Understanding the Implementation of Key Distribution Center

Sapna Malik
Department of Computer Science & Engineering
MSIT

Abstract-*Key Distribution Centre (KDC is a network authentication protocol which is used in centralized authentication server to authenticate users to servers and servers to users. It is symmetric encryption technique for providing a safe platform for converting transmission of data between the user and server. It is known to be best and optimum protocol currently but also have some limitation and drawback. This paper first aims at understand implementing the KDC and then identifying its limitations and then discussing the new protocol DISTRIBUTED KDC that aims at overcoming these limitations.*

Keywords: Key Distribution Centre, Distributed Key Distribution Center, Authentication Protocol

I. INTRODUCTION

Secure media broadcast over the Internet poses unique security challenges. One problem access control to a large number of subscribers in a public broadcast. A common solution is to encrypt the broadcast data and to disclose the decryption key to legitimate receivers only. However, how do we securely and efficiently establish a shared secret among the legitimate receivers? And most importantly, how can we efficiently update the group key securely if receivers join or leave? How can we provide reliability for key update messages in a way that scales up to large groups? Recent research makes substantial progress to address these challenges. Current schemes feature efficient key update mechanisms assuming that the key updates are communicated reliably to the receivers. In practice, however, the principal impediment to achieve a scalable system is to distribute the key updates reliably to all receivers. We have understand the implemented KDC and discussed the distributed KDC, a novel key distribution protocol

II. FEATURES OF KDC

1. KDC features perfectly reliable, super-efficient member joins. We put premium on scalability. We are interested in situation where we have widespread audio or video streaming over a network to a large number of receivers.
2. KDC uses smaller key update messages than previous protocols.[2]
3. KDC features a mechanism that allows short hint messages to be used for key recovery allowing a trade-off of communication overhead with member computation. KDC proposes to append a small amount of key update information to data packets, such that the majority of receivers can recover from lost key update messages[2]
4. KDC allows to trade off security with communication overhead. Key Distribution Center has been modelled by closed queuing network and solved directly or by an asymptotic bounds approximation to evaluate cost functions and the conclusion was increasing security in Key Distribution Center will lead to higher cost. [3]

III. PROCESS OF KEY DISTRIBUTION[1]

KDC acts as as interface between client and server. KDC abstract a random key from the database.A1,A2...An[Figure1] are the client or server systems which request keys from KDC through Carrier Device as depicted in [Figure1].

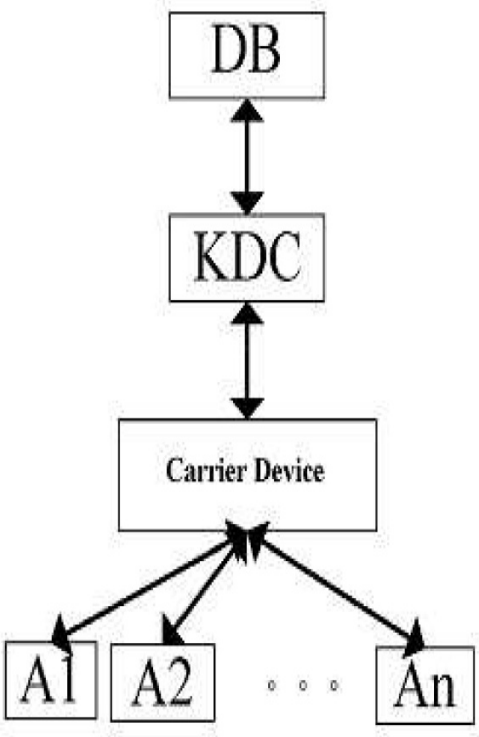

Figure 1: Key Distribution in KDC[1]

STEP 1

Client Authentication [1]:

The client sends a clear text message of the user ID to the KDC requesting services on behalf of the user. (Note: Neither the secret key nor the password is sent to the AS.) The AS generates the secret key by hashing the password of the user found at the database.

KDC checks to see if the client is in its database. If it is, then it sends back a Ticket- Granting-Ticket encrypted using the password of the client.

Once the client receives the TGT, it attempts to decrypt it with the secret key generated from the password entered by the user. If the user entered password does not match the password in the KDC database, the client's secret key will be different and thus unable to decrypt the TGT. [Figure 2]

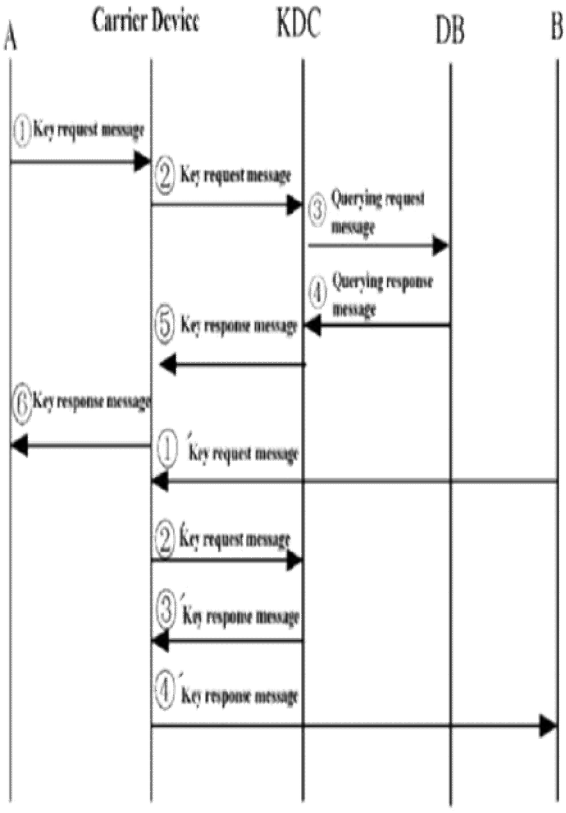

Figure 2: Client Authentication in KDC

STEP 2

Client Service Authorization [1]:

When requesting services, the client sends a message to the KDC Composed of the TGT and the ID of the requested service.

Upon receiving this message, KDC It decrypts the message using the secret key. This gives it the session key which includes the client ID, client network address and validity period. It then encrypts it and sends it to the client.[Figure 3]

STEP 3

Client Service Request [1]:

The client now has enough information to authenticate itself to the server. It sends an authenticator, which includes the client ID, timestamp and workstation address is encrypted using the session key.

The server decrypts the ticket using its own secret key to retrieve the Session Key. Using the sessions key, Server decrypts the Authenticator and sends the timestamp found in client's authenticator plus 1 ,encrypted using the Client/Server Session Key following message to the client to confirm its true identity and willingness to serve the client. The Client decrypts the confirmation using the Session Key and checks whether the timestamp is correctly updated. If so, then the client can trust the server and can start issuing service requests to the server.[Figure 4]

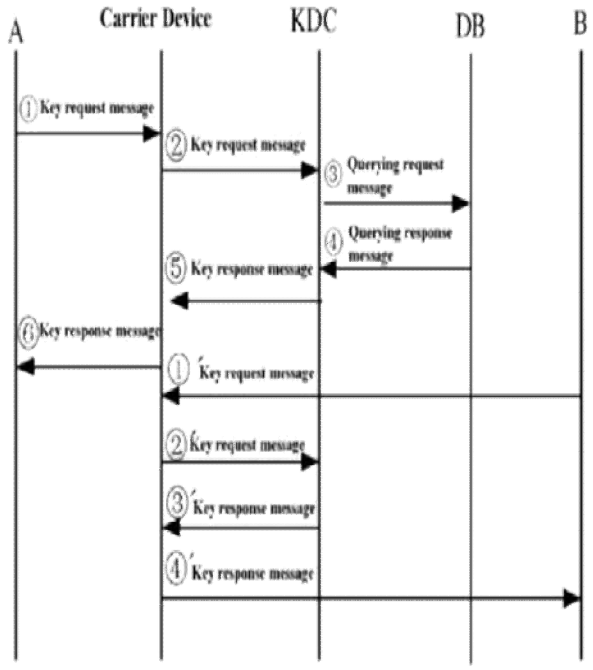

Figure 3: Client Service Authentication in KDC

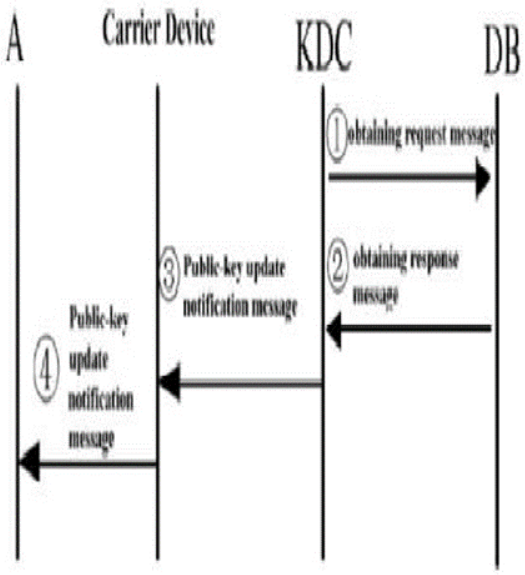

Figure 4: Client Service Request in KDC

IV. DRAWBACKS OF KDC [8][9]

1. A hierarchy of KDC's required for the large network, but must trust each other.
2. Session key lifetimes should be limited for greater security controlling purposes keys are used for lots of keys to keep track of(The encrypted key size in KDC is long(64 bits) and can therefore get lost during transmission.
3. A KDC can become a single point of failure. It requires continuous availability of a central server. When the KDC server is down, no one can log in. This can be mitigated by using multiple KDC servers.
4. Time Consuming: The time taken by packet to transfer data is too long and hence packet can get lost during transmission.
5. Vulnerable requires a trusted path through which passwords are entered. If the user enters a password to a program that Hs already been modified by an attacker(a Trojan Horse),or if the path between the user and the initial authentication program can be monitored, then an attacker may obtain sufficient information to impersonate the user.

V. DISTRIBUTED KEY DISTRIBUTION CENTRE (DKDC) [6]:

In the design of DKDC we are use multiple Key generation centers that is multiple client or multiple servers or multiple KDC as well as depicted in [Figure 5] DKDC is robust. We present lower bounds holding in the model for the main resources needed to set up and manage a distributed centre, i.e., memory storage, randomness, and bandwidth. Such as distributed Center keeps working even if some minority of the servers malfunction or misbehave under the control of a mobile adversary. Our scheme for a distributed key distribution Center is constructed using unconditionally secure proactive verifiable secret sharing schemes. We review the unconditionally secure verifiable secret sharing scheme.

VI. SECURITY REQUIREMENTS FOR GROUP KEY DISTRIBUTION[1][9]:

We consider dynamic groups where users can join or leave the group at any time. The main security properties of a group key management system for dynamic groups are:

1. Group Key Secrecy – guarantees that it is computationally infeasible for an adversary to discover any group key.
2. Forward Secrecy – (not to be confused with Perfect Forward Secrecy or PFS in key establishment protocols) guarantees that a passive adversary who knows a contiguous subset of old group keys cannot discover subsequent group keys. This property ensures that a member cannot learn about the new group keys after it leaves the group.
3. Backward Secrecy – guarantees that a passive adversary who knows a subset of group keys cannot discover preceding group keys. This property ensures that when a new member joins the group, he cannot learn about the previous group keys.

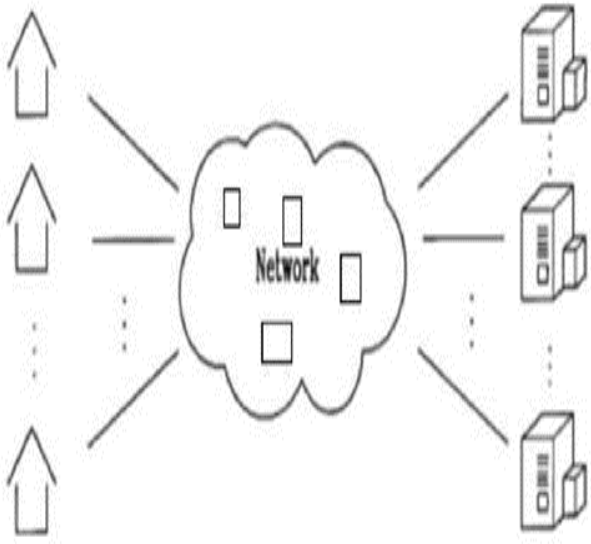

Figure 3: DKDC[6]

Table 1. Difference between KDC and DKDC

	KDC	DKDC
Cost	Require Less Cost	Higher Cost for Implementing
Security	Same	Same
Robust	A KDC can become a single point of failure: It requires continuous availability of a central server. When the KDC server is down, no one can log in. This can be mitigated by using multiple KDC servers.	It is more robust, in case of failure of one KDC, other KDC can look after The task, and data is also distributed on multiple KDC servers. It's no longer dependent on a single central server.
Time Consuming	The Time taken by packet to transfer data is too long and hence packet can get lost during transmission.	Not much is wasted as load is shifted from one centralized server to multiple KDC servers.
Reliability and trust	Everybody must trust the KDC	Here trade off with communication overhead is not just for security but also for security but also for robustness and reliability.

VII. CONCLUSION

After understanding the implementing Key Distribution Center and having the intensive literature survey of KDC. We came to conclusion that there are certain fallacies based upon cost, security, robustness ,reliability and other factors in the existing system of KDC. To overcome these drawbacks we have designed a system with multiple centers or server or client. We named this system as Distributed Key Distribution Center (DKDC) and compared this system with already existing KDC. The results of our study lead to conclusion that DKDC is a system which is much more secure, reliable, easy to be used and fast than KDC and hence much better than KDC.

REFERENCES

[1] Tie et al, "Key Distributing Method, Public Key of Key distribution Center online updating method and device", US Patent Application Publication.

[2] Adrian Perrig, Dawn Song and J.D.Tygar, "ELK, a New Protocol for Efficient Large-Group Key Distribution"

[3] Yishi Zhao and Nigel Thomas, "The Cost Model Analysis of Secure Key Distribution Model in Group-Based Key Pre-distribution", in Wireless Sensor Networks.

[4] Key Distribution Centre [Online],Available: http://en.wikipedia.org/

[5] Key Distribution Centre[Online],Available: http://www.zeroshell.net/eng/kdc/kdc- operation/

[6] Distrubuted Key Distribution Centre [Online],Available: http://web.mit.edu/dkdc/www/dialogue.html

[7] Thomas J,".Methods and Encryption for Object Encryption using Transparent Key Management.

[8] D. R. Stinson," On Some Methods for Unconditionally Secure Key Distribution and Broadcast Encryption". Designs, Codes and Cryptography, Volume 12 , Issue 3, Special issue: selected areas in cryptography I, Springer, Netherlands, 1997, pp. 215-243.

[9] M.Ramkumar, N.Memon , "Pre-Loaded Key Based Multicast and Broadcast Authentication in Mobile ad-Hoc Networks", Global Telecommunications Conference, 2003. Volume 3, Issue , 1-5 , GLOBECOM 03, IEEE,2003, pp. 1405 - 1409.