An Implementation of Elliptic Curve Cryptography Algorithm for Secured Data Transmission In Wireless Sensor Network

SUBMITTED BY:

Binay Pradhan (B415019)
Supratik Samal (B415059)
Pritish Gupta (B415039)
Rudra Narayan Panda (B415045)

INTRODUCTION

- Major issue with use of public key is the size of numbers used.
- ECC belongs to the category of Public-key Cryptography, performs the computations using elliptic curve arithmetic instead of integer or polynomial arithmetic.
- ECC provides equally good security compared to RSA, but uses smaller key size(160-256bit VS 1024-3072bit).
- Notable Advantages of ECC
  - Uses smaller keys, cipher texts and signatures.
  - ECC supports, very fast key generation.
  - ECC scores over RSA because of its moderately fast encryption and decryption.
  - ECC computations are uses less memory and CPU cycles compared to RSA, hence suited for securing Mobile Handheld devices.

PROPOSAL

- This paper centers around the execution of the scalar point multiplication, and this is the basic task of public key Elliptic Curve Cryptography(ECC).
- This cryptosystem has high security furthermore, great execution offered by ECC, compared to other options, for example RSA.
- Besides, this task can be utilized in digital signature(ECDSA), establishment of key(ECDH) and various encryption/ decryption(ECIES) protocols.

OVERVIEW

- Implementation of RSA and ECC algorithm
- Performance evaluation
- Simulation
    - Packets sent
    - Packets received
    - Packets Lost

ECC ALGORITHM

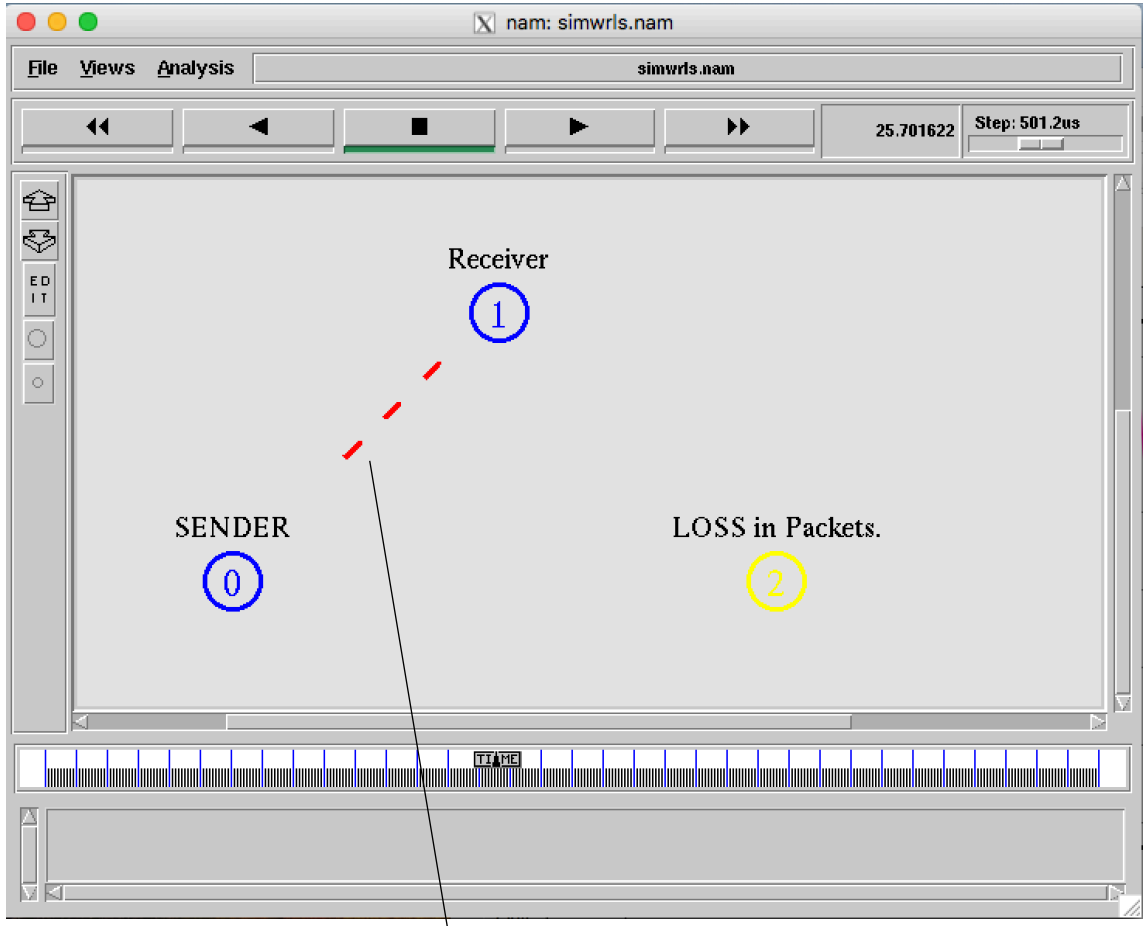
Algorithm for Encryption

1. Enter a prime no p which is used to keep the points in GF (n=1)
2. Enter curve parameters a and b
3. Collect the points(y*y) which are quadratic residue of (x*x*x)+(a*x)+b
4. Enter e1
5. A random number d is generated which is private key
6. Find public key e2=d*e1
7. Enter the message P
8. A random number r is generated for encryption
9. Find c1=r*e1
10. Find c2=P+(r*e2)
11. c1 and c2 are the encrypted points

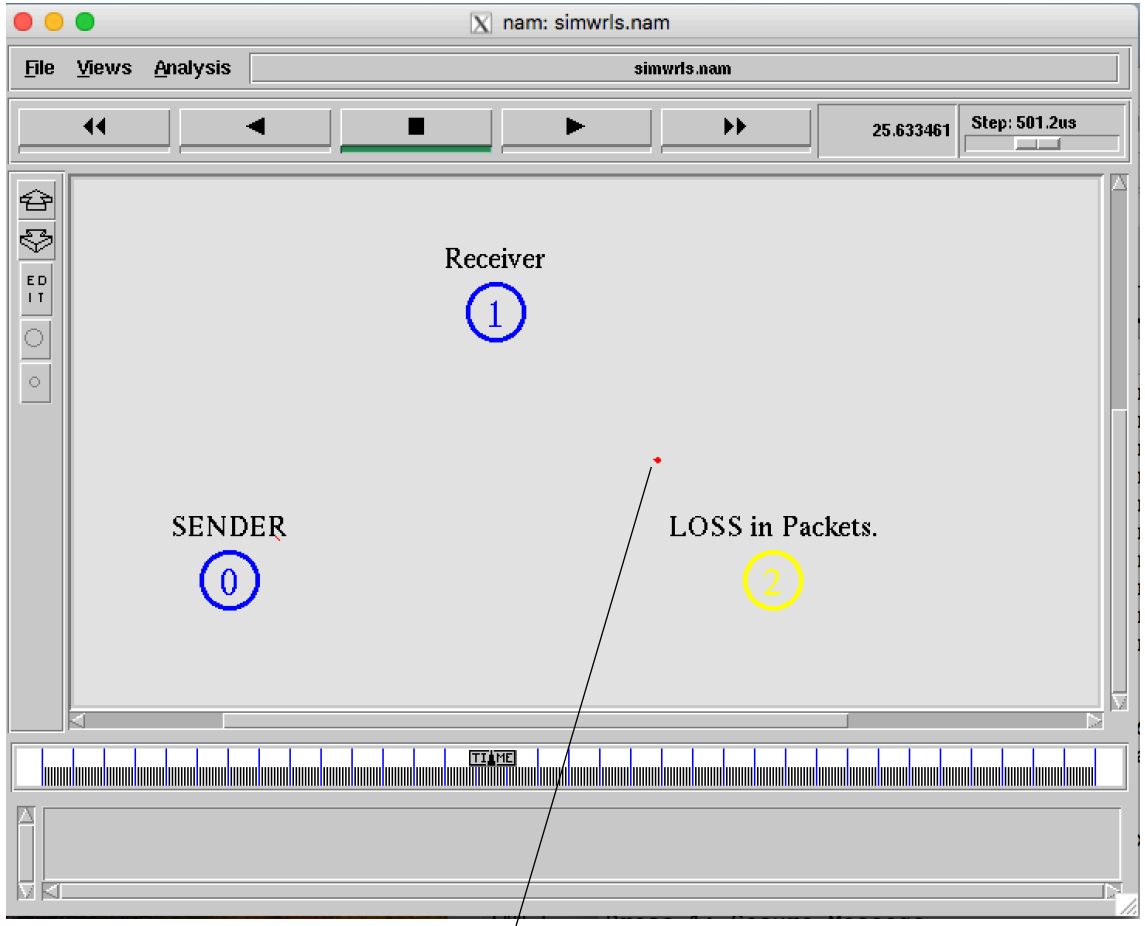
Algorithm for Decryption

1. Enter encrypted points c1 and c2
2. Enter curve parameter a and b to form the same curve as encryptor
3. Enter the prime no p to keep points in GF (n=1)
4. Enter private key d
5. Decrypted message (P) = c1-(d*c2)

RESULTS



Packets Sent



Packet Loss

GRAPH ON PACKET TRANSFER