2009 Fifth International Conference on Information Assurance and Security

Security Analysis of the Kerberos protocol using BAN logic

Kai Fan, Hui Li
Ministry of Edu. Key Lab. of Computer Network and Information Security
Xidian University
Xi'an, Shaanxi, China
kfan@mail.xidian.edu.cn, lihui@mail.xidian.edu.cn

Yue Wang
Xi'an Branch of Shaanxi Telecom Ltd.
Xi'an, Shaanxi, China
kelly8266no1@sina.com

***Abstract*—Kerberos protocol is a famous identity authentication protocol and it is widely used in the network as a standard. But there is still not a strict proof of it base on the Formal method. That is very nervous for the users. So a security analysis of the Kerberos protocol using BAN logic is proposed in this paper, and the reliability, practicability and security of Kerberos protocol are proved.**

Keywords-Kerberos protocol; Formal analysis; BAN logic

I. INTRODUCTION

As one of the core technology in network security communication systems cryptographic protocols has received great attention from all over the world and achieved great development. While it is not easy to analysis the security of a cryptographic protocol, much vulnerability difficult to find are hidden in seemingly correct protocol. As we all known, the design of cryptographic protocols is a work easy to go wrong. So it is a necessary aspect for protocol design to analysis the security of protocols. To provide accurate and credible analysis of protocol security researchers has put forward many protocol security proof technologies.

Among of them, the formal analysis methods [1] have become one of the most important means and tools to analysis and design the protocol security. The formal analysis method uses rigorous theoretical models to conduct the strict mathematical and logical deduction and demonstration for cryptographic protocols to prove the security or point out secure vulnerabilities of them. The BAN logic [2] is put forward by Burrows, Abadi and Needham is one of the most famous methods of the formal analysis method. The BAN logic has the advantages of clear concept, simple and easy to understand and use and it can effectively find the secure vulnerability difficult to detect in the protocol.

Kerberos protocol [3-4] is a three-party certification network security authentication protocol that is applied to an open network environment. Kerberos protocol has become the normal network identity authentication protocol in the field. And it is a representative technology in the Internet access control technology and it has widely used in the secure access control in the Internet. But there is still not a strict proof of Kerberos protocol base on the Formal method. That is of great significance for secure using Kerberos protocol.

In this paper we use the BAN logic to model and analysis the Kerberos protocol and have proved its reliability, practicability and security finally.

This paper is organized as follows: in Section 2, the logical postulate of BAN logic is introduced; in Section 3, the Kerberos protocol is proved by using BAN logic; finally, concluding remarks are made in Section 4.

II. THE LOGICAL POSTULATE OF BAN LOGIC

Rule (1) Message meaning rule:

$$\frac{P \text{ believes } Q \xleftrightarrow{K} P, P \text{ sees } \{X\}K}{P \text{ believes } Q \text{ said } X} \quad (1)$$

P believes *Q* has said *X* if *P* believes the key *K* is the shared key with *Q* and *P* sees *X* is encrypted by *K*.

Rule (2) Random number verification rule:

$$\frac{P \text{ believes fresh}(X), P \text{ believes } Q \text{ said } X}{P \text{ believes } Q \text{ believes } X} \quad (2)$$

P believes *Q* believes *X* if *P* believes *X* is sent currently and *Q* has said *X*.

Rule (3) Jurisdiction rule:

$$\frac{P \text{ believes } Q \text{ controls } X, P \text{ believes } Q \text{ believes } X}{P \text{ believes } X} \quad (3)$$

P believes *X* if *P* believes *Q* has the jurisdiction for *X* and *P* believes *Q* believes *X*.

Rule (4) Fresh transmission rule:

$$\frac{P \text{ believes fresh}(X)}{P \text{ believes fresh}(X,Y)} \quad (4)$$

Rule (5) Trust polymerization and trust projection rule:

$$\frac{P \text{ believes } X, P \text{ believes } Y}{P \text{ believes}(X,Y)} \quad \frac{P \text{ believes}(X,Y)}{P \text{ believes } X} \quad (5)$$

Rule (6) See rule:

978-0-7695-3744-3/09 $25.00 © 2009 IEEE
DOI 10.1109/IAS.2009.320

IEEE computer society

$$\frac{P \text{ believes} \xrightarrow{K} P, P \text{ sees}\{\text{X}\}\text{K}}{\text{P sees X}} \quad (6)$$

P can decrypt the message he has received if *P* has received messages encrypted by his own public key.

III. KERBEROS PROTOCOL PROOF USING BAN LOGIC

A. Kerberos protocol

The authentication process is as follows:

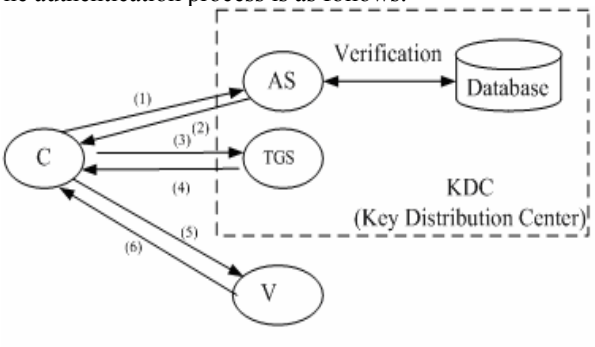

Figure 1. Kerberos protocol authentication process.

C: Client;

KDC: Key Distribution Center, is consists on *AS* and *TGS*;

AS: Authentication Serve;

TGS: Ticket Granting Server;

V: Application Server;

K_X: *X*'s shared key with *KDC*;

$K_{X,Y}$: *X*'s shared session key with *Y*;

$\{m\}K$: message *m* is encrypted by key *K*;

TGT: Ticket Granting Ticket, used to visit *TGS*;

$T_{X,Y}$: the Ticket that *X* visit *Y*;

$A_{X,Y}$: the authentication symbol of *X* and *Y*;

N_1, N_2 and N_C are random numbers

Timestamp;

Lifetime: effective survival time;

Addr: the IP address of client *C*.

The time that a client logs into the network to exits network is called a conversation in Kerberos system. Kerberos protocol is just designed for conversation. *AS* server keeps K_C, K_{TGS} and K_V and it is sharing these keys with client, *TGS* server and application server using to verify client identity when he is logging into the network. *TGS* is used to release the identity proof ticket. There are two kinds of certificates in Kerberos system-- ticket and authentication symbol. There are some information in tickets that server can use it to ensure who uses the ticket is just the one who possesses the ticket. The authentication symbol is the other certificate that will be submitted together with the ticket. The ticket and authentication symbol are both encrypted by symmetry key while they are encrypted by different key.

When *C* wants to visit the application server *V* there are three phases in the basic authentication process and they are implemented by three message exchange. The specific processes are as follows.

1) the authentication service exchange

The message exchanges (1) and (2) between the clients and *AS* is called original ticket exchange. This phase is the process that the client applies for the ticket *TGT* using to communicate with *TGS* and the session key from *KDC*. That is called authentication service exchange and called AS exchange. When the client logs into the network he needs input his user name and password. He applies for TGT from *AS* by sending message KRB_AS_REQ and *AS* responds him as message KRB_AS_REP.

a) $C \rightarrow AS$ *:* applying for the Ticket Granting Ticket (KRB_AS_REQ)

C, *TGS*, *addr*, N_1, *lifetime*

The client *C* sends the request that wants to visit *TGS* to *AS* and the request is sent in the message form. There are client name, the name of *TGS*, the IP address of the client, the random number and life cycle included in the request message. The random number N_1 is used to show this request is a fresh one to *AS*. The request message is sent in plaintext form.

b) $AS \rightarrow C$ *:* the Ticket Granting Ticket (KRB_AS_REP)

$\{K_{C,TGS}, T_{C,TGS}\}K_C$

$T_{C,TGS} = \{TGS, C, addr, N_1, lifetime, K_{C,TGS}\}K_{TGS}$

After *AS* received the request message from the client it looks up the client's shared key K_C in its database and generate the random session key $K_{C,TGS}$ and the ticket *TGT* of *TGS* as its response message. $K_{C,TGS}$ is encrypted by K_C which is used to conduct encryption communication between the client and *TGS*. The content of *TGT* includes the name of *TGS* and client, the IP address of the client, random number, the effective survival time and $K_{C,TGS}$. These data are encrypted by *TGS*'s shared key K_{TGS} to ensure only *TGS* can decrypt them. *AS* sends its response to the client which is encrypted by the client's key K_C that can ensure only *C* can decrypt that message. If the client can not decrypt that response message his identity is fake and the identity authentication is failure; otherwise, his identity is correct. After *C* receives *AS*'s response he decrypts the message and he will get the *TGS*'s ticket *TGT*. In next step he can send *TGT* to *TGS* to prove he possesses the correct identity to visit it. At the same time the client gets $K_{C,TGS}$ that can be used to conduct encrypted communication with *TGS* from *AS*.

2) the authorization service exchange

The authorization service exchange is also called the TGS exchange. The authorization service exchange process is consists on the message exchange (3) and (4). That is the process that client applies for the ticket $T_{C,V}$ and session key communicating with the application server *V* from *TGS*. TGS exchange is the message exchange between client and *TGS* and the message form is same as that in the *AS* exchange. But there is a significant difference that the session key but not the client's shared key is used as the encrypted and decrypted key in this process. The TGS exchange is consists

on two messages that are KRB_TGS_REQ and KRB_TGS_REP.

c) $C \rightarrow TGS$ *:* applying for the server ticket (KRB_TGS_REQ)

V, N_2, *lifetime*, $T_{C,TGS}$, $A_{C,TGS}$

$A_{C,TGS}$ ={C, *addr*, *timestamp*}$K_{C,TGS}$

C sends the request message to *TGS* which wants to visit *V*. The content is include the *V*'s name, *TGS*'s ticket *TGT* and authentication symbol $A_{C,TGS}$. *TGT* is encrypted by using *TGS*'s shared key K_{TGS} so only *TGS* can decrypt it. $A_{C,TGS}$ is including client's name, client's IP address and a timestamp. $A_{C,TGS}$ is encrypted by using the session key between client and *TGS* so only *TGS* can decrypt it. The ticket *TGT* could not prove anyone's identity and it can be re-used and its effective time is longer. While the authentication symbol is used to prove client's identity and it can be used only one time and its effective time is very short.

After *TGS* received client's request message it uses shared key K_{TGS} to decrypt *TGT* and knows client has got the session key $K_{C,TGS}$ with it from *AS*. Here the ticket *TGT* means that the client who uses $K_{C,TGS}$ is *C*. *TGS* uses $K_{C,TGS}$ to decrypt the authentication symbol and compares its data with *TGT*'s so it can believe *TGT*'s sender *C* is just *TGT*'s holder.

d) $TGS \rightarrow C$ *:* server ticket (KRB_TGS_REP)

{$K_{C,V}$, $T_{C,V}$}$K_{C,TGS}$

$T_{C,V}$ ={V, C, *addr*, N_2, *lifetime*, $K_{C,V}$ }$K_{V,TGS}$

After *TGS* verified client's identity is correct it generates random session key $K_{C,V}$ which is used to encrypted communicate between *C* and *V* and at the same time it generates the ticket $T_{C,V}$ which is used to visit *V*. The content of $T_{C,V}$ is includes the name of the application server and client, client's IP address, random number, effective survival time and the session key $K_{C,V}$. $T_{C,V}$ is encrypted by using session key $K_{V,TGS}$ so that only *V* can decrypt it. The session key $K_{C,V}$ and ticket $T_{C,V}$ is consists on *TGS*'s response message which is encrypted by using $K_{C,TGS}$ between *C* and *TGS*. After *C* received *TGS*'s response message he uses $K_{C,TGS}$ to decrypt it and gets $T_{C,V}$ and $K_{C,V}$.

3) the client / application server exchange

After AS exchange and TGS exchange client gets the ticket $T_{C,V}$ and session key $K_{C,V}$ for visiting *V*. The both identity authentication will be achieved after they pass the client / application server exchange. The client / application server exchange is consists on two messages which are KRB_AP_REQ and KRB_AP_REP. KRB_AP_REP is only used when there is need two-way authentication and server wants to prove its identity to client.

e) $C \rightarrow V$ *:* applying for service (KRB_AP_REQ)

V, $T_{C,V}$, $A_{C,V}$

$A_{C,V}$ ={C, *addr*, N_C}$K_{C,V}$

C sends request message to *V*. The content is includes *V*'s name, ticket $T_{C,V}$ and authentication symbol. $T_{C,V}$ can be decrypted only by *V*. The authentication symbol is includes client's name, client's IP address, random number. The authentication symbol is encrypted by using the session key between client and server so that it can be decrypted only by *V*.

After *V* received client's request message it uses $K_{V,TGS}$ to decrypt $T_{C,V}$ and knows *C* has got the session key $K_{C,V}$. Here means $T_{C,V}$ that the client who uses $K_{C,V}$ is just *C*. *V* uses $K_{C,V}$ to decrypt the authentication symbol and compares its data with $T_{C,V}$'s so that it can believes $T_{C,V}$'s sender *C* is just $T_{C,V}$'s holder. So *C*'s identity has been authenticated.

f) $V \rightarrow C$ *:* server authentication (KRB_AP_REP)

{N_C+1}$K_{C,V}$

After *V* verified *C*'s identity is correct it adds 1 to the random number that it has got from the authentication symbol and uses $K_{C,V}$ to encrypt it to send to client as its response message. That response message only can be decrypted by *C*. after *C* received and decrypted it he verifies the increased random number and compares with the effective of the random number to authenticate *V*. If it is correct *C* will believe it is just *V* has added the random number so *V*'s identity has been authenticated.

After the whole protocol exchange process there is a shared session key between client and application server and they can use that key to conduct encrypted communication to each other.

B. Kerberos protocol proof using BAN logic

Some reasonable assumptions are constructed for the analysis condition of Kerberos protocol is as follows:

(1) *C* believes $C \xleftrightarrow{K_C} AS$

(2) V believes $V \xleftrightarrow{K_{V,TGS}} TGS$

(3) V believes $V \xleftrightarrow{K_{V,TGS}} TGS$

(4) *C* believes AS controls $T_{C,TGS}$

(5) C believes TGS controls $T_{C,V}$

(6) C believes TGS controls $K_{C,V}$

(7) V believes TGS controls Tc,v

(8) C believes fresh(N_1)

(9) V believes fresh(N_C)

(10) C believes fresh(N_C)

The protocol proof using BAN logic is as follows.

1) Proof C believes $C \xleftrightarrow{K_{C,V}} V$

Because *C* believes $C \xleftrightarrow{K_C} AS$, *C* sees {$T_{C,TGS}$ }K_C according to **Rule (1)** we can get: *C* believes *AS* said $T_{C,TGS.}$

Because C believes fresh(N_1) according to **Rule (2)** we can get: *C* believes *AS* believes $T_{C,TGS.}$

Because *C* believes AS controls $T_{C,TGS}$ according to **Rule (3)** we can get: *C* believes $T_{C,TGS.}$

Then according to **Rule (5)** we can get: *C* believes $K_{C,TGS.}$

Because *C* sees {$T_{C,V}$}$K_{C,TGS}$ according to **Rule (1)** we can get: *C* believes *TGS* said $T_{C,V.}$

Because *C* believes fresh ($T_{C,V}$) according to **Rule (2)** we can get: *C* believes *TGS* believes $T_{C,V.}$

Because C believes TGS controls $T_{C,V}$ according to **Rule (3)** we can get: *C* believes $T_{C,V.}$

Finally according to **Rule (5)** we can get:

$$C \text{ believes } C \xleftrightarrow{K_{C,V}} V \quad (7)$$

2) Proof V believes $C \xleftrightarrow{K_{C,V}} V$

Because V believes $V \xleftrightarrow{K_{V,TGS}} TGS$, V sees $\{T_{C,V}\}K_{V,TGS}$ according to **Rule (1)** we can get: V believes TGS said $T_{C,V.}$

Because V believes fresh($T_{C,V}$) according to **Rule (2)** we can get: V believes TGS believes $T_{C,V.}$

Because V believes TGS controls Tc,v according to **Rule (3)** we can get: V believes $T_{C,V.}$

Finally according to **Rule (5)** we can get:

$$\text{V believes } C \xleftrightarrow{K_{C,V}} V \quad (8)$$

3) Proof C believes V believes $C \xleftrightarrow{K_{C,V}} V$

Because C believes fresh(N_C) so we can get:

$$\text{C believes fresh}(N_C+1) .$$

So we can get: C believes fresh($\{N_C+1\}K_{C,V}$).

Because C believes $K_{C,V}$, C sees $\{N_C+1\}K_{C,V}$ according to **Rule (1)** we can get: C believes V said $\{N_C+1\}K_{C,V}$.

According to **Rule (2)** we can get:

C believes V believes $\{N_C+1\}K_{C,V}$.

Finally we can get:

$$\text{C believes V believes } C \xleftrightarrow{K_{C,V}} V \quad (9)$$

As the same we can get:

$$\text{V believes C believes } C \xleftrightarrow{K_{C,V}} V \quad (10)$$

From now, we have got the final conclusion of the authentication protocol, which is the function (7), (8), (9) and (10). We can get the conclusion that Kerberos protocol can not only achieve the key establishment process between the client and server securely but also achieve the real-time communication between the two sides. In the whole process of the protocol it combines the authentication with the ticket so that that improves the security of the system. According to our analysis process the protocol has a strict architecture. We can not find any vulnerability of the protocol and the attack outside in our logic proof so we can say Kerberos protocol is correct and reliable.

IV. CONCLUSION

Because there is still not a strict proof of the Kerberos protocol base on the Formal method which has been used in the network widely the analysis and proof of the Kerberos protocol using BAN logic is proposed in this paper. And its reliability, practicability and security have been proved finally. That is of great significance for secure using Kerberos protocol.

ACKNOWLEDGMENT

This work was supported by National Science and Technology Support Plan 2007BAH08B01 and 2008BAH22B03, National 863 Program 2007AA01Z435, NSFC Grant 60772136 and National 111 Program B08038.

REFERENCES

[1] D. Dolev and A. C. Yao, "On the Security of Public Key Protocols," IEEE Trans. Information Theory, 29(2): 198-208, 1983.

[2] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," ACM Trans. Computer Systems, 8(1): 18-36, 1990.

[3] NEUMAN C. RFC 1510, The Kerberos Network Authentication Service (V5).1993.

[4] STEINER G. Kerberos, "an authentication service for open network system", Proceedings of the Winter 1988 Usenix Conference.1988.

[5] PF. Syverson and PC.Oorschot, "On unifying some cryptographic protocol logics", Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy. Los Alamitos: IEEE Computer Society Press. pp. 14-28. 1994.

[6] S. Kungpisdan, B. Srinivasan, PD. Le, "Accountbility Logic for Mobile Payment Protocols", Proceedings of the International Conference on Information Technology. Los Vegas: IEEE Computer Society Press. pp.40-44. 2004.