Today:

Continue symmetric encryption & authentication

- Cipher Block Chaining (CBC) mode
- CCA security
- Message Authentication Codes (MACs)

Begin

- Finite fields
- Shamir Secret Sharing

Cipher Block Chaining (CBC) Mode

Let \((E_k, D_k)\) be a block cipher

\[
\begin{align*}
\text{CBC:} & & \downarrow \quad & \downarrow & \downarrow & \downarrow & \downarrow \\
IV \quad & \rightarrow & m_0 & m_1 & m_2 & \cdots \\
\downarrow \quad & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
\text{random} & \quad & \downarrow & \downarrow & \downarrow & \downarrow \\
\text{initial value} & \quad & \downarrow & \downarrow & \downarrow & \downarrow \\
& \quad & E_k & E_k & E_k & \cdots \\
& \quad & C_0 & C_1 & C_2 & \cdots \\
\end{align*}
\]

Output IV, \((c_0, c_1, c_2, \ldots)\)

* If msg is not of length which is a multiple of
block length then pad (e.g., add 10...0 to each msg).

**Decrypt:** Using $D_k = E_k^{-1}$ in the obvious way.

**Claim:** If $(E_k, D_k)$ is an ideal block cipher (i.e., random permutation), then CBC mode is a CPA secure encryption scheme (assuming IV is random).

**Key generation alg**

**Def:** An encryption scheme $(Gen, Enc, Dec)$ is CPA secure if for all $m_0, m_1$, s.t. $\|m_0\| = \|m_1\|$, and efficient $A$

$$A^{Enc_k(Enc_k(m_0))} \neq A^{Enc_k(Enc_k(m_1))}$$

A is given black-box access to $Enc_k$.

Actually, A can choose $m_0, m_1$ after querying $Enc_k$

Intuitively: CBC enc. of any msg $(m_0, m_1, ...)$ is a random CT $(c_0, c_1, ...)$ ind. of $(m_0, m_1, ...)$ if $(E_k, D_k)$ is an ideal cipher.

Randomness of IV is needed to argue that it remains random even given oracle access to $Enc_k$. 

\[\text{L7.2}\]
A stronger notion of security: **CCA security**

Chosen Ciphertext Attack

**Def:** An encryption scheme \((Gen, Enc, Dec)\) is \(\text{CCA secure}\) if \(\forall M_0, M_1\) s.t. \(|M_0| = |M_1|\)

\(\forall\) efficient \(A\)

\[ A^{Enc_k, Dec_k} (Enc_k(m_0)) \approx A^{Enc_k, Dec_k} (Enc_k(m_1)) \]

\(A\) is given black-box access to both \(Enc_k\) & \(Dec_k\) for \(k \leftarrow Gen\).

Moreover \(A\) can choose \(M_0\) & \(M_1\) after querying \(Enc_k\) & \(Dec_k\), but cannot send its exact input \(Enc_k(M_0)\) as oracle query to \(Dec_k\).

**Claim:** CBC is **not** CCA secure.

(and neither are ECB or CTR)

**Pf:** \(A\) picks \(M_0 = 0^N\) & \(m_1 = 1^N\)

Given \(C \leftarrow Enc_k(M_0)\) let \(C' = 1^{\text{st half of the bits of } C}\) (w. same IV).
A queries $\text{Dec}_k$ with $c'$ (this is allowed since) $c' \neq c$

which gives $1^{st}$ half of the bits of $m_b$, revealing $b$.

**How do we design CCA secure schemes?**

1. Construct a CPA secure scheme (e.g. CBC).
2. Add authentication (so that $\text{Dec}_k$ will only decrypt msgs that are "authenticated")

**Message Authentication Code (MAC)**

Provides integrity (authenticity), not confidentiality.

Alice \[ m, \text{MAC}_k(m) \rightarrow \text{Bob} \]

Bob recomputes \[ \text{MAC}_k(m) \] and checks that it agrees with what they received. If not, reject.

- Allows Bob to verify that $m$ originated from Alice, and arrived unmodified.
- Alice & Bob need to share a secret key.
- orthogonal to confidentiality. Typically we do both
  (encrypt & append MAC on the ciphertext) for
  integrity.

**Security of MAC**

**Def**: A MAC is **secure against adaptive chosen msg attacks** if & only if a given pairs
\((m_i, \text{MAC}_k(m_i))\) for any msgs \(m_i\) of their
choice, cannot generate any new \(m'_i\) with
valid \(\text{MAC}_k(m'_i)\).

(Jumping ahead: MACs are like digital signatures but in
the symmetric key setting)

**Note**: If MAC generates tags of length \(t\), then
\(\text{Adv}\) can guess w.p. \(2^{-t}\). Therefore \(t\) needs
to be sufficiently large.

**Thm**: CPA secure encryption scheme + secure MAC \(\Rightarrow\) CCA secure encryption scheme
Intuitively, adding a MAC to the ciphertexts makes the decryption oracle useless.

**How to construct a MAC**

1. From hash functions (HMAC)
2. From block ciphers (CBC-MAC or CMAC).

**CBC-MAC**

**CBC-MAC**<sub>k</sub>(m): Encrypt m w. CBC mode with IV = 0 & output only last cipher but the key <sub>k</sub> used for the last block is different from the key <sub>k</sub> used for all other blocks!

(Both <sub>k</sub>1 & <sub>k</sub>2 are random & ind.)

**HW**

Why does <sub>k</sub>2 need to be different than <sub>k</sub>1?

Why isn’t IV random?
Finite Fields & Shamir Secret Sharing

**Def:** A field is defined by a tuple \((S, +, \cdot)\) s.t.

* \(S\) is a set containing "0" & "1".

* \((S, +)\) is an abelian (commutative) group with identity 0:
  \[
  \begin{cases}
  (a+b) + c = a + (b+c) & \forall a, b, c \in S \quad \text{(associative)} \\
  a+0 = 0+a = a & \forall a \in S \quad \text{(identity 0)} \\
  \forall a \in S \exists b \in S \text{ st. } a+b = 0 & \text{(inverse)} \\
  a+b = b+a & \forall a, b \in S \quad \text{(commutative)}
  \end{cases}
  \]

* \((S^\circ, \cdot)\) is an abelian (commutative) group with identity 1:
  \[
  S^\circ = S \setminus \{0\}
  \]
  \[
  \begin{cases}
  (a \cdot b) \cdot c = a \cdot (b \cdot c) & \forall a, b, c \in S \quad \text{(associative)} \\
  a \cdot 1 = 1 \cdot a = a & \forall a \in S \quad \text{(identity 1)} \\
  \forall a \in S^\circ \exists b \in S^\circ \text{ st. } a \cdot b = 1 & \text{(inverse)} \\
  a \cdot b = b \cdot a & \text{(commutative)}
  \end{cases}
  \]
Examples

$\mathbb{R}$ (reals) \{ familiar fields.

$\mathbb{C}$ (complex)

These are finite fields (i.e., fields w. infinitely many elements).

In crypto, we usually work w. finite fields, where $|\mathbb{F}|$ is finite.

Example: $(\mathbb{Z}_p, +, \cdot)$ where $+, \cdot$ are mod $p$.

\[ \mathbb{Z}_p = \{ 0, 1, \ldots, p-1 \} \]

Thm: There is a finite field $\mathbb{F}$ w. $q$ elements if and only if

$q = p^k$ for some prime $p$ and integer $k \geq 1$.

Moreover, for every such $q$, there is a unique field consisting of $q$ elements, denoted by $\mathbb{GF}(q)$.

$\mathbb{GF}(p)$ for prime $p$ is $(\mathbb{Z}_p, +, \cdot)$, where $+ \& \cdot$ are mod $p$. 

Galois Field