# On the Security of the CCM Encryption Mode and of a Slight Variant

Pierre-Alain Fouque, Gwenaëlle Martinet, Frédéric Valette, Sebastien Zimmer

# On the Security of the CCM Encryption Mode and of a Slight Variant

Pierre-Alain Fouque[1] and Gwenaëlle Martinet[2] and Frédéric Valette[3] and Sébastien Zimmer[1]

[1] École normale supérieure, 45 rue d'Ulm, 75005 Paris, France
{Pierre-Alain.Fouque;Sebastien.Zimmer}@ens.fr
[2] DCSSI Crypto Lab, 51 Boulevard de la Tour-Maubourg F-75700 Paris 07 SP, France Gwenaelle.Martinet@sgdn.pm.gouv.fr
[3] CELAR, 35 Bruz, France Frederic.Valette@dga.defense.gouv.fr

**Abstract.** In this paper, we present an analysis of the CCM mode of operations and of a slight variant. CCM is a simple and efficient encryption scheme which combines a CBC-MAC authentication scheme with the counter mode of encryption. It is used in several standards. Despite some criticisms (mainly this mode is *not online*, and requires *non-repeating nonces*), it has nice features that make it worth to study. One important fact is that, while the privacy of CCM is provably garanteed up to the birthday paradox, the authenticity of CCM seems to be garanteed beyond that. There is a proof by Jonsson up to the birthday paradox bound, but going beyond it seems to be out of reach with current techniques. Nevertheless, by using pseudo-random functions and not permutations in the counter mode and an authentication key different from the privacy key, we prove security beyond the birthday paradox.
We also wonder if the main criticisms against CCM can be avoided: what is the security of the CCM mode when the nonces can be repeated, (and) when the length of the associated data or message length is missing to make CCM *on-line*. We show generic attacks against authenticity in these cases. The complexity of these attacks is under the birthday paradox bound. It shows that the lengths of the associated data and the message, as well as the nonces that do not repeat are important elements of the security of CCM and cannot be avoided without significantly decreasing the security.
**Keywords:** CCM, CBC-MAC, Counter mode

## 1 Introduction

CCM stands for CTR + CBC-MAC and has been proposed by Doug Whiting, Russ Housley and Niels Ferguson. It is an *authenticated encryption* scheme based on the MAC-then-encrypt generic construction. It is interesting since it uses two very popular symmetric key schemes which are implemented in a lot of products and so, CCM can be constructed using "on the shelf" functions. It is used in many standards of wireless networks such as IEEE 802.11 [22] (WiFi), IEEE

802.15.40 (Wireless Personal Area Network/ZigBee), standards of the internet in the RFC 3610 and RFC 4309 and finally in the NIST SP 800-38C [10].

The security of CCM is very interesting since it relies on some padding or formatting functions. Such requirements are not appreciated in general and cryptographers try to avoid such properties: for example, the security should not hold only because the length of the message is included in some message block. However, such specific requirements have been used before to construct hash function as in the Merkle-Damgard transformation of compression function to hash function or in order to make secure the CBC-MAC function for messages of arbitrarily length. It is a well-known property that messages that include their length in the first block are prefix-free and such property can be used to avoid classical attacks on the CBC-MAC.

CCM has also been criticized by some authors [19] who highlight three efficiency issues: "CCM is not on-line, CCM disrupts word-alignment, and CCM can't preprocess static associated data". The main issue is that CCM is not on-line since the sender has to know the length of the message before the beginning of the encryption. The two other critiques concern the associated data. Consequently, we have tried to see whether such criticisms can be avoided in the attack part of this paper.

### 1.1 Related Works

The security notions of symmetric encryption schemes have been intensively explored [2, 3, 5, 9, 14] and are now well understood. This background has allowed to design and analyze several operating modes [2, 15, 17, 7] for symmetric authenticated encryption.

In this vein, two main authenticated encryption schemes with associated data were designed: AEX [7] and CCM [21]. They both are two-pass modes with non-repeating nonces and they both have been proved secure [12, 7] until $2^{n/2}$ encryption queries for privacy *and* integrity. This bound is a classical bound and an encryption scheme secure up to this bound is commonly considered as secure. According to Jonsson, the privacy of CCM cannot be proved beyond the birthday paradox. However maybe the scheme is a good authentication scheme beyond this bound. At the end of his paper, Jonsson explains that if the CCM security is guaranteed until $2^{n/2}$ encryption queries, no attack which reaches this bound is known. Jonsson left as an open problem to fill the gap between the better known attack in $2^n$ encryption queries and this security bound. More precisely, he conjectures a security in $2^n$ encryption queries.

### 1.2 Our results

The first part of our result concerns the presentation of an encryption scheme secure beyond the birthday paradox bound. We rely on CCM and propose a slight variant of the CCM mode for which we are able to give a security proof beyond the birthday paradox for privacy and authenticity. We do not alter CCM too much to preserve some interesting properties. Precisely, we replace the block

cipher used in the counter mode with a pseudo-random function. If one wants to base the security of the scheme on block cipher security, this pseudo-random function can be built using several block ciphers such as in [4, 11, 16]. Another alternative is to use the compression function of a hash function, where the key takes the place of the IV. This solution relies on the non classical assumption, that the compression function is a good pseudorandom function. However this assumption is more and more common [1, 8] and is realistic. The privacy proof is a consequence of the privacy of the counter (CTR) mode when it is used with a random function. The authentication proof is built upon a method from [15] using the fact that with CTR, the encryption of the tag cannot be distinguished from a random bit string. Therefore one does not have to generate the tag to simulate the encryption.

In the second part of this paper, we try to justify why the non-repeating nonces and the length of the message and of the associated data are required for the security of CCM. All the attacks do apply to CCM and to the modified version that we propose, but we focus on the consequences for CCM, since CCM is standardized. We exhibit three attacks against the authenticity of the scheme. We, among others, worry about the "non-repeating" feature of the nonces. In a two party setting, it is easy to check such requirement since the two parties can maintain a counter. However, when several parties want to communicate to each other using the same key, it is difficult to maintain a global variable distributed among the participants. Consequently, the security of CCM with random nonces is an important issue.

In our first attack, we show a generic attack that requires $2^{(\ell+t)/2} + 2^{\ell}$ encryption messages, where $\ell$ is the nonces length and $t$ is the length of the MAC. This attack is more theoretical than practical but it shows that the expected security bound of $2^n$ cannot be reached with random nonces.

Our second attack shows that when random nonces are used and when the length of the associated data is missing, $2^{\ell/2}$ encryption queries allows to forge a valid encrypted message (note that in practice $\ell < n$). It implies that if one want to remove the length of associated data to be able to preprocess static associated data, then one decreases CCM security under the proven birthday paradox bound.

Finally, our third attack shows that if random nonces are used and if the length of the message is not included in the padding function, then the authenticity of the scheme can be broken using $2^{2\ell/3}$ queries. It implies that if $\ell \leq 3n/4 = 96$ (which is realistic) and one wants to be able to make on-line encryption, then one decreases the security of CCM under the birthday paradox bound once more.

These attacks show that the security of CCM relies on the non-repeating nonce property and on the length of the message and of the associated data that is added before the message and make them prefix-free. This property is very useful to design secure encryption and authenticated schemes.

### 1.3 Organization

In section 2, we describe the CCM authenticated encrypted scheme. Then, we show our security proof beyond the birthday paradox for the authenticity and privacy in section 3. In Section 4, we describe some attacks that show why the non-classical assumptions, non-repeating nonces and prefix-free messages are important.

## 2 Security Notions

In the sequel, we briefly recall the basic security notions for blockciphers, pseudorandom functions, and symmetric encryption schemes. For the latter, we are interested into the integrity (we indistinctly use the words integrity and authentication in the sequel) and the privacy. The definitions we use are derived from [2, 5].

**Conventions** When an adversary $A$ can interact with an oracle $\mathcal{O}$ and at the end of the interaction outputs $b$, it is denoted by $A^{\mathcal{O}} \Rightarrow b$. If $B$ and $C$ are two events, the probability that the event $B$ occurs, knowing the event $C$ is denoted by $\Pr[B|C]$. When an adversary is involved in an event, the probability is considered upon the adversary random coins.

Let $\mathcal{S}$ be a set of bit strings and let $x$ and $x'$ be a couple of bit strings from $\mathcal{S}$, we denote by $x \subset x'$ the fact that $x$ is a prefix of $x'$. The set $\mathcal{S}$ is prefix-free if for all couples $(x, x') \in \mathcal{S}^2$, $x \subset x'$ implies that $x = x'$. An adversary is said prefix-free if the set of the queries that it made to all the oracles, forms a prefix-free set. Finally, we denote by $\mathsf{lsb}_k(x)$ the $k$ least significant bits of $x$.

### 2.1 Pseudorandom Functions and Pseudorandom Permutations

**Pseudorandom Permutations** Let $E\colon \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ be a permutation family. We denote by $\mathcal{S}_n$ the set of all the permutations from $\{0,1\}^n$ to $\{0,1\}^n$. The goal of a prp-adversary $\mathcal{A}$, which runs in time $T$, against $E$ is to guess the value of $b$ in the following game. The challenger chooses a bit $b$ at random; if $b = 1$ he assigns $\pi$ to a random permutation from $\mathcal{S}_n$ otherwise he chooses a random key $K$ in $\{0,1\}^k$ and assigns $\pi$ to $E(K, \cdot)$. The adversary can interact with $\pi$ making up to $q$ queries $x_i$ and receives $\pi(x_i)$. The prp-advantage of $\mathcal{A}$, denoted $\mathsf{adv}_E^{\mathtt{prp}}(\mathcal{A})$, is:

$$\left|\Pr\left[\mathcal{A}^{E(K,\cdot)} \Rightarrow 1 | K \xleftarrow{\$} \{0,1\}^k\right] - \Pr\left[\mathcal{A}^{\pi(\cdot)} \Rightarrow 1 | \pi \leftarrow \mathcal{S}_n\right]\right|.$$

**Pseudorandom Functions** Let $F\colon \{0,1\}^k \times \mathcal{D}om \to \{0,1\}^t$ be a function family. We denote by *Rand* the set of all the functions from $\mathcal{D}om$ to $\{0,1\}^t$. The goal of a prf-adversary $\mathcal{A}$, which runs in time $T$, against $F$ is to guess the value of $b$ in the following game. The challenger chooses a bit $b$ at random; if $b = 1$

he assigns $f$ to a random function from *Rand* otherwise he chooses a random key $K$ in $\{0,1\}^k$ and assigns $f$ to $F(K,\cdot)$. The adversary can interact with $f$ making up to $q$ queries $x_i$ and receives $f(x_i)$. The prf-advantage of $\mathcal{A}$, denoted $\mathsf{adv}_F^{\mathtt{prf}}(\mathcal{A})$, is:

$$\left|\Pr\left[\mathcal{A}^{F(K,\cdot)} \Rightarrow 1 | K \xleftarrow{\$} \{0,1\}^k\right] - \Pr\left[\mathcal{A}^f \Rightarrow 1 | f \leftarrow \mathcal{F}_{d,n}\right]\right|.$$

If $\mathcal{A}$ is restricted to be prefix-free then its prf-advantage is called pf-prf-advantage and is denoted $\mathsf{adv}_F^{\mathtt{pf\text{-}prf}}(\mathcal{A})$.

### 2.2 Integrity

The security notion we use to define authenticity for a symmetric encryption scheme is the integrity of the ciphertext (denoted INT-CTXT). Formally, in the integrity game, the adversary $\mathcal{A}$ is given access to an encryption oracle $\mathcal{E}(\cdot)$ and a verification oracle $\mathcal{VO}(\cdot)$ it can feed with queries of his choice. The encryption oracle encrypts the plaintext and answers by the corresponding ciphertext. The adversary feeds the verification oracle with a ciphertext, also called forgery attempt in the sequel, and the oracle answers 1 if the ciphertext is valid and 0 otherwise. The adversary goal is to generate a valid ciphertext (that is accepted by the verification oracle) which is different from all ciphertexts previously generated by the encryption oracle. Note that the adversary can send several queries to the verification oracle. The success probability of $\mathcal{A}$ is:

$$\mathsf{Succ}_{CCM}^{\mathtt{int\text{-}ctxt}}(\mathcal{A}) = \Pr[\mathcal{VO}(C) \Rightarrow 1 | \mathcal{A}^{\mathcal{E}(\cdot),\mathcal{VO}(\cdot)} \Rightarrow C].$$

### 2.3 Privacy

The security notion used to define privacy for a symmetric encryption scheme is the indistinguishability security under chosen plaintext attacks (denoted IND-CPA). Formally, in the privacy game an adversary $\mathcal{A}$ is given access to an encryption oracle $\mathcal{E}(\cdot)$ it can feed with queries of the form $(M_0, M_1)$ where $M_0$ and $M_1$ are messages of his choice. At the beginning of the game this oracle chooses a bit $b$ and always encrypts the message $M_b$. The adversary's goal is to guess $b$, that is to say to distinguish the two cases. The indistinguishability is defined in the "left or right model" which has been introduced and proved to be the strongest one in [2]. The advantage of $\mathcal{A}$ is:

$$\mathsf{adv}_{CCM}^{\mathtt{ind\text{-}cpa}}(\mathcal{A}) = \left|\Pr[\mathcal{A}^{\mathcal{E}(\cdot)} \Rightarrow 1 | b = 1] - \Pr[\mathcal{A}^{\mathcal{E}(\cdot)} \Rightarrow 1 | b = 0]\right|.$$

## 3 CCM description

In this part, we describe the original CCM authenticated encryption mode and the format of its various inputs. In [21] recommendations are also given on various choices that have to be made to implement CCM: unique key for both the CBC chain and the counter mode, nonces that cannot be repeated... Some of these restrictions can be ignored without any security problems although some other are needed for security reasons. At the end of this part we discuss these choices.

### 3.1 Notations

In this paper, the following notations will be used:

- for any string or integer $x$, $|x|_2$ denotes its bit length, $|x|_8 = \left\lceil \frac{|x|_2}{8} \right\rceil$ its octet length, and $[x]_s$ denotes the binary representation of $x$ on $s$ bits;
- $E$ is a block cipher with $n$-bit blocks and $k$-bit keys, where $n = 128$;
- $M$ is a plaintext, consisting of blocks of $n$ bits denoted $M_1, \ldots, M_{m-1}$ and a last block $M_m$ with at most $n$ bits.
- the associated data (data which is authenticated and not encrypted) is denoted by $D_1, \ldots, D_a$ and consists in $a-1$ blocks of $n$ bits and one block of at most $n$ bits;
- the ciphertext $C$ consists in $m+1$ blocks $C_0, C_1, \ldots, C_m$ where $C_i$ is $n$-bit long for $0 \leq i \leq m-1$ and $C_m$ is at most $n$ bits;
- $B = B_0, B_1, \ldots, B_r$ is the $n$-bit long formatted input used for the CBC-MAC computation, $B_0$ is called the pre-initial value;
- $A_0, A_1, \ldots, A_m$ are the inputs for the counter mode;
- the nonce used to derive the pre-initial value $B_0$ and the counter values $A_0, A_1, \ldots, A_m$ is denoted by $N$. This nonce is $\ell$-bit long, with $7 \leq \ell/8 \leq 13$ ($\ell$ has to be divisible by 8);
- $q$ is an integer such that $2 \leq q \leq 8$ and $q + \ell/8 = 15$, let $Q$ denotes the bit representation of the octet length of the message $M$ over $q$ octets, *i.e.* $Q = \left[|M|_8\right]_{8q}$;
- $t$ denotes the bit length of the MAC, it has to be divisible by 16, and $4 \leq t/8 \leq 16$.

### 3.2 CCM mode

The CCM mode can be basically viewed as an authenticate-then-encrypt composition instantiated with a CBC-MAC and a counter mode. The mode uses a block cipher $E$ both in the CBC chain and in the counter mode. The block length is equal to $n = 128$ bits. We denote by $K$ the key used in the CBC-MAC and by $K'$ the one used in the counter mode. The choice of $K$ and $K'$ is discussed in section 3.4.

Let $M = M_1 \| M_2 \| \ldots \| M_m$ be a plaintext and $D = D_1 \| D_2 \| \ldots \| D_a$ associated data that will only be authenticated and not encrypted.

At first, the encryption box chooses a nonce $N$ of $\ell$ bits. This nonce will be used to derive both the pre-initial value for the CBC-MAC and the counter blocks.

In a first step, the associated data and the plaintext blocks are authenticated. This consists in computing their CBC-MAC value. This computation is however quite different from the classical one: it authenticates a formatted input $B = B_0 \| \ldots \| B_r$ derived from $N$, $M$, and $D$. The format of $B$ is described in section 3.3. The 1-block pre-initial value $B_0$ is treated as the first message block

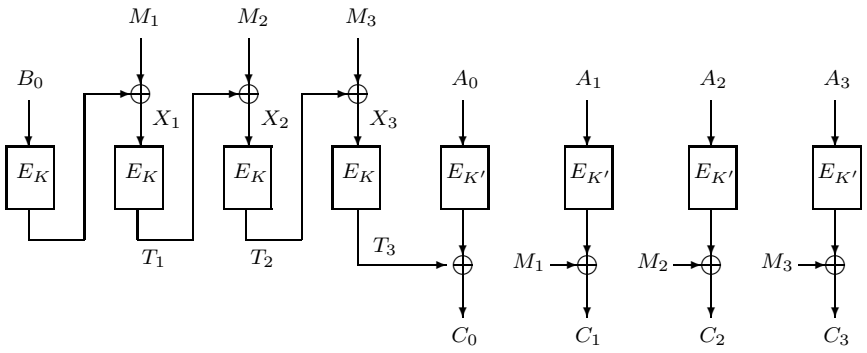


**Fig. 1.** CCM authenticated encryption mode.

in the CBC chain. Its main property is that it contains the tag length, the plaintext length and the nonce value. The CBC-MAC is a simple CBC chain without retail. Its output is denoted $T$ and is $t$-bit long, with $32 \leq t \leq n$.

In a second step, the MAC value $T$ and the plaintext $M$ are concatenated and encrypted with a counter mode. The inputs for the counter mode are blocks of $n$ bits denoted $A_0 \| A_1 \| \ldots \| A_m$. Their format is described in section 3.3. Briefly, each one contains some flag information, the nonce value, and the index of the plaintext block. The tag $T$ is encrypted as $C_0 = \mathsf{lsb}_t\left(E_{K'}(A_0)\right) \oplus T$ and for all plaintext blocks $M_i$, $1 \leq i \leq m$, $C_i = E_{K'}(A_i) \oplus M_i$.

The ciphertext $C = C_0 \| C_1 \| \ldots \| C_m$ and the associated data $D$ are then transmitted. The nonce value is transmitted if necessary (in case of non synchronizing).

Figure 1 describes the CCM mode for 3-block plaintexts and no associated data.

The decryption process consists in decrypting $C$ with the counter mode and then to compute the tag $T'$ with the formatted input $B$ computed from the nonce $N$, the associated data and the recovered plaintext. If valid, the plaintext is returned. Otherwise an error message is given. For a description of CCM encryption and decryption algorithms in pseudo-code, see appendix A.

### 3.3 Formatting the inputs

The specification [21] precisely describes the format for the different inputs.

The CBC-MAC chain uses a formatted input $B = B_0, \ldots, B_r$. The $n$-bit pre-initial value $B_0$ is determined by a nonce of $\ell$ bits, and various information. The first octet is a flag one containing one bit for future extension, one bit to indicate whether associated data are present or not, three bits to indicate the octet length of the MAC value (which is necessarily different from 000) and three bits for the octet length of the binary representation of the octet length of the plaintext $M$. The remaining octets contain the nonce value followed by

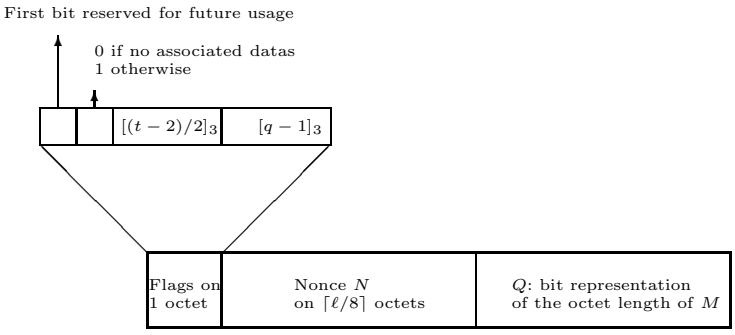


**Fig. 2.** The format of the pre-initial value $B_0$ for the CBC-MAC.

the value $q$, the bit representation of the octet length of $M$. Picture 2 represents the format of $B_0$. Note that a collision on the $B_0$ values occurs if and only if the nonces collide, associated data are either used or not for both and the plaintexts are of the same octet length.

If there are authenticated data then let $B_1 \| \dots \| \mathcal{B}_u$ be the concatenation of a particular encoding of the $D$ size (for more details see [21]), of $D$, and of as few '0' as possible such that the resulting bit string can be partitioned into $n$-bit blocks (if there is no authenticated data $B_1 \| \dots \| \mathcal{B}_u$ is the empty bit string). Let $B_{u+1} \| \dots \| \mathcal{B}_r$ be the concatenation of $M$ and of as few '0' as possible such that the resulting bit string can be partitioned into $n$-bit blocks. Remark that the encoding of $B$ is made in such a way that the set of all possible formatted inputs $B$ is prefix-free !

Finally, the inputs for the counter mode $A_0, A_1, \dots, A_m$ are encoded as follows: the first octet contains some flag information (2 bits reserved for a future usage, three bits fixed to 000 and three bits containing the binary representation of $q-1$). The remaining ones contain the nonce value, already used for formatting $B_0$ and the block number.

### 3.4 NIST requirements

The CCM specification [21] gives indications to guide implementation choices. Of course, CCM should be used with the AES so the key length is either 128, 192, or 256 bits. The block length is 128 bits.

The CCM specification also provides a requirement for the key choice. Indeed, the same key should be used for both the CBC-MAC and the counter mode. Such a choice is of course debatable since it goes against the common sense based on the key usage separation. However, the security proof given by Jonsson in [12] is made for this case and ensures that CCM provides privacy and authenticity up to $2^{n/2}$ block cipher calls. Thus, choosing the same key for the two modes combined in CCM is not a security concern since the security bound is very close to the one given for a lot of other authenticated encryption modes [18, 13].

However, even if this requirement is quite understandable in case of same security results, it becomes cumbersome if the security with two keys is much better than for a single key. In our modified CCM we can achieve a much better security with two keys. That is why in section 4, we focus on the security results for our modified CCM in case of non repeating nonces with two different keys.

Another requirement made in the CCM specification concerns the choice of the nonce values. It is explicitly defined that *"the counter blocks must be distinct within a single invocation and across all other invocations of the CTR mode under any given key"*. This is done by imposing non-repeating nonces. This requirement is here largely understandable: indeed, a collision on the nonce values can often be exploited to forge a valid ciphertext or to attack the privacy of the scheme. However, in practice, non repeating nonces could be very difficult to handle, particularly in a scenario where three or more users share the same key used with CCM. Thus, it can be interesting to carefully look at the CCM security when nonces are randomly chosen and can thus collide. This is done in section 5.

# 4 Modified CCM

## 4.1 Description

In this section we propose a modified version of CCM (mCCM) which we prove secure beyond the birthday paradox bound. The main difference between the original and the modified CCM versions is the use of a pseudorandom function to encrypt the tag and the message. Let $F$ be a pseudorandom function family from $\{0,1\}^n$ to $\{0,1\}^n$, $E$ a blockcipher over $\{0,1\}^n$ and let $K$ and $K'$ be two keys chosen independently.

To encrypt a message $M, D$, a nonce is chosen, the corresponding formatted input is deduced and a CBC-MAC tag $T$ is computed with the blockcipher $E_K$. Then, the ciphertext is computed as $C_0 = T \oplus \mathsf{lsb}_t\left(F_{K'}(A_0)\right)$ and $C_i = M_i \oplus F_{K'}(A_i)$.

The decryption process consists in decrypting $C$ with the counter mode and then to compute the tag $T'$ with the formatted input $B'$ computed from the nonce $N$, the associated data and the recovered plaintext. If valid, the plaintext is returned. Otherwise an error message is given.

We remind that in this modified version, as in the original version, we impose non repeating nonces. This implies that adversaries against modified CCM can choose the nonce to encrypt a query, as soon as, any new nonce is different from all the previous one. However, for the verification queries the adversary is allowed to use a nonce which was already used in a previous encryption query.

In the following we prove the IND-CPA and INT-CTXT security of this modified version of CCM. Note that as proven in [5] these two security notions imply the IND-CCA security (with a tight reduction), which means that this protocol achieves the best security levels for privacy and for integrity (see [5] for precise definitions and relations between these notions).

### 4.2 Privacy

The modified CCM privacy is a direct consequence of the privacy of CTR using a pseudorandom function. This security result has been stated in [2]:

**Theorem 1 (BDJR).** *Suppose $F$ is a PRF family from $\{0,1\}^n$ to $\{0,1\}^n$. Then, for any adversary $A$ against privacy of CTR mode, with running-time $T$, and which can do at most $q_e$ encryption queries of at most $s$ blocks, there exists a prf-adversary $A'$ against $F$ with running time $T$ and which can make at most $sq_e$ queries, such that:*

$$\mathsf{adv}^{\texttt{ind-cpa}}_{CTR}(\mathcal{A}) \leq \mathsf{adv}^{\texttt{prf}}_{F}(\mathcal{A}').$$

The security of mCCM is an easy consequence of this theorem:

**Theorem 2.** *Suppose $F$ is a PRF family from $\{0,1\}^n$ to $\{0,1\}^n$. Then, for any adversary $\mathcal{A}$ against privacy of modified CCM mode, with running-time $T$, and which can do at most $q_e$ encryption queries of at most $s$ blocks, there exists a prf-adversary $\mathcal{A}'$ against $F$ with running time $T$ and which can make at most $(s+1)q_e$ queries, such that:*

$$\mathsf{adv}^{\texttt{ind-cpa}}_{mCCM}(\mathcal{A}) \leq \mathsf{adv}^{\texttt{prf}}_{F}(\mathcal{A}').$$

### 4.3 Integrity

To prove the integrity of ciphertexts (INT-CTXT) in modified CCM, we need the two following results. The first one is shown in [6] and upper bounds the advantage of a pf-prf adversary against CBC-MAC.

**Theorem 3 (BPR).** *Let $\mathcal{A}$ be a prefix-free prf-adversary against the $n$-bit block CBC-MAC, $\mathcal{A}$ can make at most $q \geq 2$ queries of at most $s$ blocks and has a running-time of at most $T$. Then we have:*

$$\mathsf{adv}^{\texttt{pf-prf}}_{CBC\text{-}MAC}(\mathcal{A}) \leq \frac{sq^2}{2^n}\left(12 + \frac{64s^3}{2^n}\right).$$

The second result comes from [3] and shows that if a protocol is INT-CTXT secure against an adversary which can make at most one verification query, then it is secure against adversaries which can make several verification queries.

**Lemma 1.** *Let $\mathcal{A}$ be an adversary against the authenticity of a symmetric encryption schemes $\Pi$. Assume that $\mathcal{A}$ makes at most $q_e$ encryption queries and $q_v$ verification queries all of at most $s$ blocks. Then there exists an adversary $\mathcal{A}'$ against the authenticity of $\Pi$, such that $\mathcal{A}'$ makes at most $q_e$ encryption queries and 1 verification queries, all of at most $s$ blocks and:*

$$\mathsf{Succ}^{\texttt{int-ctxt}}_{\Pi}(\mathcal{A}) \leq q_v \cdot \mathsf{Succ}^{\texttt{int-ctxt}}_{\Pi}(\mathcal{A}').$$

*The adversaries $\mathcal{A}$ and $\mathcal{A}'$ have the same running-time.*

Thanks to these results, we can upper bound the advantage of any adversary against the INT-CTXT of mCCM.

**Theorem 4.** *Let $\mathcal{A}$ an adversary against the authentication of mCCM, with running-time at most $T$, which can make at most $q_e$ encryption queries of at most $s$ blocks and $q_v$ verification queries of at most $s$ blocks. Then its success probability is upper bounded by:*

$$\mathsf{Succ}^{\texttt{int-ctxt}}_{mCCM}(\mathcal{A}) \leq q_v \left( \frac{48(s+1)}{2^n} + 256 \left( \frac{(s+1)^2}{2^n} \right)^2 + \frac{1}{2^t} \right) + \mathsf{adv}^{\texttt{prf}}_F(\mathcal{A}_1) + \mathsf{adv}^{\texttt{prf}}_E(\mathcal{A}_2),$$

*where $A_1$ is a prf-adversary against $F$ with running-time at most $T$, which can make at most $(s+1)(q_e+q_v)$ queries and $A_2$ is a prp-adversary against $E$ with running-time at most $T$, which can make at most $(s+1)(q_e+q_v)$ queries.*

*Proof.* The following proof is a game-based proof. In the first game (game 1) the adversary faces the verification and encryption oracles which are simulated respecting strictly the protocol. In each new game we alter a bit the way we simulate the two oracles, so that in the last game we are able to upper bound the adversary success probability. Since we alter the simulation between two games, the adversary success probability is modified, so we have to upper bound this modifications; this upper bound is called the distance between two games. See [20] for details.

In the games 2 and 3 we replace successively the PRF $F$ and the PRP $E$ with respectively a true random function and a true random permutation. One can easily shows that there exists two adversaries $A_1$ and $A_2$ as stated in the theorem such that the distances between the games can be upper bounded respectively by $\mathsf{adv}^{\texttt{prf}}_F(\mathcal{A}_1)$ and $\mathsf{adv}^{\texttt{prf}}_E(\mathcal{A}_2)$.

Let $\mathcal{A}_3$ be the adversary against the authenticity of mCCM in the game 3, it can make $q_e$ encryption queries and $q_v$ verification queries, all of at most $s$ blocks. Let $\mathcal{A}'$ be an adversary against the authenticity of mCCM which makes $q_e$ encryption queries of at most $s$ blocks and 1 verification query of at most $s$ blocks such that $\mathsf{Succ}^{\texttt{int-ctxt}}_{mCCM}(\mathcal{A}_3) \leq q_v \cdot \mathsf{Succ}^{\texttt{int-ctxt}}_{mCCM}(\mathcal{A}')$ (it exists thanks to lemma 1). To upper bound the success probability of $\mathcal{A}_3$, we upper bound the success probability of $\mathcal{A}'$. For this, thanks to $\mathcal{A}'$ we construct $\mathcal{D}$ a prefix-free prf-adversary against CBC-MAC with 2 MAC queries of at most $s$ blocks, and then relate the success probability of $\mathcal{D}$ with the one of $\mathcal{A}'$.

As described in the prf security definition, the prf-distinguisher $\mathcal{D}$ faces a CBC-MAC oracle. To construct $\mathcal{D}$, we run $\mathcal{A}'$ and to every of its encryption query $(N^i, M^i, D^i)$, we answer:

- $\perp$ if there exists $k < i$ such that $N^i = N^k$,
- $(N^i, D^i, C^i = C^i_0 \| \dots \| C^i_{m_i})$ with $C^i_0 \xleftarrow{\$} \{0,1\}^n$, $C^i_k = M^i_k \oplus F(A^i_k)$.

To $\mathcal{A}'$ verification query $(N, D, C = C_0 \| \dots \| C_m)$, we answer:

– if $N \neq N^k$ for all $k \leq q_e$, then we choose randomly a $t$-bit string $T$ and compute the $m$-block message $M$ with $M_i = C_i \oplus F(A_i)$ ; we give $B$, where $B$ is the corresponding formatted input, to the CBC-MAC verification oracle which answers with $T'$ and we reply to $\mathcal{A}'$ with 1 if and only if $T = \mathsf{lsb}_t(T')$,
– if there is $k \leq q_e$ such that $N = N^k$, $D = D^k$, and $C = C^k$, then we answer $\perp$,
– if there is $k \leq q_e$ such that $N = N^k$, but $D \neq D^k$ or $C \neq C^k$, then, we compute the message blocks $M_i = F(A_i) \oplus C_i$, deduce the corresponding formatted input $B$; we send the $k^{\text{th}}$ formatted input $B^k$ (from the $k^{\text{th}}$ encryption query) to the CBC-MAC oracle and receives the corresponding tag $T^k$; then we compute the tag of $B$: $T = \mathsf{lsb}_t\left(T^k\right) \oplus C_0 \oplus C_0^k$; finally we send $B$ to the CBC-MAC oracle, receives back $T'$, check if $T = \mathsf{lsb}_t(T')$ and forward the answer to $\mathcal{A}'$ (Note that since $D \neq D^k$ or $C \neq C^k$, $B \neq B^k$ and since the formatted inputs form a prefix-free set, the two queries made to the CBC-MAC oracle are prefix-free).

At the end, $\mathcal{D}$ decides that it faces a true CBC-MAC oracle if the answer to the $\mathcal{A}'$ verification query is equal to 1.

As soon as the nonces are different from each other, for a mCCM attacker the $C_0^i$ are randomly distributed, therefore the answers to $\mathcal{A}'$ are well simulated. Since there is no collision between the nonces, the probability of success of $\mathcal{A}'$ is exactly the probability that $\mathcal{D}$ outputs 1 when it faces a true CBC-MAC oracle. When $\mathcal{D}$ faces a random function, its success probability is $1/2^t$, therefore we have:

$$\mathsf{Succ}^{\texttt{int-ctxt}}_{mCCM}(\mathcal{A}') \leq \frac{1}{2^t} + \mathsf{adv}^{\texttt{pf-prf}}_{CBC\text{-}MAC}(\mathcal{D}).$$

We remind that pf in pf-prf stands for prefix-free. Theorem 3 allows us to conclude. □

In practice, we can consider that $s \leq 2^{40} - 1$ (in fact this is probably still a large upper bound of $s$). In the following, we omit the PRF and PRP advantages for simplicity reasons, since these terms are identical in the two bounds. Let assume that $t \geq 82$, previous theorem gives an upper bound of the integrity adversaries of $q_v \cdot 2^{-80}$. For the same values, Jonsson theorem [12] gives a security of approximately $(q_v + q_e)^2 \cdot 2^{-48}$. Remark that for a value of $t = 82$, our bound is tight since the simple attack which consists in trying to guess the CBC-MAC value has a success probability of $q_v/2^t = q_v \cdot 2^{-82}$.

## 5 Random nonces

In this part we consider that the nonces used for the CCM mode are chosen at random in $\{0,1\}^\ell$. In this case, collision between two random values are possible and such an event can be exploited by an adversary to forge a valid ciphertext. Of course, confidentiality cannot be ensured as soon as two nonces collide. Indeed, if such a collision occurs, the adversary can easily distinguish which plaintext is encrypted and then break the scheme in the sense of semantic security. However, forging a valid ciphertext, *i.e.* breaking the ciphertext unforgeability, is a different

task. Attacking the privacy is independent and does not imply any weakness on the integrity, even if, when one is compromised, the other often too. However, the technique used here to forge a valid ciphertext is completely different from the one used to break the semantic security in case of collision between nonces.

Note that the following attacks do apply to both original CCM and modified CCM, as long as the nonces are random. We focus our analyze to CCM because it is a standard, but the same conclusions could be stated for the modified version of CCM. Besides, remark that the security model is slightly different from previous section. As the nonce may repeat, the adversary can make as many queries as it wants, whereas in previous section the adversary was restricted to $2^\ell$ encryption queries, the number of possible nonces. However, when in previous context the nonces used for encryption queries can be chosen by the adversary, as long as there are all distinct, in this context the nonces are chosen randomly by the challenger.

### 5.1 Generic Attack

We present in this subsection a generic attack against original and modified CCM, assuming only that the nonces may collide. The complexity of this attack is $\mathcal{O}(2^\ell + 2^{(t+\ell)/2})$ encryption queries and one verification query, where $t$ is the tag bit size and $\ell$ the nonce bit size, and its success probability is very close to 1. Let $M_3$ be a message block and $(M_1^i, M_2^i)_i$ be a set of $2^{(t+\ell)/2}$ 2-block messages. The adversary makes the $2^{(t+\ell)/2}$ encryption queries $(M_1^i, M_2^i, M_3)$ and receives the answer $N^i, (C_0^i, C_1^i, C_2^i, C_3^i)$. With high probability, there are two different indexes $i$ and $k$ such that $N^i = N^k$ and $C_0^i = C_0^k$. Since the nonces are the same, the counter values used for the encryption are the same and thus, since the encryptions of the tags are the same, there is a collision between the two tags. Since the last block of the two messages are the same, it means that the collision appears in fact before the last block and thus $CBC(B_0\|M_1^i\|M_2^i\|M_3') = CBC(B_0\|M_1^k\|M_2^k\|M_3')$ for any $n$-bit block $M_3'$ (note that the collision between the nonces implies a collision between the $B_0$ values since the plaintexts are of the same octet length). Let $M_3' \neq M_3$ and let repeat $\mathcal{O}(2^\ell)$ times the encryption query $M_1^i\|M_2^i\|M_3'$ until the answer involves the nonce $N^i$. Let denote $C_0, C_1, C_2, C_3$ the corresponding ciphertext, and let $C_0' = C_0$, $C_1' = C_1 \oplus M_1^i \oplus M_1^k$, $C_2' = C_2 \oplus M_2^i \oplus M_2^k$, and $C_3' = C_3$. The ciphertext $N^k, (C_0', C_1', C_2', C_3')$ is valid for the message $M_1^k\|M_2^k\|M_3'$ (message which was not previously asked).

Note that in the case when $t \leq \ell$ this attack requires $\mathcal{O}(2^\ell)$ encryption queries, 1 verification query, all of at most 3 blocks. Therefore if theorem 4 would apply, the success probability of such an adversary would be upper bounded by $2^{9-n} + 2^{-t}$, whereas it is nearly equal to 1. If this attack is not practical, it illustrates the fact that allowing random nonces strongly decreases the security of CCM.

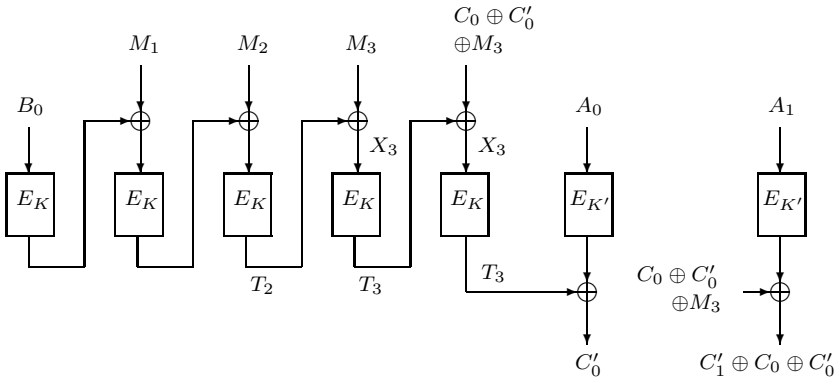


**Fig. 3.** Forgery attempt: $C_0 \oplus C'_0$ equals $T_2 \oplus T_3$.

### 5.2 If the Data Length was not Precised

In previous attack we have relieved the constraint that nonces should not collide. In the two following attacks, we still allow the nonces to collide and in addition we assume that the formatted inputs do not form a prefix-free set. We show that, in this case, the attacks can be even worse than the previous one.

We remind that in CCM, $B_0$ depends on the message length but not of the associated data length (however if there are associated data, a flag bit is set to 1). If there are associated data, their length is encoded at most on the 10 first blocks of $B$. Since the length of the associated data must be known to authenticate, the authentication cannot be done online and this reduce CCM efficiency. For the following attack we retrieve this constraint, assuming that the associated data length is not concatenated *before* the associated data themselves (it could be encoded after the associated data even if for simplicity we just skip it). The attacker will use this remark to forge a valid ciphertext from $2^{\ell/2}$ encryption queries.

We assume for the attack that associated data are always used, so that the flag bit used in $B_0$ is always set to 1. This attack exploits some relations he can deduce from the collision on the nonce values: the attacker queries the encryption oracle for messages of two blocks and three blocks. In both cases, only the last block is encrypted. He thus collects ciphertext $(N_i, C_0^i \| C_1^i)$ corresponding to the authentication of $M_1 \| M_2$ and the encryption of $M_2$ under different nonces, and ciphertexts $(N'_j, C_0^j \| C_1^j)$ corresponding to the authentication of $M_1 \| M_2 \| M_3$ and the encryption of $M_3$ under different nonces. By the birthday paradox, after $2^{\ell/2}$ queries, there is a collision between two nonces, one used for a query in the first set and the other for a query in the second set. Thus, there exists $i$ and $j$ such that $N_i = N_j$. Since plaintexts to authenticate and encrypt are of the same length in both cases, we also have $B_0^i = B_0^j$. For simplicity, the corresponding ciphertexts are denoted $(N, C_0 \| C_1)$ and $(N, C'_0 \| C'_1 \| C'_2)$.

The attacker can now compute the value

$$C_0 \oplus C'_0 = T \oplus T'$$

where $T$ is the CBC-MAC value computed for the first message and $T'$ is the CBC-MAC value for the second. He thus can forge a valid ciphertext for the message $M_1\|M_2\|M_3\|M_3 \oplus C_0 \oplus C'_0$ where only the last block is encrypted. The corresponding ciphertext is thus $(N, C'_0\|C'_1 \oplus C_0 \oplus C'_0)$ with the associated data $M_1\|M_2\|M_3$. Figure 3 resumes this forgery.

The complexity of the attack is $\mathcal{O}(2^{\ell/2})$ encryption queries. As $\ell \leq n$, it means that relieve the constraint on the format of the authenticated data would lead to better attacks than the birthday paradox bound proved by Jonsson.

### 5.3 Random nonces with inputs not formatting

Finally, in next attack we consider the case where the nonces are random and the message length is not put in the pre-initial formatted input $B_0$. This way CCM computation can be done online, but however we show that in this case, even without additional authenticated data, the security of CCM decreases.

The attacker will make 3 kinds of queries: the first ones are composed with a plaintext of a single block denoted $M_1$. This block is the same for all the queries. The second kind of queries is composed with messages of two blocks, $M_1\|M_2$ where $M_1$ is the same block as the one chosen for the first queries. Finally, in the third set of queries, messages are composed with 3 blocks $M_1\|M_2\|M_3$ where here again $M_1$ and $M_2$ are the same as before. The attackers queries the encryption oracle for these messages until a collision occurs between the nonces used for one message in each set. Thus, there exists integers $i, j, k$ such that : $N_i = N_j = N_k$ and thus $B_0^i = B_0^j = B_0^k$ and $A_0^i = A_0^j = A_0^k$ (since $B_0$ does not depend on the message length anymore). The attacker is now able to forge a valid ciphertext. We denote by $(N, C_0^1\|C_1^1)$ the corresponding ciphertext for the one block message, $(N, C_0^2\|C_1^2\|C_2^2)$ the ciphertext for the two blocks message and $(N, C_0^3\|C_1^3\|C_2^3\|C_3^3)$ the ciphertext for the three blocks message. Due to the choice of the message blocks $M_1$ and $M_2$ and since the nonces collide for these three encryptions, we remark that $C_1^1 = C_1^2 = C_1^3$ and $C_2^2 = C_2^3$. The first ones are briefly denoted by $C_1$ and the second by $C_2$. We also denote by $T_1$ the CBC-MAC value for $M_1$, $T_2$ the one for $M_1\|M_2$, and $T_3$ the one for $M_1\|M_2\|M_3$. These notations are given in figure 1.

Due to the collision between the nonces used for these three encryptions, the value $C_0^1 \oplus C_0^2$ is equal to $T_1 \oplus T_2$. Although the attackers does not know the values $T_1$ and $T_2$, he can exploit the knowledge of their bit-wise addition to forge a valid ciphertext : indeed, the ciphertext $C_0^3\|C_1\|C_2 \oplus M_2 \oplus C_0^1 \oplus C_0^2 \oplus M_3$ is valid for the plaintext $M_1\|C_0^1 \oplus C_0^2 \oplus M_3$ and the nonce $N$. The input to the third encryption box in the CBC chain is the bit-wise addition of the plaintext block $C_0^1 \oplus C_0^2 \oplus M_3$ and the previous output $T_1$. Since $C_0^1 \oplus C_0^2 = T_1 \oplus T_2$, the input is just $T_2 \oplus M_3$, that is to say the input to the fourth encryption box in the CBC for the encryption of $M_1\|M_2\|M_3$. Thus, the output is $T_3$ (unknown to

the adversary) and the first ciphertext block is $C_0^3$. The second ciphertext block is easily computable from $C_2$ due to the malleability of the counter mode.

We now estimate the complexity of this attack and in particular the average number of queries needed. The attackers queries the encryption oracle until a collision appears between the nonces for three of them. By the birthday paradox, a collision occurs for two encryption queries when $2^{\ell/2}$ nonces have been chosen at random. After $2^{2\ell/3}$ queries for $M_1$ and $M_1\|M_2$, there are in average $2^{4\ell/3}/2^{\ell}$ collisions, *i.e.* $2^{\ell/3}$ pairs of ciphertexts computed with the same nonce. If we consider this set of ciphertext pairs and the set of $2^{2\ell/3}$ ciphertexts for $M_1\|M_2\|M_3$, there are in average triplet of ciphertexts computed with the same nonce. In a general case, if $S_i$ is the number of ciphertext in each set, there is a 3-collision on the nonces used if and only if $S_1 \times S_2 \times S_3$ equals $2^{\ell}$. Choosing $S_i = 2^{2\ell/3}$ is the best compromise. Finally the attacker can forge a valid ciphertext with the help of $3 \times 2^{2\ell/3}$ encryption queries. If $\ell \leq 96 = 3n/4$ then $2\ell/3 \leq n/2$ and this attack is better than the security bound given by Jonsson.

This attack illustrates the fact that if one wants to preserve the security, one cannot increase CCM efficiency removing the particular format, with the message length appended to the beginning of the message, and allowing repeating nonces.

## 6 Conclusion

In this paper we have studied the security of the CCM authenticated encryption scheme and of a modified version. We have shown that slightly modifying CCM one can prove the $\mathcal{O}(2^n)$ security for authenticity, security which only conjectures for CCM. Additionally, the modified version of CCM provably guarantees an optimal security for privacy.

Besides, we have studied the CCM (and also modified CCM) properties that restrict its efficiency. We exhibit that if we relieve some of them (if we let nonces collide, if we break the prefix-freeness of authenticated messages removing the message and/or authenticated data length) then one can mount attacks which are better than the expected or proved security bound by Jonsson.

## Acknowledgment

This work has been partially supported by the European Commission through the IST Program under Contract IST-2002-507932 ECRYPT.

## References

1. M. Bellare. New Proofs for NMAC and HMAC: Security Without Collision-Resistance. In *Crypto '06*, LNCS 4117. Springer-Verlag, Berlin, 2006.
2. M. Bellare, A. Desai, E. Jokipii, and P. Rogaway. A concrete security treatment of symmetric encryption. In *38th FOCS*, pages 394–403. IEEE Computer Society Press, Oct. 1997.

3. M. Bellare, O. Goldreich, and A. Mityagin. The Power of Verification Queries in Message Authentication and Authenticated Encryption. Eprint cryptology archive 2004/309. Available at `http://eprint.iacr.org`, 2004.
4. M. Bellare and R. Impagliazzo. A Tool for Obtaining Tighter Security Analyses of Pseudorandom Function Based Constructions, With Applications to PRF-¿PRP conversion. Crytology ePrint archive, Report 1999/024, available at `http://eprint.iacr.org`.
5. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In T. Okamoto, editor, *ASIACRYPT 2000*, volume 1976 of *LNCS*, pages 531–545. Springer, Dec. 2000.
6. M. Bellare, K. Pietrzak, and P. Rogaway. Improved security analyses for CBC MACs. In V. Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 527–545. Springer, Aug. 2005.
7. M. Bellare, P. Rogaway, and D. Wagner. The EAX mode of operation. In B. K. Roy and W. Meier, editors, *FSE 2004*, volume 3017 of *LNCS*, pages 389–407. Springer, Feb. 2004.
8. Y. Dodis, R. Gennaro, J. Håstad, H. Krawczyk, and T. Rabin. Randomness extraction and key derivation using the CBC, cascade and HMAC modes. In M. Franklin, editor, *CRYPTO 2004*, volume 3152 of *LNCS*, pages 494–510. Springer, Aug. 2004.
9. D. Dolev, C. Dwork, and M. Naor. Non-Malleable Cryptography. In *Proc. of the 23rd STOC*. ACM Press, New York, 1991.
10. N. M. Dworkin. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality, May 2002. NIST Special Publication 800-38C.
11. C. Hall, D. Wagner, J. Kelsey, and B. Schneier. Building PRFs from PRPs. In H. Krawczyk, editor, *Advances in Cryptology – Crypto'98*, volume 1462 of *LNCS*, pages 370 – 389. Springer-Verlag, 1998.
12. J. Jonsson. On the security of CTR + CBC-MAC. In K. Nyberg and H. M. Heys, editors, *SAC 2002*, volume 2595 of *LNCS*, pages 76–93. Springer, Aug. 2003.
13. C. Jutla. Encryption Modes with Almost Free Message Integrity. In B. Pfitzmann, editor, *Advances in Cryptology – Eurocrypt'01*, volume 2045 of *LNCS*, pages 529 – 544. Springer-Verlag, 2001.
14. J. Katz and M. Yung. Characterization of security notions for probabilistic private-key encryption. *Journal of Cryptology*, 19(1):67–95, Jan. 2006.
15. H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?). In J. Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 310–331. Springer, Aug. 2001.
16. S. Lucks. The Sum of PRP is a Secure PRF. In B. Preneel, editor, *Advances in Cryptology – Eurocrypt 2000*, volume 1807 of *LNCS*, pages 470 – 484. Springer Verlag, 2000.
17. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. In *ACM CCS 01*, pages 196–205. ACM Press, Nov. 2001.
18. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A Block-Cipher Mode of Operation for Efficient Authenticated Encryption. In *Proceedings of the 8th Conference on Computer and Communications Security*, pages 196 – 205. ACM Press, 2001.
19. P. Rogaway and D. Wagner. A Critique of CCM, February 2003. Eprint cryptology archive 2003/070. Available at `http://eprint.iacr.org`.
20. V. Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Report 2004/332, 2004.

21. N. Special Publication 800-38C. Recommendation for Block Cipher Modes of Operation: The CCM Mode for Athentication and Confidentiality, May 2004. Available at `http://csrc.nist.gov/CryptoTollkit/modes/`.
22. D. Whiting, R. Housley, and N. Ferguson. IEEE 802.11-02/001r2: AES Encryption and Authentication Using CTR Mode and CBC-MAC, March 2002.

## A CCM Encryption and Decryption Algorithms

In this section we give a description of CCM encryption and decryption algorithm in pseudo-code. For the notations see subsection 3.1. We remind that the $A_i$, the inputs for the counter mode, are derived from the nonce $N$, and the $B_i$ are derived from the size of the message, the size of the associated data, the associated data itself, and the message itself.

**Algorithm 1** CCM Encryption$(M, D)$

Choose $N$,
**function** AUTHENTICATION$(N, D, M)$
  Generates $B_0, \ldots, B_r$ from $N$, $M$ and $D$.
  $X_0 \leftarrow E(B_0)$
  **for** $i \leftarrow 1,r$ **do**
    $X_i \leftarrow E(B_i \oplus X_{i-1})$
  **end for**
  $T \leftarrow \mathsf{lsb}_t(X_r)$
  **return** $T$
**end function**
**function** ENCRYPTION$(N, D, T, M)$
  Generates $A_0, \ldots, A_m$ from $N$.
  $c_0 \leftarrow \mathsf{lsb}_t(E(A_0)) \oplus T$
  **for** $i \leftarrow 1,m$ **do**
    $C_i \leftarrow E(A_i) \oplus M_i$
  **end for**
  $C \leftarrow C_0, \ldots, C_m$
  **return** $(N, C, D)$
**end function**

**Algorithm 2** CCM Decryption$(N, C, D)$

1: Generates $A_0, \ldots, A_m$ from $N$.
2: $T \leftarrow \mathsf{lsb}_t\,(E(A_0)) \oplus C_0$
3: **for** $i \leftarrow 1,m$ **do**
4: $\quad M_i \leftarrow E(A_i) \oplus C_i$
5: **end for**
6: $M \leftarrow M_0, \ldots, M_m$
7: Generates $B_0, \ldots, B_r$ from $N$, $M$ and $D$.
8: $X_0 \leftarrow E(B_0)$
9: **for** $i \leftarrow 1,r$ **do**
10: $\quad X_i \leftarrow E(B_i \oplus X_{i-1})$
11: **end for**
12: **if** $T == \mathsf{lsb}_t\,(X_r)$ **then return** $(M, D)$
13: **else return** $\perp$
14: **end if**