Current Trends in Technology and Science
ISSN : 2279-0535. Volume : 6, Issue : 2 (Feb – Mar 2016)

Securing Human Computer Interaction

Triparna Mukherjee
Student of M.Sc. Computer Science, St. Xavier's College(Autonomous). Kolkata, India, ria911.tm@gmail.com

Asoke Nath
Associate Professor, St. Xavier's College(Autonomous),Kolkata, India, asokejoy1@gmail.com

***Abstract* — The weakest link to implementing security protocols in human computer interaction happens to be the human memory , and to aid the process of remembering security information often the most important security principles are often violated. This paper puts forward some of the guidelines that need to be followed while designing a secure human-computer interaction system. Password being the most common security tool has been studied with a survey relating to people's reaction to more sophisticated and/or secure forms of it. Solutions have been provided by considering individual differences. Several factors which influence the security decisions of a user has been discussed as these form the loopholes in which most naïve users compromise security objectives.**

***Keyword* — human-computer interaction, passwords, segregation of views, tailoring views, privacy indicators.**

1. INTRODUCTION

In general, Human-computer interaction is "a discipline concerned with the design, evaluation and implementation of interactive computing systems for human use and with the study of major phenomena surrounding them". [1] HCI (human-computer interaction) devices mainly deal with how people interact with computers and to what extent computers are capable of performing successful interaction with humans. As a field of research, Human-Computer Interaction is situated at the intersection of several disciplines such as computer science, behavioral sciences, design, media studies, and several other fields of studies. The human computer interface is defined as the point of communication in such cognitive transactions between the computer and human beings. The communication flow is known as loop of communication. There are several aspects which aid the success of the cognitive ability of such interfaces. One of the principal aspects among these is security in human-computer interactive devices. However unlike devices where there is no need of human like behavior or interference implementing security protocols is relatively easier. The extent to which a human-computer interaction system is expected to be user-friendly may be curtailed when security measures have to be taken. [2] As most HCI systems require internet a great number of security threats emerge. If users do not know how to use the interface, their systems will still be vulnerable. [3] It is observed that many users, and especially those that are not educated enough with computers are not able to modify the applications they use and simply use the default settings. They may not aware of the fact that security options for the application exist, or how to modify them according to their requirements. The reason for this is that the software applications that provide security options for such sophisticated cognitive devices have been designed by technical people having a technical audience in mind. As a result, their complexity is high. As mentioned by Johnston et al. (2003)[4] , Human Computer Interaction security is defined as: "the part of a user interface which is responsible for establishing the common ground between a user and the security features of a system. HCI-S is human computer interaction applied in the area of security".

2. IMPLEMENTATION OF SECURITY TECHNIQUES IN HUMAN-COMPUTER INTERACTION

There are several guidelines that need to be followed while implementing human-computer interaction security.

1. Visible system state and security functions: Applications should not suppose that all users will investigate the application settings in order to find the security tools or have hidden features inside the application with different views. Furthermore the use of status mechanisms can keep users aware and informed about the state of the system. Status information should be periodically updated automatically and should be easily accessible. It is very likely that too much abstraction can make the user averse to using the application.
2. Security should be easily used: The interface should be carefully designed and require minimal effort in order to make use of security features. Additionally the security settings should not be placed in several different locations inside the application, because it will be hard for the user to locate each one of them.
3. Suitable for advanced as well as first time users. Show enough information for a first time user while not too much information for an experienced user. Provide shortcuts or other ways to enable advanced users to control the software more easily and quickly. It is likely that naïve users will find it hard to use the security features in their application if technical vocabulary and advanced terms are used.

5. Fault Resistant Security: The application should be planned carefully so that errors caused by the use of security features could be prevented and minimized as much as possible. However when errors occur, the messages have to be meaningful and responsive to the problem.
6. Allow customization according to changing needs without risk to be trapped: Exit paths should be provided in case some functions are chosen by mistake and the default values should be easily restored. This way the user will feel more confident with changing and configuring the application according to suit their changing needs. [5]
7. Security should not reduce performance and render the entire objective of the device as futile.

However unlike traditional systems it is quite complicated to deploy security techniques in human computer interaction systems. The security research community – which hitherto largely ignored the human factor – now acknowledges that "Security is only as good as its weakest link, and people are the weakest link in the chain." [6] The general idea to solve this issue is to compartmentalize whenever possible and have specialized duties. If there's no good reason for someone to have access to a system or information the simple solution is to not provide such a view to them. The first implication of this new perspective on security is that the traditional security approach – to address the problem by developing more, and more complex, technology - is not sufficient. A Human-Computer Interaction (HCI) design approach takes into account that users and technology work together completing a task (in order to achieve a goal) in a physical and social context.

There are several methods of authentication among which it is said that five are most popular. [7] However most security mechanisms use a two-step procedure in which identification and authentication are combined. An example of such combination are cash cards (token-based identification) combined with a PIN (knowledge-based authentication). By far the most common access control mechanism in computing is the combination of a user-id (identification) and password (authentication).

2.1. Issues With The Most Common Security Tool-Password

Even though passwords are the most popular links to human computer interaction In our view, this is a repeat of the "human error" mindset that wrecked the development of safety-critical systems until the late eighties [8]. Most password systems are implemented in the same way: The system issues a userid for every new user, and also a password (which can be changed by the user to one of his/her choice). The password is supposed to be a secret shared between the user and the computer only; that is supposed to be a secret between the computer and the person accessing the system. To log on to the system, the user has to enter his user-id , which is a token of identification and password which is a token of authentication. The system processes the entry and compares this to the entries it has stored previously. If it finds a match, the user will be given access to the computer system. If there is no match, the user will not be allowed access; she may have to contact a system administrator to have a new password issued so that he can have access. Many systems suspend an account after 3 or 5 unsuccessful login attempts, and bar further attempts until the account has been re-set. The reason for the popularity of using passwords from both a technical as well as naïve point of view because of its simplicity.

Four major factors influencing effective password usage were identified within the framework:

• Multiple passwords;
• Password content;
• Perceived compatibility with work practices; and
• Users' perceptions of organizational security and information sensitivity.

But there are also a number of usability issues connected to its use. Password mechanisms are usually implemented on a per-system basis. This means that users need to log into each password protected system individually; the time required to log into a number of systems several times a day can add up. Some operating systems store user-id and password and automatically and use it on the user's behalf (for example to mount remote volumes). Another user having access to the device do not have to try to break the security. According to FIPS [8] there are several rules for implementing password protection however all of this needs the user to remember his user-id and password at all times. So most of the users end up using something that is simple to remember and hence simple to decipher. Some systems even allow writing down the passwords which completely loosens the security objectives. This violates the first principle of knowledge-based authentication – that the password should exist only in two places – in the system (in encrypted form) and the users' mind. A technical solution to reduce the number of passwords is a single sign on (SSO) login system, which many companies are starting to deploy. This reduces not only users' memory load, but also the total number of time users spend on logins. If SSO is not feasible (e.g. because of cost), allocating users a single user-id for all systems, standardizing passwords rules, and enforcing them consistently can improve the situation somewhat. Most security policies decree that users should have different passwords for different systems, to limit the number of systems compromised if an unauthorized person gets hold of a password. Ultimately, this makes for more effective security because it gives users a chance to have strong passwords they can remember, and a strong password reduces the chances of it being compromised in the first place.

Current Trends in Technology and Science
ISSN : 2279-0535. Volume : 6, Issue : 2 (Feb – Mar 2016)

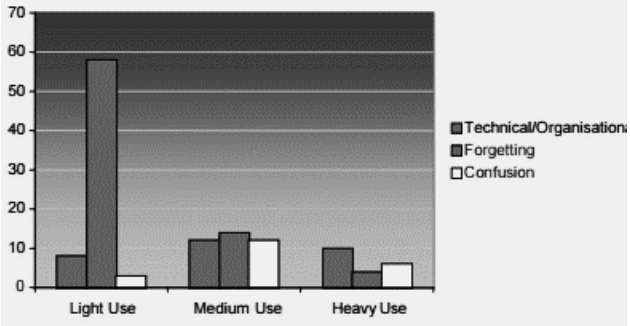

Fig. 1. Frequency and cause of problems with passwords [9]

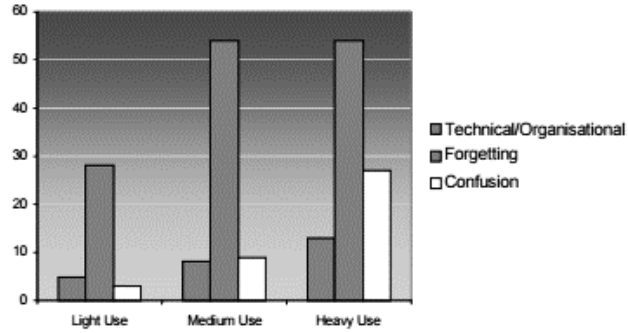

Fig. 2. Frequency and causes of problems with 6-digit PIN [9]

Security as well as usability experts [6] have stated that recalling strong passwords is a humanly impossible task because strong passwords are non-meaningful items and hence inherently difficult to remember. However, what makes a password easy to guess or crack is the fact that it is meaningful to many other people, as well as the password owner. So the key is to choose passwords that are only meaningful to the user .It is possible to create passwords that are strong and meaningful: pseudo-random combinations of letters, numbers and characters that are meaningless to anyone but the password owner/creator. Allan [10] calculates a "breaking point"; there is a maximum effective entropy - a pessimistic calculation puts this at about 18bits - for all types of password; exceeding this is likely to cause users to write passwords down, so that trying to increase password entropy by strengthening the policy will be counter-productive.

2.2 Solution To The Issues Mentioned Above

There is an ongoing debate about how to solve the issue of password policies. A solution to this issue has been addressed by scientists by the help of pass algorithms which are user based validation schemes based on knowledge based algorithms [11]. Many systems administrators use system-generated passwords for maximum security, and generate a sentence that describes something of the user's input. For example, after a user has successfully logged in via the vendor's standard log-in procedure, he or she suddenly finds a randomly generated prompt of BEL on the terminal. Since the system manager has told the user that this month's secret log-in algorithm requires providing the next alphabetic character for each character of the prompt, the user responds with the password CFM and completes a successful log-in. Furthermore, because the password changes from one log-in attempt to the next, an intruder cannot use the rejection of a password to imply anything about the next password to be tried. According to [11] "the prompt can be generated in a number of ways. It may be the day, the time, the temperature, the system real-time clock, the amount of disk space available, the company's current stock price, or the port number of the current terminal. To be particularly effective, the prompt might consist of all of these and more".

It is understandable that providing this kind of pass-algorithm protection requires that the system manager ensure that

(1) All log-in attempts, especially those to privileged accounts, execute the system log-in command file;

(2) The user cannot at any time abort the execution of the system login-command file

(3) Pass-algorithm-protection routines provided by the site, security officer, and users are invoked by the system log-in command file.

Beautement et al. [12] proposed the model of the Compliance Budget to understand how users balance the effort of complying with a security behavior required by an organization, against their own benefits in the context of their production goals. This offers a positive way forward, since the organization can manage users' compliance budget through good security design and a security-aware organizational culture. However as majority of human-computer interaction users are supposed to be naïve and mainly bothered by the simplicity and efficiency of the system expecting people to be aware about the need of unbreakable security might be too much. It is suggested by some that unusable passwords might be the key to solving this ongoing issue in human computer interaction. However the cost of passwords that are truly not re-usable isn't feasible in most scenarios. [13]

3. Some Approaches To Handle HCI – Security Issues

HCI has considerable experience with dealing with individual differences. In one approach suitable to privacy mechanisms, it has been found valuable to cluster users, and then to present different interfaces or functionality to those users depending on the type of content they access. Another approach is to allow users to tailor the systems to their own needs; however, this often requires that they obtain tailoring help from others.

3.1 Segregation of Users

- Constructing user friendly interfaces and segregating views - "This approach is similar to standard human-interface design, except that it is shaped by a concern for the variability among users. " This is particularly important for systems where people are not expert

users and where they will remain "permanent casual users."[14] In cases like this it is very important to create appropriate abstraction. Redesigning interfaces and systems so as to reduce usability as well as security issues is a praiseworthy goal. Yet, because of the complexity of privacy concerns for users, it is unlikely that a "one size fits all" approach will work adequately. The associated prospect of constructing some software that has all potential privacy functionality for a task (like the solution adopted by some word processors and office applications) may not work with privacy concerns or may be too complex for users, since the functionality is likely to cut across many tasks, systems, and applications. The possibility of task-based variability is plausible for informational websites, where thousands of information items compete for promotion to the most reachable positions. [15]

- Clustering users and creating adaptive systems - This approach promotes accommodating user differences by finding a set of user clusters and then interacting with the users through those classifications based on varying interests. This can be done in several different ways. One could also have different dialog or interaction patterns with different user classes. More currently, one might treat these differing clusters of users differently. Indeed, work on several problems shows the analytical power in examining user clusters. One set of papers examines default settings. Most users not only do not program their systems, they do not even customize them or change the default settings.[16][17] But it is also likely that different groups having different interests or skills might have different technology and working models. Clustering is a priority in such cases. Similarly, privacy mechanisms will be used very differently not only by people with differing assumptions about power and control, the efficacy of regulation and law, and the benign intent of companies. Finding suitable user clusters will be important, but may be challenging, especially for members of the pragmatic majority.[18] For protection, while the pragmatics are a substantial and very separated group in their ordinary, contextualized inclinations, it might be very conceivable to treat privacy fundamentalists and the marginally concerned population as client bunches. Thus, it might be conceivable to make usable security systems for at least each of these groups.[19]

Adaptive systems are also very helpful. These systems prevent user errors by helping users.

3.2 Tailoring Views

- This approach is to have users tailor or customize the systems to fit their needs. By needs we also mean privacy needs. Customizing usually refers to changing the surface interfaces of a system; tailoring usually refers to deeper changes to the functionality of an application. In this approach, the designer includes large amounts of functionality, most of which any given user will not use. Unlike robust interfaces, which present "one size fits all" interfaces, tailor able systems allow users to pick and choose their functionality. The system is made with a varying range of functionalities most of which is not even required by the users. The users can choose and customize the content according to their use and can also decide on privacy concerns.

However, the thought of planning for individual contrasts likewise has a drawback that is critical to remember: the potential for amplifying power irregularities and diminishing fairness of judgment. By characterizing somebody as a protection fundamentalist, say, a framework could choose he is a lot of inconvenience and set up obstructions to debilitate use. On the other hand, deceitful planners could portion clients keeping in mind the end goal to search out beginners or the possibly concerned, not to offer focused on help, but rather for moderately simple misuse.

4. FACTORS INFLUENCING SECURITY DECISIONS

While working to improve the security measures being used in a particular human computer interaction scenario it is very important to understand what do web users perceive to be security decisions and having recognized a security decision is required, on what do users base their security decisions. There are several factors that help us to evaluate this.

1. Prior use as a security indictor: A very common theme was that participants based their security decision on having used the website in question previously. It is very likely that if a person has visited a site a number of times he is well aware of the unsafe popups and what to access and what are the security warnings that need to consider seriously .Similarly not having used the particular interface might result in opening of unsafe popups or providing information in not so private platforms. Knowledge from previous use even allowed users to overlook specific security warnings. For example, while logging in to a university application portal usually google chrome prompts it as unsafe but if a person is sure of the content he is approaching he carries on by ignoring the security warnings. Lack of previous use was a reason for caution, and in some cases avoidance. Some users who have simply entered a site by browsing would not proceed to the website because they do not know about this website and found it using Google Search.
2. Checking for security indicators and certificates: When choosing if a site was secure, the most widely recognized technique was to watch that the web address began with "HTTPS" and that there was a lock present. Members noticed "The page address contains the HTTPS and the lock sign". Further, members expressed "not HTTPS" for sites without this quality. Lack of knowledge about these security indicators can result in issues. The padlock often gives them the impression that the site they are connecting to is the

real-world person or company that the site claims to be (in reality, it usually just means that the connection is encrypted to "somebody"). Even more generally, many people think that the padlock means that they are "safe" to do whatever they wish on the site without risk. Finally, there are some tricky hacker moves that can make it appear that a padlock is present when it actually is not. HTTPS is a pretty popular security indicator but there are several other security indicators such as validation certificates, color coding depending on browsers. Most important of such indicators are Domain Verification Certificates. Domain Validated SSL Certificates are no-frills, encryption-only certificates that certify the authenticity of a domain that is being accessed. In order to get a Domain Validated SSL Certificate a person just has to prove that he owns the domain by responding to an email or phone call using the information in the WHOIS record of the domain. It's easy. A company doesn't have to be validated and no organization name is entered in the certificate. But the issue with such a certificate is that it does not ensure complete authenticity. The certificates themselves still enable full, 128-bit encryption but there are other security problems. For one, any phisher can get one and can hide their identity completely. Second, they make man-in-the-middle attacks more dangerous. If an attacker was able to do some DNS poisoning, he could get a Domain Validated SSL Certificate for the concerned domain and redirect visitors to a fake site that allows him to collect visitor information. In summary, domain validated certificates do almost nothing to verify that who we are talking to is really the one who we think we are talking to. To solve this problem the concept of EV or Extended Verification came into being. An Extended Validation Certificate (EV) is a public key certificate requiring verification of the requesting entity's identity by a certificate authority. EV certificates use the same encryption as domain validated certificates: the increase in security is due to the identity validation process, which is indicated inside the certificate by the policy identifier. EV certificates actually do verify real-world identities. They also typically cause some prominent part of the browser to turn green and show the real-world entity's name and location (eg: "Bank of America Corporation (US)").

3. Perception of reputation as a security indicator: Perceptions of security, and security choices made, were strongly based on the company's reputation. Some participants stated the only information that they based their decision of whether or not to proceed on, was their trust in the company that they thought they were dealing with. Participants found various websites where they could not tell if the website was secure, even when explicitly looking for security indications. Websites, for example some banks, used techniques that embed webpages inside other webpages (e.g. using iframes). This ensures that even people very knowledgeable about computers and how HTTPS works, could not be sure if their communication was secure. Some participants went to the extent of viewing the source code for the page to see if the embedded form was secure. Others simply felt they had no knowledge or basis to judge as they did not know what could be falsified easily and what could not.
4. Unrecognizable website addresses: Participants find that they have no way of deciding if a website was who they claimed to be, when the website used their IP address (eg a number such as 66.102.11.104 as the web address, instead of www.google.com as the web address). It is agreed by many users that "You have requested an encrypted page that contains some unencrypted information" (very common) had little point. What is secure and what is not secure is in no way defined and hence cannot be used to make a decision. More significantly, some pages were found to be HTTPS pages with forms on them, but the forms targeted http webpages. In many cases, security warnings are unintelligible or are misinterpreted by naive users. [21]

It is very likely that concepts of extended validation certificates have not permeated into the general public. After this, the main factor that participants based their security decision on was the reputation of the company whose website they thought they were browsing. While company reputation is an important factor, security decisions should not be based upon reputation prior to authenticating that company is who they claim to be. Further, when private transactions are being made with the company, the channel of communication should have the confidentiality property (ie HTTPS). Designers need to be aware of the security goals, and design their browsers and websites with these security goals in mind. Designs should highlight, rather than hide, the key information that users need to make informed security decisions. This will involve the designers either acquiring the knowledge of security fundamentals themselves, or liaising with security professionals. Extended validation certificates remain the best way to authenticate to the user that they are dealing with a specific company, and communicating with them securely. Web browsers need to be adapted to more clearly indicate, and educate, users on the significance of the extended validation certificate information.

5. CONCLUSION

There are resolutions to the security problems caused by the behavior of users, but they are not commonly used (see [22] for an excellent review). To alleviate the problem of remembering multiple passwords, for example, organizations can support synchronized passwords across systems. A related solution is a single-sign-on system where users are authenticated once and then they are allowed to access multiple systems. Another technique is to reduce the memory load placed on users. It is well known that cued recall, where users

are prompted for the information they must remember, is more accurate than free recall [23].This can be used in security systems by requiring personal associates for passwords, such as "dear - god, "black - white", "spring - garden". Performance can also be improved by not asking users to recall at all, but rather to recognize certain material. Recognition is much easier and more accurate than recall [8]. There is some evidence, for example, that Passfaces are easier to remember than passwords, especially after long intervals with no use [24]. Insufficient communication with users produces a lack of a user-centered design in security mechanisms. Many of these mechanisms create overheads for users, or require unworkable user behavior. It is therefore hardly surprising to find that many users try to circumvent such mechanisms. Parker [25] points out that a major doctrine in password security, adopted from the military, is the need-to-know principle. The assumption is that the more known about a security mechanism, the easier it is to attack; restricting access to this knowledge therefore increases security. Users are often told as little as possible because security departments see them as "inherently insecure." One clear finding from this study is that inadequate knowledge of password procedures, content, and cracking lies at the root of users' "insecure" behaviors. This attitude has led to a twofold problem:

(a) users' lack of security awareness, and

(b) security departments' lack of knowledge about users, producing security mechanisms and systems that are not usable.

These two factors lower users' motivation to produce secure work practices. This in turn reinforces security departments' belief that users are "inherently insecure" and leads to the introduction of stricter mechanisms, which require more effort from users. Engineers of secure frameworks confront a genuine test. In the event that a security framework is not easy to understand, designers face disappointment in the commercial center, or clients that evade or disregard the security highlights. Despite the fact that it regularly gives the idea that security and convenience are opposite item traits, it need not be that way. For instance, Yee [26] has as of late laid out ten HCI plan rule that can be utilized to enhance the ease of use of security frameworks. Moreover, devices are rising to help designers when checking for security vulnerabilities. Frequently the outcomes and ramifications of the code sweeps can be mind boggling and hard to decipher. The field of HCI can likely add to change of security-improving improvement apparatuses. Another improvement issue is outline logic. Particularly in the domain of Web applications and administrations, outline commonly continues from the base up, driven by entrenched Web application plan designs and the limitations forced by hidden innovations, for example, open key cryptography. Be that as it may, it is frequently hard to retrofit these outline designs with adequate security structures. An option approach starts with a user centered investigation of work process and data stream (with accentuation on the limits), trailed by an outline approach that is driven from the top, taking consideration to utilize entrenched security models to implement access control and information division where proper .Designers of security mechanisms must realize that they are the key to successful security system. Unless security departments understand how the mechanisms they design are used in practice, there will remain the danger that mechanisms that look secure on paper will fail in practice.

ACKNOWLEDGMENT

The authors are very much grateful to Department of Computer Science for giving opportunity to do research work in e-learning methodologies. One of the authors Asoke Nath is also grateful to Dr. Fr. John Felix Raj for giving all inspiration and support to research work in Computer Science and Engineering.

REFERENCE

[1] Hewett et al. 1996 ACM SIGCHI Curricula for Human-Computer Interaction

[2] IT Security: A Human Computer Interaction Perspective D.Katsabas, S.M.Furnell and A.D.Phippen Network Research Group, University of Plymouth, Plymouth, United Kingdom

[3] Using Human Computer Interaction principles to promote usable security D. Katsabas, S.M. Furnell and P.S. Dowland Network Research Group, University of Plymouth, Plymouth, United Kingdom

[4] Three Faces of Human–Computer Interaction Jonathan Grudin

[5] 10 Usability Heuristics for User Interface Design by Jakob Nielsen https://www.nngroup.com/articles/ten-usability-heuristics/

[6] Schneier, B. (2000), "Secrets and Lies", John Wiley & Sons, 2000.

[7] Rejman-Greene, M. (2001): Biometrics – real identities for a personal world. BT Technical Journal Vol 19 (3), July 2001.

[8] FIPS (1985) "Password Usage". Federal Information Processing Standards Publication. May 30.

[9] Transforming the "Weakest Link": A Human-Computer Interaction Approach for Usable and Effective Security Martina Angela Sasse, Sacha Brostoff & Dirk Weirich Department of Computer Science, University College London http://discovery.ucl.ac.uk/144215/1/BTTJSECv5.pdf

[10] PASS-ALGORITHMS: A USER VALIDATION SCHEME BASED ON KNOWLEDGE OF SECRET ALGORITHMS JAMES A. HASKETT http://www.facweb.iitkgp.ernet.in/~shamik/spring2005/i&ss/papers/Pass-algorithms%20a%20user%20validation%20scheme%20based%20on%20knowledge%20of%20secret%20algorithms.pdf

[11] Allan, A. Passwords Are Near the Breaking Point: Gartner Research Note (2004) http://www.indevis.de/dokumente/gartner_passwords_breakpoint.pdf

[12] Beautement, A., Sasse, M. A., and Wonham, M. The Compliance Budget: Managing Security Behaviour in Organisations, ACM Press (2008)
[13] The True Cost of Unusable Password Policies: Password Use in the Wild Philip Inglesant & M. Angela Sasse Department of Computer Science University College London Gower Street, London WC1E 6BT, UK {p.inglesant, a.sasse}@cs.ucl.ac.uk
[14] Egan, D.E. and Gomez, L.M. Assaying, isolating and accommodating individual differences in learning a complex skill. In R. Dillon, ed., Individual differences in cognition. Vol.2. Academic Press, New York, 1985.
[15] Are users more diverse than designs? Testing and extending a 25 years old claim. Martin Schmettow University of Twente Enschede, The Netherlands m.schmettow@utwente.nl Jop Havinga University of Twente Enschede, The Netherlands jop.havinga@gmail.com
[16] Mackay, Wendy E., Thomas W. Malone, Kevin Crowston, Ramana Rao, David Rosenblitt, and Stuart K. Card. 1989. How Do Experienced Information Lens Users Use Rules? In (eds), Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems, 211-216.
[17] Mackay, Wendy E. 1990. Patterns of Sharing Customizable Software. In (eds), Proceedings of ACM CSCW'90 Conference on Computer-Supported Cooperative Work, 209-221.
[18] Orlikowski, Wanda J. 1992. Learning from Notes: Organizational Issues in Groupware Implementation. Proceedings of the Computer Supported Cooperative Work (CSCW'92) : 362-369.
[19] Grudin, Jonathan. 2004. Managerial Use and Emerging Norms: Effects of Activity Patterns on Software Design and Deployment. Proceedings of the Hawaii International Conference on System Sciences
[20] Hummes, Jakob, and Bernard Merialdo. 2000. Design of Extensible Component-Based Groupware. Computer Supported Cooperative Work Journal, 9 (1) : 53-74.
[21] How HCI Design Influences Web Security Decisions Kenneth Radke Colin Boyd Margot Brereton Juan Gonzalez Nieto Queensland University of Technology Brisbane, QLD 4001 k.radke, c.boyd, m.brereton and j.gonzaleznieto@qut.edu.au
[22] Adams, A., & Sasse, M.A. (1999). Users are not the enemy: Why users compromise computer security mechanisms and how to take remedial measures. Communications of the ACM, 42, 41–46.
[23] Crowder, R.G. (1976). Principles of learning and memory. Hillsdale, NJ: Lawrence Elrbaum Associates.
[24] Brostoff, S., & Sasse, M.A. (2000). Are Passfaces more usable than passwords? A field trial investigation. In Proceedings of HCI 2000, Sept. 5–8, Sunderland, U.K., 405-424 Springer
[25] Parker, D.B. Restating the foundation of information security. In G.C. Gable and W.J. Caelli, Eds., IT Security: The Need for International Cooperation. Elsevier Science Publishers, Holland, 1992.
[26] Yee, K.-P. (2002). User Interaction Design for Secure Systems. http://zesty.ca/sid/uidss-may-28.pdf

AUTHOR'S PROFILE

Triparna Mukherjee is a student of M.Sc,. Computer Science, Department of Computer Science, St. Xavier's College (Autonomous), Kolkata. Currently she is doing research work in cognitive radio technology , human-computer interaction and decision science. Her area of interest includes cognitive science and digital ergonomics.

Dr. Asoke Nath is Associate Professor in Department of Computer Science, St. Xavier's College (Autonomous) , Kolkata, India. Apart from his teaching assignment he is involved in various research activities. His major research areas comprises of Cryptography and Network Security, Steganography, Green Computing, Big data analytic, e-learning, distance learning, MOOCs, Cognitive Radio, Mathematical Modeling in Social Networks. He has already published more than 175 publications in International, National Journal and Conference Proceedings. He is life member of MIR Labs(USA), CSI Kolkata Chapter.