Symmetric Key Cryptosystems

- Public key cryptosystems (PKCs) and their applications are the primary focus of this course.
- But symmetric key cryptosystems are still an important tool.
- As we’ll see, PKCs are too slow for bulk data.
- But PKCs solve the key distribution problems for symmetric key systems.

Two sorts of modern symmetric key systems

Stream Ciphers
- Plaintext = stream of bits or chars
- Key = a matching stream of keys
- E.g.: One-time pad, LFSRS-pad

Block Ciphers
- Plaintext, key = a block of bits
- Long messages are broken into many blocks
- E.g.: DES, 3DES, AES

Claude Shannon’s Principles for a Good Cipher

- Diffusion Changing one character in the plaintext changes many characters in the ciphertext.
- Confusion Each part of the ciphertext should depend on several parts of the key
Plaintexts = Ciphertexts = \{0, 1\}^{64}.

**DES:**
\[\{0, 1\}^{64} \times \{0, 1\}^{56} \rightarrow \{0, 1\}^{64}\]

- From a 56 bit key \(K\), 16 other 48 bit keys are generated: \(K_1, \ldots, K_{16}\)
- Key \(K_i\) is used in round \(i\) of the encryption.
- \(\text{DES}_K = \text{DES}_{K_1, \ldots, K_{16}} = \text{DES}\) with fixed key \(K\) (or subkeys \(K_1, \ldots, K_{16}\))
- \(F: \{0, 1\}^{32} \times \{0, 1\}^{48} \rightarrow \{0, 1\}^{32}\) the Feistel function for DES
- \(\text{IP} = \text{initial permutation}\)
- \(\text{FP} = \text{final permutation} = \text{IP}^{-1}\)

**The DES Round Function**

\[\varphi_{i, \mu}: \{0, 1\}^{32} \times \{0, 1\}^{32} \rightarrow \{0, 1\}^{32} \times \{0, 1\}^{32}\]

\[\varphi_i(x, y) \text{ def } (x \oplus F_i(y), y) \quad \mu(x, y) \text{ def } (y, x)\]

\[\text{DES}_{K_1, \ldots, K_{16}}(x) = \text{IP}^{-1} \circ \varphi_{16} \circ \mu \circ \varphi_{15} \circ \mu \circ \cdots \circ \mu \circ \varphi_1 \circ \text{IP}(x)\]

\[\text{DES}_{K_{16}, \ldots, K_1}(x) = \text{IP}^{-1} \circ \varphi_1 \circ \mu \circ \varphi_2 \circ \mu \circ \cdots \circ \mu \circ \varphi_{16} \circ \text{IP}(x)\]

**Claim.**
- \(\text{IP}(\text{IP}^{-1}(x)) = x\)
- \(\mu(\mu(x, y)) = (x, y)\).
- \(\varphi_i(\varphi_i(x, y)) = (x, y)\).
- \(\text{DES}_{K_{16}, \ldots, K_1}(\text{DES}_{K_1, \ldots, K_{16}}(x)) = x\).

**Definition 1.**
A block cipher with a round function of the form \(\varphi(x, y) = (x \oplus f_i(y), y)\) is called a Feistel cipher.

**DES animation:** [http://kathrynneugent.com/des.html](http://kathrynneugent.com/des.html)
DES and AES

- DES is a much studied and much fought over cipher
  The fights are a good paper topic.
- DES’s block and key size are too small
  Attacks on DES are another good paper topic.
- 3DES – triple DES reasonable alternative
- Advanced Encryption Standard (AES) - Oct 2000
  - winner of a competition to replace DES
  - based on arithmetic over $\mathbb{F}_{2^8}$
  - key sizes 128, 192, and 256
  - compromise on speed and security
  The AES competition is yet another good paper topic.

Games you can play with block ciphers, I

Setup

\( f_k : \{0, 1\}^\ell \rightarrow \{0, 1\}^\ell \) :: a block cipher with fixed key \( k \)
\( m = m_1 \ldots m_r \) :: blocks of equal length \( \ell \)
\( (m_r \text{ padded out if need be}) \)

Electronic Code Book Mode

\[ \text{ecbEncrypt}(m) \]
Chop \( m \) into blocks \( m_1, \ldots, m_r \)
for \( i \leftarrow 1 \) to \( r \) do \( c_i \leftarrow f_k(m_i) \)
return \( c_1, \ldots, c_r \)

No sensible people use this mode!!!

Games you can play with block ciphers, II

Cipher-Block Chaining Mode

\[ \text{ecbEncrypt}(m) \]
Select \( c_0 \overset{\text{ran}}{\in} \{0, 1\}^\ell \)
Chop \( m \) into \( \ell \)-bit blocks \( m_1, \ldots, m_r \)
for \( i \leftarrow 1 \) to \( r \) do
  \( c_i \leftarrow f_k(m_i \oplus c_{i-1}) \)
return \( c_0, c_1, \ldots, c_r \)

\[ \text{ecbDecrypt}(c) \]
Chop \( c \) into \( \ell \)-bit blocks \( c_0, \ldots, c_r \)
for \( i \leftarrow 1 \) to \( r \) do
  \( m_i \leftarrow f_k^{-1}(c_i) \oplus c_{i-1} \)
return \( m_1, \ldots, m_r \)

Note: For simplicity, I’m ignoring IV’s (= initialization vectors) in the pseudo-code for block cipher modes.

- \( f_k \) is the block cipher encryption function with key \( k \).
- \( f_k^{-1} \) is the block cipher decryption function with key \( k \).

Stronger, but noise is a problem. (Why?)

Problems with ECB Mode: An illustration

![orginal](image1)
![ECB-encrypted](image2)
![securely encrypted](image3)

Identical plaintext blocks are encrypted into identical ciphertext blocks.
Cipher Feedback Mode

Suppose $x = x_1 \ldots x_k \in \{0,1\}^k$

- $\text{msb}_\ell(x) = \text{def } x_1 \ldots x_\ell$  the $\ell$ most significant bits
- $\text{lsb}_\ell(x) = \text{def } x_{k-\ell+1} \ldots x_k$  the $\ell$ least significant bits
- $x_1 \ldots x_\ell \mathbin| y_1 \ldots y_m = \text{def } x_1 \ldots x_k y_1 \ldots y_m$  string concatenation

\[
\text{cfbEncrypt}(m, x_1) = \text{def } x_1 \text{ ran } \in \{0,1\}^\ell - \text{public}
\]

Chop $m$ into $\ell$-bit blocks $m_1, \ldots, m_r$

for $i \leftarrow 1$ to $r$

\[
c_i \leftarrow m_i \oplus \text{msb}(f_k(x_i)) \quad \text{// } f_k \text{ used as a PRG}
\]

\[
x_{i+1} \leftarrow \text{lsb}_{\ell-8}(x_i) \mathbin| c_i
\]

return $c_1, \ldots, c_r$

This behaves much better with errors in transmission. (Why?)


### Counter Mode

\[
\text{crtEncrypt}(m, x_1) = \begin{cases} 
\text{ran} & x_1 \in \{0, 1\}^\ell - \text{public} \\
\text{Chop } m \text{ into } 8\text{-bit blocks } m_1, \ldots, m_r \\
\text{for } i \leftarrow 1 \text{ to } r \\
\quad o_i \leftarrow \text{msb}_8(f_k(x_i)) \\
\quad c_i \leftarrow m_i \oplus o_i \\
\quad x_{i+1} \leftarrow (x_i + 1) \mod 2^\ell \\
\text{return } c_1, \ldots, c_r
\end{cases}
\]

This behaves much better with errors in transmission. (Why?)