I. J. Computer Network and Information Security, 2018, 1, 10-17
Published Online January 2018 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2018.01.02

Security Protocol of Keys Management System for Transmission Encrypted Data

Rasha Subhi Ali[1]
[1]Computer Sciences Department, University of technology/ Baghdad, Iraq
E-mail: danafush@gmail.com

Asst. Prof. Alaa Kadhim F.[2]
[2]Computer Sciences Department, University of technology/ Baghdad, Iraq
E-mail: dralaa_cs@yahoo.com

Received: 26 June 2017; Accepted: 10 October 2017; Published: 08 January 2018

Abstract—One of the essential obstacles for the deployment of multicast is the lack of protection. And in multicast security, key management for securing organization or group communication is an important area that desires to be addressed. This paper will give an overview of four key management methods and Kerberos protocol. Cryptographic methods are frequently used for secure Data transmission wireless networks. Most cryptographic approaches can be symmetric and asymmetric, depending on the manner of the utilized keys. There are many kinds of key management methods which have been suggested for secure data transmission. This research includes a study of different key management methods to find an efficient key management for Secure and Reliable data transmission in the network. The experimental results showed that the fourth method represents the optimal key management method because it was providing a more secure way for the transmitted data, and the total time for data retrieval was (314.065, 376.119, 590.348, and 474.881) for the four key management methods sequentially to retrieve 71923records. The first three key management methods depend on symmetric key cryptography and the fourth key management method is a hybrid method, it was dependent on symmetric and asymmetric key cryptography, symmetric in the case of using user shared key and asymmetric in case of using server private key and this was unknown for any one.

Index Terms—Key management, encryption, decryption, secure data retrieval and Kerberos protocol.

I. Introduction

The main feature of the encryption/decryption method implementation is the generation of the encryption key. The primary goal of the cryptography is utilized not only to provide confidentiality, but also to provide solutions for other problems like: data integrity, authentication, non-repudiation. Cryptography is the strategies that permit information to be sent in a secure from in such a way that the only receiver able to retrieve this information [1]. In the network security, cryptography has a long history by using it to provide a way to store sensitive information or transmit it across insecure networks (i.e. The Internet), so that, it cannot be read by anyone except the intended recipient, wherein the cryptosystem there is a set of algorithms combined with keys to transform the original message (Plaintext) to the encrypted message (Cipher text) and transform it again in the intended recipient side to the original message (Plaintext) [2]. Ancient Egyptians are the oldest who encrypted text. The Internet provides essential communication and uses as a tool for many applications, i.e. secure commerce and payments to private communications and access control and so forth [3].

Key management plays an essential role in cryptography as the basis for securing cryptographic techniques, providing confidentiality, entity authentication, data origin authentication, data integrity, and digital signatures. The goal of a good cryptographic design is to reduce more complex problems to the proper management and safekeeping of a small number of cryptographic keys. Keying relationships in a communications environment include at least two parties (a sender and a receiver) in real-time. In a storage environment, there may be only a single party, which stores and retrieves data at distinct points in time [4]. There are several different types of cryptographic keys, each used for a different purpose. The keys are identified according to their classification as public, private or symmetric keys, and as to their use [5]. Cryptographic keys fall into two broad categories: 1) Symmetric key cryptography: fast implementations, good for encrypting large amounts of data; require shared secret key and 2) Asymmetric (public) key cryptography: inefficient for large data, good for authentication; no need to share a secret [6].

Yashaswini [7] key distribution for symmetric key cryptography is studied. Many protocols are used to make key distribution among the clients and authentication of the clients in distributed network. In a distributed network, there is also need for authentication of the client who

requesting the services of a server. Many authentication and key distribution protocols are used; the main two are Needham-Schroeder key distribution protocol and Kerberos protocol. In Needham-Schroeder key distribution protocol, the key distribution center generates a number once used session keys to allow access to the server services by the client. By the number of the session key allotted by KDC, sever can identify the authorized work satiations. In Kerberos protocol, each session key generated by the Kerberos KDC server will have time stamp associated with it, so that after some time it is going to expire. The server can easily identify the authorized client by checking the validity of the session key. Abdalla , Fouque and Pointcheval [8] proposed Password-based authenticated key exchange (PAKE) in the three-party scenario, in which the users trying to establish a common secret do not share a password between themselves but only with a trusted server. The natural generic construction of a three-party PAKE protocol from any two-party PAKE protocol and prove its security was presented. Tsai, Lee, and Hwang [9] presented a survey of all currently available password authentication schemes and analyzed how they work over insecure networks.

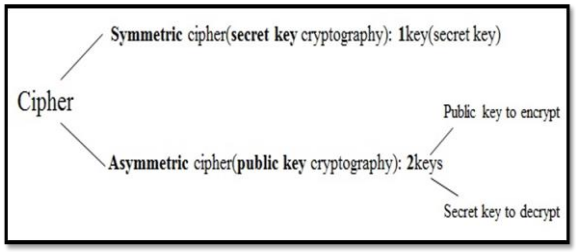

Fig.1. Symmetric Cipher and Asymmetric Cipher [6]

II. KERBEROS

Kerberos is an authentication service developed as part of Project Athena at MIT. The problem that Kerberos addresses is this: Assume an open distributed environment in which users at workstations need to access services on servers distributed over the network. We would like for servers to be able to limit access to authorized users and to be able to authenticate requests for service. In this environment, a workstation cannot be trusted to correctly identify its users correctly to network services. In particular, the following three threats exist: 1) the user can access a particular workstation, and pretend to be another user working from the workstation, 2) the user can change the network address of the workstation, so that, the requests sent from the workstation are changed and seem to come from the impersonation station, and 3) the user can eavesdrop on exchanges and used a replay attack to get access to the server or to disable operations. Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users. The first published report on Kerberos listed the following requirements [10]:

- Security: A network eavesdropper should not be able to gain the necessary information to impersonate a user.
- Reliability: Should be highly reliable and should appoint a distributed server architecture with one system able to back up another
- Transparency: Ideally, the user ought to not be aware that authentication is a career, place beyond the demand to enter a password.
- Scalability: The system must be able to support large numbers of clients and servers.

Two variations of Kerberos are in common use: v4 & v5. The core of Kerberos is the Authentication and Ticket Granting Servers – these are trusted by all users and servers and must be securely administered.

A. *Kerberos Version 4*

Version 4 of Kerberos makes utilize of DES, to provide the authentication service. Kerberos version 4 has an Authentication Server (AS) and it was having a Ticket Granting server (TGS). A Simple Authentication Dialogue [10]:

(1) C -> AS: $ID_C \| P_C \| ID_V$
- C = client.
- AS = authentication server.
- ID_C = identifier of user on C.
- P_C = password of user on C.
- ID_V = identifier of server V.
- C asks the user for the password.
- AS checks that the user supplied the right password.

(2) AS -> C: Ticket // Ticket = $E_{K(V)}$ [$ID_C \| AD_C \| ID_V$] //
- K (V) = secret encryption key shared by AS and V.
- ADC = network address of C
- The ticket cannot be altered by C or an adversary

(3) C -> V: IDC ‖ Ticket.
- Server V decrypts the ticket and checks various fields.
- ADC in the ticket binds the ticket to the network address of C.

However, this authentication scheme has problems in each time a user needs to access a different service he/she needs to enter their password (Read email several times, Print, mail, or file server, and Assume that each ticket can be used only once (otherwise open to replay attacks) and the password sent in the clear [10].

B. *Kerberos Version 5 [6]*

Kerberos Version 5 is laid out in RFC 1510 and gives a number of enhancements over version 4 in the areas of environmental shortcomings and technical deficiencies.

- Environmental shortcomings include (Encryption system dependence, Internet protocol dependence, Message byte ordering, Ticket lifetime, Authentication forwarding and Interrealm authentication).
- Technical deficiencies include (Double encryption, PCBC encryption, Session keys and Password attacks).

Kerberos Version 5 Message Exchanges:

(a) Authentication Service Exchange: to obtain ticket-granting ticket
(1) C → AS: Options ‖ ID_c ‖ $Realm_c$ ‖ ID_{tgs} ‖ $Times$ ‖ $Nonce_1$
(2) AS → C: $Realm_c$ ‖ ID_C ‖ $Ticket_{tgs}$ ‖ $E_{K_c}[K_{c,tgs}$ ‖ $Times$ ‖ $Nonce_1$ ‖ $Realm_{tgs}$ ‖ $ID_{tgs}]$
$Ticket_{tgs} = E_{K_{tgs}}[Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times]$
(b) Ticket-Granting Service Exchange: to obtain service-granting ticket
(3) C → TGS: Options ‖ ID_v ‖ $Times$ ‖ ‖ $Nonce_2$ ‖ $Ticket_{tgs}$ ‖ $Authenticator_c$
(4) TGS → C: $Realm_c$ ‖ ID_C ‖ $Ticket_v$ ‖ $E_{K_{c,tgs}}[K_{c,v}$ ‖ $Times$ ‖ $Nonce_2$ ‖ $Realm_v$ ‖ $ID_V]$
$Ticket_{tgs} = E_{K_{tgs}}[Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times]$
$Ticket_v = E_{K_v}[Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times]$
$Authenticator_c = E_{K_{c,tgs}}[ID_C \| Realm_c \| TS_1]$
(c) Client/Server Authentication Exchange: to obtain service
(5) C → V: Options ‖ $Ticket_v$ ‖ $Authenticator_c$
(6) V → C: $E_{K_{c,V}}$ [TS_2 ‖ Subkey ‖ Seq#]
$Ticket_v = E_{K_v}[Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times]$
$Authenticator_c = E_{K_{c,V}}[ID_C \| Realm_c \| TS_2 \| Subkey \| Seq\#]$

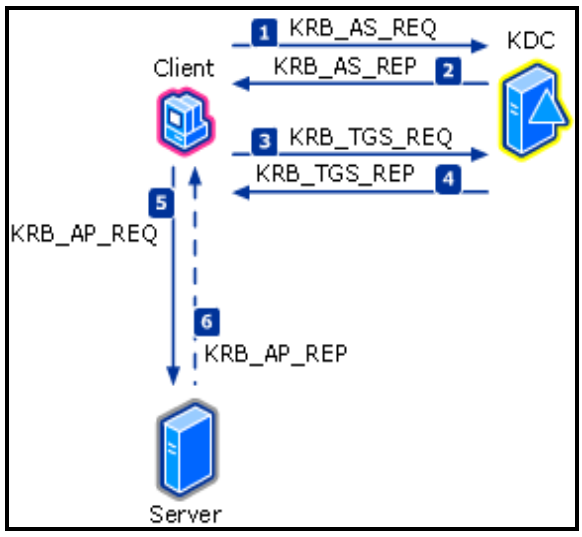

Fig.2. Kerberos Exchange and Message Summary

The Authentication Service Exchange *[10]*

1) Kerberos authentication service request (KRB_AS_REQ), and
2) Kerberos authentication service response (KRB_AS_REP).

The Ticket-Granting Service Exchange [10]

3) Kerberos ticket-granting service request (KRB_TGS_REQ), and
4) Kerberos ticket-granting service response (KRB_TGS_REP).

The Client/Server Exchange [10]

5) Kerberos application server request (KRB_AP_REQ), and
6) Kerberos application server response (KRB_AP_REP).

In a Wireless sensor networks, the issue of confidentiality should address the following requirements [11]:

i. A sensor node should not allow its readings to be accessed by its neighbors unless they are authorized to do so,
ii. Key distribution mechanism should be extremely robust,
iii. Public information such as sensor identities, and public keys of the nodes should also be encrypted in certain cases to protect against traffic analysis attacks.

III. Methodology

In this research four key management methods are used to authenticate transmitted data:

1. The shared secret key (Ks) between user and server is used in the encryption and decryption process. This key is private not known except the server and user known it. Also, the retrieval depends on this key, but this method causes load on the server site when dealing with big data, such as 1000 user wants to retrieve from database of size 6 GB then the server needs 6000 GB for storing it, also the server needs to deal with large number of keys (e.g. 1000 key for this example). Figure (3) demonstrates the steps of exchanging data by using the first key management method.

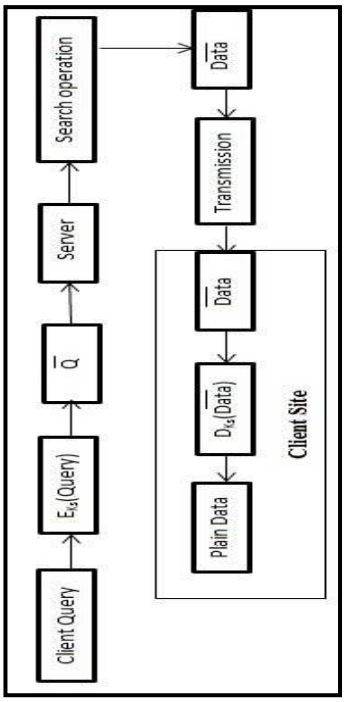

Fig.3. First Key Management Method

The steps for exchanging messages by using the first key management method is discussed in the following algorithm.

Input: User Query
Output: Relevant data in plain form.
Begin:
Step 1: Encrypt user query by using shared secret key with an encryption algorithm.
Step 2: Send the encrypted query let it be $\widetilde{Q}$ to the server.
Step 3: Search in the encrypted data about the required data according to the user query.
Step 4: If the relevant data was found then
Send the results (relevant data) to the user
Else
Send nothing was found
End If
Step 5: On the user site decrypts the received data by using same shared key and encryption algorithm
End

2. The server has a database which includes the user identifier with user keys. Also, for each user there is a secret key (Ku) shared with the server, but in this method server encrypts a database by using the server key (Ks) and the retrieved data is encrypted by using the server key only in this method. This method takes more time in the retrieval process, but it was not caused loaded on the server site because of the server encrypt the database by using one key. This method less secure than the previous method because the server key will be known for all the users who dealing with it. The retrieval process is illustrated in figure (4):

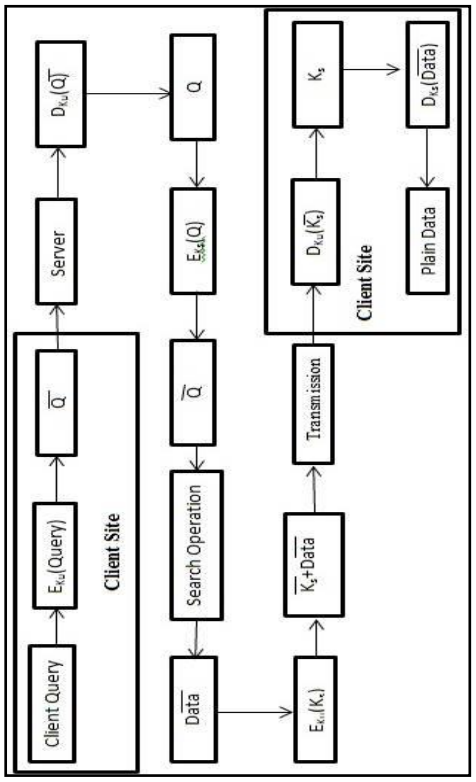

Fig.4. Second Key Management Method

The steps for exchanging messages by using the second key management method is discussed in the following algorithm

Input: User Query
Output: Relevant data in plain form.
Begin:
Step1: Encrypt user query by using user secret key with an encryption algorithm.
Step 2: Send the encrypted query let it be $\widetilde{Q}$ with the user identifier let it be (ID) to the server.
Step 3: At the server site there is a database includes users IDs' with users' keys. By using user ID get user key and decrypt $\widetilde{Q}$ // return plain query Q//
Step 4: Encrypt Q by using the server key with an encryption algorithm let it be $\widetilde{Q}1$.
Step 5: Search in the encrypted data about the required data according to the user query.
Step 6: If the relevant data was found then
Encrypt server key lets it be Ks by using user key and Send the results (relevant data let it be $\overline{Data}$) with encrypted server key let it be $\widetilde{Ks}$ the to the user
Else
Send nothing was found
End If
Step 7: At the user site decrypts the $\widetilde{Ks}$ to obtain a server key.
Step 8: decrypt the received data by using server key and encryption algorithm
End

3. Depending on the server database which also includes key with an identifier for each user, but in this method the retrieved data is encrypted by using the server key (Ks) and user key (Ku). This method gives more protection for the exchanged data, but takes more time than the previous method. This method was not caused loaded on the server site because of the server encrypt the database by using one key. The steps for exchanging data by using the third key management method was explained in figure (5).

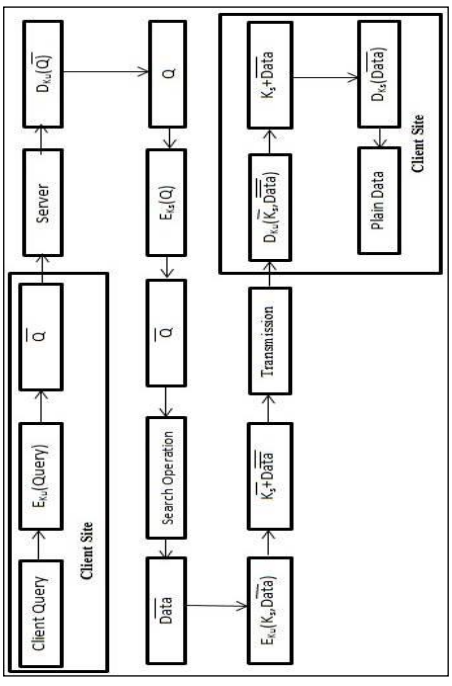

Fig.5. Third Key Management Method

The steps for exchanging messages by using the third key management method is discussed in the following algorithm

Input: User Query
Output: Relevant data in plain form.
Begin:
Step1: Encrypt user query by using user secret key with an encryption algorithm.
Step 2: Send the encrypted query let it be $\widetilde{Q}$ with the user identifier let it be (ID) to the server.
Step 3: At the server site there is a database includes users IDs' with users' keys. By using user ID get user key and decrypt $\widetilde{Q}$ // return plain query Q//
Step 4: Encrypt Q by using the server key with an encryption algorithm let it be $\widetilde{Q1}$.
Step 5: Search in the encrypted data about the required data according to the user query.
Step 6: If the relevant data was found then
Encrypt server key (Ks) and retrieved data by using user key and Send the results (double encrypted data let it be $\overline{\overline{Data}}$) with encrypted server key let it be ~~K~~s the to the user
Else
Send nothing was found
End If
Step 7: At the user site by using user key decrypts the $\widetilde{Ks}$ to obtain a server key and decrypt the $\overline{\overline{Data}}$ to obtain $\overline{Data}$.
Step 8: decrypt the $\overline{Data}$ by using server key and encryption algorithm to obtain data in plain form
End

4. Based on the server database that also includes key with an identifier for each user, but in this method the retrieved data is encrypted by using user key (Ku). The server key was kept private in this method not known by the users. The transmitted data will be decrypted by using server key and encrypted by using user key before sending it to the user. The transmitted data could not be decrypted unless using the private user key. This method gives more protection for the exchanged data than the previous discussed methods, but it was taken more time in the data retrieval process than the consumed time in the first and second key management method. Also, this method provides the key encryption and decryption time, which is consumed in the third key management method. This method was not caused loaded on the server site because of the server encrypt the database by using one key. The steps for exchanging data by using the fourth key management method was explained in figure (6).

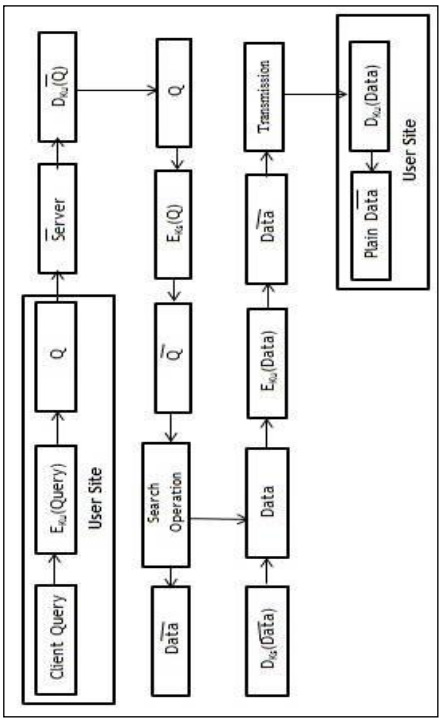

Fig.6. Fourth Key Management Method

The steps for exchanging messages by using the fourth key management method is discussed in the following algorithm

Input: User Query
Output: Relevant data in plain form.
Begin:
Step1: Encrypt user query by using user secret key with an encryption algorithm.
Step 2: Send the encrypted query let it be $\widetilde{Q}$ with the user identifier let it be (ID) to the server.
Step 3: At the server site there is a database includes users IDs' with users' keys. By using user ID get user key and decrypt $\widetilde{Q}$ // return plain query Q//
Step 4: Encrypt Q by using the server key with an encryption algorithm let it be $\widetilde{Q1}$.
Step 5: Search in the encrypted data about the required data according to the user query.
Step 6: If the relevant data was found then
1) Decrypt the retrieved data by using the server key to return data to its plain form let it be Data
2) Encrypt Data by using user key let it be $\overline{Data}$
3) Send $\overline{Data}$ to the user
Else
Send nothing was found
End If
Step 7: At the user site by using user key decrypts $\overline{Data}$ to obtain Data in plain form.
End

IV. Experimental Results and Performance Analysis

In this section the performance and effectiveness of the proposed key management algorithms were tested with different databases. Experimental results show that execution time for data retrieval process. The comparison

of total retrieving time is shown in table (2) and figure (7) and the results show the retrieving by using the first key management method faster than retrieving by using other key management method, but the first key management method caused load on the server site so it was not proper to be used when dealing with big databases.

In this work the row store database or column store database was used according to user request if the user wants to retrieve from the original database the row store database was used and if the user wants to retrieve from the file that was included only query relevant data the column store database was used; the column store database will be more faster in the query processing time because the searching operation was done on only relevant data instead of whole data.

The column store database saving more time for the query processing and this is presented in the results of tables (II and III), the average saving time are(59%, 57%, 37% and 39%) for the four key management methods sequentially.

Table 1. Number of Operations at the Server and Client Site in Each Method

	Client site	Server site
First Key Management Method	Two operations only encryption query and decryption of retrieved data	One operation only matching operation
Second Key Management Method	Three operations, encryption query, and decryption operation for the server key and retrieved data	Four operations decryption query by user key, encryption query by server key, matching operation, and encryption for the user key
Third Key Management Method	Four operations, encryption query, decryption operation for the server key and retrieved data by the user key, and decryption of the retrieved data by the server key	Five operations decryption query by user key, encryption query by server key, matching operation, encryption for the user key and the retrieved data
Fourth Key Management Method	Two operations only encryption query and decryption of retrieved data	Five operations decryption query by user key, encryption query by server key, matching operation, decryption for he retrieved data by server key and encryption for data by the user key

Table 2. Comparison the Data Retrieval Time for the Four Suggested Key Management Method Based on Row Store Database

DB NAME	Query	DB SIZE	First method row store DB	Second method row store DB	Third method row store DB	Fourth Method row store DB	NO.OF Retrived Data
Dept	كلية ابن الهيثم كيمياء	240	1.117	4.161	4.718	1.017	3
	96		0.765	3.699	4.613	1.045	1
	الجامعة التكنولوجية		0.451	3.753	4.824	0.841	6
Dhuk	ثانوية الصفا للبنين	160464	25.412	33.932	46.558	30.667	6606
	ثانوية بنات الأندلس		21.409	23.542	35.71	23.824	3463
	اعدادية حره المنطقة 92411		32.057	28.634	29.66	25.828	4
Erbel	ثانوية نوردن	174732	7.206	13.091	17.967	21.33	1607
	ثانوية نوردن 1996		26.807	34.758	29.748	25.734	34
Babel	ثانوية الحلة للمتفوقات بمدرسة بنات الهدى	422468	61.56	65.452	140.119	121.545	30395
	مركز المدينة 2-7	501984	64.375	71.236	170.886	132.13	26027
Sulaimania	شهرزور 181938		36.557	53.907	55.514	50.368	2

Table 3. Comparison the Data Retrieval Time for the Four Suggested Key Management Method Based on Column Store Database

DB NAME	Query	DB SIZE	First method column store DB	Second method column store DB	Third method column store DB	Fourth Method column store DB	NO.OF Retrived Data
Dept	كلية ابن الهيثم كيمياء	240	0.265	3.508	4.358	0.753	3
	96		0.265	3.501	4.433	0.711	1
	الجامعة التكنولوجية		0.386	3.498	4.381	0.749	6
Dhuk	ثانوية الصفا للبنين	160464	10.569	12.947	29.254	22.594	6606
	ثانوية بنات الأندلس		5.968	8.429	18.452	12.421	3463
	اعدادية حره المنطقة 92411		0.322	3.676	4.527	0.956	4
Erbel	ثانوية نوردن	174732	3.348	6.085	11.033	11.931	1607
	ثانوية نوردن 1996		0.378	3.49	4.656	1.007	34
Babel	ثانوية الحلة للمتفوقات بمدرسة بنات الهدى	422468	47.999	46.928	120.282	105.32	30395
	مركز المدينة 2-7	501984	52.323	54.818	146.939	115.671	26027
Sulaimania	شهرزور 181938		0.63	3.617	4.407	1.097	2

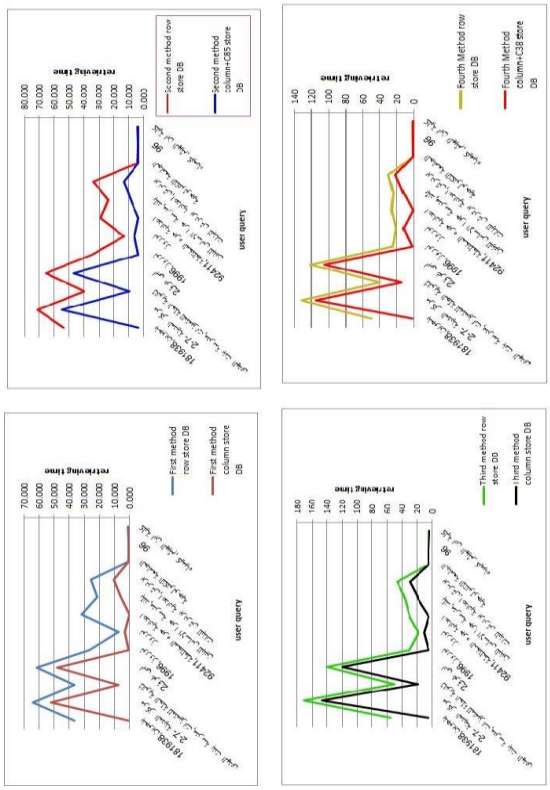

Fig.7. Comparisons Retrieving Time between Row Store Database and Column Store Database

Table 4. Advantages and Disadvantages for Each of the Proposed Key Management Methods.

	Fast Retrieval	**Caused load on the server**	**Security**
First key management method	***v***	***v***	***v***
Second key management method	***v***	***x***	***x***
Third key management method	***x***	***x***	***v***
Fourth key management method	***v***	***x***	***v***
Kerberos	***x***	***x***	***v***

From table IV it is noted that the fourth key management method has many advantages over the other key management and authentication protocol to exchange data in the secure and fast channel.

V. Conclusion

One of the most authentication problems is represented in how to exchange data in secure way between the client and server. The solution is by using secret keys in the encryption and decryption process. So the key management method must be used to exchange the keys between the two sites. In this work four key management methods are used. Different types of key management methods are discussed in this paper. In summary, symmetric key management schemes are described in four categories. Fourth symmetric key management scheme is much efficient as compared to group key schemes. First scheme has good security services, but it was caused load on the server site. Second scheme and third scheme have a problem of known the server key by the users. The first, third and fourth schemes provide better security because of using the user key in the encryption and decryption process. This means no one can open the message unless known user key. In this research, it is observed that fourth scheme of key management methods is highly secure and efficient as compared to other utilized symmetric key management schemes. It is mentioned that in the first key management method one key was needed while other three key management methods two keys are needed. The proposed key management methods solved the problem of slowing in message exchanged that was found in the Kerberos authentication protocol, it was needed for a number of operations to exchange messages.

References

[1] Gupta V., Singh G. and Gupta R., " Advance Cryptography Algorithm for Improving Data Security", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 1, January 2012 ISSN: 2277 128X.

[2] Awad Al-Hazaimeh O. Mohammad, "A New Approach For Complex Encrypting and Decrypting Data", International Journal of Computer Networks & Communications (IJCNC), Vol.5, No.2, March 2013.

[3] Hamdy M. Mousa, " DNA-Genetic Encryption Technique ", I. J. Computer Network and Information Security, 2016, 7, 1-9 Published Online July 2016 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijcnis.2016.07.01

[4] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, "Handbook of Applied Cryptography", Fifth edition, CRC Press, Inc, 2001.

[5] Barker E., Barker W., Burr W., Polk W., and Smid M., "Computer Security", National Institute of Standards and Technology, NIST Special Publication 800-57, July 2012.

[6] Chandramouli R., Iorga M., and Chokhani S., "Cryptographic Key Management Issues & Challenges in Cloud Services", National Institute of Standards and Technology, NISTIR 7956, September 2013.

[7] Yashaswini J, " Key Distribution for Symmetric Key Cryptography: A Review", International Journal of Innovative Research in Computer and Communication Engineering, ISSN(Online): 2320-9801, ISSN (Print): 2320-9798, Vol. 3, Issue 5, May 2015.

[8] Abdalla Michel, Fouque Pierre-Alain and Pointcheval David , " Password-Based uthenticated Key Exchange in the Three-Party Setting", IEE Proceedings - Information Security, Volume 153, Issue 1, March 2006, p. 27–39, DOI: 10.1049/ip-ifs:20055073 , Print ISSN 1747-0722, Online ISSN 1747-0730.

[9] Tsai Chwei-Shyong, Lee Cheng-Chi and Hwang Min-Shiang,"Password Authentication Schemes: Current

Status and Key Issues", International Journal of Network Security, Vol.3, No.2, PP.101–115, Sept. 2006 (http://ijns.nchu.edu.tw/).
[10] Stallings W., "Cryptography and Network Security Principles and Practice", Sixth Edition, Pearson Education, Inc., 2014.
[11] Jaydeep Sen, "A survey on Wireless Sensor Network Security", Technical Report 55-77, International Journal of Communication Networks and Information Security (IJCNIS) Vol 1, No2 August 2009.

Authors' Profiles

Rasha S. Ali received the B.S. in artificial intelligent in computer science from University of Technology and received the M.S. in computer science (data mining) from Baghdad University, in 2008 and 2013 respectively and received the PhD in computer science (secure retrieval from big encrypted data) from University of Technology, in 2017. His research interest includes intelligent systems, privacy, Security, management of big data, data mining and database applications

Alaa K. Farhan received the B.S. in computer science from University of Technology and received the M.S. in computer science (data security) from University of Technology, in 2003 and 2005 respectively and received the PhD in computer science from University of Technology, in 2009. His research interest includes intelligent systems, privacy, Security, management of big data, and image processing applications.

How to cite this paper: Rasha Subhi Ali, Alaa Kadhim F.,"Security Protocol of Keys Management System for Transmission Encrypted Data", International Journal of Computer Network and Information Security(IJCNIS), Vol.10, No.1, pp.10-17, 2018.DOI: 10.5815/ijcnis.2018.01.02

