SIS

LAB 2 – Encryption and integrity protection

Introduction

Confidentiality is one of the most frequently used methods in contemporary security systems. Confidentiality is achieved in the first place by means of cryptographic algorithms for encryption. Symmetric key cryptosystems require that a secret key is shared by the encrypting and the decrypting devices. Asymmetric key cryptosystems handle key differently - a pair of related keys is required to perform encryption and decryption. One of the keys has to remain private, while the other one can be made public, thus relaxing the requirement of secure key delivery.

Symmetric ciphers use the following scheme for encryption and decryption, respectively:

$$E_K(M) = C \quad \text{and} \quad D_K(C)=M$$

Where **E** represents the encryption function, **D** represents decryption function, **M** is a plaintext message, **C** is the ciphertext message corresponding to **M**, and **K** is the cryptographic key.

When the plaintext message is longer than the input block of a cipher, there’s a need to use one of the encryption modes. The following examples show some frequently used modes (see lecture slides for explanation). DES algorithm serves here as an example block algorithm of short input block of fixed length.

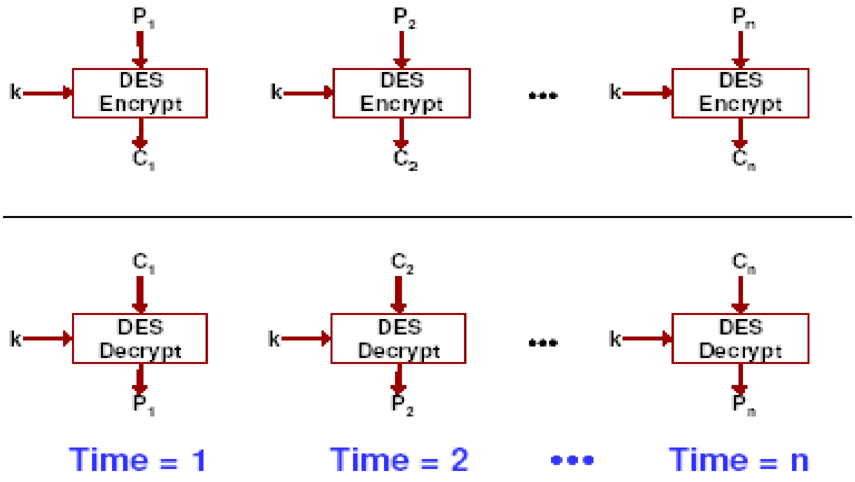

Figure 1. Electronic Codebook (ECB)

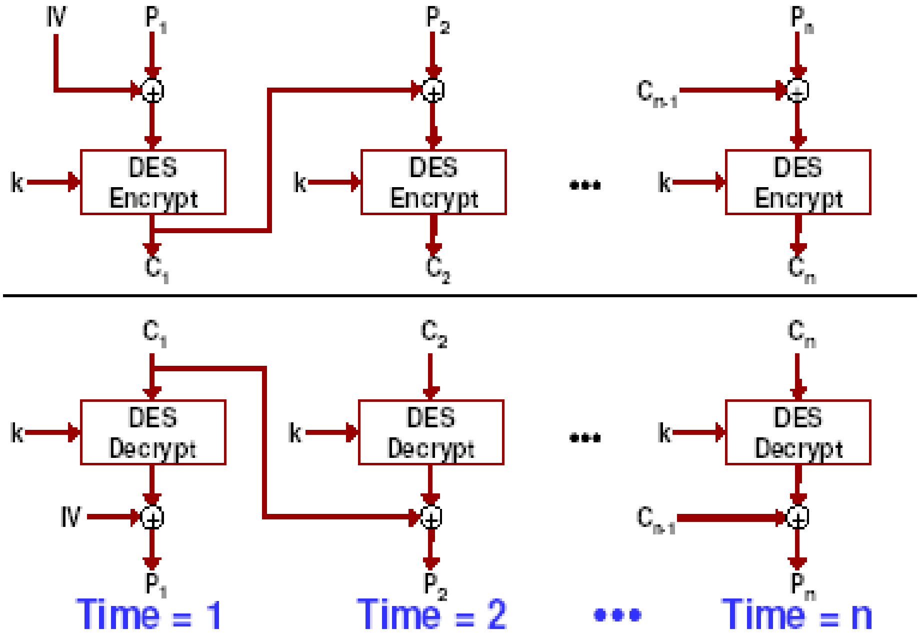

Figure 2. *Cipher Block Chaining* mode (CBC)

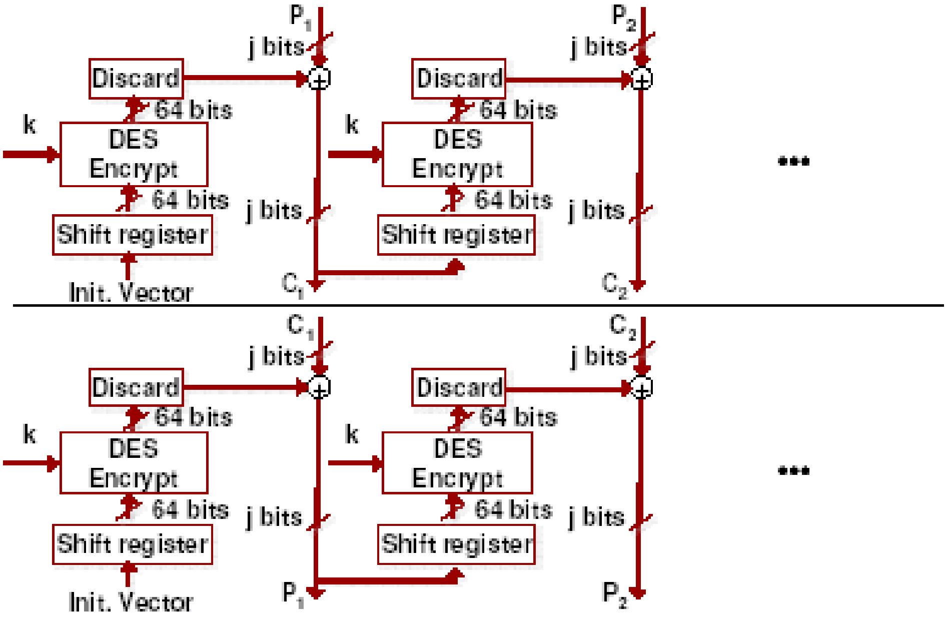

Figure 3. *Cipher Feedback* (CFB)

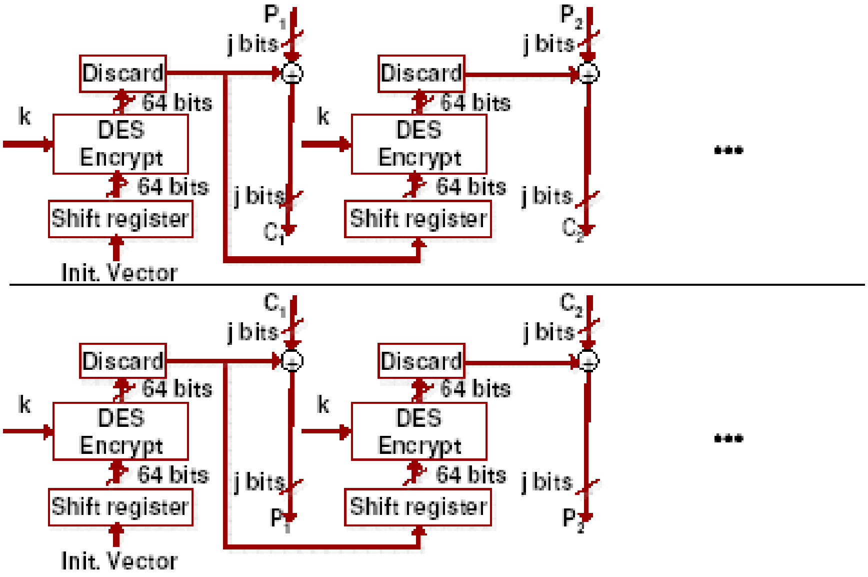

Figure 4. ***Output Feedback (OFB)***

Integrity protection is another method frequently used in security. It allows to verify that data has not been modified (tampered with) or removed by an unauthorized individual. It is often implemented by means of a control checksum, computed from the original message and appended to that original message. The control checksum is usually computed from the original message with a one way hash function.

Figure 5 shows a CBC-MAC mode, which allows to use block ciphers (such as for example DES) to compute a hash value.

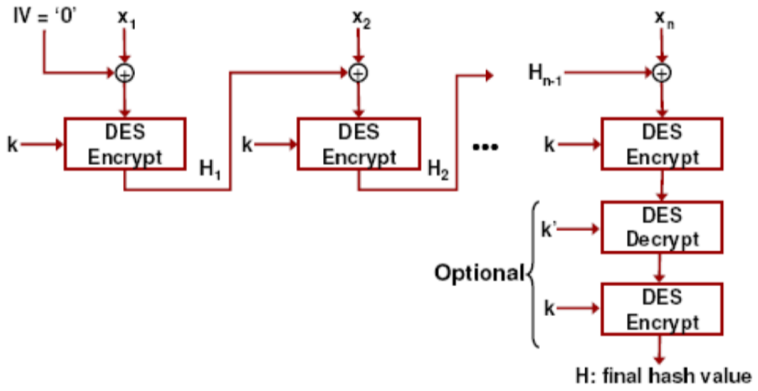

Figure 5. ***CBC-MAC keyed hash funtion***

Laboratory manual

Initial steps

1. Please boot the computer in Windows 7
2. Create a folder on the hard disk and download **`AESCipher`** application (follow link to Lab 1. on http://kt.agh.edu.pl/~niemiec/lab). The application is a software implementation of a AES symmetric block cipher. Brief information about the used block size, key length and the application itself can be found in Help (F1). Get familiar with the user interface, learn the functionality, look at the settings and then encrypt and decrypt arbitrary text as well as a test file from a disk.

   Note: when you want to see the results of the encryption process while using graphical interface, please bear in mind that the ciphertext is presented in ASCII format. Therefore some characters are blank (unprintable). An attempt to decrypt the text copied from one window to another as you see it will fail. Be patient and try to encrypt your plaintext with different encryption keys (use "Generate" button to change the encryption key). After several trials you will see, that the resulting ciphertext is free of any imprintable characters. The longer the ciphertext from your plaintext is, the higher is the probability that the ciphertext is free of any imprintable characters.
3. Download and install hexadecimal text editor `hexfre20` from http://kt.agh.edu.pl/~niemiec/lab .

   Note: in case of installation problems download **`xvi32`** from http://www.chmaas.handshake.de/delphi/freeware/xvi32/xvi32.htm
4. The **`AESCipher`** application supports several modes for encrypting large input texts: OFB, CFB, EBC, CBC and a one-way, keyed hash function which is essentially a CBC-MAC mode.
5. The names of the modes have been hidden. This is intentional. Please identify each of the modes.

   Hint: use the properties, that characterize each mode of operations, such as error propagation etc.

   Q1: When using one encryption algorithm and one secret key, but different modes of encryption, do we obtain the same ciphertext each time ? Explain your answer ?
6. Open file `polecenia_sluzbowe.txt`, which you can find on Web page http://kt.agh.edu.pl/~niemiec/lab and copy the content of the file ans save on a disk in `.txt` format (use Notepad). Assume that the file contains a confidential note with two requests regarding bank account operations. Encrypt the text file with ECB , and then use HexEdit to modify ciphertext so that, when decrypted, bank accounts would appear swapped.

   Q2: What messages can be encrypted with ECB mode ? Which mode could be used to encrypt that message ?

7. Use keyed hash CBC-MAC mode to see, how integrity protection works for long messages. Copy any text of considerable length from some web page and paste it into a `.txt` file or directly to a window in a graphical user interface. Compute the hash digest and save it on a disk (the hash is displayed in hex format). Next replace any single letter or digit in the text for an arbitrary different one and compute hash digets again.

   Q3: The whole hash digest has changed ? Explain why ?

8. Now correct the modified letter or digit back to the original one and compute hash again.

    Q4: Is the digest the same as originally ? Why ?

Go to web page: http://www.hashemaill.com/ where you can find many different hash functions. Try the most popular (MD5, SHA-1 or SHA-2, RIPEMD).

When finished, remove the directory along with the files that you donwloaded or created.

Report

- Please document the steps and the results. Please provide interpretation and argumentation.
- Answer to the questions listed above. Name the discovered encryption modes and explain what information you have used to identified the right mode.
- Write down a few statements that recapitulate major findings, general conclusion and lessons learnt.