Computer Network Security Project 2 Report

by Shen Lu

1. Introduction

Nowadays, cell phone becomes the most common communication device for people to connect to each other. However, the security of the communication becomes a big issue. If the communication is not secured properly, eavesdroppers can hijack the communication and listen to the conversation or read messages.

In this project, we created a secure messaging tool which can be used to send and receive encrypted messages so that messages are only readable for the two parties of the conversation and not readable for anybody else.

The application focuses on three major issues: one is the user authentication, another is key management and the last one is message encryption and decryption. For user authentication, we let the initiator to provide the secret key first so that whoever does not have the secrete key can not take part in the conversation. For key management, since there is no shared database for the point-to-point messaging tool, we use key exchange to do key management. Before each session is initiated, we do Diffie-Hellman key exchange to create a session and then we let the two conversation parties use the session key to encrypt and decrypt messages. For encryption and decryption, we use triple DES algorithm to secure the messages which can make sure the key has 56 bits.

2. Methodology

    1. Authentication

First of all both of the two conversation parties know the shared secret, the way they authorize each other is by checking if they are the person who knows the secret. There are several different ways to do authentication, such as password-based authentication, address-based authentication, and cryptographic authentication. For password-based authentication, it may suffer from dictionary attack. For address-based authentication, it may suffer from the fake source address attack. Cryptographic authentication is better than other authentication methods, because, we can take advantage of the encryption algorithms to make it very hard to break.

In this project, we use secrete key and random numbers to help do authentication. The authentication process can be completed with two message.

    2. Key exchange

For key exchange, we use Diffie-Hellman key exchange algorithm.both of the two conversation parties take part in the key exchange. The combination of the two keys from both of the two parties will become the secret key which can be used to encrypt and decrypt all of the messages for one session. The way to do the key exchange is that each one of the two parties generates a session key and then they send the session key to the other party, so that both of them have two session keys. Since both of them can put the two session keys into the key generating function which is also shared between both of them, they can generate the same secret key by using the two session keys.

    3. Encryption and Decryption

For encryption and decryption algorithm, we choose triple Data Encryption Standard (DES), Cipher Block Chain(CBC) with PKCS5Padding.

DES algorithm uses a 56-bit key and maps a 64-bit input block into a 64-bit output block. The key actually looks like a 64-bit quantity, but one big in each of the 8 octets is used for odd parity on each octet. Therefore, only 7 of the bits in each octet are actually meaningful as a key.

DES algorithm includes three steps: the first step is the initial permutation, the second step is the 16 round encryption and the last step is the final permutation. For the encryption step, the 56-bit key will be used to generate sixteen 48-bit per-round keys. In each round, one of the 48-bit per-round key will be chosen to do encryption. In figure 1, we show how DES algorithm is structured.

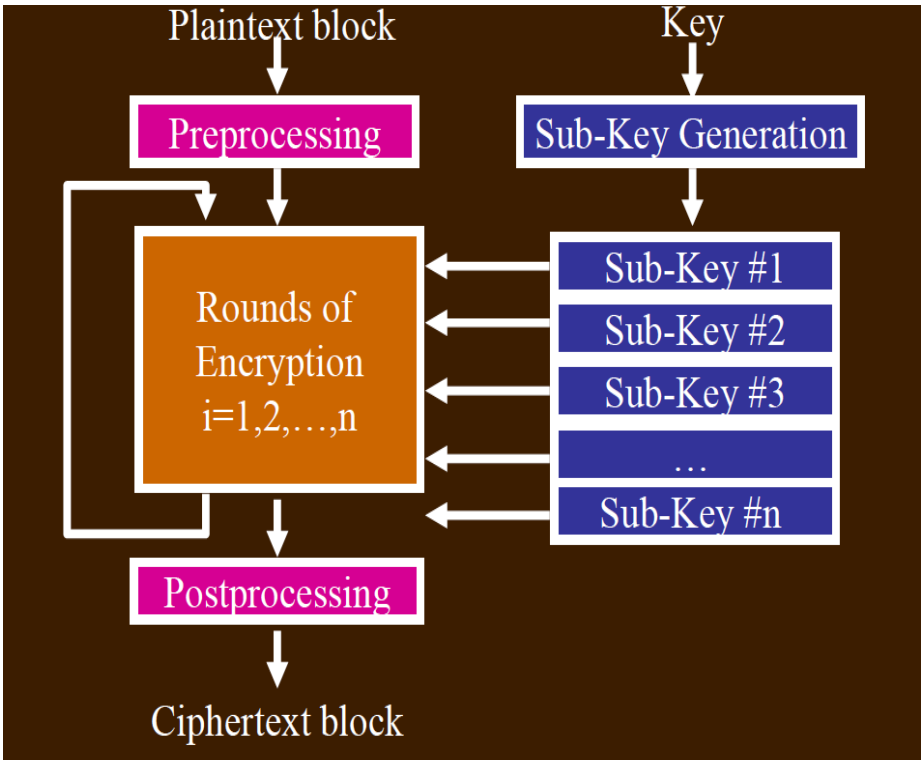


Figure 1. DES Algorithm

Triple Encryption can effectively avoid the meet-in-the-middle attack. The way it works is shown in figure.

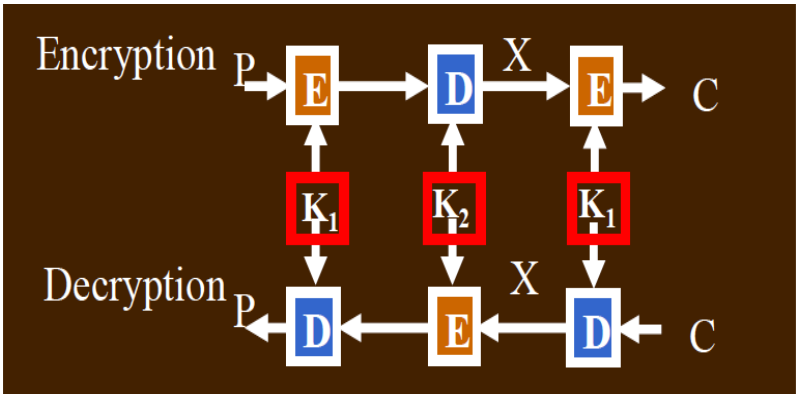


Figure 2. Triple DES Algorithm

Cipher Block Chaining algorithm can avoid the same block with the same plaintext generating the same cipher text. CBC algorithm generates a 64-bit number r_i for each plaintext mi to be encrypted. XOR the plaintext block with the random number, encrypt the result, and transmit both the unencrypted random number r_i and the cipher text block c_i. To decrypt this, we need to decrypt all the cis, and for each c_i, after decrypting it, we need to XOR it with the random number r_i. In figure 3 shows how CBC works.

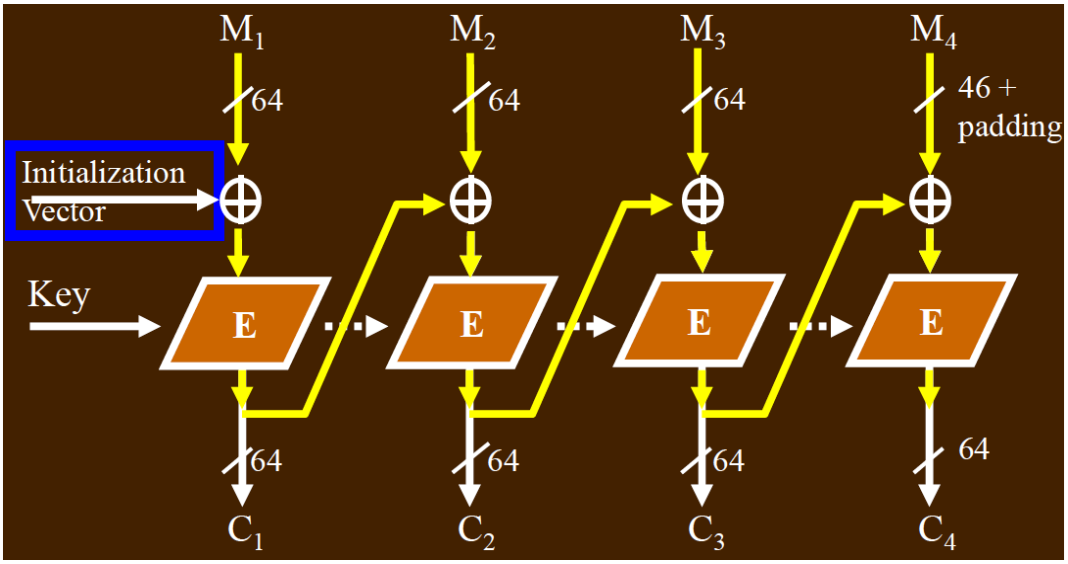


Figure 3. Cipher Block Chaining (CBC) Algorithm.

3. Implementation

1. Authentication

For authentication, each time we run the application, a random number will be generated and the random number will be encrypted with the key generated by using the shared secret. The encrypted random number would be sent to the other part. After the other party receives the random number, he will use the secrete key which is generated by using the shared secrete to decrypt the message and get the value of the random number. He will add one to the random number, encrypt the number with the secrete key which is generated by using the shared secrete, and also send the encrypted new number to the initiator. After the initiator receives the message, he will decrypt the message with the secrete key and compare the new number with the random number he generated. If the new number is equal to the random number plus one, the other party is authorized. Otherwise, the other party can not be authorized. The authentication process is shown in figure 4.

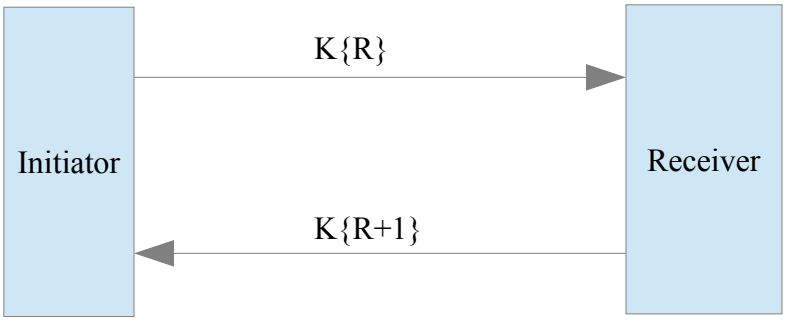


Figure 4. Authentication

2. Key exchange

For key management, normally, we need to maintain a key list or a key database which can be used to store all of the keys so that we can change keys periodically. However, since this is a point-to-point messaging system, there is no online server involved. The easy way to update the secrete key is to do key negotiation before the session starts. The way we do key negotiation is by using Diffie-Hellman algorithm which is shown in figure 5.

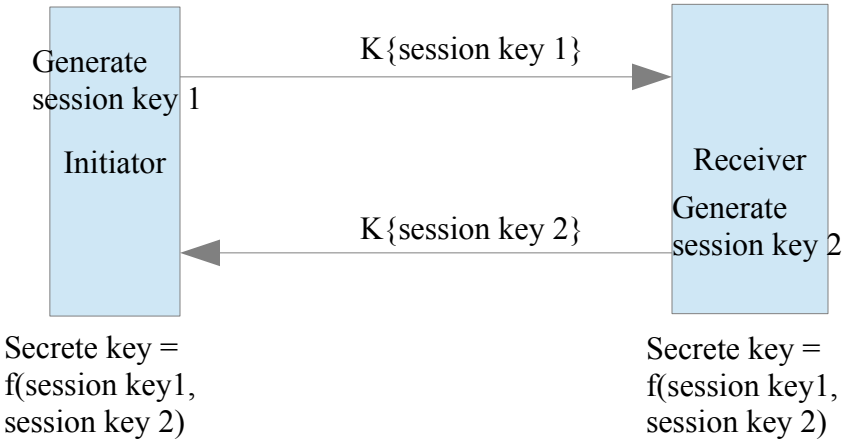


Figure 5. Key Exchange

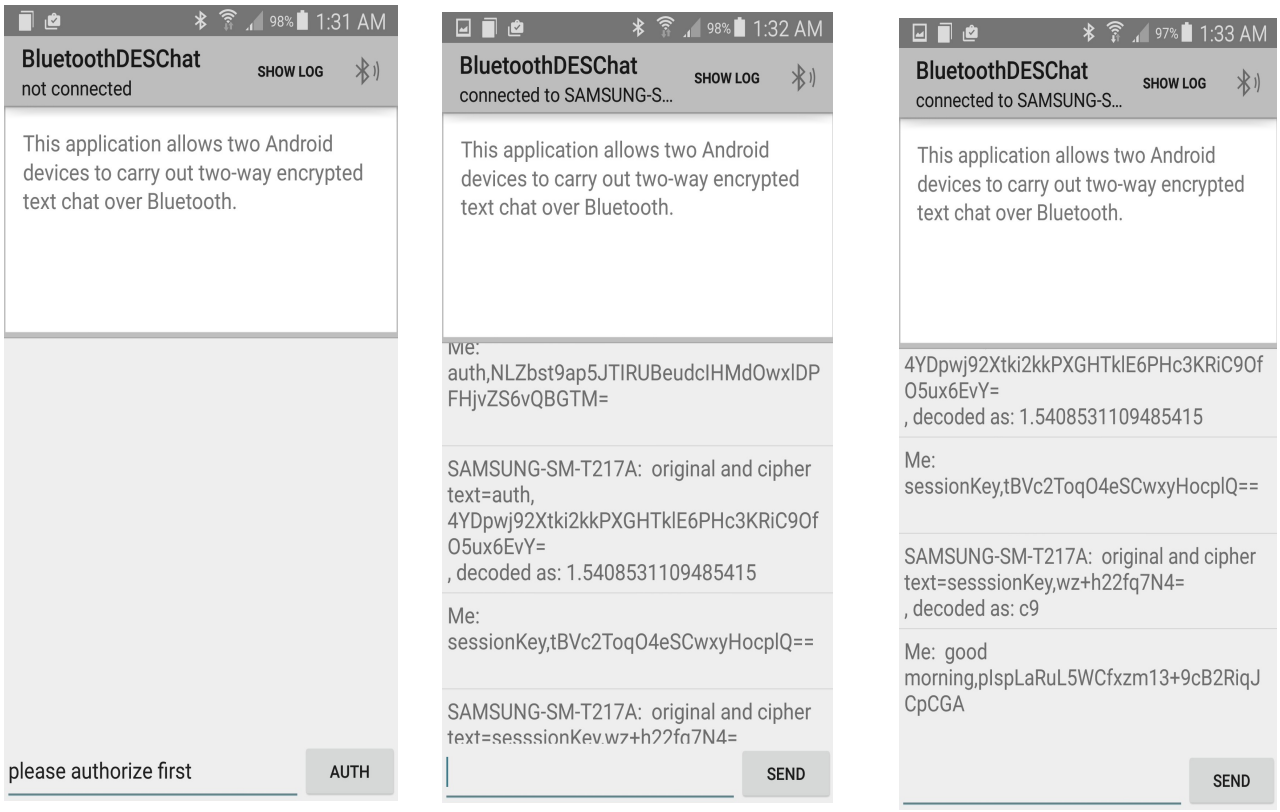
Both of the two parties generate session keys and then send session keys to each other. Since both of them use the same function and the same keys to generate the secrete key as the session key, the keys they generate should be the same and they can use the key generated by themselves to encrypt and decrypt the messages.

3. Encryption and Decryption

For encryption and decryption algorithm, we choose triple DES plus CBC algorithm with PKCS5Padding. Triple DES algorithm can effectively avoid meet-in-the-middle attack, CBC algorithm can ensure the same plaintext have different cipher text.

4. Performance

when the application just starts running, we need to choose the bluetooth device to connect with, which is shown in figure 6(a). We print authentication information and key exchange information on the screen which is shown in figure 6(b). Each authentication includes two message starting with the text “auth” and the random numbers are encrypted with the shared secret key. For key exchange, it takes two messages to negotiate the session key. The two session keys are not directly used as the shared secrete key. We use a shared function plus the two session keys as input to generate the shared secrete key as the session key so that , even though the eavesdroppers may save the traffic, without the knowledge of the shared function, they still cannot figure out what the session key is going to be. When the conversation starts, we print both plain text and cipher text on the screen, which shows that the plain text has been correctly encrypted. When we receive messages, the cipher text will be decrypted into plain text as well.



(a) (b) (c)

Figure 6. (a) the start screen (b) the screen after authentication and key exchange (c) the screen with a message

5. Conclusion

we made a point-to-point bluetooth secure messenger application which use triple DES CBC algorithm with PKCS5Padding to encrypt and decrypt messages. Since Diffie-Hellman key exchange algorithm does not provide the user authentication functionality, we use random number to do authentication before the key exchange starts. For key management, we use session keys instead of the long-term secrete key to encrypt and decrypt messages which can ensure the secrete key would be updated often and even when the previous secrete key is compromised, the conversation is still secure, which is idea of Perfect forward Secrete.