International Journal of Computer Networks and Communications Security

VOL. 2, NO. 12, DECEMBER 2014, 479–484
Available online at: www.ijcncs.org
ISSN 2308-9830

An Optimized Approach to Authenticate Users in Kerberos-Based Networks

Aliakbar Tajari Siahmarzkooh[1], Milad Shahini[2]

[1] Ph.D Student, Department of Computer Sciences, University of Tabriz, Tabriz, Iran

[2] MSc Student, Department of Computer Engineering, University of Mirdamad, Gorgan, Iran

E-mail: [1]tajari1987@gmail.com, [2]Shaini.milad@gmail.com

ABSTRACT

Authentication is a mechanism by which any entity can check whether its partner is one who claims to be in a relationship or is a disrupting that has been replaced by real party. One of the authentication protocols is Kerberos where timestamp is used to avoid sending repeated and unfeigned messages by Trudy. In this article a counter variable is considered for each client that represents number of sent messages from the client to servers. Advantage of this variable that host at ticket granting server is aware of repeated message during send it. Also, at servers we used a binary tree structure to store the messages and searching between them. Simulation results show significant improvement in the face of replay attacks by Trudy and response time to service at Kerberos-Based Network.

Keywords: *Authentication, Kerberos Protocol, Replay Attack, Ticket Granting Server, Binary Tree.*

1 INTRODUCTION

In technology security world, application sender and receiver processes are relating together instead of actual and legal individuals and authentication is a mechanism which processes is applied it to confirm digital identity of users. For example an application program which plays as a financial and credit institute server, can gain its legal identity from its owner and should evidence to institute user that is real representative of that institute [1]. Also, application program in customer situation should evidence that is real customer of who claimed to be a customer.

At security data world we are faced to A3 [2] instead of AAA. Meanings of these abbreviations are as following:

- First A: authentication is a mechanism which processes is evidenced actual or legal identity of users according to it.
- Second A: Authorization is a mechanism that determines working justification of processes which have been evidenced.
- Third A: Accounting is a mechanism which determine share of systematic and service resource of process and if any payment is transferred with receiving services or not.

Most important part of AAA operation is first A or authentication, if individual identity is not evidenced so access to systematic resource is a simple research at a table which has determined individual justification. So it is enough to reduce credit versus any services for auditing and when credit is zero services will cut.

Existence of a disrupting at every step should be supposed at authentication mechanism. Mechanism should be strong and intelligent so that could remove such disruptions. Also, authentication mechanism should be designed so that they could relate to actual individuals against wrong claims and other wrong claims would be disappointed.

Although never encryption key should not be transferred to net, Alice and Bob sometime uses “challenge and response” approach to authentication [3]. In this approach, one of parts produces a random system and send to other part. Second part should apply specific transmission and send back result. This system should be select

randomly at a wide area (minimum 125 bit) to be sure about unrepeated production of systems.

Other method of authentication is related to key distribution center (KDC) [4]. Central KDC stores and manages user's key and is an independent server. Authentication of individuals for meeting each server is possible just by KDC.

Process of authentication mechanism is as following:

- At first, Alice selects key named Ks and put it with Bob's user name (B) at a data building. Then after encryption by KA, results are sent for KDC with user name. Nobody, except Alice and KDC can decoding KA (B, Ks) and gaining the content (even Bob).
- KDC center after decoding second item with Alice's key understand what is Alice want to do and what is session key. So Alice's user name and session key is placed in a specific data building and encrypting it with Bob's secret key and send to Bob. So we insure that nobody except Bob can do decoding and extracting Ks from second message. If Bob could extract message form code right, sure that this message is from KDC because nobody knows secret key except KDC.

After decoding second message and extracting session key (Ks), KDC center mission is finished because after this section, Bob and Alice decode data and messages by Ks key and bandy together. According to this mechanism Alice meets with her bank service (Bob) and transfers some money from her account to other after authentication. Trudy which have listened to all Alice and Bob relation and store them, several day later messages will send to Bob. First day operation is repeated again and Alice is not aware about this disrupting and makes all calculation wrong, this attack is replay attack.

To resist above mechanism against replay attack, a timestamp should be inserted in each message, then receiver could distinguish if received message is old or newly produced [5, 6, 7]. In this approach, in addition to setting time of users with server, messages should be valid at a short time area (for example several minute).

In this approach, disrupting Trudy is able to initiate listened message again and make disruption during validity time.

One of protocols with is working with timestamp mechanism is Kerberos protocol [8, 9, 10]. There are three main components at authentication protocol:

1) Authentication Server (AS): Each user should states its identity at first step of entering to system (login step).
2) Ticket Granting Server (TGS): These servers issue ticket at network for receiving kind of services.
3) Server: This server supply services to customer after receiving ticket.

At version 5 of Kerberos, each ticket which issued by TGS have content like user name, name of server, address of customer, session key, ticket validity and timestamp.

2 BACKGROUND

Disrupting maybe refashion at systems time to prevent of create appropriate session between Alice and Bob [11, 12, 13]. If disrupting could change Alice's or Bob's time and remove their setting, so timestamp at all relation is not valid. Not validity of timestamp for Alice or servers is meant disrupting at message and they had to restart relation, because purpose of using timestamp is to prevent repeated message. As mentioned before, this mechanism is considered repeated or newly production of messages and if time is refashion by disrupting show that this mechanism is not working right. Maybe new message is supposed as old or old message is supposed new. So to prevent this happening, a counter variable at Kerberos protocol is used which maintain number of sent message from server to receiver. If Trudy change systems time and try to send repeated message, this variable show that an attack is occurred to times and parties should correct their times. Also, in a Kerberos-Based Network with a lot of users and servers maybe demanding on AS or TGS servers be more and other servers will be workless [14, 15]. To solve this problem a binary structure is used. Advantage of binary tree is to apply other server for reply to user when a server is busy by other users.

3 PROPOSED METHOD

The function of proposed protocol is as following:

1. Alice like other users has secret key and just AS serves know it. So at first step to login to system. Alice should login through working station. This process is started by sending Alice's user name (A), Bob's user name (B) and counter of messages (count) to collect

demands. Counter shows rate of relation between Alice and Bob. Center is collecting all demands to allocate to AS server, so that started from a random point at binary tree and if server is busy move toward next point and continue to apply a workless AS server. After selecting server, Alice's user name forms it and relates Alice and server together. This issue is shown at fig.1 that first demand is sent to AS3 which is busy (red), then go to other point AS7 which is workless (blue).

2. AS servers are considered all Alice references to Bob and compare them with counter. If number of reference is not conformed to counter, so sender was replaced by Alice and was a disrupting. Then any respond is not done and waiting for next demands. If they are conformed together, AS server produce Ks and A, Ks (KTGS) for Alice and after Cryptographic return to her and finally number of reference is increased at table and this updating is done for other servers. A sample is shown at Table1 that users and number of their references to each server are inserted.

   - Ks: a session key
   - A, Ks (KTGS): a ticket for referring to TGS servers
   - KTGS: Encrypted key of TGS servers

Because nobody at world knows secret key among AS and Alice, so everyone could decode second message and extract session key and ticket is Alice.

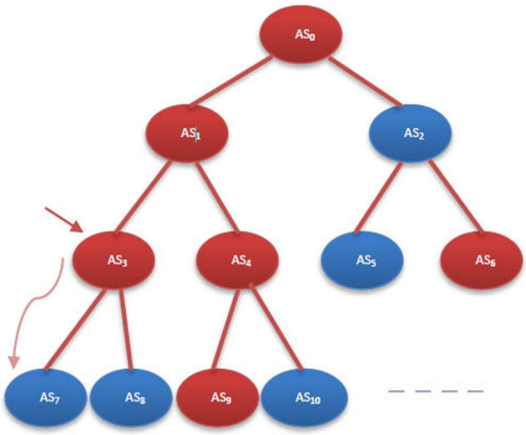

Fig. 1. binary tree related to AS.

Table 1: Table Sample of Users References to servers

Number of reference	User	Server
3	A0	B
1	A0	C
---	---	---
0	A1	B

3. Alice decodes received message from AS by secret key and extracts session key and ticket. When Alice wants to use of server should send ticket means A, Ks (KTGS) and name of servers (B) and number of references to TGS. These names first are sent to center of ticket collection. This center collects tickets and is searching a workless TGS server for each ticket (TGS servers like AS servers are stored at binary tree as fig.2). If sever is busy, binary tree search other point until server is found and ticket is delivered.

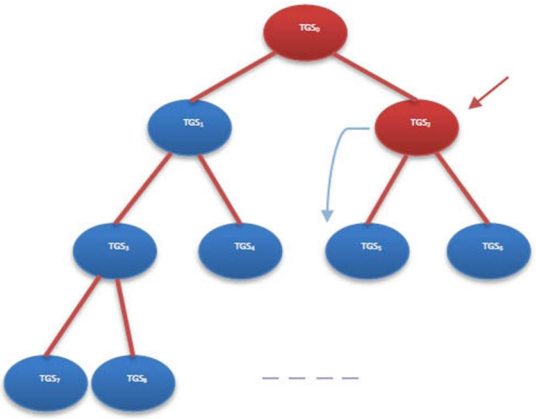

Fig. 2. binary tree related to TGS.

4. 4TGS serves at first decode ticket by secret key and Alice's user name extract message counter and session key. Then according to table of user references to server consider if sent counter is informed to number of references from Alice to Bob or not (see Table 2). If they are not conformed show that repeated message and doesn't send any respond to Alice, otherwise, produce KAB with B user name and Ks to create Ks (B, KAB), provided Alice has justification of service. Then other items as KB (A, KAB) is produced and in fact is Alice's

ticket to server B. These two items will be sent at next message, then Alice could ask service. So Alice should extract Ks of Ks(B, KAB) and then extract common key of KAB.

5. Alice sends KB (A, KAB) as one item and result of counter as second item to server [16, 17, 18].
6. Servers extract real identity of customer. If this rate is not conformed to table of user references, so existence of disrupting is certain and there is not any respond. Otherwise, increasing reference at table decodes other users referenced with KAB and send for Alice. Therefore, server identity is stated and Alice could understand that next number is own message.

Fig.3 shows proposed protocol. After finishing 6th section, Alice and server are able to relate a secure session by KAB common key.

Table 2: Table Sample of Users References to TGS Servers

Number of issued ticket	User	Server
5	A2	D
3	A0	B
---	---	---
4	A1	B

Algorithm: Optimized Protocol for Kerberos

1. Alice sends REQ(A,B,Req_Count) to Request Center
2. Request Center runs Binary Search Algorithm to find Free AS
3. AS Server compares Req_Count with A_Req_B in Req_Table
   If (Req_Count = A_Req_B +1) {
   send K_A (K_S,K_{TGS}(A,K_S)) to Alice
   A_Req_B =A_Req_B +1
   }
   Else Not_Response
4. Alice send REQ(K_{TGS}(A,K_S),B,Tck_Count) to Ticket Center
5. Ticket Center runs Binary Search Algorithm to find Free TGS
6. TGS Server compares Tck_Count with A_Tck_B in Tck_Table
   If (Tck_Count = A_Tck_B +1) {
   send K_S (B,K_{AB}), K_B (A,K_{AB}) to Alice
   A_Tck_B =A_Tck_B +1
   }
   Else Not_Response
7. Alice send K_B (A,K_{AB}), K_{AB}(B_Count) to Bob
8. Bob compares B_Count with A_B in General_Table
   If (B_Count = A _B +1) {
   A_B =A_B +1
   send K_{AB} (A_B) to Alice
   }
   Else Not_Response
9. Start Secure Session between Alice and Bob

Fig. 3. Proposed Algorithm to Defeat Trudy Attack to Timestamp.

4 SIMULATION RESULTS

We have done simulation on Kerberos protocol and proposed protocol on a net with different users as a software. Results are shown at figures. Fig.4 is shown time of respond 50 to 650 users with 5 AS serves, 5 TGS serves and 10 servers to users. Blue diagram is respond time that is not used tree structure and counter variables and orange diagram is respond time in proposed method.

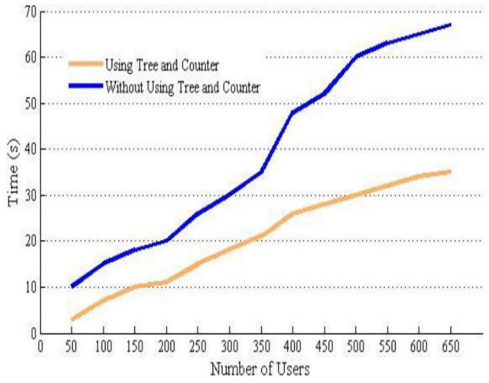

Fig. 4. Time of Responding To Users with and Without Using Tree Structure and Counter Variables.

Comparing these two diagrams show that use of tree structure to searching workless serves reduce time of searching. Fig.5 shows the result of server usage at same network and users. When tree structure and counter variable is not used a orange diagram is created which shows lack of balance and not equal demand of serves, so maybe some servers be workless while other serves are working. Blue diagram show balance at use of serves and in these diagrams we supposed that each 5 demand, 1 is faced with attack, so we done simulation and enter to network. Time to respond is considered as total time of searching for AS, TGS and workless servers. Results show that using tree structure cause to use of all servers and there is balance. Also, counter variable using doesn't cause to a lot of changes at respond time and insurance coefficient and unrepeated message is increased.

Fig.6 is shown number of unfeigned messages which are created at network and is detected by Kerberos protocol. These points are highlighted by orange color. Blue diagram in this table shows number of unfeigned messages which are detected by new protocol which uses of counter variable.

These diagrams show improvement of using counter variables at sent messages.

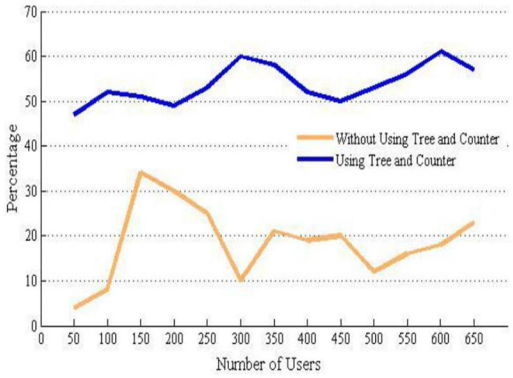

Fig. 5. Proportion of Using Servers with and Without Using Tree Structure and Counter Variables

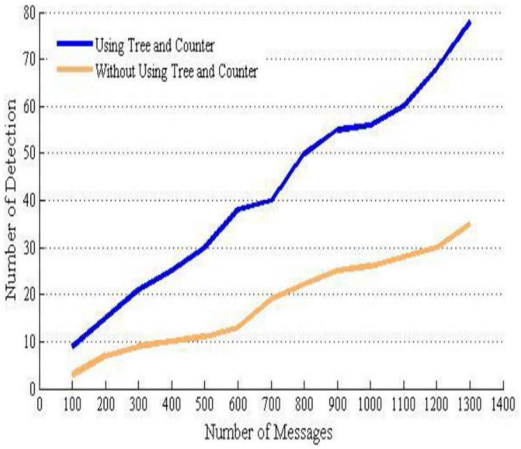

Fig. 6. Number of Diagnosed unfeigned Message with and without Using Counter Variables

5 CONCLUSION

We can calculate real send and receive messages by installing a counter to user's send messages and servers and if disrupting want to listen to messages and replaced itself by user and send listened message to servers again will lose up because they are not aware rate of new counters and these cause to server doesn't understand repeated message. Use of tree structure and binary tree for searching workless server gain of all servers' potential and less service time is allocated to users. Using these mechanisms could increase efficiency of Kerberos-Based Networks.

6 REFERENCES

[1] W. Stallings, ”Cryptography and network security: Principles and Practice”, fifth edition; Prentice Hall, 2011.

[2] A. Mendez, F. Garcia, R. Lopez, and G. Millan, ”Out-of-band federated authentication for Kerberos based on PANA”, computer communications, Vol. 36, No. 14, pp. 1527- 1538, August 2013.

[3] S. Sood, A. Sarje. A, and K. Singh, ”A secure dynamic identity based authentication protocol for multi-server architecture”, Journal of Network and Computer Applications, Vol. 34, No. 2, pp. 609- 618, March 2011.

[4] H. Aslan, ”Logical analysis of AUTHMAC_DH: a new protocol for authentication and key distribution”, Computers & Security, Vol. 23, No. 4, pp. 290- 299, June 2004.

[5] R. Lopez, F. Pereniguez, G. Lopez, and A. Mendez, ” Providing EAP-based Kerberos pre-authentication and advanced authorization for network federations”, Computer Standards & Interfaces, Vol. 33, No. 5, pp. 494- 504, September 2011.

[6] F. Pereniguez, R. Lopez, G. Kambourakis, S. Gritzalis, and A. Gomez, ”PrivaKERB: A user privacy framework for Kerberos”, Computers & Security, Vol. 30, No. 6-7, pp. 446- 463, September- October 2011.

[7] N. Fayed, E. Daydamoni, and A. Atwan, ”Efficient combined security system for wireless sensor network”, Egyptian Informatics Journal, Vol. 13, No. 3, pp. 185- 190, November 2012.

[8] F. Butler, I. Cervesato, A. Jaggard, A. Scedrov, and C. Walstad, ”Formal analysis of Kerberos 5”, Theoretical Computer Science, Vol. 367, No. 1- 2, pp. 57- 87, November 2006.

[9] L. fan, ”Design of a ticket-based single sign-on protocol”, Procedia Engineering, Vol. 23, pp. 537- 542, November 2011.

[10] Y. Huang, P. Lu, J. Tygar, and A. Joseph, ”OSNP: Secure wireless authentication protocol using one-time key”, Computers & Security, Vol. 28, No. 8, pp. 803- 815, November 2009.

[11] A. Shrestha, D. Choi, G. Kwon, and S. Han, ”Kerberos based authentication for inter-domain roaming in wireless heterogeneous network”, Computers and Mathematics with Applications, Vol. 60, No. 2, pp. 245- 255, July 2010.

[12] J. Lopez, R. Oppliger, and G. Pernul, ”Authentication and authorization infrastructures (AAIs): a comparative survey”, Computers & Security, Vol. 23, No. 7, pp. 578- 590, October 2004.

[13] H. Liu, P. Luo, and D. Wang, ”A distributed expansible authentication model based on Kerberos”, Journal of Network and Computer Applications, Vol. 31, No. 4, pp. 472- 476, November 2008.

[14] I. Cervestao, A. Jaggard, A. Scedrov, J. Tsay, and C. Walstad, ”Breaking and fixing public-key Kerberos”, Information and Computation, Vol. 206, No. 2- 4, pp. 402- 424, February- April 2008.

[15] H. Sun, H. Yeh, ”Password- based authentication and key distribution protocols with perfect forward secrecy”, Journal of Computer and System Sciences, Vol. 72, No. 6, pp. 1002- 1011, September 2006.

[16] A. Moralis, V. Pouli, S. Papavassiliou, and V. Maglaris, ”A Kerberos security architecture for web services based instrumentation grids”, Future Generation Computer Systems, Vol. 25, No. 7, pp. 804- 818, July 2009.

[17] R. Hwang, and F. Su, ”A new efficient authentication protocol for mobile networks”, Computer Standards & Interfaces, Vol. 28, No. 2, pp. 241- 252, December 2005.

[18] H. Chien, and J. Jan, ”A hybrid authentication protocol for large mobile network”, Journal of Systems and Software, Vol. 67, No. 2, pp. 123- 130, August 2003.

AUTHOR PROFILES:

Aliakbar Tajari received the MSc degree in Computer Science from University of Tabriz, in 2012. He is a Ph.D student of Computer Science. Currently, he is an Associate Professor at Mirdamad University. His interests are in System Security and Intrusion Detection Systems.

Milad Shahini received the BSc degree in Software engineering from the Golestan University, in Iran. Currently, he is a researcher in Mirdamad University. His research interests include Cloud Computing and Distributed Systems.