Modes of Encryption Secure against Blockwise-Adaptive Chosen-Plaintext Attack

Gregory V. Bard*

August 18, 2006

Keywords: Blockwise-Adaptive Attack, Encryption Schemes, Chosen-Plaintext Attack, Modes of Encryption, CBC, OFB, CFB, ABC, CTR, Infinite Garble Extension, HCBC, HPCBC, XCBC.

Abstract

Blockwise-adaptive chosen-plaintext and chosen-ciphertext attack are new models for cryptanalytic adversaries, first discovered by Joux, *et al* [JMV02], and describe a vulnerability in SSH discovered by Bellare, et al [BKN02]. Unlike traditional chosen-plaintext (CPA) or chosen-ciphertext (CCA) adversaries, the blockwise adversary can submit individual blocks for encryption or decryption rather than entire messages. This paper focuses on the search for on-line encryption schemes which are resistant to blockwise-adaptive chosen-plaintext attack. We prove that one oracle query with non-equal inputs is sufficient to win the blockwise-adaptive chosen-plaintext game if the game can be won by any adversary in PPT with non-negligible advantage.

In order to uniformly describe such encryption schemes, we define a canonical representation of encryption schemes based on functions believed to be pseudorandom (i.e. Block Ciphers). This Canonical Form is general enough to cover many modes currently in use, including ECB, CBC, CTR, OFB, CFB, ABC, IGE, XCBC, HCBC and HPCBC. An immediate result of the theorems in this paper is that CTR, OFB, CFB, HCBC and HPCBC are proven secure against blockwise-adaptive CPA, as well as S-ABC under certain conditions. Conversely ECB, CBC, IGE, and P-ABC are proven to be blockwise-adaptive CPA insecure. Since CBC, IGE and P-ABC are chosen-plaintext secure, this indicates that the blockwise-adaptive chosen-plaintext model is a non-trivial extension of the traditional chosen-plaintext attack model.

*gregory.bard@ieee.org Dept. of Applied Mathematics and Scientific Computation, University of Maryland.

1 Introduction

Cryptanalytic attacks have often been divided into the useful categories of known ciphertext, known plaintext, chosen-plaintext (CPA), and chosen-ciphertext (CCA) [DH76, KY00]. These categories represent various possible capabilities of potential attackers, by which the security of a scheme against each possible class of adversary can be expressed. In particular, Fouque, Joux, and Poupard [FJP04] show that security definitions are a pairing of an attack model with a security goal.

This research began in 2002 when Bellare, Kohno, and Namprempre published an operationally feasible attack very similar to CPA on the Secure Shell (SSH) [BKN02]—using Cipher Block Chaining (CBC) which had been proven secure against CPA [GB]. The difference stems from the simple idea that in CPA, the adversary has the capability to insert messages of his/her own creation into a stream of messages for encryption. In the SSH attack, blocks are inserted into existing messages. From this apparent mismatch between operational CPA and its theoretical definition sprang the study of blockwise-adaptive attack.

This paper establishes a generalized form for on-line encryption schemes, called Canonical Form that describes most schemes that make a single call to a pseudorandom permutation per block of encryption. The theorems that follow prove blockwise-adaptive CPA security under certain conditions and insecurity under other conditions. Most usefully, if the blockwise-adaptive chosen-plaintext game can be won by a probabilistic polynomial time (PPT) adversary with non-negligible chance of success, one oracle query with non-equal inputs is sufficient (with polynomially many queries of equal inputs). Also, several common modes of encryption are easily denoted in Canonical Form: Cipher Block Chaining (CBC), Counter Mode (CTR), Output Feedback Mode (OFB), Cipher Feedback Mode (CFB), Electronic Codebook Mode (ECB), Infinite Garble Extension (IGE) as defined by Campbell [Cam78], Accumulated Block Chaining (ABC) as described by Knudsen [Knu00], Hash Cipher Block Chaining (HCBC) and its variant HPCBC as defined by Bellare et al [BBKN01], and Extended Cipher Block Chaining (XCBC) as described by Gligor, et al [GD01]. An immediate corollary of this Canonical Form and the theorems related to it, is that CBC, ECB, IGE, and ABC with public initial conditions[1], are shown to be blockwise-adaptive CPA insecure; likewise, CTR, CFB, OFB, HCBC, HPCBC are shown to be blockwise-adaptive CPA secure, as well as ABC with certain secret conditions and choices. We have yet to determine the status of XCBC.

1.1 Blockwise-Adaptive Attack

Informally, the principal distinction between blockwise CPA and messagewise (traditional) CPA relates to the oracle queries done by the attacker. In messagewise CPA, the attacker can create messages, submit them for encryption, and observe results. In blockwise CPA, the attacker can insert blocks into existing messages, submit them for encryption, and observe results.

The name "blockwise-adaptive" alludes to the capability of the attacker to view the results of inserting one or more blocks of choice into a plaintext message before deciding on the next block, thus permitting the attacker to "adapt" the attack based upon those observations. It should be noted that blockwise-adaptive CPA includes all models of cryptanalysis except chosen-ciphertext attack, in the sense that the BACPA adversary has all the capabilities of the non-CCA adversaries of other models. Therefore it forms a convenient model for applications where the chosen-ciphertext capability is not anticipated.

[1] Public Initial Conditions including any keys for hash functions.

1.2 Previous and Recent Work

Joux, Martinet, and Valette further identified three additional operational attacks of similar form to Bellare's attack on SSH, naming this new class of attacks "blockwise-adaptive CPA" [JMV02]. While very similar to "traditional" CPA, the attacks do not conform to any of the four classes of cryptanalysis as presently defined. Also introduced in that paper is "blockwise-adaptive CCA", a variation on "traditional" CCA.

Fouque, Joux, Martinet, and Valette next provided the first security definitions for blockwise-adaptive CPA and CCA [FJMV03]. The paper was geared to smart cards in authentication devices, and so is a somewhat different context than on-line schemes. They propose a generic model, Decrypt-then-Mask, proved to be blockwise-adaptive CCA secure under certain conditions, and an example of it, CBC-MAC. However, none of these schemes are on-line and so are not included here. (See Section 1.3 for reasons).

Boldyreva and Taesombut, focusing on on-line schemes in general, showed that neither HCBC nor any on-line scheme in general is CCA secure under the general definitions of Fouque, et al. [BT04, JMV02, FJMV03]. Instead, they define a slightly different security definition, IND-BLK-CCA. They prove that the mode HPCBC, which is a variation on HCBC, is IND-BLK-CCA secure. Both HPCBC and HCBC are analyzed (and proven blockwise-adaptive CPA secure) in this paper, which confirms their results.

At almost the same time, Fouque, Martinet, and Poupard [FMP03] expanded the idea of Delayed-CBC presented in their previous paper to include a general notion of a delayed cipher. Moreover, these authors introduce an adversary with *concurrent* access to multiple blockwise oracles. All previous papers had limited the adversary to sequential access to the oracle, as does this paper. The authors proved that both delaying modes and CFB are secure under their model. However, while delaying modes allow for on-line encryption, they do not allow for on-line decryption[2], and so are not discussed here. The security of the mode CFB is also discussed in Boldyreva and Taesombut [BT04].

More recently, the author demonstrated the potential feasibility of blockwise-adaptive attack by demonstrating an exploitable flaw in the Secure Sockets Layer (SSL) [Bar04, Bar06]. The vulnerability is similar to that found by Bellare in SSH. Particular details of SSL and the method by which proxies, anonymizers, or VPNs interact with a web-browser make the attack feasible.

Recently, at SAC 2004, Fouque, Joux, and Poupard [FJP04] expanded the definitions of security in the messagewise case into the blockwise case. That paper examines the relationships between indistinguishability in the Left-Or-Right, Real-Or-Random and Find-Then-Guess senses, in the attack models of messagewise chosen-plaintext attack, and blockwise-adaptive chosen-plaintext attack with either sequential or concurrent adversaries. An analysis of these security goals for the messagewise case and their relationships is found in Katz and Yung [KY00]. The notion used for security in this paper is LORS-BCPA as defined by Fouque, Joux and Poupard, meaning that the adversary is given blockwise capability in every stage of the attack, but for sequential messages (i.e. not for concurrent messages).

1.3 On-line Schemes

Certain forms of communication are interactive by nature and would cease to be useful if queries and responses could not be exchanged in a synchronized and timely way. There are cryptologic

[2]Boldyreva and Taesombut show that there is a distinction between schemes which can have on-line encryption, and those in which both encryption and decryption are on-line. The usual meaning in the literature is for on-line encryption only, but here, as in Boldyreva and Taesombut, we require both decryption and encryption to be on-line. Note that this is required for fully interactive communications [BT04]. See also Section 1.3.

consequences of this property, denoted "on-line", where a given ciphertext block C_i does not depend on any plaintext block P_j if $j > i$.

Bellare, Boldyreva, Knudsen, and Namprempre first identified the notion of on-line ciphers [BBKN01]. There are several reasons why on-line schemes are attractive. First, they make interactive communication possible. Second, they do not require buffering. Third, in real-time scenarios delay is undesirable, and thus transmitting C_i upon receiving P_i is necessary.

Authenticated Encryption often requires a tag, hash, checksum or other digest of the plaintext or ciphertext. This cannot be verified before it is transmitted, and it cannot be transmitted before it is calculated. Any useful function of the plaintext cannot be calculated before the plaintext is known, i.e. submitted. Therefore it cannot possibly be the case that such schemes are "decryptable on the fly". This distinction was first suggested by Boldyreva and Taesombut in [BT04]. Therefore, schemes like OCB mode suggested by Rogaway *et al* [RBBK01] are out of the scope of this paper, as they do not permit on-line decryption and therefore are not useful for interactive communication.

Delayed-CBC, as suggested by Joux, *et al* [JMV02, FJMV03] calculates the ciphertexts as in normal CBC, but delays their output by one block. Thus $C_1 = F_{sk}(C_0 \oplus P_1)$ but is not transmitted until after P_2 is received. However, if P_2 will be a response to the other party's response to P_1, then deadlock occurs. This scheme is not meant for "decryption-on-the-fly" and thus is not on-line in the sense of Boldyreva and Taesombut [BT04].

1.4 Current Research on Modes of Encryption

Loosely speaking, a "mode of operation" for block ciphers is an algorithm that features the use of a symmetric-key block cipher algorithm to provide an information service, such as confidentiality or authentication [Dwo01]. An encryption scheme is a mode where the objective is confidentiality, but other objectives might be pseudorandom number generation or authenticated encryption. We use the definition of an "encryption scheme" below, as defined by Fouque, Joux and Poupard [FJP04], which is a precise version of the notion of a mode of operation.

Research into modes of encryption for block ciphers is a historically rich field, with discussions beginning with the definition of the Data Encryption Standard (DES), and two original modes of operation, namely the Electronic Code Book (ECB) and Cipher Block Chaining (CBC). More recently, NIST held two public workshops to study modes of block encryption, in Fall 2000 and Summer 2001. The result of these workshops was a two-part recommendation [Dwo01, Dwo02], the first of which specifies encryption modes. These are Electronic Codebook (ECB), Cipher Block Chaining (CBC), Counter Mode (CTR), Output Feedback Mode (OFB), and Cipher Feedback Mode (CFB).

Meanwhile many other modes of encryption have been proposed. Specifically, nineteen are listed on the NIST website, submitted by many authors. Some modes have special advantages, for example CTR mode can be easily executed in parallel [LRW00]. To analyze all nineteen of them would be beyond the scope of this paper, but the methods outlined here should be easy to apply to those cases.

Also the recently proposed modes XCBC and XECB, designed by Gligor, et al, to provide authenticity and integrity as well as privacy (to prevent encrypted message forgeries) are encryption schemes, but only XCBC is on-line [GD01].

Knudsen first proposed the general class of modes called ABC, or Accumulated Block Ciphers, which are defined in terms of a function denoted h. The mode ABC is an extension of Infinite Garble Extension (IGE), as defined by Campbell [Cam78] and analyzed by Gligor, et al [GD00]. In fact, if h is the function which always returns zero, then ABC acts identically to IGE. Bellare *et al* [BBKN01] distinguish ABC into two subclasses—P-ABC when the initial conditions are publicly

known, and S-ABC when some initial conditions are secret, including any keys to the function h. These notions were extended by Bellare *et al* to include HCBC and HPCBC, which are modes that utilize a hash function during the encryption process [BBKN01].

2 Encryption Schemes and Canonical Form

Fouque, Joux, and Poupard define an encryption scheme as a triple of algorithms [FJP04]. First, a key generation algorithm, which does not concern us. Next the plaintext is divided into blocks, $\vec{P} = P_1, P_2, \ldots P_n$, and some algorithm $E(\vec{P}) \rightarrow \vec{C}$ produces a ciphertext, again divided into blocks $\vec{C} = C_0, C_1, C_2, \ldots C_n$. A decryption function also exists, $D(\vec{C}) = \vec{P}$, to map ciphertexts back to their corresponding plaintexts.

All encryption schemes discussed in this paper are on-line and correct.[3] Since for all on-line schemes, C_i does not depend upon P_j for $j > i$, the sender can compute C_i upon receipt of P_i. Thus one can model the encryption scheme as a finite-state-machine, with a randomized initialization algorithm.

First, an algorithm $Init_{sk}(k)$, whose sole input is the security parameter, will output an initial state s_1, and an initialization vector C_0. Second, an encryption algorithm $E_{sk}(P_i, s_i)$ will take as inputs the secret key, the plaintext, and the current state. It will output the ciphertext block C_i, and the next state[4], s_{i+1}. This model is based on Fouque, Joux, and Poupard, [FJP04] and is similar to that used in Bellare and Boldyreva's several papers [BKN02, BBKN01, BDJR97, BT04]. In particular, many encryption schemes are based on a function family believed to be pseudorandom (i.e. a block cipher), and make one call per single block of encryption. To model these encryption schemes, we define Canonical Form.

2.1 Canonical Form

An encryption scheme in Canonical Form is a quadruple of algorithms, (Init, Pre, Post, and Update). The general concept is that the function $E_{sk}(P_i, s_i)$ uses some pseudorandom function family member $F_{sk}()$ to calculate the ciphertext and next state. Therefore consider the following: Some function of the inputs of E_{sk} is the input to F_{sk}. Next, some function of the output of F_{sk} and the inputs of E_{sk} becomes the ciphertext block. Finally, some function of the output of F_{sk} and the inputs of E_{sk} becomes the next state. This gives rise to the following quadruple.

- $Init_{sk}(k) \rightarrow (C_0, s_1)$
- $Pre(P_i, s_i) \rightarrow x$
- $Post(P_i, s_i, y) \rightarrow C_i$ where $y = F_{sk}(x) = F_{sk}(Pre(P_i, s_i))$.
- $Update(P_i, s_i, y) \rightarrow s_{i+1}$.

[3] An encryption scheme is correct if $D(E(\vec{P})) = \vec{P}$ for all messages $\vec{P}$.

[4] In the special case of stateless encryption (the completely insecure Electronic Codebook mode), the state can be thought of as the empty string. In all other cases, the state is some binary string, the format and significance of which depends on the chosen scheme. However, typical choices for the state are the previous ciphertext block, the previous plaintext block, or a counter.

2.2 Examples

Here we describe CBC, CTR, and OFB as three examples. The appendices will add ECB, CFB, ABC, IGE, HCBC, HPCBC and XCBC. All ten of these encryption schemes can be written in Canonical Form.

For CBC, one normally writes:

$$r_1 \xleftarrow{R} \{0,1\}^k; \quad C_0 = r_1$$
$$C_i = F_{sk}(C_{i-1} \oplus P_i)$$

However, in Canonical Form:

$$Init_{sk}(k) : r \xleftarrow{R} \{0,1\}^k; \quad C_0 = s_1 = r$$
$$Pre(P_i, s_i) = P_i \oplus s_i; \quad Post(P_i, s_i, y) = y; \quad Update(P_i, s_i, y) = y$$

Note: above the state s_i represents the previous ciphertext. For CTR, the standard notation is

$$Init_{sk}(k) : i_0 \xleftarrow{R} \{0,1\}^k; \quad C_0 = F_{sk}(i_0)$$
$$C_i = F_{sk}(i + i_0) \oplus P_i$$

However, in Canonical Form:

$$Init_{sk}(k) : r \xleftarrow{R} \{0,1\}^k; \quad C_0 = F_{sk}(r); \quad s_1 = r + 1$$
$$Pre(P_i, s_i) = s_i; \quad Post(P_i, s_i, y) = y \oplus P_i; \quad Update(P_i, s_i, y) = s_i + 1$$

Note above the state s_i represents a counter. For OFB, the classic form would be

$$x_0 \xleftarrow{R} \{0,1\}^k; \quad C_0 = x_0$$
$$x_{i+1} = F_{sk}(x_i); \quad C_i = x_i \oplus P_i$$

However, in Canonical Form:

$$Init_{sk}(k) : r \xleftarrow{R} \{0,1\}^k; \quad C_0 = s_1 = r$$
$$Pre(P_i, s_i) = s_i; \quad Post(P_i, s_i, y) = y \oplus P_i; \quad Update(P_i, s_i, y) = y$$

Note above the state s_i represents the "recycled" output x_i of the pseudorandom function, which is fed back into F_{sk} during the next block.

3 Definitions

3.1 Three Security Objectives

The following two attack categories are used in this paper to demonstrate possible blockwise-adaptive attack. One is the intuitive definition first given by Fouque, Joux and Poupard [FJMV03, FJP04], but modified slightly. The other exists to facilitate proofs; these will be shown to be equivalent. But first, the normal notion of a chosen-plaintext Left-or-Right Indistinguishability oracle must be extended to the blockwise case.

Blockwise-Adaptive Chosen Plaintext Oracles

The blockwise-adaptive oracle $LRBW_{(b,sk)}$ for an encryption scheme E_{sk} is a stateful function which takes as inputs either a pair of plaintext blocks or the special symbol **start**. Formally, the domain of $LRBW_{(b,sk)}$ is $(\{0,1\}^\ell \times \{0,1\}^\ell) \cup \mathbf{start}$ where ℓ is the plaintext block-length of E_{sk}. The range of $LRBW_{(b,sk)}$ is the set of binary strings of length equal to the ciphertext block-length of E_{sk}. The function is defined by:

$$\begin{aligned} LRBW_{(b,sk)}((M_0, M_1)) &= E_{sk}(M_b, s_t) \\ LRBW_{(b,sk)}(\textbf{start}) &= Init_{sk}(k) \end{aligned}$$

Note: The state s_t is set by $Init$ and updated by E_{sk} but never given as an output of $LRBW_{(b,sk)}$.

General Blockwise Chosen-Plaintext Security

Fouque *et al* defined the following notion called LORS-BCPA, which is based on the notion IND-LOR-CPA, or Indistinguishability in chosen-plaintext attack [FJMV03, KY00]. Essentially the adversary is given a left-or-right oracle, to build plaintext messages one block at a time, while viewing the ciphertext along the way. After polynomially many oracle uses, the adversary attempts to guess whether the oracle was left or right. For a system to be secure, for PPT adversaries this should be possible only with negligible probability (compared to the security parameter of the scheme). The adversary is allowed polynomially many messages, by starting a new message via submitting $LRBW_{(b,sk)}(\textbf{start})$.

Definition 1 *Suppose Algorithm $A(1^k)$, which returns a single bit, is given access to the blockwise chosen-plaintext oracle $LRBW_{(b,sk)}$ of an encryption scheme E_{sk}. The* **General Blockwise Chosen-Plaintext Advantage** *of A on E is:*

$$\left|\Pr[sk \leftarrow \{0,1\}^k, A^{LRBW_{(b,sk)}}(1^k) = 1 | b = 0] - \Pr[sk \leftarrow \{0,1\}^k, A^{LRBW_{(b,sk)}}(1^k) = 1 | b = 1]\right|$$

The encryption scheme E is **Generally Blockwise Chosen-Plaintext Secure** *if and only if for all PPT algorithms $A(1^k)$, the general blockwise chosen-plaintext advantage of $A(1^k)$ against E is negligible compared to k.*

Note: Our model does not include the option for any adversary to encrypt multiple messages simultaneously and adaptively under the same key. At most one message at a time is in the process of being encrypted, block by block. That corresponds well to the operational circumstances of how block ciphers are often used. However, no operational scenario has been suggested for the circumstances where multiple submissions would actually occur in an interleaved and simultaneous fashion, creating concurrent access to several oracles which are all operating under the same key. This interleaving is permitted in the model given by Fouque *et al* [FMP03]. On the other hand, Fouque *et al* show that schemes secure against sequential BCPA adversaries are not necessarily secure against concurrent BCPA adversaries [FJP04].

Primitive Blockwise Chosen-Plaintext Security

Definition 2 *The* **Primitive Blockwise-Adaptive Chosen-Plaintext Game** *is identical in all respects to the general blockwise-adaptive chosen-plaintext game, except that the adversary is only permitted at most one query $LRBW_{(b,sk)}((P, Q))$ where $P \neq Q$. All other queries must have $P = Q$.*

The **Primitive Blockwise-Adaptive Chosen-Plaintext Advantage** *is the same as the general blockwise-adaptive chosen-plaintext advantage, provided the adversary is permitted only one $P \neq Q$ query.*

The encryption scheme E is **Primitively Blockwise-Adaptive Chosen-Plaintext Secure** *if and only if for all* PPT *algorithms $A(1^k)$, the primitive blockwise-adaptive chosen-plaintext advantage of $A(1^k)$ against E is negligible in k.*

While much more restrictive than the general case, it turns out that that security in the primitive model is identical to security in the general model. The general model is more intuitive, but the primitive model will make proofs easier.

The Collision Game

Informally, the objective of this game is to create a collision in the outputs of Pre. Since the output of Pre serves as the sole input to F_{sk}, a collision on the output of Pre is also a collision on the output of F_{sk}, and thus provides a potential opportunity for attack. By collision, it is meant that that the output of Pre is equal for two distinct blocks, and also that the block-numbers of that pair are known. With this objective in mind, the definition is rather straight-forward.

The adversary $A^{E_{sk}(\cdot)}(1^k)$ is given access to a blockwise-encryption oracle $E_{sk}(\cdot)$. The adversary is not given the state s_i. This oracle is meant to represent a realistic blockwise attacker who can submit blocks for encryption and view their ciphertext, but who does not have access to the internal state of the cipher (though perhaps he/she may calculate it by other means). The adversary can submit **start** queries to begin new messages.

Formally, the game proceeds as follows. A call is made to $E_{sk}(\mathbf{start})$ and the adversary is given C_0. Next, the adversary submits P_1, and receives C_1. This continues, submitting P_i and receiving C_i for polynomially many blocks. Some of the P_i's might equal **start** resulting in several, but still polynomially many, messages. Then the adversary must output a block-number t, and a potential plaintext $P*$. The adversary is successful if the collision would actually occur, namely $Pre(P*, s_i) = Pre(P_t, s_t)$.

A scheme E is said to be collision-resistant if all PPT adversaries have negligible probability of success in the collision game against E.

3.2 Three Security Properties

Definition 3 *An encryption scheme is* **collision-resistant** *if and only if all* PPT *adversaries have negligible advantage in the collision game, compared to the security parameter of the pseudorandom function being used.*

We will prove shortly that if the following two security properties hold, then general blockwise-adaptive chosen-plaintext security follows if and only if an encryption scheme is collision-resistant. This is intuitive, because the pseudorandom function is the "work horse" of the encryption scheme, and if its input can be duplicated, then its output will be duplicated. If this condition can be detected, then an attack against the scheme can be built. (This is essentially how the original blockwise-adaptive attack operates against CBC, as discovered by Bellare [BKN02]).

Definition 4 *The function $Post$ in $\{Pre, Post, Update\}$ is* **entropy-preserving** *if, when the plaintext and state are held constant, the function from y to the ciphertext is 1-to-1 (injective).*

The reason we call this property entropy-preserving is that for these functions, if y is a random variable with entropy h, and the plaintext and state are held constant, then the ciphertext has entropy h also. (Injective functions preserve entropy). In the proof of one of the main theorems of the paper, we will exploit this property by noting that a uniform random variable has maximal

entropy over its domain, and so if the input y is uniformly random, and the plaintext and state are fixed, then the ciphertext is also uniformly random (though possibly over a different set).

Most encryption schemes in use have entropy-preserving *Post* functions. One should note that XOR with a constant is an injective map, and this is the most common function choice for *Post* along with the identity map (which is also injective). Of the ten encryption schemes analyzed in this paper, all are entropy-preserving.

A related concept is collision-verifiability.

Definition 5 *An encryption scheme* $\{Pre, Post, Update\}$ *is* **collision-verifiable** *if there exists an algorithm, given a plaintext message* $\vec{P}$*, (of length* n *blocks) a ciphertext message* $\vec{C}$*, and a block number* i*, which will with all-but-negligible probability output one when* $\vec{C}$ *decrypts to* $\vec{P}$ *and simultaneously the output of* F_{sk} *at blocks* i *and* n *are equal, and zero at all other times.*

All the schemes analyzed in this paper but XCBC are collision-verifiable.

4 Main Results

The main result of this paper is the following statement: If $\{Pre, Post, Update\}$ is an entropy-preserving and collision-verifiable encryption scheme, then it is generally blockwise-adaptive chosen-plaintext secure if and only if it is collision-resistant. This will be proven in three steps, each of which is a theorem below.

Theorem 1 *The encryption scheme* $\{Pre, Post, Update\}$ *is primitively blockwise-adaptive chosen-plaintext secure, if and only if it is generally blockwise-adaptive chosen-plaintext secure.*

Proof See the appendix. ■

Theorem 2 *If* $\{Pre, Post, Update\}$ *is a collision-resistant entropy-preserving encryption scheme, then it is primitively blockwise-adaptive chosen-plaintext secure.*

Proof Assume $\{Pre, Post, Update\}$ is an entropy-preserving encryption scheme that is collision resistant. Assume there exists a PPT Algorithm A, which can win the primitive blockwise-adaptive chosen-plaintext game, with non-negligible advantage δ_A. Then we will construct a PPT algorithm *Dist* which will win the pseudorandom game against F_{sk} with non-negligible advantage. Since F_{sk} is assumed to be pseudorandom, this is a contradiction. Thus Algorithm A does not exist. Therefore $\{Pre, Post, Update\}$ is primitive BW-CPA secure (and by the first theorem, is also generally BW-CPA secure).

Algorithm *Dist*, attempting the pseudorandom game against F_{sk}, is given an oracle F', which is either a random function (case 0), or F_{sk} for a randomly chosen value of sk (case 1). Algorithm *Dist* will use F' with $\{Pre, Post, Update\}$ acting as an encryption scheme. It will challenge Algorithm A, and perfectly simulate the primitive blockwise-adaptive chosen-plaintext game.

If Algorithm A wins the game, Algorithm *Dist* will guess that $F' = F_{sk}$ for some sk in the key-space, and output 1. If Algorithm A loses the game, Algorithm *Dist* will guess that F' is a random function, and output 0. Let us now analyze both these cases individually.

The Random Case Consider the outputs of Pre over the course of the game. We claim that the probability of any pair of them being equal is negligible. The reason for this is not obvious. Let the probability of at least one pair of outputs of Pre being equal be P_e. (Thus with probability $1 - P_e$ all the outputs of Pre are distinct).

Let n be the median [5] value of the block number of the latter block of the first matching pair, when at least one pair of outputs is equal. At least half the time, both blocks of the first pair will be before or at block n. Thus with probability at least $\frac{P_e}{2}$ there is at least one pair of blocks before block n+1, which have equal outputs from Pre.

Algorithm $Coll$ will generate a random number r, taken uniformly from the set $\{2, \ldots, n\}$. It will attempt the Collision Game by running a perfect simulation of the primitive BW-CPA game on behalf of Algorithm A. However, it will halt Algorithm A immediately after it submits the query request for block r. With probability $\frac{1}{(n-1)}\frac{P_e}{2}$, the output of Pre during the encryption of block r will collide the output of Pre for a previous block.

Algorithm $Coll$ doesn't know which block is the mate for block r, and so will guess a random value s uniformly from the set $\{1, \ldots, r-1\}$. However, this will be correct with probability at worse $\frac{1}{r-1}$. In the case when block r is indeed going to cause a collision, the probability that s will be the mate for r is given by:

$$\frac{1}{n-1}\sum_{r=2}^{r=n}\frac{1}{r-1} = \left(\frac{1}{n-1}\right)\left(1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\cdots+\frac{1}{n}\right) \approx \frac{\ln(n)+\gamma}{n-1}$$

Where γ signifies the Euler-Mascheroni Constant[6]. The Algorithm $Coll$ thus produces a correct collision at r while guessing the mate s, (and therefore winning the collision game) with probability approximately

$$\frac{P_e(\ln(n)+\gamma)}{2(n-1)^2}$$

However, we can simplify this by noting that $\frac{\ln(n)+\gamma}{2(n-1)^2} > \frac{1}{2n^2}$ and so Algorithm $Coll$ will succeed with probability greater than $\frac{P_e}{2n^2}$. Since n must be at most polynomial to k, we know that this success probability will be non-negligible compared to k if P_e is non-negligible. However, by assumption, the system is collision-resistant, and so therefore P_e is negligible.

Thus the outputs of Pre over all blocks of the game are distinct with all but negligible probability. Furthermore, a series of distinct inputs to a random function creates a sequence of independent, identically and uniformly distributed random outputs.

Recall that the inputs to $Post$ are the plaintext to be encrypted, the state, and y (which is the output of the pseudorandom function). Further, we required that $Post$ be one-to-one (an injection) when the plaintext and state are fixed. Due to this condition, when the y's are a sequences of uniform independent and identically distributed variables, the outputs of $Post$ (the ciphertexts) are also uniform independent and identically distributed variables, or white noise. These cannot convey any information about the plaintext, and the best that the Algorithm A can output is a fair coin.

Thus Algorithm A will output a 1 or 0 with probability $\frac{1}{2}+\varepsilon$, where $|\varepsilon|$ is negligible. This extra negligible advantage comes from the fact that the accidental collisions described in detail above occur with negligible but non-zero probability.

The Pseudorandom Case Here Algorithm $Dist$ is providing Algorithm A with a perfect simulation of the blockwise-adaptive chosen-plaintext game. Therefore Algorithm A will be correct with non-negligible advantage δ_A by assumption. Thus Algorithm $Dist$ will be correct (output 1) with probability $\frac{1}{2}+\delta_A$.

[5] Since the random variable (the block number) is a positive integer, call it z, there exist many m such that $Pr(z \in [1, m]) \geq 1/2$. Let the median be the least of all such m. Since any collection of positive integers has a lower bound, such a median exists.

[6] Recall that $\Sigma_{i=1}^{i=n}\frac{1}{i} \approx ln(n)+\gamma$ for very large n

The Advantage of Algorithm *Dist* In the Pseudorandom Case an output of 1 occurs with probability $\frac{1}{2} + \delta_A$, and in the Random Case with probability $\frac{1}{2} + \varepsilon$. Thus the advantage of the Algorithm *Dist* is $\delta_A - \varepsilon$, which is non-negligible. This is a contradiction, and so we know that Algorithm A must not exist. Therefore the encryption scheme is primitive blockwise-adaptive chosen-plaintext secure. ■

Theorem 3 *If $\{Pre, Post, Update\}$ is a collision-verifiable encryption scheme, and is primitively blockwise-adaptive chosen-plaintext secure, then it is collision-resistant.*

Proof Suppose $\{Pre, Post, Update\}$ is collision-verifiable and primitively blockwise-adaptive chosen-plaintext secure. Suppose further that it is not collision-resistant. This means there exists a PPT Algorithm *Coll* which can win the collision game with non-negligible probability δ_C. We will use that algorithm to construct an Algorithm A which wins the primitive blockwise-adaptive chosen-plaintext game with non-negligible advantage. This contradicts the security of $\{Pre, Post, Update\}$ and therefore Algorithm *Coll* does not exist, and the scheme is collision-resistant.

Algorithm A begins by calling Algorithm *Coll*. Algorithm A will pass Algorithm *Coll*'s oracle queries $E_{sk}(P_i)$ to the left-or-right encryption oracle as $LRBW_{(b,sk)}(P_i, P_i)$, and return the ciphertexts C_i.

When Algorithm *Coll* terminates, it outputs $P*$ and a block number t. Furthermore, if Algorithm *Coll* wins the collision game, we know $Pre(P*, s_n) = Pre(P_t, s_t)$. Let the probability that it wins the collision game be δ_C.

The split query $LRBW_{(b,sk)}(Q, P*)$ is now submitted to the blockwise oracle, where $Q \neq P*$ is a random string of appropriate length. The output C_n is received. The collision verification algorithm, which we will denote Algorithm Ver, will now be called. Its inputs are the plaintext message $P_1 \dots P_{n-1}P*$ as well as $C_0 \dots C_n$ and the block number k. Let its success probability[7] be $1/2 + \delta_V$. Algorithm A will output the bit that Ver outputs.

If Algorithm *Coll* has succeeded, and the bit is one, $P*$ was encrypted and so then $Pre(s_t, P_t) = Pre(s_n, P*)$, and therefore also $y_t = y_n$. With probability $1/2 + \delta_V$, Algorithm Ver will output one, and Algorithm A will be correct. If the bit is zero, then the plaintext block Q and not $P*$ will be encrypted, and so $C_0 \dots C_n$ will not decrypt to $P_1 \dots P_{n-1}P*$ (since $C_0 \dots C_n$ decrypts to $P_1 \dots P_{n-1}Q$). Thus, Algorithm Ver will output zero with probability $1/2 + \delta_V$, and Algorithm A will be correct.

If Algorithm *Coll* has failed, then with probability $1/2 + \delta_V$, the verification algorithm will output zero. If the hidden bit is actually zero, then this is correct. If the bit is one, then Algorithm A will be wrong. So in this case, Algorithm A is correct with probability one-half.

This is all summarized in Table 1.

$$\begin{aligned} Adv_A &= |Pr[A = 1|b = 0] - Pr[A = 1|b = 1]| \\ &= |[(1/2)(\delta_C)(1/2 - \delta_V) + (1/2)(1 - \delta_C)(1/2 - \delta_V)] \\ &\quad - [(1/2)(\delta_C)(1/2 + \delta_V) + (1/2)(1 - \delta_C)(1/2 - \delta_V)]| \\ &= |\delta_C \delta_V| \end{aligned}$$

Since the system is collision-verifiable δ_V is non-negligible, and δ_C is non-negligible by assumption. Thus the advantage of Algorithm A is then obviously non-negligible, which contradicts our assumption that Algorithm A is primitive blockwise-adaptive chosen-plaintext secure. Thus our assumption that the scheme is not collision-resistant must be false. ■

[7]Note, the success probabilities of Algorithm *Coll* and Algorithm Ver are independent.

Bit b	Alg Coll	Alg Ver	$y_n = y_t$?	Output	Probability
0	Succ	Succ	No	0	$(1/2)(\delta_C)(1/2 + \delta_V)$
0	Succ	Fail	No	1	$(1/2)(\delta_C)(1/2 - \delta_V)$
0	Fail	Succ	No	0	$(1/2)(1 - \delta_C)(1/2 + \delta_V)$
0	Fail	Fail	No	1	$(1/2)(1 - \delta_C)(1/2 - \delta_V)$
1	Succ	Succ	Yes	1	$(1/2)(\delta_C)(1/2 + \delta_V)$
1	Succ	Fail	Yes	0	$(1/2)(\delta_C)(1/2 - \delta_V)$
1	Fail	Succ	No	0	$(1/2)(1 - \delta_C)(1/2 + \delta_V)$
1	Fail	Fail	No	1	$(1/2)(1 - \delta_C)(1/2 - \delta_V)$

Table 1: Summary of Probabilities in Theorem 4

Corollary 1 *If $\{Pre, Post, Update\}$ is an entropy-preserving, collision-verifiable encryption scheme, then it is generally blockwise-adaptive chosen-plaintext secure if and only if it is collision-resistant.*

Proof Suppose the scheme is collision-resistant and entropy-preserving. By the second theorem it is primitively BW-CPA secure, and thus by the first theorem, generally blockwise-adaptive chosen-plaintext secure. Alternatively, suppose the scheme is generally blockwise-adaptive chosen-plaintext secure. Then it is primitively secure, by the first theorem, and since it is collision-verifiable, then by the third theorem it is collision-resistant. ■

5 Applications

In this section we will use the previous theorems to determine the blockwise-adaptive chosen-plaintext security of CBC, CTR and OFB. (Note that CTR has been proven blockwise-adaptive CPA secure by Fouque, Joux and Poupard [FJP04] and the insecurity of CBC has been known since Bellare, Kohno and Namprempre, discovered the SSH vulnerability [BKN02]).

Entropy Preservation

The function *Post* for CBC is the identity function, and so obviously is injective. In the case of CTR and OFB, it is an XOR with the plaintext, and so when the plaintext is held constant, *Post* is an injection into the cipher-space. Thus all three schemes are entropy-preserving.

Collision-Verifiable

For CTR and OFB, $C_i = y_i \oplus P_i$, or $y_i = C_i \oplus P_i$. Consider the expression $C_n \oplus C_k \oplus P_n \oplus P_k$. This will equal $y_k \oplus y_n$. Four possibilities must be considered. Either the plaintext is correct or not, and either the collision has occurred or not. Thus if a collision has occurred, and the plaintext blocks are actually correct, the expression will be zero. If the wrong plaintext is given for one of the blocks, and a collision occurs, the expression will be non-zero. Also, if a collision has not occurred, and the plaintext blocks are correct, $y_i \neq y_k$ and so the expression cannot be zero. Finally, if a collision has not occurred and one of the plaintext blocks is wrong, the possibility that the expression comes out to be zero anyway is obviously negligible. (In particular, the proof of the previous theorems has $Q \neq P*$ be generated randomly, so the probability the expression equals any string in particular is $2^{-\ell}$, where ℓ is the plaintext block-length of the scheme. In summary, the collision verification algorithm should output 1 if $C_n \oplus C_k \oplus P_n \oplus P_k$ is all zeroes, and 0 otherwise.

Collision-Resistance

For CBC, suppose the first query is $LRBW_{b,sk}((P_1, P_1))$. Then the ciphertext $C_1 = F_{sk}(C_0 \oplus P_1)$. In this case, let $P* = C_0 \oplus P_1 \oplus C_1$. Recalling that $s_i = C_{i-1}$, one can calculate that

$$Pre(P*, s_2) = s_2 \oplus P* = C_1 \oplus (C_0 \oplus P_1 \oplus C_1) = C_0 \oplus P_1 = Pre(P_1, s_1)$$

And so a collision can be easily caused, with $P* = C_0 \oplus P_1 \oplus C_1$, and $t = 1$, and therefore CBC is not collision-resistant. For CTR and OFB, the attacker has no control over the inputs of the pseudorandom function. For the case of CTR, the inputs are a counter, and so will only collide if a wrap-around occurs. However, the adversary is limited to a polynomial number of blocks, and so this is not possible. Thus CTR is collision-resistant.

In OFB, the output of the pseudorandom function in one step is the input in the next step. Thus, the output of the pseudorandom function in step j is $\underbrace{F_{sk}(F_{sk}(F_{sk}(\cdots F_{sk}}_{j \text{ times}}(x_0)\cdots)))$. Noting that the specification of OFB requires the block cipher to be a permutation [Dwo01], then one can calculate the distribution of orbits generated by repeated evaluation of F_{sk}.

While we omit this calculation, the final distribution of orbit lengths is uniform, and so the probability of a cycle of length $\leq z$ is $z2^{-\ell}$ or negligible. Therefore PPT adversaries will not be able to cause a collision with non-negligible probability.

Security Since CBC is entropy-preserving and not collision-resistant, it is blockwise-adaptive chosen-plaintext insecure. Since OFB and CTR are both collision-verifiable and collision-resistant, they are blockwise-adaptive chosen-plaintext secure.

6 Conclusions

The models of blockwise-adaptive chosen-plaintext attack presented here are those of Joux, et al [JMV02, FJMV03, FJP04]. Further, a framework is defined which can express encryption schemes in a canonical form. Once in Canonical Form, a scheme can usually be proven to be secure or not secure against blockwise-adaptive chosen-plaintext attack, as was the case for nine out of ten of the schemes given in this paper. Furthermore, the equivalence of primitive and general blockwise-adaptive chosen-plaintext security will surely provide system designers with a more straightforward and simple model for proving the security of their schemes against this new form of attack.

It is also interesting to note that blockwise-adaptive chosen-plaintext security strictly implies security in the known ciphertext, known plaintext and messagewise chosen-plaintext senses—in short, for sequential adversaries, all but CCA. The schemes proven secure, namely Counter Mode (CTR), Cipher Feedback Mode (CFB), Output Feedback Mode (OFB), and Hash Cipher Block Chaining (HCBC), with its variant HPCBC, and Accumulated Block Chaining with secret initial conditions (S-ABC under certain choices of the function h) are therefore good choices for encryption schemes when the capabilities of a CCA attacker, or concurrent adversaries, are not anticipated.

Acknowledgements

The problem studied in this paper was recommended to the author by Prof. Jonathan Katz, who has commented on several versions of the paper. In addition, Patrick Studdard and Prof. Lawrence Washington have also made extensive and useful comments. This work was partially completed while supported by a research assistantship under Prof. Katz and completed while a "visiting scientist" funded ECRYPT, the European Union's "Center of Excellence" in cryptography.

References

[Bar04] G. Bard. "Vulnerability of SSL to Chosen-Plaintext Attack." *IACR E-print, Report 2004/108.*

[Bar06] G. Bard. "A Challenging but Feasible Blockwise-Adaptive Chosen-Plaintext Attack on SSL." *Proc. of IEEE Intl. Conf. on Security and Cryptography (ICETE/SECRYPT'06).* 2006. Also available as *IACR E-print, Report 2006/136.*

[BKN02] M. Bellare, T. Kohno, and C. Namprempre. "Authenticated Encryption in SSH: Provably Fixing the SSH Binary Packet Protocol." *Ninth ACM Conference on Computer And Communications Security* 2002.

[BBKN01] M. Bellare, A. Boldyreva, L. Knudsen, and C. Namprempre. "On-line Ciphers, and the Hash-CBC Constructions." *Proceedings of Advances in Cryptology, 2001*

[BDJR97] 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 the 38th Annual Symposium on Foundations of Computer Science* IEEE: 1997.

[BT04] A. Boldyreva, and N. Taesombut. "On-line Encryption Schemes: New Security Notions and Constructions." *RSA, Cryptographer's Track, 2004*

[Cam78] C. Campbell. "Design and Specification of Cryptographic Capabilities." *Computer Security and the Data Encryption Standard*, National Bureau of Standards Special Publications 500-27, US Department of Commerce: 1978.

[DH76] W. Diffie, and M. Hellman. "New Directions in Cryptography." *IEEE Transactions on Information Theory.* Vol 22. No 6. 1976.

[Dwo01] M. Dworkin. "NIST Special Publication 800-38A: Recommendation for Block Cipher Modes of Operation, Methods and Techniques." National Institute for Science and Technology: 2001.

[Dwo02] M. Dworkin. "NIST Special Publication 800-38B: Recommendation for Block Cipher Modes of Operation: The RMAC Authentication Mode, Methods and Techniques." National Institute of Science and Technology: 2002.

[FJMV03] P. Fouque, A. Joux, G. Martinet, and F. Valette. "Authenticated On-Line Encryption." *Proceedings of the Selected Areas of Cryptography Conference, 2003.*

[FJP04] P. Fouque, A. Joux, and G. Poupard. "Blockwise Adversarial Model for On-line Ciphers and Symmetric Encryption Schemes." *Selected Areas of Cryptography.* 2004.

[FMP03] P. Fouque, G. Martinet, G. Poupard, "Practical Symmetric On-Line Encryption." *Proceedings of the Fast Software Encryption Conference, 2003.*

[GD00] V. Gligor, and P. Donescu. "On Message Integrity in Symmetric Encryption." *1st NIST Workshop on AES Modes of Operation.* 2000.

[GD01] V. Gligor, and P. Donescu. "Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes." *2nd NIST Workshop on AES Modes of Operation* 2001.

[GB] S. Goldwasser, and M. Bellare. *Lecture Notes on Cryptology. Web Version* http://www-cse.ucsd.edu/users/mihir/papers/gb.pdf

[JMV02] A. Joux, G. Martinet, and F. Valette. "Blockwise-Adaptive Attacks. Revisiting the (In)Security of Some Provably Secure Encryption Modes: CBC, GEM, IACBC." *Proceedings of Advances in Cryptology, 2002.*

[KY00] J. Katz, and M. Yung. "Complete Characterization of Security Notions for Probabilistic Private-Key Encryption." *Proceedings of the 32nd ACM Annual Symposium on Theory of Computing* 2000.

[Knu00] L. Knudsen. "Block Chaining Modes of Operation." *Symmetric Key Block Cipher Modes of Operation Workshop.* 2000.

[Kra94] H. Krawczyk. "LFSR-based hashing and authenticating." *Proceedings of Advances in Cryptology, 1994.*

[LRW00] H. Lipmaa, P. Rogaway, and D. Wagner. "Comments to NIST concerning AES Modes of Operation: CTR-Mode Encryption." *Symmetric Key Block Cipher Modes of Operation Workshop.* 2000.

[RBBK01] P. Rogaway, M. Bellare, J. Black, and T. Krovetz. "OCB: A Block Cipher Mode of Operation for Efficient Authenticated Encryption." *Eighth ACM Conference on Computer And Communications Security* 2001.

A Equivalence of Primitive and General Games

Theorem 4 *An encryption scheme is general blockwise-adaptive chosen-plaintext secure, if and only if it is secure in the primitive blockwise-adaptive chosen-plaintext game.*

Proof

Since the primitive blockwise-adaptive game is merely a special case of the general game, an adversary who can win the primitive game can win the general game, and so (by contrapositive) general security implies primitive security as well.

To show that primitive security implies general security, we will demonstrate that if there exists an adversary who can win the general blockwise-adaptive chosen-plaintext game, then there exists an adversary who can win the primitive blockwise-adaptive chosen-plaintext game (both in polynomial time with non-negligible advantage).

Suppose there exists a cryptosystem secure against the *primitive* blockwise-adaptive game, which means there is no adversary which can achieve non-negligible advantage in polynomial time. Suppose further, there is an adversary *Gen* who can achieve non-negligible advantage in the *general* blockwise-adaptive game, again in polynomial time. We will demonstrate that this leads to a contradiction.

Let N be the random variable that is the number of queries that *Gen* will make. Let n be such that $Pr[N \leq n] \geq 1/2$. To see that such an n exists, see the proof of Theorem 2. It is a requirement of the primitive and general blockwise-adaptive games that the secret bit b remain constant throughout the entire game. However, we will violate this rule, and create a series of games $G_1, G_2, \ldots, G_{n+1}$. The ith oracle query in Game G_j will work as follows:

$$LRBW_{(b,sk)}(P,Q) = \begin{cases} i < j & LRBW_{(b,sk)}(P,P) \\ j \leq i \leq n & LRBW_{(b,sk)}(Q,Q) \\ i > n & abort \end{cases}$$

Thus the game G_j will pretend as if the secret bit were 0 until and not including the jth query. All queries after and including the jth, until and including the nth, will be as if the secret bit were 1. Note that during the game G_1, it is as if the secret bit were always 1, and during the game G_{n+1}, it is as if the secret bit were always 0.

One cannot expect the output of adversary *Gen* to have any of its original properties since the behavior of the oracle has been changed. But nonetheless, its output is a random variable on the domain $\{0,1\}$. Suppose that the outputs (as random variables) during games G_x and G_{x+1} are computationally distinguishable, for some x in the range $1, 2, \ldots n$.

This means there is an Algorithm *Diff* which can, with probability $1/2+\delta$, correctly guess which game has been played, and that δ is non-negligible. Now Algorithm *Prim* will play the primitive blockwise-adaptive game as follows. First, it will execute Algorithm *Gen*, and receive its queries in the form (P,Q). It will submit a query to its own oracle for each of these. For queries $1, 2, \ldots, x-1$, it will be (P,P). For query x it will be honest, or (P,Q), and for queries $x, x+1, \ldots, n$ it will be (Q,Q). Finally if an $n+1$th query is given, it will give up and guess a bit equal to the value of a fair coin. It is easy to see that *Prim* only makes at most one split query, and so does not violate the rules of the primitive game. If the algorithm does not abort, Algorithm *Gen* will report a guess, and this will be passed to the distinguisher Algorithm *Diff*.

Observe that if the secret bit of the primitive game is actually one, then G_x has been played. And if the secret bit of the primitive game is actually zero, then G_{x+1} has been played. Since Algorithm *Diff* can distinguish between these correctly with probability $1/2 + \delta$, then Algorithm *Prim* should guess 0 if *Diff* returns G_{x+1} and 1 if *Diff* returns G_x. Obviously *Prim* will be correct if *Diff* is correct and there is no abortion, or correct half the time if there is an abortion.

Since the probability of an abortion is at most $1/2$, then the advantage of *Prim* is at least $\delta/2$. This is non-negligible, and so *Prim* can win the primitive blockwise-adaptive game, which is a contradiction. Therefore G_x and G_{x+1} are computationally indistinguishable, for all x in the range $1, 2, \ldots, n$.

Note further that computational indistinguishability is transitive so long as the sequence of objects compared is polynomial in length. Since *Gen* runs in polynomial time, there are polynomially many queries, and so n is upper-bounded by a polynomial. Therefore we can conclude, by transitivity, that G_1 is computationally indistinguishable from G_{n+1}.

But note, that G_1 is the general blockwise-adaptive game with the secret bit set to 1, and G_{n+1} is the same with the secret bit set to zero. Since *Gen* wins the general blockwise-adaptive game with non-negligible advantage, it does in fact distinguish between G_1 and G_{n+1} in polynomial time and with probability non-negligibly different from one-half. This is the required contradiction.

Therefore no such algorithm *Gen* can exist, and any system secure against the primitive blockwise-adaptive game is secure against the general blockwise-adaptive game as well.

■

B Specific Encryption Schemes

This section will now apply the theorems of the paper to prove secure or insecure nine of the ten modes of encryption described in this paper.

B.1 Previous Work

An indirect proof of insecurity of CBC was first given by Bellare, by demonstrating the flaw in SSH [BKN02], and is discussed in more detail by Joux, et al [JMV02]. The proof of security of HPCBC, against blockwise-adaptive CPA (as implied by IND-BLK-CCA[8]) was first given by Boldyreva, et al [BT04]. The mode CFB was proven secure almost simultaneously by both Fouque, et al [FMP03] and Boldyreva and Taesombut [BT04]. The mode IGE was analyzed by Gligor, et al, and Bellare, et al, and shown to be insecure against messagewise chosen-plaintext attack [GD00, BBKN01]. Since IGE and likewise ECB are not messagewise CPA secure, clearly they are not blockwise-adaptive CPA secure, but ECB and IGE are included to show the flexibility of Canonical Form. While XCBC can be expressed in Canonical Form, the authors have yet to determine its security. The security of CTR was first suggested by Joux, et al, [JMV02], and proven by Fouque, et al, [FJP04].

The authors believe that this paper contains the first proof of security of OFB and HCBC against blockwise-adaptive CPA. The authors also believe that this is the first analysis of the blockwise-adaptive chosen-plaintext security of ABC in its various forms. Nonetheless, the flexibility of the Canonical Form and the general theorems that follow from it allow many other proposed methods to be proven secure or insecure.

The encryption schemes CBC, CTR and OFB have already been discussed. The remaining seven follow below.

B.2 Electronic Codebook

The mode ECB is not even messagewise chosen-plaintext secure, so it certainly cannot be blockwise chosen-plaintext secure. However, we list the Canonical Form to demonstrate the flexibility of that notation:

For ECB, there is no initialization vector and one normally writes:
$C_i = F_{sk}(P_i)$

However, in Canonical Form:
$Init_{sk}(k) : C_0 = \emptyset$
$Pre(P_i, s_i) = P_i; \quad Post(P_i, s_i, y) = y; \quad Update(P_i, s_i, y) = \emptyset$
Where the symbol $\emptyset$ represents the empty string, or lack of any state.

B.3 Cipher Feedback Mode

Cipher Feedback Mode, or CFB, was proven blockwise-adaptive chosen-plaintext Secure by [BT04, FMP03].

The classic form is
$r_1 \leftarrow \{0,1\}^k; \quad C_0 = r_1$
$C_i = F_{sk}(C_{i-1}) \oplus P_i$

However, in Canonical Form:
$Init_{sk}(k) : r \leftarrow \{0,1\}^k; \quad C_0 = s_1 = r$
$Pre(P_i, s_i) = s_i; \quad Post(P_i, s_i, y) = y \oplus P_i; \quad Update(P_i, s_i, y) = y \oplus P_i$
Where the state s_i represents the previous ciphertext.

[8] The modified form of Blockwise CCA for on-line schemes.

The operation of *Post* is merely to XOR the pseudorandom function's output with the plaintext, as in CTR or OFB, and so is entropy-preserving. Moreover, CFB is collision-resistant, since the attacker has no control over the inputs of the pseudorandom function (as in OFB). Only the previous ciphertext is an input and thus there is no adversary who can win the collision game with non-negligible advantage. Therefore since CFB is entropy-preserving and collision-resistant it is blockwise adaptive chosen-plaintext secure.

B.4 Accumulated Block Ciphers

Accumulated Block Ciphers are a class of encryption schemes, first proposed by Knudsen [Knu00], based on a function denoted h. Suppose that h is publicly computable.

In the usual notation:
$P'_i \leftarrow P_i \oplus h(P'_{i-1})$
$C_i \leftarrow F_{sk}(P'_i \oplus C_{i-1}) \oplus P'_{i-1}$

The values P_0 and C_0 act as initialization vectors for the scheme. In particular, if P_0 and C_0 are secret, then one writes S-ABC, and if they are public, P-ABC. The initialization algorithm merely randomly selects P_0 and C_0.

In Canonical Form, there are actually two state variables, the previous ciphertext and (primed) plaintext. But, we can think of these as two binary strings concatenated into a larger string, since they are always of a fixed and known size. Thus we write the state $s_i = s'_i || s''_i$, where s'_i is the previous ciphertext and s''_i the previous (primed) plaintext.

$Init_{sk}(k) : r \xleftarrow{R} \{0,1\}^k; \quad C_0 = r; \quad s_1 = C_0 || P_0$
$Pre(P_i, s_i) = P_i \oplus s'_i \oplus h(s''_i); \quad Post(P_i, s_i, y) = y \oplus s''_i;$
$s'_{i+1} = y \oplus s''_i; \quad s''_{i+1} = P_i \oplus h(s''_i); \quad Update(P_i, s_i, y) = s'_{i+1} || s''_{i+1}$

Since *Post* is an XOR of the pseudorandom output and a state variable, if the state is held constant, *Post* is injective as a map from y to the ciphertext. Thus the scheme is entropy-preserving. It is useful to note that

$s''_{i+1} = P'_{i+1} = P_i \oplus h(P_{i-1} \oplus h(P_{i-2} \oplus h(P_{i-3} \oplus \cdots h(P_0) \cdots)))$

Therefore, if P_0 is public, and h is publicly computable, then s''_{i+1} is known to the adversary for all P_i where $P_1, P_2, P_3, \ldots P_i$ are all known.

Note that $C_i \oplus C_j = (y_i \oplus s''_i) \oplus (y_j \oplus s''_j)$ or that

$C_i \oplus C_j \oplus s''_j \oplus s''_i = y_i \oplus y_j$

And so therefore the value $C_i \oplus C_j \oplus s''_j \oplus s''_i$ is zero if and only if $y_i = y_j$. Thus the system is collision-verifiable, if s''_i is known to the adversary.

Let us consider for the following scenario. If the adversary receives C_0 and submits (P_1, P_1), and receives C_1. Then if P_2 is set to be $P_1 \oplus C_0 \oplus h(P_0) \oplus h(P_1 \oplus h(P_0)) \oplus C_1$ therefore

$$
\begin{aligned}
Pre(P_2, s_2) &= P_2 \oplus s_2' \oplus h(s_2'') \\
&= P_2 \oplus C_1 \oplus h(s_2'') \\
&= P_2 \oplus C_1 \oplus h(P_1 \oplus h(P_0)) \\
&= [P_1 \oplus C_0 \oplus h(P_0) \oplus h(P_1 \oplus h(P_0)) \oplus C_1] \oplus C_1 \oplus h(P_1 \oplus h(P_0)) \\
&= P_1 \oplus C_0 \oplus h(P_0) \oplus h(P_1 \oplus h(P_0)) \oplus h(P_1 \oplus h(P_0)) \\
&= P_1 \oplus C_0 \oplus h(P_0) \\
&= P_1 \oplus s_1' \oplus h(s_1'') \\
&= Pre(P_1, s_1)
\end{aligned}
$$

And thus a collision on the output of Pre has been caused in polynomial time. The value of P_2 is computable if h is publicly computable and P_0 is known. Thus if h is publicly computable, and P_0 and C_0 are known to the adversary, then the scheme is not collision-resistant, but is entropy-preserving and collision-verifiable. This is the case for P-ABC, and we can conclude that it is then not blockwise-adaptive chosen-plaintext secure.

The case of S-ABC is somewhat more complex. For example, simple attacks exist if h is a linear function, but no attacks exist if h is an Almost-XOR-Universal hash function, as proven by Bellare, Boldyreva, Knudsen, and Namprempre. [BBKN01]. Lastly, we will show momentarily that IGE (a special case of ABC) is insecure even if P_0 is secret. Therefore the secrecy of P_0 is clearly insufficient for the security of ABC.

Supposing h is an Almost-XOR-Universal hash function, assume an algorithm wins the collision game with non-negligible probability. Then blocks i and j have identical outputs of the function Pre . Then

$$
\begin{aligned}
Pre(P_i, s_i) &= Pre(P_j, s_j) \\
P_i \oplus s_i' \oplus h(s_i'') &= P_j \oplus s_j' \oplus h(s_j'') \\
P_i \oplus s_i' \oplus P_j \oplus s_j' &= h(s_i'') \oplus h(s_j'')
\end{aligned}
$$

Since h is an Almost-XOR-Universal hash function family member, this means that no algorithm (PPT or otherwise) can find a triple (x_1, x_2, y) such that $h(x_1) \oplus h(x_2) = y$ but $x_1 \neq x_2$ with non-negligible probability for a random hash key [BBKN01]. Yet we have identified a triple which meets the required equality. If $s_i'' \neq s_j''$ then h is no longer Almost-XOR-Universal, since our adversary succeeds with non-negligible probability. Yet if $s_i'' = s_j''$, this means that $P_{i-1} \oplus h(s_{i-2}'') = P_{j-1} \oplus h(s_{j-2}'')$ and so the argument can be continued on the previous block. See below, under HCBC, for more details.

Thus in the case of ABC where h is an Almost-XOR-Universal hash function family member, the scheme is collision-resistant and collision-verifiable, and therefore blockwise-adaptive chosen-plaintext secure. Also, this is considered a form of S-ABC since the hash key must be secret from the attacker (in order for Almost-XOR-Universality to hold).

B.5 Infinite Garble Extension

For IGE, or Infinite Garble Extension, one normally writes

$Init_{sk}(k) : C_0 \xleftarrow{R} \{0,1\}^k$
$C_i = F_{sk}(P_i \oplus C_{i-1}) \oplus P_{i-1}$

Where C_0 is the per message initialization vector, and P_0 is a string which can be public or kept with the secret key depending on implementation. It turns out the privacy of P_0 does not impact the blockwise-adaptive chosen-plaintext security of IGE. One can see that if h is the function which always returns zero, then IGE is a specific example of ABC.

In Canonical Form, like ABC, there are actually two state variables, the previous ciphertext and plaintext. But one can think of these as two binary strings concatenated into a larger string, since they are always of a fixed and known size. Thus we write the state $s_i = s'_i || s''_i$, where s'_i is the previous ciphertext and s''_i the previous plaintext.
$Init_{sk}(k) : s_1 \xleftarrow{R} \{0,1\}^k; \quad C_0 = s_1$
$Pre(P_i, s_i) = P_i \oplus s'_i; \quad Post(P_i, s_i, y) = y \oplus s''_i;$
$s'_{i+1} = y \oplus s''_i; \quad s''_{i+1} = P_i; \quad Update(P_i, s_i, y) = s'_{i+1} || s''_{i+1}$

Suppose the first queries are P_1, P_2 and their ciphertexts are C_1, C_2. In this case, let $P_3 = C_1 \oplus P_2 \oplus C_2$. Recalling that $s'_i = C_{i-1}$, one can calculate that

$$Pre(P_3, s_3) = s'_3 \oplus P_3 = C_2 \oplus (C_1 \oplus P_2 \oplus C_2) = C_1 \oplus P_2 = Pre(P_2, s_2)$$

And so a collision has been easily caused, and IGE is not collision-resistant.

Suppose that a collision has occurred on the inputs to the pseudorandom function, for blocks i and j. Since $C_x = y_x \oplus P_{x-1}$, then
$C_i \oplus C_j \oplus P_{i-1} \oplus P_{j-1} = y_i \oplus y_j$
Thus a collision has occurred if and only if $C_i \oplus C_j \oplus P_{i-1} \oplus P_{j-1}$ is zero, and the scheme is collision-verifiable, after the first block (all plaintexts after P_0 are known to the adversary, since she/he submitted them).

Since the function *Post* is simply an XOR with a substring of the state, when the state is kept constant, *Post* is injective and thus entropy-preserving. Since the scheme is entropy-preserving, collision-verifiable and not collision-resistant, it is not blockwise-adaptive chosen-plaintext secure.

B.6 Hash Cipher Block Chaining (HCBC)

HCBC is almost identical to CBC, but the previous ciphertext is hashed before it is XOR-ed with the plaintext.

Observe, with standard notation:
$r_1 \xleftarrow{R} \{0,1\}^k; \quad C_0 = r_1$
$C_i = F_{sk}(H_{hk}(C_{i-1}) \oplus P_i)$

However, in Canonical Form:
$Init_{sk}(k) : r \xleftarrow{R} \{0,1\}^k; \quad C_0 = s_1 = r$
$Pre(P_i, s_i) = P_i \oplus H_{hk}(s_i); \quad Post(P_i, s_i, y) = y; \quad Update(P_i, s_i, y) = y$
Note above the state s_i represents the previous ciphertext.

The mode HCBC is built around some Hash function which is Almost-XOR-Universal. This means that no algorithm (PPT or otherwise) can find a triple (x_1, x_2, y) such that $H(x_1) \oplus H(x_2) = y$ but $x_1 \neq x_2$ with non-negligible probability for a random hash key [BBKN01]. Consider the

possibility that an adversary wins the collision game. This means the outputs of Pre for blocks i and j are equal.

$$\begin{aligned} Pre(P_i, s_i) &= Pre(P_j, s_j) \\ H(s_i) \oplus P_i &= H(s_j) \oplus P_j \\ H(C_{i-1}) \oplus P_i &= H(C_{j-1}) \oplus P_j \\ H(C_{i-1}) \oplus H(C_{j-1}) &= P_i \oplus P_j \end{aligned}$$

Thus the triple $(C_{i-1}, C_{j-1}, P_i \oplus P_j)$ produces the required equality, and must do so with non-negligible probability since we assumed the collision game is won with non-negligible probability. However, it may be the case that $C_{i-1} = C_{j-1}$. Yet, C_{i-1} and C_{j-1} are particular outputs of the pseudorandom function, and since CBC requires F_{sk} to be a permutation, then the inputs to F_{sk} at blocks $i-1$ and $j-1$ are equal.

In this case $H(C_{i-2}) \oplus P_{i-1} = H(C_{j-2}) \oplus P_{j-1}$, and therefore $H(C_{i-2}) \oplus H(C_{j-2}) = P_{i-1} \oplus P_{j-1}$. Thus the triple $(H(C_{i-2}), H(C_{j-2}), P_{i-1} \oplus P_{j-1})$ meets the equality criterion. This violates the Almost-XOR-Universality of H unless $C_{i-2} = C_{j-2}$. Yet, that condition would create another triple based on C_{i-3} and C_{j-3}. Eventually this leads to C_0 which is chosen at random, and will equal a particular ciphertext with negligible probability. Therefore the collision game cannot be won with non-negligible probability.

The function post is the identity function, and so is entropy-preserving. Since the scheme is collision-resistant, and entropy-preserving, it is blockwise-adaptive chosen-plaintext secure.

B.7 Hash Cipher Block Chaining (HPCBC)

This scheme is an extension of HCBC.

Written in the usual form:
$r_1 \stackrel{R}{\leftarrow} \{0,1\}^k; \quad C_0 = r_1$
$C_i = F_{sk}(H_{hk}(P_{i-1}||C_{i-1}) \oplus P_i) \oplus H_{hk}(P_{i-1}||C_{i-1})$

However, in Canonical Form:
$Init_{sk}(k) : r \stackrel{R}{\leftarrow} \{0,1\}^k; \quad C_0 = s_1 = r$
$Pre(P_i, s_i) = P_i \oplus H_{hk}(s_i); \quad Post(P_i, s_i, y) = y \oplus H_{hk}(s_i);$
$Update(P_i, s_i, y) = P_i||(y \oplus H_{hk}(s_i))$

The scheme is collision resistant for the same reasons as HCBC. The function $Post$ is an XOR of the hash of the state, and so if the state is constant then the hash is constant, and $Post$ is entropy-preserving. Since the scheme is collision-resistant and entropy-preserving, it is blockwise-adaptive chosen-plaintext secure.

B.8 Extended Cipher Block Chaining (XCBC)

Extended Cipher Block Chaining, defined by Gligor and Donescu [GD01], adds a constant random number times the block number, to each block. Since all arithmetic is being done modulo 2^k, this requires the adversary to know something of R to mount an attack.

In traditional notation,

$Init_{sk}(k) = R \xleftarrow{R} \{0,1\}^k; \quad C_0 = F_{sk}(R)$
$Z_i = F_{sk}(P_i \oplus Z_{i-1})$
$C_i = Z_i + i \times R$
$Z_0 = F_{sk'}(R)$ where $sk \neq sk'$

The key used to encrypt R into Z_0 must be different from sk and secretly shared between sender and recipient as sk is shared. In canonical form, the state will be in three parts; as before we represent this as $s_i = s'_i||s''_i||s'''_i$. Here, s'_i will be the block number, s''_i will be the value of R, and s'''_i will be the value of Z_{i-1}.

$$Init_{sk}(k) = R \xleftarrow{R} \{0,1\}^k; \quad C_0 = F_{sk}(R); \quad s'_i = 1; \quad s''_i = R; \quad s'''_i = F_{sk'}(R); \quad s_i = s'_i||s''_i||s'''_i$$

$$Pre(P_i, s_i) = P_i \oplus s'''_i; \quad Post(P_i, s_i) = y + s'_i \times s''_i$$

$$s'_{i+1} = s'_i + 1; \quad s''_{i+1} = s''_i; \quad s'''_{i+1} = y; \quad Update(P_i, s_i, y) = s'_{i+1}||s''_{i+1}||s'''_{i+1}$$

The authors have yet to determine the security status of XCBC, but this complex method can be represented in Canonical Form, which shows the flexibility of this notational system.