2. Properties of the RSA algorithm.

I. Tasks:

1. Done

2. Prepare three plaintexts - homogeneous text, medium-diversified text, highly diversified text, (at least 1500 characters each).

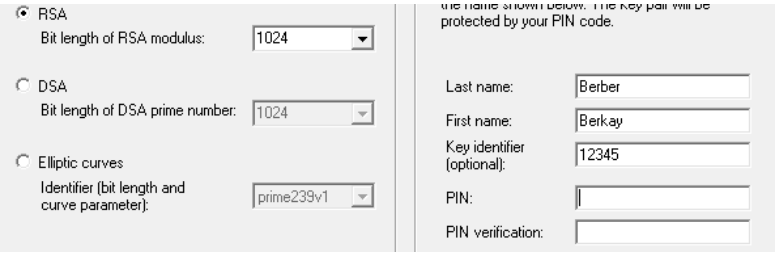
-> **Homogeneous text**: I prepared 1571 character of “b”.

-> **Medium-diversified text:** I prepared totally 1598 character of “Sun is alone too but still shines.”

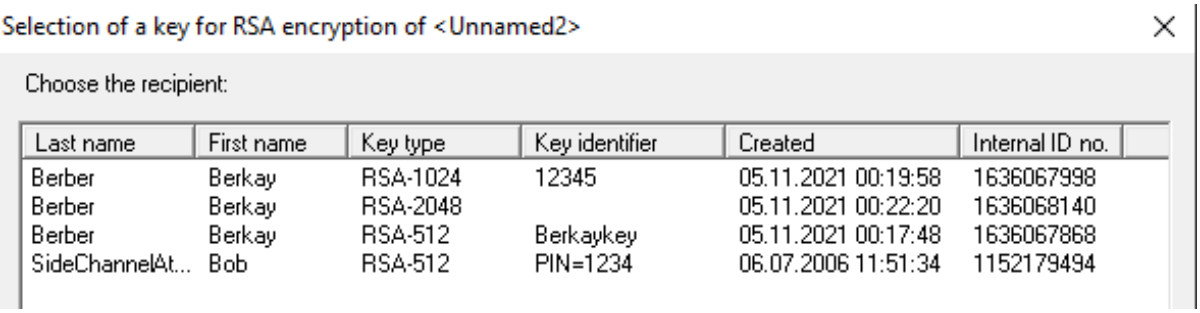
-> **highly diversified text:** I prepared totally 1544 character of “Diversification is a risk management strategy that mixes a wide variety of investments within a portfolio. A diversified portfolio contains a mix of distinct asset types and investment vehicles in an attempt at limiting exposure to any single asset or risk.”

3. Generate the following cryptographic keys (Digital Signature/PKI/Generate/Import keys):

I generated the 512-1024-2048 bits of keys following the instruction. Graph1.2 is the final presentation.



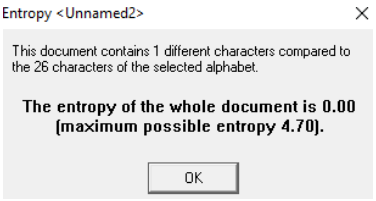
Graph1.1

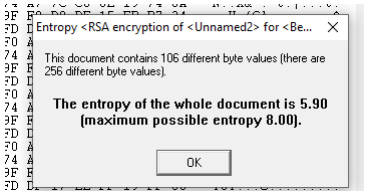


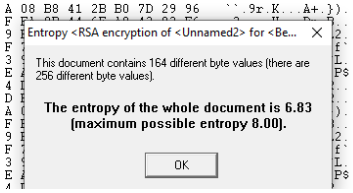
Graph1.2

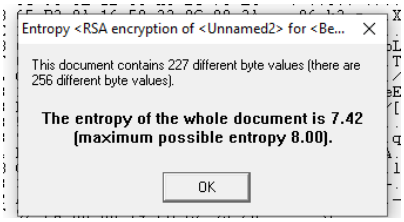
4. For different key lengths (512,1024,2048) and previously prepared plaintexts, compare the entropy of plain text with the entropy after encryption (encryption entropy). Check and compare the autocorrelation of the ciphertext and plaintext.

HOMOGENEUS:

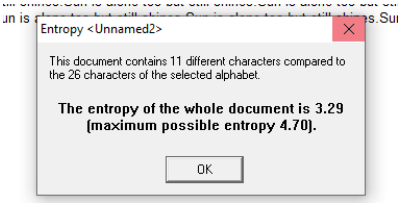


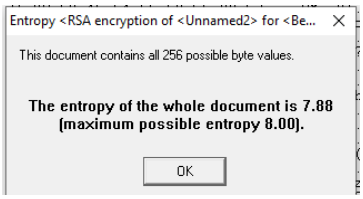


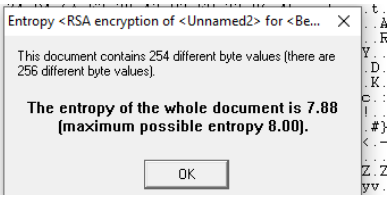


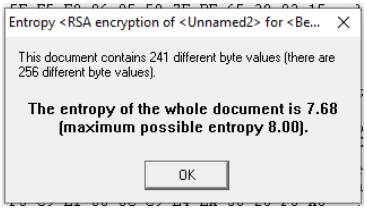


MEDİUM-DİVERSİFİED:

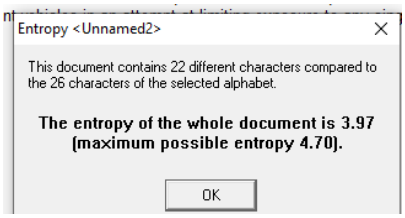


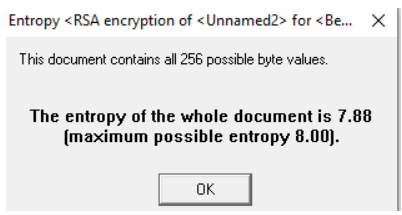


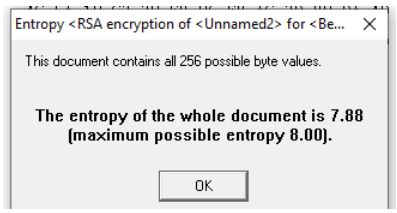


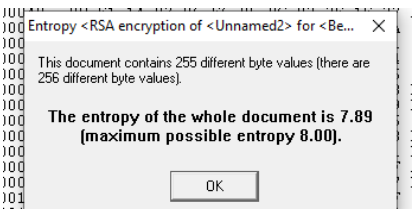


HİGHLY-DİVERSİFİED:







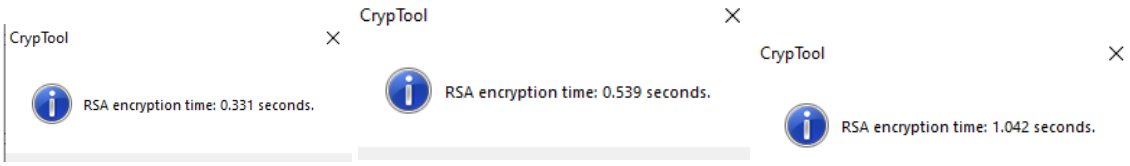


Final graph according to comparison is in the below table.

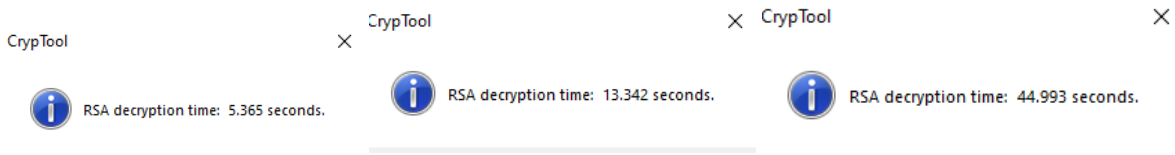
	Homogeneous	Medium-diversified	Highly-diversified
Normal entropy	0.00/4.70	3.29/4.70	3.97/4.70
RSA 512	5.90/8.00	7.88/8.00	7.88/8.00
RSA 1024	6.83/8.00	7.88/8.00	7.88/8.00
RSA 2048	7.42/8.00	7.68/8.00	7.89/8.00

5. For different RSA algorithm key lengths, measure the encryption and decryption time for files of 1MB, 2MB, 5MB (with an accuracy of 2 seconds).

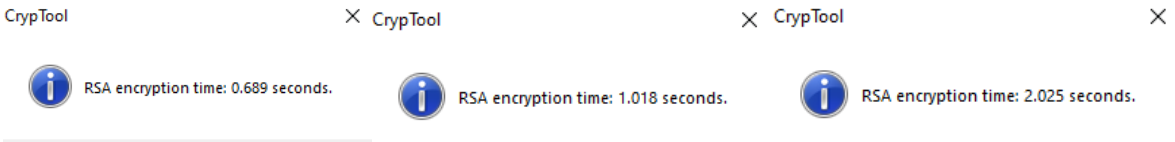
1MB ENCRYTİON:



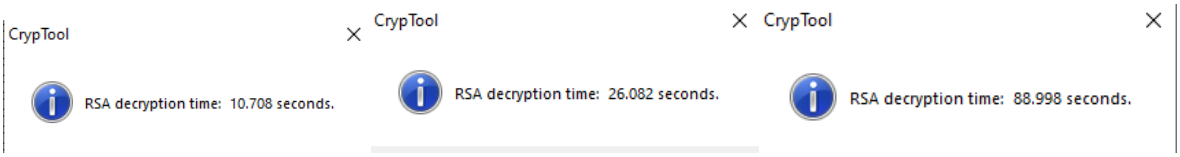
1MB DECRYPTİON:



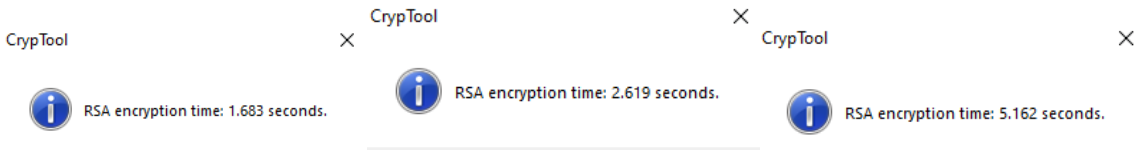
2MB ENCRYTİON:



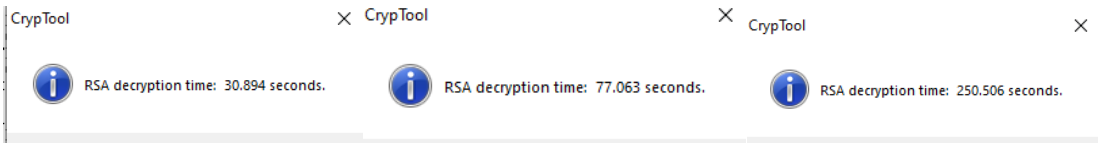
2MB DECRYPTİON:



5MB ENCRYTİON:



5MB DECRYPTİON:



Final graph according to measure of the files is in the below table.

	1MB/ENCRYPTİON	1MB/DECRYPTİON	2MB/ENCRYPTİON	2MB/DECRYPTİON	5MB/ ENCRYPTİON	5MB/DECRYPTİON
RSA 512	0.331	5.365	0.689	10.706	1.683	30.894
RSA 1024	0.539	13.342	1.018	26.082	2.619	77.063
RSA 2048	1.042	44.993	2.025	88.998	5.162	250.506

II. Questions:

1. Does the key length affect the ciphertext entropy? If so, how?

-> As far as I understood from the exercises we did so far, key length affects the ciphertext entropy. We are able to see the difference in the table of task4 that checking entropy of ciphertext without RSA has low entropy than measured with the key length. Text doesn't make such difference. However, key length doubles to normal entropy which is simple text.

2. Does the length of the key affect the autocorrelation of the ciphertext? If so, how?

-> Yes, if plaintext that we put the characters each is contains different text in it, will be easy to realize the autocorrelation more. Since, length of the key increased, than possibility of the shift in the analyze will be increased. We have better autocorrelation with the different characters.

3. Does the ciphertext entropy depend on the plaintext entropy? If so, how?

-> Yes, it all depends on the plaintext that we use for encryption. Since, Plaintext has more characters, than it will affect the ciphertext this much. While length of the RSA escalated impact on the entropy changes.

4. How does the encryption/decryption time depend on the file length?

-> Encryption and decryption methods for complexity is proportional to the size of the text character. Which means, the larger file size equals a longer encryption and decryption times as we can see in the final graph of exercise 5.

5. What is the time of encryption/decryption with asymmetric algorithms compared to the execution of these operations with a symmetric algorithm?

-> Symmetric encryption uses a single key that needs to be shared among the people who need to receive the message while asymmetric encryption uses a pair of public key and a private key to encrypt and decrypt messages when communicating. And symmetric encryption uses simpler operations, such as XOR and multiply, on smaller bits (64, 128). This is why Asymmetric algorithms are more slower than compared to symmetric algorithms.

7. Is it possible to remove a fragment from the ciphertext so that the remaining text of the plaintext after decryption is readable?

-> According to my researches its possible to remove a fragment from the ciphertext, However when I checked and deleted a part of the data, after the encryption it gave meaningless text when I decrypt. in my opinion when we removed fragment from encrypted data , we will come across to problems to read after the decryption done.

8. What are the advantages and disadvantages of asymmetric algorithms compared to symmetric ones?

-> The main advantage of symmetric encryption over asymmetric encryption is that it is fast and efficient for large amounts of data; the disadvantage is the need to keep the key secret - this can be especially challenging where encryption and decryption take place in different locations, requiring the key to be moved safely between locations.

9. In which applications is it better to use asymmetric algorithms, and in which symmetric algorithms?

-> Asymmetric algorithms are slower than symmetric algorithms. Because, Asymmetric encryption uses longer keys than symmetric encryption in order to provide better security than symmetric key encryption. While the longer key length in itself is not so much a disadvantage it contributes t slower encryption speed.

Asymmetric encryption is used in key exchange(security), email security, web security, and other encryption systems that require key exchange over the public network. Asymmetric encryption schemes are perfect for securely exchanging small amounts of data.

Some examples of where symmetric cryptography is used are: Payment applications, such as card transactions.