Chosen Ciphertext Secure (CCS): Stateful 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 stateful 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 much smaller or no MAC tags) 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. 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.

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

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

where the ciphertext is X_1, X_2, and M is an unpredictable pseudorandom value. Also, 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 would be used as a value T and the full ciphertext would be $X_1, \ldots, X_k, T$ ($k \geq 2$.) T allows the decryptor to identify the full M value in order to decrypt. If the communication peers are synchronized, then CCS requires no ciphertext expansion.

Alternatively, the $k = 2$ construction above can also be applied to longer messages (see Section 6).

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 (assuming CCS doesn't use a 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. To protect against denial of service attacks where an attacker's goal is simply to send any randomized protocol data, CCS may require a short MAC tag 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. We discuss in Section 6. Section 7 covers related work. In Section 8 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 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.

The encryption scheme S is defined to have CCA2 security for (ϵ, q) if for all probabilistic polynomial time adversaries $\mathcal{A}$

$$Pr[Exp_{CCA2}(S, n, q, \mathcal{A}) = 1] \leq 1/2 + \epsilon.$$

2.3 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).

3 CCS

In this section, we present CCS; 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 refers to a per message value, or Initialization Vector (IV), which will be unique with high probability for each message encrypted under a key K.

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 M

It is important that the value M be unpredictable to an adversary. If an adversary can predict M, (e.g., say M is a counter which is incremented), then the adversary could send a request with a ciphertext and the predicted M value to the decryption oracle and then use the resulting plaintext as m_0 for the challenge ciphertext encryption. Then the adversary wins the CCA game with high probability. We instantiate M as follows:

Suppose 8 bytes is the desired length for M. Given encryption key $\bar{K}$ (shared by the communication peers) for the block cipher $\bar{E}$ (we assume the block size is 16 bytes). Then we let

$$MSB_8(\bar{E}_{\bar{K}}(i)) = M_i, i \geq 0,$$

where i is an integer counter, in order to generate the M values that the initiator uses for its encryptions, and MSB_8 denotes the 8 most significant bytes.

For cases where a longer M value is needed (e.g., 16 bytes for a better security bound), the block cipher can be invoked separately for each direction. Given the block cipher $\bar{E}$ with blocksize equal to 16 bytes using the scheme above, a 16 byte M will be unique for the duration of the key $\bar{K}$.

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. For encryptions sent by the responder, the MSB_8 function above is replaced with LSB_8 which denotes the 8 least significant bytes.

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.

3.2.1 Key Generation

Key $\bar{K}$ is randomly generated for the pseudorandom permutation $\bar{E}_{\bar{K}}$ and the randomly generated keys $L_1, \ldots, L_k$ determine the PRF's $f_1, \ldots, f_k$. The key $K = \bar{K}, L_1, \ldots, L_k$. $\bar{E}_{\bar{K}}$ 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 $F = LSB$ if $x = resp$, and let $F = MSB$ if $x = init$.
Let $Seq(x) = M_0, \ldots, M_{x_e - 1}$.
start: $candidate(M) = F_{l/16}(\bar{E}_{\bar{K}}(u_x))$
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

If $M_0, \ldots, M_{x_e-1}$ have been used for previous encryptions but M_{x_e} hasn't, then we set $M = M_{x_e}$ and $T = LSB_{IL}(M)$. 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 $E_K(P) = X_1, \ldots, X_k, T$. For a pseudo-random 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. Finally, the encryptor increments x_e : $x_e \leftarrow x_e + 1$.

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, and T hasn't already been used for a previous decryption within the current window (defined as $|SeqNo_x[\bar{M}] - y_d| \leq w\_s$), 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 randomness used to decrypt C, T. 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 randomness 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 $LSB(\bar{M}) = T$, 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	$\alpha = 2^{P_i}$, $i = 1, \ldots, k$.
M	randomness value for encryption
w	number of possible values for M
$E_{\bar{K}}()$	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 a stateless version of CCS where the value M is included as part of the ciphertext (the value T defined above will only be used in the stateful scheme). In the stateless version, the value M is uniformly distributed. We will then show how to extend this proof to the stateful CCS scheme defined above.

Our approach is to first prove a lemma that shows that modifying a ciphertext gives another ciphertext which decrypts to an almost uniformly distributed plaintext. (The basic idea is that an existing set of decrypted ciphertexts gives a slightly greater chance for collisions when decrypting a new ciphertext but that the probabilities are still very close to uniform). We then show that our scheme is CPA secure (Theorem 4.2). Finally, using the lemma, we show that an adversary has at most a small probability of distinguishing the results from the decryption oracle in the CCA2 security experiment from independent samples of the uniform distribution. This indistinguishable view implies that a challenger can reply to the adversary's decryption oracle queries by generating uniform random answers, and therefore that the decryption oracle gives the adversary only a small advantage over the CPA security experiment. Thus the scheme is CCA2 secure in the concrete model (Theorem 4.3). We extend the same argument to the stateful scheme in Theorem 4.4.

Lemma 4.1 *Given the encryption scheme described above, where the f_i functions are randomized functions (for the idealized scheme), $1 \leq i \leq k$. Thus each f_i maps arbitrary length strings to fixed length output strings of length $|P_i|$ as described above, where $\alpha = 2^{|P_i|}$, $1 \leq i \leq k$. Given ciphertexts $A_i = X_1^i, \ldots, X_k^i, M$, $1 \leq i \leq v$. Let $B = Y_1, \ldots, Y_k, M$ be another string of the same length. Given that $Dec_K(A_i, M) = P_1^i, \ldots, P_k^i$. Let $Y = Y_1 \oplus f_1(M, Y_2, \ldots, Y_k)$. Then $Dec_K(B, M)$ is as follows:*

case i: *$Y_2, \ldots, Y_k \neq X_2^i, \ldots, X_k^i$ for all i, $1 \leq i \leq v$. Let Q_{j-1}^i be obtained from P_{j-1}^i via the same bit flips as used to obtain Y_j from X_j^i, $2 \leq j \leq k$, $1 \leq i \leq v$. (Thus $Q_{j-1}^i = P_{j-1}^i \oplus X_j^i \oplus Y_j$.) Let b be the number of distinct values from $X_{A_i} = f_k(M, P_1^i) \oplus P_k^i$, $1 \leq i \leq v$. Then*

1. *$Pr[Dec_K(Y_j, M) = Q_{j-1}^i] = c_{j,i}/\alpha + (\alpha - b)/\alpha^2$, $2 \leq j \leq k, 1 \leq i \leq v$, for some integers $c_{j,i}$ where $1 \leq b \leq v$ and $1 \leq c_{j,i} \leq b$.*
2. *If $R_{j-1} \neq Q_{j-1}^i$ for $1 \leq i \leq v$, then $Pr[Dec(Y_j, M) = R_{j-1}] = (\alpha - b)/\alpha^2$, $2 \leq j \leq k$.*
3. *Y, M decrypts to R_k with probability $1/\alpha$ for any plaintext R_k. ($Pr[Dec_K(Y, M) = R_k] = 1/\alpha$.)*

case ii: *Let $Y_1 \neq X_1^i$ for all i, $1 \leq i \leq v$. $Y_j = X_j^i$, $2 \leq j \leq k$ and $1 \leq i \leq v$. Let Q_k^i be obtained from P_k^i via the same bit flips as used to obtain Y_1 from X_1^i; $Q_k^i = P_k^i \oplus X_1^i \oplus Y_1$, $1 \leq i \leq v$. Let b_k be the number of distinct Q_k^i's, and let c be the number of distinct P_1^i's. Let $c_i = |\{P_1^j : f_k(M, P_1^j) = f_k(M, P_1^i)\}|$, $1 \leq i \leq v$.*

1. *$Pr[Dec_K(Y, M) = Q_k^i] = c_i/\alpha + (\alpha - c)/\alpha^2$, $1 \leq i \leq v$.*
2. *Let $R_k \neq Q_k^1, \ldots, Q_k^v$. Then $Pr[Dec_K(Y, M) = R_k] = (\alpha - c)/\alpha^2$.*
3. *$Pr[Dec_K(Y_j, M) = R_{j-1}] = 1/\alpha$, $2 \leq j \leq k$, for any string R_{j-1} with the same length as Y_j.*

case iii: *Let $Y_2, \ldots, Y_k \neq X_2^i, \ldots, X_k^i$ for all i, $1 \leq i \leq t$. Let $Y_1 \neq X_1^i$ for all i, $t+1 \leq i \leq v$, and $Y_j = X_j^i$, $2 \leq j \leq k$ and $t+1 \leq i \leq v$. Let Q_{j-1}^i be obtained from P_{j-1}^i via the same bit flips as used to obtain Y_j from X_j^i, $2 \leq j \leq k$, $1 \leq i \leq t$. (Thus $Q_{j-1}^i = P_{j-1}^i \oplus X_j^i \oplus Y_j$.) Let Q_k^i be obtained from P_k^i via the same bit flips as used to obtain Y_1 from X_1; $Q_k^i = P_k^i \oplus X_1^i \oplus Y_1$, $t+1 \leq i \leq v$. Then:*

1. *$Pr[Dec(Y_j, M) = Q_{j-1}^i] = c_{j,i}/\alpha + (\alpha - b)/\alpha^2$, $1 \leq i \leq t$, $2 \leq j \leq k$, where $c_{j,i}$ is the number of distinct X_{A_l}'s that satisfy $f_j(X_{A_l}) = f_j(X_{A_i})$, $1 \leq l \leq t$. Also, b is the number of distinct X_{A_l}'s, $1 \leq l \leq t$.*
2. *For a string R_{j-1} of the same length as Y_j, where $R_{j-1} \neq Q_{j-1}^i$ for $1 \leq i \leq t$, then $Pr[Dec(Y_j, M) = R_{j-1}] = (\alpha - b)/\alpha^2$, $2 \leq j \leq k$.*
3. *If $v^2 \leq \alpha$ and $\alpha \geq 2^{16}$, $Pr[Dec_K(Y, M) = Q_k^i] < 2\frac{1}{256}(1/\alpha)$, $t+1 \leq i \leq v$.*
4. *Let $R_k \neq Q_k^{t+1}, \ldots, Q_k^v$. Then $Pr[Dec_K(Y, M) = R_k] \geq (\alpha - 2v)/\alpha^2$.*

If any of the ciphertexts A_i is of the form $X_1^i, \ldots, X_k^i, \bar{M}$ where $\bar{M} \neq M$, this ciphertext would be treated as one of the case i ciphertexts or one of the first t of the case iii ciphertexts and the lemma would hold verbatim. Reordering the ciphertexts in case iii doesn't affect the result.

Proof: **case i:** $f_1(M, Y_2, \ldots, Y_k)$ is uniformly distributed and therefore so is Y.

$$Dec(Y_j, M) = Y_j \oplus f_j(Y) = Q_{j-1}^i \oplus P_{j-1}^i \oplus X_j^i \oplus f_j(Y) = Q_{j-1}^i$$

if and only if $P_{j-1}^i \oplus X_j^i = f_j(Y)$. But $P_{j-1}^i \oplus X_j^i = f_j(X_{A_i})$ where $X_{A_i} = f_k(M, P_1^i) \oplus P_k^i$, $1 \leq i \leq v$. Thus $Dec_K(Y_j, M) = Q_{j-1}^i$ if and only if $f_j(X_{A_i}) = f_j(Y)$, $1 \leq i \leq v$. Let b be the number of distinct X_{A_i}'s. Thus $1 \leq b \leq v$. Let $c_{j,i}$ be the number of distinct X_{A_l}'s that satisfy $f_j(X_{A_l}) = f_j(X_{A_i})$. Thus $1 \leq c_{j,i} \leq b$. Now we have $Pr[Dec_K(Y_j, M) = Q_{j-1}^i] = c_{j,i}/\alpha + (\alpha - b)/\alpha^2$, $2 \leq j \leq k$, $1 \leq i \leq v$.

If $R_{j-1} \neq Q_{j-1}^i$ for $1 \leq i \leq v$, then $Dec(Y_j, M) = R_{j-1}$ if and only if $f_j(Y) = Y_j \oplus R_{j-1}$. This last event requires $Y \neq X_{A_i}$, $1 \leq i \leq v$. Thus $Pr[Dec(Y_j, M) = R_{j-1}] = (\alpha - b)/\alpha^2$, $2 \leq j \leq k$.

Finally for the decryption of Y, since its uniformly distributed, so is the resulting plaintext R_k.

case ii: $Dec_K(Y, M) = Y \oplus f_k(M, Y_2 \oplus f_2(Y))$. So $Dec_K(Y, M) = Q_k^i$ if and only if $f_k(M, Y_2 \oplus f_2(Y)) = f_k(M, P_1^i)$. $Pr[Y_2 \oplus f_2(Y) = P_1^j \ where\ f_k(M, P_1^j) = f_k(M, P_1^i)] = c_i/\alpha$. $Q_k^j = Q_k^l$ if and only if $f_k(M, P_1^j) = f_k(M, P_1^l)$. Also, if $P_1^j = P_1^l$, then $Q_k^j = Q_k^l$. $Pr[Dec_K(Y, M) = Q_k^i] = c_i/\alpha + (\alpha - c)/\alpha^2$, $1 \leq i \leq v$.

Let $R_k \neq Q_k^1, \ldots, Q_k^v$. Then $Dec_K(Y, M) = R_k$ if and only if $Y \oplus f_k(M, Y_2 \oplus f_2(Y)) = R_k$. $Pr[Y_2 \oplus f_2(Y) \neq P_1^j \ for\ all\ j] = (\alpha - c)/\alpha$. So $Pr[Dec_K(Y, M) = R_k] = (\alpha - c)/\alpha^2$.

Finally, since Y is distinct from X_{A_i}, $1 \leq i \leq v$, we have that $Pr[Dec_K(Y_j, M) = R_{j-1}] = 1/\alpha$, $2 \leq j \leq k$.

case iii: The proof of (1) and (2) follow case i very closely. We prove the last two statements here. $Dec_K(Y, M) = f_k(M, Y_2 \oplus f_2(Y)) \oplus Y$; thus $Dec_K(Y, M) = Q_k^i$ if and only if $f_k(M, Y_2 \oplus f_2(Y)) = f_k(M, P_1^i)$. $Y_2 \oplus f_2(Y) = P_1^j$, where $f_k(M, P_1^j) = f(M, P_1^i)$ with probability equal to c_i/α.

$$\begin{aligned} Pr[Dec_K(Y, M) = Q_k^i] &= t/\alpha[c_i/\alpha + (\alpha - c)/\alpha^2] + (\alpha - t)/\alpha[(\alpha - c)/\alpha^2] \\ &\leq v^2/\alpha^2 + t/\alpha^2 + 1/\alpha < v^2/\alpha^2 + v/\alpha^2 + 1/\alpha \end{aligned}$$

Given $v^2 \leq \alpha$ and $\alpha \geq 2^{16}$, we have $v^2/\alpha^2 + v/\alpha^2 + 1/\alpha \leq 2/\alpha + v/\alpha^2 < 2\frac{1}{256}(1/\alpha)$.

We now prove (4): Let $R_k \neq Q_k^{t+1}, \ldots, Q_k^v$. Then $Dec_K(Y, M) = R_k$ if and only if $Dec_K(Y_2, M) \neq P_1^i$, $t + 1 \leq i \leq v$ and $R_k = Y \oplus f_k(M, Dec_K(Y_2, M))$. With probability greater than or equal to

$$\begin{aligned} 1 - \sum_{i=t+1}^{v} (c_i/\alpha + (\alpha - c)/\alpha^2) &\geq 1 - c/\alpha - (v - t)(\alpha - c)/\alpha^2 \\ &\geq (\alpha^2 - \alpha c - v\alpha)/\alpha^2 = (\alpha - c - v)/\alpha \geq (\alpha - 2v)/\alpha \end{aligned}$$

$Dec_K(Y_2, M) \neq P_1^i$, $t + 1 \leq i \leq v$. Thus $Pr[Dec_K(Y, M) = R_k] \geq (\alpha - 2v)/\alpha^2$. ∎

We now prove that our scheme is CPA-secure.

Theorem 4.2 *Let w be the number of possible values for M, where M is generated as described in Section 3.2, and let $\alpha = 2^{|P_i|}$, $1 \leq i \leq k$. The CCS encryption mode presented in the previous section is CPA-secure for (ϵ, q) with*

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

given that the adversary is restricted to q queries. The last term arises in going from the idealized model to pseudorandom functions.

Proof: The adversary is given $E_K(m_b)$ for $b \leftarrow \{0,1\}$. We will initially assume that the value M is unique for this encryption. We will also initially assume that $f_1, \ldots, f_k$ are random functions (in the idealized model). Thus X is uniformly distributed. If the X values from different encryption oracle requests collide, then the adversary has an advantage in guessing b.

$$Pr[\text{collision between } m_b \text{ encryption } X \text{ value and one of the } q \text{ query} X \text{ values}] \leq q/\alpha$$

On the other hand, if the X used for $E_K(m_b)$ hasn't been repeated from any query, then the adversary's probability of determining b equals $1/2$, since the f_i's are random functions and X is uniformly distributed. Thus

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

If a collision occurs in the M values between m_b and one of the queries, then the adversary can win the CPA game. We recall w is the number of possible values for M. The probability of a collision is bounded by q/w. Finally, the term $\sum_{i=1}^{k} Adv_{f_i}^{prf}(q)$ arises as we replace the random functions f_i with pseudorandom functions in the usual standard 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 4.3 *Let w be the number of possible values for M, where M is uniformly distributed, and let $\alpha = 2^{|P_i|}$, $1 \leq i \leq k$. The adversary submits q_i queries where the queried plaintext or ciphertext has length l_i, $1 \leq i \leq v$, and $\sum_{i=1}^{v} q_i \leq q$. Let*

$$\beta = 4^{k-1}k(2\frac{1}{256})q(q-1)/(2\alpha).$$

Let $p = (k-1)q^4/(24\alpha^3)$. Let $\epsilon_2(q) = q/\alpha + q/w + \sum_{i=1}^{k} Adv_{f_i}^{prf}(q)$. The CCS stateless encryption scheme is CCA2-secure for (ϵ, q) with

$$\epsilon = (1 - e^{\sum_{i=1}^{v}(-(q_i-1)q_i)/2^{l_i+1}}) + q/w + q^2/2^{2|P_i|+1} + (1 - e^{-\beta}) + \epsilon_2(q) + p$$

assuming the adversary is restricted to no more than q queries in the CCA2 security game.

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.

The probability of no collision is at least

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

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.)

There is also the possibility that $\mathcal{A}$ submits (C_1, M) and receives P, then submits P to receive (C_2, M) where $C_1 \neq C_2$. This event contradicts the definition of encryption, so $\mathcal{A}$ wins the game in this case. This event occurs with probability bounded by q/w. The probability can exceed q/w if multiple ciphertext queries return the same plaintext P. The probability for this last event is bounded by $q^2/2^{2|P_i|+1}$ (since $k = 2$ gives the smallest message sizes).

We now show that the distribution of plaintexts from the decryption oracle is indistinguishable from the uniform distribution with high probability, given q queries. We will show that the set of plaintexts which have a higher probability of being output by the decryption oracle, then under the uniform distribution, have at most the same number of occurrences as in the uniform distribution expected case, except with probability bounded by $f(q)$, given q queries. We give an expression for $f(q)$.

By Lemma 4.1, the $(i+1)st$ query to the decryption oracle gives rise to at most

$$\sum_{j=0}^{k-1}\binom{k}{j} i^{k-j}(\alpha-i)^j = \alpha^k - (\alpha-i)^k$$

higher probability plaintexts. This last fact can be derived by expanding $(\alpha - i + i)^k = \alpha^k$ using the binomial theorem and substracting the 0th term $((\alpha - i)^k)$.

The number of higher probability plaintexts for the decryption of the $(i+1)st$ ciphertext is bounded by

$$\alpha^k - (\alpha-i)^k \leq \alpha^{k-1} ik, \; 1 \leq i \leq q-1.$$

In the uniform distribution, we would expect these plaintexts to have less than z occurrences in the CCA2 security experiment, where $z = \lfloor \sum_{i=1}^{q-1} \alpha^{k-1} ik(\alpha^{-k}) \rfloor = \lfloor (q(q-1)/2)k\alpha^{k-1}\alpha^{-k} \rfloor = \lfloor q(q-1)k/(2\alpha) \rfloor$. (This last term will evaluate to zero for the parameters we are interested in.)

We now calculate the probability of not obtaining any high probability plaintexts (hpp's) from the decryption oracle: An upper bound on the probability of receiving a hpp, with high probability, for the decryption of the $(i+1)st$ ciphertext is

$$(\alpha^{k-1} ik)(4/\alpha)^{k-1}(2\frac{1}{256})/\alpha$$

by Lemma 4.1. Here we may select $\max c_{ji} \leq 3$ which holds as long as f_j collisions on the X_{A_l} values are such that at most 3 X_{A_l} values collide under f_j, $2 \leq j \leq k$. Thus the bound from Lemma 4.1 case iii (1) is less than $4/\alpha$.

The probability that 4 or more X_{A_l} values collide under f_j over all $f_j's$ is upper bounded by $(k-1)\binom{q}{4}(1/\alpha^3) \leq (k-1)q^4/(24\alpha^3)$. So with high probability this collision event does not occur and subject to the error term, we may assume $\max c_{ji} \leq 3$.

Since $e^{-x} \approx 1 - x$ for small x values, we have that

$$1 - \prod_{i=1}^{q-1} \left(1 - \frac{4^{k-1} ik(2\frac{1}{256})}{\alpha} \right) \approx 1 - \prod_{i=1}^{q-1} e^{-4^{k-1} ik(2\frac{1}{256})/\alpha} = 1 - e^{-4^{k-1}k(q(q-1))(2\frac{1}{256})/2\alpha} = f(q)$$

is an upper bound on the probability of receiving at least one hpp.

Thus with probability at least equal to $1 - f(q)$, the adversary cannot distinguish between being invoked by B and running in the CCA2 security game. Thus the adversary's advantage is the same as in the CPA security game when none of the above failure events occur (each of which gives rise to one of the additional terms bounding the adversary's success). ▮

Security is increased if the messages are slightly longer (e.g 8-20 bytes), or the environment is such that the adversary would only be able to submit fewer queries.

Theorem 4.4 *Given the parameters defined in Theorem 4.3, and let* $\epsilon_3(q)$ *be the adversary advantage from Theorem 4.3. The CCS stateful encryption scheme presented in the previous section is CCA2-secure for* (ϵ, q) *where* $\epsilon = \epsilon_3(q) + w\_s/2^{8IL} + Adv^{prf}_{\bar{E}_{\bar{K}}}(q) + Adv^{prf}_{\bar{E}_{\bar{K}}}(q + w\_s) - q/w$.

Proof: The challenger B can utilize the encryption oracle and maintain state for the stateful scheme. Then the proof of Lemma 4.1 follows as above. The proof of Theorem 4.2 is modified as follows: the value M is no longer uniformly distributed but is now selected using the pseudorandom function $\bar{E}_{\bar{K}}$ as described above. Thus the probability of collisions is no longer bounded by q/w but instead is bounded by $q/w + Adv^{prf}_{\bar{E}_{\bar{K}}}(q)$.

The proof of Theorem 4.3 follows as above, except for the case where the adversary submits (C_1, T) then receives P, then submits P to receive (C_2, T) where $C_1 \neq C_2$. This event contradicts the definition of encryption, so $\mathcal{A}$ wins the game in this case. However, T must be a tag not previously output by B in response to one of $\mathcal{A}$'s encryption oracle queries within the decrypt window. Let $\epsilon(q')$ be the probability that $\mathcal{A}$ outputs a tag in the decrypt window, given that $\mathcal{A}$ has seen q' previous tags. Consider the following adversary strategy for the prf game with the function $\bar{E}_{\bar{K}}$. The adversary obtains $\bar{E}_{\bar{K}}(0), \ldots, \bar{E}_{\bar{K}}(q')$ and then obtains tags $T_1, \ldots, T_s$ by taking the least significant IL bytes of the first sequence elements and deleting any duplicates within the window size. Now $\mathcal{A}$ outputs its guess at a tag in the decrypt window. The adversary's success probability is bounded by

$\epsilon(q')$. The adversary can now query w_s more elements. If the predicted tag is present, the adversary ouputs 1 for the prf game, otherwise 0. Thus $\epsilon(q) \leq w\_s/2^{8IL} + Adv^{prf}_{\bar{E}_{\bar{K}}}(q + w\_s)$ given $q' = q$ ($w\_s/2^{8IL}$ is the probability the adversary outputs 1 in the case of a random function oracle in the prf game). So the probability that T is in the decrypt window is bounded by $w\_s/2^{8IL} + Adv^{prf}_{\bar{E}_{\bar{K}}}(q + w\_s)$. If T is outside the window, then the session is aborted and $\mathcal{A}$ is unsuccessful[1]. ▮

Table 2 gives the Theorem 4.4 bounds for 8, 10, and 20 byte messages and varying numbers of adversary queries to the oracles.

msg length	α	*q (No. Adversary queries)*	$1 - e^{-\beta}$	Theorem 4.4 bound
8 bytes	2^{32}	2^{8}	.00012181	0.000594889
8 bytes	2^{32}	2^{10}	.00195292	0.00242618
8 bytes	2^{32}	2^{12}	.0308154	0.0312893
10 bytes	2^{40}	2^{10}	.00000763683	0.00048066
10 bytes	2^{40}	2^{12}	.000122259	0.000595289
10 bytes	2^{40}	2^{14}	.0019547	0.00242774
20 bytes	2^{80}	2^{12}	1.11212×10^{-16}	0.000472764
20 bytes	2^{80}	2^{14}	1.77972×10^{-15}	0.000472764
20 bytes	2^{80}	2^{16}	2.84768×10^{-14}	0.000472764
20 bytes	2^{80}	2^{20}	7.29016×10^{-12}	0.000472764

Table 2: Theorem 4.4 bounds for the adversary advantage given q queries, $w\_s = 31$, and $IL = 2$ (2 bytes of ciphertext expansion), for 8, 10, and 20 byte messages. Security increases as message length increases and as the number of bytes of ciphertext expansion increases. The security bound is approximately $8q^2/\alpha + (w\_s)2^{-8IL}$. β is from Theorem 4.3.

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, such as using Poly1305-AES [Bernstein] for the function f_1. The latter would most likely result in a substantial gain in performance (but not reduced energy utilization for short messages).

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

[1] Session termination and logging the event is the usual result given an integrity or decryption failure. If the session was allowed to continue, we would replace the $w\_s/2^{8IL}$ term with $q/2^{8IL}$.

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	753.6
16 bytes	4	4	2292.8	1227.2
20 bytes	6	6	2580.8	1515.2
24 bytes	6	6	2817.6	1752
32 bytes	6	6	3291.2	2225.6
48 bytes	8	8	4289.6	3224
64 bytes	10	10	5288	4222.4
80 bytes	12	12	6286.4	5220.8
128 bytes	18	18	9281.6	8216

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

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 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

$$\begin{cases} 4 & \text{for} \quad L \leq 16 \\ 2(B-1) + 4 = 2B + 2 & \text{for} \quad L > 16 \end{cases}$$

Table 4 assumes (1) that CCM uses the minimal recommended length message tag of 8 bytes which increases the length of the message by 8 bytes while CCS includes the 3 byte value T as described above, (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

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

$$\begin{cases} 280 + 59.2L(\mu J) & \text{for} \quad L \leq 16 \\ (32B + 32)(1.6\mu J) + (L59.2\mu J) = 228.8 + 51.2B + 59.2L(\mu J) & \text{for} \quad L > 16 \end{cases}$$

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 Discussion

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.

We can also define CCS as a block-cipher mode if we specify the PRF's and PRP in the CCS definition as block-cipher constructions.

Although our emphasis has been on utilizing CCS to protect short messages in energy constrained environments, CCS can be defined as a general case authenticated encryption algorithm:

1. We can obtain misuse resistance [RogwyShrmptn] by generalizing the initial CCS encryption step. To maximize misuse resistance, we can use a variant of CCS that gives more protection against accidental reuse of the pseudorandom value M with the same key. The initial encryption equation can be replaced with $X = f_k(M, P_1, \ldots, P_{k-1}) \oplus P_k$ which ensures that two distinct plaintexts encrypt to unrelated ciphertexts, given the computationally bounded adversary. Decryption is modified analogously. The precise claim and proof is left as future work. We note that this coincides with the CCS definition above when messages have 32 bytes or less (also see item 6 below).

2. We utilize a MAC as follows. The plaintext P is appended with zero bits prior to encryption to obtain a new plaintext $P_{new} = P, Z$ where Z is the zero bit field consisting of τ zero bits. We then segment P_{new} into $P_1, \ldots, P_k$ as before. Since CCS is CCA2 secure without the MAC, the length of the MAC will often be less than with other schemes, or it can be zero length as described above. Based on the security arguments above, we have that modifications to the ciphertext randomize the resulting plaintext with distribution close to uniform. Thus the probability that a modified ciphertext will pass the MAC check (which consists of verifying that the zero bit field has only zero bits) is approximately $2^{-\tau}$.

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.

3. Associated data that is not encrypted can be integrity protected by concatenating it with the M value prior to the prf computations.

4. Replay protection is built into the CCS scheme as a consequence of the windowing mechanism.

5. Keys for the pseudorandom functions can be derived from a single key. For example, a block cipher in counter mode can be used with the shared secret key and a secret session IV to generate keys for the prf's. Alternatively, the $k = 2$ case can be used for longer messages where we use f_3 (with a distinct key) in place of f_2 in the first encryption equation and f_3 is a block cipher in CBC mode, f_2 in the second equation uses a block cipher for CBC-MAC followed by counter-mode where the output of CBC-MAC is the counter-mode IV, and f_1 is a block cipher in CBC mode.

6. Message numbers can be encrypted and thus we provide confidentiality with respect to the number of messages previously sent.

7 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 the same number of block cipher calls but 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}.

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. But the CCS security bound and number of permitted adversary queries decreases for very small plaintexts (e.g., less than 4 bytes), where the plaintext includes any MAC bytes. Thus a 1 byte message plus a four byte MAC would be a 5 byte plaintext. CCS also includes the integration of a minimal sized IV.

CCA security for public key encryption goes back at least to [NaorYung]. The paper [Katz-Yung1] characterizes various properties for private key encryption including non-malleability and CCA security. Non-malleability where the adversary is given access to encryption and decryption oracles both before and after being given the challenge ciphertext is equivalent to CCA2 security.

8 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. 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. CCS can also be used as a general case authenticated encryption algorithm including misuse resistance, additional integrity protection, replay protection, and message number confidentiality.

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.

[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).