CMCC: Misuse Resistant Authenticated Encryption with Minimal Ciphertext Expansion

Jonathan Trostle
Consultant

Abstract

In some wireless environments, minimizing the size of messages is paramount due to the resulting significant energy savings. We present CMCC, an authenticated encryption scheme with associated data (AEAD) that is also nonce misuse resistant. The main focus for this work is minimizing ciphertext expansion, especially for short messages including plaintext lengths less than the underlying block cipher length (e.g., 16 bytes). For many existing AEAD schemes, a successful forgery leads directly to a loss of confidentiality. For CMCC, changes to the ciphertext randomize the resulting plaintext, thus forgeries do not necessarily result in a loss of confidentiality which allows us to reduce the length of the authentication tag. For protocols that send short messages, our scheme is similar to Counter with CBC-MAC (CCM) for computational overhead but has much smaller expansion. We prove both a misuse resistant authenticated encryption (MRAE) security bound and an authenticated encryption (AE) security bound for CMCC. We also present a variation of CMCC, CWM, which provides a further strengthening of the security bounds. Our contributions include both stateless and stateful versions which enable minimal sized message numbers using different network related trade-offs.

Keywords: Energy constrained cryptography, authenticated encryption.

1 Introduction

The current paradigm of providing confidentiality and integrity protection for distributed applications through the use of encryption combined with MAC's (Message Authentication Codes) is reasonably efficient for many environments. In particular, for network message sizes that range from several hundred bytes or more, having MAC's that utilize 8-20 bytes is not unduly inefficient. For resource constrained environments, where message lengths are often less than one-hundred bytes, existing MAC's impose a more significant overhead. Since it requires more energy to send longer messages, it is important to reduce message sizes in protocols used by wireless devices. This need becomes even more critical for low bandwidth networks.

A key reason that MAC's need to be long is that the most popular symmetric block cipher modes can be predictively modified by an attacker. Counter mode (CTR) can be modified by flipping bits so the attacker can precisely control the changes to the message. Cipher Block Chaining (CBC) can be modified such that changes to one block are predictable while the preceding block is randomized (see [Bel96] for attacks that utilize this property). Also, the most common schemes for CCA (Chosen Ciphertext Attack) security [KY00] utilize a CPA (Chosen Plaintext Attack) encryption scheme combined with a MAC (Message Authentication Code) [DDN00].

In this paper we present a new authenticated encryption mode, CMCC. CMCC utilizes a pseudorandom function (PRF) (e.g., AES but other choices are possible). Our construction uses multiple invocations of the PRF so that any modifications to ciphertext result in a randomized plaintext.

CBC-MAC-CTR-CBC (CMCC) mode is a general purpose authenticated encryption mode [BN00]. We apply CBC encryption in the first round, use a MAC followed by a CTR mode in the 2nd round, and CBC encryption again in the 3rd round (see Figures 1 - 3). We prove that CMCC is misuse resistant [RS06]: encryptions using the same message number, plaintext, and associated data are identifiable to the adversary as such, but security is preserved if the same message number is reused where either the plaintext or associated data is distinct. Since changes to the ciphertext randomize the resulting plaintext, with high probability, we achieve authentication by appending a string consisting of τ bits set to zero to the plaintext prior to encryption. Relative to SIV [RS06], CMCC has smaller ciphertext expansion.

We obtain MRAE and AE security with competitive security bounds using only a small number of bytes of ciphertext expansion, for a full range of message sizes.

We will make use of variable length input pseudorandom functions f_i. In order to better understand the intuition behind our scheme, consider the case where the plaintext is the concatenation of the strings P_1 and P_2 where each string's length equals the pseudorandom function output size (e.g., 16 bytes in the case of AES). Consider the scheme:

$$\begin{aligned} X &= f_3(W, P_1) \oplus P_2 \\ X_2 &= f_2(W, X) \oplus P_1 \\ X_1 &= f_1(W, X_2) \oplus X \end{aligned}$$

where the ciphertext is X_1, X_2, and W is an unpredictable pseudorandom value. For maximum security, W is unique, with high probability, for each message encrypted under a given key K. Then if the adversary flips some bits in X_1, the corresponding bits in X are flipped during decryption, and this produces random changes to P_1 during decryption (see 2nd equation). The first equation is then applied which results in random changes to P_2. A similar argument applies if we flip one or more bits in X_2. Since changes to any bits in the ciphertext result in random changes to the plaintext, we will see that the authentication tag can be a string of zero bits appended to the plaintext, and that the corresponding term in the security bound, due to this ciphertext expansion, is smaller than in comparable schemes.

A common scenario is one where some packet loss and/or packet reordering may occur so that the communication peers aren't fully synchronized. We present two versions of our scheme with different trade-offs to handle loss of synchronization. The stateless version uses a public message number and its size is constrained thus limiting the number of messages that can be encrypted under a single key while avoiding resuse of the message numbers. The stateful version uses a private message number which is encrypted and the last few bytes of the resulting encryption are sent with the ciphertext. This mechanism enforces a different trade-off; the limit here is on the maximum amount of disorder between encryption order and decryption order. It also hides the number of messages previously sent.

1.1 Definitions for Authenticated Encryption (AE)

We give motivation for our definition of authenticated encryption.

Consider OCB or a counter mode variant (e.g., GCM) with a 4 byte authentication tag. Then for the AE security game (see Section 2.1.1 for definition), submit the message (plaintext) with all 1's and also the message with all 0's. The adversary obtains a ciphertext response corresponding to one of the plaintexts. Then randomly flip bits in this ciphertext for each new ciphertext query and attach a random authentication tag. Then the probability of winning is $q(2^{-32})$. The reason is that this bound is the probability that one of the submitted ciphertexts is valid. If it's valid then we get the plaintext back which shows us the bits that we flipped. And if the flipped bits are zero, then the original message had all 1's and vice versa. Now compare this to CMCC with a 4 byte zero bit authentication string. Then our AE security bound is approximately $q(q-1)(2^{-65})$ for a 12 byte message. Thus CMCC has stronger AE security given a short authentication tag. If we run the same attack against CMCC as in the preceding paragraph, then the probability of a valid ciphertext is approximately the same. But the corresponding plaintext would be randomized with high probability and thus would give us no information about the challenge plaintext.

The MRAE–AE definition in [RS06] does not distinguish between the security levels in the two cases above, but the PRI (Pseudo Random Injection) definition in [RS06] does distinguish them.

This distinction becomes more important given short authentication tags; in particular, classifying a forgery as a a complete loss of security is not always appropriate. Depending on the application, a single forgery may not be enough to disrupt the application (e.g., VoIP), and depending on the encryption scheme, it may be detectable during higher layer protocol checks. Our security definition should be general enough to handle the case of a valid ciphertext query where changes to the ciphertext randomize the resulting plaintext so that the upper layer protocol checks detect and reject the message. (None of our security bounds include any factor related to upper layer protocol checks.)

Our definition gives the Adversary encryption and decryption oracles (real world) vs. a random injection function and its inverse and asks the Adversary to distinguish between the two (see Section 2). This definition is the same as the PRI definition in [RS06].

1.2 Applications

For constructing a secure channel (with both confidentiality and authentication) using our encryption scheme, it follows that we can shorten or eliminate our MAC tag since the adversary cannot make a predictable change to the encrypted message, as in many counter-mode based schemes. (These other schemes depend on the MAC to detect such a change). With our scheme, a change to the packet is highly likely to cause the packet to be rejected due to a failure to satisfy application protocol checks. Another possibility (e.g., Voice over IP (VoIP)) is that the randomized packet will have a minimal effect. With only a small probability can the adversary achieve a successful integrity attack. Since network transmission and reception incurs significant energy utilization, it follows that we can expect to achieve significant energy savings. Our analytical results for wireless sensor networks show that energy utilization is proportional to packet length, and that the cryptographic computational processing impact on energy use is minor.

If we consider VoIP, a 20 byte payload is common. The transport and network layer headers (IP, UDP, and RTP) bring another 40 bytes, but compression [CJ99, BBD$^+$01] is used to reduce these fields down to 2-4 bytes. The link layer headers add another 6 bytes. Thus the total packet size is 30 bytes, assuming the UDP checksum of 2 bytes is included. In this case, by omitting the recommended 10 byte authentication tag and using CMCC with 2 bytes of expansion, we obtain a 1/5 savings in message size and corresponding savings in energy utilization. Furthermore if the encryption boundary is just after the CID field (which is used to identify the full headers), then the UDP checksum is encrypted and acts as a 2 byte authentication tag. Even if the adversary was lucky enough to obtain the correct checksum, the resulting Voice payload would be noise, with high probability.

Wireless sensor networks also use short packets [VA08] to maximize resource utilization; these packets are often in the range of 10-30 bytes. For the adversary, large numbers of queries are likely to be either impossible or highly anomalous in these constrained low bandwidth networks.

1.3 Our Contributions

Our contributions are as follows:

1. We give a new family of private key encryption schemes with minimal ciphertext expansion. We obtain AE security with a competitive security bound using only a small number of bytes of ciphertext expansion, for a full range of message sizes. When message numbers are not reused for CMCC, we obtain a security bound which is dominated by $q(q-1)2^{-\tau-1}(1/\beta + 2^{-\tau}) + 2e(q-1)/\beta$ where $\beta = min\{\alpha, 2^B\}$, B is the block cipher block length in bits, and $\alpha = 2^{8m}$ where Len is the byte length of the minimal length plaintext query response, $m = \lfloor Len/2 \rfloor$ and τ is the bit length of the authentication tag.

2. CMCC is a general purpose misuse resistant authenticated encryption mode. We define security for misuse resistant authenticated encryption and prove a MRAE security bound for CMCC. CMCC has less ciphertext expansion than SIV [RS06]. In particular, the ciphertext expansion τ due to the SIV IV contributes a $q(q-1)/2^{\tau}$ term to the SIV security bound, whereas the CMCC ciphertext expansion due to the authentication tag adds a $q(q-1)/2^{2^{\tau}}$ term to the CMCC AE bound, and a $q/2^{\tau}$ term to the CMCC MRAE security bound.

3. We give both stateless and stateful versions of our schemes where we minimize message number sizes in both versions. As discussed above, each version enables a different trade-off based on the network and application parameters.

4. We present a variant of CMCC, CMCC with MAC, or CWM. CWM replaces the authentication tag consisting of zero bits in CMCC with an authentication tag consisting of a MAC computed over the plaintext. When message numbers are not reused for CWM, we obtain a security bound which is dominated by $q^2/2^{3\tau} + q^2/(2^{2\tau}\beta) + q/(2^{\tau-1}\beta)$ and if message numbers can be reused then we obtain a bound dominated by $q/2^{2\tau-1} + q^2/2^{3\tau-1} + q^2/(2^{2\tau}\beta) + q/(2^{\tau-1}\beta) + q^2/(2\alpha) + q^2/(2\beta)$.

5. We give a rough comparison for CPU overhead, network overhead, and energy consumption between CCM and CMCC, where energy is based on a wireless sensor node, the Mica2Dots platform. CMCC uses less energy since its ciphertext expansion is smaller, while the number of block cipher invocations is similar.

1.4 Related Work

There was originally work in the IETF IPsec Working Group on a confidentiality-only mode; the original version of ESP provided confidentiality without integrity protection [Atk95]. However, [Bel96] showed that CBC and stream-cipher like constructions were vulnerable to attacks that could be prevented by adding a MAC.

Given a message with redundancy, the idea that authenticity can be obtained by enciphering it with a strong pseudorandom permutation goes back to [BR00]. The authors formally prove a bound on adversary advantage against authenticity which requires that the probability that an arbitrary string decodes to a valid message is low. In [AB01], the authors show that public redundancy is not always sufficient and that private (keyed) redundancy leads to stronger authentication properties. Struik [Str11] presented application requirements and constraints, independently of this work at roughly the same time this work was started.

In [Des00], Desai gives CCA-secure symmetric encryption algorithms that don't use a MAC and don't provide explicit integrity protection outside of the CCA-security. The most efficient one is UFE which utilizes variable length pseudorandom functions. Its ciphertext expansion is $|r|$ bits where r is a uniform random value; security can be compromised if the same r is used for multiple messages. Since r is uniform random, collisions are likely after $2^{|r|/2}$ messages. The UFE security bound is $q(q+1)/2^{|r|}$. If the adversary can make 2^{20} queries, then Theorem 4.6 gives a security bound around 2^{-57} for CMCC with a 6 byte authentication string, given a 14 byte message. UFE would require a 13 byte ciphertext expansion to assure the same security level.

Rogaway and Shrimpton introduced misuse resistant authenticated encryption (MRAE) in the seminal paper [RS06], where they present the MRAE schemes SIV and PTE. SIV includes a MRAE scheme where the expansion includes the block cipher block size (e.g., 16 byte) IV plus the nonce. Thus CMCC is a MRAE scheme with smaller expansion (which is important for short messages), and comparable security for applications that require less than a 16 byte MAC. Some applications can utilize a 4 byte or smaller MAC and meet security requirements. The SIV ciphertext expansion adds a $q(q-1)/2^{\tau}$ term to the SIV security bound, while the CMCC ciphertext expansion adds a $q(q-1)/2^{2\tau}$ term to the CMCC AE bound, and a $q/2^{\tau}$ term to the CMCC MRAE security bound. The RFC 5297 specification of SIV has the same number of block cipher invocations as CCM, and roughly the same number of block cipher invocations as CMCC (see Table 5). Our security definition is the same as the PRI security definition in [RS06].

CMCC uses the same authentication construction as PTE. However, the TES that [RS06] recommends for PTE is not capable of encrypting messages with less than the block size of the underlying block cipher.

Collisions in the IV [RS06] (or random message number in [Des00]) will result in loss of privacy for the affected messages. Thus security is increased if the IV is long (e.g., 16 bytes for SIV). In other words, decreasing ciphertext expansion results in less security. Security for our scheme increases as message length grows, so privacy is stronger when ciphertext expansion is minimal, given message lengths between 10 and 32 bytes. The parameter X in our scheme is similar to the σ parameter in [Des00] and to the IV in [RS06]. These last two parameters create ciphertext expansion whereas X does not. Our scheme is targeted at environments where minimizing ciphertext expansion is valuable.

Other fully nonce-misuse resistant schemes include AEZ [HKR15], HS1-SIV [Kro14], Julius [Bah14], MRO [GJMN16], HBS [IY09b], and BTM [IY09a] with the first three being Caesar Authenticated Encryption competitors along with CMCC. Of the above schemes, similarly to CMCC

AEZ addresses smaller length messages and minimal ciphertext expansion. The ciphertext expansion, or stretch, is a user controlled parameter that is an input to the encryption function. The AEZ paper does not give a security bound when message length plus stretch is less than 16 bytes. For some message/stretch sizes between 16 and 32 bytes, the CMCC security bounds are stronger. AEZ also makes use of a nonstandard 4 round AES function.

Processing performance for CMCC is similar to SIV, whereas the above schemes are significantly more efficient (for processing but not energy usage) than SIV.

[BZD$^+$16] surveys Internet facing https servers and proxies to detect nonce reuse for AES-GCM in TLS. Their study uncovered nonce reuse thus showing the value of nonce-misuse resistance.

Shrimpton and Terashima [ST13] use a 3 round unbalanced Feistel network approach to obtain schemes TCT1 and TCT2 where the latter has BBB (Beyond Birthday Bound) security for longer messages (messages of length $\geq 2n$ where the underlying blockcipher has length n. Both schemes are STPRP's (Strong Tweakable PRP's, e.g., the adversary may reuse tweaks.)

There is additional work in the area of small domain encryption including [RY13].

1.5 Organization

In Section 2, we give basic cryptographic definitions. In Section 3, we present the CMCC authenticated encryption scheme with minimal ciphertext expansion. Section 4 gives the proof that establishes security bounds for CMCC authenticated encryption and misuse resistant authenticated encryption. We also present CWM in this section. Section 5 gives our performance analysis and results, including a comparison of energy utilization between CMCC and CCM, for wireless sensor nodes. In Section 6 we draw conclusions.

2 Definitions

2.1 Pseudorandomness

All strings are binary strings (if S is a string, then $S \in \{0,1\}^*$.) The concatenation of two strings S and T is denoted by $S||T$, or S, T where there is no danger of confusion. For a string S, $|S|$ is its length (in bits). If $1 \leq i \leq j \leq |S|$, then $S[i \ldots j]$ is the substring from the *ith* to the *jth* characters, inclusive.

We write $w \leftarrow W$ to denote selecting an element w from the set W using the uniform distribution. We write $x \leftarrow f()$ to denote assigning the output of the function f, or algorithm f, to x. S^C denotes the complement of set S.

Throughout the paper, the adversary is an algorithm which we denote as $\mathcal{A}$.

We follow [GGM86] as explained in [Sho04] for the definition of a pseudo-random function: Let l_1 and l_2 be positive integers, and let $\mathcal{F} = \{h_L\}_{L \in K}$ be a family of keyed functions where each function h_L maps $\{0,1\}^{l_1}$ into $\{0,1\}^{l_2}$. Let H_{l_1,l_2} denote the set of functions from $\{0,1\}^{l_1}$ to $\{0,1\}^{l_2}$.

Given an adversary $\mathcal{A}$ which has oracle access to a function in H_{l_1,l_2} or $\mathcal{F}$. The adversary will output a bit and attempt to distinguish between a function uniformly randomly selected from $\mathcal{F}$ and a function uniformly randomly selected from H_{l_1,l_2}. We define the PRF-advantage of $\mathcal{A}$ to be

$$Adv_{\mathcal{F}}^{prf}(\mathcal{A}) = |Pr[L \leftarrow K : \mathcal{A}^{h_L}() = 1] - Pr[f \leftarrow H_{l_1,l_2} : \mathcal{A}^f() = 1]|$$

$$Adv_{\mathcal{F}}^{prf}(q,t) = \max_{\mathcal{A}}\{Adv_{\mathcal{F}}^{prf}(\mathcal{A})\}$$

where the maximum is over adversaries that submit at most q queries and run in time t.

Intuitively, $\mathcal{F}$ is pseudo-random if it is hard to distinguish a random function selected from $\mathcal{F}$ from a random function selected from H_{l_1,l_2}.

We also define $Adv_{\mathcal{F}}^{prp}(q,t)$ in the same manner where the comparison is with a random permutation and $\mathcal{F}$ is a family of keyed permutations.

2.1.1 Authenticated Encryption (AE) and Misuse Resistant Authenticated Encryption (MRAE)

Given plaintext (message) set $\mathcal{P}$, associated data set $\mathcal{AD}$, ciphertext set $\mathcal{C}$, key set $\mathcal{K}$, header string set $\mathcal{H}$, and message number set $\mathcal{N}$. An authenticated encryption scheme (AE) is a tuple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ such that $\mathcal{E} : \mathcal{K} \times \mathcal{H} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{P} \to \mathcal{C}$, $\mathcal{D} : \mathcal{K} \times \mathcal{H} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{C} \to \mathcal{P} \cup \{\perp\}$, and $\mathcal{D}(K, H, N, A, \mathcal{E}(K, H, N, A, P)) = P$ for all $H \in \mathcal{H}, N \in \mathcal{N}, A \in \mathcal{AD}$, $P \in \mathcal{P}$. If there is no $P \in \mathcal{P}$ such that $C = \mathcal{E}(K, H, N, A, P)$, then $\mathcal{D}(K, H, N, A, C) =\perp$. We write D_K and E_K in place of $\mathcal{D}(K, ...)$ and $\mathcal{E}(K, ...)$.

For our security definition, we define the ideal world object as a random injective function. The expansion function is $e : \mathcal{H} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{P} \to \mathbb{N}$. The expansion function depends only on the length of its arguments. Let $Inj_e^{\mathcal{H},\mathcal{N},\mathcal{A}}(\mathcal{P}, \mathcal{C})$ be the set of injective functions f from $\mathcal{H} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{P}$ into $\mathcal{C}$ such that $|f(H, N, A, P)| = |P| + e(|H|, |N|, |A|, |P|)$.

Let $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be an AE with message space $\mathcal{P}$, associated data set $\mathcal{AD}$, header string set $\mathcal{H}$, message number set $\mathcal{N}$, and expansion e. The AE-advantage of adversary $\mathcal{A}$ against Π is

$$\begin{aligned}\mathbf{Adv}_{\Pi}^{AE(q,t,\mu)}(\mathcal{A}) = Pr[K \leftarrow \mathcal{K} : \mathcal{A}^{\mathcal{E}_K(.,.,.,.)\mathcal{D}_K(.,.,.,.)} \Rightarrow 1] - \\ Pr[f \leftarrow Inj_e^{\mathcal{H},\mathcal{N},\mathcal{A}}(\mathcal{P}, \mathcal{C}) : \mathcal{A}^{f(.,.,.,.),f^{-1}(.,.,.,.)} \Rightarrow 1]\end{aligned}$$

when encryption oracle queries use unique message numbers and $\mathcal{A}$ is restricted to asking q queries totaling μ blocks in running time t. $f^{-1}(H, N, A, C) = P$ if $f(H, N, A, P) = C$ and returns $\perp$ if no such tuple (H, N, A, P) exists. We define MRAE-advantage and $Adv_{\Pi}^{MRAE(q,t,\mu)}$ analogously except encryption oracle queries are allowed to repeat message numbers. We also define $Adv_{\Pi}^{AE(q,t,\mu)} = \max Adv_{\Pi}^{AE(q,t,\mu)}(\mathcal{A})$ over all adversaries $\mathcal{A}$ that ask q queries totaling μ blocks in time t. We define $Adv_{\Pi}^{MRAE(q,t,\mu)} = \max Adv_{\Pi}^{MRAE(q,t,\mu)}(\mathcal{A})$ over all adversaries $\mathcal{A}$ that ask q queries totaling μ blocks in time t for the MRAE environment where message numbers may be repeated in encryption oracle queries. We will also consider the case where the game is restricted if the adversary submits a decryption oracle query which returns $\perp$; in this case, the adversary will not be allowed to make additional oracle queries prior to its output. We define $Adv_{\mathcal{E}}^{priv}(\mathcal{A}) = |Pr[L \leftarrow K : \mathcal{A}^{E_L}() = 1] - Pr[\mathcal{A}^{\$} = 1]|$ for encryption scheme $\mathcal{E}$ with expansion τ where \$ returns a random string with τ plus the input string's bitlength bits. We also define $Adv_{\mathcal{E}}^{priv}(q, t, \mu) = \max Adv_{\mathcal{E}}^{priv}(\mathcal{A})$ over all adversaries $\mathcal{A}$ that ask q queries totaling μ blocks in time t. $CTR_K(N, P)$ denotes Counter Mode encryption with key K, nonce N, and plaintext P.

3 CMCC

In this section, we present CMCC. CMCC includes a stateless version with public message numbers, and a stateful version with private message numbers. The stateless version has full misuse resistance against reuse of the message numbers, whereas the stateful version has resistance as well, but some private message numbers may result in decryption failures if too far outside the decrypt window.

Figures 1 - 3 describe the stateless version of CMCC, and Figures 4 - 5 describe the stateful version.

3.1 CMCC Stateless Encryption

We now present CBC-MAC-Counter-CBC (CMCC) mode. CMCC is a general purpose authenticated encryption mode which is misuse resistant and optimized for energy constrained environments.

3.1.1 Overview

For stateless version encryption, we initially utilize CBC mode and obtain the value X. Here we utilize $E_{\bar{K}}$ to create the CBC IV W from the message number M. This prevents the adversary from being able to manipulate M and P_1 in a way that allows collisions in X values to be created. Then we apply a MAC algorithm to W, X and use the result as the IV for counter mode encryption to encrypt P_1 and obtain X_2. Note that if the message has length less than or equal to 32 bytes, then the output of the MAC function is xor'd with P_1 to obtain X_2 and additional counter blocks are not needed. Finally we create the other half of the ciphertext, X_1 using CBC mode applied to X_2 and exclusive-or with X.

3.1.2 Notation

We use $\oplus$ to denote bitwise xor. When we xor two strings with different lengths, the longer string is first truncated to the length of the shorter string. b^j is the bit b repeated j times. S^j denotes the bit string S repeated j times. Thus $(0110)^2 = 01100110$. $A\ and\ B$ is the logical AND operation on two equal length strings A and B. The notation $R_{128} = 0^{120}10000111$ denotes the bit string with 120 zero bits, followed by the bits 1,0,0,0,0,1,1, and 1. $x << n$ denotes the left shift operator (filling vacated bits with zero bits), after shifting the string x by n bits to the left. B denotes the block length of the underlying block cipher (128 bits for AES). E_k denotes encryption using the block cipher and input key k.

$LSB_j(x)$ and $MSB_j(x)$ denote the j least significant bytes and j most significant bytes of byte string x respectively.

3.1.3 Padding

We will apply the padding scheme from the AES-CMAC algorithm to our mode when CBC encryption is performed. One difference is that we will sometimes need to pad by a full block length ($B/8$ bytes)[1] and we use the same padding scheme as when the padding is between 1 and $B/8-1$ bytes.

1. Given the CBC encryption key K, and byte strings S_1 and S_2, where $|S_1| \leq |S_2|$. We define $pad(S_1)_{S_2}$ as follows:

[1] If S_1 is a multiple of B and S_2 is one byte longer, than we pad S_1 with $B/8$ bytes. If both strings are the same length which is a multiple of B then we do not add any padding bytes.

2. *pad_length* is the number of bits (which is a multiple of 8) needed to bring S_1 up to the length of S_2 and then bring S_1 up to a multiple of the block size. More formally,

$$pad\_length = |S_2| - |S_1| + B - (|S_2| \mod B)$$

   where mod values are taken between 1 and B.

3. We define $L = E_K(0^B)$. If the most significant bit of L is zero, then define $K1 = L << 1$, otherwise, we define $K1 = (L << 1) \oplus R_{128}$. If the most significant bit of $K1$ is zero, then define $K2 = K1 << 1$. Otherwise, we define $K2 = (K1 << 1) \oplus R_{128}$.

   If $pad\_length = 0$, then $|S_1|$ is a multiple of B; let F be the last block of S_1. We define $pad(S_1)_{S_2}$ to be S_1 with its last block replaced with $F \oplus K1$.

   If $1 \leq pad\_length \leq B$, then we append the following string to the last (possibly empty) block F of $S_1 : 10^{pad\_length-1}$. We denote this string as $\bar{S}_1$. $pad(S_1)_{S_2}$ is $\bar{S}_1$ with the last B bits of $\bar{S}_1$ replaced with $F||10^{pad\_length-1} \oplus K2$.

3.2 CMCC Stateful Encryption - Informal Design Intuition for Private Message Numbers

For stateful encryption, the only difference is in how the message numbers are handled: the message number tag is $T = LSB_{IL}(E_{\bar{K}}(i))$ for message number i. This follows the description in Section 3.3.

We allow the caller to use private message numbers. In this case,

$$E_{\bar{K}}(i)) = M_i, i \geq 0,$$

for private message number i where encryption key $\bar{K}$ is shared by the communication peers for the block cipher E. If the sender and receiver communication is synchronized, then M doesn't need to be transmitted. Otherwise, we send the least significant 2-3 (IL) bytes of the value M_i as described above except we eliminate M_i values from the sequence if the least significant IL byte(s) duplicate a previous M_j's least significant IL byte(s) where $(\gamma - j) \leq 2(window\_size) + 1$ given M_i as the γth element in the sequence (after eliminating previous last IL-byte duplicates and M_j is the jth element of the resulting sequence). In other words, M_i's that are close together are selected to have distinct least significant byte(s). This does require a small amount of additional computation to compute the sequence of M_i values but doesn't require significant additional work over the case where the least significant bytes are allowed to collide (since $2(window\_size) + 1$ will be less than the birthday bound). The *window_size* parameter (w_s) controls how much the encryptor and decryptor are allowed to fall out of synchronization.

Private message numbers allow the number of messages previously sent to be hidden and also minimize the number of bytes transmitted on the wire but the scheme is stateful.

3.3 CMCC Private Message Numbers

The two communication peers are denoted as the initiator (*init*) and responder (*resp*), respectively. There are two channels; one with the initiator as the encryptor and the responder as the decryptor, and the other with the initiator as the decryptor and the responder as the encryptor. We will describe the private message number (stateful) case.

```
**Algorithm** CMCC Encrypt($P$, $\bar{K}, L_3, L_2, \bar{L}_2, L_1$, $N$, $A$)
$M \leftarrow (10110110)^{16-|N|/8}||N$
$Z \leftarrow 0^{\tau}$
$W \leftarrow E_{\bar{K}}(M)$
$Q \leftarrow P||Z$
$L \leftarrow |Q|/8$
**if** $L = 0 \ mod \ 2$ **then**
    $P_1 \leftarrow MSB_{L/2}(Q)$
    $P_2 \leftarrow LSB_{L/2}(Q)$
**else**
    $P_1 \leftarrow MSB_{(L-1)/2}(Q)$
    $P_2 \leftarrow LSB_{(L+1)/2}(Q)$
**end if**
$X \leftarrow CBC(W, pad(P_1)_{P_2}, L_3) \oplus P_2$
$Y \leftarrow X||A$
$V \leftarrow MAC(W||Y, L_2)$
$i \leftarrow \lfloor |P_1|/B \rfloor$
$P_1 = \bar{P}_{1,1}|| \ldots ||\bar{P}_{1,i}||\bar{P}_{1,i+1}$ where $|\bar{P}_{1,1}| = \ldots = |\bar{P}_{1,i}| = B$ and $|\bar{P}_{1,i+1}| = |P_1| \ mod \ B$.
$U \leftarrow V \ and \ (1^{64}||0^1||1^{31}||0^1||1^{31})$
$X_2 \leftarrow V \oplus \bar{P}_{1,1}||E_{\bar{L}_2}(U+1) \oplus \bar{P}_{1,2}|| \ldots ||E_{\bar{L}_2}(U+i) \oplus \bar{P}_{1,i+1}$
$X_1 \leftarrow CBC(W, pad(X_2)_X, L_1) \oplus X$
```

Figure 1: CMCC Encryption: Encryption inputs are plaintext P, key $K = \bar{K}, L_3, L_2, \bar{L}_2, L_1$, public message number N, and associated data A. $CBC(IV, P, Key)$ is CBC encryption with initialization vector IV, plaintext P, and key Key. $MAC(P, Key)$ is the CMAC MAC algorithm [Dwo05] with plaintext P and key Key. $pad()$ is the padding algorithm defined in Section 3.1. $E_{\bar{K}}$ is the block cipher with key $\bar{K}$. $|P|$, the bitlength of P, is a multiple of 8, as is τ. U is obtained from V by zeroing bits 31 and 63 to enable faster addition (prevent carries) [Har08]. $U + j$ is integer addition, $1 \leq j \leq i$. When xor'ing two strings of different length, the longer string is first truncated to the length of the shorter string.

```
**Algorithm** CMCC Decrypt($X_1, X_2, \bar{K}, L_3, L_2, \bar{L}_2, L_1, N, A$)
$M \leftarrow (10110110)^{16-|N|/8}||N$
$Z \leftarrow 0^{\tau}$
$W \leftarrow E_{\bar{K}}(M)$
$X \leftarrow CBC(W, pad(X_2)_{X_1}, L_1) \oplus X_1$
$Y \leftarrow X||A$
$V \leftarrow MAC(W||Y, L_2)$
$i \leftarrow \lfloor |X_2|/B \rfloor$
$X_2 = \bar{X}_{2,1}|| \ldots ||\bar{X}_{2,i}||\bar{X}_{2,i+1}$ where $|\bar{X}_{2,1}| = \ldots = |\bar{X}_{2,i}| = B$ and $|\bar{X}_{2,i+1}| = |X_2| \; mod \; B$.
$U \leftarrow V \; and \; (1^{64}||0^1||1^{31}||0^1||1^{31})$
$P_1 \leftarrow V \oplus \bar{X}_{2,1}||E_{\bar{L}_2}(U+1) \oplus \bar{X}_{2,2}|| \ldots ||E_{\bar{L}_2}(U+i) \oplus \bar{X}_{2,i+1}$
$P_2 \leftarrow CBC(W, pad(P_1)_X, L_3) \oplus X$
$Q = P_1||P_2,$
$U = LSB_{\tau/8}(Q)$
**if** $(U \; != Z)$ **return** $\perp$
**else**
$Q = \tilde{P}||Z$ and return **Plaintext** $\tilde{P}$
**end if**
```

Figure 2: CMCC Decryption: Decryption inputs are ciphertext X_1X_2, key $K = \bar{K}, L_3, L_2, \bar{L}_2, L_1$, public message number N, and associated data A.

3.3.1 Key Generation

Keys $\bar{K}_1$ and $\bar{K}_2$ are randomly generated for the pseudorandom permutations $E_{\bar{K}_i}$ $i = 1, 2$.

3.3.2 Initial State

$u_{init} = u_{resp} = 0$. $init_e = init_d = resp_e = resp_d = 0$. ($init_e$ and $init_d$ are part of the initiator state; $resp_e$ and $resp_d$ are part of the responder state.) IL is the number of bytes that are transmitted to the peer for recovering the message number. w_s is initialized to a positive integer. $m_1 = 2(w\_s)+1$. Initially the sequences of M values, $Seq(init)$ and $Seq(resp)$ are empty.

3.3.3 Creating the Sequences of Private Message Numbers

Let x be the encryptor, $x \in \{init, resp\}$. Let $v = 1$ if $x = init$, and let $v = 2$ if $x = resp$. Let $Seq(x) = M_0, \ldots, M_{x_e - 1}$.

```
start: $candidate(M) = E_{\bar{K}_v}(u_x)$
IF $LSB_{IL}(candidate(M)) = LSB_{IL}(M_i)$ for any $i$, $0 \leq i \leq x_e - 1$, where $(x_e - i) \leq m_1$,
$u_x = u_x + 1$, go to start;
ELSE
{
$M_{x_e} = candidate(M)$; $Seq(x) = M_0, \ldots, M_{x_e}$
$u_x = u_x + 1$;
}
ENDIF
```

$SeqNo_x[M] = i$ if M is the ith element in the sequence $Seq(x)$.

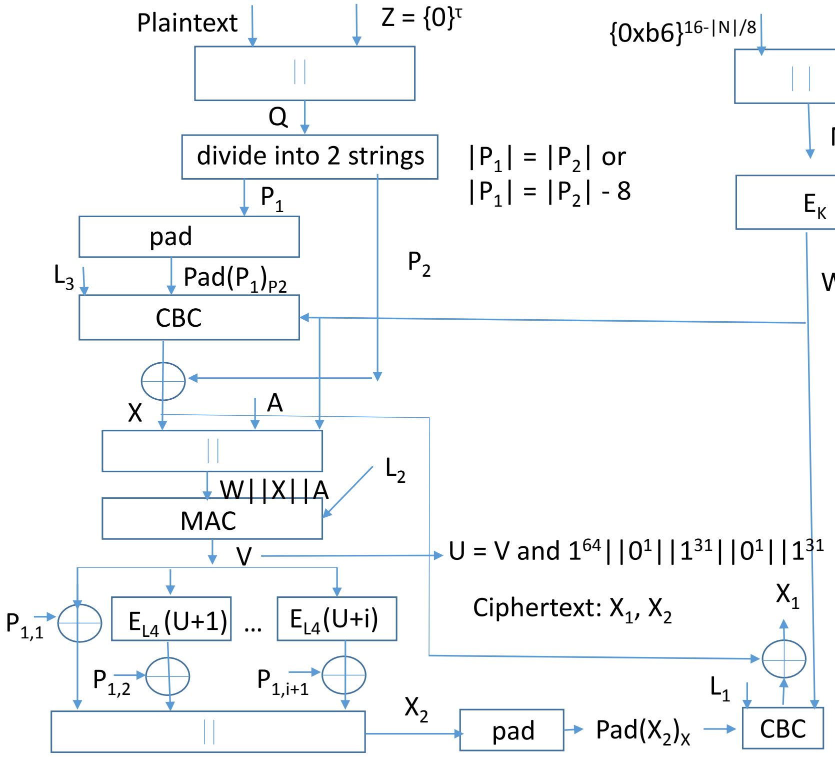

Figure 3: CMCC Stateless Encryption: $L_4 = \bar{L}_2$

3.3.4 Channel Assumption

The decryption algorithm returns $\perp$ if the ciphertext was created using a message number M that was too far out of synchronization. The following assumption guarantees that decryption is successful (i.e., does not output $\perp$).

Let $y \in \{init, resp\}$ where $y \neq x$. The next ciphertext that is decrypted, $X_1|| \dots ||X_k||T$ is such that there exists $\bar{M}$ in $Seq(x)$ such that $LSB_{IL}(\bar{M}) = T$ and $|SeqNo_x[\bar{M}] - y_d| \leq w\_s$.

Given the channel assumption, there exists $\bar{M}$ such that $LSB_{IL}(\bar{M}) = T$, and the algorithm for creating the sequence ensures that $\bar{M}$ is unique.

Table 1 summarizes the parameters for the stateful scheme.

4 Proof of Security

We first give some examples illustrating attacks against CMCC. We will then prove a MRAE security bound for CMCC (see Theorem 4.3). A key point is that ciphertext queries that do not return invalid can be used to create new plaintexts that satisfy a relation (see examples below) that is less likely to be satisfied given a random injection. Of course the MRAE security bound is also an AE security bound for CMCC, but we prove a smaller AE security bound in Theorem 4.6. In

```
**Algorithm** CMCC Stateful Encrypt($P$, $\bar{K_1}, \bar{K_2}, L_3, L_2, \bar{L}_2, L_1$, $i$, $A$)
Select $M$ such that $SeqNo_x[M] = i$.
$Z \leftarrow 0^\tau$
$Q \leftarrow P||Z$
$L \leftarrow |Q|/8$
**if** $L = 0 \ mod \ 2$ **then**
    $P_1 \leftarrow MSB_{L/2}(Q)$
    $P_2 \leftarrow LSB_{L/2}(Q)$
**else**
    $P_1 \leftarrow MSB_{(L-1)/2}(Q)$
    $P_2 \leftarrow LSB_{(L+1)/2}(Q)$
**end if**
$X \leftarrow CBC(M, pad(P_1)_{P_2}, L_3) \oplus P_2$
$Y \leftarrow X||A$
$V \leftarrow MAC(M||Y, L_2)$
$j \leftarrow \lfloor |P_1|/B \rfloor$
$P_1 = \bar{P}_{1,1}|| \ldots ||\bar{P}_{1,j}||\bar{P}_{1,j+1}$ where $|\bar{P}_{1,1}| = \ldots = |\bar{P}_{1,j}| = B$ and $|\bar{P}_{1,j+1}| = |P_1| \ mod \ B$.
$U \leftarrow V \ and \ (1^{64}||0^1||1^{31}||0^1||1^{31})$
$X_2 \leftarrow V \oplus \bar{P}_{1,1}||E_{\bar{L}_2}(U+1) \oplus \bar{P}_{1,2}|| \ldots ||E_{\bar{L}_2}(U+j) \oplus \bar{P}_{1,j+1}$
$X_1 \leftarrow CBC(M, pad(X_2)_X, L_1) \oplus X$
$T = LSB_{IL}(M)$.
```

Figure 4: CMCC Stateful Encryption: Encryption inputs are plaintext P, key $K = \bar{K_1}, \bar{K_2}, L_3, L_2, \bar{L}_2, L_1$, private message number i, and associated data A. State initialization is per the Key Generation, Initial State, and Creating the Sequence of Private Message Numbers subsections above. $CBC(IV, P, Key)$ is CBC encryption with initialization vector IV, plaintext P, and key Key. $MAC(P, Key)$ is the CMAC MAC algorithm [Dwo05] with plaintext P and key Key. $pad()$ is the padding algorithm defined in Section 3.1. $E_{\bar{K}}$ is the block cipher with key $\bar{K}$. $|P|$ is a multiple of 8, as is τ. U is obtained from V by zeroing bits 31 and 63 to enable faster addition (prevent carries) [Har08]. $U + l$ is integer addition, $1 \leq l \leq j$. If xor'ing two strings of different lengths, the longer string is first truncated to the length of the shorter string.

```
**Algorithm** CMCC Stateful Decrypt($X_1, X_2, \bar{K_1}, \bar{K_2}, L_3, L_2, \bar{L}_2, L_1, T, A$)
$x \in \{init, resp\}$ and $x$ has created the ciphertext.
Let $y \in \{init, resp\}$ where $y \neq x$.
There exists at most one $\bar{M}$ in $Seq(x)$ such that $LSB_{IL}(\bar{M}) = T$ and $|SeqNo_x[\bar{M}] - y_d| \leq w\_s$.
**if** $\bar{M}$ exists, **then**
    $M = \bar{M}$
**else**
    return $\perp$
**end if**
$i \leftarrow SeqNo_x[M]$
**if** $SeqNo_x[M] > y_d$, **then**
    $y_d = SeqNo_x[M]$.
**end if**
$Z \leftarrow 0^{\tau}$
$X \leftarrow CBC(M, pad(X_2)_{X_1}, L_1) \oplus X_1$
$Y \leftarrow X||A$
$V \leftarrow MAC(M||Y, L_2)$
$j \leftarrow \lfloor |X_2|/B \rfloor$
$X_2 = \bar{X}_{2,1}|| \ldots ||\bar{X}_{2,j}||\bar{X}_{2,j+1}$ where $|\bar{X}_{2,1}| = \ldots = |\bar{X}_{2,j}| = B$ and $|\bar{X}_{2,j+1}| = |X_2|\ mod\ B$.
$U \leftarrow V\ and\ (1^{64}||0^1||1^{31}||0^1||1^{31})$
$P_1 \leftarrow V \oplus \bar{X}_{2,1}||E_{\bar{L}_2}(U+1) \oplus \bar{X}_{2,2}|| \ldots ||E_{\bar{L}_2}(U+j) \oplus \bar{X}_{2,j+1}$
$P_2 \leftarrow CBC(M, pad(P_1)_X, L_3) \oplus X$
$Q = P_1||P_2$,
$U = LSB_{\tau/8}(Q)$
**if** $(U\ != Z)$ **return** $\perp$
**else**
    $Q = \tilde{P}||Z$ and return **Plaintext** $\tilde{P}$, $i$
**end if**
```

Figure 5: CMCC Stateful Decryption: Decryption inputs are ciphertext X_1, X_2, key $K = \bar{K_1}, \bar{K_2}, L_3, L_2, \bar{L}_2, L_1$, message number tag T, and associated data A. State initialization is per the Key Generation, Initial State, and Creating the Sequence of Private Message Numbers subsections above.

Parameter	*Description*
M	per message value obtained by using PRP on private message number
$E_{\bar{K}}()$	PRP used to create M values
l	number of bits in the strings mapped by $E_{\bar{K}}()$; assume $l = 128$
q	bound on number of adversary queries
IL	number of bytes of ciphertext expansion
w_s	bound on ciphertext reordering that still ensures decrypt success

Table 1: Summary of Parameters for Stateful CMCC Scheme

```
Algorithm SIV-G Encrypt $\mathrm{E}_{L_2,\bar{L}_2}(H, N, A, P)$
X $\leftarrow f(P,T)$
$IV \leftarrow CMAC_{L_2}(N||X||A)$
$C \leftarrow CTR_{\bar{L}_2}(IV, P)$
return $Y = IV||C$
```

Figure 6: SIV-G Encryption: Encryption inputs are header $H = T$, nonce N, associated data A, and plaintext P.

Section 4.1, we present a CMCC variant, CMCC with MAC, or CWM. We prove a MRAE security bound for CWM in Theorem 4.7 and a AE security bound for CWM in Theorem 4.8.

To give more insight into the best attacks and security properties of CMCC, we utilize the following examples.

Example 1: Without the encoding step (for the zero bit authentication tag), CMCC is not MRAE secure (the adversary advantage is large in the MRAE security game). To illustrate this fact, the adversary submits a plaintext query followed by a ciphertext query using the same message number M and value X_2. Both queries are twice the block length of the underlying block cipher. The adversary can compute $X_1 \oplus \bar{X}_1 = X \oplus \bar{X}$. The adversary then creates two new plaintexts by modifying both P_2 and $\bar{P}_2$ so that the two corresponding ciphertexts have equal X values. Note that the two plaintexts have distinct P_1 values (P_{11} and P_{12}). The adversary submits both plaintexts along with the message number M and receives the two ciphertexts whose X_2 values xor to $P_{11} \oplus P_{12}$. This relation is only satisfied with probability $1/\alpha$ for a random injection and thus the adversary advantage is large.

Example 2: Given a collision of X values for two plaintext queries in the MRAE security game (message numbers may be reused). Then the adversary can modify the respective P_2 values to create two new plaintexts such that the corresponding ciphertexts have equal X values. Then the adversary can win with high probability as in the preceding example. This attack works even if the zero bit authentication tag is being used. Thus $q(q-1)/2\alpha$ will be part of the security bound for CMCC MRAE security.

Remark: For the stateless scheme, if there is a field in the associated data which is distinct for each message (e.g., sequence number field), then this can be utilized for the message number and the advantage is that no additional bytes for the message number are sent over the network.

```
**Algorithm** SIV-G Decrypt $\mathrm{D}_{L_2,\bar{L}_2}(H, N, A, Y)$
if $|Y| < B$, then return $\perp$ .
IV $\leftarrow Y[1 \ldots B], C \leftarrow [B+1 \ldots |Y|]$
$P \leftarrow CTR_{\bar{L}_2}(IV, C)$
$X \leftarrow f(P, T)$
$IV_2 \leftarrow CMAC_{L_2}(N||X||A)$
if $IV = IV_2$, return $P$, else return $\perp$ .
```

Figure 7: SIV-G Decryption: Decryption inputs are header $H = T$, nonce N, associated data A, and Y

Lemma 4.1 *([RS06] - Theorems 2 and 7) SIV has MRAE security bound*

$$Adv_{SIV}^{MRAE(q,t,\mu)} \leq Adv_{CMAC}^{prf}(q,t) + Adv_{CTR}^{priv}(q,t,\mu) + 5q/2^B + q^2/2^{B+9}.$$

Lemma 4.2 *Consider the following generalization of the SIV [RS06] algorithm, SIV-G: We include a distinguished string T as part of the header H. We replace the plaintext P in the PRF calculation with $f(P,T)$ where f is an injective function (thus $f(P,T) = f(\bar{P},\bar{T})$ implies $P = \bar{P}$ and $T = \bar{T}$.) See Figure 6 and 7. The security bound for SIV-G is unchanged from SIV: SIV-G has MRAE security bound*

$$Adv_{SIV-G}^{MRAE(q,t,\mu)} \leq Adv_{CMAC}^{prf}(q,t) + Adv_{CTR}^{priv}(q,t,\mu) + 5q/2^B + q^2/2^{B+9}.$$

Theorem 4.3 *Let b_i =number of bytes in ith query response, $1 \leq i \leq q$. Let $\mu = \sum_{i=1}^{q} \lceil b_i/32 \rceil$. B is the cipher block length. Let $\beta = min\{\alpha, 2^B\}$. Let the CMCC MAC function be CMAC [Dwo05]. Let s be the maximum number of CMAC blocks in a query; c_1 is a constant. CMCC encryption (stateless version) is a misuse resistant authenticated encryption scheme with MRAE-advantage bounded by*

$$\begin{gathered} q(q-1)/2\alpha + q(q-1)/2\beta + 1 - (1 - 1/\beta - 2^{-\tau})^x + (5s^2+1)q^2/2^B + \\ Adv_E^{prp}(sq+1, t+c_1 sq) + Adv_E^{prp}(sq,t) + sq(sq-1)/2^{B+1} + \mu(\mu-1)/2^{B+1} + \\ Adv_{CTR}^{priv}(q,t,\mu) + 5q/2^B + q^2/2^{B+9} + 2q(q-1)/2^{B+1} + Adv_E^{prp}(q,t) \end{gathered}$$

given that the adversary is restricted to q queries, E is the underlying block cipher for CMAC (e.g., AES), $\alpha = 2^{8m}$ where Len is the byte length of the minimal length plaintext query response, $m = \lfloor Len/2 \rfloor$, assuming up to x invalid ciphertexts do not result in session termination, and τ is the number of bits in the authentication tag.

Remark: Intuitively, there are three types of relations that distinguish CMCC from a random injection:

1. For messages where $|\alpha|$ is shorter than the block length, and $M = \bar{M}$, we have the relation $X_2 \oplus \bar{X}_2 = P_1 \oplus \bar{P}_1$ with higher probability equal to $1/\alpha + (\alpha - 1)/\alpha^2$ for CMCC versus $1/\alpha$ for the random injection. The reason is that we may have a collision of X values with probability $1/\alpha$ and if that does not occur, the resulting V values may still be equal in the first $\log_2(\alpha)$ bits.

2. If $M = \bar{M}$, $X_2 = \bar{X}_2$, and $P_1 = \bar{P}_1$, then $X_1 \oplus \bar{X}_1 = P_2 \oplus \bar{P}_2$. The latter occurs with probability $1/\beta$ for CMCC but it occurs with probability $1/\beta^2$ for a random injection.

3. For messages such that $|X_1|$ = block length, $M = \bar{M}$, $P_2 = \bar{P}_2$, and $P_1 \neq \bar{P}_1$, we have the relation $X_2 \oplus \bar{X}_2 = P_1 \oplus \bar{P}_1$ with probability $1/2^B$ given a random injection, but with probability 0 for CMCC.

Proof: **case i:** All plaintexts have length $\leq 2 * B + 1 - \tau$ bits: We use a games based proof to establish the bound claim for the theorem. Game G_0 is depicted in Figure 8. Game G_0 gives the adversary the CMCC encryption and decryption oracles and the adversary's probability of success is equal to the adversary's MRAE-advantage against CMCC.

Game G_1 is the same as game G_0 except we replace the CMAC MAC function with a random function. Now consider an adversary $\mathcal{A}^{\mathcal{E},\mathcal{D}}$ where $\mathcal{E}$ and $\mathcal{D}$ are either the game G_0 encrypt and decrypt oracles or the game G_1 encrypt and decrypt oracles. When $\mathcal{A}$ submits P, A, N, then X_1, X_2 is returned and we give the distinguisher D $X_2 \oplus P_1 = F(P, A, N)$ where F is either CMAC or a random function. When $\mathcal{A}$ submits X_1, X_2, A, N then P is returned and we give the distinguisher D $X_2 \oplus P_1 = F(P, A, N)$. When $\mathcal{A}$ outputs b, D also outputs b ($b \in \{0, 1\}$). Then $\mathcal{A}'s$ probability of success is bounded by the probability bound for any adversary to distinguish CMAC from a random function which is $(5s^2 + 1)q^2/2^B + Adv_E^{prp}(sq + 1, t + c_1 sq)$ [IK03] where E is the underlying block cipher, e.g., AES, and s is the maximum number of blocks in any query.

Thus

$$|Pr[\mathcal{A}^{G_1} \Rightarrow 1] - Pr[\mathcal{A}^{G_0} \Rightarrow 1]| \leq (5s^2 + 1)q^2/2^B + Adv_E^{prp}(sq + 1, t + c_1 sq)$$

Game G_2 is the same as game G_1 except the block ciphers used in CBC encryption for computing X_1 and X are replaced with random functions. Consider the game F (see Figure 9) where prf game adversary $\mathcal{B}$ has oracle access to functions f_1 and f_2 and distinguishes between the following:

1. $f_1 = E_{L_3}$, $f_2 = E_{L_1}$, and

2. $f_1 = g_1 \in H_{128,128}$, $f_2 = g_2 \in H_{128,128}$ (g_1 and g_2 are random functions.)

$f_1 = E_{L_3}$ if and only if $f_2 = E_{L_1}$. $\mathcal{B}$ will run $\mathcal{A}^{G_i}$ as a subroutine, $i = 1, 2$. If $f_1 = E_{L_3}$, then $\mathcal{A}$ is in game G_1, and if $f_1 = g_1$ then $\mathcal{A}$ is in game G_2.

Each encryption query from $\mathcal{A}$ results in $\mathcal{B}'s$ query of $W \oplus pad(P_1)_{P_2}$ to f_1. $\mathcal{A}$ will output a bit indicating whether it is in game G_1 or game G_2. $\mathcal{B}$ outputs the same bit for the prf game. Thus $\mathcal{A}'s$ probability of success is bounded by $\mathcal{B}'s$ probability of success. Let q be the number of queries to f_1. Then $Adv(\mathcal{A}, q, t) \leq Adv_E^{prf}(q, t)$ where E is the block cipher.

Thus we obtain

$$|Pr[\mathcal{A}^{G_2} \Rightarrow 1] - Pr[\mathcal{A}^{G_1} \Rightarrow 1]| \leq Adv_E^{prf}(q, t) \leq Adv_E^{prp}(q, t) + q(q-1)/2^{B+1}$$

Game G_3 is the same as game G_2 except:

1. Initialize is modified: Initially we set $QD(N, A) = \emptyset$ for all N, A. $QD(N, A)$ is a subset of the plaintexts.

2. The line: if $(U! = Z)$ return $\perp$; otherwise $Q = \tilde{P}||Z$ and return Plaintext $\tilde{P}, A, N$ is replaced with:
   $\bar{Q}$ is a random string of length $|Q|$ such that the prefix of $\bar{Q}$ of length $|Q| - \tau$ is in $QD(N, A)^C$, $\bar{U} = LSB_{\tau/8}(\bar{Q})$. If $(\bar{U}! = Z)$ return $\perp$, else $\bar{Q} = \tilde{P}||Z$, return $\tilde{P}, A, N$.

3. If the adversary submits the encryption query P, A, N, then we set $QD(N, A) = QD(N, A) \cup \{P\}$.

Then the advantage of $\mathcal{A}$ in distinguishing G_3 and G_2 is bounded by the probability of obtaining a valid response from the decryption oracle. Consider the adversary's optimal strategy for obtaining a valid ciphertext response in game G_2; given the ciphertext query $\bar{X}_1, \bar{X}_2, \bar{N}$. Clearly if no encryption queries have been submitted (so no query responses have been received) then the probability of a valid response is $2^{-\tau}$. Suppose we have submitted one previous encryption query: P_1, P_2, N, A returning X_1, X_2.

case a: $\bar{N} = N$ and $\bar{X}_2 \neq X_2$.

Then the probability of a valid response is independent of this previous query since we evaluate the random function at a new domain point. Thus $\bar{X}$ is uniform random, and the value P_2 will be uniform random, so the probability of a valid response is $2^{-\tau}$.

case b: $\bar{N} \neq N$ and $\bar{X}_2 = X_2$.

The argument as in case a applies; the probability of a valid response is $2^{-\tau}$.

case c: $\bar{N} \neq N$ and $\bar{X}_2 \neq X_2$.

The adversary may select $\bar{X}_1 = X_1$. Then $\bar{X} = X$ with probability 2^{-B}. The input to the random function for computing P_2 will also be the same with probability 2^{-B}; otherwise, the probability of a valid response will be $2^{-\tau}$. Thus the probability of a valid response is $2^{-\tau} + 2^{-B}(2^{-B} + 2^{-\tau})$.

case d: $\bar{N} = N$ and $\bar{X}_2 = X_2$.

We have $Pr[\bar{P}_1 = P_1] = 1/\beta$ and in that case if the last τ bits of $\bar{X}_1$ equal the last τ bits of X_1 then the query is valid. We have $\bar{P}_1 \neq P_1$ with probability $(\beta - 1)/\beta$. In this case, P_2 is uniform random so the probability that the query is valid is $2^{-\tau}$. Thus the probability of a valid query is $1/\beta + ((\beta - 1)/\beta)2^{-\tau}$.

Case d maximizes the probability of a valid response. There are two strategies for additional queries: multiple encryption queries followed by decryption queries or a single encryption query followed by decryption queries. Multiple encryption queries are likely to result in distinct X_2 values; in any case, two responses with equal N and X_2 values allows the Adversary to distinguish CMCC from a PRI with high probabiity without any decryption queries (see Games G_4 and G_5.) Thus the optimal strategy for multiple queries using the case d strategy is a single encryption query followed by decryption queries.

For cases a and b, multiple encryption queries followed by ciphertext queries does not increase the probability of a valid decryption query beyond $2^{-\tau}$. Thus these strategies are suboptimal in the multiple queries case as well.

For case c, multiple encryption queries followed by multiple decryption queries does increase the probability of a valid decryption query. The success probability is dominated by $q^2(2^{-B-\tau})$ which is less than the optimal case d strategy.

Thus the optimal adversary strategy is a single plaintext query followed by successive ciphertext queries that match the N and X_2 values from the plaintext query.

The bound for Adversary success, assuming at most $x, 1 \leq x \leq q$, invalid ciphertext queries prior to session termination, is

$$|Pr[\mathcal{A}^{G_3} \Rightarrow 1] - Pr[\mathcal{A}^{G_2} \Rightarrow 1]| \leq 1 - (1 - 1/\beta - 2^{-\tau})^x.$$

Game G_4 is the same as game G_3 except the line
$X = CBC(W, pad(P_1)_{P_2}, L_3) \oplus P_2$,
is replaced with
$X = CBC(W, pad(P_1)_{P_2}, L_3) \oplus P_2$; if $X \in set\_of\_used\_X$, $bad_5 = true$ and reselect $X : X \leftarrow set\_of\_used\_X^C$. If $X \notin set\_of\_used\_X$, $set\_of\_used\_X = set\_of\_used\_X \cup \{X\}$. Then

$$|Pr[\mathcal{A}^{G_4} \Rightarrow 1] - Pr[\mathcal{A}^{G_3} \Rightarrow 1]| \leq q(q-1)/2\alpha + q(q-1)/2^{B+1}.$$

Game G_5 is depicted in Figure 10. Then game G_5 and game G_4 are indistinguishable except that collisions are possible in the strings S_2 where C includes $S_1||S_2$. When such a collision occurs, the games are distinguishable; the bound on collisions is $q(q-1)/2\beta$. It is possible in game G_4 that a ciphertext query that is not invalid will return a plaintext and another encrypt query with a different plaintext returns the same ciphertext. This last sequence is not possible in game G_5. However, the bound from Game G_3 allows us to assume that no valid ciphertext queries occur. Thus

$$|Pr[\mathcal{A}^{G_5} \Rightarrow 1] - Pr[\mathcal{A}^{G_4} \Rightarrow 1]| \leq q(q-1)/2\beta + q(q-1)/2^{B+1}.$$

Thus the bound claimed in the theorem statement holds.

case ii: Some plaintexts have length greater than $2 * B + 1 - \tau$ bits:
We note that this case is a suboptimal strategy for the adversary. Game G_1 is unchanged and for game G_2 the term $q(q-1)/2^{B+1}$ from above is generalized to $sq(sq-1)/2^{B+1}$. The game G_3 bound holds. For $CBC(W, pad(X_2)_X)$ in game G_3, if the every input to each random function invocation is a previously unseen input (fresh input), then the output is random (the function is a random function). This bound on failure here is $\mu(\mu-1)/2^{B+1} + q(q-1)/2^{B+1}$.

Lemma 4.2 applies if all of the X values from the queries are distinct. For the function f in the Lemma, we use $P = 2nd\ to\ last\ blocks\ of\ P_1$, $T = P_2||1st\ block\ of\ P_1$, and $f(P,T) = X = CBC(W, pad(P_1)_{P_2}, L_3) \oplus P_2$. The probability that the X values from the queries is not distinct is bounded by $q(q-1)/2\alpha + q(q-1)/2^{B+1}$ (where $\alpha \geq 2^B$.) The X_1 values and first block of X_2 are random strings when these failure events do not occur and thus the CMCC adversary's advantage is the same as the SIV-G advantage. Thus the CMCC adversary's advantage in distinguishing between games G_3 and G_5 is bounded by the sum of the two terms above plus the SIV-G security bound. ▮

Initialize: Select the CMCC key, using the uniform random distribution. Let Z be the bit string with τ zero bits. $bad_4 = bad_5 = false$. Let $set\_of\_used\_X = \emptyset$.
Encrypt(P, A, N): See Figure 1 for definition.
Decrypt(C, A, N): See Figure 2 for definition.
Output: Return the adversary's output.

Figure 8: CMCC MRAE proof Game G_0

Remark: (i) We can replace the $2^{-\tau}$ term in the above theorem with $2^{-(\tau+\gamma)}$ where γ quantifies the number of higher level protocol check bits. (ii) We can eliminate the $2^{-\tau}$ term if $|P_2| \leq \tau$.

Initialize: $\mathcal{B}$ selects keys $\bar{K}$, L_2, $\bar{L}_2$, using the uniform distribution. $\mathcal{B}$ has oracle access to f_1 and f_2.
Response to $\mathcal{A}'s$ encrypt query: $\mathcal{B}$ computes and returns X_1, X_2 to $\mathcal{A}$.
Response to $\mathcal{A}'s$ decrypt query: $\mathcal{B}$ computes and returns P_1, P_2 to $\mathcal{A}$.
Output: Return $\mathcal{A}$'s output.

Figure 9: Game F with PRF Adversary $\mathcal{B}$.

Initialize: Select a random injection $f \in Inj_e^{\mathcal{N},\mathcal{A}}(\mathcal{P},\mathcal{C})$. Let Z be the bit string with τ zero bits. $e(N, A, P) = \tau$ for all N, A, and P.
Encrypt(P, A, N): Return $f(N, A, P)$.
Decrypt(C, A, N): $f^{-1}(N, A, C) = P$ if $f(N, A, P) = C$ and return $\perp$ if no such triple (N, A, P) exists.
Output: Return the adversary's output.

Figure 10: CMCC MRAE proof Game G_5

We now prove a security bound for the CMCC stateless AEAD algorithm; here message numbers are not allowed to be repeated in plaintext queries. In the following, games H_1, H_2, H_3, and H_4 are identical to games G_1, G_2, G_3, and G_5 respectively, except the H_i games are in the AE security game where queries may not reuse message numbers from previous queries.

Lemma 4.4 *Let $q - 1 \leq 2^\tau$. Given the adversary strategy in game H_2 (in the AE game) where the adversary submits a plaintext query P_1, P_2, N and obtains the response X_1, X_2. The adversary then submits a succession of ciphertext queries of the form $\bar{X}_1, X_2, N$ where the last τ bits of $\bar{X}_1$ are equal to the last τ bits of X_1. Given the relation*

$$\hat{X}_1 \oplus \bar{X}_1 = \hat{P}_2 \oplus \bar{P}_2 \tag{1}$$

Then

$$Pr[there\ are\ 2\ distinct\ queries\ \hat{P}_1, \hat{P}_2, N, \hat{X}_1, X_2\ and\ \bar{P}_1, \bar{P}_2, N, \bar{X}_1, X_2\ satisfying\ (1)\ is] \leq$$
$$(q-1)\sum_{i=0}^{q-2}\binom{q-2}{i}\lambda_1/2^{i\tau} < \lambda_1 e(q-1) < 2e(q-1)/\beta$$

where $\lambda_1 = 1/\beta + (\beta - 1)/\beta^2$.

Proof: We use induction over the number of queries. If $q = 2$, we have

$$Pr[(1)\ holds] = \lambda_1 = (q-1)\sum_{i=0}^{q-2}\binom{q-2}{i}\lambda_1/2^{i\tau} < \lambda_1 e.$$

Suppose the lemma is valid for $k = q - 1$. We now prove the $k = q$ case. We have

$$\begin{aligned}
Pr[(1)\ in\ H_2\ with\ q\ queries] = Pr[(1)\ in\ H_2\ with\ first\ q-1\ queries] + \\
Pr[not\ (1)\ in\ H_2\ with\ first\ q-1\ queries\ \cap (1)\ in\ H_2\ with\ qth\ query] \leq \\
Pr[(1)\ in\ H_2\ with\ first\ q-1\ queries] + Pr[(1)\ in\ H_2\ with\ qth\ query] \leq \\
(q-2)\sum_{i=0}^{q-3}\binom{q-3}{i}\lambda_1/2^{i\tau} + \lambda_1 + (1-\lambda_1)\left(\sum_{i=0}^{q-2}\binom{q-2}{i}2^{-i\tau}(1-2^{-\tau})^{q-2-i}i\lambda_1\right) < \\
(q-2)\sum_{i=0}^{q-3}\binom{q-3}{i}\lambda_1/2^{i\tau} + \lambda_1 + \sum_{i=0}^{q-2}\binom{q-2}{i}i\lambda_1/2^{i\tau} = \\
\sum_{i=0}^{q-3}\left(\binom{q-3}{i}(q-2)\lambda_1/2^{i\tau} + \binom{q-2}{i}i\lambda_1/2^{i\tau}\right) + \lambda_1 + (q-2)\lambda_1/2^{(q-2)\tau} = \\
\sum_{i=0}^{q-3}\binom{q-2}{i}(q-2)\lambda_1/2^{i\tau} + (q-2)\lambda_1/2^{(q-2)\tau} + \lambda_1 = \\
\lambda_1 + (q-2)\sum_{i=0}^{q-2}\binom{q-2}{i}\lambda_1/2^{i\tau} < \\
(q-1)\sum_{i=0}^{q-2}\binom{q-2}{i}\lambda_1/2^{i\tau}.
\end{aligned}$$

Also,

$$\sum_{i=0}^{q-2}\binom{q-2}{i}1/2^{i\tau} < \sum_{i=0}^{q-2}1/i! < e$$

which completes the proof. ▮

Lemma 4.5 *Let $q - 1 \leq 2^\tau$. Given the adversary strategy in game H_3 above where the adversary submits a plaintext query P_1, P_2, N and obtains the response X_1, X_2. The adversary then submits a succession of ciphertext queries of the form $\bar{X}_1, X_2, N$ where the last τ bits of $\bar{X}_1$ are equal to the last τ bits of X_1. Then*

$$Pr[there\ are\ 2\ distinct\ queries\ \hat{P}_1, \hat{P}_2, N, \hat{X}_1, X_2\ and\ \bar{P}_1, \bar{P}_2, N, \bar{X}_1, X_2\ satisfying\ (1)\ is] \geq (q-1)2^{-\tau}/\beta$$

Proof: The probability that (1) is satisfied is bounded below by

$$1 - (1 - 2^{-\tau}/\beta)^{q-1} = 1 - \sum_{i=0}^{q-1}\binom{q-1}{i}(-2^{-\tau}/\beta)^i \geq 1 - (1 - (q-1)2^{-\tau}/\beta) = (q-1)2^{-\tau}/\beta$$

▮

Theorem 4.6 *Let b_i =number of bytes in ith query response, $1 \leq i \leq q$. Let $\mu = \sum_{i=1}^{q}\lceil b_i/32 \rceil$. B is the cipher block length. Let $\beta = min\{\alpha, 2^B\}$. Let the CMCC MAC function be CMAC [Dwo05].*

Let s be the maximum number of CMAC blocks in a query; c_1 is a constant. $L = max_{1 \leq i \leq q}\{b_i\}$. CMCC encryption (stateless version) is an authenticated encryption with associated data (AEAD) scheme with AE-advantage bounded by

$$q(q-1)2^{-\tau-1}(1/\beta + 2^{-\tau}) + 2e(q-1)(1/\beta + (L-1)/2^{B+\tau} + 2^{-B}) + (5s^2+1)q^2/2^B + \\ Adv_E^{prp}(sq+1, t+c_1sq) + Adv_E^{prp}(sq,t) + sq(sq-1)/2^{B+1} + \mu(\mu-1)/2^{B+1} + \\ Adv_{CTR}^{priv}(q,t,\mu) + 5q/2^B + q^2/2^{B+9} + 2q(q-1)/2^{B+1} + Adv_E^{prp}(q,t)$$

given that the adversary is restricted to q queries, E is the underlying block cipher for CMAC (e.g., AES), $\alpha = 2^{8m}$ where Len is the byte length of the minimal length plaintext query response, $m = \lfloor Len/2 \rfloor$, and $\tau > 0$ is the number of bits in the authentication tag. We also assume $q-1 \leq 2^{\tau}$.

Proof: **case 1:** All plaintexts have length $\leq 2 * B + 1 - \tau$ bits:

For the transition from game H_2 to game H_3 we have two mechanisms for the adversary to distinguish between the two: $X_2 \oplus \bar{X}_2 = P_1 \oplus \bar{P}_1$, and $X_1 \oplus \bar{X}_1 = P_2 \oplus \bar{P}_2$ (1) for two distinct queries X_2, X_1, N, P_1, P_2 and $\bar{X}_2, \bar{X}_1, \bar{N}, \bar{P}_1, \bar{P}_2$.

We first consider distinguishing between H_2 and H_3 via (1):

case a: Here the adversary uses the strategy from Lemma 4.4: the adversary submits a single plaintext query with message number N and receives a response with X_1 and X_2, followed by ciphertext queries with $\bar{N} = N$, and $\bar{X}_2 = X_2$, where the last τ bits for $\bar{X}_1$ are equal to the last τ bits of X_1 from the plaintext query. Then we have

$$|Pr[\mathcal{A}^{H_2} \Rightarrow 1] - Pr[\mathcal{A}^{H_3} \Rightarrow 1]| \leq \\ 2e(q-1)/\beta - (q-1)2^{-\tau}/\beta \leq 2e(q-1)/\beta$$

where we have applied both Lemma 4.4 and Lemma 4.5 from above.

case b: Games H_2 and H_3 can also be distinguished if a collision occurs on $W \oplus pad(P_1)_{P_2}$ and $W \oplus pad(X_2)_X$ between 2 distinct plaintext queries in game H_2 which gives a slightly higher probability for the relation $X_1 \oplus \bar{X}_1 = P_2 \oplus \bar{P}_2$ in H_2 versus H_3. This probability is bounded by $q(q+1)2^{-2B-1}$. We can ignore the corresponding case where one or both queries are ciphertext queries since the probability would be less. Furthermore, this strategy is sub-optimal compared to the case a strategy above.

case c: Neither of the above two cases: then at least one of the CBC random function replacements get evaluated on a point distinct from the point in any other query. Thus the probability of (1) is the same in both H_2 and H_3.

We now check the adversary's optimal strategy to distinguish between H_2 and H_3 based on

$$X_2 \oplus \bar{X}_2 = P_1 \oplus \bar{P}_1 \tag{2}$$

case d: Given two previous valid ciphertext queries with identical X_2, N, and last τ bits of X_1 values, the adversary may leverage the technique from the examples above to create a new encryption query that will have the same N value and which will match one of the previous query's X value. Then this query response can be used to distinguish between H_2 and H_3. The adversary advantage is bounded by $q(q-1)2^{-\tau-1}(1/\beta + 2^{-\tau})$.

case e: Given a combination of zero or more plaintext queries and one or more ciphertext queries, with at least two total queries. If we have a match on the last τ bits of X_1 values for some queries as well as a collision on $W \oplus pad(X_2)_X$ then the adversary can follow the approach in case d above and distinguish between H_2 and H_3 based on (2) above. Note that the X_2 and N values are distinct across the queries. The probability of such a collision between two queries is at best 2^{-B} and therefore this strategy is suboptimal.

case f: The new query (either $\bar{X}_1, \bar{X}_2, \bar{N}$ or $\bar{P}_1, \bar{P}_2, \bar{N}$) is such that $\bar{N}$ is distinct from the N in previous queries. Then $X_2 \oplus \bar{X}_2 = P_1 \oplus \bar{P}_1$ occurs with the same probability in both H_3 and H_2 since $\bar{N}$ results in a previously unseen point for the domain of the CMAC random function replacement.

case g: The new ciphertext query is such that $\bar{X}_2$ and $\bar{N}$ match the corresponding values in a set of previous queries: Then the corresponding X values are distinct. So $X_2 \oplus \bar{X}_2 = P_1 \oplus \bar{P}_1$ occurs with the same probability in both H_3 and H_2. (Here we assume that the last τ bits of the X_1 values are distinct, or alternatively, that all of the previous queries are plaintext queries, to distinguish this case from case d above.)

case h: The new ciphertext query is such that $\bar{X}_2$ is distinct from and $\bar{N}$ matches the corresponding values in a set of previous queries:

Note that only one of the previous queries is a plaintext query whereas the others must be valid ciphertext queries. Then we have a similar scenario as for case a above, and we can apply Lemma 4.4 with the collision bound $2^{-\tau+1}/\beta$ in place of $1/\beta + (\beta-1)/\beta^2$. Since the latter value is larger, this strategy is suboptimal.

case i: None of the above cases. Then the inputs to the $CBC(W, pad(X_2)_X)$ random function replacement are distinct across all queries. Thus the probability of $X_1 \oplus \bar{X}_1 = X \oplus \bar{X}$ is $1/\beta$ for any two queries. Also, the above cases are exhaustive for $(X, N) = (\bar{X}, \bar{N})$. Thus the probability of (2) is the same in both H_2 and H_3.

case 2: Some plaintexts have length greater than $2 * B + 1 - \tau$ bits:

The case with longer plaintexts/ciphertexts is similar to the Theorem 4.3 case ii above. The term $2e(q-1)/\beta$ is generalized to $2e(q-1)(1/\beta + (L-1)/2^{B+\tau} + 2^{-B})$. Also, the bound on X collisions is $q(q-1)/2^{B+1}$. ▮

4.1 CMCC with MAC (CWM)

In this section, we present a variant, CMCC with MAC (CWM). Figures 11 and 12 specify CWM. For the proof of CWM AE security, the main distinction with CMCC above is that we no longer restrict $q-1 \leq \tau$. By requiring the MAC computation, CWM achieves a stronger security bound at the cost of additional processing, when compared with CMCC.

We give the MRAE security bound and the AE security bound for CWM in the next two theorems.

Theorem 4.7 *Let b_i =number of bytes in ith query response, $1 \leq i \leq q$. Let $\mu = \sum_{i=1}^{q} \lceil b_i/32 \rceil$. B is the cipher block length. Let $\beta = min\{\alpha, 2^B\}$. Let the CMCC MAC function be CMAC [Dwo05].*

```
**Algorithm** CWM Encrypt($P,\ \bar{K}, L_3, L_2, \bar{L}_2, L_1,\ N,\ A$)
$M \leftarrow (10110110)^{16-|N|/8}||N$
$Z \leftarrow MAC(P, \bar{K})$
$W \leftarrow E_{\bar{K}}(M)$
$Q \leftarrow P||Z$
$L \leftarrow |Q|/8$
**if** $L = 0\ mod\ 2$ **then**
    $P_1 \leftarrow MSB_{L/2}(Q)$
    $P_2 \leftarrow LSB_{L/2}(Q)$
**else**
    $P_1 \leftarrow MSB_{(L-1)/2}(Q)$
    $P_2 \leftarrow LSB_{(L+1)/2}(Q)$
**end if**
$X \leftarrow CBC(W, pad(P_1)_{P_2}, L_3) \oplus P_2$
$Y \leftarrow X||A$
$V \leftarrow MAC(W||Y, L_2)$
$i \leftarrow \lfloor |P_1|/B \rfloor$
$P_1 = \bar{P}_{1,1}|| \ldots ||\bar{P}_{1,i}||\bar{P}_{1,i+1}$ where $|\bar{P}_{1,1}| = \ldots = |\bar{P}_{1,i}| = B$ and $|\bar{P}_{1,i+1}| = |P_1|\ mod\ B$.
$U \leftarrow V\ and\ (1^{64}||0^1||1^{31}||0^1||1^{31})$
$X_2 \leftarrow V \oplus \bar{P}_{1,1}||E_{\bar{L}_2}(U+1) \oplus \bar{P}_{1,2}|| \ldots ||E_{\bar{L}_2}(U+i) \oplus \bar{P}_{1,i+1}$
$X_1 \leftarrow CBC(W, pad(X_2)_X, L_1) \oplus X$
```

Figure 11: CWM (Stateless) Encryption: Encryption inputs are plaintext P, key $K = \bar{K}, L_3, L_2, \bar{L}_2, L_1$, public message number N, and associated data A. $CBC(IV, P, Key)$ is CBC encryption with initialization vector IV, plaintext P, and key Key. $MAC(P, Key)$ is the CMAC MAC algorithm [Dwo05] with plaintext P and key Key. $pad()$ is the padding algorithm defined in Section 3.1. $E_{\bar{K}}$ is the block cipher with key $\bar{K}$. $|P|$ is a multiple of 8, as is τ. U is obtained from V by zeroing bits 31 and 63 to enable faster addition (prevent carries) [Har08]. $U + j$ is integer addition, $1 \leq j \leq i$. If xor'ing two strings of different lengths, the longer string is first truncated to the length of the shorter string.

```
**Algorithm** CWM Decrypt$(X_1, X_2, \bar{K}, L_3, L_2, \bar{L}_2, L_1, N, A)$
$M \leftarrow (10110110)^{16-|N|/8}||N$
$W \leftarrow E_{\bar{K}}(M)$
$X \leftarrow CBC(W, pad(X_2)_{X_1}, L_1) \oplus X_1$
$Y \leftarrow X||A$
$V \leftarrow MAC(W||Y, L_2)$
$i \leftarrow \lfloor |X_2|/B \rfloor$
$X_2 = \bar{X}_{2,1}||\ldots||\bar{X}_{2,i}||\bar{X}_{2,i+1}$ where $|\bar{X}_{2,1}| = \ldots = |\bar{X}_{2,i}| = B$ and $|\bar{X}_{2,i+1}| = |X_2|\ mod\ B$.
$U \leftarrow V\ and\ (1^{64}||0^1||1^{31}||0^1||1^{31})$
$P_1 \leftarrow V \oplus \bar{X}_{2,1}||E_{\bar{L}_2}(U+1) \oplus \bar{X}_{2,2}||\ldots||E_{\bar{L}_2}(U+i) \oplus \bar{X}_{2,i+1}$
$P_2 \leftarrow CBC(W, pad(P_1)_X, L_3) \oplus X$
$Q = P_1||P_2,$
$U = LSB_{\tau/8}(Q)$
$Q = \tilde{P}||U$
**if** $(U\ != MAC(\tilde{P}, \bar{K}))$ **return** $\perp$
**else**
return **Plaintext** $\tilde{P}$
**end if**
```

Figure 12: CWM (Stateless) Decryption: Decryption inputs are ciphertext X_1X_2, key $K = \bar{K}, L_3, L_2, \bar{L}_2, L_1$, public message number N, and associated data A.

Let s be the maximum number of CMAC blocks in a query; c_1 is a constant. $L = max_{1\leq i\leq q}\{b_i\}$. CWM encryption (stateless version) is a misuse resistant authenticated encryption scheme with MRAE-advantage bounded by

$$q(q-1)/2\alpha + q(q-1)/2\beta + (1-1/\beta)((q-1)/(2^{\tau-1}\beta) + (q-1)(q-2)/(2^{2\tau}\beta) +$$
$$(L-1)((q-1)/(2^{B+\tau-1}) + (q-1)(q-2)/2^{B+2\tau}) + (q-1)/2^{2\tau-1} + \binom{q-1}{2}1/2^{3\tau-1}) +$$
$$(5s^2+1)q^2/2^B + Adv_E^{prp}(sq+1, t+c_1sq) + Adv_E^{prp}(sq,t) + sq(sq-1)/2^{B+1} +$$
$$\mu(\mu-1)/2^{B+1} + Adv_{CTR}^{priv}(q,t,\mu) + 5q/2^B + q^2/2^{B+9} + 2q^2/2^{B+1} + Adv_E^{prp}(q,t)$$

given that the adversary is restricted to q queries, E is the underlying block cipher for CMAC (e.g., AES), $\alpha = 2^{8m}$ where Len is the byte length of the minimal length plaintext query response, $m = \lfloor Len/2 \rfloor$, assuming up to x invalid ciphertexts do not result in session termination, and τ is the number of bits in the authentication tag.

Proof sketch: The proof is similar to the proof of Theorem 4.3 above with the main difference being the bound for the strategy in Lemma 4.4. Also, we use the game structure from Theorem 4.6. Consider the strategy from Lemma 4.4 for the case where plaintexts have short length ($\leq 2B+1-\tau$). Then we have

$$Pr[(1)] = Pr[(1)\ with\ 1st\ query] + Pr[(1)\ without\ 1st\ query] =$$
$$(q-1)(1/(2^\tau\beta) + (\beta-1)/(2^\tau\beta^2)) + \binom{q-1}{2}(1/(2^{2\tau}\beta) + (\beta-1)/(\beta^2 2^{2\tau})) <$$
$$(q-1)/(2^{2\tau-1}\beta) + (q-1)(q-2)/(2^{2\tau}\beta)$$

This term generalizes to

$$(q-1)/(2^{\tau-1}\beta)+(q-1)(q-2)/(2^{2\tau}\beta)+(L-1)((q-1)/(2^{B+\tau-1})+(q-1)(q-2)/2^{B+2\tau})$$

for the arbitrary length messages case.

Also, we have that

$$Pr[(2)]=(q-1)/2^{2\tau-1}+\binom{q-1}{2}1/2^{3\tau-1}$$

We also have

$$Pr[\mathcal{A}^{G_2}\Rightarrow 1]\leq Pr[(1)]+Pr[(2)]$$

and

$$Pr[\mathcal{A}^{G_3}\Rightarrow 1]\geq (1/\beta)(Pr[(1)]+Pr[(2)]$$

Thus

$$|Pr[\mathcal{A}^{G_2}\Rightarrow 1]-Pr[\mathcal{A}^{G_3}\Rightarrow 1]|\leq (1-1/\beta)(Pr[(1)]+Pr[(2)]$$

▮

Theorem 4.8 *Let b_i =number of bytes in ith query response, $1\leq i\leq q$. Let $\mu=\sum_{i=1}^{q}\lceil b_i/32\rceil$. B is the cipher block length. Let $\beta=min\{\alpha,2^B\}$. Let the CMCC MAC function be CMAC [Dwo05]. Let s be the maximum number of CMAC blocks in a query; c_1 is a constant. $L=max_{1\leq i\leq q}\{b_i\}$. CWM encryption (stateless version) is an authenticated encryption with associated data (AEAD) scheme with AE-advantage bounded by*

$$\begin{aligned}(1-1/\beta)q(q-1)2^{-3\tau-1}+(q-1)/(2^{\tau-1}\beta)+(q-1)(q-2)/(2^{2\tau}\beta)+\\(L-1)((q-1)/(2^{B+\tau-1})+(q-1)(q-2)/2^{B+2\tau})+(5s^2+1)q^2/2^B+\\Adv_E^{prp}(sq+1,t+c_1sq)+Adv_E^{prp}(sq,t)+sq(sq-1)/2^{B+1}+\\\mu(\mu-1)/2^{B+1}+Adv_{CTR}^{priv}(q,t,\mu)+5q/2^B+q^2/2^{B+9}+2q^2/2^{B+1}+Adv_E^{prp}(q,t)\end{aligned}$$

given that the adversary is restricted to q queries, E is the underlying block cipher for CMAC (e.g., AES), $\alpha=2^{8m}$ where Len is the byte length of the minimal length plaintext query response, $m=\lfloor Len/2\rfloor$, and $\tau>0$ is the number of bits in the authentication tag.

Proof sketch: The proof is similar to the proof of Theorem 4.6 above with the main difference being the bound for the strategy in Lemma 4.4 and the bound for the other potentially optimal strategy from case 1d in the proof of Theorem 4.6. The bound for the case 1d strategy is

$$(1-1/\beta)q(q-1)2^{-3\tau-1}$$

which replaces

$$(1-1/\beta)q(q-1)2^{-2\tau-1}$$

in Theorem 4.6 above.

For the strategy in Lemma 4.4, we have

$$(q-1)/(2^{\tau-1}\beta)+(q-1)(q-2)/(2^{2\tau}\beta)+(L-1)((q-1)/(2^{B+\tau-1})+(q-1)(q-2)/2^{B+2\tau})$$

which replaces the term $2e(q-1)(1/\beta+(L-1)/2^{B+\tau}+2^{-B})$. ▮

Algorithm	*AE security bound*	*MRAE security bound*
CMCC	$q(q-1)2^{-\tau-1}(1/\beta+2^{-\tau})+2e(q-1)/\beta$	$q/2^{\tau}+q(q-1)/2\alpha+q(q-1)/2\beta$
CWM	$q^2/2^{3\tau}+q^2/(2^{2\tau}\beta)+q/(2^{\tau-1}\beta)$	$q/2^{2\tau-1}+q^2/2^{3\tau-1}+q^2/(2^{2\tau}\beta)$ $+q/(2^{\tau-1}\beta)+q(q-1)/2\alpha+q(q-1)/2\beta$

Table 2: Dominant Terms for Security Bounds for CMCC and CWM

4.2 Security Bound Summary

Table 2 summarizes the dominant terms from the security bounds for CMCC and CWM for short messages (less than 32 bytes), for both AE and MRAE security.

5 Performance Analysis for Wireless Sensor Networks

We discuss and compare performance to other schemes (e.g. CCM [WHF03] and others) for short messages, including energy utilization. Energy utilization is important for low power constrained devices and we use the measurements from [WGE$^+$05] to make an estimate for energy consumption on wireless sensor platforms. We compare CCM to CMCC for energy utilization.

In [WGE$^+$05], the authors measure energy utilization for a variety of cryptographic algorithms due to CPU utilization and networking for the Berkeley/Crossbow motes platform, specifically on the Mica2dot sensor platform. Table 3 gives the results from [WGE$^+$05] with respect to AES encryption, message transmission, and message receipt.

Operation	*Energy Utilization*
Energy to transmit one byte	59.2 μJ
Energy to receive one byte	28.6 μJ
Energy per byte of AES encryption including key setup, averaged over messages of 64-1024 bytes	1.6 μJ

Table 3: Energy Utilization for Operations on the Mica2Dots Platform from [WGE$^+$05]

A key point, which is not specific to the Mica2dot platform, is that energy utilization for transmitting or receiving a byte from the wireless network is 10-100 times greater than the energy needed per byte of AES encryption processing, for wireless sensor nodes.

We estimate energy utilization for CCM and CMCC based on the number of AES encryption operations (pseudorandom function evaluations) and sizes of messages. The other CPU operations such as exclusive-or are minor usages and not counting them will not affect our results significantly. Table 4 gives the results.

Let $R = \lceil L/16 \rceil$, where L is the message length in bytes. For CCM, the number of AES block encryptions is equal to $2R+2$. For CMCC, the number of prf invocations (AES block encryptions) is $4W+1 = 3W+\max\{W-1,0\}+2$ where $W = \lceil L/32 \rceil$. The number drops by 1 if we assume precomputation of the message numbers which is likely in the stateful version and possible in the

Message Length	*No. CCM prf calls*	*No. CMCC prf calls*	*CCM energy use*	*CMCC energy use*
8 bytes	4	5	1819.2	838.4
16 bytes	4	5	2292.8	1312
20 bytes	6	5	2580.8	1548.8
24 bytes	6	5	2817.6	1785.6
32 bytes	6	5	3291.2	2259.2
48 bytes	8	9	4289.6	3308.8
64 bytes	10	9	5288	4256
80 bytes	12	13	6286.4	5305.6
128 bytes	18	17	9281.6	8249.6

Table 4: Energy utilization (μJ) for sending network messages with CCM and CMCC protection, Mica2dot platform.

stateless version as well. CCM eliminates R prf invocations with precomputation, so CMCC has an advantage for messages with 32 bytes or less (for number of prf invocations given precomputation), but CCM has an advantage for longer messages.

Table 4 assumes (1) that CCM uses the minimal recommended length MAC tag of 8 bytes which increases the length of the message by 8 bytes while CMCC includes the 2 byte message number tag T as described above along with a 2 byte authentication string for a total of 4 bytes (2) that both CCM and CMCC are applied to the full length message which will cause our measurements to favor CCM slightly,[2] and (3) Messages are less than 2^{16} bytes so CCM sends a 13 byte nonce with each message.

The amount of energy used for CCM is

$$(32R + 16)(1.6\mu J) + (L59.2\mu J) + 16(1.6\mu J) + 21(59.2\mu J) = 1294.4 + 59.2L + 51.2R(\mu J)$$

and the amount of energy for CMCC is

$$4\lceil L/32\rceil 16(1.6\mu J) + (L + 4)(59.2\mu J) + 25.6\mu J = 102.4\lceil L/32\rceil + 59.2L + 262.4\mu J$$

Thus we see that energy utilization is proportional to message length. For faster schemes (e.g., OCB, etc.), the more efficient computations will result in an even closer correlation between message length (including the MAC bytes) and energy utilization. The reason is that the main energy use is in the networking, and reducing the computational load will result in a higher percentage of energy use by networking.

We haven't included length fields in either CCM or CMCC as part of the comparison. Including such fields would give results very close to the ones above.

[2] CMCC can be applied to the application payload or additional payloads as well (e.g., IPsec). For example, the transport layer checksum and port numbers both act as tag fields for CMCC. In other words, a random change to these fields is likely to cause a failure in transport layer processing leading to message rejection. If link layer encryption/integrity protection is employed, then an integrity failure can be detected prior to sending a large application layer message through multiple wireless network hops. In this case, using CMCC can result in significant energy savings regardless of the size of the application layer messages.

Message Length	*No. CMCC prf calls*	*No. SIV prf calls*	*No. CWM prf calls*
1-16 bytes	5	4	6
17-32 bytes	5	6	7
33-48 bytes	9	8	12
49-64 bytes	9	10	13
65-80 bytes	13	12	18
81-96 bytes	13	14	19

Table 5: Number of Block Cipher Calls For CMCC, SIV, and CWM for Varying Message Sizes (CMCC, CWM message sizes include message tag).

5.1 Implementation and Number of Block Cipher Calls

We have completed an initial implementation as part of our submission to the Caesar competition for authenticated encryption. Details can be accessed at http://groups.google.com/group/crypto-competitions.

Table 5 compares the number of block cipher calls for the CMCC, SIV, and CWM algorithms, for varying message sizes. CMCC requires $3\lceil Length/32\rceil + 2 + \lceil (Length/32) - 1\rceil$ block cipher calls, where *Length* is the message length (including tag).

6 Conclusions

We have presented CMCC, a scheme providing provably secure misuse resistant authenticated encryption, and it leverages existing modes such as CBC, Counter, and CMAC. The main focus for this work is minimizing ciphertext expansion, especially for short messages including plaintext lengths less than the underlying block cipher length (e.g., 16 bytes). Depending on the environment, we obtain security with only 2-6 bytes of ciphertext expansion. Since changes to the ciphertext randomize the plaintext, we can leverage the protocol checks in higher layer protocols as additional authentication bits allowing us to reduce the length of the authentication tag. Our CWM variation provides a further strengthening of the security bounds for the short messages scenario at the cost of an additional MAC operation over the plaintext.

We have given a comparison of energy utilization in wireless sensor networks between CMCC and CCM and showed that energy use is proportional to packet length. Thus CMCC can achieve significant energy savings when applied to protocols that send short messages due to its small ciphertext expansion. Our contributions include both stateless and stateful versions which enable minimal sized message numbers using different network related trade-offs.

References

[AB01] Jee Hea An and Mihir Bellare. Does encryption with redundancy provide authenticity? In *Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6-10, 2001, Proceeding*, pages 512–528, 2001.

[Atk95] Ran Atkinson, 1995. IP Encapsulating Security Payload (ESP) RFC 1827.

[Bah14] L. Bahack, 2014. Julius. http://competitions. cr.yp.to/caesar-submissions.html.

[BBD$^+$01] C. Bormann, C. Burmeister, M. Degermark, H. Fukuhsima, H. Hannu, L-E. Jonsson, R. Hakenberg, T. Koren, K. Le, Z. Liu, A. Martensson, A. Miyazaki, K. Svanbro, T. Wiebke, T. Yoshimura, and H. Zheng, 2001. RObust Header Compression: Framework and Four Profiles: RTP, UDP, ESP, and Uncompressed (ROHC). RFC 3095.

[Bel96] Steven M. Bellovin. Problem areas for the IP security protocols. In *Proceedings of the 6th USENIX Security Symposium, San Jose, CA, USA, July 22-25, 1996*, 1996.

[BN00] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, pages 531–545, 2000.

[BR00] Mihir Bellare and Phillip Rogaway. Encode-then-encipher encryption: How to exploit nonces or redundancy in plaintexts for efficient cryptography. In *Advances in Cryptology - ASIACRYPT 2000, 6th International Conference on the Theory and Application of Cryptology and Information Security, Kyoto, Japan, December 3-7, 2000, Proceedings*, pages 317–330, 2000.

[BZD$^+$16] Hanno Bock, Aaron Zauner, Sean Devlin, Juraj Somorovsky, and Phillip Jovanovic. Nonce-disrespecting adversaries: Practical forgery attacks on GCM in TLS. *IACR Cryptology ePrint Archive*, 2016:475, 2016.

[CJ99] S. Casner and V. Jacobson, 1999. Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. RFC 2508.

[DDN00] D. Dolev, C. Dwork, and M. Naor. Non-malleable cryptography. *SIAM Jpurnal on Computing*, 30(2):391–437, 2000.

[Des00] Anand Desai. New paradigms for constructing symmetric encryption schemes secure against chosen-ciphertext attack. In *Advances in Cryptology - CRYPTO 2000, 20th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2000, Proceedings*, pages 394–412, 2000.

[Dwo05] Morris J. Dworkin. SP 800-38B. Recommendation for block cipher modes of operation: The CMAC mode for authentication. Technical report, Gaithersburg, MD, United States, 2005.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.

[GJMN16] Robert Granger, Philipp Jovanovic, Bart Mennink, and Samuel Neves. Improved masking for tweakable blockciphers with applications to authenticated encryption. In *Advances in Cryptology – EUROCRYPT 2016*, pages 263–293. Springer Berlin Heidelberg, 2016.

[Har08] Dan Harkins, 2008. Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES). RFC 5297.

[HKR15] Viet Tung Hoang, Ted Krovetz, and Phillip Rogaway. Robust authenticated-encryption AEZ and the problem that it solves. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology – EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, pages 15–44, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[IK03] Tetsu Iwata and Kaoru Kurosawa. OMAC: one-key CBC MAC. In Thomas Johansson, editor, *Fast Software Encryption: 10th International Workshop, FSE 2003, Lund, Sweden, February 24-26, 2003. Revised Papers*, pages 129–153. Springer Berlin Heidelberg, 2003.

[IY09a] Tetsu Iwata and Kan Yasuda. BTM: a single-key, inverse-cipher-free mode for deterministic authenticated encryption. In Michael J. Jacobson, Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography: 16th Annual International Workshop, SAC 2009, Calgary, Alberta, Canada, August 13-14, 2009, Revised Selected Papers*, pages 313–330. Springer Berlin Heidelberg, 2009.

[IY09b] Tetsu Iwata and Kan Yasuda. HBS: a single-key mode of operation for deterministic authenticated encryption. In Orr Dunkelman, editor, *Fast Software Encryption*, pages 394–415, Berlin, Heidelberg, 2009. Springer-Verlag.

[Kro14] T. Krovetz, 2014. HS1-SIV. http://competitions.cr.yp.to/caesar-submissions.html.

[KY00] Jon Katz and Moti Yung. Complete characterization of security notions for probabilistic private key encryption. In *Proceedings of the 32nd Annual Symposium on Theory of Computing*, pages 245–254. ACM, 2000.

[RS06] Phillip Rogaway and Thomas Shrimpton. A provable-security treatment of the key-wrap problem. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St. Petersburg, Russia, May 28 - June 1, 2006, Proceedings*, pages 373–390, 2006.

[RY13] Thomas Ristenpart and Scott Yilek. The mix-and-cut shuffle: Small-domain encryption secure against N queries. In *Advances in Cryptology - CRYPTO 2013 - 33rd Annual Cryptology Conference, Santa Barbara, CA, USA, August 18-22, 2013. Proceedings, Part I*, pages 392–409, 2013.

[Sho04] Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. *IACR Cryptology ePrint Archive*, 2004:332, 2004.

[ST13] Thomas Shrimpton and R. Seth Terashima. A modular framework for building variable-input-length tweakable ciphers. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part I*, pages 405–423, 2013.

[Str11] Rene Struik, 2011. Cryptography for Highly Constrained Networks. NIST CETA Workshop 2011.

[VA08] M. C. Vuran and I. F. Akyildiz. Cross-layer packet size optimization for wireless terrestrial, underwater, and underground sensor networks. In *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, April 2008.

[WGE$^+$05] A.S. Wander, N. Gura, H. Eberle, V. Gupta, and S. C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *Third IEEE International Conference on Pervasive Computing and Communications*, pages 324–328, 2005.

[WHF03] D. Whiting, R. Housley, and N. Ferguson, 2003. Counter with CBC-MAC (CCM) RFC 3610.