**Eastern Mediterranean University**

**Computer Engineering Department**

**CMSE-353 Security of Software Systems**

 **Midterm Exam**

**Three A4 sheets of handwritten paper may be used for your help. Photocopies, printouts, etc. are not allowed! Electronic devices are not allowed**

**Duration: 100 minutes November 13, 2019**

**Std Id\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Std Name\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Instructor Alexander Chefranov**

**Totally 4 questions, 7 pages**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** |  |  |  |  | **Total** |
| **Points** | **20** | **15** | **35** | **30** | **100** |
| **Grade** |  |  |  |  |  |

**Q1.** **(20 points).** Explain concerns of the following security requirements:

1. (7 points) Confidentiality

Data shall be available to authorized people only, e.g., students’ records shall be available to their academic advisor only.

1. (7 points) Integrity

Data shall meet integrity constraints, e.g., student id number in the students table shall not appear twice. Data shall not be altered in unauthorized way.

1. (6 points) Availability

Data shall be available to the users according to an agreement between them and service provider, e.g., 7 days, 24 hours a day

# Q2. (15 points). What social engineering is? (5 points) Explain Quid Pro Quo technique used in social engineering. (10 points)

Social engineering is a set of methods alloing a social engineer to get sensitive information from other people, e.g., bribery. Quid pro quo technique uses people’s psychology: if a social engineer does something useful to a potential victim, the victim wants returning something good to the social engineer, and, this may be, in particular, some his/her sensitive information, e.g., a password.

**Q3.** **(35 points).** Using Hill cipher with size 2 block,

1. (25 points) Encrypt the **first** block of the following message “Good morning, guys!” preserving blanks, commas, exclamation marks, ~~commas~~, and dots.

What numerical codes of the symbols you use? Construct an appropriate key matrix. (4 points)

Numerical codes are:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|  | , | ! | . |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 26 | 27 | 28 | 29 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

What conditions must be satisfied by the key matrix? (4 points)

What modulo value shall be used? Show that the matrix you construct satisfies the conditions. (4 points)

N=30

Calculate inverse of the key matrix. (13 points)

, since

Thus, . Check its correctness:

Encrypt x=“GO”=(6, 14):

1. (10 points) Decrypt back the **first** ciphertext block encrypted in a)

The first ciphertext block encrypted is c=”E!” = (4, 28). Thus,

We see that decrypted x’=”GO” is the same as the original plaintext, x=”GO”.

**Q4.** **(30 points).** Consider the following description of a DES round from the Lecture notes:

“The round key Ki is 48 bits. The R input is 32 bits. This R input is first expanded to 48 bits by Expansion/Permutation (E table):

|  |
| --- |
| Expansion/Permutation (E table) |
| 32 | 1 2 3 4 | 5 |
| 4 | 5 6 7 8 | 9 |
| 8 | 9 10 11 12 | 13 |
| 12 | 13 14 15 16 | 17 |
| 16 | 17 18 19 20  | 21 |
| 20 | 21 22 23 24 | 25 |
| 24 | 25 26 27 28 | 29 |
| 28 | 29 30 31 32 | 1 |

The resulting 48 bits are XORed with Ki. This 48 bit result passes through a substitution function that produces 32-bit output, which is permuted by Permutation function (P):

|  |
| --- |
| Permutation function( P ) |
| 16 7 20 21 29 12 28 171 15 23 26 5 18 31 102 8 24 14 32 27 3 919 13 30 6 22 11 4 25 |

The role of S-boxes is illustrated in Fig. 3.9:



The substitution consists of a set of 8 S-boxes, each of which accepts 6 bits input and produces 4 bits as output.

These transformations are:



Each row of an S-box defines a general reversible substitution: middle 4 bits of each group of 6-bit input are substituted by S-box output, 1st and last 6th bits define what particular substitution out of four to use.”

Assume, the output of XOR of the 48-bit expanded-permuted right half is 0x56234112abcd, in hexadecimal.

1. (6 points) How many bits will be output by S-box S2? 4 bits
2. (18 points) What bit values will be output by S-box S2? Give explanations

Output of XOR in binary is

0x56234112abcd= 0101 0110 0010 0011 0100 0001 0001 0010 1010 1011 1100 1101

The first 6 bits, 010101 go to S1, the second 6 bits, 100010, go to S2. According to Expansion/permutation table, two end bits, 10, define row number, that is 2, and 4 middle bits, 0001, define column number that is 1. Thus, on the cross of row 2 and column 1 in the S-box S2, we find decimal value, 14, that, in binary, is 1110, which is output by S2.

1. (6 points) In what position of the output, Ri, will be the first bit output by S2? Give explanations.

The first bit output by S2, is in the position 5 after S-boxes, and according to permutation function, P, will appear in the position 13 of Ri as appearing in Row 2, Column 5 of the permutation P table above.