Introduction to Modern Cryptography

Michele Ciampi

(Slides courtesy of Prof. Jonathan Katz)

Lecture 11, Part 2
Authenticated Encryption
Secrecy and Integrity Combined?

- **Secrecy**: PRF/block cipher in a mode of operation
- **Integrity**: message authentication code (MAC)

**Question**

Can we combine both secrecy and integrity in a single private-key scheme?
Constructions

Three natural approaches

1. Encrypt-and-authenticate (E-and-A)
2. Authenticate-then-encrypt (A-then-E)
3. Encrypt-then-authenticate (E-then-A)
Encrypt-and-authenticate (E-and-A)

Sender and receiver share two keys: $k_1$ for encryption, $k_2$ for authentication

$k_1, k_2$

\[ m \]  
\[ c \leftarrow \text{Enc}_{k_1}(m) \]  
\[ t = \text{Mac}_{k_2}(m) \]

$c, t$

$m = \text{Dec}_{k_1}(c)$  
$\text{Vrfy}_{k_2}(m, t) = 1?$

$k_1, k_2$
Encrypt-and-authenticate (E-and-A)

- Sender sends $c = \text{Enc}_{k_1}(m)$, $t = \text{Mac}_{k_2}(m)$
- Receiver decrypts $m = \text{Dec}_{k_1}(c)$ and verifies $\text{Vrfy}_{k_2}(m, t) = 1$
E-and-A Weaknesses

Not CPA-secure

If the MAC is deterministic (as is CBC-MAC), then the tag leaks whether the same message is encrypted twice

- i.e. E-and-A will not be CPA-secure, even if $\text{Enc}$ is CPA-secure
E-and-A Weaknesses

Not EAV-secure

The tag \( t \) might leak information about \( m \)
- Nothing in the definition of security for a MAC implies that it hides information about \( m \)
- E-and-A may not even be EAV-secure

Example

- Let \( \Pi = (\text{Gen}, \text{Mac}, \text{Vrfy}) \) be a secure MAC
- Define \( \text{Mac}'_k = (m, \text{Mac}_k(m)) \)
- \( \implies \Pi' = (\text{Gen}, \text{Mac}', \text{Vrfy}) \) is a secure MAC
- \( \Pi' \) reveals \( m \) \( \implies \) E-and-A using \( \Pi' \) is not CPA-secure
Authenticate-then-encrypt (A-then-E)

Sender and receiver share two keys: $k_1$ for encryption, $k_2$ for authentication.

$t = \text{Mac}_{k_2}(m)$
$c \leftarrow \text{Enc}_{k_1}(m \mid t)$

$m \mid t = \text{Dec}_{k_1}(c)$
$\text{Vrfy}_{k_2}(m, t) = 1$
Authenticate-then-encrypt (A-then-E)

Sender computes tag \( t = \text{Mac}_{k_2}(m) \) and sends \( c = \text{Enc}_{k_1}(m | t) \)

Receiver decrypts \((m, t) = \text{Dec}_{k_1}(c)\) and verifies \( \text{Vrfy}_{k_2}(m, t) = 1 \)
A-then-E Weaknesses

Problems with A-then-E

- Padding-oracle attack
- Other counter-examples are also possible
  - The combination may not be CCA-secure
A-then-E: Padding Oracle Attack

A-then-E scheme $\Pi$

- Encode $m$ applying $T(m)$ as:
  - replace 0 $\rightarrow$ 00, replace 1 $\rightarrow$ 01 or 10
- Decode $m$ from $T(m)$ as:
  - replace 00 $\rightarrow$ 0, replace 01 or 10 $\rightarrow$ 1
  - if 11 return ⊥ (error)
- Let $\text{Enc}$ be a cipher that generates a PR sequence and XORs it with $m$
  - e.g. PRF/block cipher in CTR mode
- Define $\text{Enc}'_k(m) = \text{Enc}_k(T(m))$
- Let $\Pi$ be an A-then-E scheme using $\text{Enc}'$
A-then-E: Padding Oracle Attack

---

Padding-oracle attack on $\Pi$

- $A$ attacks $\Pi$ following the CCA experiment
- $A$ gets challenge $c = \text{Enc}_{k_1}^{'}(T(m, \text{Mac}_{k_2}(m)))$
- $A$ flips first 2 bits of $c$ to get $c'$
- $A$ submits $c'$ to the decryption oracle $O$
- If $O$ returns $\perp$ $\implies$ $A$ infers first bit of $c$ to be 0
- Otherwise $A$ infers the first bit of $c$ to be 1
- $\implies \Pi$ not CCA-secure
Encrypt-then-authenticate (E-then-A)

Sender and receiver share two keys: \(k_1\) for encryption, \(k_2\) for authentication
Encrypt-then-authenticate (E-then-A)

- Sender sends $c = \text{Enc}_{k_1}(m)$, $t = \text{Mac}_{k_2}(c)$
- Receiver verifies $\text{Vrfy}_{k_2}(c, t) = 1$ and (if $t$ is valid) decrypts $m = \text{Dec}_{k_1}(c)$
Security of E-then-A

<table>
<thead>
<tr>
<th>Theorem</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>If the underlying encryption scheme is CPA-secure and the MAC is secure (i.e. existentially unforgeable) then the E-then-A combination is a CCA-secure encryption scheme</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Proof</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Omitted</em></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>The encryption and authentication keys $k_1$ and $k_2$ must be independent</td>
</tr>
</tbody>
</table>
A CCA-secure Scheme

**Encrypt-then-authenticate**

E-and-A is the right way to combine secrecy with integrity:

- Use a CPA-secure encryption scheme to encrypt the message
- Use a MAC to prevent the ciphertext from being modified
A stronger notion than CCA

Observation

The E-then-A approach results in a stronger notion than CCA-security:

- The MAC is applied on the ciphertext produced by the sender.
- The adversary is not able to obtain any valid ciphertext that was not generated by the legitimate parties, thus rendering the decryption oracle useless.
- This property is not implied by CCA-security, where the attacker is allowed to query the decryption oracle on any chosen ciphertexts and receive the corresponding plaintexts.
Authenticated Encryption

A stronger property than CCA

Given ciphertexts \((c_1, t_1), (c_2, t_2), \ldots\) corresponding to (chosen) plaintexts \(m_1, m_2, \ldots\), it is infeasible for an attacker to generate any new valid ciphertext \((c, t)\).

- i.e. if an attacker injects his own ciphertext, the decryption oracle will output an error (rather than the corresponding plaintext)

Authenticated encryption (AE) scheme

Schemes with the above property are called authenticated encryption schemes
Authenticated Encryption

Theorem

If the underlying encryption scheme is CPA-secure and the MAC is secure then the E-then-A combination is an AE scheme.

E-then-A is the recommended generic approach to constructing an AE scheme.

- “Generic” = using any CPA-secure scheme and any secure MAC.
Direct AE Constructions

Other, more-efficient AE constructions exist:

- OCB, CCM, GCM
- Finalists from the CAESAR competition
  > [https://competitions.cr.yp.to/caesar-submissions.html](https://competitions.cr.yp.to/caesar-submissions.html)
End of Symmetric-key Part
Summary of Symmetric-key Topics

- Historical ciphers: Shift cipher, Vigenère
- Perfect secrecy
- One-time pad (OTP)
- Computational secrecy
- Pseudorandom generators (PRG)
- Pseudo-OTP
- Security against chosen-plaintext attacks (CPA)
- Pseudorandom functions / permutations (PRF / PRP)
Summary of Symmetric-key Topics

- CPA-secure encryption using PRF/PRP
- Modes of operation: block ciphers
- Malleability
- Security against chosen-ciphertext attacks (CCA)
- Non-CCA secure schemes: padding-oracle attacks
- Secrecy vs. integrity: message authentication codes (MAC)
- Hash functions
- Secrecy and integrity; authenticated encryption
What next?

<table>
<thead>
<tr>
<th>Observe</th>
<th>The security of symmetric-key schemes ultimately depends on the <strong>secrecy of the key</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Problem</td>
<td>How do we distribute the keys in the first place?</td>
</tr>
<tr>
<td>Solution</td>
<td>Public-key cryptography.</td>
</tr>
</tbody>
</table>
End

References: Sec 4.5.1, 4.5.2 (not Theorem 4.19)