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). Our work can be viewed as extending the line of work starting with [HR03] to plaintext sizes smaller than the block cipher block length which is a problem posed in [Hal04]. 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. 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 [Bellovin] 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 authenticated encryption mode, CMCC. CMCC utilizes a pseudo-random 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 [BellrNamp]. 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 and 2). 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. 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 [RogwyShrmptn], CMCC has smaller ciphertext expansion.

We also show that CMCC satisfies both MRAE and AE security with small concrete security bounds using only 2-3 bytes of ciphertext expansion.[1]

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

[1] An existing TES (Tweakable Enciphering Scheme) [16] 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).

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 [RogwyShrmptn].

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 [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 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 [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.

1.3 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 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 $q(q-1)2^{-2\tau} + 2e(q-1)/\beta$ where $\beta = min\{\alpha, 2^B\}$ 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 [RogwyShrmptn]. 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 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 [Atknsn]. However, [Bellovin] 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. 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 4.5 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 [RogwyShrmptn], 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. Our security definition is the same as the PRI security definition in [RogwyShrmptn].

CMCC uses the same authentication 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 σ 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]. Shrimpton and Terashima [ShrmptnTrshm] 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.)

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 then CMC and EME*. The EME ciphers are more parallelizable.

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

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. 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

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. 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 [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

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

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

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 $\mathcal{K}$, and message number set $\mathcal{N}$. An authenticated encryption scheme (AE) is a tuple $\Pi = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ such that $\mathcal{E} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{P} \rightarrow \mathcal{C}$, $\mathcal{D} : \mathcal{K} \times \mathcal{N} \times \mathcal{AD} \times \mathcal{C} \rightarrow \mathcal{P} \cup \{\perp\}$, and $\mathcal{D}(K, N, A, \mathcal{E}(K, N, A, P)) = P$ for all $N \in \mathcal{N}, A \in \mathcal{AD}$, $P \in \mathcal{P}$. If there is no $P \in \mathcal{P}$ such that $C = \mathcal{E}(K, N, A, P)$, then $\mathcal{D}(K, N, A, C) = \perp$. 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 $e : \mathcal{N} \times \mathcal{AD} \times \mathcal{P} \rightarrow \mathbb{N}$. The expansion function depends only on the length of its arguments. Let $Inj_e^{\mathcal{N},\mathcal{A}}(\mathcal{P}, \mathcal{C})$ be the set of injective functions f from $\mathcal{N} \times \mathcal{AD} \times \mathcal{P}$ into $\mathcal{C}$ such that $|f(N, A, P)| = |P| + e(N, A, P)$.

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

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

when encryption oracle queries use unique message numbers and $\mathcal{A}$ is restricted to asking q queries totaling μ blocks in running time t. $f^{-1}(N, A, C) = P$ if $f(N, A, P) = C$ and returns $\perp$ if no such triple (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.

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.

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. 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.

3.1.1 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. $|S|$ 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.

3.1.2 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 ($B/8$ 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,

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

   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

[2]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.

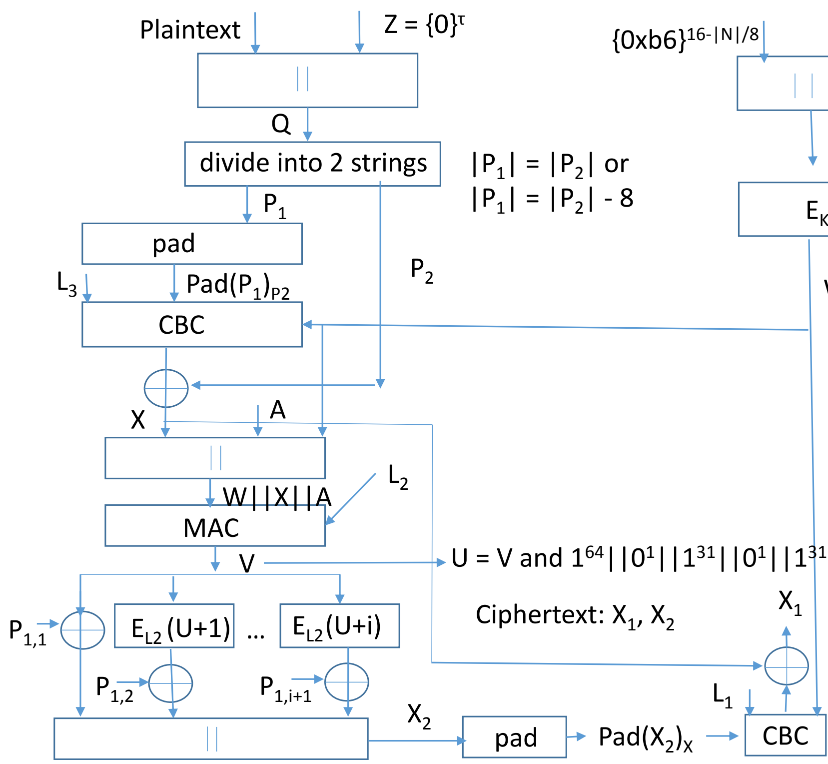

Figure 1: CMCC Stateless Encryption

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

Figure 2: CMCC Encryption: Encryption inputs are plaintext P, key $K = \bar{K}, L_3, L_2, L_1$, 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 [CMAC] 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. U is obtained from V by zeroing bits 31 and 63 to enable faster addition (prevent carries) [Hrkn]. $U+j$ is integer addition, $1 \leq j \leq i$. $E_{L_2}(U+i)$ is truncated to the length of $P_{1,i+1}$.

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

Figure 3: CMCC Decryption: Decryption inputs are ciphertext X_1X_2, key $K = \bar{K}, L_3, L_2, L_1$, 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 $Seq(x)$.

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.

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

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 4.2). 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 4.5.

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

Algorithm CMCC Stateful Encrypt(P, $\bar{K_1}, \bar{K_2}, L_3, L_2, L_1$, 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 $|\bar{P}_{1,j+1}| = |P_1|\ mod\ B$.
$U \leftarrow V\ and\ (1^{64}||0^1||1^{31}||0^1||1^{31})$
$X_2 \leftarrow V \oplus \bar{P}_{1,1}||E_{L_2}(U+1) \oplus \bar{P}_{1,2}|| \ldots ||E_{L_2}(U+j) \oplus \bar{P}_{1,j+1}$
$X_1 \leftarrow CBC(M, pad(X_2)_X, L_1) \oplus X$
$T = LSB_{IL}(M)$.

Figure 4: CMCC Stateful Encryption: Encryption inputs are plaintext P, key $K = \bar{K_1}, \bar{K_2}, L_3, L_2, L_1$, 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 [CMAC] 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. U is obtained from V by zeroing bits 31 and 63 to enable faster addition (prevent carries) [Hrkn]. $U+l$ is integer addition, $1 \leq l \leq j$. $E_{L_2}(U+j)$ is truncated to the length of $P_{1,j+1}$.

Algorithm CMCC Stateful Decrypt($X_1, X_2, \bar{K_1}, \bar{K_2}, L_3, L_2, L_1, T, A$)
$x \in \{init, resp\}$ 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
$Z \leftarrow 0^\tau$
$X \leftarrow CBC(M, pad(X_2)_X, L_1) \oplus X_1$
$Y \leftarrow X||A$
$V \leftarrow MAC(M||Y, L_2)$
$j \leftarrow \lfloor |X_2|/B \rfloor$
$X_2 = \bar{X}_{2,1}|| \ldots ||\bar{X}_{2,j}||\bar{X}_{2,j+1}$ where $|\bar{X}_{2,1}| = \ldots = |\bar{X}_{2,j}| = B$ and $|\bar{X}_{2,j+1}| = |X_2|\ mod\ B$.
$U \leftarrow V\ and\ (1^{64}||0^1||1^{31}||0^1||1^{31})$
$P_1 \leftarrow V \oplus \bar{X}_{2,1}||E_{L_2}(U+1) \oplus \bar{X}_{2,2}|| \ldots ||E_{L_2}(U+j) \oplus \bar{X}_{2,j+1}$
$P_2 \leftarrow CBC(M, pad(P_1)_X, L_3) \oplus X$
$Q = P_1||P_2$,
$U = LSB_{\tau/8}(Q)$
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 $K = \bar{K_1}, \bar{K_2}, L_3, L_2, L_1$, 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.

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.

We may select c such that

$$ct/2^B + q^2/2^{B+1} \leq Adv_E^{prf}(q,t) \leq ct/2^B + q^2/2^B$$

for the block cipher E.

Lemma 4.1 *Given the experiment S where adversary $\mathcal{A}$ attempts to distinguish between a block cipher E and a random function. We also consider the experiment S' : here an adversary $\mathcal{B}$ attempts to distinguish between a random function and a first instance of a block cipher for $q/2$ queries, and then between the random function and a 2nd instance of a block cipher for the remaining $q/2$ queries (the block cipher is randomly rekeyed for the last $q/2$ queries). $\mathcal{B}$ has an oracle for a random function for the first $q/2$ queries if and only if it again has an oracle for the random function during the 2nd set of $q/2$ queries. Then*

$$|Pr(\mathcal{B}^{random}(q,t)=1) - Pr(\mathcal{B}^{E_{K_1},E_{K_2}}(q,t)=1)| \leq Adv_E^{prf}(q,t)$$

Proof: For the first $q/2$ queries, we have

$$|Pr(\mathcal{B}^{random}(q/2,t_1)=1) - Pr(\mathcal{B}^{E_{K_1}}(q/2,t_1)=1)| \leq Adv_E^{prf}(q/2,t_1) \leq ct_1/2^B + q^2/2^{B+2}$$

Similarly,

$$|Pr(\mathcal{B}^{random}(q/2,t_2)=1) - Pr(\mathcal{B}^{E_{K_2}}(q/2,t_2)=1)| \leq Adv_E^{prf}(q/2,t_2) \leq ct_2/2^B + q^2/2^{B+2}$$

where $t = t_1 + t_2$. Also,

$$\begin{aligned} |Pr(\mathcal{B}^{random}(q,t)=1) - Pr(\mathcal{B}^{E_{K_1},E_{K_2}}(q,t)=1)| \leq \\ |Pr(\mathcal{B}^{random}(q/2,t_1)=1) - Pr(\mathcal{B}^{E_{K_1}}(q/2,t_1)=1)| + \\ |Pr(\mathcal{B}^{random}(q/2,t_2)=1) - Pr(\mathcal{B}^{E_{K_2}}(q/2,t_2)=1)| \leq \\ Adv_E^{prf}(q/2,t_1) + Adv_E^{prf}(q/2,t_2) \leq ct/2^B + q^2/2^{B+1} \leq Adv_E^{prf}(q,t). \end{aligned}$$

∎

Theorem 4.2 *Let μ be the maximum number of P_1 blocks in a query request or query response. B is the cipher block length. Let $\beta = min\{\alpha, 2^B\}$. Let the CMCC MAC function be CMAC [CMAC]. 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*

$$\begin{aligned} q(q-1)/2\alpha + q(q-1)/2\beta + 1 - (1 - 1/\beta - 2^{-\tau})^x + (5s^2+1)q^2/2^B + Adv_E^{prp}(sq+1, t+c_1sq) + \\ Adv_E^{prp}(2sq,t) + 2sq(2sq-1)/2^{B+1} + q(q-1)(\mu-1)^2/2^{B+1} + q^2/2^{B+1} + Adv_E^{prf}(q,t) \end{aligned}$$

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,* $m = \lfloor Len/2 \rfloor$*, 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 less than $2 * B - \tau$ bits: We use a games based proof to establish the bound claim for the theorem. Game G_0 is depicted in Figure 6. 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 $(5s^2 + 1)q^2/2^B + Adv_E^{prp}(sq + 1, t + c_1 sq)$ [IwataKrswa] 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 H where adversary $\mathcal{B}$ distinguishes between the following:

1. A random function

2. A block cipher E_{K_1} for $q/2$ queries and then the block cipher is rekeyed (E_{K_2}) for the 2nd set of $q/2$ queries. Denote this as $\mathcal{B}^{E_{K_1}, E_{K_2}}$. By Lemma 4.1, we have

$$|Pr(\mathcal{B}^{random}(q, t) = 1) - Pr(\mathcal{B}^{E_{K_1}, E_{K_2}}(q, t) = 1)| \leq Adv_E^{prf}(q, t)$$

Suppose D is a distinguisher such that D runs in time t, makes q queries, and will distinguish between the xor sum of two block cipher encryptions (each block cipher is independently keyed) and the xor sum of two random function invocations.

For an adversary $\mathcal{A}$ that attempts to distinguish between games G_1 and G_2, $\mathcal{A}$ can submit $X_1 \oplus P_2$ to D for each query that $\mathcal{A}$ makes. When $\mathcal{A}$ outputs a bit b, D outputs the same bit b. Then $\mathcal{A}'s$ probability of success is bounded by $D's$ probability of success.

Given the adversary $\mathcal{B}$ for the game H. At the end of the sequence of queries, $\mathcal{B}$ gives the exclusive or sum of queries i and $q/2+i$, $1 \leq i \leq q/2$ to D. D outputs a bit b and $\mathcal{B}$ outputs the same bit b. Then

$$\begin{aligned} |Pr(D^{random\ xor\ sum}(q/2,t)=1) - Pr(D^{block\ cipher\ xor\ sum}(q/2,t)=1)| \leq \\ |Pr(\mathcal{B}^{random}(q,t)=1) - Pr(\mathcal{B}^{E_{K_1},E_{K_2}}(q,t)=1)| \leq \\ Adv_E^{prf}(q,t). \end{aligned}$$

Thus we obtain

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

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 $\bar{X}_1, \bar{X}_2, \bar{N}$:

case a: The effect of previous queries where $\bar{N}$ matches the message number in these previous queries which have distinct X_2 values $(X_2 \neq \bar{X}_2)$.

Then the probability of a valid response is independent of these previous queries; $\bar{X}$ is uniform random since the block cipher has been replaced with a random function. Thus the value P_2 will be uniform random, and using only the information from these previous queries, the probability of a valid response is $2^{-\tau}$.

case b: The effect of previous queries where $\bar{N}$ is distinct from the message number in the previous queries but $\bar{X}_2$ matches their X_2 values:

The same argument as in case i applies; using only the information from these previous queries, the probability of a valid response is $2^{-\tau}$.

case c: The effect of i previous queries where $\bar{N}$ and $\bar{X}_2$ are distinct from the message numbers and X_2 values in the previous i queries:

For each previous query, the probability of a match on both of the inputs to the random functions for computing X and X_1 is 2^{-2B}. Thus the probability of a valid response, using only the information from these previous i queries, is bounded by $i/2^{2B} + 2^{-\tau}$.

case d: The effect of i previous queries where $\bar{N}$ and $\bar{X}_2$ are equal to the message number and X_2 values in all of the i previous queries:

If $\bar{P}_1$ matches P_1 from a previous query, we obtain $\bar{X}_1 \oplus X_1 = \bar{P}_2 \oplus P_2$. Thus the adversary can select $\bar{X}_1$ and P_2 such that the query $\bar{X}_1, \bar{X}_2, \bar{N}$ is a valid query. Then, using only the information from these i previous queries, the probability of a valid response is bounded by $i/\beta + 2^{-\tau}$.

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

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

Game G_5 is depicted in Figure 7. 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

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

Thus the bound claimed in the theorem statement holds.

case ii: At least some plaintexts have length greater than or equal to $2 * B - \tau$ bits:
We note that this case is a suboptimal strategy for the adversary. Here we modify the bound by adding in the $q(q-1)(\mu-1)^2/2^{B+1}$ and $q^2/2^{B+1}$ terms for counter mode block collisions and padding collisions for plaintexts of different lengths, respectively. The term $2q(2q-1)/2^{B+1}$ from above is generalized to $2sq(2sq-1)/2^{B+1}$. ∎

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 $|P_2| \leq \tau$.

We now prove a security bound for the CMCC stateless AEAD algorithm; here message numbers are not allowed to be repeated in plaintext queries.

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 2 for definition.
Decrypt(C, A, N): See Figure 3 for definition.
Output: Return the adversary's output.

Figure 6: CMCC MRAE proof Game G_0

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 7: CMCC MRAE proof Game G_5

Lemma 4.3 *Let $q - 1 \leq 2^\tau$. Given the adversary strategy in game G_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

$$
\begin{aligned}
&Pr[(1)\ in\ G_2\ with\ q\ queries] = Pr[(1)\ in\ G_2\ with\ first\ q-1\ queries] + \\
&Pr[not\ (1)\ in\ G_2\ with\ first\ q-1\ queries\ \cap (1)\ in\ G_2\ with\ qth\ query] \leq \\
&Pr[(1)\ in\ G_2\ with\ first\ q-1\ queries] + Pr[(1)\ in\ G_2\ with\ qth\ query] \leq \\
&(q-2)\sum_{i=0}^{q-3}\binom{q-3}{i}\lambda_1/2^{i\tau} + \lambda_1 + (1-\lambda_1)\left(\sum_{i=0}^{q-2}\binom{q-2}{i}2^{-i\tau}(1-2^{-\tau})^{q-2-i}i\lambda_1\right) < \\
&(q-2)\sum_{i=0}^{q-3}\binom{q-3}{i}\lambda_1/2^{i\tau} + \lambda_1 + \sum_{i=0}^{q-2}\binom{q-2}{i}i\lambda_1/2^{i\tau} = \\
&\sum_{i=0}^{q-3}\left(\binom{q-3}{i}(q-2)\lambda_1/2^{i\tau} + \binom{q-2}{i}i\lambda_1/2^{i\tau}\right) + \lambda_1 + (q-2)\lambda_1/2^{(q-2)\tau} = \\
&\sum_{i=0}^{q-3}\binom{q-2}{i}(q-2)\lambda_1/2^{i\tau} + (q-2)\lambda_1/2^{(q-2)\tau} + \lambda_1 = \\
&\lambda_1 + (q-2)\sum_{i=0}^{q-2}\binom{q-2}{i}\lambda_1/2^{i\tau} < \\
&(q-1)\sum_{i=0}^{q-2}\binom{q-2}{i}\lambda_1/2^{i\tau}.
\end{aligned}
$$

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 4.4 *Let $q - 1 \leq 2^{\tau}$. Given the adversary strategy in game G_5 above (except we are in the AE game where message numbers are not allowed to be repeated) 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 4.5 *Let μ be the maximum number of P_1 blocks in a query request or query response. B is the cipher block length. Let $\beta = min\{\alpha, 2^B\}$. Let the CMCC MAC function be CMAC [CMAC]. Let s be the maximum number of CMAC blocks in a query; c_1 is a constant. CMCC encryption (stateless version) is an authenticated encryption with associated data (AEAD) scheme with AE-advantage bounded by*

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

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 less than $2 * B - \tau$ bits:

Games G_0, G_1, and G_2 are identical to the ones in the proof of the MRAE case above, except that plaintext queries with repeated message numbers N are not allowed. Game G_3 is identical to game G_5 in Theorem 4.2 above, except that plaintext queries with repeated message numbers N are not allowed (see Figure 7.) For the transition from game G_2 to game G_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 G_2 and G_3 via (1):
case a: Here the adversary uses the strategy from Lemma 4.3: 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

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

where we have applied both Lemma 4.3 and Lemma 4.4 from above.

case b: Games G_2 and G_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 G_2 which gives a slightly higher probability for the relation $X_1 \oplus \bar{X}_1 = P_2 \oplus \bar{P}_2$ in G_2 versus G_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 G_2 and G_3.

We now check the adversary's optimal strategy to distinguish between G_2 and G_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 G_2 and G_3. The adversary advantage is bounded by $(1 - 1/\beta)q(q-1)2^{-2\tau-1}$.

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 G_2 and G_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 G_3 and G_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 G_3 and G_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 4.3 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 $CBC(W \oplus pad(X_2)_X)$ 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 G_2 and G_3.

case 2: At least some plaintexts have length greater than or equal to $2 * B - \tau$ bits:

The case with longer plaintexts/ciphertexts is similar to the Theorem 4.2 case ii above. We note that this case is a suboptimal strategy for the adversary. Here we modify the bound by adding in the $q(q-1)(\mu-1)^2/2^{B+1}$ and $q^2/2^{B+1}$ terms for counter mode block collisions and padding collisions for plaintexts/ciphertexts of different lengths, respectively. The term $2e(q-1)/\beta$ is generalized to $e(q-1)(2/\beta + (\mu-1)/2^{B+\tau-1})$.

∎

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 2 gives the results from [WanGurEblGupShtz] 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 2: 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 3 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 3 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,[3] and (3) Messages are less than 2^{16} bytes so CCM sends a 13 byte nonce

[3]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

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

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.

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 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). Our work can be viewed as

savings regardless of the size of the application layer messages.

extending the line of work starting with [HR03] to plaintext sizes smaller than the block cipher block length which is a problem posed in [Hal04]. Depending on the environment, we obtain security with only 2-3 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.

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.

Future work includes CWM (CMCC with MAC) that replaces the zero bit string Z in CMCC with a MAC computed over the plaintext.

References

[AnBellr] An, J., Bellare, M.: Does encryption with redundancy provide authenticity? In: Advances in Cryptology EUROCRYPT 2001, LNCS vol. 2045, pp. 512–528. Springer, Heidelberg (2001)

[Atknsn] Atkinson R.: IP Encapsulating Security Payload (ESP). RFC 1827 (1995).

[BellrNamp] Bellare, M., Namprempre, C.: Authenticated encryption: relations among notions and analysis of the generic composition paradigm. In: Advances in Cryptology ASIACRYPT 2000, LNCS vol. 1976, pp. 531–545. Springer, Heidelberg (2000)

[BellrRogwy] Bellare, M., Rogaway, P.: The Game-Playing Technique. Technical Report 2004/278, IACR ePrint archive, 2004.

[Bellovin] Bellovin, S.M.: Problem Areas for the IP Security Protocols. In: Proceedings of the 6th USENIX Security Symposium, (1996).

[Bernstein] Bernstein, D. J.: A short proof of the unpredictability of cipher block chaining. Document ID: 24120a1f8b92722b5e15fbb6a86521a0. URL: http://cr.yp.to/papers.htmleasycbc. Date: 2005.01.09.

[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., Sarkar, P.: HCH: A new tweakable enciphering scheme using the hash-encrypt-hash approach. In: INDOCRYPT'06, LNCS vol. 4329, pp. 287–302. Springer, Heidelberg (2006)

[CS06b] Chakraborty, D., 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, LNCS vol. 4047, pp. 293–309. Springer, Heidelberg (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. In: CRYPTO 2000: pp. 394–412, Springer, (2000)

[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., 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, LNCS vol. 3348, pp. 315–327. Springer, (2004)

[Hal07] Halevi, S.: Invertible Universal Hashing and the TET Encryption Mode. In: Advances in Cryptology CRYPTO 2007. Long version available on-line at http://eprint.iacr.org/2007/014/.

[HR03] Halevi, S., Rogaway, P.: A tweakable enciphering mode. In: Advances in Cryptology CRYPTO 2003, LNCS vol. 2729, pp. 482–499. Springer, Heidelberg (2003)

[HR04] Halevi, S., Rogaway, P.: A parallelizable enciphering mode. In: The RSA conference Cryptographer's track, RSA-CT'04, LNCS vol. 2964, pp. 292–304. Springer, Heidelberg (2004)

[Hrkn] Harkins, D: Synthetic Initialization Vector (SIV) Authenticated Encryption Using the Advanced Encryption Standard (AES). RFC 5297. October 2008.

[IwataKrswa] Iwata, T., Kurosawa, K.: OMAC: One-Key CBC MAC. In: Fast Software Encryption, FSE 2003, LNCS vol. 2887, pp. 129–153, Springer, Heidelberg (2003)

[Jutla] Jutla C.: Encryption modes with almost free message integrity. *Journal of Cryptology*, 21(4):547–578, (2008)

[Katz-Yung1] Katz, J., Yung, M.: Complete Characterization of Security Notions for Probabilistic Private Key Encryption. In: Proceedings of the 32nd Annual Symposium on Theory of Computing, pp. 245–254, ACM (2000)

[Katz-Yung2] Katz, J., Yung, M.: Unforgeable encryption and chosen-ciphertext secure modes of operation. In: Fast Software Encryption - FSE 2000, LNCS vol. 1978, pp. 284–299. Springer (2000)

[16] Liskov, M., Rivest, R., Wagner, D.: Tweakable block ciphers. In Advances in Cryptology CRYPTO 2002, LNCS vol. 2442, pp. 31–46, Springer, Heidelberg (2002)

[McGrewViega] McGrew D. and Viega J.: The security and performance of the Galois/Counter Mode (GCM) of operation. INDOCRYPT 2004, LNCS vol. 3348, pp. 343–355, Springer, (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*, pp. 427–437, ACM (1990)

[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)

[Ristpt] Ristenpart, T., Yilek, S.: The Mix-and-Cut Shuffle: Small-Domain Encryption Secure against N Queries. In: Advances in Cryptology CRYPTO 2013, pp. 392–409. Springer, Heidelberg (2013)

[RogwyBellr] Rogaway P., Bellare, M.: Encode-then-Encipher Encryption: How to Exploit Nonces or Redundancy in Plaintexts for Efficient Cryptography. In: Advances in Cryptology - ASIACRYPT 2000, LNCS vol. 1976, pp. 317–330. Springer, Heideberg (2000)

[RogwyShrmptn] Rogaway P., Shrimpton T.: A Provable-Security Treatment of the Key-Wrap Problem. In: Advances in Cryptology – EUROCRYPT 2006, LNCS vol. 4004, pp. 373–390, 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.

[ShrmptnTrshm] Shrimpton, T., Terashima, R. S.: A Modular Framework for Building Variable-Input-Length Tweakable Ciphers. In: Advances in Cryptology - ASIACRYPT 2013, LNCS vol. 8269, pp. 405–423. Springer, Heideberg (2013)

[SongPoovnLeeIwata] Song, J., Poovendran, R., Lee, J., Iwata, T.: The AES-CMAC Algorithm. RFC 4493 (June 2006).

[SongPoovnLeeIwata] Song J., Poovendran R., Lee J., 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 In: 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. In: Third IEEE International Conference on Pervasive Computing and Communications, 2005 (PerCom 2005), pp. 324–328, March 2005

[WFW05] Wang, P., Feng, D., Wu, W.: HCTR: A variable-input-length enciphering mode. In *Information Security and Cryptology CISC'05*, LNCS vol. 3822, pp. 175–188. Springer, (2005)

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