CENG471 Cryptography Term Project

<table>
<tr><th>Delivery date</th><th>Required Modules</th><th>Cryptosystems</th></tr>
<tr><td rowspan="5">Stage 1:
21.04.2019,
23:55</td><td>“Euclid’s Extended Algorithm” to calculate GCD, multiplicative inverse and to check the relatively prime condition</td><td rowspan="5">In this stage, the main asymmetrical primitives (listed on the left) are applied. Most of them will also be used for the following stages of the term project. Therefore, please design your personal re-usable code modules.
After coding the primitives, implement the following steps for symmetrical encryption:
1) Sender and receiver generate their own public-private keys.
2) Both parties generate a common secret key by DH Key Exchange scheme.
3) With any AES/DES usage, a data file is encrypted by sender and decrypted by receiver.
Please measure the performance of your AES/DES code (in terms of memory, execution time, and security) for different lengths of data files, i.e. 1-page length, 10 page-length, 100 page-length, and 1000 page-length documents. Insert the measurement results to tables and show them in the figures.</td></tr>
<tr><td>“FLT” to test the primality of any number</td></tr>
<tr><td>“Fast Exponentiation”</td></tr>
<tr><td>“DH Key Exchange” scheme</td></tr>
<tr><td>“AES or DES” (You can use any ready open implementation from internet)</td></tr>
<tr><td>Stage 2:
28.04.2019,
23:55</td><td>“RSA” encryption and decryption cryptosystem</td><td>In this second stage, sender and receiver first generate their own RSA public-private key pairs and share their public keys.
Then, RSA encryption scheme is used to encrypt and decrypt a file.
Please also measure the performance of your code for your data files which are used at the previous step and use figures and tables to explain your results.
Finally, compare the results at the first and second stages.</td></tr>
<tr><td rowspan="2">Stage 3:
12.05.2019,
23:55</td><td>“SHA 2 or 3 with 256” hash (You can use any ready open implementation from internet)</td><td rowspan="2">In this step; sender would like to send a document after he/she sign it. Then, the receiver should verify the sender’s signature.</td></tr>
<tr><td>“DSS” digital signature signing and verification</td></tr>
</table>

		You can implement this solution to satisfy secrecy, identification, integrity, and non-repudiation requirements. Please show how these requirements are satisfied in this scenario.