Symmetric Key Encryption II

Anders Västberg

vastberg@kth.se

Slides by M. Goodrich, R. Tamassia , Göran Andersson

Symmetric Key Cryptography

Recap

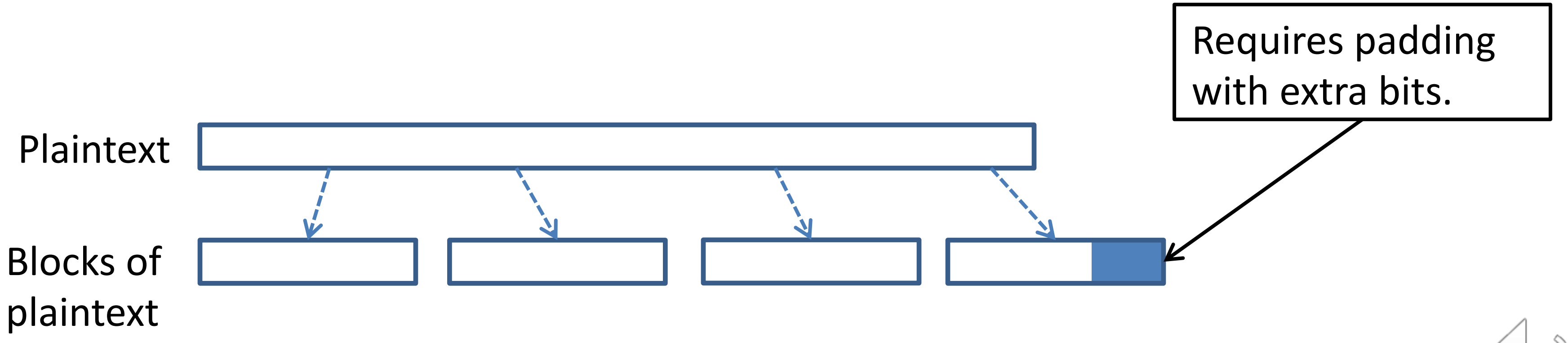
- Substitution Ciphers
- S-boxes
- Polyalphabetic Ciphers
- One-time pad
- Stream ciphers

Symmetric Cryptography

- Block ciphers
  - Padding
- Hill cipher
- Transposition cipher
- AES
- CBC
- Practical examples

Block Ciphers

- In a **block cipher:**
  - Plaintext and ciphertext have fixed length b (e.g., 128 bits)
  - A plaintext of length n is partitioned into a sequence of m **blocks**, $M[0], \ldots, M[m-1]$, where $n \le bm < n + b$
- Each message is divided into a sequence of blocks and encrypted or decrypted in terms of its blocks.



Padding

- Block ciphers require the length n of the plaintext to be a multiple of the block size b
- Padding the last block needs to be unambiguous (cannot just add zeroes)
- When the block size and plaintext length are a multiple of 8, a common padding method (PKCS5) is a sequence of identical bytes, each indicating the length (in bytes) of the padding.
- We need to always pad the last block, which may consist only of padding
- PKCS5 assumes that the block size is 8 bytes, PKCS7 uses the same method, but with arbitrary number bytes in the blocks.
- Example for $b = 128$ (16 bytes)
  - Plaintext: “Roberto” (7 bytes)
  - Padded plaintext: “Roberto999999999” (16 bytes), where 9 denotes the number and not the character

Problem

- A plain text of 220 characters is coded by 8 bit ASCII code and divided into 128 bit blocks. The last block is padded by PKCS7. How is the last block padded?

- Solution

$220 \cdot 8 = 1760$ bits
$\lfloor 1760/128 \rfloor = 13$ blocks
$1760 - 13 \cdot 128 = 96$ bits of plain text in the last block
$\frac{96}{8} = 12$ bytes in the last block
The padding in the last block is in the form of 4 bytes with the value 4.

The Hill Cipher

- Block cipher invented Lester Hill 1929
- English letters are treated as numbers $\bmod 26$
- The key $\boldsymbol{K}$ is an invertible $n \times n$ matrix $\bmod 26$
- Message partitioned in n-block (column vectors) and padded
- Encryption: $\boldsymbol{C} = \boldsymbol{K} \bullet \boldsymbol{M} \bmod 26$
- Decryption: $\boldsymbol{M} = \boldsymbol{D} \bullet \boldsymbol{C} \bmod 26$, where $\boldsymbol{D} = \boldsymbol{K}^{-1} \bmod 26$

- If $\boldsymbol{K}^{-1}$ and $d = \det(\boldsymbol{K})$ are known then
  $\boldsymbol{D} = [(d^{-1} \bmod 26)\ (d\ \boldsymbol{K}^{-1})] \bmod 26$
  as $d^{-1}\, d \pmod{26} = 1$

The Hill Cipher

Example:

$$\boldsymbol{K} = \begin{pmatrix} 1 & 0 & 11 \\ 11 & 16 & 24 \\ 7 & 17 & 1 \end{pmatrix}$$

$$d = 433 \Rightarrow d^{-1} \equiv_{26} 23$$

$$\text{check} : 433 \times 23 = 9959 \equiv_{26} 1$$

$$d\,\boldsymbol{K}^{-1} = \begin{pmatrix} -392 & 187 & -176 \\ 157 & -76 & 97 \\ 75 & -17 & 16 \end{pmatrix}$$

$$23\,d\,\boldsymbol{K}^{-1} = \begin{pmatrix} -9016 & 4301 & -4048 \\ 3611 & -1748 & 2231 \\ 1725 & -391 & 368 \end{pmatrix} \equiv_{26} \begin{pmatrix} 6 & 11 & 8 \\ 23 & 20 & 21 \\ 9 & 25 & 4 \end{pmatrix} = \boldsymbol{D}$$

The Hill Cipher

message = "CATANDHOUND"

$$M = \begin{pmatrix} 2 & 0 & 7 & 13 \\ 0 & 13 & 14 & 3 \\ 19 & 3 & 20 & 1 \end{pmatrix}$$

$$C = K \cdot M = \begin{pmatrix} 1 & 0 & 11 \\ 11 & 16 & 24 \\ 7 & 17 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & 0 & 7 & 13 \\ 0 & 13 & 14 & 3 \\ 19 & 3 & 20 & 1 \end{pmatrix} \equiv_{26} \begin{pmatrix} 3 & 7 & 19 & 24 \\ 10 & 20 & 1 & 7 \\ 7 & 16 & 21 & 13 \end{pmatrix}$$

cipher = "DKHHUQTBVYHN"

$$R = D \cdot C = \begin{pmatrix} 6 & 11 & 8 \\ 23 & 20 & 21 \\ 9 & 25 & 4 \end{pmatrix} \cdot \begin{pmatrix} 3 & 7 & 19 & 24 \\ 10 & 20 & 1 & 7 \\ 7 & 16 & 21 & 13 \end{pmatrix} \equiv_{26} \begin{pmatrix} 2 & 0 & 7 & 13 \\ 0 & 13 & 14 & 3 \\ 19 & 3 & 20 & 1 \end{pmatrix}$$

Problem

- Find the inverse key to

$$\begin{pmatrix} 0 & 15 \\ 1 & 5 \end{pmatrix}$$

- Solution

$$|\mathbb{K}| = \begin{vmatrix} 0 & 15 \\ 1 & 5 \end{vmatrix} = 0 \times 5 - 1 \times 15 = -15 \equiv_{26} 11$$

$$\mathbb{K}^{-1} = 11^{-1} \begin{pmatrix} 5 & -15 \\ -1 & 0 \end{pmatrix} \equiv_{26} -7 \begin{pmatrix} 5 & 11 \\ -1 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} -35 & -77 \\ 7 & 0 \end{pmatrix} \equiv_{26} \begin{pmatrix} 17 & 1 \\ 7 & 0 \end{pmatrix}$$

Transposition Cipher

- Plaintext shuffled around according to permutation
- The encryption key π consists of permutation cycles
- The decryption key is the inverse permutation π^{-1}
- Encryption: $C = \pi(M)$
- Decryption: $M = \pi^{-1}(C)$

Example:

$M =$ "CATANDHOUND"

$\pi = (1, 6, 11, 9, 8)\,(4, 7, 5)$

$C = \pi(M) =$ "OATNHCAUDND"

$\pi^{-1} = (1, 8, 9, 11, 6)\,(4, 5, 7))$

$M = \pi^{-1}(C) =$ "CATANDHOUND"

π not unique

$\pi = (1, 8, 9, 6)\,(4, 7, 5)$

another possibility

Explanation, C:

$M_1 \to M_6 \to M_{11} \to M_9 \to M_8 \to M_1$

$M_4 \to M_7 \to M_6 \to M_4$

M_2, M_3, M_{10} fixed

Attacks on Block Ciphers

- Both Hill Ciphers and transposition ciphers are susceptible to known plain text attacks.
- Hill Ciphers are linear and the encryption key can be found if having enough plain text and corresponding cipher text.
  - The encryption key can then be found by linear algebra.
- Transposition ciphers can be found by examining each position in the plain text in order.

DES, the Data Encryption Standard

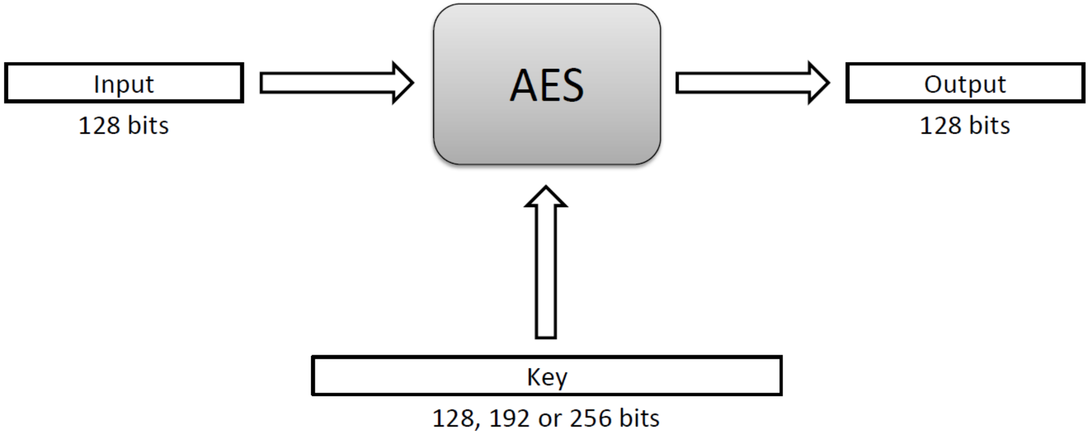
- Encryption standard used between 1975 and 2005.
- Initially thought of lasting 10-15 years but by regularly revision of the standard, it lasted much longer.
- Block encryption system of 2^{64} symbols.
- Key is only 56 bits, which means that its possible for a computer to test all 2^{56} possible keys in reasonable time.
- 1999 triple DES with three 56-bit keys

The Advanced Encryption Standard (AES)

In 1997, the U.S. National Institute for Standards and Technology (NIST) put out a public call for a replacement to DES.

It narrowed down the list of submissions to five finalists, and ultimately chose an algorithm that is now known as the **Advanced Encryption Standard** (**AES**).

AES is a block cipher that operates on 128-bit blocks. It is designed to be used with keys that are 128, 192, or 256 bits long, yielding ciphers known as AES-128, AES-192, and AES-256.



AES Round Structure

The 128-bit version of the AES encryption algorithm proceeds in ten rounds (10, 12 or 14).

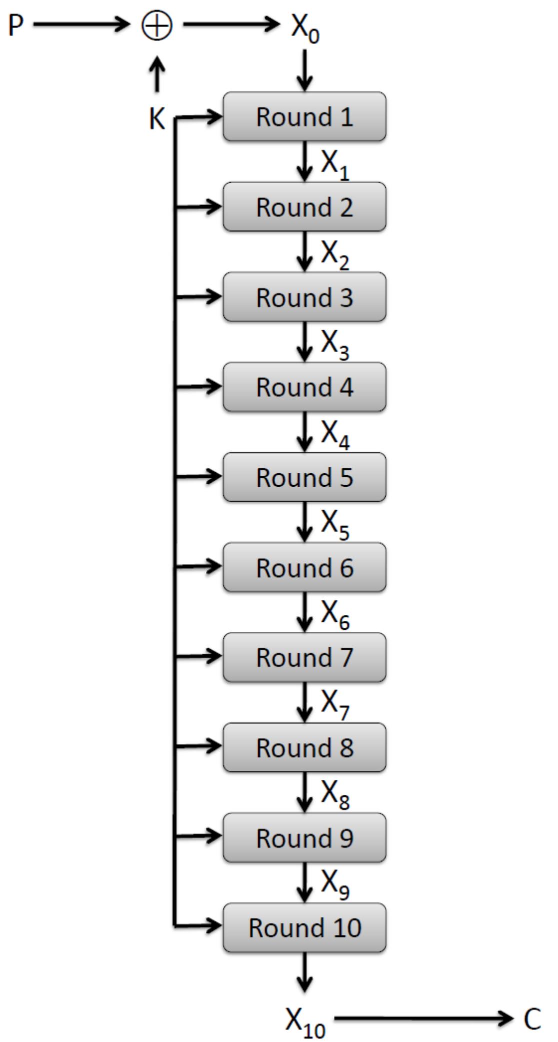
Each round performs an invertible transformation on a 128-bit array, called **state**.

The initial state X_0 is the XOR of the plaintext P with the key K:

$$X_0 = P \text{ XOR } K.$$

Round i (i = 1, …, 10) receives state X_{i-1} as input and produces state X_i.

The ciphertext C is the output of the final round: $C = X_{10}$.



AES Rounds

Each round is built from four basic steps:

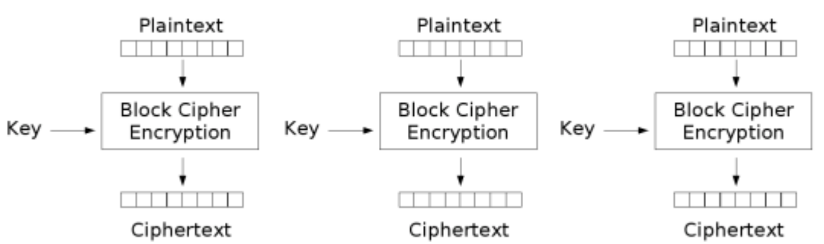
1. **SubBytes step**: an S-box substitution step
2. **ShiftRows step**: a permutation step
3. **MixColumns step**: a matrix multiplication step
4. **AddRoundKey step**: an XOR step with a **round key** derived from the 128-bit encryption key

Block Cipher Modes

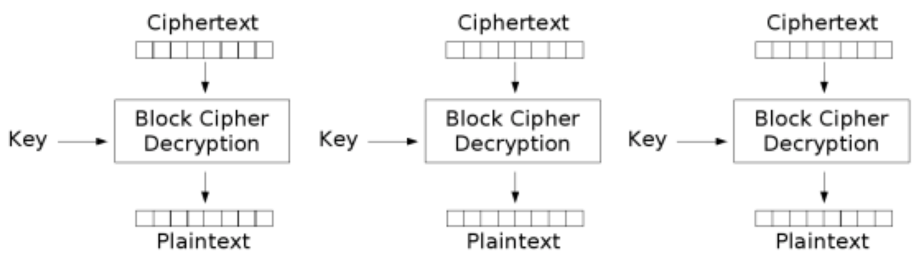
A block cipher mode describes the way a block cipher encrypts and decrypts a sequence of message blocks.

Electronic Code Book (ECB) Mode (is the simplest):

- Block P[i] encrypted into ciphertext block $C[i] = E_K(P[i])$
- Block C[i] decrypted into plaintext block $M[i] = D_K(C[i])$



Electronic Codebook (ECB) mode encryption



Electronic Codebook (ECB) mode decryption

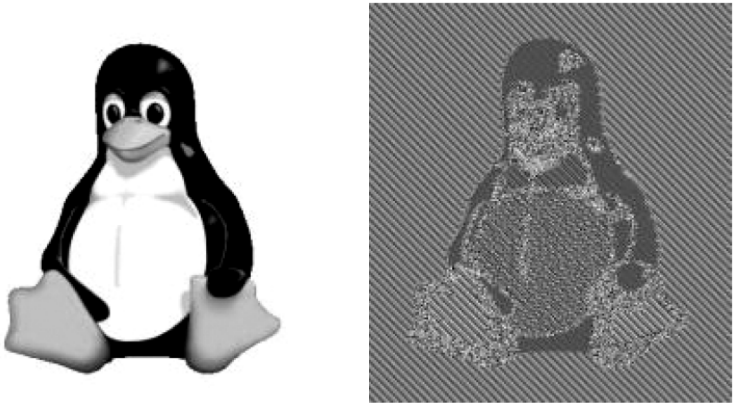
Strengths and Weaknesses of ECB

Strengths:

- Is very simple
- Allows for parallel encryptions of the blocks of a plaintext
- Can tolerate the loss or damage of a block

Weakness:

- Documents and images are not suitable for ECB encryption since patters in the plaintext are repeated in the ciphertext:



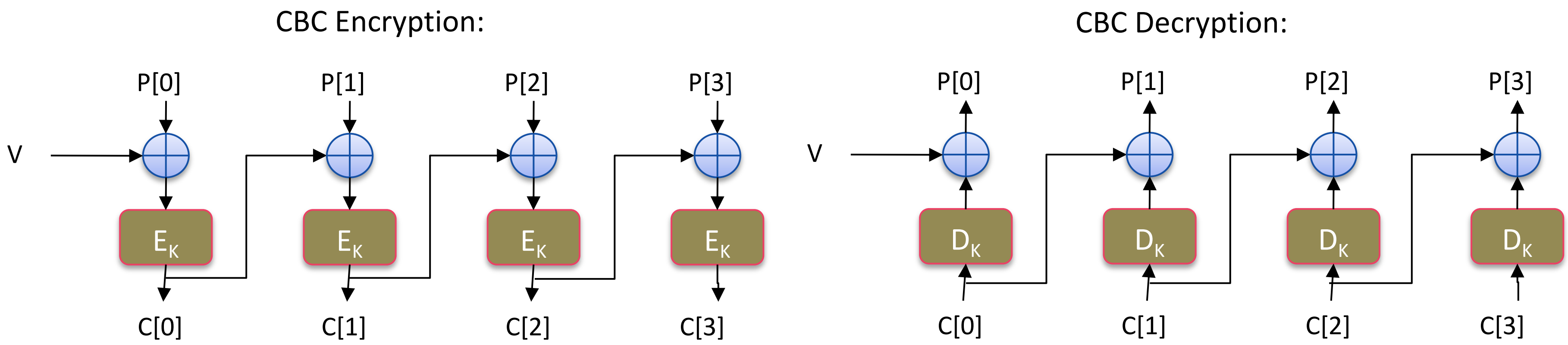
(a) (b)

Figure 8.6: How ECB mode can leave identifiable patterns in a sequence of blocks: (a) An image of Tux the penguin, the Linux mascot. (b) An encryption of the Tux image using ECB mode. (The image in (a) is by Larry Ewing, lewing@isc.tamu.edu, using The Gimp; the image in (b) is by Dr. Juzam. Both are used with permission via attribution.)

Cipher Block Chaining (CBC) Mode

In Cipher Block Chaining (CBC) Mode

- The previous ciphertext block is combined with the current plaintext block $C[i] = E_K(C[i-1] \oplus P[i])$
- $C[-1] = V$, a random block separately transmitted encrypted (known as the initialization vector)
- Decryption: $P[i] = C[i-1] \oplus D_K(C[i])$



Strengths and Weaknesses of CBC

Strengths:

- Doesn’t show patterns in the plaintext
- Is the most common mode
- Is fast and relatively simple

Weaknesses:

- CBC requires the reliable transmission of all the blocks sequentially
- CBC is not suitable for applications that allow packet losses (e.g., music and video streaming)

Java AES Encryption Example

Source

http://docs.oracle.com/javase/8/docs/api/javax/crypto/package-summary.html

Generate an AES key

```
KeyGenerator keygen = KeyGenerator.getInstance("AES");
SecretKey aesKey = keygen.generateKey();
```

Create a cipher object for AES in ECB mode and PKCS5 padding

```
Cipher aesCipher;
aesCipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
```

Encrypt

```
aesCipher.init(Cipher.ENCRYPT_MODE, aesKey);
byte[] plaintext = "My secret message".getBytes();
byte[] ciphertext = aesCipher.doFinal(plaintext);
```

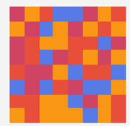
Decrypt

```
aesCipher.init(Cipher.DECRYPT_MODE, aesKey);
byte[] plaintext1 = aesCipher.doFinal(ciphertext);
```

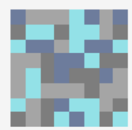
Mathematica AES Example

```
In[1]:= msg = "This is a secret that cannot be revealed!";

In[2]:= keyAES = GenerateSymmetricKey[
           Method → <|"Cipher" → "AES256", "BlockMode" → "CBC"|>]
```

Out[2]= SymmetricKey[ cipher: AES256 block mode: CBC key length: 256 bits]

```
In[3]:= cipherAES = Encrypt[keyAES, msg]
```

Out[3]= EncryptedObject[ data length: 48 bytes IV length: 128 bits original form: String]

```
In[4]:= Decrypt[keyAES, cipherAES]

Out[4]= This is a secret that cannot be revealed!
```

Mathematica AES Example

The information can be extracted by

In[5]:=
```
Normal[keyAES]
```

Out[5]=
```
SymmetricKey[<|Cipher -> AES256, BlockMode -> CBC,
  Key -> ByteArray[ 32 bytes ], InitializationVector -> None|>]
```

In[6]:=
```
keyBytesAES = Normal[keyAES["Key"]]
```

Out[6]=
```
{232, 52, 238, 192, 18, 48, 133, 184, 61, 168,
 24, 182, 179, 182, 92, 29, 56, 52, 100, 192, 168,
 241, 3, 142, 35, 129, 185, 162, 31, 38, 100, 139}
```

The cipher can be transformed into bytes by

In[7]:=
```
cipherBytesAES = Normal[cipherAES["Data"]]
```

Out[7]=
```
{213, 71, 155, 182, 109, 226, 117, 251, 120, 170, 154, 68,
 185, 221, 185, 69, 230, 59, 194, 44, 222, 88, 73, 201,
 163, 32, 53, 146, 3, 135, 94, 114, 217, 147, 225, 85,
 130, 192, 52, 100, 95, 150, 129, 91, 239, 38, 116, 242}
```

Summary Symmetric Cryptography II

- Block ciphers
- Hill cipher
- Transposition cipher
- AES
- Block Cipher Modes

That's all folks!