Design and Implementation of Multi-tier Authentication Scheme in Cloud

Maninder Singh[1] and Sarbjeet Singh[2]

[1]Computer Science and Engineering, UIET
Panjab University, Chandigarh-160014, India

[2]Computer Science and Engineering, UIET
Panjab University, Chandigarh-160014, India

Abstract

The purpose of this paper is to present the design of a secure and more advanced authentication scheme for executing secure financial transactions over Internet. There has been continuous change in technology day by day, so security mechanisms like authentication schemes are also required to be updated. The security measures have very crucial role in banking and financial sector. For any internet application which deals with personal and private information exchange, single tier authentication is inadequate. Authentication schemes that imply more than one tier for authentication are comparatively safer than single tier authentication scheme. Properly designed secure authentication mechanisms are more fraud deterrent. Various multi-tier authentication schemes have been proposed and implemented in various computing domains. The main drawback of these schemes is that most of them do not provide security against insider attacks. In case of Cloud Computing, another drawback is that the whole authentication control lies toward the server side. It is very hard to trust the third party server in Cloud Computing. This work proposes a scheme in which authentication process is carried out in two levels or two tiers. First tier uses simple username and password. Second tier is pre-determined series of steps. The advantage of this scheme is that it does not require any additional hardware and software. So this can be used and accessed from anywhere across the globe.

Keywords: *Authentication schemes, insider attacks, multi-tier authentication, Cloud Computing.*

1. Introduction

From the past few decades, there has been very rapid advancement in computing technology. Systems have been designed which have high resource handling capability, capacity and computing power. For the last decade both hardware and software advancement had been the main goal for the researchers. Through the advancement of internet technology, many works are done online. This includes chatting, entertainment, information gathering and financial transactions etc. All these online activity require some type of authentication. Authentication means to check the identity of the user, which means whether the person is same which he pretends to be. In case of financial transactions, security of information is required to carry out secure transaction. Information in case of online financial transaction includes individual's authentication parameters and some other account related information etc. For authentication, various techniques are used, e.g. username-passwords, biometric face recognition, public key infrastructure and symmetric key based authentication schemes etc. Authentication schemes are key techniques to verify the correctness of the identities of all communication entities [1]. Authentication is quite challenging and difficult in the case of Cloud Computing. In Cloud Computing, a third party is responsible for providing computational power, storage space and application support etc. Every data which is used by a user is stored in Cloud database. Cloud database is maintained by third party Cloud provider, so user hesitates to keep his data at Cloud database. In order to utilize the resources of Cloud, user has to prove with some identity stating that it is valid person seeking permission to use their resources. If a user needs to use or control a remote server or process financial transactions, the user needs to pass the authentication phase first [2].

This paper presents the design and implementation of a multi-tier authentication scheme in Cloud. Section II presents literature review. Section III discusses the limitations of existing approaches. Proposed authentication scheme is described in Section IV. Results have been discussed in Section V and finally Sections VI and VII conclude the paper and talks about future plans.

2. Literature Review

In most applications, authentication is achieved through username and password only. With password cracking tools available free online, hackers take few minutes to identify the user's password [3]. To safeguard users against this threat, NIST (National Institute for Standards and Technology) and FFIEC (Federal Financial Institutions Examination Council) gives details of instructions to carry out financial transactions. [4], [5] and [6] states that 2 tier authentication or 2 factor authentication (2FA)

mechanisms should be adopted. One tier login password is not enough. [4] and [5] Specify various authentication and authorization models. The application must use more than one tier for authentication. That's why users are asked to enter secret code which is sent to their mobile [5].

There are certain guidelines in [6] which focused on risk management controls necessary to authenticate the identity of retail and commercial customers accessing Internet-based financial services. Since 2001, there have been significant legal and technological changes in computing. Protection of customer information is taken as higher priority in the guidelines [6]. These guidelines focus on increasing incidents of fraud and identity theft. These guidelines also give measures to improve authentication technologies. [6] and [7] states that the Financial Institutions (FI) should periodically ensure the following information:

- FI should identify risk mitigation actions including appropriate authentication strength.
- FI should adjust, as appropriate, their information security program in light of any relevant changes in technology. FI should safeguard its customer information, and internal or external threats to information.

There are a variety of technologies and methodologies financial institutions can use to authenticate customers. These methods include the use of customer passwords, personal identification numbers (PINs), digital certificates using a public key infrastructure (PKI), physical devices such as smart cards, one-time passwords (OTPs), USB plug-ins or other types of "tokens", transaction profile scripts, biometric identification, and others [6]. [6] and [7] states various multifactor authentications which include:

i. Shared Secrets: Shared secrets (something a person knows) are information elements that are known or shared by both the customer and the trusted third party.

ii. Tokens: Tokens are physical devices (something the person has) and are a part of multi-tier authentication scheme. For example, use of mobile internet device to gain access to internet connection.

iii. Biometrics: Biometric technologies identify or authenticate the identity of a living person on the basis of a physiological or physical characteristic (something a person is).

iv. Non-Hardware-Based One-Time-Password (OTP) Scratch Card: In this technique, user is given a scratch card. The scratch card acts as one time password. User is asked to fill particular numbers present at particular place in the scratch card.

v. Out Of Band (OOB) Authentication: In this authentication scheme, the user is authenticated in two tiers. Firstly, username and password is entered. Secondly, the user is asked to enter the code which is received on mobile phone.

vi. Internet Protocol Address (IPA) Location and Geo-Location: This scheme checks for physical presence of user by identifying his geographical position. For Example if user has made his transactions in one country then his next transactions would be assumed to be carried out in that country only.

According to the technique presented in [8], the user will be required to authenticate once to SSO server. Once the user is authenticated, SSO will take care of further authentication required by any other application. SSO (Single Sign-On) is a process of authenticating once and gain access of multiple resources. Advantage of SSO technique is to reduce number of logins done by a user for various different applications. Disadvantage of SSO is that if SSO server is hacked then entire Cloud application is hacked.

The technique presented by [9] authenticates the Cloud access in multiple levels. It generates the password and concatenates the generated password at multiple levels. At each level the user has to input password to gain access. Advantage of this technique is that it uses multi-tier approach. It is quite difficult to break multilevel security as compared to single level. Disadvantage of this technique is that it uses passwords at every level but password remembrance is very hectic task for users. If user forgets the password then all the passwords at each level need to be reconsidered.

[10] Focuses on problem associated with keeping private data on third party Cloud. It states that the problem is made worse by the fact that the client has to rely on the claims of security capabilities of the Cloud service provider. [10] Proposes that the user should keep one private security Cloud. This private security Cloud will check the compatibility of data with private Cloud's policy. If data is compatible with policies than access will be given otherwise the data packets are rejected. Advantage of this technique is that user is able to add its own security policies inside private Cloud. Disadvantage of this methodology is that overhead is increased, maintenance is hectic task and initial cost of building private Cloud is increased.

In the architecture presented in [11], application logic is maximized towards client side via RIA. On the client-side, RIA implements UI presentation, web services, application logic and transaction logic. On the server-side, the functionality is simplified and minimized into only storing and querying data via Amazon's simple DB Cloud [11]. Advantage of this mechanism is that it gives architecture for keeping maximum control of data from server side to client side and then storing the data at Cloud side in encrypted form.

[12] Proposes a framework which provides identity management, mutual authentication, session key establishment between the users and the Cloud server. The

proposed scheme verifies user authenticity using two-step verification, which is based on password, smartcard and out of band (i.e. strong two factors) authentication [12]. The advantage of this scheme is that it gives motivation to keep certain authentication control towards client side to resist from attacks. Disadvantage of this technique is that additional hardware and software are required to carry out the processes, which make it little hectic.

[13], [14] and [15] presents other ways of authentication by using either bio-metric or by using some other physical characteristics. The advantages of these techniques are that these use multi-tier authentication. In [16], a framework for handling security issues is discussed which makes use of WS-* security specifications for handling authentication and related issues. [17-20] discuss models for handling privacy, trust and policy based access but authentication is not handled in a multi-tiered way. So the proposed multi-tiered authentication scheme can be adopted in [17-20]. Other multi-tier authentication techniques are discussed in [21], [22] and [23]. Disadvantages of these techniques are that they require additional hardware and software equipment.

3. Limitations of Existing Techniques

Various approaches have been discussed in literature review. As discussed in Section II, there are certain advantages and disadvantages of reviewed techniques. The disadvantages of various techniques can be categorize into four parameters namely security from insider attack, presence of authentication control towards server or client, extra hardware and software needed and number of security tiers required. Figure 1 shows the comparison of various techniques based upon above parameters.

Scheme\parameter	Security from insider attack	Presence of authentication control towards	Extra hardware software needed	No. of security tiers
Authentication using single sign-on [8]	NO	SERVER	NO	1
Multilevel authentication technique [9]	YES	SERVER	NO	MORE THAN 2
2-tier [11]	NO	CLIENT\SERVER	NO	2
Strong user authentication [12]	YES	SERVER	NO	2
PAS [14]	NO	SERVER	YES	2
AQP [13]	NO	SERVER	YES	2
Architecture based on proactive model [10]	NO	CLIENT\SERVER	YES	1
SOA [15]	NO	SERVER	NO	2
Multitier authentication	YES	CLIENT \SERVER	NO	2

Fig. 1 Comparison of various authentication techniques

A brief summary of the comparison is presented below:

i. Security from insider attack: This parameter is based on the assumption that it is easy for an insider to gain access to first tier authentication credentials. This is not tolerable. So second tier authentication is also required.

ii. Presence of authentication Control towards server or client: In Cloud Computing, the data is present on Cloud side and under the supervision of third party Cloud service provider. So there should be some mechanism to authenticate user at client side [15].

iii. Extra hardware software needed: Some of the techniques require extra hardware and software [10], [13], [14]. This adds overhead to the performance of the techniques. The working of such kind of authentication techniques depends upon the working of additional hardware i.e. if the additional hardware fails, then the authentication technique fails.

iv. Number of security tiers: [6] and [7] states that multi-tier authentication schemes are more secure than single-tier schemes. Single-tier schemes are prone to insider attack. So it is desirable to have more than one tier for authentication.

v. Security under pressurized circumstances: This parameter focused on providing security even when the authentication credentials are disclosed to some other person.

4. Proposed Scheme

The proposed authentication scheme is divided into two tiers. First tier authentication uses the encryption-decryption mechanism as followed in normal authentication schemes. The second tier authentication requires the user to perform a sequence of predetermined activities on the fake screen. This fake screen is loaded by the Cloud server in order to capture second tier authentication details from the user. The sequence of activities which the user performs on fake screen must be same which he has chosen during registration. If the details entered by the user are correct then the original screen of application is loaded, otherwise, the user is left over the fake screen. The overall working is explained in 9 steps as shown in Figure 2. These steps are marked in sequential order of their execution in the scheme.

Step wise working of proposed scheme is explained below:

Step 1: User enters URL of application in his browser. Login GUI is loaded in the browser.

Step 2: User enters his first tier credentials (username and password). These credentials are passed to the Cloud server for validation as shown in Figure 2.

Step 3: Cloud checks for first tier credentials. If the username and password are correct then Cloud sends

validation reply to observer (this is application program) at client side through step 3.

Step 4: Upon receiving validation reply from Cloud server, observer initiate the code to load fake screen. The data from fake database is taken during this step.

Step 5: Once the fake data from database is fetched, the fake screen is loaded in the browser.

Step 6: After step 5, observer continuously sense the fake screen in the browser to check for second tier authentication credentials. These second tier credentials are some sequence of registered activities. Observer checks for these activities continuously.

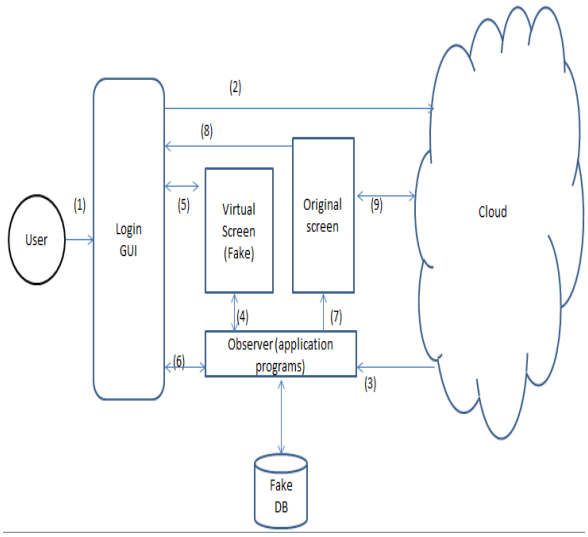

Fig. 2 Working of Proposed Authentication Scheme

Step 7: If the user performs correct sequence of activities then the program for original screen is initiated.

Step 8: Upon successful completion of step 7, original screen is loaded in the browser.

Step 9: The direct communication between client and Cloud server is established in this step.

The difference in this approach is that the details which are entered by the user for second tier authentication are independent of additional hardware and software. These details are certain activities registered by the user, which he will perform after login first tier authentication. These second tier authentication activities in this proposed work could be any among the following three.

i. Menu Activity: This activity registers the sequences of menu clicks which the user will follow after login credentials registration. The user follows his/her registered sequences of clicks on the menu items on the fake screen. If the sequence of mouse event is correct then the user will be authenticated. If the user fails to follow the registered step on the fake screen then he is not shown the original screen of the application.

ii. Mouse Activity: This activity registers the mouse events on the fake screen. Mouse events could comprise of number of mouse clicks decided by the user at particular coordinates. The mouse event also includes dragging of mouse from one coordinate to another coordinate. These mouse events are monitored and if the events go according to the registered sequence by the user, then second tier authentication is validated.

iii. Text Field Activity: In this activity, the user needs to register any phrase which he can memorize easily. Then the initial letters of each word in the phrase are taken and stored as second tier password for that particular user. Now, when the user is presented fake screen, a text field filled with English words is shown. The user can validate second tier authentication by double clicking on the words in text field which starts with the same alphabets of the phrase which he has registered during registration. For example, if the user registered his phrase as "I Love My Country", then "ILMC" letters are extracted from the phrase and these act as second tier password key for user. Now during second tier authentication the user only need to double click on the words which start with "I" then double clicking on word that start with "L" and so on till the last word of his phrase. In this way when the combination of words clicked matches with the combination registered, then the user is authenticated.

5. Results

The multi-tier authentication technique proposed in section 4 has been implemented on cloud using Google App Engine platform and Eclipse IDE (Integrated Development Environment). Various necessary details required for developing the cloud application using GAE is discussed in [24-27]. Development of application on cloud server has been made using Eclipse IDE by installing GAE plugin package [28], [29] and [30]. The communication between user and cloud server is established using RPC (Remote Procedure Call) [31], [32] and [33]. The analysis of various parameters of results has been checked on Google's Dashboard, on the basis of which following analysis has been made.

Security Analysis

In proposed scheme, the data is stored at Cloud side in hashed form, so it is safer from insider attack. This technique can be extended to any levels, but here, it has two levels. Let the two outcomes of the registered events as success and failure be S and F respectively. Then outcome of two levels is {SS, SF, FF, FS} and n(S) = 4, in this case. If probability of success at each level is 'p', then breaking multilevel authentication for success SS, denoted by P(E), is equal to $p^{2.}$ Now failure in breaking multilevel

authentication is 1-P(E) =1-p^2. If probability at each level for success is p= 0.1 (say) then probability of breaking multilevel authentication is 0.01 ($p^{2)}$. For first factor authentication, if encryption key is of length 128 bits then there are 2^{128} different combination for a particular key.

In the proposed scheme, the strength of second tier authentication is as follows:

- **Menu Login:** if there are n menus, then there are n! different combination for menus. Only one chance is given to authenticate user at a given time, so probability of success is 1/n!
- **Mouse Login:** the user can choose any number of mouse clicks as he wants on particular places to authenticate himself. So it is very hard to determine that sequence of clicks. And again only one chance is given, so it also provides a very high level of security.
- **Text Phrase Login:** this is based on the number of words that a phrase has. If phrase have k words, then there are k! different combination. Probability of success is 1/k! at second level.

As only one chance is given to the user, the second factor security is proportional to 1/ (n! or k!). If the variable n or k is increased by a small factor, the probability of breaking the authentication technique will decrease accordingly. For example, take value of n=5, now n! = 120. The probability of breaking second factor is 1/n! i.e. 1/120 = 0.0083. Now if the value of n becomes 6, then n! = 720, the probability of breaking second factor becomes 1/720 =0.0019. For n = 10, the value is 1/3628800 = 0.00000028. From Figure 3, it is clear that the probability of breaking second factor authentication follows exponential pattern.

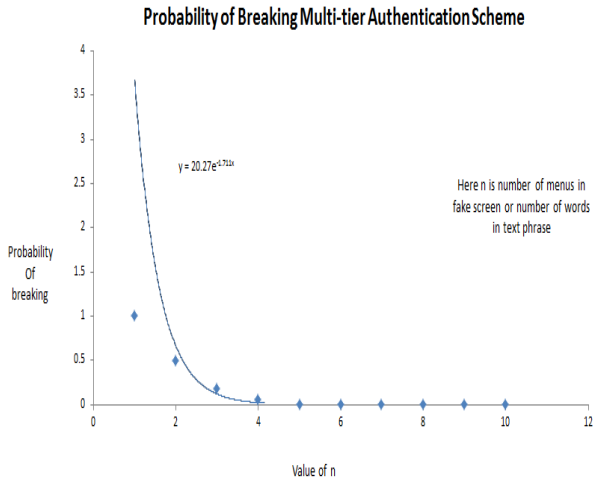

Fig. 3 Probability of Failure

The probability approaches to zero as the value of n or k increases.

Space Requirements

The space is another parameter which is analyzed for the proposed scheme. To evaluate storage space requirement effectively, another technique is designed which works with single factor authentication. The results were taken for both techniques and this has been analyzed that both single factor authentication technique and multitier authentication technique respectively consume space which is linear function of data entered as shown in Figure 4. The space required by second factor authentication and single factor authentication is shown in the graph of Figure 4.

It is clear from the graph that as the data entries are increased in multitier authentication scheme, the space consumed also increases linearly. On an average, multitier technique uses 253 bytes to store one user's credential and single tier requires 114 bytes to store one user's credential. If at any time 100000 users are registered, then server would need 100000*(253-114) = 13900000 (13.3 MB approx.) bytes of additional storage, which is not a big issue in comparison to more security provided by multitier technique.

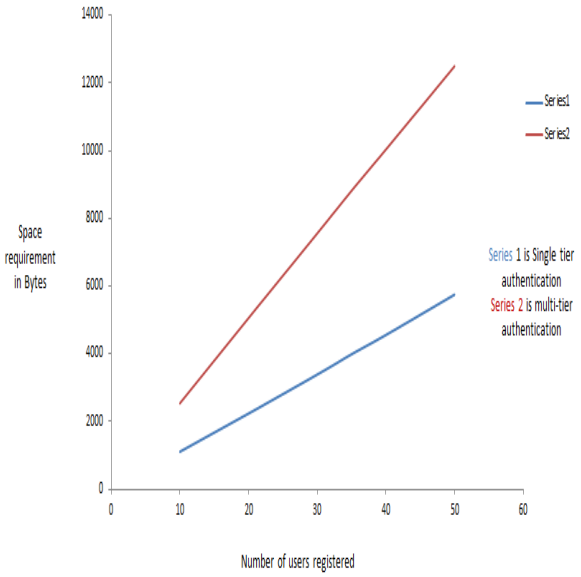

Fig. 4 Space Requirements

Performance Analysis

This parameter is very important to evaluate how effective is this proposed multitier authentication technique in terms of CPU processing time. To evaluate performance, single

factor and multitier authentication techniques are compared. The testing conditions were fixed for both the techniques and equal number to successful login attempts were made to each technique. The values obtained were number of login attempts made versus CPU seconds used per second as shown in Figure 5.

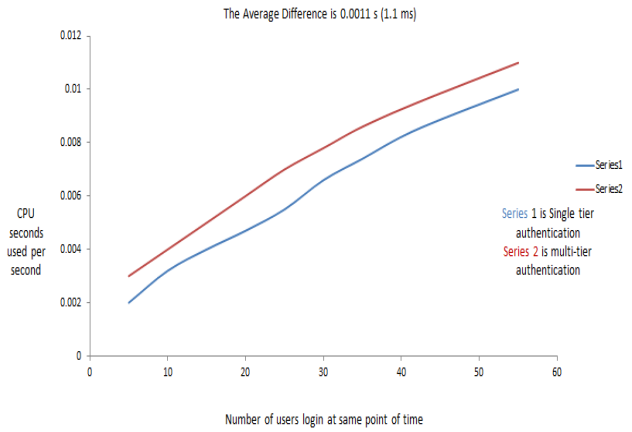

Fig. 5 Performance Analysis

On the basis of this the graph in Figure 5 is plotted which shows that multitier uses only 0.0011s more than single tier. In terms of Cloud Computing, which provides us unlimited, scalable resource, this difference is negligible. Hence it can be said that the proposed multitier authentication technique takes slight higher processing time than single tier technique.

6. Conclusion

The strength of any authentication technique depends upon the probability of breaking that technique. As shown in the Figure 3, the probability is inversely proportional to the various sequential combinations of user activities. Figure 3 shows that as the number of menu items or words in phrases (denoted by n and k respectively) increases, the probability of breaking the multitier authentication technique approaches zero. Hence, it can be concluded that according to security analysis, there is very less probability of breaking second factor authentication. In case of storage space required, multitier authentication technique takes space which is linear function of data entries, so it does not cause much fetching and processing overhead to server. It also provides better mechanism to handle pressurized circumstances by adding concept of fake screen to the system. It is independent of additional hardware and software requirements. In terms of performance, multitier authentication technique no doubt takes slightly higher CPU time, but this slight difference is in milliseconds, which is negligible and can be overlooked in case of Cloud Computing which provides high processing power, storage capacity and scalability.

7. Future Scope

The emphasis of this work is on the design and implementation of multi-tier authentication scheme, which is free from any hardware or software requirements. In the proposed authentication scheme, the work is still needed to be done in situations where the user wants to change his username and password for first tier and for second tier. In other techniques like [21] and [22] the new password is sent to the email registered by the user. But this password generation technique is not appropriate to our proposed authentication scheme. As email itself provides only one tier of security, if new password is sent to registered email of the user, then overall security again will become equal to single tier. So in multitier authentication schemes, the new password should also be recovered in multi-tier manner i.e. half of the password should be sent through one way and the other half should be sent by any other mean. The other multi-tier ways of recovering the password are in the future scope of this work.

References

[1] Chun-I Fan, Pei-HsiuHo, and Ruei-Hau Hsu, "Provably Secure Nested One-Time Secret Mechanisms for Fast Mutual Authentication and Key Exchange in Mobile Communications", IEEE/ACM Transactions on Networking, Vol. 18, No. 3, JUNE 2010.

[2] Wen-Shenq, Juang, Sian-Teng Chen, and Horng-TwuLiaw, "Robust and Efficient Password-Authenticated Key Agreement Using Smart Cards", IEEE, Transaction on Industrial Electronics, Vol. 55, No. 6, June 2008.

[3] White paper for authentication and authorization, "http://www.cryptocard.com/images/stories/pdfs/Authentication_WP.PDF".

[4] Prof. More V.N, "Authentication and Authorization Models", International Journal of Computer Science and Security (IJCSS), Volume (5): Issue (1): 2011.

[5] David Chou, "Strong User Authentication on the Web", Microsoft Corporation, August 2008 Available at: http://msdn.microsoft.com/en-us/library/cc838351.aspx

[6] "Authentication in an Internet Banking Environment", Federal Financial Institutions Examination Council, Government of USA, 2005.

[7] William E. Burr et al., "Electronic Authentication Guideline by U.S. Department of Commerce", NIST Special Publication 800-63, Version 1.0.2, April 2006.

[8] Ashish G. Revar and Madhuri D. Bhavsar, "Securing User Authentication Using Single Sign On in Cloud Computing", Institute of Technology, Nirma University, IEEE, December 2011.

[9] Dinesha et al.,"Multi-level Authentication Technique for Accessing Cloud Services", International Conference on Computing, Communication and Applications (ICCCA), IEEE, 22-24 February 2012, pp 1-4.

[10] Prashant et al., "An Architecture Based on Proactive model for Security in Cloud", International Conference on Recent Trends in IT, IEEE, 3-5 June 2011, pp 661-666.

[11] Wenjun Zhang, "2-Tier Cloud Architecture with Maximized RIA", Research Institute of Applied Computer Technology, IEEE, Vol. 6, 2010, pp 52-56.
[12] Amlan et al. , "A Strong User Authentication Framework for Cloud Computing", Asia- Pacific Services Computing Conference, IEEE Computer Society, 2011, pp 110-115.
[13] Adrian Kapczynski and Marcinsobota, "Distributed Authentication Systems Enhanced by Quantum Protocols", Fifth International Conference on Information Technology: New Generations, IEEE, 2008, pp 928- 931.
[14] Mohammed RazaKanjee, KalyaniDivi, and Hong Liu,"A Physiological Authentication Scheme in Secure Healthcare Sensor Networks", Proceedings of IEEE Secon, 2010.
[15] Fengyu Zhao, XinPeng, Wenyun Zhao, "Multi-Tier Security Feature Modeling for Service-Oriented Application Integration", Eigth IEEE/ACIS International Conference on Computer and Information Science, IEEE, 2009, Page 1178-83.
[16] S. Singh and S. Bawa, "Design of a Framework for Handling Security Issues in Grids", in International Conference on Information Technology, 2006, ICIT'06, 18-21 Dec. 2006, pp. 178-179.
[17] Sarbjeet Singh and Seema Bawa, "A Privacy Policy Framework for Grid and Web Services", Information Technology Journal 6, 2007, pp. 809-817.
[18] Seema, Sarbjeet Singh and Dolly Sharma, "An Access Control Framework for Grid Environment", Indian Journal of Computer Science and Engineering", Vol. 2, No. 6, Dec 2011 – Jan 2012, pp. 937-948.
[19] S. Singh, "Trust Based Authorization Framework for Grid Services", Journal of Emerging Trends in Computing and Information Sciences, Vol. 2, No. 3, March 2011, pp. 136-144.
[20] S. Singh and S. Bawa, "A Privacy, Trust and Policy based Authorization Framework for Services in Distributed Environments", International Journal of Computer Science, Vol. 2, No. 1, 2007, pp. 85-92.
[21] Charles Miller, "Password Recovery", available at http://fishbowl.pastiche.org/archives/docs/PasswordRecovery.pdf
[22] Google Account Recovery, methods available at https://accounts.google.com/RecoverAccount
[23] Peter Mell and Timothy Grance, "Recommendations of the National Institute of Standards and Technology", NIST Special Publication 800 145, Computer Security Division, Information Technology, September 2011.
[24] Daniel Guermeur and Amy Unruh, "Google App Engine Java and GWT Application Development", Packt Publication, Chapter 1, November 2010.
[25] Fay Chang et al., "Bigtable: A Distributed Storage System for Structured Data", Google Incorporation, Available at: http://research.google.com/archive/bigtable.html
[26] "Google Web Toolkit Get Started", Available at: https://developers.google.com/webtoolkit/doc/latest/FAQ_GettingStarted
[27] Daniel Guermeur and Amy Unruh, "Google App Engine Java and GWT Application Development", Packt Publication, Chapter 2, November 2010, pp 24-29.
[28] "Google Plugin for Eclipse 3.7 Installation Instructions", Available at: https://developers.google.com/eclipse/docs/install-eclipse-3.7
[29] "Google Web Toolkit: Organize Projects", Available at: https://developers.google.com/webtoolkit/doc/latest/DevGuideOrganizing Projects
[30] "Create a GWT Project", Available at: https://developers.google.com/web-toolkit/doc/latest/tutorial/create
[31] "Communicating with Server", Available at: https://developers.google.com/web-toolkit/doc/latest/tutorial/clientserver
[32] "Making Remote Procedure Calls", Available at: https://developers.google.com/web-toolkit/doc/latest/tutorial/RPC
[33] "Benefits of Using RPC over HTTP", Available at:http://technet.microsoft.com/en-us/library/aa997284(v=exchg.65).aspx

Maninder Singh received his B.Tech degree in Computer Science & Engineering from Punjabi University, Patiala, Punjab, India in 2009. He received M.E degree in Computer Science and Engineering from Panjab University, Chandigarh, India in 2012. His research interests include Distributed Systems, Cloud Computing and security issues in Cloud Systems.

Sarbjeet Singh received his B.Tech degree in Computer Science & Engineering from Punjab Technical University, Jalandhar, Punjab, India in 2001 and M.E. degree is Software Engineering from Thapar University, Patiala, India in 2003. He also received Ph.D degree in Computer Science & Engineering from Thapar University, Patiala, India in 2009, working on grid security systems architecture.
Currently he is working as Reader in Computer Science & Engineering at UIET, Panjab University, Chandigarh, India. He has more than 20 research publications in International conferences and journals to his credit. His research interests include distributed systems, distributed security architectures, Cloud Computing, privacy and trust related issues in distributed environments.
Dr. Singh is a life member of Computer Society of India and Indian Society for Technical Education.