IK2206 – Internet Security and Privacy
Chapter 3 – Secret Key Cryptography

3.1 Introduction (READ)

3.2 Generic Block Encryption (READ)

- How large blocks of data are suitable to work within a block encryption scheme?

  64 bits is a reasonable length

  - What are the drawbacks of choosing a "too small" block size? (compare with monoalphabetic ciphers)

    For example one octet (eight bits) -> not secure, could construct a table to be used for decryption

  - What are the drawbacks of choosing a "too large" block size?

    Unnecessarily complex and possibly having performance problem

- What is a permutation?

  A permutation changes the positions of each character of the input value. It specifies, for each of the k input bits, the output position to which it goes. -> Bit shuffle

- What is a substitution? Why is it impractical to let a "substitution box" work on the whole block of data?

  Specifies for each of the 2^k possible values of the input, the k bit output.

  ?

- What is a round?

  Do a substitution on each small chunk, and then take the outputs of all the substitutions and run them through a permuter that is as big as the input, which shuffles the bits around.

- What properties should the secret key have on the output? How much influence should every bit of the key have on the output?

- Are you able to describe an example design of a block encryption scheme such as the one shown in figure 3-1?

An input is divided in smaller pieces. On each piece a substitution is executed. In the end we combine the pieces and we permute the bits.

- What does it mean that a mechanism is reversible? (In the lecture slides and in a later section (figure 3-6) you can read about "Feistel designs".

3.3 Data Encryption Standard (DES) (READ)

- Can you name two properties of DES which makes it unsuitable for use in current cryptographic systems?

3.3.1 DES Overview (READ)

- Can you comment the structure of DES given a figure like 3-2?
  - *The 64-bit input is permuted to obtain a 64-bit result*
  - *The 56-bit key is used to generate sixteen 48-bit per round keys, by taking a different 48-bit subset of the 56 bits for each of the keys.*
  - *Then each round takes as input the output of the previous round and the 48-bit key and produces a 64-bit output.*
  - *After the 16th round, the output has its halves swapped*
  - *In the end the output is permuted again (inverse of the initial permutation)*

    ->Decryption works by essentially running DES backwards.
    We have to use the same key generation but in the opposite order (first use Key 16)

3.3.2 The Permutations of the Data (SKIP)

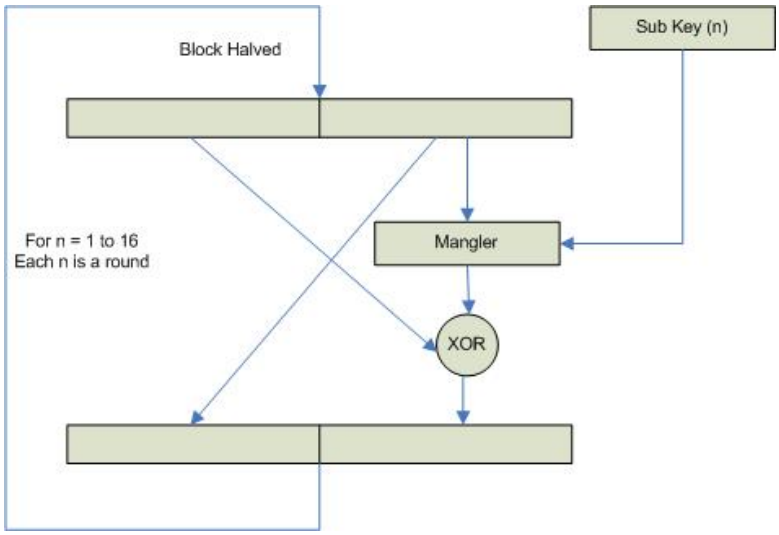
3.3.3 Generating the Per-Round Keys (SKIP)

3.3.4 A DES Round (READ)

- Can you explain the "Feistel design" in figure 3-6? How can the same design be used for encryption and decryption without running the mangler function backwards?
  - *64-bit input is divided into two 32-bits halves called Ln and Rn*
  - *The round generates as output 32-bit quantities Ln+1 and Rn+1*
  - *The 64-bit output of the round is the concatenation of Ln+1 and Rn+1*
  - *Ln+1 is Rn*
  - *Rn+1 is obtained as follows: Rn and Kn are input to mangler function, which output is 32-bit. The output is XOR'd with Ln -> Rn+1*
  - *The mangler function takes as input 32 bits of the data plus 48 bits of the key*

    DES decryption?
    - *Rn is just Ln+1*
    - *Then we know Rn, Ln+1, Rn+1 and Kn*
    - *In addition we know that Rn+1 = Ln XOR magler(Rn,Kn)*
    - *Therefore we compute mangler (Rn, Kn) and XOR the result of the magler with Rn+1 -> result will be Ln*

    ➔ *Mangler never run backwards*



3.3.5 The Mangler Function (READ BRIEFLY)

- Only read as much so that you get a feeling where the secret key is used in the DES structure.

3.3.6 Weak and Semi-Weak Keys (SKIP)

3.3.7 What's So Special About DES (READ)

- Why is it desirable to have an open design process of security protocols?

  Otherwise we don't know if the particular details were well-chosen for strength, or even whether the particular details were well-chosen to have some sort of weakness that could only be exploited by someone involved in the design process

- Security protocols commonly contain "random" parameters. How do you think these values should be selected?

  For instance by being the digits of an irrational number such as $\sqrt{2}$

3.4 Internationial Data Encryption Algorithm (IDEA) (SKIP)

3.5 Advanced Encryption Standard (AES) (READ)

- Describe some of the cicumstances behind NIST's initiative for a new encryption standard?

  Need a new secret key standard because DES' s key was to small, Triple DES too slow. Some branches of the U.S. government trying everything they could to hinder deployment of secure cryptography.

- Why do you think NIST chose to arrange an open contest for the new standard?
  - *That the design process is public and therefore they don't face the same problem as with DES*
  - *Should be free to implement*
  - *If many people can have a look on the design then the chance is better to identify weaknesses of a new algorithm*

3.5.1 Basic Structure (READ)

- What block size and key size(s) does AES support?

  Block size: 128, 160, 192, 224 and 256 bits
  Key size: is the number of 32-bit words in an encryption key.
  AES-128 -> Nk = 4, AES-192 Nk=6, AES-256 Nk = 8 -> Rijndael allows any Nk between 4 and 8 inclusive

- How many rounds are used in AES? Why is it a variable amount?

  This parameter is a function of the other two parameters. The number of round needs to be larger for longer keys and as well larger for bigger block sizes.
  Rijndael specifies Nr = 6 + max(Nb (number of words in an encryption block), Nk (number of words in an encryption key)

3.5.2 Primitive Operations (SKIP)

3.5.3 Key Expansion (SKIP)

3.5.4 Rounds (SKIP)

3.5.5 Inverse Rounds (SKIP)

3.5.6 Optimization (SKIP)

3.6 RC4 (READ)

- What is a "stream cipher"?

  Generates a one-time pad and applies it to a stream of plaintext with XOR.

- What is a "one-time pad"?

  A long random string used to encrypt a message with a simple XOR operation

- How does a stream cipher differ from a block cipher? (When you read the next chapter (Modes of operation) you may find that the border line between block and stream ciphers is not as clear, since block ciphers can be run in "stream cipher mode".)

  ***Stream ciphers** combine plain-text bits with a pseudorandom cipher bits stream with the use of XOR (exclusive-or) operation. Stream ciphers encrypt plain-text digits one at a time with varying transformations for successive digits*

  ***Block ciphers** operate on blocks (groups of bits) with fixed-length. Block ciphers use a fixed (unvarying) transformation for all digits in the block. For example, when an x-bit block plain-text (along with a secret key) is provided as input to the block cipher engine, it produces the corresponding x-bit block of ciphertext.*