EFFICIENT METHOD FOR SECURELY MANAGING PASSWORDS

[1]ASMA TANVEER, [2]AIHAB KHAN, [3] MALIK SIKANDER HAYAT KHIYAL, [4]SYED AFAQ HUSSAIN

[1]Student of Department of Software Engineering Fatima Jinnah Women University, Rawalpindi.
[2]Assistanct Prof,Department of Computer Science Fatima Jinnah Women University, Rawalpindi.
[3]Chairperson,Department of Computer Science and Software Engineering Fatima Jinnah Women University, Rawalpindi.
[4]Prof. Faculty of science and Engineering, Riphah International University, Rawalpindi.

Abstract: *A password is a secret string that is used for security and authentication purposes and to provide access to a resource. Therefore it is more vulnerable to attacks such as hacking, phishing, identity theft, Cyber stalking and website cloning. But it is difficult for humans to remember large no of passwords.This research is formulated to provide a source to users by which they can conveniently and securely manage their passwords. The user is authorized to enter the site information, ID and password in a list which is secured by a Master Password which is encrypted. AES encryption algorithm is also compared with DES encryption algorithms w.r.t time, size of file after encryption and we have concluded that AES algorithm is faster than DES. AES has only known attack that is brute force attack which allows Hacker to try different combination of words and characters to break security. AES has high performance as compared to DES.*

Keywords: *Phishing, Authentication, Password, Master Password Security.*

1. INTRODUCTION

To secure user accounts password is a top secret string of characters or word or numbers which is used for verification, to prove identity and to access resource (example: a known password type is access code). The password must be kept secret from the persons who are not allowed to access your resources. On web the most commonly use type of authentication is password but passwords have many usage problems and security threats. [1]

Passwords are inconvenient but are compulsory to avoid others from accessing our online accounts, files, and computers etc. The Project is to develop an efficient method by which user can securely manage his password. Security of password depends on choosing passwords that are distinctive and not easily guessed, however long passwords may be hard to remember and reuse properly. For security users are asked to create and remember large number of passwords for uses as well as accounts of host, email servers, e-commerce sites etc.This research is conducted to implement the convenience and management of passwords by generating secure short master password for all sites.

Unfortunately, the large number of passwords which users have to memorize seems unsatisfactory to accumulate distinctive, secure passwords for all user accounts, and it is probable to remain constant as the number of passwords increases. The easiest passwords to select and remember may be exposed to dictionary attacks, where an attacker tries to guess the password by using common passwords from the list of words. To change the passwords regularly helps user defend against attack, but the problem is passwords are not easily memorized. Easier password is easy for hacker to guess. Complicated passwords are not easily memorized because users have to write down or store the password, there is possibility that user may re-use the same passwords also; the strict necessities for password strength (e.g. passwords are a mix of uppercase and lowercase letters and digits) has greater degree to which users threaten the system. Therefore password authentication occupy a switch.[2]

Our aim is to develop a method which is convenient for user to remember and secure password using hashing algorithms and Phishing protection techniques.

Passwords are difficult to remember but necessary way to prevent hackers from accessing our online accounts, computers, files, etc therefore many convenient and secure procedures and methods are proposed for strengthening passwords.

A password must meet the following security issues.

Fraud we enter username and password to a site which seems legal can trap our emails.

Like passwords are easy to guess e.g using row of letters from keyboard (e.g. “iop”, “fghj”), names (like pet, relative, user, significant other), dates (marriage, birth), or arrangement of the above.

If our passwords cannot be guessed easily, the answers to password secret questions, like Pet Name, that websites offer in case user forgets his password can be guessed by the person who knows us or looks for personal information about you online.

The hackers who try to guess our password use programs that try each and every word and its combination in the dictionary.

In public places like libraries, net cafes people can see our passwords as we type the password.[3]

Some programs like Viruses and Trojan horses secretly install programs that grasp and correspond what we type on our system.

Convenience issues include human factors. It is difficult for humans to memorize unique passwords for all accounts. A simple study shows that normal human being cannot remember even fifty random combinations of characters that have one of more common solutions: also users select non-random passwords that are easy to remember and break. To remember the password the users write down the passwords on paper or maintain a file of passwords on their system with the major issue of security [4].

Password differentiates between the authorized and unauthorized users. It is used for the security purpose that provides control to only a restricted group of users who know the password. The security measures are always taken so that it is not easily accessible by phishers or hackers.[5]

A typical computer user needs the passwords in many respects e.g. logging into online accounts, recovering emails from servers, accessing files and databases, network websites. But it is not possible for humans to remember large passwords for hundreds of accounts. [6]

Phishing uses emails and websites which are designed to look like emails and websites from legitimate organizations to deceive users into disclosing their personal or financial information. The hostile party can use this information for criminal purposes such as theft, fraud.

Some assumptions have been made to limit the scope of the work. These assumptions are:

Any information flowing across the network is secure.

Secure method for users to manage many passwords against phishing attacks using encryption algorithms.

Method could be vulnerable to dictionary attacks.

Due to growing usage of internet services security threats are also increasing. For accounts we need passwords which are difficult to manage and it is also not convenient for human to memorize a lot of different passwords. So the technique of generating short master password on a list of all user account passwords provides security and convenience to users this short password will be easily memorized and user can access all accounts. This password will be free from dictionary attacks. Limitations are always there so my project has also some limitations that I am not concerned about other security issues.

The selection of topic convenient method for securely managing password is that it is security and convenience is main problem these days a lot of techniques have been developed for securely managing passwords using hash functions, encryption, decryption etc but generating strengthened master password for all sites is most convenient computationally low and cost effective technique.

Our research statement is about developing an Efficient method for securely managing passwords.

- To ensure the security and convenience of passwords
- To ensure that the password is easily memorable by humans

- To create a short master password on a list of all the passwords of accounts that is secure and can be easily memorized.
- To ensure that the master password is applicable on all user accounts.
- To ensure that the password cannot be decrypted.

2. RELATED WORK

Halderman et.al.[1] proposed a technique in which strengthened cryptographic hash function is used to compute secure passwords for arbitrarily many accounts and user needs to memorize short password. This method works completely on the client side; server-side changes are not needed. Unlike earlier approaches, their design is extremely opposed to brute force attacks and approximately stateless, allow users to pick up their passwords from any location so as long as they can achieve their program and remember a short secret password. Master passwords in a hash-based password invention system may be exposed to brute force attacks due to low-entropy of passwords memorable by humans and the ability of a challenger to perform a low-cost offline password guessing attack. There has been much effort put into interchange password entry method that increases unforgettable entropy, common character string passwords are still central method. Our approach focuses on to raise the cost of a brute force attack to make guessing time-exhaustive. Hash-based system is basically exposed to an offline guessing attack of master password by a competitor who has get hold of one or more of the user's site detailed passwords. This competitor can apply a simple brute force attack by identify the potential master passwords and for each one performorming the hash algorithm for that specific site and comparing it to site-specific password. The amount of time to execute this attack will be the number of passwords tested multiplied by the execution time for the hashing algorithm. If the algorithm is planned quickly, like SHA-1, this attack can be relatively useful.

Khiyal et.al.[4] analyze a technique of password hashing is to compute safe passwords. By using this we can get hash value by implementing a cryptographic hash function to a string of the passwords and the other value is salt. Salt value consists of current constraints of the system and prevents attackers to build the list of hash values for known common passwords. Commonly used cryptographic hash functions are MD5 and SHA1. We applied these algorithms and have found that SHA-1 is more secure as MD5 but slow in execution includes more rounds than MD5 in calculating hashes.

Ross et.al.[7] described the browser extension, PwdHash, which visibly create a diverse password for each site, improving web password security and protecting against password phishing and other different attacks. It means that the user just need to type same password for each site, but various sites will receive different encoded password. PwdHash take all user input to a password field and sends hash (pwd,dom) to the remote site, dom is resultant from the domain name of the remote site. We refer dom as salt. The hash is applied using a Pseudo Random Function enter by the password. In this paper, they describe the design, user interface, and application of a browser extension, PwdHash, which reinforce web password verification and by providing customized passwords, above SSL, the threat of password attacks will decrease with no server changes and minor or no change to the user practice. They have discussed the main challenges in building PwdHash which are salting; encoding, dictionary attacks, javascript attacks, roaming, auto-complete and password reset, and then describe solutions to these challenges. Finally they implement the prototype for Internet Explorer and Mozilla Firefox.

3. Proposed Model & Technique

The Proposed model is given in fig1 on next page.

The definition of the terms used in proposed model is as follows:

EP: Encryption Function,

PWD: password

User opens the entry form. Store user information in list and information includes user name, password, and site information

Save the list with master password. Apply encryption and save the list. The list will be saved in windows registry.

User can open his list by entering valid password. Apply decryption and it will authenticate the user and give access to list.

User can see his password and he can simply login to his desired password.

In this technique, the Master password is encrypted with AES, DES and Blowfish. Phisher cannot break this.

We provide the user authentication by applying the decryption with the addition of salt on the password provided by the user. The addition of salt prevent the phisher to produce the hash values list of the database entries to get the user common password to access the legitimate site (online banking).

In this technique, we used three encryption algorithms (DES, AES and Blow fish) that strengthen the master password authentication.

In our scheme, the user verifies the original Master password do not have to execute any public key confirmation. So my scheme needs less public key processes.

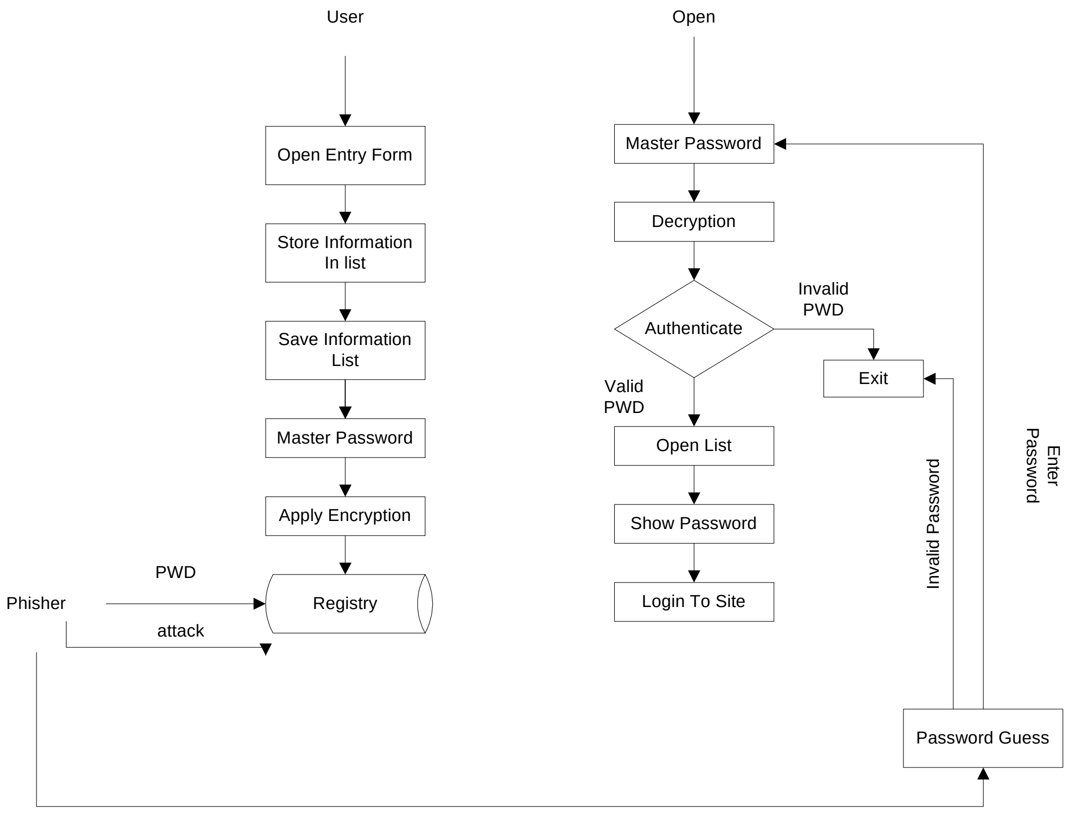


Figure 1. Proposed cryptographic Model of Efficient method for securely managing passwords

Limitations are always there our model has also some limitations.

This technique is implemented in Visual C# and backend database is handled in Windows Registry.
Master Password for list is elaborated in Fig 2.

```
public MasterPasswordForm(bool SaveMode )
        {
            InitializeComponent();
            saveMode = SaveMode;
            if (saveMode)
                this.btnOK.Text = "OK";
            else
            {
                LoadPassword();
                this.btnOK.Text = "Authenticate";
            }
        }
```

Figure 2: Master Password for list

The code of master password which is applied on list of user information.
Encryption algorithm on Master Password is elaborated in Fig 3.

```
private void LoadPassword()
        {
            RegistryKey regKeyAppRoot =
Registry.CurrentUser.CreateSubKey(strPath);
            string stPassword =
Convert.ToString(regKeyAppRoot.GetValue("mp
wd", ""));

            RegistryKey regKeyAppRootMode =
Registry.CurrentUser.CreateSubKey(strEncryptio
nMode);
            string strMode =
Convert.ToString(regKeyAppRoot.GetValue("E
METHOD", ""));
            if(string.Equals(strMode,"DES"))
                password =
DESEncryption.Decrypt(stPassword);
            else if (string.Equals(strMode, "AES"))
                password =
AESEncryption.Decrypt(stPassword);
            else if (string.Equals(strMode, "BF"))
            {
                Byte[] data =
System.Text.ASCIIEncoding.ASCII.GetBytes(stP
assword);
                password = (new
BlowFish(System.Text.ASCIIEncoding.ASCII.G
etBytes(blowFishKey)).Decipher(data,
data.Length));

            }
            regKeyAppRoot.Close();
        }
```

Figure 3: Encryption algorithm on Master Password

We have applies three encryption algorithms on master password AES, DES and Blowfish. All are secure and resists brute force attacks.
Decryption algorithm on Master Password is elaborated in Fig 4.

```

DataObject data = new DataObject();
            CryptoCore decryptor = new
CryptoCore();
            data.SetData(DataFormats.Text, true,
decryptor.DecryptString(Program.CurrentStorage
.Entries[Convert.ToInt32(listView.SelectedItems[
0].Tag)].Password));
            Clipboard.SetDataObject(data, true);
```

Figure 4: Decryption algorithm on Master Password

We have applied CryptoCore decryptor to decrypt the password for opening the information list.
Algorithm of Master Password Authentication is elaborated in Fig 5.

```
{
            InitializeComponent();
            saveMode = SaveMode;
            if (saveMode)
                this.btnOK.Text = "OK";
            else
            {
                LoadPassword();
                this.btnOK.Text = "Authenticate";
            }
        }
```

Figure 5: User Authentication

The algorithm for authentication of user if user will enter valid Master password it will authenticate the user other wise invalid password.
Code for Master Password Storage is elaborated in Fig 6.

```
RegistryKey regKeyAppRoot =
Registry.CurrentUser.CreateSubKey(strPath);
                RegistryKey regKeyAppRootMode =
Registry.CurrentUser.CreateSubKey(strEncryptio
nMode);

                string strToSave = txtPassword.Text;
```

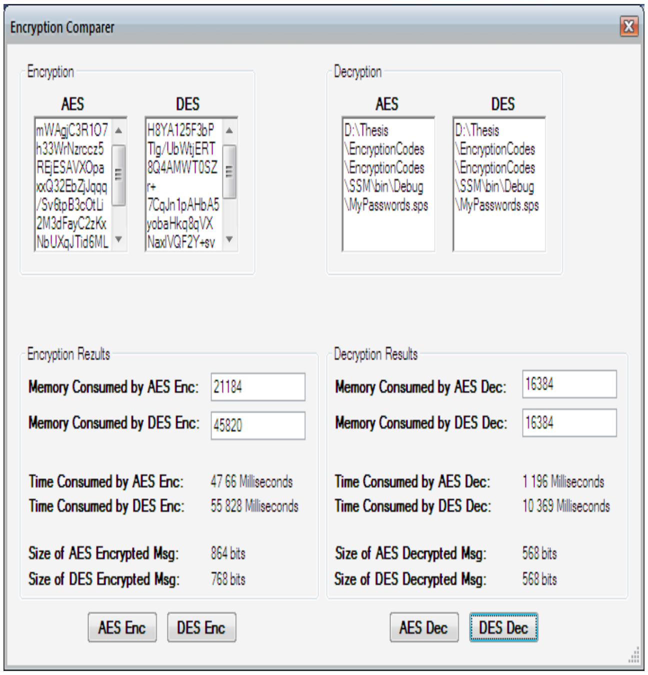
```
          if(rbDES.Checked)
             strToSave =
DESEncryption.Encrypt(txtPassword.Text);
          else if(rbAES.Checked)
             strToSave =
AESEncryption.Encrypt(txtPassword.Text);
          else if (rbBlowFish.Checked)
          {
             Byte[]
data=System.Text.ASCIIEncoding.ASCII.GetByt
es( txtPassword.Text);
            strToSave =(new
BlowFish(System.Text.ASCIIEncoding.ASCII.G
etBytes(blowFishKey)).Encipher(data,
data.Length));
          }
regKeyAppRoot.SetValue("mpwd", strToSave);

          if (rbDES.Checked)

regKeyAppRoot.SetValue("EMETHOD",
"DES");
          else if (rbAES.Checked)

regKeyAppRoot.SetValue("EMETHOD",
"AES");
          else

regKeyAppRoot.SetValue("EMETHOD", "BF");


          regKeyAppRoot.Close();
```

Figure 6: Master Password storage

The Master password is stored in window registry we can encrypt the master password with any one of the encryption algorithm AES, DES or Blowfish.

4.RESULTS

Comparison of different Encryption/Decryption Algorithms. Shown in Figure 7.



Encryption Comparer

Encryption

AES	DES
mWAgjC3R1O7 h33WrNzrccz5 REjESAVXOpa xxQ32EbZjJqqq /Sv6tpB3cOtLi 2M3dFayC2zKx NbUXqJTid6ML	H8YA125F3bP Tlg/UbWtjERT 8Q4AMWT0SZ r+ 7CqJn1pAHbA5 yobaHkq8qVX NaxlVQF2Y+sv

Decryption

AES	DES
D:\Thesis \EncryptionCodes \EncryptionCodes \SSM\bin\Debug \MyPasswords.sps	D:\Thesis \EncryptionCodes \EncryptionCodes \SSM\bin\Debug \MyPasswords.sps

Encryption Rezults

Memory Consumed by AES Enc: 21184
Memory Consumed by DES Enc: 45820

Time Consumed by AES Enc: 47 66 Milliseconds
Time Consumed by DES Enc: 55 828 Milliseconds

Size of AES Encrypted Msg: 864 bits
Size of DES Encrypted Msg: 768 bits

Decryption Results

Memory Consumed by AES Dec: 16384
Memory Consumed by DES Dec: 16384

Time Consumed by AES Dec: 1 196 Milliseconds
Time Consumed by DES Dec: 10 369 Milliseconds

Size of AES Decrypted Msg: 568 bits
Size of DES Decrypted Msg: 568 bits

Figure 7 Comparison of different Encryption& Decryption Algorithms

DES and AES are compared with respect to time consuming in calculating encrypted file, memory consumed for each encryption function, size after calculating Encryption of password and their results are shown in the following table.1 & table.2.

Table 1 Comparison of different Encryption Algorithms

Encryption Algorithm	Size of Encrypted MSG	Consumed Memory	Time Consumed for Encryption
AES	704 bits	53556 bytes	10 msec
DES	608 bits	146972 bytes	55.828 msec

Different graphs obtained from the table values are given in fig8 & fig 9.

Table 2 Comparison of Different Decryption Algorithm

Decryption Algorithm	Size of Decrypted MSG	Consumed Memory	Time Consumed for Decryption
AES	440 bits	16384 bytes	10 msec
DES	440 bits	16384 bytes	10.369 msec

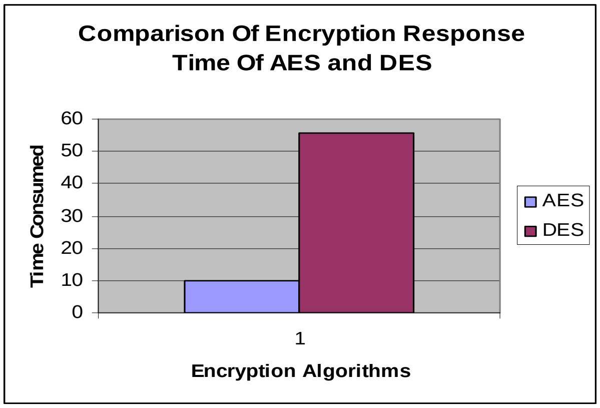


Figure 8 Comparison of Encryption Response Time of AES & DES

From this graph it is analyzed that encryption response time of AES is less than DES

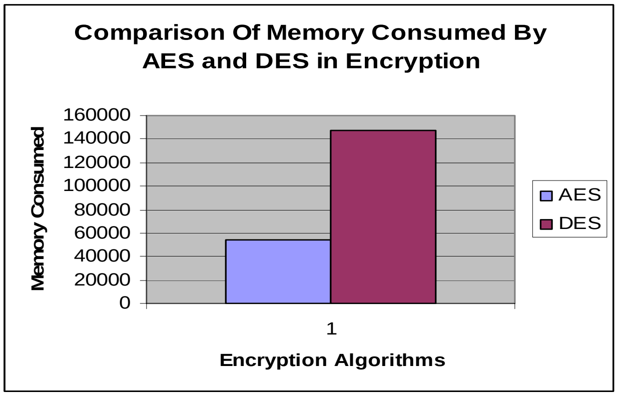


Figure 9 Comparison of Memory consumed in Encryption by AES & DES in Encryption

From this graph it is analyzed that memory consumed in encryption by AES is less than DES

Different graphs obtained from the table values are given in fig 10 & fig 11.

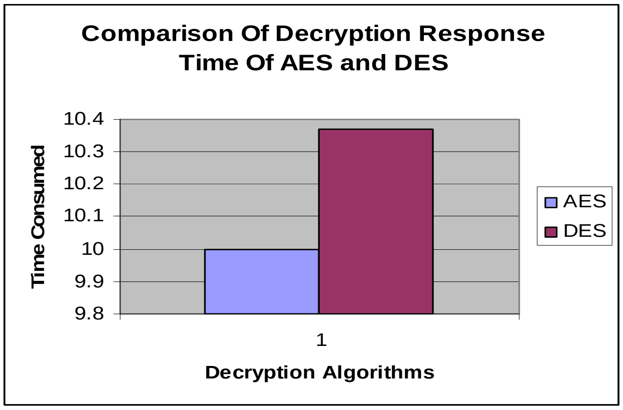


Figure 10 Comparison of Decryption Response time of AE$S & DES

From this graph it is analyzed that decryption response time of AES is less than DES

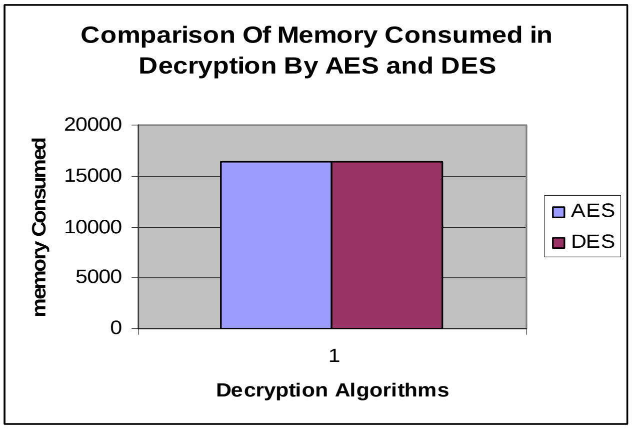


Figure 11 Comparison of Memory Consumed in Decryption by AES & DES

From this graph it is analyzed that memory consumed in decryption by AES and DES is equal.

5. CONCLUSION

A password is a form of secret confirmation data that is used to control access to a resource. Password is kept secret from those who are not allowed access, and those who wish to gain access are tested on whether or not they know the password and are granted or denied access accordingly. We have implemented encryption techniques on master password to secure our managed passwords. For management of user information we have created a list of user information in which user name, ID, password and site information is stored. This list is saved in windows registry with a secure master password. We have compared the encryption algorithms for encryption response time, decryption response time, and memory usage.

We have concluded that AES algorithm is faster than DES. The one known attack to AES is the brute force attack that allows an attacker to try combination of characters in order to break the security, as for performance AES has high performance.

In future we can manage our passwords in Databases for very large number of data. We can also secure the password by authenticating the user by issuing digital certificate.

REFERENCES:

[1] J.A.Halderman,B. Waters, and E.Felten. A convenient method for securely managing passwords. Proceeding of the 14th International World Wide Web Conference (WWW 2005),(2005, IBM-FAIRUCE, 471-479)

[2]] Rachna Dhamija & J.D.Tygar, “The Battle against Phishing Dynamic Security skins.” “University of California”, Berkeley. In Proc.2005 Symposium on Usable Privacy and Security. Vol.93, SOUP 05. New York , ACM press, NY 77-88.

[3] KaPing-Yee and Kragen Sitaker. Passpet: Convenient Password Management and phishing protection. In proceeding of the second Symposium on Usable Privacy and security (SOUP 2006), (32-43 Yue, C and Wang, H.2008) Antiphishing in defense and offense.345-354, In proceedings of the Annual Computer Securety Applications Conference(ACSAC).

[4] M.S.H.Khiyal, A.Khan, N.Bibi and T.Ashraf, Analysis of Password Login Phishing Based Protocols for Security Improvements. In proceeding of IEEE 5th International Conference on Emerging Technologies (ICET 2009), pp: 376-379, October 19-20, 2009. FAST National University of Computer and Emerging Sciences (FAST-NUCES), A.K. Brohi Road, H-11/4, Islamabad.

[5] Engin Kirda and Christopher Kruegel. Protecting Users Against Phishing Attacks.554-561,2006,Computer Journals 49(5). Published by Oxford University Press on behalf of the British Computer Society, 2005

[6] Mercan Topkara Ashish Kamra Mikhail J. Atallah Cristina Nita-Rotaru Visible Watermarking based Defense against Phishing, 15-17 september,2005, International Workshop on Digital Watermarking, Siena, Italy.

[7] B. Ross, C. Jackson, N. Miyake, D. Bonch, J.C. Mitchell. Stronger Password Authentication Using Browser Extension. 14th Usenix Security Symposium, 2005.