International Journal of Security and Its Applications
Vol. 10, No. 9 (2016), pp.275-284
http://dx.doi.org/10.14257/ijsia.2016.10.9.26

Lightweight Key Distribution Protocol for Streaming Media DRM

Hao Li, Tang Xuyue, Jiayin Tian, Jianbo Liu and Cheng Yang

School of Information Engineering, Communication University of China, Beijing 100024, China;
muyue_8710@163.com, 15652906586@163.com, 34tianjiayin@163.com, cafeeyang@163.com

Abstract

In the daily-real systems, protecting the data only is far less enough. The process about the generation, distribution, storage, and the revocation of the key is the core problem in the system-security consideration. If the management of the key is insecure, attackers could easily get the key used in the encryption steps to the context, leading to useless processing in the encryption no matter how secure the system be.

Unlike the period of validity for the different key, the key can easily be attacked by various methods due to its distribution through the complex net links, which would bring about the threatening of the security. In this paper, we analyze three models of the key distribution, associated with the actual applications for the encryption in the hypermedia video stream. Besides, we raise up a new protocol based on the IKEV2 distribution protocol and evaluate the stability when it suffer the Man-in –the-Middle Attacks, DoS and Replay attacks.

Keywords*: key management, lightweight, low power consumption, KEP, KDP, DoS, MITM, replay attack*

1. Introduction

The period of validity for the key consists of the generation, distribution, identification, renovation, storage and revocation. Each step is relevant to the knowledge based on the Cryptography.

The key's generation faces the challenges like the weak key problem in the DES algorithm. [1] The identification of the key usually take Hash functions to use such as the MD5 and SHA algorithm. According to the theory raised by Shannon in the Information Theory, the One-time pad model gain the most secure status in key update. The renovation plays an essential role in the process of the real-time communication. While the data needs to preserve for a long time after being encrypted, we had better use the Perfect forward secrecy in order to ensure it is time-efficient.[2] The security on the terminals are much concerned in the key's storage. Right Issuer take the right Encryption key to encrypt according to the OMA DRM, and then it is involved in the RO package encrypted by the DRM Agent.

Compared to the other parts, the key distribution needs to be transferred to kinds of terminals through the complex environment of the network, therefore it face more security challenges, which is the focus of this paper. According to the model of the secure communication set by Shannon, key distribution is transmitted through the secure channel specified in the picture. And such a secure channel can be varied, preset key based and PKI based.[3] Zimmermann proposed a method based on SAS(Short Authentication String) to resist the Man-in-the-Middle Attack of the key exchange in Diffie-Hellman to establish a secure channel in the ZRTP. [4]

This paper is organized into four sections. We briefly introduce the management of the key and its period of validity in this section to raise up the current problem of its distribution. In the next section, we analyze different distribution models. In section three, two sets of key distribution protocols with different orders are proposed on the basis of the problem of the difference between server and client. The anti-attack performance of these protocols is analyzed in the section four.

2. Three Key Distribution Models

No matter which kind of key distribution scheme, we are supposed to execute the mutual authorization and authentication to ensure the security of the connection through the transmission between client and server. It is also protect the channel for the key distribution. The whole step above is called the security association. The both sides of the communication need to negotiate the configuration information such as encryption algorithm, initial offset, Hash algorithm and so on. There are three main methods of the security authorization currently. [5]

These three methods have advantages and disadvantages in the key distribution and management system.

The method of presenting the password provides a simple encryption operation, which can meet the requirements of low power consumption and low computational requirements for mobile phones and other devices. Device ID [6] and email [7] are used to be a unique identifier. However, due to the user's access to the server and demand is sudden, and time is relatively concentrated, a large number of pre key query work increases the burden on the server.

The complete PKI authentication model need the participation of the third party CA, and at the same time need to streamline the certification process. Compared with the Symmetric encryption algorithm, asymmetric encryption consists of more operations on the encryption and decryption based on public key encryption technology, meanwhile it cannot meet the standard mentioned before as the low power consumption and easily-authorized. [8]

DH is also the asymmetric encryption principle with key sharing instead of the complete PKI model, not requiring the trusted third party CA. It can meanwhile ensure the relatively small encryption and decryption computation, but would easily be attacked by MITM. Zimmermann proposed a SAS (Authentication String Short) based method in ZRTP to protect DH from it. In the first landing, the two sides will verify the SAS, and save every required verification through the SAS key (continuity key). For the verification of the SAS itself, ZRTP proposed two kinds of methods: one of it is to read out the SAS to be paired in the same time; the other method is to verify by an already existing PKI. [4]

3. Proposed Protocols

The purpose of this paper is to achieve a complete set of DRM for the Hypermedia, key management center (KMC), digital content encryption system (DCES) and digital content system (DCDS), which is the three part of the client. Therefore the kinds of key distribution can be divided into two parts. The digital content encryption system (DCES) is supposed to encrypt the hypermedia documents, after applying for the encryption key through the key management center (KMC). In addition, digital content decryption system (DCDS) decrypt the hypermedia documents, which also need to apply for the decryption key at first.

Therefore we proposed two different sets of key distribution protocols for the different situations in this paper. The first is the KEP Protocol[1] between Key Management System and Digital Content Encrypt System. The other is the KDP Protocol[2] between Key Management System and Digital Content Decrypt System.

[1] Key distribution protocol between key management system and digital content encrypt system.

[2] Key distribution protocol between key management system and digital content decrypt terminal.

3.1. KEP Protocol

Security is the key transmission between KMC and DCES belongs to Server-side secret sharing operation. The interaction among the servers differs when it is between the server and the client. And it is of different characteristics:

1. The server has strong ability of calculation.
2. The server requires higher security level.
3. The location of the server, including its actual location and the network location, is relatively fixed, and of course the mobility of the new agreement or the mobility of the protocols is relatively small.

Based on the three features above, KEP propose the DH algorithm [9] as the core strategy to carry on the secret sharing. As the DH algorithm is vulnerable to DOS attacks, MITM attacks and replay attacks, KEP use the appropriate method of defense in the different stages of the protocols respectively.

The message packet is initialized as shown in the following table:

Table 1. KEP Packet

name	length	initial value	meaning
IP	33byte	NULL	sender IP
StepID	4byte	0	current step of the protocol
CKY_I	17byte	0	sender cookie
CKY_R	17byte	NULL	receiver cookie
p	8byte	0	DH element
g	8byte	0	DH element
MESSAGE_code	4byte	-1	type of the information
DH_ClientData	8byte	0	$\mathrm{g}^a \bmod p$
DH_ServerData	8byte	0	$g^b \bmod p$
EncData	257byte	NULL	
MD5	17byte	NULL	

The meaning of some columns in the table is as follows:

EncData: The nonce of the public-key encryption is used by the sender or the responder, which can achieve the purpose of authentication to prevent the middle attack and the replay attack. There are two generating formulas. $E\{N_i\}_{Kr} / E\{N_i + N_r\}_{Ki}$

MD5: This value is used to verify the integrity of the message, and can be generated in two formulas: $MD5(K_{ir}, CKY\_R | CKY\_I | g^a | g^b | N_r)$ 、 $MD5(K_{ir}, CKY\_R | CKY\_I | g^a | g^b | N_i)$. $K_{ir} = MD5(0, N_i | N_r)$, N_i and N_r is respectively the nonce generated by the sender and the responder.

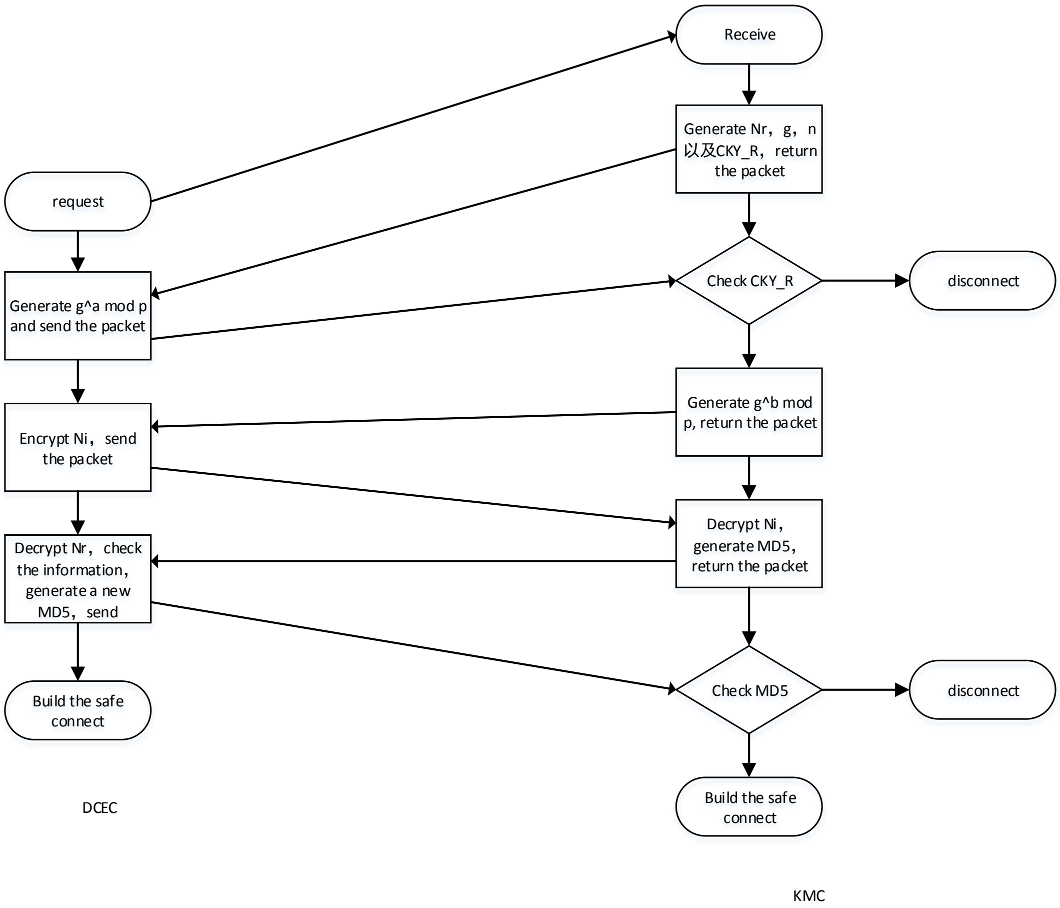
Receive
Generate Nr，g，n 以及CKY_R，return the packet
request
Generate g^a mod p and send the packet
Check CKY_R
disconnect
Generate g^b mod p, return the packet
Encrypt Ni，send the packet
Decrypt Ni，generate MD5，return the packet
Decrypt Nr，check the information，generate a new MD5，send
Check MD5
disconnect
Build the safe connect
Build the safe connect
DCEC
KMC

Figure 5-7. KEP Flowchart

The two parties of the communication generate the shared secret key by the following formula:

$$\left(g^{b} \bmod p\right)^{a} \bmod p = \left(g^{a} \bmod p\right)^{b} \bmod p$$

KEP protocol has been completed at this time and the both sides of the communication used the same shared key. In the implementation process of KEP, we have added the cookie, the RSA public key encryption and the nonce-using to resist the threat from DOS, MITM and replay attacks on the protocol framework and protocol content.

3.2. KDP Protocol

The key transmission between the KMC and DCDS is the transversion of that the client terminal apply for the decryption key from the server and decrypt the document by the key, which is the typical model of the interaction between the server and the client. From now on, we define the sender as the client and the responder as the server during the communication. Compared with the server to the server, the communication between the client and the server has a new objective environment.

1. The client is of various species, such as a PC, laptop or mobile phones, iPad, which means the computing ability is at different level.

2. The client DCDS program requires low power consumption, which is to ensure that devices with no battery can be used on the long time.

3. The requirements of the security level are lower them of the server to server.

4. The location of the server, including its actual location and the network location, is random. And the mobility of the new agreement or the removement of the protocols is relatively high, and time-efficient.

KDP protocol is in on the basis of the Oakley protocol. According to the characteristics of client and the difference status of the sender and responder, an improved key exchange protocol with authenticated safe is proposed to defend against DoS attacks of a certain extent, and to identify the MITM attack, denial of replay attacks.

Compared to the message structure of the KEP protocol, the message structure of the KDP protocol add a record of the client log-in user name. In order to reduce the computation of the certification stage, the protocol between the client and the server use the username and password to authentication, which can reduce the amount of computation by using the user name and password applied before as the preset key and by using a symmetric encryption replace RSA public key encryption.

The message packet is initialized as shown in the following table:

Table 2. KDP Packet

name	length	initial value
IP	33byte	NULL
name	25byte	NULL
StepID	4byte	0
CKY_I	17byte	0
CKY_R	17byte	NULL
p	8byte	0
g	8byte	0
MESSAGE_code	4byte	-1
DH_ClientData	8byte	0
DH_ServerData	8byte	0
EncData	257byte	NULL
MD5	17byte	NULL

The exchange process of shared keys between the two parties:

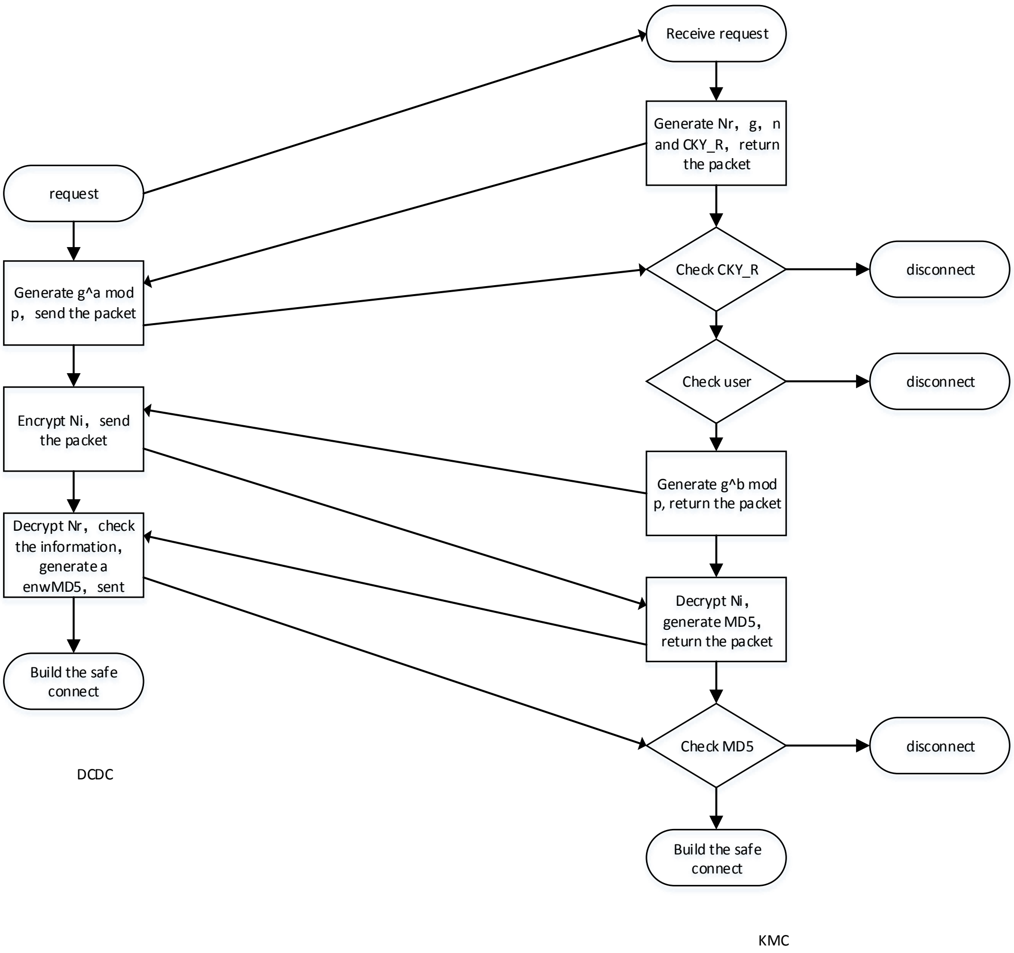
Receive request
Generate Nr，g，n and CKY_R，return the packet
request
Generate g^a mod p，send the packet
Check CKY_R
disconnect
Check user
disconnect
Encrypt Ni，send the packet
Generate g^b mod p, return the packet
Decrypt Nr，check the information，generate a enwMD5，sent
Decrypt Ni，generate MD5，return the packet
Build the safe connect
Check MD5
disconnect
DCDC
Build the safe connect
KMC

Figure 5-8. KDP Flowchart

The two parties of the communication generate the shared secret key by the following formula:

$$\left(g^b \bmod p\right)^a \bmod p = \left(g^a \bmod p\right)^b \bmod p$$

4. Ability Against the Three Attacks

The most common DoS attacks have two ways. One is for the attacks of computer network bandwidth. The other is the attacks to the computer itself connectivity performance. [10] The Oakley protocol proposes a "Cookies" program to defend against denial of service attacks, and the Cookies program is proposed by Karn Photuris in the Phil key exchange protocol. [11] KEP protocol and KDP Protocol are capable against DoS attack. Both protocols use the cookies procedure, which demands the DCES/DCDS sent the request to the KMC for new key/decryption key as starting the first step. And this step commands the DCES/DCDS must send their cookie to KMC and the value of CKY_I cannot be 0. Then the responder use the CKY_I receive to generate CKY_R, and returned it to the sender. When the sender response the CKY_R, it can verify the correctness of CKY-R, resulting in determine whether there is a false IP attack which may be the possibility of DoS attack. As the sender cannot receive the data came from responder in the cookies procedure if the IP of responder in incorrect, an error can be occurred during the verification of the responder so that the use of false network address of the malicious sponsor can no longer continue to implement the protocols thereby resisting the denial of service attacks.

As for how to defend the MITM attack, Xiao Daoju proposed the idea at the beginning of his paper that if the client cannot provide authentication or get the verification of the server's certificate, the attackers will entirely eavesdrop the conversation by using the man-in-middle attack. [12] KEP protocol and KDP Protocol are capable to defend against MITM attack. With adding Eve existing between Alice and Bob who can arbitrary bug and modify the information in the transmission of Alice and Bob, the attack can be defined as success if Eve can generate a shared key with Alice and Bob trough the KEP protocol and KDP Protocol.

It can be clearly seen in the following list shown as what Eve can obtain from the KDP protocol:

Table 5-7. The Data Status during the KDP Protocol

Alice		public channel	Bob	
privet	public		private	public
N$_i$, pwd	$CKY\_I, name, MD5(pwd)$	①→		
		←②	CKY_R， p， g	N_r
a	$g^a \bmod p$	③→		
		←④	$g^b \bmod p$	b
	$E(N_i)_{pwd}$	⑤→		
		←⑥	$E(N_i + N_r)_{pwd}$ $MD5(K_{ir}, CKY\_R \mid$ $CKY\_I \mid g^a \mid g^b \mid N_i)$	
	$MD5(K_{ir}, CKY\_R \mid$ $CKY\_I \mid g^a \mid g^b \mid N_r)$	⑦→		

As to the MITM attack, Eve can bug and modify the information in the insecure transmission between Alice and Bob. We discuss the KDP process firstly. After the process four, Eve establish a shared key with Alice and Bob respectively, which is shown below:

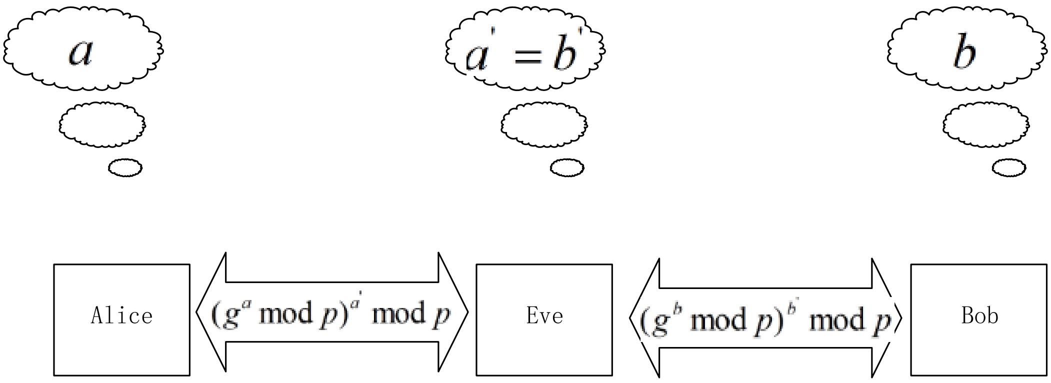

Figure 5-8. MITM Attack

After the process five, due to Eve did not know pwd , namely the encrypt key of N_i , Eve cannot decrypt the content of $E(N_i)_{pwd}$, which means that N_i is unknown. In the process six, after decrypting the $E(N_i)_{pwd}$ to get N_i , and transferring the $E(N_i + N_r)_{pwd}$, $MD5(K_{ir}, CKY\_R | CKY\_I | g^a | g^b | N_i)$ to Alice, Eve cannot obtain the N_i end N_r , resulting in the indeterminacy of MD5. Of course at the side Alice, failed check will be found if the value of MD5 be modified after decrypting and calculating the used MD5 algorithm $MD5(K_{ir}, CKY\_R | CKY\_I | g^a | g^b | N_i)$. After the same steps later, Bob will be able to check the value passed over by Alice, which prevents MITM attacks.

KEP protocol takes the similar principle. The only difference is that, KDP protocol with key pre distribution, using the username and password, using symmetric encryption system to check the value of encryption, and KEP protocol used the RSA public key encryption. In the compare, KEP protocol has a better security level, but the energy consumption is larger, while the KDP protocol is of lower secure ability, but it also reduces the energy loss.

Replay Attacks, is also known as Freshness Attacks. Liu Jiafen and Zhou Mingtian proposed classification of the new replay attack. The first one is the lack of novelty checking mechanism leading to replay attack. The next one is the lack of the main signs leading to replay attack. The third one is the same information format leading to replay attack. The forth type is of category lack leading to replay attack. The last one is multi protocol interaction leading to replay attacks. [13] There are two main schemes for preventing replay attacks, which are time stamp [14] and challenge response mechanism [15]. KEP/KDP protocol is also available for defense of the replay attack. KEP/KDP protocol is through the nonce-way to defend the replay attack. Eve can obtain the certification information through the protocol exchange between Alice and Bob which are $E(\mathrm{N}_i)_{pwd}$, $E(\mathrm{N}_i + \mathrm{N}_r)_{pwd}$, $MD5(K_{ir}, CKY\_R | CKY\_I | g^a | g^b | N_i)$ and

$MD5(K_{ir}, CKY\_R \mid CKY\_I \mid g^a \mid g^b \mid N_i)$. And Eve attempts to disguise the identity of Alice/Bob through these old authentication information and exchange with the Bob/Alice protocol to generate shared keys. We can prevent from using the old certification information by nonce-using. Both side of the communication will generate the random number N_i and N_r at the beginning of the protocol exchange. Certification process need to add the encryption information including the nonce, and protocol KEP/KDP preset username and password and RSA public key encryption mode to encrypt and exchange of the nonce-using respectively. The MD5 value is used in the feedback, and if one side of the two is currently reproducing information, MD5 authentication will fail.

In the step five, if Eve camouflage Alice to communicate with Bob and the value Eve transfer to Bob is outdated, Bob will send the MD5 to Alice (Eve) which is combined with the outdated information and N_{rNew} generated by Bob. The information Eve return back to Bob resulting from Eve's unknown N_{rNew} generated by Bob leads to unsuccessful Hash verification ($N_r \neq N_{rNew}$).

Instead, if Eve camouflage Bob to communicate with Alice, is unapproachable for Eve. And after the step six, Hash verification will fail at the side of Alice.

Acknowledgement

The work on this paper was supported by the National Science and Technology Support Plan (2014BAH10F00) and the New Century Talent Scheme from Education Ministry of China(NCET-12-0944).

References

[1] Wang Y., "Algebraic structure of the DES's weak keys[J]", Journal of Xidian University, **(1989)**.

[2] Orman H. and Orman H., "RFC 2412: The OAKLEY Key Determination Protocol [J]", IETF RFC 2412, **(1998)**.

[3] F.-C. Kuo, H. Tschofenig, F. Meyer and X. Fu, "Comparison Studies between Pre-Shared and Public Key Exchange Mechanisms for Transport Layer Security", 9th IEEE Global Internet Symposium 2006, **(2006)**.

[4] M. Petraschek T H O J H H W G. Security and Usability Aspects of Man-in-the-Middle Attacks on ZRTP. [J]. Journal of Universal Computerence, vol. 14, no. 5, **(2008)**, pp. 673-692.

[5] R. Pecori and L. Veltri, "A Key Agreement Protocol for P2P VoIP Applications, Software, Telecommunications & Computer Networks", 2009. SoftCOM 2009. 17th International Conference, **(2009)**.

[6] Kuo F., Tschofenig H. and Meyer F., "Comparison Studies between Pre-Shared and Public Key Exchange Mechanisms for Transport Layer Security[C]", //Infocom IEEE International Conference on Computer Communications. IEEE, **(2006)**, pp. 1-6.

[7] Garfinkel S. L., "Email-based identification and authentication: an alternative to PKI?[J]", IEEE Security & Privacy Magazine, vol. 1, no. 6, **(2003)**, pp. 20-26.

[8] Zhang J., Chen H. and Geng Q., "An Efficient Certificate-Based Signature Scheme without Pairings[J]", Second International Workshop on Computer Science & Engineering, , vol. 2: **(2009)**, pp. 44-48.

[9] Rescorla E., "RFC 2631: Diffie-Hellman Key Agreement Method[J]", IETF RFC 2631, **(1999)**.

[10] He L. I. and Prof A., "Prof. Research on Safeguard Techniques against DoS and DDoS Attacks[J]", China Safety Science Journal, **(2009)**.

[11] P. Karn and W. Simpson, "RFC 2522: Photuris: Session-Key Management Protocol[J]", IETF RFC 2522, **(1999)**.

[12] Daoju X., Guo J. and C. Xiaosu, "Research on a Defending Strategy for Man-in-the-Middle Attacks[J]", Computer Engineering & Science, vol. 26, no. 9, **(2004)**, pp. 7-10.

[13] Liu J. F. and Zhou M. T., "Research and taxonomy of replay attacks on security protocol[J]", Jisuanji Yingyong Yanjiu/ Application Research of Computers, vol. 24, no. 3, **(2007)**, pp. 135-139.

[14] Denning D. E. and Sacco G. M., "Timestamps in Key Distribution Protocols [J]", Communications of the Acm, vol. 24, no. 8, **(1981)**, pp. 533-536.

[15] Mitchell C., "Limitations of challenge-response entity authentication[J]", Electronics Letters, vol. 25, no. 17, **(1989)**, pp. 1195 - 1196.

Authors

Hao Li, He is a PhD student studies in the Communication University of China. His current research interests include digital right management, key management and users' privacy protection.