Integrity: MACs & CCA security

Malleability: get $C = Enc(k, m)$ for unknown $m$, transform it into $C'$ so that $Dec(k, c')$ is related to $m$ in a predictable way

Ex: ECB, CBC, OFB

Attacks:
1. Server sends "authentication token" $C = Enc(k, m)$, Attacker obtains $c'$ which is authentication token for another user!
2. Server can tell you whether $Dec(k, c)$ is a valid XML file. Can send many related ciphertexts $c'$ to server and actually learn entire contents of $Dec(k, c)$

MAC: message authentication code

$MAC(k, m) \rightarrow \text{"tag"}$

Idea: only the person who knows $k$ can come up with $(m, tag)$ s.t. $MAC(k, m) = tag$ "symmetric key analog of signatures"
Security game:

Adv gets to see $\text{MAC}(k, x)$ for many $x$ of its choice.

Adv outputs $(x^*, t^*)$ and wins the game if:

1. $\text{MAC}(k, x^*) = t^*$
2. $x^*$ was not queried earlier

MAC is secure $\iff$ No Adv can win this game except w/ exponential effort.

Note: $t$ doesn't hide $x$.

Constructions:

CBC-MAC:

Do CBC encryption with all-zeros IV, output last block only.

Only for same-length messages (seeing MACs on 1-block messages, can forge MAC on 2-block message).
HMAC: uses hash function
\[ H(K_1 \ || \ H(K_2 \ || \ m)) \]
- derived from master key \( K \)

Naive MAC: \( H(k \ || \ m) \)
- depending on \( H \), suffers from length-extension attack
  - seeing MAC of \( m \) \( \Rightarrow \) can forge MAC on a longer msg that has \( m \) as prefix.

---

Why does CPA security not capture the XML-enc attack?

CPA:
- Adv can request \( Enc(k,m) \) for any \( m \)
- Adv asks for \((m_0, m_1)\)
  - receives \( C^* = \begin{cases} \text{Enc}(k,m_0) \text{ or } \text{Enc}(k,m_1) \end{cases} \)
- Adv can request \( Enc(k,m) \) for any \( m \)
- Adv tries to guess contents of \( C^* \)

XML attack: Server's behavior depends on result of decrypting adversarially-generated ciphertexts
Adv learns partial information about dec.

**CCA:**
- Adv can request $\text{Enc}(k, m)$ for any $m$
- Adv can request $\text{Dec}(k, c)$ for any $c$
- Adv asks for $(m_0, m_1)$
  - receives $C^* = \begin{cases} \text{Enc}(k, m_0) \quad \text{or} \\ \text{Enc}(k, m_1) \end{cases}$
- Adv can request $\text{Enc}(k, m)$ for any $m$
- Adv can request $\text{Dec}(k, c)$ for any $c \neq C^*$
- Adv tries to guess contents of $C^*$

**Note:** Can combine CPA-secure enc + MAC = CCA secure enc

Can use block cipher modes that directly give CCA security (authenticated modes)

**CPA**

"Semantic security"

**CCA**

"Authenticated encryption"