Secure Authentication Gateway using Kerberos

Vagish Yagnik 2K18/CO/381
Vishesh Jain 2K18/CO/391

OBJECTIVE

- Authenticating users to network services can prove dangerous when the method used by the protocol is inherently insecure, as evidenced by the transfer of unencrypted passwords over an unsecured network using the traditional FTP and Telnet protocols.
- To solve password storing and account managing problem a centralized user/password database is required, and then all the servers should consult that database to authenticate usernames and passwords.
- A multiuser/multiserver environment should allow to have any number of employees use the same credentials to log into resources throughout their domain

WHAT IS KERBEROS

- Kerberos is a network authentication protocol that uses tickets to allow entities to prove their identity over potentially insecure channels or distributed networks to provide mutual authentication.
- It also uses symmetric encryption to protect protocol messages from eavesdropping and replay attacks.

So our project would be basically a simulation of the authentication functionality of kerberos and we would use it over a small distributed system containing some clients who wants to access certain functionality provided by 2-3 servers.

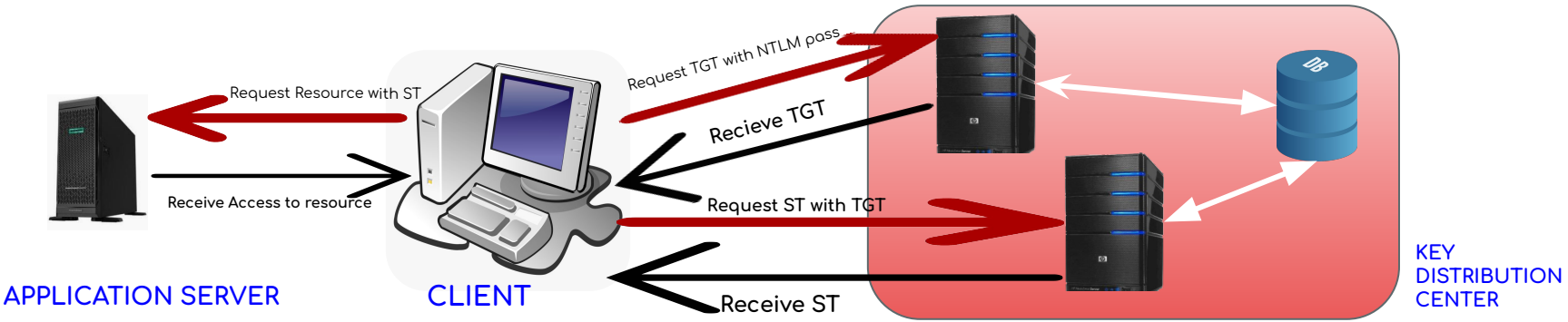
PROCEDURE

The basic operation of Kerberos is as follows :

1. Initial client authentication request.
2. KDC verifies the client credentials, The client sends the encrypted TGT to the TGS and requests a service ticket for access to the application server. This ticket has two copies of the session key: One copy is encrypted with the client secret key, and the other copy is encrypted with the Ticket granting server (TGS) secret key.
3. The client decrypts the message

contd...

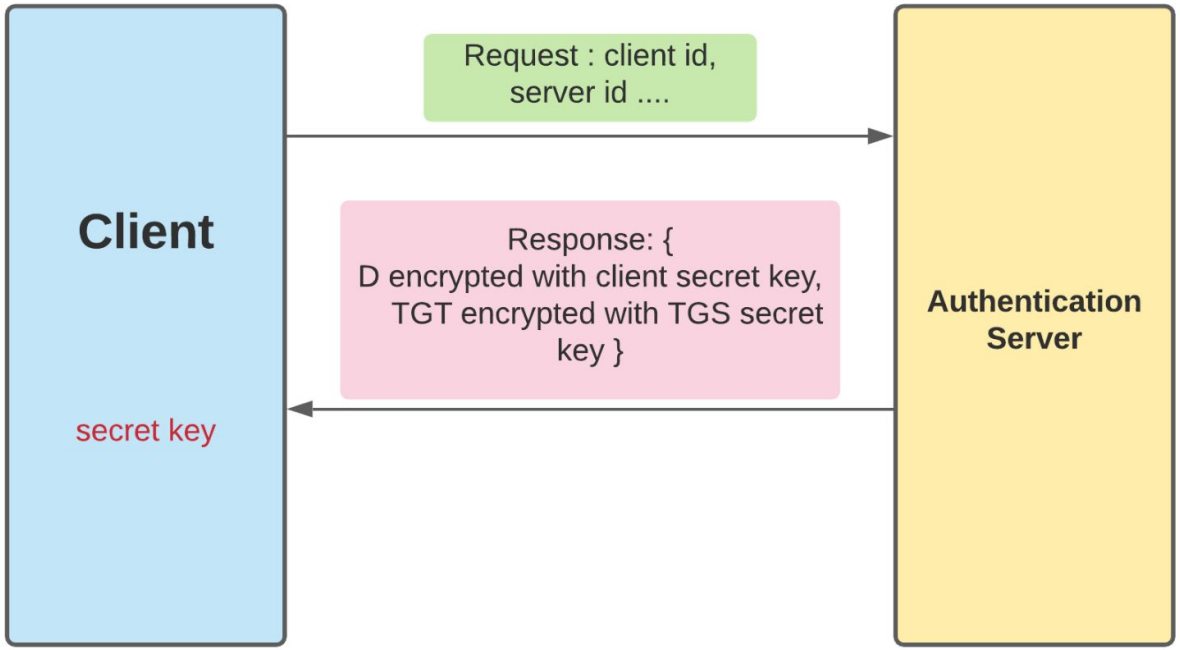
4. Client uses TGT to request access.
5. The KDC creates a ticket for the file server.
6. The client uses the file ticket to authenticate.
7. The target server receives, decryption and authentication.



Innovation

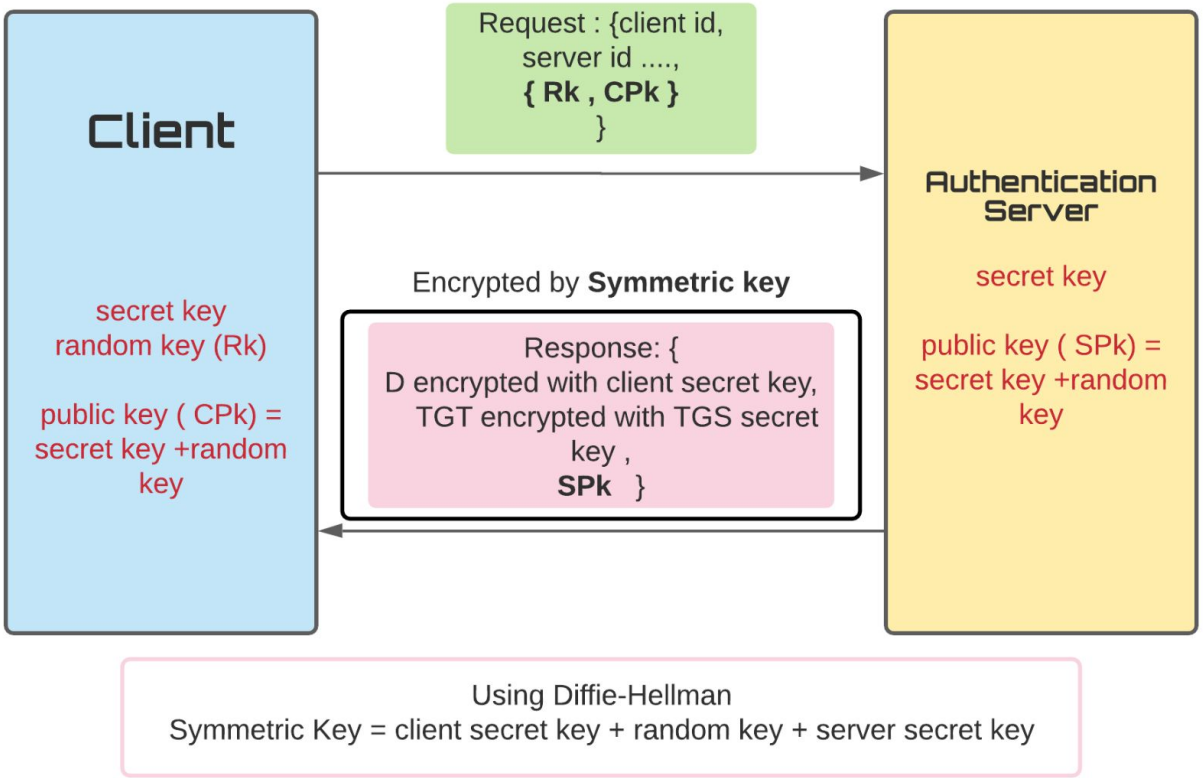
- The message sent from Authentication Server to Client is encrypted by users private key which is produced with user password, and the users private key is dealt with user password by using one-way Hash function.
- Therefore, the attackers can collect a lot of data which is from AS back to C, and make dictionary attacks. When the user password is not strong, it cannot effectively prevent the dictionary attacks. When the user password is not strong, it cannot effectively prevent the dictionary attacks.
- Once the user password is obtained, the user identity can be pretended to be a random one.
- So, to make the message passing secure from Authentication server to client we are introducing a way using diffie-hellman algorithm without adding more latency to the system.

Existing system



client - authentication server communication

Proposed system



Proposed Steps:

1. Clients should have a secret private key (CPvtKey) and one random key(RKey) would be generated at run time.
2. Using DH, a public key (CPubKey) is created which is a combination of the client's secret key and random key.
3. Now when a client sends a request to the authentication server, one more json object would be added containing both CPubKey and Rkey.
4. Authentication server on receiving the request first create a private Symmetric Key, say SymmKey and creates the server public key (SPubKey) from the combination of the server's secret key and random key.
5. Authentication server before sending response to client, encrypts the whole packet using AES symmetric encryption algorithm with key SymmKey and server’s public key.
6. On receiving response from server, client first generates symmetric private key using server public key (SPubKey) and it’s private key (CPvtKey) and then decrypts the message.

REFERENCES

- Authors Notes by MIT https://web.mit.edu/kerberos/
- Kerberos as used by Microsoft https://docs.microsoft.com/en-us/windows-server/security/kerberos/kerberos-authentication-overview
- https://medium.com/@robert.broeckelmann/kerberos-and-windows-security-kerberos-on-windows-3bc021bc9630
- GitHub Repository by MIT https://medium.com/@robert.broeckelmann/kerberos-and-windows-security-kerberos-on-windows-3bc021bc9630
- Comparitive Study https://medium.com/@robert.broeckelmann/kerberos-and-windows-security-kerberos-on-windows-3bc021bc9630
- Secure Key Distribution Prototype Based on Kerberos https://sci-hub.ee/10.1007/978-3-030-48149-0

THANK YOU