Lecture 9 — Authenticated Encryption

COSC-260 Codes and Ciphers
Adam O’Neill

Adapted from http://cseweb.ucsd.edu/~mihir/cse107/
Setting the Stage

We have previously studied the goals of privacy and authenticity in isolation.
We have previously studied the goals of privacy and authenticity in isolation.

- **Privacy**: Symmetric-key encryption and IND-CPA security.
Setting the Stage

We have previously studied the goals of privacy and authenticity in isolation.

- **Privacy**: Symmetric-key encryption and IND-CPA security.
- **Authenticity**: Message Authentication Codes and UF-CMA security.
Setting the Stage

We have previously studied the goals of privacy and authenticity in isolation.

• Privacy: Symmetric-key encryption and IND-CPA security.

• Authenticity: Message Authentication Codes and UF-CMA security.

But many (most) applications require both.
Medical Example

Privacy: Nobody but Alice can read her medical record.

Authenticity: Alice is assured that only her doctor modified her medical record.
Authenticated Encryption (AE)

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

- $K$ is a key-generation algorithm that outputs a key $K$
- $E$ is an encryption algorithm that on inputs a key $K$ and a message $M$ outputs a ciphertext $C$.
- $D$ is a decryption algorithm that on inputs a key $K$ and ciphertext $C$ outputs a message $M$ or $1$. $HKM \ D_k^E_k(C) = M$
Privacy of AE

just the standard IND-CPA definition.
Integrity of AE (INT-CTXT)

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

The int-ctxt advantage of $A$ is

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

We are interested in constructing symmetric-key encryption schemes that are both
Our Goal

We are interested in constructing symmetric-key encryption schemes that are both

• IND-CPA (to provide privacy).
Our Goal

We are interested in constructing symmetric-key encryption schemes that are both:

- **IND-CPA** (to provide privacy).
- **INT-CTXT** (to provide integrity).
Plain Encryption Doesn’t Work

**Alg** $\mathcal{E}_K(M)$

\[
\begin{align*}
\text{C}[0] & \leftarrow \{0, 1\}^n \\
\text{For } i = 1, \ldots, m \text{ do} \\
& \quad \text{C}[i] \leftarrow E_K(\text{C}[i-1] \oplus M[i]) \\
\text{Return } C
\end{align*}
\]

**Alg** $\mathcal{D}_K(C)$

\[
\begin{align*}
\text{For } i = 1, \ldots, m \text{ do} \\
& \quad M[i] \leftarrow E_K^{-1}(\text{C}[i]) \oplus C[i-1] \\
\text{Return } M
\end{align*}
\]

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

*No!* Decryption doesn’t return 1.
The Attack
Encryption with Redundancy

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

\begin{align*}
\textbf{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
\end{align*}

\begin{align*}
\textbf{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 } \bot
\end{align*}
The Attack

Adversary A

Let $M$ be arbitrary

$c[0] \rightarrow \cdots \rightarrow c[m+2] \leftarrow Enc(M||h(m))$

ret $c[0] \rightarrow \cdots \rightarrow c[m+1]$
WEP Attack

In around 2000 the 802.11 protocol WEP was attacked exactly for this reason; it used a particular CRC as the redundancy.
Generic Composition

Build an authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ by combining

- a given IND-CPA symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$
- a given PRF $F : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^n$
Generic Composition

Build an authenticated encryption scheme $\mathcal{AE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ by combining

- a given IND-CPA symmetric encryption scheme $\mathcal{SE} = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$
- a given PRF $F : \{0, 1\}^k \times \{0, 1\}^* \rightarrow \{0, 1\}^n$

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

\[
\begin{align*}
\text{Alg } &\mathcal{K} \\
K_e &\leftarrow^* \mathcal{K}'
\end{align*}
\]

Return $K_e \parallel K_m$
Possibilities

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

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

$\textbf{Alg}$ $\mathcal{E}_{K_e \| K_m}(M)$

\[
\begin{align*}
C' & \leftarrow \mathcal{E}'_{K_e}(M) \\
T & \leftarrow \mathcal{F}_{K_m}(M) \\
\text{Return } & C' \| T
\end{align*}
\]

$\textbf{Alg}$ $\mathcal{D}_{K_e \| K_m}(C' \| T)$

\[
\begin{align*}
M & \leftarrow \mathcal{D}'_{K_e}(C') \\
\text{If } (T = & \mathcal{F}_{K_m}(M)) \text{ then return } M \\
\text{Else return } & \bot
\end{align*}
\]
Security Analysis

IND-CPA

INT-CTX

Adversary $A$

Let $m_1, m_2$ be arbitrary messages.

$c_{11}, c_{12} \leftarrow LR(m_1, m_2)$
$c_{21}, c_{22} \leftarrow LR(m_1, m_1)$

IF $c_{22} = c_{22}$ then $0$ ELSE set $1$
MAC-then-Encrypt

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

\[
\begin{align*}
\text{Alg } & \mathcal{E}_{K_e\|K_m}(M) \\
& T \leftarrow F_{K_m}(M) \\
& C \leftarrow \mathcal{E}'_{K_e}(M\|T) \\
& \text{Return } C
\end{align*}
\]

\[
\begin{align*}
\text{Alg } & \mathcal{D}_{K_e\|K_m}(C) \\
& M\|T \leftarrow \mathcal{D}'_{K_e}(C) \\
& \text{If } (T = F_{K_m}(M)) \text{ then return } M \\
& \text{Else return } \bot
\end{align*}
\]
Idea: Let $SE=(K,E,D)$ be any IND-CPA secure encryption scheme and let $F:K\times\{0,1\}^n\rightarrow\{0,1\}^*$ be a secure IND-CPA secure AE symmetric encryption scheme such that MAC-then-encrypt of $F$ and $SE'$ is not INT-CTXT.
Encrypt-then-MAC

$AE = (K, E, D)$ is defined by

\[
\begin{align*}
\text{Alg} & \ E_{K_e || K_m}(M) \\
C' & \leftarrow E'_{K_e}(M) \\
T & \leftarrow F_{K_m}(C') \\
\text{Return } & \ C' || T
\end{align*}
\]

\[
\begin{align*}
\text{Alg} & \ D_{K_e || K_m}(C' || T) \\
M & \leftarrow D'_{K_e}(C') \\
\text{If } (T = F_{K_m}(C')) & \text{ then return } M \\
\text{Else return } & \bot
\end{align*}
\]
One can prove that Encrypt-then-MAC yields an IND-CPA + INT-CTXT authenticated encryption scheme on any IND-CPA secure symmetric-key encryption scheme & secure PRF.
# 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
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

\[ E_{K_e\|K_M} = E'_{K_e}(M\|F_{K_m}(M)) \]

which we saw is not INT-CTXT-secure in general. But \( E' \) 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.
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
- $C \leftarrow \mathcal{E}_K(N, AD, M)$
- Send $(N, AD, C)$

Receiver
- Receive $(N, AD, C)$
- $M \leftarrow \mathcal{D}_K(N, AD, C)$

Sender must never re-use a nonce.

But when attacking integrity, the adversary may use any nonce it likes.
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
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
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 [FWSKLLK], SOBER-128 [HR]

- 1-pass schemes are fast
- 2-pass schemes are patent-free
- Stream cipher based schemes are fast