Enhanced Password-Based User Authentication Using Smart Phone*

Inkyung Jeun, Mijin Kim, and Dongho Won**

Sungkyunkwan University,
300 Cheoncheon-dong, Jangan-gu, Suwon, Gyeonggi-do, 440-746 Korea
{ikjeun,mjkim,dhwon}@security.re.kr

Abstract. Today, an internet environment has become a single society in which all of the services required for daily living are implemented online. To reliably use such an internet environment for our daily living, safe user identification and authentication are required. Of course, there are many ways to perform authentication online. However, thus far, in many e-services include cloud services, passwords are mainly used for user authentications. Although there are many safer authentication methods than a password, which has the risk of personal information leakage and is a relatively poor authentication method, passwords are frequently used due to their high user convenience and ease of implementation. In this article, to resolve the weaknesses of the current password environment, we suggest a new user authentication method that can offer both safety and convenience combined with the recent mobile internet environment. For this purpose, in this article, a smart phone is used as the storage space for the user password. A user can conveniently use passwords in the wireless e-service as well as the wired e-service.

1 Introduction

Smart phones represented by Steve Jobs in Apple inc. have recently been gaining a global attention. They are intelligent terminals combining the functions of mobile phones with features such as internet communication and information search. The representative aspect that distinguishes smart phones from other mobile phones is that users can install applications ("apps" hereinafter) in it[1]. The smart phones aren't just means of communication, but they provide us functions like a PC. As a result, the mobile cloud envirinment is established in our world. We can see and modify our document on the PC as well as our smart phone using the mobile cloud services. So, the internet is changing from a simple means of information exchange to a cyber society. We call this e-society. The convenience that we can connect to the internet anytime and anywhere also increased the use of Social network service(SNS) like Facebook or Twitter, so

* This research was supported by the KCC(Korea Communications Commission), Korea, under the RnD program supervised by the KCA(Korea Communications Agency)"(KCA-2012-12-912-06-003).

** Corresponding author.

R. Li, J. Cao, and J. Bourgeois (Eds.): GPC 2012, LNCS 7296, pp. 350–360, 2012.

the world has established an e-society.E-society in a cyber space has some good points, as it facilitates communication among users and can serve as a vital source of information in the event of a disaster or emergency, as was recently seen during the Japanese earthquakes. Nevertheless, the more the e-society is becoming a giant and the e-services which are members of e-services have being increased, the more threats are also increasing. Among them, one of the leading threats is the problem of a personal information exposure. When the use of e-services increases, the threats of a personal information exposure also increase. There have been many cases reported related to the distribution of false information using stolen ID, or causing damages to others by misrepresenting celebrities.

SNS Profiles have also become a channel for ID theft by hackers. The user information contained in an SNS Profile (e.g. address, place of birth, school, phone number, date of birth, color of vehicle and name of pet) can help hackers find questions and answers for ID tracking. Hackers can use the information contained in an SNS Profile to answer the ID search question, and abuse the account by identifying ID[2]. This is possible because users use multiple SNSs with a single ID, allowing the ID linkability of multiple SNSs.

In order to avoid these risks, each e-services are using variety user authentication method like a password as well as a digital certificate, one time password (OTP), bio information. Nevertheless, the most e-services still use a password as a user authentication method for reasons of user-friendliness and ease of implementation. But, the problems of password are steadily being pointed. The password can be guessed, forgotten, written down and stolen, eavesdropped or deliberately being told to other people. Most users tend to use a simple password even though they know the importance of his/her password for the inconvenience of memory. Also even if they use a secure password, the password can be exposed to others by key logging program of malicious hackers. In addition, despite the increase of smart phone users, the use of wired and wireless environments for e-services has not been lined. That is, even if the same e-service, user authentications are performed separately in wired and wireless environment, and especially smart phone users tend to use weak password for user authentication, because the input is inconvenient compared to the input of keyboard in PC. And sometimes they use automatic login function in smart phone, which can increase the risk of password exposure.

Against this backdrop, this paper proposed a way to safely and conveniently enjoy e-services by using smart phones. The content of this paper are follows. Chaper 2 featured the problem of password for an user authenticationm and chapter 3 proposed the enhanced password-based user authentication protocol using smart phone. Chapter 4 analyzed our proposed model and chapter 6 provided conclusions.

2 Problems of Password for an User Authentication

Our world is being tied into the huge e-society as the off-line based services such as an e-health service, e-business and e-banking are available through on-line internet. Recently, as many people use the excellent wireless devices such as smart

phones, e-society that we can access to e-services at anytime and anywhere is being built. To use this e-service safety, the trusty worth user authentication method is our priorities. In an e-society environment, the proper authentication method should be provided considering the risk level of each services. The authentication methods are divided into various types such as user holding, user itself and user knows.

The most widely used means of authentication among various authentication methods is clearly a password. This is easier to implement, as well as easy to use, so it is used in the most e-services. However, the vulnerability of passwords has been raised in the past until now. Passwords can be exposed to hackers by key logging programs, as well as hackers can find out the passwords if we use a short and simple password. Thus, the e-services take place some financial transactions such as e-banking, e-payment use multi-factor authentication. For that, the password is used in conjunction with digital certificate, OPT, etc. methods for user authentication

The specific issues if we use a password as an authentication method are as follows.

(1) Difficulty of memory

The more we use e-services, the more the passwords we should remember increase. We should use different passwords in all services due to the possibility of the risk of exposing the password, but it's impossible to set up the different passwords according to the e-services. Secure passwords have a long length including letters, number, as well as special characters, but this also limits to remember. As a result, it's very difficult to configure easy-to-remember passwords for users. So most users use the weak and same password in all e-services, and besides they write their password on the post-it or the notebook. A recent study by the security firm Trustwave indicated that the most common password is Password1, a variation on the historically common default password, Password[3]. As we can see in this study, most users tend to use easy password.

(2) Key logging

Recently, the key logging tools that installed on a PC and can expose an input password via keyboard are exploited. The services that require high security technology such as e-banking service installed and used some anti-hacking software to prevent key logging attack, but it is impossible that all e-services use that software. So, we can say that the risks of exposing a password exist anywhere.

(3) Phishing Attack

Hackers can seizure the input password as they built a fake web-site instead of actual web-site. Phishing attack is to be used in order to seize your financial information to earn the money, but simply it can be used means to steal user's password and some personal information.

(4) Non-Repudiation

Password user can deny their act on the internet. In other words, if a user access the SNS and write some wrong comments, the user can repudiate his acts and

say that it is performed by others due to the password theft. In this case, the service provider is hard to prove it.

Despite these many problems, the password is used widely and easy to use, so it is very necessary to strengthen it's safety. Accordingly, this paper proposed an enhanced user authentication scheme using smart phone considering recent e-society environment. To this, we proposed password-based authentication model, one of the most commonly used means for user authentication and suggested using of smart phone as a storage of password. So we hope it can be classified as a multi-factor authentication, and it can increase security level of password.

3 Enhanced Password-Based User Authentication Scheme Using SmartPhone

In this section, we proposed the enhanced authentication method that will minimize the changes of the password-based authentication mechanism which is the most used in the current e-services. Our proposed method can solve the problems of password such as the difficulties of memory, the risk of personal information exposure due to use of weak password, etc. E-services that need a high level authentication method such as e-banking service use an additional authentication information for user authentication like PKI certificate or bio information.[4] But, these methods are impractical in general e-society services like SNS because they require excessive costs for an additional implementation. Another high level authentication method is OTP. Google web-site uses OTP to who wants to use stronger user authentication.[5] It is two-factor and strong authentication method based on one time password. Instead of authenticating with a simple and weak password, each user carries a OPT generator ("token") to generate passwords that are valid only one time. To generate one time password, the user has to enter his personal PIN into the OPT token. And the one-time password which is generated in the token is used for user authentication on the e-services. So OPT authentication is two factor authentication method using the OTP token and a PIN ("something you have and something you know"). This is obviously more secure than just simple password, because an attacker should know the PIN as well as the token device. This can be quite expensive and it is very inconvenient to the users. To solve this problem, the mobile-OTP was developed using a smart phone instead of a token device.[7] But, Mobile-OTP also needs an additional implementation cost to recognize and process the OTP.

In this paper, we proposed the password-based user authentication mechanism similar with current e-services. The e-service providers don't need any additional implementation, and they can just process the password. And the user can use their own password without difficulty of memory using smart phone and mobile application. Also, the security level of the password would be increased.

3.1 Terms and Abbreviations

– ESP(E-Service Provider) : It is e-service which is configure the e-society like as e-banking, e-payment, cloud service provider, etc.

- Password token : The data element containing the personal information of a user, including password, ID, etc. It will be saved in smart phone safety, and sent to e-service provider for an user authentication if necessary.
- SmartID_App : It is application installed in the smart phone of users, and it saves, manages, and transfers the password token.
- ID(Identity) : It is identifier for user identification in ESP, and it is generated by users.
- salt : It consists of random bits, creating one of the inputs to a one-way hash function with password.
- SS(Shared Secret) : It is shared secret between ESP and user's smart phone, and used as encryption key when they transfer confidence information between them.

3.2 Symbols and Notions

- $T(s, r, a)$: For $s \in \{ESP, USER, SmartID\_App\}$, $r \in \{ESP, USER\}$ and a, it means transfer a from s to r.
- $S(i, a)$: For $i \in \{ESP, USER\}$ and a, it means to save a to i.
- $E(a)$: For $a \in \{SmartID\_App\}$, it means to execute a by user on the smart phone.
- $ENC_{KEY}(p)$: It means to encrypt p using encryption key KEY.
- $DEC_{KEY}(p)$: It mean to decrypt p using encryption key KEY.
- $PKEY(PublicKey)$: It means a public key for encryption in the public encryption algorithm, $PKEY_{ESP}$ is a public key of ESP.
- $SKEY(SecretKey)$: It means a secret key for decryption in the public encryption algorithm, $SKEY_{ESP}$ is a secret key of ESP.
- PW_{ESP} : It is a password for user authentication in ESP, input strings by user.
- PW_{APP} : It is a password for user authentication when the app starts, input strings by user's smart phone.
- HPW_{ESP} : It is a hashed password value saved in ESP.
- $KDF_i(PW)$: It means a Password-based key derivation function. It generates SS based on the input PW. On here, i means a key length.
- $a \parallel b$: It is a concatenation of a and b.
- TK_{ESP} : It means a password token include PW_{ESP}, ID used in ESP.
- $TKG(PW_{ESP}, ID)$: It is a function to generation TK_{ESP} using PW_{ESP}, ID.
- $P(a, b)$: For a, b, If $a = b$, then it proceed to the next stage of the protocol, but if $a \neq b$, then the protocol is stopped.

3.3 Assumptions

Before describe the details, the assumptions of this paper are as follows.

1. Users should pre-install SmartID_App on their smart phone to use the password token in the e-services. This app has a roll like a password wallet, so

the SmartID_App should be developed and installed via safety method.[8] Also, before the SmartID_App is run in a user's smart phone, the user of the app should be authenticated using a password. This password is set when the SmartID_App is installed first on the smart phone

2. SmarID_App should store ESP lists which can use the SmartID_App for user authentication and the public keys of each ESPs. Users can select the ESP name that the user wants to access now on SmartID_App, and if there are changes in the status of ESP lists supporting SmartID_App, the status should be updated through an application upgrade method on the smart phone. The public key of ESP is used to transfer the secret key which is needed between SmarID_App and ESP server.

3.4 Service Protocol

This mechanism uses the current authentication method used in e-service but the users use their smart phone instead of key board to input the password. That is, users can use the e-services same as the past, but they just select and transfer the password token in smart phone instead of entering the password via the keyboard in PC. From now on, we call this authentication method as SmarID_App-based method. The process of SmartID_App-based method consists of three major steps for user authentication. First process is an user registration to ESP and the password stored ESP as well as smart phone. And the second process is an user authentication process using password token which is store in smart phone. Last process is an user management process which is used when the user lost or change his smart phone.

Now, the detailed protocols of our SmartID_App -based authentication method are as follows.

Step1. User Registration Protocol

The process for user registration to ESP is similar with a general ESP registration process, but the password that used for an user authentication in ESP is stored in the smart phone as well as ESP server as a password token type. The password that stored in the user's smart phone is used when the user try to access an e-service instead of entering the password on PC. The password token selected by user in his smart phone would be transferred from the smart phone to the ESP for user authentication. The detailed protocol is as follows.

1. User applies for membership in ESP that the user intends to use. The user should enter his ID, password and his smart phone number to ESP after ESP performed an identity proofing of the user. The detailed identity proofing method is not mentioned here.
   (1) $T(USER, ESP, (ID, PW, PhoneNum))$

2. ESP server sends an authentication code which is randomly generated to the smart phone for the smart phone authentication. After the user confirms this

authentication code, he inputs this authentication code to ESP, and then the ESP can confirm the reality of smart phone number. User runs SmartID_App in his smart phone, and input the password for app operation. This password is set when the SmartID_App is installed in the smart phone by a user.
(2) $(ESP, SmartID\_App, authenticationcode)$
(3) $T(USER, ESP, authenticationcode)$
(4) $E(SmartID\_App)$
(5) $T(USER, SmarID\_App, PW_{App})$

3. The User selects the ESP name in the SmartID_App, and input the ID used on his smart phone. At this time, SmartID_App generates the shared secret key(SS) which is needed for secret communication between ESP andSmartID_App. SS is made with PW_{app} by key derivation function and its length is 128 bits. SS is sent from SmartID_App to ESP after encrypt using the public key of $ESP(PKEY_{ESP})$ which is stored in SmartID_App with ESP name.
(6) $SS = KDF_{128}(PW_{App})$
(7) $T(USER, SmartID\_App, (NameofESP, ID))$
(8) $T(SmartID\_App, ESP, (ID, ECN_{PKEY_{ESP}}(SS))$

4. ESP decrypts the encrypted secret key which is sent from SmartID_App using his private key($SKEY_{ESP}$). And he sent the hashed password which is matching with ID to SmartID_App after encrypt the hashed password using SS.
(9) $SS = DECS_{KEY_{ESP}}(ECN_{PKEY_{ESP}}(SS))$
(10) $T(ESP, SmartID\_APP, ENC_{SS}(Hash(PW_{ESP} \parallel salt)))$

5. SmartID_App will store the encryption value of the hashed password with ID and this is called password token. SmartID_App will store the password token with an icon to distinguish the token from others.
(11)$TK_{ESP} = TKG(ENC_{SS}(Hash(PW_{ESP} \parallel salt)), ID)$
(12) $S(SmartID\_App, (TK_{ESP}))$

6. ESP should save the user's registration information such as ID, the smart phone number and hashed password. PW is stored in ESP server after hashing using one-way hash function. For a password hash, the salt is used with the password. This makes the hash value of password to strong from some dictionary attack to stole the password.
(13) $(ID, PhoneNum, Hash(PW_{ESP} \parallel salt))$

Step2. User Authentication Protocol

To use ESP in the PC e-service environment, the user just inputs the ID without the password, and the password is sent from the user's smart phone after the user select the ESP icon in the SmartID_App. The detailed protocol is as follows.

1. The user inputs the ID on the ESP via PC. Then ESP sends SMS to the smart phone to request the password which is used to access the ESP site for user authentication.
   (1) $T(USER, ESP, (ID))$
   (2) $T(ESP, SmartID\_App, (PasswordRequestSMS))$

2. The user runs the SmartID_App after reading the SMS from ESP in his smart phone, and he inputs the password(PW_{APP}) to run the SmartID_App. The password token(TK_{ESP}) list is showed as icon type, the user selects the ESP name.
   (3) $E(StartID/_{A}app)$
   (4) $T(USER, SmartID/_{A}app, PW_{App})$

3. SmartID_App generates the shared secret key(SS) based on the user's input password(PW_{APP}). This secret key is used to decrypt the encrypted hashed password and the length of it should be longer that 128 bits. In here, we assume the length of the secret key is 128 bits. The user can select the password token SmartID_App to send a password to ESP. The hashed password which is included in the password token is sent to ESP after encrypted using SS and the SS is sent after encrypted by a ESP' public key.
   (5) $SS = KDF_{128}(PW_{APP})$
   (6) $SELECT(TK_{ESP})$
   (7) $T(SmartID/_{A}app, ESP, (ENC_{PKEY_{ESP}}(SS), TK_{ESP}))$

4. ESP decrypts the SS using his private key and encrypts the hashed password using SS. And then ESP confirms whether the decrypted hashed password is same with the stored hashed password in ESP. If the two values are same, the user's login to ESP is allowed.
   (8) $SS = DEC_{SKEY_{ESP}}(ECN_{PKEY_{ESP}}(SS))$
   (9) $P(Hash(PW_{ESP} \parallel salt), DEC_{SS}(ENC_{SS}(Hash(PW_{ESP} \parallel salt))))$

   As we can see this protocol, ESP just uses the password same with the past e-services. But the password is sent from the user's smart phone instead of keyboard input. In other words, ESP improves the password-based user authentication method with a minimal cost using a smart phone, which is widely used recently. Also the user don't need to remember his password to access ESP, and he can use the ESP just run SmartID_App and click the password token in SmartID_App. The password stored in the smart phone is hashed and encrypted, so the risk of exposure can be minimized. The password which is used to encrypt the hashed password is not save anywhere, and just entered by a user when he run a SmartID_App. So the possibility of exposure is close to zero.

Step3. Management Protocol of Smart Phone Number

If the user wants to change the smart phone number, he should login to the ESP specified in Step.2 and then he can changed the smart phone number. At that

time, the protocol of Step.1 is run again, and the new TK_{ESP} is generated and saved in the new smart phone.

But, if the user lost or stole the smart phone, the user must lock the login function by smart phone in ESP. For this, ESP should build the emergency login function in it. For this function, ESP can use to confirm the smart phone number after the user input the ID and password in ESP.

4 Analysis

In this section, well look our proposed SmartID_App-based user authentication scheme which is proposed to solve the problem of password-based authentication problems and further it will increase the level of trust of the password-based user authentication method.

1. SmartID_App-based-authentication method reduces the user's inconvenience, possibility of key logging, and phishing attack and provides user's non-repudiation.
   - Users Inconvenience : To use the normal password, the user should remember and input the password whenever the user access to the ESP, but in this method, there is no need to input of password. The user authentication is completed when the user select the password token which is saved in smart phone and generated during registration process of users. Only thing that the user should memory is just the password of SmartID_App. In other words, the user can access all e-serivces after login SmartID_App.
   - Key Logging : The password can be exposed by a malicious key logging tools during the user input the password in his PC even though the password is very safety consist of long and complex characters. This problem occurs when the user enters his password via a keyboard. But in our proposed method, there is no process that the user inputs his password, so it is safe from the key logging attacks.
   - Phishing Attack : Phishing attack occurs when the hackers built a fake site to know the user's password and he user input his password on that fake site. At this case, the hacker can hijack the password, and it can be used on the real site illegally. But on our proposed method, the fake site can't send SMS to the user's smart phone because it doesn't know the user's smart phone number. And the user doesn't need to input his password on the site. So the phishing attack to seize a password is impossible.
   - Non-repudiation : To use our proposed method, the user should have a smart phone before access to e-services. In other words, the user who access to the e-service must have had his smart phone, so he can't repudiate his action except the smart phone is stolen or lost.

2. SmartID_App based-authentication method improves the security level of password-based user authentication method.

   Our proposed method uses the password token instead of a normal password from the smart phone to ESP, so it provides two-factor authentication function. In other words, if the user doesn't have a smart phone, he cant use ESP, because the password is stored in the smart phone as a form of token. Thus, it's security level is enhanced to the multi-factor authentication capabilities such as OPT.

3. SmartID_App based-authentication method is interoperable with the wireless and wired e-service environment.

   E-services such as SNS, etc. are operated on the wireless network using smart phone as well as wired network using PC. So, many ESPs are providing the wireless services in the form of app and mobile web browsing services. In the case of wireless services using smart phone, the using of ID and password is very inconvenient due to the constrained mobile phone environment. So, some apps use the auto login function where their ID and PW are store in smart phone and then used in log-in, but if the smart phone is been rooting or hacking that information could be exposed to the other person. But, in the our proposed method, the ID and password are stored in the smart phone as a form of token, so the password token is used on the wireless serivce as itself and it can avoid the risk of lost.

As a results, SmartID_App based-authentication method can be used for an user authentication in the wireless e-services as well as wired e-services.

5 Conclusion

This paper proposed SmartID_App based password as a model to improve the safety of e-Society. Many services is opreated in wireless as well as wired environment. For user authentication using mobile phone like as smart phone, users tend to use easy password due to dufficulty of memory. So the security level of password can be low. But our proposed method store the password to the smart phone. So the password can be used on the PC and mobile internet using smart phone. As a result, it is expected that users will be able to conveniently use e-services using their smart phone. The password is not any more weak authentication method using the SmartID_App by this paper.

References

1. Ballagas, R., et al.: The smart phone: a ubiquitous input device. IEEE Pervasive Computing (2006)
2. ENISA, Security Issued and Recommendations for Online Social Networks, ENISA Position Papter No.1
3. The daily caller, Ten ways your smartphone is vulnerable to hackers (March 2012)

4. Housley, R., et al.: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (April 2002)
5. Google, `http://support.google.com`
6. Zhang, L., et al.: A Dynamic Password Identity Authentication System Based on Mobile Phone Token. Communications Technology (2009)
7. Cheng, F.: A Secure Mobile OTP Token. In: Cai, Y., Magedanz, T., Li, M., Xia, J., Giannelli, C. (eds.) Mobilware 2010. LNICST, vol. 48, pp. 3–16. Springer, Heidelberg (2010)
8. Jeun, I., Lee, K., Won, D.: Enhanced Code-Signing Scheme for Smartphone Applications. In: Kim, T.-H., Adeli, H., Slezak, D., Sandnes, F.E., Song, X., Chung, K.-I., Arnett, K.P. (eds.) FGIT 2011. LNCS, vol. 7105, pp. 353–360. Springer, Heidelberg (2011)
9. Mulliner, C.R.: Security of Smart Phone, Master's Thesis of University of California (June 2006)