Two-Pass Authenticated Encryption Faster than Generic Composition

Stefan Lucks

University of Mannheim, Germany
http://th.informatik.uni-mannheim.de/people/lucks/

Abstract. This paper introduces CCFB and CCFB+H, two patent-free authenticated encryption schemes. CCFB+H also supports the authentication of associated data. Our schemes can employ any block cipher and are provably secure under standard assumptions. The schemes and their proofs of security are simple and straightforward. CCFB and CCFB+H restrict the sizes of nonce and authentication tags and can, depending on these sizes, perform significantly better than both generic composition and other two-pass schemes for authenticated encryption, such as the EAX mode.

Keywords: authenticated encryption, associated data, provable security, OMAC

1 Introduction

An *Authenticated Encryption (AE)* scheme is a secret-key cryptosystem designed for simultaneously protecting *both* a message's privacy *and* its authenticity. Traditionally, these two security goals had been handled separately by the means of encryption schemes and message authentication codes (MACs). In practice, however, the same message often needs to be kept both private and authentic, and gluing together encryption and message authentication is surprisingly tricky and error-prone. Hence, a couple of block cipher based AE schemes have been developed recently.

Even more recently, people discovered that AE is not quite sufficient. Often, some header *(associated data, AD)* is not confidential, but vital for authentication. *Authenticated Encryption with Associated Data (AEAD)* schemes authenticate both the message and the associated data, but only encrypt the message. Most of today's AE and AEAD schemes are either "two-pass" schemes and thus as slow as encrypting and authenticating independently, or "one-pass" schemes whose usage is hindered by the patent situation. This paper proposes a new two-pass scheme. Depending on the size of the authentication tag, our solution can run significantly faster than generic composition or other non-patented two-pass AE(AD) schemes. Another advantage is simplicity: compared to other AE(AD) schemes, our solution and its proof of security is very simple and straightforward.

1.1 The Development of Authenticated Encryption

In 2000, Bellare and Namprempre proposed *generic composition*: a privacy-protecting encryption scheme and a MAC are used jointly (but securely) under independent keys [3]. This is not very efficient – it takes the time to encrypt plus the time to authenticate and makes block cipher based authenticated encryption twice as slow as either encryption or authentication. The generic approach can provide AEAD as well as AE. Generic composition can be *minimal-expanding*[1], i.e. the size of a ciphertext is the *plaintext size* plus τ *bit for the authentication tag*, where τ is a plaintext-size-independent constant, and the forgery probability is close to $1/2^\tau$.

In the same year, Katz and Yung presented the RPC block cipher mode for authenticated encryption [8]. It is a *single-pass* AE scheme, but the message expansion is not minimal – it is linear in the plaintext size. Depending on the size of the authentication tag, RPC can run significantly faster than generic AE, but always less than twice as fast[2]. For historical reasons, the authors of RPC did not consider AEAD.

In 2001, several *single-pass* minimal-expanding AE schemes have been proposed: IAPM, OCB and XCBC [7,13,4]. These combine minimal expansion with a close-to-optimal running time: for large messages, these schemes are almost as fast as conventional encryption (without authenticity), i.e. twice as fast as the generic approach. In 2002, a single-pass AEAD scheme based on OCB has been proposed [12].

Unfortunately, several patents cover the usage of the fast single-pass schemes. The patent situation has turned out to be a significant deterrence. To avoid patents, new *two-pass* AEAD schemes have been developed, with one pass for encryption and another one for authentication. The first was CCM [15], followed by EAX, CWC, and GCM [1,2,9,10,11], which addressed some shortcomings [14] of CCM. All these modes are minimal expanding, but as (in)efficient as generic composition. Their main advantage over generic composition is that a single block cipher key suffices for the entire scheme.

1.2 Contributions and Outline of this Paper

This paper proposes CCFB (Counter-CipherFeedBack) – another two-pass AE mode for block ciphers, but with a different separation of duties between the passes. It has been developed with low-end devices in the mind, such as smart-cards, small embedded systems, sensor network motes, and RFID tags. CCFB is related to RPC, which has been published *before* the patented single-pass schemes. The first pass of CCFB is for privacy and "local" authentication, while the second computes a single "global" authentication tag from the local ones. CCFB+H is

[1] ... depending on the underlying encryption and MAC scheme.

[2] E.g., AES-RPC with 32-bit authentication tags is 50 % faster than AES-based generic composition.

- a new *minimal-expanding* and *two-pass* AEAD scheme (avoiding the patents on single-pass schemes [3], similarly to EAX, CWC, and GCM),
- which can run significantly faster than previously published two-pass schemes[4], especially on low-end devices.

Like EAX, CWC, and GCM,

- CCFB+H can use any block cipher and even a pseudorandom function (PRF) as the underlying primitive,
- it uses a single block cipher (or PRF) key for all its work, and a block cipher is only used in encryption mode,
- CCFB+H allows the (pre-)processing of the header, independently from the message,
- CCFB+H is provably secure under standard assumptions on the security of the underlying block cipher or PRF,
- and we analyse our schemes' concrete security.

A drawback, inherited from RPC, is that the sizes for nonces and authentication tags are limited (in contrast to EAX, CWC, and GCM). More specifically, if n is the block size of the underlying block cipher or PRF, then

$$\overbrace{\text{maximum size of nonce}}^{\delta} = \overbrace{\text{block size}}^{n} - \overbrace{\text{size of authentication tag}}^{\tau}.$$

Section 2 describes CCFB, Section 4 analyses it with respect to the notions of security defined in Section 3. Section 5 extends CCFB to an AEAD scheme CCFB+H (CCFB with Header). Using OMAC [5,6], a block cipher based message authentication code, Sections 6 and 7 develop a block cipher based instantiation of CCFB. Section 8 compares CCFB+H and EAX security-wise and performance-wise. The proof of Theorem 2 and some figures are deferred to the appendix.

2 CCFB Authenticated Encryption

We define *CCFB authenticated encryption* under a function $F : \{0,1\}^n \rightarrow \{0,1\}^n$. Fix the tag size $\tau \leq n/2$. Set $\delta = n - \tau$. The notation "$(d,t) := F(\cdot)$" implies $d \in \{0,1\}^\delta$ and $t \in \{0,1\}^\tau$. For $i \in \{1,\ldots,2^\tau - 1\}$, we write $\langle i \rangle_\tau$ for the corresponding τ-bit string. We write "$\|$" for the concatenation of bit-strings. If X is a bit-string of length $\geq \lambda$, we write $\mathrm{MSB}_\lambda(X)$ for the first λ bits of

[3] We neither have, nor are aware of any patents or pending patents relevant to CCFB+H. We do not intend to apply for such patents.

[4] EAX and our instantiation of CCFB+H are dominated by the block cipher operations, and can run on any low-end device capable of running block cipher operations. This enables a "platform-independent" performance evaluation by counting the number of block cipher calls, see Section 8. In the same section, we also explain why CWC and GCM appear to be poor choices for low-end devices.

X. The input for CCFB encryption consists of a nonce $N \in \{0,1\}^\delta$ (shorter nonces can be padded), and a message M of any length $|M|$ between 1 bit and $(2^\tau - 3)\delta$ bit. The algorithm is described in Figure 1. See also Figures 2 and 3 for an illustration of CCFB encryption.

```
Algorithm: CCFB encryption.
Input: nonce $N \in \{0,1\}^\delta$ and $M \in \{0,1\}^*$, $1 \le |M| \le (2^\tau - 3)\delta$;
First pass:
    1. parse $M$ as $(M_1, \ldots, M_m)$ with
           $|M_1| = \cdots = |M_{m-1}| = \delta$, $|M_m| \in \{1, \ldots, \delta\}$;
    2. $C_0 := N$;
    3. for $1 \le i \le m-1$:    $(\mathrm{tmp}, A_i) := F(C_{i-1}, \langle i \rangle_\tau)$;
                                 $C_i := \mathrm{tmp} \oplus M_i$;
    4. $(\mathrm{tmp}, A_m) := F(C_{m-1}, \langle m \rangle_\tau)$;
    5. if $|M_m| = \delta$ then $d := 1; \mathrm{pad} := ()(*\text{empty string}*)$;
                      else $d := 2; \mathrm{pad} := (1||0^{\delta-|M_m|-1})$;
    6. $C_m := \mathrm{MSB}_{|M_m|}(\mathrm{tmp}) \oplus M_m$;
    7. $C' := \mathrm{tmp} \oplus (M_m||\mathrm{pad})$;
    8. $(\mathrm{dummy}, A_{m+1}) := F(C', \langle m+d \rangle_\tau)$;
Second pass:
    9. $T := A_1 \oplus \cdots \oplus A_{m+1}$;
Output: ciphertext $(C_1, \ldots, C_m, T)$ with
       $C_1, \ldots, C_{m-1} \in \{0,1\}^\delta$, and $C_m \in \{0,1\}^{|M_m|}$.
```

Fig. 1. CCFB encryption under $F : \{0,1\}^\delta \times \{0,1\}^\tau \to \{0,1\}^\delta \times \{0,1\}^\tau$.

Observe that if the length $|M|$ of M is a multiple of δ, i.e., $|M_m| = \delta$, steps 3 to 8 simplify to the following short algorithm:

- for $1 \le i \le m$: $(\mathrm{tmp}, A_i) := F(C_{i-1}, \langle i \rangle_\tau)$;
  $C_i := \mathrm{tmp} \oplus M_i$;
- $(\mathrm{dummy}, A_{m+1}) := F(C_m, \langle m+1 \rangle_\tau)$;

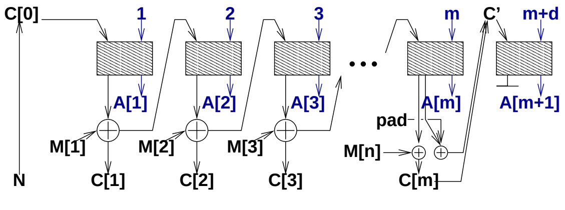


Fig. 2. 1st phase of CCFB encryption: compute the C_i and the local tags A_i; $d \in \{1, 2\}$.

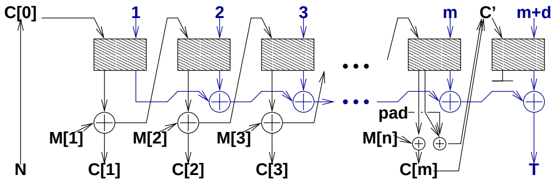


Fig. 3. Full CCFB encryption: The global tag T is computed in the second phase.

An $|M|$-bit message M is split into $m = \lceil |M|/\delta \rceil + 1$ blocks M_i, and encrypting M requires $2m$ XORs and $m + 1$ random function (or block cipher) calls. Thus, CCFB runs at essentially the same speed as RPC [8]. The most important differences between CCFB and RPC, cf. Figure 7 in the appendix, are:

- CCFB employs CipherFeedBack, where RCB uses the ECB mode. Accordingly, RPC assumes F to be a permutation.
- The output of RPC consists of the encryption blocks C_i and the local authentication tags A_i. CCFB extends RPC by the second pass, which makes CCFB minimal-expanding. The output of CCFB is a single "global" authentication tag $T = \bigoplus A_i$.
- To protect against cut-and-paste attacks, RPC requires a message encoding with reserved "start" and "stop" blocks. CCFB does not need a message encoding.

Given a nonce $N \in \{0,1\}^\delta$ and a ciphertext $C = (C_1, \ldots, C_m, T)$, **CCFB decryption** is straightforward and needs as much computation as the encryption, see Figure 6 in the appendix.

As ususal for modes of operations, nonces must not be re-used. E.g., if we encrypt two messages $(M_1, \ldots, M_m)$ and $(M'_1, \ldots, M'_{m'})$ under the same nonce, the corresponding first ciphertext blocks satisfy $C_1 \oplus C'_1 = M_1 \oplus M'_1$.

3 Notions of Security for Authenticated Encryption

Before we analyse the security of CCFB (and later CCFB+H), we have to specify what we mean by "secure". Our notions of security are standard, see e.g. [1,2]. An **AEAD scheme** is a pair (E, D) of deterministic algorithms E for encryption and D for decryption:

$$E: \text{KEY} \times \text{NONCE} \times \text{HEADER} \times \text{MESSAGE} \rightarrow \text{CIPHERTEXT},$$
$$D: \text{KEY} \times \text{NONCE} \times \text{HEADER} \times \text{CIPHERTEXT} \rightarrow \text{MESSAGE} \cup (\text{none}).$$

The sets KEY, NONCE, HEADER, MESSAGE, and CIPHERTEXT are bit-strings, i.e., subsets of $\{0,1\}^*$. For simplicity, we assume KEY to be finite. An **adversary** with access to an **encryption oracle** $E(K, \cdot, \cdot, \cdot)$ chooses triples (N^1, H^1, M^1), $\ldots$, $(N^q, H^q, M^q) \in \text{NONCE} \times \text{HEADER} \times \text{MESSAGE}$ and receives the corresponding ciphertexts $C^i = E(K, N^i, H^i, M^i)$. The adversary is **nonce-respecting**, if for all $i \neq j$, $N^i \neq N^j$. If NONCE is finite, a **nonce-randomising** adversary chooses a fresh uniformly distributed random $N^i \in$ NONCE for each query (N^i, H^i, M^i).

In a privacy attack, the adversary is either given access to the real encryption oracle, or to a fake oracle $F(K, \cdot, \cdot, \cdot)$, which on input (N^i, H^i, M^i) returns a random ciphertext $F(N^i, H^i, M^i)$ of the same length as the real ciphertext $C^i = E(K, N^i, H^i, M^i)$. The adversay has to distinguish between both oracles. Let K be a random key. An AEAD scheme is p-**private** against a class of adversaries, if for all adversaries A of that class, the advantage in distinguishing E from F is

$$\left|\Pr\left[A^{E(K,\cdot,\cdot,\cdot)} = 1\right] - \Pr\left[A^{F(\cdot,\cdot,\cdot)} = 1\right]\right| \leq p.$$

A forger asks queries (N^1, H^1, M^1), …, (N^q, H^q, M^q), receives the corresponding ciphertexts C^1, …, C^q, and finally chooses a ciphertext C, a nonce N, and a header H. The forger succeeds, if $(C, H) \notin \{(C^1, H^1) \dots, (C^q, H^q)\}$[5] and $D(K, N, H, C) \neq$ (none).

An AEAD scheme is p-**authentic** against a class of forgers, if for all forgers A_F of that class and a random key K

$$\Pr\left[A_F \text{ succeeds}\right] \leq p.$$

An **AE scheme** is an AEAD scheme without a choice for the headers: HEADER $= \{0\}$.

4 Analysis of CCFB Authenticated Encryption

Consider a chosen plaintext scenario where the adversary $\mathcal{A}$ selects q messages $M^1 = (M^1_1, \dots, M^1_{m_1})$, …, $M^q = (M^q_1, \dots, M^q_{m_q})$ with $r = \sum_{1 \leq i \leq q} m_i$ blocks in total. We write $N^1 = C^1_0$, …, $N^q = C^q_0$ for the corresponding nonces chosen by $\mathcal{A}$, and $C^1 = (C^1_1, \dots, C^1_{m_1}, T^1)$, …, $C^q = (C^q_1, \dots, C^1_{m_q}, T^q)$ for the ciphertexts. Consider the inputs for F:

$$D^i_k = \begin{cases} (C^i_k, k+1) & \text{if } k < m_i \\ ((C')^i, m_i + d) & \text{if } k = m_i \ (d = 1 \text{ if } |M_m| = \delta, \text{ else } d = 2). \end{cases} \tag{1}$$

Here $(C')^i$ corresponds to the “internal” value C' from Figure 1. An “input-collision” is an input-pair (D^i_k, D^j_k) with

$$D^i_k = D^j_k \text{ with } 1 \leq i < j \leq q \text{ and } k \in \{0, \dots, \min\{m_i, m_j\}\}. \tag{2}$$

We assume the adversaries to ask q queries to the encryption oracle with, in total, r message blocks, i.e. $r = \sum_{1 \leq i \leq q} m_i$.

[5] Even if the forger is nonce-respecting $N \in \{N^1, \dots, N^q\}$ is permissable.

Lemma 1. *For CCFB encryption under a random function F, the probability for any nonce-respecting adversary to generate an input-collision is at most*

$$\frac{qr}{2^{\delta+1}}.$$

Similarly, the probability for any nonce-randomising adversary to generate an input-collision is at most

$$\frac{q(r+q)}{2^{\delta+1}}.$$

Proof. First, consider a nonce-respecting adversary. There is no input-collision with $k = 0$. Thus, we can concentrate on $k \geq 1$.

A collision $D^i_k = D^j_k$ implies $F(D^i_{k-1}) = F(D^j_{k-1})$, and if $D^i_{k-1} \neq D^j_{k-1}$, then $\Pr[D^i_k = D^j_k] \leq 1/2^\delta$. The number of triples (i, j, k) with $1 \leq i < j \leq q$ and $1 < k \leq \min\{m_i, m_j\}$, is at most $(q-1)r/2$. The probability that at least one of these triples collides is thus at most $\frac{(q-1)r}{2} * \frac{1}{2^\delta} = \frac{(q-1)r}{2^{\delta+1}}$.

Second, consider a nonce-randomising adversary. If there is no input-collision with $k = 0$, then the adversary happens to be nonce-respecting. The additional chance to generate an input-collision at the level $k = 0$ – which is in fact a nonce-collision – is a most $(q(q-1)/2)/2^\delta \leq q^2/2^{\delta+1}$. The second claim follows from $qr + q^2 = q(r+q)$. □

Theorem 1 (Information-Theoretic Privacy of CCFB).
CCFB encryption using a random F is

$\frac{qr}{2^{\delta+1}}$*-private against nonce-respecting adversaries and*

$\frac{q(r+q)}{2^{\delta+1}}$*-private against nonce-randomising adversaries.*

Proof. Without any input-collision D^i_k $(k \geq 0)$, all the inputs to the random function F are different, all of its outputs are distributed uniformly at random. Thus, the outputs from the "real" ecnryption oracle and the fake oracle are distributed equally. To distinguish the oracles, the adversary would need an input-collision. The claims follow from the bounds given in Lemma 1. □

Theorem 2 (Information-Theoretic Authenticity of CCFB).
CCFB encryption, using a random F, is

$\left(\frac{qr}{2^{\delta+1}} + \frac{1}{2^\tau}\right)$*-authentic with respect to nonce-respecting adversaries and*

$\left(\frac{q(r+q)}{2^{\delta+1}} + \frac{1}{2^\tau}\right)$*-authentic with respect to nonce-randomising adversaries.*

The proof will be given in the appendix.

5 The CCFB+H AEAD Mode and its Analysis

Let $F' : \{0,1\}^* \rightarrow \{0,1\}^\delta$ be an additional random function, chosen independently from F. Note that F' is defined for a variable input length, in contrast to F. We write $H \in \{0,1\}^*$ for the associate ("header") data and tweak both the CCFB encryption algorithm and its decryption counterpart by changing instruction 2 in Figure 1 and in Figure 6: replace $\boxed{C_0 := N;}$ by: $\boxed{C_0 := N \oplus F'(H);}$ see Figure 4 for an illustration of the modified encryption. Since CCFB+H is a tweaked CCFB, we conveniently inherit most the analysis from CCFB.

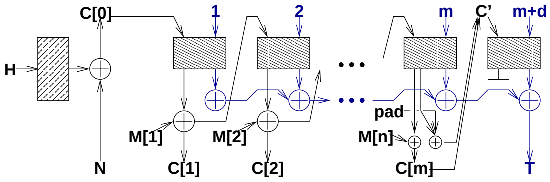


Fig. 4. CCFB Authenticated Encryption of Message $M[\cdot]$ with Associated Data H.

Recall the definition of q and r from the previous section.

Lemma 2. *For CCFB encryption under a random F, the probability for a nonce-respecting or nonce-randomising adversary to generate an input-collision is at most*

$$\frac{q(r+q)}{2^{\delta+1}}.$$

Proof. For a nonce-randomising adversary, the result follows immediately from the second claim of Lemma 1. We will show that for a nonce-respecting adversary, the probability for an input-collision at level $k = 0$ is no more than $q^2/2^{\delta+1}$. We write H^i for the header of the i-th chosen ciphertext query. If $H^i = H^j$, then $D^i \neq D^j$, since the adversary is nonce-respecting.

Consider $H^i \neq H^j$ and $\Delta(i,j) := F'(H^i) \oplus F'(H^j) \in \{0,1\}^\delta$. We get $C_0^i = C_0^j$ if and only if $N^i \oplus N^j = \Delta(i,j)$, i.e. with at most the probability $1/2^\delta$. There are $q(q-1)/2$ pairs (i,j) with $1 \leq i < j \leq q$, so the probability for an input-collision at level $k = 0$ is $q(q-1)/2^{\delta+1}$. □

The proofs for privacy and authenticity of CCFB+H are the same as their counterparts in Section 4. We consider adversaries, who are either nonce-respecting or nonce-randomising.

Theorem 3 (Information-Theoretic Privacy of CCFB+H).
CCFB encryption using a random F is

$$\frac{qr+q^2}{2^{\delta+1}}\text{-}\textit{private}.$$

Theorem 4 (Information-Theoretic Authenticity of CCFB+H). *CCFB encryption using a random F is*

$$\left(\frac{qr+q^2}{2^{\delta+1}}+\frac{1}{2^\tau}\right)\text{-}\textit{authentic.}$$

6 Using a Single Random Function *f*

For CCFB, we can instantiate the random function $F:\{0,1\}^n \rightarrow \{0,1\}^n$ by a PRF – or by a block cipher E_K under a secret key K. Thus, CCFB can obviously be viewed as a block cipher mode of operation. But for CCFB+H, we need an additional random function $F':\{0,1\}^* \rightarrow \{0,1\}^\delta$, which is supposed to be independent from F.

We propose to use a single variable-input-length random function $f:\{0,1\}^* \rightarrow \{0,1\}^n$, defining F and F' by

$$\begin{array}{ll} F(x) = f(x) & \text{for } x \in \{0,1\}^n \\ F'(y) = \mathrm{MSB}_{|M_m|}(f(0^n||y)) & \text{for } y \in \{0,1\}^* \end{array}$$

By the definition of CCFB and CCFB+H, the first τ bits of any input for F represent a number between 1 and $2^{\tau-1}$, i.e. are never zero. Thus, inputs x and $0^n||y$ for f are never the same,[6] and F and F' behave exactly like two independent random functions.

7 Instantiating *f* by OMAC

OMAC [5,6], described in Figure 5,[7] is a message authentication code under a function $E_K:\{0,1\}^n \rightarrow \{0,1\}^n$. It

- can use any block cipher or PRF as the underlying primitive,
- uses a block cipher E only in encryption mode,
- uses a single block cipher (or PRF key) K,
- and is provably secure in the standard model, see Theorem 5 for OMAC's **information-theoretical security as a variable-input-size PRF** in a concrete security setting.

Theorem 5 (Lemma 5.2 of [6]). *Consider OMAC under a random permutation $E_K:\{0,1\}^n \rightarrow \{0,1\}^n$. An adversary asking at most q' queries, each at most $\mu < 2^n/4$ blocks long, cannot distinguish OMAC from a random function with an advantage exceeding*

$$\frac{(5\mu^2+1)q'}{2^n}.$$

[6] In fact, we could replace $0^n||y$ by $0^\tau||y$. The only reason why we propose the longer 0^n-prefix is the improved efficiency for our OMAC based instantiation of f.

[7] [6] describes two flavours of OMAC, OMAC1 and OMAC2. In this paper, we set OMAC=OMAC1, but we could use OMAC2 just as well.

For the definition of u and "$*$" in GF(2^n) see [5,6]. We stress that computing $L*u$ and $L*u^2$ can be done very efficiently by shifting and conditional XORing.

```
Algorithm: OMAC.
Init: $L_0 := E_K(0)$; $L_1 := L_0 * u$; $L_2 := (L_0 * u^2)$; (* in GF($2^n$) *)
Input: $X \in \{0,1\}^n$.
    1. parse $X$ as $X_1, \dots X_m$;
    2. $Z := 0^n$;
    3. for $i$ in $1 \le i \le m-1$: $Y := X_i \oplus Z$;
                                     $Z := E_K(Y)$;
    4. if $|M_m| = n$ then $Y := Y \oplus L_1$;
                      else $Y := Y \oplus L_2$;
Output: authentication tag $E_K(Y)$.
```

Fig. 5. OMAC

Thus, we propose to instantiate f by OMAC under a block cipher E (e.g., E=AES) and a secret block cipher key K. The performance figures are:

- Computing $F'(H) = \text{OMAC}_K(0^n||H)$ can be done by calling the block cipher E_K only $\lceil |H|/n \rceil$ times. The first iteration of the loop in Figure 5 can easily be optimised away, since it produces $E_K(0) = L_0$, which has been computed before, in the initialisation phase.
- Each computation of a value $F(C_{i-1}, i)$ or $F(C', m+d)$ boils down to a single block cipher call.

⇒ Computing CCFB+H(H,M) thus needs

$$\left\lceil \frac{|H|}{n} \right\rceil + \left\lceil \frac{|M|}{\delta} + 1 \right\rceil \text{ block cipher calls.} \tag{3}$$

8 A Comparison: EAX ↔ CCFB+H

In this section, we extrapolate the performance of CCFB+H from EAX' performance. Based on these results, one can compare the performance of CCFB+H with other modes, such as CWC and GCM, and one can verify these findings by benchmarking CCFB+H directly.

This has not been done in the current paper, which's focus is on low-end systems. As stressed in [9], CWC has not been developed for low-end devices. CWC combines counter-mode encryption with a Carter-Wegman hash function over GF($2^{127} - 1$). Due to the heavy use of large-scale integer multiplications, CWC actually appears to be very unattractive for low-end devices. Similarly to CWC, GCM combines counter-mode encryption with a universal hash function, namely a polynomial hash over some binary field GF(2^w). Efficient software implementations would need large tables, i.e. more storage space than available on usual low-end systems. It thus seems natural to consider EAX as the main "competitor" for CCFB+H.

To the security architect, CCFB offers a trade-off between the size τ of the authentication tag and the size δ of the message blocks. This has an obvious impact on the performance, but also determines the security level. Table 1 highlights this. Apart from the bound on τ, what is the impact of replacing a term $\Theta(r^2/2^n)$ (for EAX) by $\Theta(qr/2^{n-\tau})$ (for CCFB+H)?

	provable privacy	provable authenticity	limit for τ
EAX	$\Theta\left(\frac{r^2}{2^n}\right)$	$\Theta\left(\min\left\{\frac{r^2}{2^n}, \frac{1}{2^\tau}\right\}\right)$	$\tau \leq n$
CCFB+H	$\Theta\left(\frac{qr}{2^{n-\tau}}\right)$	$\Theta\left(\min\left\{\frac{qr}{2^{n-\tau}}, \frac{1}{2^\tau}\right\}\right)$	$\tau \leq n - \delta$

$r = \sum m_i$: accumulated number of message blocks q: number of messages

Table 1. Asymptotical Security of EAX and CCFB+H.

The maximum message length for CCFB+H is $(n-\tau)(2^\tau - 3)$ bit, i.e. appxoximately 2^τ blocks. Thus, if the average message size is large, CCFB+H can be about as secure as EAX. On the other hand, CCFB+H has been designed with low-end devices in mind. Typical applications for low-end devices mostly transmit small messages. So let us consider a concrete example with small messages:

block cipher: E=AES, and thus $n = 128$,
tag size: $\tau = 32$, and thus $\delta = 128 - 32 = 96$,
number of messages in the lifetime of a secret key: $q \leq 2^{28}$
average message size: ≤ 16 blocks ($16 * \delta = 1536$ bit) $\Rightarrow r \leq 2^{32}$.

While EAX would provide better security than CCFB+H, we still get good privacy and almost the authenticity we would expect from an ideal MAC with 32-bit authentication tags:

$$\text{privacy (Thm. 3):} \quad \tfrac{qr+q^2}{2^{\delta+1}} \approx 2^{60}/2^{97} \approx 2^{-37}$$

$$\text{authenticity (Thm. 4):} \quad \tfrac{qr+q^2}{2^{\delta+1}} + \tfrac{1}{2^\tau} \approx 2^{-37} + 2^{-32} \approx 2^{-32}$$

The above results apply in an information-theoretic setting. Since we propose to use OMAC as a pseudorandom function, Theorem 5 comes into play. Note that each header H and each message block M_i in the CCFB+H setting is, from OMAC's point of view, a message of its own right – OMAC thus authenticates $q' = q+r$ messages. Theorem 5 also considers the length μ of the largest message (in blocks). By the specification of CCFB, we have $\mu \leq 2^\tau - 3$. Even if we assume $\mu \approx 2^\tau$, the advantage is bounded by

$$\text{the pseudorandomness of OMAC (Thm. 5):} \quad \frac{(5\mu^2+1)q'}{2^n} \leq 2^{-64}.$$

This is negligible, compared to the 2^{-37} and 2^{-32} from above.

Finally, we also compare the performance of the concrete CCFB+H example with the security of AES-based EAX. CCFB+H allows the precomputation of a header checksum $F'(H)$ in advance, before knowing M. EAX offers a similar feature. Thus, in Table 2, we consider authenticated encryption with and without header precomputation. It turns out that

- The header-dependent work is the exactly same for EAX and CCFB+H: Computing OMAC(H) by making $\lceil |H|/128 \rceil$ AES calls.
- Apart from the header-dependent work, we see the following:
  - For short messages ($|M| \leq 96$), CCFB+H makes two AES calls, while EAX makes three. E.e., CCFB+H is 50 % faster than EAX.
  - With $|M|$ increasing, CCFB+H is at least as fast as EAX ($97 \leq |M| \leq 128$), and at most 66.7 % faster ($128 \leq |M| \leq 192$).
  - In the long run, EAX makes about $|M|/64$ calls. CCFB+H with $|M|/96$ calls is 50 % faster.

	full computation	header has been preprocessed
EAX	$\left\lceil \frac{\lvert M\rvert}{128} \right\rceil + \left\lceil \frac{\lvert M\rvert}{128} \right\rceil + 1 + \left\lceil \frac{\lvert H\rvert}{128} \right\rceil$	$\left\lceil \frac{\lvert M\rvert}{128} \right\rceil + \left\lceil \frac{\lvert M\rvert}{128} \right\rceil + 1$
CCFB+H	$\left\lceil \frac{\lvert M\rvert}{96} \right\rceil + 1 + \left\lceil \frac{\lvert H\rvert}{128} \right\rceil$	$\left\lceil \frac{\lvert M\rvert}{96} \right\rceil + 1$

$|M|$ = message length $|H|$ = header length nonce-length $\leq$ 128 bit

Table 2. Performance of AES-based EAX and CCFB+H in # of AES calls.

Acknowledgement

The author thanks Ulrich Kühn for suggesting the name "Counter CFB", Nico Schmoigl for his survey on EAX and CWC, and the referees for their support in improving this presentation.

References

1. M. Bellare, P. Rogaway, D. Wagner. EAX: a conventional authenticated encryption mode. FSE 2004.
2. M. Bellare, P. Rogaway, D. Wagner. EAX: a conventional authenticated encryption mode. Extended version of [1]. http://www.cs.berkeley.edu/~daw/papers/eax-fse04.ps
3. M. Bellare, C. Namprempre. Authenticated Encryption: relations among notions and analysis of the generic composition paradigm. Asiacrypt 00.

4. V. Gligor, P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. FSE 01.
5. T. Iwata and K. Kurosawa. OMAC: One-Key CBC MAC. Fast Software Encryption, FSE 03.
6. T. Iwata and K. Kurosawa. OMAC: One-Key CBC MAC. Extended Version of [5]. http://crypt.cis.ibaraki.ac.jp/omac/docs/omac.pdf
7. C Jutla. Encryption modes with almost free message integrity. Eurocrypt 01.
8. J. Katz, M. Yung. Unforgeable encryption and adaptively secure modes of operation. FSE 00.
9. T. Kohno, J. Viega, D. Whiting. CWC: a high performance conventional authenticated encryption mode. FSE 04.
10. T. Kohno, J. Viega, D. Whiting. CWC: a high performance conventional authenticated encryption mode. Extended version of [9]. http://eprint.iacr.org/2003/106.ps.gz
11. D. McGrew, J. Viega. The Security and Performance of the Galois/Counter Mode of Operation (Full Version). http://eprint.iacr.org/2004/193
12. P. Rogaway. Authenticated encryption with associated data. Computer and Communications Security, ACM, 2002.
13. P. Rogaway, M. Bellare, J. Black, T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. Computer and Communications Security, ACM, 2001.
14. P. Rogaway, D. Wagner. A critique of CCM. Unpublished manuscript. February 2, 2003. http://www.cs.berkeley.edu/˜daw/papers/ccm.html
15. D. Whiting, R. Hously, N. Ferguson. Counter with CBC-MAC (CCM). Submission to NIST.

Appendix: Deferred Proof and Figures

Theorem 2 (Information-Theoretic Authenticity of CCFB)
CCFB encryption, using a random F, is

$$\left(\frac{qr}{2^{\delta+1}} + \frac{1}{2^\tau}\right)\text{-authentic with respect to nonce-respecting adversaries and}$$

$$\left(\frac{q(r+q)}{2^{\delta+1}} + \frac{1}{2^\tau}\right)\text{-authentic with respect to nonce-randomising adversaries.}$$

Proof. We will show that the chance to succeed in forging a message *without* having found an input-collision is at most $1/2^\tau$. The claimed theorem then follows from Lemma 1.

The adversary's knowledge about the local authentication tags A^i_j can be described by

$$q \text{ linear equations} \qquad T^i = A^i_1 \oplus \bigoplus_{2\le j\le m_i+1} A^i_j \qquad \text{with } 1 \le i \le q$$

over $\mathrm{GF}(2^\tau)$. We stress that only the T^i are known – the unknowns A^i_j ($j \ge 1$) are uniformly distributed independent random values from $\mathrm{GF}(2^\tau)$ (since we assumed no input-collision). Due to the statistical independence of the A_1^i, all q linear equations are linearly independent.

A forgery (C_0, C) with $C = (C_1, \ldots, C_{m-1}, C_m, T)$ succeeds if and only if C is different from all the other ciphertexts C^i and the linear equation

$$T = A_1 \oplus \bigoplus_{2 \le j \le m+1} A_j \tag{4}$$

holds. We claim that Equation 4 is linearly independent from the q equations above. I.e., we show that the sum of Equation 4 with any subset of equations $T^i = \ldots$ is the sum of some non-dissappearing unknowns A_j^i or A_j with $j \ge 1$ and $1 \le i \le q$.

If $D_0 \notin \{D_0^1, \ldots, D_0^q\}$,[8] the term A_1 cannot dissappear. So assuming w.l.o.g. $D_0 = D_0^1$, this is equivalent to $C_0 = C_0^1$, from which $A_1^1 = A_1$ follows. By adding T and T^1, we get

$$T^1 \oplus T = \bigoplus_{2 \le j \le m_1+1} A_j^1 \oplus \bigoplus_{2 \le j \le m+1} A_j.$$

Any terms $A_j^1 = A_j$ cancel out if $D_j = D_j^k$. We define the set

$$A^* = \{A_j^1 | 2 \le j \le m_1 + 1, D_j^1 \ne D_j\} \cup \{A_j | 2 \le j \le m + 1, D_j \ne D_j^1\}$$

of terms which don't cancel out and rewrite $T^1 \oplus T$ as

$$T^1 \oplus T = \bigoplus_{A \in A^*} A.$$

Since $C \ne C^i$, the set A^* is not empty.[9] For $i > 1$, each equation $T^i = \ldots$ with $i > 1$ added to $T^1 \oplus T$ introduces a non-disappearing term A_1^i to the sum. Thus, equation 4 is linearly independent from the equations for the T^i, as claimed.

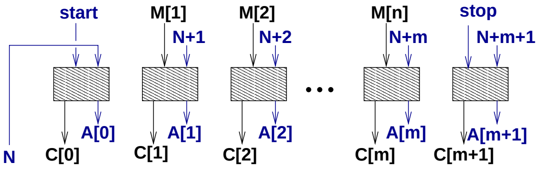
Since Equation 4 is linearly independent from the q equations for the T^i, the sum $T = A_1 \oplus \bigoplus_{2 \le j \le m+1} A_j$ can take any value in $T \in \mathrm{GF}(2^\tau)$, and the number of solutions for each T is the same. All $T \in \mathrm{GF}(2^\tau)$ are equally likely to be the "correct" solution, which finally yields the claimed probability $1/2^\tau$. □

[8] The inputs D_j for F are defined similarly to the D_j^i in Equation 1.

[9] Technically, $C \ne C^i$ could mean $T \ne T^i$, $m = m_i$, and $C_j = C_j^i$ for $1 \le j \le m$. But this type of forgery would fail: $A^* = \{\}$ and thus $T^1 \oplus T = 0$, contradicting $T \ne T^i$.

```
**Algorithm:** CCFB decryption.
**Input:** nonce $N \in \{0,1\}^\delta$ and $C \in \{0,1\}^*$, $\tau + 1 \le |C| \le (2^\tau - 3)\delta + \tau$;
**First pass:**
    1. parse $C$ as $(C_1, \ldots, C_m, T)$ with
           $|C_1| = \cdots = |C_{m-1}| = \delta$, $|C_m| \in \{1, \ldots, \delta\}$, $|T| = \delta$;
    2. $C_0 := N$;
    3. for $1 \le i \le m-1$:    $(\mathrm{tmp}, A_i) := F(C_{i-1}, \langle i \rangle_\tau)$;
                                 $M_i := \mathrm{tmp} \oplus C_i$;
    4. $(\mathrm{tmp}, A_m) := F(C_{m-1}, \langle m \rangle_\tau)$;
    5. if $|C_m| = \delta$ then $d := 1; \mathrm{pad} := ()(*\text{empty string}*)$;
                    else $d := 2; \mathrm{pad} := (1||0^{\delta - |C_m| - 1})$;
    6. $M_m := \mathrm{MSB}_{|C_m|}(\mathrm{tmp}) \oplus C_m$;
    7. $C' := \mathrm{tmp} \oplus (M_m||\mathrm{pad})$;
    8. $(\mathrm{dummy}, A_{m+1}) := F(C', \langle m + d \rangle_\tau)$;
**Second pass:**
    9. $T' := A_1 \oplus \cdots \oplus A_{m+1}$;
**Output:** If $T = T'$
            then output plaintext $(M_1, \ldots, M_m)$ with
                $M_1, \ldots, M_{m-1} \in \{0,1\}^\delta$, and $M_m \in \{0,1\}^{|M_m|}$
        else output (none).
```

Fig. 6. CCFB decryption under $F : \{0,1\}^\delta \times \{0,1\}^\tau \to \{0,1\}^\delta \times \{0,1\}^\tau$.



plaintext: $(M_1, \ldots M_m) \in \{0,1\}^{\delta m}$, $M_i \not\in \{\mathrm{start}, \mathrm{stop}\}$;
nonce: $N \in \{0,1\}^\tau$; ciphertext: $((C_0, A_0), \ldots, (C_{m+1}, A_{m+1})) \in \{0,1\}^{n(m+2)}$

Fig. 7. RPC encryption under a permutation