AES-CMCC v1

Designer: Jonathan Trostle

Submitter: Jonathan Trostle
jon49175@yahoo.com

2014.03.01

Chapter 1

Specification

1.1 Parameters

AES-CMCC v1 has the following parameters:

1. Stateful or stateless
2. Key size: 128 bits, 192 bits, or 256 bits.
3. Authentication Tag Length: 0 bytes up to 16 bytes.
4. Secret Message Number (SMN) length (stateful only): 16 bytes
5. Stateful scheme ciphertext expansion: 0-8 bytes
6. Public Message Number (PMN) length (stateless only: 0-16 bytes)
7. The MAC algorithm for computing V (see Section 1.3) can be any MAC algorithm with the standard MAC security property (forgery under an adaptive chosen message attack), except that if it takes a nonce as one of its input parameters, the MAC algorithm must be misuse resistant when a nonce is reused. [1]

Each parameter is an integer number of bytes.

1.2 Recommended Parameter Sets

All recommended parameter sets have 128 bit keys, and the MAC algorithm for computing V is AES-CMAC.[2] All stateful versions have a 16 byte SMN.

(1) Stateless: Authentication tag length 8 bytes, PMN length 4 bytes.
(2) Stateless: Authentication tag length 4 bytes, PMN length 4 bytes.
(3) Stateless: Authentication tag length 4 bytes, PMN length 2 bytes.

[1] This MAC algorithm is used as part of the encryption process and does not produce the authentication tag in the 3rd parameter above. The authentication tag is a string of zero bits.

[2] NIST Special Publication 800-38B Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication

(4) Stateless: Authentication tag length 2 bytes, PMN length 4 bytes.
(5) Stateless: Authentication tag length 2 bytes, PMN length 2 bytes.
(6) Stateful: Authentication tag length 4 bytes, IL length 2 bytes.
(7) Stateful: Authentication tag length 2 bytes, IL length 2 bytes.

1.3 Authenticated Encryption

1.3.1 CMCC Stateless Scheme

Figure 1.1 describes the stateless version of CMCC.

1.3.2 CMCC Stateful Scheme

$LSB_j(x)$ 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.

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 $L_1, \ldots, L_k$ determine the PRF's $f_1, \ldots, f_k$. The key $K = \bar{K}_1, \bar{K}_2, L_1, \ldots, L_k$. $E_{\bar{K}_i}$ is a permutation on the set of binary strings with l bits.

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.

Creating the Sequences of Secret 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 $Seq(x)$.

CMCC Mode

$CBC(IV, P, Key)$ is CBC encryption with initialization vector IV, plaintext P, and key Key. CBC padded with zero bytes if needed. B is the block length of the underlying block cipher.
$MAC(IV, P, Key)$ is MAC algorithm with output string of length $l/8$ bits (one block) with initialization vector IV, plaintext P, and key Key.
$E_{\bar{K}}$ is the block cipher with key $\bar{K}$.
Encryption Inputs: plaintext P, key K as above, public message number N, associated data A.
Given constant $0xb6b6b6b6b6b6b6b6b6b6b6b6b6b6b6b6$, 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 τ zero bits (τ 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.)
$X = CBC(W, P_1, L_3) \oplus P_2$ where P_1 is zero padded out to the length of P_2 and then zero padded up to the next block size multiple, if needed; X is truncated to the length of P_2.
$Y = X|A$
If $|Y| \leq B$ and $|P_1| \leq B$, then $X_2 = E_{L_2}(Y|\text{zero padding}) \oplus P_1$ else
$P_1 = \bar{P}_{1,1}| \ldots |\bar{P}_{1,i}|\bar{P}_{1,i+1}$ where $i \geq 0$, $\bar{P}_{1,1}, \ldots, \bar{P}_{1,i}$ are full blocks and $\bar{P}_{1,i+1}$ is a partial (possibly empty) block,
$V = MAC(W, Y, L_2)$,
$X_2 = V \oplus \bar{P}_{1,1}|E_{\bar{L}_2}(V + 1) \oplus \bar{P}_{1,2}| \ldots |E_{\bar{L}_2}(V + i) \oplus \bar{P}_{1,i+1}$.
($E_{\bar{L}_2}(V + j)$ is truncated to the length of $\bar{P}_{1,j+1}$ for $j \geq 1$, prior to $\oplus$.)
$X_1 = CBC(W, X_2, L_1) \oplus X$
(where X_2 is zero padded out to the length of X and then zero padded up to the next block size multiple, if needed; X_1 is truncated to the length of X.)
Ciphertext: X_1, X_2, M
Decryption Inputs: X_1, X_2, M, A
$W = E_{\bar{K}}(M)$.
$X = CBC(W, X_2, L_1) \oplus X_1$
$Y = X|A$. If $|Y| \leq B$ and $|X_2| \leq B$, then $P_1 = E_{L_2}(Y|\text{zero padding}) \oplus X_2$, else
$V = MAC(W, Y, L_2)$, $X_2 = \bar{X}_{2,1}| \ldots |\bar{X}_{2,i}|\bar{X}_{2,i+1}$ where $i \geq 0$ and $\bar{X}_{2,1}, \ldots, \bar{X}_{2,i}$ are full blocks and $\bar{X}_{2,i+1}$ is a partial empty block, $P_1 = V \oplus \bar{X}_{2,1}|E_{\bar{L}_2}(V + 1) \oplus \bar{X}_{2,2}| \ldots |E_{\bar{L}_2}(V + i) \oplus \bar{X}_{2,i+1}$
$P_2 = CBC(W, P_1, L_3) \oplus X$
$Q = P_1|P_2$
$U = LSB_{\tau/8}(Q)$
if ($U\ != Z$), return $\perp$, otherwise $Q = \tilde{P}|Z$ and return **Plaintext** $\tilde{P}, M$

Figure 1.1: CMCC Mode - Stateless Version

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, $X_1|\ldots|X_k|T$ 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 1.1 summarizes the parameters for the stateful scheme.

Parameter	*Description*
α	$\alpha = 2^{\lvert P_i \rvert}$, $i = 1, 2$.
M	per 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 1.1: Summary of Parameters for Stateful CCS Scheme

Figure 1.2 gives the stateful version.

Encryption Inputs: plaintext P, key K as above, private message number i, associated data A. CBC padded with zero bytes if needed. Z is the bit string with τ 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 $i = SeqNo_x[M]$.
$X = CBC(M, P_1, L_3) \oplus P_2$ where P_1 is zero padded out to the length of P_2 and then zero padded up to the next block size multiple, if needed; X is truncated to the length of P_2.
$Y = X|A$
If $|Y| \leq B$ and $|P_1| \leq B$, then $X_2 = E_{L_2}(Y|\text{zero padding}) \oplus P_1$ else
$P_1 = \bar{P}_{1,1}|\ldots|\bar{P}_{1,i}|\bar{P}_{1,i+1}$ where $i \geq 0$, $\bar{P}_{1,1}, \ldots, \bar{P}_{1,i}$ are full blocks and $\bar{P}_{1,i+1}$ is a partial (possibly empty) block,
$V = MAC(M, Y, L_2)$,
$X_2 = V \oplus \bar{P}_{1,1}|E_{\bar{L}_2}(V + 1) \oplus \bar{P}_{1,2}|\ldots|E_{\bar{L}_2}(V + i) \oplus \bar{P}_{1,i+1}$.
($E_{\bar{L}_2}(V + j)$ is truncated to the length of $\bar{P}_{1,j+1}$ for $j \geq 1$, prior to $\oplus$.)
$X_1 = CBC(M, X_2, L_1) \oplus X$
(where X_2 is zero padded out to the length of X and then zero padded up to the next block size multiple, if needed; 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 $y_d = SeqNo_x[M]$.
$X = CBC(M, X_2, L_1) \oplus X_1$
$Y = X|A$. If $|Y| \leq B$ and $|X_2| \leq B$, then $P_1 = E_{L_2}(Y|\text{zero padding}) \oplus X_2$, else
$V = MAC(M, Y, L_2)$, $X_2 = \bar{X}_{2,1}|\ldots|\bar{X}_{2,i}|\bar{X}_{2,i+1}$ where $i \geq 0$ and $\bar{X}_{2,1}, \ldots, \bar{X}_{2,i}$ are full blocks and $\bar{X}_{2,i+1}$ is a partial empty block, $P_1 = V \oplus \bar{X}_{2,1}|E_{\bar{L}_2}(V + 1) \oplus \bar{X}_{2,2}|\ldots|E_{\bar{L}_2}(V + i) \oplus \bar{X}_{2,i+1}$
$P_2 = CBC(M, P_1, L_3) \oplus X$
$Q = P_1|P_2$
$U = LSB_{\tau/8}(Q)$
if (U ! = Z), return $\perp$, otherwise $Q = \tilde{P}|Z$ and return **Plaintext** $\tilde{P}, i = SeqNo_x(M)$

Figure 1.2: CMCC Mode - Stateful Version

Chapter 2

Security Goals

goal	aes-cmcc v1, 8 byte tag	aes-cmcc v1 4 byte tag	aes-cmcc v1, 2 byte tag
confidentiality for plaintext	128	128	128
integrity for plaintext	64	32	16
integrity for Assoc. Data	64	32	16
integrity for PMN	64	32	16

Table 2.1: Security goals: recommended parameters for stateless

Table 2.1 gives the security strengths for the recommended parameters for the stateless case, but the integrity strengths do not include higher layer checks that act as authentication bits. These checks are protocol specific. The numbers in the table do not account for the case where the attacker is able to obtain messages encrypted under the key - see below for bounds for this latter case.

The stateless case does not include a Secret Message Number (SMN). Although longer key lengths are possible (192 and 256 bits), the recommended key size for these parameter sets is 128 bits.

goal	aes-cmcc v1, stateful, 4 byte tag	aes-cmcc v1, stateful, 2 byte tag
confidentiality for plaintext	128	128
confidentiality for SMN	112	112
integrity for plaintext	32	16
integrity for Assoc. Data	32	16
integrity for SMN	32	16

Table 2.2: Security goals: recommended parameters for stateful

Table 2.1 gives the security strengths for the recommended parameters for the stateful case, but the integrity strengths do not include higher layer checks that act as authentication bits. These checks are protocol specific. The stateful case does not include a Public Message Number (PMN) (the PMN has zero bytes). Although longer key lengths are possible (192 and 256 bits), the recommended key size for these parameter sets is 128 bits.

2.1 Additional Security Goals

In addition to authenticated encryption, CMCC has the following security goals:

1. The cipher is designed to provide the maximum possible robustness against message-number reuse, i.e., that the cipher maintains full integrity and confidentiality, except for leaking collisions of (plaintext,associated data,secret message number,public message number) via collisions of ciphertexts.

2. Ciphertext modification results in unpredictable changes to the plaintext; thus
   (a) modifications to a ciphertext will like cause a failure in higher level processing (resulting in session termination most likely)
   (b) data that is consumed immediately will be randomized and thus anomalous to the consuming agent, again causing alerts and/or session termination.

   The implication of these properties is that, for many applications, the number of authentication bits that are part of the ciphertext can be reduced. The benefit is reduced network overhead.

3. Stateful version: private message numbers will hide the number of messages previously sent.

4. Stateful version: replay protection can be enforced by the receiver.

2.1.1 Resistance to Additional Specific Attacks

Here we discuss additional security features.

1. Large number of legitimate messages encrypted, ciphertexts decrypted: Let $\mathcal{M}$ be a bound on the maximum number of blocks in a query, and B is the cipher block length. The adversary submits q_i queries where the queried plaintext or ciphertext has length l_i, $1 \leq i \leq v$, and $\sum_{i=1}^{v} q_i \leq q$. CMCC encryption (stateless and stateful versions) is CCA2 MRAE secure for (ϵ, q) with

   $$\epsilon = (1 - e^{\sum_{i=1}^{v}(-(q_i-1)q_i)/2^{l_i+1}}) + q(q-1)/\alpha + \sum_{i=1}^{3} Adv_{f_i}^{prf}(q) + Adv_{E_{\bar{K}}}^{prf}(q) + 2q\mathcal{M}/2^B + \max\{q\mathcal{M}^2/2^B, z/\alpha + \mathcal{M}(q-z)/2^B\} + 2^{-2\tau}$$

   given that the adversary is restricted to q queries, $\alpha = 2^m$ where m is the minimal bit size for the adversary queries, τ is the number of bits in the authentication tag, and $z \leq q$.

2. Relationships among keys and related key attacks: There are no key relationships (the keys are independent) and related key attacks are not a threat.

3. Software and Hardware Side Channels: Note that encryption and decryption are both performed using AES encryption (not decryption) and that padding is never checked (both

encryptor and decryptor will pad but neither will verify any padding). So there is no padding oracle. The adversary cannot manipulate ciphertext to produce specific plaintext relations or patterns so the adversary cannot learn information about the plaintext or keys from integrity failures.

Specific implementations may be vulnerable to side channels based on observable differences arising from distinct keys or plaintexts. Thus developers should consider methods for minimizing side channels in implementations.

Chapter 3

Security Analysis

3.1 Definitions

3.1.1 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}.

3.1.2 (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 (ϵ, 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 $Adv^{CCA2}_{S,n,q}(\mathcal{A}) = [Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1] - 1/2$.

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 (ϵ, 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 $Adv^{CCA2\_MRAE}_{S,n,q}(\mathcal{A}) = [Pr[Exp_{CCA2\_MRAE}(S, n, q, \mathcal{A}) = 1] - 1/2$.

3.1.3 (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 (ϵ, q). We have the analogous definitions for $Adv^{CPA}_{S,n,q}(\mathcal{A})$ and $Adv^{CPA\_MRAE}_{S,n,q}(\mathcal{A})$.

3.2 Proof of CCA2 Security

We prove security for the generalized scheme first (CCS):

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

where the ciphertext is X_1, X_2, together with M, a public message number and the f_i are pseudorandom functions. For maximum security, M is unique, with high probability, for each message encrypted under a given key K.

We will first prove CCA security for the stateless version of CCS. We will then show how to extend this proof to the stateful CCS scheme defined above.

We now prove that our scheme is CPA-secure.

Theorem 3.2.1 *The CCS encryption presented in the previous section is CPA MRAE secure for* (ϵ, 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* m *is the minimal bit size for the adversary queries.*

Proof: 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 ith ciphertext collides with the X value for one of the first $i-1$ ciphertexts. Thus

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

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 the probability of collisions in the X_2 values.

Thus the ciphertexts give no information about the X values.

We have

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

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 $\mathcal{A}$. ∎

We now prove that CCS is CCA2-secure. Our proof strategy is as follows. We will construct a challenger B that invokes the adversary $\mathcal{A}$ and answers $\mathcal{A}$'s decryption queries with uniformly random plaintexts. We will show that with high probability, $\mathcal{A}$ can't distinguish between the game without B and when being run by B. In other words, the probability distributions on outputs from B and the decryption oracle are indistinguishable with high probability. Thus $\mathcal{A}$'s probability of success will be the same as in the CPA game, after accounting for indistinguishability and collisions.

Theorem 3.2.2 *The adversary submits q_i queries where the queried plaintext or ciphertext has length l_i, $1 \leq i \leq v$, and $\sum_{i=1}^{v} q_i \leq q$. The CCS encryption presented in the previous section is CCA2 secure for (ϵ, q) with*

$$\epsilon = (1 - e^{\sum_{i=1}^{v}(-(q_i-1)q_i)/2^{l_i+1}}) + q(q-1)/\alpha + \sum_{i=1}^{k} Adv_{f_i}^{prf}(q)$$

given that the adversary is restricted to q queries and given $\alpha = 2^m$ where m is the minimal bit size over the adversary queries.

Proof: Challenger B invokes the adversary $\mathcal{A}$ for the CCA2 security game. B answers $\mathcal{A}$'s queries as follows:

1. If $\mathcal{A}$ makes an encryption oracle query, B transmits the query to the encryption oracle and returns the answer to $\mathcal{A}$. B records the plaintext ciphertext pair, $(P, (C, M))$.
2. If $\mathcal{A}$ makes a decryption oracle query, B checks the existing list of plaintext ciphertext pairs, and if the query ciphertext is present on the list, it returns the corresponding plaintext. Otherwise, B generates a random, uniformly distributed plaintext and returns that to $\mathcal{A}$. B records the plaintext ciphertext pair.

If $\mathcal{A}$ submits (C_1, M) and (C_2, M), $C_1 \neq C_2$ on different queries, (or $\mathcal{A}$ receives (C_1, M) and submits (C_2, M) where the plaintexts are identical), then there is a small probability that the returned plaintexts are identical. In other words, a collision has occurred. In this case, $\mathcal{A}$ wins the game since the two encryptions aren't both possible ($\mathcal{A}$ can distinguish between the Challenger B game and the real decryption oracle with probability equal to 1).

The probability of no collision is at least

$$\begin{aligned} p &= \frac{2^{l_i}-1}{2^{l_i}}\frac{2^{l_i}-2}{2^{l_i}}\cdots\frac{2^{l_i}-(q-1)}{2^{l_i}} \\ &= (1-1/2^{l_i})(1-2/2^{l_i})\dots(1-(q-1)/2^{l_i}) \\ &\approx e^{-1/2^{l_i}}e^{-2/2^{l_i}}\dots e^{-(q-1)/2^{l_i}} \\ &= e^{(-(q-1)q)/2^{l_i+1}} \end{aligned}$$

where the plaintexts are of length l_i. Thus the probability of a collision, over all the queries, is bounded by $1 - e^{\sum_{i=1}^{v}(-(q_i-1)q_i)/2^{l_i+1}}$. (As an aside: in general this term will be less than $q(q-1)/2^{l+1}$ where l is the minimum length of query strings submitted. Thus this term won't contribute significantly to the bound.)

We now show that the Challenger B game is indistinguishable from the uniform distribution, except with small probability.

We will assume initially that f_1 and f_2 are random functions. Since f_1 is a random function it follows using the same argument as in the CPA security proof that the X values are distinct with high probability: For $2 \leq i \leq q$, $(i-1)/\alpha$ is an upper bound on the probability that the X value for the *ith* 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.

Given distinct X values except for the small probability above, then since f_2 is random, it follows that P_1 is uniformly distributed. Thus P_2 is also uniformly distributed. Therefore, it follows that the game with Challenger B is indistinguishable from the game without Challenger B (where the adversary interacts with the real decryption oracle). Thus we have reduced the adversary's probability of success in the CCA2 security game to the probability of success in the CPA MRAE security game. We can replace f_1 and f_2 with pseudorandom functions and the proof then follows the prf argument in the CPA security theorem. Thus the claim for CCA2 security follows.

∎

Theorem 3.2.3 *Given the parameters defined in Theorem 3.2.2. The CCS stateful encryption scheme presented in Section 1.3.2 is CCA2 secure for (ϵ, q) with*

$$\epsilon = (1 - e^{\sum_{i=1}^{v}(-(q_i-1)q_i)/2^{l_i+1}}) + q(q-1)/\alpha + \sum_{i=1}^{k} Adv_{f_i}^{prf}(q)$$

given that the adversary is restricted to q queries and given $\alpha = 2^m$ where m is the minimal bit size for the adversary queries.

Proof: The challenger B can utilize the encryption oracle and maintain state for the stateful scheme. The adversary strategy is now a subset of the possible strategies in the stateless case, so the theorem follows. ∎

3.3 CMCC Proof

Although our emphasis has been on utilizing CCS to protect short messages in energy constrained environments, we now discuss CBC-MAC-Counter-CBC (CMCC) mode security. CMCC is a general purpose authenticated encryption mode 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. 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 from the message number M. This prevents the adversary from being able manipulate M and P_1 in a way that allows collisions in X values to be created. Then we apply a MAC algorithm to 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 1.3.1.

Figure 1.1 describes the stateless version of CMCC, and Figure 1.2 gives the stateful version. The proof of CCA MRAE security for CMCC follows the proof of Theorem 3.2.2, except we depend on the uniform randomness of $E_{\bar{K}}(M)$ which gives rise to the additional term $Adv^{prf}_{E_{\bar{K}}}(q)$, and the factors that arise due to the probability of collisions in counter and CBC modes. Additionally, if the adversary can submit multiple ciphertext and plaintext queries with the same message number, it can cause collisions in X values and obtain additional advantage. Thus our proof of CCA MRAE security will depend on the authentication tag as defined in Figure 1.1 and Figure 1.2 to defend against arbitrary ciphertext queries. We now state the CCA security theorem.

Theorem 3.3.1 *Let $\mathcal{M}$ be a bound on the maximum number of blocks in a query, and B is the cipher block length. The adversary submits q_i queries where the queried plaintext or ciphertext has length l_i, $1 \le i \le v$, and $\sum_{i=1}^{v} q_i \le q$. CMCC encryption (stateless and stateful versions) is CCA2 MRAE secure for (ϵ, q) with*

$$\epsilon = (1 - e^{\sum_{i=1}^{v}(-(q_i-1)q_i)/2^{l_i+1}}) + q(q-1)/\alpha + \sum_{i=1}^{3} Adv^{prf}_{f_i}(q) + Adv^{prf}_{E_{\bar{K}}}(q) + \\ 2q\mathcal{M}/2^B + \max\{q\mathcal{M}^2/2^B, z/\alpha + \mathcal{M}(q-z)/2^B\} + 2^{-2\tau}$$

given that the adversary is restricted to q queries, $\alpha = 2^m$ where m is the minimal bit size for the adversary queries, τ is the number of bits in the authentication tag, and $z \le q$.

Proof: The proof follows the proof of Theorem 3.2.2, except we have the following modifications in order to bound the probability of collisions in the counter variant and CBC modes. We first consider the counter-mode variant.

case i: Suppose the challenge ciphertext has $P_1 \le B$ and $|Y| \le B$ (short length plaintext). Then queries with $|P_1| \le B$ and $|Y| \le B$ can't match with the keystream block for the challenge ciphertext since the X values are distinct. Now consider queries where $|P_1| > B$ or $|Y| > B$. Then a collision may occur; they occur with probability 2^{-B}. Thus $\mathcal{M}q/2^B$ is a bound on the probability of collision across the q queries.

case ii: The challenge ciphertext has $|P_1| > B$ or $|Y| > B$. Then queries would have to result in matches on B bits in order to have collisions in counter mode keystream blocks; the probability of a collision is bounded by $2q\mathcal{M}/2^B$.

We now consider the CBC construction for X_1. To ensure security, we require that the input blocks for the challenge ciphertext be distinct from all of the input blocks for all of the queries (for the X_1 calculation.)

case i: $|X_2| < B$ for the challenge ciphertext $|X_2|$.

Then z queries with the same length X_2 values will result in a z/α probability of collision ($z \leq q$). The other queries will need to match on B bits and we have a $\mathcal{M}(q-z)/2^B$ bound on probability of collision. Thus the probability of collision is bounded by $z/\alpha + \mathcal{M}(q-z)/2^B$.
case ii: $|X_2| \geq B$ for the challenge ciphertext X_2 value.
Then we obtain a $\mathcal{M}^2 q/2^B$ bound if we want to bound any collisions between any 2 input blocks. Thus the total bound is the maximum of the two bounds for the CBC cases summed with the bound above.

We add the bound from the probability of submitting a valid ciphertext that was not obtained from the encryption oracle and manipulating plaintexts such that the authentication tag bit positions remain zero bits. ∎

Chapter 4

Features

Security features are covered in Section 2. For completeness, we list the security features again:

1. The cipher is designed to provide the maximum possible robustness against message-number reuse, i.e., that the cipher maintains full integrity and confidentiality, except for leaking collisions of (plaintext,associated data,secret message number,public message number) via collisions of ciphertexts.

2. Ciphertext modification results in unpredictable changes to the plaintext; thus
   (a) modifications to a ciphertext will like cause a failure in higher level processing (resulting in session termination most likely)
   (b) data that is consumed immediately will be randomized and thus anomalous to the consuming agent, again causing alerts and/or session termination.

3. Stateful version: private message numbers will hide the number of messages previously sent.

4. Stateful version: replay protection can be enforced by the receiver.

We now discuss the main performance feature of CMCC.

4.1 Reduced Message Expansion

A key performance feature is the reduction in message expansion that CMCC offers. In energy constrained environments, the number of bytes that are sent and received are a major influence on energy consumption. Thus it is important to minimize the ciphertext expansion (including padding, authentication bits) as well as the message number sizes that are transmitted over the network in these environments.

From Section 1.2, the recommended parameters sets for CMCC have between 2 and 12 bytes of expansion. We note that the smaller number of bits for message numbers is complemented by the misuse resistance property in the sense that if a message number is reused, then the security impact is minimized.

CMCC compares favorably with existing ciphers for reducing energy consumption due to the reduced message expansion.

4.2 Precomputation of the Message Numbers/Encrypted Message Numbers and Reduced Number of Block Cipher Calls

Although not as computationally efficient as some one-pass algorithms such as OCB, CMCC does allow for some improved computational efficiencies compared with existing two-pass algorithms, especially for shorter plaintexts.

If precomputation can be leveraged, then the number of block cipher calls is reduced by one in the stateless case. Also, the stateful scheme leverages precomputation. Table 4.1 compares the number of block cipher calls for CMCC and CCM for various length plaintexts, assuming no precomputation.

4.3 Comparison with AES-GCM

Compared to GCM [McGrewViega], CMCC imposes less message expansion and is also misuse resistant. If GCM uses an 8 byte MAC with a 12 byte nonce, the total expansion is 20 bytes. CMCC with 4 bytes of authentication and a 4 byte PMN has 8 bytes of expansion in comparison. Although it is arguable whether these are equivalent security levels, CMCC does not require the same authentication overhead as GCM for some protocols since changes to a ciphertext creates a randomized plaintext. Since GCM uses counter mode, changes to a GCM ciphertext create a predictable plaintext.

CMCC can leverage associated data for overcoming message number repetitions. This factor combined with misuse resistance implies that there is more safety margin with a shorter message number. CMCC is targeted for lower bandwidth, energy constrained environments which also fits with using shorter message numbers.

GCM is well suited for high bandwidth networks where parallelism and leveraging precomputation are important. In this sense, GCM and CMCC can be viewed as tools for distinct application scenarios.

4.4 Comparison with AES-CCM

Compared with CCM [WhitHousFerg], CMCC is more computationally efficient for messages with 32 or fewer bytes (see Table 4.1), and is misuse resistant. In addition, CMCC has less message expansion. CMCC (RFC 3610) can use a nonce as small as 7 bytes; if we assume an 8 byte MAC, then the overhead is 15 bytes. CMCC with 4 bytes of authentication and a 4 byte PMN has 8 bytes of expansion.

Although it is arguable whether these are equivalent security levels, CMCC does not require the same authentication overhead as CCM for some protocols since changes to a ciphertext creates a randomized plaintext. Since CCM uses counter mode, changes to a CCM ciphertext create a predictable plaintext.

With precomputation, CCM is more computationally efficient for messages with more than 32 bytes whereas CMCC is more efficient for shorter ($\leq$ 32 byte) messages.

Message Length	*No. CCM block cipher calls*	*No. CMCC block cipher calls*
8 bytes	4	4
16 bytes	4	4
20 bytes	6	4
24 bytes	6	4
32 bytes	6	4
48 bytes	8	8
64 bytes	10	8
80 bytes	12	12
128 bytes	18	16

Table 4.1: Comparing Number of Block Cipher Calls for AES-CMCC and AES-CCM for Different Length Plaintexts

4.5 Comparison with AES-OCB

AES-OCB [KrovtzRogwy] is a highly computationally efficient one pass AE algorithm. Compared to OCB, CMCC provides misuse resistance and has less message expansion. CMCC is not as computationally efficient as OCB for longer messages.

4.6 Comparison with AES-SIV

SIV is a misuse resistant AE algorithm invented by Rogaway and Shrimpton [RogwyShrmptn]. SIV (as specified in RFC 5297) has a similar number of block cipher calls as CCM (see Table 4.1. Thus CMCC has fewer block cipher calls for some message lengths and this can be further reduced by one block cipher call if precomputation is available. SIV has the 16 byte IV overhead together with the nonce overhead. If similar size nonces are used, and CMCC uses 8 bytes of authentication, then CMCC has 8 bytes less message expansion. However, the probability of an authentication forgery would be lower with SIV, but for many protocols, 8 bytes of authentication is sufficient.

CMCC is also misuse resistant.

4.7 Comparison with PTE

PTE is another misuse resistant approach from [RogwyShrmptn]. It utilizes a TES (Tweakable Enciphering Scheme) [LskvRvstWgnr, HR03] combined with authentication consisting of padding with zero bytes. CMCC uses the same approach. However, the TES that [RogwyShrmptn] proposes does not accomodate plaintexts smaller than the block cipher length, unless padding is used. Thus CMCC has smaller expansion for smaller than blocklength messages.

4.8 Applications

CMCC is well suited for applications that have one or more of the following properties:

1. Protocols encapsulated with the ciphertext (higher layer protocols) have control fields that act as authentication bits given the randomizing that occurs if a ciphertext is modified.

2. A single randomized plaintext will either have minimal effect on the session (e.g., VoIP) or will result in the termination of the session due to failing a protocol check.

3. Due to limitations of the network environment, it is difficult for an adversary to generate a large number of queries.

These applications can obtain reasonable security with smaller additional authentication overhead and can also function with smaller message numbers. CMCC is well-suited for VoIP and wireless sensor networks.

4.9 Justification for Recommended Parameter Sets

4.9.1 Stateless: Authentication tag length 8 bytes, PMN length 4 bytes

The first recommended parameter set is the default recommendation. In particular, for environments where the set of applications that will be used is not fully understood, then this first parameter set should be used. It offers the most authentication security given the 8 bytes of authentication it provides. Thus it provides security for a wider set of applications. The protocols that are included in the ciphertext (higher layer protocols) will, in most cases, still provide additional authentication bits given the protocol specific checks that occur.

The 4 byte PMN size implies that the PMN will cycle after 2^{32} plaintexts which is a bound on the number of plaintexts encrypted under a single key unless distinct associated data is present for all messages in which case cycling on the PMN does not affect security.

This parameter set is suitable for plaintexts of all lengths.

4.9.2 Stateless: Authentication tag length 4 bytes, PMN length 4 bytes

The second recommended parameter set can be used for environments where less authentication security is needed and message compactness is a higher priority. For example, environments where an adversary can make a smaller number of queries for a single key (e.g., less network bandwidth), or where higher layer protocols provide sufficient additional authentication bits are appropriate.

The 4 byte PMN size implies that the PMN will cycle after 2^{32} plaintexts which is a bound on the number of plaintexts encrypted under a single key unless distinct associated data is present for all messages in which case cycling on the PMN does not affect security.

This parameter set is suitable for plaintexts of all lengths.

4.9.3 Stateless: Authentication tag length 4 bytes, PMN length 2 bytes

This parameter set can be used for environments where less authentication security is needed and message compactness is a higher priority. For example, environments where an adversary can make a smaller number of queries for a single key (e.g., less network bandwidth), or where higher layer protocols provide sufficient additional authentication bits are appropriate.

The 2 byte PMN size implies that the PMN will cycle after 2^{16} plaintexts which is a bound on the number of plaintexts encrypted under a single key unless distinct associated data is present for all messages in which case cycling on the PMN does not affect security.

This parameter set is suitable for plaintexts of all lengths.

4.9.4 Stateless: Authentication tag length 2 bytes, PMN length 4 bytes

Usage of this parameter set requires a precise understanding of the higher layer protocols. In this case, security depends to a greater degree on the authentication bits that are provided through the higher layer protocol checks that occur (since a modified ciphertext will randomize the plaintext bits), and the degree of impact that will result due to a successful forgery. Some higher level protocols may not provide enough authentication bits, or the impact of an authentication forgery may be too high, which would make this parameter set an inappropriate choice.

The 4 byte PMN size implies that the PMN will cycle after 2^{32} plaintexts which is a bound on the number of plaintexts encrypted under a single key unless distinct associated data is present for all messages in which case cycling on the PMN does not affect security.

This parameter set is suitable for plaintexts with lengths that are 4 bytes or longer.

4.9.5 Stateless: Authentication tag length 2 bytes, PMN length 2 bytes

Usage of this parameter set requires a precise understanding of the higher layer protocols. In this case, security depends to a greater degree on the authentication bits that are provided through the higher layer protocol checks that occur (since a modified ciphertext will randomize the plaintext bits), and the degree of impact that will result due to a successful forgery. Some higher level protocols may not provide enough authentication bits, or the impact of an authentication forgery may be too high, which would make this parameter set an inappropriate choice.

Reduced PMN size makes sense when either a reduced number of messages (up to 2^{16} for the 2 byte PMN) will be encrypted under a single key, or distinct associated data is present for all messages in which case cycling on the PMN does not affect security.

This parameter set is suitable for plaintexts with lengths that are 4 bytes or longer.

4.9.6 Stateful: Authentication tag length 4 bytes, IL length 2 bytes

The stateful version provides private message numbers which enables hiding the number of messages previously sent. It also provides replay protection. Relative to the stateless version, there is no limit on the number of plaintexts that can be encrypted (but this number is still bounded by the AES limit) due to the IL value. However, $IL = 2$ limits the maximum difference in out of sequence message numbers to 128. Security is maintained if senders select SMN out of sequence, per the requirements in the Caesar final call, but replay protection may cause messages to be rejected.

This parameter set can be used for environments where less authentication security is needed and message compactness is a higher priority. For example, environments where an adversary can make a smaller number of queries for a single key (e.g., less network bandwidth), or where higher layer protocols provide sufficient additional authentication bits are appropriate.

This parameter set is suitable for plaintexts of all lengths.

4.9.7 Stateful: Authentication tag length 2 bytes, IL length 2 bytes

The stateful version provides private message numbers which enables hiding the number of messages previously sent. It also provides replay protection. Relative to the stateless version, there is no limit on the number of plaintexts that can be encrypted (but this number is still bounded by the

AES limit) due to the IL value. However, $IL = 2$ limits the maximum difference in out of sequence message numbers to 128. Security is maintained if senders select SMN out of sequence, per the requirements in the Caesar final call, but replay protection may cause messages to be rejected.

Usage of this parameter set requires a precise understanding of the higher layer protocols. In this case, security depends on the authentication bits that are provided through the higher layer protocol checks that occur (since a modified ciphertext will randomize the plaintext bits), and the degree of impact that will result due to a successful forgery.

This parameter set is suitable for plaintexts with lengths that are 4 bytes or longer.

Chapter 5

Design Rationale

The main goals of this cipher are reduced message expansion and misuse resistance. Since CMCC is CCA-secure even when zero bytes of authentication are used, we can reduce the size of the authentication overhead. We have also utilized two mechanisms (stateless and stateful schemes respectively) to reduce the size of the message number information that is sent over the network.

CMCC requires additional block cipher operations vs. a one-pass algorithm such as OCB. So our design requires additional computational overhead in order to obtain the property where modified ciphertexts lead to randomized plaintexts.

The designer/designers have not hidden any weaknesses in this cipher. The security proof for CMCC rules out any weaknesses outside AES.

Chapter 6

Intellectual Property

There are no known patents, patent applications, planned patent applications, or other intellectual property constraints relevant to the use of this cipher. If any of this information changes, the submitter/submitters will promptly (and within at most one month) announce these changes on the crypto-competitions mailing list.

Chapter 7

Consent

The submitter/submitters hereby consent to all decisions of the CAESAR selection committee regarding the selection or non-selection of this submission as a second-round candidate, a third-round candidate, a finalist, a member of the final portfolio, or any other designation provided by the committee. The submitter/submitters understand that the committee will not comment on the algorithms, except that for each selected algorithm the committee will simply cite the previously published analyses that led to the selection of the algorithm. The submitter/submitters understand that the selection of some algorithms is not a negative comment regarding other algorithms, and that an excellent algorithm might fail to be selected simply because not enough analysis was available at the time of the committee decision. The submitter/submitters acknowledge that the committee decisions reflect the collective expert judgments of the committee members and are not subject to appeal. The submitter/submitters understand that if they disagree with published analyses then they are expected to promptly and publicly respond to those analyses, not to wait for subsequent committee decisions. The submitter/submitters understand that this statement is required as a condition of consideration of this submission by the CAESAR selection committee.

Bibliography

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

[BellrRogwyWagnr] Bellare M., Rogaway P., Wagner D.: The EAX mode of operation. *FSE 2004*, LNCS vol. 3017, Springer, pp. 389–407, 2004.

[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).

[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/.

[Gligor] Gligor V. and Donescu P.: Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes.*Fast Software Encryption: 8th International Workshop, FSE 2001*, Yokohama, Japan, April 2-4, 2001.

[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-Velrag, 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.

[KohnViegWhit] Kohno T., Viega J., and Whiting D.: CWC: A high-performance conventional authenticated encryption mode. *Fast Software Encryption*, pp. 408-426, 2004.

[KrovtzRogwy] Krovetz T., Rogaway P.: The Software Performance of Authenticated-Encryption Modes. *Fast Software Encryption 2011*, 2011.

[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)

[RogwyBellrBlack] Rogaway P., Bellare M., and Black J.: OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption. *ACM TISSEC* 6(3):365-403, 2003.

[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).

[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).