Unforgeable Encryption and Chosen Ciphertext Secure Modes of Operation

Jonathan Katz[1] and Moti Yung[2]

[1] Department of Computer Science, Columbia University
jkatz@cs.columbia.edu
[2] CertCo, NY, USA
moti@cs.columbia.edu, moti@certco.com

Abstract. We find certain neglected issues in the study of private-key encryption schemes. For one, private-key encryption is generally held to the same standard of security as public-key encryption (i.e., indistinguishability) even though usage of the two is very different. Secondly, though the importance of secure encryption of single blocks is well known, the security of modes of encryption (used to encrypt multiple blocks) is often ignored. With this in mind, we present definitions of a new notion of security for private-key encryption called *encryption unforgeability* which captures an adversary's inability to generate valid ciphertexts. We show applications of this definition to authentication protocols and adaptive chosen ciphertext security.
Additionally, we present and analyze a new mode of encryption, RPC (for Related Plaintext Chaining), which is unforgeable in the strongest sense of the above definition. This gives the first mode provably secure against chosen ciphertext attacks. Although RPC is slightly less efficient than, say, CBC mode (requiring about 33% more block cipher applications and having ciphertext expansion of the same amount when using a block cipher with 128-bit blocksize), it has highly parallelizable encryption and decryption operations.

1 Introduction

1.1 Motivation

Much work has been devoted to developing precise definitions of security for encryption schemes [2,3,16] and to constructing cryptosystems meeting these enhanced notions of security. Currently, the same notions of security are used to analyze both public-key and private-key encryption. In the public-key setting, however, encryption is available to everyone; in this case, therefore, one need only worry about the possibility of an adversary decrypting an encrypted message. This is in contrast to the private-key setting where one must also worry about the potential harm an adversary can cause by generating the encryption of some message (an action which a protocol designer may not expect to occur). So, while one can "borrow" security notions from the public-key to the private-key setting, one has to recognize that the security goals of the latter may be different.

B. Schneier (Ed.): FSE 2000, LNCS 1978, pp. 284–299, 2001.

Additionally, private-key cryptography is used to transmit large amounts of data (in particular, more than one block at a time using some mode of operation), while public-key cryptography is generally used to send short messages (typically, session keys). Finally, private-key encryption is the basis of many authentication and security handshake protocols [14].

For these reasons, we introduce here a higher standard of security for private-key encryption called *encryption unforgeability* (simply *unforgeability* in the sequel), which characterizes an adversary's inability to generate valid ciphertexts. This notion turns out to be quite useful: it guarantees security for certain authentication protocols, provides message integrity without additional computation or cryptographic primitives, and offers some level of security under chosen ciphertext attacks.

Unforgeability is particularly interesting in the context of modes of encryption. The encryption of single blocks, both in theory and in practice, is well understood; however, we believe that the security of modes of encryption has been (comparatively) neglected. To remedy this, we present a new mode of encryption which is unforgeable in the strongest sense of the definition. We then show, using a concrete security analysis [2], that this mode is secure against the strongest form of chosen plaintext/ciphertext attacks, and is non-malleable as well.

1.2 Applicability

Unforgeability seeks to capture the following intuition: an adversary, after intercepting various ciphertext messages, should not be able to generate a new ciphertext corresponding to any valid plaintext. As an application of this, say Alice and Bob carry out a handshaking protocol using a shared private key K. Alice sends Bob $\mathcal{E}_K(\mathsf{timestamp})$, and Bob must reply with $\mathcal{E}_K(\mathsf{timestamp} + 1)$ (this is similar to the protocol implemented in Kerberos V4 [14]). This is meant to prove to Alice that Bob knows the secret key K. However, if an adversary can somehow "forge" an encryption of $\mathsf{timestamp} + 1$, he can authenticate himself to Alice without knowledge of the key. Note that this differs from a non-malleability attack; in the case of non-malleability, the adversary does not know the plaintext corresponding to the "challenge" ciphertext. In this case, however, the current time is known to all participants, so Bob *does* know the plaintext corresponding to Alice's encrypted message.

Extending this further, if Alice and Bob are communicating over an insecure channel by encrypting messages using a shared key, it is clearly undesirable for an adversary to be able to insert ciphertext into the channel which will be decrypted by one of the parties and interpreted as a valid (potentially malicious) message. This threat can be reduced by using message authentication, but most schemes for integrating encryption and authentication are either inefficient or are not provably secure. In fact, efficient integration of encryption, authentication, and message integrity *using a single shared key* is an important and intensely studied problem in network security [14,23] which is solved by the use of an unforgeable encryption scheme.

1.3 Importance of a Single-Key Solution

A trivial solution to the problem of unforgeability is to share two keys—one for encryption and one for authentication (e.g., using a MAC). Also, various constructions (requiring multiple shared keys) are known to exist which allow for chosen-ciphertext secure encryption of large (i.e., many block) messages [20,5,8]. However, we have in mind applications such as integrated public-key/private-key encryption (e.g., PGP for e-mail encryption) in which it is advantageous to share only one key. One public-key encryption of a session key followed by (slow) private-key encryption of the message is still faster than two public-key encryptions and (fast) private-key encryption, for "short" messages. Hardware implementations of encryption may also benefit from the single-key requirement. Furthermore, it may be required to integrate a new mode of encryption into existing software which already calls for encryption of only one session key.

1.4 Previous Work

Notions of Security. Beginning with the paper by Goldwasser and Micali [10], which first provided a rigorous definition of "semantic security", the cryptography community has progressed to the currently accepted definitions of indistinguishability (polynomially equivalent to semantic security) and non-malleability (introduced by Dolev, Dwork, and Naor [9]; later reformulated by Bellare, et al. [3]). Indistinguishability describes an adversary's inability to derive any information from a ciphertext about the corresponding plaintext. Non-malleability characterizes an adversary's inability, given access to a challenge ciphertext, to generate a different ciphertext meaningfully related to the challenge ciphertext. We refer the reader elsewhere [3,16] for formal definitions.

Unforgeability. Previous work dealing with concurrent encryption plus message authentication implicitly uses many of the ideas of unforgeability [13,23]. However, the formal definition presented here is new.

Unforgeability is distinct from non-malleability. In the latter, the adversary's goal is to generate one ciphertext meaningfully related to another. In the former (depending on the type of attack, see below), the adversary may succeed by generating *any* valid ciphertext. Furthermore, in a non-malleability-based attack, the adversary does not know the plaintext corresponding to the challenge ciphertext. This is in contrast to an unforgeability-based attack, in which queries are made to an encryption oracle and therefore the adversary may know the underlying decryption of some ciphertexts. Thus, the level of security considered here is much stronger.

Chosen Ciphertext Security. Chosen ciphertext [21] and adaptive chosen ciphertext [25] attacks are very powerful attacks in which the adversary can obtain decryptions of her choice (in the case of adaptive attacks, even after seeing the challenge ciphertext). As it is not our intention to survey the literature on chosen ciphertext security, we merely point out that most research thus far has focused on public-key encryption. Little attention has been paid to chosen ciphertext security for private-key encryption (an exception is [9]), and even less to chosen ciphertext secure modes of encryption.

Modes of Encryption. The commonly-used modes of encryption include those defined as part of the DES [1,11,22], an XOR mode suggested by Bellare, et al. [2], and the PCBC mode [19] used in Kerberos V4. The security of these modes of encryption lags behind the security of available block encryption algorithms. None of the above modes are non-malleable, and all are vulnerable to an adaptive chosen ciphertext attack. This has been recognized in the cryptographic literature for some time [19], but the previously-mentioned modes continue to be used even though potentially serious security flaws may result [15].

Security analyses of modes of encryption have focused on chosen plaintext attacks. Examples include a concrete analysis of the CBC and XOR modes [2], Biham's and Knudsen's analyses of modes of operation of DES and triple-DES [6,7], and others [13,24].

A mode of encryption as secure as a pseudorandom permutation is given by Bellare and Rogaway [5]. The only other examples of (potentially) chosen-ciphertext-secure modes of encryption of which we are aware [20,8] study a slightly different problem, and are therefore not *provably* chosen-ciphertext secure. Our suggestion (RPC) lends itself to simpler analysis and has certain practical advantages over these other constructions; we refer the reader to the Discussion in Section 5.

1.5 Summary of Results

We begin in Section 2 with a review of some notation. In Section 3 we present definitions for three levels of unforgeability for private-key encryption; this section concludes with a theorem formalizing the relation between security in the sense of unforgeability and security under chosen ciphertext attacks. Section 4 describes two simple modes of encryption which are unforgeable under the strongest definition. We conclude with a discussion of the practicality of these modes, and a comparison with previously suggested modes which are potentially chosen ciphertext secure.

The unforgeable modes we present are actually quite straightforward. The intuition is as follows: to prevent an adversary from "splicing" together blocks from different ciphertexts, we "tag" each ciphertext block with a sequence number. To prevent an adversary from shortening a ciphertext to generate a new, valid one, we "mark" the beginning and end of each ciphertext with special start and end tokens. Encryption of $x_1, \ldots, x_n$ is simply given by:

$$\mathcal{E}(\mathsf{start}, i), \mathcal{E}(x_1, i+1), \ldots, \mathcal{E}(x_n, i+n), \mathcal{E}(\mathsf{end}, i+n+1),$$

where the nature of i depends on the details of the mode.

It follows from the properties of RPC that it can provide a single-key, one-pass, provably secure mechanism for "concurrent encryption, authentication, and message integrity" using a single shared key (an open question [14,23]). The encryption and decryption operations are parallelizable, and the mode works better with larger blocksizes (i.e., it is better suited for AES than for DES). Because of these properties and its simplicity, we suggest RPC as a practical addition to (or substitute for) the modes of encryption currently in use.

2 Notation

Indistinguishability. We follow the standard notation for private-key encryption [16], but note that we consider both probabilistic and stateful encryption schemes. We use the notion of indistinguishability as our measure of security. The definition below is similar to those given elsewhere [2,16]; we rephrase it to allow for a concrete security analysis. Also, since we deal explicitly with the security of modes of encryption, we parameterize the adversary's advantage by the length of the submitted plaintexts. The notation for notions of security follows [16]; thus, IND-PX-CY means an indistinguishability-based attack, with encryption oracle access at level X and decryption oracle access at level Y. Level 0 denotes no oracle access, level 1 denotes access before being presented with the challenge ciphertext (non-adaptive access), and level 2 denotes access both before *and after* the challenge ciphertext is revealed (adaptive access). Of course, different levels of access can be chosen separately for each oracle. The definition below corresponds to security under an IND-P2-C2 attack.

Definition 1. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an encryption scheme accepting variable length messages, and let $A = (A_1, A_2)$ be an adversary. Let* $\mathsf{Adv}_{A,\Pi}^{\mathrm{IND-P2-C2}} \stackrel{\mathrm{def}}{=}$

$$2 \cdot \Pr \Big[sk \leftarrow \mathcal{K}; (x_0, x_1, s) \leftarrow A_1^{\mathcal{E}_{sk}(\cdot), \mathcal{D}_{sk}(\cdot)}; b \leftarrow \{0,1\}; y \leftarrow \mathcal{E}_{sk}(x_b) : \\ A_2^{\mathcal{E}_{sk}(\cdot), \mathcal{D}_{sk}(\cdot)}(x_0, x_1, s, y) = b \Big] - 1.$$

We insist that $|x_0| = |x_1| = \ell$. Furthermore, all queries to the encryption oracle consist of an integral number of blocks (the size of which depends on the underlying block cipher algorithm); thus, no "padding" is ever required.

We say that Π is $(t, q_e, b_e, q_d; \ell, \epsilon)$-secure in the sense of IND-P2-C2 if for any adversary A distinguishing between plaintexts of length ℓ which runs in time at most t, submits at most q_e queries to the encryption oracle (these totaling at most b_e bits), and submits at most q_d queries to the decryption oracle we have $\mathsf{Adv}_{A,\Pi}^{\mathrm{IND-P2-C2}} \leq \epsilon$.

Note that IND-P2-C2 security implies security under all other notions of indistinguishability and non-malleability [16].

Super Pseudorandom Permutations (following [2]). A *permutation family* is a set F of permutations all having the same domain and range. We assume the permutations are indexed by some key $k \in K$, such that F_k specifies a particular permutation in F. Usually, K is the set of all strings of some fixed length. We write $f \leftarrow F$ to denote selection of a permutation at random from F according to the distribution given by picking a random k and setting $f = F_k$. Let P_n be the permutation family consisting of all permutations on $\{0,1\}^n$. Thus, $f \leftarrow P_n$ means selection of a random permutation on n-bit strings.

Let F, G be function families with the same domain and range. Consider a *distinguisher* D, given oracle access to a function f and its inverse f^{-1}, that attempts to distinguish between the case where f is chosen randomly from F and the case where f is chosen randomly from G. Let

$$\mathsf{Dist}_D(F,G) = \Pr\left[f \leftarrow F : D^{f(\cdot),f^{-1}(\cdot)} = 1\right] - \Pr\left[f \leftarrow G : D^{f(\cdot),f^{-1}(\cdot)} = 1\right].$$

A super pseudorandom permutation (super-PRP) [18] family F on $\{0,1\}^n$ has the property that the input-output behavior of f, f^{-1} "looks random" to someone who does not know the randomly selected key k. Accordingly, define the advantage of the distinguisher by:

$$\mathsf{Adv}_D^{\mathsf{SPRP}}(F) = \mathsf{Dist}_D(F, P_n).$$

Definition 2. *We say that super-PRP family F is $(t, q_1, q_2; \epsilon)$-secure if for any distinguisher D which makes at most q_1 oracle queries of f, q_2 oracle queries of f^{-1}, and runs in time at most t it is the case that $\mathsf{Adv}_D^{\mathsf{SPRP}}(F) \leq \epsilon$.*

Note the distinction from a pseudorandom permutation (PRP); in the latter case, the distinguisher is only given access to the function f (not its inverse f^{-1}). However, any family of PRPs can be converted to a family of super-PRPs [18].

3 Unforgeability

3.1 Definitions

We define three levels of unforgeability for encryption, progressing from the weakest to the strongest. In the following definitions, we assume that the set of valid encryption oracle queries is exactly the same as the valid message space; this eliminates technical problems arising from having to randomly pad short messages. Note that in all cases our definitions do not restrict the length of the forged ciphertext or the length of the plaintext queries submitted to the encryption oracle. Thus, our definitions include cases where an adversary might try to extend previous ciphertexts, paste two ciphertexts together, or delete blocks from a valid ciphertext in an attempt to forge a message.

Our definitions focus on the settings with maximum access to an encryption oracle. Weaker definitions, with non-adaptive access or no access, are also possible.

Random Plaintext Unforgeability. The framework of the attack is as follows: a challenge plaintext x is chosen at random from the message space M. The adversary succeeds if he can output a ciphertext y such that x is the decryption of y. This essentially means that the adversary has "broken" one direction of the encryption, since he can forge ciphertext for just about any message he chooses.

Definition 3. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a private-key encryption scheme, and let A be an adversary. Define:*

$$\mathsf{Adv}^{\text{random}}_{A,\Pi} \stackrel{\text{def}}{=} \Pr\left[sk \leftarrow \mathcal{K}; x \leftarrow M; y \leftarrow A^{\mathcal{E}_{sk}(\cdot)}(x) : \mathcal{D}_{sk}(y) = x\right].$$

We insist that A does not ask the oracle to encrypt x. We say that Π is $(t, q, b; \epsilon)$-secure in the sense of random plaintext unforgeability if for any adversary A which runs in time at most t and asks at most q queries, these totaling at most b bits, we have $\mathsf{Adv}^{\text{random}}_{A,\Pi} \le \epsilon$.

Chosen Plaintext Unforgeability. In this attack, the goal of the adversary is simpler. Instead of having to find the encryption of a "challenge" plaintext, the adversary is free to output the encryption of any plaintext he chooses. But, the adversary must know the plaintext message to which this ciphertext corresponds. This is similar to the *valid pair creation* attack defined previously [12].

Definition 4. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a private-key encryption scheme, and let A be an adversary. Define:*

$$\mathsf{Adv}^{\text{chosen}}_{A,\Pi} \stackrel{\text{def}}{=} \Pr\left[sk \leftarrow \mathcal{K}; (x, y) \leftarrow A^{\mathcal{E}_{sk}(\cdot)} : \mathcal{D}_{sk}(y) = x\right].$$

We insist, above, that A has never received ciphertext y in return from its encryption oracle. We say that Π is $(t, q, b; \epsilon)$-secure in the sense of chosen plaintext unforgeability if for any adversary A which runs in time at most t and asks at most q queries, these totaling at most b bits, we have $\mathsf{Adv}^{\text{chosen}}_{A,\Pi} \le \epsilon$.

Existential Unforgeability. This represents the strongest notion of unforgeability, as it corresponds to the simplest attack for an adversary. The adversary succeeds by producing *any* new valid ciphertext, even without knowing the corresponding plaintext.

Definition 5. *Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a private-key encryption scheme, and let A be an adversary. Define:*

$$\mathsf{Adv}^{\text{exist}}_{A,\Pi} \stackrel{\text{def}}{=} \Pr\left[sk \leftarrow \mathcal{K}; y \leftarrow A^{\mathcal{E}_{sk}(\cdot)} : \mathcal{D}_{sk}(y) \ne \perp\right].$$

We insist, above, that A has never received ciphertext y in return from its encryption oracle. We say that Π is $(t, q, b; \epsilon)$-secure in the sense of existential unforgeability if for any adversary A which runs in time at most t and asks at most q queries, these totaling at most b bits, we have $\mathsf{Adv}^{\text{exist}}_{A,\Pi} \le \epsilon$.

3.2 Unforgeability and Chosen Ciphertext Security

The notion of existential unforgeability is a strong one; as such, we expect some relation between existential unforgeability and chosen ciphertext security. The intuition is clear: since any (new) ciphertexts generated by an adversary are likely to be invalid, the adversary cannot gain much by submitting them to the decryption oracle. Thus, if a scheme cannot be broken with no access to a decryption oracle, it cannot be broken much more often even when access to a decryption oracle is given. This is formalized by the following theorem, which shows that unforgeability (along with chosen plaintext security) implies adaptive chosen plaintext/ciphertext security.

Theorem 1. *Let Π be an encryption scheme which is $(t_1, q, b; \epsilon_1)$-secure in the sense of existential unforgeability, and $(t_2, q_e, b_e; \ell, \epsilon_2)$-secure in the sense of IND-PX-C0 (for $X \in \{0, 1, 2\}$). Then Π is $(t', q'_e, b'_e, q_d; \ell, \epsilon')$-secure in the sense of IND-PX-C2, where $t' = \min\{t_1, t_2\}$; $q'_e = \min\{q-1, q_e\}$; $b'_e = \min\{b-\ell, b_e\}$; and $\epsilon' = \epsilon_2 + q_d\epsilon_1$.*

Sketch of Proof Say adversary A attacks Π in the sense of IND-PX-C2, running in time t', making at most q'_e encryption oracle queries totaling at most b'_e bits, and making q_d decryption oracle queries. We assume without loss of generality that A never queries the decryption oracle on a ciphertext which it received in return from the encryption oracle (in fact, there is no reason for A to do so). We can construct the following adversaries:

1. Adversaries $\{B_i\}$ (for $i = 1, \ldots, q_d$) attacking Π in the sense of existential unforgeability. B_i runs A as a subroutine, and answers the first $i-1$ decryption oracle queries made by A with $\perp$, then returns as its output the i^{th} decryption oracle query made by A.
2. Adversary C attacking Π in the sense of IND-PX-C0, which runs A as a subroutine but answers all decryption oracle queries made by A with $\perp$.

Define Valid_i to be the event that the i^{th} decryption oracle query submitted by A was the first one to be valid. Extending this notation, let Valid_∞ be the event that none of the decryption oracle queries submitted by A are valid. Let Succ be (informally) the event that A succeeds in distinguishing the challenge ciphertext it is given. We have:

$$\begin{aligned}
\mathsf{Adv}_{A,\Pi}^{\mathrm{IND-PX-C2}} &= \\
&\Pr\left[\mathsf{Succ}|\mathsf{Valid}_{q_d} \vee \cdots \vee \mathsf{Valid}_1\right] \Pr\left[\mathsf{Valid}_{q_d} \vee \cdots \vee \mathsf{Valid}_1\right] \\
&\;+ \Pr\left[\mathsf{Succ}|\mathsf{Valid}_\infty\right] \Pr\left[\mathsf{Valid}_\infty\right] \\
&\le \sum_{i=1}^{q_d} \Pr\left[\mathsf{Valid}_i\right] + \Pr\left[\mathsf{Succ}|\mathsf{Valid}_\infty\right] \Pr\left[\mathsf{Valid}_\infty\right] \\
&\le \sum_{i=1}^{q_d} \mathsf{Adv}_{B_i,\Pi}^{\mathrm{exist}} + \mathsf{Adv}_{C,\Pi}^{\mathrm{IND-PX-C0}}.
\end{aligned}$$

Since the advantages of adversaries $\{B_i\}$ and C are bounded by ϵ_1 and ϵ_2 respectively, we have:

$$\mathsf{Adv}_{A,\Pi}^{\mathrm{IND-PX-C2}} \leq q_d\epsilon_1 + \epsilon_2.$$

This gives the stated result. □

3.3 The Forgeability of Previous Modes

It is instructive to demonstrate attacks on the original DES modes of encryption [1,11,22]. It is clear that all of these modes are susceptible to an existential unforgeability attack (even a passive attack, with no encryption oracle access), since any ciphertext string of appropriate length decrypts to a valid plaintext. Some modes are even weaker. ECB mode is susceptible to a random plaintext unforgeability attack (with adaptive encryption oracle access) as follows: to find the encryption of $M_1, M_2, \ldots, M_l$ simply submit to the encryption oracle the two queries M_1 and $M_2, \ldots, M_l$ and paste the responses together. OFB has even worse characteristics—it is susceptible to a random plaintext unforgeability attack with *non-adaptive* encryption oracle access. Simply have the adversary submit $0^{n\ell}$ to the encryption oracle and receive ciphertext $C_0, C_1, \ldots, C_\ell$. To forge encryption of $M_1, \ldots, M_\ell$, compute $C_i' = C_i \oplus M_i$ and return $C_0, C_1', \ldots, C_\ell'$. Attacks in the sense of chosen plaintext unforgeability exist for CBC and CFB modes as well [15,19]. These examples indicate the weaknesses of these modes; this further implies that these modes are not adaptive-chosen-ciphertext secure.

4 Unforgeable Modes of Operation

We present two modes of encryption (one stateful, one probabilistic) which are secure under the strongest definition of unforgeability and are additionally secure under adaptive chosen plaintext/ciphertext attack. The encryption (decryption) algorithms use underlying block cipher (using secret key *sk*) F_{sk}. The mode is parameterized by n and r, where the underlying block cipher operates on n-bit blocks and r is the length of the padding.

4.1 A Stateful Mode

We begin by describing the stateful mode of encryption. The variable *ctr* is an r-bit binary number; it is initialized to 0, and addition is modulo 2^r. (We assume that the start and end symbols do not represent valid message blocks.)

```
Algorithm $\mathcal{E}$-RPC$_{n,r}(ctr, M)$
  parse $M$ as $M_1, \ldots, M_\ell$,
   where $|M_i| = n - r$
  $C_0 = F_{sk}(\mathsf{start}, ctr)$
  for $i = 1, \ldots, \ell$ do
    $C_i = F_{sk}(M_i, ctr + i)$
  $C_{\ell+1} = F_{sk}(\mathsf{end}, ctr + \ell + 1)$
  $ctr := ctr + \ell + 1$
  return $(ctr, C = C_0, \ldots, C_{\ell+1})$
```

```
Algorithm $\mathcal{D}$-RPC$_{n,r}(C)$
  parse $C$ as $C_0, \ldots, C_{\ell+1}$,
   where $|C_i| = n$
  if $\ell + 1 < 3$ return $\perp$
  for $i = 0, \ldots, \ell + 1$ do
    $(M_i, ctr_i) = F_{sk}^{-1}(C_i)$
  if $(M_0 \neq \mathsf{start}) \vee (M_{\ell+1} \neq \mathsf{end})$ return $\perp$
  for $i = 1, \ldots, \ell$ do
    if $ctr_i \neq ctr_0 + i$ return $\perp$
    if $(M_i = \mathsf{start}) \ \vee (M_i = \mathsf{end})$ return $\perp$
  if $ctr_{\ell+1} \neq ctr_0 + \ell + 1$ return $\perp$
  return $M = M_1, \ldots, M_\ell$
```

Theorems 2 and 3 quantify the security of RPC mode when instantiated with a fully random permutation:

Theorem 2. *Let Π be an encryption scheme using RPC$_{n,r}$ mode instantiated with a block cipher chosen randomly from P_n. Then Π is $(t, q, b(n-r); \epsilon)$-secure in the sense of existential unforgeability (for $b + q \leq 2^r$), where:*

$$\epsilon \leq \frac{2^{n-r-1}}{2^n - b - 2q}.$$

Sketch of Proof Recall the technical assumption that the adversary submits oracle queries whose lengths are integer multiples of $n - r$ bits (i.e., no padding is required). Without this assumption, it is unclear how to define a notion of unforgeability. Thus, the adversary submits b blocks of plaintext to the oracle.

Due to the construction of the mode, the adversary will have to introduce at least one new (previously-unseen) ciphertext block in the output ciphertext. The *ctr* variable derived from this block must "match up" with the remainder of the message. There are at most 2^{n-r-1} blocks whose *ctr* will match up properly (there are $n - r$ data bits, and these cannot take on the values start or end). Furthermore, there is a pool of at least $2^n - b - 2q$ ciphertext blocks to choose from after eliminating those blocks which the adversary has already received in return from the encryption oracle (b blocks of data and $2q$ blocks to account for encryption of start and end tokens). Since we are dealing with a random permutation, this gives the stated result. □

This bound is tight, as an adversary can submit one plaintext block to its encryption oracle, receive in return ciphertext $C = C_0, C_1, C_2$, and then output $C' = C_0, C_1', C_2$ as an attempted forgery. Clearly, C' is valid iff the *ctr* variable derived from C_1' "matches up" with the remainder of the ciphertext, and this occurs with the probability specified in the theorem.

Theorem 3. *Let Π be an encryption scheme using $RPC_{n,r}$ mode instantiated with a block cipher chosen randomly from P_n. Then Π is $(t, q_e, b_e(n-r), q_d; \ell, \epsilon)$-secure in the sense of IND-P2-C2 (for $b_e + q_e + \ell + 2 \leq 2^r$), where:*

$$\epsilon = \frac{q_d 2^{n-r-1}}{2^n - b_e - 2q_e}.$$

Proof Due to the stateful mode of operation and the fact it uses a random permutation, the advantage of any adversary attacking Π in the sense of IND-P2-C0 is 0. Application of Theorem 1 gives the desired result. □

These results translate into the following "real-world" security:

Theorem 4. *Suppose F is a $(t, q_1, q_2; \epsilon)$-secure super-PRP family ($q_2 \geq 2$). Let Π be an encryption scheme using $RPC_{n,r}$ mode instantiated with a block cipher chosen randomly from F. Then Π is $(O(t - b \log b), q, b(n-r); \epsilon')$-secure in the sense of existential unforgeability (for $b + q \leq 2^r$ and $b + 2q \leq q_1$), where:*

$$\epsilon' = \epsilon + \frac{2^{n-r-1}}{2^n - b - 2q}.$$

Sketch of Proof Assume adversary A attacks Π in the sense of existential unforgeability. Without loss of generality, we may assume that A does not output a ciphertext which it has obtained in response from its encryption oracle. Construct a distinguisher D for F which will use A as a subroutine. D simulates A's encryption oracle by maintaining an internal *ctr* variable, "padding" A's oracle queries according to the definition of $\mathrm{RPC}_{n,r}$, and submitting the resulting blocks to its own oracle for f. When A returns a (supposedly forged) ciphertext, D finds a block in this ciphertext which it did not previously receive from f and submits that block (and an adjacent block, if that too has never been received from f) to its oracle for f^{-1}. (Note that submitting the entire ciphertext is unnecessary, since our analysis in Theorem 2 bounds the probability of forging even one block.) If the *ctr* variables "match up", D outputs 1 (guessing that f was chosen from F); otherwise, D outputs 0.

This requires D to submit $b + 2q$ queries to its oracle for f, and 2 queries to its oracle for f^{-1}. Running time includes time to update the counter and to sort and search through the submissions/responses from oracle f. The theorem follows. □

Theorem 5. *Suppose F is a $(t, q_1, q_2; \epsilon)$-secure super-PRP family. Let Π be an encryption scheme using $RPC_{n,r}$ mode instantiated with a block cipher chosen randomly from F. Then Π is $(O(t - b_e \log b_e), q_e, b_e(n-r), q_d; \ell, \epsilon')$-secure in the sense of IND-P2-C2 (for $b_e + q_e + \ell + 2 \leq 2^r$ and $2q_d \leq q_2$), where:*

$$\epsilon' = \epsilon + \frac{q_d 2^{n-r-1}}{2^n - b_e - 2q_e}.$$

Sketch of Proof The proof is similar to that of Theorem 4. Construct distinguisher D from adversary A. For each ciphertext submitted by A to the decryption oracle, D finds a block in the ciphertext which was not previously received from its oracle for f. D submits that block (and an adjacent block, if that too has never been received from f) to its oracle for f^{-1}. If the *ctr* variables "match up", D outputs 1 (guessing that f was chosen from F); otherwise, D returns $\perp$ to A. D also outputs 1 if A successfully distinguishes the ciphertext even though all its decryption oracle queries were answered by $\perp$. The proof follows. □

4.2 A Probabilistic Mode

The probabilistic mode is a straightforward extension of the stateful mode. We present it here for completeness (*rand* represents an r-bit binary number, and addition is done modulo 2^r):

```
Algorithm $\mathcal{E}$-RPC$\$_{n,r}(M)$
  parse $M$ as $M_1, \ldots, M_\ell$,
    where $|M_i| = n - r$
  $rand \leftarrow \{0,1\}^r$
  $C_0 = F_{sk}(\mathsf{start}, rand)$
  for $i = 1, \ldots, \ell$ do
    $C_i = F_{sk}(M_i, rand + i)$
  $C_{\ell+1} = F_{sk}(end, rand + \ell + 1)$
  return $C = C_0, \ldots, C_{\ell+1}$
```

```
Algorithm $\mathcal{D}$-RPC$\$_{n,r}(C)$
  parse $C$ as $C_0, \ldots, C_{\ell+1}$,
    where $|C_i| = n$
  if $\ell + 1 < 3$ return $\perp$
  for $i = 0, \ldots, \ell + 1$ do
    $(M_i, rand_i) = F_{sk}^{-1}(C_i)$
  if $(M_0 \neq \mathsf{start}) \vee (M_{\ell+1} \neq \mathsf{end})$ return $\perp$
  for $i = 1, \ldots, \ell$ do
    if $rand_i \neq rand_0 + i$ return $\perp$
    if $(M_i = \mathsf{start}) \vee (M_i = \mathsf{end})$ return $\perp$
  if $rand_{\ell+1} \neq rand_0 + \ell + 1$ return $\perp$
  return $M = M_1, \ldots, M_\ell$
```

The following theorems quantify the security of RPC$ mode with respect to existential unforgeability and chosen ciphertext attacks.

Theorem 6. *Let Π be an encryption scheme using RPC$\$_{n,r}$ mode instantiated with a block cipher chosen randomly from P_n. Then Π is $(t, q, b(n-r); \epsilon_{\mathrm{unf}})$-secure in the sense of existential unforgeability, where:*

$$\epsilon_{\mathrm{unf}} = \frac{(b+q)(q-1)}{2^r} + \frac{2^{n-r-1}}{2^n - b - 2q}.$$

Sketch of Proof Let A be an adversary attacking Π in the sense of existential unforgeability. Let $rand^i$ be the nonce associated with query i, for $i = 1, \ldots, q$, and let b^i be the number of plaintext blocks in the i^{th} query. Let $\mathsf{Overlap}$ be the event that $rand^i + k = rand^j + k'$ for some $i \neq j$ and $0 \leq k \leq b^i, 0 \leq k' \leq b^j$. In other words, $\mathsf{Overlap}$ is the event that there is an overlapping sequence in the random padding used.

The success probability of A is given by:

$$\begin{aligned}\Pr\left[\mathsf{Success}\right] &= \Pr\left[\mathsf{Success}|\mathsf{Overlap}\right]\Pr\left[\mathsf{Overlap}\right]\\ &\quad + \Pr\left[\mathsf{Success}|\overline{\mathsf{Overlap}}\right]\Pr\left[\overline{\mathsf{Overlap}}\right]\\ &\le \Pr\left[\mathsf{Overlap}\right] + \Pr\left[\mathsf{Success}|\overline{\mathsf{Overlap}}\right].\end{aligned}$$

Now, in the case of $\overline{\mathsf{Overlap}}$, the advantage of A is the same as in the stateful version of RPC (Theorem 2). Furthermore, we have (following [2]):

$$\Pr\left[\mathsf{Overlap}\right] < \frac{(b+q)(q-1)}{2^r}.$$

This gives the stated result. □

To see that this bound is essentially tight, consider an adversary A making $q = 2$ queries, totaling b blocks, achieving success probability approximately $\frac{2^{n-r-1}}{2^n-b-2q} + \frac{b-2}{2^r}$, which operates as follows: A submits to the encryption oracle the plaintext $0^{(b/2)(n-r)}$ *twice*. If any of the ciphertext blocks received in return from the oracle are equal (except in the case that the two ciphertexts received are entirely equal), A can cut-and-paste the ciphertexts to form a new ciphertext decrypting to a (longer or shorter) valid plaintext consisting of all zeros. The probability of this occurring is precisely $\frac{b-2}{2^r}$. If this does not happen, A guesses a value for a ciphertext block and submits this (as before).

Theorem 7. *Let Π be an encryption scheme using RPC$\$_{n,r}$ mode instantiated with a block cipher chosen randomly from P_n. Then Π is $(t, q_e, b_e(n-r), q_d; k, \epsilon)$-secure in the sense of IND-P2-C2, where:*

$$\epsilon = \frac{(k-1)q_e + b_e}{2^r} + q_d\epsilon_{\mathrm{unf}}.$$

Sketch of Proof We first analyze the success probability of an adversary A when attacking Π in the sense of IND-P2-C0. Let $rand^i$ be the nonce associated with query i, for $i = 1, \ldots, q_e$, and let b^i be the number of plaintext blocks in the i^{th} query. Let $rand^*$ be the nonce associated with the challenge ciphertext. It is clear that the adversary can succeed only when $rand^i + r = rand^* + r'$, with $1 \le r \le b^i, 1 \le r' \le k$ (indeed, an adversary can submit the plaintexts $0^{k(n-r)}$, $1^{k(n-r)}$ and then query the encryption oracle repeatedly at $0^{k'(n-r)}$ for various values of k' to try for a repeated block). The chance of such a collision is maximized if all nonce sequences generated by encryption oracle queries are no less than $k-1$ blocks apart. Then, a collision occurs if $rand^*$ is $k-1$ or fewer blocks before any previous sequence or in a block occupied by some previous sequence. So:

$$\mathsf{Adv}_{A,\Pi}^{\mathrm{IND-P2-C0}} \le \frac{(k-1)q_e + b_e}{2^r}.$$

Application of Theorems 6 and 1 gives the final result. □

Bounds for RPC$ instantiated with a super-PRP block cipher are straightforward extensions of the above (as in Theorems 4 and 5), and are omitted.

4.3 Forthcoming: An Incremental Mode

The present results have led us to develop a chosen-ciphertext-secure incremental encryption mode [4]; details will appear in a forthcoming manuscript [17]. An incremental mode of encryption is one in which updating the encryption of a document (for instance, when a document is edited) is much faster than re-encryption of the entire document. An incremental version of the modes presented above partially offsets their relative inefficiency (depending on the application).

5 Discussion

It is instructive to compare RPC mode to other modes of encryption which might potentially achieve security under chosen ciphertext attacks. Note, however, that previous work in this area has concentrated on extending super-PRPs on n bits to super-PRPs on kn bits, and not on chosen ciphertext security. While a super-PRP on kn bits is desirable, it does not necessarily imply security when the adversary is allowed to submit ciphertexts of varying lengths. We note that RPC is not intended to be a super-PRP; instead, it is meant to give chosen ciphertext security via unforgeability.

One mode of encryption was suggested by Naor and Reingold [20]. They provide a construction which extends a block cipher on n-bits to a super-PRP on $2in$ bits, for any $i \geq 1$. However, it is unclear how to extend their construction to handle variable input lengths. Furthermore, the construction requires shared keys for both the underlying block cipher and two additional hash functions. Finally, since the construction requires two applications of a hash function to strings as long as the plaintext message, the construction is not inherently parallelizable.

A different mode of encryption was suggested by Bleichenbacher and Desai [8]. Their construction gives a super-PRP on messages of arbitrary length, and requires the parties to share keys to three underlying block ciphers. However, it is not clear (and we were unable to prove) that their mode is secure under chosen ciphertext attacks when the adversary is allowed to submit ciphertexts of varying lengths. Also, the scheme is relatively inefficient, as it requires three applications of the underlying block cipher for every block of the plaintext message, and the required computation is not parallelizable.

RPC mode is simple and leads to a straightforward security analysis. The drawback of the mode is the ciphertext expansion and resulting slowdown: for practical security one would want $r \geq 32$ which gives impractical expansion when using a 64-bit block cipher. When using block ciphers with 128-bit or larger block-sizes (e.g., AES), this is less of a concern (only 33% expansion). Advantages of RPC include the fact that the parties need share only one key and that the scheme is completely parallelizable. In contrast, authentication using a MAC requires an additional key and another cryptographic computation per block (which in many cases requires highly sequential computation). Minimizing shared key lengths is important in any integration of public-key and private-key encryption (such as e-mail encryption software). We further note that the ciphertext expansion in the current state of increased communication and storage bandwidth does not seem like a real limitation.

As an open research direction, we note that the mode presented here has the ciphertext a (constant) multiplicative factor longer than the plaintext. Can this be improved to obtain a provably secure, single-key mode in which the ciphertext is longer than the plaintext by only an additive constant?

References

1. ANSI X3.106, “American National Standard for Information Systems—Data Encryption Algorithm—Modes of Operation,” American National Standards Institute, 1983.
2. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A Concrete Security Treatment of Symmetric Encryption: Analysis of the DES Modes of Operation. FOCS 1997.
3. M. Bellare, A. Desai, D. Pointcheval, and P. Rogaway. Relations Among Notions of Security for Public-Key Encryption Schemes. CRYPTO 1998.
4. M. Bellare, O. Goldreich, and S. Goldwasser. Incremental Cryptography and Application to Virus Protection. STOC 1995.
5. M. Bellare and P. Rogaway. On the Construction of Variable-Input-Length Ciphers. FSE 1999.
6. E. Biham. Cryptanalysis of Multiple Modes of Operation. J. of Cryptology 1998.
7. E. Biham and L.K. Knudsen. Cryptanalysis of the ANSI X9.52 CBCM Mode. EUROCRYPT 1998.
8. D. Bleichenbacher and A. Desai. A Construction of a Super-Pseudorandom Cipher. Manuscript, February 1999.
9. D. Dolev, C. Dwork, and M. Naor. Non-malleable Cryptography. SIAM J. Computing, to appear; a preliminary version appears in STOC 1991.
10. S. Goldwasser and S. Micali. Probabilistic Encryption. JCSS, 28: 270-299, 1984.
11. ISO 8372, “Information Processing—Modes of Operation for a 64-bit Block Cipher Algorithm,” International Organization for Standardization, Geneva, Switzerland, 1987.
12. M. Jakobsson, J.P. Stern, and M. Yung. Scramble All, Encrypt Small. FSE 1999.
13. C.J.A. Jansen and D.E. Boekee. Modes of Blockcipher Algorithms and Their Protection Against Active Eavesdropping. EUROCRYPT 1987.
14. C. Kaufman, R. Perlman, and M. Speciner. “Network Security: Private Communication in a Public World,” Prentice Hall, New Jersey, 1995, pp. 89–92.
15. J. Katz and B. Schneier. A Chosen Ciphertext Attack Against Several E-mail Encryption Protocols. 9th USENIX Security Symposium, to appear.
16. J. Katz and M. Yung. Complete Characterization of Security Notions for Probabilistic Private-Key Encryption. STOC 2000.
17. J. Katz and M. Yung. Chosen-Ciphertext Secure Incremental Encryption. Manuscript, February 2000.
18. M. Luby. Chapter 14, “Pseudorandomness and Cryptographic Applications,” Princeton University Press, 1996.
19. C.H. Meyer and S.M. Matyas. “Cryptography: A New Dimension in Computer Data Security,” John Wiley & Sons, New York, 1982.
20. M. Naor and O. Reingold. On the Construction of Pseudorandom Permutations: Luby–Rackoff Revisited. STOC 1997; also: personal communication, December 1999.

21. M. Naor and M. Yung. Public-Key Cryptosystems Provably Secure Against Chosen Ciphertext Attacks. STOC 1990.
22. National Bureau of Standards, NBS FIPS PUB 81, "DES Modes of Operation," U.S. Department of Commerce, 1980.
23. B. Preneel. Cryptographic Primitives for Information Authentication—State of the Art. State of the Art in Applied Cryptography, 1997.
24. B. Preneel, M. Nuttin, V. Rijmen, and J. Buelens. Cryptanalysis of the CFB Mode of the DES with a Reduced Number of Rounds. CRYPTO 1993.
25. C. Rackoff and D. Simon. Non-Interactive Zero-Knowledge Proof of Knowledge and Chosen Ciphertext Attack. CRYPTO 1991.