Department of Electronics
Faculty of Electrical Engineering, Mechanical Engineering and
Naval Architecture (FESB)
University of Split, Croatia

# Laboratory Exercises II: Symmetric and Asymmetric Cryptography

**Keywords: CrypTool, historical ciphers (Caesar, Vernam), DES, CBC and ECB encryption/decryption, RSA, Diffie-Hellman key agreement**

Dr. Mario Čagalj

mario.cagalj@fesb.hr
http://www.fesb.hr/~mcagalj

November 14, 2007

# Introduction

In this set of exercises we will study the basics of *symmetric* (secret key) and *asymmetric* (public key) cryptography. For this purpose, we will use *CrypTool*, an excellent program by means of which cryptographic functions and mechanisms can be easily demonstrated and analyzed (Figure 1). We will learn about both classical and modern cryptography. We will begin our study with a couple of exercises related to classical ciphers such as Caesar's and Vernam. Using CrypTool, we will demonstrate their weaknesses and limitations. Then, we will focus on modern ciphers, primarily Data Encryption Standard (DES), and on basic operation modes of block ciphers such as Electronic Codebook (EBC) and Cipher Block Chaining (CBC). The "cut-and-paste" attack against EBC and CBC will be demonstrated. Finally, in the last set of exercises we will study basic public key cryptosystems RSA and Diffie-Hellman.

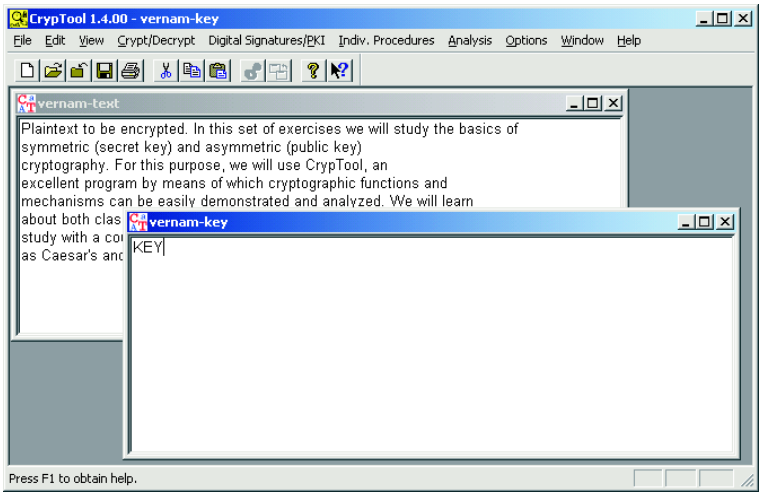


Figure 1: CrypTool.

# Exercise 1

In this exercise we will demonstrate and study two representatives of classical ciphers, namelly, Caesar's cipher and Vernam (one-time pad) cipher. We will also perform cryptanalysis of a simple permutation cipher.

Recall, Caesar's cipher falls in the category of monoalphabetic substitution ciphers (i.e., each element from the plaintext will be replaced with a unique element from the space of ciphertexts). For this reason, a cipehertext preserves the relative frequency at which plaintext elements appear in the corresponding plaintext. In Vernam cipher encryption is performed by means of *eXclusive-OR* (XOR) logical operation (plaintext is XORed with an encryption key). If an encryption key is chosen randomly and is

at least as long as the plaintext to be encrypted, XOR encryption (*one-time pad*) is provably (perfectly) secure.

## Task 1.1. Caesar's Cipher

1. Given that the Caesar's cipher is used, recover the plaintext that corresponds to the following ciphertext: `Sodlqwhaw wr eh hqfubswhg.` Describe your approach to cryptanalysis of the ciphertext. (Try to recover the plaintext without assistance of CrypTool.)

2. Use CrypTool to check your answer to the first challenge (task). In order to do this, you should first create a new document in CrypTool by clicking on the icon "New" (or choose "File ▷ New"). Write the challenge ciphertext in a newly open document. In the main menu, under "Analysis" tab, select "Symmetric Encryption (classic) ▷ Ciphertext only ▷ Caesar" and follow the instructions.

3. Assume that we choose `M` (that is, `K=13`) as an encryption key in Caesar's cipher. What is the result ($\mathtt{C}_i$) of the following (double) encryption

$$\mathtt{C}_i = \mathbf{E}\Big(\mathtt{K}, \mathbf{E}(\mathtt{K}, \mathtt{M}_i)\Big) \ ,$$

   where $\mathtt{M}_i$ is an arbitrary plaintext element? Explain your answer.

4. How many encryptions with key `K=2` is needed before observing the same effect as in the previous example (with `K=13`)?

## Task 1.2. Vernam and One-Time Pad Cipher

1. Consider the following letter encodings:

| letter | A | E | I | M | O | R | T | V |
|---|---|---|---|---|---|---|---|---|
| encoding | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

   A message `M = MARIO` is Vernam encrypted into ciphertext `C = AOAMV`. Find the corresponding encryption key (please note that the key is 4 letters long). Provide details of your cryptanalysis.

2. Consider the following two ciphertexts $\mathtt{C}_1$= `MAOEE` and $\mathtt{C}_2$=`RTITR` that are obtained by Vernam encrypting messages $\mathtt{M}_1$ and $\mathtt{M}_2$, respectively, under the same encryption key. The letter encodings is the same as in the first task. Encrypted messages are two names. Let us denote with $\mathtt{m}_{i,k}$ the $k$th letter in message $\mathtt{M}_i$. The following is known about messages (names): $\mathtt{m}_{1,1}$=`R` and $\mathtt{m}_{2,4}$=`T`. Using this information, try to recover messages $\mathtt{M}_1$ and $\mathtt{M}_2$, as well as the encryption key. Provide details of your cryptanalysis.

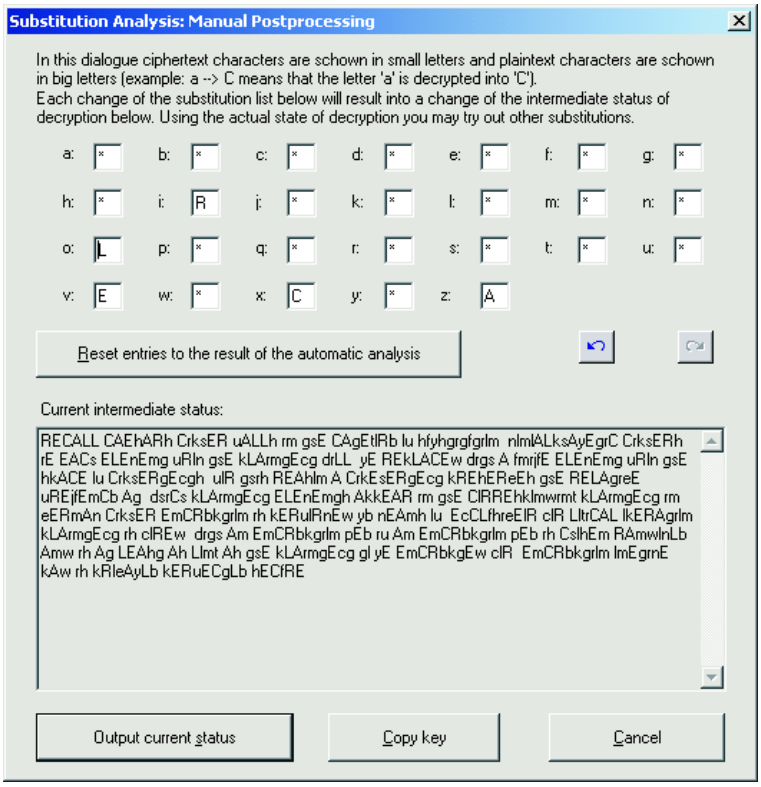


Figure 2: Manual analysis of a ciphertext obtained using a simple substitution cipher.

## Task 1.3. Monoalphabetic Substitution Cipher

This type of a cipher is similar to the Caesar's cipher, i.e. every letter of the plaintext is replaced by a different letter of the alphabet, with the difference that the encryption key can be any permutation of the plaintext elements. In this way the key space increases from 26 (in the case of Caesar's cipher) to 26!. This cipher, however, is still vulnerable to the"relative frequency"-based attack.

Your task is to decrypt the following message that was encrypted using a monoalphabetic substitution cipher.

```
Ivxzoo, Xzvhzi'h xrksvi uzooh rm gsv xzgvtlib lu hfyhgrgfgrlm
nlmlzokszyvgrx xrksvih (r.v., vzxs vovnvmg uiln gsv kozrmgvcg droo
yv ivkozxvw drgs z fmrjfv vovnvmg uiln gsv hkzxv lu xrksvigvcgh).
Uli gsrh ivzhlm, z xrkvsvigvcg kivhvievh gsv ivozgrev uivjfvmxb zg
dsrxs kozrmgvcg vovnvmgh zkkvzi rm gsv xliivhklmwrmt kozrmgvcg. Rm
Evimzn xrksvi, vmxibkgrlm rh kviulinvw yb nvzmh lu vCxofhrev-LI
(CLI) oltrxzo lkvizgrlm (kozrmgvcg rh CLIvw drgs zm vmxibkgrlm
pvb). Ru zm vmxibkgrlm pvb rh xslhvm izmwlnob zmw rh zg ovzhg zh
```

```
olmt zh gsv kozrmgvcg gl yv vmxibkgvw, CLI vmxibkgrlm (lmv-grnv
kzw) rh kilezyob (kviuvxgob) hvxfiv.
```

Recover the corresponding encryption key knowing that the first ciphertext word corresponds to plaintext "`Recall`".

Use CrypTool as follows for your cryptanalysis. Write the challenge ciphertext in a newly open document. In the main menu, under "Analysis" submenu, select "Symmetric Encryption (classic) ▷ Manual Analysis ▷ Substitution..." and perform cryptanalysis using the fact that the first word in the ciphertext corresponds to plaintext "`Recall`" (Figure 2).

# Exercise 2

In this exercise we will focus on modern ciphers, primarily on the most prominent block cipher DES. We will study a relationship between DES and its extension 3-DES (triple DES). Recall, 3-DES was introduced in order to compensate for a short encryption key (only 56 bits) in the original DES cipher. 3-DES is still widely used for protection of data confidentiality. In October 2000, Rijndael encryption algorithm (AES) was chosen as a more robust and more flexible replacement of the DES encryption algorithm. Similar to DES, AES is also a block cipher but it supports a variable block length and a variable key length (i.e., 128, 192 and 256 bits).
We will further study two basic operation modes of block ciphers, namely, ECB and CBC. More specifically, we will show that it is possible to rearrange the ECB and CBC ciphertext blocks in such a way that all (or some) of the blocks decrypt correctly. This is a well-known *cut-and-paste* attack.

## Task 2.1. Histogram Analysis of the Data Encryption Standard

In this task we compare a frequency histogram of a document before and after encryption with the DES cipher. We use CrypTool to accomplish this. Quoting CrypTool help: "The histogram of a document expresses the frequency distribution of the characters of a document in graphical form in a corresponding window."

Create a new document in CrtypTool and fill it with a longer English text. Alternatively, open an existing English text. Save this document for later reference.

1. In the main menu, click on the "View" submenu and select "As HexDump". This is to convert ASCII representation of the plaintext document into the corresponding hexadecimal representation.

2. In the main menu, open the "Analysis" submenu and select "Tools for Analysis ▷ Histogram" to obtain the histogram of the plaintext (un-encrypted) document (Figure 4). Save the result.

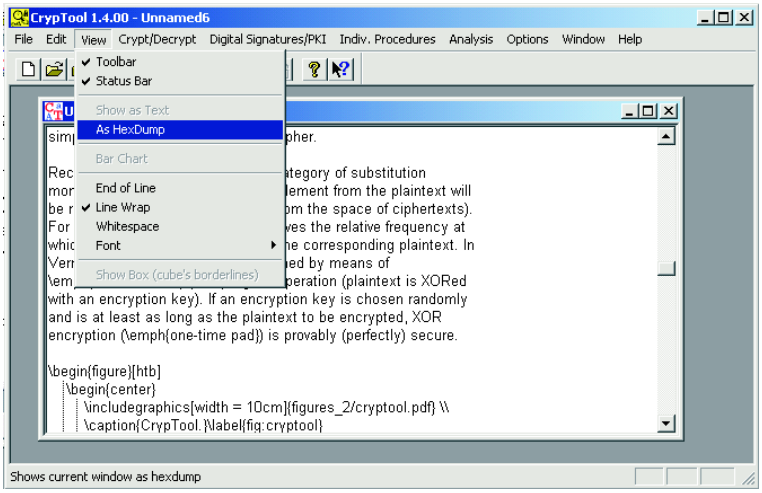


Figure 3: CrypTool: Hexadecimal representation of an ASCII text.

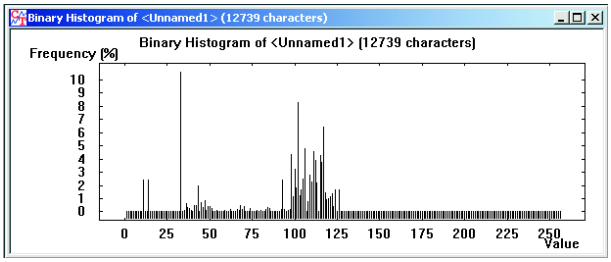


Figure 4: Histogram of an unencrypted message.

3. Encrypt the plaintext document (English text) using the DES cipher. In CrypTool click on the window with the plaintext document in order to make it active. In the main menu, under the "Crypt/Decrypt" submenu select "Symmetric (modern) ▷ DES CBC...". Enter the encryption key and encrypt the plaintext document.

4. Repeat step 2 above to obtain the histogram of the encrypted text (ciphertext). Compare the result with the histogram obtained in step 2. Describe/explain your observations.

## Task 2.2. Histogram Analysis of the Caesar's Cihper

Repeat all the steps from the previous task, while using Caesar's cihper with an arbitrary key instead of the DES cipher. Compare the obtained results with the results from the previous task. Describe/explain your observations.

## Task 2.3. Triple DES (3-DES)

1. Assume that a message m is encrypted using 3-DES (in the ECB mode) with the following key (hex value):

   K = 11 22 33 44 55 66 77 88 AA BB CC DD EE FF FF FF

   The encrypted message is to be decrypted using only regular 1-DES (not 3-DES). Explain how is this done.

2. Use CrypTool and encrypt your name with 3-DES in the ECB mode under the following two encryption keys:

   $\mathtt{K}_1$ = 11 22 33 44 55 66 77 88 AA BB CC DD EE FF FF FF
   $\mathtt{K}_2$ = 11 22 33 44 55 66 77 88 11 22 33 44 55 66 77 88

   Decrypt resulting ciphertexts using 1-DES cipher. Provide any intermediate results that you obtain. One among the keys $\mathtt{K}_1$ and $\mathtt{K}_2$ enables "fast" decryption with the 1-DES cipher (a single application of 1-DES). Which one? Please explain your answer.

## Task 2.4. "Cut-and-Paste" Attack on ECB and CBC Modes

Let us denote with $\mathtt{M}_1$ $\mathtt{M}_2$... $\mathtt{M}_k$ a message M broken up into $k$ 64 bit segments. Also, let us denote with K the corresponding encryption key. Then, the ECB encryption mode can be mathematically described as follows:

$$\begin{aligned} \mathtt{C}_i &= \mathbf{E}(\mathtt{K}, \mathtt{M}_i),\ i = \{1, 2, \ldots, k\} \\ \mathtt{M}_i &= \mathbf{D}(\mathtt{K}, \mathtt{C}_i),\ i = \{1, 2, \ldots, k\}\ . \end{aligned}$$

Similarly, for the CBC mode we have:

$$\begin{aligned} \mathtt{C}_1 &= \mathbf{E}(\mathtt{K}, \mathtt{M}_1 \oplus \mathtt{IV}) \\ \mathtt{C}_i &= \mathbf{E}(\mathtt{K}, \mathtt{M}_i \oplus \mathtt{C}_{i-1}),\ i = \{2, 3, \ldots, k\} \\ \mathtt{M}_1 &= \mathbf{D}(\mathtt{K}, \mathtt{C}_1) \oplus \mathtt{IV} \\ \mathtt{M}_i &= \mathbf{D}(\mathtt{K}, \mathtt{C}_i) \oplus \mathtt{C}_{i-1},\ i = \{2, 3, \ldots, k\}\ . \end{aligned}$$

1. Consider the following message M:

   M=Bob's salary is $25000--Tom's salary is $15000.

   Break the message (plaintext) up into 64 bit long plaintext segments ($\mathtt{M}_1$ $\mathtt{M}_2$... $\mathtt{M}_k$). Note that each letter in the message is an 8 bit ASCII character; each "space" (blank) counts as a single ASCII character. Use ␣ sign to denote blank characters. For example, the first 64 bit plaintext segment is $\mathtt{M}_1$=Bob's␣sa .

2. Using CrypTool, encrypt the above message with DES in the ECB mode using key `K = 11 22 33 44 55 66 77 88`. Write down resulting 64 bit ciphertext blocks $\texttt{C}_1$ $\texttt{C}_2$... $\texttt{C}_k$.

3. Exchange ciphertext blocks $\texttt{C}_1$ and $\texttt{C}_4$ in the above sequence of ciphertext blocks to obtain the following sequence of ciphertext blocks:

   $$\texttt{C}_4\ \texttt{C}_2\ \texttt{C}_3\ \texttt{C}_1\ \texttt{C}_5 \ldots\ \texttt{C}_k$$

   Decrypt the resulting ciphertext using the key from step 2. What message do you obtain? Please explain.

4. Repeat steps 1-3 but now use DES in the CBC mode. Contrast the decrypted text with the one obtained when the ECB mode is used. Explain your observations.

### Task 2.5. Controlled Plaintext Changes in the CBC Mode

Your task is to cause a controlled change in the decrypted message by modifying an appropriate CBC ciphertext block.

1. Use CrypTool and encrypt message

   `M=Bob's salary is $25000--Tom's salary is $15000.`

   with DES in the CBC mode. Choose the encryption key at will.

2. In the resulting ciphertext sequence modify an appropriate ciphertext block so that it causes the following change in the decrypted message: `$15000` $\rightarrow$ `$.5000`. Provide details of your actions. (Hint: Use CrypTool to accomplish this task.)

3. Do all ciphertext blocks decrypt correctly after this modification? Explain your answer.

4. Does the CBC (and/or ECB) mode of encryption ensure data integrity? Please explain using experience gained from the present and the previous task.

## Exercise 3

In this exercise we will focus on asymmetric or public key cryptography. We will study some important properties of a (textbook) RSA public key cryptosystem. We will demonstrate certain attacks against RSA - "factoring-based" and "chosen-ciphertext" attack. Finally, we will touch upon the Diffie-Hellman key exchange protocol and its security in face of passive and active attacks.

## Task 3.1. RSA Encryption

The RSA encryption algorithm works with numbers. As your task is to encrypt some textual messages, we obviously need a method for coding of a message into numbers. We next describe one such a method that is used in CrypTool. The method is called *b-adic* (where $b$ is the number of plaintext elements) and works as follows. Suppose that the plaintext alphabet consists the following elements:

<space>ABCDEFGHIJKLMNOPQRSTUVWXYZ ,

that is, there are in total 27 different plaintext elements. Next, the alphabet elements are coded as follows:

$$\begin{array}{rcl} \texttt{<space>} & \rightarrow & 0 \\ \texttt{A} & \rightarrow & 1 \\ \texttt{B} & \rightarrow & 2 \\ & \vdots & \\ \texttt{Z} & \rightarrow & 26\,. \end{array}$$

Depending on the bit length of the RSA modulus $N$ and the selected alphabet, in CrypTool you can adjust the block length used with RSA encryption. Now, assume that we want to encrypt plaintext: MARIO. Assuming that we want to work with block of length 2, we obtain the following blocks:

MA#RI#O<space> .

Encoding this with the above code, we obtain:

13 01 # 18 09 # 15 00 .

Finally, *b-adic*, that is, 27-adic coding of a numerical representation of the message is obtained according to the following formula:

<letter 1> $\times$ 27 + <letter 2> .

By applying this formula to the numerical representation of our message, we finally obtain:

352 # 495 # 405 .

It is this sequence of numbers that is encrypted into ciphertext (in the "block-by-block" fashion).

1. You are asked to encrypt your name using RSA with a small modulus $N$ (that is, $N < 100000$). Choose $N$ such that the length of the resulting RSA modulus $N$ allows you to work with blocks of length 2. Use CrypTool to accomplish this as follows:

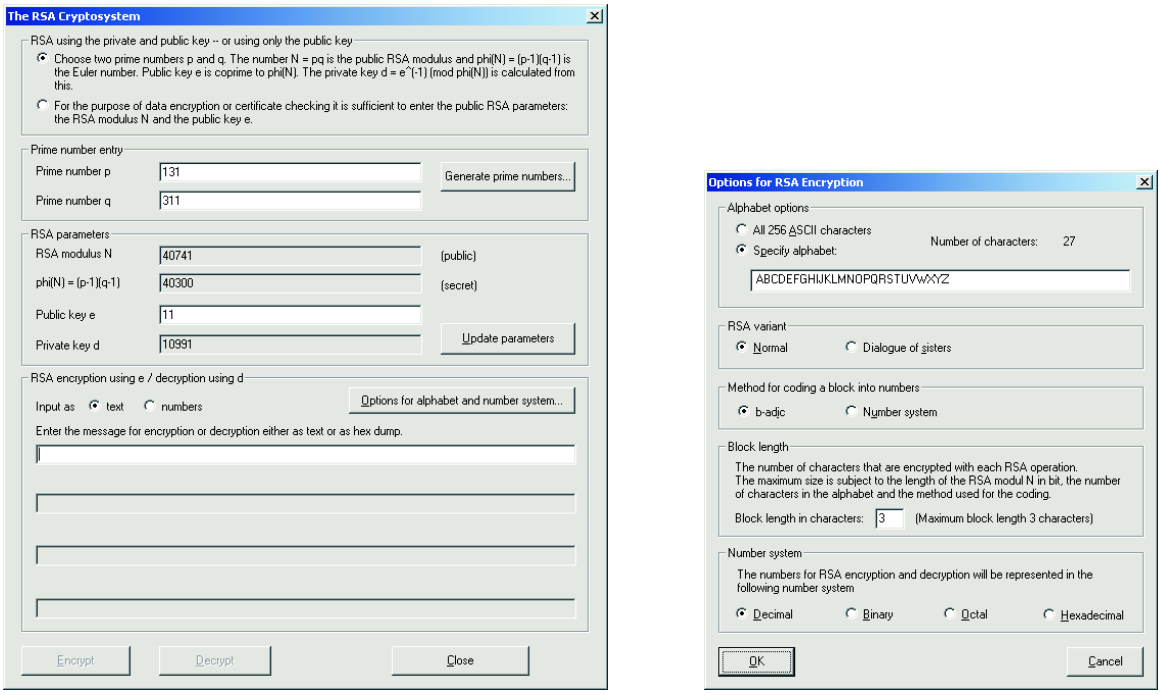


Figure 5: CrypTool: The RSA Cryptosystem.

- In the main menu, under "Crypt/Decrypt" select "Asymmetric ▷ RSA Demonstration..." to open "The RSA Cryptosystem" window (Figure 5(left)).
- Click on button "Options for alphabet and numeric system..." to open "Options for RSA Encryption" (Figure 5)(right).
- In the "Options..." window, in tab "Alphabet options" check "Specify alphabet:", in tab "RSA variant" check "Normal", in tab "Method for coding a block into numbers" check "b-adic", set the block length to 2, and finally choose the "Decimal" number system. Click "OK" button to save the settings.

2. Show that CrypTool RSA encryption works by encrypting the first block of the message "manually". You are allowed to use only multiplication and squaring operations (and of course your calculator). In order to make this process manageable, use "reduction by modulo" property, that is,

$$(a \times b) \bmod N \equiv \Big((a \bmod N) \times (b \bmod N)\Big) \bmod N \ .$$

   Provide the steps of your calculation.

3. After decryption of a given ciphertext, we have to decode the result back to its initial plaintext representation. How would you do this? Decode 354 into a

plaintext alphabet `<space>ABCDEFGHIJKLMNOPQRSTUVWXYZ`, assuming the block length of 2 and 27-adic coding.

## Task 3.2. Chosen Ciphertext Attack on RSA

This is an attack against the textbook version of the RSA algorithm. In this attack, an attacker first chooses a message and encrypts it the victim's public key. Then, the attacker asks the victim to sign (decrypt) for him a specially crafted related message. Due to the following property of RSA

$$\mathbf{E}(\mathtt{PU}, \mathtt{M}_1) \times \mathbf{E}(\mathtt{PU}, \mathtt{M}_2) = \mathbf{E}(\mathtt{PU}, \mathtt{M}_1 \times \mathtt{M}_2) \ , \tag{1}$$

the attacker can easily recover any message encrypted with the victim's private key, without ever learning this private key.
For example, the attacker wants to decrypt the following ciphertext $C = M^e \bmod N$, without knowing the private key $d$. The attacker proceeds as follows. Knowing the victim's public key $e$, he prepares the following message

$$X = (C \times 2^e) \bmod N \ ,$$

gives it to the victim and asks her to sign it. The victim signs message $X$ with its private key and sends the result $Y$ back to the attacker.

$$Y = X^d \bmod N$$

Using $Y$ and equation (1), the attacker can retrieve the encrypted message $M$ as follows:

$$\begin{aligned} X^d &= \Big((C \bmod N) \times (2^e \bmod N)\Big)^d \\ &= \Big((M^e \bmod N) \times (2^e \bmod N)\Big)^d \\ &= \Big((2 \times M)^e \bmod N\Big)^d \\ &= (2 \times M)^{ed} \bmod N \\ &= 2 \times M \ . \end{aligned}$$

1. Show by example that equation (1) holds for the RSA encryption algorithm. Please provide details of your solution. Use CrypTool to accomplish this task.

2. Demonstrate by example the chosen ciphertext attack against RSA. Please provide details of your solution. Use CrypTool to accomplish this task.

## Task 3.3. Attack on RSA by Factoring Modulus $N$

By factoring the RSA modulus $N$, the attacker learn prime numbers $p$ and $q$. From this, the attack can calculate Euler's $\Phi(N)$ function as follows

$$\Phi(N) = (p-1)(q-1) \ .$$

Furthermore, the public key $e$ is also known by the attacker. Therefore, he can calculate the corresponding private key $d$, which is the inverse modulo $\Phi(N)$ of $e$, that is,

$$d = e^{-1}\Big(\text{mod } \Phi(N)\Big) \ .$$

1. Decrypt the ciphertext given below knowing that the public key $e$ is 11 and the RSA modulus $N$ is 40741. The plaintext alphabet is encrypted using the block length of 2 and 27-adic coding (for details on encoding please check Task 3.1).

   (Hint: Use CryptTool to factor the modulus $N$. In the main menu, under "Analysis") submenu, select "Asymmetric Encryption ▷ Factorization of a Number..." and follow instructions therein.

   Ciphertext (read from left-to-right then down):

   ```
   01437 # 32647 # 36721 # 14238 # 24974 # 27041 # 01170 # 31888 #
   08891 # 20670 # 07453 # 36364 # 38274 # 06244 # 11809 # 28159 #
   12942 # 30673 # 21533 # 12400 # 18298 # 34309 # 36364
   ```

## Task 3.4. Diffie-Hellman Key Exchange Protocol

1. The Diffie-Hellman protocol is secure against passive (eavesdropping) attacks. Explain why.

2. Describe a possible *man-in-the-middle* (MITM) attack on the Diffie-Hellman protocol and explain its consequences.

3. (**Note: This problem originally appears in book "Information Security: Principles and Practice" by Mark Stamp.**) Assume that Malice mounts the MITM attack on Diffie-Hellman key exchange protocol as illustrated in Figure 6. Let $Y_A = g^{X_A} \bmod p$, and $Y_B = g^{X_B} \bmod p$ be Diffie-Hellman public keys of Alice and Bob, respectively. Private keys $X_A$, $X_B$ and $X_M$ remain known only to the respective owners Alice, Bob and Malice. Suppose that Malice wants to establish a single Diffie-Helmman key, $K_{ABM} = g^{X_A X_B X_M} \bmod p$, that he, Alice and Bob all share. Does the attack illustrated in Figure 6 accomplish this? Explain your answer.


Figure 6: The MITM attack against Diffie-Hellman protocol executed between Alice and Bob.