**Project Report**

**Project Study:**

Our Project goal focuses on encryption/decryption implementation for secure payment application using the Salsa20 algorithm and secure secret key delivery with EC El-Gamal and Schnorr signature.

At first we went to research the three main algorithms with which we will implement the application.

We started reading about each algorithm individually, looking at its code online and doing depth research on it.

**Salsa20 algorithm-** is a modern and efficient stream symmetric cipher that works on data blocks size of 64 bytes. It is considered to be a well-designed and efficient algorithm. There aren't any known and effective attacks on the family of Salsa20 ciphers.

**EC El-Gamal-** is an asymmetric key encryption algorithm for public-key cryptography which is based on the Diffie–Hellman key exchange.

**Schnorr signature-** It is one of the protocols used to implement “Proof Of Knowledge”. The signature proof is supposed to convince the Verifier that they are communicating with a user who knows the private key corresponding to the public key.

In our secure payment application we have implemented the ability of Alice to transfer a certain amount of money to Bob in an encrypted manner and sign the message so that the receiver side will know that the message (amount of money) was sent from a reliable source. Bob receives the message. The encrypted message that Bob received is deciphered by him and gets the original message. The message is verified that the sender is Alice, in a secure way such that nobody can see the original message except the destination user.

**How to make sure that the payment is secure:**

* Each side creates 2 keys: 1)Public key, 2) Private key (total: 4 keys).
* The sender encrypts the message using the Salsa20 algorithm with SalsaKey and nonce. At this point the message is secure.
* To send the encrypted message, you need to secure the SalsaKey.
* SalsaKey is secured by the EL-Gamal algorithm which encrypts the SalsaKey using a receiver public key. At this point the SalsaKey is secure.
* After following two steps above, the sender creates a signature for his plaintext message. This process makes sure that message belongs only to the sender.
* SCHNORR algorithm creates signatures on the plaintext message using sender private key, generator and prime numbers. At this point we secure that the message belongs only to the sender.
* Only the receiver that the message was sent to him, can decrypt the message with his private key, sender public key, encrypted key, encrypt message and the signature.

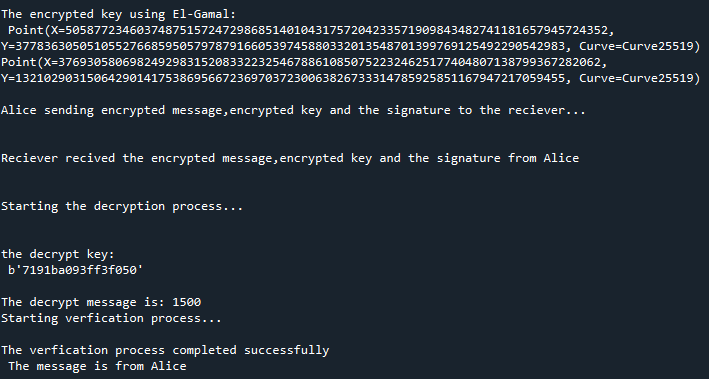
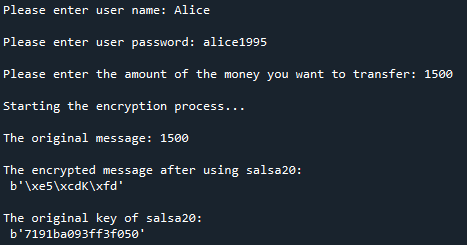
**By the following steps we secure the payment application for both sides.**

**Project Flow:**

**Alice- The sender side (Encryptor) Bob- The received side (Decryptor)**

1. Login to the application by the sender : Alice.
2. Creating 2 keys for each side: one private, one public to Alice.  
   one private, one public to Bob.
3. Alice enters the amount of money(the message) she wants to send to Bob.
4. The application creates randoms private SalsaKey and public nonce.
5. Alice encrypts the message using the Salsa20 algorithm with SalsaKey and nonce.
6. Alice uses Bob's public key and encrypts SALSA's key with EC El-Gamal algorithm.
7. Alice takes the plaintext message and creates a signature using SCHNORR signature with her private key, generator number and prime number.
8. Alice sends to Bob the encrypted message + the encrypted SALSA's key + the signature.
9. Bob receives the encrypted message + the encrypted SALSA's key + the signature from Alice.
10. Bob decrypts the encrypted SALSA’s key with EC El-Gamal algorithm, with his private key.
11. Bob decrypts the encrypted message using the decypher key he received after step 10 and public nonce number with the SALSA20 algorithm.
12. Bob uses SCHNORR signature on the decrypted message with Alice's public key, generator number and prime number to verify that the message comes from Alice.
13. If the verification process is completed successfully , Bob will be sure that the message comes from Alice otherwise not from Alice.

**Obtained Results:**

****

**Conclusions:**

By using a symmetrical salsa 20 algorithm, we encrypt the message quickly and reliably.

It has been proven to be safe against differential attack and also ensures resistance against timing attack.

Furthermore we use El-Gamal decryption ,which is an asymmetric algorithm where the key for decryption is different from the encryption key. So, In this way we guarantee that only the person we want can decrypt the message and see it.

By using symmetrical algorithm(salsa 20) for encryption and asymmetric algorithm (El-gamal), we use a hybrid method that merges two encryption systems. The combination of which creates hybrid coding is considered the most secure type of encryption as long as the public and private keys are completely secure.