The Security of Ciphertext Stealing

Phillip Rogaway[1], Mark Wooding[2], and Haibin Zhang[1]

[1] Dept. of Computer Science, University of California, Davis, USA
[2] Thales e-Security Ltd, UK

Abstract. We prove the security of CBC encryption with ciphertext stealing. Our results cover all versions of ciphertext stealing recently recommended by NIST. The complexity assumption is that the underlying blockcipher is a good PRP, and the security notion achieved is the strongest one commonly considered for chosen-plaintext attacks, indistinguishability from random bits (ind\$-security). We go on to generalize these results to show that, when intermediate outputs are slightly delayed, one achieves ind\$-security in the sense of an online encryption scheme, a notion we formalize that focuses on what is delivered across an online API, generalizing prior notions of blockwise-adaptive attacks. Finally, we pair our positive results with the observation that the version of ciphertext stealing described in Meyer and Matyas's well-known book (1982) is not secure.

Keywords: blockwise-adaptive attacks, CBC, ciphertext stealing, cryptographic standards, modes of operation, provable security.

1 Introduction

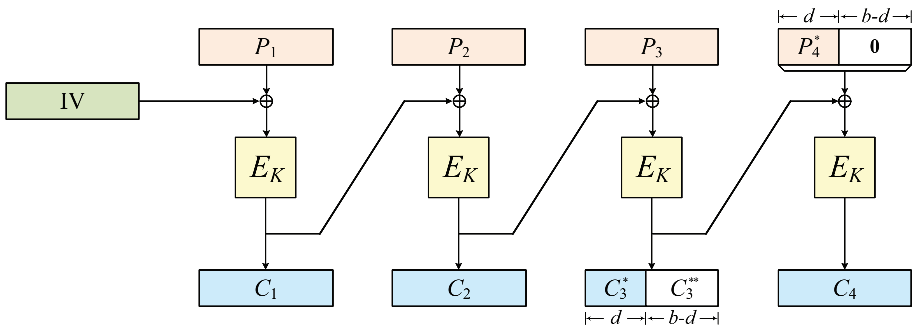
Ciphertext stealing. Many blockcipher modes require the input be a sequence of complete blocks, each having a number of bits that is the blockcipher's blocksize. One approach for dealing with inputs not of this form is *ciphertext stealing*. The classical combination is CBC encryption and ciphertext stealing, a mode going back to at least 1982 [14].

In 2010, NIST put out an addendum [8] to Special Publication 800-38A [7], the document that had defined blockcipher modes ECB, CBC, CFB, OFB, and CTR. The addendum defines three ways to enrich CBC with ciphertext stealing. The modes are named CBC-CS1, CBC-CS2, and CBC-CS3. See Fig. 1 for the definition of these modes, which differ only in the ordering of ciphertext bits.

Despite the classicism of ciphertext-stealing, its adoption in standards, and the strong preferences, these days, for proven-secure modes, there has, until now, been no proof offered for CBC with ciphertext stealing. This paper fills in this gap.

Our contributions. We begin by looking at the NIST ciphertext-stealing modes, which we collectively call CBC-CS. Assuming a random IV, we show that the CBC-CS schemes achieve the strongest conventional form of chosen-plaintext-attack (CPA) security: what we call ind\$, indistinguishability from random bits under an adaptive chosen-plaintext attack. The definition, easily shown to imply all conventional formulations of CPA-style semantic security, formalizes that a ciphertext C is indistinguishable from as many random bits.

Next we show that *delayed* versions of CBC-CS achieve an analogous IND\$ notion that we define for *online* security. The idea of delayed CBC is from Fouque, Martinet, and Poupard [11]. Our formulation for online security generalizes their and subsequent work (further history and credits coming shortly). In particular, prior definitional approaches were specific to blockcipher-based schemes of a specified form—restrictions not in keeping with identifying a general notion of security. We levy no such restrictions, but do imagine that the encryption scheme is written to an incremental API (application programming interface). Each time a user presents a piece of plaintext to encrypt she will get back a corresponding chunk of ciphertext. The length of both is arbitrary. One can understand our definition of online security as establishing that a



```
10    **algorithm** $\text{CBC-CS}_K^{IV}(P)$
11    $n \leftarrow \lceil |P|/b \rceil$
12    $P_1 \cdots P_{n-1} P_n^* \leftarrow P$ **where** $|P_1| = \cdots = |P_{n-1}| = b$
13    $P_n \leftarrow P_n^* \, 0^{b-d}$ **where** $d \leftarrow |P_n^*|$
14    $C_0 \leftarrow IV$;
15    $C_1 \cdots C_n \leftarrow \text{CBC}_K^{IV}(P_1 \cdots P_n)$ **where** $|C_1| = \cdots = |C_n| = b$
16    $C_{n-1}^* \leftarrow \text{MSB}_d(C_{n-1})$
17-1  **return** $C_1 \cdots C_{n-2} C_{n-1}^* C_n$                                        $\Leftarrow$ for CS1
17-2  **if** $d{=}b$ **return** $C_1 \cdots C_{n-2} C_{n-1}^* C_n$ **else return** $C_1 \cdots C_{n-2} C_n C_{n-1}^*$  $\Leftarrow$ for CS2
17-3  **return** $C_1 \cdots C_{n-2} C_n C_{n-1}^*$                                        $\Leftarrow$ for CS3
```

```
20    **algorithm** $\text{CBC}_K^{IV}(P_1 \cdots P_n)$ **where** $|P_1| = \cdots = |P_n| = b$
21    $C_0 \leftarrow IV$
22    **for** $i \leftarrow 1$ **to** $n$ **do** $C_i \leftarrow E_K(C_{i-1} \oplus P_i)$
23    **return** $C_1 \cdots C_n$
```

Fig. 1. Encryption under NIST modes CBC-CS1, CBC-CS2, and CBC-CS3. The schemes differ only in which version of line 17 is used. The schemes depend on a blockcipher $E\colon \mathcal{K} \times \{0,1\}^b \to \{0,1\}^b$ that determines the key space $\mathcal{K}$, the IV space $\mathcal{IV}$, and the message space $\mathcal{P} = \{0,1\}^{\geq b}$. We insist that $K \in \mathcal{K}$, $IV \in \mathcal{IV}$, and $P \in \mathcal{P}$.

specified incremental API introduces no new security vulnerabilities. Technically, we reconceptualize an encryption scheme *as* the incremental interface. We regard a general definition for online security—a definition motivated by cryptographic APIs and not the characteristics of any particular encryption mode—as an important and independent contribution of this paper.

The workings of delayed CBC—the naturalness of this scheme and how much one must delay—are clarified by freeing the definition of online security from a demand on a scheme being blockcipher-based. Now it is the security analysis, not the syntax, that surfaces by just how much one must delay—an amount that is, in fact, slightly different for the CS1/CS2 and the CS3 versions of the scheme. Absent a careful treatment of such matters the author of an incremental API could well get these things wrong, buffering more than what is necessary or less than what is needed.

Finally, we point out that a 30-year-old version of ciphertext stealing described in the book of Meyer and Matyas [14] is essentially wrong: it will not achieve any desirable security notion we know. The apparently unnoticed observation highlights the importance of having proofs in this domain, and underscores NIST's wisdom in selecting the versions of ciphertext stealing that it did.

Additional history. The provable-security treatment of CBC, and of other blockcipher-based encryption modes, begins with Bellare, Desai, Jokipii, and Rogaway [3]. The stronger ind$-definition that we adopt here is from Rogaway, Bellare, Black, and Krovetz [16]. For online security, the delayed-CBC scheme that we embellish with NIST's versions of ciphertext stealing is due to Fouque, Martinet, and Poupard [11].

Our definition of online security springs from the line of work on blockwise-adaptive attacks that starts with Bellare, Kohno, and Namprempre [4] and Joux, Martinet, and Valette [13] and continues with Fouque, Martinet, and Poupard [11], Fouque, Joux, and Poupard [10], and Bard [2]. As explained, our own security definitions take a different turn by divorcing the notion of online security from its former association with blockcipher-based schemes. We instead assume an arbitrary symmetric encryption scheme that is presented to the user by way of an incremental API. The user provides the plaintext as a sequence of chunks and the encryption algorithm, buffering what it needs, returns corresponding ciphertext chunks. The approach echos Gennaro and Rohatgi [12], which likewise transplants a primitive (digital signatures) from a setting that sees messages as atomic to one that sees messages as something produced and consumed across an expanse of time.

Discussion. A possible reaction to any discussion of ciphertext stealing is to say: forget it, use CTR mode instead. We are sympathetic to this point of view, knowing no convincing reason to favor CBC encryption over CTR mode, which natively handles plaintexts of arbitrary length. But the fact remains that CBC encryption is widely used, and that ciphertext stealing is a classical, standardized, and elegant way to extend it. This makes it worth attending to.

In justifying the use of ciphertext stealing in a mode that employed it, Matt Ball writes that "[d]espite lacking a formal security proof, ciphertext stealing still has general approval in the cryptographic community" [1, p. 5]. Probably this statement is at some level true, but "general approval" is hard to gauge and far removed from being a proof.

We think that security notions that attend to the vulnerabilities introduced by the specifics of an envisioned API comprise an interesting direction in narrowing the gap between conventional abstractions of cryptographic primitives and what cryptographic practice actually exports. It is not just that protocols may segment conceptually atomic messages (the original motivation for dealing with blockwise adaptivity); rather, it is that the segmentation is actually surfaced to users, and therefore desirable to directly model.

We do not discuss the security of CBC-CS when the IV fails to be unpredictable; it would seem that no interesting or desirable security notion is achieved in this case. NIST SP800-38A appropriately demands an unpredictable IV for CBC [7, Appendix C].

The CBC-CS schemes predate NIST's addendum [8]: CBC-CS2 goes back to at least 1996 [17], while older versions of ciphertext stealing go back to at least 1982 [14]. Looking at these schemes from a modern vantage is long overdue.

2 Preliminaries

Notation. Strings are assumed to be binary, elements of $\{0,1\}^*$. Both $A \,\|\, B$ and $A\,B$ denote the concatenation of strings A and B. If X is a string then $|X|$ is its length. The empty string is denoted ε. Throughout this paper we fix an integer $b \geq 1$ called the *blocksize.* For a string X and a number $d \leq |X|$ let $\mathrm{MSB}_d(X)$ and $\mathrm{LSB}_d(X)$ be the leftmost and rightmost d bits of X.

Blockciphers. A *blockcipher* is a map $E\colon \mathcal{K} \times \{0,1\}^b \to \{0,1\}^b$ where $\mathcal{K} \subseteq \{0,1\}^*$ is finite and $E_K(\cdot) = E(K, \cdot)$ is a permutation for each $K \in \mathcal{K}$. Let $\mathrm{Perm}(b)$ be the set of all permutations on b bits. This may be regarded as a blockcipher with a $(2^b!)$-size key space. Let $\mathbf{Adv}_E^{\mathrm{prp}}(A) = \Pr[\,A^{E_K(\cdot)} \Rightarrow 1\,] - \Pr[\,A^{\pi(\cdot)} \Rightarrow 1\,]$ with $K \overset{\$}{\leftarrow} \mathcal{K}$ and $\pi \overset{\$}{\leftarrow} \mathrm{Perm}(b)$. Similarly define $\mathbf{Adv}_E^{\mathrm{prf}}(A) = \Pr[\,A^{E_K(\cdot)} \Rightarrow 1\,] - \Pr[\,A^{\rho(\cdot)} \Rightarrow 1\,]$ with $K \overset{\$}{\leftarrow} \mathcal{K}$ and $\rho \overset{\$}{\leftarrow} \mathrm{Func}(b)$ for $\mathrm{Func}(b)$ the set of all functions from b bits to b bits. Here $E_K(\cdot)$ need not be a permutation.

Encryption schemes. It has become traditional to regard blockciphers as fixed functions but encryption schemes as tuples, as in $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$. To simplify and unify matters we formalize an encryption scheme more like a blockcipher: an (IV-based, symmetric) *encryption scheme* is a function $\mathcal{E}\colon \mathcal{K} \times \mathcal{IV} \times \mathcal{P} \to \mathcal{P}$. We call $\mathcal{K}$, $\mathcal{IV}$, and $\mathcal{P}$ the *key space*, *IV space*, and *message space*. For simplicity we assume that $\mathcal{K}$ is finite and $\mathcal{IV}$ is the set of all strings of some one particular length. We write $\mathcal{E}_K^{IV}(P)$ instead of $\mathcal{E}(K, IV, P)$. To keep things simple we require that $\mathcal{E}_K^{IV}(\cdot)$ be a length-preserving permutation for all $K \in \mathcal{K}$ and $IV \in \mathcal{IV}$. The condition implies that $\mathcal{E}$ has a unique inverse, the map $\mathcal{D}$ where $\mathcal{D}_K^{IV}(C) = P$ when $\mathcal{E}_K^{IV}(P) = C$. Because there is no formal need to specify the decryption direction $\mathcal{D}$ of an encryption scheme $\mathcal{E}$, we never do so. Of course it is important in practice that $\mathcal{E}$ and $\mathcal{D}$ have efficient realizations, it is simply that this doesn't show up in the statement of definitions or security results.

Let $\mathcal{E}\colon \mathcal{K} \times \mathcal{IV} \times \mathcal{P} \to \mathcal{P}$ be an IV-based encryption scheme and let A be an adversary (algorithm) with one of two types of oracles. A *real* encryption oracle Real$(\cdot)$ chooses a random $K \xleftarrow{\$} \mathcal{K}$ and then, on input $P \in \mathcal{P}$, returns $C \leftarrow IV \parallel \mathcal{E}_K^{IV}(P)$ for a random $IV \xleftarrow{\$} \mathcal{IV}$. A *fake* encryption oracle Fake$(\cdot)$ takes an input $P \in \mathcal{P}$ and returns $C \xleftarrow{\$} \{0,1\}^c$ where $c = |IV| + |P|$ (for $IV \in \mathcal{IV}$). Define $\mathbf{Adv}_{\mathcal{E}}^{\text{ind\$}}(A) = \Pr[\,A^{\text{Real}(\cdot)} \Rightarrow 1\,] - \Pr[\,A^{\text{Fake}(\cdot)} \Rightarrow 1\,]$. This "indistinguishability-from-random-bits" definition is easily shown to imply all conventional (CPA) formulations of indistinguishability and semantic security [3]; that we have selected a different syntax makes no difference in the proofs.

Note that even though the encryption function is formalized as taking, besides the key, an IV and a plaintext, the security definition does not allow the adversary to specify the IV; the adversary asks P and the IV is randomly generated, used, and and returned. Our security notion thus formalizes security for random IVs, not, for example, security for nonce IVs.

3 Conventional Security of the CBC-CS Schemes

We begin with a simple proposition about the security of conventional CBC encryption (no ciphertext stealing) with a random IV. The result is needed insofar as we deduce the security of CBC-CS from it. Recall that the mode was defined in Fig. 1 and was proven secure by Bellare *et al.* [3]. That proof, however, is for a somewhat weaker definition than the one we use here. The proof below is a simple application of the game-playing technique [5, 18].

Lemma 1. *Suppose A asks queries totaling at most σ blocks. Then we have* $\mathbf{Adv}_{\text{CBC[Perm}(b)]}^{\text{ind\$}}(A) \le \sigma^2/2^b$.

Proof. The difference between $\mathbf{Adv}_{\text{CBC[Perm}(b)]}^{\text{ind\$}}(A)$ and $r = \mathbf{Adv}_{\text{CBC[Func}(b)]}^{\text{ind\$}}(A)$ is at most $0.5\,\sigma^2/2^b$; this is a standard application of PRP/PRF switching [5]. It thus suffices to bound r by $r \le 0.5\,\sigma^2/2^b$. To that end, consider the games of Fig. 2. Observe that, with $\mathcal{E} = \text{CBC[Func}(b)]$, $\Pr[A^{\text{Real}(\cdot)} \Rightarrow 1] = \Pr[A^{G_1(\cdot)} \Rightarrow 1]$, while $\Pr[A^{\text{Fake}(\cdot)} \Rightarrow 1] = \Pr[A^{G_0(\cdot)} \Rightarrow 1]$. As a consequence, we have that $r = \Pr[A^{\mathcal{G}_1(\cdot)} \Rightarrow 1] - \Pr[A^{\mathcal{G}_0(\cdot)} \Rightarrow 1]$ and, the two games being identical-until-*bad*, we know that $r \le \Pr[A^{G_0}$ sets *bad*$]$. Because in game G_0 all of the C_i values are uniform and independent of P_i, so too all of the X_i values are uniform and independent of one another, so the probability that *bad* gets set—the probability some two of the X_i's collide—is at most $(1 + 2 + \cdots (\sigma - 1))/2^b \le 0.5\,\sigma^2/2^b$. This completes the proof.

Turning now to the CBC-CS modes, we claim that these inherit CBC's security with no quantitative degradation. The needed observation is that CBC-CS1$_K^{IV}(P)$ is just CBC$_K^{IV}(P0^*)$ (minimal padding to the next multiple of b bits) with some bits excised and some bits reordered. Which bits are excised and how bits are rearranged depends only on $|P|$. Thus if CBC$_K^{IV}(\cdot)$ looks random, so too will look CBC-CS1$_K^{IV}(\cdot)$. The same comments hold for CBC-CS2 and CBC-CS3; these are just different rearrangements of the bits of CBC$_K^{IV}(P\,0^*)$. The observation and proof are formalized by the proposition below.

```
algorithm $\mathrm{Enc}(P)$                                              Game $G_0$
$P_1 \cdots P_n \leftarrow P$ where $|P_1| = \cdots = |P_n| = b$                 $\boxed{\text{Game } G_1}$
$C_0, \ldots, C_n \xleftarrow{\$} \{0,1\}^b$
for $i \leftarrow 1$ to $n$ do
    $X_i \leftarrow P_i \oplus C_{i-1}$
    if $\rho(X_i)$ then $bad \leftarrow \mathsf{true}$, $\boxed{C_i \leftarrow \rho(X_i)}$
    $\rho(X_i) \leftarrow C_i$
return $C_0\, C_1 \cdots C_n$
```

Fig. 2. Proof of the ind\$-security of CBC encryption with a random IV. This application of game-playing is probably simple and well-known enough to be considered folklore. Game G_1 includes the boxed statement following the setting of *bad*; game G_0 omits it. Variable *bad* is initialized to false and ρ is initialized to everywhere undefined, a value treated as false if used as a boolean.

Theorem 1. *Let $\mathcal{E}$ be any of* CBC-CS1[Perm(b)], CBC-CS2[Perm(b)], *or* CBC-CS3[Perm(b)] *and suppose adversary A asks queries totaling at most σ blocks. Then $\mathbf{Adv}^{\mathrm{ind\$}}_{\mathcal{E}}(A) \le \sigma^2/2^b$.*

Proof. Suppose that A, asking σ total blocks of queries, gets advantage δ at distinguishing oracles $\mathcal{E} = \text{CBC-CS1}(\cdot)$ and $\$(\cdot)$. The first of these oracles chooses a random permutation $\pi \xleftarrow{\$} \mathrm{Perm}(n)$ and then, when asked a query $P \in \{0,1\}^{\ge b}$, returns $IV \parallel \text{CBC-CS1}^{IV}_{\pi}(P)$ for a random $IV \xleftarrow{\$} \{0,1\}^b$; the second oracle, when asked a query P, returns a random string of length $b+|P|$. We construct from A an adversary B that, also asking σ blocks worth of queries, also gets advantage δ, but now at distinguishing between $\mathrm{CBC}(\cdot)$ and $\$(\cdot)$. The first of these oracle chooses a random permutation $\pi \xleftarrow{\$} \mathrm{Perm}(n)$ and then, when asked a query $P \in (\{0,1\}^b)^+$, returns $IV \parallel \mathrm{CBC}^{IV}_{\pi}(P)$ for a random $IV \xleftarrow{\$} \{0,1\}^b$. Adversary B now works as follows: it runs A and when A generates a query of $P \in \{0,1\}^{\ge b}$ adversary B queries its own oracle on $P' = P\,0^*$, meaning P padded on the right with the minimal number of zero-bits so that P' is a multiple of b bits. Suppose this returns a ciphertext $C = C_0 C_1 \cdots C_n$ where $|C_i| = n$. Then B returns to A the string $C^* = C_0 C_1 \cdots C^*_{n-1} C_n$ where $C^*_{n-1} = \mathrm{MSB}_d(C_{n-1})$ and $d = b - (|P| \bmod b)$. We observe that $\Pr[B^{\text{CBC-CS1}(\cdot)} \Rightarrow 1] = \Pr[A^{\mathrm{CBC}(\cdot)} \Rightarrow 1]$ (we have reordered bits exactly as required by CBC-CS1) and that $\Pr[B^{\$(\cdot)} \Rightarrow 1] = \Pr[A^{\$(\cdot)} \Rightarrow 1]$ (reordered and pruned uniform random bits are still uniform), and so $\delta = \mathbf{Adv}^{\mathrm{ind\$}}_{\text{CBC-CS1[Perm}(b)]}(A) = \mathbf{Adv}^{\mathrm{ind\$}}_{\text{CBC[Perm}(b)]}(B)$. By Proposition 1 we thus have $\delta \le 0.5\,\sigma^2/2^b$. This establishes the first of the three results. The analogous results for CBC-CS2 and CBC-CS3 are obtained simply by modifying the string C^* returned to A: for CBC-CS2 return $C^* = C_0 C_1 \cdots C_{n-2} C^*_{n-1} C_n$ when $|P|$ is a multiple of b and $C^* = C_0 C_1 \cdots C_{n-2} C_n C^*_{n-1}$ otherwise; for CBC-CS3 always return $C^* = C_0 C_1 \cdots C_{n-2} C_n C^*_{n-1}$. This completes the theorem.

The proof's simplicity stems from having unidentified a clean abstraction boundary: directly modifying the proof of Lemma 1 to attend to the ciphertext stealing would be much more complex.

Finally, one can pass from the information-theoretic result to its complexity-theoretic analog in the standard way, trading the family of random permutations for a conventional blockcipher. Stating the result for completeness, we have the following.

Corollary 1. *Let $E\colon \mathcal{K} \times \{0,1\}^b \to \{0,1\}^b$ be a blockcipher and let $\mathcal{E}$ be any of the encryption schemes* CBC-CS1[E], CBC-CS2[E], *or* CBC-CS3[E]. *Suppose A asks queries that total σ blocks, runs in time t, and achieves advantage $\delta = \mathbf{Adv}^{\mathrm{ind\$}}_{\mathcal{E}}(A)$. Then there is an adversary B, explicitly known and constructed from A in a blackbox manner, that asks at most σ queries, runs in time at most $t + \lambda b\sigma$, and achieves advantage $\mathbf{Adv}^{\mathrm{prp}}_{E}(B) \ge \delta - \sigma^2/2^b$. Here λ is an absolute constant depending only on details of the model of computation.*

4 Defining Online Security

Syntax. We adjust the syntax of an encryption scheme to accommodate the staged presentation of plaintexts and ciphertexts. Rather than messages being atomic objects that get encrypted all at once, messages may be arbitrarily partitioned into chunks, each of which gets fed into a stateful encryption engine. Breaking with former treatments, we do not assume that chunks are single blocks, nor multiples of blocks, where the length of a block is the blocksize of some underlying blockcipher. Instead, we provide a general definition where one assumes nothing about the structure of the underlying encryption scheme (in particular, there is no assumption that it is blockcipher-based). As each installment of plaintext is provided to the encryption interface, it is up to the algorithm to decide how much ciphertext to spit out. The algorithm will thus return not only a ciphertext chunk, but also an updated state.

Realizing the idea above, we choose to define an *online encryption scheme* as a function $\mathcal{E}\colon \mathcal{K} \times \mathcal{V} \times \{0,1\} \times \{0,1\}^* \to \{0,1\}^* \times \mathcal{V}$. We write $\mathcal{E}_K^{V,\delta}(P)$ for $\mathcal{E}(K,V,\delta,P)$. We call $\mathcal{K}$ and $\mathcal{V}$ the *key space* and *state space*, respectively. The key space is finite and the state space is a finite set of strings. The third argument to $\mathcal{E}$, a bit, is the *end-of-message indicator*. The final argument to $\mathcal{E}$ is the next chunk of *message*. An online encryption scheme $\mathcal{E}$ must have an associated *IV space* $\mathcal{IV} \subseteq \mathcal{V}$ and *message space* $\mathcal{P} \subseteq \{0,1\}^*$. The former contains strings of some one fixed length. Formally, an online encryption scheme is the tuple $(\mathcal{E}, \mathcal{IV}, \mathcal{P})$, but we will usually use the first component as shorthand for the whole.

We also impose a number of "syntactic" requirements on an online encryption scheme $(\mathcal{E}, \mathcal{IV}, \mathcal{P})$. First we define some additional notation. We write $(C_1, \ldots, C_n) \leftarrow \mathcal{E}_K^{IV}(P_1, \ldots, P_n)$ for the sequence:

```
$V_0 \leftarrow IV$
**for** $i \leftarrow 1$ to $n-1$ **do** $(C_i, V_i) \leftarrow \mathcal{E}_K^{V_{i-1},0}(P_i)$
$(C_n, V_n) \leftarrow \mathcal{E}_K^{V_{n-1},1}(P_n)$
**return** $(C_1, \ldots, C_n)$.
```

Alternatively, we can think of $\mathcal{E}_K^{IV}(P_1, \ldots, P_n)$ as returning a single string, setting $\mathcal{E}_K^{IV}(P_1, \ldots, P_n)$ to $C = C_1 \cdots C_n$ where $(C_1, \ldots, C_n) \leftarrow \mathcal{E}_K^{IV}(P_1, \ldots, P_n)$.

Now fix an online encryption scheme $(\mathcal{E}, \mathcal{IV}, \mathcal{P})$.

- The *consistency requirement* says that you get the same ciphertext regardless of how you split up the plaintext. More formally, if $P_1 \parallel \cdots \parallel P_n = P'_1 \parallel \cdots \parallel P'_{n'} = P \in \mathcal{P}$ then $\mathcal{E}_K^{IV}(P_1, \ldots, P_n) = \mathcal{E}_K^{IV}(P'_1, \ldots, P'_{n'})$. We can therefore write this as $\mathcal{E}_K^{IV}(P)$ without ambiguity.
- The *invertibility requirement* is that $\mathcal{E}_K^{IV}(\cdot)$ is injective on $\mathcal{P}$ (for all $K \in \mathcal{K}$ and $IV \in \mathcal{IV}$).
- The *length requirement* is that the length of the first and second components of $\mathcal{E}_K^{V,\delta}(P)$ depend only on $|V|$, $|P|$, and δ. This ensures that, when $(C_1, \cdots, C_m) \leftarrow \mathcal{E}_K^{IV}(P_1, \cdots, P_m)$, the lengths of $C_1, C_2, \ldots, C_m$ reveal nothing about $P = P_1 \cdots P_m$ beyond how it was partitioned up.

Indistinguishability. We define a very strong form of indistinguishability for an online encryption scheme: indistinguishability from random bits. Fix an online encryption scheme $(\mathcal{E}, \mathcal{IV}, \mathcal{P})$ and consider the following two $\mathcal{E}$-dependent oracles.

- Real(i, M, δ): At the beginning, set $K \xleftarrow{\$} \mathcal{K}$ and $V_i \xleftarrow{\$} \mathcal{IV}$ for all $i \in \mathbb{N}$. Then, on query $(i, P, \delta) \in \mathbb{N} \times \{0,1\}^* \times \{0,1\}$, compute $(C, V_i) \xleftarrow{\$} \mathcal{E}_K^{V_i,\delta}(P)$ and return C.
- Fake(i, P, δ): At the beginning, set $K \xleftarrow{\$} \mathcal{K}$ and $V_i \xleftarrow{\$} \mathcal{IV}$ for all $i \in \mathbb{N}$. Then, on query $(i, P, \delta) \in \mathbb{N} \times \{0,1\}^* \times \{0,1\}$, compute $(C, V_i) \xleftarrow{\$} \mathcal{E}_K^{V_i,\delta}(P)$ and return $|C|$ random bits.

We define $\mathbf{Adv}_{\mathcal{E}}^{\mathrm{IND\$}}(A) = \Pr[A^{\mathrm{Real}} \Rightarrow 1] - \Pr[A^{\mathrm{Fake}} \Rightarrow 1]$. Informally, an online encryption scheme is IND\$-secure if an adversary can't distinguish the ciphertexts it is receiving from random bits.

Discussion. Some of our high-level definitional choices differ for conventional and online encryption schemes. A conventional encryption scheme does not spit out its IV, while an online scheme does. The former is needed to match NIST's definitions for the CBC-CS schemes, but it works less well in the online setting, as here it is important that the algorithm can decide if and when to release the IV. Typically, the IV does get discharged, and as the first part of the ciphertext, so we say that an online encryption scheme $(\mathcal{E}, \mathcal{IV}, \mathcal{P})$ is *IV-prefixed* if $C = \mathcal{E}_K^{IV}(P)$ is always IV followed by some $|P|$ additional bits (assuming $K \in \mathcal{K}$ and $IV \in \mathcal{IV}$). An IV-prefixed online encryption scheme $(\mathcal{E}, \mathcal{IV}, \mathcal{P})$ determines a conventional encryption scheme $\hat{\mathcal{E}}$ in the natural way, setting $\hat{\mathcal{E}}_K^{IV}(P)$ to be $\mathcal{E}_K^{IV}(P)$ stripped of its initial $|IV|$ bits. Conversely, a conventional encryption scheme $\hat{\mathcal{E}}\colon \mathcal{K} \times \mathcal{IV} \times \mathcal{P} \to \mathcal{P}$ is realized by an IV-prefixed online encryptions scheme $(\mathcal{E}, \mathcal{IV}, \mathcal{P})$ if the latter determines the former in the manner just defined. In this way one can speak of an encryption scheme $\mathcal{E}\colon \mathcal{K} \times \mathcal{IV} \times \mathcal{P} \to \mathcal{P}$ as being online; the statement means that it has a secure online realization (the notion of security soon to be defined).

While our notions make sense regardless of whether or not $\mathcal{V}$ is finite, its being finite is the essence of what it means to be online: that one can encrypt (and decrypt) streaming messages without having to buffer more than a constant number of bits. Equivalently, that one can implement an incremental API with a fixed-size context. Our notions allow one to consider things in a more quantitative manner, using $|\mathcal{V}|$ as a measure of worth. We say that $\mathcal{E}\colon \mathcal{K} \times \mathcal{V} \times \{0,1\} \times \{0,1\}^* \to \{0,1\}^* \times \mathcal{V}$ *uses v-bits of state* if v is the smallest number such that $\mathcal{V} \subseteq \{0,1\}^{\le v}$.

Since we concern ourselves only with chosen-plaintext security, we do not formalize syntax or security for the decryption direction of an online encryption scheme. Still, we comment that if an incremental encryption scheme is online then it has an online (that is, finite state-space) decryption.

We regard the initialization vector IV as the initial value of the saved state V. The embedding of the IV space into the state space doesn't prevent a scheme from performing "special" initialization; one can always distinguish the first chunk of a message from subsequent chunks of message by arranging that point in $\mathcal{IV}$ are never returned as a modified state.

An online encryption function has control over if and when the IV is revealed. This can be essential for security: in particular, the Delayed CBC scheme we will soon describe is insecure if the IV is revealed too soon.

Note that the IND\$-definition allows interleaved querying of multiple streams; this is the purpose of the index i. Fouque, Martinet, and Poupard earlier observed that, with respect to their definitions for online indistinguishability, this made for a stronger security notion [11]. The same is true for us; it is easy to see that if the adversary were restricted to asking a sequence of messages with nondecreasing indexes, a restriction that amounts to forbidding the interleaving of encryptions, the resulting security notion would be properly weaker.

We do not find it necessary to demand that, once an oracle query $(i, \cdot, 1)$ is made, there are no subsequent queries $(i, \cdot, \cdot)$. Nonetheless, this is the expected behavior, as the setting of $\delta = 1$ is meant to indicate that the message is complete.

5 Online Security of the CBC-CS Schemes

Delayed CBC. We now present an online version of CBC mode. For the moment, assume all messages have a multiple of b bits. The most obvious approach for defining an online version of CBC is to just spit out ciphertext blocks as they are formed. But this does not work: if an adversary knows C_{i-1} it can choose P_i such that $C_{i-1} \oplus P_i = P_j \oplus C_{j-1}$ for some $j < i$, whence C_i will be C_j if the adversary has a "real" encryption oracle, while this is unlikely if the adversary has a "fake" encryption oracle. We can defend against this attack and, more broadly,

```
**algorithm** $\mathrm{DCBC}_K^{V,\delta}(P)$
**if** $|V| < b$ **then return** error
$C_0\, P_0 \leftarrow V$ **where** $|C_0| = b$
$P \leftarrow P_0\, P;\quad n \leftarrow \lfloor |P|/b \rfloor$
$P_1 \cdots P_n\, P^* \leftarrow P$ **where** $|P_1| = \cdots = |P_n| = b$
**if** $\delta = 1$ **and** $P^* \neq \varepsilon$ **then return** error
**for** $i \leftarrow 1$ **to** $n$ **do** $C_i \leftarrow E_K(P_i \oplus C_{i-1})$
**if** $\delta = 0$ **then** $(C, V') \leftarrow (C_0 \cdots C_{n-1},\ C_n P^*)$
**if** $\delta = 1$ **then** $(C, V') \leftarrow (C_0 \cdots C_n,\ \varepsilon)$
**return** $(C, V')$
```

Fig. 3. Mode DCBC. An online encryption scheme, encryption now depends on the saved state $V \in \{0,1\}^*$. The first b bits of V comprise the *pending ciphertext*, C_0, while the remaining 0 to $b-1$ bits are *unprocessed plaintext*, P_0. Bit δ signals if the plaintext is over.

```
**algorithm** $\mathrm{Enc}(j, P, \delta)$                                   Game $G_0$
**if** $|V_j| < b$ **then return** error                                 $\boxed{\text{Game } G_1}$
$C_0\, P_0 \leftarrow V_j$ **where** $|C_0| = b$
$P \leftarrow P_0\, P;\quad n \leftarrow \lfloor |P|/b \rfloor$
$P_1 \cdots P_n\, P^* \leftarrow P$ **where** $|P_1| = \cdots = |P_n| = b$
**if** $\delta = 1$ **and** $P^* \neq \varepsilon$ **then return** error
**for** $i \leftarrow 1$ **to** $n$ **do**
    $X_i \leftarrow P_i \oplus C_{i-1}$
    $C_i \xleftarrow{\$} \{0,1\}^b$
    **if** $\rho(X_i) \neq$ undefined **then** $bad \leftarrow$ true, $\boxed{C_i \leftarrow \rho(X_i)}$
    **else** $\rho(X_i) \leftarrow C_i$
**if** $\delta = 0$ **then** $(C, V_j') \leftarrow (C_0 \cdots C_{n-1},\ C_n P^*)$
**if** $\delta = 1$ **then** $(C, V_j') \leftarrow (C_0 \cdots C_n,\ \varepsilon)$
**return** $C$
```

Fig. 4. Proof of the IND\$-security of Delayed CBC. Game G_1 includes the boxed statement following the setting of *bad*; game G_0 omits it. The variable *bad* is initialized to false, V_j is initialized to a random b-bit string chosen uniformly at random for each $j \in \mathbb{N}$, and ρ is initialized to everywhere undefined.

get online-secure scheme, simply by *delaying* the last ciphertext block from each plaintext chunk, holding onto it until the relevant blockcipher has already been made. The idea is due to Fouque, Martinet, and Poupard [11]. The contents of this section are a strengthening and extension of that work, adding ciphertext stealing, employing less restrictive syntax, and establishing a stronger notion of security.

The algorithm, detailed in Fig. 3, is called *delayed CBC*, or DCBC. The state consists of two parts: a *pending ciphertext block*, which initially contains a randomly generated IV, and *unprocessed plaintext*, a partial block, possibly empty, carried over from the previous message chunk. If the blockcipher acts on b bits then the state will be at most $v = 2b - 1$ bits. In the pseudocode of Fig. 3, regard $C_i \cdots C_j$ as the empty string if $i > j$.

Informally, the algorithm of Fig. 3 proceeds as follows. The algorithm receives a key K, a state V, an end-of-message indicator δ, and a plaintext chunk P. It parses the state into a b-bit delayed ciphertext block C_0, and a partial plaintext block P_0 with $0 \leq |P_0| < b$. The algorithm then adjusts the incoming plaintext chunk P by prefixing it with P_0. Next it splits P into b-bit blocks $P_1, \ldots, P_n$, leaving a leftover and possibly empty partial block P^*. Since DCBC can only cope with messages that are an integral number of blocks long, the algorithm fails (it reports an error) if $P^* \neq \varepsilon$ when $\delta = 1$. The algorithm next performs the CBC encryption: for $1 \leq i \leq n$, set $C_i = E_K(P_i \oplus C_{i-1})$. Finally, if $\delta = 1$ the algorithm outputs all of $C = C_0 \cdots C_n$, and clears the state. If $\delta = 0$ then it outputs $C = C_0 \cdots C_{n-1}$, holding $V' = C_n \parallel P^*$ in the revised state.

ONLINE SECURITY OF DCBC. We now show that Delayed CBC achieves IND\$ security.

Theorem 2. *Suppose that adversary A asks queries totaling at most σ blocks (each query P contributes $\lceil |P|/b \rceil$ blocks). Then $\mathbf{Adv}^{\mathrm{IND\$}}_{\mathrm{DCBC[Perm}(b)]}(A) \leq \sigma^2/2^b$.*

Proof. Let $r = \mathbf{Adv}^{\mathrm{IND\$}}_{\mathrm{DCBC[Func}(b)]}(A)$. As in the proof of Lemma 1, the PRF/PRP switching gives us that $\left|\mathbf{Adv}^{\mathrm{IND\$}}_{\mathrm{DCBC[Perm}(b)]}(A) - r\right| \leq \sigma^2/2^{b+1}$. It remains to show that $r \leq \sigma^2/2^{b+1}$, for which we use the games in Fig. 4.

The games are constructed so that, with $\mathcal{E} = \mathrm{DCBC[Func}(b)]$, we have $\Pr[A^{\mathrm{Real}(\cdot,\cdot,\cdot)} \Rightarrow 1] = \Pr[A^{G_1(\cdot,\cdot,\cdot)} \Rightarrow 1]$ and $\Pr[A^{\mathrm{Fake}(\cdot,\cdot,\cdot)} \Rightarrow 1] = \Pr[A^{G_0(\cdot,\cdot,\cdot)} \Rightarrow 1]$. Furthermore, games G_0 and G_1 are identical-until-*bad*, and hence we get $r = \left|\mathbf{Adv}[A^{G_1(\cdot,\cdot,\cdot)} \Rightarrow 1] - \mathbf{Adv}[A^{G_0(\cdot,\cdot,\cdot)} \Rightarrow 1]\right| = \Pr[A^{G_0}$ sets *bad*$]$.

In game G_0, all of the C_i values are uniform and independent: all except C_0 in the initial call are generated explicitly by the oracle—and that C_0 is the initial state V_j, chosen uniformly as part of the initialization.

Since the P_i are determined solely by the adversary's inputs, we can think of them as being selected directly by the adversary. We claim that the adversary must choose each P_i before receiving any information about C_{i-1}. For $i > 1$ this is clear, since C_{i-1} is chosen uniformly at random after P_i has been determined. It remains to show that C_0 is uniformly distributed and independent of the adversary's view until P_1 is determined. (Ensuring this property is the reason for delaying the ciphertext block.) We do this inductively, and separately for each index $j \in \mathbb{N}$. The base case is the first encryption query with index j: then $C_0 = V_j$ is the randomly selected initialization vector. Here the adversary can't know anything about its value at this stage since it hasn't been used in any computations at all. The state is empty and we return an immediate error if the previous call's end-of-message indicator was set, so there is no C_0 to concern ourselves with. In the remaining case, the value of C_0 is equal to the value of C_n from the previous encryption query with the same index; the inductive step, therefore, is to show that C_n is uniform and independent of the adversary's view if C_0 is also and $\delta = 0$. But nothing dependent on C_n is part of the oracle's output if $\delta = 0$, and C_n is either freshly generated (if $n > 0$), or equal to C_0 and therefore uniform and independent of the adversary by the induction hypothesis (if $n = 0$).

It immediately follows that each P_i is independent of C_{i-1}, and therefore all of the X_i values are uniform and independent of one another. Hence the probability that two X_i collide—and *bad* is set—is at most $\sigma^2/2^{b+1}$, completing the proof.

As usual, it is easy to pass from the information-theoretic setting to complexity-theoretic one.

Delayed CBC with ciphertext stealing. The algorithms DCBC-CS1, DCBC-CS2, and DCBC-CS3 are defined in Fig. 5. Implicitly, the modes are all parameterized by a blockcipher $E\colon \mathcal{K} \times \{0,1\}^b \to \{0,1\}^b$. The state V once again maintains two portions: the *pending ciphertext* and the *unprocessed plaintext*. The pending ciphertext is a single block—or possibly two blocks in the cases of DCBC-CS3—that the algorithm retains until it is "safe" to spit this out. This is followed by 0 to $b-1$ bits of unprocessed plaintext. The dividing line between the two portions is always clear from the length of the string V. Note that for DCBC-CS3, the state has grown from $2b-1$ bits to $2b$ bits while, for DCBC-CS1 and DCBC-CS2 the state remains at $2b-1$ bits.

The IND\$ security of the DCBC-CS schemes can be inferred from the IND\$ security of the DCBC schemes. This is done in the proof below.

Theorem 3. *Let $\mathcal{E}$ be any of* DCBC-CS1[Perm(b)], DCBC-CS2[Perm(b)], *or* DCBC-CS3[Perm(b)], *and suppose A asks queries totaling at most σ blocks. Then $\mathbf{Adv}^{\mathrm{IND\$}}_{\mathcal{E}}(A) \leq \sigma^2/2^b$.*

```
40     algorithm DCBC-CS$_K^{V,\,\delta}(P)$
41     if $|V| < b$ then return error
42     $C_{-1}\,C_0\;\,P_0 \leftarrow V$ where $|C_{-1}| \in \{0,b\}$,  $|C_0| = b$,  $|P_0| < b$
43     $P \leftarrow P_0\,P$
44     if $\delta = 0$ then $P_1 \cdots P_n\;P^* \leftarrow P$ where $n \leftarrow \lfloor |P|/b \rfloor$, $|P_1| = \cdots = |P_n| = b$
45     else $P_1 \cdots P_n \leftarrow P\,0^{b-d}$ where $n \leftarrow \lceil |P|/b \rceil$, $d \leftarrow b+|P|-nb$, $|P_1| = \cdots = |P_n| = b$
46     for $i \leftarrow 1$ to $n$ do $C_i \leftarrow E_K(P_i \oplus C_{i-1})$
47     if $\delta = 0$ then
48-1       $(C, V') \leftarrow (C_0 \cdots C_{n-1},\;\; C_n P^*)$                              $\Longleftarrow$ for CS1
48-2       $(C, V') \leftarrow (C_0 \cdots C_{n-1},\;\; C_n P^*)$                              $\Longleftarrow$ for CS2
48-3       $(C, V') \leftarrow (P^* = \varepsilon)$? $(C_{-1}C_0 \cdots C_{n-2},\; C_{n-1}C_n)$ :        $\Longleftarrow$ for CS3
                                    $(C_{-1}C_0 \cdots C_{n-1},\; C_n\,P^*)$
49     if $\delta = 1$ then
50         if $n > 0$ then $C_{n-1} \leftarrow \mathrm{MSB}_d(C_{n-1})$
51-1       $(C, V') \leftarrow (C_0 \cdots C_{n-2}C_{n-1}C_n,\; \varepsilon)$                       $\Longleftarrow$ for CS1
51-2       $(C, V') \leftarrow (d = b)$? $(C_0 \cdots C_{n-2}C_{n-1}C_n,\; \varepsilon)$ :            $\Longleftarrow$ for CS2
                                   $(C_0 \cdots C_{n-2}C_nC_{n-1},\; \varepsilon)$
51-3       $(C, V') \leftarrow (C_{-1}C_0 \cdots C_{n-2}C_nC_{n-1},\; \varepsilon)$                  $\Longleftarrow$ for CS3
52     return $(C, V')$
```

Fig. 5. Delayed CBC with ciphertext stealing: DCBC-CS. Each online scheme depends on $E\colon \mathcal{K} \times \{0,1\}^b \to \{0,1\}^b$. String $C_{-1}C_0$ is pending ciphertext (with C_{-1} used only for DCBC-CS3). String P_0 is unprocessed plaintext from the prior call.

```
400    algorithm $\mathcal{DCBC}\text{-}\mathcal{CS}_K^{V,\,\delta}(P)$
401    if $|V| \le b$ then $(W, C_{-1}, \ell) \leftarrow (V, \varepsilon, 0)$ else $[W, C_{-1}, \ell] \leftarrow V$
402    $m \leftarrow \ell + |P|$,  $d \leftarrow m - b\lfloor (m-1)/b \rfloor$
403    if $\delta = 1$ then $P \leftarrow P\,0^{b-d}$
404    $(C, W') \leftarrow \mathrm{DCBC}_K^{W,\,\delta}(P)$
405    $C_0\;\cdots\;C_n \leftarrow C$ where $n \leftarrow |C|/b - 1$  and  $|C_0| = \cdots = |C_n| = b$
406    if $\delta = 0$ then
407-1       $(C', C'_{-1}) \leftarrow (C_0 \cdots C_n, \varepsilon)$                                  $\Longleftarrow$ for CS1
407-2       $(C', C'_{-1}) \leftarrow (C_0 \cdots C_n, \varepsilon)$                                  $\Longleftarrow$ for CS2
407-3       $(C', C'_{-1}) \leftarrow (d = b)$? $(C_0 \cdots C_{n-1}, C_n)$ : $(C_0 \cdots C_n, \varepsilon)$      $\Longleftarrow$ for CS3
408        $\ell' \leftarrow (d = b)$? $0$ : $d$
409    if $\delta = 1$ then
410        if $n > 0$ then $C_{n-1} \leftarrow \mathrm{MSB}_d(C_{n-1})$
411-1      $C' \leftarrow C_0 \cdots C_{n-2}C_{n-1}C_n$                                            $\Longleftarrow$ for CS1
411-2      $C' \leftarrow (d = b)$? $C_0 \cdots C_{n-2}C_{n-1}C_n$ : $C_0 \cdots C_{n-2}C_nC_{n-1}$          $\Longleftarrow$ for CS2
411-3      $C' \leftarrow C_0 \cdots C_{n-2}C_nC_{n-1}$                                            $\Longleftarrow$ for CS3
412        $\ell' \leftarrow 0$,  $C'_{-1} \leftarrow \varepsilon$
413    return $(C, [W', C'_{-1}, \ell'])$
```

Fig. 6. Defining DCBC-CS in terms of DCBC. The notation $[x_1, \ldots, x_n]$ denotes an unambiguous non-compressing encoding of the items $x_1, \ldots, x_n$; used on the left-hand side of an assignment, it implies a decoding operation.

Proof. We use a different description of DCBC-CS, shown in Fig. 6, now writing the algorithm in terms of DCBC. The state vector consists of three components: a state W for DCBC, which is not interpreted; an additional delayed ciphertext block C_{-1}, which corresponds to C_{-1} in Fig. 5; and a length $0 \le \ell < b$, which keeps track of the amount of unprocessed plaintext maintained in W, so that $\ell = |P_0|$.

The theorem will follow from three observations about this new description of DCBC. First, $\mathcal{DCBC}\text{-}\mathcal{CS}$ is functionally identical to DCBC-CS. Second, if the call to function DCBC at line 404 were to instead call a function that returned a random strings of the appropriate length, then so too would $\mathcal{DCBC}\text{-}\mathcal{CS}$. This observation is immediate, since the strings returned DCBC-CS$'$ are derived from those returned by $\mathrm{DCBC}_K^{V,\,\delta}$ by discarding and reordering particular fixed bits.

Third, $\mathcal{DCBC\text{-}CS}$ can be implemented using only oracle access to the DCBC function: it doesn't need to inspect or interpret the DCBC state vector W, nor examine the key K, and it uses the state only in the "single-threaded" way permitted by the online IND\$ oracle.

Consequently, for any adversary A attacking DCBC-CS, we can construct an adversary B attacking DCBC: B will run A against a simulated oracle built from B's (real or fake) DCBC oracle using $\mathcal{DCBC\text{-}CS}$ and, in the end, output A's guess as its own. We have

$$\begin{aligned}
\mathbf{Adv}^{\text{IND\$}}_{\text{DCBC-CS}}(A) &= \Pr[A^{\text{Real}} \Rightarrow 1] - \Pr[A^{\text{Fake}} \Rightarrow 1] \\
&= \Pr[A^{\mathcal{DCBC\text{-}CS}[\text{Real}]} \Rightarrow 1] - \Pr[A^{\mathcal{DCBC\text{-}CS}[\text{Fake}]} \Rightarrow 1] \\
&= \Pr[B^{\text{Real}} \Rightarrow 1] - \Pr[B^{\text{Fake}} \Rightarrow 1] \\
&= \mathbf{Adv}^{\text{IND\$}}_{\mathcal{E}}(B) \le \sigma^2/2^b
\end{aligned}$$

appealing to Theorem 2 for the final inequality.

As before, one can immediately conclude the corresponding complexity-theoretic statement, which would read as follows.

Corollary 2. *Let* $E\colon \mathcal{K} \times \{0,1\}^b \to \{0,1\}^b$ *be a blockcipher and let* $\mathcal{E}$ *be any of the encryption schemes* DCBC-CS1$[E]$, DCBC-CS2$[E]$, *or* DCBC-CS3$[E]$. *Suppose* A *asks queries that total* σ *blocks, runs in time* t, *and achieves advantage* $\delta = \mathbf{Adv}^{\text{IND\$}}_{\mathcal{E}}(A)$. *Then there is an adversary* B, *explicitly known and constructed from* A *in a blackbox manner, that asks at most* σ *queries, runs in time* $t + \lambda b\sigma$, *and achieves advantage* $\mathbf{Adv}^{\text{prp}}_{E}(B) \ge \delta - \sigma^2/2^b$. *Here* λ *is an absolute constant depending only on details of the model of computation.*

6 Insecurity of the Meyer-Matyas CBC-CS

The CBC ciphertext-stealing construction by Meyer and Matyas, what we will call CBC-CSX, is defined in Fig. 7. This well-known scheme—it has been used since the early 1980's under the IBM CUSP architecture—is susceptible to a simple chosen-plaintext attack, a fact that appears not to have been pointed out before. Thus NIST did well in choosing not to standardize this form of ciphertext stealing, but the alternative, "correct" variant.

Here is an attack on the ind\$-security of CBC-CSX. The adversary makes two encryption queries: $M = 1^b 0^{b-1}$ and $M' = 1^b 0^{b-1}$. As the IV is randomized, asking the same plaintext twice is not without purpose. The oracle returns $C = C_0C_1C_2$ and $C' = C'_0C'_1C'_2$ where C_0 and C'_0 are randomly chosen *IV*s and $|C_1| = |C'_1| = b - 1$. If $C_2 = C'_2$ the adversary returns 1; otherwise, it returns 0. Now if the adversary is given a CBC-CSX oracle, the probability that $C_2 = C'_2$ is at least 1/2; otherwise, it's about $1/2^b$. Thus we have a trivial but effective ind\$-attack.

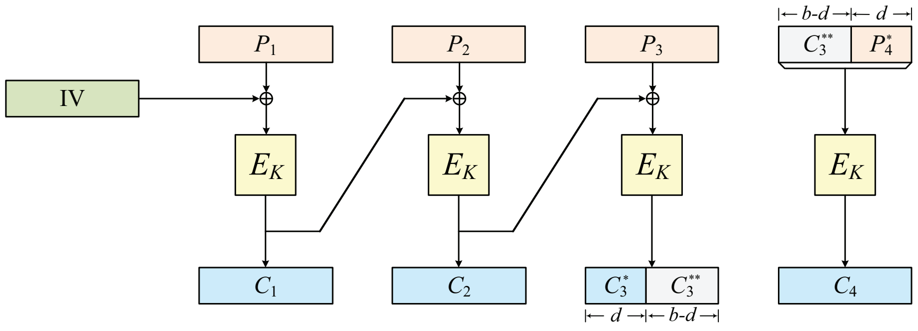
We remark that, not surprisingly, CBC-CSX is not secure under conventional, weaker notions of security, like left-or-right indistinguishability [3]; a similar attack can easily be described. It is not that the definition is too strong; from a modern point of view, the scheme is simply wrong.

Acknowledgments

Authors Rogaway and Zhang received support for this project under NSF grant CNS 0904380. Many thanks to the NSF for their support.

References

1. M. Ball. Follow-up to NIST's consideration of XTS-AES as standardized by IEEE Std 1619-2007. http://tinyurl.com/nist-ball-xts, Public comments to NIST, 2008.



```
90  algorithm CBC-CSX$_K^{IV}(P)$
91  $n \leftarrow \lceil |P|/b \rceil$
92  $P_1 \cdots P_{n-1} P_n^* \leftarrow P$ where $|P_1| = \cdots = |P_{n-1}| = b$ and $|P_n^*| = d$
94  $C_0 \leftarrow IV$
95  for $i \leftarrow 1$ to $n-1$ do $C_i \leftarrow E_K(P_i \oplus C_{i-1})$
96  $C_n \leftarrow E_K((\mathrm{LSB}_{b-d}(C_{n-1}) \,\|\, P_n^*)$
97  $C_{n-1}^* \leftarrow \mathrm{MSB}_d(C_{n-1})$
98  return $C_0\, C_1\, \cdots\, C_{n-2}\, C_{n-1}^*\, C_n$
```

Fig. 7. Mode CBC-CSX. The mode is insecure and should not be used. This version of ciphertext stealing is from Meyer and Matyas [14]. The mode depends on a blockcipher $E\colon \mathcal{K} \times \{0,1\}^b \to \{0,1\}^b$. That can be ideal, and the IV random, and still the mode will fail to achieve standard (CPA) privacy definitions.

2. G. Bard. Blockwise-adaptive chosen-plaintext attack and online modes of encryption. *Cryptography and Coding 2007*, LNCS 4887, Springer, pp. 129–151, 2007.
3. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption: analysis of the DES modes of operation. *FOCS 97*, IEEE Press, pp. 394–403, 1997.
4. M. Bellare, T. Kohno and C. Namprempre. Breaking and provably repairing the SSH authenticated encryption scheme: a case study of the encode-then-encrypt-and-MAC paradigm. *ACM Transactions on Information and System Security* (TISSEC), 7:2, pp. 206–241, 2004. Earlier version from *CCS 2002*.
5. M. Bellare and P. Rogaway. Code-based game-playing proofs and the security of triple encryption. *EUROCRYPT 2006*, LNCS vol. 4004, Springer, pp. 409–426, 2006.
6. A. Boldyreva and N. Taesombut. Online encryption schemes: new security notions and constructions. *CT-RSA 2004*, LNCS vol. 2964, Springer, pp. 1–14, 2004.
7. M. Dworkin. Recommendation for block cipher modes of operation: method and techniques. NIST Special Publication 800-38A, 2001 Edition. December 2001.
8. M. Dworkin. Recommendation for block cipher modes of operation: three variants of ciphertext stealing for CBC mode. Addendum to NIST Special Publication 800–38A. October 2010.
9. P. Fouque, A. Joux, G. Martinet, and F. Valette. Authenticated on-line encryption. *SAC 2003*, LNCS vol. 3006, Springer, pp. 145–159, 2003.
10. P. Fouque, A. Joux and G. Poupard. Blockwise adversarial model for on-line ciphers and symmetric encryption schemes. *SAC 2004*, LNCS vol. 3357, Springer, pp. 212–226, 2004.
11. P. Fouque, G. Martinet, G. Poupard. Practical symmetric on-line encryption. *FSE 2003*, LNCS vol. 2887, Springer, pp. 362–375, 2003.
12. R. Gennaro and P. Rohatgi. How to sign digital streams. *CRYPTO 97*, LNCS vol. 1294, Springer, pp. 180–197, 1997.
13. A. Joux, G. Martinet, and F. Valette. Blockwise-adaptive attackers: revisiting the (in)security of some provably secure encryption models: CBC, GEM, IACBC. *CRYPTO 2002*, LNCS vol. 2442, Springer, pp. 17–30, 2002.
14. C. Meyer and M. Matyas. Cryptography: a new dimension in data security. *John Wiley & Sons*, New York, 1982.
15. NIST. Proposal to extend CBC mode by “ciphertext stealing.” Anonymous draft, May 6, 2007. Available from NIST’s website.
16. P. Rogaway, M. Bellare, and J. Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *ACM Transactions on Information and System Security*, 6:3, pp. 365–403, 2003. Earlier version, with T. Krovetz, in *ACM CCS 01*.

17. B. Schneier. *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C*. New York, Wiley, 1996.
18. V. Shoup. Sequences of games: a tool for taming complexity. ePrint archive 2004/332. Revised 2006.
19. S. Vaudenay. Security flaws induced by CBC padding — applications to SSL, IPSEC, WTLS .... *EUROCRYPT 2002*, LNCS vol. 2332, Springer, pp. 534–545, 2002.