Symmetric Key Encryption

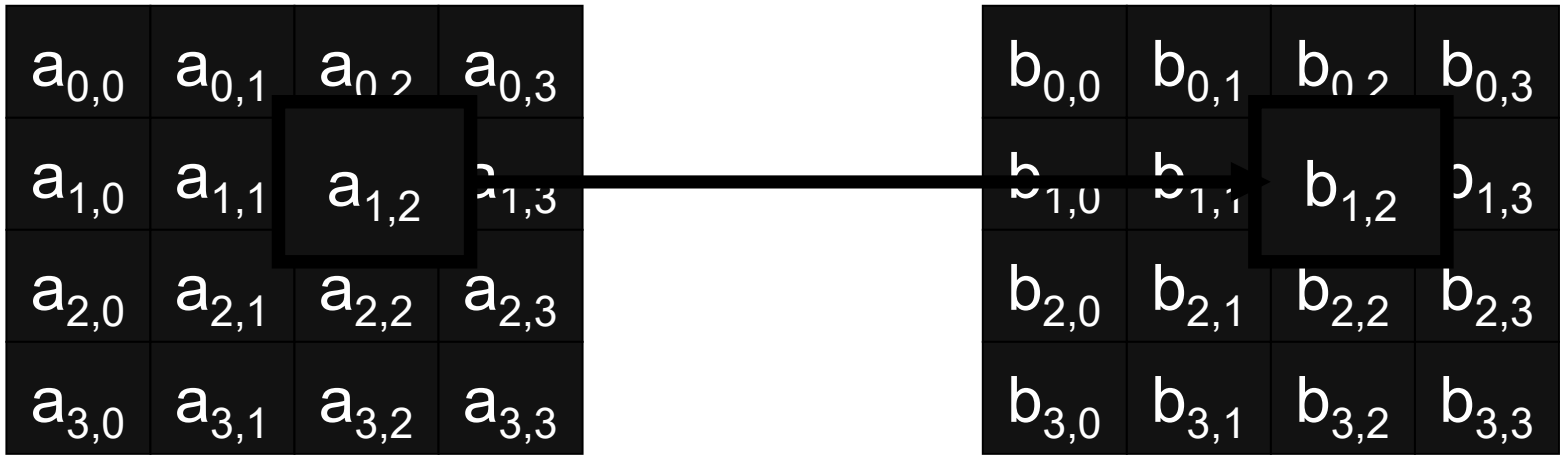
Eike Ritter and David Oswald

Computer Security and Networks:
Lecture 2

Last Lecture

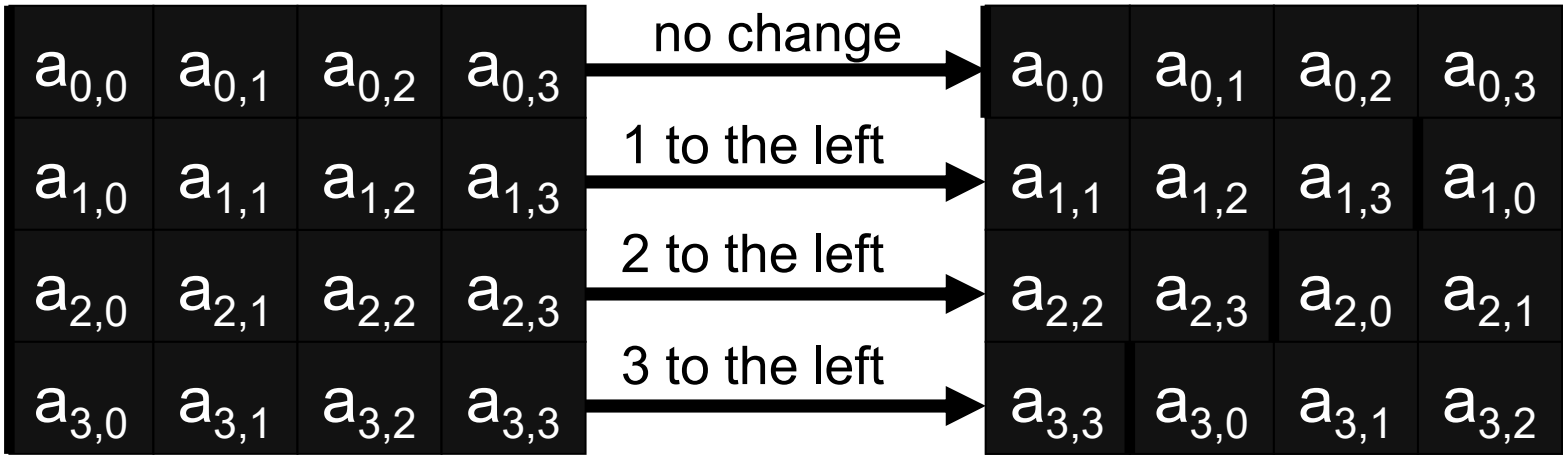
- How data is encode
  - bytes, binary, hex, ascii..
- One Time Pads
  - and how to xor data.
- AES
  - block cipher
  - Encrypting with AES in Java

SubBytes: S-box



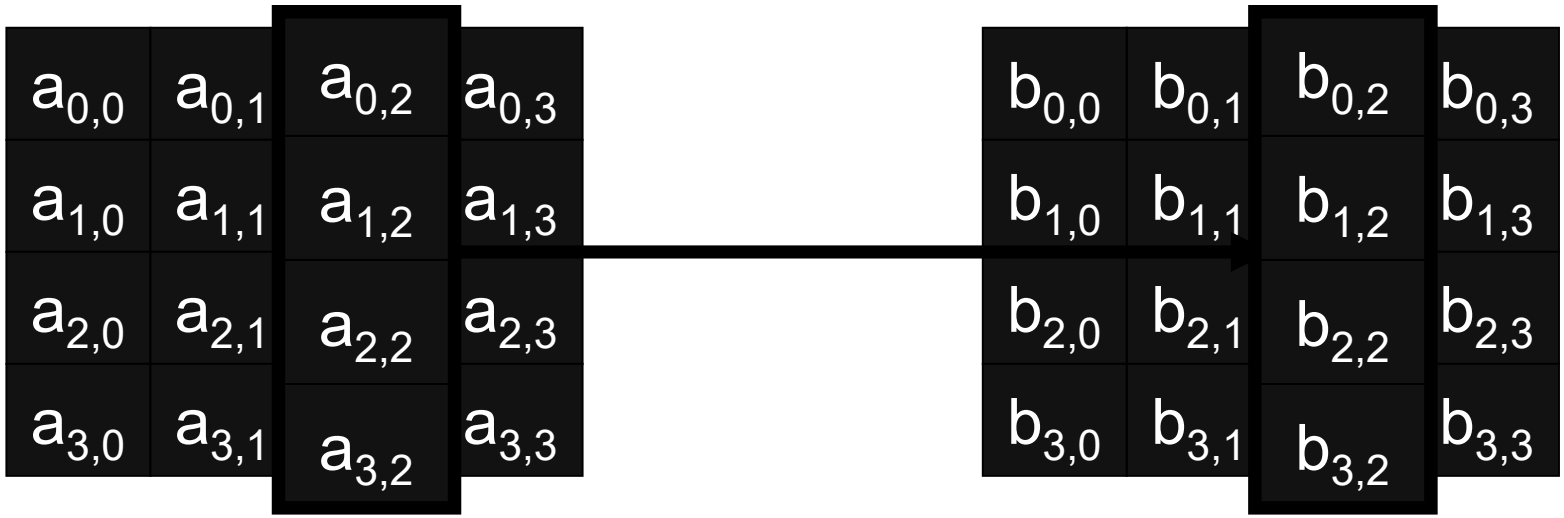
- The "SubByte" is a fixed substitution based on matrix multiplication, one byte at a type.

ShiftRows



- "ShiftRows" moves the
  - 2nd row one byte to the left,
  - the 3rd row two bytes
  - and the 4th row 3 bytes.

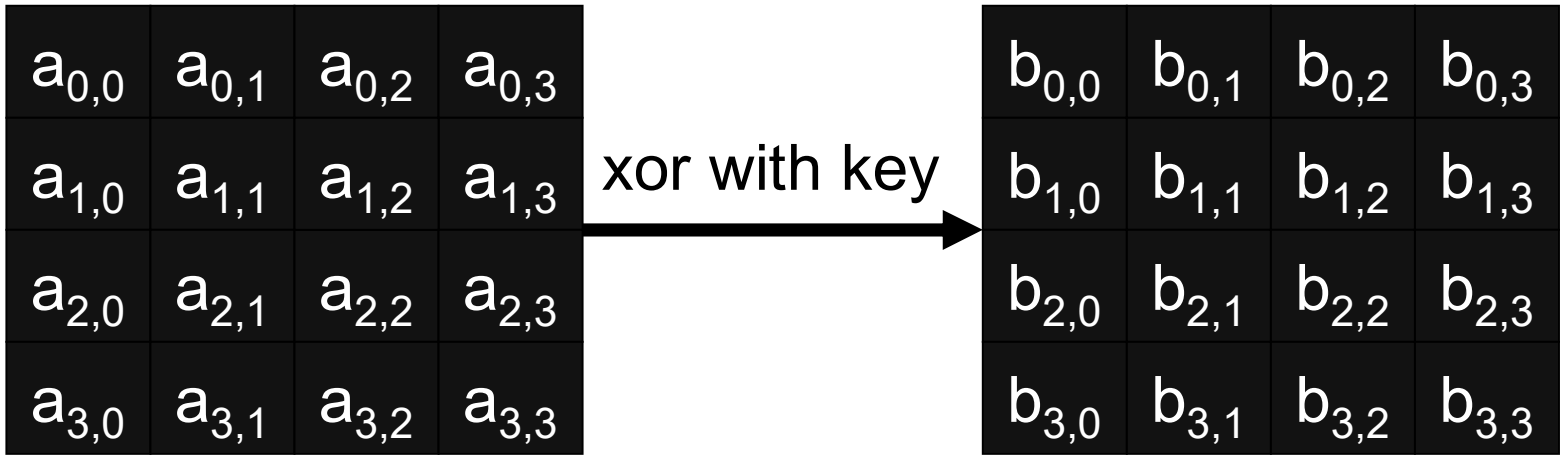
MixColumn



- "MixColumn" is a substitution of each column such that:

$$(a_0.x^3 + a_1.x^2 + a_2.x + a_3) \times (3.x^3 + x^2 + x + 2) \bmod (x^4+1) = (b_0.x^3 + b_1.x^2 + b_2.x + b_3)$$

AddRoundKey



- "AddRoundKey" xor's the block with the 128-bit round key (which was generated from the main key).

  – $b_{i,j} = a_{i,j}$ xor $k_{i,j}$

Symmetric Key Encryption

- DES and 3-DES
- Padding
  - What if it's less than 1 block?
- Block cipher modes
  - What if it's more than 1 block?

Advanced Encryption Standard (AES)

- AES is a state-of-the-art block cipher.
- It works on blocks of 128-bits.
- It generates 10 round keys from a single 128-bit key.
- In uses one permutation: ShiftRows and three substitutions SubBytes, MixColumns, AddRoundKey.

DES

The Data Encryption Standard (DES), was the previous standard.

Designed by IBM in early 1970’s

DES

The Data Encryption Standard (DES), was the previous standard.

Designed by IBM in early 1970's

Before it was accepted as a standard the NSA stepped in and added S-boxes and fixed the key length at 56 bits

DES

- S-boxes are a type of substitution.
- It was unclear at the time why the NSA added S-boxes to the design.
- Many believed these were a back door for the NSA.

DES

- In 1990, Biham & Shamir discovered differential cryptanalysis.
- The S-boxes had made DES resistant to differential cryptanalysis.
- It seems that the NSA knew about differential cryptanalysis, at the start of the 1970s and had step into to protect DES.

Cost to Break DES

- 1977, Diffie and Hellman, theoretically: $20 million, break in 1 day.

Cost to Break DES

- 1977, Diffie and Hellman, theoretically: $20 million, break in 1 day.
- 1993, theoretically $1 million, in 7 hours.

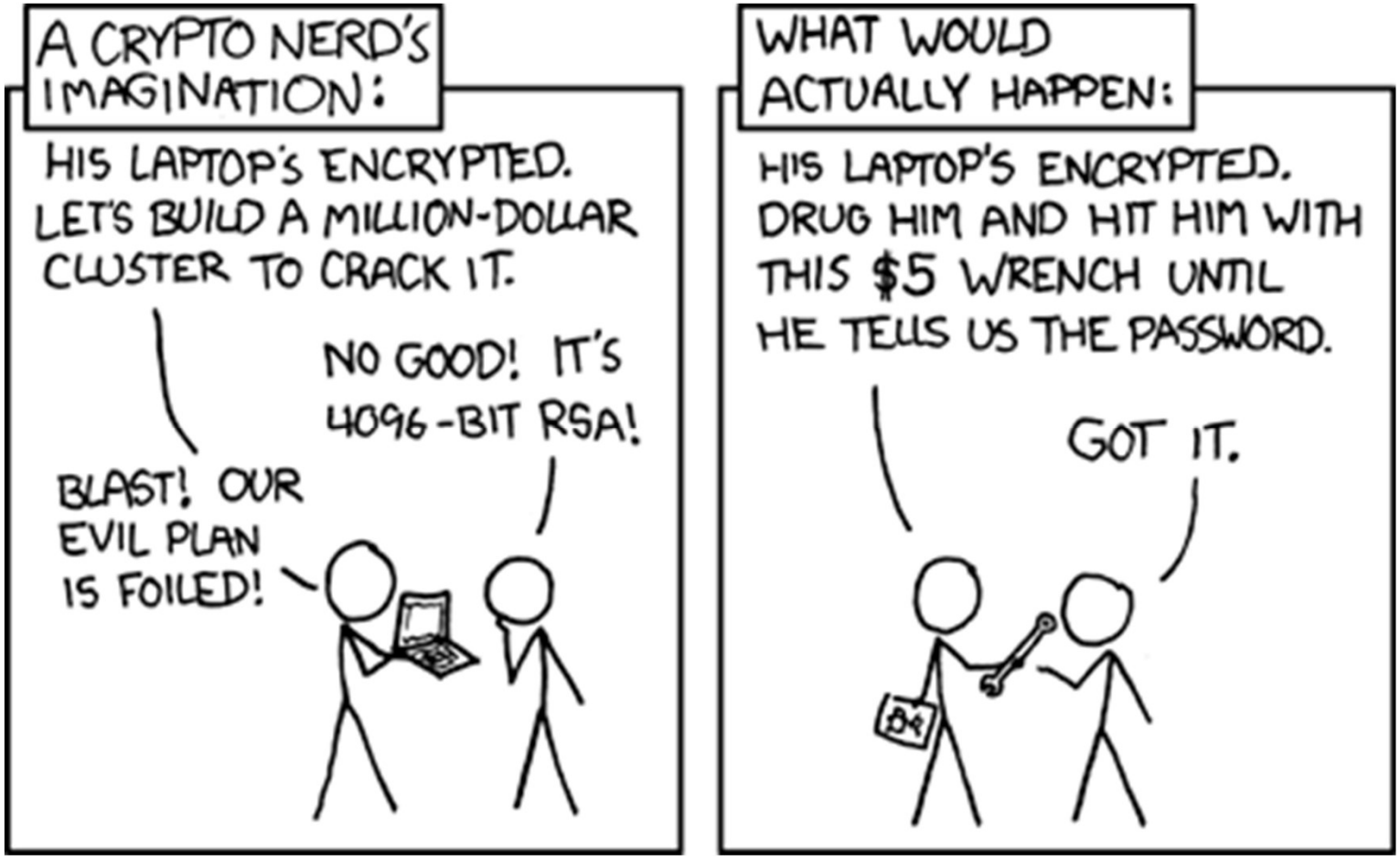
Cost to Break DES

- 1977, Diffie and Hellman, theoretically: $20 million, break in 1 day.
- 1993, theoretically $1 million, in 7 hours.
- 1997, RSA Security offer $10,000 for a real break,
  - won by a distributed computing project, at “no cost”
  - EFF (Electronic rights group) break in 56 hours for $250,000

Cost to Break DES

- 1977, Diffie and Hellman, theoretically: $20 million, break in 1 day.
- 1993, theoretically $1 million, in 7 hours.
- 1997, RSA Security offer $10,000 for a real break,
  - won by a distributed computing project, at “no cost”
  - EFF (Electronic rights group) break in 56 hours for $250,000
- 2006, COPACOBANA, general purpose brute force, break DES for $10,000

A word about key length



Source:

3-DES

- Triple DES, was a stop gap until AES
- 3-DES takes 3 keys, K_1, K_2 & K_3.

$$E_{K1K2K3}(M) = E_{K3}(D_{K2}(E_{K1}(M)))$$

- Setting $K_1=K_2=K_3$ gives you DES
- Expected to be good until 2030
- Used in bank cards and RFID chips

Padding

- Block ciphers only work on fixed size blocks.
- If the message isn’t of the right block size we need to pad the message.
- But receiver need to tell the difference between the padding and message.

Padding

- Add random bytes to the end of the block?

—

—

Padding

- ~~Add random bytes to the end of the block?~~

—

—

Padding

- ~~Add random bytes to the end of the block?~~

—

- Add zeros to the end of the block?

—

Padding

- ~~Add random bytes to the end of the block?~~

—

- ~~Add zeros to the end of the block?~~

—

Padding

- ~~Add random bytes to the end of the block?~~

—

- ~~Add zeros to the end of the block?~~

—

- Write “this is padding”?

Padding

- ~~Add random bytes to the end of the block?~~

—

- ~~Add zeros to the end of the block?~~

—

- ~~Write “this is padding”?~~

Padding: PKCS 5/7

- If there is 1 byte of space write 01
- If there are 2 byte of space write 0202
- If there are 3 byte of space write 030303
- …
- If the message goes to the end of the block add a new block of 16161616..

- PKCS 7: 16 byte block, PKCS 5: 8 byte block

Block Cipher Modes

- Block Ciphers can be used in a number of modes:

- **1) Electronic codebook mode** (**ECB)**
  - each block is encrypted individually,
  - encrypted blocks are assembled in the same order as the plain text blocks.
  - if blocks are repeated in the plain text, this is revealed by the cipher text.

Demo Block Problems

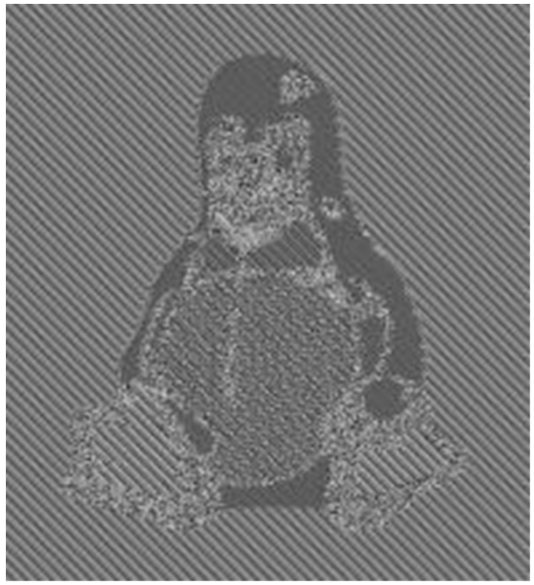
Block Cipher Modes

Original

Images from wikipedia

Block Cipher Modes

Original



ECB

Block Cipher Modes

2) Cipher Block Chaining mode (CBC)

- each block XOR'd with previous block
- start with a random Initialization Vector (IV)
- helps overcome replay attack.

- Suppose the plain text is $B_1, B_2, \ldots, B_n$.

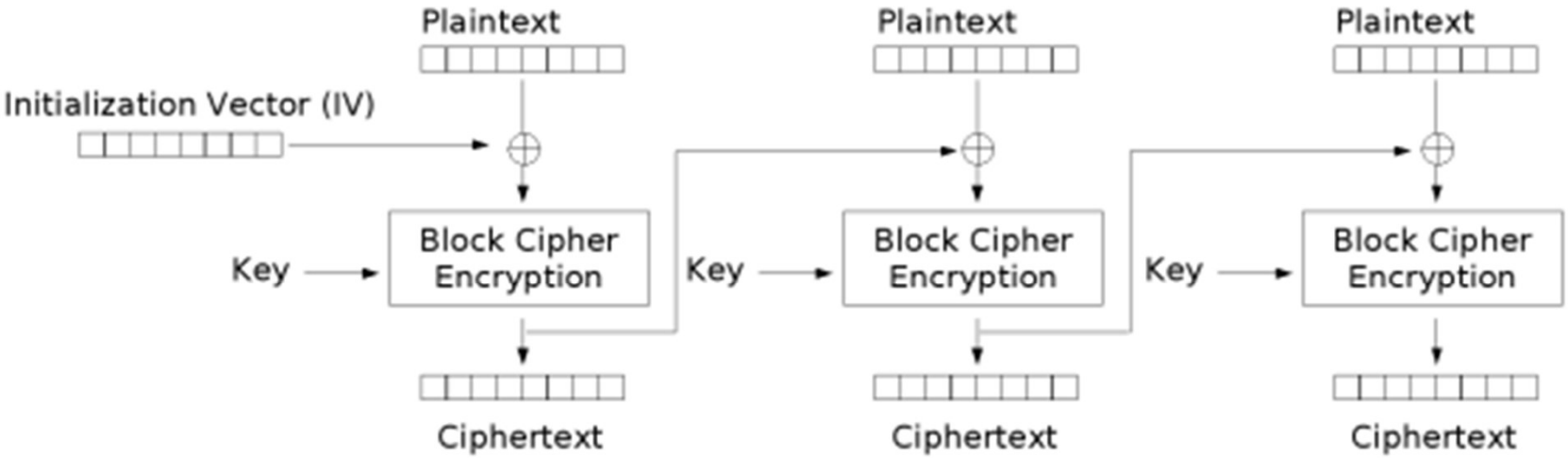
  IV = random number (sent in the clear)

  C_1 = encrypt(B_1 xor IV),

  C_2 = encrypt(B_2 xor C_1).

  ….. C_i = encrypt(B_i xor C_{i-1}).

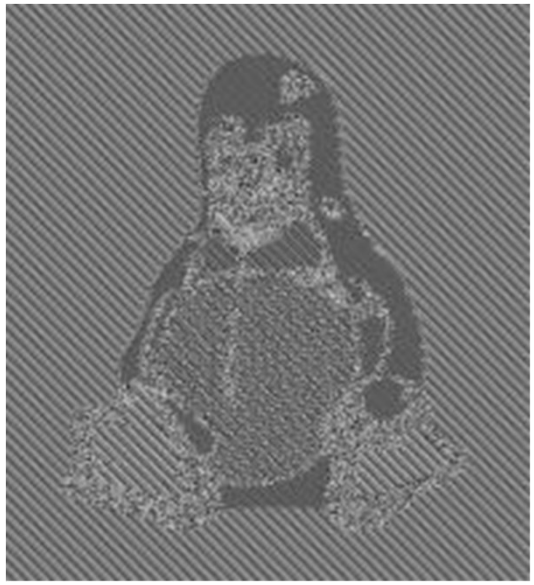
Block Cipher Modes



Cipher Block Chaining (CBC) mode encryption

Block Cipher Modes

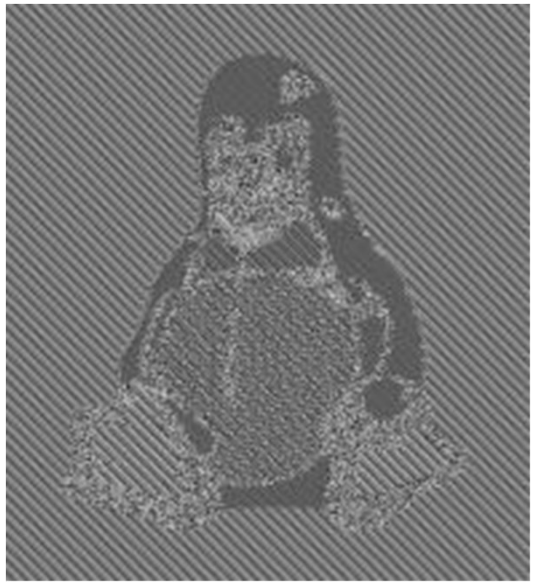
Original



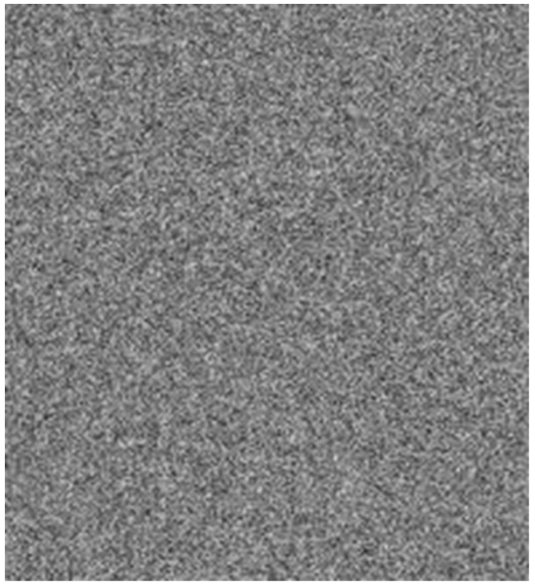
ECB

Block Cipher Modes

Original

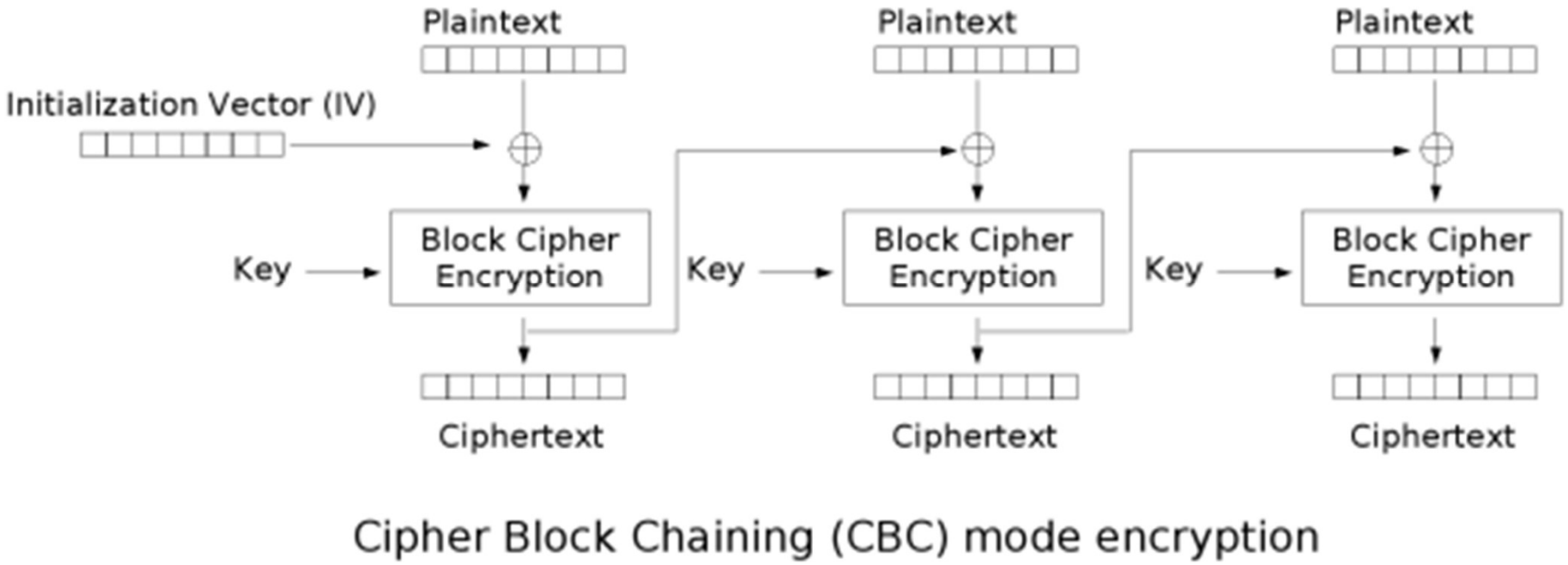


ECB



CBC

CBC encrypt



Cipher Block Chaining (CBC) mode encryption

http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

CBC decrypt

- Receive IV
- Receive cipher text = $C_1, C_2, C_3, \ldots$
- Plain text = $B_1, B_2, B_3, \ldots$ where:

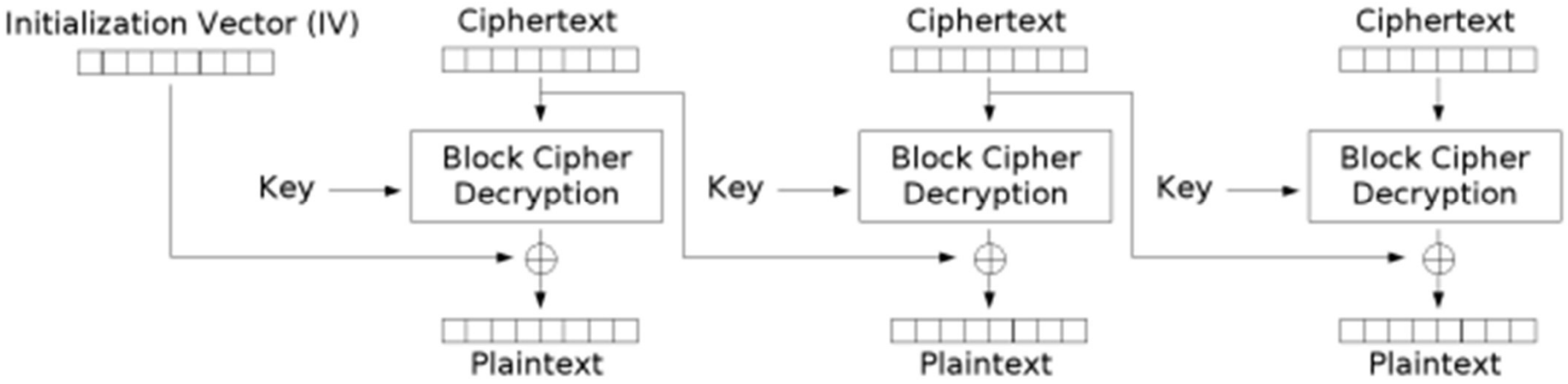
$$B_1 = \text{decrypt}_K(C_1) \text{ xor } IV,$$

$$B_2 = \text{decrypt}_K(C_2) \text{ xor } C_1,$$

$$\ldots\ldots$$

$$B_i = \text{decrypt}_K(B_i) \text{ xor } C_{i-1}.$$

CBC decrypt



Cipher Block Chaining (CBC) mode decryption

http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

Probabilistic Encryption

- Probabilistic encryption schemes use random elements to make every encryption different.
- CBC with a random IV is a good way to make encryption probabilistic.
- Using CBC & random IVs lets me encrypt the same message, and with the same key, without an attacker realising.

Sony PlayStation

Sony needs to stop
games being copied.

CD & full disk encryption

User can read and write
areas of the hard disk,
for own files, notes, etc

Why won't CBC work?

Sony PlayStation

- With CBC, you need to encrypt, or decrypt, the whole file to get to the end.
- The Sony PlayStation uses ECB full disk encryption, to stop people copying games.
- User can access files they made themselves
- Hardware controls user access to data.

Sony PlayStation Disk Encryption Attack

1. Remove disk and make a copy.
2. Replace disk in Playstation.

Sony PlayStation Disk Encryption Attack

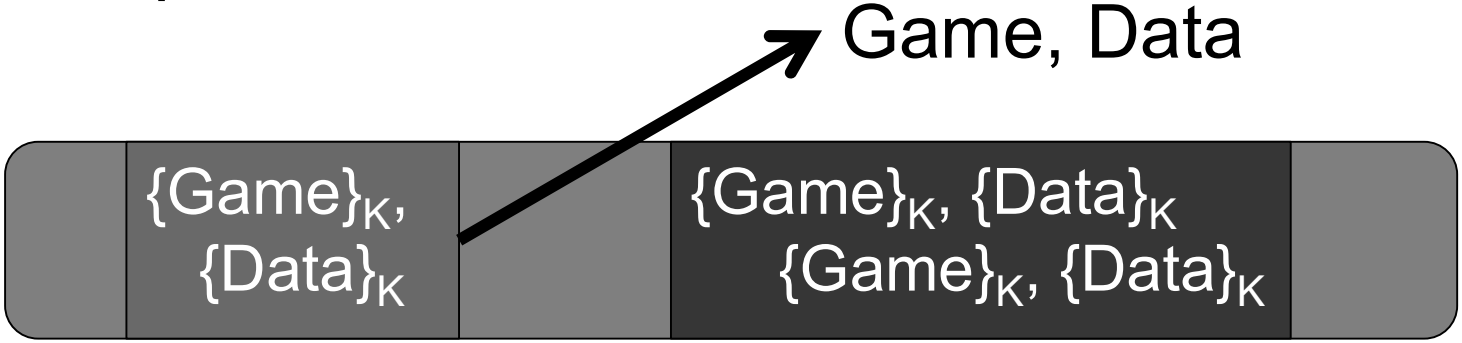
1. Remove disk and make a copy.
2. Replace disk in Playstation.
3. Copy a file to the disk.
4. Remove disk and find the bit of disk that changed (that's your encrypted file).

Sony PlayStation Disk Encryption Attack

5. Copy target data to the user area.

Sony PlayStation Disk Encryption Attack

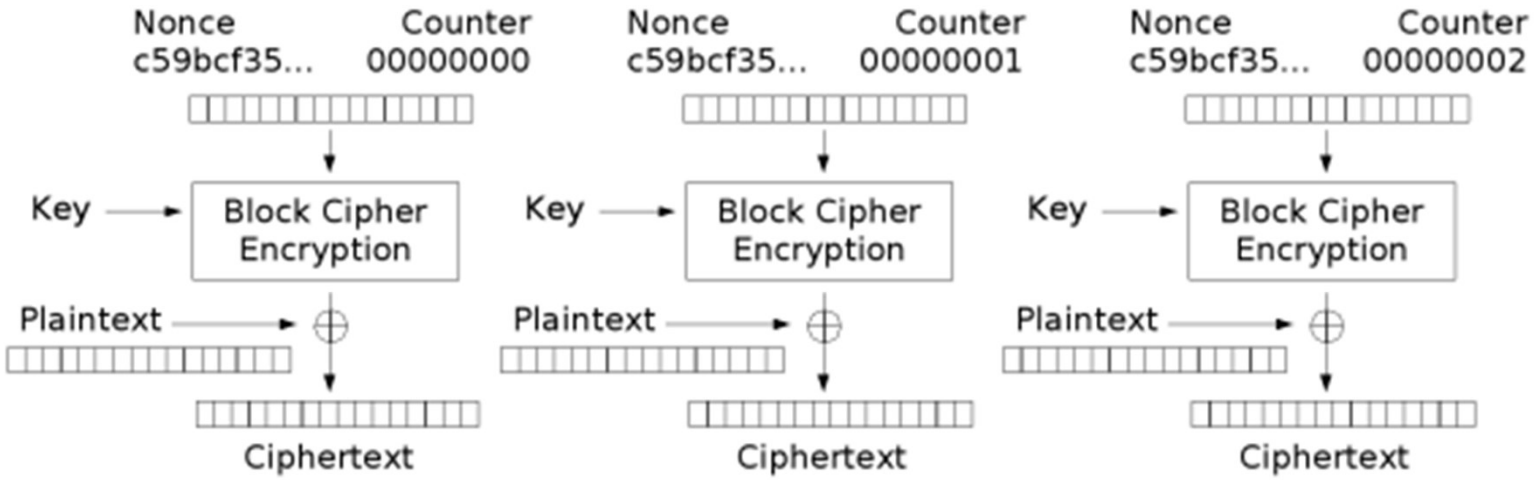
5. Copy target data to the user area.
6. Restart the PlayStation and ask for your file back.
7. PlayStation decrypts the file and gives you the plain text.



Counter Mode (CTR)

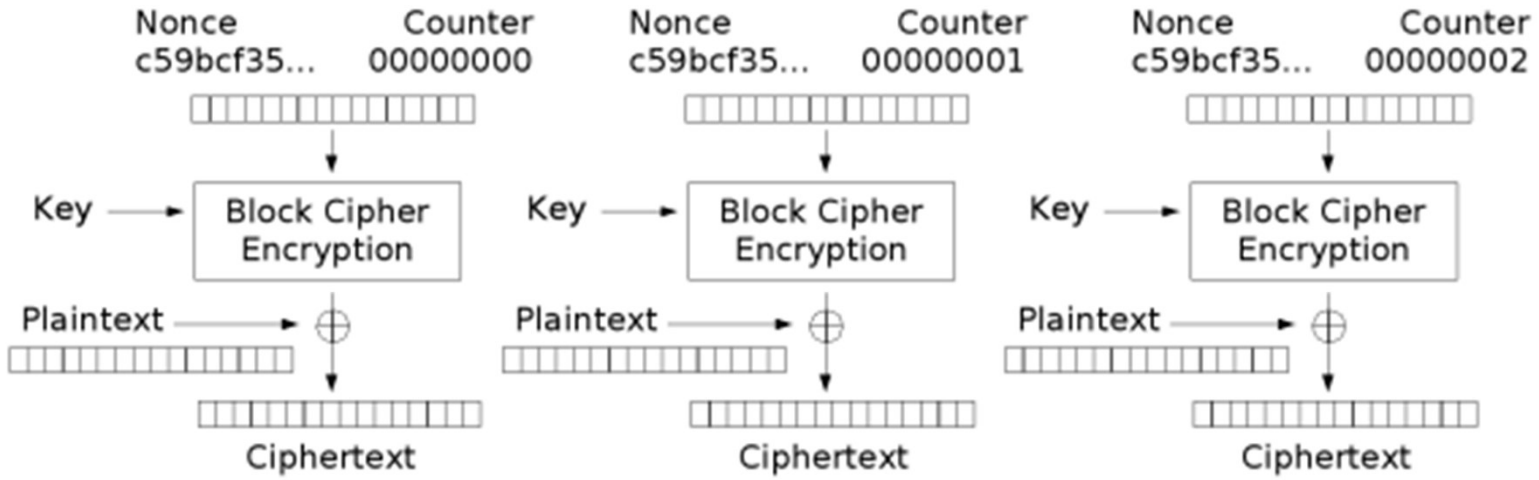
- Plain text = $B_1, B_2, B_3, \ldots$
- IV = random number (sent in clear)
- Cipher text = $C_1, C_2, C_3, \ldots$ where

$$C_1 = B_1 \text{ xor } \text{encrypt}_K(IV),$$
$$C_2 = B_2 \text{ xor } \text{encrypt}_K(IV+1),$$
$$C_3 = B_3 \text{ xor } \text{encrypt}_K(IV+2),$$
$$\ldots\ldots$$
$$C_i = B_i \text{ xor } \text{encrypt}_K(IV+ i-1),$$

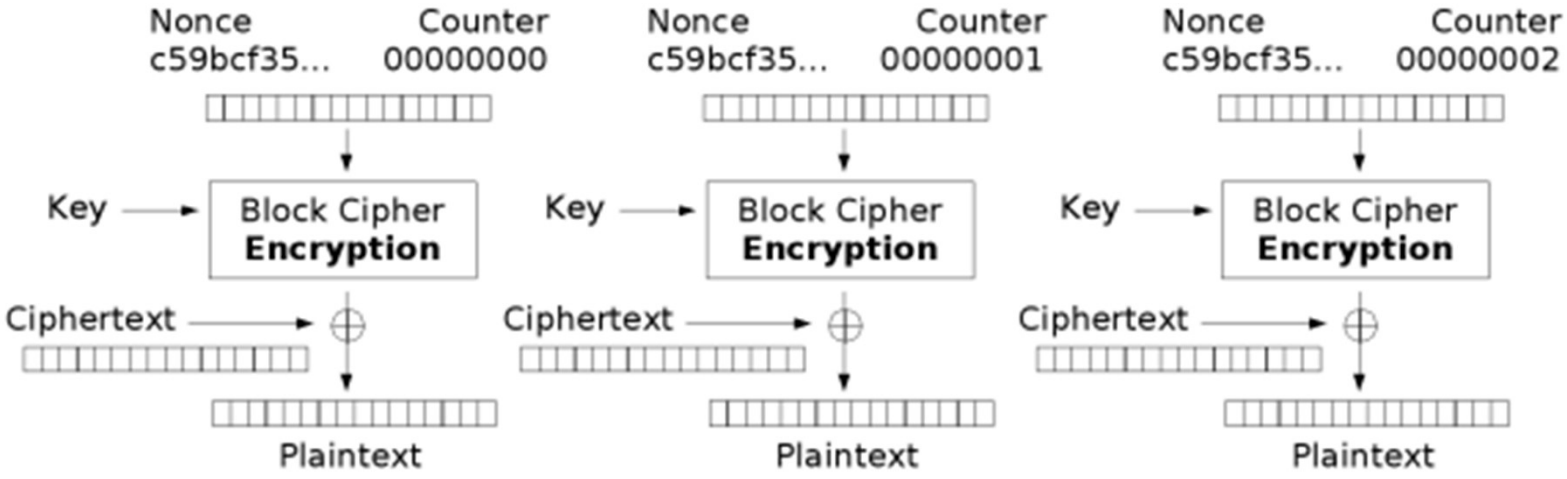


Counter (CTR) mode encryption

http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation



Counter (CTR) mode encryption



Counter (CTR) mode decryption

http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

Next Lecture

- Public Key Crypto