

## Assignment6

### Lab on Short Message RSA Attacks and Padding (100 points)

In **short message attack of RSA**, if it is known that Alice is sending a four-digit number to Bob, Eve can easily try plaintext numbers from 0000 to 9999 to find the plaintext. Therefore, short message must be padded with random bits. **If you are Eve, show that you are able to find the plaintext containing four digit numbers given ciphertext.**

Optimal asymmetric encryption padding (OAEP) is recommended when short messages are encrypted with RSA algorithms. The following is the encryption and decryption processes of OAEP.

- Encryption
  - Pad the message to make m-bit message M, if M is less than m-bit
  - Choose a random number r
  - User one-way function G that inputs r-bit integer and outputs m-bit integer. This is the mask.
  - $P1 = M \oplus G^{\circ}$
  - $P2 = H(P1) \oplus r$ , function H inputs m-bit and outputs k-bit
  - $C = E(P1 || P2)$ . User RSA encryption here
- Decryption
  - $P = D(P1 || P2)$
  - Bob first recreates the value of r:  
 $H(P1) \oplus P2 = H(P1) \oplus H(P1) \oplus r = r$
  - Bob recreates msg:  
 $G(r) \oplus P1 = G(r) \oplus G(r) \oplus M = M$

**Pad your message with OAEP padding and then encrypt by RSA.**

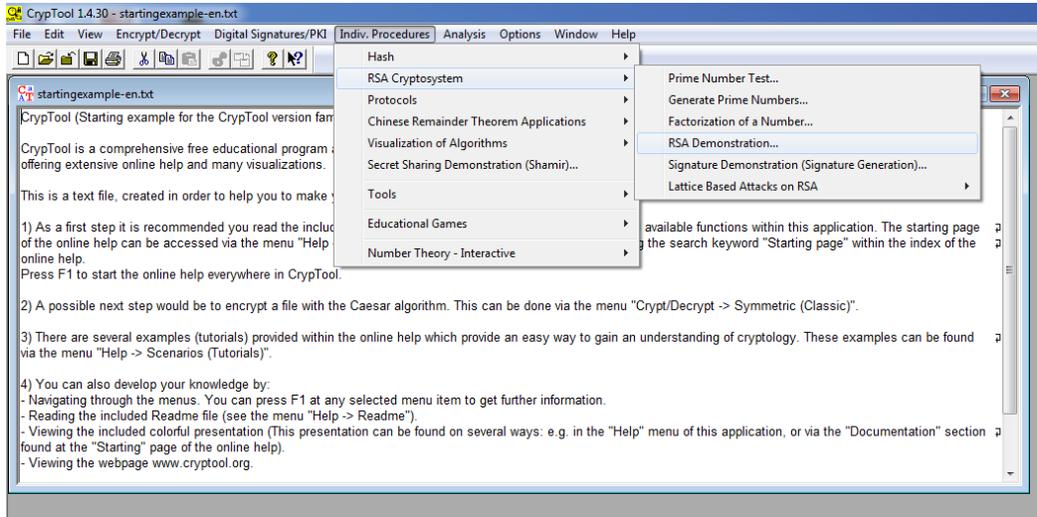
**What to submit:**

A report describes how you find the unpadded short plaintext (30 points), describes what you have observed after you apply OAEP padding (30 points), and discusses feasibility of short message attack after padding (10 points).

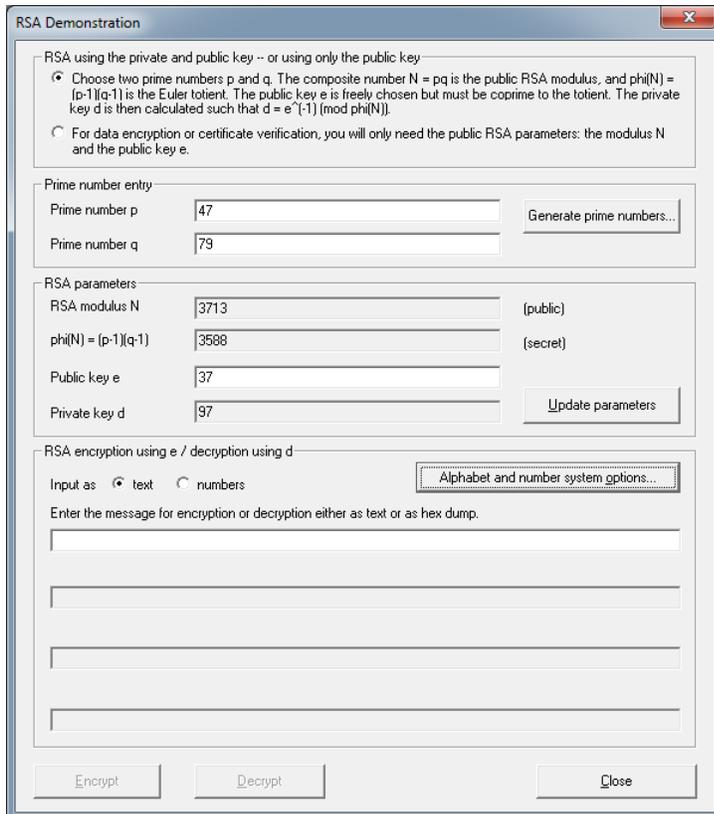
# Lab on RSA Encryption and Factorization Attacks (practice)

Encryption or decryption of messages using the RSA key pair.

## 1. Select Individual Procedures/RSA Cryptosystem/RSA Demonstration



2. Enter the RSA key  $p=47$ ,  $q=79$ ,  $e=37$ . The parameters  $N = p \cdot q=3713$  and  $\phi(N)=3588$  and  $d=97$  are calculated.



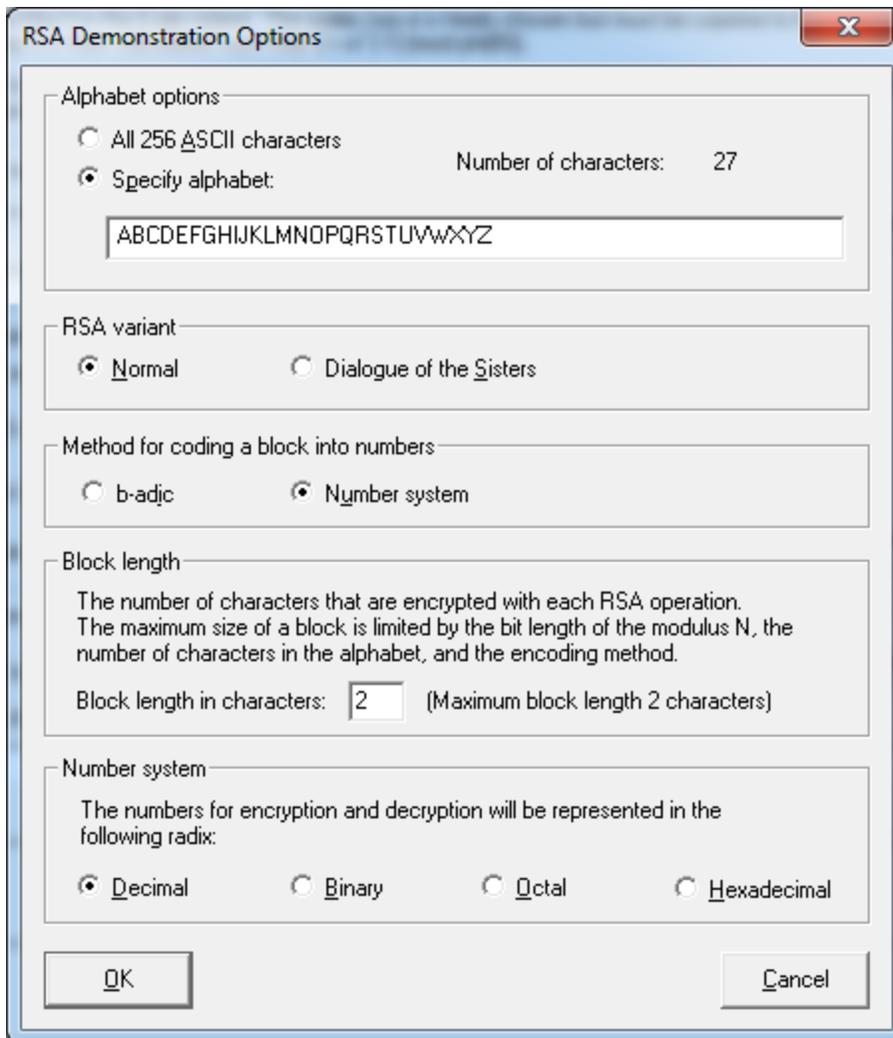
3. Click **Alphabet and number system options**

The image shows a dialog box titled "RSA Demonstration Options" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Alphabet options:** Contains two radio buttons: "All 256 ASCII characters" (selected) and "Specify alphabet:". To the right of the second radio button is the text "Number of characters: 256". Below these is a text input field containing the string "ABCDEFGHIJKLMNOPQRSTUVWXYZ".
- RSA variant:** Contains two radio buttons: "Normal" (selected) and "Dialogue of the Sisters".
- Method for coding a block into numbers:** Contains two radio buttons: "b-adjc" (selected) and "Number system".
- Block length:** Contains a text input field with the value "1" and the text "(Maximum block length 1 characters)". Above the input field is a paragraph: "The number of characters that are encrypted with each RSA operation. The maximum size of a block is limited by the bit length of the modulus N, the number of characters in the alphabet, and the encoding method."
- Number system:** Contains a paragraph: "The numbers for encryption and decryption will be represented in the following radix:" followed by four radio buttons: "Decimal" (selected), "Binary", "Octal", and "Hexadecimal".

At the bottom of the dialog are two buttons: "OK" and "Cancel".

4. Choose **specify alphabet** under Alphabet Options and **number system** under Method for coding of text into number. Enter **2** in Block length in characters.



5. To confirm your entries, click on OK. You can now enter the input the text, “**WORKSHOP AT CHATTANOOGA**”, in the input line and click on the **Encrypt** button.

**RSA Demonstration**

RSA using the private and public key -- or using only the public key

- Choose two prime numbers p and q. The composite number  $N = pq$  is the public RSA modulus, and  $\phi(N) = (p-1)(q-1)$  is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that  $d = e^{-1} \pmod{\phi(N)}$ .
- For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.

Prime number entry

Prime number p: 47

Prime number q: 79

Generate prime numbers...

RSA parameters

RSA modulus N: 3713 (public)

$\phi(N) = (p-1)(q-1)$ : 3588 (secret)

Public key e: 37

Private key d: 97

Update parameters

RSA encryption using e / decryption using d

Input as:  text  numbers

Alphabet and number system options...

Input text

WORKSHOP AT CHATTANOOGA

The Input text will be separated into segments of Size 2 (the symbol '#' is used as separator).

W O # R K # S H # O P # A # T # C H # A T # T A # N O # O G # A

Numbers input in base 10 format.

2315 # 1811 # 1908 # 1516 # 0001 # 2000 # 0308 # 0120 # 2001 # 1415 # 1507 # 0100

Encryption into ciphertext  $c[i] = m[i]^e \pmod{N}$

1999 # 3408 # 2545 # 2798 # 0001 # 3284 # 3613 # 1404 # 2932 # 0208 # 1095 # 3306

Encrypt Decrypt Close

6. To decrypt, copy text in Encryption into ciphertext **1999 # 3408 # 2545 # 2798 # 0001 # 3284 # 3613 # 1404 # 2932 # 0208 # 1095 # 3306** to input text area. And click **Decrypt** button.



**Encryption of the message with block length 1 v.s. encryption of the message with block length 2.**

1. Create the RSA key  $p=251$ ,  $q=269$ ,  $e=65537$ . The value of  $N$  is \_\_\_\_\_, the value of  $\phi(N)$  is \_\_\_\_\_, the value of private key  $d$  is \_\_\_\_\_.

**RSA Demonstration**

RSA using the private and public key -- or using only the public key

- Choose two prime numbers p and q. The composite number  $N = pq$  is the public RSA modulus, and  $\phi(N) = (p-1)(q-1)$  is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that  $d = e^{-1} \pmod{\phi(N)}$ .
- For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.

Prime number entry

Prime number p: 251

Prime number q: 269

Generate prime numbers...

RSA parameters

RSA modulus N: 67519 (public)

$\phi(N) = (p-1)(q-1)$ : 67000 (secret)

Public key e: 65537

Private key d: 2473

Update parameters

RSA encryption using e / decryption using d

Input as:  text  numbers

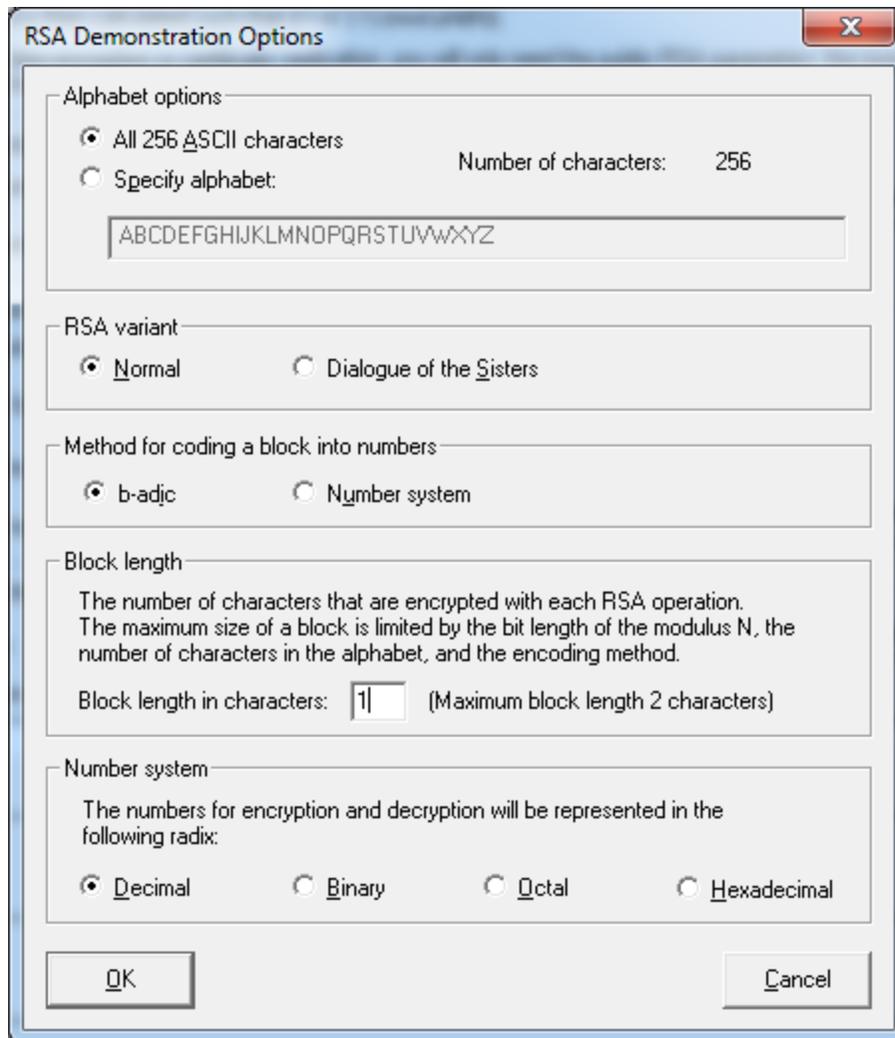
Alphabet and number system options...

Enter the message for encryption or decryption either as text or as hex dump.

Encrypt Decrypt Close

2. Click **Alphabet and number system options**

Choose **All 256 ASCII characters** under Alphabet options, **b-adic** under Method for coding and a **block** into numbers and **1** in Block length in characters.



3. To confirm your entries, click on **OK**. You can now enter the input the text, "**RUBY FALLS!**", in the input line and click on the **Encrypt** button.

**RSA Demonstration** [X]

RSA using the private and public key -- or using only the public key

- Choose two prime numbers p and q. The composite number  $N = pq$  is the public RSA modulus, and  $\phi(N) = (p-1)(q-1)$  is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that  $d = e^{-1} \pmod{\phi(N)}$ .
- For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.

Prime number entry

Prime number p: 251 Generate prime numbers...

Prime number q: 269

RSA parameters

RSA modulus N: 67519 (public)

$\phi(N) = (p-1)(q-1)$ : 67000 (secret)

Public key e: 65537

Private key d: 2473 Update parameters

RSA encryption using e / decryption using d

Input as:  text  numbers Alphabet and number system options...

Input text

RUBY FALLS!

The Input text will be separated into segments of Size 1 (the symbol '#' is used as separator).

R # U # B # Y # # F # A # L # L # S # !

Numbers input in base 10 format.

082 # 085 # 066 # 089 # 032 # 070 # 065 # 076 # 076 # 083 # 033

Encryption into ciphertext  $c[i] = m[i]^e \pmod{N}$

58455 # 30900 # 02593 # 37260 # 28092 # 04562 # 16634 # 36959 # 36959 # 39103 # 03240

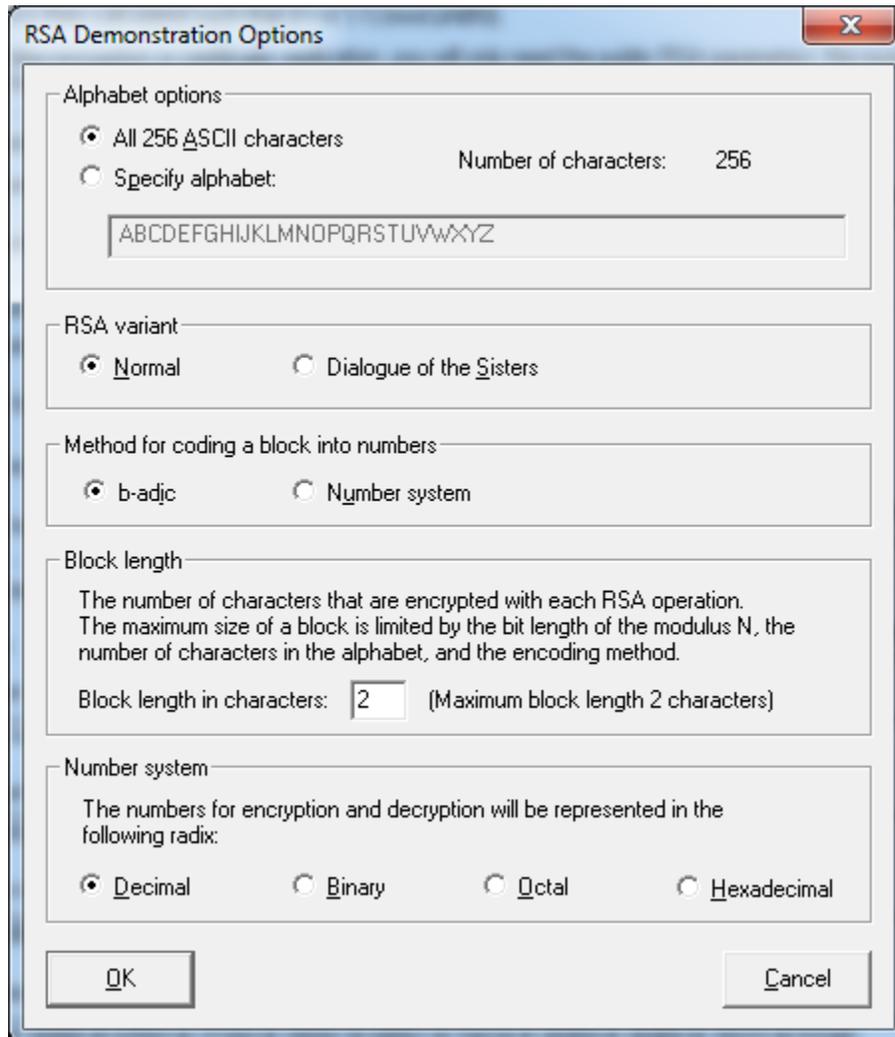
Encrypt Decrypt Close

The encrypted version of this is the number sequence is \_\_\_\_\_

The number “#” serves here to visually split up the individual numbers. If you insert these numbers into the input line and then choose **Decrypt**, the original plaintext will be restored.

4. Click **Alphabet and number system options**

Choose **All 256 ASCII characters** under Alphabet options, **b-adic** under Method for coding and a block into numbers and **2** in Block length in characters.



The image shows a dialog box titled "RSA Demonstration Options" with a close button (X) in the top right corner. The dialog is divided into several sections:

- Alphabet options:** Contains two radio buttons. The first, "All 256 ASCII characters", is selected. To its right, the text "Number of characters: 256" is displayed. The second radio button is "Specify alphabet:", followed by a text input field containing the string "ABCDEFGHIJKLMNOPQRSTUVWXYZ".
- RSA variant:** Contains two radio buttons: "Normal" (selected) and "Dialogue of the Sisters".
- Method for coding a block into numbers:** Contains two radio buttons: "b-adic" (selected) and "Number system".
- Block length:** Contains a text input field with the value "2". To the right of the field, the text "(Maximum block length 2 characters)" is shown. Above the field, there is explanatory text: "The number of characters that are encrypted with each RSA operation. The maximum size of a block is limited by the bit length of the modulus N, the number of characters in the alphabet, and the encoding method."
- Number system:** Contains four radio buttons: "Decimal" (selected), "Binary", "Octal", and "Hexadecimal". Above these buttons, there is explanatory text: "The numbers for encryption and decryption will be represented in the following radix:".

At the bottom of the dialog, there are two buttons: "OK" on the left and "Cancel" on the right.

5. To confirm your entries, click on **OK**.

**RSA Demonstration** X

RSA using the private and public key -- or using only the public key

- Choose two prime numbers p and q. The composite number  $N = pq$  is the public RSA modulus, and  $\phi(N) = (p-1)(q-1)$  is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that  $d = e^{-1} \pmod{\phi(N)}$ .
- For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.

Prime number entry

Prime number p:  Generate prime numbers...

Prime number q:

RSA parameters

RSA modulus N:  (public)

$\phi(N) = (p-1)(q-1)$ :  (secret)

Public key e:

Private key d:  Update parameters

RSA encryption using e / decryption using d

Input as:  text  numbers Alphabet and number system options...

Input text

The Input text will be separated into segments of Size 2 (the symbol '#' is used as separator).

Numbers input in base 10 format.

Encryption into ciphertext  $c[i] = m[i]^e \pmod{N}$

6. You will receive a cipher text that is only half as long:

---

## Attack on RSA encryption with short RSA modulus (practice)

The analysis is performed in two stages: first of all the prime factorization of the RSA modulus is calculated using factorization, and then in the second stage the secret key for encryption of the message is determined. After this, the cipher text can be decrypted with the cracked secret key.

We will figure out plaintext given

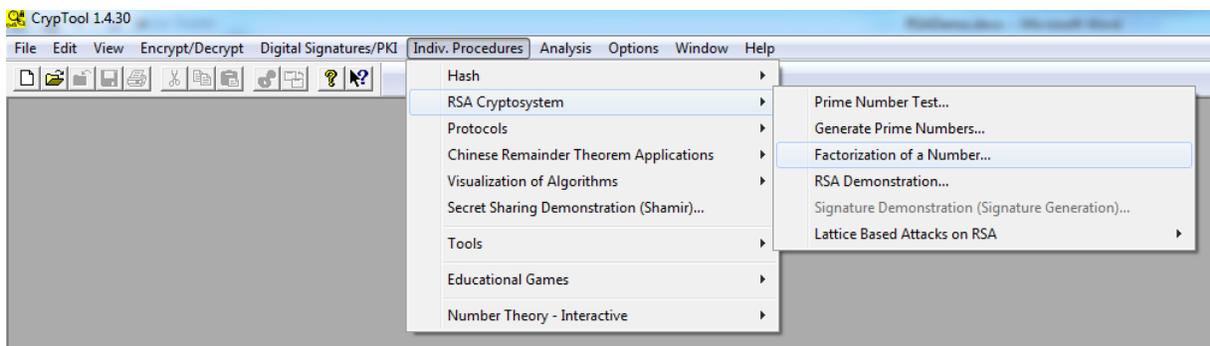
RSA modulus  $n = 63978486879527143858831415041$

Public exponent  $e = 17579$

Cipher text = 45411667895024938209259253423, 16597091621432020076311552201, 46468979279750354732637631044, 32870167545903741339819671379

1. Factorization of the RSA modulus with the aid of prime factorization.

To break down the natural number, select menu **sequence Indiv. Procedure/RSA Cryptosystem / Factorization of a Number.**

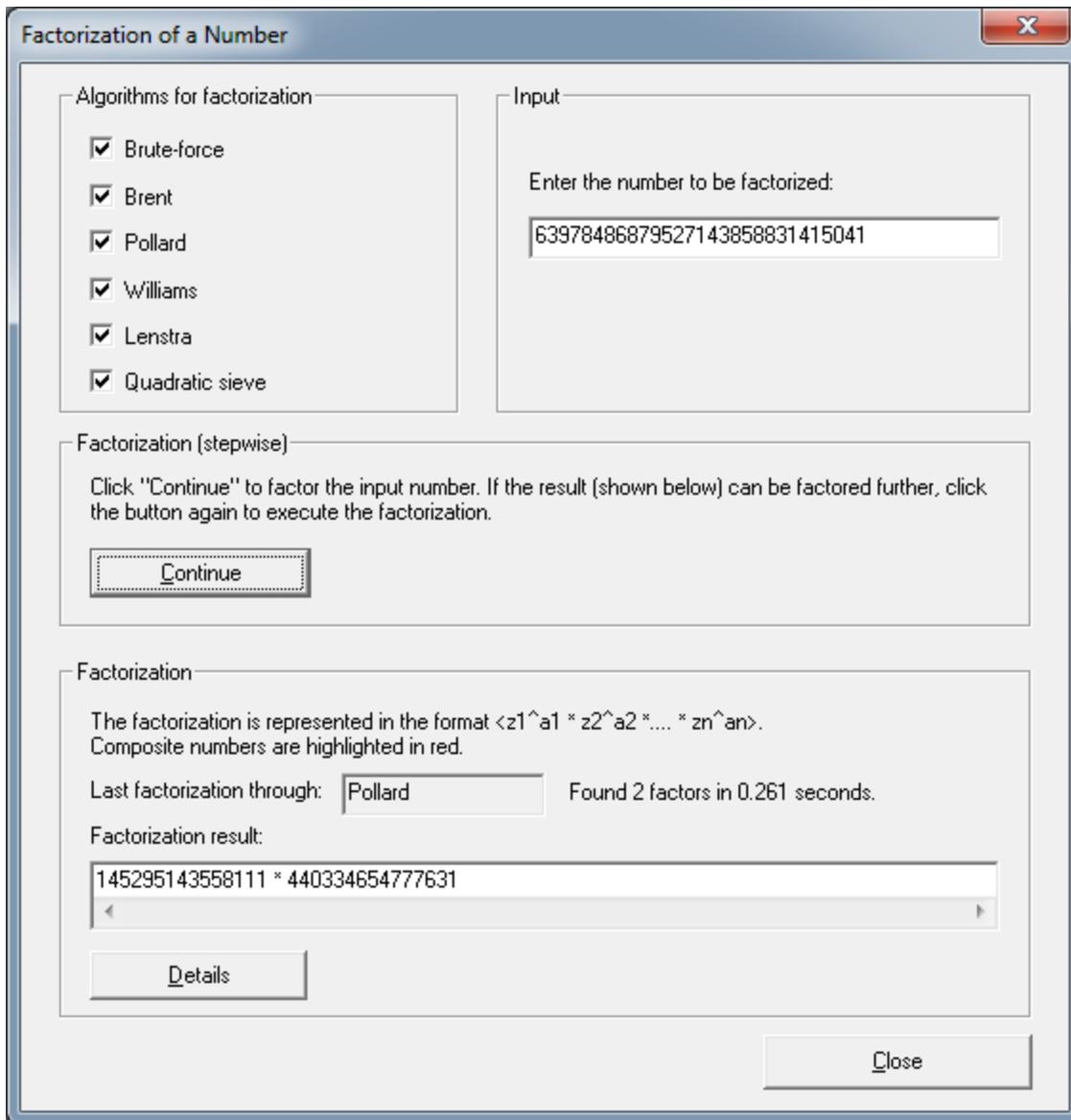


2. The two components of the public key is

RSA modulus  $n = 63978486879527143858831415041$

Public exponent  $e = 17579$

Enter  $n=63978486879527143858831415041$  as input and click **Continue.**



It is interesting to see which procedure broke down the RSA modulus the fastest.

2. Calculate the secret key **d** from the prime factorization of **n** and the public key **e**:

With the knowledge of the prime factors  $p = 145295143558111$  and  $q = 440334654777631$  and the public key  $e = 17579$ , we are in a position to decrypt the ciphertext.

3. Open the next dialog box via menu selection **Indiv. Procedure/RSA Cryptosystem/RSA Demonstration**:

4. Enter **p = 145295143558111** and **q = 440334654777631** and the public key **e = 17579**.

5. Click on **Alphabet and number system options** and make the following settings:

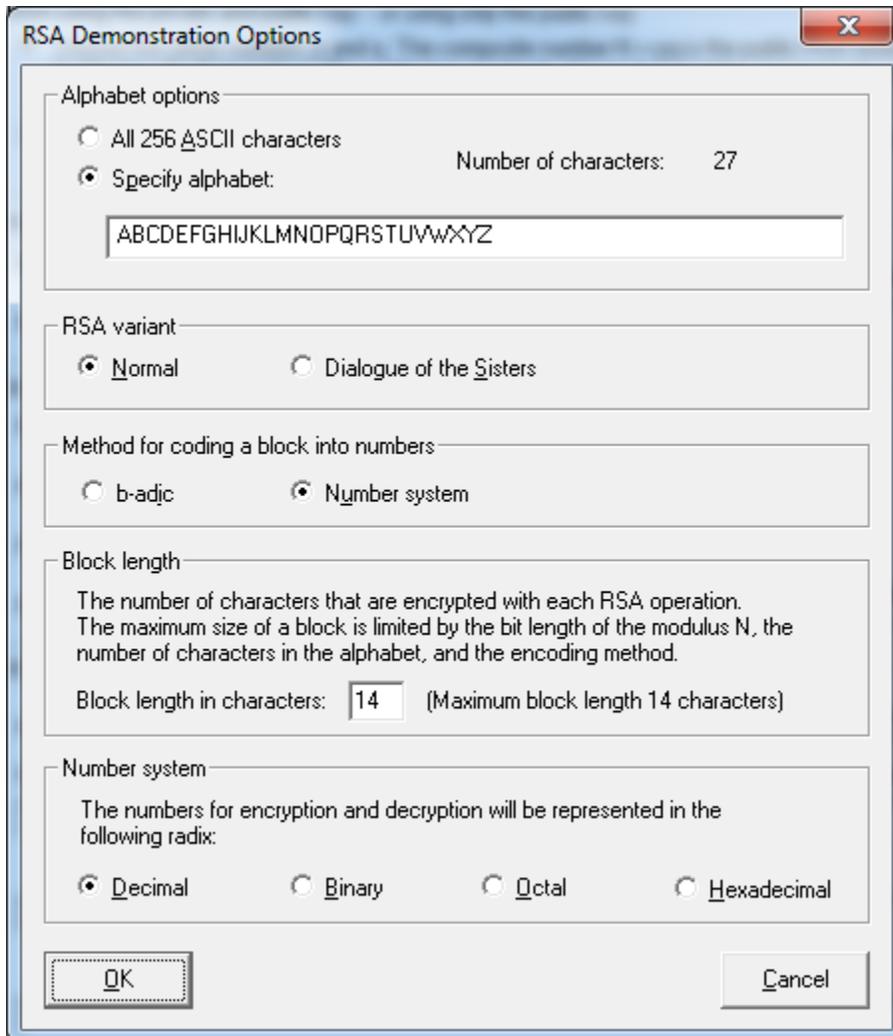
Alphabet options: **Specify alphabet**

RSA variant: **Normal**

Method for coding a block into number: **Number system**

Block length: **14**

Number system: **Decimal**



6. Enter the following cipher text in the input text field. And click **Decrypt** button.

45411667895024938209259253423,  
16597091621432020076311552201,  
46468979279750354732637631044,  
32870167545903741339819671379

**RSA Demonstration** ✕

RSA using the private and public key -- or using only the public key

- Choose two prime numbers p and q. The composite number  $N = pq$  is the public RSA modulus, and  $\phi(N) = (p-1)(q-1)$  is the Euler totient. The public key e is freely chosen but must be coprime to the totient. The private key d is then calculated such that  $d = e^{-1} \pmod{\phi(N)}$ .
- For data encryption or certificate verification, you will only need the public RSA parameters: the modulus N and the public key e.

Prime number entry

Prime number p:

Prime number q:

RSA parameters

RSA modulus N:  (public)

$\phi(N) = (p-1)(q-1)$ :  (secret)

Public key e:

Private key d:

RSA encryption using e / decryption using d

Input as:  text  numbers

Ciphertext coded in numbers of base 10

Decryption into plaintext  $m[i] = c[i]^d \pmod{N}$

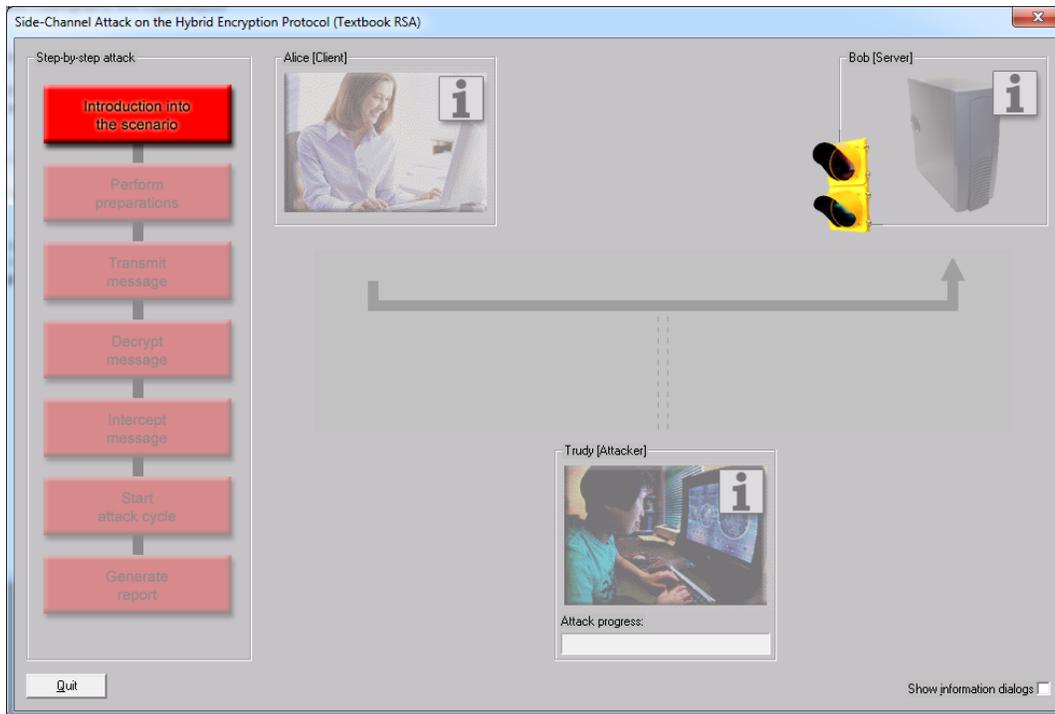
Output text from the decryption (into segments of size 14; the symbol '#' is used as separator).

Plaintext

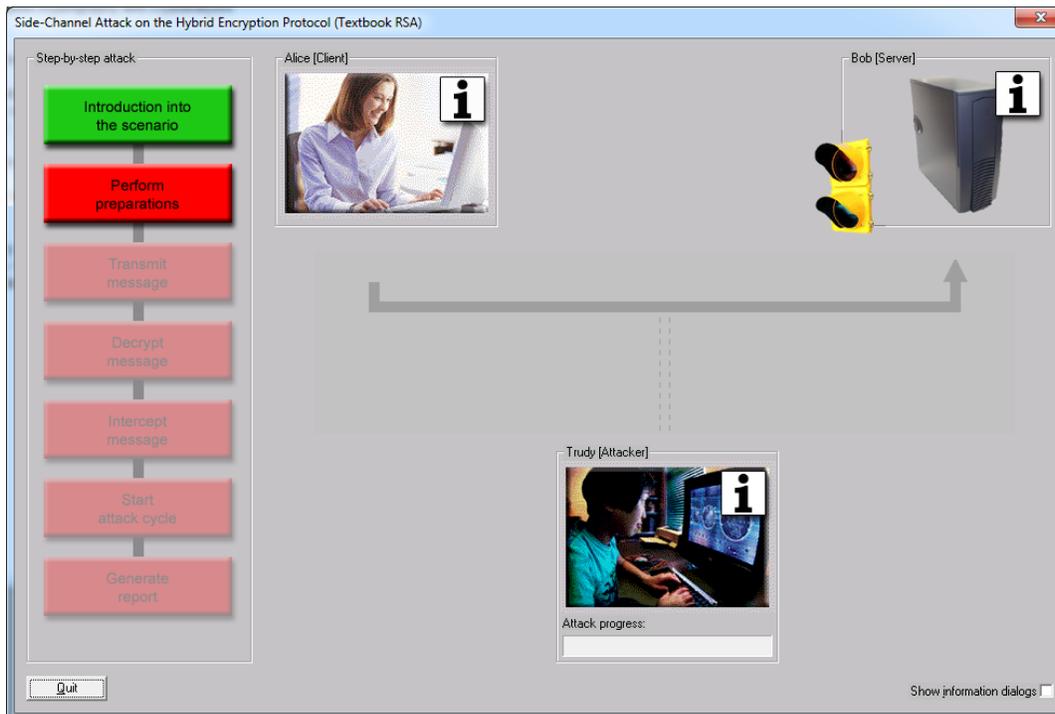
Check your results: **“NATURAL NUMBERS ARE MADE BY GOD”**

## Side Channel Attack to RSA: (10 points)

1. Select from menu: “Analysis” \ “Asymmetric Encryption” \ “Side-Channel Attack on Textbook RSA”



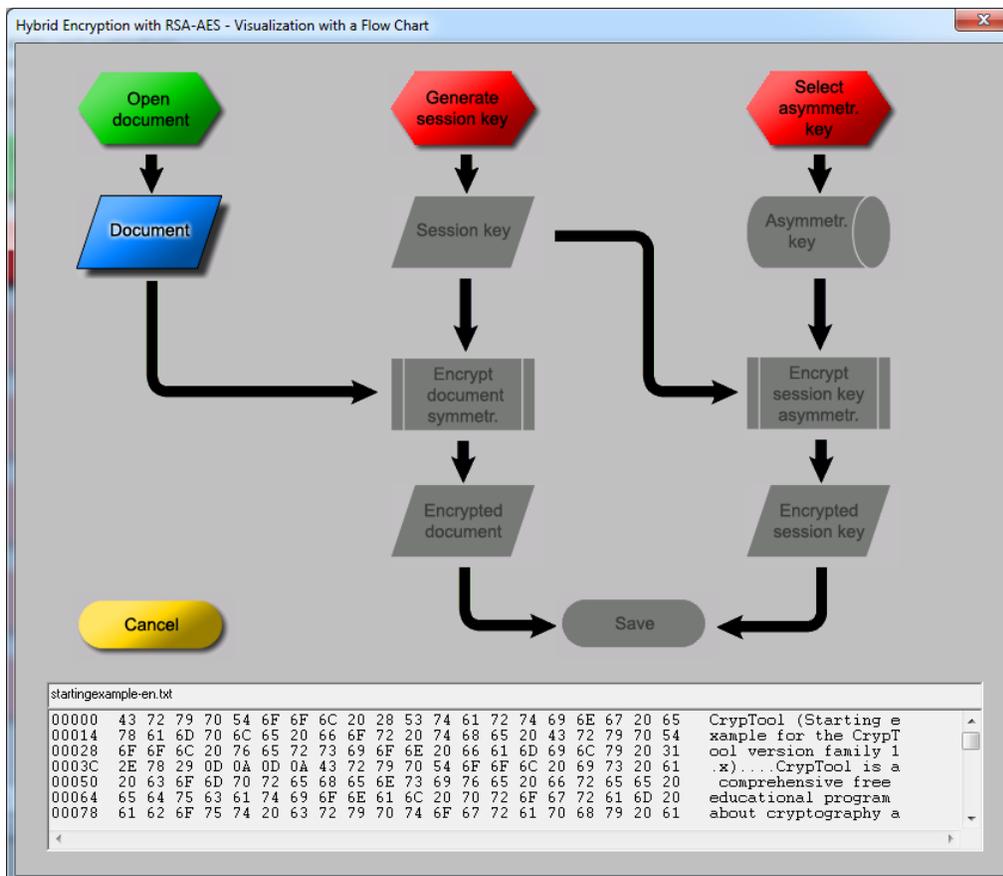
2. Click “Introduction to the scenario”.



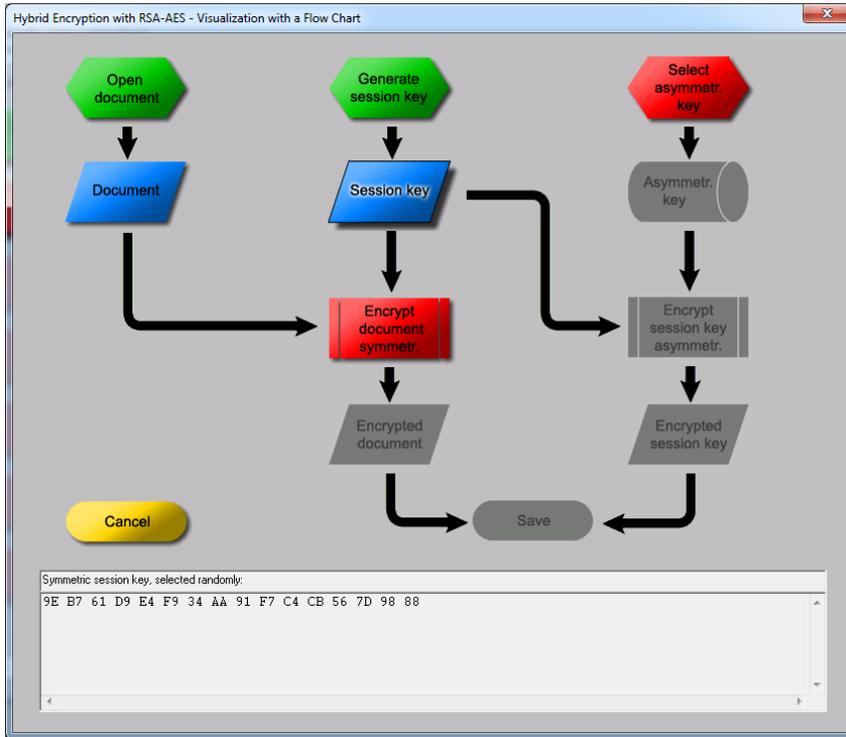
3. Click "Perform preparation" and click "OK"



4. Click "OK" again.



- Click **“Generate session key”** and **“Session Key”**. The generated session key is **“9E B7 61 D9 E4 F9 34 AA 91 F7 C4 CB 56 7D 98 88”**.

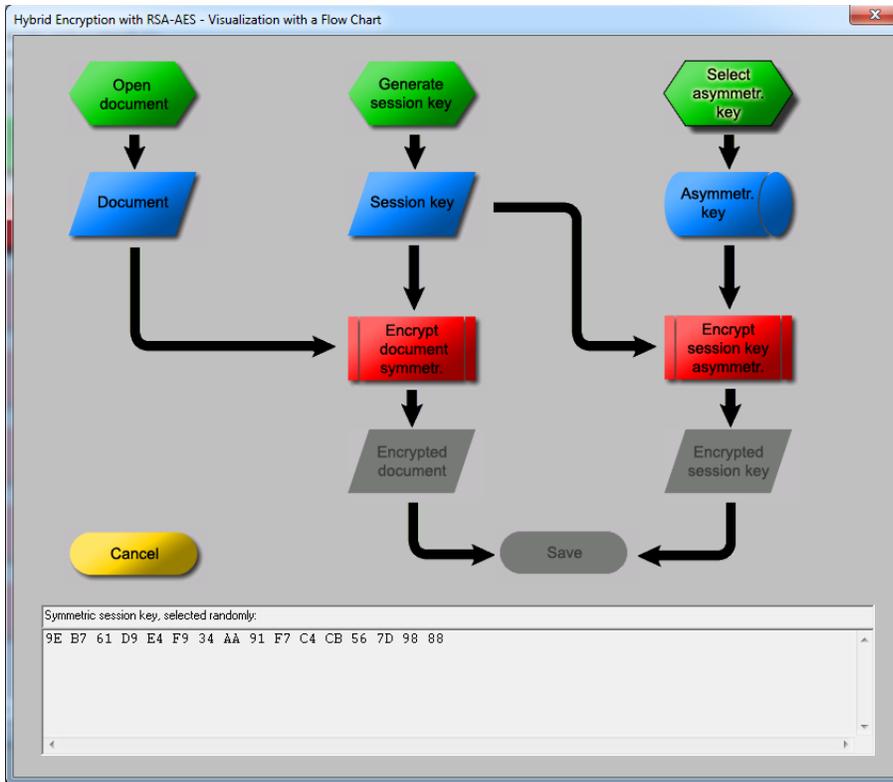


- Click **“Select asymmetr. key”**.

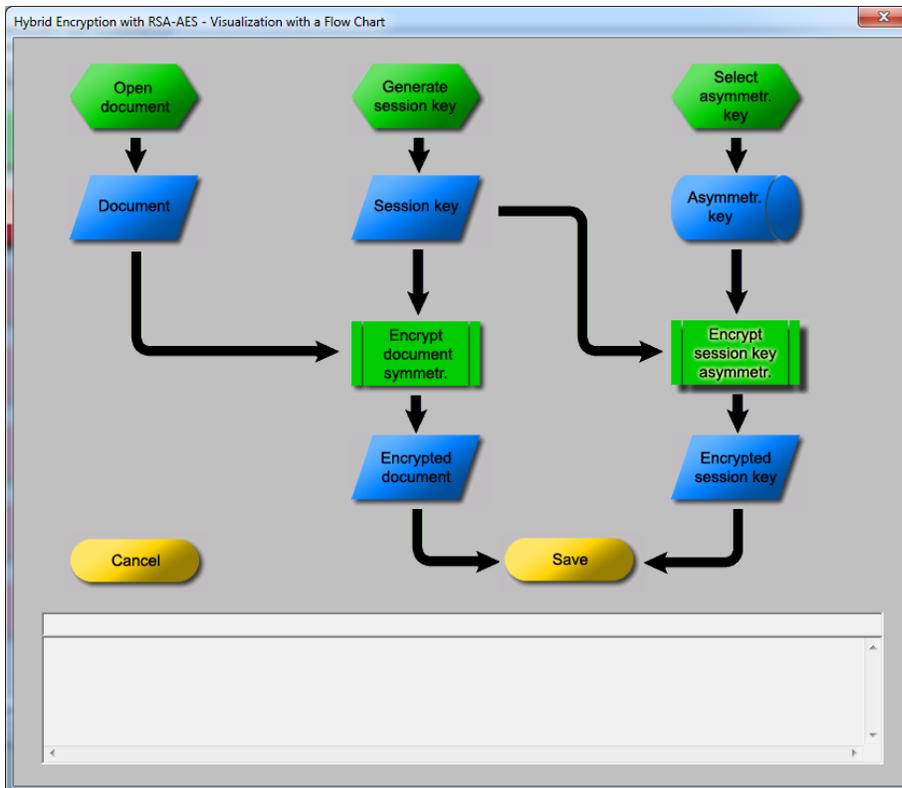
Last name	First name	Key type	Key identifier	Created	Internal ID no.
SideChannelAt...	Bob	RSA-512	PIN=1234	06.07.2006 05:51:34	1152179494
Smith	John	RSA-1024	Smith Key	12.07.2011 17:09:15	1310504955
Smith	Mary	RSA-304	Mary key	13.07.2011 09:54:04	1310565244

Note: Here only names are displayed, which have an RSA key.

- Select Bob’s key and click **“OK”**.



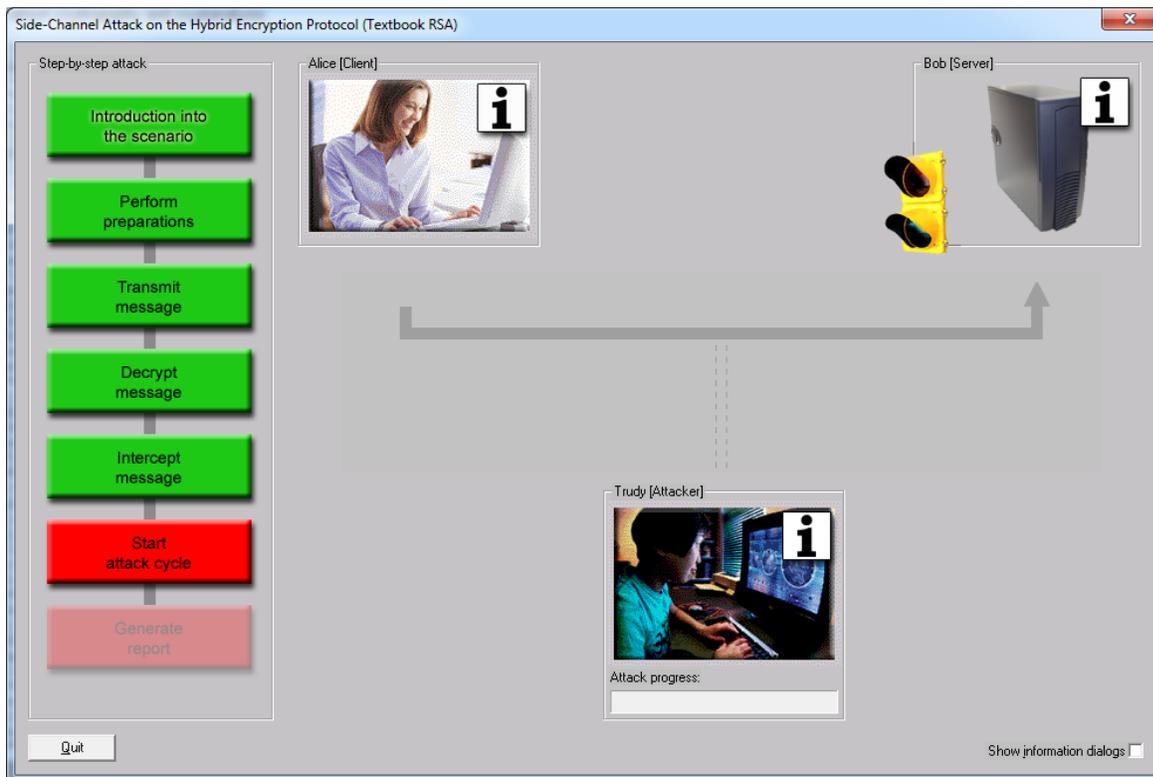
8. Click “Encrypt document symmetry.”, “Encrypt session key asymmetry.” and “Save”.



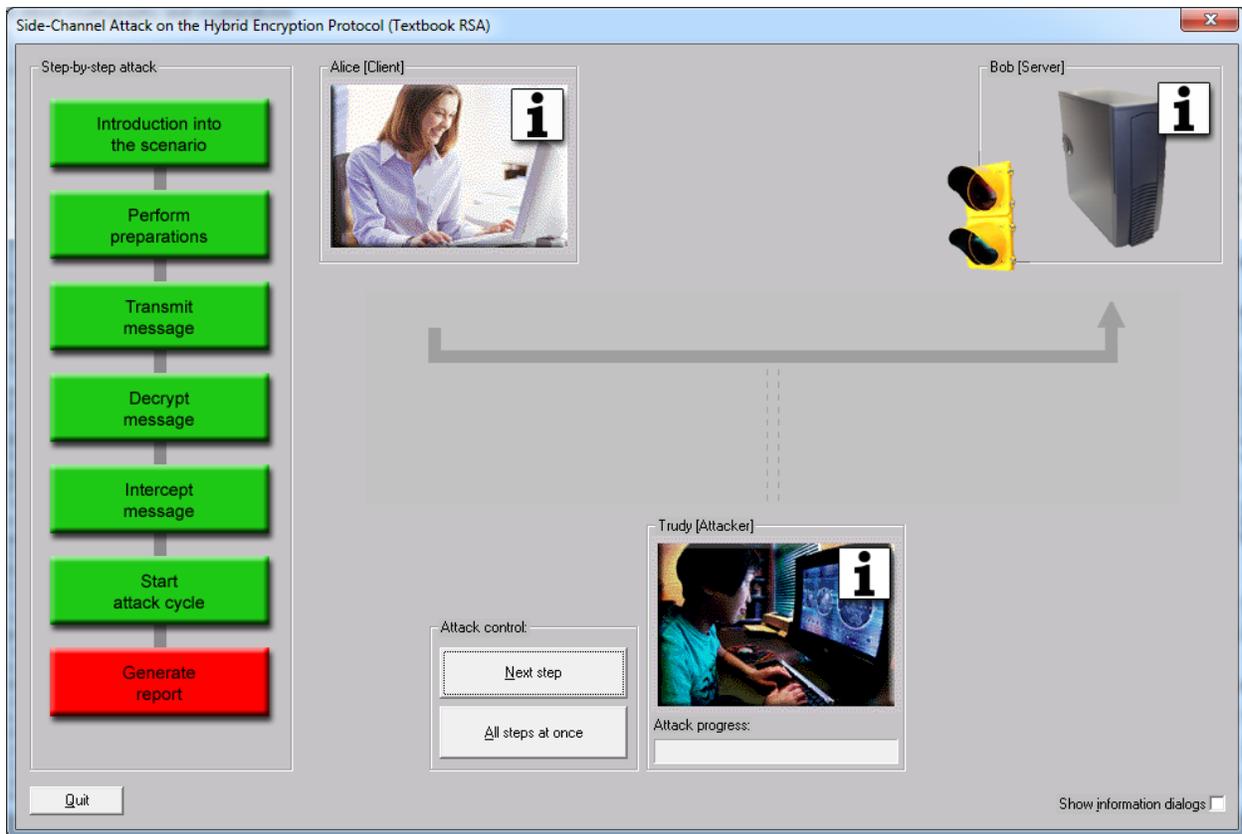
9. Click “Transmit message” and “Decrypt message”.



10. Enter **1234** and click “OK”.



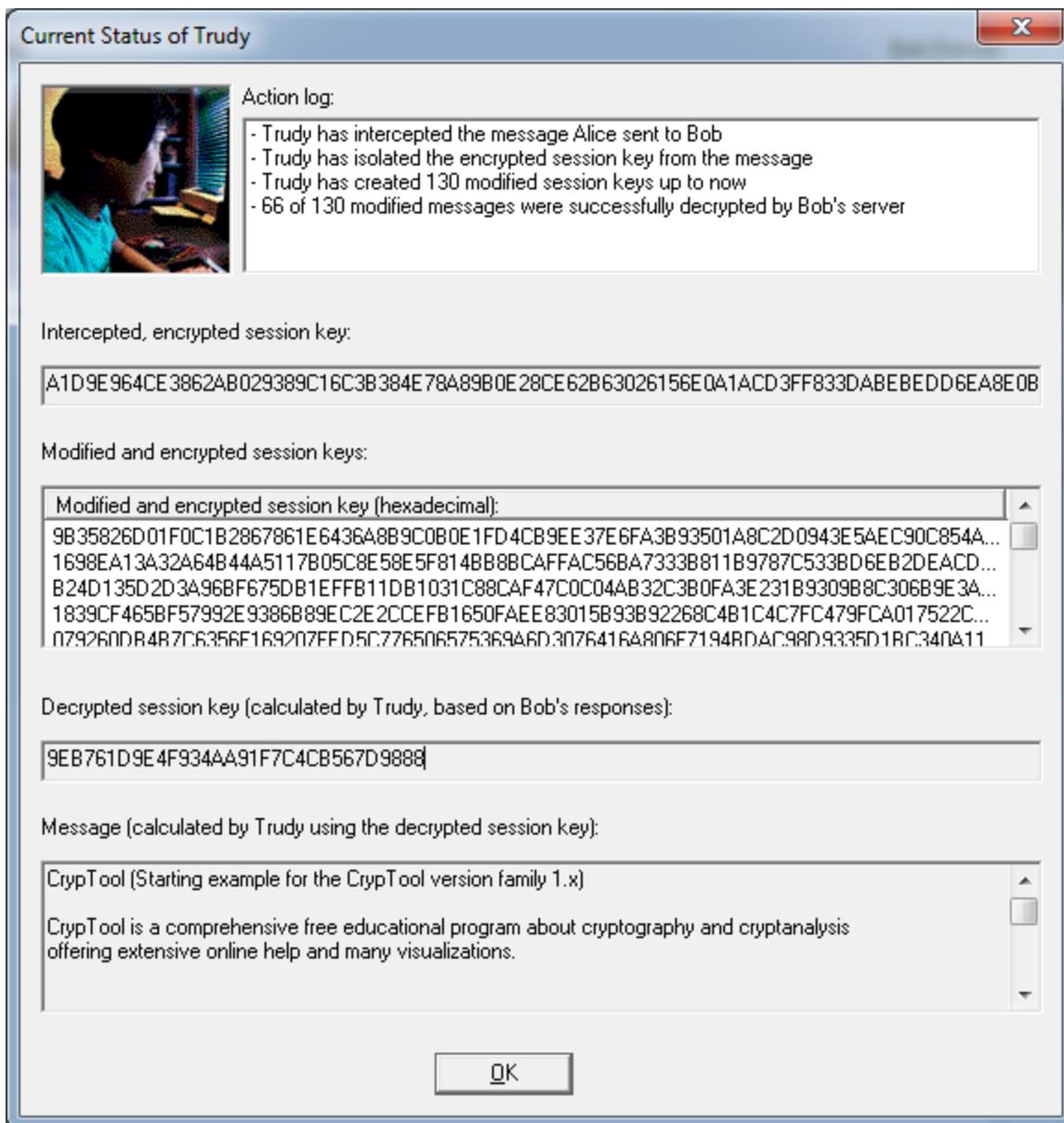
11. Click “Intercept message” and “Start attack cycle”.



12. Click “All steps at once” button.



13. Click “OK” and icon of **Trudy (Attacker)**.



The session key is 9EB761D9E4F934AA91F7C4CB567D9888 which matches the one generated in Step 5.