UNIVERSITÀ DI PISA

Department of Computer Engineering

Master’s degree program in Cybersecurity

Applied Cryptography course - Project Report

A Secure Bank Application

Professor:
Prof. Gianluca Dini

Students:

Claudio Costanzo
Calogero Costa

AY 2022-2023

Table of contents

1. Project Specification 1
2. Design Choices 2
3. Protocol Schemes 4

Figure Index

Figure 1 – PFS Protocol scheme.. 4

Figure 2 – Communication Protocol scheme 4

1. Project Specification

Develop a Secure Bank Application (SBA), that is a client-server application that allows users to issue operations on their own bank accounts.

Each user is modeled by username and a password whereas each bank account by an accountID and a balance. For simplicity we assume that a user owns a single account and that an account is owned by a single owner.

A user may issue the following operations on its own bank account:

• ***Balance () → (accountID, balance)***
The function returns the user's bank account's accountId and balance.

• ***Transfer (username, amount):***
The function the username of another user and an amount of money as input parameters and transfers such an amount from the user's bank account to the other user's bank account. The operation returns false if the value of balance is smaller than the value of amount; it returns true otherwise.

• ***History () → (TransactionList)***
The function returns the last T transfers performed by the user, where T is a system configuration parameter. Each transfer is a triple (user, amount, timestamp).

The SBA application is hosted by a server which maintains users and accounts. Users interact with the SBA server through a secure channel that must be established before issuing operations.

Requirements

1. Each principal owns a pair of private and public keys.
2. The SBA server maintains the public key of every user.
3. Each user maintains the public key of the server.
4. Users are initialized at registration time which is off-line.
5. OpenSSL API TLS cannot be used.
6. The authentication protocol must fulfill Perfect Forward Secrecy (PFS).
7. Client-serve communication must fulfill confidentiality, integrity, no-replay, and non-malleability.
8. Password and history of transfers of any given user cannot be stored in the clear.

2. Design Choices

Username and Password

- The username of the User must be in lowercase, with no digits nor special characters.
- Usernames uniquely identify users. For this reason, each username must be unique. Username’s length must be at max 20.
- Each user can own a single bank account, and each bank account is owned by a single user. For this reason, the bank account id must be unique.
- Passwords must include at least 1 uppercase, 1 lowercase, 1 special character and 1 digit. Passwords’ s length must be at least 8.

Public/Private Keys and Symmetric Key

- At registration time, (that is offline), Server and Client exchange their Public Keys. So the user goes physically to the bank and do the registration procedure. We created different files named “*username/secret.pem*” to store the Private Keys of the users. Same for the Server. This is done to “simulate” that each entity of the SBA stores its Private Keys in a secure way (at home for instance)
- Server’s Public Key is in a shared directory named *PublicKeys,* in a file named “*Server.pem*”, to make sure that each user can access the Server’s Public Keys. In this directory there are also the Users’s Public Keys.
- The Symmetric Key used by the Server to encrypt and decrypt the transaction list is stored in a file named “*DBkey.txt*”.

Encryption and Decryption

- For the Authentication Protocol *RSA* is used to share the *DHE Public Key* in a secure way.
- For the Communication Protocol we used *AES-GCM* using the session key established in the Authentication Protocol. Session key size is 16 bytes. The IV size is 12 bytes while the Tag size is 16 bytes. They are sent in clear, concatenated with the ciphertext, and newly generated every (encryption-decryption) pair.
- For encrypting and decrypting the transaction list, we used *AES_GCM* with the Symmetric Key stored in the file “*DBkey.txt*”, while the IV and Tag are stored in the file “*T-IV.txt*”

Additional Functions

- **Deposit(amount)**

  This function takes as parameter a positive integer amount and add it to the current balance of the user himself. Then the balance is updated with the new one.

- **Logout**

  This function is used to logout after the user finish the desired operations.

- **Exit**

  This function is used to close the Client application.

- **Help**

  This function is a local function, that the user can type if he wants additional information on how to interact with the client interface.

3. Protocol Schemes

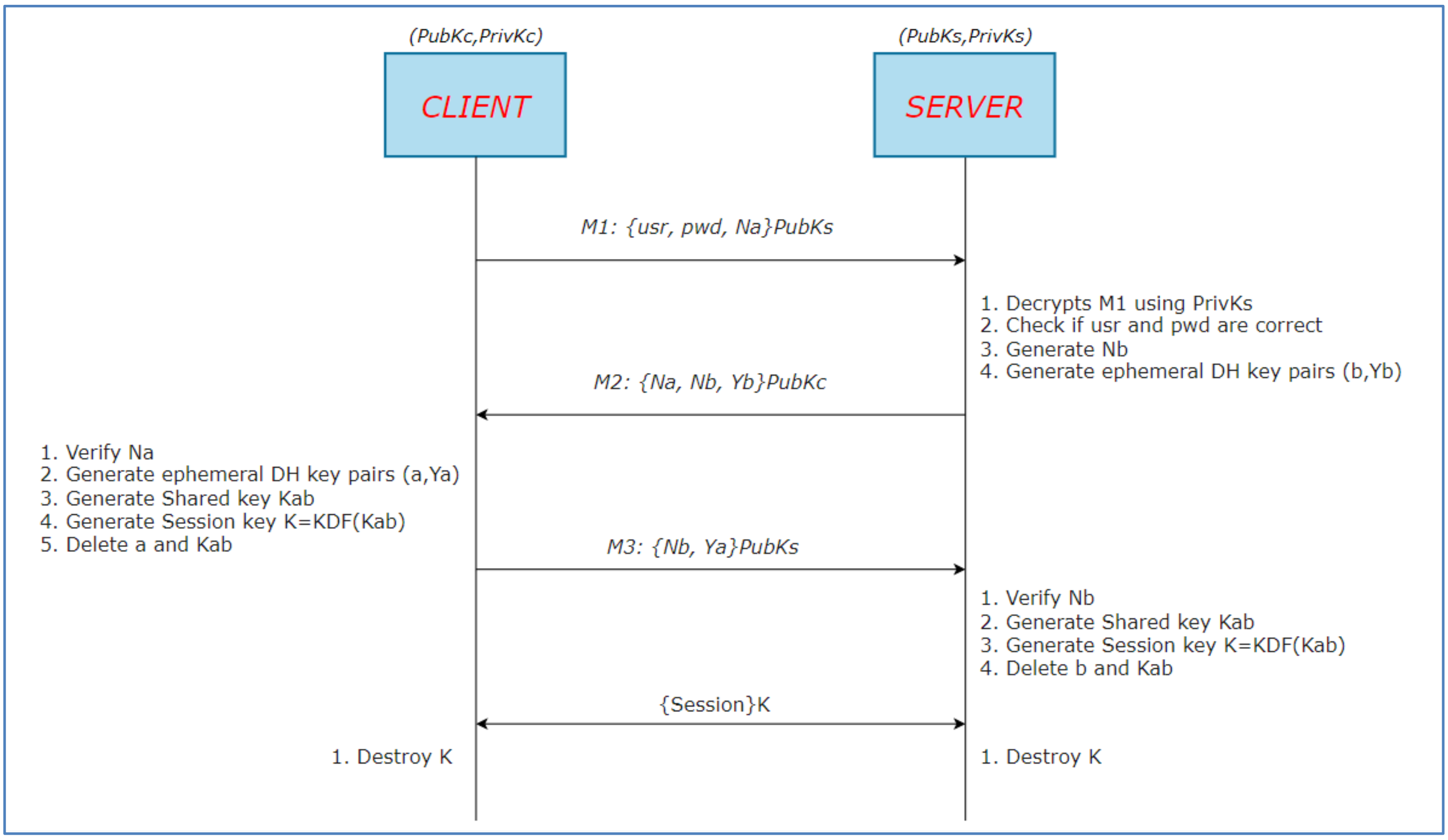
(PubKc,PrivKc)
CLIENT
(PubKs,PrivKs)
SERVER
M1: {usr, pwd, Na}PubKs
1. Decrypts M1 using PrivKs
2. Check if usr and pwd are correct
3. Generate Nb
4. Generate ephemeral DH key pairs (b,Yb)
M2: {Na, Nb, Yb}PubKc
1. Verify Na
2. Generate ephemeral DH key pairs (a,Ya)
3. Generate Shared key Kab
4. Generate Session key K=KDF(Kab)
5. Delete a and Kab
M3: {Nb, Ya}PubKs
1. Verify Nb
2. Generate Shared key Kab
3. Generate Session key K=KDF(Kab)
4. Delete b and Kab
{Session}K
1. Destroy K
1. Destroy K

Figure 1:Authentication Protocol Scheme

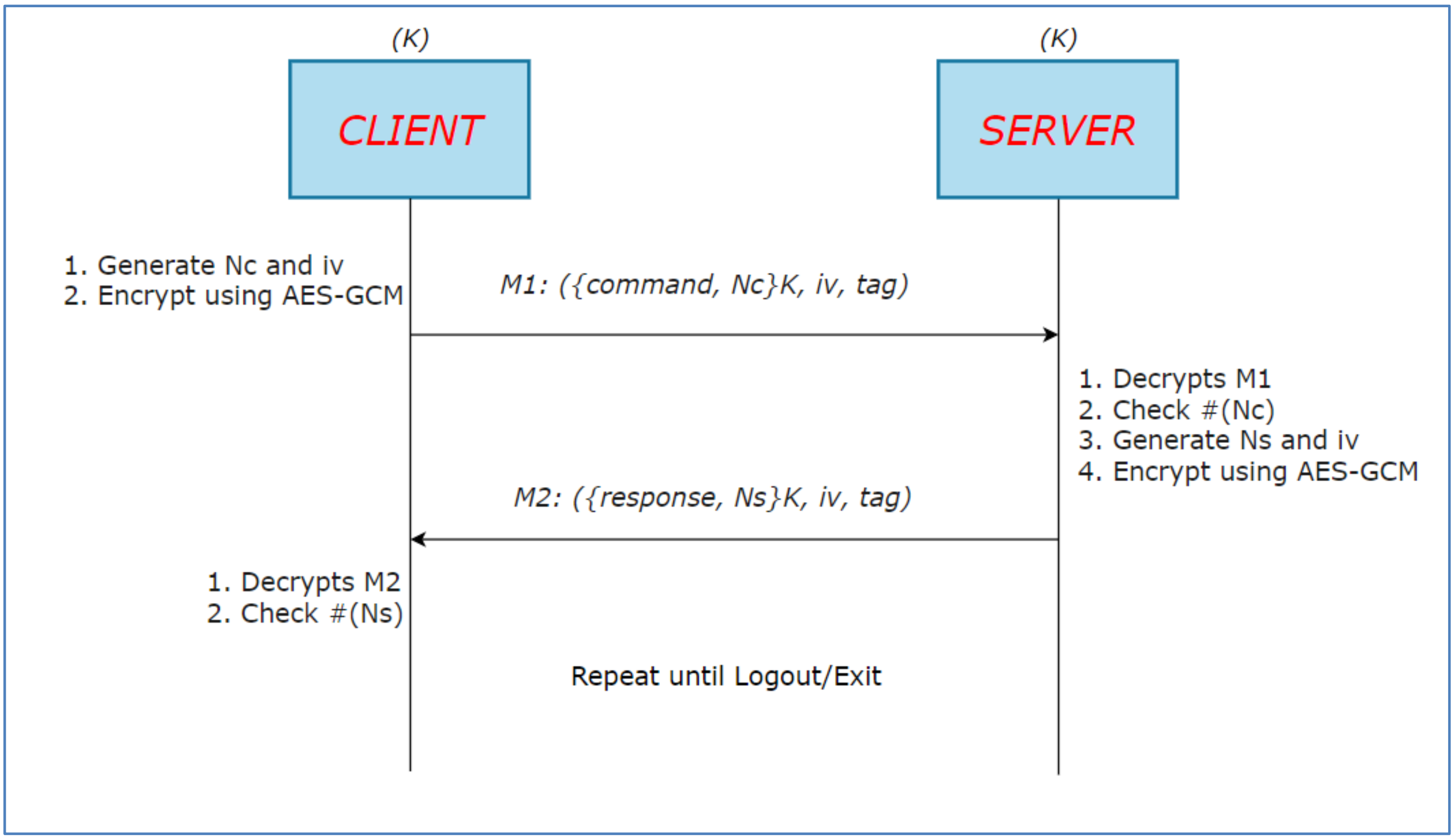

Figure 2: Communication Protocol Scheme