ANURAG UNIVERSITY

Hybrid Cryptography

Team Details

1. Vishwas(20EG105411)
2. Abhinav(20EG105426)
3. Chaitanya(20EG105437)

Project Supervisor
B Ravinder Reddy
Assistant Professor

Introduction

ANURAG UNIVERSITY

Cryptography and Information Security: Hybrid of two algorithms ECC and diffie-hellman.

A hybrid algorithm that combines Elliptic Curve Cryptography (ECC) and the Diffie-Hellman key exchange can provide a secure way of key generation and exchanging process.

The ECC key exchange provides secrecy and protection against quantum attacks, while Diffie-Hellman offers a well-established method for secure key exchange.

Literature

SI No	Author(s)	Method	Advantages	Disadvantages
01.	Sachin rana , satarupa biswas and Anushika pansari	-ECC algorihm -Diffie hellman	Reduced Encryption and decryption time	Large key size
02	Dr.Vivek Kapoor and Rahul Yadav	-RSA algorithm -DES algorithm -SHA128	128-Bit Key Strength. Secure Data Transmission. Data Integrity	Performance Impact and need high computational power.
03	Arpit Agarwal and Gunjan Patnakar	-RSA algorithm -Diffie-hellman -SHA1	Scalability. Highly Efficienct. Secure key sharing.	Require technical expertise and users need to understand how to use encryption properly.

04	Y Alkady, M. I. Habib and R. Y. Rizk	-RSA algorithm -ECC algorithm -MD5	better performance in terms of computation time and the size of cipher text.	High complexity , might need more resources.
05	Prakash Kuppuswamy and Sayeed Q.Y. Al-khalidi	-Symmetric Key Algorithm - Linear Block Cipher Algorithm.	Better Performance. Enhanced Security. Resistance to Attacks.	Increased implementation challenges. Security of Key Exchange. Potential vulnerabilities

Problem Statement

The combination of Diffie-Hellman acts a secure key transmission agent for the RSA and the RSA accounts for the security of message. However, this method has a very huge key length which effects the performance for the system.

Objective

To overcome this problem to ensure improved data security with reduced key size , we use hybrid algorithm of ECC with Diffie Hellman.

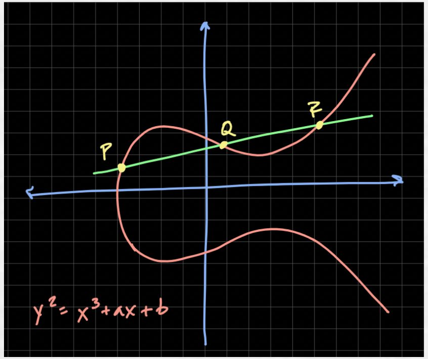
The parameters improved are:

- encryption time
- decryption time
- key size

Proposed Method

ECC, an alternative technique to RSA, is a powerful cryptography approach. It generates security between key pairs for public key encryption by using the mathematics of elliptic curves.

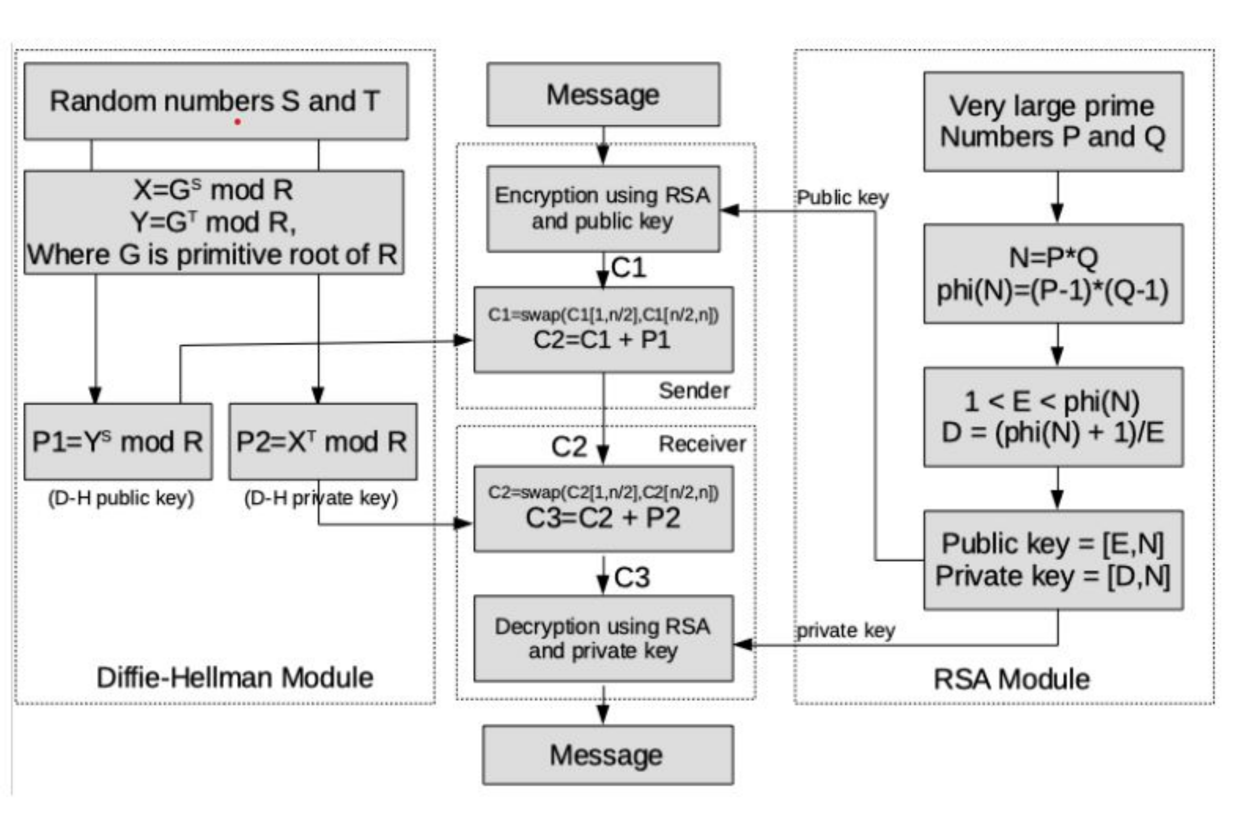
- It makes use of elliptic curves.
- The curves are symmetric to x-axis.
- A line is drawn at any random place on the curve , the points where the line touches are taken as public and private values



Proposed Method

Start
Generate ECC Key Pair (Alice)
Select ECC Parameters (Curve, Base Point)
Generate ECC Key Pair (Bob)
Alice and Bob exchange public keys
Alice computes Shared Secret Key
Bob computes Shared Secret Key
Shared Secret Key is now established
End

FlowChart



Random numbers S and T

$X=G^S \bmod R$
$Y=G^T \bmod R$,
Where G is primitive root of R

$P1=Y^S \bmod R$ (D-H public key)

$P2=X^T \bmod R$ (D-H private key)

Diffie-Hellman Module

Message

Encryption using RSA and public key

C1

C1=swap(C1[1,n/2],C1[n/2,n])
C2=C1 + P1

Sender

C2

Receiver

C2=swap(C2[1,n/2],C2[n/2,n])
C3=C2 + P2

C3

Decryption using RSA and private key

Message

Very large prime Numbers P and Q

N=P*Q
phi(N)=(P-1)*(Q-1)

1 < E < phi(N)
D = (phi(N) + 1)/E

Public key = [E,N]
Private key = [D,N]

Public key

private key

RSA Module

The ECDH (Elliptic Curve Diffie–Hellman Key Exchange) is anonymous key agreement scheme, which allows two parties, each having an elliptic-curve public–private key pair, to establish a shared secret over an insecure channel. ECDH is very similar to the classical DHKE (Diffie–Hellman Key Exchange) algorithm, but it uses ECC point multiplication instead of modular exponentiations. ECDH is based on the following property of EC points:

(a * G) * b = (b * G) * a

If we have two secret numbers a and b (two private keys, belonging to Alice and Bob) and an ECC elliptic curve with generator point G, we can exchange over an insecure channel the values (a * G) and (b * G) (the public keys of Alice and Bob) and then we can derive a shared secret: secret = (a * G) * b = (b * G) * a. Pretty simple. The above equation takes the following form:

alicePubKey * bobPrivKey = bobPubKey * alicePrivKey = secret

The ECDH algorithm (Elliptic Curve Diffie–Hellman Key Exchange) is trivial:

1.Alice generates a random ECC key pair: {alicePrivKey, alicePubKey = alicePrivKey * G}

2.Bob generates a random ECC key pair: {bobPrivKey, bobPubKey = bobPrivKey * G}

3.Alice and Bob exchange their public keys through the insecure channel (e.g. over Internet)

4.Alice calculates sharedKey = bobPubKey * alicePrivKey

5.Bob calculates sharedKey = alicePubKey * bobPrivKey

6.Now both Alice and Bob have the
same sharedKey == bobPubKey * alicePrivKey == alicePubKey * bobPrivKey

Project status

Sl. No	List of Functions	Status
01	Algorithm Building	Completed
02	Preparing the flowchart	Completed
03	Ecc value pair generation	In progress
04	Public variables digest	In progress
05	Key Generation using diffie-hellman	Not Yet Started
06	Encryption of the plain text	Not Yet Started
07	Decryption of the cipher text	Not Yet Started

References

1. Sachin rana , Satarupa biswas and Anushika pansari,"Hybrid Cryptography Algorithm For Secure And Low Cost Communication" , International Conference on Computer Science Engineering and Applications(2020)

2. Dr.Vivek Kapoor and Rahul Yadav, "A Hybrid cryptography technique for increasing