Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC

Phillip Rogaway

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

Abstract. We describe highly efficient constructions, XE and XEX, that turn a blockcipher $E\colon \mathcal{K}\times\{0,1\}^n \to \{0,1\}^n$ into a tweakable blockcipher $\widetilde{E}\colon \mathcal{K}\times\mathcal{T}\times\{0,1\}^n \to \{0,1\}^n$ having tweak space $\mathcal{T}=\{0,1\}^n\times\mathbb{I}$ where $\mathbb{I}$ is a set of tuples of integers such as $\mathbb{I}=[1\mathinner{..}2^{n/2}]\times[0\mathinner{..}10]$. When tweak T is obtained from tweak S by incrementing one if its numerical components, the cost to compute $\widetilde{E}_K^T(M)$ having already computed some $\widetilde{E}_K^S(M')$ is one blockcipher call plus a small and constant number of elementary machine operations. Our constructions work by associating to the i^{th} coordinate of $\mathbb{I}$ an element $\alpha_i \in \mathbb{F}_{2^n}^*$ and multiplying by α_i when one increments that component of the tweak. We illustrate the use of this approach by refining the authenticated-encryption scheme OCB and the message authentication code PMAC, yielding variants of these algorithms that are simpler and faster than the original schemes, and yet have simpler proofs. Our results bolster the thesis of Liskov, Rivest, and Wagner [10] that a desirable approach for designing modes of operation is to start from a tweakable blockcipher. We elaborate on their idea, suggesting the kind of tweak space, usage-discipline, and blockcipher-based instantiations that give rise to simple and efficient modes.

1 Introduction

Liskov, Rivest and Wagner [10] defined the notion of a tweakable blockcipher and put forward the thesis that these objects make a good starting point for doing blockcipher-based cryptographic design. In this paper we describe a good way to build a tweakable blockcipher $\widetilde{E}$ out of an ordinary blockcipher E. Used as intended, our constructions, XE and XEX, add just a few machine instructions to the cost of computing E. We illustrate the use of these constructions by improving on the authenticated-encryption scheme OCB [15] and the message authentication code PMAC [4].

Tweakable blockciphers. Schroeppel [16] designed a blockcipher, Hasty Pudding, wherein the user supplies a non-secret *spice* and changing this spice produces a completely different permutation. Liskov, Rivest, and Wagner [10] formally defined the syntax and security measures for such a *tweakable* blockcipher, and they suggested that this abstraction makes a desirable starting point to design modes of operation and prove them secure. They suggested ways to build a tweakable blockcipher $\widetilde{E}$ out of a standard blockcipher E, as well as ways to modify existing blockcipher designs to incorporate a tweak. They illustrated the use of these objects. Formally, a tweakable blockcipher is a map $\widetilde{E}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ where each $\widetilde{E}_K^T(\cdot) = \widetilde{E}(K,T,\cdot)$ is a permutation and $\mathcal{T}$ is the set of *tweaks*.

Our contributions. We propose efficient ways to turn a blockcipher $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ into a tweakable blockcipher $\widetilde{E}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$. (See Appendix A for the best constructions formerly known.) Our *powering-up* constructions, XE and XEX, preserve the key space and blocksize of E but endow $\widetilde{E}$ with a tweak space $\mathcal{T} = \{0,1\}^n \times \mathbb{I}$ where $\mathbb{I}$ is a set of tuples of integers, like $\mathbb{I} = [1 \,..\, 2^{n/2}] \times [0 \,..\, 10]$. The XE construction turns a CPA-secure blockcipher into a CPA-secure tweakable blockcipher, while XEX turns a CCA-secure blockcipher into a CCA-secure tweakable blockcipher. (CPA stands for chosen-plaintext attack and CCA for chosen-ciphertext attack.) The methods are highly efficient when tweaks arise in sequence, with most tweaks $(N, \mathbf{i})$ being identical to the prior tweak $(N, \mathbf{i}')$ except for incrementing a component of $\mathbf{i}$.

As an illustrative and useful example, consider turning a conventional blockcipher $E\colon \mathcal{K}\times\{0,1\}^n \to \{0,1\}^n$ into a tweakable blockcipher $\widetilde{E}\colon \mathcal{K}\times\mathcal{T}\times\{0,1\}^n \to \{0,1\}^n$ by defining $\widetilde{E}_K^{N\,i\,j}(M) = E_K(M \oplus \Delta) \oplus \Delta$ where offset $\Delta = 2^i 3^j\,\mathsf{N}$ and $\mathsf{N} = E_K(N)$. Arithmetic is done in the finite field $\mathbb{F}_{2^n}$. For concreteness, assume $n = 128$ and a tweak space of $\mathcal{T} = \{0,1\}^n \times [1 \,..\, 2^{64}] \times [0 \,..\, 10]$. We show that $\widetilde{E}$ is secure (as a strong, tweakable PRP) as long as E is secure (as a strong, untweakable PRP). Computing $\widetilde{E}_K^{N\,i\,j}(X)$ will usually cost about 1 shift, 1 conditional, and 3–4 xors more than computing $E_K(X)$.

We illustrate how the use of tweakable blockciphers during mode design, followed by the instantiation of the tweakable blockcipher with an ordinary blockcipher using one of our constructions, can give rise to modes that are simpler, faster, and easier to prove correct than what designing directly from a blockcipher has delivered. We do this by refining two already-optimized modes, OCB [15] and PMAC [4], yielding new modes, OCB1 and PMAC1, that are are easier to understand, easier to implement, and faster. Computing offsets in the new modes does not involve Gray-code sequence or counting the number of trailing zero bits in successive integers. OCB1 eliminates the utility of preprocessing, saving a blockcipher call.

Intuition. The idea behind the powering-up constructions can be explained like this. Apart from Gray-code reordering, PMAC authenticates an m-block message using a sequence of offsets $L, 2L, 3L, \ldots, (m-1)L$, where multiplication is in the finite field $\mathbb{F}_{2^n}$ and $L = E_K(0^n)$ is a variant of the underlying key K. When a special kind of offset is needed, a value $\mathit{huge} \cdot L$ is added (xored) into the current offset, where *huge* is so large that it could never be among $\{1, 2, \ldots, m-1\}$. What we now do instead is to use the easier-to-compute sequence of offsets $2^1L, 2^2L, \ldots, 2^{m-1}L$. We insist that our field be represented using a primitive polynomial instead of merely an irreducible one, which ensures that $2^1, 2^2, 2^3, \ldots, 2^{2^n-1}$ will all be distinct. When a special offset is needed we can no longer add to the current offset some huge constant times L and expect this never to land on a point in $2^1L, 2^2L, \ldots, 2^{m-1}L$. Instead, we multiply the current offset by 3 instead of 2. If the index of 3 (in $\mathbb{F}_{2^n}^*$) is enormous relative to the base 2 then multiplying by 3 is equivalent to multiplying by 2^{huge} and 2^i3L won't be among of $2^1L, 2^2L, \ldots, 2^{m-1}L$ for any reasonable value of m. The current paper will make all of the ideas of this paragraph precise.

Further related work. Halevi and Rogaway [7] used the sequence of offsets $2L, 2^2L, 2^3L, \ldots,$ in their EME mode. They give no general results about this construction, and EME did not use tweakable blockciphers, yet this offset ordering was our starting point.

2 Preliminaries

The field with 2^n points. Let $\mathbb{F}_{2^n}$ denote the field with 2^n points and let $\mathbb{F}_{2^n}^*$ be its multiplicative subgroup. We interchangeably think of a point $a \in \mathbb{F}_{2^n}$ as an n-bit string, a formal polynomial of degree $n-1$, or as an integer in $[0 \mathinner{..} 2^n{-}1]$. To represent points select a primitive polynomial, say the lexicographically first one among the degree n polynomials having a minimum number of nonzero coefficients. For $n = 128$ the indicated polynomial is $p_{128}(\mathtt{x}) = \mathtt{x}^{128} + \mathtt{x}^7 + \mathtt{x}^2 + \mathtt{x} + 1$. Saying that $p_n(\mathtt{x})$ is primitive means that it is irreducible over $\mathbb{F}_2$ and 2 (i.e., $\mathtt{x}$) generates all of $\mathbb{F}_{2^n}^*$. It is computationally simple to multiply $a \in \{0,1\}^n$ by 2. To illustrate for $n = 128$, $2a = a{\ll}1$ if $\mathsf{firstbit}(a) = 0$ and $2a = (a{\ll}1) \oplus 0^{120}10^41^3$ if $\mathsf{firstbit}(a) = 1$. One can easily multiply by other small constants, as $3a = 2a \oplus a$ and $5a = 2(2a) \oplus a$ and so forth.

Blockciphers and tweakable blockciphers. We review the standard definitions for blockciphers and their security [2] and the extension of these notions to tweakable blockciphers [10]. A *blockcipher* is a function $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ where $n \geq 1$ is a number and $\mathcal{K}$ is a finite nonempty set and $E(K, \cdot) = E_K(\cdot)$ is a permutation for all $K \in \mathcal{K}$. A *tweakable blockcipher* is a function $\widetilde{E}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ where n and $\mathcal{K}$ are as above and $\mathcal{T}$ is a nonempty set and $\widetilde{E}(K, T, \cdot) = \widetilde{E}_K^T(\cdot)$ is a permutation for all $K \in \mathcal{K}$ and $T \in \mathcal{T}$. For blockciphers and tweakable blockciphers we call n the *blocksize* and $\mathcal{K}$ the *key space*. For tweakable blockciphers we call $\mathcal{T}$ the *tweak space*.

Let $\mathrm{Perm}(n)$ be the set of all permutations on n bits. Let $\mathrm{Perm}(\mathcal{T}, n)$ be the set of all mappings from $\mathcal{T}$ to permutations on n bits. In writing $\pi \stackrel{\$}{\leftarrow} \mathrm{Perm}(n)$ we are choosing a random permutation $\pi(\cdot)$ on $\{0,1\}^n$. In writing $\pi \stackrel{\$}{\leftarrow} \mathrm{Perm}(\mathcal{T}, n)$ we are choosing a random permutation $\pi(T, \cdot) = \pi_T(\cdot)$ on $\{0,1\}^n$ for each $T \in \mathcal{T}$. If $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ is a blockcipher then its inverse is the blockcipher $D = E^{-1}$ where $D\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ is defined by $D(K, Y) = D_K(Y)$ being the unique point X such that $E_K(X) = Y$. If $\widetilde{E}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is a tweakable blockcipher then its inverse is the tweakable blockcipher $\widetilde{D} = \widetilde{E}^{-1}$ where $\widetilde{D}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is defined by $\widetilde{D}(K, T, Y) = \widetilde{D}_K^T(Y)$ being the unique point X such that $\widetilde{E}_K^T(X) = Y$.

An adversary is a probabilistic algorithm with access to zero or more oracles. Without loss of generality, adversaries never ask a query for which the answer is trivially known: an adversary does not repeat a query, does not ask $D_K(Y)$ after receiving Y in response to a query $E_K(X)$, and so forth. Oracles will have an implicit domain of valid queries and, for convenience, we assume that all adversarial queries lie within that domain. This is not a significant restriction because membership can be easily tested for all domains of interest to us.

Definition 1 (Blockcipher/tweakable-blockcipher security). *Let* $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ *be a blockcipher and let* $\widetilde{E}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ *be a tweakable blockcipher. Let* A *be an adversary. Then* $\mathbf{Adv}_E^{\mathrm{prp}}(A)$, $\mathbf{Adv}_E^{\pm\mathrm{prp}}(A)$, $\mathbf{Adv}_{\widetilde{E}}^{\widetilde{\mathrm{prp}}}(A)$, *and* $\mathbf{Adv}_{\widetilde{E}}^{\pm\widetilde{\mathrm{prp}}}(A)$ *are defined by:*

$$\Pr[K \xleftarrow{\$} \mathcal{K}:\ A^{E_K(\cdot)} \Rightarrow 1] - \Pr[\pi \xleftarrow{\$} \mathrm{Perm}(n):\ A^{\pi(\cdot)} \Rightarrow 1]$$
$$\Pr[K \xleftarrow{\$} \mathcal{K}:\ A^{E_K(\cdot)\,D_K(\cdot)} \Rightarrow 1] - \Pr[\pi \xleftarrow{\$} \mathrm{Perm}(n):\ A^{\pi(\cdot)\,\pi^{-1}(\cdot)} \Rightarrow 1]$$
$$\Pr[K \xleftarrow{\$} \mathcal{K}:\ A^{\widetilde{E}_K(\cdot,\cdot)} \Rightarrow 1] - \Pr[\pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{T},n):\ A^{\pi(\cdot,\cdot)} \Rightarrow 1]$$
$$\Pr[K \xleftarrow{\$} \mathcal{K}:\ A^{\widetilde{E}_K(\cdot,\cdot)\,\widetilde{D}_K(\cdot,\cdot)} \Rightarrow 1] - \Pr[\pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{T},n):\ A^{\pi(\cdot,\cdot)\,\pi^{-1}(\cdot,\cdot)} \Rightarrow 1] \quad \square$$

Of course D and $\widetilde{D}$ denote the inverses of blockciphers E and $\widetilde{E}$. In writing $A \Rightarrow 1$ we are referring to the event that the adversary A outputs the bit 1.

In the usual way we lift advantage measures that depend on an adversary to advantage measures that depend on named resources: $\mathbf{Adv}_{\Pi}^{\mathrm{xxx}}(\mathcal{R}) = \max_A\{\mathbf{Adv}_{\Pi}^{\mathrm{xxx}}(A)\}$ over all adversaries A that use resources at most $\mathcal{R}$. The resources of interest to us are the total number of oracle queries q and the total length of those queries σ and the running time t. For convenience, the total length of queries will be measured in n-bit blocks, for some understood value of n, so a query X contributes $|X|_n$ to the total, where $|X|_n$ means $\max\{|X|/n, 1\}$. Running time, by convention, includes the description size of the algorithm relative to some standard encoding. When we speak of authenticity, the block length of the adversary's output is included in σ.

3 The XE and XEX Constructions

Goals. We want to support tweak sets that look like $\mathcal{T} = \{0,1\}^n \times \mathbb{I}$ where $\mathbb{I}$ is a set of tuples of integers. In particular, we want to be able to make $\mathbb{I}$ the cross product of a large subrange of integers, like $[1 \mathinner{..} 2^{n/2}]$, by the cross product of small ranges of integers, like $[0 \mathinner{..} 10] \times [0 \mathinner{..} 10]$. Thus an example tweak space is $\mathcal{T} = \{0,1\}^n \times [1 \mathinner{..} 2^{n/2}] \times [0 \mathinner{..} 10] \times [0 \mathinner{..} 10]$. Tweaks arise in some sequence $T_1, T_2, \ldots$ and we will obtain impressive efficiency only to the extent that most tweaks are an increment of the immediately prior one. When we say that tweak $T = (N, i_1, \ldots, i_k)$ is an increment of another tweak we mean that one of $i_1, \ldots, i_k$ got incremented and everything else stayed the same. The second component of tweak $(N, i_1, \ldots, i_k)$, meaning i_1, is the component that we expect to get incremented most often. We want there to be a simple, constant-time procedure to increment a tweak at any given component of $\mathbb{I}$. To increment a tweak it shouldn't be necessary to go to memory, consult a table, or examine which number tweak this in in sequence. Incrementing tweaks should be endian-independent and avoid extended-precision arithmetic. Efficiently incrementing tweaks shouldn't require precomputation. Tweaks that are not the increment of a prior tweak will also arise, and they will typically look like $(N, 1, 0 \ldots, 0)$. Constructions should be reasonably efficient in dealing with such tweaks.

We emphasize that the efficiency measure we are focusing on is not the cost of computing $\widetilde{E}_K^T(X)$ from scratch—by that measure our constructions will not be particularly good. Instead, we are interested in the cost of computing $\widetilde{E}_K^T(X)$ given that one has just computed $\widetilde{E}_K^S(X')$ and T is obtained by incrementing S at some component. Most often that component will have been the second component of S. It is a thesis underlying our work, supported by the design of OCB1 and PMAC1, that one will often be able to arrange that most tweaks are an increment to the prior one.

Tweaking with $\Delta = 2^i\,\mathsf{N}$. Recall that we have chosen to represent points in $\mathbb{F}_{2^n}$ using a primitive polynomial, not just an irreducible one. This means that the point 2 is a generator of $\mathbb{F}_{2^n}$: the points $1, 2, 2^2, 2^3, \ldots, 2^{2^n-2}$ are all distinct. This property turns out to be the crucial one that lets us construct from a blockcipher E: $\mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ a tweakable blockcipher $\widetilde{E}$: $\mathcal{K} \times (\{0,1\}^n \times [1\,..\,2^n-2]) \times \{0,1\}^n \to \{0,1\}^n$ by way of

$$\widetilde{E}_K^{N\,i}(M) = E_K(M \oplus \Delta) \oplus \Delta \quad \text{where} \quad \Delta = 2^i\,\mathsf{N} \quad \text{and} \quad \mathsf{N} = E_K(N).$$

The tweak set is $\mathcal{T} = \{0,1\}^n \times \mathbb{I}$ where $\mathbb{I} = [1\,..\,2^n-2]$ and the tweakable blockcipher just described is denoted $\widetilde{E} = \mathrm{XEX}[E, 2^{\mathbb{I}}]$. When computing the sequence of values $\widetilde{E}_K^{N\,1}(M_1), \ldots, \widetilde{E}_K^{N\,m-1}(M_{m-1})$ each $\widetilde{E}_K^{N\,i}(M_i)$ computation but the first uses one blockcipher call and one doubling operation. Doubling takes a shift followed by a conditional xor. We call the construction above, and all the subsequent constructions of this section, *powering-up* constructions.

Tweaking by $\Delta = 2^i 3^j\,\mathsf{N}$. To facilitate mode design we may want tweaks that look like (N, i, j) where $N \in \{0,1\}^n$ and i is an integer from a large set $\mathbb{I}$, like $\mathbb{I} = [1\,..\,2^{n/2}]$, and j is an integer from some small set $\mathbb{J}$, like $\mathbb{J} = \{0,1\}$. To get the "diversity" associated to the various j-values we just multiply by 3 instead of 2. That is, we construct from a blockcipher E: $\mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ a tweakable blockcipher $\widetilde{E}$: $\mathcal{K} \times (\{0,1\}^n \times \mathbb{I} \times \mathbb{J}) \times \{0,1\}^n \to \{0,1\}^n$ by way of

$$\widetilde{E}_K^{N\,i\,j}(M) = E_K(M \oplus \Delta) \oplus \Delta \quad \text{where} \quad \Delta = 2^i 3^j\,\mathsf{N} \quad \text{and} \quad \mathsf{N} = E_K(N).$$

The tweakable blockcipher just described is denoted $\widetilde{E} = \mathrm{XEX}[E, 2^{\mathbb{I}}3^{\mathbb{J}}]$. Incrementing the tweak at component i is done by doubling, while incrementing the tweak at component j is done by tripling.

The XEX construction. Generalizing the two examples above, we have the following definition.

Definition 2 (XEX). *Let* $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ *be a blockcipher, let* $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_{2^n}^*$*, and let* $\mathbb{I}_1, \ldots, \mathbb{I}_k \subseteq \mathbb{Z}$*. Then* $\widetilde{E} = \mathsf{XEX}[E, \alpha_1^{\mathbb{I}_1} \cdots \alpha_k^{\mathbb{I}_k}]$ *is the tweakable blockcipher* $\widetilde{E}\colon \mathcal{K} \times (\{0,1\}^n \times \mathbb{I}_1 \times \cdots \times \mathbb{I}_k) \times \{0,1\}^n \to \{0,1\}^n$ *defined by* $\widetilde{E}_K^{N i_1 \ldots i_k}(M) = E_K(M \oplus \Delta) \oplus \Delta$ *where* $\Delta = \alpha_1^{i_1} \alpha_2^{i_2} \cdots \alpha_k^{i_k}\, \mathsf{N}$ *and* $\mathsf{N} = E_K(N)$.

The XE construction. As made clear in the work of Liskov, Rivest, and Wagner [10], constructions of the form $\widetilde{E}_K^T(M) = E_K(M \oplus \Delta) \oplus \Delta$ aim for chosen-ciphertext attack (CCA) security, while for chosen-plaintext attack (CPA) security one can omit the outer xor. Thus we consider the construction $E_K(M \oplus \Delta)$. This is slightly more efficient than XEX, saving one xor.

Definition 3 (XE). *Let* $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ *be a blockcipher,* $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_{2^n}^*$*, and* $\mathbb{I}_1, \ldots, \mathbb{I}_k \subseteq \mathbb{Z}$*. Then* $\widetilde{E} = \mathsf{XE}[E, \alpha_1^{\mathbb{I}_1} \cdots \alpha_k^{\mathbb{I}_k}]$ *is the tweakable blockcipher* $\widetilde{E}\colon \mathcal{K} \times (\{0,1\}^n \times \mathbb{I}_1 \times \cdots \times \mathbb{I}_k) \times \{0,1\}^n \to \{0,1\}^n$ *defined by* $\widetilde{E}_K^{N i_1 \ldots i_k}(M) = E_K(M \oplus \Delta)$ *where* $\Delta = \alpha_1^{i_1} \alpha_2^{i_2} \cdots \alpha_k^{i_k}\, \mathsf{N}$ *and* $\mathsf{N} = E_K(N)$. □

4 Parameter Sets Yielding Unique Representations

It is easy to see that the XE and XEX constructions can only "work" if $\alpha_1^{i_1} \cdots \alpha_k^{i_k}$ are distinct throughout $(i_1, \ldots, i_k) \in \mathbb{I}_1 \times \cdots \times \mathbb{I}_k$. This motivates the following definition.

Definition 4 (Unique representations). *Fix a group* G*. A* ***choice of parameters*** *is a list* $\alpha_1, \ldots, \alpha_k \in G$ *of* ***bases*** *and a set* $\mathbb{I}_1 \times \cdots \times \mathbb{I}_k \subseteq \mathbb{Z}^k$ *of* ***allowed indices****. We say that the choice of parameters provides* ***unique representations*** *if for every* $(i_1, \ldots, i_k), (j_1, \ldots, j_k) \in \mathbb{I}_1 \times \cdots \times \mathbb{I}_k$ *we have that* $\alpha_1^{i_1} \cdots \alpha_k^{i_k} = \alpha_1^{j_1} \cdots \alpha_k^{j_k}$ *implies* $(i_1, \ldots, i_k) = (j_1, \ldots, j_k)$. □

In other words, representable points are uniquely representable: any group element $\alpha_1^{i_1} \cdots \alpha_k^{i_k}$ that can be represented using allowed indices can be represented in only one way (using allowed indices).

For tweak spaces of practical interest, discrete-log calculations within $\mathbb{F}_{2^n}^*$ can be used to help choose and verify that a given choice of parameters provides unique representations. The following result gives examples for $\mathbb{F}_{2^{128}}^*$.

Proposition 1. [Can use 2, 3, 7 when $n = 128$] *In the group* $\mathbb{F}_{2^{128}}^*$ *the following choices for parameters provide unique representations:*

(1) $\alpha_1 = 2$ *and* $\mathbb{I}_1 = [-2^{126} \mathinner{..} 2^{126}]$.

(2) $\alpha_1, \alpha_2 = 2, 3$ *and* $\mathbb{I}_1 \times \mathbb{I}_2 = [-2^{115} \mathinner{..} 2^{115}] \times [-2^{10} \mathinner{..} 2^{10}]$.

(3) $\alpha_1, \alpha_2, \alpha_3 = 2, 3, 7$ *and* $\mathbb{I}_1 \times \mathbb{I}_t \times \mathbb{I}_3 = [-2^{108} \mathinner{..} 2^{108}] \times [-2^7 \mathinner{..} 2^7] \times [-2^7 \mathinner{..} 2^7]$.

Proof. For statement (1) recall that 2 is a generator of the group (by our choice of irreducible polynomial) and the order of the group $\mathbb{F}_{2^{128}}^*$ is $2^{128} - 1$ and so $2^i = 2^j$ iff $i = j \pmod{2^{128} - 1}$ and so any contiguous range of $2^{128} - 1$ or fewer integers will provide unique representations with respect to base 2.

To prove statement (2) we need to compute $\log_2 3$ in the group $\mathbb{F}^*_{2^{128}}$:

$$\log_2 3 = 338793687469689340204974836150077311399 \quad \text{(decimal)}$$

This and subsequent discrete logs were computed using a Maple-implementation combining the Pohlig-Hellman [11] and Pollard-rho [12] algorithms. (A naive implementation computes discrete logs in $\mathbb{F}^*_{2^{128}}$ in a few hours.) Now note that $2^a3^b = 2^{a'}3^{b'}$ iff $2^a 2^{b\log_2 3} = 2^{a'}2^{b'\log_2 3}$ iff $2^{a+b\log_2 3} = 2^{a'+b'\log_2 3}$ iff $a + b\log_2 3 = a' + b'\log_2 3 \pmod{2^{128}-1}$ because 2 is a generator of the group $\mathbb{F}^*_{2^{128}}$. Thus $2^a3^b = 2^{a'}3^{b'}$ iff $a - a' = (b'-b)\log_2 3 \pmod{2^{128}-1}$. If $b, b' \in [-2^{10} \mathinner{..} 2^{10}]$ then $\Delta_b = b' - b \in [-2^{11} \mathinner{..} 2^{11}]$ and computer-assisted calculation then shows that the smallest value of $\Delta_b \log_2 3 \pmod{2^{128}-1}$ for $\Delta_b \in [-2^{11}..2^{11}]$ and $\Delta_b \neq 0$ is $1600 \log_2 3 =$ 00113a0ce508326c006763c0b80c59f9 (in hexadecimal) which is about $2^{116.1}$. (By "smallest" we refer to the distance from 0, modulo $2^{128}-1$, so 2^{100} and $(2^{128}-1) - 2^{100}$ are equally small, for example.) Thus if a, a' are restricted to $[-2^{115} \mathinner{..} 2^{115}]$ and b, b' are restricted to $[-2^{10} \mathinner{..} 2^{10}]$ then $\Delta_a = a - a' \leq 2^{116}$ can never equal $\Delta_b \log_2 3 \pmod{2^{128}-1} > 2^{116}$ unless $\Delta_b = 0$. This means that the only solution to $2^a3^b = 2^{a'}3^{b'}$ within the specified range is $a = a'$ and $b = b'$.

To prove statement (3) is similar. First we need the value

$$\log_2 7 = 305046802472688182329780655685899195396 \quad \text{(decimal)}$$

Now $2^a3^b7^c = 2^{a'}3^{b'}7^{c'}$ iff $a - a' = (b'-b)\log_2 3 + (c'-c)\log_2 7 \pmod{2^{128}-1}$. The smallest value for $\Delta_b \log_2 3 + \Delta_c \log_2 7 \pmod{2^{128}-1}$ when $\Delta_b, \Delta_c \in [-2^8 \mathinner{..} 2^8]$ and at least one of these is non-zero is $-48\log_2 3 + 31 \log_2 7 \pmod{2^{128}-1} =$ 00003bfabac91e02b278b7e69a379d18 (hexadecimal) which is about $2^{109.9}$. So restricting the index for base-2 to $[-2^{108} \mathinner{..} 2^{108}]$ ensures that $a - a' \leq 2^{109}$ while $(b'-b)\log_2 3 + (c'-c)\log_2 7 > 2^{109}$ unless $b = b'$ and $c = c'$ and $a = a'$. □

We emphasize that not just any list of bases will work. Notice, for example, that $3^2 = 5$ in $\mathbb{F}^*_{2^n}$ so the list of bases $2, 3, 5$ does *not* give unique representations, even for a tiny list of allowed indices like $\mathbb{I}_1 \times \mathbb{I}_2 \times \mathbb{I}_3 = \{0, 1, 2\}^3$.

Similar calculations can be done in other groups; here we state the analogous result for $\mathbb{F}^*_{2^{64}}$.

Proposition 2. [Can use 2, 3, 11 when $n = 64$] *In the group $\mathbb{F}^*_{2^{64}}$ the following choices for parameters provide unique representations:*

(1) $\alpha_1 = 2$ *and* $[-2^{62} \mathinner{..} 2^{62}]$.

(2) $\alpha_1, \alpha_2 = 2, 3$ *and* $[-2^{51} \mathinner{..} 2^{51}] \times [-2^{10} \mathinner{..} 2^{10}]$.

(3) $\alpha_1, \alpha_2, \alpha_3 = 2, 3, 11$ *and* $[-2^{44} \mathinner{..} 2^{44}] \times [-2^7 \mathinner{..} 2^7] \times [-2^7 \mathinner{..} 2^7]$. □

This time 2, 3, 7 does *not* work as a list of bases, even with a small set of allowed indices like $[1 \mathinner{..} 64] \times \{0, 1, 2\} \times \{0, 1, 2\}$, due to the fact that $2^{64} = 3^2 \cdot 7$ in this group. Machine-assisted verification seems essential here; a relation like that just given is found immediately when computing the possible values for $\Delta_b \log_2 3 + \Delta_c \log_2 7 \pmod{2^{64}-1}$ but it might not otherwise be anticipated.

5 Security of XE

The following result quantifies the security of the XE construction.

Theorem 1. [Security of XE] *Fix $n \geq 1$ and let $\alpha_1, \ldots, \alpha_k \in \mathbb{F}^*_{2^n}$ be base elements and let $\mathbb{I}_1 \times \cdots \times \mathbb{I}_k$ be allowed indices such that these parameters provide unique representations. Fix a blockcipher $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and let $\widetilde{E} = \mathsf{XE}[E, \alpha_1^{\mathbb{I}_1} \cdots \alpha_k^{\mathbb{I}_k}]$. Then $\mathbf{Adv}^{\widetilde{\mathrm{prp}}}_{\widetilde{E}}(t,q) \leq \mathbf{Adv}^{\mathrm{prp}}_{E}(t', 2q) + \frac{4.5\, q^2}{2^n}$ where $t' = t + ckn(q+1)$ for some absolute constant c.* □

In English, the XE construction promotes a CPA-secure blockcipher to a CPA-secure tweakable blockcipher, assuming that the chosen base elements and range of allowed indices provide unique representations. The proof is in [14].

6 Security of XEX

Some added care is needed to address the security of XEX. Suppose, to be concrete, that we are looking at $\mathsf{XEX}[E, 2^{\mathbb{I}}]$ and $\mathbb{I} = [0 \mathinner{..} 2^{n-2}]$. Let the adversary ask a deciphering query with ciphertext $C = 0^n$ and tweak $(0^n, 0)$. If the adversary has a construction-based deciphering oracle then it will get a response of $M = \widetilde{D}_K^{0^n\, 0}(0^n) = D_K(\Delta) \oplus \Delta = D_K(\mathsf{N}) \oplus \mathsf{N} = 0^n \oplus \mathsf{N} = \mathsf{N}$, where $\mathsf{N} = E_K(0^n) = \Delta$. This allows the adversary to defeat the CCA-security. For example, enciphering $2M = 2\mathsf{N}$ with a tweak of $(0^n, 1)$ and enciphering $4M = 4\mathsf{N}$ with a tweak of $(0^n, 2)$ will give identical results (if the adversary has the construction-based enciphering oracle). Corresponding to this attack we exclude any tweak $(N, i_1, \ldots, i_k)$ for which $(i_1, \ldots, i_k)$ is a representative of 1—that is, any tweak $(N, i_1, \ldots, i_k)$ for which $\alpha_1^{i_1} \ldots \alpha_k^{i_k} = 1$. In particular, this condition excludes any tweak $(N, 0, \ldots, 0)$. The proof of the following is omitted, as Theorem 3 will be more general.

Theorem 2 (Security of XEX). *Fix $n \geq 1$ and let $\alpha_1, \ldots, \alpha_k \in \mathbb{F}^*_{2^n}$ be base elements and let $\mathbb{I}_1 \times \cdots \times \mathbb{I}_k$ be allowed indices such that these parameters provide unique representations. Assume $\alpha_1^{i_1} \cdots \alpha_k^{i_k} \neq 1$ for all $(i_1, \ldots, i_k) \in \mathbb{I}_1 \times \cdots \times \mathbb{I}_k$. Fix a blockcipher $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and let $\widetilde{E} = \mathsf{XEX}[E, \alpha_1^{\mathbb{I}_1} \cdots \alpha_k^{\mathbb{I}_k}]$. Then $\mathbf{Adv}^{\pm\widetilde{\mathrm{prp}}}_{\widetilde{E}}(t,q) \leq \mathbf{Adv}^{\pm\mathrm{prp}}_{E}(t', 2q) + \frac{9.5\, q^2}{2^n}$ where $t' = t + ckn(q+1)$ for some absolute constant c.* □

7 An Almost-Free Alternative to Key Separation

When combining two blockcipher-based cryptographic mechanisms into a composite mechanism, it is, in general, essential to use two different keys. Either these two keys together comprise the key for the joint mechanism, or else each key is obtained from an underlying one by a key-derivation technique. The first possibility increases the key length in the composite mechanism while the second involves extra computation at key setup. Both possibilities incur the inefficiency of blockcipher re-keying when the combined mode runs. For all of these reasons, some new "composite" modes of operation have gone to considerable trouble in order to make do (for their particular context) with a *single* blockcipher key. Examples include EAX, CCM, and OĊB [3, 13, 17]. Using a single key complicates proofs—when the mechanism works at all—because one can no longer reason about generically combining lower-level mechanisms.

Tweakable blockciphers open up a different possibility: the same underlying key is used across the different mechanisms that are being combined, but one arranges that the tweaks are disjoint across different mechanisms. In this way one retains the modularity of design and analysis associated to using separate keys—one reasons in terms of generic composition—yet one can instantiate in a way that avoids having extra key material or doing extra key setups. Because the tweak space for XE and XEX is a Cartesian product of ranges of integers, it is easy, for these constructions, to separate the different tweaks.

8 Combining XE and XEX

Some blockcipher-based constructions need CCA-security in some places and CPA-security in other places. One could assume CCA-security throughout, later instantiating all blockcipher calls with a CCA-secure construction, but it might be better to use a CPA-secure construction where sufficient and a CCA-secure one where necessary. Regardless of subsequent instantiation, it is good to be able to talk, formally, about *where* in a construction one needs *what* assumption.

To formalize where in a construction one is demanding what, we *tag* each blockcipher call with an extra bit. We say that a tweakable blockcipher $\widetilde{E}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ is *tagged* if $\mathcal{T} = \{0,1\} \times \mathcal{T}^*$ for some nonempty set $\mathcal{T}^*$. Think of $\mathcal{T}^*$, the *effective tweak space*, as the tweak space actually used by the mode. The extra bit indicates what is demanded for each tweak. A first bit of 0 indicates a demand of CPA security, and 1 indicates a demand for CCA security. For a given $T \in \mathcal{T}$ one should be asking for one or the other.

An adversary A launching an attack on a tagged blockcipher is given two oracles, $e(\cdot,\cdot)$ and $d(\cdot,\cdot)$, where the second oracle computes the inverse of the first (meaning $d(T,Y)$ is the unique X such that $e(T,X) = Y$). The adversary must respect the semantics of the tags, meaning that the adversary may not make any query $d(T,Y)$ where the first component of T is 0, and if the adversary makes an oracle query with a tweak $(b,\ T^*)$ then it may make no subsequent query with a tweak $(1-b,\ T^*)$. As always, we insist that there be no pointless queries: an adversary may not repeat an $e(T,X)$ query or a $d(T,Y)$ query, and it may not ask $d(T,Y)$ after having learned $Y = e(T,X)$, nor ask $e(T,X)$ after having learned $X = d(T,Y)$. The definition for security is now as follows.

Definition 5 (Security of a tagged, tweakable blockcipher). *Let* $\widetilde{E}\colon \mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ *be a tagged, tweakable blockcipher and let* A *be an adversary. Then* $\mathbf{Adv}^{[\pm]\widetilde{\mathrm{prp}}}_{\widetilde{E}}(A)$ *is defined as* $\Pr[K \xleftarrow{\$} \mathcal{K} :\ A^{\widetilde{E}_K(\cdot,\cdot)\,\widetilde{D}_K(\cdot,\cdot)} \Rightarrow 1] - \Pr[\pi \xleftarrow{\$} \mathrm{Perm}(\mathcal{T}, n) :\ A^{\pi(\cdot,\cdot)\,\pi^{-1}(\cdot,\cdot)} \Rightarrow 1]$ □

Naturally $\widetilde{D}$, above, is the inverse of $\widetilde{E}$. Security in the $\widetilde{\mathrm{prp}}$-sense and security in the $\pm\widetilde{\mathrm{prp}}$-sense are special cases of security in the $[\pm]\widetilde{\mathrm{prp}}$ sense (but for the enlarged tweak space).

If we combine XE and XEX using our tagging convention we get the tagged, tweakable blockcipher XEX*.

Definition 6 (XEX*). *Let E: $\mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher, let $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_{2^n}^*$, and let $\mathbb{I}_1, \ldots, \mathbb{I}_k \subseteq \mathbb{Z}$. Then $\widetilde{E} = XEX^*[E,\ \alpha_1^{\mathbb{I}_1} \cdots \alpha_k^{\mathbb{I}_k}]$ is the tweakable blockcipher $\widetilde{E}$: $\mathcal{K} \times (\{0,1\} \times \{0,1\}^n \times \mathbb{I}_1 \cdots \times \mathbb{I}_k) \times \{0,1\}^n \to \{0,1\}^n$ defined by $\widetilde{E}_K^{0\,N\,i_1 \ldots i_k}(M) = E_K(M \oplus \Delta)$ and $\widetilde{E}_K^{1\,N\,i_1 \ldots i_k}(M) = E_K(M \oplus \Delta) \oplus \Delta$ where $\Delta = \alpha_1^{i_1} \alpha_2^{i_2} \cdots \alpha_k^{i_k}\,\mathsf{N}$ and $\mathsf{N} = E_K(N)$.* $\square$

9 Security of the Combined Construction

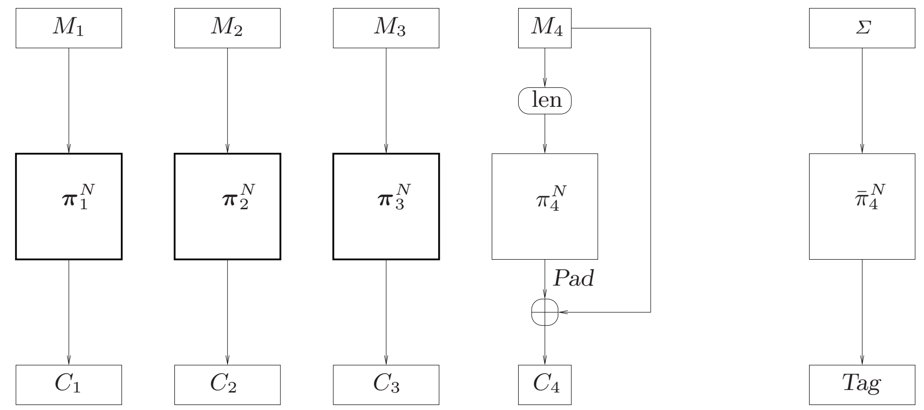
We now specify the security of the XEX* construction. The result encompasses that XE is $\widetilde{\mathrm{prp}}$-secure and XEX is $\pm\widetilde{\mathrm{prp}}$-secure. The proof is in [14].

Theorem 3 (Security of XEX*). *Fix $n \geq 1$ and let $\alpha_1, \ldots, \alpha_k \in \mathbb{F}_{2^n}^*$ be base elements and let $\mathbb{I}_1 \times \cdots \times \mathbb{I}_k$ be allowed indices such that these parameters provide unique representations and such that $\alpha_1^{i_1} \cdots \alpha_k^{i_k} \neq 1$ for all $(i_1, \ldots, i_k) \in \mathbb{I}_1 \times \cdots \times \mathbb{I}_k$. Fix a blockcipher E: $\mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and let $\widetilde{E} = XEX^*[E, \alpha_1^{\mathbb{I}_1} \cdots \alpha_k^{\mathbb{I}_k}]$. Then $\mathbf{Adv}_{\widetilde{E}}^{[\pm]\widetilde{\mathrm{prp}}}(t, q) \leq \mathbf{Adv}_{E}^{\pm\mathrm{prp}}(t', 2q) + \frac{9.5\,q^2}{2^n}$ where $t' = t + ckn(q+1)$ for some absolute constant c.* $\square$

10 The OCB1 Authenticated-Encryption Scheme

We recast OCB [15] to use a tweakable blockcipher instead of a conventional blockcipher. Liskov, Rivest, and Wagner first did this [10], but our formulation is different from theirs. First, guided by what we have done so far, we choose a tweak space of $\mathcal{T} = \{0,1\} \times \{0,1\}^n \times [1\,..\,2^{n/2}] \times \{0,1\}$. The first bit of the tweak is the tag; the effective tweak space is $\mathcal{T}^* = \{0,1\}^n \times [1\,..\,2^{n/2}] \times \{0,1\}$. Second, we want tweaks to increase monotonically, and so we switch the "special" processing done in OCB from the penultimate block to the final block. The resulting algorithm is shown in Fig. 1. Algorithm OCB1 is parameterized by a tweakable blockcipher $\widetilde{E}$: $K \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ and a number $\tau \in [0\,..\,n]$. For clarity, we write $\boldsymbol{\pi}_i^N$ for $\widetilde{E}_K^{1\,N\,i\,0}$ and π_i^N for $\widetilde{E}_K^{0\,N\,i\,0}$ and $\bar{\pi}_i^N$ for $\widetilde{E}_K^{0\,N\,i\,1}$.

The security of OCB1[Perm$(\mathcal{T}, n)$] is much simpler to prove than the security of OCB[Perm(n)]. (Liskov, Rivest, and Wagner [10] had made the same point for their tweakable-blockcipher variant of OCB.) To state the result we give a couple of definitions from [15]. For privacy of a nonce-based encryption scheme $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ we use the notion of indistinguishability-from-random-strings, which defines $\mathbf{Adv}_{\Pi}^{\mathrm{priv}}(A)$ as $\Pr[K \xleftarrow{\$} \mathcal{K}:\ A^{\mathcal{E}_K(\cdot,\cdot)} \Rightarrow 1] - \Pr[A^{\$(\cdot,\cdot)} \Rightarrow 1]$. Here $\$(\cdot,\cdot)$ is an oracle that, on input (N, M), returns $|M|$ random bits. The adversary is not allowed to repeat a nonce N. For authenticity we use the nonce-based notion of integrity of ciphertexts: the adversary is given an encryption



Algorithm OCB1.Encrypt$_K^N$ (M)
Partition M into $M[1] \cdots M[m]$
for $i \in [1 \,..\, m-1]$ **do** $C[i] \leftarrow \boldsymbol{\pi}_i^N(M[i])$
$\mathit{Pad} \leftarrow \pi_m^N(\mathrm{len}(M[m]))$
$C[m] \leftarrow M[m] \oplus \mathit{Pad}$
$C \leftarrow C[1] \cdots C[m]$
$\Sigma \leftarrow M[1] \oplus \cdots \oplus M[m-1] \oplus$
$\quad C[m]0^* \oplus \mathit{Pad}$
$\mathit{Tag} \leftarrow \bar{\pi}_m^N(\Sigma)$
$T \leftarrow \mathit{Tag}$ [first τ bits]
return $\mathcal{C} \leftarrow C \,\|\, T$

Algorithm OCB1.Decrypt$_K^N$ $(\mathcal{C})$
Partition $\mathcal{C}$ into $C[1] \cdots C[m]\, T$
for $i \in [1 \,..\, m-1]$ **do** $M[i] \leftarrow \left(\boldsymbol{\pi}_i^N\right)^{-1}(C[i])$
$\mathit{Pad} \leftarrow \pi_m^N(\mathrm{len}(C[m]))$
$M[m] \leftarrow C[m] \oplus \mathit{Pad}$
$M \leftarrow M[1] \cdots M[m]$
$\Sigma \leftarrow M[1] \oplus \cdots \oplus M[m-1] \oplus C[m]0^* \oplus \mathit{Pad}$
$\mathit{Tag} \leftarrow \pi_m^N(\Sigma)$
$T' \leftarrow \mathit{Tag}$ [first τ bits]
if $T = T'$ **then return** M
else return INVALID

Fig. 1. OCB1$[\widetilde{E}, \tau]$ with a tweakable blockcipher $\widetilde{E}$: $\mathcal{K} \times \mathcal{T} \times \{0,1\}^n \rightarrow \{0,1\}^n$ and tweak space $\mathcal{T} = \{0,1\} \times \{0,1\}^n \times [1 \,..\, 2^{n/2}] \times \{0,1\}$ and tag length $\tau \in [0 \,..\, n]$. We write $\boldsymbol{\pi}_i^N$ and π_i^N and $\bar{\pi}_i^N$ for $\widetilde{E}_K^{1\,N\,i\,0}$ and $\widetilde{E}_K^{0\,N\,i\,0}$ and $\widetilde{E}_K^{0\,N\,i\,1}$.

oracle $\mathcal{E}_K(\cdot,\cdot)$ and is said to *forge* if it outputs an $(N, \mathcal{C})$ that is valid and $\mathcal{C}$ was not the result of any prior (N, M) query. The adversary is not allowed to repeat a nonce N while it queries its encryption oracle. We write $\mathbf{Adv}_{\Pi}^{\mathrm{auth}}(A)$ for $\Pr[K \xleftarrow{\$} \mathcal{K} :\ A^{\mathcal{E}_K(\cdot,\cdot)} \text{ forges }]$. We have the following theorem for the information-theoretic security of OCB1. The proof is in [14].

Theorem 4 (OCB1 with an ideal tweakable blockcipher). *Fix $n \geq 1$, $\tau \in [0 \,..\, n]$, and $\mathcal{T} = \{0,1\} \times \{0,1\}^n \times [1 \,..\, 2^{n/2}] \times \{0,1\}$. Let A be an adversary. Then* $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{OCB1}[\mathrm{Perm}(\mathcal{T},n),\tau]}(A) = 0$ *and* $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{OCB1}[\mathrm{Perm}(\mathcal{T},n),\tau]}(A) \leq 2^{n-\tau}/(2^n - 1)$. $\square$

Note that the authenticity bound is close to $2^{-\tau}$; in particular, $2^{n-\tau}/(2^n-1) \leq 1/(2^\tau - 1)$ for all $\tau \geq 2$. The bounds do not degrade with the number of queries asked by the adversary, the length of these queries, or the time the adversary runs. For the complexity-theoretic analog we have the following.

Algorithm OCB1.Encrypt$_K^N$ (M)	**Algorithm** OCB1.Decrypt$_K^N$ $(\mathcal{C})$
Partition M into $M[1]\cdots M[m]$	Partition $\mathcal{C}$ into $C[1]\cdots C[m]\,T$
$\Delta \leftarrow 2\,E_K(N)$	$\Delta \leftarrow 2\,E_K(N)$
$\Sigma \leftarrow 0^n$	$\Sigma \leftarrow 0^n$
for $i \in [1\mathinner{..}m-1]$ **do**	**for** $i \in [1\mathinner{..}m-1]$ **do**
$\quad C[i] \leftarrow E_K(M[i] \oplus \Delta) \oplus \Delta$	$\quad M[i] \leftarrow E_K^{-1}(C[i] \oplus \Delta) \oplus \Delta$
$\quad \Delta \leftarrow 2\Delta$	$\quad \Delta \leftarrow 2\Delta$
$\quad \Sigma \leftarrow \Sigma \oplus M[i]$	$\quad \Sigma \leftarrow \Sigma \oplus M[i]$
$\mathit{Pad} \leftarrow E_K(\mathrm{len}(M[m]) \oplus \Delta)$	$\mathit{Pad} \leftarrow E_K(\mathrm{len}(C[m]) \oplus \Delta)$
$C[m] \leftarrow M[m] \oplus \mathit{Pad}$	$M[m] \leftarrow C[m] \oplus \mathit{Pad}$
$C \leftarrow C[1]\cdots C[m]$	$M \leftarrow M[1]\cdots M[m]$
$\Sigma \leftarrow \Sigma \oplus C[m]0^* \oplus \mathit{Pad}$	$\Sigma \leftarrow \Sigma \oplus C[m]0^* \oplus \mathit{Pad}$
$\Delta \leftarrow 3\Delta$	$\Delta \leftarrow 3\Delta$
$\mathit{Tag} \leftarrow E_K(\Sigma \oplus \Delta)$	$\mathit{Tag} \leftarrow E_K(\Sigma \oplus \Delta)$
$T \leftarrow \mathit{Tag}$ [first τ bits]	$T' \leftarrow \mathit{Tag}$ [first τ bits]
return $\mathcal{C} \leftarrow C \,\|\, T$	**if** $T = T'$ **then return** M
	else return INVALID

Fig. 2. OCB1$[E,\tau]$ with a conventional blockcipher E: $\mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ and a tag length $\tau \in [0\mathinner{..}n]$. This coincides with OCB1$[\widetilde{E},\tau]$ where $\widetilde{E} = \mathrm{XEX}[E, 2^{[1\mathinner{..}2^{n/2}]}3^{\{0,1\}}]$.

Corollary 1 (OCB1 with a tweakable blockcipher). *Fix $n \geq 1$, $\tau \in [0\mathinner{..}n]$, $\mathcal{T} = \{0,1\} \times \{0,1\}^n \times [1\mathinner{..}2^{n/2}] \times \{0,1\}$, and $\widetilde{E}$: $\mathcal{K} \times \mathcal{T} \times \{0,1\}^n \to \{0,1\}^n$ a tagged, tweakable blockcipher. Then $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{OCB1}[\widetilde{E},\tau]}(t,\sigma) \leq \mathbf{Adv}^{\widetilde{\mathrm{prp}}}_{\widetilde{E}}(t',\sigma)$ and $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{OCB1}[\widetilde{E},\tau]}(t,\sigma) \leq \mathbf{Adv}^{[\pm]\widetilde{\mathrm{prp}}}_{\widetilde{E}}(t',\sigma) + 2^{n-\tau}/(2^n-1)$, where $t' = t + cn\sigma$ for some absolute constant c.* □

The proof requires CPA-security for privacy but authenticity uses the notion that combines CPA- and CCA-security (Definition 5). It is here that one has formalized the intuition that the first $m-1$ tweakable-blockcipher calls to OCB1 need to be CCA-secure but the last two calls need only be CPA-secure.

To realize OCB1 with a conventional blockcipher E: $\mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$, use XEX*, instantiating OCB1$[\widetilde{E},\tau]$ by way of $\widetilde{E} = \mathrm{XEX}^*[E,\, 2^{\mathbb{I}}3^{\mathbb{J}}]$ where $\mathbb{I} = [1\mathinner{..}2^{n/2}]$ and $\mathbb{J} = \{0,1\}$. Overloading the notation, we write this scheme as OCB1$[E,\tau]$. The method is rewritten in Fig. 2.

Corollary 2 (OCB1 with a blockcipher). *Fix $n \geq 1$ and $\tau \in [0\mathinner{..}n]$. Assume that $2, 3$ provide unique representations on $[1\mathinner{..}2^{n/2}] \times \{0,1\}$ and $2^i3^j \neq 1$ for all $(i,j) \in [1\mathinner{..}2^{n/2}] \times \{0,1\}$. Let E: $\mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a blockcipher. Then*

– $\mathbf{Adv}^{\mathrm{priv}}_{\mathrm{OCB1}[E,\tau]}(t,\sigma) \;\leq\; \mathbf{Adv}^{\mathrm{prp}}_{E}(t',2\sigma) + \dfrac{4.5\sigma^2}{2^n}$ *and*

– $\mathbf{Adv}^{\mathrm{auth}}_{\mathrm{OCB1}[E,\tau]}(t,\sigma) \;\leq\; \mathbf{Adv}^{\pm\mathrm{prp}}_{E}(t',2\sigma) + \dfrac{9.5\sigma^2}{2^n} + \dfrac{2^{n-\tau}}{2^n-1}$

where $t' = t + cn\sigma$ for some absolute constant c. □

Propositions 1 and 2 establish that $n = 128$ and $n = 64$ satisfy the requirement for unique representations. They also guarantee that there is no representative of 1 within $[1 \mathinner{..} 2^{n/2}] \times \{0, 1\}$. To see this, note that the propositions imply that $(0, 0)$ is the only representative for 1 within a space $\mathbb{I}_1 \times \mathbb{I}_2$ that includes $[1 \mathinner{..} 2^{n/2}] \times \{0, 1\}$, and so there can be *no* representative of 1 within a subspace of $\mathbb{I}_1 \times \mathbb{I}_2$ that excludes $(0, 0)$.

Blockcipher-based OCB1 is more efficient than OCB. With OCB one expects to use preprocessing to compute a value $L = E_K(0^n)$ and a collection of $2^i L$-values. This is gone in OCB1; preprocessing is not useful there beyond setting up the underlying blockcipher key. Beyond this, with OCB processing the j^{th} block involved xoring into the current offset a value $L(i) = 2^i L$ where $i = \mathsf{ntz}(j)$ was the number of trailing zero-bits in the index j. In the absence of preprocessing, offset-calculations were not constant time. This too is gone.

The previous paragraph notwithstanding, the time difference or chip-area difference between optimized implementations of OCB and OCB1 will be small, since the overhead of OCB over a mode like CBC was already small. The larger gain is that the mode is simpler to understand, implement, and prove correct.

11 The PMAC1 Message Authentication Code

As with OCB, one can recast PMAC [4] to use a tweakable blockcipher and, having done so, one can instantiate the tweakable blockcipher, this time with the XE construction. The resulting algorithm, PMAC1, is simpler and more efficient than PMAC. In the latter construction one had to xor into the current offset a value $L(i) = 2^i L$ where i was the number of trailing zero-bits in the current block index j. This is gone in PMAC1, and an implementation no longer needs to concern itself with Gray codes, precomputing $L(i)$-values, or finding the most efficient way to bring in the right $L(i)$ value. Details are in [14].

To make an authenticated encryption scheme that handles associated-data, combine OCB1 and PMAC1 [13]. Encrypt message M under key K, nonce N, and header H by $\text{OCB1}.\mathsf{Encrypt}_K^N(M) \oplus \text{PMAC1}_K(H)$ where the $\oplus$ xors into the end. Omit the $\oplus\ \text{PMAC1}_K(H)$ if $H = \varepsilon$. We call this scheme AEM.

12 Comments

Under the approach suggested by this paper, to get good efficiency for a design that uses a tweakable-blockcipher, the designer must accept certain design rules. In particular, the tweak space needs to look like $\{0, 1\}^n \times \mathbb{BIG} \times \mathbb{SMALL}$ for appropriate sets $\mathbb{BIG}$ and $\mathbb{SMALL}$, and one needs to arrange that most tweaks be obtained by incrementing the prior one. It is a thesis implicit in this work that these restrictions are not overly severe.

Besides simplifying the design and proof for OCB and PMAC, we have improved their efficiency. The improvement are not large (the modes were already highly efficient), but performance improvements, of any size, was not a benefit formerly envisaged as flowing from the tweakable-blockcipher abstraction.

Somewhat strangely, our constructions depend on the relative *easiness* of computing discrete logarithms. I know of no other example where one needs to compute discrete logs in order to design or verify a mode of operation.

I end this paper by acknowledging that everyone writes *block cipher*, not *blockcipher*. Still, the time has come to spell this word solid. I invite you to join me.

Acknowledgments

Thanks to David Wagner for pointing out an oversight in an early draft. Useful comments were also received from John Black and the anonymous referees.

This research was supported by NSF 0208842 and by a gift from Cisco System. Thanks to the NSF (particularly Carl Landwehr) and to Cisco (particularly David McGrew) for their kind support of my research.

References

1. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption: Analysis of the DES modes of operation. *Symposium on Foundations of Computer Science, FOCS '97*, IEEE Computer Society, pp. 394–403, 1997.
2. M. Bellare, J. Kilian, and P. Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, vol. 61, no. 3, Dec 2000. Earlier version in *CRYPTO '94.*
3. M. Bellare, P. Rogaway, and D. Wagner. The EAX Mode of operation. *Fast Software Encryption, FSE 2004.* Lecture Notes in Computer Science, vol. 3017, Springer-Verlag, pp. 389–407, 2004.
4. J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. *Advances in Cryptology — Eurocrypt '02.* Lecture Notes in Computer Science, vol. 2332, Springer-Verlag, pp. 384–397, 2002.
5. V. Gligor and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. *Fast Software Encryption, FSE 2001.* Lecture Notes in Computer Science, vol. 2355, Springer-Verlag, pp. 92–108, 2001.
6. S. Goldwasser and S. Micali. Probabilistic encryption. *Journal of Computer and System Sciences*, vol. 28, April 1984, pp. 270–299.
7. S. Halevi and P. Rogaway. A parallelizable enciphering mode. *Topics in Cryptology — CT-RSA 2004.* Lecture Notes in Computer Science, vol. 2964, Springer-Verlag, pp. 292–304, 2004.
8. J. Kilian and P. Rogaway. How to protect DES against exhaustive key search (an analysis of DESX). *J. of Cryptology*, vol. 14, no. 1, pp. 17–35, 2001.
9. C. Jutla. Encryption modes with almost free message integrity. *Advances in Cryptology — EUROCRYPT 2001.* Lecture Notes in Computer Science, vol. 2045, Springer-Verlag, pp. 529–544, 2001.
10. M. Liskov, R. Rivest, and D. Wagner. Tweakable block ciphers. *Advances in Cryptology — CRYPTO '02.* Lecture Notes in Computer Science, vol. 2442, Springer-Verlag, pp. 31–46, 2002.

11. S. Pohlig and M. Hellman. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. *IEEE Transactions on Information Theory*, vol 24, pp. 106–110, 1978.
12. J. Pollard. Monte Carlo methods for index computation (mod p). *Mathematics of Computation*, vol. 32, pp. 918–924, 1978.
13. P. Rogaway. Authenticated-encryption with associated-data. *ACM Conference on Computer and Communications Security 2002, CCS 2002.* ACM Press, pp. 98–107, 2002.
14. P. Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. Manuscript, 2004. Full version of this paper, available from the author's web page.
15. P. Rogaway, M. Bellare, and J. Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security*, vol. 6, no. 3, pp. 365–403, 2003. Earlier version, with T. Krovetz, in *CCS 2001*.
16. R. Schroeppel. The hasty pudding cipher. AES candidate submitted to NIST, 1998.
17. D. Whiting, R. Housley, and N. Ferguson. Counter with CBC-MAC (CCM). Network Working Group RFC 3610. The Internet Society, September 2003.

A Tweakable Blockciphers Implicit in Prior Work

When tweaks increase in sequence, the most efficient constructions formerly known for a tweakable blockcipher are those implicit in earlier modes [4, 5, 9, 15], recast in view of Liskov, Rivest, and Wagner [10]. In particular:

– Jutla [9] might be seen as suggesting a construction (among others) of $\widetilde{E}\colon (\mathcal{K}\times\mathcal{K}')\times(\{0,1\}^n\times\mathbb{Z}_p^+)\times\{0,1\}^n\to\{0,1\}^n$ by way of $\widetilde{E}^{N,i}_{KK'}(X) = E_K(X\oplus\Delta)\oplus\Delta$ where $\Delta = i\ell \bmod p$ and $\ell = E_{K'}(N)$ and p is the largest prime less than 2^n.
– Gligor and Donescu [5] might be seen as suggesting constructions like $\widetilde{E}\colon (\mathcal{K}\times\{0,1\}^n)\times[1\mathinner{..}2^n-1]\to\{0,1\}^n$ by $\widetilde{E}^i_{K,r}(X) = E_K(X+\delta)$ where $\delta = ir$ and addition is done modulo 2^n.
– Rogaway, Bellare, and Black [15] might be seen as implicitly suggesting making a tweakable blockcipher $\widetilde{E}\colon \mathcal{K}\times(\{0,1\}^n\times[0\mathinner{..}2^{n-2}])\times\{0,1\}^n\to\{0,1\}^n$ from an ordinary blockcipher $E\colon\mathcal{K}\times\{0,1\}^n\to\{0,1\}^n$ by way of $\widetilde{E}^{N,i}_K(X) = E_K(X\oplus\Delta)\oplus\Delta$ where $\Delta = \gamma_i L\oplus R$ and $L = E_K(0^n)$ and $R = E_K(N\oplus L)$ and γ_i is the i-th Gray-code coefficient.
– Black and Rogaway [4] might be seen as making $\widetilde{E}\colon \mathcal{K}\times[0\mathinner{..}2^{n-2}]\times\{0,1\}^n\to\{0,1\}^n$ out of $E\colon\mathcal{K}\times\{0,1\}^n\to\{0,1\}^n$ by $\widetilde{E}^i_K(X) = E_K(X\oplus\Delta)$ where $\Delta = \gamma_i L$ and $L = E_K(0^n)$ and γ_i is as before.
– The last two definitions ignore the "special" treatment afforded to blocks modified by xoring in $2^{-1}L$. The implicit intent [4, 15] was to use this mechanism to enlarge the tweak space by one bit, effectively taking the cross product with $\{0,1\}$.