Online Ciphers from Tweakable Blockciphers

Phillip Rogaway and Haibin Zhang

Dept. of Computer Science, University of California, Davis, California 95616, USA
{rogaway,hbzhang}@cs.ucdavis.edu

Abstract. Online ciphers are deterministic length-preserving permutations $\mathcal{E}_K$: $(\{0,1\}^n)^+ \to (\{0,1\}^n)^+$ where the i-th block of ciphertext depends only on the first i blocks of plaintext. Definitions, constructions, and applications for these objects were first given by Bellare, Boldyreva, Knudsen, and Namprempre. We simplify and generalize their work, showing that online ciphers are rather trivially constructed from tweakable blockciphers, a notion of Liskov, Rivest, and Wagner. We go on to show how to define and achieve online ciphers for settings in which messages need *not* be a multiple of n bits.

Key words: Online ciphers, modes of operation, provable security, symmetric encryption, tweakable blockciphers.

1 Introduction

Background. Informally, a cryptographic transform is said to be *online* if it can be computed by an algorithm that reads in the (unknown number of) input bits—in order, one at a time—as it writes out the corresponding output bits—again in order, one at a time—never using more than a constant amount of memory or incurring more than a constant amount of latency.[1] Most blockcipher modes of operation *are* online—for example, modes like CBC, HMAC, and GCM certainly are. But one kind of transformation is not online, and can never be online: a *general cipher* [20], one secure in the customary sense of a PRP (a pseudorandom permutation). Such objects take a key K and a plaintext M of unbounded length and produce a ciphertext C of length $|M|$, doing so in such a way that the mapping resembles a random permutation. EME2 is a soon-to-be-standardized example [14]. The reason an online cipher can't be PRP-secure is simple: the first bit of output, for example, has got to depend on every bit of input, else it is trivial to distinguish the cipher from a random permutation. This requirement makes bounded memory, or latency, an impossibility.

One can weaken PRP security to capture what *is* possible in the online setting. Bellare, Boldyreva, Knudsen, and Namprempre (BBKN) were the first to do so, defining *online ciphers* [3]. The authors fix a parameter n (likely the blocksize of some underlying blockcipher). They then demand that the i-th (n-bit) block of ciphertext depend only on the first i blocks of plaintext (and, of

[1] For online ciphers, two alternative formulations—one corresponding to the opening sentence of the abstract, plus one other—will subsequently be described.

course, the key).[2] BBKN did not explicitly demand that encryption and decryption be computable with constant memory and latency, but follow-on work by Boldyreva and Taesombut strengthened BBKN's definition in a way that ensures this is so [9].

Like other kinds of ciphers, online ciphers can be secure in either the CPA (chosen plaintext) or CCA (chosen ciphertext, or "strong") sense, depending on whether the adversary is given oracle access to the decryption (or "backwards") functionality as well as the encryption (or "forwards") functionality. For online ciphers it initially seemed as though CCA-security was harder to achieve than CPA-security [3], but, at present, the most efficient CCA-secure construction has essentially the same overhead as the most efficient CPA-secure one, needing just one extra xor per block [22].

Online ciphers are useful tools. For example, BBKN demonstrate a simple recipe for turning a CCA-secure online cipher into an authenticated-encryption scheme (one prepends and appends a random value $R \in \{0,1\}^n$) [4]; Boldyreva and Taesombut (following, Fouque Joux, Martinet, and Valette [10]) show how to turn a CCA-secure online cipher into either an online encryption scheme secure against "blockwise-adaptive" CCA attacks, or else an online authenticated-encryption (AE) scheme likewise secure against BA-CCA attacks [9]; and Amanatidis, Boldyreva, and O'Neill describe the use of online ciphers to solve a database-security problem [1].

Our contribution. In this paper we make two contributions. First, we recast the constructions of BBKN [3, 4], plus a subsequent construction by Nandi [21, 22], into the language of *tweakable blockciphers*, a notion of Liskov, Rivest, and Wagner [17]. The new starting point yields constructions more general and transparent than those before. See Fig. 1. Second, we show how to relax the notion of an online cipher to deal with messages that are *not* a multiple of n bits. Besides definitions, we provide a simple and efficient construction to handle this setting. Dealing with arbitrary-length inputs is a necessary precursor to practical schemes, which we also describe.

Discussion. The original BBKN paper had fairly complex schemes and proofs [3]. Nandi found some bugs in these proofs and offered up his own [21, 22]. BBKN corrected the issues in their proofs, which they regarded as minor, but the proofs remain complex [4]. BBKN's modes relied on xor-universal hash functions, and subsequent work did too, or else doubled the number of blockcipher calls [3, 4, 21, 22]. Our own constructions are simple, and they are natural generalizations of the existing schemes. The proofs are simple too. We do not regard this simplicity as a defect. Without the tweakable-blockcipher abstraction, constructions and proofs in this domain are *not* simple, as the above history suggests.

The question of fractional final blocks was earlier asked by Nandi [22, p. 361]. Note that one cannot just say to pad to the next multiple of n (as suggested, for

[2] It follows that the i-th block of plaintext will likewise depend only on the key and the first i blocks of ciphertext.

TC1

algorithm $\mathcal{E}_K(M)$
if $M \notin (\{0,1\}^n)^+$ **return** $\perp$
$m \leftarrow |M|/n$; $T \leftarrow 0^n$
for $j \leftarrow 1$ **to** m **do**
  $C[j] \leftarrow \widetilde{E}_K^T(M[j])$
  $T \leftarrow C[j]$
return C

TC2

algorithm $\mathcal{E}_K(M)$
if $M \notin (\{0,1\}^n)^+$ **return** $\perp$
$m \leftarrow |M|/n$; $T \leftarrow 0^{2n}$
for $j \leftarrow 1$ **to** m **do**
  $C[j] \leftarrow \widetilde{E}_K^T(M[j])$
  $T \leftarrow M[j] \,\|\, C[j]$
return C

TC3

algorithm $\mathcal{E}_K(M)$
if $M \notin (\{0,1\}^n)^+$ **return** $\perp$
$m \leftarrow |M|/n$; $T \leftarrow 0^n$
for $j \leftarrow 1$ **to** m **do**
  $C[j] \leftarrow \widetilde{E}_K^T(M[j])$
  $T \leftarrow M[j] \oplus C[j]$
return C

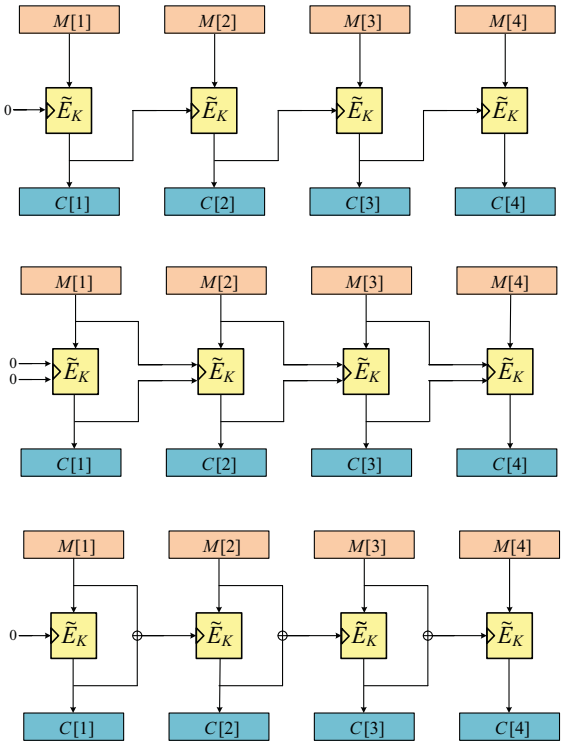

Fig. 1. Modes TC1, TC2, and TC3. The first is a CPA-secure online cipher; the next two are CCA-secure. Plaintexts must have a length divisible by n. In the diagrams, the object $\widetilde{E}_K$ is a keyed tweakable blockcipher; the tweak comes in at left, the n-bit input comes in at the top, and the n-bit output emerges from the bottom. TC1, TC2, and TC3 simplify and generalize HCBC1 [3, 4], HCBC2 [4], and MHCBC [21, 22], respectively.

example, by Bard [2, p. 134]). This doesn't make sense definitionally, because it leaves one without any notion for what it *means* to have an online-encipher outside of $(\{0,1\}^n)^+$, and it doesn't make sense procedurally, because, if you *did* pad and then encipher, you would no longer have a cipher at all (ciphers preserve length).

Dealing with arbitrary-length inputs is important for efficiency: if we are going to turn an online cipher into an authenticated-encryption scheme or a blockwise-adaptive (BA) online encryption scheme, there will be ciphertext expansion if is forced to pad.

Correctly constructing arbitrary-input-length online ciphers would be difficult without the tweakable blockcipher abstraction. Liskov, Rivest, and Wagner had argued that tweakable blockciphers would be useful tools for designing symmetric protocols [17]. Our results bolster this point of view.

Joux, Martinet, and Valette [15] implicitly argue that our notion of security for online ciphers is not strong enough, since it treats the encryption operation as "atomic," not attending to attacks that, for example, select the second block of a plaintext being encrypted based on the encryption of the first. We do not dispute this insight, but nonetheless prefer not to deal with these blockwise-adaptive adversarial attacks. First, enriching the security notion to allow for them hardly changes things in the CPA-setting [11, Theorem 8]. Second, "atomic" online ciphers are already useful for higher-level applications, as described above. Finally, "true" CCA security becomes impossible in the BA setting, leading to more subtle definitions for what actually is achievable [9].

2 Preliminaries

NOTATION. A *string* is a member of $\{0,1\}^*$. The notation $A \| B$, or just AB, denotes the concatenation of strings A and B. If X is a string then $|X|$ denotes its length. The empty string is denoted ε. Throughout this paper we fix a positive number n called the *blocksize.* The set $(\{0,1\}^n)^+$ is the set of all strings having length jn for some $j \geq 1$. If $X \in (\{0,1\}^n)^+$ we let $X[i]$ denotes its ith n-bit block, so $X = X[1] \cdots X[m]$ where $m = |X|/n$. We will later extend this notation to the case when X is not a multiple of n bits. We write $X[i..j]$ for $X[i] \cdots X[j]$.

CIPHERS. A map $f\colon \mathcal{X} \to \mathcal{X}$ for $\mathcal{X} \subseteq \{0,1\}^*$ is a *length-preserving function* if $|f(x)| = |x|$ for all $x \in \{0,1\}^*$. It is a length-preserving *permutation* if it is also a permutation. A *cipher* is a map $\widetilde{E}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ where $\mathcal{K}$ is a nonempty set (finite or otherwise endowed with some distribution), $\mathcal{M} \subseteq \{0,1\}^*$ is a nonempty set, and $\mathcal{E}_K = \mathcal{E}(K,\cdot)$ is a length-preserving permutation for all $K \in \mathcal{K}$. The set $\mathcal{K}$ is called the *key space* and $\mathcal{M}$ is called the *message space.* If $\mathcal{E}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ is a cipher then its *inverse* is the cipher $\mathcal{E}^{-1}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ defined by $\mathcal{E}^{-1}(K,Y) = \mathcal{E}_K^{-1}(Y)$ being the unique point X such that $\mathcal{E}_K(X) = Y$.

BLOCKCIPHERS AND TWEAKABLE BLOCKCIPHERS. A *blockcipher* is a function $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ where $\mathcal{K}$ is a finite nonempty set and $E_K(\cdot) = E(K,\cdot)$ is a permutation on $\{0,1\}^n$ for every $K \in \mathcal{K}$. Equivalently, a blockcipher is a cipher with message space $\mathcal{M} = \{0,1\}^n$. A *tweakable* blockcipher is a function $\widetilde{E}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ where $\mathcal{K}$ is a finite nonempty set and $\mathcal{T}$ is a nonempty set (the *tweak space*) and $\widetilde{E}_K^T(\cdot) = \widetilde{E}(K,T,\cdot)$ is a permutation on $\{0,1\}^n$ for every $K \in \mathcal{K}$, $T \in \mathcal{T}$.

Let $\mathrm{Perm}(n)$ be the set of all permutations on n bits, $\mathrm{Perm}(\mathcal{M})$ be the set of all length-preserving permutations on the finite set $\mathcal{M} \subseteq \{0,1\}^*$, and $\mathrm{Perm}(\mathcal{T},n)$ the set of all functions $\pi\colon \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ where $\pi_T(\cdot) = \pi(T,\cdot)$ is a permutation for each $T \in \mathcal{T}$. We may regard $\mathrm{Perm}(n)$, $\mathrm{Perm}(\mathcal{M})$, and $\mathrm{Perm}(\mathcal{T},n)$ as blockciphers, ciphers, and tweakable blockciphers, respectively; they are the ideal blockcipher on n bits, the ideal cipher on $\mathcal{M}$, and the ideal tweakable blockcipher on n bits and tweak space $\mathcal{T}$. When an adversary $\mathcal{A}$ is run with an oracle $\mathcal{O}$ we let $\mathcal{A}^{\mathcal{O}} \Rightarrow 1$ denote the event that $\mathcal{A}$ outputs 1. Define

the prp, ±prp, $\widetilde{\text{prp}}$, and ±$\widetilde{\text{prp}}$ *advantage* of $\mathcal{A}$ against E or $\widetilde{E}$ by:

$$\begin{aligned}
\mathbf{Adv}^{\text{prp}}_{E}(\mathcal{A}) &= \mathbf{Pr}[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{E_K} \Rightarrow 1] - \mathbf{Pr}[\pi \xleftarrow{\$} \text{Perm}(n) : \mathcal{A}^{\pi} \Rightarrow 1] \\
\mathbf{Adv}^{\pm\text{prp}}_{E}(\mathcal{A}) &= \mathbf{Pr}[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{E_K, E_K^{-1}} \Rightarrow 1] - \mathbf{Pr}[\pi \xleftarrow{\$} \text{Perm}(n) : \mathcal{A}^{\pi, \pi^{-1}} \Rightarrow 1] \\
\mathbf{Adv}^{\widetilde{\text{prp}}}_{\widetilde{E}}(\mathcal{A}) &= \mathbf{Pr}[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\widetilde{E}_K} \Rightarrow 1] - \mathbf{Pr}[\pi \xleftarrow{\$} \text{Perm}(\mathcal{T}, n) : \mathcal{A}^{\pi} \Rightarrow 1] \\
\mathbf{Adv}^{\pm\widetilde{\text{prp}}}_{\widetilde{E}}(\mathcal{A}) &= \mathbf{Pr}[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\widetilde{E}_K, \widetilde{E}_K^{-1}} \Rightarrow 1] - \mathbf{Pr}[\pi \xleftarrow{\$} \text{Perm}(\mathcal{T}, n) : \mathcal{A}^{\pi, \pi^{-1}} \Rightarrow 1]
\end{aligned}$$

Online ciphers. A length-preserving function $f\colon (\{0,1\}^n)^+ \to (\{0,1\}^n)^+$ is *online* if, for all i, $f(X)[1..i]$ depends only on $X[1..i]$. Here we say that $f(X)[1..i]$ depends only on $X[1..i]$ if $f(XY)[1..i] = f(XY')[1..i]$ for all $X \in \{0,1\}^{in}$ and $Y, Y' \in \{0,1\}^*$ where $f(XY)$ and $f(XY')$ are defined. A cipher $\mathcal{E}\colon \mathcal{K} \times (\{0,1\}^n)^+ \to (\{0,1\}^n)^+$ is online if each $\mathcal{E}_K$ is. Let Online$(n)\colon \mathcal{K} \times (\{0,1\}^n)^+ \to (\{0,1\}^n)^+$ be the *ideal* online cipher on n bits: each key names one of the possible online ciphers, the set being given the uniform distribution in the natural way. If $\mathcal{E}\colon \mathcal{K} \times (\{0,1\}^n)^+ \to (\{0,1\}^n)^+$ is an online cipher and $\mathcal{A}$ is an adversary we define:

$$\begin{aligned}
\mathbf{Adv}^{\text{oprp}}_{\mathcal{E}}(\mathcal{A}) &= \mathbf{Pr}[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K} \Rightarrow 1] - \mathbf{Pr}[\pi \xleftarrow{\$} \text{Online}(n) : \mathcal{A}^{\pi} \Rightarrow 1] \\
\mathbf{Adv}^{\pm\text{oprp}}_{\mathcal{E}}(\mathcal{A}) &= \mathbf{Pr}[K \xleftarrow{\$} \mathcal{K} : \mathcal{A}^{\mathcal{E}_K, \mathcal{E}_K^{-1}} \Rightarrow 1] - \mathbf{Pr}[\pi \xleftarrow{\$} \text{Online}(n) : \mathcal{A}^{\pi, \pi^{-1}} \Rightarrow 1]
\end{aligned}$$

We comment that the definitions allow variable-input-length (VIL) attacks: the adversary may ask queries of varying lengths. On the other hand, definitions only countenance ciphers on $\mathcal{M} = (\{0,1\}^n)^+$, and it is not obvious what to do beyond this domain, Online(n) being quite specific to it.

Discussion. The notion for an online cipher just given, taken from BBKN [3], can be criticized for not prohibiting, for example, that that computation of $C[m]$ requires one to retain all of $M[1] \cdots M[m]$. A stronger notion appears in Boldyreva and Taesombut [9], the definition asserting that $C[i]$ may only depend on $M[i]$, $M[i-1]$, $C[i-1]$, and the underlying key. We believe that this requirement does not make for a desirable security definition: the cipher in which each $C[i]$ is a random permutation of $M[i]$, tweaked by $M[i-1] \,\|\, C[i-1]$, ought not to be regarded ideal, since one can easily do better and still, intuitively, be "online." Still, our *constructions* enjoy the BT-style locality property, ensuring that they can be implemented with constant latency and memory.

An alternative notion for an online cipher would capture the intuition from the opening paragraph of this paper, saying that a cipher $\mathcal{E}\colon \mathcal{K} \times \mathcal{M} \to \mathcal{M}$ is Online$[m]$ if it can be implemented by an algorithm that is fed in bits one at a time, and that retains just m bits of state. This would natively handle ciphers on arbitrary bit strings. We leave it as an open question to explore these ideas.

3 Online Ciphers Achieving CPA-Security

Let $\widetilde{E}\colon \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable blockcipher. From this primitive we define a cipher $\mathcal{E} = \text{TC1}[\widetilde{E}]$ with key space $\mathcal{K}$ and message space

$\mathcal{M} = (\{0,1\}^n)^+$. See Fig. 1. The construction is online CPA-secure, as formalized below.

Theorem 1 (TC1 is oprp-secure). *Let* $\widetilde{\pi} = \mathrm{Perm}(\{0,1\}^n, n)$. *If* $\mathcal{A}$ *asks queries having at most* σ *blocks then* $\mathbf{Adv}^{\mathrm{oprp}}_{\mathrm{TC1}[\widetilde{\pi}]}(\mathcal{A}) \leq 1.5\,\sigma^2/2^n$. ∎

We omit the proof because we will in a moment be proving, by analogous but slightly more involved means, what is essentially a stronger result: online CCA-security for the equally efficient cipher TC3. The complexity-theoretic analog for the theorem, using a "real" tweakable PRP $\widetilde{E}$ instead of the ideal tweakable PRP $\widetilde{\pi}$, follows by standard techniques. One would need $\widetilde{E}$ to be secure in the prp-sense. We omit the theorem statement, showing later how it would look for scheme TC3.

Mechanism TC1 is a generalization of BBKN's mode of operation HCBC1 [4] (formerly named HCBC [3]); the latter can be realized as a special case of TC1 by selecting the tweakable blockcipher $\widetilde{E}\colon (\mathcal{K}_1 \times \mathcal{K}_2) \times \{0,1\}^n \to \{0,1\}^n$ to be $\widetilde{E}^T_{K1\ K2}(X) = E_{K1}(M \oplus H_{K2}(T))$ where $H\colon \mathcal{K}_2 \times \{0,1\}^n \to \{0,1\}^n$ is an almost-xor universal hash function and E is a blockcipher.[3] This is in fact the "standard" construction of a tweakable blockcipher from an ordinary one [17]. Of course one can instantiate the tweakable blockcipher $\widetilde{E}$ from an ordinary blockcipher E in a variety of other ways as well. We comment that TC1 can also be regarded as Liskov, Rivest, and Wagner's "tweak block chaining" mode [17, Section 4] but with a zero IV.

Note that TC1 is not a secure online cipher with respect to CCA attacks. A simple attack is as follows. The adversary makes a decryption query of $C \,\|\, C \,\|\, C$ for any $C \in \{0,1\}^n$. The oracle returns $M_1 \,\|\, M_2 \,\|\, M_3$ as the reply. If $M_2 = M_3$, return 1; otherwise, return 0. Under the TC1 construction, one will always have that $M_2 = M_3$, but with a random on-line cipher this will rarely be true.

4 Online Ciphers Achieving CCA-Security

Let $\widetilde{E}\colon \mathcal{K} \times \{0,1\}^{2n} \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable blockcipher. From this primitive we define a cipher $\mathcal{E} = \mathrm{TC2}[\widetilde{E}]$ with key space $\mathcal{K}$ and message space $\mathcal{M} = (\{0,1\}^n)^+$. Again see Fig. 1. The construction is CCA-secure, as formalized below.

Theorem 2 (TC2 is ±oprp-secure). *Let* $\widetilde{\pi} = \mathrm{Perm}(\{0,1\}^{2n}, n)$. *If* $\mathcal{A}$ *asks queries having at most* σ *blocks then* $\mathbf{Adv}^{\pm\mathrm{oprp}}_{\mathrm{TC2}[\widetilde{\pi}]}(\mathcal{A}) \leq 1.5\,\sigma^2/2^n$. ∎

We again omit the proof, and the complexity-theoretic analog, which would this time need the $\pm\widetilde{\mathrm{prp}}$ assumption, preferring, for concision, to do this just for TC2's more efficient cousin, TC3.

[3] We recall the definition, due to Krawczyk [16], that $H\colon \mathcal{K}2 \times \mathcal{X} \to \{0,1\}^n$ is ϵ-almost XOR universal (ϵ-AXU) if for all distinct $X, X' \in \mathcal{X}$ and all $C \in \{0,1\}^n$ we have that $\Pr[H_K(X) \oplus H_K(X') = C] \leq \epsilon$, the probability over $K \xleftarrow{\$} \mathcal{K}2$. Simple constructions achieve $\epsilon = 2^{-n}$, the minimum value possible.

Mechanism TC2 is a generalization of BBKN's mode HCBC2 [4] (formerly named HPCBC), which can be regarded as TC2 with a tweakable blockcipher $\widetilde{E}\colon (\mathcal{K}_1 \times \mathcal{K}_2) \times \{0,1\}^n \to \{0,1\}^n$ of $\widetilde{E}^T_{K1\ K2}(X) = E_{K1}(M \oplus H_{K2}(T)) \oplus H_{K2}(T)$ where $H\colon \mathcal{K}_2 \times \{0,1\}^{2n} \to \{0,1\}^n$ is an almost-xor universal hash function and E is a blockcipher. This is also the "standard" construction of a strong tweakable blockcipher from an ordinary one [17].

We are now ready to consider TC3. Let $\widetilde{E}\colon \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a tweakable blockcipher. From this primitive define the online cipher $\mathcal{E} = \mathrm{TC3}[\widetilde{E}]$ with key space $\mathcal{K}$ and message space $\mathcal{M} = (\{0,1\}^n)^+$. Again see Fig. 1. The construction is CCA-secure, as formalized below.

Theorem 3 (TC3 is ±oprp-secure). *Let $\widetilde{\pi} = \mathrm{Perm}(\{0,1\}^n, n)$. If $\mathcal{A}$ asks queries having at most σ blocks then* $\mathbf{Adv}^{\pm\mathrm{oprp}}_{\mathrm{TC3}[\widetilde{\pi}]}(\mathcal{A}) \le 1.5\,\sigma^2/2^n$. ∎

The idea of the proof is to "give up"—regard the adversary as having won—if we ever generate a "new" tweak that collides with any prior one.

Proof. Without loss of generality we can assume that $\mathcal{A}$ is deterministic and makes queries totaling exactly σ blocks. We can further assume that it never repeats an encryption query, never repeats a decryption query, never asks a decryption query of a value that it earlier received from an encryption query, and never asks an encryption query of a value that it earlier received from a decryption query. For strings $X,\ X_1, \ldots, X_I \in (\{0,1\}^n)^*$, let $\mathrm{find}(X;\ X_1, \ldots, X_I)$ be the unique pair of numbers $(\imath, \ell)$ for which X and $X_\imath$ share a common prefix $X[1..\ell] = X_\imath[1..\ell]$, no $X_\jmath$ ($\jmath \in [1..I]$) shares a *longer* common prefix with X ($X[1..\ell+1] = X_\jmath[1..\ell+1]$), and $\imath$ is the *smallest* index in $[1..I]$ for which the above is true. If $X = \varepsilon$ define $\mathrm{find}(X;\ X_1, \ldots, X_I) = (0,0)$. By way of examples, if $a, b, c \in \{0,1\}^n$ are distinct blocks then $\mathrm{find}(abca;\ abaa, abcb, abcc) = (2,3)$, $\mathrm{find}(abca;\ a, abc, abcab) = (3,4)$, and $\mathrm{find}(abca;\ bbab, cba, b) = (1,0)$.

We employ the code-based games [6] shown in Fig. 2. Booleans are silently initialized to false and integers to 0. The one variable that is a set, $\mathcal{T}$, is silently initialized to $\{0^n\}$. (This is done because $\mathcal{T}$ will be used to record the set of tweaks that have been utilized and, in effect, 0^n is a tweak that is always used—it is used in processing each query's first block.) Partial functions π_x (where $x \in \{0,1\}^*$) are, initially, everywhere *undefined*. As they grow we refer to their current domain and range by $\mathrm{domain}(\pi_x)$ and $\mathrm{range}(\pi_x)$. We write $\mathrm{codomain}(\pi_x)$ and $\mathrm{corange}(\pi_x)$ for the complements relative to $\{0,1\}^n$.

We begin Fig. 2 with game G_1, which precisely emulates the TC3 construction with the ideal tweakable blockcipher $\widetilde{E}$. We end with game G_6, which precisely emulates the ideal online cipher. Thus we have that $\mathbf{Adv}^{\mathrm{oprp}}_{\widetilde{E}}(\mathcal{A}) = \Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_6^{\mathcal{A}} \Rightarrow 1]$. Games G_2, G_3, G_4, and G_5 are hybrid games in between these two extremes, and we bound the $\Pr[G_1^{\mathcal{A}} \Rightarrow 1] - \Pr[G_6^{\mathcal{A}} \Rightarrow 1]$ as $\sum_{1 \le j \le 5} \big(\Pr[G_j^{\mathcal{A}} \Rightarrow 1] - \Pr[G_{j+1}^{\mathcal{A}} \Rightarrow 1]\big)$.

Passing from games G_1 to G_2 is just the usual approach of lazy sampling [6]; the games G_1 and G_2 are adversarially indistinguishable. By the game-playing lemma, $\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \Pr[G_3^{\mathcal{A}} \Rightarrow 1]$ is at most the probability that game $\mathcal{A}$ manages

```
100 procedure E(M)
101 m ← |M|/n;  (ι, ℓ) ← find(M; M_1, ..., M_i)
102 C[1..ℓ] ← C_ι[1..ℓ]
103 for j ← ℓ + 1 to m do
104   if j = 1 then t ← 0^n
105   else t ← M[j − 1] ⊕ C[j − 1];
106   C[j] ← π_t(M[j])
107 i ← i + 1;  (M_i, C_i) ← (M, C)
108 return C
```

```
150 procedure D(C)
151 m ← |C|/n;  (ι, ℓ) ← find(C; C_1, ..., C_i)
152 M[1..ℓ] ← M_ι[1..ℓ]
153 for j ← ℓ + 1 to m do
154   if j = 1 then t ← 0^n
155   else t ← M[j − 1] ⊕ C[j − 1];
156   M[j] ← π_t^{−1}(C[j])
157 i ← i + 1;  (M_i, C_i) ← (M, C)
158 return M                          Game G_1
```

```
200 procedure E(M)
201 m ← |M|/n;  (ι, ℓ) ← find(M; M_1, ..., M_i)
202 C[1..ℓ] ← C_ι[1..ℓ]
203 for j ← ℓ + 1 to m do
204   if j = 1 then t ← 0^n
205   else t ← M[j − 1] ⊕ C[j − 1];
206   x ← M[j]
207   if x ∈ domain(π_t) then
208     bad_1 ← true; [ C[j] ← π_t(x); next ]
209   y ←$ {0,1}^n
210   if y ∈ range(π_t) then
211     bad_2 ← true; [ y ←$ corange(π_t) ]
212   π_t(x) ← y;  C[j] ← y;  t ← x ⊕ y
213   if t ∈ T then bad_3 ← true
214   T ← T ∪ {t}
215 i ← i + 1;  (M_i, C_i) ← (M, C)
216 return C
```

```
250 procedure D(C)
251 m ← |C|/n;  (ι, ℓ) ← find(C; C_1, ..., C_i)
252 M[1..ℓ] ← M_ι[1..ℓ]
253 for j ← ℓ + 1 to m do
254   if j = 1 then t ← 0^n
255   else t ← M[j − 1] ⊕ C[j − 1];
256   y ← C[j]
257   if y ∈ range(π_t) then
258     bad_1 ← true; [ M[j] ← π_t^{−1}(y); next ]
259   x ←$ {0,1}^n
260   if x ∈ domain(π_t) then
261     bad_2 ← true; [ x ←$ codomain(π_t) ]
262   π_t(x) ← y;  M[j] ← x;  t ← x ⊕ y
263   if t ∈ T then bad_3 ← true
264   T ← T ∪ {t}
265 i ← i + 1;  (M_i, C_i) ← (M, C)  [ Game G_2 ]
266 return M                          Game G_3
```

```
300 procedure E(M)
301 m ← |M|/n;  (ι, ℓ) ← find(M; M_1, ..., M_i)
302 C[1..ℓ] ← C_ι[1..ℓ]
303 for j ← ℓ + 1 to m do C[j] ←$ {0,1}^n
304 i ← i + 1;  (M_i, C_i) ← (M, C)
305 return C
```

```
350 procedure D(C)
351 m ← |C|/n;  (ι, ℓ) ← find(C; C_1, ..., C_i)
352 M[1..ℓ] ← M_ι[1..ℓ]
353 for j ← ℓ + 1 to m do M[j] ←$ {0,1}^n
354 i ← i + 1;  (M_i, C_i) ← (M, C)
355 return M                          Game G_4
```

```
400 procedure E(M)
401 m ← |M|/n;  (ι, ℓ) ← find(M; M_1, ..., M_i)
402 C[1..ℓ] ← C_ι[1..ℓ]
403 for j ← ℓ + 1 to m do
404   P ← M[1..j − 1]; x ← M[j]; y ←$ {0,1}^n
405   if y ∈ range(π_P) then
406     bad ← true; [ y ←$ corange(π_P) ]
407   π_P(x) ← y;  C[j] ← y
408 i ← i + 1;  (M_i, C_i) ← (M, C)
409 return C
```

```
450 procedure D(C)
451 m ← |C|/n;  (ι, ℓ) ← find(C; C_1, ..., C_i)
452 M[1..ℓ] ← M_ι[1..ℓ]
453 for j ← ℓ + 1 to m do
454   P ← M[1..j − 1]; y ← C[j]; x ←$ {0,1}^n
455   if x ∈ domain(π_P) then
456     bad ← true; [ x ←$ codomain(π_P) ]
457   π_P(x) ← y;  M[j] ← x
458 i ← i + 1;  (M_i, C_i) ← (M, C)   Game G_5
459 return M                          [ Game G_6 ]
```

Fig. 2. Games used in the proof of Theorem 3. Game G_2 includes the bracketed statements while game G_3 does not. Similarly, game G_6 includes the bracketed statements while game G_5 does not.

to set one of the bad_j variables in game G_3. The crux of the proof is the following observation:

Claim: Every execution of game G_3 that sets flag bad_1 also sets flag bad_3.

In fact, flag bad_3 was introduced as a trick for bounding the probability that bad_1 gets set. The proof of the claim is as follows. Suppose we are executing adversary $\mathcal{A}$ with game G_2 and, at some point in time it happens that, at line 207, we have $x \in \text{domain}(\pi_t)$, so that bad_1 will get set in the following line. Fix the current values M, m, $(\imath, \ell)$, $C[1..\ell]$, t, and x. Now x belonging to $\text{domain}(\pi_t)$ means that some triple (t, x, y) was already added into the set of triples that constitute the partial function π (that is, (t, x, y) is "in" π if we have set $\pi_t(x) = y$). This triple had to have been added to π at some earlier execution of line 212 or 262. We distinguish two possibilities: (t, x, y) is the *only* triple in π with this t-value; or else there are, already, at least two distinct triples (t, x, y), (t, x', y') with this particular t-value. In the latter case, when we added the temporally second of these two triples into π we checked, at line 213 or 263, if t was *already* in $\mathcal{T}$. It would have been, so bad_3 would already have been set. What remains is the case that (t, x, y) is the *only* triple in π with the given value t. Focus on the fact that $M[1] \cdots M[\ell]$ matches $M_\imath[1] \cdots M_\imath[\ell]$. Now if the latter string is *all* of $M_\imath$ then there were two prior times that t was generated: one is when (t, x, y) got added to π, and another is when the final t-value was generated in response to the $\imath$-th query—that is, when we executed the final statement at line 212 or 262. The temporally second of these t-producing events would have resulted in production of a t that was already in $\mathcal{T}$ and bad_3 would have been set. If, instead, $M[1] \cdots M[\ell] = M_\imath[1] \cdots M_\imath[\ell]$ and $M_\imath$ continues with at least one nonempty block $M_\imath[\ell{+}1]$, then we know that $M[\ell{+}1] \neq M_\imath[\ell{+}1]$ and the one and only triple in π with the given t-value must be $(t, M_\imath[\ell{+}1], C_\imath[\ell{+}1])$, so $x = M[\ell + 1] \neq M_\imath[\ell + 1]$ could not have caused line 207 to evaluate to true.

The case where bad_1 gets set on a decryption query, at line 258, is symmetric with the paragraph above: again bad_3 will already have been set. This completes the proof of the claim. □

Continuing, we now know that $\Pr[bad_1 \wedge bad_2 \wedge bad_3] = \Pr[bad_2 \wedge bad_3]$, which is at most $\Pr[bad_2] + \Pr[bad_3]$. The first probability is at most $0.5\,\sigma(\sigma - 1)/2^n$ and the second is at most $0.5\,\sigma(\sigma + 1)/2^n$ (recall that $\mathcal{T}$ was initially seeded with a point). We thus have that $\Pr[G_2^{\mathcal{A}} \Rightarrow 1] - \Pr[G_3^{\mathcal{A}} \Rightarrow 1] \leq \sigma^2/2^n$.

Games G_3 and G_4 are easily seen to be adversarially indistinguishable; we have simply eliminated the pointless code. Games G_4 and G_5 are adversarially indistinguishable; here we are introducing using lazy sampling. Passing from G_5 to G_6 can be regarded as a form of the PRP/PRF switching lemma (cf. [4, Lemma 3.7]) and the probability that bad gets set to true in game G_6 is at most $0.5\,\sigma^2/2^n$. The theorem now follows. ▮

The complexity-theoretic analog easily follows. This time, we show how the theorem looks.

Corollary 1 (TC3 is ±oprp-secure). *Let* $\widetilde{E}\colon \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ *be a tweakable blockcipher. Let* $\mathcal{A}$ *be an adversary that runs in time* t *and asks queries totaling at most* σ *blocks. Then there exists an adversary* $\mathcal{B}$ *such that* $\mathbf{Adv}^{\widetilde{\mathrm{prp}}}_{\mathrm{TC3}[\widetilde{E}]}(\mathcal{B}) \geq \mathbf{Adv}^{\mathrm{oprp}}_{\mathcal{E}}(\mathcal{A}) - 1.5\,\sigma^2/2^n$. *Adversary* $\mathcal{B}$ *runs in time at most* $t + cn\sigma$, *for some absolute constant* c, *and asks at most* σ *queries.* ∎

Mode TC3, beyond being a natural simplification to the generalization to mode HCBC2 [4], is a generalization of Nandi's mode MHCBC [22]; the latter can be realized as a special case of TC3 by selecting the tweakable blockcipher $\widetilde{E}\colon (\mathcal{K}_1 \times \mathcal{K}_2) \times \{0,1\}^n \to \{0,1\}^n$ to be $\widetilde{E}^T_{K1\,K2}(X) = E_{K1}(M \oplus H_{K2}(T)) \oplus H_{K2}(T)$ where $H\colon \mathcal{K}_2 \times \{0,1\}^n \to \{0,1\}^n$ is an almost-xor universal hash function and E is a blockcipher [17].

5 Online Ciphers for Arbitrary-Length Strings

We start out by extending the notation $M[i]$ so that, for each nonempty string M we have that $M = M[1] \cdots M[m-1]\, M[m]$ where $m = \lfloor |M|/n \rfloor$, $|M[i]| = n$ for all $1 \leq i \leq m-1$, and $n \leq |M[m]| \leq 2n-1$. In other words, when M is not a multiple of n bits its final block $M[m]$ is chosen to be *long*, having between $n+1$ and $2n-1$ bits. All other blocks remain n-bits in length. With this notation in hand we define $\mathrm{Online}^*(n)$ to be the set of all length-preserving permutations $\pi\colon \{0,1\}^* \to \{0,1\}^*$ such that $\pi(M[1] \cdots M[i])$ depends only on $M[1] \cdots M[i]$ (for all $i \geq 1$). This set can be regarded as an idealized cipher, just like $\mathrm{Perm}(n)$ and $\mathrm{Online}(n)$. Now if $\mathcal{E}\colon \mathcal{K} \times \{0,1\}^* \to \{0,1\}^*$ is an online cipher and $\mathcal{A}$ is an adversary we can extend our prior definitions by using $\mathrm{Online}^*(n)$ in our reference experiment:

$$\mathbf{Adv}^{\mathrm{oprp}}_{\mathcal{E}}(\mathcal{A}) = \mathbf{Pr}[K \xleftarrow{\$} \mathcal{K}\colon \mathcal{A}^{\mathcal{E}_K} \Rightarrow 1] - \mathbf{Pr}[\pi \xleftarrow{\$} \mathrm{Online}^*(n)\colon \mathcal{A}^{\pi} \Rightarrow 1]$$
$$\mathbf{Adv}^{\pm\mathrm{oprp}}_{\mathcal{E}}(\mathcal{A}) = \mathbf{Pr}[K \xleftarrow{\$} \mathcal{K}\colon \mathcal{A}^{\mathcal{E}_K,\, \mathcal{E}_K^{-1}} \Rightarrow 1] - \mathbf{Pr}[\pi \xleftarrow{\$} \mathrm{Online}^*(n)\colon \mathcal{A}^{\pi,\, \pi^{-1}} \Rightarrow 1]$$

There is an alternative notion of security where, when M is not a multiple of n bits, the final block is *short* (having 1 to $n-1$ bits) instead of long (having $n+1$ to $2n-1$ bits). There are problems with this alternative notion. First, it is too weak. If the adversary learns $C = E_K(X \,\|\, 0)$, where $X \in (\{0,1\}^n)^+$, then it also knows $C' = E_K(X \,\|\, 1)$, which is just C with its final bit flipped. Second, despite this alternative notion being weak, instantiations are hard. This is because it is not known how to construct from an n-bit blockcipher an efficient and provably-secure cipher, with good bounds, for arbitrary input lengths less than n; see the literature on "format-preserving encryption" for a discussion of this problem [5]. While this short-string enciphering problem cannot be avoided if the original message M has fewer than n bits, there is no need to deal with it when $|M|$ has more than n bits, which, in applications, is likely to be most or all the time.

Now turning to constructions, let $\widetilde{E}\colon \mathcal{K} \times \{0,1\}^n \times \{0,1\}^{\leq 2n-1} \to \{0,1\}^{\leq 2n-1}$ be a tweakable cipher. From this primitive define the online cipher $\mathcal{E} = \mathrm{TC3}^*[\widetilde{E}]$ with key space $\mathcal{K}$ and message space $\mathcal{M} = \{0,1\}^*$. See Fig. 3. The construction is CCA-secure, as formalized below. We will take up in the next section how one constructs a tweakable cipher with message space $\{0,1\}^{\leq 2n-1}$.

TC3*

algorithm $\mathcal{E}_K(M)$
if $M = \varepsilon$ **then return** ε
$m \leftarrow \lfloor |M|/n \rfloor$; $T \leftarrow 0^n$
for $j \leftarrow 1$ **to** $m-1$ **do**
  $C[j] \leftarrow \widetilde{E}_K^T(M[j])$
  $T \leftarrow M[j] \oplus C[j]$
$C[m] \leftarrow \widetilde{E}_K^T(M[m])$
return C

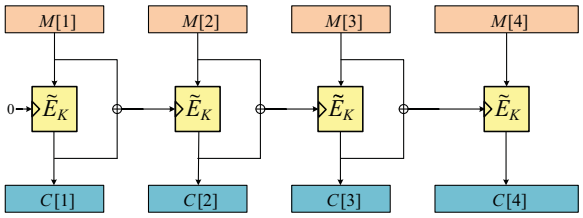

Fig. 3. Mode TC3*. The CCA-secure online cipher now takes an input of arbitrary bit length, but it depends on a richer primitive than does TC3: we start with a cipher $\widetilde{E}\colon \{0,1\}^{\leq 2n-1} \to \{0,1\}^{\leq 2n-1}$. The block input to $\widetilde{E}$ is "usually" n bits, but a single long-block call (up to $2n-1$ bits) will be used when $|M| \geq n$ and n doesn't divide $|M|$, while a single short-block call will be needed if $|M| < n$.

Theorem 4 (TC3* is $\pm$oprp-secure). *Let* $\widetilde{\pi} = \mathrm{Perm}(\{0,1\}^{\leq 2n-1})$. *If* $\mathcal{A}$ *asks queries having at most* σ *blocks then* $\mathbf{Adv}^{\pm\mathrm{oprp}}_{\mathrm{TC3}^*[\widetilde{\pi}]}(\mathcal{A}) \leq 1.5\,\sigma^2/2^n$. ∎

We omit the proof since it is almost the same as that for Theorem 3.

6 Instantiating the Schemes

Let us consider how to instantiate TC3 starting from a conventional (instead of a tweakable) blockcipher $E\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$. The simplest and most natural solution is to create the tweakable blockcipher by way of $\widetilde{E}_K^T(X) = E_{K_1}(X \oplus \Delta) \oplus \Delta$ where $\Delta = T \cdot K_2$ and $K = K_1 \,\|\, K_2$ for $|K_1| = k$ and $|K_2| = n$. Here multiplication, $T \cdot K_2$, is in $\mathrm{GF}(2^n)$, representing field points as n-bit strings in the usual way. We know that $\widetilde{E}\colon \mathcal{K} \times \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ will be a CCA-secure tweakable PRP as long as E is a CCA-secure conventional PRP; this is the well-known construction from Liskov, Rivest, and Wagner [17, Theorem 2], together with the fact that multiplication in $\mathrm{GF}(2^n)$—that is, $H_K(X) = K \cdot X$—is a 2^{-n}-AXU hash function.

The above construction is quite efficient, involving one blockcipher invocation and one $\mathrm{GF}(2^n)$ multiply for each message block. This is comparable to the work involved with the authenticated-encryption scheme GCM [19], which has, for example, been implemented by Gueron and Kounavis to run as fast as 3.54 cycles per byte [13] (on Intel processors supporting AES and PCLMULQDQ assembly instructions). This timing figure, however, would overestimate the expected speed of TC3, on similar hardware, since the blockcipher chaining in TC3 will decrease instruction-level parallelism.

Following Boldyreva and Taesombut, TC3 can be augmented, with little overhead, to provide a solution to the problem of blockwise-adaptive CCA-secure authenticated-encryption [9]. Doing so would give an AE scheme with efficiency roughly comparable to GCM but provably achieving a useful security property that GCM does not achieve.

Instantiating TC3* from a conventional blockcipher is more involved than instantiating TC3, as now we need a map $\widetilde{E}\colon \mathcal{K} \times \{0,1\}^n \times \{0,1\}^{\leq 2n-1} \to \{0,1\}^{\leq 2n-1}$. Actually our tweakable cipher will not have to deal with messages having fewer than n bits unless the higher-level construction $\mathcal{E}$ is itself asked to encipher messages of fewer than n bits, so let us put this case aside. Our problem then is to create from an ordinary n-bit blockcipher E a VIL-secure tweakable cipher that can encipher messages of n to $2n-1$ bits. Fortunately there are some ready solutions to this problem. Four or more rounds of Feistel would be the classical approach [18]. One would use a blockcipher-based, tweak-dependent round function. A different possibility is the EME2 cipher (formerly named EME*) of Halevi [14]; the mechanism was recently approved as the IEEE standard P1619.2-2010. The scheme is simple, provably secure, and, using five blockcipher calls and a modest amount of additional overhead, provides a tweakable and VIL cipher over the domain that we need. More efficient still would be the XLS construction of Ristenpart and Rogaway [24]. This can encipher strings of $n+1$ to $2n-1$ bits using three blockcipher calls and very little extra work.[4]

Enciphering strings of fewer than n bits takes special techniques. One proposal is FFX [7], which uses a conventional, unbalanced, or alternating Feistel network on these small domains. We note that if one is going to deal with short final blocks by a patchwork of techniques, one for strings in $\{0,1\}^{\leq n-1}$ and one for other strings that are not a multiple of n bits, it is important to use distinct keys, or to use other techniques, to provably ensure VIL security.

Acknowledgments

Many thanks for the perceptive comments from the anonymous referees. Our apologies that we have not expanded on points where this ought be done.

The authors gratefully acknowledge the support of NSF grant CNS 0904380.

References

1. G. Amanatidis, A. Boldyreva, and A. O'Neill. Provably-secure schemes for basic query support in outsourced databases. *Working Conference on Data and Applications Security* (DBSec 2007), LNCS vol. 4602, Springer, pp. 14–30, 2007.
2. G. Bard. A challenging but feasible blockwise-adaptive chosen-plaintext attack on SSL. *SECRYPT 2006*, *International Conference on Security and Cryptography*, INSTICC Press, pp. 99–109, 2006.
3. M. Bellare, A. Boldyreva, L. Knudsen, and C. Namprempre. On-line ciphers and the hash-CBC constructions. *CRYPTO 2001*, LNCS vol. 2139, Springer, pp. 292–309, 2001.

[4] In fact, none of the methods just surveyed was designed specifically to create a tweakable cipher with n-bit tweaks and a message space of n to $2n-1$ bits. With this limited goal in mind we believe that provably-good methods related to those of Naor-Reingold [23] will give a cipher needing just two blockcipher calls for strings of $n+1$ to $2n-1$ bits.

4. M. Bellare, A. Boldyreva, L. Knudsen, and C. Namprempre. On-line ciphers and the hash-CBC constructions. Cryptology ePrint report 2007/197, June 29, 2007. Full version of [3].
5. M. Bellare, T. Ristenpart, P. Rogaway, and T. Stegers. Format preserving encryption. *Selected Areas in Cryptography, SAC 2009*. LNCS vol. 5867, Springer, pp. 295–312, 2009.
6. 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.
7. M. Bellare, P. Rogaway, and T. Spies. The FFX mode of operation for format-preserving encryption (draft 1.1). NIST submission, February 2010. See also the addendum (September 2010) by the same authors.
8. D. Bernstein and P. Schwabe. New AES software speed records. *INDOCRYPT 2008*, LNCS vol. 5365, Springer, pp. 322–336, 2008.
9. A. Boldyreva and N. Taesombut. Online encryption schemes: new security notions and constructions. *CT-RSA 2004*, LNCS vol. 2964, Springer, pp. 1–14, 2004.
10. P. Fouque, A. Joux, G. Martinet, and F. Valette. Authenticated on-line encryption. *SAC 2003*, LNCS vol. 3006, Springer, pp. 145–159, 2003.
11. 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.
12. P. Fouque, G. Martinet, G. Poupard. Practical symmetric on-line encryption. *FSE 2003*, LNCS vol. 2887, Springer, pp. 362–375, 2003.
13. S. Gueron and M. Kounavis. Intel carry-less multiplication instruction and its usage for computing the GCM mode (revision 2). White paper, available at www.intel.com. May 2010.
14. S. Halevi. EME*: extending EME to handle arbitrary-length messages with associated data. *INDOCRYPT 2004*, LNCS vol. 3348, Springer, pp. 315–327, 2004.
15. 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.
16. H. Krawczyk. LFSR-based hashing and authentication. *CRYPTO 1994*, LNCS vol. 839, Springer, pp. 129–139, 1994.
17. M. Liskov, R. Rivest, and D. Wagner. Tweakable block ciphers. *CRYPTO 2002*, LNCS vol. 2442, Springer, pp. 31–46, 2002.
18. M. Luby and C. Rackoff. How to construct pseudorandom permutations from pseudorandom functions. *SIAM Journal of Computing*, 17(2), pp. 373–386, 1988.
19. D. McGrew and J. Viega. The security and performance of the Galois/counter mode (GCM) of operation. *INDOCRYPT 2004*, LNCS vol. 3348, Springer, pp. 343–355, 2004.
20. C. Meyer and M. Matyas. *Cryptography: A New Dimension in Data Security*. John Wiley & Sons, New York, 1982.
21. M. Nandi. A simple security analysis of Hash-CBC and a new efficient one-key online cipher. Cryptology ePrint report 2007/158. May 7, 2007.
22. M. Nandi. Two New Efficient CCA-secure online ciphers: MHCBC and MCBC. *INDOCRYPT 2008*, LNCS vol. 5365, Springer, pp. 350–362, 2008. Also Cryptology ePrint report 2008/401, September 20, 2008.
23. M. Naor and O. Reingold. On the construction of pseudorandom permutations: Luby-Rackoff revisited. *Journal of Cryptology*, 12(1), pp. 29–66, 1999.

24. T. Ristenpart and P. Rogaway. How to enrich the message space of a cipher. *FSE 2007*, LNCS vol. 4593, pp. 101–118, 2007.