The AEM Authenticated-Encryption Mode

(Specification 1.3 — November 6, 2003)

Phillip Rogaway*

1 Overview

This note specifies AEM, a mode of operation giving authenticated encryption. AEM is a refinement to Rogaway, Bellare, and Black's OCB mode [10], while OCB was, in turn, a refinement to Jutla's IAPM [5]. AEM is also a successor to the work of Gligor and Donescu's [4] and to the broader line of research that has defined and investigated authenticated encryption [1, 2, 6–8]. The acronym AEM stands for *authenticated-encryption mode* and *advanced encryption mode*. Prominent characteristics of AEM are:

(1) AEM is a mode of operation parameterized by an n-bit block cipher E and a tag length $\tau \in [0 \mathinner{..} n]$.

(2) Encryption and decryption depend on an n-bit nonce N, which must be selected as a new value for each encryption. The nonce need not be random or secret.

(3) AEM allows an arbitrary header H to be specified when one encrypts or decrypts a string.

(4) The message M and header H can have any bit length, and the ciphertext $\mathcal{C}$ one gets by encrypting M in the presence of H will always have τ bits more than M.

(5) AEM encryption protects the privacy and authenticity of M and the authenticity of H and N.

(6) AEM uses $\|M\| + \|H\| + 2$ block-cipher calls, where $\|\cdot\|$ is the length of the specified string measured in n-bit blocks.

(7) If the header H is fixed during a session then, after preprocessing, there is effectively no cost to have H authenticated—the mode will use $\|M\| + 2$ block-cipher calls regardless of $\|H\|$.

(8) AEM uses a single key K for the underlying block cipher, and all block-cipher calls are keyed by K.

(9) AEM is *on-line*: one need not know the length of H or M to proceed with encryption, and one need not know the length of H or $\mathcal{C}$ to proceed with decryption.

(10) AEM is *parallelizable*: the bulk of its block-cipher calls may be performed simultaneously.

(11) The main computational work beyond the block-cipher calls consists of one doubling operation and three xors for each n-bit block. Doubling consists of one shift and one conditional xor.

(12) If the header H is empty, no key setup is necessary or useful for AEM. If the header H is nonempty, key setup is a single block-cipher call.

(13) AEM enjoys provable security. One must assume that the block cipher E is secure in the customary (strong PRP) sense. Security falls off in $\sigma^2/2^n$ where σ is the total number of blocks one acts on.

Like all authenticated-encryption modes discussed in the literature, AEM becomes completely insecure if one acts on a total number of blocks approaching $\sigma = 2^{n/2}$. Care must be taken to re-key well before then. Security is also sacrificed if a nonce is re-used.

Two major factors differentiate AEM from its predecessor OCB: the first is the presence of the header H, an issue discussed in [8], and the second is the simpler way in which AEM computes offsets, which no longer involves counting the number-of-trailing-zero bits in a counter or doing any other non-constant-time

* Department of Computer Science, One Shields Ave., University of California, Davis, California, 95616, USA; and Department of Computer Science, Faculty of Science, Chiang Mai University, Chiang Mai 50200 Thailand. E-mail: `rogaway@cs.ucdavis.edu` WWW: `www.cs.ucdavis.edu/~rogaway/`

calculation for each n-bit block. The theory underlying AEM is described in Rogaway [9] and the body of work on which that builds. The current note gives only a specification—no security definitions or proofs.

2 Notation

By a *string* we mean a finite sequence of zeros and ones. The length of a string X (in bits) is written $|X|$. The string of length 0 is called the *empty string* and is denoted ε. By 0^i we mean the string of i zero-bits and by 1^j we mean the string of j one-bits. The set of all strings is denoted $\{0,1\}^*$ and the set of all strings of length n is denoted $\{0,1\}^n$. If X and Y are strings then XY or $X \parallel Y$ is their concatenation. If X is a string of length n and $\tau \in [0 \mathinner{..} n]$ then we write $X\,[\text{first } \tau \text{ bits}]$ for the first τ bits of X. If X and Y are strings of equal length then $X \oplus Y$ denotes their bitwise xor, while if $|X| < |Y|$ then $X \oplus Y$ is $X \oplus Y\,[\text{first } |X| \text{ bits}]$. If X is a string of length at most and n and n is understood, then $X0^* = X0^{|X| \bmod n}$ if $X \neq \varepsilon$ and $X0^* = 0^n$ if $X = \varepsilon$. If X is a string of length at most $2^n - 1$ and n is understood then by $\mathrm{len}(X)$ we mean the n-bit that encodes the length of X, in binary, most-significant-bit first and least-significant-bit last.

Let X be a 128-bit string. We define what it means to *double* X, or multiply X by two, giving a 128-bit string we denote by $2\,X$. This is defined as $2\,X = X{\ll}1$ if the first bit of X is a 0, and $2\,X = (X{\ll}1) \oplus 0^{120}10^41^3$ if the first bit of X is a 1. Here $X{\ll}1$ means the left shift of X by one position: if $X = X_1 \cdots X_{128}$ then $X{\ll}1 = X_2 \cdots X_{128}0$. We similarly define what it means to multiple X by three and five, defining $3\,X = 2X \oplus X$ and $5\,X = (2(2X)) \oplus X$. We emphasize that $2X$, $3X$, and $5X$ should not be confused with multiplication in the integers: the semantics and the mechanics of computing these values is very different from integer multiplication.

An n-bit block cipher is a function $E\colon \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ where $n \geq 1$ is a number (the block length) and $\mathcal{K}$ is a finite nonempty set (the keys) and $E(K, \cdot) = E_K(\cdot)$ is a permutation for all $K \in \mathcal{K}$. We write $E_K^{-1}(Y)$ for the string X such that $E_K(X) = Y$.

3 Definition of AEM

Parameters. AEM is parameterized by an n-bit block cipher E and a tag length $\tau \in [0 \mathinner{..} n]$. The parameters E and τ must be fixed for a given session and would typically be fixed for a given application. If parameters are negotiated between communicating parties at the beginning of a session then they must be negotiated in an authenticated way. This specification assumes the use of a block cipher having block length of $n = 128$ bits. Although this specification could easily be extended to allow other values, block lengths of $n = 64$ bits would have to be disparaged due to the $\sigma^2/2^n$ security degradation discussed in Section 1, while block lengths other than $n = 64$ and $n = 128$ have little practical value. We expect that E will usually be AES (meaning AES128, AES192, or AES256). To be more explicit in specifying parameters one may write AEM-E or AEM[E] or AEM[E, τ].

Encryption interface. Once the parameters E and τ have been fixed, AEM provides a method to *encrypt* and a method to *decrypt*. To encrypt, AEM.Encrypt takes

- a key K (a random string drawn from the key space of the underlying block cipher),
- a nonce N (an n-bit string that must not be repeated, during all encryption requests, during a session),
- a header H (an arbitrary string), and
- a message M (an arbitrary string)

and produces as output

- a ciphertext (having τ bits more than M).

Ciphertext $\mathcal{C} = \text{AEM.Encrypt}_K^{N\,H}(M)$ protects the privacy of M and the authenticity of M, N, and H. If one does not need the header H let it be the empty string $H = \varepsilon$. Note that a small amount of material may be authenticated by placing it in N since all of N is authenticated, too.

Decryption interface. To decrypt, AEM.Decrypt takes

- a key K (a random string drawn from the key space of the underlying block cipher),

- a nonce N (an n-bit string)
- a header H (an arbitrary string), and
- a ciphertext $\mathcal{C}$ (an arbitrary string)

and produces as output either

- a message M (having τ bits fewer than $\mathcal{C}$) or
- the distinguished symbol Invalid (indicating that (N, H, C) is invalid with respect to K).

The message $M = \text{AEM.Decrypt}_K^{N\,H}(C)$ is always returned if $\mathcal{C} = \text{AEM.Encrypt}_K^{N\,H}(M)$. The symbol Invalid is usually returned as $\text{AEM.Decrypt}_K^{N\,H}(\mathcal{C})$ if $\mathcal{C}$ was produced in a manner other than by setting $\mathcal{C} = \text{AEM.Encrypt}_K^{N\,H}(M)$ for some M. It is beyond the scope of this note to explain the scientific meaning of the last sentence; see a work such as [9] for that.

Specification. Our definition of AEM.Encrypt and AEM.Decrypt are given in Figure 1. The definitions make use of functions OCB1.Encrypt, OCB1.Decrypt, and PMAC1, which are defined in the same figure. The instruction "Parse $\mathcal{C}$ into $C \parallel T$" means to let C be the first $|\mathcal{C}| - \tau$ bits of $\mathcal{C}$ and to let T be the last τ bits of $\mathcal{C}$. The instruction "Parse M into $M[1] \cdots M[m]$" means to let $m = \max\{\lceil |M|/n \rceil, 1\}$ and to let $M[1], \ldots, M[m]$ be the unique strings such that $M[1] \cdots M[m] = M$ and $|M[1]| = \cdots = |M[m-1]| = n$. The instruction "Parse $\mathcal{C}$ into $C[1] \cdots C[m]\, T$" means to let C be the first $|\mathcal{C}| - \tau$ bits of $\mathcal{C}$ and to let T be the last τ bits of $\mathcal{C}$ and to let $m = \max\{\lceil |C|/n \rceil, 1\}$ and to let $C[1], \ldots, C[m]$ be the unique strings such that $C[1] \cdots C[m] = C$ and $|C[1]| = \cdots = |C[m-1]| = n$. Illustrations of OCB1.Encrypt and PMAC1 are given in Figures 2 and 3.

Acknowledgments

This research was supported by NSF 0208842 and by a gift from Cisco System. Thanks to the NSF (particularly Carl Landwehr) and to Cisco (particularly David McGrew) for their kind support of my research.

References

[1] M. Bellare and P. Rogaway. Encode-then-encipher encryption: how to exploit nonces or redundancy in plaintexts for efficient cryptography. *Advances in Cryptology — Asiacrypt '00*, Lecture Notes in Computer Science, vol. 1976, Springer-Verlag, 2000. www.cs.ucdavis.edu/~rogaway

[2] M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. *Advances in Cryptology — Asiacrypt '00.* Lecture Notes in Computer Science, vol. 1976, T. Okamoto., ed., Springer-Verlag, 2000. www-cse.ucsd.edu/users/mihir

[3] J. Black and P. Rogaway. A block-cipher mode of operation for parallelizable message authentication. *Advances in Cryptology — Eurocrypt '02.* Lecture Notes in Computer Science, vol. 2332, pp. 384–397. Springer-Verlag, 2002. www.cs.ucdavis.edu/~rogaway

[4] V. Gligor and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authentication modes. *Fast Software Encryption*, Lecture Notes in Computer Science, vol. 2355, Springer-Verlag, pp. 92–108, 2001. www.eng.umd.edu/~gligor

[5] C. Jutla. Encryption modes with almost free message integrity. *Advances in Cryptology — EUROCRYPT 2001.* Lecture Notes in Computer Science, vol. 2045, B. Pfitzmann, ed., Springer-Verlag, 2001.

[6] J. Katz and M. Yung. Unforgeable encryption and adaptively secure modes of operation. *Fast Software Encryption '00*, Lecture Notes in Computer Science, vol. 1978, Springer-Verlag, pp. 284–299, 2001.

[7] H. Krawczyk. The order of encryption and authentication for protecting communications (or: How secure is SSL?) *Advances in Cryptology — CRYPTO '01.* Lecture Notes in Computer Science, vol. 2139, Springer-Verlag, pp. 310–331, 2001.

[8] P. Rogaway. Authenticated-encryption with associated-data. ACM CCS'02. ACM Press, 2002. www.cs.ucdavis.edu/~rogaway

Algorithm $\text{AEM.Encrypt}_K^{N\,H}\,(M)$

$\mathcal{C} \leftarrow \text{OCB1.Encrypt}_K^N(M)$
Parse $\mathcal{C}$ into $C \,\|\, T$
if $H \neq \varepsilon$ **then** $T \leftarrow T \oplus \text{PMAC1}_K(H)$
return $C \,\|\, T$

Algorithm $\text{OCB1.Encrypt}_K^N\,(M)$

Parse M into $M[1] \cdots M[m]$
$\Delta \leftarrow E_K(N)$
$\Sigma \leftarrow 0^n$
for $i \leftarrow 1$ **to** $m-1$ **do**
    $\Delta \leftarrow 2\,\Delta$
    $C[i] \leftarrow E_K(M[i] \oplus \Delta) \;\oplus\; \Delta$
    $\Sigma \leftarrow \Sigma \oplus M[i]$
$\Delta \leftarrow 2\,\Delta$
$\mathit{Pad} \leftarrow E_K(\text{len}(M[m]) \oplus \Delta)$
$C[m] \leftarrow M[m] \oplus \mathit{Pad}$
$C \leftarrow C[1] \cdots C[m]$
$\Sigma \leftarrow \Sigma \oplus C[m]0^* \oplus \mathit{Pad}$
$\Delta \leftarrow 3\,\Delta$
$\mathit{Tag} \leftarrow E_K(\Sigma \oplus \Delta)$
$T \leftarrow \mathit{Tag}$ [first τ bits]
return $\mathcal{C} \leftarrow C \,\|\, T$

Algorithm $\text{PMAC1}_K\,(M)$

Parse M into $M[1] \cdots M[m]$
$\Theta \leftarrow 5\,E_K(0^n)$
$\Sigma \leftarrow 0^n$
for $i \leftarrow 1$ **to** $m-1$ **do**
    $\Theta \leftarrow 2\,\Theta$
    $Y \leftarrow E_K(M[i] \oplus \Theta)$
    $\Sigma \leftarrow \Sigma \oplus Y$
$\Theta \leftarrow 2\,\Theta$
if $|M[m]| = n$ **then** $\Theta \leftarrow 3\,\Theta, \quad \Sigma \leftarrow \Sigma \oplus M[m]$
        else $\Theta \leftarrow 5\,\Theta, \quad \Sigma \leftarrow \Sigma \oplus M[m]10^*$
$\mathit{Tag} \leftarrow E_K(\Sigma \oplus \Theta)$
$T \leftarrow \mathit{Tag}$ [first τ bits]
return T

Algorithm $\text{AEM.Decrypt}_K^{N\,H}\,(\mathcal{C})$

if $|\mathcal{C}| < \tau$ **then return** INVALID
Parse $\mathcal{C}$ into $C \,\|\, T$
if $H \neq \varepsilon$ **then** $T \leftarrow T \oplus \text{PMAC1}_K(H)$
$M \leftarrow \text{OCB1.Decrypt}_K^N(C \,\|\, T)$
return M

Algorithm $\text{OCB1.Decrypt}_K^N\,(\mathcal{C})$

Parse $\mathcal{C}$ into $C[1] \cdots C[m]\,T$
$\Delta \leftarrow E_K(N)$
$\Sigma \leftarrow 0^n$
for $i \leftarrow 1$ **to** $m-1$ **do**
    $\Delta \leftarrow 2\,\Delta$
    $M[i] \leftarrow E_K^{-1}(C[i] \oplus \Delta) \;\oplus\; \Delta$
    $\Sigma \leftarrow \Sigma \oplus M[i]$
$\Delta \leftarrow 2\,\Delta$
$\mathit{Pad} \leftarrow E_K(\text{len}(C[m]) \oplus \Delta)$
$M[m] \leftarrow C[m] \oplus \mathit{Pad}$
$M \leftarrow M[1] \cdots M[m]$
$\Sigma \leftarrow \Sigma \oplus C[m]0^* \oplus \mathit{Pad}$
$\Delta \leftarrow 3\,\Delta$
$\mathit{Tag} \leftarrow E_K(\Sigma \oplus \Delta)$
$T' \leftarrow \mathit{Tag}$ [first τ bits]
if $T = T'$ **then return** M
        else return INVALID

Figure 1: The AEM mode of operation. The plaintext is M, the ciphertext is $\mathcal{C} = C \,\|\, T$, the key is K, and the header is H. The underlying block cipher is E.

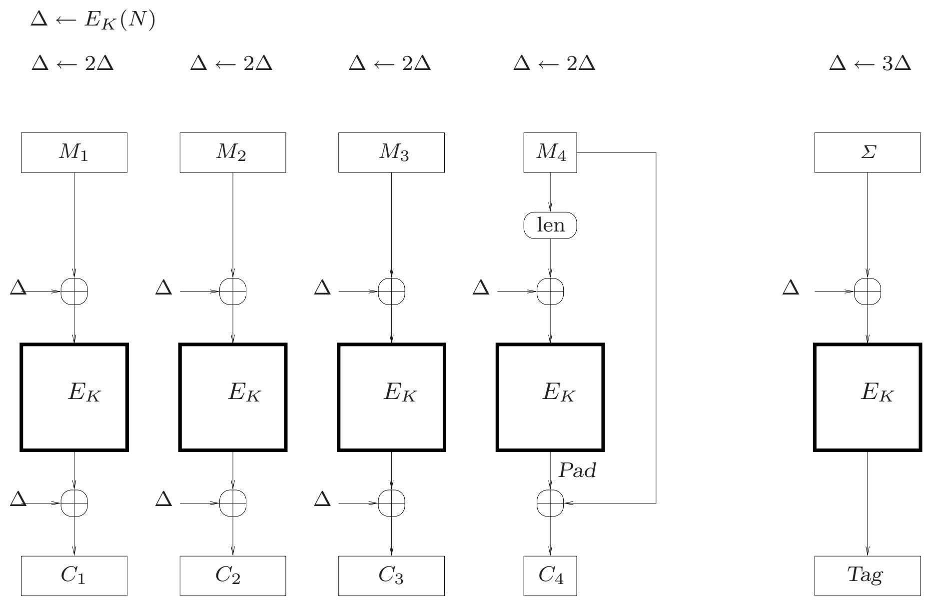

Figure 2: An illustration of OCB1 encryption acting on a message of four blocks. OCB1 is the core of AEM. Read the diagram in vertical strips, from top-to-bottom and from left-to-right.

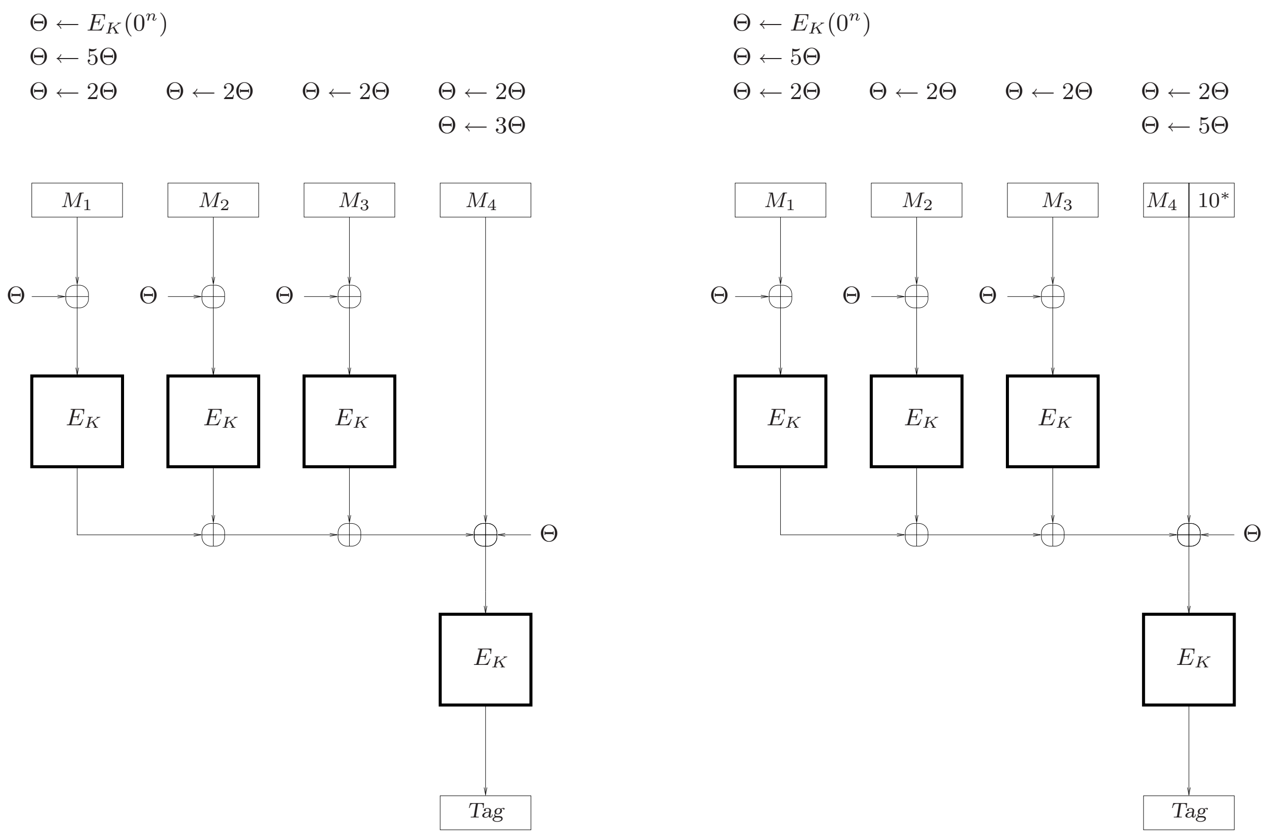

Figure 3: An illustration of PMAC1 authentication of a four-block message. On the left is the case where the final block is a full block, and on the right is the case where the final block is a partial block. Read each picture in vertical strips, from top-to-bottom and from left-to-right.

[9] P. Rogaway. Efficient instantiations of tweakable blockciphers and refinements to modes OCB and PMAC. Manuscript, 2003. www.cs.ucdavis.edu/~rogaway

[10] P. Rogaway, M. Bellare, and J. Black. OCB: A block-cipher mode of operation for efficient authenticated encryption. TISSEC, August 2003. Earlier version, with T. Krovetz, in *ACM Conference on Computer and Communications Security* (CCS-8), ACM Press, pp. 195–205, 2001. www.cs.ucdavis.edu/~rogaway