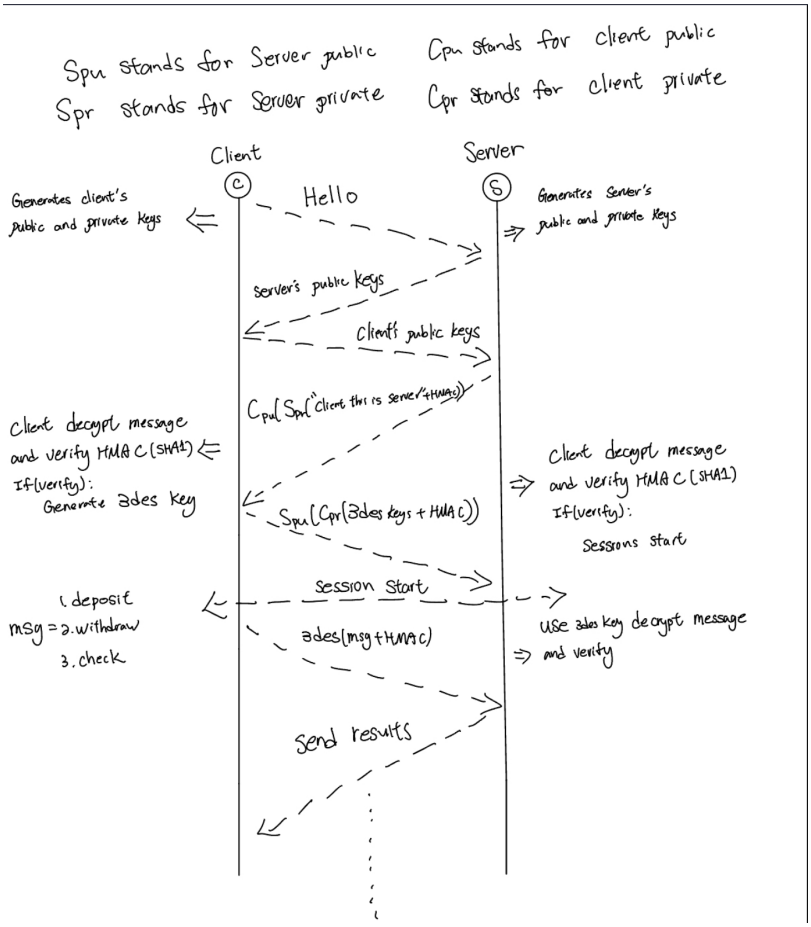
Chien Hsiang (Sean) Hung

Cryptography and Network Security

8/8/2022

# White Hat Report

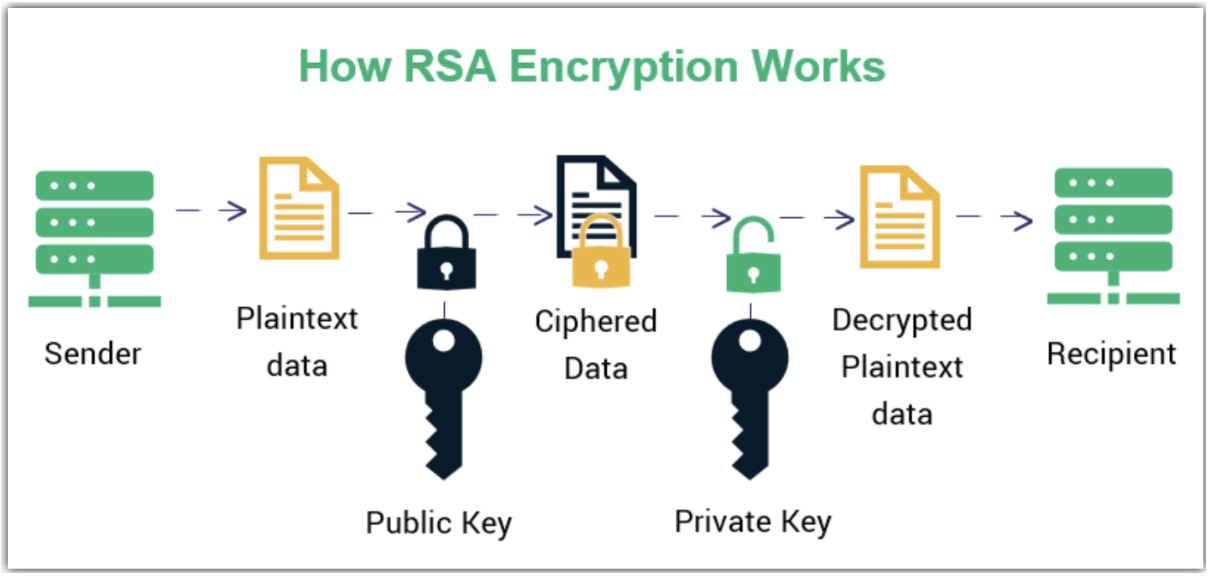
For my report I will explain the secure transport protocol that we design to protect the communications between an ATM (client) and a bank (server). My team consisted of Nick Pardave and Sam. The system will allow the user to deposit money, withdraw money, and check their current balance from the ATM.




The graph shows the overall architecture of our system. The protocol and encryption algorithm that we had implemented were using TCP connection, RSA public and private key encryption, Triple Des ECB, and for MAC we used HMAC, for hash functions we used SHA1.

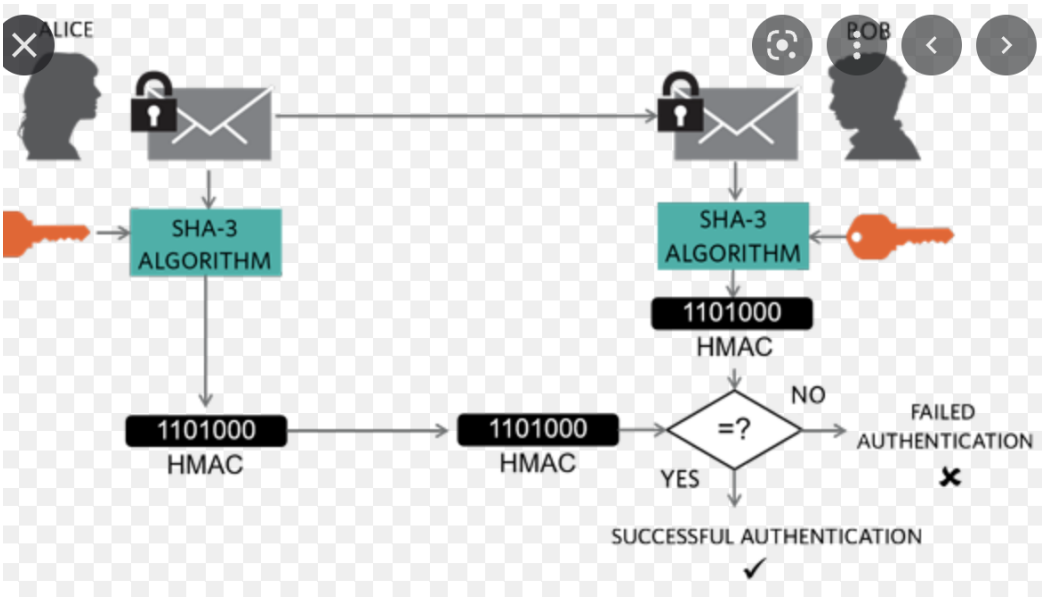
When a socket is initiated the client and server will generate their own public key and private key with the RSA algorithm. RSA is short for the Rivest-Shamir-Adleman algorithm. The reason we choose RSA is because it is

currently the most commonly used public key encryption algorithm.




The principle of the RSA public key cryptosystem is that it is relatively easy to find two large prime numbers, but it is very difficult to factor their product so that the product can be used publicly as an encryption key. The security of RSA depends on the decomposition of large numbers, but whether it is equivalent to the decomposition of large numbers has not been theoretically proved, nor has it been theoretically proved to be deciphered. The difficulty of RSA is equivalent to the difficulty of factoring large numbers. Because there is no proof to crack RSA, large number factorization must be done. The security strength of the RSA algorithm increases as its key length increases. However, the longer the key, the longer it takes to encrypt and decrypt. Therefore, it is necessary to comprehensively consider the sensitivity of the protected information, the cost of cracking by an attacker, and the required response time of the system. However, since all calculations are large numbers, whether implemented in software or hardware, RSA is the fastest case several times slower than DES. Speed has always been a shortcoming of RSA. Usually only used for small amounts of data encryption. RSA is about 1000 times slower than the corresponding symmetric encryption algorithm for the same level of

security. After the client generates the public key and private key it will send the message to server(Bank) “ Client: Hi Bob, This is Alice”, After the server has connected to the client it will show the IP address and port number of the client. Now, the server will send its own public key to the client. Now both the client and server have each other's public key, which means we can start the encryption. The server then encrypts the "client, this is server" and mixes it with HMAC then encrypts it with their private key, then encrypted with the client's public key, and sends the cipher text to the client. The client then decrypts the message using their private key and verifies the HMAC that the current message is not corrupted using an HMAC implementation of SHA-1. If the verification is successful, the client generates a 3DES key. The client then sends through the 3DES key with HMAC and encrypts it with their private key, which is then encrypted with the server's public key.




(The idea is about the same, instead we are using SHA1 instead of SHA3)

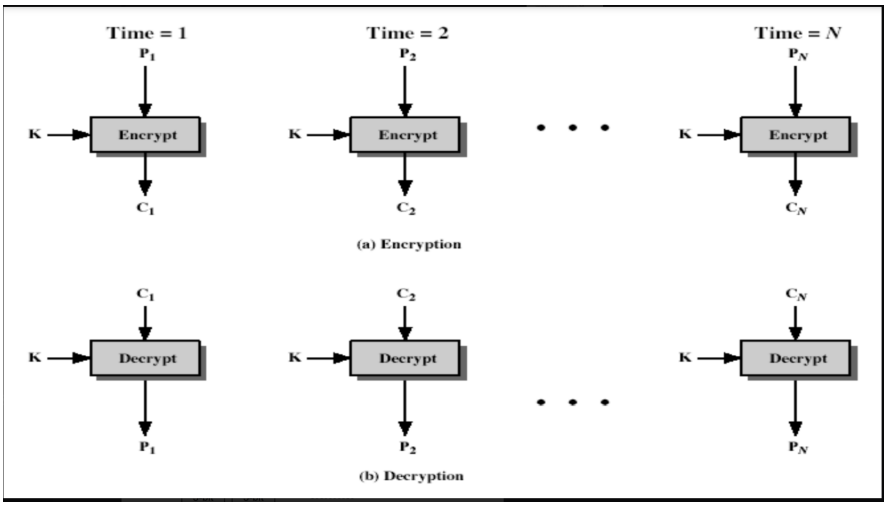
The HMac plays an important role when sending the message. Hmac is mainly used in authentication, and its use process is as follows: 1. The client sends a request .

2. The server returns a random value and records this random value in the session

3.The client uses the random value as the key, performs hmac operation on the message , and then submits it to the server.

4. The server reads the message and the random value sent in step 2, performs the same hmac operation as the client. The sent results are compared, and if the results are consistent, the user is authenticated.

In this process, the random value sent by the server and the hmac result sent by the user may be attacked. For the hacker who intercepts these two values, these two values are meaningless and will never obtain the user's password. The introduction of random values makes hmac valid only in the current session, greatly enhancing security and practicality.




Instead of using DES/SDES we choose to use 3DES to encrypt the message. 3DES, also known as Triple DES, is a mode of the DES encryption algorithm, which uses three 56-bit keys to encrypt data three times. The Data Encryption Standard (DES) is a long-standing encryption standard in the United States that uses

symmetric key encryption and was standardized by the ANSI organization in 1981 as ANSI X.3.92. DES uses a 56-bit key and cipher block method, and in the cipher block method, text is divided into 64-bit sized text blocks and then encrypted. 3DES is more secure than the original DES.

3DES (i.e. Triple DES) is an encryption algorithm that transitions from DES to AES (in 1999, NIST designated 3-DES as a transitional encryption standard). The specific implementation of the encryption algorithm is as follows: Let Ek() and Dk() represent The encryption and decryption process of the DES algorithm, K represents the key used by the DES algorithm, P represents the plaintext, and C represents the ciphertext, so: The 3DES encryption process is: C=Ek3(Dk2(Ek1(P))), 3DES decryption The process is: P=Dk1(EK2(Dk3(C))). After safely sending the 3DES key the session between server and client started. Throughout the session, clients will encrypt their messages including HMAC and using 3DES to encrypt then send them to the server. The server will decrypt the messages using the 3DES key previously received by the client. At the same time, the server verifies that the message is corrupted using HMAC and sends the result if the verification is successful. Thus, the server and the client are able to communicate within a secure network between each other.