle query of $(N_2, M_2) = (\mathbf{1}, \mathbf{1})$, getting back a ciphertext C_2. If $C_1 = C_2$ then the adversary outputs 1 (it believes it has a real encryption oracle) and otherwise the adversary outputs 0 (it knows that it has a fake encryption oracle). The adversary is extremely efficient and has advantage close to 1.

The attack above motivates a natural alternative to CBC: encipher the IV before using it, as shown in Fig. 3. We call the scheme CBC1. The key space Key for the encryption scheme remains the key space for the underlying block cipher and the nonce space Nonce is $\{0, 1\}^n$.

The scheme CBC1 still doesn't work. Let the adversary ask query $(N_1, M_1) = (\mathbf{0}, \mathbf{00})$, obtaining ciphertext $C_1^1 C_1^2$ (where C_1^1 and C_1^2 are n bits). Note that if the adversary was provided a real encryption oracle then $C_1^2 = E_K(C_1^1)$. So next the adversary asks $(N_2, M_2) = (C_1^1, C_1^2 \oplus C_1^1)$, getting result C_2^1. If $C_2^1 = C_1^2$ then the adversary outputs 1 (it guesses that it has a real encryption oracle) and otherwise it outputs 0 (it is sure that it has a fake encryption oracle). The adversary is very efficient and is easily seen to have advantage close to 1.

5 Secure Schemes

Despite the two examples above, it is easy to construct encryption schemes that are secure in the ind\$-sense. Consider first the scheme CBC2 shown in Fig. 4. The key space for the encryption scheme is Key × Key, where Key is the key

Algorithm CBC2.Encrypt$^N_{K1\ K2}(M)$	**Algorithm** CBC2.Decrypt$^N_{K1\ K2}(C)$
if $\lvert M\rvert \notin \{n, 2n, 3n, \ldots\}$ **then return** $\star$	**if** $\lvert C\rvert \notin \{n, 2n, 3n, \ldots\}$ **then return** $\star$
Parse M into $M_1 \cdots M_m$ where $\lvert M_i\rvert = n$	Parse C into $C_1 \cdots C_m$ where $\lvert M_i\rvert = n$
$C_0 \leftarrow E_{K1}(N)$	$C_0 \leftarrow E_{K1}(N)$
for $i \leftarrow 1$ **to** m **do**	**for** $i \in [1 \mathinner{..} m]$ **do**
$\quad C_i \leftarrow E_{K2}(C_{i-1} \oplus M_i)$	$\quad M_i \leftarrow C_{i-1} \oplus E^{-1}_{K2}(C_i)$
return $C_1 \cdots C_m$	**return** $M_1 \cdots M_m$

Fig. 4. Scheme CBC2. The scheme is now ind\$-secure.

space for the underlying block cipher. The nonce space is $\{0,1\}^n$. The message space remains $(\{0,1\}^n)^+$.

The following result shows that CBC2 is a secure encryption scheme. We state the theorem in the information-theoretic setting. Passing to the complexity-theoretic case is standard. By $\mathcal{P}_n$ we mean the set of all permutations on $\{0,1\}^n$. These are block ciphers in the natural way. Thus by CBC2$[\mathcal{P}_n \times \mathcal{P}_n]$ we mean the scheme where E_{K1} and E_{K2} are random permutations from n bits to n bits.

Theorem 1. *Let* $n, \sigma \geq 1$. *Then* $\mathbf{Adv}^{\mathrm{ind\$}}_{\mathrm{CBC2}[\mathcal{P}_n \times \mathcal{P}_n]}(\sigma) \leq \sigma^2/2^n$ $\diamondsuit$

To avoid having two block-cipher keys one can modify the scheme using tricks like those from [4, 8]. However, it is not necessary to use a CBC-like scheme at all; simple forms of counter mode (CTR) work fine, and such modes have the advantage of being parallelizable and working directly on messages of any bit length. See Fig. 5 and Fig. 6 for two counter-based encryption schemes that are secure in the ind\$-sense. The first has a nonce space of $\{0,1\}^{n/2}$ (assume that n is even) and the second has a nonce space of $\{0,1\}^n$ but uses one extra block-cipher call. When S is an n-bit string and i is a number we denote by $S + i$ the n-bit string which is obtained by treating S as a number (msb first, lsb last), adding i modulo 2^n to this number, and then turning the result back into an n-bit string (msb first, lsb last).

One should anticipate use of CTR1 only on strings of at most $2^{n/2}$ blocks, though the ind\$-security of the scheme has already vanished by that point when the block cipher E is a PRP. Similarly, one should anticipate use of CTR2 only on strings of at most 2^n blocks, though the ind\$-security of the scheme has long before vanished when the function E is a PRP.

Theorem 2. *Let* $n, \sigma \geq 1$. *Then* $\mathbf{Adv}^{\mathrm{ind\$}}_{\mathrm{CTR1}[\mathcal{P}_n \times \mathcal{P}_n]}(\sigma) \leq \sigma^2/2^n$ $\diamondsuit$

Theorem 3. *Let* $n, \sigma \geq 1$. *Then* $\mathbf{Adv}^{\mathrm{ind\$}}_{\mathrm{CTR2}[\mathcal{P}_n \times \mathcal{P}_n]}(\sigma) \leq \sigma^2/2^n$ $\diamondsuit$

Recall our conventions that when multiple strings are encoded into a single one, as in a query (N, M), one sums the length of each component in the resource bound σ. This explains the absence of a term like $q^2/2^n$ in the second bound (where q is the number of queries).

Algorithm CTR1.Encrypt$_K^N(M)$	**Algorithm** CTR1.Decrypt$_K^N(C)$
$S \leftarrow N \parallel 0^{n/2}$	$S \leftarrow N \parallel 0^{n/2}$
$m \leftarrow \lceil \lvert M \rvert / n \rceil$	$m \leftarrow \lceil \lvert M \rvert / n \rceil$
$P \leftarrow E_K(S+0) \parallel \cdots \parallel E_K(S+m-1)$	$P \leftarrow E_K(S) \parallel \cdots \parallel E_K(S+m-1)$
$C \leftarrow M \oplus P$[first $\lvert M \rvert$ bits]	$M \leftarrow C \oplus P$[first $\lvert C \rvert$ bits]
return C	**return** M

Fig. 5. Scheme CTR1. The nonce space is Nonce $= \{0,1\}^{n/2}$.

Algorithm CTR2.Encrypt$_K^N(M)$	**Algorithm** CTR2.Decrypt$_K^N(C)$
$S \leftarrow E_K(N)$	$S \leftarrow E_K(N)$
$m \leftarrow \lceil \lvert M \rvert / n \rceil$	$m \leftarrow \lceil \lvert M \rvert / n \rceil$
$P \leftarrow E_K(S+0) \parallel \cdots \parallel E_K(S+m-1)$	$P \leftarrow E_K(S) \parallel \cdots \parallel E_K(S+m-1)$
$C \leftarrow M \oplus P$[first $\lvert M \rvert$ bits]	$M \leftarrow C \oplus P$[first $\lvert C \rvert$ bits]
return C	**return** M

Fig. 6. Scheme CTR2. The nonce space is Nonce $= \{0,1\}^n$.

6 Stronger Notions of Security

One desirable property of a nonce-based encryption scheme is that an adversarial-produced ciphertext, coupled with its nonce, should be deemed *invalid* by the receiver unless, of course, it is a copy a prior ciphertext and its nonce. We recall the definition of this property and then look at some other strong properties for a nonce-based encryption scheme.

AUTHENTICITY. A notion of *authenticity of ciphertexts* for nonce-based encryption schemes was formalized in [9, 10] following [2, 3, 6]. Fix an encryption scheme $\Pi = (\mathcal{E}, \mathcal{D})$ with key space Key. Let A be a nonce-respecting adversary having an encryption oracle $\mathcal{E}_K$. We say that A *forges* if it outputs a pair (N, C) such that C was not the response to any $\mathcal{E}_K(N, M)$ query and $\mathcal{D}_K^N(C) \neq \star$. We write

$$\mathbf{Adv}_{\Pi}^{\mathrm{auth}}(A) = \Pr\left[K \xleftarrow{\$} \mathsf{Key} :\ A^{\mathcal{E}_K(\cdot,\cdot)} \text{ forges}\right]$$

and lift this to give resource-bounded definitions in the usual way.

CHOSEN-CIPHERTEXT SECURITY. We define indistinguishability from random bits under an adaptive chosen-plaintext-and-ciphertext-and-IV attack. The defining game is as with ind\$-cpa except that the adversary is given access to a decryption oracle as well. Queries may not be repeated, and one forbids the adversary from making a decryption query of (N, C) if the adversary already encrypted some (N, M) and got back an answer C; and one similarly forbids the adversary from encrypting (N, M) if the adversary already decrypted some

(N, C) and got back an answer M. These restrictions must hold regardless of the adversary's coins and query responses. Only such an adversary is deemed to be *valid*.

In defining chosen-ciphertext security one restricts attention to valid, nonce-respecting adversaries. Be clear that the nonce-respecting condition applies only to encryption-queries; the adversary is free to repeat nonces in its decryption oracle. This reflects the understanding that the party encrypting a message is the one that is responsible for providing fresh nonces; the receiver may be stateless.

Let adversary A be a valid, nonce-respecting adversary and let $\Pi = (\mathcal{E}, \mathcal{D})$ be a nonce-based encryption scheme with key space Key. We define

$$\mathbf{Adv}_{\Pi}^{\text{ind\$-cca}}(A) = \Pr\left[K \stackrel{\$}{\leftarrow} \mathsf{Key}:\ A^{\mathcal{E}_K(\cdot,\cdot)\ \mathcal{D}_K(\cdot,\cdot)} \Rightarrow 1\right] - \Pr\left[A^{\$(\cdot,\cdot)\ \mathcal{D}_K(\cdot,\cdot)} \Rightarrow 1\right]$$

The notion can be modified to ind-cca in the natural way.

Nonmalleability. The notion of nonmalleability [5] can likewise be adapted to the nonce-based setting. The adversary's goal will be to create a ciphertext (perhaps by modifying ciphertexts that have already been seen) whose underlying plaintext is something about which the adversary can say something interesting. More specifically, the adversary will output a tuple (N, C, f, Y) where N is a nonce and C is a ciphertext and $f\colon \{0,1\}^* \cup \{\star\} \to \{0,1\}^*$ is a function (encoded as a program) and Y is a string. The output should be interpreted as the adversary guessing that the value of $f(M)$ is Y, where $M = \mathcal{D}_K^N(C)$. The formalization captures the idea that the adversary should be right about this guess no more often than that which is inherent for the game.

We define nonmalleability under chosen-ciphertext attack (meaning chosen-plaintext-and-ciphertext-and-IV attack). Fix an encryption scheme $\Pi = (\mathcal{E}, \mathcal{D})$ having key space Key. Consider a valid, nonce-respecting adversary A with access to oracles $\mathcal{E}_K(\cdot,\cdot)$ and $\mathcal{D}_K(\cdot,\cdot)$. At the end of the adversary's execution, let $\mathit{Dec}(N, C)$ be $\{M\}$ if the adversary asked some query $\mathcal{E}_K(N, M)$ and this returned C or the adversary asked some query $\mathcal{D}_K(N, C)$ and this returned M. If the adversary asked no such query, let $\mathit{Dec}(N, C) = \star$. One can regard $\mathit{Dec}(N, C)$ as a "fake" decryption of C for nonce N: if the adversary trivially knows the decryption to be M then the value is M; otherwise, the value is the "guess" that the ciphertext is invalid. Then define $\mathbf{Adv}_{\Pi}^{\text{nm-cca}}(A)$ as

$$\Pr\left[K \stackrel{\$}{\leftarrow} \mathsf{Key};\ (N, C, f, Y) \stackrel{\$}{\leftarrow} A^{\mathcal{E}_K(\cdot,\cdot)\ \mathcal{D}_K(\cdot,\cdot)};\ M \leftarrow \mathcal{D}_K(N, C):\ f(M) = Y\right] -$$
$$\Pr\left[K \stackrel{\$}{\leftarrow} \mathsf{Key};\ (N, C, f, Y) \stackrel{\$}{\leftarrow} A^{\mathcal{E}_K(\cdot,\cdot)\ \mathcal{D}_K(\cdot,\cdot)};\ M \leftarrow \mathit{Dec}(N, C):\ f(M) = Y\right]\ .$$

The corresponding notion for nonmalleability under a chosen-plaintext attack (nm-cpa) is obtained by insisting that the adversary asks no decryption queries.

Though the above notions might not look like nonmalleability, really they are: the case of creating a ciphertext C whose plaintext M is related to the plaintext M' of some other ciphertext C' is just a special case.

Implications and separations. As with probabilistic encryption, one can work out the complete set of implications and separations between the defined notions of nonce-based encryption. The most useful relations are that ind$ plus auth implies both ind$-cca and nm-cca. The intuition is clear: the auth-condition effectively renders useless the decryption oracle, since it almost always returns an answer that the adversary can anticipate. We omit further details.

Achieving ind$+auth by generic composition. None of the ind$-secure encryption schemes given so far (CBC$, CTR1, CTR2) achieve the auth-notion of authenticity (nor do they achieve ind-cca or nm-cca). We now explain the most natural way to modify an ind$-secure encryption scheme so as to achieve authenticity (while preserving ind$-security, of course).

Let $\Pi = (\mathcal{E}, \mathcal{D})$ be a nonce-based encryption scheme having nonce space $\mathsf{Nonce} = \{0,1\}^n$ and key space Key. Think of Π as being ind$-secure. Let $F\colon \mathsf{Key}' \times \mathsf{Dom} \to \{0,1\}^n$ be a function. Think of it as being good as a pseudorandom function. We want to combine Π and F to give an encryption scheme $\bar{\Pi} = (\bar{\mathcal{E}}, \bar{\mathcal{D}})$ that will be ind$-secure and auth-secure. The simplest possibilities are as follows.

- *Encrypt-then-PRF.* Let $\bar{\mathcal{E}}^N_{K\,K'}(M) = C \parallel T$ where $T = F_{K'}(N \parallel C)$ and $C = \mathcal{E}^N_K(M)$. Decryption (including the test for authenticity) proceeds in the natural way.
- *PRF-then-Encrypt.* Let $\bar{\mathcal{E}}^N_{K\,K'}(M) = \mathcal{E}^N_K(M \parallel T)$ where $T = F_{K'}(N \parallel M)$. Decryption (including the test for authenticity) proceeds in the natural way.

The definition above assumes that the encryption scheme Π and the PRF F have appropriately matching domains.

The situation is different from conventional probabilistic encryption [2]; for nonce-based encryption, *both* encrypt-then-PRF and PRF-then-encrypt work correctly. The proofs are straightforward; see [10] for the slightly more complex setting in which "associated data" is present as well.

7 Directions

With the syntax of an encryption having been modified to surface the IV, a number of weaker notions of security for IV-based encryption make sense. For example, to capture the requirement that "the IVs are to be some fixed sequence of distinct values" have the adversary provide a deterministic algorithm F that gives distinct n-bit strings $F(1), F(2), \ldots, F(2^n)$. Require indistinguishability from random bits with respect to the resulting scheme.

In this paper we have only treated symmetric encryption. Public-key encryption schemes traditionally do *not* surface an IV. But they do use random bits, and it makes just as much sense to consider nonce-based public-key encryption schemes as it does to consider nonce-based symmetric encryption schemes. This provides an approach to effectively weakening the requirement for randomness on the sender.

Acknowledgments

Some of the ideas of this note were developed in the course of preparing some lectures for Helsinki University of Technology, Finland (April 2002); thanks to Helger Lipmaa for inviting me to give those lectures. Kind thanks also to Mihir Bellare, John Black, and Tom Shrimpton for their useful comments and suggestions. This work was supported under NSF CCR-0208842 and a gift from CISCO Systems.

References

1. M. Bellare, A. Desai, E. Jokipii and P. Rogaway. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. *Proceedings of 38th Annual Symposium on Foundations of Computer Science* (FOCS 97), IEEE, 1997.
2. 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.
3. 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.
4. J. Black and P. Rogaway. CBC MACs for arbitrary-length messages: The three-key constructions. *Advances in Cryptology—CRYPTO '00,* Lecture Notes in Computer Science, vol. 1880, M. Bellare, ed., Springer-Verlag, pp. 197–215, Aug 2000.
5. D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM J. Computing*, vol. 30, no. 2, pp. 391–437, 2000.
6. J. Katz and M. Yung. Unforgeable encryption and chosen ciphertext secure modes of operation. *Fast Software Encryption* (FSE 2000), Lecture Notes in Computer Science, vol 1978, B. Schneier, ed., Springer, pp. 284–299, 2001.
7. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, vol. 28, pp. 270–299, April 1984.
8. T. Iwata and K. Kurosawa. One-key CBC MAC. *Fast Software Encryption* (FSE 2003). Lecture Notes in Computer Science (to appear), 2003.
9. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. *Proceedings of the 8th ACM Conference on Computer and Communications Security* (CCS '01), ACM Press, pp. 196–205, 2001.
10. P. Rogaway. Authenticated-encryption with associated-data. *Proceedings of the 9th ACM Conference on Computer and Communications Security* (CCS '02), ACM Press, pp. 98–107, 2002.