Tweakable Block Ciphers

Moses Liskov[1], Ronald L. Rivest[1], and David Wagner[2]

[1] Laboratory for Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA
mliskov@theory.lcs.mit.edu, rivest@mit.edu
[2] University of California Berkeley
Soda Hall
Berkeley, CA 94720, USA
daw@cs.berkeley.edu

Abstract. We propose a new cryptographic primitive, the "*tweakable block cipher.*" Such a cipher has not only the usual inputs—message and cryptographic key—but also a third input, the "tweak." The tweak serves much the same purpose that an initialization vector does for CBC mode or that a nonce does for OCB mode. Our proposal thus brings this feature down to the primitive block-cipher level, instead of incorporating it only at the higher modes-of-operation levels. We suggest that (1) tweakable block ciphers are easy to design, (2) the extra cost of making a block cipher "tweakable" is small, and (3) it is easier to design and prove modes of operation based on tweakable block ciphers.

Keywords: block ciphers, tweakable block ciphers, initialization vector, modes of operation

1 Introduction

A conventional block cipher takes two inputs—a *key* $K \in \{0,1\}^k$ and a *message* (or *plaintext*) $M \in \{0,1\}^n$—and produces a single output—a *ciphertext* $C \in \{0,1\}^n$. The signature for a block cipher is thus (see Figure 1(a)):

$$E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n \ . \tag{1}$$

On the other hand, the corresponding operators for variable-length encryption have a different signature. These operators are usually defined as "modes of operation" for a block cipher, but they may also be viewed abstractly as another set of encryption operators. They take as input a *key* $K \in \{0,1\}^k$, an *initialization vector* (or *nonce*) $V \in \{0,1\}^v$, and a *message* $M \in \{0,1\}^*$ of arbitrary length, and produce as output a *ciphertext* $C \in \{0,1\}^*$. The signature for a typical encryption mode is thus:

$$\mathcal{E} : \{0,1\}^k \times \{0,1\}^v \times \{0,1\}^* \rightarrow \{0,1\}^* \ .$$

Block ciphers (pseudorandom permutations) are inherently deterministic: every encryption of a given message with a given key will be the same. Many modes of operation and other applications using block ciphers have nonetheless a requirement for "essentially different" instances of the block cipher in order to prevent attacks that operate by, say, permuting blocks of the input. Attempts to resolve the conflict between keeping the same key for efficiency and yet achieving variability often results in a design that uses a fixed key, but which attempts to achieve variability by manipulating the input before encryption, the output after encryption, or both. Such designs seem inelegant—they are attempting to solve a problem with a primitive (a basic block cipher) that is not well suited for the problem at hand. Better to rethink what primitives are really wanted for such a problem.

This paper proposes to revise the signature of a block cipher so that it contains a notion of variability as well. The revised primitive operation, which we call a *tweakable block cipher*, has the signature:

$$\widetilde{E} : \{0,1\}^k \times \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n \; . \qquad (2)$$

For this operator, we call the new (second) input a "tweak" rather than a "nonce" or "initialization vector," but the intent is similar. A tweakable block cipher thus takes three inputs—a *key* $K \in \{0,1\}^k$, a *tweak* $T \in \{0,1\}^t$, and a *message* (or *plaintext*) $M \in \{0,1\}^n$—and produces as output a *ciphertext* $C \in \{0,1\}^n$ (see Figure 1(b)).

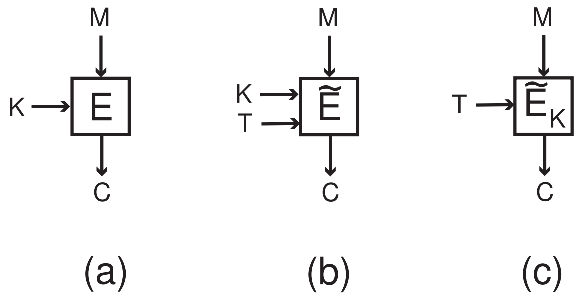

Fig. 1. (a) *Standard block cipher* encrypts a message M under control of a key K to yield a ciphertext C. (b) *Tweakable block cipher* encrypts a message M under control of not only a key K but also a "tweak" T to yield a ciphertext C. The "tweak" can be changed quickly, and can even be public. (c) Another way of representing a tweakable block cipher; here the key K shown inside the box.

In designing a tweakable block cipher, we have certain goals. First of all, obviously, we want any tweakable block ciphers we design to be as efficient as possible (just as with any scheme). Specifically, a tweakable block cipher should have the property that changing the tweak should be less costly than changing the key. Many block ciphers have the property that changing the encryption key is relatively expensive, since a "key setup" operation needs to be performed. In contrast, changing the tweak should be cheaper. [1]

A tweakable block cipher should also be secure, meaning that even if an adversary has control of the tweak input, we want the tweakable block cipher to remain secure. We'll define what this means more precisely later on. But intuitively, *each fixed setting of the tweak gives rise to a different, apparently independent, family of standard block cipher encryption operators.* We wish to carefully distinguish between the function of the *key*, which is to provide uncertainty to the adversary, and the role of the *tweak*, which is to provide variability. The tweak is not intended to provide additional uncertainty to an adversary. Keeping the tweak secret need not provide any greater cryptographic strength.

The point of this paper is to suggest that by cleanly separating the roles of cryptographic key (which provides uncertainty to the adversary) from that of tweak (which provides independent variability) we may have just the right tool for many cryptographic purposes.

1.1 Related Work

One motivating example for this introduction of tweakable block ciphers is the DESX construction introduced by Rivest (unpublished). The reason for introducing DESX was to cheaply provide additional key information for DES. The security of DESX has been analyzed by Kilian and Rogaway [10]; they show that DESX with n-bit inputs (and tweaks) and k-bit keys has an effective key-length of $k+n-1-\lg m$ where the adversary is limited to m oracle calls. In the DESX construction secret pre- and post-whitening values were added as additional key information.

Even and Mansour [8] have also investigated a similar construction where the inner encryption operator F is fixed and public, and encryption is performed by $E_{K_1K_2}(M) = K_2 \oplus F(K_1 \oplus M)$. They show (see also Daemen[7]) that the effective key length here is $n - \lg l - \lg m$ where the adversary is allowed to make l calls to the encryption/decryption oracles and m calls to an oracle for F or F^{-1}.

Similarly, if one looks at the internals of the recently proposed "offset codebook mode" (OCB mode) of Rogaway et al. [12], one sees DESX-like modules that may also be viewed as instances of a tweakable block ciphers. That is, the pre- and post-whitening operations are essentially there to provide distinct families of encryption operators, i.e. they are "tweaked."

[1] Some cryptographic modes of operation such as the Davies-Meyer hash function (see Menezes et al. [11, Section 9.40]) have fallen into disfavor because they have a feedback path into the key input of the block cipher. Since for many block ciphers it is relatively expensive to change the key, these modes of operation are relatively inefficient compared to similar modes that use the same key throughout. See, for example, the discussion by Rogaway et al. [12] explaining the design rationale for the OCB mode of operation, which uses the same cryptographic key throughout.

In a similar vein, Biham and Biryukov [4] suggest strengthening DES against exhaustive search by (among other things) applying a DESX-like construction to each of DES's S-boxes.

Finally, two block cipher proposals, the Hasty Pudding Cipher (HPC) [14] and the Mercy cipher [6] include an extra input for variability, called in their design specifications a "spice," a "randomiser," or a "diversification parameter." These proposals include a basic notion of what kind of security is needed for a block cipher with this extra input, but no formal notions or proofs are given.

1.2 Outline of this paper

In Section 2 we then discuss and formalize the notion of security for tweakable block ciphers. In Section 3 we suggest several ways of constructing tweakable block ciphers from existing block ciphers, and prove that the existence of tweakable block ciphers is equivalent to the existence of block ciphers. Then in Section 4 we suggest several new modes of operation utilizing tweakable block ciphers, and give simple proofs for some of them. Section 5 concludes with some discussion and open problems.

2 Definitions

The security of a block cipher E (e.g. parameterized as in equation (1)) can be quantified as $\mathrm{Sec}_E(q,t)$–the maximum advantage that an adversary can obtain when trying to distinguish $E(K,\cdot)$ (with a randomly chosen key K) from a random permutation $\Pi(\cdot)$, when allowed q queries to an unknown oracle (which is either $E(K,\cdot)$ or $\Pi(\cdot)$) and when allowed computation time t. This advantage is defined as the difference between the probability the adversary outputs 1 when given oracle access to E and the probability the same adversary outputs 1 when given oracle access to Π. A block cipher may be considered secure when $\mathrm{Sec}_E(q,t)$ is sufficiently small.

We may measure the security of a tweakable block cipher $\widetilde{E}$ (parameterized as in equation (2)) in a similar manner as the maximum advantage $\mathrm{Sec}_{\widetilde{E}}(q,t)$ an adversary can obtain when trying to distinguish $\widetilde{E}(\cdot,\cdot)$ from a "tweakable random permutation" $\widetilde{\Pi}(\cdot,\cdot)$ where $\widetilde{\Pi}$ is just a family of independent random permutations parametrized by T. That is, for each T, we have that $\widetilde{\Pi}(T,\cdot)$ is an independent randomly chosen permutation of the message space. Note that the adversary is allowed to choose both the message and tweak for each oracle call. A tweakable block cipher $\widetilde{E}$ may be considered secure when $\mathrm{Sec}_{\widetilde{E}}(q,t)$ is sufficiently small.

A tweakable block cipher should also be efficient: both encryption $\widetilde{E}_K(\cdot,\cdot)$ and decryption $\widetilde{D}_K(\cdot,\cdot)$ should be easy to compute.

2.1 Strong tweakable block ciphers

A stronger definition for a block cipher, $\mathrm{Sec}'_E(q,t)$, can be defined as the maximum advantage than an adversary can obtain when trying to distinguish the pair of oracles $E(K,\cdot), D(K,\cdot)$ from the pair Π, Π^{-1}, when allowed q queries and computation time t. This advantage is defined as the difference between the probability the adversary outputs 1 when given oracle access to E, D and the probability the same adversary outputs 1 when given oracle access to Π, Π^{-1}. A block cipher is considered "chosen-ciphertext" secure when $\mathrm{Sec}'_E(q,t)$ is sufficiently small.

Similarly, we define $\mathrm{Sec}'_{\widetilde{E}}(q,t)$ as the maximum advantage an adversary can obtain when trying to distinguish $\widetilde{E}_K(\cdot,\cdot), \widetilde{D}_K(\cdot,\cdot)$ from $\widetilde{\Pi}, \widetilde{\Pi}^{-1}$, when given q queries and t time. We say a tweakable block cipher is chosen-ciphertext secure when $\mathrm{Sec}'_{\widetilde{E}}(q,t)$ is sufficiently small, and we call such a secure tweakable block cipher a "strong tweakable block cipher."

3 Constructions

In this section we show that the existence of block ciphers and the existence of tweakable block ciphers are equivalent. One direction is easy: if we let $E_K(M) = \widetilde{E}_K(0^t, M)$, it is easy to see that if $\widetilde{E}$ is a secure tweakable block cipher then E must be a secure block cipher.

The other direction is more difficult. Some simple attempts to construct a tweakable block cipher from a block cipher fail.

For example, the DESX analogue:

$$\widetilde{E}_K((T_1,T_2),M) = E_K(M \oplus T_1) \oplus T_2$$

fails because an adversary can notice that flipping the same bits in both T_1 and m has no net effect.

Similarly, taking an ordinary block cipher and splitting its key into a key for the tweakable cipher and a tweak:

$$\widetilde{E}_K(T,M) = E_{K\|T}(M)$$

or xoring the tweak into the key:

$$\widetilde{E}_K(T,M) = E_{K\oplus T}(M)$$

need not yield secure tweakable block ciphers, since a block cipher need not depend on every bit of its key. (Biham's related-key attacks of Biham [3] would be relevant to this sort of design.)

The following theorem gives a construction that works.

Theorem 1. *Let*

$$\widetilde{E}_K(T,M) = E_K(T \oplus E_K(M)).$$

$\widetilde{E}$ *is a secure tweakable block cipher. More precisely,*

$$\mathrm{Sec}_{\widetilde{E}}(q,t) < \mathrm{Sec}_E(q,t) + \Theta(q^2/2^n) \ .$$

Proof. We assume that E has security function $\mathrm{Sec}_E(q,t)$ and assume that an adversary $A^?$ exists that achieves and advantage $\mathrm{Sec}_{\widetilde{E}}(q,t)$ when distinguishing $\widetilde{E}$ from a tweakable random permutation.

We have the following cases.

Case i: A can distinguish between $\widetilde{E}_K$ and H^1, where $H^1(T,M) = \Pi(T \oplus \Pi(M))$. If this is the case, we can use A to distinguish E from Π.

Case ii: A can distinguish between H^1 and H^2 where $H^2(T,M) = R(T \oplus R(M))$, where R is a random function. It is easy to see that the advantage in distinguishing a random function from a random permutation is $\Theta(q^2/2^n)$.

Case iii: A can distinguish H^2 from H^3, where $H^3(T,M) = R_2(T \oplus R_1(M))$, where R_1 and R_2 are random functions. Suppose $(T_1,M_1),\ldots,(T_q,M_q)$ are all the queries A makes to the oracle, and suppose no collisions of the following type happen: $T_i \oplus R(M_i) = M_j$. With no such collisions, H^2 cannot be distinguished from H^3 as the outer application of R takes place on a set of inputs disjoint from the inputs to the inner application of R, and so the outer outputs are independently random, just as the outputs of R_2 would be.

Furthermore, the probability of any such collisions occuring is $\Theta(q^2)/2^n$. What is the probability that (T_i,M_i) collides with any previous pair? If M_i is a new value then it is easy to see that the probability is at most $(i-1)/2^n$. What if M_i is not new? In this case, either (T_i,M_i) will collide or it won't, since all the random decisions have been made. However it is important to note that if no collisions have happened before, then every oracle response the adversary gets is just a new random value. Thus, the values the adversary gets are independent from the T's the adversary produces. Conversely, T_i must be independent from the distribution of R. Thus, even though T_i is not necessarily chosen randomly, no matter how the adversary picks T_i, it has a probability of at most $(i-1)/2^n$ of being one that causes a collision. Adding all these probabilities up, we see that the probability that any collision occurs is $\Theta(q^2)/2^n$.

Case iv: A can distinguish H^3 from H^4, where $H^4(T,M) = R(T,M)$, where R is a random function.

In order for there to be a difference between H^3 and H^4, the output of R must be constrained for two different input pairs. Thus, there must be a pair i,j such that $T_i \oplus R_1(M_i) = T_j \oplus R_1(M_j)$ for $i \neq j$. What is the probability that this happens for any given j? Well, if M_j is a new M, this will only happen with probability $(j-1)/2^n$. Now suppose that up through the jth query there have been no collisions. The adversary then receives purely random values back in response. Thus, since the outputs the adversary sees are independent of the queries, the queries must be independent of the values $T_i \oplus R_1(M_i)$.

If M_j is not a new value, but T_j has never been asked with M_j before, then the probability of a collision is at most $(j-2)/2^n$, since the only possible values to collide with are those where $M_i \neq M_j$. This critically relies on the observation that the adversary's queries are independent of the values so long as there have been no collisions. Thus, we can bound the total probability of collisions in the same way. The probability of distinguishing can be bounded by $q^2/2^n$ where q is the number of queries.

Case v: A can distinguish between H^4 and $\widetilde{\Pi}$. Note that $R(T, M)$ differs from $\widetilde{\Pi}(T, M)$ only in that for any given T, one will be a random function and the other will be a random permutation. Since random functions and random permutations are indistinguishable, this is impossible: we use a simple hybrid argument, providing permutations for more and more T.

Thus, we see that this construction only "degrades" $\mathrm{Sec}_E q, t$ by $\Theta(q^2/2^n)$ to obtain $\mathrm{Sec}_{\widetilde{E}}(q, t)$. □

Note that this construction has the nice property that changing the tweak is easy (no "key setup" required). Furthermore, we do not require a longer key than the block cipher did for the same level of security. However, the construction has an overall cost (running time) that is twice that of the underlying block cipher.

This completes our proof that the existence of (secure) tweakable block ciphers is equivalent to the existence of (secure) block ciphers. We leave it as an open problem to devise a construction with a tighter bound than Theorem 1.

3.1 Another construction

We can do better than this, however. We now give a construction that is more efficient, and is also a strong tweakable block cipher.

First, we need a definition. A set $\mathcal{H}$ of functions with signature $\{0,1\}^t \to \{0,1\}^n$ is said to be ϵ-almost 2-xor-universal (ϵ-AXU_2, for short) if $\Pr_h[h(x) \oplus h(y) = z] \le \epsilon$ holds for all x, y, z, where the probability is taken over h chosen uniformly at random from $\mathcal{H}$.

With these definitions, we prove

Theorem 2. *Let* $\widetilde{E}_{K,h}(T, M) = E_K(M \oplus h(T)) \oplus h(T)$*, and let* $\mathcal{H}$ *be an* ϵ*-*AXU_2 *family with* $\epsilon \ge 1/2^n$*. Then* $\widetilde{E}$ *is a strong tweakable block cipher. Specifically,*

$$\mathrm{Sec}'_{\widetilde{E}}(q, t) \le \mathrm{Sec}'_E(q, t) + 3\epsilon q^2.$$

We give the proof in Appendix A.

As there are plenty of known constructions of AXU_2 hash families with $\epsilon \approx 1/2^n$, the security theorem shows that we can obtain a construction with good security against adaptive chosen-ciphertext attacks for up to the birthday bound, i.e., for $q \ll 2^{n/2}$.

Moreover, we expect that our construction will be reasonably fast. For instance, for $t = n = 128$, a generalized division hash runs in something like 300 cycles [15], UMAC/UHASH runs in about 200 cycles [5], hash127 runs in about 150 cycles [2] and a DFC-style decorrelation module should run in about 200 cycles [9] (all speeds on a Pentium II class machine, and are rough estimates). If we compare to AES, which runs in about 230–300 cycles [1], we expect that a version of AES tweaked in this way will run about 50–80% slower than the plain AES. Though this is likely to be faster than the previous construction, it does require a longer key.

4 Tweakable Modes of Operation

The new "tweak" input of a tweakable block ciphers enables a multitude of new modes of operation. Indeed, these new modes may really be the "payoff" for introducing tweakable block ciphers. In this section we sketch three such possible modes, and leave the remainder to your imagination. We just describe the first two, and prove secure the third, which is the most interesting of the three (it is an analogue to OCB mode for authenticated encryption).

4.1 Tweak Block Chaining (TBC)

Tweak block chaining (TBC) is similar to cipher block chaining (CBC). An initial tweak T_0 plays the role of the initialization vector (IV) for CBC. Each successive message block M_i is encrypted under control of the encryption key K and a tweak T_{i-1}, where $T_i = C_i$ for $i > 0$. See Figure 2.

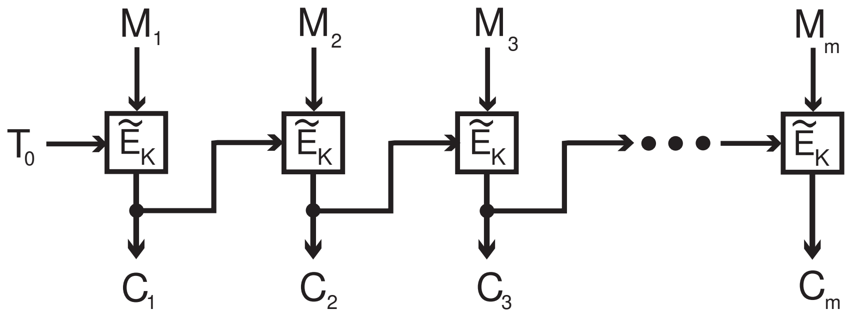

Fig. 2. Tweak block chaining: a chaining mode for a tweakable block cipher. Each ciphertext becomes the tweak for the next encryption.

To handle messages whose length is greater than n but not a multiple of n, a variant of ciphertext-stealing [13] can be used; see Figure 3.

One can also adapt the TBC construction to make a TBC-MAC in the same manner that one can use the CBC construction to make a CBC-MAC, though these constructions still need a security analysis.

4.2 Tweak Chain Hash (TCH)

To make a hash function, one can adapt the Matyas-Meyer-Oseas construction (see Menezes et al. [11, Section 9.40]). See Figure 4, using a fixed public key K in the tweakable block cipher, and chaining through the tweak input.

We don't know if this construction is secure. With a strong additional property on the tweakable block cipher, namely that for a fixed known key and fixed

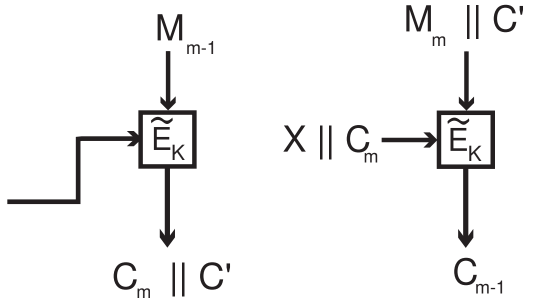

Fig. 3. Ciphertext stealing for tweak block chaining handles messages whose length is at least n bits long but not a multiple of n. Let r denote the length of the last (short) block M_m of the message. Then $|C_m| = |M_m| = r$ and $|C'| = n - r$. Here X denotes the rightmost $n - r$ bits of C_{m-2} (or of T_0 if $m = 2$).

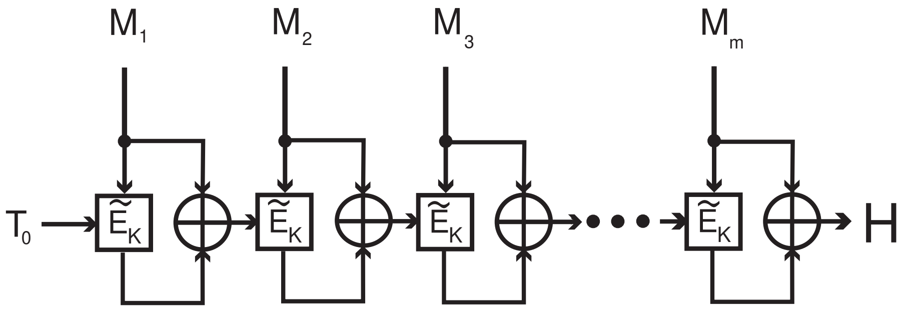

Fig. 4. The tweak chain hash (TCH). Here T_0 is a fixed initialization vector. A fixed public key K is used in the tweakable block cipher. The message M is padded in some fixed reversible manner, such as by appending a 1 and then enough 0's to make the length a multiple of n. The value H is the output of the hash function.

unknown tweak, we still get a pseudorandom permutation, we could adapt the proof of the Davies-Meyer hash function. However, as we noted in section 2, this is not the case for all tweakable block ciphers.[2]

4.3 Tweakable Authenticated Encryption (TAE)

In this section we suggest an authenticated mode of encryption (TAE) based on the use of a tweakable block cipher. This mode can be viewed as a paraphrase or restatement of the architecture of the OCB (offset codebook) mode proposed by Rogaway et al. [12] to utilize tweakable block ciphers rather than DESX-like modules. The result is shown in Figure 5. (The reader may need to consult the OCB paper to follow the rather terse description given here.)

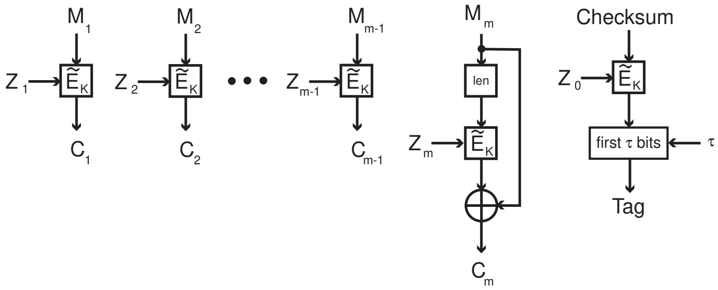

Fig. 5. Authenticated encryption based on a tweakable block cipher. This mode takes as input an $n/2$-bit nonce N. The tweak Z_i for $i > 0$ is defined as the concatenation of the nonce N, an $n/2-1$-bit representation of the integer i, and a zero bit 0: $Z_i = N\|i\|0$. The tweak Z_0 is defined as the concatentation of the nonce N, an $n/2-1$-bit representation of the integer b, where b is the bit-length of the message M, and a one bit 1: $Z_0 = N\|b\|1$. The message M is divided into $m-1$ blocks $M_1, \ldots, M_{m-1}$ of length n and one last block M_m of length r for $0 < r \leq n$ (except that if $|M| = 0$ then the last (and only) block has length 0). Each ciphertext block C_i has same length as M_i. The function $\mathrm{len}(\mathrm{M_m})$ produces an n-bit binary representation of the length r of the last message block. The last message block M_m is padded with zeros if necessary to make it length n before xoring. The checksum is $(M_1 \oplus \cdots \oplus M_{m-1} \oplus (M_m\|0^*))$. The parameter τ, $0 \leq \tau \leq n$ specifies the desired length of the authentication tag.

The OCB paper goes to considerable effort to analyze the probability that various encryption blocks all have distinct inputs. We feel that an authenticated encryption mode such as TAE should be much simpler to analyze, since the use of tweaks obviate this concern.

[2] One tweakable block cipher construction that does have this property is $\widetilde{E}_K(T, M) = E_K(E_T(E_K(M)))$, but this is not as desirable a construction as it is not easy to change the tweak.

We will in fact give a fairly easy proof that a tweakable block cipher used in TAE mode gives all the security properties claimed for OCB mode. Rogaway et al claim that OCB mode is:

– *Unforgeable.* Any nonce-respecting[3] adversary can forge a new valid encryption with probability at worst negligibly greater than $2^{-\tau}$.
– *Pseudorandom.* To any nonce-respecting adversary, the output of OCB mode is pseudorandom. In other words, no adversary can distinguish between an OCB mode oracle and a random function oracle. [12]

We now prove that TAE mode satisfies these properties.

Theorem 3. *If $\widetilde{E}$ is a secure tweakable block cipher, then $\widetilde{E}$ used in TAE mode will be unforgeable and pseudorandom.*

Proof. To prove that TAE mode is pseudorandom, we note that no tweak is ever repeated when the adversary is nonce-respecting. Now, if an adversary A were able to distinguish between a random function oracle and a TAE mode oracle, then we could distinguish the tweakable block cipher $\widetilde{E}$ from $\widetilde{\Pi}$ as follows. Given an oracle $\mathcal{O}$, we simply run A, and answer A's oracle queries by simulating TAE mode with $\mathcal{O}$ instead of $\widetilde{E}$. Now, if $\mathcal{O} = \widetilde{E}$ then we are in fact providing A with a TAE mode oracle. However, if $\mathcal{O} = \widetilde{\Pi}$ we are providing a random oracle. To see this, note that since no tweak is ever repeated, every part of every output is an independent random value. Thus, if we just give the answer A gives, we are correct whenever A is correct, and thus we defeat the security of $\widetilde{E}$.

To prove that TAE mode is unforgeable, we do the same thing. Suppose some adversary A can forge encryptions in TAE mode. We will break $\widetilde{E}$ as follows. Given an oracle $\mathcal{O}$ we just run A and answer A's oracle queries by simulating TAE mode with $\mathcal{O}$. When A gives an answer, we check to see if the answer is a successful forgery. If it is, we guess that $\mathcal{O} = \widetilde{E}$ and if not, we guess that $\mathcal{O} = \widetilde{\Pi}$. Since A is a successful adversary, if $\mathcal{O} = \widetilde{E}$, it forges successfully with probability nonnegligibly greater than $2^{-\tau}$. We will now show that if $\mathcal{O} = \widetilde{\Pi}$ then A forges with probability at most $2^{-\tau}$. Once we prove this we'll be done, since this reduction will be correct non-negligibly more often than it is incorrect.

Suppose now that $\mathcal{O} = \widetilde{\Pi}$. First we note that if A returns an answer with a new nonce, then Z_0 for the answer will be new and thus the correct answer will be a totally random τ-bit string. In other words, A will be correct with probability exactly $2^{-\tau}$. Secondly, if A returns an answer with an old nonce, then there are several cases. In the first case, all the ciphertext blocks are the same ciphertext blocks that were returned when that nonce was used previously. In this case, the forgery cannot possibly be correct since either it will be wrong or

[3] By "nonce-respecting," it is meant that while the adversary has oracle access and control of the nonce, the adversary may never ask that a nonce be used more than once; the idea is that any oracle a real adversary would have access to would still not repeat nonces, even if manipulated in order to accept nonces from elsewhere.

it will not be new. In the second case, the message is a different length than the message this nonce was queried with before. In this case, the forgery is correct with probability $2^{-\tau}$ since Z_0 will be different from before. In the final case, the message is the same length but there is at least one new ciphertext block. Now, the preimage of every block which is different is a random new value. Thus, the checksum is a random value, so the input to the $\widetilde{\Pi}$ that computes the tag is a new value. Thus, with probability exactly $2^{-\tau}$, the forgery is correct. Thus, the probability that the forgery is correct is at most $2^{-\tau}$, which concludes the proof. □

It is interesting to note that the construction *loses nothing* in terms of its advantage compared to the advantage of the tweakable block cipher! This is somewhat remarkable, and helps to emphasize our main point that tweakable block ciphers may be the most natural and useful construct for designing higher-level modes of operation. What's more, we note that if we use $\widetilde{E}$ from section 3.1, TAE mode is very similar to OCB mode. One critical difference is that OCB mode (essentially) derives its choice of h from the key K whereas our construction would require h to be additional key information. Also, OCB mode uses a Gray code to fine-tune efficiency, which we do not. However, this proof is significantly shorter and simpler than the proof for OCB mode, which further strengthens our point that tweakable block ciphers are the right primitive for this kind of task.

5 Conclusions and Open Problems

By introducing tweakable block ciphers, we have "re-partitioned" the design problem into two (new) parts: designing good tweakable block ciphers, and designing good modes of operation based on tweakable block ciphers. We feel that this re-partitioning is likely to be more useful and fruitful than the usual structure, since certain issues (e.g. having to do with collisions, say) can be handled once and for all at the lower level, and can then be ignored at the higher levels, instead of having to be dealt with repeatedly at the higher levels.

We feel that the notions of a *tweakable block cipher* and *tweakable modes of operation* (that is, modes of operation based on tweakable block ciphers) are interesting and worthy of further study.

One advantage of this framework is the new division of issues between design and analysis of the underlying primitive and the design and analysis of the higher-level modes of operation. We feel that the new primitive may result in a more fruitful partition.

Some interesting open problems are:

- What is the security of TAES (AES with our proposed "standard tweak")?
- Design efficient and secure tweakable block ciphers directly.
- Improve the construction of Theorem 1 to achieve a tighter bound.
- Analyze the security of the tweak-block-chaining mode of encryption.
- Analyze the security of the tweak chain hash.

– Devise and analyze the security of other modes of operation based on tweakable block ciphers.
– Define, devise and analyze the security of tweakable stream ciphers.

Acknowledgments

We would like to thank Rogaway, Bellare, Black, and Krovetz for inspiring this line of research with their proposed OCB mode and Mihir Bellare, Burt Kaliski, Zulfikar Ramzan, and Matthew Robshaw for very helpful discussions. Moses Liskov would like to acknowledge support from NTT grant #6762700.

References

1. Kazumaro Aoki and Helger Lipmaa. Fast implementations of AES candidates. In *Third AES Candidate Conference*, April 2000.
2. D.J. Bernstein. Floating-point arithmetic and message authentication, March 2000.
3. Eli Biham. New types of cryptanalytic attacks using related keys. *Journal of Cryptology*, 7(4):229–246, Fall 1994. Also available at: `citeseer.nj.nec.com/biham94new.html`.
4. Eli Biham and Alex Biryukov. How to strengthen DES using existing hardware. In *Proceedings ASIACRYPT '94*, volume 917 of *Lecture Notes in Computer Science*, pages 398–412. Springer-Verlag, 1994. Also available at: `citeseer.nj.nec.com/biham94how.html` .
5. John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway. UMAC: Fast and secure message authentication. In *Proceedings CRYPTO '99*, volume 1666 of *Lecture Notes in Computer Science*, pages 216–233. Springer-Verlag, 1999.
6. Paul Crowley. Mercy: A fast large block cipher for disk sector encryption. In *Fast Software Encryption: 7th International Workshop*, volume 1978 of *Lecture Notes in Computer Science*, pages 49–63. Springer-Verlag, 2000. Also available at: `www.ciphergoth.org/crypto/mercy` .
7. Joan Daemen. Limitations of the Even-Mansour construction. In *Proceedings ASIACRYPT '91*, volume 739 of *Lecture Notes in Computer Science*, pages 495–499. LNCS, Springer-Verlag, 1991. Also available at: `citeseer.nj.nec.com/daemen92limitation.html` .
8. Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–161, Summer 1997. Also available at: `citeseer.nj.nec.com/even91construction.html` .
9. L. Granboulan, P. Nguyen, F. Noilhan, and S. Vaudenay. DFCv2. In *Selected Areas in Cryptography*, volume 2012 of *Lecture Notes in Computer Science*, pages 57–71. Springer-Verlag, 2001.
10. Joe Kilian and Phillip Rogaway. How to protect DES against exhaustive search (an analysis of DESX). In *Proceedings CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 252–267. Springer, 1996. See `http://www.cs.ucdavis.edu/~rogaway/papers/desx.ps` for an updated version.
11. Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997.

12. Phillip Rogaway, Mihir Bellare, John Black, and Ted Krovetz. A block-cipher mode of operation for efficient authenticated encryption. In *Eighth ACM Conference on Computer and Communications Security (CCS-8)*, pages 196–205. ACM Press, Aug 16 2001. See `http://www.cs.ucdavis.edu/~rogaway/ocb/ocb-doc.htm`.
13. Bruce Schneier. *Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C.* John Wiley & Sons, New York, 1996.
14. Rich Schroeppel. The hasty pudding cipher. Available at `http://www.cs.arizona.edu/~rcs/hpc/` ., 1999.
15. Victor Shoup. On fast and provably secure message authentication based on universal hashing. In *Proceedings CRYPTO '96*, volume 1109 of *Lecture Notes in Computer Science*, pages 313–328. Springer, 1996.
16. Serge Vaudenay. Provable security for block ciphers by decorrelation. In *Proceedings STACS '98*, volume 1373 of *Lecture Notes in Computer Science*, pages 249–275. Springer-Verlag, 1998.

A Proof of Theorem 2

In this section, we give a proof of Theorem 2. First, though, we establish some notation. We use $\Pr_0[\cdot]$ to represent the probability measure in the case where A interacts with $\widetilde{E}_{K,h}$, where the probability is taken over the choice of $K \in \{0,1\}^k$ and $h \in \mathcal{H}$ uniformly and independently at random. Also, we let $\Pr_1[\cdot]$ denote the measure where A interacts with $\widetilde{\Pi}$. In either case, we write $\mathcal{O}$ for A's oracle, so in the former case $\mathcal{O} = \widetilde{E}_K$, and in the latter case $\mathcal{O} = \widetilde{\Pi}$.

We let the random variable T_i denote the tweak input on A's i-th oracle call, and we let M_i and C_i denote the plaintext and ciphertext corresponding to this call, so that $\mathcal{O}(T_i, M_i) = C_i$. In other words, if A's i-th oracle query is an encryption query (to $\mathcal{O}$), then (T_i, M_i) denotes the input and C_i the return value, whereas if A's i-th oracle query is a decryption query (to $\mathcal{O}^{-1}$), then the input is (T_i, C_i) and the result of the query is M_i. Moreover, we define the random variables N_i, B_i by $N_i = M_i \oplus h(T_i)$ and $B_i = C_i \oplus h(T_i)$. Note that if $\mathcal{O} = \widetilde{E}_K$, then $E_K(N_i) = B_i$. We define the random variable τ_n by $\tau_n = \langle (T_1, M_1, C_1), \ldots, (T_n, M_n, C_n) \rangle$, and we use $\tau = \tau_q$ to represent the full transcript of interaction with the oracle.

We fix an adversary A, and we assume without loss of generality that A does not make any repeated or redundant queries to its oracle. As a consequence of this assumption, the pairs (T_i, M_i) are all distinct, or in other words, for all $i \neq j$, we have $(T_i, M_i) \neq (T_j, M_j)$. Similarly, the pairs (T_i, C_i) are also distinct, as are the (T_i, N_i)'s and the (T_i, B_i)'s. Also, the output of A can be viewed as a function of the transcript τ, so we sometimes write the output of A as $A(\tau)$.

Our proof is separated into two parts. In the information-theoretic part, we let $E_K = \Pi$ denote a permutation chosen uniformly at random, we set $\widetilde{E'}(T, M) = \Pi(M \oplus h(T)) \oplus h(T)$, and we show that $\widetilde{E'}$ is a secure tweakable block cipher. Then, in the computational part, we let E be arbitrary, and we show that if E_K and Π are computationally indistinguishable, then $\widetilde{E}$ will also be a secure tweakable block cipher.

The information-theoretic part of the proof uses the following strategy. We define a bad event Bad. We show that when conditioning on the complement event, the probability measures $\Pr_0[\cdot|\overline{\mathsf{Bad}}]$ and $\Pr_1[\cdot|\overline{\mathsf{Bad}}]$ are in fact identical. Then, we show that $\Pr_0[\mathsf{Bad}]$ and $\Pr_1[\mathsf{Bad}]$ are both small. The result will then follow using standard arguments.

In our arguments, we define Bad_n to be the event that, for some $1 \leq i < j \leq n$, either $N_i = N_j$ or $B_i = B_j$. Also, we let $\mathsf{Bad} = \mathsf{Bad}_q$.

Lemma 1. *For every possible transcript t, if $E_K = \Pi$, then* $\Pr_0[\tau = t|\overline{\mathsf{Bad}}] = \Pr_1[\tau = t|\overline{\mathsf{Bad}}]$.

Proof. We show this by induction on the length of the transcript, q. Consider the q-th oracle query: it is either an encryption or a decryption query. Suppose first that the q-th oracle query is an encryption query, with inputs (T_q, M_q). By the inductive hypothesis, we can assume that the distribution of τ_{q-1} is the same for both the $\Pr_0[\cdot|\overline{\mathsf{Bad}_{q-1}}]$ and $\Pr_1[\cdot|\overline{\mathsf{Bad}_{q-1}}]$ probability measures, hence the same is true of the distribution of (τ_{q-1}, T_q, M_q).

Now fix any h such that $N_q \notin \{N_1, \ldots, N_{q-1}\}$, so that the only remaining random choice is over Π or $\widetilde{\Pi}$. When $\mathcal{O} = \widetilde{\Pi}$, $C_q = \widetilde{\Pi}(T_q, M_q)$ is uniformly distributed on the set $S = \{0,1\}^n \setminus \{C_i : T_i = T_q \text{ and } 1 \leq i < q\}$, if we condition on $\overline{\mathsf{Bad}_{q-1}}$ (but before conditioning on $\overline{\mathsf{Bad}_q}$). When $\mathcal{O} = \widetilde{E'}$, we find something slightly different: $B_q = \Pi(N_q)$ is uniformly distributed on the set $\{0,1\}^n \setminus \{B_1, \ldots, B_{q-1}\}$ (conditioned on $\overline{\mathsf{Bad}_{q-1}}$, but before conditioning on $\overline{\mathsf{Bad}_q}$), hence $C_q = B_q \oplus h(T_q)$ is uniformly distributed on the set $S' = \{0,1\}^n \setminus \{C_i \oplus h(T_i) \oplus h(T_q) : i = 1, \ldots, q-1\}$. In both cases, the probabilities are independent of the choice of h. Also, note that $S' \subseteq S$, since when $T_i = T_q$ we have $C_i \oplus h(T_i) \oplus h(T_q) = C_i$. Adding the condition $\overline{\mathsf{Bad}_q}$ amounts to adding the condition that $B_q \in \{0,1\}^n \setminus \{B_1, \ldots, B_{q-1}\}$, i.e., that $C_q \in S'$. Thus, after conditioning on $\overline{\mathsf{Bad}_q}$, we see that C_q is uniformly distributed on S' and independent of the rest of the transcript, and hence the distribution of τ is the same for both the $\Pr_0[\cdot|\overline{\mathsf{Bad}_q}]$ and $\Pr_1[\cdot|\overline{\mathsf{Bad}_q}]$ probability measures. (Here we have used the following simple fact: if the random variable X is uniform on a set S, and if S' is some subset of S, then after conditioning on the event $X \in S'$ we find that the resulting r.v. is uniform on S'.)

This covers the case where the q-th query is a chosen-plaintext query. The other case, where the q-th query is a chosen-ciphertext query, is treated similarly. This concludes the proof of Lemma 1.

Lemma 2. *If $\mathcal{H}$ is ϵ-AXU_2, then* $\Pr_1[\mathsf{Bad}_q] \leq \epsilon q(q-1)$.

Proof. Note that, when $\mathcal{O} = \widetilde{\Pi}$, h is independent of the transcript τ. Hence, we can defer the choice of h until after A completes all q of its queries and the values of T_i, M_i, C_i are fixed. Then, we find $\Pr_1[\mathsf{Bad}_q] = \Pr_h[\exists i,j. N_i = N_j \vee B_i = B_j]$(by the definition of Bad_q) $\leq \sum_{1 \leq i < j \leq q} \Pr_h[N_i = N_j] + \Pr_h[B_i = B_j]$(by a union bound) $= \sum_{1 \leq i < j \leq q} \Pr_h[h(T_i) \oplus h(T_j) = M_i \oplus M_j] \Pr_h[h(T_i) \oplus h(T_j) = C_i \oplus C_j]$(by the definition of N_i, B_i) $\leq \sum_{1 \leq i < j \leq q} 2\epsilon = \epsilon q(q-1)$.(since $\mathcal{H}$ is

ϵ-AXU$_2$)

Lemma 3. *If $\mathcal{H}$ is ϵ-AXU$_2$ for $\epsilon \geq 1/2^n$, and if $E_K = \Pi$, then* $\Pr_0[\mathsf{Bad}_q] \leq 1.5\epsilon q(q-1)$.

Proof. We will prove $\Pr_0[\mathsf{Bad}_q] \leq 1.5\epsilon q(q-1)$ by induction on q. Let E denote that event that, for some i, we have $N_i = N_q$, and let E' denote the event that, for some i, we have $B_i = B_q$. Note that $\Pr_0[\mathsf{Bad}_q] = \Pr_0[\mathsf{Bad}_{q-1}] + \Pr_0[\mathsf{Bad}_q|\overline{\mathsf{Bad}}_{q-1}]\Pr_0[\overline{\mathsf{Bad}}_{q-1}]$. By the inductive hypothesis, $\Pr_0[\mathsf{Bad}_{q-1}] \leq 1.5\epsilon(q-1)(q-2)$. Also, $\Pr_0[\overline{\mathsf{Bad}}_{q-1}] \leq 1$. Hence all that remains is to bound the term $\Pr_0[\mathsf{Bad}_q|\overline{\mathsf{Bad}}_{q-1}]$.

Applying a union bound shows $\Pr_0[\mathsf{Bad}_q|\overline{\mathsf{Bad}}_{q-1}] \leq \Pr_0[\mathsf{E}|\overline{\mathsf{Bad}}_{q-1}] + \Pr_0[\mathsf{E}'|\overline{\mathsf{E}} \wedge \overline{\mathsf{Bad}}_{q-1}]$. We next bound each of these two terms in turn. By Lemma 1, and since $\mathcal{H}$ is ϵ-AXU$_2$, we see $\Pr_0[\mathsf{E}|\overline{\mathsf{Bad}}_{q-1}] = \Pr_1[\mathsf{E}|\overline{\mathsf{Bad}}_{q-1}] \leq \epsilon(q-1)$. Moreover, $\Pr_0[\mathsf{E}'|\overline{\mathsf{E}} \wedge \overline{\mathsf{Bad}}_{q-1}] = \Pr_0[\Pi(N_q) \in \{B_1, \ldots, B_{q-1}\}|\overline{\mathsf{E}} \wedge \overline{\mathsf{Bad}}_{q-1}] \leq (q-1)/(2^n - q + 1) \leq 2(q-1)/2^n \leq 2\epsilon(q-1)$, since $\Pi(N_q)$ is uniformly distributed on a set of size at least $2^n - q + 1$ and since $\epsilon \geq 1/2^n$. Finally, $1.5\epsilon(q-1)(q-2) + \epsilon(q-1) + 2\epsilon(q-1) \leq 1.5\epsilon q(q-1)$. The statement of the lemma now follows.

We are now ready to prove the security theorem.

Proof (of Theorem 2). First, we do a simple calculation:

$$\begin{aligned}
\mathrm{Sec}'_{\widetilde{E'}}(q,t) &= \max_A |\Pr{}_0[A(\tau)=1] - \Pr{}_1[A(\tau)=1]| && \text{(by definition)}\\
&= \max_A |\Pr{}_0[A(\tau)=1|\overline{\mathsf{Bad}}]\Pr{}_0[\overline{\mathsf{Bad}}] + \Pr{}_0[A(\tau)=1|\mathsf{Bad}]\Pr{}_0[\mathsf{Bad}]\\
&\qquad - \Pr{}_1[A(\tau)=1|\overline{\mathsf{Bad}}]\Pr{}_1[\overline{\mathsf{Bad}}] - \Pr{}_1[A(\tau)=1|\mathsf{Bad}]\Pr{}_1[\mathsf{Bad}]|\\
& && \text{(by conditional probabilities)}\\
&\leq \max_A |\Pr{}_0[A(\tau)=1|\overline{\mathsf{Bad}}]\Pr{}_0[\overline{\mathsf{Bad}}] - \Pr{}_1[A(\tau)=1|\overline{\mathsf{Bad}}]\Pr{}_1[\overline{\mathsf{Bad}}]|\\
&\qquad + |\Pr{}_0[A(\tau)=1|\mathsf{Bad}]\Pr{}_0[\mathsf{Bad}] - \Pr{}_1[A(\tau)=1|\mathsf{Bad}]\Pr{}_1[\mathsf{Bad}]|\\
& && \text{(by the triangle inequality)}\\
&\leq \max_A |\Pr{}_0[A(\tau)=1|\overline{\mathsf{Bad}}]\Pr{}_0[\overline{\mathsf{Bad}}] - \Pr{}_1[A(\tau)=1|\overline{\mathsf{Bad}}]\Pr{}_1[\overline{\mathsf{Bad}}]|\\
&\qquad + 1.5\epsilon q(q-1) && \text{(since } \Pr[\mathsf{Bad}] \leq 1.5\epsilon q(q-1))\\
&\leq \max_A \Pr[A(\tau)=1|\overline{\mathsf{Bad}}] \cdot |\Pr{}_0[\overline{\mathsf{Bad}}] - \Pr{}_1[\overline{\mathsf{Bad}}]| + 1.5\epsilon q(q-1)\\
& && \text{(by Lemma 1)}\\
&\leq 3\epsilon q(q-1) && \text{(since } 1 - 1.5\epsilon q(q-1) \leq \Pr[\overline{\mathsf{Bad}}] \leq 1)
\end{aligned}$$

The result then follows from the triangle inequality: $\mathrm{Sec}'_{\widetilde{E}}(q,t) = \max_A |\Pr{}_0[A^{\widetilde{E}_K, \widetilde{E}_K^{-1}} = 1] - \Pr{}_1[A^{\widetilde{\Pi}, \widetilde{\Pi}^{-1}} = 1]| \leq \max_A |\Pr{}_0[A^{\widetilde{E}_K, \widetilde{E}_K^{-1}} = 1] - \Pr{}_0[A^{\widetilde{E'}_K, \widetilde{E'}_K^{-1}} = 1]| + |\Pr{}_0[A^{\widetilde{E'}_K, \widetilde{E'}_K^{-1}} = 1] - \Pr{}_1[A^{\widetilde{\Pi}, \widetilde{\Pi}^{-1}} = 1]| \leq Sec'_E(q,t) + \mathrm{Sec}'_{\widetilde{E'}}(q,t) \leq \mathrm{Sec}'_E(q,t) + 3\epsilon q(q-1)$.