A SUBMISSION TO CAESAR COMPETITION FOR
AUTHENTICATED ENCRYPTION

CBA Mode (v1-1)

Designers:
Hossein HOSSEINI
Shahram KHAZAEI

Submitter:
Shahram KHAZAEI
shahram.khazaei@sharif.ir

This version addresses some issues in CBA v1, mainly regarding the offsets. The details of changes are listed in Appendix A.

March 31, 2014

CBA Mode (v1-1)

A Submission to CAESAR Competition for Authenticated Encryption

Hossein Hosseini[1] and Shahram Khazaei[2]

Sharif University of Technology
[1]School of Electrical Engineering
[2]Department of Mathematical Sciences

Abstract. This paper presents the Code-Book Authentication mode (CBA), a submission to the CAESAR competition for authenticated encryption. CBA is a blockcipher mode of encryption that provides confidentiality and authenticity for plaintexts and authenticity for associated data. The proposed mode improves the OCB mode in the sense that it saves up to one blockcipher call to encrypt and authenticate the plaintext. The CBA mode is one pass, uses one-key for encryption and authentication and employs a fixed-length arbitrary nonce.

Keywords: CAESAR Competition, Authenticated Encryption, Blockcipher Mode, Code-Book Authentication.

1 Introduction

Authenticated encryption (AE) is a shared-key encryption scheme, which provides confidentiality and authenticity simultaneously. A straightforward approach for constructing an authenticated-encryption scheme is to combine an encryption scheme and a message authentication code (MAC), appropriately. However, composition methods are slow, require at least two keys, and in practice are more likely to be misused. Therefore, designing a mode of operation for blockciphers has become a popular approach to achieve AE. In recent years, several modes have been proposed, such as CCM [10], GCM [7], EAX [2], IAPM [4], OCB [9,8,5] and CCFB [6].

This article is a submission to CAESAR Competition for authenticated encryption [1]. We present a new blockcipher mode of operation, called Code-Book Authentication (CBA). CBA inherits many features from the OCB, which in turn was a refinement of the IAPM. It is one-pass, uses a single key for both the encryption and authentication processes, and employs a fixed-length arbitrary nonce.

However, CBA improves the OCB mode in the sense that it saves up to one blockcipher call to encrypt and authenticate the plaintext. The improvement is achieved by possibly merging the last two blocks into one block and also considering that most applications need relatively small or moderate amount of data to be encrypted, per key. The trade off is that the last block in the encryption algorithm, possibly, loses its parallelizability, and the decryption algorithm has to wait one block before processing the received block.

2 Notation

A string is a finite sequence of symbols over alphabet $\{0,1\}$. Let $\{0,1\}^*$ denote the set of all strings and $A, B \in \{0,1\}^*$. Let i be a non-negative integer. Let $\{0,1\}^i$ denote the set of all strings of length i. The following notations and operations are used in the description of the CBA mode:

- AB or (A, B): concatenation of the two strings A and B.
- 0^i: string of i 0's.
- 1^i: string of i 1's.
- $|A|$: bit-length of A.
- $\mathrm{pad}_i(A) = \begin{cases} A10^{i-|A|-1} & \text{if } |A| < i \\ A & \text{if } |A| = i \end{cases}$.
- $\mathrm{pad}(A) = \mathrm{pad}_{128}(A)$.
- $\mathrm{MSB}_i(A)$: first i bits of A, where $i \leq |A|$.
- $\mathrm{LSB}_i(A)$: last i bits of A, where $i \leq |A|$.
- $A \oplus B$: bitwise exclusive-or of two equal-length strings A and B.
- $\lfloor x \rfloor$: largest integer not greater than x.
- $\lceil x \rceil$: smallest integer not less than x.
- $[i]_8$: the 8-bit string that encodes i as a binary number, where $i \in \{0, \ldots, 255\}$.
- $A \ll i$: a string with the same bit-length as A, which is a left shift of A by i bits, where $i \leq |A|$ and the first i bits are discarded.
- $A \ggg i$: a string with the same bit-length as A, which is a right rotation of A by i bits; i.e., the last i bits are moved to the beginning.

Let A be a string of length 64. Then, we let:

- $2 \cdot A = (A \ll 1) \oplus \mathrm{MSB}_1(A) \cdot 0^{59}11011$.
- $3 \cdot A = A \oplus (2 \cdot A)$.
- $4 \cdot A = 2 \cdot (2 \cdot A)$.
- $5 \cdot A = 3 \cdot (3 \cdot A)$.
- $6 \cdot A = 3 \cdot (2 \cdot A)$.

3 Specification

3.1 Parameters

The CBA mode uses a blockcipher E (whose inverse is denoted by D), with n-bit blocksize and k-bit key, along with a ν-bit nonce.

Parameter space: CBA is parametrized with a tag-length τ and usage-capacity b, where $0 \leq \tau \leq n$ and $b \leq n/2 - 16$. By usage-capacity b we mean that CBA can be used to encrypt and authenticate at most 2^b message blocks, including the associated data, per key. There is no secret message number; i.e., the secret message number is empty.

- From here on, we fix $n = 128$ and $\nu = 96$.
- All over the paper, we let $\ell = \min(n - 2b - 32, \tau)$.

Recommended parameter sets The recommended parameter sets are as follows:

	E	k	τ	b
(1)	AES	128	32	16
(2)	AES	128	32	32
(3)	AES	128	64	16
(4)	AES	128	64	32
(5)	AES	128	64	48
(6)	AES	128	96	16
(7)	AES	128	96	32
(8)	AES	128	96	48
(9)	AES	192	64	32
(10)	AES	256	96	48

3.2 The Scheme

The CBA mode consists of the algorithm ENC for the encryption and authentication process and the algorithm DEC for the decryption and verification process. We refer to these algorithms as encryption and decryption algorithms, for simplicity. These algorithms use the functions H and $\mathrm{F}_{\alpha,\beta}$ for authenticating the associated data and updating the offset values, respectively.

The encryption algorithm takes a key $K \in \{0,1\}^k$, a public message number (nonce) $N \in \{0,1\}^\nu$, an associated data $A \in \{0,1\}^*$ and a message $M \in \{0,1\}^*$. The output is the ciphertext $C \in \{0,1\}^*$, with the same size as the plaintext, plus τ extra bits to authenticate C. The decryption algorithm takes a key, a nonce, an associated data and a ciphertext, and it returns either a message or a special symbol $\perp$, indicating that the decrypted plaintext is invalid.

The algorithm H takes the key K and the associated data A to compute the τ-bit tag value $T_A = \mathrm{H}(K, A)$. The value T_A will be used in the processes of message encryption and decryption.

Figures 1-5 illustrate the functionality of the CBA encryption algorithm. In the figures, we show the offset values by $\varDelta_i$'s, $\varDelta'$ and $\varDelta^*$ for clarity. The decryption algorithm is straightforward.

CBA Definition In the following, we provide the details of the CBA ENC and the CBA DEC algorithms and the two subroutines H and $\mathrm{F}_{\alpha,\beta}$. Note that, for the encryption and decryption algorithms, if the inputs do not conform to the assumptions, they return $\perp$ as output.

```
[assumes: $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a blockcipher,
          $0 \le \tau \le n,\ b \le n/2 - 16$ and $\ell = \min(n - 2b - 32, \tau)$,
          $K \in \{0,1\}^k, N \in \{0,1\}^\nu$ and $A, M \in \{0,1\}^*$.]
function $\textsc{Enc}(K, N, A, M)$
    $T_A \leftarrow \mathrm{H}(K, A)$
    $R \leftarrow E_K(0^{n-\nu-16}[\tau]_8[b]_8 N)$
    $\Delta \leftarrow \mathrm{F}_{2,2}(\mathrm{MSB}_{n-\ell-2}(R)1^{\ell+2} \ggg \lfloor \frac{\ell}{2} \rfloor + 1)$
    if $|M| \le \ell$ then
        $C_0 \leftarrow M \oplus \mathrm{LSB}_{|M|}(R)$
        $\Delta \leftarrow \mathrm{F}_{3,3}(\Delta)$
        $C_1 \leftarrow \mathrm{MSB}_\tau\big(E_K(0^{n-\tau}T_A \oplus \mathrm{pad}(C_0) \oplus \Delta)\big)$
        return $C_0 C_1$
    $M_0 M' \leftarrow M$ where $|M_0| = \ell$
    $C_0 \leftarrow M_0 \oplus \mathrm{LSB}_\ell(R)$
    $T \leftarrow T_A \oplus 0^{\tau-\ell} C_0$
    $M_1 \cdots M_m \leftarrow M'$ where $|M_i| = n$ for $1 \le i \le m-1$, $1 \le |M_m| \le n$
    $S \leftarrow 0^n$
    for $i = 1$ to $m - 1$ do
        $S \leftarrow S \oplus M_i$
        $\Delta \leftarrow \mathrm{F}_{2,2}(\Delta)$
        $C_i \leftarrow E_K(M_i \oplus \Delta) \oplus \Delta$
    if $|M_m| \le n - \tau$ and $m \ge 2$ then
        $C_{m-1} \leftarrow \mathrm{MSB}_{|M_m|+\tau}(C_{m-1} \oplus \Delta) \oplus M_m T$
        $\Delta \leftarrow \mathrm{F}_{3,3}(\Delta)$
        $C_m \leftarrow S \oplus E_K(\mathrm{pad}(C_{m-1}) \oplus \Delta)$
        return $C_0 C_1 \cdots C_m$
    else
        $S \leftarrow S \oplus \mathrm{pad}(M_m)$
        $\Delta \leftarrow \mathrm{F}_{2,4}(\Delta)$
        $C_m \leftarrow M_m \oplus \mathrm{MSB}_{|M_m|}\big(E_K(0^{n-\tau}T \oplus \Delta)\big)$
        if $|M_m| < n$ then
            $\Delta \leftarrow \mathrm{F}_{3,5}(\Delta)$
        else
            $\Delta \leftarrow \mathrm{F}_{5,3}(\Delta)$
        $C_{m+1} \leftarrow \mathrm{MSB}_\tau\big(E_K(S \oplus 0^{n-\tau}T \oplus \Delta)\big)$
        return $C_0 C_1 \cdots C_m C_{m+1}$
```

```
[assumes: $E : \{0,1\}^k \times \{0,1\}^n \to \{0,1\}^n$ is a blockcipher with inverse $D$,
            $0 \le \tau \le n,\ b \le n/2 - 16$ and $\ell = \min(n - 2b - 32, \tau)$,
            $K \in \{0,1\}^k, N \in \{0,1\}^\nu$ and $A, C \in \{0,1\}^*$.]
function $\textsc{Dec}(K, N, A, C)$
    if $|C| < \tau$ then return $\perp$
    $T_A \leftarrow \mathrm{H}(K, A)$
    $R \leftarrow E_K(0^{n-\nu-16}[\tau]_8[b]_8N)$
    $\mathit{\Delta} \leftarrow \mathrm{F}_{2,2}(\mathrm{MSB}_{n-\ell-2}(R)1^{\ell+2} \ggg \lfloor \frac{\ell}{2} \rfloor + 1)$
    if $|C| \le \ell + \tau$ then
        $C_0C_1 \leftarrow C$ where $|C_1| = \tau$
        $M \leftarrow C_0 \oplus \mathrm{LSB}_{|C_0|}(R)$
        $\mathit{\Delta} \leftarrow \mathrm{F}_{3,3}(\mathit{\Delta})$
        $\mathrm{Temp} \leftarrow \mathrm{MSB}_\tau\big(E_K(0^{n-\tau}T_A \oplus \mathrm{pad}(C_0) \oplus \mathit{\Delta})\big)$
        if $C_1 = \mathrm{Temp}$ then
            return $M$
        else
            return $\perp$
    $C_0C' \leftarrow C$ where $|C_0| = \ell$
    $M_0 \leftarrow C_0 \oplus \mathrm{LSB}_\ell(R)$
    $T \leftarrow T_A \oplus 0^{\tau-\ell}C_0$
    $C_1 \cdots C_m \leftarrow C'$ where $|C_i| = n$ for $1 \le i \le m-1, 1 \le |C_m| \le n$
    $S \leftarrow 0^n$
    for $i = 1$ to $m - 2$ do
        $\mathit{\Delta} \leftarrow \mathrm{F}_{2,2}(\mathit{\Delta})$
        $M_i \leftarrow D_K(C_i \oplus \mathit{\Delta}) \oplus \mathit{\Delta}$
        $S \leftarrow S \oplus M_i$
    if $|C_m| > \tau$ and $m \ge 2$ then
        $C'_{m-1}C'_m \leftarrow C_{m-1}C_m$ where $|C'_m| = n$
        $\mathit{\Delta}' \leftarrow \mathrm{F}_{6,6}(\mathit{\Delta})$
        $M_{m-1} \leftarrow C'_m \oplus S \oplus E_K(\mathrm{pad}(C'_{m-1}) \oplus \mathit{\Delta}')$
        $\mathit{\Delta} \leftarrow \mathrm{F}_{2,2}(\mathit{\Delta})$
        $M_mT' \leftarrow C'_{m-1} \oplus \mathrm{MSB}_{|C'_{m-1}|}(E_K(M_{m-1} \oplus \mathit{\Delta}))$ where $|T'| = \tau$
        if $T = T'$ then
            return $M_0M_1 \cdots M_m$
        else
            return $\perp$
    else
        if $m = 1$ then
            $C'_{m-1}C'_m \leftarrow C_m$ where $|C'_m| = \tau$
        else
            $C'_{m-1}C'_m \leftarrow C_{m-1}C_m$ where $|C'_m| = \tau$
        $\mathit{\Delta} \leftarrow \mathrm{F}_{2,4}(\mathit{\Delta})$
        $M'_{m-1} \leftarrow C'_{m-1} \oplus \mathrm{MSB}_{|C'_{m-1}|}\big(E_K(0^{n-\tau}T \oplus \mathit{\Delta})\big)$
        $S \leftarrow S \oplus \mathrm{pad}(M'_{m-1})$
        if $|M'_{m-1}| < n$ then
            $\mathit{\Delta} \leftarrow \mathrm{F}_{3,5}(\mathit{\Delta})$
        else
            $\mathit{\Delta} \leftarrow \mathrm{F}_{5,3}(\mathit{\Delta})$
        $\mathrm{Temp} \leftarrow \mathrm{MSB}_\tau\big(E_K(S \oplus 0^{n-\tau}T \oplus \mathit{\Delta})\big)$
        if $C'_m = \mathrm{Temp}$ then
            return $M_0 \cdots M_{m-2}M'_{m-1}$
        else
            return $\perp$
```

```
[*assumes:* $K \in \{0,1\}^k$ *and* $A \in \{0,1\}^*$]
**function** $\mathrm{H}(K, A)$
    $A_1 \cdots A_a \leftarrow A$ where $|A_i| = n$ for $1 \leq i \leq a-1$, $|A_a| < n$
    $S \leftarrow 0^n$
    $L \leftarrow E_K([\tau]_8[b]_8 0^{n-16})$
    $\mathit{\Delta} \leftarrow \mathrm{F}_{2,2}(\mathrm{MSB}_{n-2}(L)1^2 \ggg 1)$
    **for** $i = 1$ **to** $a-1$ **do**
        $\mathit{\Delta} \leftarrow \mathrm{F}_{2,2}(\mathit{\Delta})$
        $S \leftarrow S \oplus E_K(A_i \oplus \mathit{\Delta})$
    **if** $|A_a| > 0$ **then**
        $\mathit{\Delta} \leftarrow \mathrm{F}_{3,3}(\mathit{\Delta})$
        $S \leftarrow S \oplus E_K(\mathrm{pad}(A_a) \oplus \mathit{\Delta})$
    $T_A \leftarrow \mathrm{MSB}_\tau(S)$
    **return** $T_A$
```

```
[*assumes:* $\mathit{\Delta} \in \{0,1\}^{128}$]
**function** $\mathrm{F}_{\alpha,\beta}(\mathit{\Delta})$
    $(A, B) \leftarrow \mathit{\Delta}$ where $|A| = |B| = 64$
    **return** $(\alpha \cdot A, \beta \cdot B)$
```

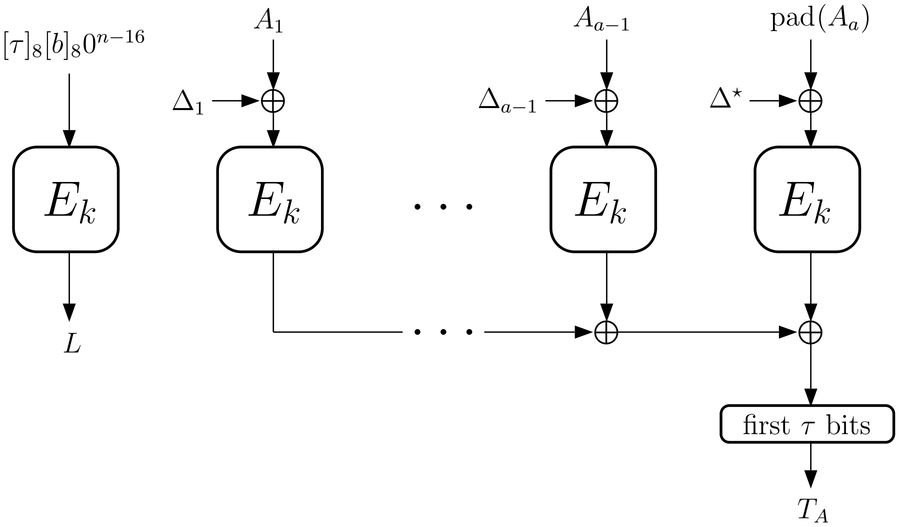

Fig. 1: **CBA Encryption–associated data processing.** Case $|A|$ is not a multiple of n (where $A = A_1 \cdots A_a$ and $|A_i| = n$ for $1 \leq i \leq a-1$ and $0 < |A_a| < n$). In the figure, we have $\Delta_i = \mathrm{F}_{2,2}(\Delta_{i-1})$ where $\Delta_0 = \mathrm{F}_{2,2}(\mathrm{MSB}_{n-2}(L)1^2 \ggg 1)$, and $\Delta^* = \mathrm{F}_{3,3}(\Delta_{a-1})$.

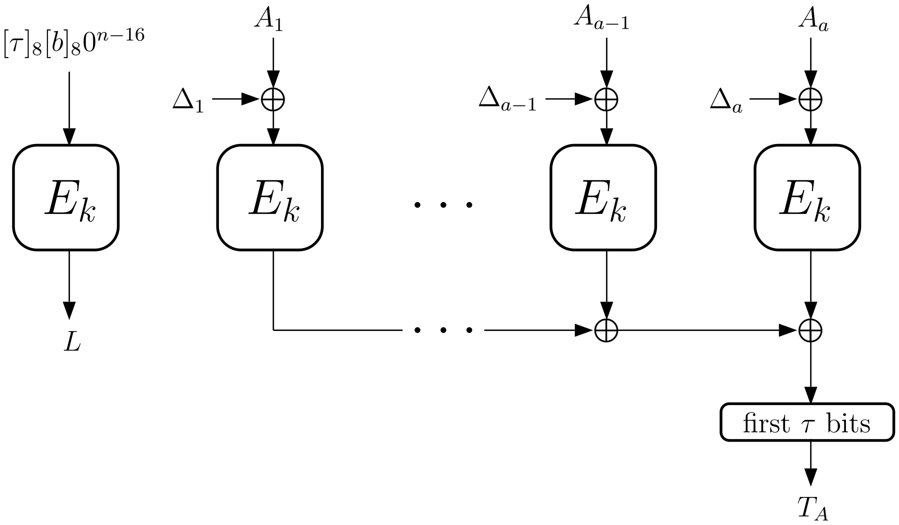

Fig. 2: **CBA Encryption–associated data processing.** Case $|A|$ is a multiple of n (where $A = A_1 \cdots A_a$ and $|A_i| = n$). In the figure, we have $\Delta_i = \mathrm{F}_{2,2}(\Delta_{i-1})$ where $\Delta_0 = \mathrm{F}_{2,2}(\mathrm{MSB}_{n-2}(L)1^2 \ggg 1)$.

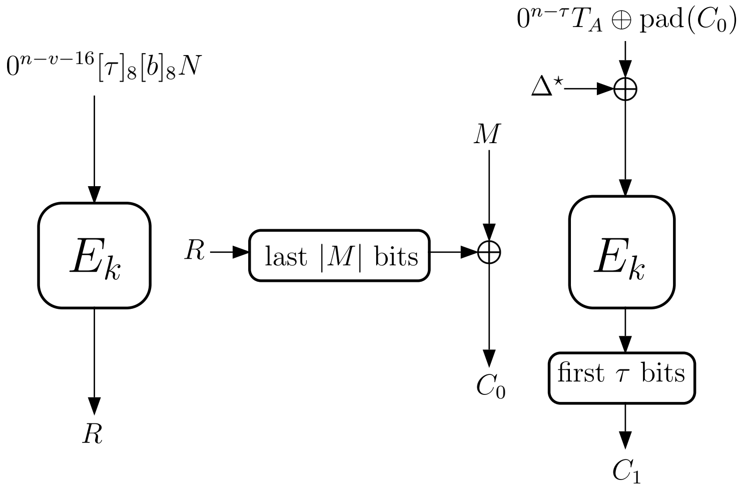

Fig. 3: **CBA Encryption—message processing.** Case $|M| \leq \ell$. In the figure, we have $\Delta^* = \mathrm{F}_{3,3}(\Delta_0)$ where $\Delta_0 = \mathrm{F}_{2,2}(\mathrm{MSB}_{n-\ell-2}(R)1^{\ell+2} \ggg \left\lfloor \frac{\ell}{2} \right\rfloor + 1)$.

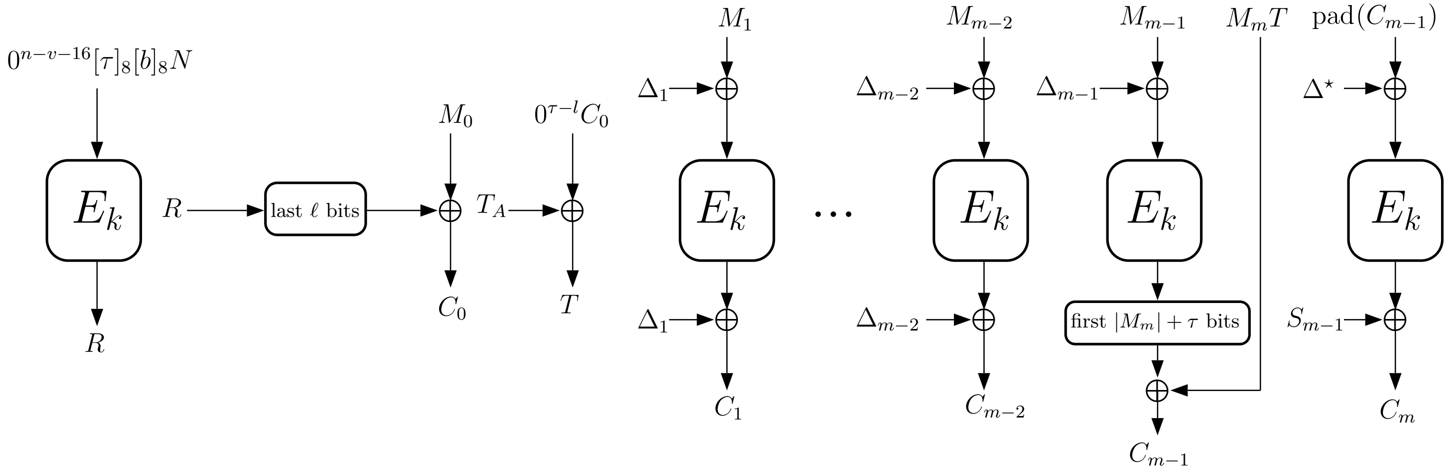

Fig. 4: **CBA Encryption—message processing.** Case $|M_m| \leq n - \tau$ and $m \geq 2$ (where $M = M_0M_1 \cdots M_m$, $|M_0| = \ell$, $|M_i| = n$ for $i = 1, \cdots, m-1$). In the figure, we have $S_{m-1} = M_1 \oplus \cdots \oplus M_{m-1}$, $\varDelta_i = \mathrm{F}_{2,2}(\varDelta_{i-1})$ where $\varDelta_0 = \mathrm{F}_{2,2}(\mathrm{MSB}_{n-\ell-2}(R)1^{\ell+2} \ggg \lfloor \frac{\ell}{2} \rfloor + 1)$, and $\varDelta^* = \mathrm{F}_{3,3}(\varDelta_{m-1})$.

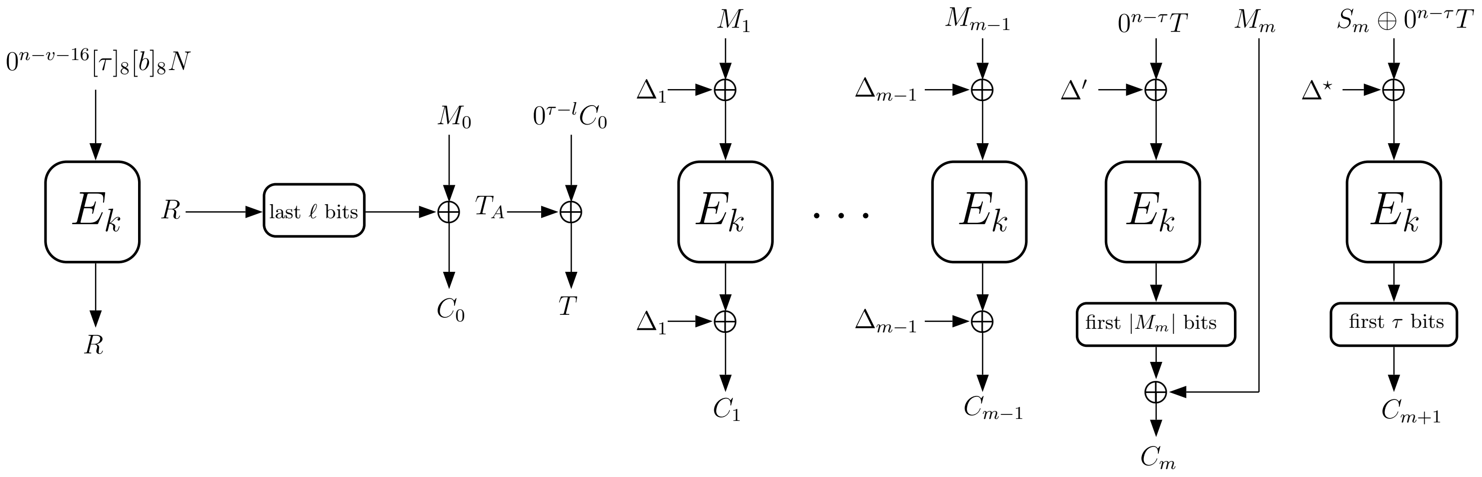

Fig. 5: **CBA Encryption—message processing.** Case $|M_m| > n - \tau$ or $m = 1$ (where $M = M_0M_1 \cdots M_m$, $|M_0| = \ell$, $|M_i| = n$ for $i = 1, \cdots, m-1$). In the figure, we have $S_m = M_1 \oplus \cdots \oplus M_{m-1} \oplus \mathrm{pad}(M_m)$, $\Delta_i = \mathrm{F}_{2,2}(\Delta_{i-1})$ where $\Delta_0 = \mathrm{F}_{2,2}(\mathrm{MSB}_{n-\ell-2}(R)1^{\ell+2} \ggg \lfloor \frac{\ell}{2} \rfloor + 1)$, $\Delta' = \mathrm{F}_{2,4}(\Delta_{m-1})$ and $\Delta^* = \mathrm{F}_{3,5}(\Delta')$ when $|M_m| < n$ and $\Delta^* = \mathrm{F}_{5,3}(\Delta')$ when $|M_m| = n$.

4 Security goals and analysis

CBA has been designed to support the strongest notions of security for both confidentiality and authenticity. This is formalized using the standard indistinguishability games for confidentiality and unforgeability games for authenticity. However, the security of the CBA mode is not yet proved and is a work in progress.

The following requirements should be satisfied in order to use CBA securely:

1. Each key should be randomly generated.
2. Each key and nonce pair should not be used to encrypt more than one message.
3. If the verification fails, the decryption algorithm returns nothing, but $\perp$.
4. For each key, the CBA should not be used to encrypt more than a total of 2^b message blocks, including the associated data.
5. The CBA usage-capacity must satisfy $b \leq n/2 - 16$.

If the above requirements are satisfied, we have the following claims:

Claim 1. Confidentiality of CBA degrades as per $q^2 \times 2^{-(n-\ell-2)}$, where q is the total number of blocks that the adversary acquires.

Claim 2. Authenticity of CBA degrades as per $q^2 \times 2^{-(n-\ell-2)}$, where q is the total number of blocks that the adversary acquires.

5 Features

CBA inherits most of the desirable properties of OCB. However, it improves OCB in the sense that it saves up to one blockcipher call for both encryption and decryption. Thus, CBA performs better specifically for short-length messages. We specify the CBA features in the following:

- CBA protects the confidentiality of M and the authenticity of A, N, and M. It does this using, on average, $\lceil \frac{|A|}{n} \rceil + \lceil \frac{|M|}{n} \rceil + 1 + \frac{\tau - \ell}{n}$ blockcipher calls.
- For a given message, the CBA returns a ciphertext of minimal length.
- CBA requires a single blockcipher key for both encryption and authentication.
- It achieves AE using only a single pass over the message M.
- The nonce need not be random or secret.
- CBA is on-line: one does not need to know the length of A or M to proceed with encryption, nor need one know the length of A or C to proceed with decryption. However, the decryption algorithm has to wait one block before processing the received block.

– If the associated data is fixed during a session, then it can be pre-processed so that there is effectively no per-message cost to providing the authenticity of associated data.
– CBA is parallelizable: the bulk of its blockcipher calls can be performed simultaneously, except, possibly, for the last message block.

Recommended parameters: We have three parameters to take into account.

– **key-length.** The CBA supports 128, 192 and 256-bit keys. The parameter sets consider all key-lenghts with an emphasize on the 128-bit key, because it is believed to be enough for many applications. However, other key-lengths may be chosen if higher levels of security are required.
– **tag-length.** We consider different tag lengths of 32, 64 and 96, which are appropriate for various applications.
– **usage-capacity.** To increase the efficiency, we suggest different settings for the maximum number of allowed message and associated data blocks. We consider three categories, which are low, moderate and high amount of data, corresponding to at most 2^{16}, 2^{32} and 2^{48} blocks of data.

In comparison with AES-GCM, we expect the CBA to be at least two times faster. Moreover, CBA is more flexible in terms of the parameters' selection.

6 Design rationale

CBA is designed such that it requires up to one less blockcipher call, compared to the OCB mode, when the nonce is random. The improvement is achieved by two contribution to the OCB. The first is that if the maximum number of message blocks is to be lower than 2^b, then we can use ℓ bits of R to encrypt ℓ bits of the message in a stream cipher fashion, and yet preserve the security. Recall that R is an encryption of a nonce-dependent block. This means that the message length, needed to be encrypted, is decreased by ℓ bits. Note that this decrease does not necessarily result in one less blockcipher call. In fact, on average, we need ℓ/n less blockcipher calls. This improvement is particularly important for very short-length messages, which is the case in many applications.

The second improvement is originated from the fact that OCB truncates more than or equal to n bits, when $|M_m| + \tau \leq n$. Thus, in this case, it is possible to make an improvement by merging the last two blocks into one block and, therefore, omit one blockcipher call. Assuming that the length of the last message block is a random variable with uniform distribution between 1 and n, we save, on average, $(n - \tau)/n$ blockcipher calls.

Let g denote the gain of the CBA mode with respect to the OCB, i.e., the average number of blockcipher call reductions. We have:

$$g = \frac{n-\tau}{n} + \frac{\ell}{n} = 1 - \frac{\tau - \ell}{n}\ .$$

Since $0 \leq \ell \leq \tau$, we have:

$$1 - \frac{\tau}{n} \leq g \leq 1 \, .$$

This shows that CBA saves a number of blockcipher calls between $1 - \tau/n$ and 1. The following table lists the value of ℓ and the CBA gain for our recommended parameter sets.

parameter set	1	2	3	4	5	6	7	8	9	10
ℓ	32	32	64	32	0	64	32	0	32	0
gain g	1	1	1	3/4	1/2	3/4	1/2	1/4	3/4	1/4

Moreover, we suggest a simpler offset generation, compared to the OCB. In [5], the idea of the *stretch-then-shift* hash is proposed for further reduction of, on average, 0.98 blockcipher calls in case that the nonce is a counter. Although this approach is also applicable to the CBA, we seek for solutions that are efficient for various applications and platforms, without increasing the internal state or going for table look-up. There are many approaches for updating the offset strings to guarantee the maximum period for them [3]. However, we mention that since the message size is at most 2^{48} blocks, it is of no use for the offsets to have the maximum period of 2^{128}. Therefore, we suggest an efficient updating function that provides the period of 2^{64}. The function is $(A, B) \leftarrow (\alpha \cdot A, \beta \cdot B)$, where A and B are 64-bit strings and the elements α and β and the multiplication are in GF(2^{64}). We found that this function performs reasonably well on 32, 64 or 128-bit processors.

The last point is to prevent the algorithm from misusing with different lengths of tag or usage-capacity. For this, we pad the tag-length and the usage-capacity to the 96-bit nonce, so that their integrity will be protected along with nonce.

The designers have not hidden any weaknesses in this cipher. CBA is a refinement over OCB, which has received years of in-depth analysis in the cryptography society and enjoys provable security.

7 Intellectual property

The design of the CBA mode has largely been influenced by the OCB, which in turn was a refinement of the IAPM. There are US patents 7046802, 7200227, 7949129, and 8321675 on OCB. In addition, there are patents 6963976, 6973187, 7093126, and 8107620 on IAPM.

The authors are not aware of any other patents related to the CBA mode. If any of this information changes, the submitter will promptly (and within at most one month) announce these changes on the crypto-competitions mailing list.

8 Consent

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

Acknowledgment. The authors would like to thank Alexandre Duc for his helpful comments.

References

1. CAESAR—Competition for Authenticated Encryption: Security, Applicability, and Robustness. `http://competitions.cr.yp.to/caesar.html`.
2. M. Bellare, P. Rogaway, and D. Wagner. The EAX Mode of Operation. In B. K. Roy and W. Meier, editors, *FSE*, volume 3017 of *Lecture Notes in Computer Science*, pages 389–407. Springer, 2004.
3. D. Chakraborty and P. Sarkar. A General Construction of Tweakable Block Ciphers and Different Modes of Operations. *IEEE Transactions on Information Theory*, 54(5):1991–2006, 2008.
4. C. S. Jutla. Encryption Modes with Almost Free Message Integrity. In B. Pfitzmann, editor, *EUROCRYPT*, volume 2045 of *Lecture Notes in Computer Science*, pages 529–544. Springer, 2001.
5. T. Krovetz and P. Rogaway. The Software Performance of Authenticated-Encryption Modes. In A. Joux, editor, *FSE*, volume 6733 of *Lecture Notes in Computer Science*, pages 306–327. Springer, 2011.
6. S. Lucks. Two-Pass Authenticated Encryption Faster Than Generic Composition. In H. Gilbert and H. Handschuh, editors, *FSE*, volume 3557 of *Lecture Notes in Computer Science*, pages 284–298. Springer, 2005.
7. D. A. McGrew and J. Viega. The Security and Performance of the Galois/Counter Mode (GCM) of Operation. In A. Canteaut and K. Viswanathan, editors, *INDOCRYPT*, volume 3348 of *Lecture Notes in Computer Science*, pages 343–355. Springer, 2004.
8. P. Rogaway. Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In P. J. Lee, editor, *ASIACRYPT*, volume 3329 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2004.
9. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: a block-cipher mode of operation for efficient authenticated encryption. In M. K. Reiter and P. Samarati, editors, *ACM Conference on Computer and Communications Security*, pages 196–205. ACM, 2001.
10. D. Whiting, R. Housley, and N. Ferguson. AES encryption & authentication using CTR mode & CBC-MAC. In *IEEE P802.11 doc 02/001r2, May 2002*.

A Changelog

The changes with respect to the CBA v1 are listed in the following:

- We modified the offset updating rules and defined the function $\mathrm{F}_{\alpha,\beta}$ instead of the functions F and F^*.
- We initialize $\varDelta$ as $\mathrm{F}_{2,2}(\mathrm{MSB}_{n-\ell-2}(R)1^{\ell+2} \ggg \lfloor \frac{\ell}{2} \rfloor + 1)$ instead of $1\mathrm{MSB}_{n-\ell}(R)0^{\ell-2}1$ in the encryption and decryption algorithms.
- We initialize $\varDelta$ as $\mathrm{F}_{2,2}(\mathrm{MSB}_{n-2}(L)1^2 \ggg 1)$ instead of L in the function H.
- We generate R as $E_K(0^{n-\nu-16}[\tau]_8[b]_8N)$ instead of $E_K([\tau]_8[b]_8 0^{n-\nu-16}N)$.
- We included a special case of plaintext encryption that we had missed.
- We corrected some syntax issues in the paddings.
- The encryption and decryption algorithms are described more clearly.
- We corrected the definition of the function H so that it outputs a zero string when the associated data is empty.
- We corrected the security bounds in the claims 1 and 2.