Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 4 (4): 176-185, 2012
ISSN 1815-7432 / DOI: 10.3923/rjit.2012.176.185

Inter-domain Authentication Scheme in a Distributed Mobile Network

M. Murali and R. Srinivasan
SRM University, Chennai, Tamil Nadu, India

Corresponding Author: M. Murali, SRM University, Chennai, Tamil Nadu, India

ABSTRACT

Over the recent period, transactions conducted on mobile network are gaining popularity due to the convenience and portability of mobile devices. The applications running on the mobile devices may require access to different servers just like in wired environment. Security over the mobile platform is more critical because wireless connections accessible to mobile devices are more vulnerable to attacks because of the openness of the wireless access points. It is easier for an opponent to gain access to the wireless network and perform fraudulent activities such as eavesdropping and impersonation. Moreover, security is more difficult to implement on a mobile platform because of the resource limitation of mobile devices. In this paper the concept of mobile gateway, which serves as a fixed agent for the mobile clients, is used. With the mobile gateway, all mobile clients and applications are connected to different servers through the mobile gateway server. The mobile client is authenticated to the mobile gateway through simple authentication mechanism such as protected password login and the mobile gateway in turn executes complex security transactions with Kerberos server on behalf of the mobile clients.

Key words: Authentication, remote server, Kerberos, realm, agent

INTRODUCTION

In a mobile computing environment, users are based on some wireless computing devices. However, security over the mobile platform is more critical due to the openness of wireless networks. Moreover, security is more difficult to implement on a mobile platform because of the resource limitation of mobile devices. Therefore, it is important to have some security mechanisms suitable for mobile clients. In this paper mobile gateway, which serves as a fixed agent for the mobile clients like base station. The mobile client is authenticated to the mobile gateway through simple authentication mechanism such as password login and the mobile gateway in turn executes complex security transaction with Kerberos server (Neuman *et al*., 2005; Butler *et al*., 2006; Cervesato *et al*., 2005) on behalf of the mobile clients. Considering the resource limitations of mobile devices such as battery power, processing capacity and vulnerable wireless connection all security related transactions will be executed only at the mobile gateway.

A mobile user wants to conduct a transaction with the database server, initiates the security mechanism (Kemmerer *et al*., 1994; Meadows, 1999; Diffie *et al*., 1992) from his mobile device. The mechanism is implemented between the mobile device and the mobile gateway. The mechanism starts with the client C sending an authentication request to the server S. In response to the request, the protocol messages are exchanged between server and client. They do not connect directly to the database server. Instead, they establish a trusted connection with the mobile gateway, which in turn executes transactions with the Kerberos server on behalf of the end users.

In this study the security architecture provides a secure means for authenticating end users and transacting with different servers based on tokens given by the Kerberos realms (Neuman *et al*., 2005; Butler *et al*., 2006; Cervesato *et al*., 2005). The combination of the mobile gateway and Kerberos server serves as an effective security protection.

The various attacks against the distributed system have resources (Seixas *et al*., 2009; Vieira *et al*., 2009) including the web servers, the communication links, the authentication and authorization mechanisms etc. Rehbock and Hunt (2009) proposes solutions based on authentication standards for enabling TNC (Trusted Network Connect) in open, web-based scenarios. Basso and Sicco (2009) presents MosaHIP, a Mosaic-based Human Interactive Proof (HIP), which is able to prevent massive automated access to web resources. El-Yamany *et al*. (2010) applies three different mining techniques based on the association rules to help predicting attacks. Han *et al*. (2009) uses three-party key establishment to enable secure communications for Service Requester and Service Provider through web services. Butler *et al*. (2006) have discussed about the Kerberos 5 protocol in the distributed web environment.

KERBEROS PROTOCOL

In the distributed environment, an unauthorized user may be able to gain access to the data that he is not authorized to access. In order to protect user information and resources, we need, that client systems to be authenticated. Kerberos (Neuman *et al*., 2005; Butler *et al*., 2006; Cervesato *et al*., 2005) is a widely used protocol and it is designed to authenticate a client to access all the required services in different realms. Kerberos allows clients and servers to reliably verify each others identity before connection is established. It provides advantages such as mutual authentication (Kemmerer *et al*., 1994; Meadows, 1999; Diffie *et al*., 1992) and message integrity as well as data confidentiality. Kerberos must go through a process of establishing a secure authenticated network connection.

Kerberos protocol (Stallings, 2003; Butler *et al*., 2006) includes two representative realms, namely Realm A and Realm B. Realm A includes Client, Authentication Server (AS), Ticket-Granting Server (TGS) and Local server. Realm B includes AS, TGS, Remote server. Authentication Server keeps a database containing the private keys of the clients and all of servers. Realm A and Realm B are connected with mobile gateway, thus the client and the Server can communicate each other.

The Kerberos Server (Stallings, 2003; Butler *et al*., 2006) must have the user ID and password of all participating users in its database. All mobile users are registered with the Kerberos server through the mobile gateway. The Kerberos server must share a secret key with each server. All servers are registered with the Kerberos server. All mobile users information are exchanged between the mobile gateways. Such an environment is known as realm. However, users in one realm may need access to servers in other realms and some servers may provide service to users from other realms, provided the users are authenticated. Mobile gateways in each realm will authenticate with each other as when transactions between a pair of realms is needed.

The terms used:

Realm:	Indicates realm of the client
Client:	Requires to gain access to Local server in Realm A or Remote server in Realm B
AS:	Authenticates servers to the client

TGS:	Grants service-granting ticket to the client
Local server:	Stores resource and data for the local users to share directly in same realm
Remote server:	Stores resource and data for the remote users to share in another realm
Options:	Used to request that certain flags be set in the returned ticket
Nonce:	A random value to be repeated in message (2) to assure that the response is fresh and has not been replayed by an opponent
Times:	Used by the client to request the following time settings in the ticket
From:	The desired start time for the requested ticket
To:	The requested expiration time for the requested ticket
Rtime:	Requested renew-till time

With the above said terms, we have described the mechanism as in Fig. 1. A user wants service on a server in another realm needs a ticket for that server. The user's client follows the usual procedures to gain access to the local TGS and then requests a ticket-granting ticket for a remote TGS. The client can then apply to the remote TGS for a service-granting ticket for the server in another realm of the remote TGS.

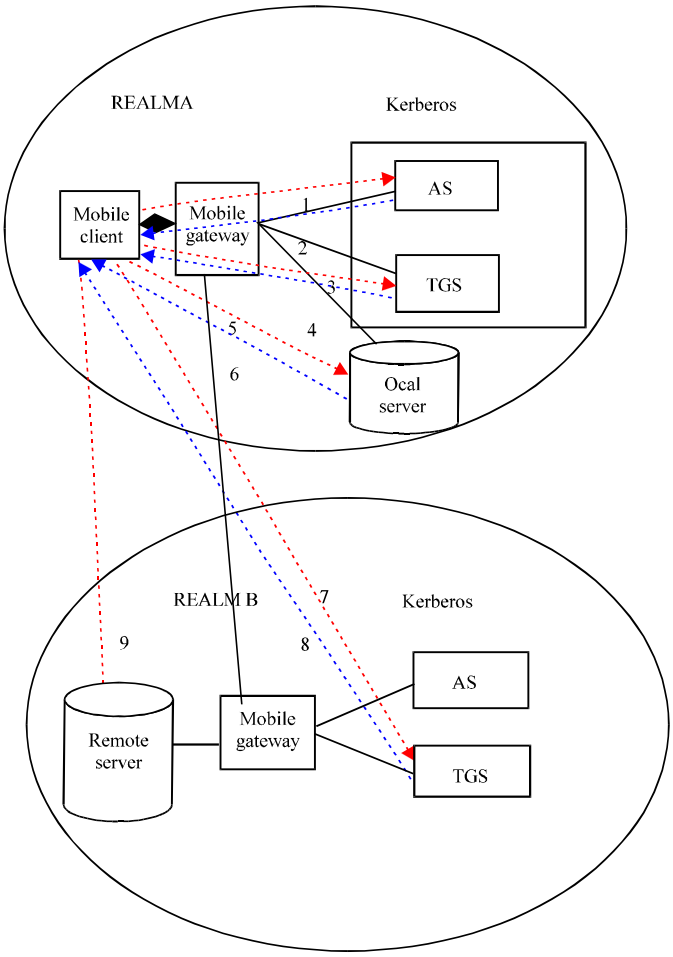

Fig. 1: Security architecture for distributed data environment

Message exchanges: To obtain ticket-granting ticket

C→AS→Options||IDc||Realmc||ID tgs||Times||Nonce1

AS→C: Realmc||IDc||Tickettgs||EKc (Kc,tgs||Times||Nonce1||Realmtgs||ID tgs

Tickettgs = Ektgs (Flags||Kc,tgs||Realmc||IDc||ADc||Times)

The Table 1 describes the meaning of the symbols used in 1 and 2.
To obtain service-granting ticket:

C→TGS: Options||IDls||Times||Nonce2||Tickettgs||Authenticatorc

TGS→C: Realmc||IDc||Ticketls||Ekc,tgs||K,c,ls||Times||Nonce2||Realmls||IDls)

Tickettgs = Ektgs (Flags||Kc,tgs||Realmc||IDc||ADc||Times)

Tickettgs = Ekv (Flags||Kc,v||Realmc||IDc||ADc||Times)

Authenticatorc = Ekc,tgs (IDc||Realmc||TS1)

The Table 2 describes the meaning of the symbols used in 3 and 4.
To obtain service:

C→LS: Options||Ticketls||Authenticatorc

Table 1: Symbols used in message 1 and 2

Symbol meaning
C: Client
AS: Authentication server
Options: Used to request that certain flags be set in the returned ticket
IDc: Tells AS identity of user from this client
Realmc: Tells AS realm of user from this client
IDtgs: Tells AS identity of TGS that user requests access
Times: Used by the client to request to the time settings in the ticket, it consists the desired start time, the requested expiration time and the requested renew expiration time
Nonce1: A random value that client produces to be repeated in message (2) to assure that the response is fresh and has not been replayed by an attacker
Tickettgs: Ticket to be used by client to access TGS
Kc: Encryption is based on client's password, enabling AS and user to verify password and protecting contents of message (2)
Kc,tgs: Copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring to share a permanent key
Realmtgs: Tells client realm of TGS
Ktgs: Ticket is encrypted with key known only to AS and TGS, to prevent tampering
ADc: Prevents use of ticket from client other than one that initially requested the ticket
E(): Encryption function based on Rijndael encryption algorithm of AES

Table 2: Symbols of message 3 and 4

Symbol meaning
Idls: Tells TGS identity of Local server
Nonce2: A random value that client produces to be repeated in message (4) to assure that the response is fresh and has not been replayed by an attacker
Authenticatorc: Client transmits an authenticator, which includes the ID and address of client's user and a timestamp
Ticketls: Ticket to be used by client to access Local server
Kc,ls: Copy of session key accessible to client created by TGS to permit secure exchange between client and Local server without requiring to share a permanent key
K ls: Ticket is encrypted with key known only to TGS and Local server, to prevent tampering
Realmls: Tells client realm of Local server
TS1: Informs TGS of time this authenticator was generated

Table 3: Symbols of message 5 and 6

Symbol meaning
LS: Local server
TS2: Informs Local server of time this authenticator was generated
Subkey: The client's choice for an encryption key to be used to protect this specific application session. If this field is omitted, the session key uses the ticket
Seq#: An optional field that specifies the starting sequence number to be used by Local server for messages sent to the client during this session. Message may be sequence number to detect replays

LS→C: Ekc,ls (TS2||subkey||Seq#)

The Table 3 describes the meaning of the symbols used in 5 and 6.

Message 1: Is a client request for a ticket-granting ticket

Message 2: Returns ticket-granting ticket, identifying information for the client and a block encrypted using the encryption key based on the user's password. This includes the session key to be used between the client and the TGS, times specified in message (1) the nonce from message (1) and TGS identifying information

Message 3: Includes an authenticator, a ticket and the name of the requested service

Message 4: Structure is same as message (2), returning a ticket plus information needed by the client

Message 5: The client may request as an option that mutual authentication is required. The authenticator includes new fields subkey, sequence number

Message 6: Includes timestamp from the authenticator

Kerberos version 5 ticket flags: The following flags field supports expanded functionality of Kerberos version 5:

Forwardable: (TGT only) Tells the ticket-granting service that it can issue a new TGT with a different network address based on the presented TGT.

Forwarded: Indicates either that a TGT has been forwarded or that a ticket was issued from a forwarded TGT.

Proxiable: (TGT only) Tells the ticket-granting service that it can issue tickets with a different network address than the one in the TGT.

Proxy: Indicates that the network address in the ticket is different from the one in the TGT used to obtain the ticket.

Renewable: Used in combination with the endtime and renew-till fields to cause tickets with long life spans to be renewed at the KDC periodically.

Initial: (TGT only) Indicates that this is a TGT.

Remote authentication procedure: The Table 1 gives the details about the symbols used in message 1 and 2.

Message 1: Client requests ticket-granting ticket for local TGS
Message 2: AS returns ticket-granting ticket for local TGS

The Table 2 gives the details about the symbols used in message 3 and 4.

Message 3: Client requests ticket-granting ticket for remote TGS
Message 4: Local TGS returns ticket-granting ticket for remote TGS

The Table 3 gives the details about the symbols used in message 5 and 6.

Message 5: Client requests service Local server
Message 6: Optional authentication of Local server to client
Message 7: Client requests ticket-granting ticket for Remote server

C→RTGS: Options||IDrs||Times||Nonce 3||Ticket rtgs||Authenticatorc

Message 8: Remote TGS returns ticket-granting ticket for Remote server:

RTGS→C: Realmc||IDc||Ticket rs||EKc,rtgs (Kc,rs||Times||Nonce3||Realmrs||IDrs)

Message 9: Client requests Remote server for remote service

C→RS: Options||Ticket rs||Authenticatorc

Registration phase: When the user wants to become a legal client to access the services, the user must register himself to the registration center through mobile gateway, at the same time, the service providing servers register themselves with the registration center.

Login phase: When the user wants to login to the server, the client's identity and password and the server's identity requesting access to the TGS are sent through the mobile gateway.

Authentication and session key agreement phase: After receiving the login request message from the user, the service provider authenticates the user through the mobile gateway. The AS includes several elements of the ticket in a form accessible to client. This enables client to confirm that this ticket is for the TGS and to know its expiration time.

Authentication server and user phase: When the AS has received authentication key from the registration center, this AS uses this authentication key to verify the user. After authentication is complete, a session key is generated to encrypt/decrypt all communication messages between the server and the user.

Remote authentication: In message (3) Client requests ticket-granting ticket for remote TGS and Local TGS returns ticket-granting ticket for remote TGS in message (4). In message (7) Client requests ticket-granting ticket for Remote server, the Remote TGS returns ticket-granting ticket for Remote server in message (8). In message (9) client requests Remote server for remote service.

Security analysis: The security of distributed server provided by authentication service based on Kerberos. The following is the summary of the Kerberos dialogue.

To obtain ticket-granting ticket:

C→AS: Options||IDc||Realmc||ID tgs||Times||Nonce1

AS→C: Realmc||IDc||Tickettgs||EKc (Kc,tgs|| Times||Nonce1||Realmtgs||ID tgs)

Tickettgs = Ektgs (Flags||Kc,tgs||Realmc||IDc||ADc||Times)

The Table 1 describing all the symbols used in message 1 and 2.

To obtain service-granting ticket:

C→TGS: Options||IDls||Times||Nonce2||Tickettgs||Authenticatorc

TGS→C: Realmc||IDc||Ticketls||Ekc,tgs||K,c,ls||Times||Nonce2||Realmls||IDls)

Tickettgs = Ektgs (Flags||Kc,tgs||Realmc||IDc||ADc||Times)

Tickettgs = Ekv (Flags||Kc,v||Realmc||IDc||ADc||Times)

Authenticatorc = Ekc,tgs (IDc||Realmc||TS1)

The Table 2 describing all the symbols used in message 3 and 4.

To obtain service:

C→LS: Options||Ticketls||Authenticatorc

LS→C: Ekc,ls (TS2||subkey||Seq#)

The Table 3 describing all the symbols used in message 5 and 6.

Masquerade attack: An opponent can act as a legal user.

In message (3) C sends the TGS a message that includes the ticket plus the ID of the (local server) requested service. The ticket was reusable and it was easy for the opponent to act as a legal

user. In addition, C transmits an authenticator, which includes the ID and address of C = s user and a timestamp. So, the authenticator is intended for use only once and has a very short lifetime. Nonce2 is a random value that client produces to be repeated in message (4) to assure that the response is fresh and has not been replayed by an attacker.

Malicious server attack: It is impossible for an opponent to masquerade as the server to cheat a remote user or the registration center.

For mutual authentication, the server can reply as in message (6). The server returns the Local server of time this authenticator was generated. Subkey also used as a client's choice for an encryption key to be used to protect the specific application session. Sequence number is an optional field that specifies the starting sequence number to be used by Local server for messages sent to the client during this session. Message may be sequence number to detect replays.

In message (7) client requests ticket-granting ticket for Remote server, with ID of the remote server, Nonce3 a random value that client produces to be repeated in message (8) to assure that the response is fresh and has not been replayed by an attacker. Realmc tells as realm of user from this client. Times used by the client to request to the time settings in the ticket, it consists the desired start time, the requested expiration time and the requested renew expiration time. Realmrs Tells client realm of remote server. IDrs tells TGS identity of the remote server.

Guessing attack: In message (2) encryption is based on user's password. User is always using the strong password. In message (4) Ekc,tgs key shared only by C and TGS which protects the contents of the message. The authenticator is encrypted with key known only to client and TGS, to prevent tampering or guessing.

Security of session key: An opponent cannot guess the session key. In message (2) AS returns ticket-granting ticket. The encryption is based on user's password, enabling AS and client to verify password and protecting contents of message (2). The copy of session key accessible to client created by AS to permit secure exchange between client and TGS without requiring them to share a permanent key. IDtgs confirms that this ticket is for the TGS. In addition it has expiration time also.

Offline dictionary attack: In offline dictionary attack, the attacker can record messages and attempts to guess user identity IDc and password from recorded messages. Here an attacker has to guess the identity IDc and password correctly at the same time. The authenticator c is generated by the client to validate the ticket. In addition The times prevents the attack after the ticket has expired. So it is difficult to guess all parameters at once in real time.

Man-in-the-middle attack: In this type of attack, the attacker intercepts the messages sent between the client and the server and replay these intercepted messages. An attacker can act as client to server or vice-versa with recorded messages. In message (4) Ekc,tgs, ticket is encrypted with key known only to AS and TGS to prevent tampering. In message (2) and message (4) Kc,tgs copy of session key accessible to TGS, used to decrypt authenticator. The authenticator is used by client to validate ticket. In message (6) Ekc,ls authenticator is encrypted with key known only to client and local server to prevent tampering. The TGS uses the session key to decrypt the authenticator. The TGS can then check the name and address from the authenticator with that of

the ticket and with the network address of the incoming message. If all match, then the TGS is assured that the sender of the ticket is indeed the ticket's real owner. Moreover, an opponent cannot compute the session key very easily. Therefore it is secure against man-in-the-middle attack.

DISCUSSION

In this study we have provided security mechanism in distributed database systems. In distributed database systems, the data are shared among users with different locations, yielding to a number of security issues. The major issues are ensuring that appropriate security measures when retrieving data from the distributed database. In distributed and mobile systems, it is difficult to consider a boundary, containing all the confidential information. This study describes a security mechanism for protecting transactions conducted over the mobile platform. Kerberos protocol provides cross-realm authentication, which enables a user to transparently access data on the server. Kerberos is a widely deployed protocol that is designed to authenticate a client to access the server in different realm. Kerberos provides dedicated message formats to protect the communication. Further, the security analysis proves that this authentication process is no masquerade attack and no malicious server attack.

CONCLUSION

This authentication mechanism gives way for increasing distributed server security and provides guarantee, which prevents unauthorized mobile clients. In this study, a mobile gateway is used with Kerberos authentication mechanism to access remote servers. By this way, only the gateway will be heavy but not the clients in each realm. Moreover, the opponents cannot obtain the necessary information to act as a legal user. And the system is capable of supporting clients and servers in large numbers.

REFERENCES

Basso, A. and S. Sicco, 2009. Preventing massive automated access to web resources. Comput. Security, 28: 174-188.

Butler, F., I. Cervesato, A.D. Jaggard, A. Scedrov and C. Walstad, 2006. Formal analysis of Kerberos 5. Theoretical Comput. Sci., 367: 57-87.

Cervesato, I., A.D. Jaggard, A. Scedrov and C. Walstad, 2005. Specifying Kerberos 5 Cross-realm authentication. Proceedings of the Workshop on Issues in the Theory of Security, January 10-11, 2005, Long Beach, CA, USA, pp: 12-26.

Diffie, W., P.V. Oorschot and M. Wiener, 1992. Authentication and authenticated key exchanges. Des. Codes Cryptogr., 2: 107-125.

El-Yamany, H.F., M.A.M. Capretz and D.S. Allison, 2010. Intelligent security and access control framework for service-oriented architecture. Inform. Software Technol., 52: 220-236.

Han, S., T. Dillon, E. Chang and B. Tian, 2009. Secure web services using two-way authentication and three-party key establishment for service delivery. J. Syst. Architecture, 55: 233-242.

Kemmerer, R., C. Meadows and J. Millen, 1994. Three systems for cryptographic protocol analysis. J. Cryptol., 7: 79-130.

Meadows, C., 1999. Analysis of the internet key exchange protocol using the nrl protocol analyzer. Proceedings of the Symposium Security and Privacy, May 12, 1999, Oakland, California, USA., pp: 216-231.

Neuman, C., T. Yu, S. Hartman and K. Raeburn, 2005. The Kerberos network authentication service. (V5) (2005), http://www.ietf.org/rfc/rfc4120

Rehbock, S. and R. Hunt, 2009. Trustworthy clients: Extending TNC to web-based environments. Comp. Commun., 32: 1006-1013.

Seixas, N., J. Fonseca and M. Vieira, 2009. Looking at web security vulnerabilities from the programming language perspective: A field study. Software Reliab. Eng., 1: 129-135.

Stallings, W., 2003. Cryptography and Network Security: Principles and Practice. 3rd Edn., Prentice Hall, London, UK., ISBN: 9780130914293, Pages: 681.

Vieira, M., N. Antunes and H. Madeira, 2009. Using web security scanners to detect vulnerabilities in web services. Proceedings of the IEEE/IFIP International Conference on Dependable Systems and Networks, June 29-July 2, 2009, Lisbon, Portugal, pp: 566-571.