1. Network Security

a. Consider the security of the above system, discuss three potential security issues and provide countermeasures. For each of the issues, specify the related security service(s), attack(s) and mechanism(s). The demonstrated issues must not relate to the same security service(s). (3 marks)

Customers using a mobile network or Wi-Fi connection to make transactions may be at risk of eavesdropping, communication jamming, cryptographic threats and mainly modification of data and rogue attacks. A user may be fooled into linking to a rogue access point by an adversary and transmits information to make their own login by that user. Wi-Fi protected access (WPA) is an IEEE architecture protocol used to ensure wireless security. It can authenticate a user's strong encryption and authentication measures.

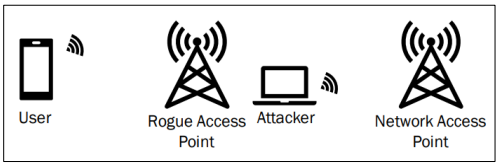


Figure 1: A user may unknowingly be accessing a rogue access point.

The bank's servers can be a victim to attacks such as user impersonation, network address impersonation, eavesdropping and replay attacks. Most common attacks involve a user's dishonesty to the bank to be something or someone they are not. Kerberos is a widely known and used protocol design with two barriers of entry to grant a user access to the primary server's services. This access is dependant upon the server authentication to verify a user and a ticket granting server to permit the user into the system for a period of time.

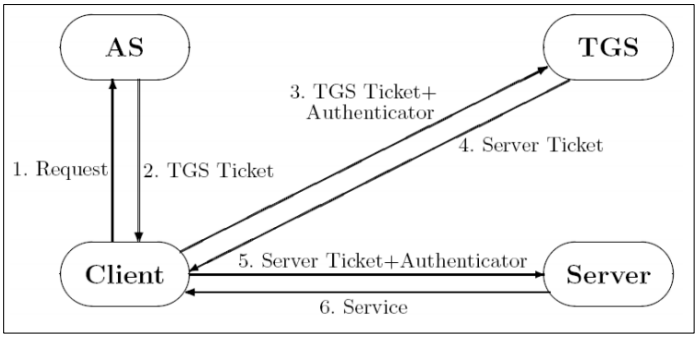


Figure 2: The Kerberos architecture showing how a user would have access to the bank's server.

Travelling employees may be subject to recall attacks and threats across the internet. To combat this, internet protocol security measures can be implemented as an additional network layer. Employees can access the bank's services as work without exchanging confidential information across the internet with the IPSec security protocol. However, it is necessary that both the user's device and the bank's server must have IPSec in their devices.

b. Consider that a bank employee requests to modify a bank customer's daily cash transfer limit. Briefly describe the essential security-related step(s) that demonstrate the security checks for the operation. For each step, specify the aimed security service(s). (3 marks)

For an employee to modify a customer's daily cash transfer limit, the bank must ensure that the request made is a secure operation. The employee must not be vulnerable to external or inside attacks by adversaries.

Firstly, the bank must have a sturdy authorisation mechanism to enduring the data in the system is protected from any unauthorised access. No random person must have the ability to change a customer's transfer limit. For better authentication, the bank must use two levels of authentication. The first level being the password and the second being the employee's personal information such as passport or identification number. The password must also have a decent length and a form of complexity (special characters, upper-case, numbers, etc). Furthermore, the password duration must also have its lifespan for the employee to use the service to change the transfer limit of the customer to lower risk of adversaries compromising the password. Biometric identification is also a great suggestion in identifying a person's physical feature such as retina or thumb print.

Secondly, another feature the bank should have is the ability to regulate an employee's activities to be recorded for analysis and develop additional security measures. Any irregularities would trigger a response protocol in which the administrators would need to verify if the employee is updating a customer's transfer limit to an amount outside of the usual amount.

Finally, any leakage of data to the customer must be kept confidential to the customer only. As the updated information about the customer's daily cash transfer limit has been changed, the customer must be notified which is primarily done via an email the bank would send. An outgoing protocol can be used scanned on all types of http, TCP and FTP. These features help scan the data in email bodies and attachments, reducing accidental data breaches.

c. An employee accesses the internal system with proper authentication and authorisation. Consider Kerberos, SAML, and OAuth, which one is better for internal system authentication and authorisation? Justify your answer. (2 marks)

Synchronous KerberosV5 would be the better system authentication and authorisation than SAML and OAUTH 2.0 as it is primarily implemented to prevent attacks such as impersonation, eavesdropping and replay attacks. This is especially important in a bank server as attacks can cause catastrophic loss for the company. Kerberos is widely accepted by the industry and has many forms of its protocol. Once the client (employee) has authorised itself into the bank's server, it can choose which server to work on. Despite SAML's web browser authorisation and authentication, it lacks security issues such as mutual authentication (impersonation) and is also prone to attacks such as man-in-the-middle and replay attacks. OAUTH is the worst authorisation and authentication choice. Despite the security of third-party applications, it does not require a third party to share resources in the beginning as it is relying on a separate company's security which may be unpredictable.

d. To provide secure connection services for the travelling employees, which of IPSec, SSL/TLS, and SSH, would be a better option? Justify your answer. (2 marks)

IPSec applications would be the most suitable protocol for a travelling employee's secure connection. This protocol allows employees to access resources of a server as long as both the employee's device and the bank's router have IPSec enabled. This emulates the same process of an employee inside the company protected by the network provided by the bank. Due to the added modification of the IP packet (IP Header, IPSec Information & IP Payload) and the Server's router handling decrypting mechanisms of the modified IP packet to their modified packet, it allows for a secure connection between the employee and the server.

2. Programming Task

```
D:\University\2020\Semester 2\SENG2250 - System Network Security\Assignment3>java A3
Server is listening on port: 6868
Setup_Request: Hello
```

The server sets up a “Hello” request for any listeners. This is to make sure that it is in fact communicating with some receiver.

```
calculating public key yA...

publicYA:98421863645128132634071496872521834890081346760223162940012131344306339642950613428630785765601903597272421307837793149456408370735918702627342093643892468575182444495861422039629817565890051072115567505690716840305915120513651010853885268684289197746494065559600034533985678468495261513589689911630700941674

RECEIVED Server yA: 98421863645128132634071496872521834890081346760223162940012131344306339642950613428630785765601903597272421307837793149456408370735918702627342093643892468575182444495861422039629817565890051072115567505690716840305915120513651010853885268684289197746494065559600034533985678468495261513589689911630700941674

calculating public key yB...

publicYB:32281548006445130868863559974729658887539396586948641665855990685497709697526514622365099557574995343524255394534616382664685364512036448352569409565000712245547028566669773713982629202750937473630160691369845047802979369284001049570642420481774403372115159446460193151319191108646705365833557754709398536080

calculating session key

SESSION KBA: 126215143856637478557391650939045763194322664301415034852554040018839143770250590054257569710860742220654969499408756825972263024783951926952041993864372603803840654660422343495292988768822930137378781152840572158107502549144591893851301553850730595717316931838717628039021197017864220902051471031600067541372

Diffie public key is set.

RECEIVED public key yB: 32281548006445130868863559974729658887539396586948641665855990685497709697526514622365099557574995343524255394534616382664685364512036448352569409565000712245547028566669773713982629202750937473630160691369845047802979369284001049570642420481774403372115159446460193151319191108646705365833557754709398536080

calculating session key

SESSION KAB: 126215143856637478557391650939045763194322664301415034852554040018839143770250590054257569710860742220654969499408756825972263024783951926952041993864372603803840654660422343495292988768822930137378781152840572158107502549144591893851301553850730595717316931838717628039021197017864220902051471031600067541372

Diffie public key is set.
```

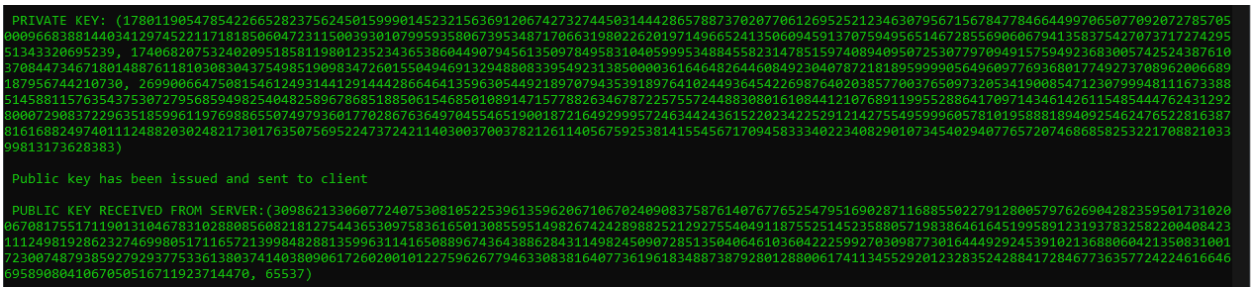
The server and client communicate to each other about public keys and using that to establish a secure connection to each other, the Diffie-Hellman exchange. A common session key is made and the process continues to execute as long as the session key is the same.

```
converting message ("raris and rovers, these hoes love chief sosa, hit em widda cobra") to BigInteger...

message in BigInteger form: 599060083366011373805481663803111137664845862299246916871351965183980568553007864654528492582984516313528778298851859271168210173710160507403207415394774 5

pre-hash: raris and rovers, these hoes love chief sosa, hit em widda cobra

pos-hash: 876830955528588523305078633573197915340085013594393101201669660664438369577 32

k = C4B3314D7CA76D7F3AC0AC92CE41E15EDD700D3159B67A4BFADA1BF7E4BF0FC2

hmac = A09D151ECE7AE1C59213BD99EC899D18F03AB73E0F6B7F81C9C783BF1A3CC6F7

HMACValue: -43144492707606801553783933952539962674541553664689478126845191636350270650633
```

The server hashes the plaintext to generate a HMAC value which will be later sent to the client to verify the message the server is sending.

```
SIGNATURE: 19977217217561516404839580961953292044399859163904978627304659274711886287452706424884130922863742927208924731739352543704128286576784190701658530812281498239791435642879297026172638243648516483596556568143731945381383885819231988758249422043391959554972988969285073401278616810266935433393807041996802436102912455667405471158377380226667216904958193334857992715258015524343211443105022991097150485601982069192575388247613950635680536309439606898659043717692684561329735044652447581693516292908894437383950935360164955131198057516644431321179399185068546508146961758027516590324346944709630561197479988808909092942168
```

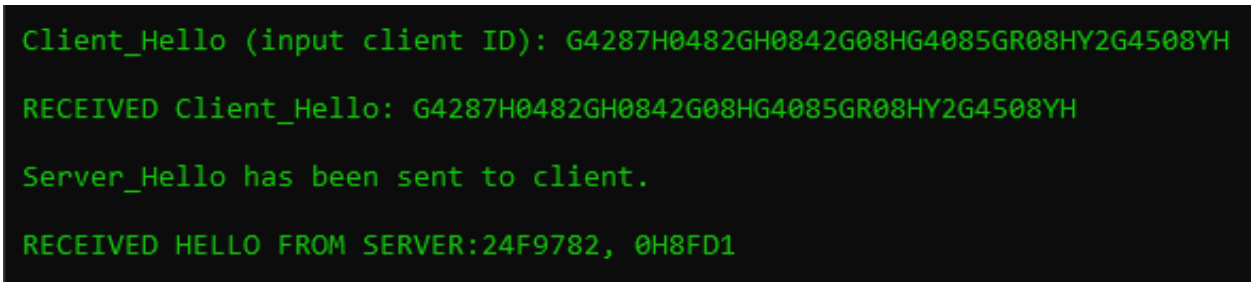
The server also generates a signature in the process.


PRIVATE KEY: (178011905478542266528237562450159990145232156369120674273274450314442865788737020770612695252123463079567156784778466449970650770920727857050009666838814403412974522117181850604723115003930107995935806739534871706631980226201971496652413506094591370759495651467285569060679413583754270737172742955134332069523917404820951858119801235234365386044907945613509784958310405999534884558231478515974089409507253077970949157594923683005742524387610370844734671801488761181030830437549851909834726015504946913294880833954923138500003616464826446084923040787218189599990564960977693680177492737089620066891879567442107305, 269900664750815461249314412914442866464135963054492189707943539189764102449364542269876402038577003765097320534190085471230799948111673388514588115763543753072795685949825404825896786851885061546850108914715778826346787225755724488830801610844121076891199552886417097143461426115485444762431292800072908372296351859961197698865507497936017702867636497045546519001872164929995724634424361522023422529121427554959996057810195888189409254624765228163878161688249740111248820302482173017635075695224737242114030037003782126114056759253814155456717094583334022340829010734540294077657207468685825322170882103399813173628383)

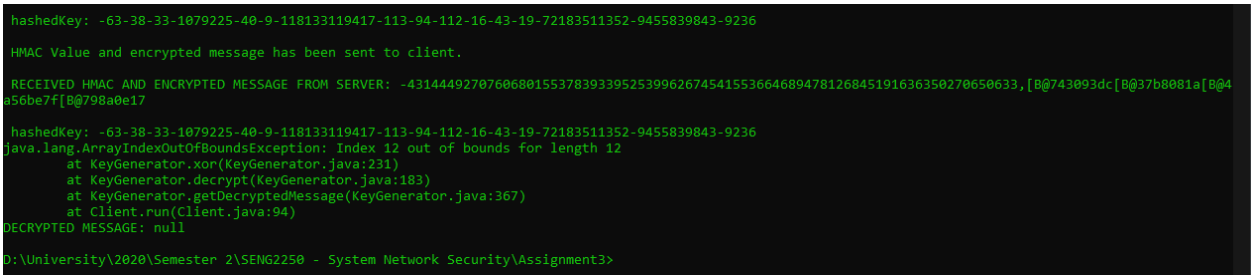
Public key has been issued and sent to client

PUBLIC KEY RECEIVED FROM SERVER:(309862133060772407530810522539613596206710670240908375876140767765254795169028711688550227912800579762690428235950173102006708175517119013104678310288085608218127544365309758361650130855951498267424289882521292755404911875525145235880571983864616451995891231937832582200408423111249819286232746998051711657213998482881359963114165088967436438862843114982450907285135040646103604222599270309877301644492924539102136880604213508310017230074879385927929377533613803741403809061726020010122759626779463308381640773619618348873879280128800617411345529201232835242884172846773635772422461664669589080410670505167119237144 70, 65537)

The server issues a private key with the given information which is sent to the client. It wouldn't matter who gets this public key.



Establishing hello connection between Server and client.



A hashed key is generated and used to encrypt the message to be sent to the client. The client receives the encrypted message and the HMAC value. The encrypted message is decrypted and the HMAC value should have been verified before accepting the final received decrypted message as legitimate. However, due to time-shortage, I was unable to finish this part off.