Bulletin of Environment, Pharmacology and Life Sciences
Bull. Env.Pharmacol. Life Sci., Vol 4 [Spl issue 1] 2015: 97-104
©2014 Academy for Environment and Life Sciences, India
Online ISSN 2277-1808
Journal's URL:http://www.bepls.com
CODEN: BEPLAD
Global Impact Factor 0.533
Universal Impact Factor 0.9804

FULL LENGTH ARTICLE **OPEN ACCESS**

Password Authentication Scheme in the Peer-to-peer Network and Security Analysis

Amir Hossein Rahimi[1], Ali Zaghian[2] and Maryam Rahimi[3]
[1] Department of Mathematics and Cryptography Malekashtar University of Technology, Isfahan, Iran, and Department of Mathematics, Qom Branch, Islamic Azad University, Qom, Iran. (Gmail: Amir.Rahimi361@Gmail.Com)
[2] Department of Mathematics and Cryptography Malekashtar University of Technology, Isfahan, Iran. (Email: A_Zaghian@mut-es.ac.ir P.Box:83145/115)
[3] (Gmail: AHRahimi.1982@Gmail.Com)

ABSTRACT
Depending on the way of computer configuration and information access in the network, the networks are divided into two main groups: Peer-to-Peer and Client-Server. In peer to peer networks, there is no specific server and no hierarchy in terms of computers. Authentication in P2P environment is owned by the public keys, and is done by establishing a secure mechanism and storage of cryptographic keys on a user device. The basic factor for password-based authentications in P2P systems includes registering a user account and making a connection. In this paper, an efficient scheme is presented to support logging in to systems based on users' user name-password authentication and Client-Server systems. In addition, password authentication protocols to log in a P2P networks such as account registration, login, password change, re-login, log-out of a remote device, password retrieval, restarting the forgotten password via e-mail or security questions will be introduced.
Keywords: *peer-to-peer (P2P) network, password authentication, user account, login, password retrieval, device.*

INTRODUCTION

Depending on the way of computer configuration and information access in the network, the networks are divided into two main groups of Peer-to-Peer and Client-Server. In peer to peer networks, there is no specific server and no hierarchy in terms of computers and all computers are considered equivalent and equal, and each computer in the network acts both as a client and a server. Therefore, central security is not usually required and security is often provided locally on each computer, and users specify which data can be shared on their computer based on classified documents. Peer to peer networks have a maximum of ten computers.
Client-Server: due to growth in the networks and increased users and resources, a peer to peer network is unable to meet the high volume demand for common resources. To keep up with increasing demand and provide the required services, networks should use specific servers. Client - server networks are considered as a standard model for setting up networks. As the network grows , "the number of computers connected , physical distance , current traffic" , the number of network servers can be increased.
Most P2P systems do not authenticate users. However, as long as issues such as P2P storage, system supports, and the importance of online social networks exist, user Authentication would be an acceptable or even preferable requirement. In such applications, the data available depend on a Client. Our goal is to create a username- password authentication component through encryption keys in P2P systems.
P2P schemes are usually used to re-log in , change the password , and retrieve a forgotten password. In such schemes , a user selects a device and can easily enter into the system. In fact , the device stores username and password data in the form of the original text , and for password change , it is required to know the old password. Authentication in P2P environment is owned by the public keys , and is done by establishing a secure mechanism and storage of cryptographic keys on a user device and sharing the public key file should always be unknown. One drawback to this method appears when the system users have access to multiple devices. Another disadvantage is that the users need to transfer keys , or assume a new username , that increases the complexity. In addition , using passwords increase security risks for the

user , in the that an intruder can easily copy switches between devices. Next problem is re-login to the system. In fact , a thief cannot be prevented from accessing an account , but using standard and safe proposed security protocols , the thief cannot change the username and password , and also the authorized user can always revoke re-certifications that are a tool for theft. Authentication in P2P environment is used in systems such as OneSwarm , Safebook , and Tribler , and works well until the user needs to access the services of a secondary device.

RELATED USER

Issues of the secure authentication IDs in P2P systems have already been studied by researchers such as Aberer , Datta , and Hauswirth. In these systems, user authentication is performed through nodes and node authentication is performed through a sign key , that is automatically created and stored in the node. However, equalizing a node to a user with multiple devices fails and for each user , a personal device must be used. The main problem in P2P systems is data retrieval and authentication system support. However, re-login and forgotten password retrieval does not have any position in the best techniques for password authentication. Also some P2P storage systems, use techniques that are related to personal characteristics of the user. For example, GNUnet systems based on DHT decision trees and Free net use keywords field to obtain a public-private key pair , i.e. use a private key for data signaling and a hashed public key to identify data storage. Both of these systems used a series of keywords where users could only identify the remembered keyword string to obtain and store data. In terms of the forgotten passwords , data retrieval in P2P systems have already been studied by Vu. He proposed a secret sharing composition according to the predefined time threshold by selecting passwords and encryption sharing's. Frykholm and Juels proposed a password retrieval mechanism based on security questions. But the problem of password change had not been resolved in their scheme. In this paper , we propose a scheme that completely solves the password change problem.

Table 1: scientific terminology protocol

Username	$uname$
password	$passwd$
Random bit sequences	$salt$
Encryption key for authentication insertion	K_W
Key storage file	F_{KS}
Fila nameF_{KS}	f_{KS}
Encryption key to encrypt the fileF_{KS}	K_{KS}
Login data file , file name , and related key	F_{LI} ,f_{LI} ,K_{LI}
Device login data file , file name , and related key	F_{DL} ,f_{DL} ,K_{DL}
Device user and D ID	D ,D_{ID}
Encryption keys for after-login application	K_{x1} ,K_{x2} , …..
Routing from device IDs to device login data files and related keys	$devmap$

The proposed protocol analysis

This paper presents a scheme to support login based on username-password binary authentication and client- server system. Also , asymmetric password authentication protocols are investigated to log in a P2P networks such as account registration , login , password change , re-login , log-out of a remote device , password retrieval , restarting the forgotten password via e-mail through security questions or behavioral characteristics (biometrics).

The proposed system protocols are based on DHT , where a pair of random sampling protocol , and a distributed data storage as DHT are required to search a user. Both DHT and distributed data storage are related to P2P system protocols , except for the DHT that keeps the user account registration. In such a system , if a user account is registered once , no one else can register the same username. Two operations are needed by DHT , put (key , value) and (key) get , that respectively retrieve insertion of the value/key operator , and performing operations on that key/value. Therefore , once DHT runs registration operations , and then replaces the operation with the same key , which will not affect the system mode. The use of distributed data storage functions are similar to DHT , with three differences: distributed storage is allowed to select "filename" for the user , data are need to be updated; and access control is assumed when data is recorded.

API storage gradually becomes official with three operations: first , (Data) Create that creates a new data file and retrieves the file name. Second , (filenameWrite data) that inserts a file named **filename** with content of **data**. Third , (filename) read that reads the contents of a file. Secure protocols does not need a copy of data for non-access of others , and each stored file has an owner. In fact , owner is the user that has created the file , and only the owner of the file can runs (Write) operation. For authentication of file ownership , we consider a public key cryptography system. Finally , for the components of pairs sampling , in a **get (peer)** method is required to retrieve a randomly selected pair with a uniform distribution package. BitTorrent Mainline DHT protocol (Distributed protocol that makes distribution software possible and shares very large peer to peer files) is recommended with two different Client designs.

Login based on P2P password

This process facilitates logging in a safe system and equal and correct access to services on the server for the user. For password-based authentications in P2P systems , the basic factor for account registration is communication and storage way of information on a device. The following requirements are defined from the ISO 27002 modern technologies standard for P2P systems login , and user ownership of his/her user accounts would be added to certain devices.

• Passwords should not be stored or transferred in the form of the original text "Resistance to impersonation attack , self , maintaining user anonymity".

However, it cannot be denied that if a server's administrator receives user ID and password in the form of original text during the registration phase on the network , he may act to impersonate registered users.

• A user should be able to freely select and change his passwords , so that there would no need to restart the system for changing password. The chances of finding the system default passwords using trial and error would also be reduced.

• The files with password contents should be stored separately from data application to reduce the context for data distortion and manipulation attacks , password revelation , eavesdropping , impersonation , and password guessing.

• Considering an expiration date for each password to prevent repetition attacks and impersonation attacks.

• Password should not be found in dictionaries , and should consist of a combination of mixed numbers and letters with more length , so that intruders could not guess the password with trial and error methods.

• System scalability should be high. "increased network users should not reduce its efficacy of server".

• * A user should consider the same password to log in through any device.

• * Each user should not retrieve a password by a stolen device with a digital certificate.

• * User should be able to access a user account block from a stolen device , to be able to easily change secret keys "revealing the server secret key ".

This standard also defines restrictions on failed password logins and the maximum and minimum log time. For example , the proposal to implement a security alarm system and remote management technology based on SMS , limit the number of failed logins , and lock user accounts to prevent guessing attacks. However , this proposal cannot remove all defects and deficiencies in P2P networks in this modern world of growing technology.

User account registration

To register a new user account , the user first selects a username **uname** and password **passwd**. Then , the user creates a file to store key (key data center) , and next creates a symmetric K_{KS} key , and encrypts the content of F_{KS} storage file with this key , and saves the encrypted text to the file named f_{KS}. Then , user creates a log in file of F_{LI} with a random byte field of **salt** and obtains asymmetric key K_{LI} from **passwd** password and **salt**. Then f_{KS} , K_{KS} and authentication certificate K_W is encrypted with $K_{LI.}$ It also saves **salt** and three previous encrypted values with the file name of f_{LI}. Salt is stored in the main text , so the next user can obtain K_{LI} decryption key only by accessing the password. Finally , the user runs operations of *puton* insertion on DHT with **uname** as the key , and f_{LI} as the value. Now , if the username is captured , then the user must apply for a new username and carry out all operations once again record it in the system.

Login

A user already registered can login to the system , retrieve encryption keys stored in the storage file of key F_{KS} , by entering the username and password from each device. With a request with **Uname** parameter to DHT , the user gains a result with file name of f_LI for login data file. However , this distributed stored file is retrieved , including salt in the form of the original text. Next , the username feeds a key extraction function and user password to obtain K_LI key; that enables user to decrypt all the contents of the login file. Finally , the user stores storage file F_KS from the system , and decrypts it with K_KS key; so , now the way of user login to the system using P2P is in hand of K_x1 , K_x2 , ... keys , and these keys must be kept in secret. If the user wants to login again on a local device , a new device will be created for login data file to system F_DL. Remember that the in device , only f_DL file and K_DL key are locally stored. In addition , a reference is added to the login data file to device with value of *Devmap* that includes a (mapping) function

from ID password of the device. When the user wants to re-login from the same device , then the locally stored values (K_DL , f_DL) are used to retrieve data of login file and decrypt it. Whereby , access to key storage file will be achieved. Therefore , re-login feature enables user to login without entering a password. Also re-logins to systems remain valid even if some keys change in key storage file.
However , there are mitigation measures to prevent intruders activities , so that if the login request is rejected three times , then the user account is locked automatically. "Implementing remote alarm system by sending an SMS to the administrator to prevent an intruder and at the same time sending a text message to a central server to lock the intruder's system".

Table 2. Retrieval protocol glossary

Retrieval through Security question	
sharing (n ,k)secret sharingK_{LI}	q^{S_i}
Challenging security question	Q_i
Answer to questionQ_i	A_i
Random byte sequence	q^{Salt_i}
Key withq^{S_i} sharing encryption	q^{K_i}
Retrieval through E-mail	
Conditioning retrieval key to login	K_R
sharing (n ,k)secret sharingK_{LI}	e^{S_i}
Retrieving user's e-mail address	Email
randomly selected peer	$peer_i$
Random byte sequence (in search for E-mail authentication)	$esalt_i$
Random byte sequence (in search for eK_i)	$ksalt_i$
Encryption requirement to E-mail address	C_i
Key withq^{S_i} sharing encryptioneS_i	eK_i

Password change
At this point , to change the password by using the old password , the user logs in to K_LI. With this information , password change would be done freely , and server would not interfere. Therefore , user apply for a new password and creates a new salt. The key extraction function is used to create a new K_{LI}^{new}key for login data file. Then , the content of key storage file is decrypted by the old key , and a newK_{LI}^{new} key is created and used again for storing encrypted content of the key. We note that the content is previously stored in the storage system , in name of K_{LI}^{new}. Finally , the login file is updated: K_{KS}^{new} , f_{KS}^{new} , writing K_W certificates and designing a new empty device $devmap^{new}$that is encrypted with the new keyK_{LI}^{new}. The encrypted text is inserted plus a new salt for distributed storage , finally by using the reference file name f_{LI} certification K_W , the *write* operator will be authenticated. Eventually , the keys stored in the key repository , using P2P protocols will be updated. Doing this , the potential for insider attack , and confidential data retrieval attack decreases.
Logout
To logout , a user does not require to interact with DHT or storage system. Clearly , a local repository data is deleted and all of the essential keys for each login are retrieved. If a user is selected to re-login on another device , the corresponding login data file to device , F_DL , can be removed from storage on the system.
Problems related to logout is to cancel certifications on re-login on another device (e.g. , a stolen phone of a user).
To accomplish this step , first the password change operation is performed , so that all devices are locked by re-login to the system , since K_KS storage key , as well as f_KS file name changes. Later , devmap routing device is used to inform every device in connection with the new key (file name) , with the exception of the device that is canceled. For informing a device about changing , the corresponding values are updated in F_DL log file , that can be accessed to from the device using local storage certificates. After the execution of password change operation , all devices that should not be canceled , offer re-login to the system (referred to devmap routing device). The name of f_DL login file and K_DL key are readable and the new key stores K_{KS}^{new}key and name of f_{KS}^{new} file is encrypted to F_{DL} under the K_{DL} key of device. Finally , modified devmap is stored on login data file F_{LI}.
Password retrieval
As part of the password-based login , there is a potential for users to retrieve their user account (as a mechanism to retrieve the password) , if users forget their password. The goal of password retrieval mechanism is providing a secondary method for user authentications. Three unique mechanism to

retrieve passwords including password guidelines , security questions , and email-based retrieval will be discussed.
For the login procedure , the set of basic requirements for the password retrieval are defined according to the ISO 27002 modern technologies standards. A set of our own requirements are also added to the list (which are predetermined by an asterisk).
• Authentication methods of a user's former ID to allow the user to select a new password.
• Connection under influence of security incidents retrieval "the regular removal of security patches and elimination of ways to intrude to the system."
• Ways to allow the retrieval and restitution of business operations and accessibility of data in a scalable time manner "Traceability in certain cases with maintaining security , trust and anonymity features "
• An authorized user should be able to retrieve the broken keys of device or lost keys (forgotten) "without worrying about the possibility of disclosing the password."
• * The retrieval method should enable the user to replace a new password , without disclosing the old passwords.
• * The retrieval method should be easy to use.
• * The important data should be kept confidential to be retrieved "Resistance against insider attack".
If a password is restarted via security questions alone , the system would not be tended to " be affected by that". In the proposed system , it is important to note that no one knows the answer to the security questions of users , Since many systems ask similar security questions , some detailed annexes must be attached to the component of previous protocols.
Retrieval based on security questions
Security questions are a password retrieval technique that trusts to answers to the questions asked from the user during registration. The answer should be such that it cannot be easily guessed or sought by an intruder.
Suppose the user has provided , n A_i answers to Q_i security questions. To retrieve the password , the user is required to correctly answer each K output of n questions. Selecting a K-combination is a relationship between security and ability. A successful retrieval of K_LI key to login data files , allow the user to change the password. This process does not require the user to provide a new answer after a regular password change. In addition , it avoids storage of the original answers to the questions. To make a retrieval mechanism based on the security question , first , n sharings of qS_1 , ... qS_n are created under a secret sharing scheme (k , n). For each share , a *qsalt*$_i$ is created , which obtains a qK_i key from the salt and answer A_i that are used for sharing encryption to achieve S_i^{enc}. Finally , the login data file is developed with all Q_i questions , $t_i$salts , S_i^{enc}encrypted sharings , and K_i^{enc}encrypted keys. When the key is retrieved , the user has reproduced at least K replies , which are stored together with the salt and can be used to obtain K keys of K_i , which in turn can decrypt K qS_i sharings. When K_{LI} is changed (for example , leads to a regular password changes) , for the new key of K_{LI}^{new} , a new set of sharing will be created.
We imply that no one of qK_i keys do not change , salt_i salts , so the user can still use the same answers for retrieval. Finally , the updated sharings are stored as backup to data log files allowing the encrypted keys to be updated more.
Retrieval based on email
According to the password retrieval based on email , user sends an email with some information. This link will be sent to the email address with which he/she has registered his/her accounts. This scheme , is confirmed by selecting a random pair of number , followed by using secret sharing (k , n) to retrieve the password. To provide a stable retrieval mechanism , independent of the change in one key K_{LI} (for example , as a result of password change) , as a result of retrieval , a retrieval key of K_R , causes the last model K_{LI} to be always encrypted. K_R retrieval key and n sharing of keyeS_1 .... eS_n1 is created by using secret sharing of (k , n).
For each share , a pair of *peer*$_i$ is randomly selected and two salts of $esalt_i$ and $ksalt_i$ are created and an authentication (confirmation) encryption of C_i of salts $esalt_i$ and $ksalt_i$ is obtained together with the email. Since the pair cannot decrypt the sharing , different salt is required to and encrypted confirmation and share are stored in the pair. F_{LI} is extended with a list of selected pairs of $peer_i$, and based on the salts $esalt_i$ and $ksalt_i$, ate retrieved by encryption keys (password change is allowed). To retrieve the password , the user searches an available login file including a list of requested pairs. If the request is authorized , the pair sends encrypted share to e-mail. As soon as the user collected K replies , K_{LI} can be retrieved. When K_LI varies , $K_R^{enc}$is updated to allow further updates.
Combination of two retrieval methods
To ensure the security of the system , any of the above two mechanisms can be combined in continuous or parallel ways. Through a continuous (sequential) combination , the user must correctly answer both security questions , and also receive the e-mail. Through parallel combination , each of the two

mechanisms can be used alone to retrieve password. For sequential combination , the user selects a uniform random sequence r with the same length as the key K_{LI}. User saves r in one of the mechanisms , and r □ K_{LI} in the second mechanism , where □ represents the OR operator. If one retrieves both of them , K_{LI} can be calculated , if identifies only one of the sections , nothing will be retrieved. It should be noted that both r and r $\oplus K_{LI}$ are uniform and random. In general , for combinatorics of n mechanism in an arbitrary method , secret sharing of (K and n) can be used. In order to maintain the confidentiality of information , a random number is required. If the secret key is revealed , then encrypted keys are exposed to revelation risk. Key agreement can prevent outsider attacks , repetition , parallel sessions , reflection , and man-in-the-middle attacks.

Security

Some security risks are related to inherent functionality of the password, and occur despite the technique implementations. For example, in the retrieval by e-mail, an intruder can find the victim's email account and retrieve his/her password. In this section, the relevant security protocols proposals are carefully reviewed. If the proposed protocol interrupt re-login certificate cancellation, many devices may be canceled. Some of these operations have a transaction implication. Some features provided in centralized systems include cancelling the certificate stored in the device and password retrieval based on email without revealing email address before retrieving actual events. In a password-based login, the user may provide some data before and after the reaction, enter your user name, but before he enters the password. Especially when a username was identified, F_{LI} file name can be retrieved from DHT, and the file can be read. Therefore file decryption and further process is only possible when the users enter their passwords. In addition, when an intruder can destroy a number of nodes, he/she will be able to access all information known to that node. In case of an active intruder, even next actions of the node can be controlled. An intruder can also make requests to the system; for example, reading distributed storage files. This function may be worse for small systems, because there are fewer nodes there that mean that the probability of finding a data on a node becomes higher. Considering that the proposed protocols are created by application of several standard components, vulnerability in those components can affect the system. In the proposed mechanism for email-based retrieval, secret sharing (k, n) is used, and passwords are stored on n random nodes in the system. If an intruder controls k or more nodes, he/she can retrieve the password and access the victim's account. Thus, when the nodes are matched, pairs can keep both $ksalt_i$ and email address retrieval to support an attack. It should be implied that e-mail address will be disclosed only to a single pair during the retrieval method. The opposition pairs may soon try to guess it, but they need to check both of $ksalt_i$ and $esalt_i$ guesses. These two parameters are stored in the user's login file, which can only reveal the known username. For example, to create a fake e-mail, so that if clicking on user's password, the effort to decryption and eavesdropping begins. Thus, the method should be anonymous. Anonymity has a special place in electronic payment systems (especially electronic money), i.e. the identity of the buyer shall be kept confidential. And attacking them will impose serious consequences. However, the new authentication protocol can protect the system against these attacks as well as impersonation attack both at client and server sides. Moreover, the intruder can go a step further and install a malicious operating system on a chip in the system to threaten the memory bus, and launch the basis of a physical attack.

Offline guessing

In password authentication based on a peer to peer system , an intruder cannot be prevented from launching an online attack against the encrypted passwords "significant problem". The database of encrypted password should be kept securely to cause failure of parallel attacks such as trial and error and Brute Force attacks against multiple users. Here, a system is recommended by introducing a gradual KDF per each user salt, such as crypt to avoid offline guessing. Also, this protocol can be modified to storage reduction by using the username as a salt. Server problem, on offline attacks against the database password is reported by Ford and Kaliski. Their technique is based on a server-client, and all need online service to login to the system. It should be reminded that guessing attacks can be prevented by limiting the number of failed logins to the system. It should be considered that passwords and messages are completely independent that can somewhat protect the system against offline attacks. Another feature of this scheme is that it protects the insider attack (with key agreement) and prevents the issuance of traditional password management. In fact, passwords are sent over the network in form of original texts "Protecting user anonymity."

Updating the functional keys

When a password is changed, or a tool is canceled, effective access to next updates of key storage will be canceled. However, a malicious device may store be the last keys that are able to access. Thus, in the method of updating the password (which can be used to cancel the device), the keys update their needs to P2P application, and then these new keys need to insert the key storage. However, with the gradual searches on password guess, intruders may feign key extraction method. It should be noted that disclosing the server secret key will be followed by risk of encrypting session keys.

Denial-of-Service (DoS)

For denial of service of a device on the network, an intruder may launch DoS attack by writing lots of usernames in the DHT, and block the attack by registering authorized users. The intruder can also send an application login messages continuously to keep the server occupied, that will lead to DoS attacks and cause server overload. This attack may also be programmed to eliminate fraudulent user accounts in centralized systems. In general, DoS attacks are aimed at breaking a server or network device, so that the system must be restarted or stay out service for a while. This attack is launched whether within the network with high filling internal bandwidth of network with high volume of unnecessarily information, or outside the network, by sending useless traffic from inside to outside of the network for wasting bandwidth of the server. Safe book, similarly, suggests that these attacks only specify a new user account available (By checking the user ID before creating a user account). Other related attack can occur when an intruder can prevent a user from access to the system ((For example, by controlling all copies of (user's logins in DHT). In such attacks, the users can prevent likelihood of login to the system by deleting the key (with selecting a large repetition factor). The IDS (intrusion detection system) and the IDP systems can also be used in network to prevent intruders to the system.

CONCLUSION

Based on the functional analysis and efficient comparisons, we conclude that our proposal scheme requires very little computational and administrative cost, and always promise high security and stability to all known attacks. This system can also be used as an intelligent system with a high degree of security, anonymity, and operational stability for foreign exchange and daily transactions of remote users.

REFERENCES

1. Kreitz, G., Bodriagov, O. , Greschbach, B. , Cano, G. R. , Buchegger, S. "Passwords in Peer-to-Peer," IEEE Communications Society subject matter experts for publication in the IEEE P2P 2012 proceeding
2. Herley, C., van Oorschot, P. C. "A research agenda acknowledging the persistence of passwords," IEEE Security & Privacy, vol. 10, no. 1,pp. 28–36, 2012.
3. Cutillo, L. A. , Molva, R , Strufe, T"Safebook: A privacy-preserving online social network leveraging on real-life trust," IEEE Communications Magazine, vol. 47, no. 12, pp. 94–101, 2009.
4. Abbas ,S. M. A , Pouwelse, J. A. , Epema, D. H. J. , Sips, H. J "A gossip-based distributed social networking system," in WETICE,S. Reddy, Ed. IEEE Computer Society, 2009, pp. 93–98.
5. Aberer, K , Datta ,A , M. Hauswirth, "Efficient, self-contained handling of identity in peer-to-peer systems," IEEE Transactions on Knowledge and Data Engineering, vol. 16, no. 7, pp. 858–869, 2004.
6. Vu L.-H. , Aberer, K , Buchegger, S , Datta, A "Enabling secure secret sharing in distributed online social networks," in ACSAC. IEEE Computer Society, 2009, pp. 419–428.
7. Frykholm, N. , Juels, A "Error-tolerant password recovery," in CCS. ACM, 2001, pp. 1–9.
8. Wehrle, K , ¨otz S. G , Rieche S , "Distributed hash tables," in Peer-to-Peer Systems and Applications, vol. 3485. Springer, 2005, pp. 79–93.
9. Jim´enez, R , Osmani, F , Knutsson, B "Sub-second lookups on a large-scale Kademlia-based overlay," in P2P. IEEE Computer Society,2011, pp. 82–91.
10. Jelasity, M , Voulgaris, S , Guerraoui, R , Kermarrec, A.-M. , van Steen, M. "Gossip-based peer sampling," ACM Computing Surveys, vol. 25,no. 3, 2007.
11. Bortnikov, E , Gurevich, M , Keidar, I , Kliot, G , Shraer,A "Brahms:Byzantine resilient random membership sampling," Computer Networks,vol. 53, no. 13, pp. 2340–2359, 2009.
12. International Electrotechnical Commission, "ISO/IEC 27002:2005. Information technology – Security techniques – Code of practice for information security management," International Organization for Standardization,2005.
13. Ford, W , Kaliski Jr, B "Server-assisted generation of a strong secret from a assword," in WET ICE. IEEE Computer Society, 2000, pp.176–180.
14. Scoville, G "Good security questions," http://goodsecurityquestions.com/ (19th April, 2012).
15. Urdaneta, G , Pierre, G , van Steen,M "A survey of DHT security techniques," ACM Computing Surveys, vol. 43, no. 2, p. 8, 2011.
16. Provos, N , Mazi`eres,D "A future-adaptable password scheme," in USENIX Annual Technical Conference, FREENIX Track. USENIX,1999, pp. 81–91.
17. Rzadca, K , Datta ,A , Buchegger, S "Replica placement in P2Pstorage: Complexity and game theoretic analyses," in ICDCS. IEEE Computer Society, 2010, pp. 599–609.
18. Tolia, N , Andersen, D. G , Satyanarayanan,M "Quantifying interactive user experience on thin clients," IEEE Computer Society, vol. 39,no. 3, pp. 46–52, 2006.
19. He1, D. , Zhao2, W. , Wu3, Sh. "Security Analysis of a Dynamic ID-based Authentication Scheme for Multi-server Environment Using Smart Cards," International Journal of Network Security, Vol.15, No.1, PP.288-294, Jan. 2013
20. Ramasamy, R, Muniyandi, A. P "An Efficient Password Authentication Scheme for Smart Card," International Journal of Network Security, Vol.14, No.3, PP. 180-186, May 2012

21. Smith, R-E. “Authentication from Passwords to Public Keys” Copyright 2002 by Addison-Wesley
22. Panek, W. , Wentworth , T. , Chellis , J. “Network Infrastructure Configuration” Windows Server 2008.
23. Sadowsky , G. ,Depsey , J-X. , Greeberg , A. “IT Security Handbook” infodev , Wordbank 2003.