Wrocław 01.10.2022

C Y B E R S E C U R I T Y

L A B 5

1. Introduction

The laboratory aims to become familiar with elements of cryptanalysis on symmetrical and asymmetrical algorithms. Below is a list of terms containing theoretical attack models and their practical application.

Brute Force attacks

This type of attack consists of checking every possible combination in a key to find the "key" to correctly decipher the message. Brute Force attacks can take less time for smaller keyspaces, but for larger key spaces they take an extremely long time. Brute Force attacks are impractical for modern encryption algorithms.

Ciphertext only attacks

It is also called “known ciphertext attacks”. In a “cipher-only attack”, the attacker knows the ciphertext of various messages that have been encrypted using the same encryption algorithm. The attacker must find a key which can then be used to decrypt all messages. This is one of the easiest attacks to carry out because it is easy to intercept the ciphertext (by sniffing), but it is difficult to implement because knowledge of the encryption process is limited. The methods that can be used:

- Attack on Two-Time Pad,
- Frequency Analysis,

Chosen-Plaintext Attack

A “chosen-plaintext” attack is similar to a 'known plaintext' attack, but in this case, the attacker selects their texts to help them crack the key. A properly selected sequence of characters with a generated cipher can restore the key. The found key can be used to obtain information about the whole encryption process and understand how it is carried out. The attacker analyzes the system behavior and output ciphertext, based on any kind of input data.

Known-plaintext Attack

In a known-plaintext attack, the attacker knows some of them and encrypts them. Reverse engineering is then used to find the key. Attacks of this type were only used to break simple ciphers.

Differential cryptanalysis

The attacker observes changes in two texts up to an encrypted form. Based on the changes taking place, the attacker tries to find the key. This is a kind of 'chosen plaintext' attack because the attacker selects plain text to observe the changes. The attack was used to break symmetrical algorithms like DES, but over time, the subsequent algorithms have become resistant.

Linear cryptanalysis

An attacker carries out a known-plaintext attack on several messages that have been encrypted with the same key. This gives the attacker an insight into the probability of a particular key occurring. If more than one message is attacked, there is a higher probability of finding a specific "key".

Man-in-the-Middle Attack

The attacker joins the communication as a hidden intruder. With two keys, he intercepts communication between the two entities. Using the previously prepared set of keys, the attacker decrypts, intercepts the content, encrypts, and then sends it to the target entity. To minimize the occurrence of such an attack, strong authentication methods must be used. An example is the use of asymmetric algorithms or key exchange using the Diffie-Hellman protocol.

Frequency analysis

Frequency analysis is similar to the attack on a known ciphertext. The attack consists of analyzing letters and groups of letters for a specific language. Knowledge of a language is important because each language has its frequency of occurrence of given letters. Popular characters in the ciphertext are searched for and converted into popular characters in a given language. Substitution Ciphers group can be given this type of attack.

Replay attack

An intruder sends the same message to the victim as in the victim's communication impersonating one of the parties. This allows the intruder to be authenticated by one of the parties and treated as a trusted entity in the communication. The intruder may gain unauthorized access and may extract the desired information. This type of attack is similar to a Man-in-the-Middle attack.

2. Prerequisites

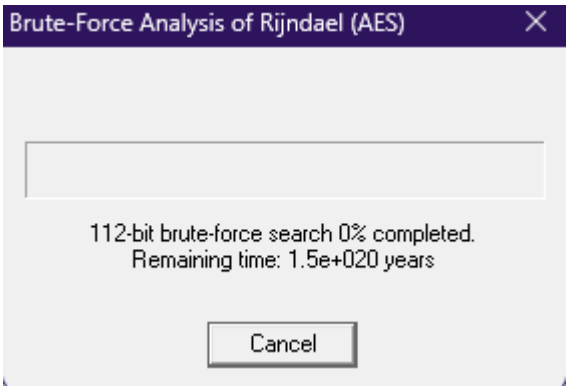
In this laboratory, the Cryptool tool is used to perform some process of cryptanalysis. The materials available at the lecture should be read. In Cryptool, options from the Analyses tab are used. Get acquainted with the cryptanalysis elements (types). **3. Cryptanalysis for the symmetric algorithm**

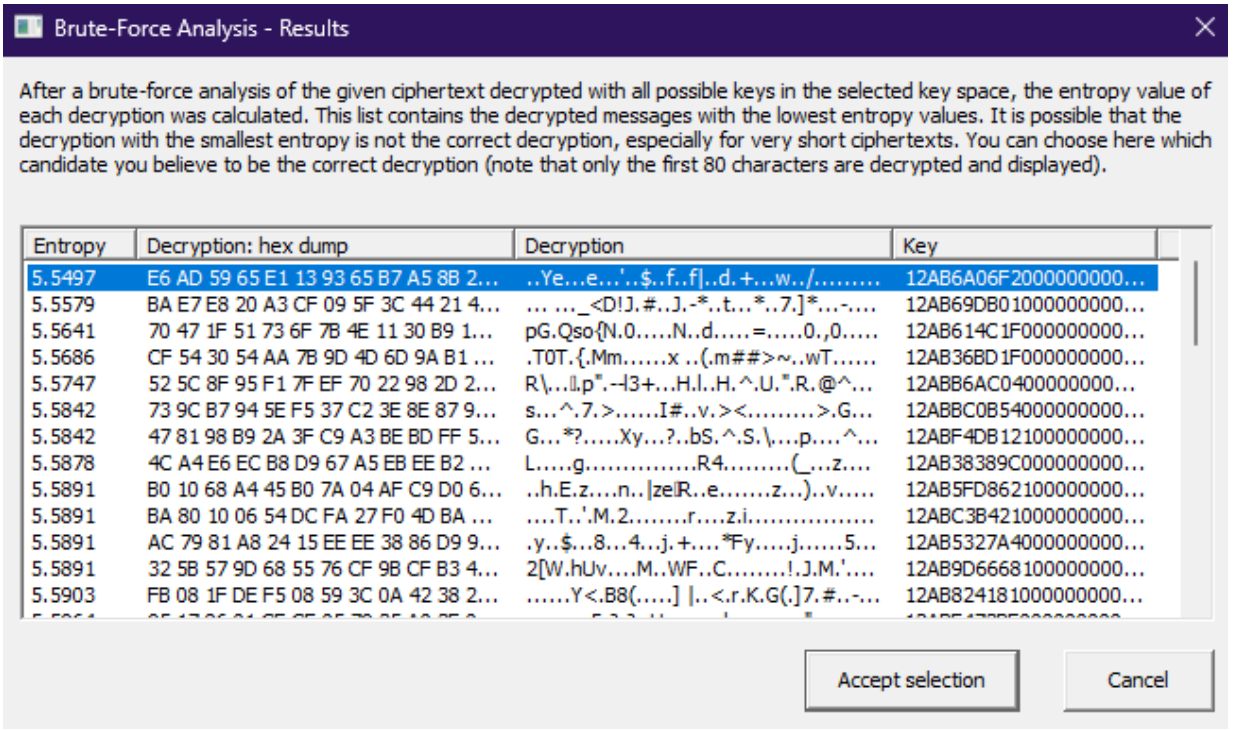
I. **Tasks:**

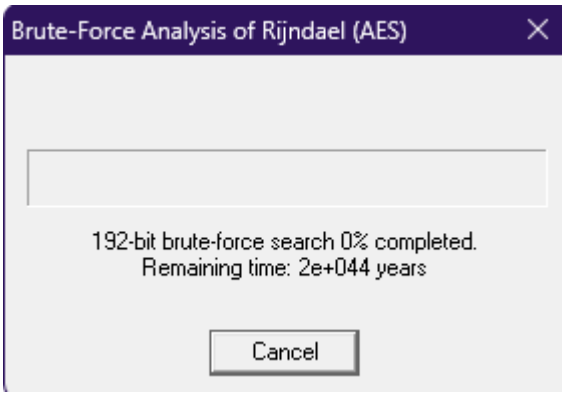
1. Evaluate the time needed to find a full key of 64, 128, 192, 256 bits.

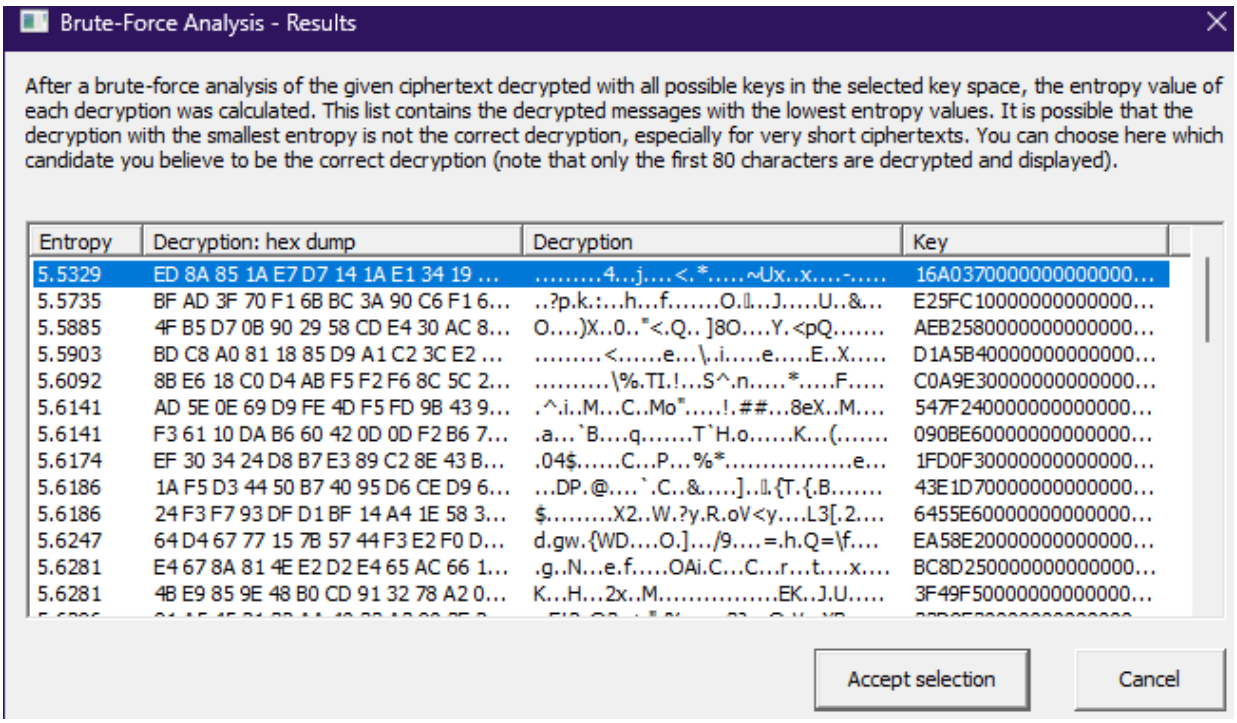
> For the text “This is a text to check brute-force analysis of different key lengths for AES” , Key: AB12

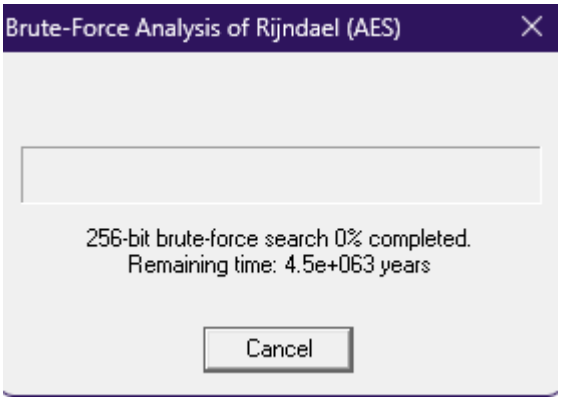
I chose modern (AES) so only three (128,192, 256) were available.

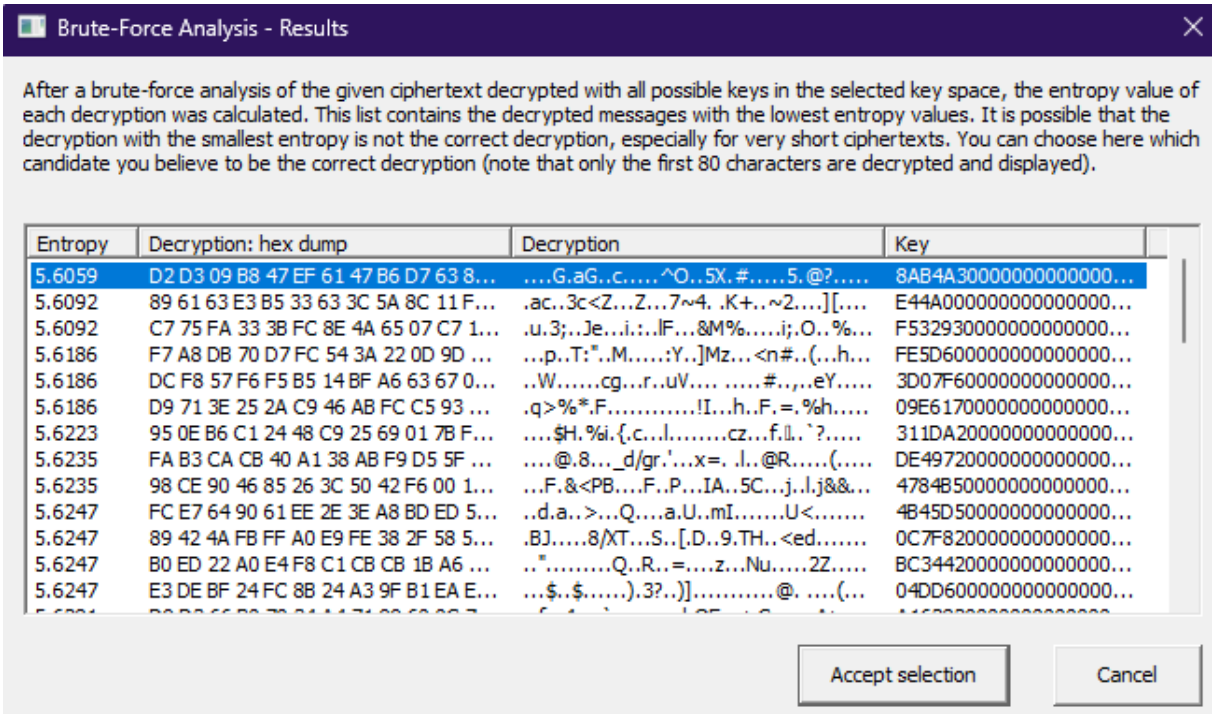



Brute-Force Analysis - Results
After a brute-force analysis of the given ciphertext decrypted with all possible keys in the selected key space, the entropy value of each decryption was calculated. This list contains the decrypted messages with the lowest entropy values. It is possible that the decryption with the smallest entropy is not the correct decryption, especially for very short ciphertexts. You can choose here which candidate you believe to be the correct decryption (note that only the first 80 characters are decrypted and displayed).
Entropy
Decryption: hex dump
Decryption
Key
5.5497 E6 AD 59 65 E1 13 93 65 B7 A5 8B 2... ..Ye...e...'..$..f..f|..d.+...w../..... 12AB6A06F2000000000...
5.5579 BA E7 E8 20 A3 CF 09 5F 3C 44 21 4... ... ..._<D!J.#..J.-*..t...*..7.]*...-.... 12AB69DB01000000000...
5.5641 70 47 1F 51 73 6F 7B 4E 11 30 B9 1... pG.Qso{N.0.....N..d.....=.....0.,0..... 12AB614C1F000000000...
5.5686 CF 54 30 54 AA 7B 9D 4D 6D 9A B1 ... .T0T.{.Mm.....x ..(.m##>~..wT..... 12AB36BD1F000000000...
5.5747 52 5C 8F 95 F1 7F EF 70 22 98 2D 2... R\...ñ.p".-ï3+...H.l..H.^.U.".R.@^... 12ABB6AC0400000000...
5.5842 73 9C B7 94 5E F5 37 C2 3E 8E 87 9... s...^.7.>.....I#..v.><.....>.G... 12ABBC0B54000000000...
5.5842 47 81 98 B9 2A 3F C9 A3 BE BD FF 5... G...*?.....Xy...?..bS.^.S.\....p....^... 12ABF4DB12100000000...
5.5878 4C A4 E6 EC B8 D9 67 A5 EB EE B2 ... L.....g.....R4.....(_...z.... 12AB38389C000000000...
5.5891 B0 10 68 A4 45 B0 7A 04 AF C9 D0 6... ..h.E.z....n..|zeR..e.....z...)..v..... 12AB5FD862100000000...
5.5891 BA 80 10 06 54 DC FA 27 F0 4D BA ... .T..'.M.2.....r....z.i..... 12ABC3B421000000000...
5.5891 AC 79 81 A8 24 15 EE EE 38 86 D9 9... .y..$...8...4...j.+....*Fy.....j.....5... 12AB5327A4000000000...
5.5891 32 5B 57 9D 68 55 76 CF 9B CF B3 4... 2[W.hUv....M..WF..C.....!.J.M.'.... 12AB9D6668100000000...
5.5903 FB 08 1F DE F5 08 59 3C 0A 42 38 2... .....Y<.B8(.....] |..<.r.K.G(.]7.#..-... 12AB824181000000000...
Accept selection
Cancel


Brute-Force Analysis of Rijndael (AES)
192-bit brute-force search 0% completed.
Remaining time: 2e+044 years
Cancel

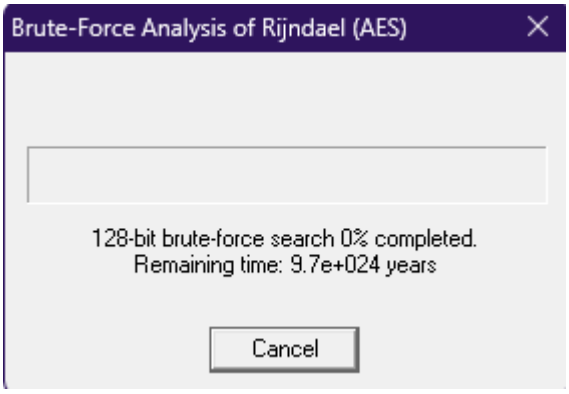

Brute-Force Analysis - Results
After a brute-force analysis of the given ciphertext decrypted with all possible keys in the selected key space, the entropy value of each decryption was calculated. This list contains the decrypted messages with the lowest entropy values. It is possible that the decryption with the smallest entropy is not the correct decryption, especially for very short ciphertexts. You can choose here which candidate you believe to be the correct decryption (note that only the first 80 characters are decrypted and displayed).
Entropy
Decryption: hex dump
Decryption
Key
5.5329 ED 8A 85 1A E7 D7 14 1A E1 34 19 ... .4...j....<.*.....~Ux..x....-..... 16A0370000000000000...
5.5735 BF AD 3F 70 F1 6B BC 3A 90 C6 F1 6... ..?p.k.:...h...f.....O.ñ...J.....U..&... E25FC10000000000000...
5.5885 4F B5 D7 0B 90 29 58 CD E4 30 AC 8... O....)X..0.."<.Q..]8O....Y.<pQ..... AEB2580000000000000...
5.5903 BD C8 A0 81 18 85 D9 A1 C2 3C E2 ... .<.....e...\..i.....e.....E..X..... D1A5B40000000000000...
5.6092 8B E6 18 C0 D4 AB F5 F2 F6 8C 5C 2... .....\%.TI.!...S^.n.....*.....F..... C0A9E30000000000000...
5.6141 AD 5E 0E 69 D9 FE 4D F5 FD 9B 43 9... .^.i..M...C..Mo".....!.##...8eX..M.... 547F240000000000000...
5.6141 F3 61 10 DA B6 60 42 0D 0D F2 B6 7... .a...`B....q.....T`H.o.....K...(..... 090BE60000000000000...
5.6174 EF 30 34 24 D8 B7 E3 89 C2 8E 43 B... .04$.....C...P...%*.....e... 1FD0F30000000000000...
5.6186 1A F5 D3 44 50 B7 40 95 D6 CE D9 6... ...DP.@....`.C..&.....]..ñ.{T.{.B..... 43E1D70000000000000...
5.6186 24 F3 F7 93 DF D1 BF 14 A4 1E 58 3... $.....X2..W.?y.R.oV<y....L3[.2.... 6455E60000000000000...
5.6247 64 D4 67 77 15 7B 57 44 F3 E2 F0 D... d.gw.{WD....O.].../9....=.h.Q=\f.... EA58E20000000000000...
5.6281 E4 67 8A 81 4E E2 D2 E4 65 AC 66 1... .g..N...e.f.....OAi.C...C...r...t....x.... BC8D250000000000000...
5.6281 4B E9 85 9E 48 B0 CD 91 32 78 A2 0... K...H...2x..M.....EK..J.U..... 3F49F50000000000000...
Accept selection
Cancel

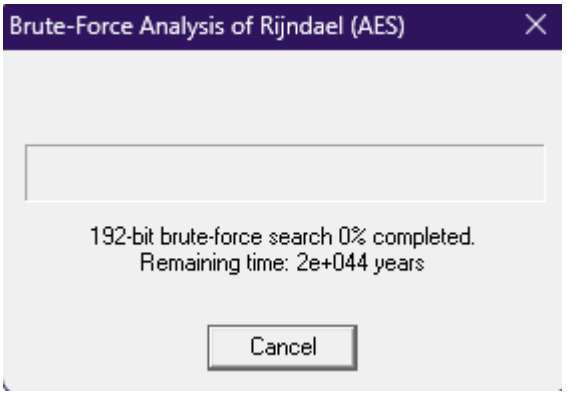


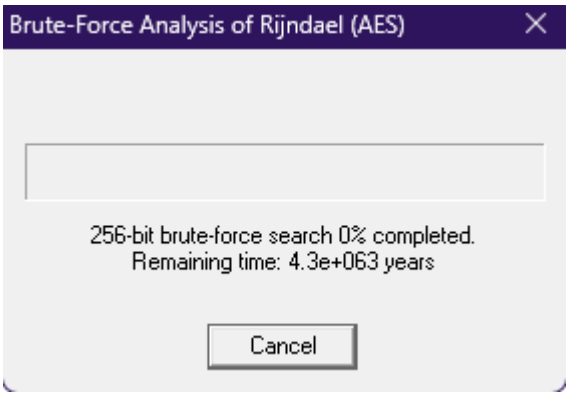


2. Compare the time it takes to find keys of the same length (e.g. 128 bits) for different algorithms.

>Here 128 bits with default keys,



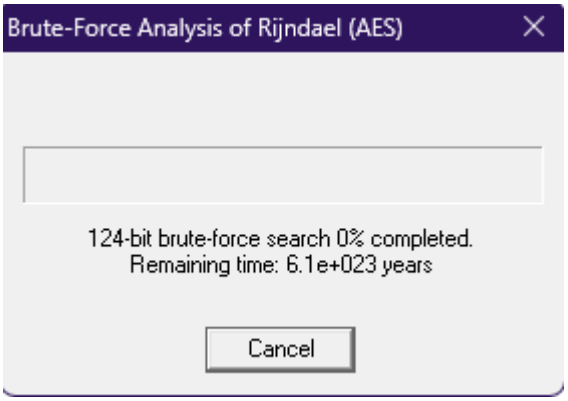


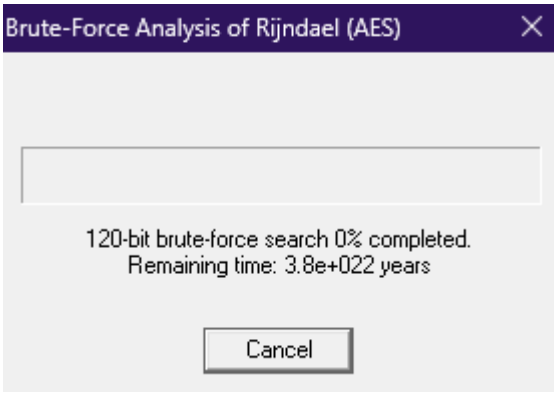


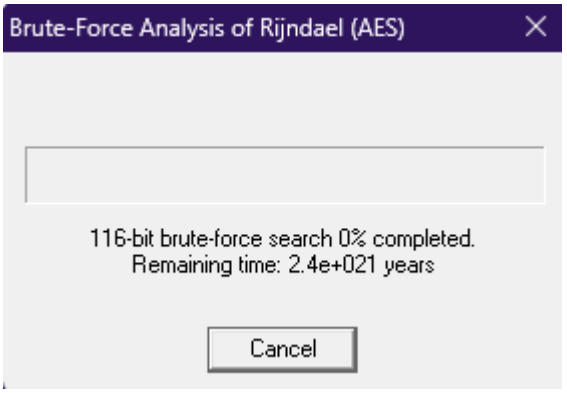
Complexity fluctuates for different bit lengths compared to 128 bits.

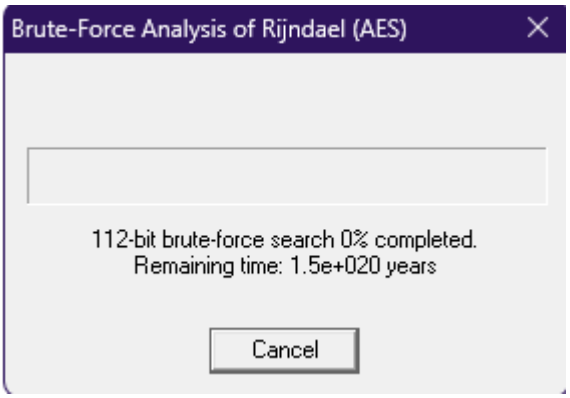
3. Determine the time needed to find a key if the unknown number of bits of the key is: 4, 8, 12, 16, 20, 24, 30, 34. (One star in the key equals 4 bits)

   >Changing 4 bits at a time with the keys ADFCB.. we see,





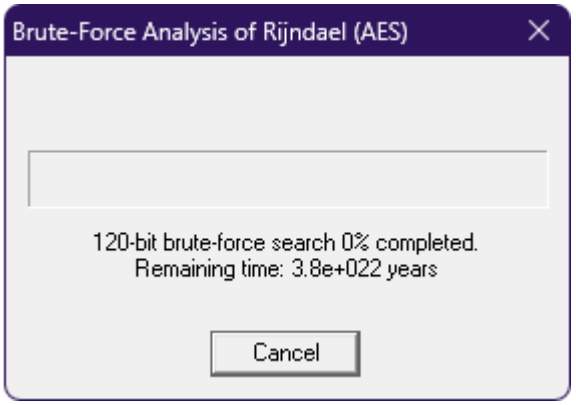


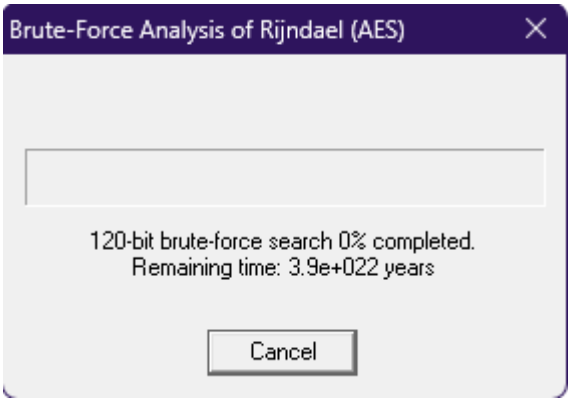


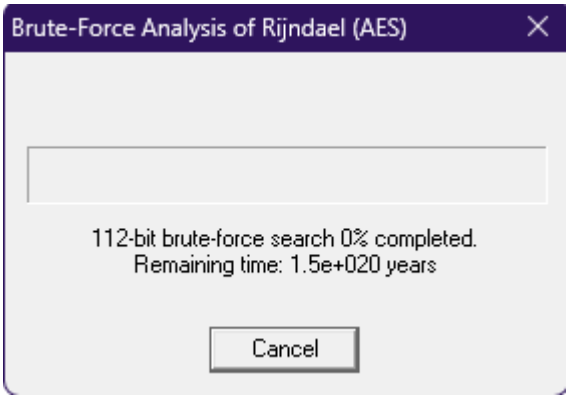
So the downward going time and bit complexity is visible.

4. Check whether the position of the unknown bits in the key affects the key search time.

   > Using positions randomly,







It seems like from previous comparisons changing bits does not change the time for the attack.

5. Evaluate the performance of the cryptanalysis tools.
> There was not an exact match. With enough time and trials we can at least assume the right direction with the tool.

6. Do we always get the correct key?
> I did not notice an exact match.

7. Does the number of the bits searched affect the quality of the key being restored?

> It does.

8. Does the position of unknown bits affect the quality of the key being restored?

> Comparison shows there are no significant changes.

II. **Questions:**

1. Can modern block algorithms be considered safe (referring to the above experiments)?
   > It is important to note that no cryptographic algorithm is completely immune to attack, and all algorithms have some level of vulnerability. Depending on design AES is very safe.

2. What length of the key offers us a sufficient level of security for particular algorithms? Why?
   > A complexity of 290+ is very secure. Starters, 256 is sufficient.

3. Does the size of the ciphertext influence the possibility of breaking it?

> The size of the ciphertext generally does not significantly influence the possibility of breaking the cipher.

4. Does the format and earlier processing of the document affect the possibility of its cryptanalysis (e.g. compression etc.)?
>Yes, for example, compressing the document can reduce the amount of data available to an attacker to analyze and thus make it more difficult to decipher the encrypted message. Additionally, the use of a different file format can make it more difficult for an attacker to recognize patterns in the data.

5. How many possible passwords can we check in a year of continuous work on one computer, which checks one million passwords per second ($^{20}2$)?
>A computer that can check one million passwords per second can check 6.87x10^13 passwords in a year.

6. What does this result say about the security of modern algorithms?
> Modern algorithms providing secure solutions. While no algorithm is 100% secure, modern algorithms are designed to reduce the potential for malicious actors to exploit weaknesses in the system. These results suggest that modern algorithms are successful in providing secure solutions and can be trusted to protect user data.

4. Cryptanalysis for the asymmetric algorithm

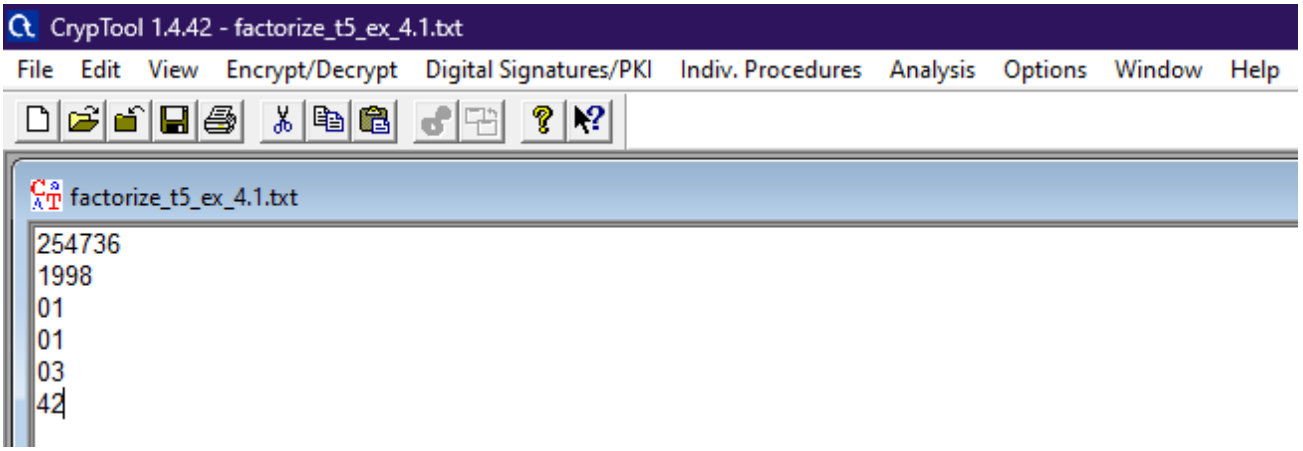
I. **Tasks:**

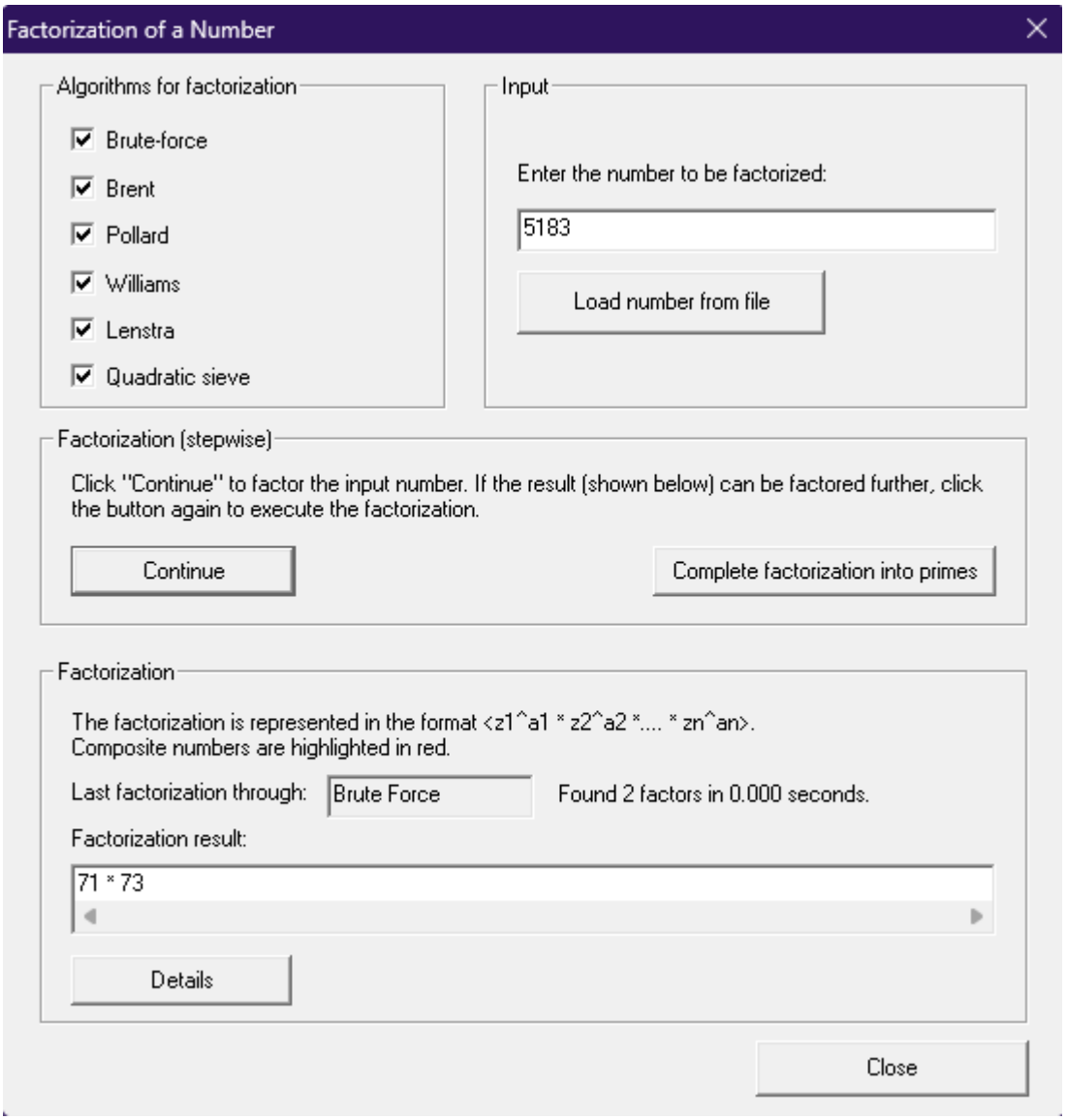
1. Please try to factorize the number created by the combination of
   - Student's ID number
   - Year
   - Month
   - Day
   - Hour
   - Minutes

   Example: 123456202011051408

Tab: Indiv. Procedures -> RSA Cryptosystem -> Factorization of a number

> for the shown values,

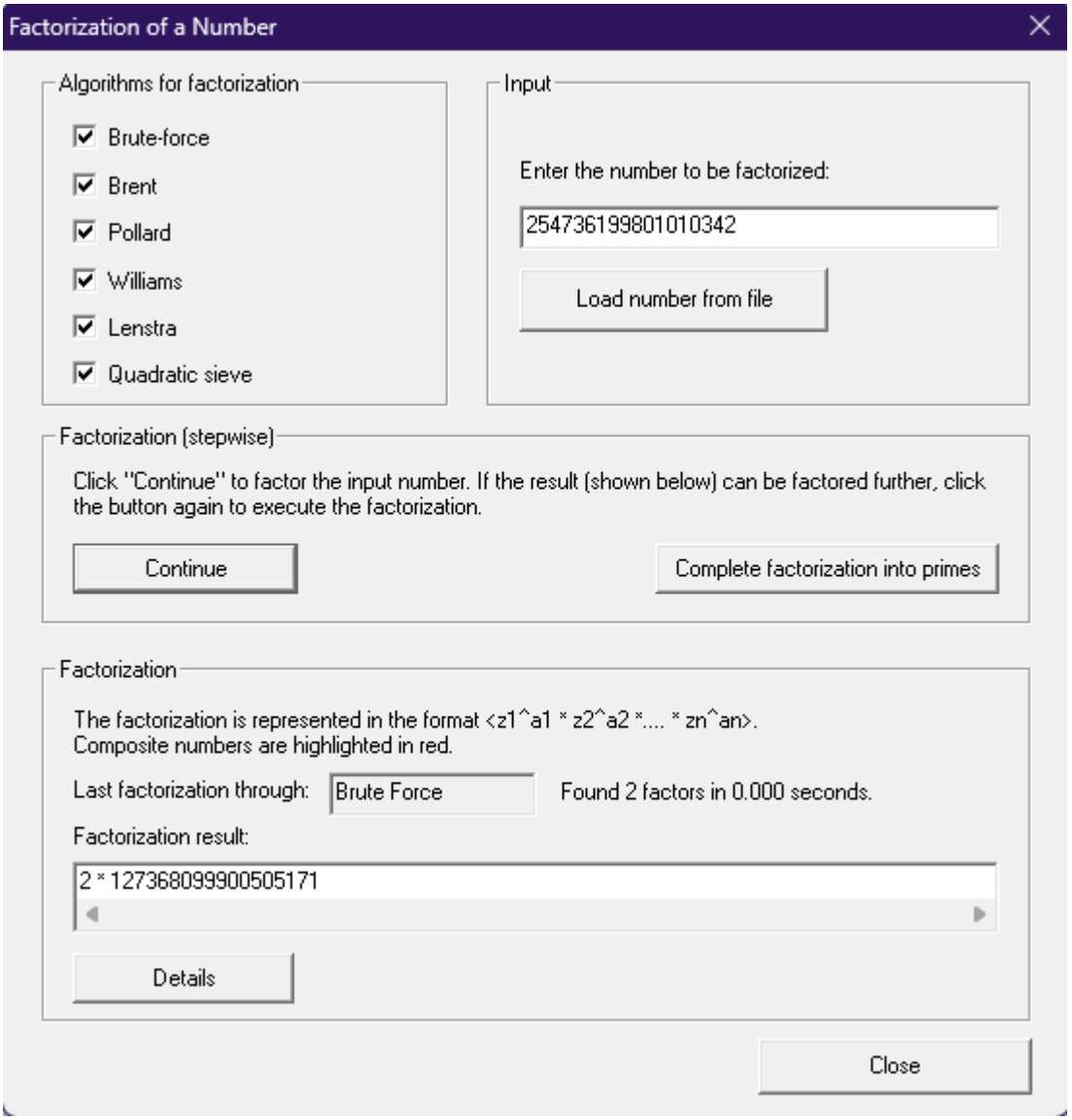

CrypTool 1.4.42 - factorize_t5_ex_4.1.txt
File Edit View Encrypt/Decrypt Digital Signatures/PKI Indiv. Procedures Analysis Options Window Help
factorize_t5_ex_4.1.txt
254736
1998
01
01
03
42


Factorization of a Number
Algorithms for factorization
Brute-force
Brent
Pollard
Williams
Lenstra
Quadratic sieve
Input
Enter the number to be factorized:
5183
Load number from file
Factorization (stepwise)
Click "Continue" to factor the input number. If the result (shown below) can be factored further, click the button again to execute the factorization.
Continue
Complete factorization into primes
Factorization
The factorization is represented in the format <z1^a1 * z2^a2 *.... * zn^an>.
Composite numbers are highlighted in red.
Last factorization through:
Brute Force
Found 2 factors in 0.000 seconds.
Factorization result:
71 * 73
Details
Close

2. Please check how the time of searching for prime numbers increases with the value of searching interval:

Tab: Analysis -> Asymmetric encryption -> Generate Prime Numbers

> for the same input as before,



Time (should) increased due to larger prime number.

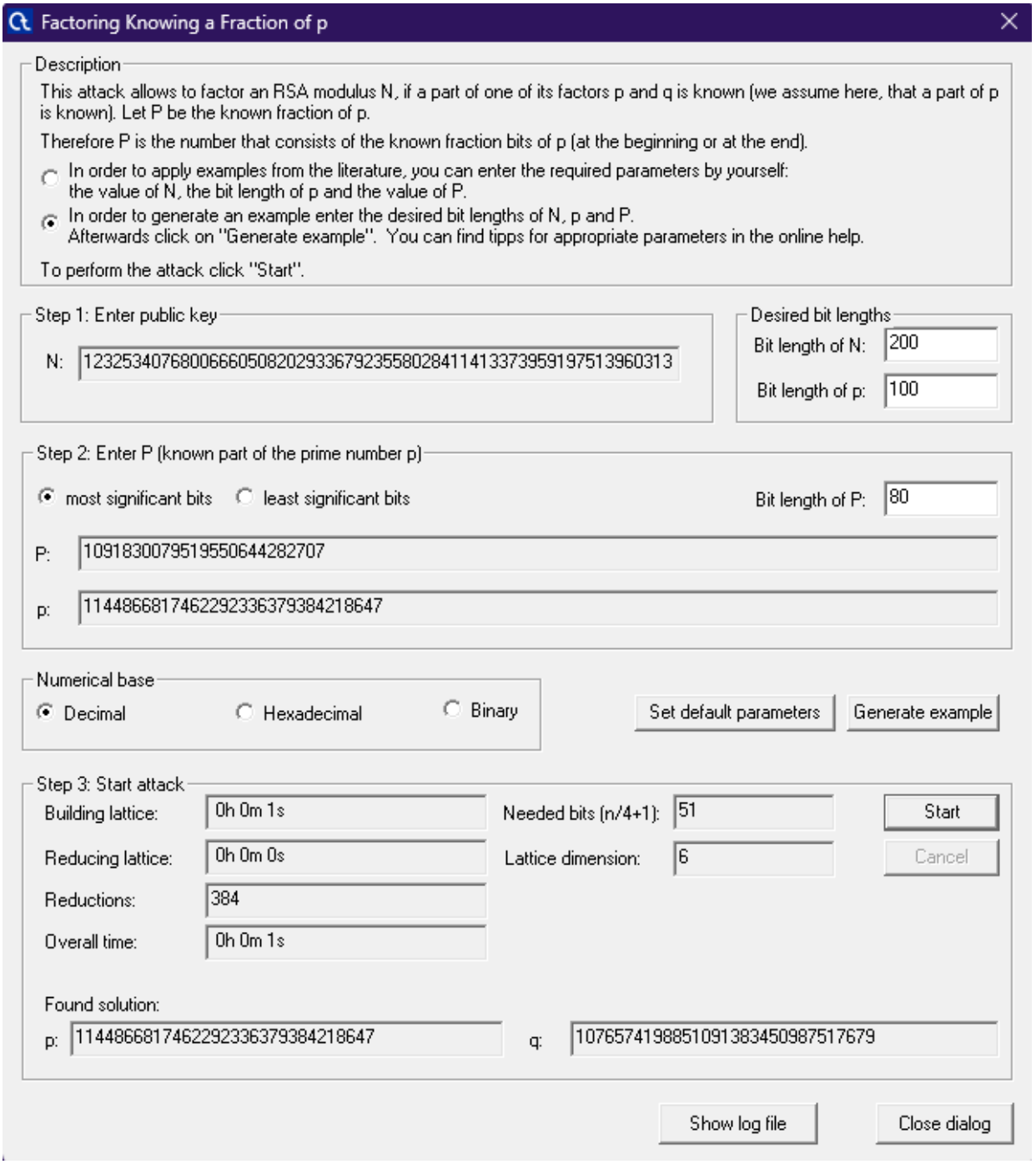
3. Check how different parameter values influence the effectiveness and time of need for the RSA module N factorization attack:

Length N, Length p, Length of known bit string P (lab_5_numbers.txt)

Tab: Analysis -> Asymmetric encryption -> Lattice-Based Attacks on RSA -> Factoring with a hint

> an example,

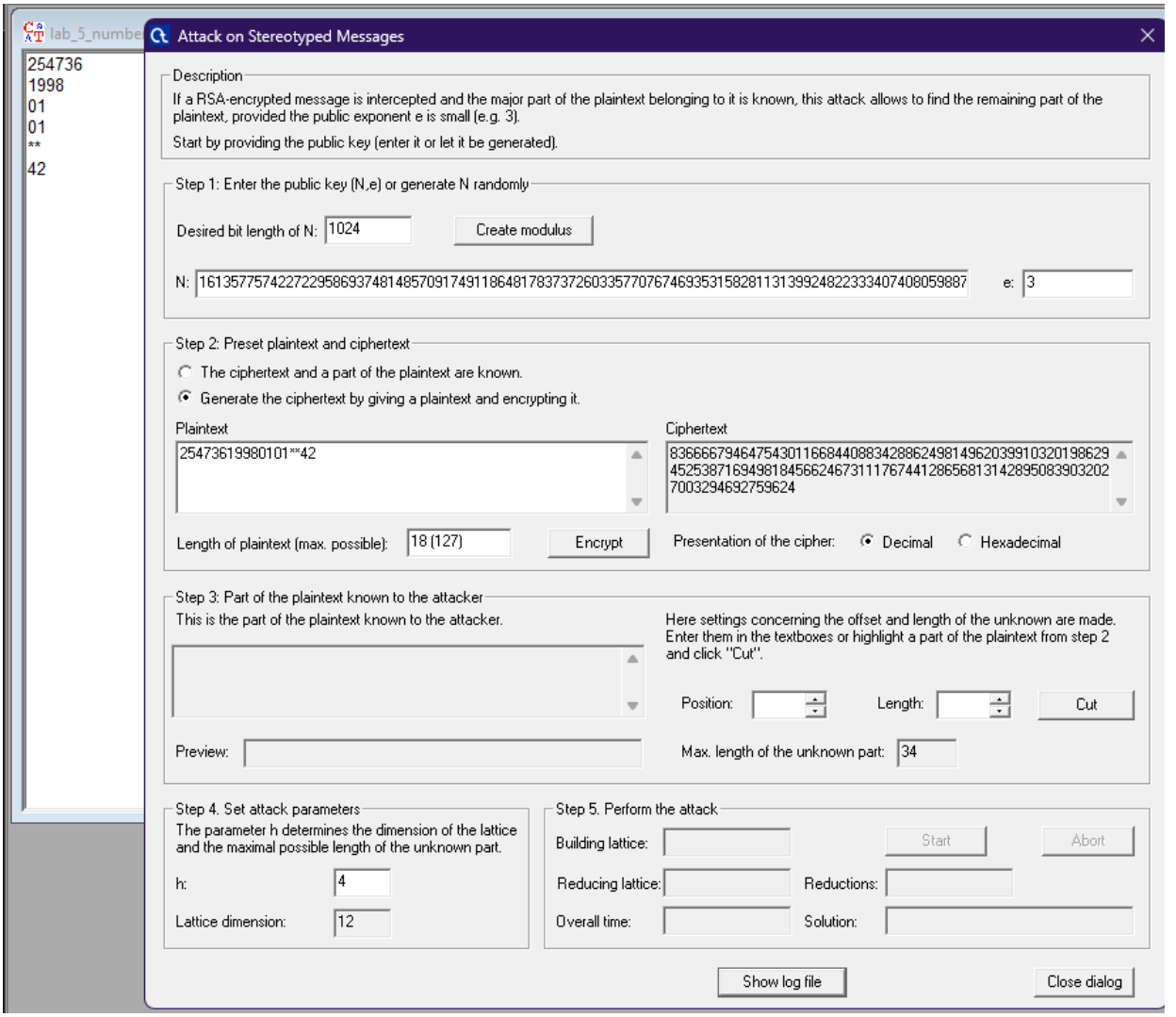
>



4. Find the missing text fragment from lab_5_number.txt using the attack method available in the tab:

Tab: Analysis -> Asymmetric encryption -> Lattice-Based Attacks on RSA -> Attack on Stereotyped Messages

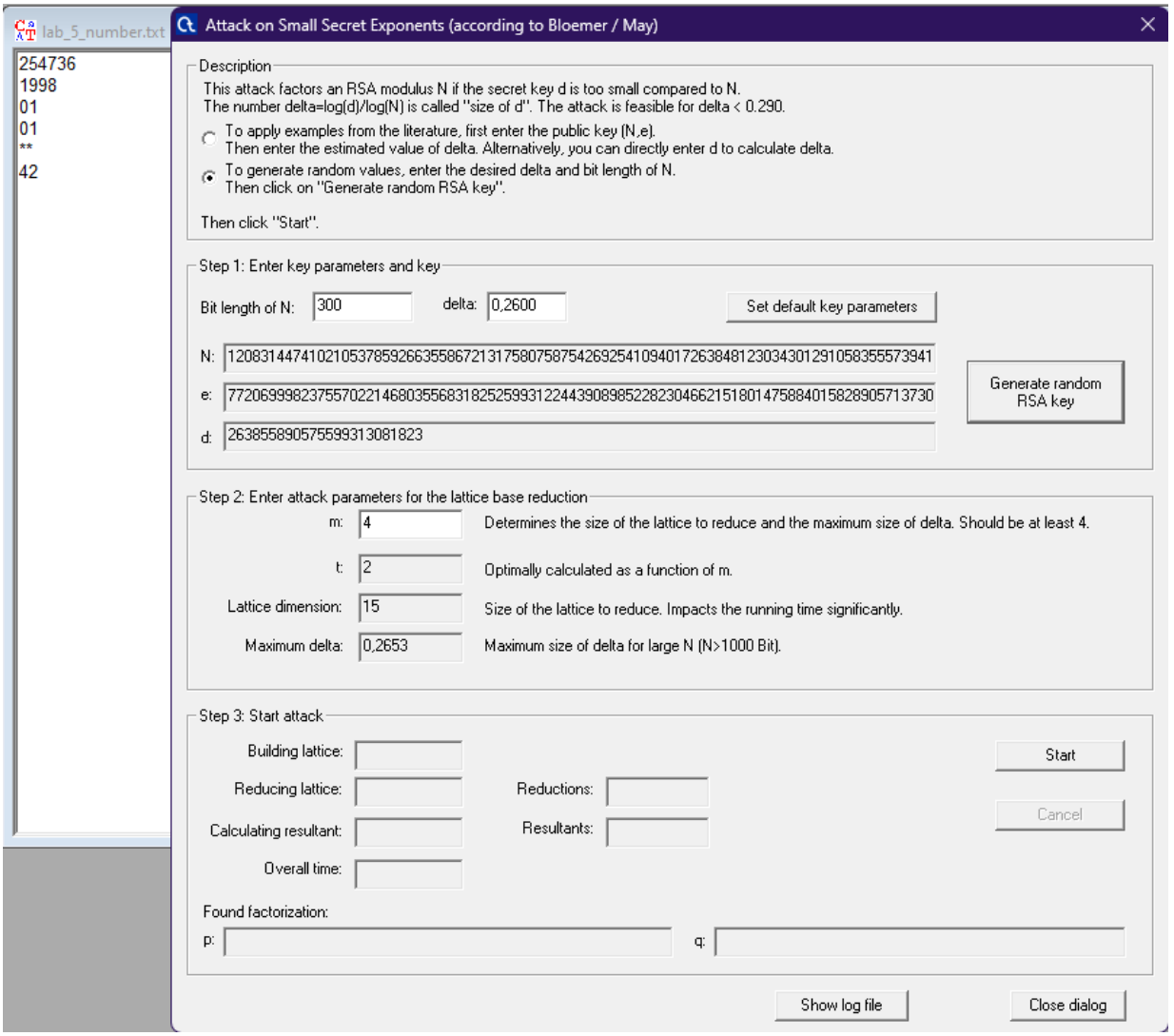
>



5. Factorize the modules from the file lab_5_number.txt using the additional data contained in this file (values: d, e, delta, ...) and the attack method available under the tab:

Tab: Analysis -> Asymmetric encryption -> Lattice-Based Attacks on RSA -> Attack on Small Secret Keys

>



II. **Questions:**

1. What is the minimum length of the module (number N) of the RSA algorithm, which guarantees that its distribution (finding its prime factors) will be difficult enough?

> The minimum length of the modulus (number N) of the RSA algorithm that guarantees that it will be difficult to factorize (find the prime factors of) depends on the level of security that is required and the resources that are available to an attacker. In general, the larger the modulus, the more secure the RSA key will be.

As a rough guide, a modulus with a length of 2048 bits is generally considered to be secure for most purposes. A modulus with a length

of 3072 bits or 4096 bits may be required for more stringent security requirements. It is important to note that these values are only approximate and may change over time as computing power and cryptographic techniques advance.

2. Is it possible to carry out an effective factoring attack based on partial knowledge of the value of one parameter for the module length assumed in the previous point as safe? (task 3)?

> No, it is not possible to perform an effective factoring attack based on partial knowledge of the value of one parameter for the module length. Factoring attacks typically require full knowledge of the value of all parameters for the module length, as well as the values of any other parameters associated with the cryptographic algorithm used to generate the key.

3. In what cases can RSA encryption be threatened by the attack in point 4?

> RSA encryption can be threatened by the attack in stereotyped messages when an attacker is able to collect a large amount of encrypted data and identify patterns or similarities in the encrypted messages. For example, if a large amount of messages are sent with the same payload, an attacker may be able to break the encryption by analyzing the similarities in the encrypted messages.

4. In what cases can RSA encryption be threatened by the attack in point 5?

> RSA encryption can be threatened by an attack on small secret keys when the key is too small. If the key is less than 2048 bits, then it is considered small and the encryption can be cracked by a brute force attack. The larger the key size, the more secure the encryption is.