Chosen Ciphertext Secure (CCS): Symmetric Key CCA 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 a new symmetric encryption scheme: CCS or Chosen Ciphertext Secure scheme. CCS has the property that modifications to the ciphertext randomizes the resulting plaintext. Using this property, we prove the scheme is CCA2 secure. Thus we obtain CCA2 encryption schemes with minimal ciphertext expansion which are applicable to resource constrained wireless environments. For protocols that send short messages, our scheme is similar to Counter with CBC-MAC (CCM) for computation but has much shorter messages (since we can use a smaller MAC tag) for a similar level of security. A key idea is that various protocol fields in the underlying plaintext act as an authentication tag given changes to the message ciphertext. To the best of our knowledge, CCS is the first scheme that achieves CCA2 security with only 2-3 bytes of ciphertext expansion.

Keywords: Private key CCA2 encryption, energy constrained cryptography.

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 symmetric encryption scheme Chosen Ciphertext Secure (CCS) that utilizes a pseudorandom function (PRF) (e.g., AES). Our construction uses multiple invocations of the PRF so that any modifications to ciphertext result in a randomized plaintext. We will show that this property implies that our scheme has CCA2 security. CCS is a tweakable enciphering scheme (TES) [LskvRvstWgnr, HR03]. To the best of our knowledge, CCS is the first scheme that achieves CCA2 security with a small concrete security bound using only 2-3 bytes of ciphertext expansion.[1]

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

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

where the ciphertext is X_1, X_2, together with M, a public message number. For maximum security, M is unique, with high probability, for each message encrypted under a given key K. Then if the adversary flips some bits in X_1, the corresponding bits in X are flipped during decryption, and this produces random changes to P_1 during decryption (see 2nd equation). The first equation is then applied which results in random changes to P_2. A similar argument applies if we flip one or more bits in X_2. Since changes to any bits in the ciphertext result in random changes to the plaintext, it follows that the decryption oracle in the CCA2 security experiment isn't useful to the adversary either before or after the challenge ciphertext is obtained.

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

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

where the resulting ciphertext is $X_1, \ldots, X_k$. One possible instantiation of $f_1, \ldots, f_k$ would be with AES-CMAC-PRF-128 (RFC 4615) [SongPoovnLeeIwata], but other choices are possible. A common scenario is one where some packet loss and/or packet reordering may occur so that the communication peers aren't fully synchronized. In this case, the least significant 2-3 bytes of M are used as a message number tag T and the full ciphertext is $X_1, \ldots, X_k, T$ ($k \geq 2$.) T allows the decryptor to identify the full M value in order to decrypt, and it also hides the number of messages previously sent. If the communication peers are synchronized, then CCS requires no ciphertext expansion.

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

1.1 Applications

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

If we consider VoIP, a 20 byte payload is common. The transport and network layer headers (IP, UDP, and RTP) bring another 40 bytes, but compression [cRTP, Bormann] is used to reduce these fields down to 2-4 bytes. The link layer headers add another 6 bytes. Thus the total packet size is 30 bytes, assuming the UDP checksum of 2 bytes is included. In this case, by omitting the recommended 10 byte authentication tag and using CCS with 2 bytes of expansion, we obtain a 1/5 savings in message size and corresponding savings in energy utilization. (Actually, the savings is larger since encryption schemes send randomness (e.g., an IV) as well. For example, CCM sends a 13 byte nonce with each message.) Furthermore if the encryption boundary is just after the CID field (which is used to identify the full headers), then the UDP checksum is encrypted and acts as a 2 byte authentication tag. Even if the adversary was lucky enough to obtain the correct checksum, the resulting Voice payload would be noise, with high probability.

Wireless sensor networks also use short packets [VuranAkyldz] to maximize resource utilization; these packets are often in the range of 10-30 bytes.

1.2 Our Contributions

Our contributions are as follows:

1. We give a new family of private key encryption schemes with minimal ciphertext expansion. To the best of our knowledge, CCS is the first scheme that achieves CCA2 security with a small concrete security bound using only 2-3 bytes of ciphertext expansion.

2. We show that changes to a ciphertext result in a randomized plaintext with a distribution close to the uniform distribution. Based on this property we show that our scheme has CCA2 security in the concrete standard model, based on the assumption that pseudorandom functions exist.

3. We give a rough comparison for CPU overhead, network overhead, and energy consumption between CCM and CCS, where energy is based on a wireless sensor node, the Mica2Dots platform. Although the CCM MAC gives CCM a theoretical advantage over CCS with respect to integrity protection (in the case where CCS uses a smaller MAC), in practice we expect CCS to enjoy sufficient integrity protection for some network protocols since many network protocols have their own checks which will act as MAC tags under CCS. For example, VoIP has the checks above plus a randomized voice packet will not constitute a successful integrity

attack. To protect against denial of service attacks where an attacker's goal is simply to send any randomized protocol data, CCS may require a similar length MAC for equivalent security. Whether this is needed will depend on the protocols involved and the security policy.

1.3 Organization

In Section 2, we give basic cryptographic definitions. In Section 3, we present our symmetric key encryption scheme that has minimal ciphertext expansion. Section 4 gives the proof that our scheme has CCA2 security. Section 5 gives our performance analysis and results, including a comparison of energy utilization between CCS and CCM, for wireless sensor nodes. Section 6 covers related work. In Section 7 we draw conclusions.

2 Definitions

2.1 Pseudorandomness

The concatenation of two strings S and T is denoted by S, T.

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

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

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

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

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

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

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

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

2.2 (Misuse Resistant) CCA Encryption

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

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

4. A random bit $b \leftarrow \{0,1\}$ is selected. The ciphertext $c \leftarrow Enc_K(m_b)$ is computed and given to $\mathcal{A}$.

5. The adversary continues to have oracle access to $Enc_K()$ and $Dec_K()$. However, the adversary is not allowed to query the decryption oracle with the ciphertext c. The adversary is limited to q total queries (including the queries issued before the challenge ciphertext is generated).

6. The adversary outputs a bit $\bar{b}$. The output of the experiment is 1 if $\bar{b} = b$ and 0 otherwise.

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

The encryption scheme S is defined to have CCA2 security for (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 $Adv^{CCA2\_MRAE}_{S,n,q}(\mathcal{A}) = [Pr[Exp_{CCA2\_MRAE}(S, n, q, \mathcal{A}) = 1] - 1/2$.

2.3 (Misuse Resistant) CPA Encryption

Given the CCA2 encryption experiment above, except we remove the decryption oracle from the experiment. We define the resulting experiment as the CPA encryption experiment, and if the adversary probability of success is bounded as above, we say that the encryption scheme is CPA secure for (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 CCS

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

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

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

We will segment a plaintext message P into k input blocks. The input block size for P is the largest size less than or equal to the output block size such that the message P can be divided into k input blocks each with the input block size or one byte less than the input block size if needed. If P divides into k equal sized blocks, then input block size $= |P|/k$. We define α to be 2 raised to the input block size, in bits. As an example, consider a pseudorandom function constructed using the

AES encryption algorithm [AES]. The output block size is 16 bytes. If P has 33 bytes, then $k = 3$, the input block size is 11 bytes, and $\alpha = 2^{88}$.

3.1 Informal Design Intuition for Message Numbers

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

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

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

for private message number i where encryption key $\bar{K}$ is shared by the communication peers for the block cipher E (we assume the block size is 16 bytes). If the sender and receiver communication is synchronized, then M doesn't need to be transmitted. Otherwise, we send the least significant 2-3 (IL) bytes of the value M_i as described above except we eliminate M_i values from the sequence if the least significant IL byte(s) duplicate a previous M_j's least significant IL byte(s) where $(\gamma - j) \leq 2(window\_size) + 1$ 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 size of the ciphertext but the scheme is stateful.

3.2 CCS Specification

$LSB_j(x)$ and $MSB_j(x)$ denote the j least significant bytes and j most significant bytes of byte string x respectively. The two communication peers are denoted as the initiator ($init$) and responder ($resp$), respectively. There are two channels; one with the initiator as the encryptor and the responder as the decryptor, and the other with the initiator as the decryptor and the responder as the encryptor. We will describe the private message number (stateful) case; for public message numbers (stateless case), $\bar{K}_1$, $\bar{K}_2$, $E_{\bar{K}_1}$, and $E_{\bar{K}_2}$ are not used, and Sections 3.2.2, 3.2.3, and 3.2.6 are not needed. Also, M replaces the message number tag T in Sections 3.2.4 and 3.2.5.

3.2.1 Key Generation

Keys $\bar{K}_1$ and $\bar{K}_2$ are randomly generated for the pseudorandom permutations $E_{\bar{K}_i}$ $i = 1, 2$ and the randomly generated keys $L_1, \ldots, L_k$ 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.

3.2.2 Initial State

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

3.2.3 Creating the Sequences of M Values

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

```
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 $Seq(x)$.

3.2.4 Encryption

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

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

where $X_1, \ldots, X_k, T$ is the resulting ciphertext. We write $Enc_K(P, M) = X_1, \ldots, X_k, T$. For a pseudorandom function based on an underlying cryptographic algorithm with a block size (e.g., an AES based prf), padding may be necessary. In this case, we pad (append) the first and last equation inputs with zero bits. The other equation inputs are padded with the bits of M, least significant bits first. If additional padding is needed, zero bits are used.

3.2.5 Decryption

Let $y \in \{init, resp\}$ where $y \neq x$. Given C, T where $C = X_1, \ldots, X_k$. There exists at most one $\bar{M}$ in $Seq(x)$ such that $LSB_{IL}(\bar{M}) = T$ and $|SeqNo_x[\bar{M}] - y_d| \leq w\_s$. If it exists, then set $M = \bar{M}$ and compute the sequence

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

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

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

3.2.6 Channel Assumption

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

Let $y \in \{init, resp\}$ where $y \neq x$. The next ciphertext that is decrypted, $X_1, \ldots, X_k, T$ 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 summarizes the parameters for the stateful scheme.

Parameter	*Description*
k	Number of plaintext segments: $P = P_1, \ldots, P_k$.
α	$\alpha = 2^{\lvert P_i \rvert}$, $i = 1, \ldots, k$.
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: Summary of Parameters for Stateful CCS Scheme

4 Proof of CCA2 Security

We will first prove CCA security for the stateless version of CCS (the message number tag T defined above will only be used in the stateful scheme). We will then show how to extend this proof to the stateful CCS scheme defined above.

We now prove that our scheme is CPA-secure.

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

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

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. If $k > 2$, then CCS encryption is CPA secure for (ϵ, q) for the same value of ϵ provided that message numbers are not reused.

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

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

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 $f_2(M, X_2)$ is also uniform random and thus the X_1 values gives no information about the X values. Thus the ciphertexts give no information about the X values.

We have

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

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 $Adv_{(h_1,h_2)}^{CPA\_MRAE} \leq q(q-1)/2\alpha$. 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.

$$\begin{aligned} Adv_{f_2}^{prf}(q) \geq Adv_{f_2}^{prf}(D^g) &= |Pr[D^{(h_1,f_2)}() = 1] - Pr[D^{(h_1,h_2)}() = 1]| \\ &\geq Adv_{(h_1,f_2,n,q)}^{CPA\_MRAE}(\mathcal{A}) - q(q-1)/2\alpha. \end{aligned}$$

Thus $Adv^{CPA\_MRAE}_{(h_1,f_2,n,q)}(\mathcal{A}) \leq Adv^{prf}_{f_2}(q) + q(q-1)/2\alpha$ 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

$$\begin{aligned} Adv^{prf}_{f_1}(q) \geq Adv^{prf}_{f_1}(D^g) &= |Pr[D^{(f_1,f_2)}() = 1] - Pr[D^{(h_1,f_2)}() = 1]| \\ &\geq Adv^{CPA\_MRAE}_{(f_1,f_2,n,q)}(\mathcal{A}) - Adv^{prf}_{f_2}(q) - q(q-1)/2\alpha. \end{aligned}$$

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

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

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 4.2 *The adversary submits q_i queries where the queried plaintext or ciphertext has length l_i, $1 \leq i \leq v$, and $\sum_{i=1}^{v} q_i \leq q$. If $k = 2$, then the CCS encryption presented in the previous section is CCA2 MRAE secure for (ϵ, q) with*

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

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. If $k > 2$, then CCS encryption is CCA2 secure for (ϵ, q) for the same value of ϵ provided that message numbers are not reused.

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 first handle the $k=2$ case.

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 $2 \le i \le q$, $(i-1)/\alpha$ 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.

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 MRAE 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 MRAE security follows.

For the $k>2$ case, since the message numbers M are all distinct in the plaintext queries, we again have the same probability bound to ensure the X values are distinct. Thus we obtain the $k>2$ result in the same manner as for the above argument.

$\blacksquare$

Theorem 4.3 *Given the parameters defined in Theorem 4.2. If $k=2$, then the CCS stateful encryption scheme presented in Section 3 is CCA2 MRAE secure for (ϵ, q) with*

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

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. If $k>2$, then CCS stateful encryption is CCA2 secure for (ϵ, q) for the same value of ϵ provided that message numbers are not reused.

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. ▮

Table 2 gives the Theorem 4.3 bounds for 6, 8, 10, and 20 byte messages and varying numbers of adversary queries to the oracles.

msg length	α	*q (No. Adversary queries)*	$q(q-1)/2\alpha$	CCA2 security bound
6 bytes	2^{24}	2^{6}	.000120163	0.000120163
8 bytes	2^{32}	2^{8}	.0000076	0.0000076
8 bytes	2^{32}	2^{10}	.000121951	0.000121951
8 bytes	2^{32}	2^{12}	.00195264	0.00195264
10 bytes	2^{40}	2^{10}	.000000476371	0.000000476371
10 bytes	2^{40}	2^{12}	.00000762753	0.00000762753
10 bytes	2^{40}	2^{14}	.000122063	0.000122063
20 bytes	2^{80}	2^{12}	6.9372×10^{-18}	6.9372×10^{-18}
20 bytes	2^{80}	2^{14}	1.110155×10^{-16}	1.110155×10^{-16}
20 bytes	2^{80}	2^{16}	1.77633×10^{-15}	1.77633×10^{-15}
20 bytes	2^{80}	2^{20}	4.54747×10^{-13}	4.54747×10^{-13}

Table 2: Theorem 4.3 bounds for the adversary advantage given q queries for 6, 8, 10, and 20 byte messages. Security increases as message length increases. The security bound is approximately $q^2/2\alpha$.

We note that the computations of $f_2, \ldots, f_k$, can be parallelized. Also, if two blocks of the message are known in advance, X may be precomputed, so that $f_2(X), \ldots, f_k(X)$ can also be precomputed. f_1 could be instantiated with a parallelizable function; for example, it may be possible to use Poly1305-AES [Bernstein] for the function f_1 with a 16 byte value M as the nonce.

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 instantiate our pseudorandom functions $f_1, \ldots, f_k$ with AES-CMAC-PRF-128 (RFC 4615) [SongPoovnLeeIwata], but other choices are possible.

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

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 CCS based on the number of AES encryption

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

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

Message Length	*#CCM prf calls*	*#CCS prf calls*	*CCM energy use*	*CCS energy use*
8 bytes	4	4	1819.2	812.8
16 bytes	4	4	2292.8	1286.4
20 bytes	6	4	2580.8	1523.2
24 bytes	6	4	2817.6	1760
32 bytes	6	4	3291.2	2233.6
48 bytes	8	8	4289.6	3283.2
64 bytes	10	8	5288	4230.4
80 bytes	12	12	6286.4	5280
128 bytes	18	16	9281.6	8224

Table 4: Energy utilization (μJ) for sending network messages with CCM and CCS protection, Mica2dot platform, RFC 4615 instantiation for CCS

operations (pseudorandom function evaluations) and sizes of messages. The other CPU operations such as exclusive-or are minor usages and not counting them will not affect our results significantly. Table 4 gives the results.

Let $B = \lceil L/16 \rceil$, where L is the message length in bytes. For CCM, the number of AES block encryptions is equal to $2B+2$. For CCS, the number of prf invocations (AES block encryptions given the RFC 4615 prf instantiation of the prf) is $4W = 3W + \max\{W-1, 0\} + 1$ 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.

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

[2]CCS 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 CCS. 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 CCS can result in significant energy savings regardless of the size of the application layer messages.

The amount of energy used for CCM is

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

and the amount of energy for CCS is

$$4\lceil L/32\rceil 16(1.6\mu J) + (L + 4)(59.2\mu J) = 102.4\lceil L/32\rceil + 59.2L + 236.8\mu J$$

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 CCS as part of the comparison. Including such fields would give results very close to the ones above.

6 Related Work

We briefly overview related work in this section.

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.

Counter with CBC-MAC (CCM) [WhitHousFerg] is standardized as IETF RFC 3610. It specifies the use of AES in counter mode with CBC-MAC for integrity protection. As discussed above, our scheme has fewer block cipher calls and imposes less message expansion.

RFC 4493 [SongPoovnLeeIwata] specifies the AES-CMAC algorithm. This algorithm is a variable input length PRF with a fixed output length. Thus it can be used to instantiate our encryption scheme.

The line of work on authenticated encryption (typically with a single pass over the data using one key) is aimed at creating efficient primitives that provide both confidentiality and integrity for network messages. This approach was initiated by [Jutla] and includes [Gligor, KohnViegWhit, BellrRogwyWagnr] as well as the OCB variants [RogwyBellrBlack, KrovtzRogwy]. The efficiency gains over traditional combinations of encryption and MAC algorithms are mainly with respect to computation vs. message overhead. OCB [RogwyBellrBlack] recommends a tag with 64 bits. The other algorithms also create ciphertext expansion. In [KrovtzRogwy], the authors show that the OCB variants are more computationally efficient than CCM and GCM [McGrewViega].

The most widely used algorithms for CCA encryption consist of CPA secure encryption algorithms (e.g., CBC encryption or counter mode encryption) combined with a MAC that is existentially unforgeable under adaptive chosen message attack. Another approach for CCA encryption is given in [Katz-Yung2], but this approach also results in ciphertext expansion.

In [Desai], Desai gives CCA-secure symmetric encryption algorithms that don't use a MAC and don't provide explicit integrity protection outside of the CCA-security. CCS shares this CCA-security without a MAC property. The most efficient one is UFE which utilizes variable length pseudorandom functions. Its ciphertext expansion is $|r|$ bits where r is a uniform random value; security can be compromised if the same r is used for multiple messages. Since r is uniform random, collisions are likely after $2^{|r|/2}$ messages. Furthermore, with small probability, collisions will occur after a much smaller number of messages. The UFE security bound is $q(q+1)/2^{|r|}$. If the adversary can make 2^{20} queries, then Table 2 for 20 byte messages gives a security bound around 2^{-11} (CCS with 2 bytes of ciphertext expansion). UFE would require a 7 byte ciphertext expansion to assure the same security level. Alternatively, given 2^{12} queries, the respective expansions would be 2 and 5 bytes for CCS and UFE to assure a security bound around 2^{-11}. UFE is not resistant to nonce reuse; the privacy of the affected messages will be lost.

Another line of work, originally motivated by the problem of storage encryption includes CMC [HR03], [NR], TET [Hal07], XCB [FM04], EME [HR04], EME* [Hal04], HCH [CS06a], HCTR [WFW05], PEP [CS06b], and HEH, iHCTR, HOH [Sarkar]. These schemes can also be applied to network encryption. These schemes require plaintexts to be at least as long as the block size of the underlying block cipher whereas CCS can encrypt plaintexts that are shorter than the block size (e.g. 16 bytes) which is valuable for short messages. CCS also includes the integration of a minimal sized message number that enables the number of messages previously sent to be hidden.

SIV and PTE [RogwyShrmptn] were the first misuse resistant authenticated encryption (MRAE) schemes. SIV includes a MRAE scheme where the expansion includes the block cipher block size (e.g., 16 byte) IV plus the nonce.

7 Conclusions

We have presented CCS; to our knowledge it is the first symmetric encryption scheme that achieves CCA2 security with a small concrete security bound using only 2-3 bytes of ciphertext expansion. The security assumption that CCS relies on is the existence of pseudorandom functions. Based on this assumption, we have proved that CCS is CCA2 secure, and that the $k = 2$ case is CCA2 misuse resistant secure. Based on using AES as the underlying pseudorandom function, we have presented a comparison of energy utilization in wireless sensor networks between CCS and CCM and showed that energy use is proportional to packet length. Thus CCS can achieve significant energy savings when applied to protocols that send short messages due to its small ciphertext expansion.

References

[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 tweak- able 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).