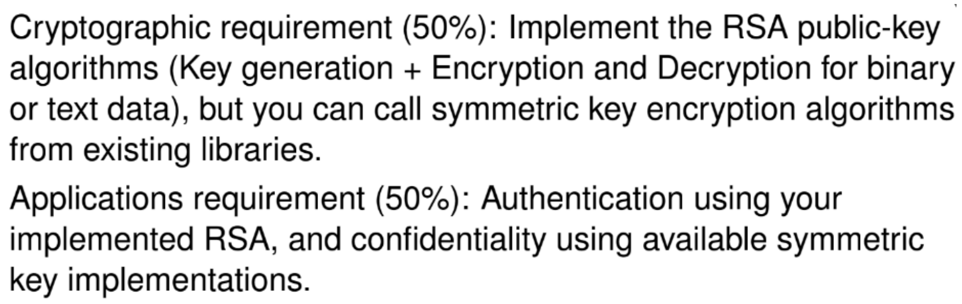
Project Requirements



Cryptographic requirement (50%): Implement the RSA public-key algorithms (Key generation + Encryption and Decryption for binary or text data), but you can call symmetric key encryption algorithms from existing libraries.

Applications requirement (50%): Authentication using your implemented RSA, and confidentiality using available symmetric key implementations.

Authentication processes (Examples)

- ☐ ***Sender*** creates a clear-text messages
- ☐ ***Sender*** creates a SHA-1 Message digest of the clear text message
- ☐ ***Sender*** encrypts the SHA-1 message digest using the RSA asymmetric encryption algorithm with the ***sender’s*** private key, producing a digital signature that is attached to the clear-text message.
- ☐ ***Receiver*** uses the RSA asymmetric encryption algorithm with the ***sender’s*** public key to decrypt the digital signature and recover the SHA-1 message digest.
- ☐ ***Receiver*** generates a SHA-1 message digest from the clear-text message and compares the generated SHA-1 message digest with the decrypted SHA-1 message digest; *if they match, then the message is accepted as authentic.*

Confidentially protection processes *(Examples)*

- ☐ ***Sender*** generates a random 128-bit number to be used as a session shared secret key (SSSK) for this message only
- ☐ ***Sender*** encrypts the clear-text message, and appends a digital signature,using a symmetric encryption algorithm, such as CAST-128, IDEA or 3DES, with the SSSK.
- ☐ ***Sender*** then encrypts the SSSK using RSA with each recipient’s public key(s) and then appends each uniquely encrypted copy of the SSSK to the black-text message.
- ☐ Each ***receiver*** uses RSA with its private key to decrypt and recover their copy of the SSSK.
- ☐ The decrypted SSSK is used to decrypt the black-text message thereby recovering the clear-text message.

Topics relevant

- □ E-voting system in organization
- □ Chat (message) application
- □ Email securing platforms
- □ Evaluation system (Hung’s example)