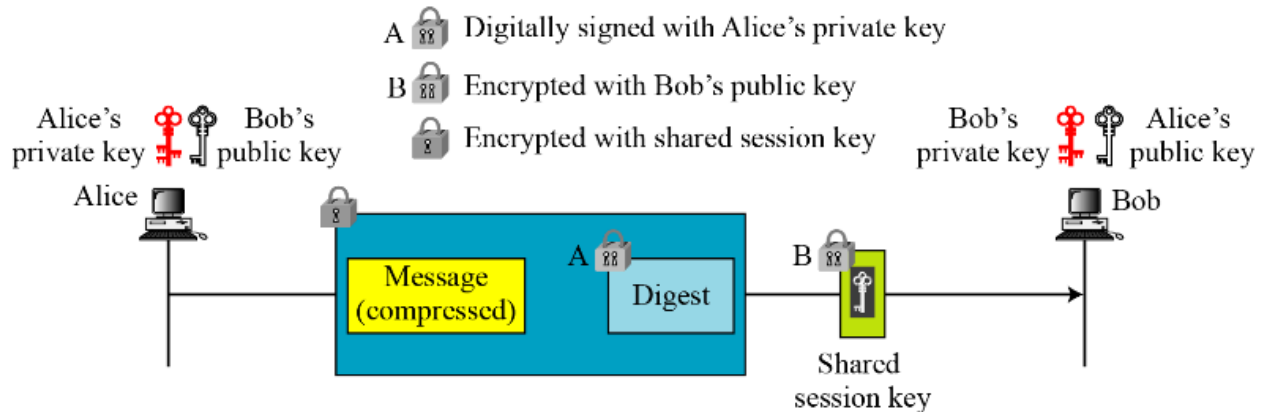


LAB ASSIGNMENT - 2

Deadline: 09/07/2024 (Monday) 11.59 PM

Following diagram shows a secure message transmission protocol:



Implement all these algorithms using only functions of crypto++ library

Algorithms to Implement:

- **Key Generation:**
 - Generate two pairs of RSA keys (public and private).
 - Each key should be stored as separate binary files.
- **Digital Signature:**
 - Take the data file and create a digital signature of the data using the sender's RSA private key.
- **Message Encryption:**
 - Concatenate the generated signature with the data file.
 - Generate a random session key (AES-256)
 - Encrypt the concatenated data file using the generated session key.
 - Encrypt the session key with the recipient's RSA public key.
 - Store the encrypted data file and the encrypted key as two separate binary files.
- **Message Decryption:**
 - Extract the session key by decrypting the encrypted key file using the recipient's RSA private key.

- Decrypt the data file using the session key.
- Extract the plaintext data and the digital signature separately.
- **Signature Verification:**
 - Verify the integrity of the data by checking the signature using the sender's RSA public key.

Implementation Guidelines:

- Implement the following functions:

- `generate_keys()`
- `sign_message()`
- `encrypt_message()`
- `decrypt_message()`
- `verify_signature()`

Ensure that the implemented functions are non-interactive. All necessary files and inputs should be provided as command-line arguments. For example:

- `./generate_keys`

Output: `public_key_file, private_key_file`

- `./sign_message private_key_file data_file`

Output: `signature_file`

- `./encrypt_message public_key_file data_file
signature_file`

Output: `encrypted_key_file, encrypted_data_file`

- `./decrypt_message private_key_file encrypted_data_file
encrypted_key_file`

Output: `decrypted_data_file, decrypted_signature`

- `./verify_signature public_key_file
decrypted_data_file, decrypted_signature`

Output: Prints Success/Failure On Screen