AUTHENTICATED ENCRYPTION

So Far ...

We have looked at methods to provide privacy and authenticity separately:

<table>
<thead>
<tr>
<th>Goal</th>
<th>Primitive</th>
<th>Security notion</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data privacy</td>
<td>symmetric encryption</td>
<td>IND-CPA</td>
</tr>
<tr>
<td>Data authenticity</td>
<td>MAC</td>
<td>UF-CMA</td>
</tr>
</tbody>
</table>

Authenticated Encryption

In practice we often want both privacy and authenticity.

**Example:** A doctor wishes to send medical information $M$ about Alice to the medical database. Then

- We want data privacy to ensure Alice’s medical records remain confidential.
- We want authenticity to ensure the person sending the information is really the doctor and the information was not modified in transit.

We refer to this as authenticated encryption.

Authenticated Encryption Schemes

Syntactically, an authenticated encryption scheme is just a symmetric encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where
Privacy of Authenticated Encryption Schemes

The notion of privacy for symmetric encryption carries over, namely we want IND-CPA.

Integrity of Authenticated Encryption Schemes

Adversary’s goal is to get the receiver to accept a “non-authentic” ciphertext $C$.

Integrity of ciphertexts: $C$ is “non-authentic” if it was never transmitted by the sender.

INT-CTXT

Let $\mathcal{AE} = (K, E, D)$ be a symmetric encryption scheme and $A$ an adversary.

Game INTCTXT$_{\mathcal{AE}}$

<table>
<thead>
<tr>
<th>procedure Initialize</th>
<th>procedure Finalize($C$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K \leftarrow K^*$; $S \leftarrow \emptyset$</td>
<td>$M \leftarrow D_K(C)$</td>
</tr>
<tr>
<td>$\text{procedure Enc}(M)$</td>
<td>if $(C \not\in S \land M \neq \bot)$ then</td>
</tr>
<tr>
<td>$C \leftarrow E_K(M)$</td>
<td>$\text{return true}$</td>
</tr>
<tr>
<td>$S \leftarrow S \cup {C}$</td>
<td>Else return false</td>
</tr>
<tr>
<td>Return $C$</td>
<td></td>
</tr>
</tbody>
</table>

The int-ctxt advantage of $A$ is

$$\text{Adv}^{\text{int-ctxt}}_{\mathcal{AE}}(A) = \Pr[\text{INTCTXT}^A_{\mathcal{AE}} \Rightarrow \text{true}]$$

Integrity with privacy

The goal of authenticated encryption is to provide both integrity and privacy. We will be interested in IND-CPA + INT-CTXT.
Plain Encryption Does Not Provide Integrity

**Algorithm** $E_K(M)$

$C[0] \leftarrow \{0, 1\}^n$

For $i = 1, \ldots, m$

$C[i] \leftarrow E_K(C[i-1] \oplus M[i])$

Return $C$

**Algorithm** $D_K(C)$

For $i = 1, \ldots, m$

$M[i] \leftarrow E_K^{-1}(C[i]) \oplus C[i-1]$

For $i = 1, \ldots, m$

$C[i] \leftarrow E_K(C[i-1] \oplus M[i])$

Return $C$

**Question:** Is CBC encryption INT-CTXT secure?

**Answer:** No, because any string $C[0]C[1] \ldots C[m]$ has a valid decryption.

Mihir Bellare  
UCSD  
9

Plain Encryption Does Not Provide Integrity

**Algorithm** $E_K(M)$

$C[0] \leftarrow \{0, 1\}^n$

For $i = 1, \ldots, m$

$C[i] \leftarrow E_K(C[i-1] \oplus M[i])$

Return $C$

**Algorithm** $D_K(C)$

For $i = 1, \ldots, m$

$M[i] \leftarrow E_K^{-1}(C[i]) \oplus C[i-1]$

For $i = 1, \ldots, m$

$M[i] \leftarrow E_K(C[i-1]) \oplus C[i]$

Return $M$

**Question:** Is CBC encryption INT-CTXT secure?

**Answer:** No, because any string $C[0]C[1] \ldots C[m]$ has a valid decryption.

Mihir Bellare  
UCSD  
10

Plain Encryption Does Not Provide Integrity

**Algorithm** $E_K(M)$

$C[0] \leftarrow \{0, 1\}^n$

For $i = 1, \ldots, m$

$C[i] \leftarrow E_K(C[i-1] \oplus M[i])$

Return $C$

**Algorithm** $D_K(C)$

For $i = 1, \ldots, m$

$M[i] \leftarrow E_K^{-1}(C[i]) \oplus C[i-1]$

Return $C$

**Question:** Is CBC encryption INT-CTXT secure?

**Answer:** No, because any string $C[0]C[1] \ldots C[m]$ has a valid decryption.

Mihir Bellare  
UCSD  
11

Encryption with Redundancy

Here $E: \{0, 1\}^k \times \{0, 1\}^m \rightarrow \{0, 1\}^n$ is our block cipher and $h: \{0, 1\}^* \rightarrow \{0, 1\}^n$ is a “redundancy” function, for example

- $h(M[1] \ldots M[m]) = 0^n$
- $h(M[1] \ldots M[m]) = M[1] \oplus \cdots \oplus M[m]$
- A CRC
  - $h(M[1] \ldots M[m])$ is the first $n$ bits of SHA1($M[1] \ldots M[m]$).

The redundancy is verified upon decryption.

Mihir Bellare  
UCSD  
12
Encryption with Redundancy

Let $E: \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be our block cipher and $h: \{0,1\}^n \rightarrow \{0,1\}^n$ a redundancy function. Let $SE = (K, E', D')$ be CBC encryption and define the encryption with redundancy scheme $AE = (K, E, D)$ via

\[
\text{Alg } E_K(M) \\
M[1] \ldots M[m] \leftarrow M \\
M[m+1] \leftarrow h(M) \\
C \leftarrow E'_K(M[1] \ldots M[m]M[m+1]) \\
\text{return } C
\]

\[
\text{Alg } D_K(C) \\
M[1] \ldots M[m]M[m+1] \leftarrow D'_K(C) \\
\text{if } (M[m+1] = h(M)) \text{ then } \text{return } M[1] \ldots M[m] \\
\text{else return } \perp
\]

Arguments in Favor of Encryption with Redundancy

The adversary will have a hard time producing the last enciphered block of a new message.

Encryption with Redundancy Fails

WEP Attack

A “real-life” rendition of this attack broke the 802.11 WEP protocol, which instantiated $h$ as CRC and used a stream cipher for encryption [BGW]. What makes the attack easy to see is having a clear, strong and formal security model.

This attack succeeds for any (not secret-key dependent) redundancy function $h$. 

Mihir Bellare UCSD 13
Mihir Bellare UCSD 14
Mihir Bellare UCSD 15
Mihir Bellare UCSD 16
Build an authenticated encryption scheme $\mathcal{AE} = (K, E, D)$ by combining
- a given IND-CPA symmetric encryption scheme $\mathcal{SE} = (K', E', D')$
- a given PRF $F : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^n$

| CBC$\$-AES | CTR$\$-AES | ...
|----------------|----------------|---
| HMAC-SHA1      |                |   
| CMAC           |                |   
| ECBC           |                |   
|                |                |   

A key $K = K_e || K_m$ for $\mathcal{AE}$ always consists of a key $K_e$ for $\mathcal{SE}$ and a key $K_m$ for $F$:

$\textbf{Alg } K$

$K_e \xleftarrow{\$} K'$; $K_m \xleftarrow{\$} \{0, 1\}^k$

Return $K_e || K_m$

The order in which the primitives are applied is important. Can consider

<table>
<thead>
<tr>
<th>Method</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Encrypt-and-MAC (E&amp;M)</td>
<td>SSH</td>
</tr>
<tr>
<td>MAC-then-encrypt (MtE)</td>
<td>SSL/TLS</td>
</tr>
<tr>
<td>Encrypt-then-MAC (EtM)</td>
<td>IPSec</td>
</tr>
</tbody>
</table>

We study these following [BN].
Encrypt-and-MAC

\[ \mathcal{AE} = (K, \mathcal{E}, \mathcal{D}) \text{ is defined by} \]

\[
\begin{align*}
\text{Alg } E_{K_{||K_m}}(M) & \quad \text{Alg } D_{K_{||K_m}}(C'||T) \\
C' & \leftarrow E'_{K_0}(M) & M & \leftarrow D'_{K_0}(C') \\
T & \leftarrow F_{K_m}(M) & \text{if } (T = F_{K_m}(M)) \text{ then return } M \\
\text{Return } C'||T & \text{else return } \bot
\end{align*}
\]

Security | Achieved? \\
--- | --- \\
IND-CPA | NO \\
INT-CTX | NO

Why? \( T = F_{K_m}(M) \) is a deterministic function of \( M \) and allows detection of repeats.

MAC-then-Encrypt

\[ \mathcal{AE} = (K, \mathcal{E}, \mathcal{D}) \text{ is defined by} \]

\[
\begin{align*}
\text{Alg } E_{K_{||K_m}}(M) & \quad \text{Alg } D_{K_{||K_m}}(C'||T) \\
C' & \leftarrow E'_{K_0}(M) & M & \leftarrow D'_{K_0}(C') \\
T & \leftarrow F_{K_m}(M) & \text{if } (T = F_{K_m}(M)) \text{ then return } M \\
\text{Return } C'||T & \text{else return } \bot
\end{align*}
\]

Security | Achieved? \\
--- | --- \\
IND-CPA | NO \\
INT-CTX | NO

Security Achieved?

IND-CPA | NO \\
INT-CTX | NO

Why? May be able to modify \( C' \) in such a way that its decryption is unchanged.
MAC-then-Encrypt

\( AE = (K, \mathcal{E}, D) \) is defined by

- \( \text{Alg } \mathcal{E}_{K_a \| K_m}(M) \)
  - \( T \leftarrow F_{K_a}(M) \)
  - \( C \leftarrow \mathcal{E}'_{K_a}(M \| T) \)
  - Return \( C \)

- \( \text{Alg } D_{K_a \| K_m}(C) \)
  - \( M \leftarrow T \leftarrow D'_{K_a}(C) \)
  - If \( (T = F_{K_m}(M)) \) then return \( M \)
  - Else return \( \bot \)

<table>
<thead>
<tr>
<th>Security</th>
<th>Achieved?</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND-CPA</td>
<td>YES</td>
</tr>
<tr>
<td>INT-CTX</td>
<td>NO</td>
</tr>
</tbody>
</table>

**Why?** \( SE' = (K', \mathcal{E}', D') \) is IND-CPA secure.

---

MAC-then-Encrypt

\( AE = (K, \mathcal{E}, D) \) is defined by

- \( \text{Alg } \mathcal{E}_{K_a}(M) \)
  - \( T \leftarrow F_{K_a}(M) \)
  - \( C \leftarrow \mathcal{E}'_{K_a}(M \| T) \)
  - Return \( C \)

- \( \text{Alg } D_{K_a}(C) \)
  - \( M \leftarrow T \leftarrow D'_{K_a}(C) \)
  - If \( (T = F_{K_m}(M)) \) then return \( M \)
  - Else return \( \bot \)

<table>
<thead>
<tr>
<th>Security</th>
<th>Achieved?</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND-CPA</td>
<td>YES</td>
</tr>
<tr>
<td>INT-CTX</td>
<td>NO</td>
</tr>
</tbody>
</table>

**Why?** May be able to modify \( C \) in such a way that its decryption is unchanged.

---

Encrypt-then-MAC

\( AE = (K, \mathcal{E}, D) \) is defined by

- \( \text{Alg } \mathcal{E}_{K_a \| K_m}(M) \)
  - \( T \leftarrow F_{K_a \| K_m}(M) \)
  - \( C \leftarrow \mathcal{E}'_{K_a}(M \| T) \)
  - Return \( C \)

- \( \text{Alg } D_{K_a \| K_m}(C \| T) \)
  - \( M \leftarrow T \leftarrow D'_{K_a}(C) \)
  - If \( (T = F_{K_m}(M)) \) then return \( M \)
  - Else return \( \bot \)

<table>
<thead>
<tr>
<th>Security</th>
<th>Achieved?</th>
</tr>
</thead>
<tbody>
<tr>
<td>IND-CPA</td>
<td>YES</td>
</tr>
<tr>
<td>INT-CTX</td>
<td>NO</td>
</tr>
</tbody>
</table>
Encrypt-then-MAC

$\mathcal{AE} = (K, E, D)$ is defined by

| $\text{Alg } E_{K_e||K_m}(M)$ | $\text{Alg } D_{K_e||K_m}(C'\|T)$ |
|-------------------------------|-----------------------------------|
| $C' \leftarrow E_{K_e}(M)$   | $M \leftarrow D_{K_e}(C')$       |
| $T \leftarrow F_{K_m}(C')$   | If $(T = F_{K_m}(C'))$ then return $M$ |
| Return $C'\|T$               | Else return $\bot$               |

Security | Achieved?   \\
|----------|-------------|
| IND-CPA  | YES         \\
| INT-CTX  |             |

Why? $SE' = (K', E', D')$ is IND-CPA secure.

---

Encrypt-then-MAC

$\mathcal{AE} = (K, E, D)$ is defined by

| $\text{Alg } E_{K_e||K_m}(M)$ | $\text{Alg } D_{K_e||K_m}(C'\|T)$ |
|-------------------------------|-----------------------------------|
| $C' \leftarrow E_{K_e}(M)$   | $M \leftarrow D_{K_e}(C')$       |
| $T \leftarrow F_{K_m}(C')$   | If $(T = F_{K_m}(C'))$ then return $M$ |
| Return $C'\|T$               | Else return $\bot$               |

Security | Achieved?   \\
|----------|-------------|
| IND-CPA  | YES         \\
| INT-CTX  |             |

Two keys or one?

We have used separate keys $K_e, K_m$ for the encryption and message authentication. However, these can be derived from a single key $K$ via $K_e = F_K(0)$ and $K_m = F_K(1)$, where $F$ is a PRF such as a block cipher, the CBC-MAC or HMAC.

Trying to directly use the same key for the encryption and message authentication is error-prone, but works if done correctly.

Why? If $C\|T$ is new then $T$ will be wrong.
Exercise

Let $E = AES$. Let $K$ return a random 128-bit AES key $K$. Let $S\mathcal{E} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ where $\mathcal{E}, \mathcal{D}$ are below. Here, $X[i]$ denotes the $i$-th 128-bit block of a string whose length is a multiple of 128.

\begin{align*}
\text{Alg } \mathcal{E}_K(M) & \quad \text{if } |M| \neq 512 \text{ then return } \bot \quad \\
M[1] \ldots M[4] & \leftarrow M \\
C_e[0] & \leftarrow \{0, 1\}^{128} \\
C_m[0] & \leftarrow 0^{128} \\
\text{for } i = 1, \ldots, 4 \text{ do } & \\
C_e[i] & \leftarrow E_K(C_e[i-1] \oplus M[i]) \\
C_m[i] & \leftarrow E_K(C_m[i-1] \oplus M[i]) \\
T & \leftarrow C_m[4]; \text{ return } (C_e, T)
\end{align*}

\begin{align*}
\text{Alg } \mathcal{D}_K((C_e, T)) & \quad \text{if } |C_e| \neq 640 \text{ then return } \bot \quad \\
C_m[0] & \leftarrow 0^{128} \\
\text{for } i = 1, \ldots, 4 \text{ do } & \\
M[i] & \leftarrow E_K^{-1}(C_e[i]) \oplus C_e[i-1] \\
C_m[i] & \leftarrow E_K(C_m[i-1] \oplus M[i]) \\
\text{if } C_m[4] \neq T \text{ then return } \bot & \\
\text{return } M
\end{align*}

1. Is $S\mathcal{E}$ IND-CPA-secure? Why or why not?
2. Is $S\mathcal{E}$ INT-CTXT-secure? Why or why not?
3. Is $S\mathcal{E}$ an Encrypt-and-MAC construction? Justify your answer.

Exercise

INT-CTXT security of Encrypt-then-MAC

You are given
- An IND-CPA symmetric encryption scheme $S\mathcal{E}^* = (\mathcal{K}^*, \mathcal{E}^*, \mathcal{D}^*)$
- A PRF $F: \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^n$

Construct a symmetric encryption scheme $S\mathcal{E}' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ such that
1. $S\mathcal{E}'$ is IND-CPA, but
2. The MtE combination of $S\mathcal{E}'$ and $F$ is not INT-CTXT-secure.

Specify $S\mathcal{E}'$ by giving pseudocode for all the constituent algorithms.

Then prove (1) by a reduction and prove (2) by giving pseudocode for an efficient adversary achieving int-ctxt advantage 1.

Encrypt-then-MAC is INT-CTXT-secure assuming PRF-security of $F$:

**Theorem:** Let $S\mathcal{E} = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ be a symmetric encryption scheme. Let $F: \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^n$ be a family of functions. Let $S\mathcal{E} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be obtained by composing $S\mathcal{E}'$ and $F$ in the Encrypt-then-MAC combination. Let $A$ be an int-ctxt adversary against $S\mathcal{E}$ make $q_e$ Enc queries and having running time $t$. Then we can construct a prf-adversary $B$ against $F$ such that

$$\text{Adv}^{\text{int-ctxt}}_{S\mathcal{E}}(A) \leq \text{Adv}^{\text{prf}}_{F}(B) + \frac{1}{2^n}.$$ 

$B$ makes $q_e$ queries to its $\text{Fn}$ oracle and has running time $t$ plus some overhead.
The adversary $B$

**adversary $B$**

$K_e \leftarrow K' \; \; ; \; S \leftarrow \emptyset$

$C' \parallel T \leftarrow A_{EncSim}$

If $(C', T) \in S$ then return 0

If $T = \text{Fn}(C')$ then return 1

Else return 0

**Subroutine** $\text{EncSim}(M)$

$C' \leftarrow \mathcal{E}'(K_e, M) \; ; \; T \leftarrow \text{Fn}(C')$

$S \leftarrow S \cup \{(C', T')\}$

Return $C' \parallel T$

Note that $B$ itself picks $K_e$ so that it can simulate $\text{Enc}$ for $A$.

\[
\begin{align*}
\Pr[\text{Real}_F \Rightarrow 1] &= \text{Adv}_{\mathcal{A}E}^{\text{int-ctxt}}(A) \\
\Pr[\text{Rand}_{\{0,1\}^n} \Rightarrow 1] &\leq \frac{1}{2^n}
\end{align*}
\]

Exercise: IND-CPA security of Encrypt-then-MAC

Encrypt-then-MAC is IND-CPA-secure assuming IND-CPA-security of $\mathcal{S}\mathcal{E}'$:

**Theorem:** Let $\mathcal{S}\mathcal{E} = (K', \mathcal{E}', \mathcal{D}')$ be a symmetric encryption scheme. Let $F : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a family of functions. Let $\mathcal{A}\mathcal{E} = (K, \mathcal{E}, \mathcal{D})$ be obtained by composing $\mathcal{S}\mathcal{E}$ and $F$ in the Encrypt-then-MAC combination. Let $A$ be an ind-cpa adversary against $\mathcal{A}\mathcal{E}$ make $q$ LR queries and having running time $t$. Then we can construct an ind-cpa adversary $B$ against $\mathcal{S}\mathcal{E}'$ such that

\[
\text{Adv}_{\mathcal{A}\mathcal{E}}^{\text{ind-cpa}}(A) \leq \text{Adv}_{\mathcal{S}\mathcal{E}'}^{\text{ind-cpa}}(B).
\]

$B$ makes $q$ queries to its LR oracle and has running time $t$ plus some overhead.

The exercise is to prove this theorem.

Exercise

There is a lot going on in the above proof! The exercise is to work through it slowly, checking each step and claim.

Generic Composition in Practice

<table>
<thead>
<tr>
<th>AE in</th>
<th>is based on</th>
<th>which in general is</th>
<th>and in this case is</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSH</td>
<td>E&amp;M</td>
<td>insecure</td>
<td>secure</td>
</tr>
<tr>
<td>SSL</td>
<td>MtE</td>
<td>insecure</td>
<td>insecure</td>
</tr>
<tr>
<td>SSL + RFC 4344</td>
<td>MtE</td>
<td>insecure</td>
<td>secure</td>
</tr>
<tr>
<td>IPSec</td>
<td>EtM</td>
<td>secure</td>
<td>secure</td>
</tr>
<tr>
<td>WinZip</td>
<td>EtM</td>
<td>secure</td>
<td>insecure</td>
</tr>
</tbody>
</table>

Why?

- Encodings
- Specific “E” and “M” schemes
- For WinZip, disparity between usage and security model
AE in SSH

SSH2 encryption uses inter-packet chaining which is insecure [D, BKN]. RFC 4344 [BKN] proposed fixes that render SSH provably IND-CPA + INT-CTXT secure. Fixes recommended by Secure Shell Working Group and included in OpenSSH since 2003. Fixes included in PuTTY since 2008.

AE in SSL

SSL uses MTE

$$\varepsilon_{K_e}||\varepsilon_{K_m} = \varepsilon_{K_m}^{\prime}(M||F_{K_m}(M))$$

which we saw is not INT-CTXT-secure in general. But $\varepsilon^{\prime}$ is CBC$\$ in SSL, and in this case the scheme does achieve INT-CTXT [K].

$F$ in SSL is HMAC.

Sometimes SSL uses RC4 for encryption.

AEAD

The goal has evolved into Authenticated Encryption with Associated Data (AEAD) [Ro].

- Associated Data (AD) is authenticated but not encrypted
- Schemes are nonce-based (and deterministic)

Sender: Has $K, N, AD$
- $C \leftarrow \varepsilon_K(N, AD, M)$
- Send $C$

Receiver: Has $K, N, AD$
- Receive $C$
- $M \leftarrow D_K(N, AD, C)$

Sender must never re-use a nonce.

But when attacking integrity, the adversary may use any nonce it likes.

AEAD Schemes

Generic composition: E&M, MtE, EtM extend and again EtM is the best but others work too under appropriate conditions [NRS14].

1-pass schemes: IAPM [J], XCBC/XEBC [GD], OCB [RBBK, R]

2-pass schemes: CCM [FHW], EAX [BRW], CWC [KVW], GCM [MV]

Stream cipher based: Helix [FWSKLK], SOBER-128 [HR]

- 1-pass schemes are fast
- 2-pass schemes are patent-free
- Stream cipher based schemes are fast
OCB [RBBK]

\[
\begin{align*}
M[1] & \xrightarrow{E_K^{N,1,0}} C[1] \\
\text{Checksum} & \xrightarrow{E_K^{N,1,1}} S \\
& \text{ } \\
& \text{ } \\
& \text{ } \\
& \text{ } \\
& \text{ } \\
& C[4]
\end{align*}
\]

Checksum = \(M[1] \oplus M[2] \oplus M[3]\)

\(S = \text{PMAC}_K(AD)\) using separate tweaks.

Output may optionally be truncated.

Some complications (not shown) for non-full messages.

Optional in IEEE 802.11i

OCB2 broken despite proof, OCB and OCB3 standing.

Patents on 1-pass schemes

- Jutla (IBM) 7093126
- Gligor and Donescu (VDG, Inc.) 6973187
- Rogaway 7046802, 7200227

2-pass AEAD

- Tailored generic composition of specific base schemes
- Single key

Philosophical questions:
- What is the advantage of one key versus two given that can always derive the two from the one?
- Why not just do specific generic composition of specific base schemes?

CCM [FHW]

MtE-based but single key throughout. CTR-ENC is nonce-based counter mode encryption, and CBC-MAC is the basic CBC MAC. Ciphertext is \(C||T\). In NIST SP 800-38C, IEEE 802.11i.
GCM [MV]

CTR-ENC is nonce-based counter mode encryption. GCM-HASH is a polynomial-based hash. $K_H$ is derived from $K$ via $E$. Can be used as a MAC. In NIST SP 800-38D.

Performance Comparisons x32

No clear answer. Ask yourself
- What performance do I need?
- Single or multiple keys?
- Patents ok or not?
- Do I need to comply with some standard?

Gladman’s C code

Gladman’s C code

Which AEAD scheme should I use?
Authenticated encryption today

- Dedicated schemes: OCB, OCBx (x=1,2,3), GCM, CCM, EAX
- TLS uses GCM
- CAESAR competition to standardize new schemes:
  http://competitions.cr.yp.to/caesar.html