**Cryptology Final Project**

**Encryption/decryption for secure payment application using AES algorithm + secret key delivery and signature with EC El-Gamal**

Submitted by: Bradley Feitsvaig 311183073, David Nashash 321522302, Aviv Okun 312426570

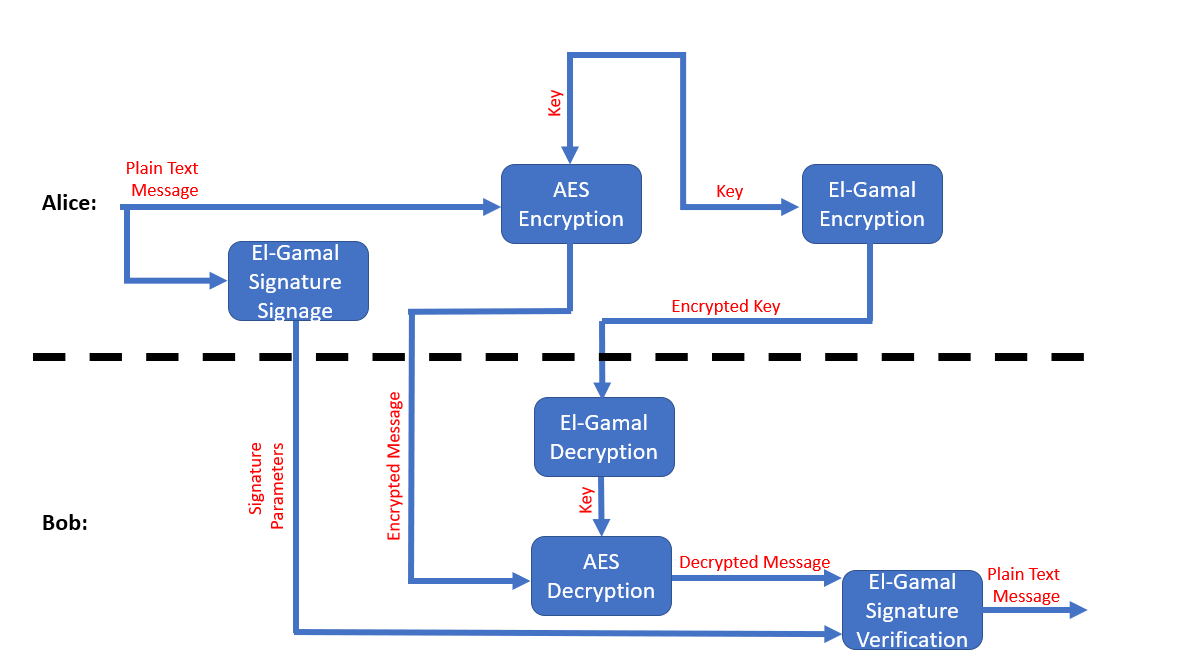
Submitted to Prof. Zeev Volkovich and Dr.Renata Avros.

**Introduction**

In today's world, providing credit card information to pay for various products through the internet has become the norm. From fast food ordering to house furniture purchasing, all payments can be done online. Therefore, finding a way for secure payment methods is a necessity. Our project integrates different kinds of cypher algorithms like AES and El-Gamal in a way that provides a solution for the online payment problem.

**Study process**

The first step of our learning process was to understand the idea behind the encryption process for both the message and the provided key. We depicted a basic scheme that visualized the encryption method for better understanding of the entire process:



After great consideration, we've decided upon signing only the original message sent from Alice because of the following reasons:

1. If the signature was only on the encrypted message, in case of a wrong key and a correct signature verification, there wouldn’t be a way to know if the decrypted message is right or wrong.
2. In case of signing the encrypted message, because of the previous reason, the key must be signed too, what will lead to double signage.
3. Signing both the encrypted message and the encrypted key will lead to over complicated code because of the need for two signing functions specially modified for both the encrypted message and the encrypted key.
4. For our scenario, signing only the original message is enough because, in a case where the encrypted key for AES is corrupted, the encrypted message will be decrypted wrong, therefore, the signature verification will fail, and the final message will be considered as invalid.

One downside of signing only the original message is that in case of anything wrong with the signature verification, it is very difficult to find out exactly where the problem is stems from.

Nevertheless, most of the times the message is expected to be sent correctly, so for the uncommon event in which the message will be corrupted, it can be sent again instead of adding a computational cost for each transaction.

The next step was to understand exactly how AES works and watching the AES Rijndael animation again helped us a lot.

The third step was to find out how to integrate El-Gamal with Elliptic Curve. Regarding the decryption process, we had to choose appropriate Elliptic Curve equation that will fit our model and for the signature process, the signage and verification algorithm was depicted in lecture 9 PowerPoint slides.

**Project flow**

All of the code was done in Python using Spyder5.0

For the Elliptic Curve parameters, we chose **secp256k1** which is a set of parameters used in Bitcoin digital signature. The finite field for the Elliptic Curve domain parameters is defined by:

The curve over is defined by:

So that the final aquation is:

First, we implemented the AES algorithm with 128 bits key and message size, then we made sure that a message can be encrypted and decrypted successfully.

For the El-Gamal using Elliptic Curve implementation, we had to find a way to convert a key to a point on the Elliptic Curve. We addressed the key as an X value on a curve point and then, we calculated the Y value using the Elliptic Curve equation.

We added a signature, and a signature verification functions according to lecture 9 PowerPoint slides.

We used sha256 hash function on the message for the signature and signature verification.

Finally, we implemented the main code using all of the implemented cypher algorithms with the signature and the signature verification functions.

**Obtained results**

For the input 4580 1111 2222 3333 Alice is sending to Bob:

1. The encrypted message, after it went through the AES cypher.
2. C1 and C2 parameters which are the results of encrypting the key with El-Gamal cypher.
3. r and s parameters which are the result of signing the message.

Bob actions after receiving the data from Alice are:

1. Bob decrypts the encrypted key using El-Gamal decryption.
2. Bob uses the key to decrypt the message with AES.
3. Bob verifies the signature on the decrypted message sent from Alice.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

**Conclusions**

Through the work on this project, we managed to better understand how cyphers like El-Gamal and AES works. We've watched the workflow of transferring sensitive data via the internet and the risks involved. The hybrid encryption model of AES and El-Gamal provides more security by increasing the complexity. The AES algorithm is faster than El-Gamal, but the El-Gamal algorithm is more secure than AES. Therefore, the hybrid model that consists of encrypting the plaint text message with AES and the provided key with El-Gamal, is more secure with less computational cost in comparison to using El-Gamal encryption for the plain text message. Furthermore, as was described in the study process, we found that signing the plaintext of the message is more suited to our task.