

# 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 CCS which is a new family of tweakable enciphering schemes (TES). 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). CMCC is an instantiation of the scheme providing misuse resistant authenticated encryption with associated data (AEAD), and it leverages existing modes such as CBC, Counter, and CMAC. Our work can be viewed as extending the line of work starting with [HR03] on TES's to plaintext sizes smaller than the block cipher block length which is a problem posed in [Hal04]. 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. 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 CCA2 security and misuse resistant authenticated encryption (MRAE) security for different variants of CMCC. Our contributions include both stateless and stateful versions which enable minimal sized message numbers using different network related trade-offs.

**Keywords:** Private key CCA2 encryption, 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 [Bellare] for attacks that utilize this property). Also, the most common schemes for CCA (Chosen Ciphertext Attack) security [Katz-Yung1] utilize a CPA (Chosen Plaintext Attack) encryption scheme combined with a MAC (Message Authentication Code) [DolvDwkNaor].

In this paper we present a new symmetric encryption scheme, Chosen Ciphertext Secure (CCS), that 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. We will show that this property implies that our scheme has CCA2 security. CCS is a tweakable enciphering scheme (TES) [LskvRvstWgnr, HR03]. We obtain CCA2 security with a small concrete security bound using only 2-3 bytes of ciphertext expansion.<sup>1</sup>

We will make use of variable length input pseudorandom functions  $f_i$  that have a fixed length output size. 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). Our encryption scheme is:

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

where the ciphertext is  $X_1, X_2$ , and  $M$  is a public message number (or the tweak [LskvRvstWgnr]). For maximum security,  $M$  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, it follows that the decryption oracle in the CCA2 security experiment (or inverse permutation in the definition of TES security) has limited usefulness to the adversary.

For longer messages, the plaintext  $P$  is split into the equal length substrings  $P_1, \dots, P_k$ , (the lengths may differ by one byte if necessary) and we have:

$$\begin{aligned} X &= f_k(M, P_1) \oplus P_k \\ X_k &= f_k(X) \oplus P_{k-1} \\ &\vdots \\ X_2 &= f_2(X) \oplus P_1 \\ X_1 &= f_1(M, X_2, \dots, X_k) \oplus X \end{aligned}$$

where the resulting ciphertext is  $X_1, \dots, X_k$ .

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 reuse of the message numbers. The stateful version uses a private message number which is encrypted and the last few bytes of the resulting ciphertext are sent with the ciphertext. This mechanism enforces a different trade-off; the limit here is on the

---

<sup>1</sup>An existing TES could also be used but a limitation of existing schemes is that the plaintext must be at least as long as the block cipher length (16 bytes for AES).

maximum amount of disorder between encryption order and decryption order. It also hides the number of messages previously sent. If the communication peers are synchronized, then CCS requires no additional overhead for message numbers.

Alternatively, the  $k = 2$  construction above can also be applied to longer messages (see Section 3.3). We describe a particular instantiation of CCS: CBC-MAC-CTR-CBC (CMCC) mode. CMCC is a general purpose authenticated encryption mode [BellrNamp]. We apply CBC encryption in the first equation above (and replace  $f_2$  with  $f_3$ ), use a MAC followed by a CTR mode variant in the 2nd equation, and CBC encryption again in the 3rd equation. We prove that CMCC is misuse resistant [RogwyShrmptn]: 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. Our approach for including authentication bits follows PTE [RogwyShrmptn]: the authentication tag consists of a string of zero bits and a modification of the ciphertext will randomize the zero bit string which is detected after decryption. We also consider the case where the authentication string is the MAC of the plaintext and associated data: CMCC with MAC or CWM. For CWM, we obtain stronger security bounds. Relative to SIV [RogwyShrmptn], CMCC has smaller ciphertext expansion.

## 1.1 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 CCM or other 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. Our scheme is computationally comparable to existing schemes such as CCM [WhitHousFerg], but yields reduced message sizes. 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 [cRTP, Bormann] 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 CCS with 2 bytes of expansion, we obtain a 1/5 savings in message size and corresponding savings in energy utilization. (Actually, the savings is larger since encryption schemes send randomness (e.g., an IV) as well. For example, CCM sends a 13 byte nonce with each message.) 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 [VuranAkyldz] 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.

## 1.2 Our Contributions

Our contributions are as follows:

1. We give a new family of private key encryption schemes with minimal ciphertext expansion. We obtain CCA2 security with a small concrete security bound using only 2-3 bytes of ciphertext expansion, for a full range of message sizes. Our work can be viewed as extending the line of work starting with [HR03] to plaintext sizes smaller than the block cipher block length. Halevi posed this problem in [Hal04]. When message numbers are not reused for CMCC, we obtain a security bound which is dominated by  $q/\beta$  where  $\beta$  is the minimum of the block length and half the length of the plaintext plus the length of the authentication tag for the challenge ciphertext (here we allow a zero length authentication tag). When message numbers are not reused and we include an authentication tag with  $\tau$  bits, then we obtain a bound dominated by  $1/\beta$  if invalid queries result in session termination,  $q/\beta$  if invalid queries do not result in session termination,  $q/(2^\tau\beta)$  if the authentication tag is computed using a keyed MAC algorithm (CWM) but invalid queries are allowed and  $2^{-\tau}/\beta$  for CWM where invalid queries result in session termination.
2. We instantiate the CCS construction with a new block cipher mode: CMCC. CMCC is a general purpose misuse resistant authenticated encryption mode. We prove security for the misuse resistant scheme in the TES security model [HR03]. Our misuse resistant scheme has less ciphertext expansion than SIV [RogwyShrmpn].
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 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. Here we assume an 8 byte authentication tag for CCM and a 2 byte authentication string for CMCC. This gives a comparable level of CCA2 security for plaintexts in the 12-16 byte range (see Remark in Section 4). Authentication security is increased beyond 2 bytes for CMCC given the higher layer protocol checks. The exact strength is dependent on the specific protocols including the application layer. For some protocols, a 2 byte authentication string would not be sufficient for authentication and would have to be increased in length. On the other hand, a protocol such as VoIP has application layer checks and a randomized voice packet will not constitute a successful attack. Thus a 2 byte authentication string for CMCC may be sufficient for an application such as VoIP.

## 1.3 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 [Atknsn]. However, [Bellare] 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 [RogwyBellr]. 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 [AnBellr], the authors show that public redundancy is not always sufficient and that private (keyed) redundancy leads to stronger authentication properties. Struik [Struik] presented application requirements and constraints, independently of this work at roughly the same time this work was started.

In [Desai], 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. CCS shares this CCA-security without a MAC property. 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. Furthermore, with small probability, collisions will occur after a much smaller number of messages. The UFE security bound is  $q(q+1)/2^{|r|}$ . If the adversary can make  $2^{20}$  queries, then Theorem 4.2 gives a security bound around  $2^{-60}$  for CMCC, given a 20 byte message. UFE would require a 13 byte ciphertext expansion to assure the same security level.

SIV and PTE [RogwyShrmptn] are misuse resistant authenticated encryption (MRAE) schemes. 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 RFC 5297 specification of SIV has the same number of block cipher invocations as CCM.

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

Collisions in the IV [RogwyShrmptn] (or random message number in [Desai]) 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  $\sigma$  parameter in [Desai] and to the IV in [RogwyShrmptn]. These last two parameters create ciphertext expansion whereas  $X$  does not. Our scheme is targeted at environments where minimizing ciphertext expansion is valuable.

CMC [HR03] is the first of the tweakable enciphering schemes (TES), originally motivated by the problem of disk encryption. CMC sandwiches a masking layer (involving xor and a pass over the message blocks) in between two encryption layers. CMC plaintexts must be a multiple of the block cipher length. EME [HR04] and EME\* [Hal04] are improved schemes with the latter able to encrypt any length equal or longer than the block length. Halevi [Hal04] poses the open problem of encrypting short plaintexts with lengths less than the block length.

Naor and Reingold [NR] initiated another approach for constructing a TES: hash-ECB-hash. The schemes here include PEP [CS06b], TET [Hal07], HEH [Sarkar], iHCTR and HOH [Sarkar]. The hashing layers use finite field multiplications so they obtain a performance advantage over the earlier schemes when finite field operations become significantly faster than block cipher operations. A third approach, hash-CTR-hash, is embodied in HCTR [WFW05] and HCH [CS06a].

Since our scheme uses encryption only in the forward direction combined with xor, our construction is able to handle messages of varying lengths including lengths shorter than the underlying block length which is an advantage over CMC and the above schemes. The stateful version of our scheme includes the integration of a minimal sized message number that enables the number of

messages previously sent to be hidden. We also require one less block cipher invocation than CMC and EME\*. The EME ciphers are more parallelizable.

## 1.4 Organization

In Section 2, we give basic cryptographic definitions. In Section 3, we present our tweakable enciphering scheme CCS; we also give the CMCC instantiation including the authenticated encryption scheme with minimal ciphertext expansion. Section 4 gives the proof that our scheme has CCA2 security and (misuse resistant) TES security. Section 5 gives our performance analysis and results, including a comparison of energy utilization between CCS and CCM, for wireless sensor nodes. In Section 6 we draw conclusions.

# 2 Definitions

## 2.1 Pseudorandomness

The concatenation of two strings  $S$  and  $T$  is denoted by  $S|T$ , or  $S,T$  where there is no danger of confusion.

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  $x$ .

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

We follow [GGM86] as explained in [Shoup] 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

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

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

where the maximum is over adversaries that run with number of queries bounded by  $q$ .

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  $H_{l_1, l_2}$ .

## 2.2 (Misuse Resistant) CCA Encryption

Given the symmetric key encryption scheme  $S = (Gen, Enc, Dec)$ . We define the CCA2 encryption experiment  $Exp_{CCA2}(S, n, q, \mathcal{A})$  here:

1. The algorithm  $Gen(1^n)$  is run and the key  $K$  is generated.
2. The adversary  $\mathcal{A}$  is given the input  $1^n$  and oracle access to  $Enc_K()$  and  $Dec_K()$ .
3. The adversary outputs a pair of messages  $m_0$  and  $m_1$  of the same length.

4. A random bit  $b \leftarrow \{0, 1\}$  is selected. The ciphertext  $c \leftarrow Enc_K(m_b)$  is computed and given to  $\mathcal{A}$ .
5. The adversary continues to have oracle access to  $Enc_K()$  and  $Dec_K()$ . However, the adversary is not allowed to query the decryption oracle with the ciphertext  $c$ . The adversary is limited to  $q$  total queries (including the queries issued before the challenge ciphertext is generated).
6. The adversary outputs a bit  $\bar{b}$ . The output of the experiment is 1 if  $\bar{b} = b$  and 0 otherwise.

Inputs to  $Enc_K()$  are of the form  $(P, M)$ , and inputs to  $Dec_K()$  are of the form  $(C, M)$  where  $M$  is a message number, and the adversary may not reuse  $M$  with the same key. If  $Dec_K(C, M) = P$ , for adversary query  $(C, M)$ , then the adversary will not subsequently submit  $(P, M)$  to  $Enc_K()$ .

The encryption scheme  $S$  is defined to have CCA2 security for  $(\epsilon, q)$  if for all probabilistic polynomial time adversaries  $\mathcal{A}$  limited to  $q$  queries,  $Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1] \leq 1/2 + \epsilon$ . We define  $Adv_{S, n, q}^{CCA2}(\mathcal{A}) = Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1] - 1/2$ .

We also define the CCA2 MRAE security experiment which is identical to the experiment above except the adversary may reuse the message number  $M$  with the same key. However, no query can be submitted twice. In particular,  $m_0$  and  $m_1$  must be new queries. The encryption scheme  $S$  is defined to have CCA2 MRAE security for  $(\epsilon, q)$  if for all probabilistic polynomial time adversaries  $\mathcal{A}$  limited to  $q$  queries,  $Pr[Exp_{CCA2\_MRAE}(S, n, q, \mathcal{A}) = 1] \leq 1/2 + \epsilon$ . We define  $Adv_{S, n, q}^{CCA2\_MRAE}(\mathcal{A}) = Pr[Exp_{CCA2\_MRAE}(S, n, q, \mathcal{A}) = 1] - 1/2$ .

### 2.3 (Misuse Resistant) CPA Encryption

Given the CCA2 encryption experiment above, except we remove the decryption oracle from the experiment. We define the resulting experiment as the CPA encryption experiment, and if the adversary probability of success is bounded as above, we say that the encryption scheme is CPA secure for  $(\epsilon, q)$ . We have the analogous definitions for  $Adv_{S, n, q}^{CPA}(\mathcal{A})$  and  $Adv_{S, n, q}^{CPA\_MRAE}(\mathcal{A})$ .

### 2.4 Tweakable Enciphering Scheme Security

We follow [HR03] for our definitions and notation. A tweakable enciphering scheme (TES) is a function  $\mathcal{E} : \mathcal{K} \times \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$  where for every  $K \in \mathcal{K}$  and  $T \in \mathcal{T}$ ,  $\mathcal{E}_K^T$  is a length preserving permutation.  $Perm^{\mathcal{T}}(\mathcal{M})$  denotes the set of all functions  $\pi : \mathcal{T} \times \mathcal{M} \rightarrow \mathcal{M}$  such that  $\pi(T, \cdot)$  is a length preserving permutation for all  $T \in \mathcal{T}$ .

Let  $\mathcal{E}$  be a TES, and let  $A$  be an adversary. We define the advantage of  $A$  in distinguishing between  $\mathcal{E}$  and a random tweakable length-preserving permutation and its inverse as

$$\mathbf{Adv}_{\mathcal{E}}^{\pm \widetilde{prp}}(A) = Pr[K \leftarrow \mathcal{K} : A^{\mathcal{E}_K(\cdot, \cdot)} \mathcal{E}_K^{-1}(\cdot, \cdot) \Rightarrow 1] - Pr[\pi \leftarrow Perm^{\mathcal{T}}(\mathcal{M}) : A^{\pi(\cdot, \cdot)} \pi^{-1}(\cdot, \cdot) \Rightarrow 1]$$

$$\pi^{-1}(Y, T) = X \text{ if there exists } X \text{ such that } \pi(X, T) = Y, \text{ otherwise } \pi^{-1}(Y, T) = \perp .$$

## 3 CCS

In this section, we present CCS. CCS includes a stateless version with public message numbers, and a stateful version with private message numbers. CCS is based on a variable input length pseudorandom function (we give examples of these later in the paper). The terminology  $f_i$  refers to a keyed pseudorandom function (keyed with key  $K_i$ ).  $M$  is the message number.

We assume  $f_i$  maps an arbitrary length domain string to a fixed length output string, where the output length is the same across all  $i$ . We call the output length the output block size.  $k$  is the number of bytes in the plaintext divided by the output block size (in bytes), and then rounded up to the nearest integer. If this integer is one, then  $k = 2$  :

$$k = \max\{\lceil |P|/\text{output block size} \rceil, 2\}.$$

We will segment a plaintext message  $P$  into  $k$  input blocks. The input block size for  $P$  is the largest size less than or equal to the output block size such that the message  $P$  can be divided into  $k$  input blocks each with the input block size or one byte less than the input block size if needed. If  $P$  divides into  $k$  equal sized blocks, then input block size =  $|P|/k$ . We define  $\alpha$  to be 2 raised to the input block size, in bits. As an example, consider a pseudorandom function constructed using the AES encryption algorithm [AES]. The output block size is 16 bytes. If  $P$  has 33 bytes, then  $k = 3$ , the input block size is 11 bytes, and  $\alpha = 2^{88}$ .

### 3.1 Informal Design Intuition for Message Numbers

$M$  is a per message value that can be selected by the caller of the encryption API. Our goal is to allow the caller to use any strategy or algorithm for selecting  $M$ . For the  $k > 2$  case, the caller must not reuse  $M$ ; reusing  $M$  will result in a loss of CPA security. The  $k = 2$  case is misuse resistant when the authentication field is included; security is maintained provided that the same message number is not reused with the same key and plaintext. When the caller explicitly selects  $M$ , then the scheme uses  $M$  as the public message number and is stateless.

We also 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  $E$  (we assume the block size is 16 bytes). 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  $(\gamma - j) \leq 2(\text{window\_size}) + 1$  given  $M_i$  as the  $\gamma$ th element in the sequence (after eliminating previous last  $IL$ -byte duplicates and  $M_j$  is the  $j$ th 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  $2(\text{window\_size}) + 1$  will be less than the birthday bound). The  $\text{window\_size}$  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 size of the ciphertext but the scheme is stateful.

### 3.2 CCS Specification

$LSB_j(x)$  and  $MSB_j(x)$  denote the  $j$  least significant bytes and  $j$  most significant bytes of byte string  $x$  respectively. 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; for public message numbers (stateless case),  $\bar{K}_1$ ,  $\bar{K}_2$ ,  $E_{\bar{K}_1}$ , and  $E_{\bar{K}_2}$  are not used, and Sections 3.2.2, 3.2.3, and 3.2.6 are not needed. Also,  $M$  replaces the message number tag  $T$  in Sections 3.2.4 and 3.2.5.

### 3.2.1 Key Generation

Keys  $\bar{K}_1$  and  $\bar{K}_2$  are randomly generated for the pseudorandom permutations  $E_{\bar{K}_i}$   $i = 1, 2$  and the randomly generated keys  $L_1, \dots, L_k$  determine the PRF's  $f_1, \dots, f_k$ . The key  $K = \bar{K}_1, \bar{K}_2, L_1, \dots, L_k$ .  $E_{\bar{K}_i}$  is a permutation on the set of binary strings with  $l$  bits.

### 3.2.2 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 of ciphertext expansion.  $w_s$  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.

### 3.2.3 Creating the Sequences of $M$ Values

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

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

{  
 $M_{x_e} = candidate(M)$ ;  $Seq(x) = M_0, \dots, M_{x_e}$

$u_x = u_x + 1$ ;

}

ENDIF

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

### 3.2.4 Encryption

Given private message number  $i$  where  $i = SeqNo_x[M]$ . We set  $T = LSB_{IL}(M)$ .  $T$  is the message number tag. We assume  $P$  is a plaintext byte string (the number of bits in  $P$  is divisible by 8). For  $k \geq 2$ , the plaintext  $P$  is split into the equal length substrings, where length is the input block size,  $P_1, \dots, P_k$ ; (the lengths may differ by one byte per our discussion above, but for convenience we will assume they are equal length for the remainder of the paper and all of our results hold with only minor changes in the non equal case) the encryptor computes the following values sequentially (but the 2nd through 2nd to last values can be computed in parallel):

$$\begin{aligned}
X &= f_k(M, P_1) \oplus P_k \\
X_k &= f_k(X) \oplus P_{k-1} \\
&\vdots \\
X_2 &= f_2(X) \oplus P_1 \\
X_1 &= f_1(M, X_2, \dots, X_k) \oplus X
\end{aligned}$$

where  $X_1| \dots | X_k|T$  is the resulting ciphertext. We write  $Enc_K(P, M) = X_1| \dots | X_k|T$ . For a pseudorandom function based on an underlying cryptographic algorithm with a block size (e.g., an AES based prf), padding may be necessary. In this case, we pad using the padding algorithm from the CMAC specification [CMAC].

### 3.2.5 Decryption

Let  $y \in \{init, resp\}$  where  $y \neq x$ . Given  $C|T$  where  $C = X_1| \dots | X_k$ . 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 it exists, then set  $M = \bar{M}$  and compute the sequence

$$\begin{aligned}
X &= f_1(M, X_2, \dots, X_k) \oplus X_1 \\
P_1 &= X_2 \oplus f_2(X) \\
&\vdots \\
P_{k-1} &= X_k \oplus f_k(X) \\
P_k &= X \oplus f_k(M, P_1)
\end{aligned}$$

and output  $Dec_K(C, T) = P_1| \dots | P_k$ . Otherwise, output  $Dec_K(C, T) = \perp$ .

If  $Dec_K(C, T) \neq \perp$ , then we say  $M$  is the message number used to decrypt  $C, T$ ;  $SeqNo_x[M]$  is the corresponding private message number. In this case, if  $SeqNo_x[M] > y_d$ , then set  $y_d = SeqNo_x[M]$ .

### 3.2.6 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.

## 3.3 CMCC

Although our emphasis has been on utilizing CCS to protect short messages in energy constrained environments, we now discuss further a specific instantiation of the  $k = 2$  case of CCS: CBC-MAC-Counter-CBC (CMCC) mode. CMCC is a general purpose authenticated encryption mode

<i>Parameter</i>	<i>Description</i>
$k$	Number of plaintext segments: $P = P_1   \dots   P_k$ .
$\alpha$	$\alpha = 2^{ P_i }$ , $i = 1, \dots, k$ .
$M$	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 CCS Scheme

which is misuse resistant and optimized for energy constrained environments. As before, we will have a stateless version with public message numbers, and a stateful version with private message numbers. Also as before, the scheme is identical to  $k \geq 2$  when the message has 32 bytes or less. 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.

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 a variant of counter mode encryption to encrypt  $P_1$  and obtain  $X_2$ . Finally we create the other half of the ciphertext,  $X_1$  using CBC mode applied to  $X_2$  and exclusive-or with  $X$ .

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.

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

### 3.3.1 Notation

We use  $|$  to denote concatenation of strings, and  $\oplus$  denotes bitwise xor.  $b^j$  is the bit  $b$  repeated  $j$  times. 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.  $x \ll n$  denotes the left shift operator (filling vacated bits with zero bits), after shifting the string  $x$  by  $n$  bits to the left.  $|S|$  denotes the length of the string  $S$ .  $B$  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.

### 3.3.2 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

$(B/8 \text{ bytes})^2$  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:
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,

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

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  $K1 = L \ll 1$ , otherwise, we define  $K1 = (L \ll 1) \oplus R_{128}$ . If the most significant bit of  $K1$  is zero, then define  $K2 = K1 \ll 1$ . Otherwise, we define  $K2 = (K1 \ll 1) \oplus R_{128}$ .

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  $S_1$  :  $10^{pad\_length-1}$ .  $pad(S_1)_{S_2}$  is  $S_1$  with the last block of  $S_1$  replaced with  $F|10^{pad\_length-1} \oplus K2$ .

Figure 1 gives a pictorial description of CMCC stateless encryption.  
Figure 4 gives the stateful version.

## 4 Proof of Security

We first give some examples illustrating attacks against CMCC. We then prove CPA security for the stateless version of CCS (the message number tag  $T$  defined above will only be used in the stateful scheme). We will then prove CCA2 security for CMCC (with a zero length authentication tag); the bound is dominated by  $q/\beta$ . When message numbers are not reused and we include an authentication tag with  $\tau$  bits, then we obtain a bound dominated by  $1/\beta$  if invalid queries result in session termination,  $q/\beta$  if invalid queries do not result in session termination,  $q/(2^\tau \beta)$  if the authentication tag is computed using a keyed MAC algorithm (CWM) but invalid queries are allowed and  $2^{-\tau}/\beta$  for CWM where invalid queries result in session termination.

For MRAE security, we use the TES definition of security. The TES message space will be the encoded messages obtained by appending the authentication tag consisting of  $\tau$  zero bits to the plaintext (see Figure 2). Thus arbitrary ciphertext queries most likely receive the value  $\perp$  as a response.

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 CCA-MRAE secure. 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

---

<sup>2</sup>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.

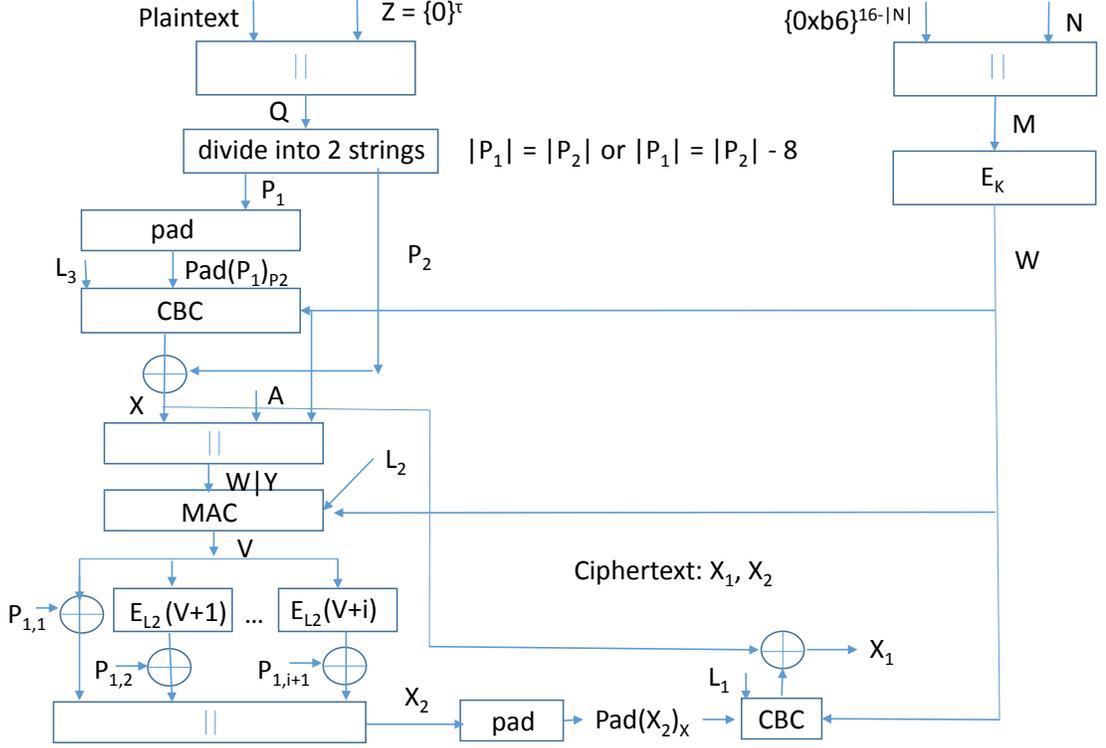


Figure 1: CMCC Stateless Encryption

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. The adversary submits one of these plaintexts in a query and uses the other one as one of the plaintexts for the challenge ciphertext. The adversary wins with high probability.

**Example 2:** Given a collision of  $X$  values for two plaintext queries in the CCA-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 privacy.

**Example 3:** Consider the CPA security game. Suppose the adversary submits identical plaintext queries  $(P_1, P_2)$  with unique message numbers, where  $|P_1| = |P_2| = 2^B$ . The challenge ciphertext is the encryption of the same plaintext or  $(P_1, \bar{P}_2)$  where  $P_2 \neq \bar{P}_2$ . Then the  $X$  values for the queries are distinct and also distinct for the encryption of  $(P_1, P_2)$ . But there is a probability of collision bounded by  $q/2^B$  if the challenge ciphertext is the encryption of  $(P_1, P_2)$ . The adversary advantage is bounded by  $q/2^B$ .

### CMCC Mode - Encryption

$CBC(IV, P, Key)$  is CBC encryption with initialization vector  $IV$ , plaintext  $P$ , and key  $Key$ .  $MAC(IV, P, Key)$  is MAC algorithm with output string of length  $l/8$  bits (one block) with initialization vector  $IV$ , plaintext  $P$ , and key  $Key$ .  $pad()$  is the padding algorithm defined in Section 3.3.  $E_{\bar{K}}$  is the block cipher with key  $\bar{K}$ .

**Encryption Inputs:** plaintext  $P$ , key  $K = \bar{K}, L_3, L_2, L_1$ , public message number  $N$ , and associated data  $A$ .

Given constant  $0xb6\dots0xb6$ , (repeated 16 times), we take the  $16 - |N|$  most significant bytes and prepend them to  $N$  to obtain  $M$ , where  $|N|$  denotes the length of  $N$  in bytes.

Let  $Z$  be the bit string with  $\tau$  zero bits ( $\tau$  is the number of authentication bits).

Let  $W = E_{\bar{K}}(M)$ .

$Q = P|Z$ .

Let  $Q = P_1|P_2$  where  $|P_1| = |P_2|$  or  $|P_1| = |P_2| - 8$  ( $P_1$  may be one byte shorter than  $P_2$ .)

$X = CBC(W, pad(P_1)_{P_2}, L_3) \oplus P_2$ ,  $X$  is truncated to the length of  $P_2$ .

$Y = X|A$ ,  $V = MAC(W, Y, L_2)$ ,

$P_1 = \bar{P}_{1,1}|\dots|\bar{P}_{1,i}|\bar{P}_{1,i+1}$  where  $i \geq 0$ ,  $\bar{P}_{1,1}, \dots, \bar{P}_{1,i}$  are full blocks and  $\bar{P}_{1,i+1}$  is a partial (possibly empty) block,

$X_2 = V \oplus \bar{P}_{1,1}|E_{L_2}(V + 1) \oplus \bar{P}_{1,2}|\dots|E_{L_2}(V + i) \oplus \bar{P}_{1,i+1}$ .

( $E_{L_2}(V + j)$  is truncated to the length of  $\bar{P}_{1,j+1}$  for  $j \geq 1$ , and bits 31,63 of  $V$  are zeroed for  $j=1$ .)

$X_1 = CBC(W, pad(X_2)_X, L_1) \oplus X$ ,  $X_1$  is truncated to the length of  $X$ .

**Ciphertext:**  $X_1, X_2$

Figure 2: CMCC Mode Encryption - Stateless Version

**Decryption Inputs:**  $X_1, X_2, M, A$

$$W = E_{\bar{K}}(M).$$

$$X = CBC(W, pad(X_2)_{X_1}, L_1) \oplus X_1$$

$$Y = X|A, V = MAC(W, Y, L_2)$$

$X_2 = \bar{X}_{2,1}|\dots|\bar{X}_{2,i}|\bar{X}_{2,i+1}$  where  $i \geq 0$  and  $\bar{X}_{2,1}, \dots, \bar{X}_{2,i}$  are full blocks and  $\bar{X}_{2,i+1}$  is a partial empty block,  $P_1 = V \oplus \bar{X}_{2,1}|E_{L_2}(V + 1) \oplus \bar{X}_{2,2}|\dots|E_{L_2}(V + i) \oplus \bar{X}_{2,i+1}$

$$P_2 = CBC(W, pad(P_1)_X, L_3) \oplus X$$

$$Q = P_1|P_2, U = LSB_{\tau/8}(Q)$$

if  $(U \neq Z)$ , return  $\perp$ , otherwise  $Q = \tilde{P}|Z$  and return **Plaintext**  $\tilde{P}, M$

Figure 3: CMCC Mode Decryption - Stateless Version

We now prove that CCS is CPA secure.

**Theorem 4.1** *If  $k = 2$ , then the CCS encryption presented in the previous section is CPA MRAE secure for  $(\epsilon, q)$  with*

$$\epsilon = q(q-1)/\alpha + \sum_{i=1}^k Adv_{f_i}^{prf}(q)$$

*given that the adversary is restricted to  $q$  queries and given  $\alpha = 2^m$  where  $Len$  is the bit length of the challenge ciphertext and  $m = \lfloor Len/2 \rfloor$ . If  $k \geq 2$ , then CCS encryption is CPA secure for  $(\epsilon, q)$  for the same value of  $\epsilon$  provided that message numbers are not reused.*

**Proof:** We first handle the  $k = 2$  case. We will initially assume that  $f_1$  and  $f_2$  are random functions (in the idealized model). We will first obtain the probability bound for ensuring no collisions in the  $X$  values for the adversary queries. For  $2 \leq i \leq q$ ,  $(i-1)/\alpha$  is an upper bound on the probability that the  $X$  value for the  $i$ th ciphertext collides with the  $X$  value for one of the first  $i-1$  ciphertexts. Thus

$$\left(1 - \frac{q-1}{\alpha}\right) \dots \left(1 - \frac{1}{\alpha}\right) \approx e^{-q(q-1)/2\alpha}$$

is a lower bound on the probability of no collisions in the  $X$  values for the adversary queries. For sufficiently small values of  $q(q-1)/2\alpha$ , we can approximate the right hand side in the above inequality by  $1 - (q(q-1)/2\alpha)$  and use  $q(q-1)/2\alpha$  as the upper bound on the probability of collisions in the  $X$  values.

Since the  $X$  values are distinct, and  $f_2$  is a random function, it follows that the  $f_2(X)$  values are uniformly distributed and independent. Thus the  $X_2$  values give no information about  $P_1$ . Since  $X_2$  is uniform random, it follows that  $f_1(M, X_2)$  is also uniform random and thus the  $X_1$  values give no information about the  $X$  values, except if there is a collision between two query  $X_2$  values. As discussed above for collisions between  $X$  values, we can use  $q(q-1)/2\alpha$  as the upper bound on

**Encryption Inputs:** plaintext  $P$ , key  $K = \bar{K}, L_3, L_2, L_1$ , private message number  $i$ , and associated data  $A$ .

$Z$  is the bit string with  $\tau$  zero bits;  $Q = P|Z$

Let  $Q = P_1|P_2$  where  $|P_1| = |P_2|$  or  $|P_1| = |P_2| - 8$  ( $P_1$  may be one byte shorter than  $P_2$ .)

State initialization is per the Key Generation, Initial State, and Creating the Sequence of Secret Message Numbers subsections above.

Let  $i = SeqNo_x[M]$ .  $X = CBC(M, pad(P_1)_{P_2}, L_3) \oplus P_2$ ,  $X$  is truncated to the length of  $P_2$ .

$Y = X|A$   
 $V = MAC(M, Y, L_2)$ ,

$P_1 = \bar{P}_{1,1}|\dots|\bar{P}_{1,i}|\bar{P}_{1,i+1}$  where  $i \geq 0$ ,  $\bar{P}_{1,1}, \dots, \bar{P}_{1,i}$  are full blocks and  $\bar{P}_{1,i+1}$  is a partial (possibly empty) block,

$X_2 = V \oplus \bar{P}_{1,1}|E_{L_2}(V + 1) \oplus \bar{P}_{1,2}|\dots|E_{L_2}(V + i) \oplus \bar{P}_{1,i+1}$ .  
( $E_{L_2}(V + j)$  is truncated to the length of  $\bar{P}_{1,j+1}$  for  $j \geq 1$ , and bits 31,63 of  $V$  are zeroed for  $j=1$ .)

$X_1 = CBC(M, pad(X_2)_X, L_1) \oplus X$ , where  $X_1$  is truncated to the length of  $X$ .

**Ciphertext:**  $X_1, X_2, T = LSB_{IL}(M)$

**Decryption Inputs:**  $X_1, X_2, T, A$

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 it exists, then  $M = \bar{M}$ , otherwise return  $\perp$ .

If  $Dec_K(C, T) \neq \perp$ , then we say  $M$  is the message number used to decrypt  $C, T$ ;  $SeqNo_x[M]$  is the corresponding private message number. In this case, if  $SeqNo_x[M] > y_d$ , then set  $y_d = SeqNo_x[M]$ .  
 $X = CBC(M, pad(X_2)_{X_1}, L_1) \oplus X_1$

$Y = X|A$ .  
 $V = MAC(M, Y, L_2)$ ,

$X_2 = \bar{X}_{2,1}|\dots|\bar{X}_{2,i}|\bar{X}_{2,i+1}$  where  $i \geq 0$  and  $\bar{X}_{2,1}, \dots, \bar{X}_{2,i}$  are full blocks and  $\bar{X}_{2,i+1}$  is a partial empty block,  $P_1 = V \oplus \bar{X}_{2,1}|E_{L_2}(V + 1) \oplus \bar{X}_{2,2}|\dots|E_{L_2}(V + i) \oplus \bar{X}_{2,i+1}$

$P_2 = CBC(M, pad(P_1)_X, L_3) \oplus X$

$Q = P_1|P_2$   
 $U = LSB_{\tau/8}(Q)$

if  $(U \neq Z)$ , return  $\perp$ , otherwise  $Q = \tilde{P}|Z$  and return **Plaintext**  $\tilde{P}$ ,  $i = SeqNo_x(M)$

Figure 4: CMCC Mode - Stateful Version

the probability of collisions in the  $X_2$  values. Thus the ciphertexts give no information about the  $X$  values.

We have

$$\begin{aligned}
Pr[\mathcal{A} \text{ guesses } b] &= Pr[\mathcal{A} \text{ guesses } b \wedge \text{collision}] + Pr[\mathcal{A} \text{ guesses } b \wedge \text{no collision}] \\
&\leq Pr[\text{collision}] + Pr[\mathcal{A} \text{ guesses } b \wedge \text{no collision}] \\
&\leq q(q-1)/\alpha + Pr[\mathcal{A} \text{ guesses } b | \text{no collision}] \\
&= q(q-1)/\alpha + 1/2.
\end{aligned}$$

Now we prove the case where the  $f_i$  functions are pseudorandom functions (prfs). We construct an adversary  $D^g$  where  $g$  is either  $(h_1, h_2)$  or  $(h_1, f_2)$  and  $h_i$ ,  $1 \leq i \leq 2$  are random functions and  $f_2$  is a prf. Then  $Adv_{(h_1, h_2)}^{CPA\_MRAE} \leq q(q-1)/\alpha$ .  $D^g$  will attack  $f_2$  as a prf. Let  $\mathcal{A}$  be an adversary that attacks our encryption scheme.  $D^g$  runs  $\mathcal{A}$ .  $D$  uses  $g$  to answer  $\mathcal{A}$ 's encryption and decryption oracle queries. When  $\mathcal{A}$  outputs bit  $b$ ,  $D$  also outputs bit  $b$ .

$$\begin{aligned}
Adv_{f_2}^{prf}(q) &\geq Adv_{f_2}^{prf}(D^g) = |Pr[D^{(h_1, f_2)}() = 1] - Pr[D^{(h_1, h_2)}() = 1]| \\
&\geq Adv_{(h_1, f_2, n, q)}^{CPA\_MRAE}(\mathcal{A}) - q(q-1)/\alpha.
\end{aligned}$$

Thus  $Adv_{(h_1, f_2, n, q)}^{CPA\_MRAE}(\mathcal{A}) \leq Adv_{f_2}^{prf}(q) + q(q-1)/\alpha$  for all adversaries  $\mathcal{A}$ . Now let  $g = (h_1, f_2)$  or  $g = (f_1, f_2)$  where  $f_1$  and  $f_2$  are prfs and  $h_1$  is a random function. Then

$$\begin{aligned}
Adv_{f_1}^{prf}(q) &\geq Adv_{f_1}^{prf}(D^g) = |Pr[D^{(f_1, f_2)}() = 1] - Pr[D^{(h_1, f_2)}() = 1]| \\
&\geq Adv_{(f_1, f_2, n, q)}^{CPA\_MRAE}(\mathcal{A}) - Adv_{f_2}^{prf}(q) - q(q-1)/\alpha.
\end{aligned}$$

for all adversaries  $\mathcal{A}$ . Thus  $Adv_{(f_1, f_2, n, q)}(\mathcal{A}) \leq q(q-1)/\alpha + \sum_{i=1}^2 Adv_{f_i}^{prf}$  for all adversaries  $\mathcal{A}$ .

For the  $k \geq 2$  case, we note that MRAE security isn't possible since the adversary can submit distinct queries (e.g., with distinct  $P_3$  and equal  $P_1, P_2$ , and  $M$  values) that result in ciphertexts with equal  $X$  values and a high probability of winning the CPA security game. Since the message numbers  $M$  are all distinct, we again have the same probability bound to ensure the  $X$  values are distinct. Thus we obtain the  $k \geq 2$  result in the same manner as for the above argument.  $\blacksquare$

We now prove CCA2 security (no message number reuse) for CMCC. The intuition for the proof is as follows: We cannot, except for padding based collisions in  $CBC(W, P_1)$  (based on encoding distinct  $P_1$  values to identical encoded values), create two new plaintexts that will yield identical  $X$  values. The reason is that  $M$  is fresh for the challenge ciphertext. Given the challenge ciphertext, we have the  $q/\beta$  bound for  $X$  collisions and  $P_1$  collisions involving plaintext and ciphertext queries respectively. We also have the bound for collisions in the counter mode blocks and CBC blocks. Other than these events,  $X$  is fresh and  $CBC(W, P_1)$  is also fresh so  $P_1$  and  $P_2$  aren't leaked to the adversary ( $W$  is fresh for all plaintext queries, and  $P_1$  is fresh for all the ciphertext queries).

$$\text{Let } \chi(m) = \begin{cases} 1 & \text{if } m > 128 \\ 0 & \text{otherwise.} \end{cases}$$

**Theorem 4.2** *The adversary is restricted to  $q$  queries; given  $\alpha = 2^m$  where  $Len$  is the bit length of the challenge ciphertext and  $m = \lfloor Len/2 \rfloor$ .  $B$  is the block length and  $\mu$  is the total number of blocks in all the query plaintexts and ciphertexts. Let  $\beta = \min\{\alpha, 2^B\}$ . The stateless and stateful*

versions of CMCC (where the authentication string  $Z$  can be any length including zero length) are CCA2 secure for  $(\epsilon, q)$  with

$$\epsilon = q/\beta + 2q/2^B + \chi(m)\lceil\mu/2 + q\rceil^2/2^{B+2} + \sum_{i=1}^3 Adv_{f_i}^{prf}(q) + Adv_{E_{\bar{K}}}^{prf}(q)$$

**Proof:** **case i:** challenge ciphertext has  $\alpha < 2^B$ .

We assume that the truncated block cipher encryptions are pseudorandom functions, and we initially work in the idealized model where these functions are random functions. Let the challenge ciphertext be  $M, X_1, X_2$ . Given  $q_e$  encryption queries and  $q_d$  decryption queries where  $q_e + q_d = q$ .

Let  $Bad_1$  be the event where the  $Y$  value for the challenge ciphertext equals the  $Y$  value for one of the encryption query ciphertexts. (We will see that the adversary's optimal strategy will not have any  $Y$  collisions between the challenge ciphertext and the ciphertext query ciphertexts.) Let  $Bad_2$  be the event where the returned  $P_1$  value for a ciphertext query equals the  $P_1$  value for the challenge ciphertext. For the computation of  $V$ , if event  $Bad_1$  does not occur, then the idealized MAC PRF function is invoked on a fresh value and  $V$  is uniformly distributed. Thus  $X_2$  leaks no information about  $P_1$  or  $X$ . Also, the idealized pseudorandom function for computing  $X_1$  leaks no information about  $X$  or  $P_2$ , given that  $Bad_2$  does not occur. (The optimal adversary strategy for ciphertext queries is to submit queries of the form  $M, \bar{X}_1, X_2$  for  $\bar{X}_1 \neq X_1$ .)

Thus

$$\begin{aligned} Adv_{S,n,q}^{CCA2}(\mathcal{A}) &= Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1] - 1/2 \\ &\leq Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1 | \overline{Bad_1} \wedge \overline{Bad_2}] Pr[\overline{Bad_1} \wedge \overline{Bad_2}] + \\ &\quad Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1 | Bad_1] Pr[Bad_1] + \\ &\quad Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1 | Bad_2] Pr[Bad_2] - 1/2 \\ &\leq Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1 | \overline{Bad_1} \wedge \overline{Bad_2}] + Pr[Bad_1] + Pr[Bad_2] - 1/2 \\ &\leq 1/2 + q_e/\alpha + q_d/\alpha - 1/2 \\ &= q/\alpha \end{aligned}$$

where we assume no encoding (padding of  $P_1$  for decryption queries) collisions which accounts for the  $q/2^B$  term. In going from the idealized model to pseudorandom functions, the argument is similar to the proof of Theorem 4.1 and we pick up the corresponding terms.

**case ii:** the challenge ciphertext has  $\alpha \geq 2^B$ : Here we obtain an additional  $q/2^B$  term due to the fact that the pseudorandom permutation ( $\alpha = 2^B$ ) leaks information regarding which  $X$  values are no longer possible. We also have potential counter mode block collisions. In this case,  $|X| > B$ , counter block collisions are detectable, and  $\lceil\mu/2 + q\rceil^2/2^{B+2}$  is a bound on the probability of these collisions. ■

**Remark:** The above theorem is stated generally, and includes the case where the MAC function for computing  $V$  does not make use of the value  $W$  (stateless case) or  $M$  (stateful case). For our implementation of CMCC, the MAC is computed over the string  $W|Y$  (stateless case) and thus  $Bad_1$  in the above theorem cannot occur. The security bound is still unchanged though as discussed in the theorem proof. The authentication tag is either not present (zero length) or is present where invalid queries do not terminate the session.

**Theorem 4.3** *The adversary is restricted to  $q$  queries; given  $\alpha = 2^m$  where  $Len$  is the bit length of the challenge ciphertext and  $m = \lfloor Len/2 \rfloor$ .  $B$  is the block length and  $\mu$  is the total number of blocks in all the query plaintexts and ciphertexts. Let  $\beta = \min\{\alpha, 2^B\}$ . The stateless and stateful versions of CMCC where the authentication string  $Z$  has nonzero length  $\tau$  is CCA2 secure where the security bound is approximately*

$$\epsilon = 1/\beta + 2q/2^B + \chi(m)\lceil\mu/2 + q\rceil^2/2^{B+2} + \sum_{i=1}^3 Adv_{f_i}^{prf}(q) + Adv_{E_{\bar{K}}}^{prf}(q)$$

*if an invalid query results in session termination. Otherwise the security bound is*

$$\epsilon = \frac{q}{\beta} + 2q/2^B + \chi(m)\lceil\mu/2 + q\rceil^2/2^{B+2} + \sum_{i=1}^3 Adv_{f_i}^{prf}(q) + Adv_{E_{\bar{K}}}^{prf}(q).$$

*If we have CMCC with MAC (CWM) where the string  $Z$  is replaced with a keyed MAC computed over the plaintext and associated data, then the security bound (where invalid queries do not terminate the session) is*

$$\epsilon = \frac{q}{2^\tau\beta} + 2q/2^B + \chi(m)\lceil\mu/2 + q\rceil^2/2^{B+2} + \sum_{i=1}^3 Adv_{f_i}^{prf}(q) + Adv_{E_{\bar{K}}}^{prf}(q).$$

*For CWM where an invalid query terminates the session, the security bound is*

$$\epsilon = 2^{-\tau}/\beta + 2q/2^B + \chi(m)\lceil\mu/2 + q\rceil^2/2^{B+2} + \sum_{i=1}^3 Adv_{f_i}^{prf}(q) + Adv_{E_{\bar{K}}}^{prf}(q)$$

*Finally, if  $\alpha \leq 2^\tau$ , then the security bound is*

$$\epsilon = 2q/2^B + \chi(m)\lceil\mu/2 + q\rceil^2/2^{B+2} + \sum_{i=1}^3 Adv_{f_i}^{prf}(q) + Adv_{E_{\bar{K}}}^{prf}(q).$$

**Proof:** We first note that distinct  $W$  values (since message numbers are not reused) eliminates the possibility that  $W|Y$  values for distinct queries are equal. Thus the adversary's optimal strategy is to make ciphertext queries (except for the challenge plaintexts).

Suppose that an invalid query results in session termination. Then the security bound is

$$\epsilon = \frac{1}{\beta} \frac{1 - (2^{-\tau}(\beta - 1)/\beta)^q}{1 - (2^{-\tau}(\beta - 1)/\beta)} + q/2^B + \chi(m)\lceil\mu/2 + q\rceil^2/2^{B+2} + q/2^B + \sum_{i=1}^3 Adv_{f_i}^{prf}(q) + Adv_{E_{\bar{K}}}^{prf}(q)$$

where the first term represents the Adversary's probability of success for obtaining a ciphertext query based collision for  $P_1$  values (finite geometric series). The other terms are the same terms from the proof of Theorem 4.2. The first term is very close to  $1/\beta$ .

Thus the 2nd security bound is still the same as for Theorem 4.2. Then the security bound from Theorem 4.2 is modified to reflect that the ciphertext queries will be invalid, with high probability, for the third and fourth security bounds. The last security bound reflects that the ciphertext query attack cannot produce a valid ciphertext with matching  $X_2$ ,  $M$ , and  $P_1$  values, given the constraint.

■

**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.

**Theorem 4.4** *Let  $\mu$  be the total number of blocks in the adversary queries, and  $B$  is the cipher block length. Let  $\beta = \min\{\alpha, 2^B\}$ . CMCC encryption (stateless and stateful versions) is a TES with TES advantage  $\epsilon$  where*

$$\begin{aligned} \epsilon = & q(q-1)/2\alpha + q(q-1)/2\beta + q^2/2^{B+2} + \sum_{i=1}^3 Adv_{f_i}^{prf}(q) \\ & + Adv_{E_{\bar{K}}}^{prf}(q) + \chi(m)\mu^2/2^{B-1} + 2^{-\tau} + q^2/2^B \end{aligned}$$

*given that the adversary is restricted to  $q$  queries,  $\alpha = 2^m$  where  $Len$  is the bit length of the minimal length query response and  $m = \lfloor Len/2 \rfloor$ , and  $\tau$  is the number of bits in the authentication tag.<sup>3</sup>*

**Proof:** There are two types of relations that distinguish CMCC from a random permutation:

1. For messages shorter than the block length, we have the relation  $X_2 \oplus \bar{X}_2 = P_1 \oplus \bar{P}_1$  with higher probability equal to  $1/\alpha + (\alpha-1)/\alpha$  for CMCC versus  $1/\alpha$  for the random permutation. 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 permutation

Our strategy for handling these differences for the first relation above is that our security bound rules out  $X$  collisions. Thus the probability for the first relation is now the same as for a random permutation. For the second relation, our security bound rules out  $X_2$  collisions in the plaintext queries. Our security bound also rules out ciphertext queries.

The  $2^{-\tau}$  term reflects the fact that we assume there are no valid decryption oracle queries. We also bound the probability of  $X_2$  collisions between any of the queries ( $q(q-1)/2\beta$  term). The term  $q^2/2^{B+2}$  is a bound on the probability of padding encoding distinct plaintexts into the same input string to block cipher encryption.

We have the following bounds on the probability of collisions in the counter variant. Any collision can result in the adversary being able to distinguish CMCC from a random permutation.  $\mu^2/2^{B-1}$  is a bound on the probability of these collisions.

■

**Remark:** (i) We can replace the  $2^{-\tau}$  term in the above theorem with  $2^{-(\tau+\gamma)}$  where  $\gamma$  quantifies the number of higher level protocol check bits. (ii) We can eliminate the  $2^{-\tau}$  term if  $|P_2| \leq \tau$ . (iii) We can replace the  $2^{-\tau}$  term in the above theorem with  $2^{-2\tau}$  if we use CMCC with MAC (CWM). In this case,  $Z$  holds a MAC computed over the plaintext and associated data  $A$  (e.g., using CMAC) instead of a zero bit string.

**Remark:** Consider CCM or another counter mode variant with a 4 byte authentication tag. Then for the CCA2 security game, 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.

<sup>3</sup>Replace  $2^{-\tau}$  with  $x2^{-\tau}$  if up to  $x-1$  invalid ciphertexts do not result in session termination.

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 CCA2 security bound is approximately  $q(2^{-64})$  for a 12 byte message. Thus CMCC has much stronger CCA2 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. For a 2 byte authentication tag, the numbers would be  $q(2^{-16})$  and  $q(2^{-56})$ .

Table 2 gives the Theorem 4.3 bounds for varying message lengths and varying numbers of adversary queries to the oracles.

<i>plaintext length</i>	$\beta$	$q$	$\mu$	bound with invalid queries	bound without invalid queries	CWM bound with invalid queries	CWM bound without invalid queries
2 bytes	$2^{24}$	$2^{20}$	$2^{24}$	$2^{-107}$	$2^{-107}$	$2^{-107}$	$2^{-107}$
4 bytes	$2^{32}$	$2^{20}$	$2^{24}$	$2^{-107}$	$2^{-107}$	$2^{-107}$	$2^{-107}$
6 bytes	$2^{40}$	$2^{20}$	$2^{24}$	$2^{-20}$	$2^{-40}$	$2^{-52}$	$2^{-72}$
8 bytes	$2^{48}$	$2^{20}$	$2^{24}$	$2^{-28}$	$2^{-48}$	$2^{-60}$	$2^{-80}$
8 bytes	$2^{48}$	$2^{24}$	$2^{30}$	$2^{-24}$	$2^{-48}$	$2^{-56}$	$2^{-80}$
10 bytes	$2^{56}$	$2^{24}$	$2^{30}$	$2^{-32}$	$2^{-56}$	$2^{-64}$	$2^{-88}$
12 bytes	$2^{64}$	$2^{24}$	$2^{30}$	$2^{-40}$	$2^{-64}$	$2^{-72}$	$2^{-96}$
16 bytes	$2^{80}$	$2^{24}$	$2^{30}$	$2^{-56}$	$2^{-80}$	$2^{-88}$	$2^{-103}$
20 bytes	$2^{96}$	$2^{24}$	$2^{30}$	$2^{-72}$	$2^{-96}$	$2^{-103}$	$2^{-103}$

Table 2: Theorem 4.3 bounds for the adversary advantage given  $q$  queries for 2, 4, 6, 8, 10, 12, 16, and 20 byte messages, where  $\tau = 32$ . Security increases as message length increases (or if the length is less than or equal to  $\tau$ .) The security bound is approximately  $q/\beta$  when invalid queries are allowed,  $1/\beta$  when an invalid query terminates the session,  $q2^{-\tau}/\beta$  when invalid queries are allowed for CMCC with MAC (CWM), and  $2^{-\tau}/\beta$  for the smaller lengths when an invalid query terminates the session for CWM.

## 5 Performance Analysis for Wireless Sensor Networks

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

In [WanGurEblGupShtz], 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 [WanGurEblGupShtz] with respect to AES encryption, message transmission, and message receipt.

<i>Operation</i>	<i>Energy Utilization</i>
Energy to transmit one byte	59.2 $\mu J$
Energy to receive one byte	28.6 $\mu J$
Energy per byte of AES encryption including key setup, averaged over messages of 64-1024 bytes	1.6 $\mu J$

Table 3: Energy Utilization for Operations on the Mica2Dots Platform from [WanGurEblGupShtz]

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 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,<sup>4</sup> 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

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

---

<sup>4</sup>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.

<i>Message Length</i>	<i>No. CCM prf calls</i>	<i>No. CMCC prf calls</i>	<i>CCM energy use</i>	<i>CMCC energy use</i>
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 ( $\mu J$ ) for sending network messages with CCM and CMCC protection, Mica2dot platform.

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.

## 5.1 Implementation

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>.

## 6 Conclusions

We have presented CCS which is a new family of tweakable enciphering schemes (TES). 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). CMCC is an instantiation of the scheme providing misuse resistant authenticated encryption, and it leverages existing modes such as CBC, Counter, and CMAC. Our work can be viewed as extending the line of work starting with [HR03] on TES's to plaintext sizes smaller than the block cipher block length which is a problem posed in [Hal04]. Depending on the environment, we obtain CCA2 security with only 2-3 bytes of expansion (for the message number). 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.

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

- [AnBellr] An J. and Bellare M.: Does encryption with redundancy provide authenticity? *Advances in Cryptology Eurocrypt 01*, LNCS vol. 2045, Springer, pp. 512528, 2001.
- [Atknsn] Atkinson R.: IP Encapsulating Security Payload (ESP). RFC 1827 (1995).
- [BellrNamp] Bellare M. and Namprempre, C.: Authenticated encryption: relations among notions and analysis of the generic composition paradigm. *Advances in Cryptology Asiacrypt 00*, LNCS vol. 1976, Springer, pp. 531545, 2000.
- [Bellovin] Bellovin S.M.: Problem Areas for the IP Security Protocols. *Proceedings of the 6th USENIX Security Symposium* (1996).
- [Bernstein] Bernstein D.: The Poly1305-AES message-authentication code. *FSE 2005*, LNCS vol. 3557, Springer, pp. 3249, 2005.
- [Bormann] Bormann, C., Burmeister, C., Degermark, M., Fukuhsima, H., Hannu, H., Jonsson, L-E., Hakenberg, R., Koren, T., Le, K., Liu, Z., Martensson, A., Miyazaki, A., Svanbro, K., Wiebke, T., Yoshimura, T. and H. Zheng, RObust Header Compression: Framework and Four Profiles: RTP, UDP, ESP, and uncompressed (ROHC). RFC 3095, July 2001.
- [CS06a] Chakraborty, D. and Sarkar, P.: HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In *INDOCRYPT'06*, volume 4329 of Lecture Notes in Computer Science, pages 287–302. Springer, 2006.
- [CS06b] Chakraborty, D. and Sarkar, P.: A new mode of encryption providing a tweakable strong pseudo-random permutation. In *The 13th International Workshop on Fast Software Encryption FSE'06*, volume 4047 of Lecture Notes in Computer Science, pages 293–309. Springer, 2006.
- [cRTP] Casner, S., Jacobson, V.: Compressing IP/UDP/RTP Headers for Low-Speed Serial Links. RFC 2508, February 1999.
- [Desai] Desai A.: New Paradigms for Constructing Symmetric Encryption Schemes Secure Against Chosen-Ciphertext Attack. *CRYPTO 2000*: 394-412.
- [DolvDwkNaor] Dolev D., Dwork C., Naor M.: Non-malleable cryptography. *SIAM Journal on Computing*, 30(2):391-437, (2000).
- [CMAC] Dworkin, M.: NIST Special Publication 800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication.
- [FM04] Fluhrer, S., and McGrew D.: The extended codebook (XCB) mode of operation. Technical Report 2004/278, IACR ePrint archive, 2004. <http://eprint.iacr.org/2004/278/>.

- [GGM86] Goldreich, O., Goldwasser, S., Micali, S.: How to construct random functions. *Journal of the ACM*, 33:210-217, 1986.
- [Hal04] Halevi, S.: EME: extending EME to handle arbitrary-length messages with associated data. In *INDOCRYPT'04*, volume 3348 of LNCS, pages 315–327. Springer, 2004.
- [Hal07] Halevi, S.: Invertible Universal Hashing and the TET Encryption Mode. In *Advances in Cryptology CRYPTO '07*, 2007. Long version available on-line at <http://eprint.iacr.org/2007/014/>.
- [HR03] Halevi S. and Rogaway, P.: A tweakable enciphering mode. In D. Boneh, editor, *Advances in Cryptology CRYPTO '03*, volume 2729 of LNCS, pages 482–499. Springer, 2003.
- [HR04] Halevi S. and Rogaway, P.: A parallelizable enciphering mode. In *The RSA conference Cryptographer's track, RSA-CT'04*, volume 2964 of Lecture Notes in Computer Science, pages 292–304. Springer-Verlag, 2004.
- [Jutla] Jutla C.: Encryption modes with almost free message integrity. *Journal of Cryptology*, 21(4):547-578, 2008.
- [Katz-Yung1] Katz J. and M. Yung M.: Complete Characterization of Security Notions for Probabilistic Private Key Encryption. In *Proceedings of the 32nd Annual Symposium on Theory of Computing*, ACM 2000, pp. 245-254.
- [Katz-Yung2] Katz J. and M. Yung M.: Unforgeable encryption and chosen-ciphertext secure modes of operation. In *Fast Software Encryption - FSE 2000*, volume 1978 of *Lecture Notes in Computer Science*, pp. 284-299. Springer 2001.
- [LskvRvstWgnr] Liskov, M., Rivest, R. and Wagner D.: Tweakable block ciphers. In *Advances in Cryptology CRYPTO '02*, volume 2442 of Lecture Notes in Computer Science, pages 31–46. Springer, 2002.
- [McGrewViega] McGrew D. and Viega J.: The security and performance of the Galois/Counter Mode (GCM) of operation. *INDOCRYPT 2004*, LNCS vol. 3348, Springer, pp. 343-355, 2004.
- [NaorYung] Naor M. and Yung M.: Public-key cryptosystems provably secure against chosen ciphertext attacks. *Proceedings of the 22nd Annual Symposium on Theory of Computing*, ACM (1990), pp. 427-437.
- [NR] Naor, M., and Reingold, O.: On the construction of pseudorandom permutations: Luby-Rackoff revisited. *J. Cryptology*, 12(1):29–66, 1999.
- [AES] National Institute of Standard and Technology.: Specification for the Advanced Encryption Standard (AES). FIPS **197** (2001)
- [RogwyBellr] Rogaway P., Bellare M.: Encode-then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. in *AsiaCrypt00*, T. Okamoto, Ed., Lecture Notes in Computer Science, Vol. 1976, Springer, 2000.

- [RogwyShrmptn] Rogaway P., Shrimpton T.: Deterministic Authenticated-Encryption. *Advances in Cryptology – EUROCRYPT 06*, Lecture Notes in Computer Science, vol. 4004, Springer, 2006.
- [Sarkar] Sarkar, P.: Efficient Tweakable Enciphering Schemes from (Block-Wise) Universal Hash Functions. Technical Report 2008/004, IACR ePrint archive, 2008. <http://eprint.iacr.org/2008/004/>.
- [Shoup] Shoup. V.: Sequences of games: a tool for taming complexity in security proofs, manuscript, Nov. 30, 2004. Revised, May 27, 2005; Jan. 18, 2006. <http://www.shoup.net/papers/games.pdf>.
- [SongPoovnLeeIwata] J. Song, R. Poovendran, J. Lee, and T. Iwata.: The AES-CMAC Algorithm. RFC 4493 (June 2006).
- [SongPoovnLeeIwata] Song J., Poovendran R., Lee J., and Iwata T.: The Advanced Encryption Standard-Cipher-based Message Authentication Code Pseudo-Random Function-128 (AES-CMAC-PRF-128) Algorithm for the Internet Key Exchange Protocol (IKE) RFC 4615 (August 2006).
- [Struik] Struik, R.: Cryptography for Highly Constrained Networks. *NIST CETA Workshop 2011*, November 2011.
- [VuranAkyldz] Vuran, M., Akyildiz I.: Cross-layer Packet Size Optimization for Wireless Terrestrial, Underwater, and Underground Sensor Networks *Proceedings of IEEE Infocomm* 2008.
- [WanGurEblGupShtz] Wander A.S., Gura N., Eberle H., Gupta V., Shantz S. C.: Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks. *Third IEEE International Conference on Pervasive Computing and Communications, 2005 (PerCom 2005)*. pp. 324-328, March 2005
- [WFW05] Wang, P., Feng, D., and Wu, W.: HCTR: A variable-input-length enciphering mode. In *Information Security and Cryptology CISC'05*, volume 3822 of Lecture Notes in Computer Science, pages 175–188. Springer, 2005.
- [WhitHousFerg] Whiting D., Housley R., Ferguson, N.: Counter with CBC-MAC (CCM). RFC 3610 (2003).