ECMM462: Fundamentals of Security

Continuous Assessment

Diego Marmsoler

Deadline: **12:00pm (noon) 31st July 2023**

The CA is worth 40% of your final mark and intended to last for 40 hours. You can get up to 100 marks in total split into four exercises:

Topic	Marks	Tasks	Target Time
Symmetric Encryption	25	4	10 h
Asymmetric Encryption	25	4	10 h
P			

Format of Submissi ... d a file for every task (t1-t4). ... d exercise should be in E2/t1 ... ubmission system BART (http ... MT.

Python Libraries Fo ... Python you can use only basic ... encryption libraries).

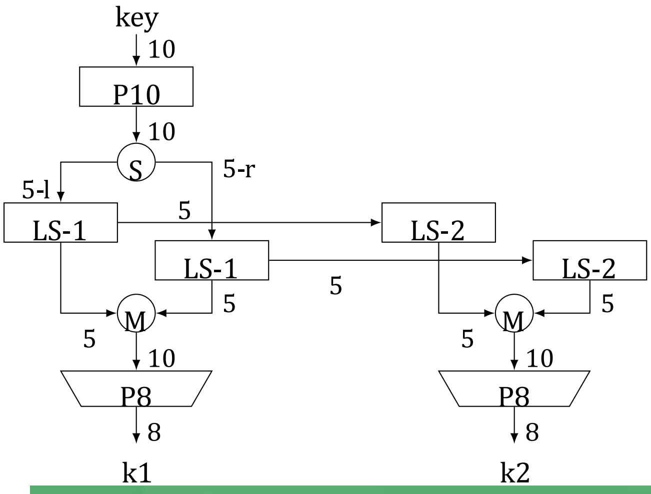
Questions If you ha ... post them to the correspondi ... close your identity then the f

1 Symmetric E

In the lecture we ... mple of a modern symmetric encryption cipher. In the following, we briefly introduce a simplified version of the DES.

1.1 Key Generation

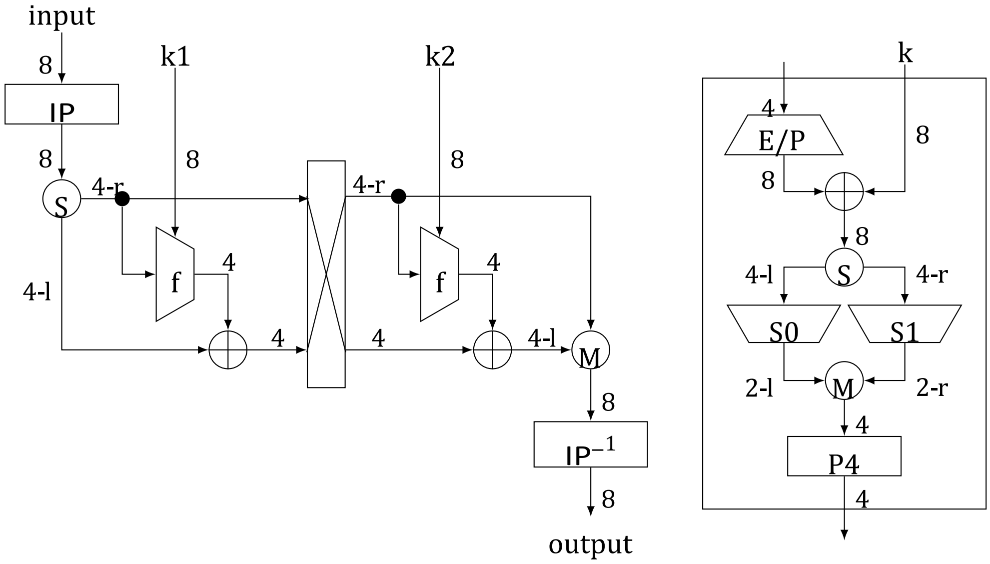
In simplified DES, one master key is used to generate multiple sub-keys (so called round keys). Figure 1 depicts the algorithm to generate round keys in our simplified version of DES: It takes a 10-bit master key as input and produces two 8-bit round keys using permutations and shifts.



The operator S s … of length 5: 5-l (for left) den … 5 bits. The operator M concat … quence of length 10 (again, t … he second 5 bits). The functio … shifts LS-1 and LS-2 are des … put bit in the output bit sequ … g P10, for example, the 3^{th} in … es the 2^{nd} output bit, etc.

1.2 Encryption an

Encryption in our … structure depicted in Fig. 2. … Fig. 3. The permutation table



...ction f

The substitution ... tables get 4 bits as input: the ... l the third specify the column ... es the row 01 (which corres... 0 (which corresponds to 2 i... table S0, for example, can b... 1 (which corresponds to 01...

Decryption in ou... pt that we swap the round ke...

1.3 Tasks

In the following yo... simplified version of DES:

Table 1: Permutation Tables.

	1	2	3	4	5	6	7	8	9	10
P10	3	5	2	7	4	10	1	9	8	6
LS-1	2	3	4	5	1	-	-	-	-	-

LS-2	3	4	5	1	2	-	-	-	-	-
P8	6	3	7	4	8	5	10	9	-	-

To this end you should use the following key:

1001110000

T1.1 Compute the round keys (including intermediate results).

T1.2 Convert the above text to bit representation in ASCII and encrypt the first letter using the algorithm (including intermediate results).

T1.3 Decrypt the first letter using the algorithm (including intermediate results).

T1.4 Implement our simplified version of DES in Python. The program should be called myDes and take three input parameters:

- the firs
- the sec
- the thir

The program

```
>myDes
>010101
```

Table 2: Permutation Tables.

Table 3: S_0

Table 4: S_1,

	1	2	3	4
IP	2	6	3	1
IP-1	4	1	3	5
E/P	4	1	2	3
P4	2	4	3	1

	0	1	2
0	1	0	3
1	3	2	1
2	0	2	1
3	3	1	3

	0	1	2	3
0	0	1	2	3
1	2	0	1	3
2	3	0	1	0
3	2	1	0	3

2 Asymmetric Encryption

In the following we describe a simple description of an asymmetric encryption mechanism. The idea is to represent encryption and decryption with table lookups. To this end, three types of tables are used:

M1 is just a sequence of N elements and contains a random permutation of all integers between 1 and N. For example $m1$ =(4,3,2,5,1) could be an example for N = 5. It is used to co … ve assume that our priv … ing to our example tabl …

M2 is an N ×N ma … tion of all integers betw …

3 1 2

2 3 5 1

It is used for … 2 and key 2, we get $m2$ …

M3 is an N × N mat … ation of all integers betw …

The tables must be constructed in a way such that for all k and p, with $1 \le k,p \le N$ the following property holds:

$$M3(M2(M1(k),p),k)= p \qquad (1)$$

2.1 Tasks

T2.1 Construct an example of M1, M2, and M3 for $N = 5$. Hint: first, randomly create M1 and M2 and then construct M3 such that property Eq. (1) holds.

T2.2 Encrypt the number 3 and then decrypt it again.

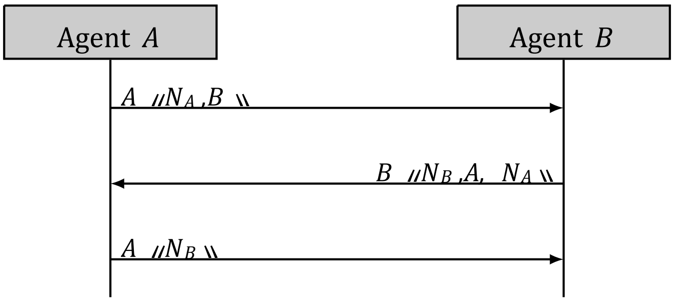
T2.3 Is the scheme secure? Explain why/why not.

T2.4 Implement the key generation scheme in Python. It should be called myPKE and take one input parameter which represents N. It should then generate three random tables $m1$, $m2$, and $m3$ which satisfy Eq. 1. For example:

```
>myPKE 5
>m1:
>4,3,2,5, 1
>m2:
>3,1,2,5
>1,4,3,2
>4,5,2,3
>3,2,1,4
>2,3,5,1
>m3:
>.....
>.....
>.....
>.....
>.....
```

3 Protocol Verification

Consider the following protocol where $X\langle\langle M\rangle\rangle$ denotes a message M digitally signed by agent X:



The aim of the protocol is to establish authentication between two agents. In particular, at the end of the protocol, agent B needs to be sure that nonce N_A was indeed send by age

3.1 Tasks

T3.1 Formalize the

T3.2 State the secu .

T3.3 The protocol

T3.4 One simple fi fix of the protocol and

4 Access Cont

Assume you are de ing to the Bell-LaPadula mod "high" for the modules they they take. Moreover, exams a ments as "low" for the corre

4.1 Tasks

T4.1 Define a starting state $z_0 = (b_0, m_0, f_0)$ in which the following holds:

- *Alice* is a lecturer for module *Security*. *Bob* is a student of *Security* and *Eve* a student of *Logics*.

- *Ex1* is an exam for module *Logics*. *Hw1* is a homework for *Security* and *A1* an assignment for *Logics*.
- *Alice* has given edit (read/write) rights for *Ex1*, read rights for *A1*, and write rights for *Hw1*. *Bob* has read/write rights for *Hw1* and *Eve* for *A1*.
- Currently *Bob* is editing (reading and writing) *Hw1* whereas *Alice* is reading *A1*.
- The current security level of all subjects to an object is initialized with their maximum security level for this object.

T4.2 Argue whether or not the state described above is secure.

T4.3 Describe the new state arising when *Bob* stops writing to *Hw1* and *Alice* changes the exam (i.e., executes read/write rights on the exam), and use the security theorem to argue whether or not the new state is secure.

T4.4 Assume *Alice* ... ute write rights on it. ... sponding protection st... s, and (iii) show that it ...