سارا برادران
9624193

Question 1)

Fermats little theorem

Let p be a prime.Then prove for every positive integer a:

$$a^p \equiv a \pmod p$$

Proof: we can use induction

Base: a = 1 → for all p which p is a prime number; $1^p \equiv 1 \pmod p$

Assumption: a = n → for all p which p is a prime number; $n^p \equiv n \pmod p$
→ $n^p + 1 \equiv n + 1 \pmod p$

Sentence: a = n + 1 → for all p which p is a prime number? $(n+1)^p \equiv n + 1 \pmod p$

We know $(x + y)^p \equiv x^p + y^p$:

$(n + 1)^p \equiv n^p + 1^p \pmod p \equiv n^p + 1 \pmod p \equiv n + 1 \pmod p$ →? $(n+1)^p \equiv n + 1 \pmod p$

Question 2)

Chinese remainder theorem

Let m_1 and m_2 be two positive integers that are relatively prime. Given any two integers a and b, there exists an integers x such that:

$$x \equiv a \pmod{m_1}$$
$$x \equiv b \pmod{m_2}$$

Prove any two solutions of these equations are congruent to each other modulo m_1m_2.

Proof:

$x \equiv a \pmod{m1}$ → $x = m_1q + a$
$x \equiv b \pmod{m2}$ → $x = m_2q' + b$

$m_1q + a = m_2q' + b$ → $m_1q - m_2q' = b - a$

Can we find such q and q' for all a, b, m_1, and m_2 where $(m_1, m_2) = 1$? Bezout's identity tells us 1is a Z-linear combination of m_1 and m_2, and therefore every integer is a Z-linear combination of m_1 and m_2.
Therefore integers q and q' satisfying $m_1q - m_2q' = b - a$ exist .
Uniqueness of Solution: If x=c and x=c′ both satisfy

$$x \equiv a \pmod{m1}$$
$$x \equiv b \pmod{m2}$$

Then we have $c \equiv c'$ mod m_1 and $c \equiv c'$ mod m_2. Then $m_1 \mid (c-c')$ and $m_2 \mid (c-c')$.
Since $(m_1, m_2) = 1$, the product $m_1 m_2$ divides $c-c'$, which means $c \equiv c'$ mod $m_1 m_2$.
This shows all solutions to the initial pair of congruences are the same modulo $m_1 m_2$

Question 3)

Diffie-Hellman Key Exchange

In the DHKE protocol, the private keys are chosen from the set $\{2,\ldots, p-2\}$. Why are the values 1 and $p-1$ are not considered?

If (prA or a) = 1 → A = (α^{a}) mod p = α → $K_{AB} = \alpha^{b}$ mod p = B
If (prB or b) = 1 → B = (α^{b}) mod p = α → $K_{AB} = \alpha^{a}$ mod p = A

Both A and B are not hidden and available to everyone → these keys are not secure

If (prA or a) = p - 1 → A = (α^{p-1}) mod p → $K_{AB} = (\alpha^{p-1})^{b}$ mod p = α^{pb-b} mod p = $(\alpha^{b})^{p} * (\alpha^{b})^{-1}$ mod p = $B^{p} * B^{-1}$ mod p = B^{p-1} mod p = 1 (Fermats little theorem)

If (prB or b) = p - 1 → B = (α^{p-1}) mod p → $K_{AB} = (\alpha^{p-1})^{a}$ mod p = α^{pa-a} mod p = $(\alpha^{a})^{p} * (\alpha^{a})^{-1}$ mod p = $A^{p} * A^{-1}$ mod p = A^{p-1} mod p = 1 (Fermats little theorem)
Common Key = 1 → it is Guessable and not secure

Question 4)

4.1. Compute the two public keys and the common key for the DHKE scheme with the parameters p=467,=2 ,a=228,b=57

A = α^{a} mod 467 = 2^{228} mod 467 = $(2^{10})^{22} * 2^{8}$ mod 467 = $(90)^{22} * 2^{8} = (161)^{11} * 2^{8} = (236)^{5}$ * 161 * 256 = $(123)^{2}$ * 236 * 161 * 256 = 185 * 236 * 161 * 256 = 229 * 120 = 394

B = α^{b} mod 467 = 2^{57} mod 467 = $(2^{10})^{5} * 2^{7}$ mod 467 = $(90)^{5} * 2^{7} = (161)^{2} * 90 * 2^{7}$ = 236 * 90 * 128 = 236 * 312 = 313

$K_{AB} = \alpha^{ba} = 2^{228 * 57}$ mod 467 = 394^{57} mod 467 = 192^{28} * 394 mod 467 = 438^{14} * 394 mod 467 = 374^{7} * 394 = 243^{3} * 374 * 394 = 207 * 243 * 374 * 394 = 332 * 251 = 206

4.2. We now design another DHKE scheme with the same prime p=467 as in problem 4.1. this time, we use the element α=4. The element 4 has order 233 and generates a subgroup with 233 elements. Compute kAB for:

a = 400, b = 134 →

$kAB = 4^{400 * 134}$

4^{400} mod 467 = $(4^{10})^{40}$ mod 467 = 161^{40} mod 467 = $(161^2)^{20}$ = $(236^2)^{10}$ = $(123^2)^5$ = $(185^2)^2$ * 185 = 134^2 * 185 = 210 * 185 = 89

89^{134} mod 467 = $(89^2)^{67}$ = $(449^2)^{33}$ * 449 = $(324^2)^{16}$ * 324 * 449 = $(368^2)^8$ * 324 * 449 = $(461^2)^4$ * 324 * 449 = $(36^2)^2$ * 324 * 449 = 362^2 * 324 * 449 = 284 * 324 * 449 = 17 * 449 = 161

a = 167, b = 134 →

$kAB = 4^{167 * 134}$
4^{167} mod 467 = $(4^{10})^{16}$ * 4^7 mod 467 = 161^{16} * 39 mod 467 = $(161^2)^8$ * 39 = $(236^2)^4$ * 39 = $(123^2)^2$ * 39 = 185^2 * 39 = 134 * 39 = 89
89^{134} → Similar to the above calculations

4.3. Why are the session keys identical?

The element 4 has order 233: 4^{233} mod 467 = 1
4^{167} mod 467 = 89

4^{233} * 4^{167} mod 467 = 4^{400} mod 467 = 89 * 1 = 89 → 4^{167} mod 467 = 4^{400} mod 467 →

$(4^{167})^{134}$ mod 467= $(4^{400})^{134}$ mod 467

Question 5)

Explain Attack Man-in-the-middle to Diffie–Hellman Key Exchange.

Alice		Attacker		Bob
Knows p , α		Knows p , α		Knows p , α
Sends $A = \alpha^a$	→	receives A and Sends $C = \alpha^c$ creates $k_A = A^c = \alpha^{ca}$	→	receives C creates key = $C^b = \alpha^{cb}$
receives C creates key = $C^a = \alpha^{ca}$	←	receives B and sends C creates $k_B = B^c = \alpha^{cb}$	←	Sends $B = \alpha^b$
encrypts message with key and sends	→	receives message and decrypts with k_A then encrypts message with k_B and sends	→	receives message and decrypts with key then encrypts replay message with key and sends
receives replay message and decrypts with key and sends	←	receives replay message and decrypts with k_B then encrypts replay message with k_A and sends	←	
. . .		. . .		. . .

Question 6)

6.1. Find a primitive root module 11.

$K_{11}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$, $| K_{11}^* | = 10$

If a = 2

$2^{10} \bmod 11 = 1$
$2^9 \bmod 11 = 6$
$2^8 \bmod 11 = 3$
$2^7 \bmod 11 = 7$
$2^6 \bmod 11 = 9$

$2^5 \bmod 11 = 10$
$2^4 \bmod 11 = 5$
$2^3 \bmod 11 = 8$
$2^2 \bmod 11 = 4$
$2^1 \bmod 11 = 2$

➔ $K_{11}^* = \{2^i \bmod 11 \mid 1 \le i \le \varphi(11)\}$ → **2 is a primitive root module 11**

6.2. Find a primitive root modulo 11^2, modulo $2 .11^2$, and modulo 11^{100}

$\alpha \in Z_n^*$ is a generator of Z_n^* if and only if $\alpha^{\left(\frac{\varphi(n)}{p}\right)} \neq 1 (\bmod n)$ for each prime divisor p of $\varphi(n)$

$11^2 = 121 \rightarrow \varphi(121) = 121 * (1 - \frac{1}{11}) = 110 = 2 * 5 * 11$

$2^{\left(\frac{110}{2}\right)} = 2^{55} \bmod 121 = (2^{11})^5 \bmod 121 = 112^5 \bmod 121 = 81^2 * 112 \bmod 121 = 27 * 112 = 120$

$2^{\left(\frac{110}{5}\right)} = 2^{22} \bmod 121 = (2^{11})^2 \bmod 121 = 112^2 \bmod 121 = 81$

$2^{\left(\frac{110}{11}\right)} = 2^{10} \bmod 121 = 56$

→2 is a primitive root module 11^2

$2.11^2 = 121 \rightarrow \varphi(242) = 242 * (1 - \frac{1}{11}) (1 - \frac{1}{2}) = 110 = 2 * 5 * 11$

$2^{\left(\frac{110}{2}\right)} = 2^{55} \bmod 242 = (2^{11})^5 \bmod 242 = 112^5 \bmod 242 = 202^2 * 112 \bmod 242 = 148 * 112 \bmod 242 = 120$

$2^{\left(\frac{110}{5}\right)} = 2^{22} \bmod 242 = (2^{11})^2 \bmod 242 = 112^2 \bmod 242 = 202$

$2^{\left(\frac{110}{11}\right)} = 2^{10} \bmod 242 = 56$

→2 is a primitive root module 2.11^2

$11^{100} \rightarrow \varphi(11^{100}) = 11^{100} * (1 - \frac{1}{11}) = 11^{99} * 10 = 2 * 5 * 11^{99}$

If a is a primitive root of p (p is a prime number)

then a is a primitive root of p^k if $a^{p-1} \neq 1 (\bmod p^2)$

$2^{11-1} = 2^{10} \bmod 121 = 56$

→ 2 is a primitive root module 11^{100}

Question 7)

ElGamal Encryption System

If Bob uses ELGamal with p=44927,=7,d=22105 , find Bob's public key, encode the message m=10101, and then decode the associated ciphertext.

Bob calculates public_key and sends it to Alice :

$\beta = \alpha^d \bmod p = 7^{22105} \bmod 44927 = 40909$ → Bob_public_key(p , α , β) = (44927 , 7 , 40909)

Alice receives Bob's public key and select a random i and calculates k_E :

Random i ∈ {2, 3, … , p-2} , for example : i = 2 → $\alpha^i \bmod p = 7^2 \bmod 44927 = 49 = k_E$

Alice calculate k_m and encrypt message with k_m and sends (cipher , k_E) to Bob:

$k_m = \beta^k \bmod p = 40909^2 \bmod 44927 = 15531$
cipher = m * β mod p = 10101 * 15531 mod 44927 = 38474

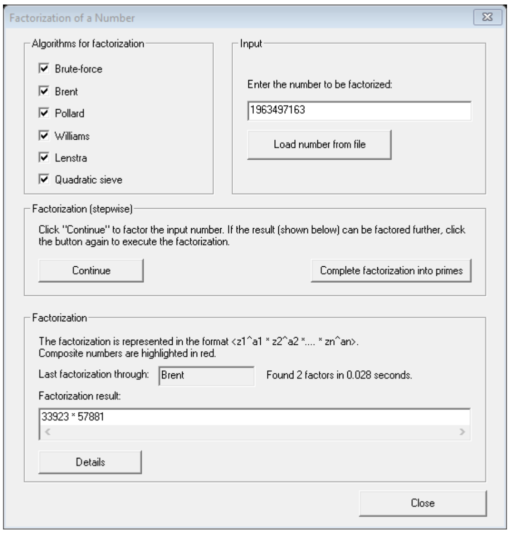
Bob receives (cipher , k_E) and calculates k_m and k_m^{-1} and decrypt cipher with k_m^{-1} :

$k_m = k_E^d \bmod p = 49^{22105} \bmod 44927 = 15531$
if p is a prime number $\rightarrow \varphi(p) = p - 1 \rightarrow \varphi(44927) = 44926$
$\rightarrow k_m^{-1} = k_m^{44926-1} \bmod p = 15531^{44925} \bmod 44927 = 13888$
plaintext = 38474 * k_m^{-1} mod p = 38474 * 13888 mod 44927 = 10101

CrypTool:

Question 1)

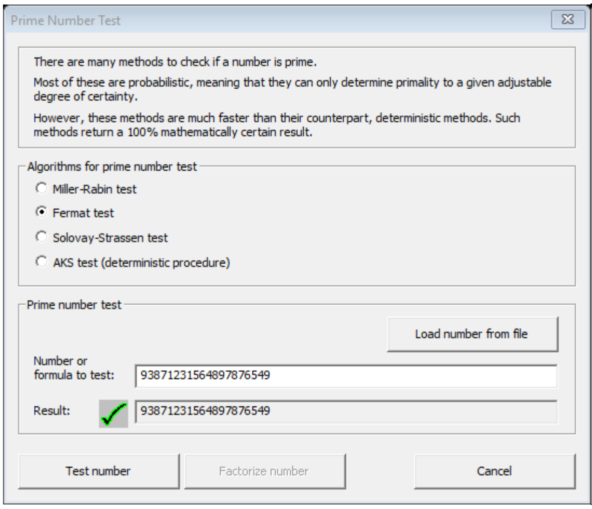
1963497163 → prime factors = 33923 , 57881

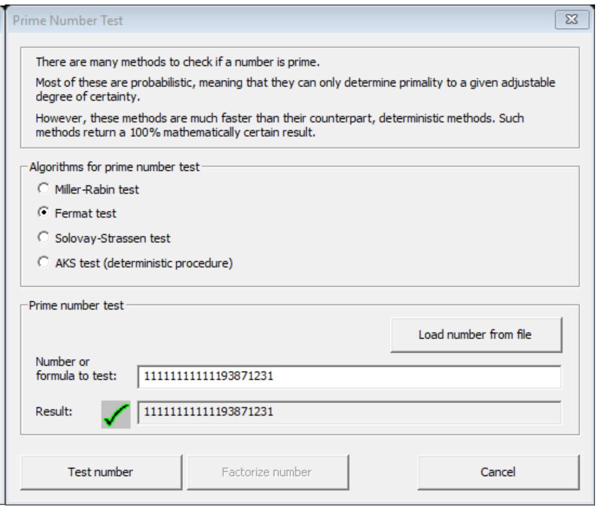


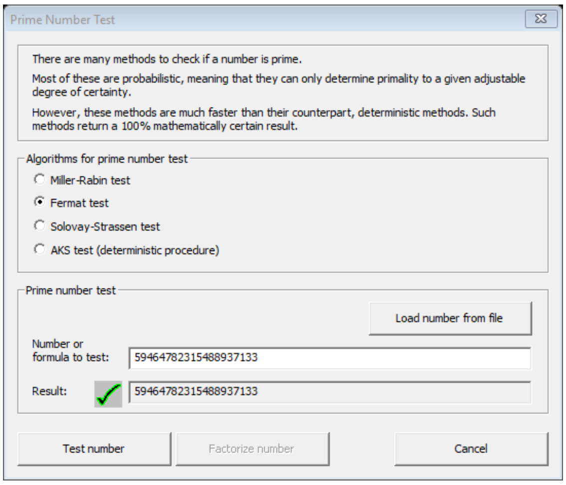
Question 2)

Three large prime numbers:
1111111111193871231
59464782315488937133
93871231564897876549

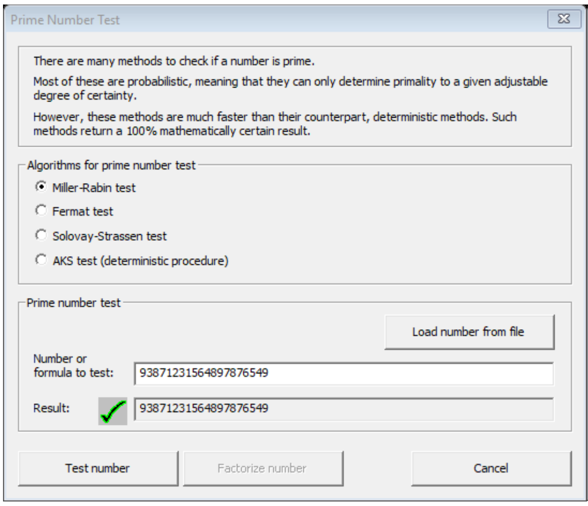
Fermat test

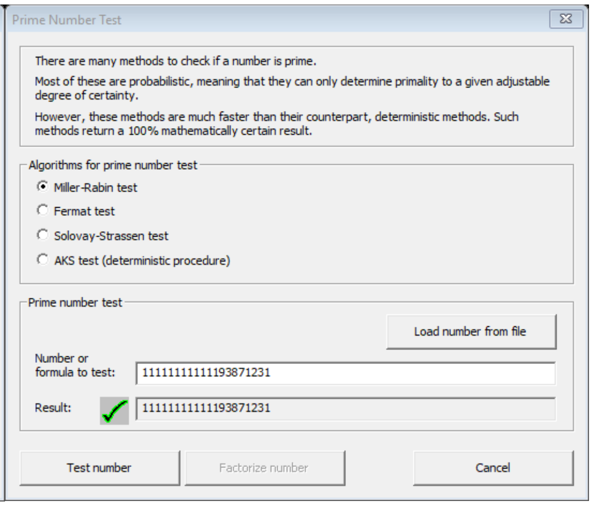


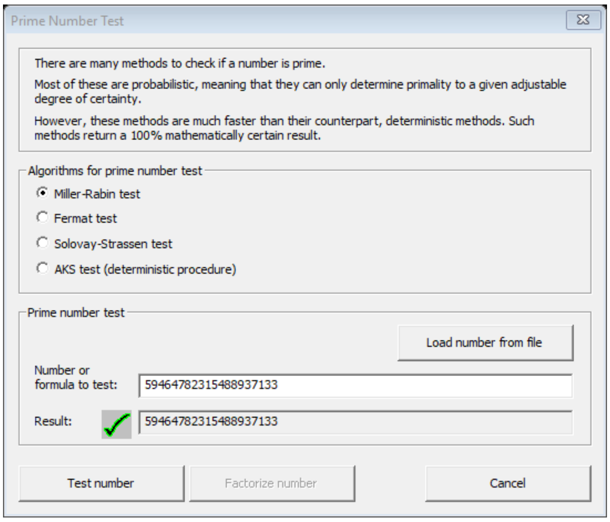




Miller-Rabin test



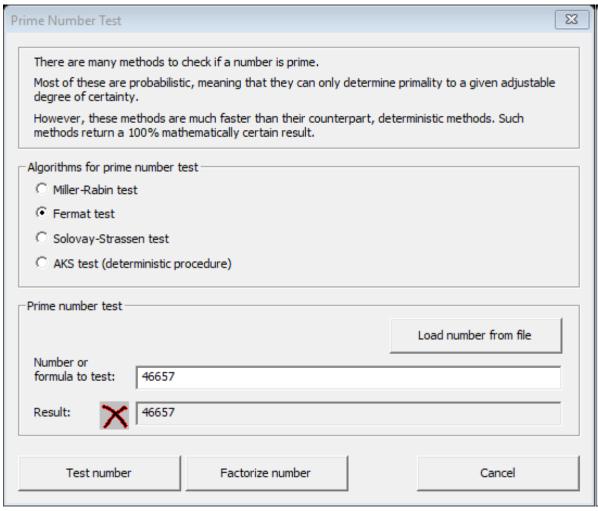


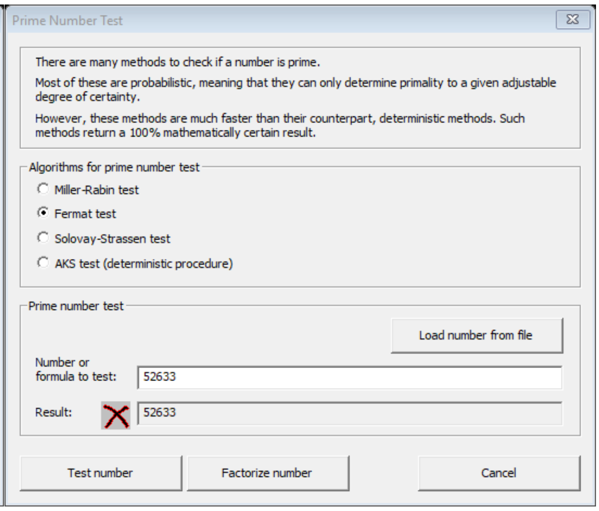


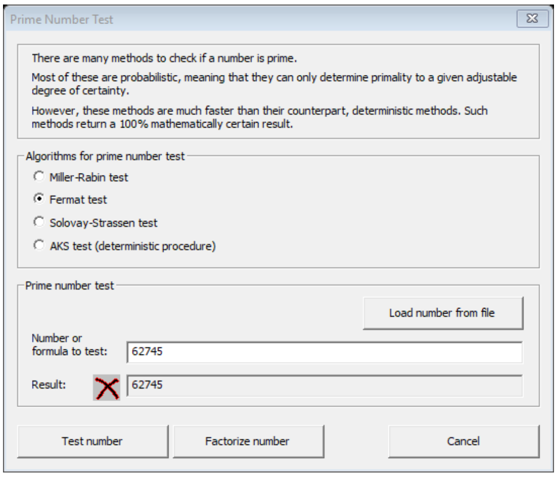
Three Carmichael numbers:

46657 , 52633 , 62745

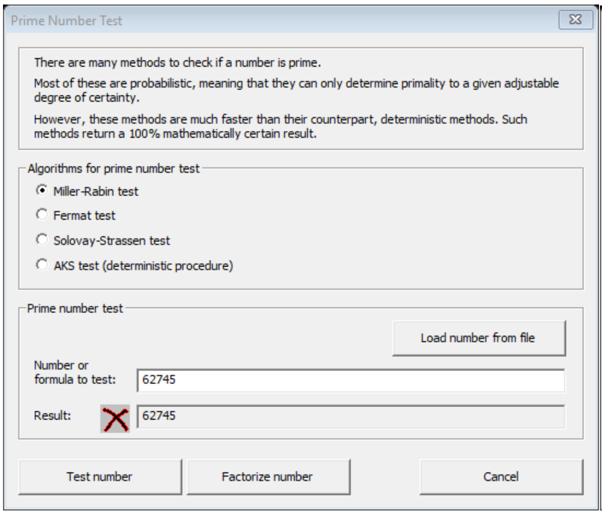
Fermat test

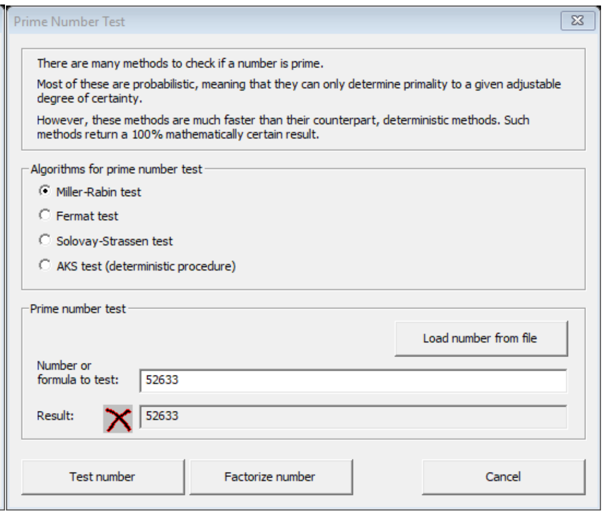


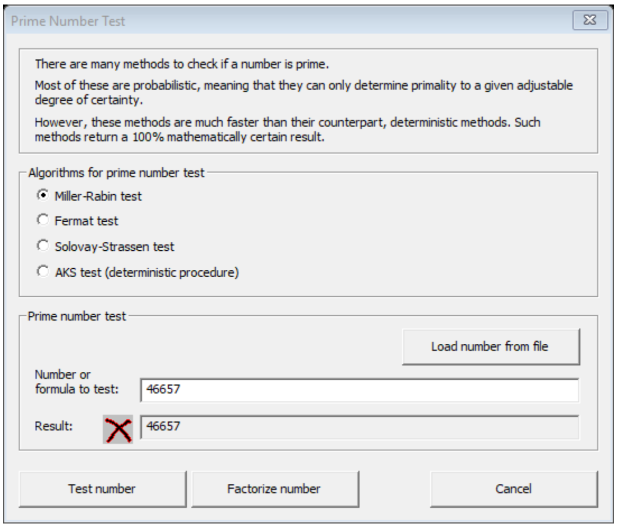




Miller-Rabin test



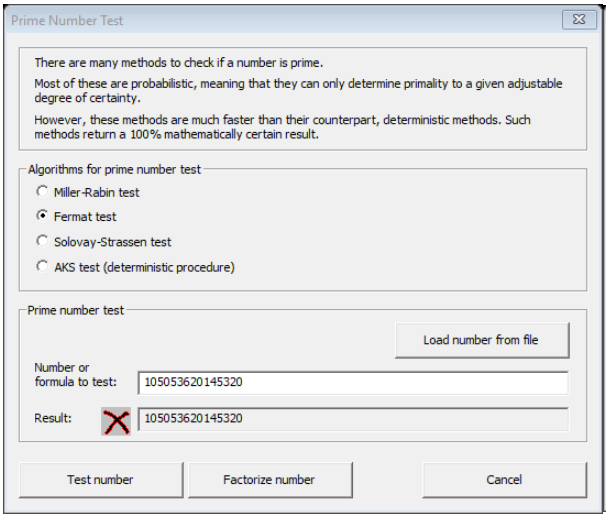


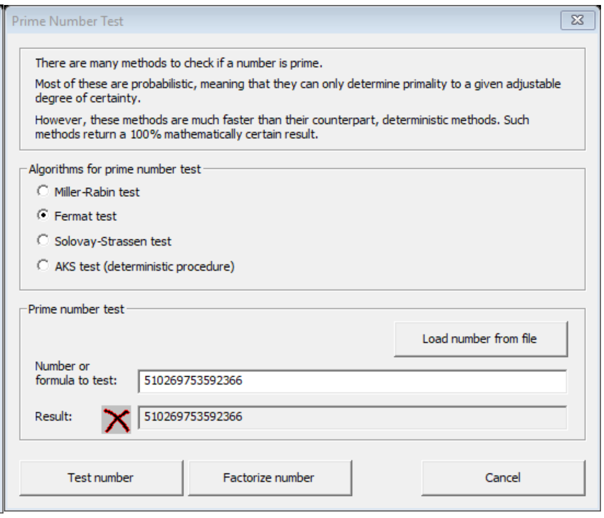


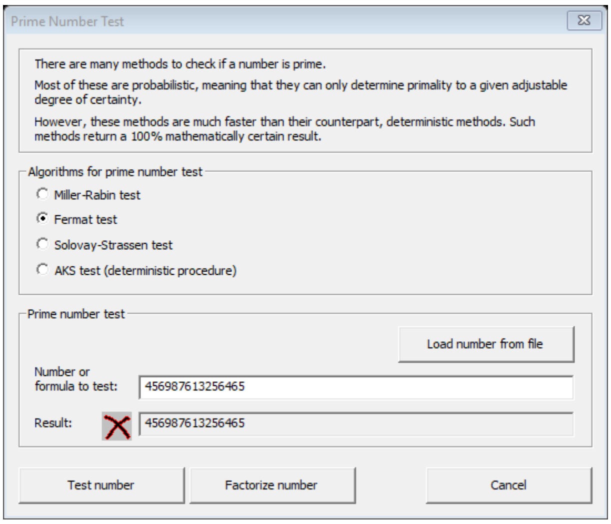
Three composite numbers:

105053620145320
510269753592366
456987613256465

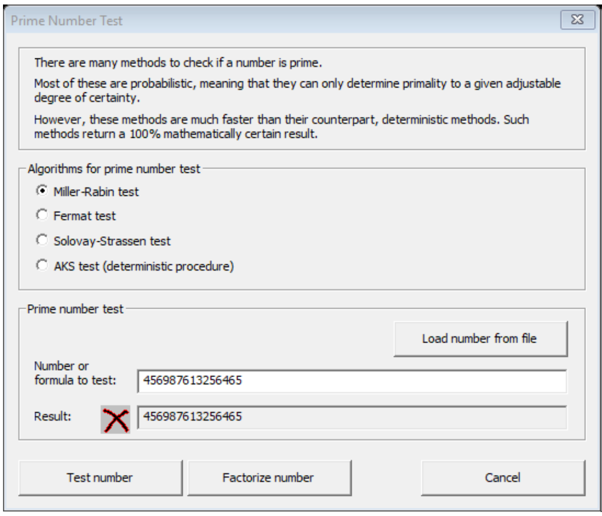
Fermat test

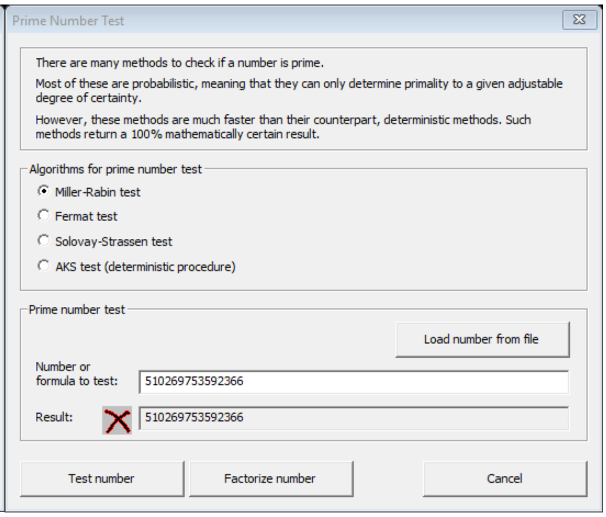


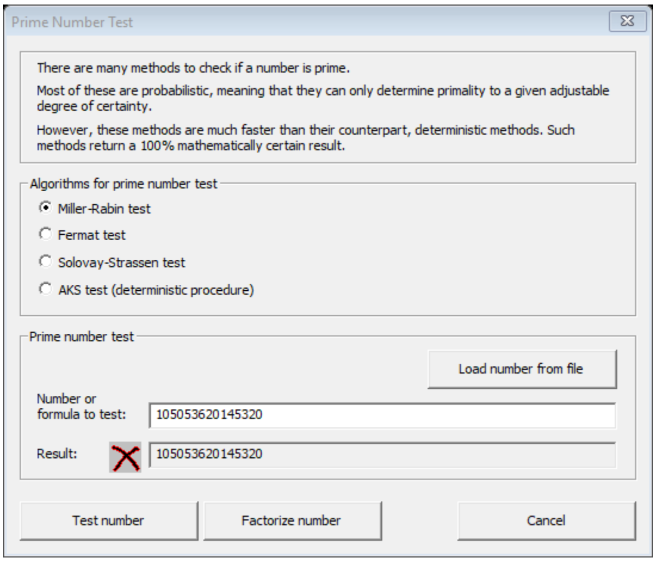




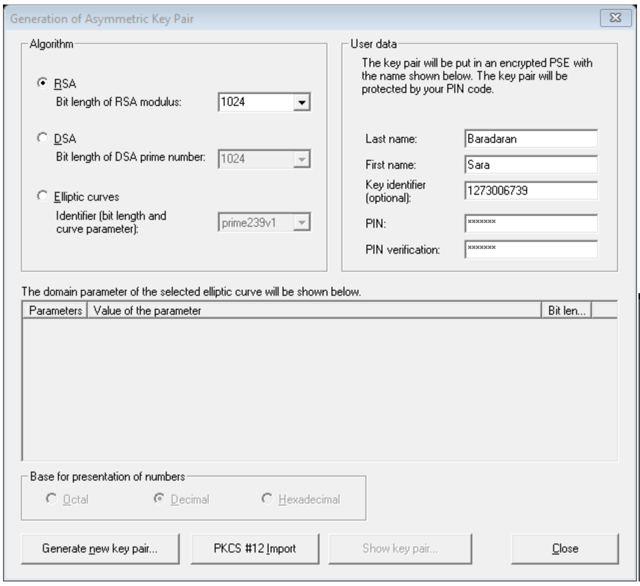
Miller-Rabin test

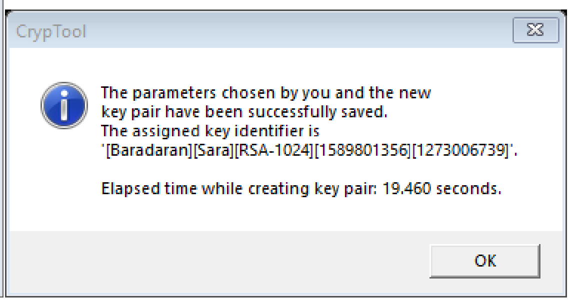






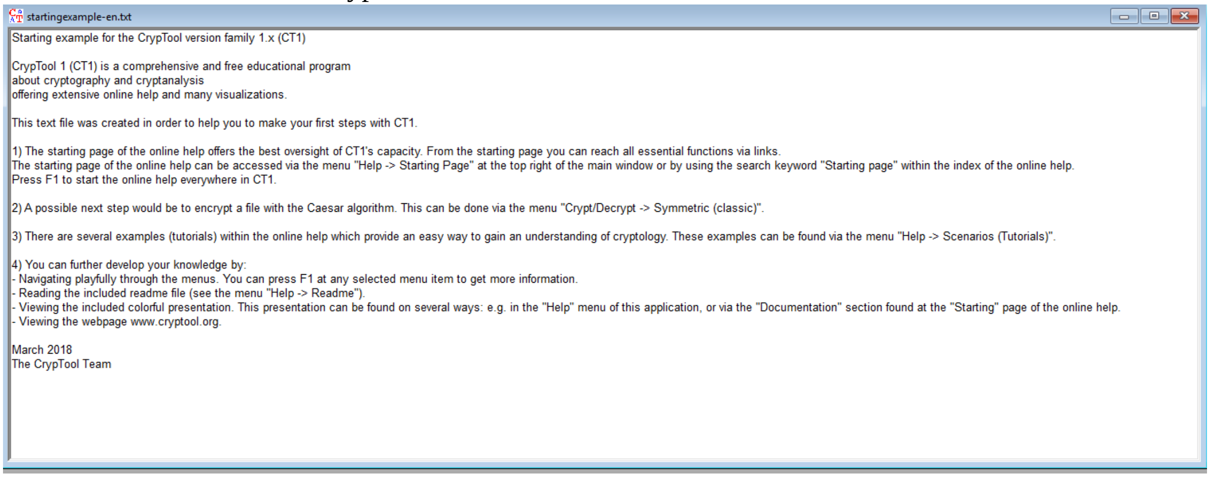
Question 3)





Question 4)

The below text has been encrypted :



startingexample-en.txt

Starting example for the CrypTool version family 1.x (CT1)

CrypTool 1 (CT1) is a comprehensive and free educational program
about cryptography and cryptanalysis
offering extensive online help and many visualizations.

This text file was created in order to help you to make your first steps with CT1.

1) The starting page of the online help offers the best oversight of CT1's capacity. From the starting page you can reach all essential functions via links.
The starting page of the online help can be accessed via the menu "Help -> Starting Page" at the top right of the main window or by using the search keyword "Starting page" within the index of the online help.
Press F1 to start the online help everywhere in CT1.

2) A possible next step would be to encrypt a file with the Caesar algorithm. This can be done via the menu "Crypt/Decrypt -> Symmetric (classic)".

3) There are several examples (tutorials) within the online help which provide an easy way to gain an understanding of cryptology. These examples can be found via the menu "Help -> Scenarios (Tutorials)".

4) You can further develop your knowledge by:
- Navigating playfully through the menus. You can press F1 at any selected menu item to get more information.
- Reading the included readme file (see the menu "Help -> Readme").
- Viewing the included colorful presentation. This presentation can be found on several ways: e.g. in the "Help" menu of this application, or via the "Documentation" section found at the "Starting" page of the online help.
- Viewing the webpage www.cryptool.org.

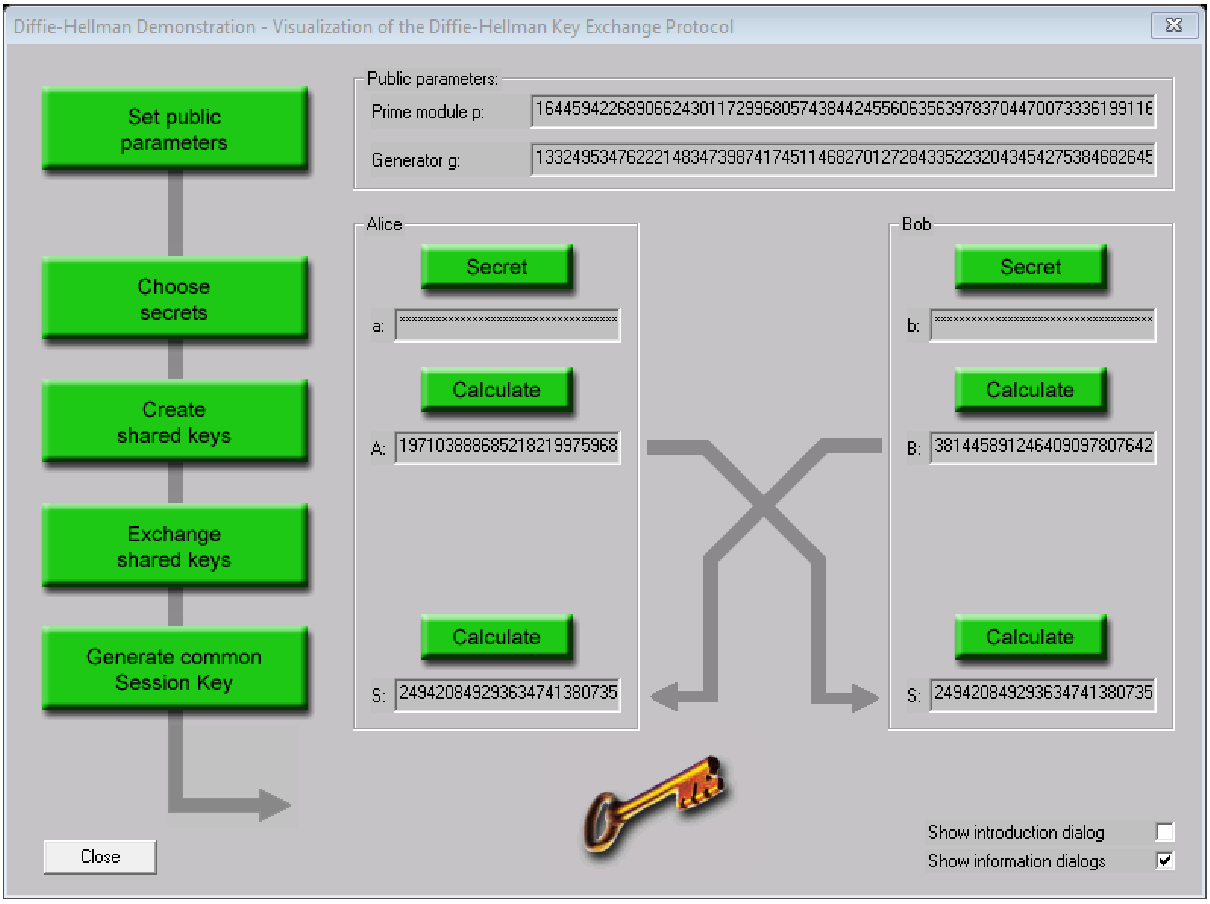
March 2018
The CrypTool Team

Cry-RSA-enc.hex : encrypted file

Cry-RSA-dec.hex : decrypted file

Question 5)

Diffie-Hellman



P:

164459422689066243011729968057438442455606356397837044700733361991165383163799

G:

133249534762221483473987417451146827012728433522320434542753846826453428574083

A public key:

19710388868521821997596828767080585947582043005941985175943283370322287047734

B public key:

3814458912464090978076429337600943099396550277082398129763588030204916946843

Session key:

24942084929363474138073590888863267595522440315900558039603861529856773827676

Openssl:

Question 1)

i. Generate a 2048bit RSA key, and save it in file named “private.key”.

```
→ ~ openssl genrsa -out private.key 2048
Generating RSA private key, 2048 bit long modulus (2 primes)
........+++++
..............+++++
e is 65537 (0x010001)
→ ~
```

Private.key file has been attached.

ii. Extract the public key out of the previously generated key and save it into a file named “public.key”.

```
→ ~ openssl rsa -in private.key -pubout -out public.key
writing RSA key
→ ~ cat public.key
-----BEGIN PUBLIC KEY-----
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAv5pzAMY39Av1WiZGuR0r
Nm46AOaI3AmUS8hh/49VuHJ6H6Ax7KvMe/lNkuIVnYcrgEaIH9ns4dvjT7jTLsI1
tVDrYSTcnRFmbOztX9l5WUgubzE6ZDFGshn2GKbK6to4wJ/exWT+oTNlU5xogqxf
RN4FDN141fzdzpXlyeCPPusyjhbpq+Nn3dbaxb8QVAPoTwy3MoB1UbLpZ+Bji9+G
grnt8wYNNH3a+4Sgog0pMSNftqXxUTAtvNsTx6J1rS7VuEEJPLsFaCtE/9ICIYaW
iPTgJz1MLzj+8886KUhC4HL38/sSIrnKSEoAvWVjb/WXj2vYrqU04EzLnp01lJy1
5QIDAQAB
-----END PUBLIC KEY-----
→ ~
```

Public.key file has been attached.

iii. Extract all parts that contribute to the construction of the respective private key, including prime factors, private and public exponents, and so on. Save the results in a file named “structure.txt”.

```
➜ ~ openssl rsa -text -in private.key > structure.txt
writing RSA key
➜ ~
```

Structure.txt file has been attached.

Question 2)

```
➜ ~ openssl s_client -connect www.feistyduck.com:443 -showcerts
```

cert.txt file has been attached.

In