Content of this part

♦ Encryption with Block Ciphers: Modes of Operation
  • Electronic Code Book mode (ECB)
  • Cipher Block Chaining mode (CBC)
  • Output Feedback mode (OFB)
  • Cipher Feedback mode (CFB)
  • Counter mode (CTR)
  • Galois Counter Mode (GCM)
♦ Exhaustive Key Search Revisited
♦ Increasing the Security of Block Ciphers
♦ Cryptanalysis principles
Block Ciphers

♦ A block cipher is much more than just an encryption algorithm, it can be used ...
  • to build different types of block-based encryption schemes
  • to realize stream ciphers
  • to construct hash functions
  • to make message authentication codes
  • to build key establishment protocols
  • to make a pseudo-random number generator
  • ...
♦ The security of block ciphers also can be increased by
  • key whitening
  • multiple encryption

Encryption with Block Ciphers

♦ There are several ways of encrypting long plaintexts, e.g., an e-mail or a computer file, with a block cipher ("modes of operation")

♦ These modes of operation have three goals:
  • In addition to confidentiality, some of them provide authenticity and integrity:
    » Is the message really coming from the original sender? (authenticity)
    » Was the ciphertext altered during transmission? (integrity)
Electronic Code Book mode (ECB)

- \( e_k(x_i) = \) encryption of a \( b \)-bit plaintext block \( x_i \) with key \( k \)
- \( e_k^{-1}(y_i) = \) decryption of \( b \)-bit ciphertext block \( y_i \) with key \( k \)
- Messages which exceed \( b \) bits are partitioned into \( b \)-bit blocks
- Each Block is encrypted separately

\[
\begin{array}{ccc}
X_1 & & X_i \\
\downarrow & & \downarrow \\
e_k & & e_k \\
Y_1 & & Y_i \\
\end{array}
\]

- Padding of last block:
  - Include non-data values (e.g., Null)
  - Include number of bytes in padding
  - And/or number of plaintext bytes

ECB: advantages/disadvantages

- **Advantages**
  - no block synchronization between sender and receiver is required
  - bit errors caused by noisy channels only affect the corresponding block but not succeeding blocks
  - Block cipher operations can be parallelized
    » advantage for high-speed implementations

- **Disadvantages**
  - ECB encryption highly deterministic
    » identical plaintexts result in identical ciphertexts
    » an attacker recognizes if the same message has been sent twice
    » plaintext blocks are encrypted independently of previous blocks
      - an attacker may reorder ciphertext blocks which results in valid but incorrect plaintext
Substitution Attack on ECB

- Once a particular plaintext to ciphertext block mapping $x_i \rightarrow y_i$ is known, a sequence of ciphertext blocks can easily be manipulated.

- Suppose an electronic bank transfer:

<table>
<thead>
<tr>
<th>Block #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sending Bank A</td>
<td>Sending Account #</td>
<td>Receiving Bank B</td>
<td>Receiving Account #</td>
<td>Amount $</td>
<td></td>
</tr>
</tbody>
</table>

- The encryption key between banks does not change frequently.
- The attacker sends $1 transfers from his account at bank A to his account at bank B several times.
  - He can check for ciphertext blocks that repeat, and he stores blocks 1, 3, and 4 of these transfers.
- He can now replace block 4 of other transfers (having the same blocks 1 & 3) with the block 4 that he stored before.
  - All transfers from some account of bank A to some account of bank B are redirected to go into the attacker’s B account.
  - Integrity violation – does not break the cipher.

Example of encrypting bitmaps in ECB mode

- Identical plaintexts are mapped to identical ciphertexts.

- Statistical properties in the plaintext are preserved in the ciphertext.
Cipher Block Chaining mode (CBC)

♦ There are two main ideas behind the CBC mode:
  ♦ The encryption of all blocks are “chained together”
    » ciphertext $y_i$ depends not only on block $x_i$ but on all previous plaintext blocks as well
  ♦ The encryption is randomized by using an initialization vector (IV)

Encryption (first block): $y_1 = e_k(x_1 \oplus IV)$
Encryption (general block): $y_i = e_k(x_i \oplus y_{i-1}), \ i \geq 2$
Decryption (first block): $x_1 = e^{-1}_k(y_1) \oplus IV$
Decryption (general block): $x_i = e^{-1}_k(y_i) \oplus y_{i-1}, \ i \geq 2$

 Cipher Block Chaining mode (CBC)

♦ For the 1st plaintext block $x_1$ there is no previous ciphertext
  ♦ an IV is added to the first plaintext to make each CBC encryption nondeterministic
  ♦ the first ciphertext $y_1$ depends on plaintext $x_1$ and the IV
♦ The 2nd ciphertext $y_2$ depends on the IV, $x_1$ and $x_2$
♦ The 3rd ciphertext $y_3$ depends on the IV and $x_1$, $x_2$ and $x_3$, and so on

> Image of a flowchart illustrating the encryption and decryption processes for CBC mode.
### Substitution Attack on CBC

- Consider the last example:
  - If the IV is properly chosen for every wire transfer, the attack will not work at all
  - If the IV is kept the same for several transfers, the attacker would recognize the transfers from his account at bank A to bank B
  - If we choose a new IV every time we encrypt, the CBC mode becomes a probabilistic encryption scheme, i.e., two encryptions of the same plaintext look entirely different
  - It is not needed to keep the IV secret - the IV should be a non-secret nonce (value used only once)
  - Integrity can still be violated (e.g., replacing blocks 4&5)
  - Can not be encrypted in parallel, decryption?
  - If a bit in a block is flipped in transmission all remaining blocks are garbled

<table>
<thead>
<tr>
<th>Block #</th>
<th>Sending Bank A</th>
<th>Sending Account #</th>
<th>Receiving Bank B</th>
<th>Receiving Account #</th>
<th>Amount</th>
</tr>
</thead>
</table>

### Output Feedback mode (OFB)

- Used to build a synchronous stream cipher from a block cipher
- Key stream is not generated bitwise but in a block-wise fashion
- The cipher output generates key stream bits $S_i$
- Key stream independent of ciphertext - can be generated in parallel

**Encryption (first block):** $s_1 = e_k(IV)$ and $y_1 = s_1 \oplus x_1$

**Encryption (general block):** $s_i = e_k(s_{i-1})$ and $y_i = s_i \oplus x_i$, $i \geq 2$

**Decryption (first block):** $s_1 = e_k(IV)$ and $x_1 = s_1 \oplus y_1$

**Decryption (general block):** $s_i = e_k(s_{i-1})$ and $x_i = s_i \oplus y_i$, $i \geq 2$
Cipher Feedback mode (CFB)

- Uses a block cipher as a building block for asynchronous stream cipher (similar to OFB mode), better name: “Ciphertext Feedback Mode”
- Key stream $S_i$ generated in a block-wise fashion and is also a function of the ciphertext
- By using IV, CFB encryption is also nondeterministic

\[ y_1 = e_k(IV) \oplus x_1 \]
\[ y_i = e_k(y_{i-1}) \oplus x_i, \quad i \geq 2 \]
\[ x_1 = e_k(IV) \oplus y_1 \]
\[ x_i = e_k(y_{i-1}) \oplus y_i, \quad i \geq 2 \]

- Can be used when short plaintext blocks (e.g., byte) are to be encrypted - shift input to the left and use the 8-bit ciphertext

CFB – byte at a time

- If one ciphertext byte is “lost” all remaining bytes will be decrypted incorrectly.
**Counter mode (CTR)**

- A block cipher generates a **stream cipher** (like OFB and CFB)
- The key stream is computed in a block-wise fashion
- The input to the block cipher is a counter which assumes a different value every time the block cipher computes a new key stream block
  - Shorter IV
  - Need not to be replaced every time
- Unlike OFB and CFB modes, the CTR mode can be parallelized since the 2nd encryption can begin before the 1st one has finished
  - Desirable for high-speed implementations, e.g., in network routers

\[
\begin{align*}
\text{Encryption:} & \quad y_i = e_k (\text{IV} \ || \ \text{CTR}_i) \oplus x_i, \quad i \geq 1 \\
\text{Decryption:} & \quad x_i = e_k (\text{IV} \ || \ \text{CTR}_i) \oplus y_i, \quad i \geq 1
\end{align*}
\]

- AES-CTR is the basis for Wi-fi Protected Access WPA2

---

**Galois Counter Mode (GCM)**

- It also computes a **message authentication code (MAC)**, i.e., a cryptographic checksum is computed for a message (see Chapter 12 in *Understanding Cryptography*)
- By making use of GCM, two additional services are provided:
  - **Message Authentication**
    - the receiver can make sure that the message was really created by the original sender
  - **Message Integrity**
    - the receiver can make sure that nobody tampered with the ciphertext during transmission
- **Inputs:** plaintext \((x_i)\), IV and AAD = Additional Authentication Data
  - IV and AAD transmitted in the clear
  - AAD can include network address, port, sequence number etc
- **Outputs:** ciphertext \((y_i)\) and authentication tag \(T\)
Galois Counter Mode (GCM)

- For encryption
  - An initial counter is derived from an IV
  - The initial counter value is incremented then encrypted and XORed with the first plaintext block
  - For subsequent plaintexts, the counter is incremented and then encrypted

- For authentication
  - For every plaintext an intermediate authentication parameter \( g_i \) is derived
    - \( g_i \) is computed as the XOR of the current ciphertext and the last \( g_{i-1} \), and multiplied by a constant \( H \)
    - \( H \) is generated by encryption of the zero input with the block cipher
  - Initial \( g \): \( g_0 = H \times AAD \) (Galois field multiplication)
  - All multiplications are in the 128-bit Galois field \( GF(2^{128}) \) mod the irreducible polynomial
    \[ P(x) = x^{128} + x^7 + x^2 + x + 1 \]

---

Encryption:
a. Derive a counter value \( CTR_0 \) from the IV and compute \( CTR_i = CTR_{i-1} + 1 \)
b. Compute ciphertext: \( y_i = e_k(CTR_i) \oplus x_i, \quad i \geq 1 \)

Authentication:
a. Generate authentication subkey \( H = e_k(0) \)
b. Compute \( g_0 = AAD \times H \) (Galois field multiplication)
c. Compute \( g_i = (g_{i-1} \oplus y_i) \times H, \quad 1 \leq i \leq n \) (Galois field multiplication)
d. Final authentication tag: \( T = (g_n \times H) \oplus e_k(CTR_n) \)
**Exhaustive Key Search Revisited**

♦ A simple exhaustive search for a DES key knowing one pair $(x_1, y_1)$:

$$DES_k^{(i)}(x_1) \stackrel{?}{=} y_i \quad i = 0, 1, \ldots, 2^{56} - 1$$

♦ However, for most other block ciphers a key search is somewhat more complicated

♦ A brute-force attack can produce **false positive** results

```
\begin{array}{c}
\text{x}_1 \\
\vdots \\
\text{y}_1 \\
\end{array}
```

- The likelihood of this is related to the relative size of the key space and the plaintext space
- A brute-force attack is still possible, but several pairs of plaintext-ciphertext are needed

---

**An Exhaustive Key Search Example**

♦ Assume cipher block width 64 bit and key size 80 bit

♦ If we encrypt $x_1$ under all possible $2^{80}$ keys, we obtain $2^{80}$ ciphertexts

- However, there exist only $2^{64}$ different ones

♦ If we run through all keys for a given plaintext-ciphertext pair, we find up to $2^{80} / 2^{64} = 2^{16}$ keys that perform the mapping $e_k(x_1) = y_1$

```
\begin{array}{c}
\text{x}_1 \\
\vdots \\
\text{y}_1 \\
\end{array}
```

Given a block cipher with a key length of $k$ bits and block size of $n$ bits, as well as $t$ plaintext-ciphertext pairs $(x_1, y_1), \ldots, (x_t, y_t)$, the expected number of **false** keys which encrypt all plaintexts to the corresponding ciphertexts is: $2^{k-n}$

♦ In this example assuming two plaintext-ciphertext pairs, the likelihood is $\frac{2^{80} - 2^{64}}{2} = 2^{48}$

- for almost all practical purposes two plaintext-ciphertext pairs are sufficient
Increasing the Security of Block Ciphers

- If we wish to increase the security of block ciphers, e.g., DES available in hardware for legacy reasons

- Two approaches are possible
  - Multiple encryption: theoretically more secure, but in practice may increase the security very little
  - Key whitening

**Double Encryption:**

- A plaintext $x$ is first encrypted with key $k_L$, and the resulting ciphertext is encrypted again using a 2nd key $k_R$

- Assuming a key length of $k$ bits, an exhaustive key search would require $2^k \cdot 2^k = 2^{2k}$ encryptions or decryptions

---

Meet-in-the-Middle Attack

- A Meet-in-the-Middle attack requires $2^k + 2^k = 2^{k+1}$ operations

- **Phase I:** for the given $(x_i, y_i)$ the left encryption is brute-forced for all $k_L, i=1,2,...,2^k$ and a lookup table with $2^k$ entry (each $n+k$ bits wide) is computed
  - the lookup table should be ordered by the result of the encryption ($z_{L,i}$)

- **Phase II:** the right encryption is brute-forced (using decryption) and for each $z_{R,i}$ it is checked whether $z_{R,i}$ is equal to any $z_{L,i}$ value in the table of the first phase

- **Computational Complexity**
  - number of encryptions and decryptions = $2^k + 2^k = 2^{k+1}$
  - number of storage locations = $2^k$

- Double encryption is not much more secure then single encryption
Triple Encryption

- Encryption of a block three times $y = e_{k3}(e_{k2}(e_{k1}(x)))$
- In practice a variant scheme is often used EDE (encrypt-decrypt-encrypt) $y = e_{k3}(e^{-1}_{k2}(e_{k1}(x)))$
  - Advantage: choosing $k1=k2=k3$ performs single DES encryption
- Still we can perform a meet-in-the-middle attack, and it reduces the effective key length of triple encryption from $3k$ to $2k$
- The attacker must run $2^{112}$ tests in the case of 3DES
- Triple encryption effectively doubles the key length

Key Whitening

- Makes block ciphers such as DES much more resistant against brute-force attacks
- In addition to the regular cipher key $k$, two whitening keys $k_1$ and $k_2$ are used to XOR-mask the plaintext and ciphertext
- It does not strengthen block ciphers against most analytical attacks such as linear and differential cryptanalysis
- It is not a "cure" for inherently weak ciphers
- The additional computational load is negligible
- Its main application is ciphers that are relatively strong against analytical attacks but possess too short a key space especially DES
  - A variant of DES which uses key whitening is called DESX
Cryptanalysis

- **Linear cryptanalysis** - search for an approximated linear relation among bits of plaintext-ciphertext pairs and bits of the key due to imperfect cipher structure
  - DES: $2^{43}$ known plaintext-ciphertext pairs to find the key (1993)
- **Differential cryptanalysis** - track how differences between two plaintexts affect the corresponding difference between the ciphertexts. Search for non-random behavior
  - DES: $2^{47}$ plaintext-ciphertext pairs to find the key (1990)

Lessons Learned

- There are many different ways to encrypt with a block cipher. Each mode of operation has some advantages and disadvantages
- Several modes turn a block cipher into a stream cipher
- There are modes that perform encryption together with authentication, i.e., a cryptographic checksum protects against message manipulation
- The straightforward ECB mode has security weaknesses, independent of the underlying block cipher
- The counter mode allows parallelization of encryption and is thus suited for high speed implementations
- Double encryption with a given block cipher only marginally improves the resistance against brute-force attacks
- Triple encryption with a given block cipher roughly *doubles* the key length, e.g., Triple DES (3DES) has an effective key length of 112 bits
- Key whitening enlarges the DES key length without much computational overhead.