Text used for analysis:

Hear me, Subjects of Ymir. My name is Eren Jaeger. I'm addressing my fellow Subjects of Ymir, speaking to you directly through the power of the Founder. All the walls on the Island of Paradis have crumbled to the ground, and the legions of Titans buried within have begun their march. My only goal is to protect the lives of the people of Paradis, the island where I was born. Right now, the nations of the world are united in their desire to exterminate my people. And it won't end with our island. They won't be satisfied until every last subject of Ymir is dead. I won't let them have their way. The Titans of the walls....will continue their march, until every trace of life beyond our shores is trampled flat. And the people of Paradis are all that remains of Humanity.

PART 3

I

TASK 1-2

Time needed to find full key for different algorithms(years)				
key Length	64	128	192	256
DES(CBC)	2.2*10^4	-	-	-
AES(CBC)	-	1.1*10^25	2.5*10^44	5.3* 10^63
IDEA	-	9.4 * 10^25	-	-

It is evident that the time required to discover the key through brute force is exceedingly lengthy, surpassing a minimum of 22,000 years. This renders them impervious to such types of attacks.

TASK 3-4

For these tasks AES(CBC) with key length 256 will be used

The time needed if the unknown number of bits is :

with 4 unknown bits:

- at beginning: immediate,

- at the end: immediately,

- in the middle: immediately,

- with 8 unknown bits:

- at beginning (group): immediately,

- at beginning (alternately): immediately,,

- at end (in groups): immediately,,

- at end (alternating): immediately,,

- in the middle (group): immediately,,

- in the middle (alternate): immediate,

- at 12 unknown bits:

- at beginning (group): immediate,

- at beginning (alternating): immediately,,

- at end (group): immediate, at end (alternate): immediate, at end (alternate): immediate,

- at end (alternating): immediate,,

- in the middle (group): immediately,,

- in the middle (alternate): immediately

at 16 unknown bits:

- at start (group): 0,25 s,

- at beginning (alternating): 0,25 s,

- at end (grouped): 0,25 s,

- at the end (alternating): 0,25 s,

- in the middle (group): 0,25 s,

- in the middle (alternating): 0,25 s

at 20 unknown bits:

- in the beginning (group): 3 s,

- at the beginning (alternating): 3 s,

- at the end (in groups): 3 s,

- at the end (alternating): 3 s,

- in the middle (in groups): 3 s,

- in the middle (alternating): 3s

at 24 unknown bits:

- in the beginning (group): 23 s,

- at the beginning (alternating): 23 s,

- at the end (in groups): 23 s,

- at the end (alternating): 23 s,

- in the middle (in groups): 23 s,

- in the middle (alternating): 23s

at 28 unknown bits:

- in the beginning (group): 6 minutes,

- at the beginning (alternating): 6 minutes,

- at the end (in groups): 6 minutes,

- at the end (alternating): 6 minutes,

- in the middle (in groups): 6 minutes,

- in the middle (alternating): 6 minutes

at 32 unknown bits:

- in the beginning (group): 1h 37 minutes,

- at the beginning (alternating): 1h 37 minutes,

- at the end (in groups): 1h 37 minutes,

- at the end (alternating): 1h 37 minutes,

- in the middle (in groups): 1h 37 minutes,

- in the middle (alternating): 1h 37 minutes

The duration of key search increases with the number of unknown bits, while the position of these unknown bits within the key does not impact the search time.

TASK 5

The effectiveness of the decryption algorithm can be characterized as highly satisfactory:

TASK 6

Each instance where the decryption time was relatively short (i.e., less than 1 hour) resulted in the acquisition of a completely accurate key. In cases where the decryption time exceeded 1 hour, key correctness verification was omitted.

TASK 7

The quality of the reconstructed key remains unaffected by the number of bits searched. When searching all possible keys, a correct one is guaranteed to be among them. The primary challenge lies in the time required for key exploration, which escalates with an increase in the number of unknown bits.

TASK 8

The positioning of unknown bits has no bearing on the quality of the reproduced key. Both the key itself and the time required for its retrieval remain consistent irrespective of the location of the missing bits.

II

TASK 1

Most modern block symmetric algorithms are generally considered secure. However, algorithms employing key lengths below 128 bits are less secure, as the time required for brute force key recovery is considerably shorter compared to longer keys. With the utilization of more advanced attack methods, even longer keys may become susceptible to reproduction. In the case of keys with a length of at least 128 bits, the recovery time is sufficiently long, providing resistance against brute force attacks. Nevertheless, it is crucial to emphasize the importance of keeping the encryption key as concealed as possible. Knowledge of a significant portion of the key can substantially reduce the time needed for key reconstruction, making the cipher vulnerable to brute force attacks. When at least half of the key bits are unknown, block ciphers can be deemed fully resistant to brute force attacks, although this does not imply increased resistance against other types of attacks.

TASK 2

Experimental findings reveal that keys with a length of at least 128 bits are the most secure. The time required to reconstruct such keys using the "brute force" method is on the order of 10^25 years, ensuring robust resistance against this type of attack. In contrast, for smaller keys, this time diminishes significantly, substantially compromising the security of the algorithm and rendering it more susceptible to any brute force attack.

TASK 3

The size of the encrypted text plays a significant role in the realm of cryptanalysis. Longer texts offer an advantage as they provide more material for analysis, increasing the likelihood of identifying various relationships within the text. Cyclic repetitions, autocorrelation plots, histograms, and n-grams yield more accurate insights, enabling a more effective and efficient cryptanalysis. We also know that in block algorithms operating in appropriate modes, certain blocks of text can sometimes affect others, which can also provide a clue to a potential intruder.

Conducting this type of analysis becomes more challenging with smaller texts. In such cases, obtained results may inadvertently display false information, such as a distorted autocorrelation plot suggesting repetitions that do not actually exist in a text reproduced multiple times. Additionally, there is insufficient material for histogram and n-gram analysis, as a very short text does not reveal meaningful differences in letter frequency.

However, the effectiveness of breaking ciphers through force attacks is independent of text length. Regardless of the text's length, as all possible keys are systematically searched, the text can be decrypted after a shorter or longer duration.

TASK 4

The pre-encryption processing of a document plays a crucial role in influencing the susceptibility to cryptanalysis. For instance, altering the file format typically enhances its entropy, thereby rendering attempts to break the ciphertext, especially those based on differential analysis, more challenging. When a document undergoes compression, its size is reduced compared to the original, resulting in a lower entropy. This reduction alone poses increased difficulty for typical differential cryptanalysis techniques such as histograms, n-grams, cyclicities, and frequency analysis.

Compression not only diminishes the size of the document but also reduces data redundancy, making it more challenging to identify dependencies within the text. However, compression may introduce periodic repetitions in certain parts of the file, particularly with strong compression of various types of images. This periodicity can complicate the task of deciphering relationships present in the text or images. While compression doesn't necessarily make guessing the original content significantly easier, it does compromise the accuracy of its representation, thereby posing a potential threat to the security of the cryptogram.

TASK 5

365 days * 24 h * 3600 s * 1 000 000 passwords = 3.15 * 10^13 passwords / year

TASK 6

With a key length of 128 bits, represented by 32 hexadecimal characters, each of the 32 positions can assume one of 16 possible values. This leads to a total of 16^32 possibilities, equivalent to 2^128 or 4^64. The sheer magnitude of this numerical value surpasses the computational capacity to exhaustively check within a year. Consequently, cryptographic algorithms utilizing a key length of at least 128 bits can be considered secure and resistant to straightforward "brute force" attacks.

PART 4

I

TASK 1

Student’s ID number:266617

Year:2023

Month:11

Day:15

Hour:15

Minutes:16

Number:266617202311151516

nput number:

266617202311151516

Factorized number	Factor 1	Factor 2	Method	Time
266617202311151516	2	133308601155575...	Brute Force	0.004 seconds.
133308601155575758	2	66654300577787879	Brute Force	0.004 seconds.
66654300577787879	11	6059481870707989	Brute Force	0.004 seconds.
6059481870707989	23	263455733509043	Brute Force	0.004 seconds.
263455733509043	616211623	427541	Pollard	0.100 seconds.

TASK 2

Time of searching for prime numbers	
Interval	Time
0-2^10	immediately
0-2^15	3.4s
0-2^20	1min 25s
0-2^21	1min 54s
0-2^22	3min 40s

TASK 3

Example 1:

N = 896078837981342956751440941147801908171367987837182352662341

P = 990733116324678635714929

bit length for parameter:

p = 80

Time needed: immediately

Example 2:

N =
171251722243075942180137273745347863249901011251258756111876138476471462393049344854222905 9

P = 120391363595085290461224673297232083 9

bit length for parameter:

p = 120

Time needed: 1.3s

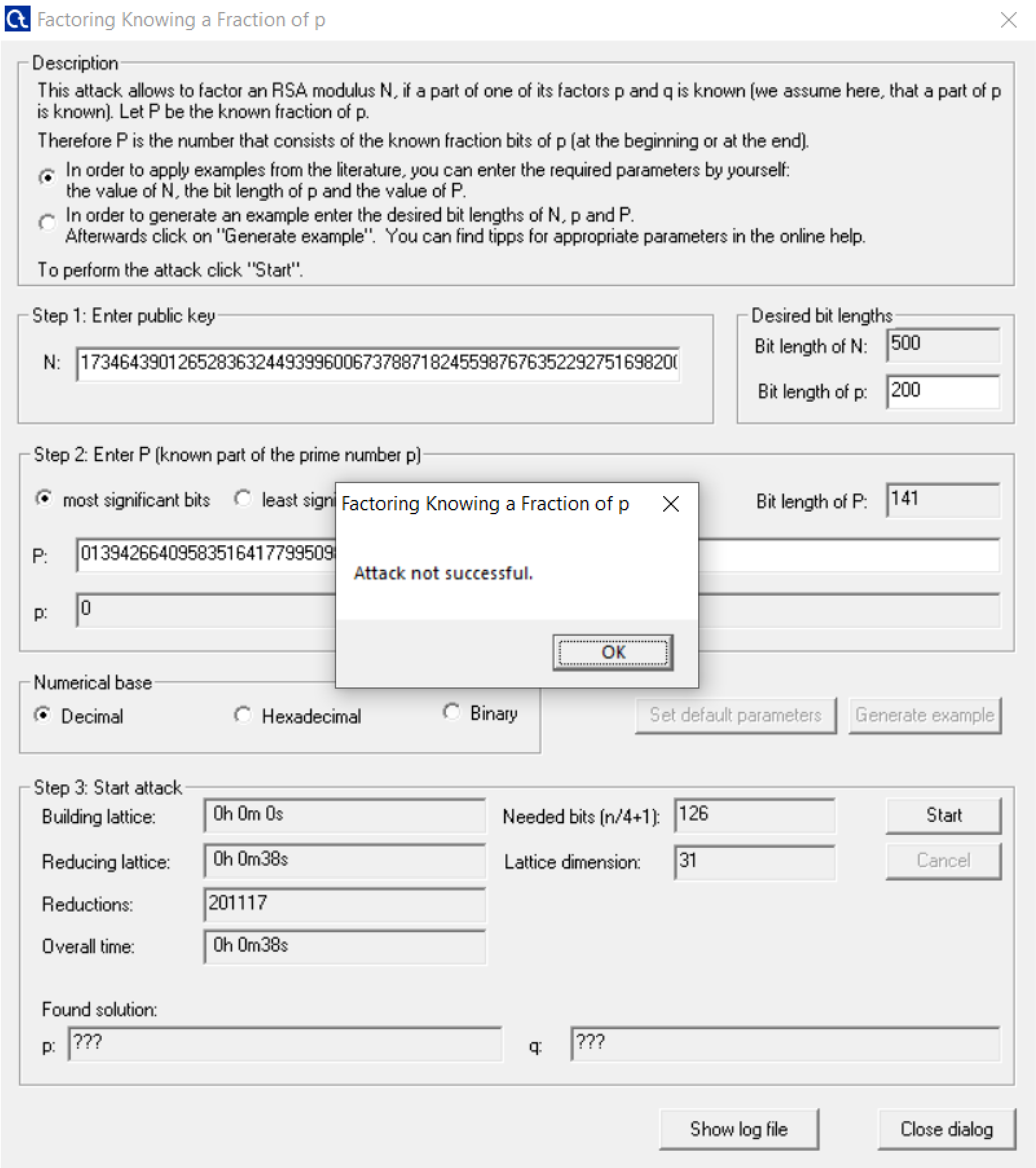
Example 3:

N =
1734643901265283632449399600673788718245598767635229275169820004796491388594632128473063778355083208725319875053472743931611036903602587847506733778057

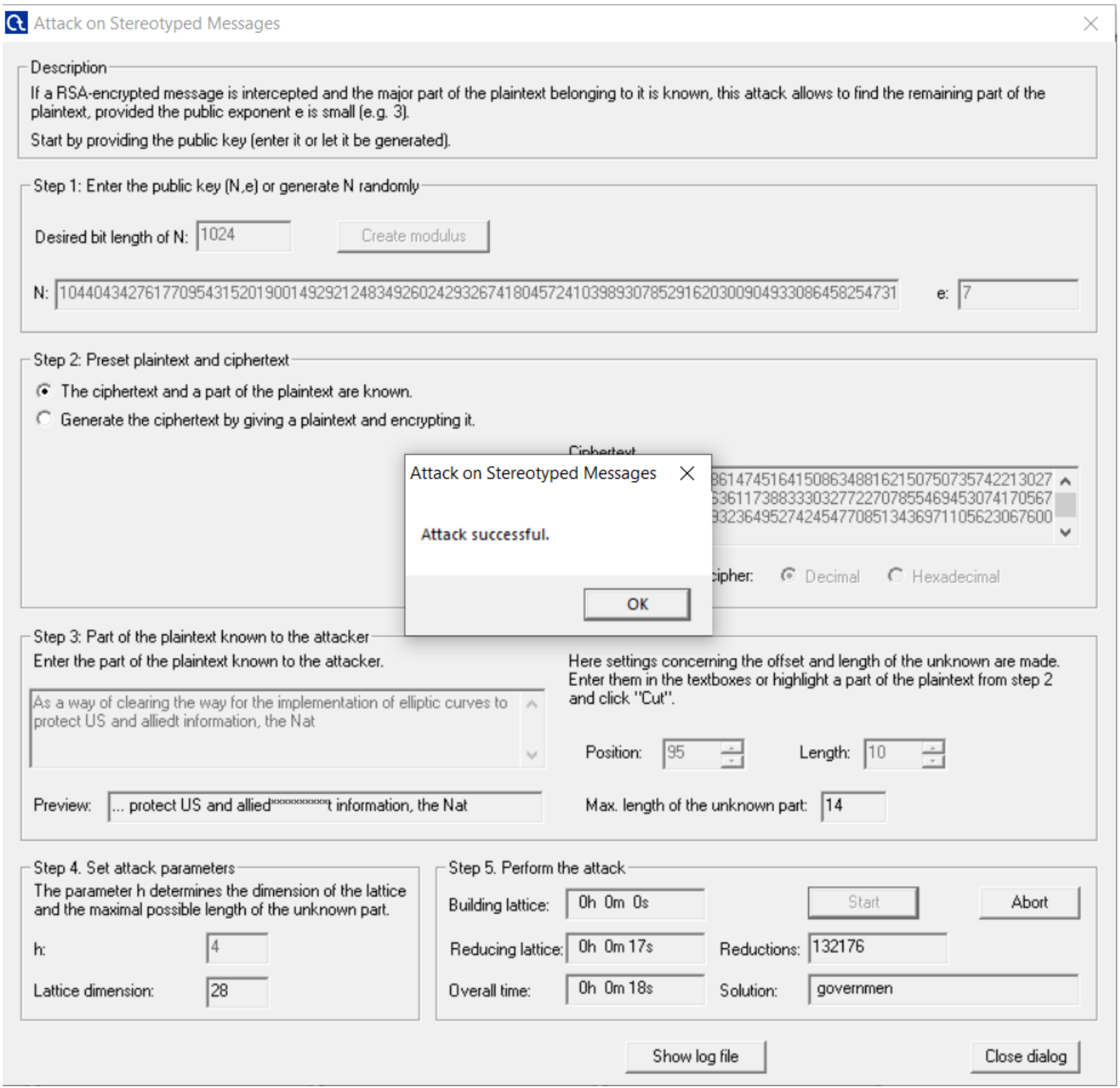
P = 0139426640958351641779950989194304947653679

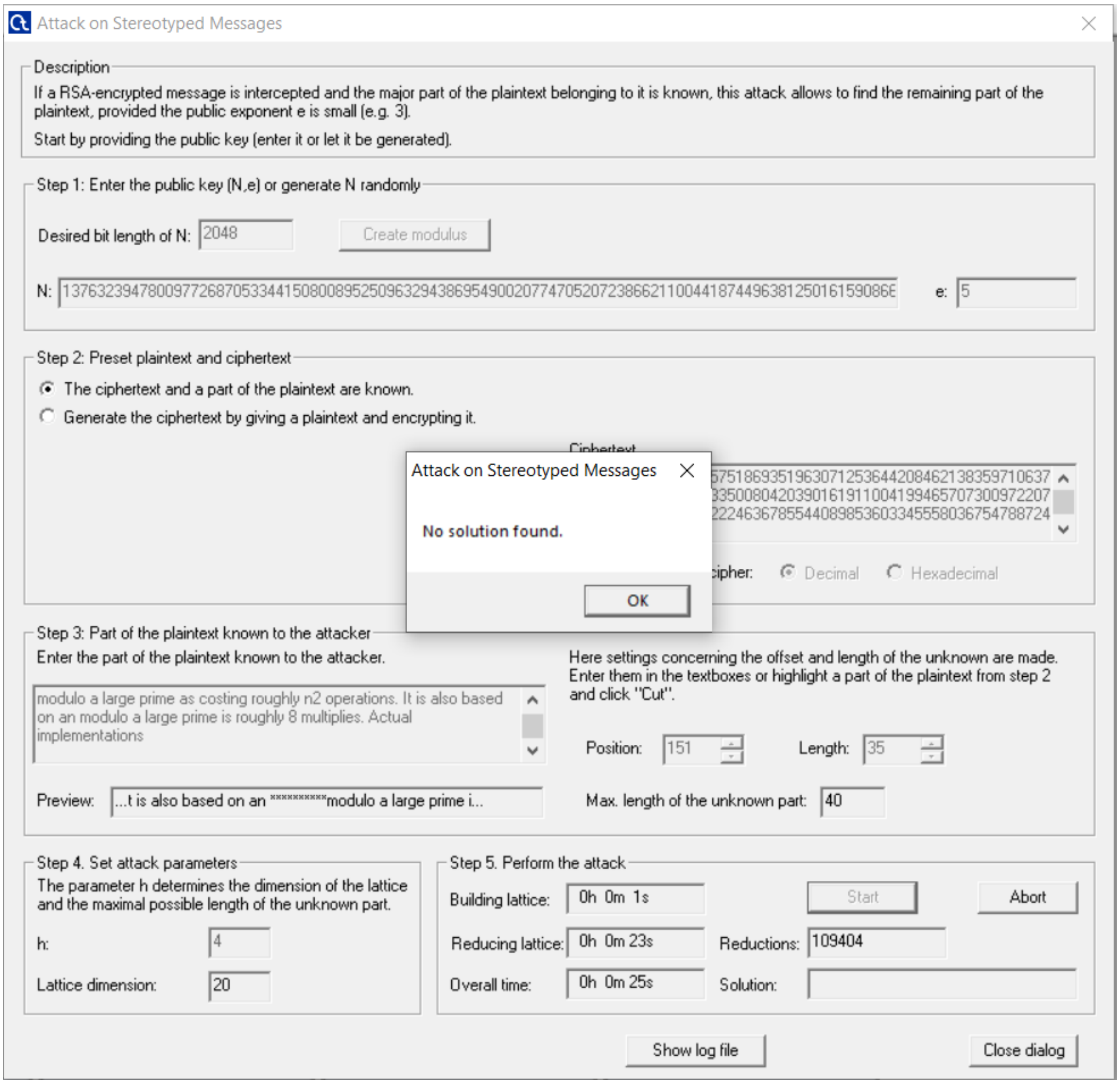
bit length for parameter:

p = 200

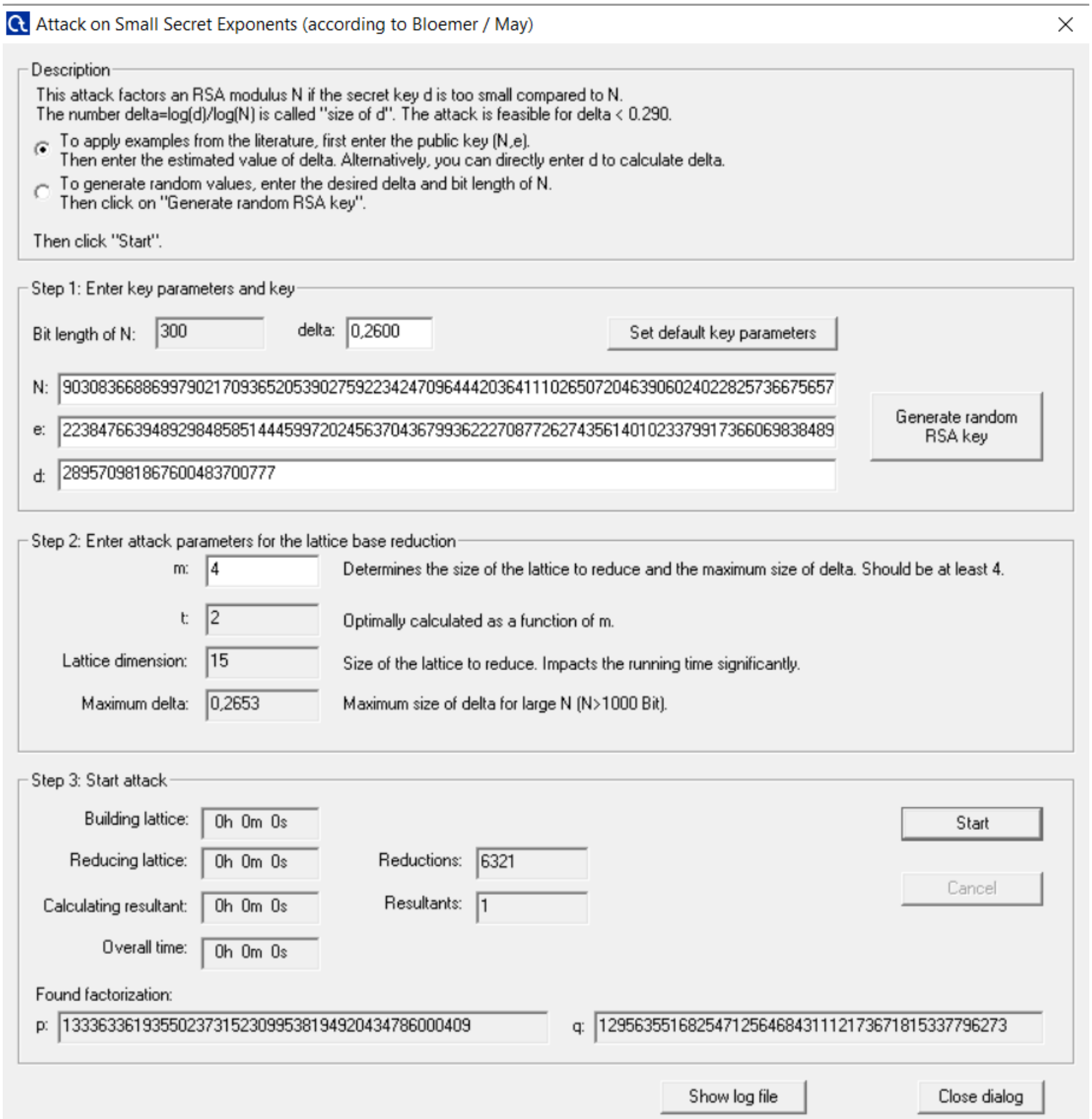


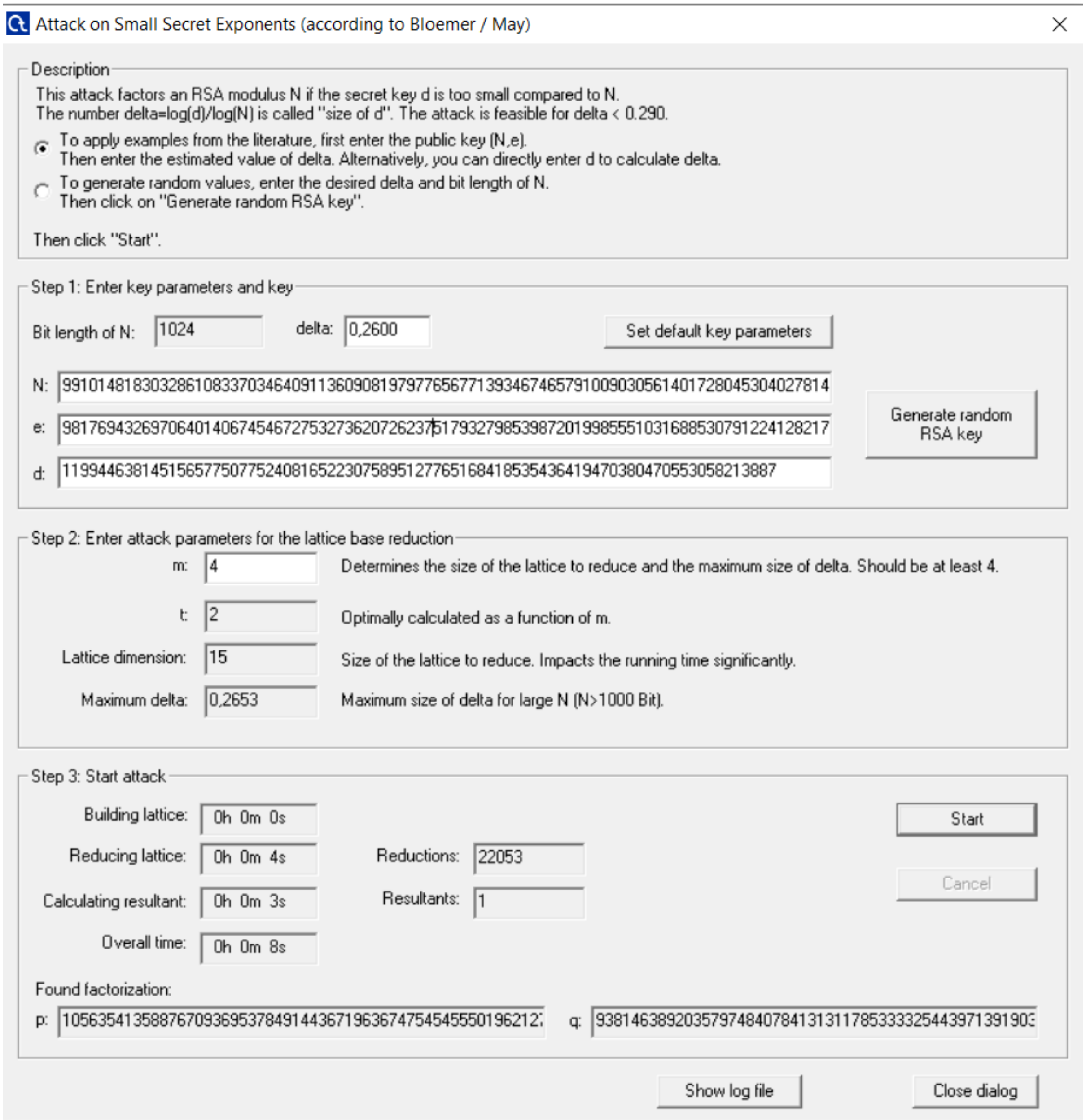
TASK 4

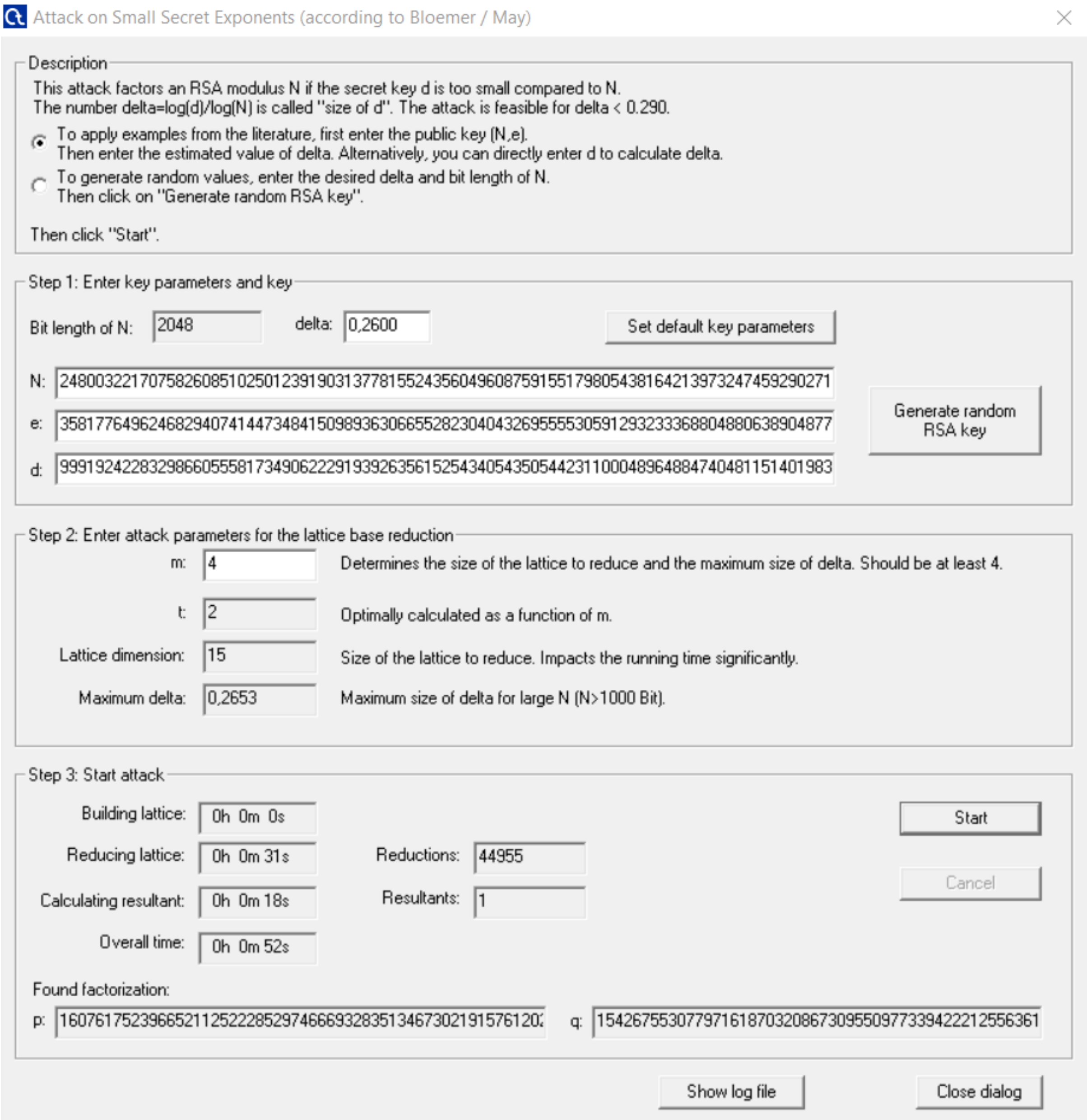

Attack on Stereotyped Messages
Description
If a RSA-encrypted message is intercepted and the major part of the plaintext belonging to it is known, this attack allows to find the remaining part of the plaintext, provided the public exponent e is small (e.g. 3).
Start by providing the public key (enter it or let it be generated).
Step 1: Enter the public key (N,e) or generate N randomly
Desired bit length of N: 1024
Create modulus
N: 10440434276177095431520190014929212483492602429326741804572410398930785291620300904933086458254731
e: 7
Step 2: Preset plaintext and ciphertext
The ciphertext and a part of the plaintext are known.
Generate the ciphertext by giving a plaintext and encrypting it.
Ciphertext
Decimal
Hexadecimal
Attack on Stereotyped Messages
Attack successful.
OK
Step 3: Part of the plaintext known to the attacker
Enter the part of the plaintext known to the attacker.
As a way of clearing the way for the implementation of elliptic curves to protect US and alliedt information, the Nat
Here settings concerning the offset and length of the unknown are made. Enter them in the textboxes or highlight a part of the plaintext from step 2 and click "Cut".
Position: 95
Length: 10
Preview: ... protect US and allied**********t information, the Nat
Max. length of the unknown part: 14
Step 4. Set attack parameters
The parameter h determines the dimension of the lattice and the maximal possible length of the unknown part.
h: 4
Lattice dimension: 28
Step 5. Perform the attack
Building lattice: 0h 0m 0s
Reducing lattice: 0h 0m 17s
Overall time: 0h 0m 18s
Start
Abort
Reductions: 132176
Solution: governmen
Show log file
Close dialog


Attack on Stereotyped Messages
Description
If a RSA-encrypted message is intercepted and the major part of the plaintext belonging to it is known, this attack allows to find the remaining part of the plaintext, provided the public exponent e is small (e.g. 3).
Start by providing the public key (enter it or let it be generated).
Step 1: Enter the public key (N,e) or generate N randomly
Desired bit length of N: 2048
Create modulus
N: 1376323947800977268705334415080089525096329438695490020774705207238662110044187449638125016159086E
e: 5
Step 2: Preset plaintext and ciphertext
The ciphertext and a part of the plaintext are known.
Generate the ciphertext by giving a plaintext and encrypting it.
Ciphertext
5751869351963071253644208462138359710637
3350080420390161911004199465707300972207
2224636785544089853603345558036754788724
cipher: Decimal Hexadecimal
Attack on Stereotyped Messages
No solution found.
OK
Step 3: Part of the plaintext known to the attacker
Enter the part of the plaintext known to the attacker.
modulo a large prime as costing roughly n2 operations. It is also based on an modulo a large prime is roughly 8 multiplies. Actual implementations
Preview: ...t is also based on an **********modulo a large prime i...
Here settings concerning the offset and length of the unknown are made. Enter them in the textboxes or highlight a part of the plaintext from step 2 and click "Cut".
Position: 151
Length: 35
Max. length of the unknown part: 40
Step 4. Set attack parameters
The parameter h determines the dimension of the lattice and the maximal possible length of the unknown part.
h: 4
Lattice dimension: 20
Step 5. Perform the attack
Building lattice: 0h 0m 1s
Reducing lattice: 0h 0m 23s
Overall time: 0h 0m 25s
Start
Abort
Reductions: 109404
Solution:
Show log file
Close dialog

TASK 5


Attack on Small Secret Exponents (according to Bloemer / May)
Description
This attack factors an RSA modulus N if the secret key d is too small compared to N.
The number delta=log(d)/log(N) is called "size of d". The attack is feasible for delta < 0.290.
To apply examples from the literature, first enter the public key (N,e).
Then enter the estimated value of delta. Alternatively, you can directly enter d to calculate delta.
To generate random values, enter the desired delta and bit length of N.
Then click on "Generate random RSA key".
Then click "Start".
Step 1: Enter key parameters and key
Bit length of N: 300
delta: 0,2600
Set default key parameters
N: 903083668869979021709365205390275922342470964442036411102650720463906024022825736675657
e: 223847663948929848585144459972024563704367993622270877262743561401023379917366069838489
d: 28957098186760048370077 7
Generate random RSA key
Step 2: Enter attack parameters for the lattice base reduction
m: 4
Determines the size of the lattice to reduce and the maximum size of delta. Should be at least 4.
t: 2
Optimally calculated as a function of m.
Lattice dimension: 15
Size of the lattice to reduce. Impacts the running time significantly.
Maximum delta: 0,2653
Maximum size of delta for large N (N>1000 Bit).
Step 3: Start attack
Building lattice: 0h 0m 0s
Reducing lattice: 0h 0m 0s
Reductions: 6321
Calculating resultant: 0h 0m 0s
Resultants: 1
Overall time: 0h 0m 0s
Start
Cancel
Found factorization:
p: 1333633619355023731523099538194920434786000409
q: 1295635516825471256468431112173671815337796273
Show log file
Close dialog


Attack on Small Secret Exponents (according to Bloemer / May)
Description
This attack factors an RSA modulus N if the secret key d is too small compared to N.
The number delta=log(d)/log(N) is called "size of d". The attack is feasible for delta < 0.290.
To apply examples from the literature, first enter the public key (N,e).
Then enter the estimated value of delta. Alternatively, you can directly enter d to calculate delta.
To generate random values, enter the desired delta and bit length of N.
Then click on "Generate random RSA key".
Then click "Start".
Step 1: Enter key parameters and key
Bit length of N:
1024
delta:
0,2600
Set default key parameters
N:
991014818303286108337034640911360908197977656771393467465791009030561401728045304027814
e:
981769432697064014067454672753273620726237517932798539872019985551031688530791224128217
d:
119944638145156577507752408165223075895127765168418535436419470380470553058213887
Generate random RSA key
Step 2: Enter attack parameters for the lattice base reduction
m:
4
Determines the size of the lattice to reduce and the maximum size of delta. Should be at least 4.
t:
2
Optimally calculated as a function of m.
Lattice dimension:
15
Size of the lattice to reduce. Impacts the running time significantly.
Maximum delta:
0,2653
Maximum size of delta for large N (N>1000 Bit).
Step 3: Start attack
Building lattice:
0h 0m 0s
Reducing lattice:
0h 0m 4s
Reductions:
22053
Calculating resultant:
0h 0m 3s
Resultants:
1
Overall time:
0h 0m 8s
Start
Cancel
Found factorization:
p:
1056354135887670936953784914436719636747545455501962127
q:
93814638920357974840784131311785333325443971391903
Show log file
Close dialog



II

TASK 1

A 2048-bit key for RSA is generally deemed secure for most applications, providing a satisfactory level of security. However, for enhanced and more resilient long-term protection, it is advisable to consider employing a 3072-bit or even a 4096-bit key.

TASK 2

With knowledge of a significant number of bits, particularly more than half, of one of the numbers involved, we can execute this attack relatively swiftly, even when dealing with lengthy and ostensibly secure module lengths. Consequently, the search space is significantly reduced, as it becomes unnecessary to scan the entire range, focusing instead on a subset that fulfills the specified conditions.

TASK 3

RSA encryption is susceptible to the "Factoring with a hint" attack in situations where an attacker gains partial information about the factors of the RSA modulus. This can occur when there is a leak of bits from one of the prime factors, the same message is encrypted repeatedly, plaintexts have low entropy or are predictable, or short key lengths are used. Mitigating this risk involves employing longer key lengths, avoiding predictable plaintexts, and staying updated on cryptographic best practices.

TASK 4

RSA encryption faces a risk from the "Attack on Small Secret Keys" when utilizing inadequate key lengths. Short key lengths create a smaller search space, making brute-force attacks or factorization more feasible for attackers. To address this vulnerability, it is crucial to adhere to recommended key length standards to enhance the security of RSA encryption.