Part 1 (Required)

Implementation and Analysis of RSA in CrypTool 1

INSTALLATION

Install CrypTool 1 (version: 1.4.31 Beta 06 - English) by going to https://www.cryptool.org/en/ct1-download-en

Get familiar with the options of the program and its On-Line Help.

1. RSA DEMONSTRATION

Please go through all steps of the RSA Demonstration (available under Indiv. Procedures => RSA Cryptosystem => RSA Demonstration…).

Perform this procedure for at least the following sizes of the RSA keys (understood as the size of N in bits):

a) 16 bits (CrypTool default)
b) 256 bits
c) 2048 bits.

Assume the equal sizes of P and Q.

For each case, record values of

A) all components of a public key
B) all components of a private key
C) message
D) ciphertext.

Try to encrypt 1, 10, N-1, and N-10 and see if the results match your expectations. Record and discuss your findings.

Hint: In order to do that you will need to set your Input as numbers (as opposed to text) Please see the corresponding setting of CrypTool in the RSA Demonstration window.
2. FACTORING

Using the beginning of the RSA Demonstration generate values of N for at least the following initial sizes:

a) 50 bits  
b) 100 bits  
c) 150 bits.

For each size of N, double click on N, and then copy the entire value of N to the Input field in the window Factorization of a Number, obtained by choosing Indiv. Procedures => RSA Cryptosystem => Factorization of a Number…

For each case, include in the report:

A. value of N  
B. values of P and Q obtained after factoring N  
C. sizes of N, P, and Q in decimal digits  
D. factoring time  
E. method used for factoring (listed after clicking on “Details”, and then choosing “Save list into main window”)

Find experimentally the size of N for which the factoring time is consistently greater than

A. one minute (required)  
B. five minutes (bonus).

Then, generate the same size number(s) randomly (e.g., by typing arbitrary digits until the required size is reached). Factor these random numbers and record the same information as in case of N obtained using RSA Key Generation.

Hint: Please keep clicking on “Continue” until the number is fully factored.

Have you noticed any changes in the execution time or method(s) used?

3. GENERATION AND CERTIFICATION OF RSA KEYS

Generate a pair of RSA keys, with the secure size of the key, 2048 bits, using option Digital Signatures/PKI => Generate/Import Keys…

Provide user data required for the generation of a certificate and protecting access to the private key.

Press on “Show key pair”, and then on “Show public parameters…” and “Show certificate”.

Record all generated data. Explain the meaning of the abbreviation PSE used in CrypTool (Hint: Use Help Index Search).
4. SIGNATURE DEMONSTRATION

Open an arbitrary text document you would like to sign.

Choose option
**Indiv. Procedures => RSA Cryptosystem => Signature Demonstration (Signature Generation)…**

Follow the demonstration by clicking on active icons.

Use the “Provide certificate” icon to import the RSA certificate and the key.

Record all generated data.

Then, verify certificate using **Digital Signatures/PKI => Verify Signature**

5. DECRIPTION USING CHINESE REMAINDER THEOREM

Using the Big Integer Calculator, available at

http://www.javascripter.net/math/calculators/100digitbigintcalculator.htm

or any other web-based calculator capable of performing operations on large integers, demonstrate RSA decryption using Chinese Remainder Theorem, for the case of a 256-bit RSA key pair generated using CrypTool 1 as a part of Task 1, and the ciphertext obtained using the public key of this pair for the message M=10.

Record the results of all intermediate calculations you performed to execute this decryption.

**Hint:** The second part of the CRT calculations, devoted to the calculation of M given M_P and M_Q, can be performed using the following function of CrypTool:

**Indiv. Procedures => Chinese Remainder Theorem Applications => Planetary Motion and the Chinese Remainder Theorem**

6. MESSAGES NOT CONCEALED BY RSA

A. For your earlier choice of a 16-bit RSA key, used in Task 1: RSA Demonstration, calculate values of 9 messages that are not concealed by RSA. Verify this special property of these inputs using RSA Demonstration. Document your findings.

B. Generate an RSA key that does not conceal any message. Verify the properties of this key using RSA Demonstration. Document your findings.
Part 2 (Bonus)

Implementation and Analysis of RSA in CrypTool 2

INSTALLATION

Install CrypTool 2 (version 2.0, Stable Build 6222.1) by going to https://www.cryptool.org/en/ct2-download-en
Get familiar with the options of the program and its On-Line Help.

RSA DEMONSTRATION

Using visual programming available in CrypTool 2, prepare a demonstration of the operation of a hybrid system based on the use of RSA and AES.

AES should be used for the secret-key encryption of messages, and RSA for the exchange of AES session keys.

The demonstration should visualize all major operations performed on the sender’s side and the receiver’s side, and should allow exchange of medium size messages in English.

The users are assumed to know each other’s public keys.

Do your best to support various security levels recommended by NIST.

As a part of your solution, please submit your CrypTool 2 project in an electronic form, and write a short report including screenshots illustrating your project operation on the sender’s side and on the receiver’s side.

RSA BREAKING

Investigate components available in CrypTool 2 for breaking RSA using factorization. Determine the sizes of keys that can be broken using these tools within a predetermined time limit, e.g., 5 minutes. Document your findings and submit your report, as well as an electronic version of your CrypTool 2 project.

Part 3 (Bonus)

Breaking Historical Ciphers using CrypTool 1 or CrypTool 2

Solve the following problem using capabilities of CrypTool 1 or CrypTool 2:

Below please find 6 ciphertexts of different messages encrypted using the following 6 classical ciphers: Caesar (shift cipher with k=3), Vigenere, Hill (with d=3), Affine, Playfair, and Permutation. Do your best to match ciphertexts with a cipher that could
have been used to obtain the given ciphertext. If you are uncertain, you can list several ciphers per each ciphertext.

Break at least 2 out of 6 ciphers. You will obtain extra points for any additional cipher broken using ciphertext-only attack. All attacks must be documented. Brute-force attacks do not count.

Please note that spaces and punctuation characters have been removed before encryption.

The ciphertext has been divided into blocks of the size of 5 letters.

**Ciphertext 1**

XINKR NCWEI RRKER EGBIF AIHRM MFEOH JIEQP RRGEO FLFYS KEXGV OYPRI MTEFB IXHKS BMMEZ UAZSN GNQVE SZEEE IUXOC VMFEQ HJIEQ LVSI ORR Ker TOOAT SMKHR IITUT UZWGG TRDVG URORG HKVBG INZHV JKAZE ZIKO BXNTLE BMEXS RHAPY IVZEN IHOCN GYMNC AY CVR

**Ciphertext 2**

WRIJC IXPCI QZRIZ QYHAE PLWZZ LCSRW LXPCV OWLIA VCXHY LWJCW VIVIZ WAVOR CPCZR CYOWL LVAZN CBEXM CXNYZ RCSAL APAHZ RCWPU GWVNE ZNYZR CSAXZ CVZAH ZRCWP SRIPI SZCPQ LG

**Ciphertext 3**

EDOOU RNLI O TOSH M OAST REBEH AFIDA ODREE BSOEG ROTMT EOTEP IRTBE RUBEM AOSET BSMPP GNOUC REEDS NRNCR YOUL A ANETO RCNYI

**Ciphertext 4**

BRXPX VWEHW HKFD QJHBR XZDQW WRVHH LQW KH ZRUOGPJ

**Ciphertext 5**

TNFRC KZLVI HWYRD DAXAL LWZRW LVBJM XZYBV EBLWX IRGYH BNQGM IZLVJ BCGKP TYDRS XHHAQ QEXDO XZUFF AZPKW VIEZ J BHIU EO

**Ciphertext 6**

KBMPT QDKZL QZKPL MFCNK GLGEX HUKYF FMPIT QDKZL QZKPL MEONX FKKUP MKPAM