2014 IEEE International Conference on Cloud Engineering

Cloud Password Manager Using Privacy-Preserved Biometrics

Bian Yang, Huiguang Chu, Guoqiang Li, Slobodan Petrovic, Christoph Busch
Norwegian Information Security Laboratory at
Gjøvik University College, Gjøvik, Norway
{bian.yang; huiguang.chu; guoqiang.li; slobodan.petrovic; christoph.busch}@hig.no

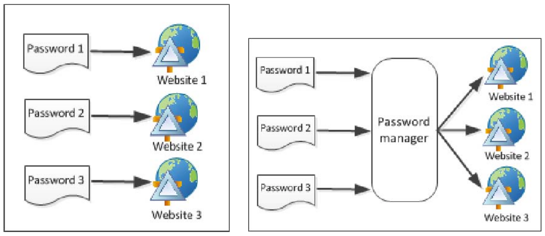
***Abstract*—Using one password for all web services is not secure because the leakage of the password compromises all the web services accounts; while using independent passwords for different web services is inconvenient for the identity claimant to memorize. A password manager is used to address this security-convenience dilemma by storing and retrieving multiple existing passwords using one master password. On the other hand, a password manager liberates human brain by enabling people to generate strong passwords without worry about memorizing them. While a password manager provides a convenient and secure way to managing multiple passwords, it centralizes the passwords storage and shifts the risk of passwords leakage from distributed service providers to a software or token authenticated by a single master password. Concerned about this one master password based security, biometrics could be used as a second factor for authentication by verifying the ownership of the master password. However, biometrics based authentication is more privacy concerned than a non-biometric password manager. In this paper we propose a cloud password manager scheme exploiting privacy enhanced biometrics, which achieves both security and convenience in a privacy-enhanced way. The proposed password manager scheme relies on a cloud service to synchronize all local password manager clients in an encrypted form, which is efficient to deploy the updates and secure against untrusted cloud service providers.**

Keywords—password manager; security; privacy preservation; biometrics; cloud

I. INTRODUCTION

Utilizing password for identity verification is a common authentication method for website login. Many people prefer using one password for all web services, which is insecure because the leaked password can compromises all web service accounts. Using independent passwords for different web services (as shown in Figure 1(a)) is inconvenient for the identity principal to memorize. In order to address this security-convenience dilemma, a password manager (as shown in Figure 1(b)) can be used to store and retrieve these passwords. An identity principal needs only to remember one password (called master key) which is used to log into password manager, and then login to various web services will be performed by the password manager automatically.

There are several methods to implement the password manager, such as storing the plain text password [1], using a cipher to encrypt passwords by means of a master key [2][3], using biometrics authentication [5][6], or using two-factor authentication by means of a master key and biometrics [12]. As a popular browser, Google Chrome [1] has a function to manage user's passwords and enables the users to save the usernames and passwords when the users login to the website for the first time. However, Google Chrome stores the password in plaintext format (as an example, Figure 2 shows a screenshot from Google Chrome). This could be risky if the computer's system account is compromised by an attacker or the computer is left unattended without logout. Storing password in plaintext format is also adopted by other prevalent browsers as analyzed in [4]. KeePass [2] and LastPass [3] are two password manager products which encrypt passwords by a master key. KeePass is a locally installed password manager, while LastPass is a cross-browser's extension with cloud synchronization. Such master-key-only based password managers, while providing data security by standard encryption, are subject to the risk of leakage of the master key.



(a) (b)

Figure 1: (a) Non-password-manager authentication and (b) password manager based authentication

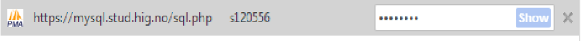

https://mysql.stud.hig.no/sql.php s120556 ••••••••

Figure2: Password management in the Google Chrome

Another threat during the authentication between users and web service providers is the passwords leakage from web service providers' side. Such event can happen in the real world [4]. An attacker can use the password hacked from a web service provider to impersonate a legitimate user. In order to overcome this vulnerability, biometrics based authentication systems [13] and some biometric password managers have been proposed to prevent the accidental or corrupted data leakage. For instance, M2SYS [5] is a biometric based password manager which implements a centralized biometric password management repository for single sign-on, while the way this product protects the biometric information from

978-1-4799-3766-0/14 $31.00 © 2014 IEEE
DOI 10.1109/IC2E.2014.91

leakage is unknown. Other hardware and biometric based password products, such as APC Touch Biometric Pod [6], can offer more convenience but they are token based and not suitable for the cloud identity authentication use. Although incorporating biometrics in password manager design can help alleviate the risk of leaking the master key for a password manager, it is almost impossible to assign a new biometric identifier [7] if the biometric characteristics (such as fingerprint, iris, etc.) are hacked. That is also a reason, among others, why people concerned about the privacy issues towards using biometric data for authentication [8] [9]. Therefore, it is desirable to address this privacy concern when it comes to a biometrics based password manager.

Bearing in mind the convenience, security, and privacy issues mentioned above, a cloud password manager with privacy-preserving biometrics is proposed in this paper. By the proposed scheme, a user needs only to login to the cloud password manager using one password (called the master key) and his / her freshly-captured biometric data before the authentication for a web service is automatically performed by the cloud password manager.

The rest of this paper is structured as follows: Section II gives the related work; Section III presents the details of the proposed design; and we conclude this paper in Section IV.

II. RELATED WORK

In this section, we briefly describe the two techniques related to our proposed scheme: (1) Single Sign On (SSO), which is a commonly-adopted mechanism in Internet community to manage passwords; and (2) Biometric Template Protection (BTP), which is a technology to secure the biometric data during authentication.

A. Single Sign On (SSO)

Single Sign On (SSO) is an independent software system giving users convenience by granting them access to all systems without needing login to each of them after login once. With this benefit, a user can access all computers and systems where the user has access permission by using a single action of authentication and authorization [13]. Using SSO can reduce the inconvenience of having an individual username and password for specific web site [14]. One of the advantages of SSO is that all the data used for authentication are stored in a central database which is easy to maintain. However, this feature also introduces some vulnerability. A security study of commercially-deployed SSO web service [15] discovered 8 confirmed security flaws. In short, SSO provides a solution to the challenge of managing multiple passwords, but it needs strengthening in security.

B. Biometric Template Protection (BTP)

Addressing the privacy concern is desired when we incorporate biometric data into the authentication process of a password manager. Biometric Template Protection (BTP) is such a technology to transform the biometric data into a protected template and store it in the database for direct comparisons without leaking biometric information.

Some template protection approaches are proposed [11], such as salting, which transforms biometric features by using a function defined by a user-specific key or password; noninvertible transform, which employs one-way transformation function; key-generating biometric, which generates cryptographic key from biometrics directly; and key-binding biometric cryptosystem, which binds the biometric template with a key within a cryptographic framework, *e.g.*, fuzzy commitment [10] shown in Figure 3. The idea of fuzzy commitment scheme is using Error-Correcting Code (ECC) to tolerate the intrinsic fuzziness of biometric signals and using exclusive-or to combine a biometric feature vector with ECC of a randomly-generated secret which can be used as a cryptographic key. This idea of biometric-secret combination is borrowed in the proposed scheme in this paper to combine an irreversibly-protected biometric template with a password which is to be saved by the password manager.

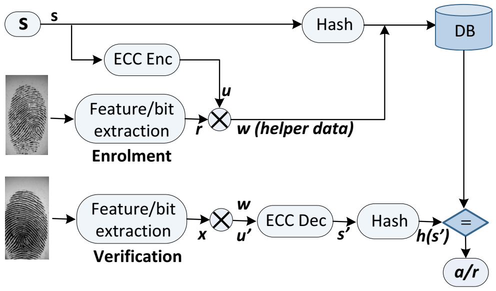


Figure 3: Fuzzy commitment scheme for biometric template protection

III. PROPOSED SCHEME

We propose in this section a cloud password manager scheme design which has the following merits:

(1) Biometrics is incorporated as the second factor into the password manager design, in addition to a master key.

(2) The authentication of the two factors (the master key and the biometrics) is an integrated process, which gives no information to an attacker about authentication result of each factor separately.

(3) Biometric template protection is used to implement a one-way function to generate a protected biometric template, from which no information about the raw biometric characteristic can be derived. The one-way function is designed to be rigidly one way so that it is computationally difficult to obtain the biometric input even if both the transformation parameter and the protected template are available for the attacker. Such rigidly one-way function can be those template protection methods permitting use of public parameters, *e.g.*, the dynamic random projection method in [17].

(4) Both the master key and the biometric data are protected in a way that leakage of one arbitrary factor of the two shall cause neither the leakage of the other one nor the leakage of the password saved in the password manager.

(5) The password manager is designed in a way that the two factors shall not be compromised even if the protected password is leaked from the service provider side. It even holds that, when both the protected password and any arbitrary one factor are leaked, the other factor can still be safe, thanks to the rigidly one-way function.

(6) The protected password can be updated at an arbitrary one client end and the updated password can be synchronized via the cloud server among all distributed clients.

The proposed password manager scheme design has two main functional components – password binding (PB), which combines a master key K_m, biometric feature B, and a password PSW to generate a password vault consisting of elements W_s and W_p; and password releasing (PR), which releases a password PSW from W_s and W_p via access to K_m and B. The details are described as follows.

A. Password binding (PB)

Password binding is a process which generates a password vault $\{W_s, W_p\}$ by taking as input a master key K_m, the biometric feature B, and the password PSW to be saved by the password manager. The whole process is presented in Figure 4:

Step 1: A True Random Number Generator (TRNG) is used to generate a random number s, which is used to hide the master key K_m by exclusive-or (XOR) operation to obtain the first vault element W_s;

Step 2: s is also used as an external parameter to a rigidly-irreversible (*i.e.*, irreversible under the situation where both the external parameter and the protected template are exposed to an attacker) biometric template protection method (denoted as $\text{BTP}^{\rightarrow}$) to generate a protected template PT;

Step 3: The generated PT is used to hide the Error-Correction-Code (ECC) encoded password (PSW) by exclusive-or (XOR) operation to obtain the second vault element W_p. The ECC adds some robustness to a PT, which is generated by those distance-preserving transformation algorithms such as the dynamic random projection [17] and thus modulates some fuzziness inherent in B.

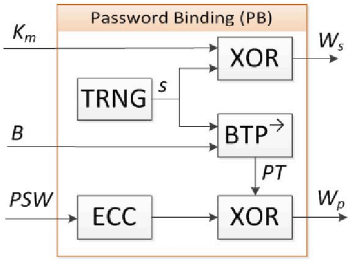


Figure 4 Password binding process

Note that after the password binding process, both the randomly-generated number s and the generated protected template PT are discarded. The fact that only W_s and W_p are saved in the password vault requires that the user simultaneously give correct K_m and B as input in order to release the correct password PSW. This can be illustrated further by the password releasing process shown in Figure 5.

B. Password releasing (PR)

Password releasing is a process which releases a password PSW' from the password vault $\{W_s , W_p\}$ by taking as input a master key K'_m and the biometric feature B'. The whole process is presented in Figure 5:

Step 1: A secret s' is recovered by exclusive-or (XOR) operation of K'_m and W_s;

Step 2: The recovered s' is used as an external parameter to the rigidly-irreversible biometric template protection method $\text{BTP}^{\rightarrow}$ to generate a protected template PT';

Step 3: A password PSW' is derived by firstly performing exclusive-or (XOR) operation of PT' and W_p, and secondly Error-Correction-Code decoding (ECC^{-1}) the XOR result.

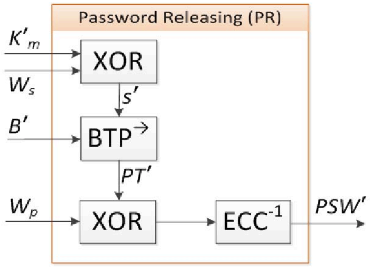


Figure 5 Password releasing process

Note that after the password binding process, both the recovered number s' and the re-generated protected template PT' are discarded. There is no verification for either the recovered s' or the released PSW' during the whole password releasing process (PSW' shall be forwarded directly to the service provider, *e.g.*, web email, system login, *etc.*, for verification), making the password manager difficult for hacking (*i.e.*, an attacker is unable to hack the master key K_m or the biometric input B separately).

C. Password manager maintenance

Table I shows an example of the data structure of the proposed password manager scheme in the cloud server end. Three attributes, namely User_Index, Record_Index, and Encrypted_Record, are available for a user to maintain in the cloud end. User_Index are indices to different users' cloud storage of their password manager records. To locate the cloud storage of a user i ($1 \leq i \leq N$, assuming in total N users registered in the cloud database), his / her username *UserName_i* and master key K_{m_i} are required to generate an index by a mathematical hash function H(.), *e.g.*, SHA-2/3: $H(H(UserName\_i),H(K_{m\_i}))$, where $H(K_{m\_i})$ as a unique identifier is used to add salt to the username. After locating the user i's whole storage, login service information *ServiceName_t* ($1 \leq t \leq M$, assuming in total M records stored in user i's password manager) and the user's account name *AccountName_t* are required together with $H(K_{m\_i})$ to generate a record index $H(H(ServiceName\_t, AccountName\_t),H(K_{m\_i}))$ used to locate the t-th record for user i. The t-th record is encrypted by standard encryption algorithm, *e.g.*, AES-256, into an encrypted record $\text{Enc}_{Km\_i}\,(W_{s\_i\_t} , W_{p\_i\_t})$. Note that the records are encrypted separately so that they can be updated and synchronized individually without worry about the data exchanging efficiency even if a user has a huge amount of records saved in a password manager.

Table I. An example of data structure of the proposed password manager in the cloud server

Password manager server database		
User_Index (*e.g.*, *UserName* = John_Smith)	**Record_Index** (*e.g.*, *ServiceName* = www.google.com, *AccountName* = jsmith1984)	**Encrypted_Record**
H(H(*UserName*_1), H(K_{m_1}))	…*UserName*_1's record index(indices)	…*UserName*_1's encrypted password manager record(s)
H(H(*UserName*_2), H(K_{m_2}))	…*UserName*_2's record index(indices)	…*UserName*_2's encrypted password manager record(s)
…	…	…
H(H(*UserName_i*), H(K_{m_i}))	None (Virtual Service for user authentication)	$Enc_{K_{m\_i}}\left(W_{s\_i\_0}, W_{p\_i\_0}, h_{psw\_i\_0}\right)$
	H (H(*ServiceName*_1, *AccountName*_1), H(K_{m_i}))	$Enc_{K_{m\_i}}\left(W_{s\_i\_1}, W_{p\_i\_1}\right)$
	H (H(*ServiceName*_2, *AccountName*_2), H(K_{m_i}))	$Enc_{K_{m\_i}}\left(W_{s\_i\_2}, W_{p\_i\_2}\right)$
	…	…
	H (H(*ServiceName_M*, *AccountName_M*), H(K_{m_i}))	$Enc_{K_{m\_i}}\left(W_{s\_i\_M}, W_{p\_i\_M}\right)$
…	…	…
H(H(*UserName_N*), H(K_{m_N}))	… *UserName_N*'s record index (indices)	… *UserName_N*'s encrypted password manager record(s)

The password manager can be maintained in the following ways. Before any maintenance operation, a password manager client shall prompt the user to submit the User_Index first and then download the user authentication virtual record for identity verification (detailed in the following (1)).

(1) Authentication to the cloud password manager

Before downloading the actual password records to the client end, or creating new records, updating records, or deleting records, in the client end, the user shall be prompt for identity authentication. To achieve this goal, a virtual service implemented with a virtual record (see Table 1) Enc$_{Km\_i}$($W_{s\_i\_t}$, $W_{p\_i\_t}$, $h_{psw\_i\_t}$) for user authentication is first used to verify if the hash value $h'_{psw\_i\_t}$ = H(PSW') of the released PSW' can match $h_{psw\_i\_t}$, which is the hash value of the virtual password *PSW* assigned earlier by the password manager. Note that this *PSW* is merely generated for this identity verification purpose.

(2) Creating a new (or updating an existing) password record

A new password record can come into creation by generating a new password *PSW* for the login service or adopting the exsiting password *PSW* registered by the login service. In both cases the password *PSW* shall be bound with the master key K_m and biometric input *B* as in Figure 4. The record is always created locally in the password manager client end and then updated to the cloud server end.

(3) Deleting a password record

A password record can only be deleted first in the client end and then the status "deleted" will be updated to the cloud server end.

(4) Synchronization

For synchronization, the records in the client end are updated to the corresponding records in the cloud server end following an identity authentication process described in (1) in the client end. The synchronized data in the client database can be decrypted by the master key K_m to obtain the password vault elements W_s and W_p.

IV. Conclusion

The privacy-preserving biometrics (using rigidly-irreversible template protection method) based password manager scheme proposed in this paper enables people to use two factors (a master key and the biometrics) for password binding and releasing in a secure way. Cloud infrastructure is used to synchronize all the password manager clients with the updated encrypted records. Neither the biometric features nor the master key is needed to be transmitted to the cloud end and the whole authentication process takes place merely in the client end which is highly privacy-respected. This is especially suitable for the password manager services hosted by untrusted cloud service providers.

References

[1] https://support.google.com/chrome/answer/95606, "Manage your website passwords," *Google Chrome help* .[Accessed on 18-01, 2013]

[2] http://keepass.info/help/base/security.html, "Security - Detailed information about the security of KeePass," *KeePass Help Center*. [Accessed on 18-01, 2013]

[3] https://lastpass.com/how-it-works/, "The secure and trusted way to store passwords," *LastPass home page*.[Accessed on 18-01, 2013]

[4] S. Gaw and E. W. Felten, “Password management strategies for online accounts,” *Proceedings of the second ACM symposium on usable privacy and security*, 2006. p. 44-55.

[5] http://www.m2sys.com/EBS.htm, “Biometric secure single sign-on software,” *M2SYS Enterprise Biometrics Suite*. [Accessed on 18-01, 2013]

[6] http://www.apc.com/resource/include/techspec_index.cfm?base_sku=BIOPOD, “APC Touch Biometric Pod Password Manager,” *APC home page*. [Accessed on 18-11, 2013]

[7] Y. Sutcu, H. T. Sencar, N. Memon, “A secure biometric authentication scheme based on robust hashing,” *Proceedings of the 7th workshop on Multimedia and security*, pp. 111-116, 2005..

[8] C. Rathgeb, A. Uhl. “A Survey on Biometric Cryptosystems and Cancelable Biometrics,” *EURASIP Journal on Information Security*, 2011(3), Springer Verlag, 2011.

[9] J. Breebaart, B. Yang, I. Buhan-Dulman, and C. Busch, “Biometric template protection,” Datenschutz und Datensicherheit-DuD, vol.33, no.5, pp.299-304, 2009.

[10] A. Juels and M. Wattenberg, “A fuzzy commitment scheme,” *Proceedings of the 6th ACM conference on Computer and communications security*, pp. 28-36, 1999.

[11] A. K. Jain, K. Nandakumar, and A. Nagar, “Biometric template security,” *EURASIP Journal on Advances in Signal Processing*, Volume 2008, Article ID 579416, 2008.

[12] A.A.Yassin, H. Jin, A. Ibrahim, D. Zou, “Anonymous password authentication scheme by using digital signature and fingerprint in cloud computing,” In *proceedings of Second International Conference on Cloud and Green Computing*, 2012.

[13] J. D. Clercq. “Single sign-on architectures,” *Infrastructures Security*. Springer Berlin Heidelberg, pp.40-58, 2002.

[14] E. Tsyrklevich and V. Tsyrklevich, “Single sign-on for the Internet: a security story,” *BlackHat USA*, Las Vegas, 2007.

[15] R. Wang, S. Chen, X. Wang, “Signing me onto your accounts through Facebook and Google: a traffic-guided security study of commercially deployed single-sign-on web services,” *IEEE Symposium on Security and Privacy*, pp. 365-379, 2012.

[16] R. Veldhuis and K. Tom, “Biometric Template Protection,” *Introduction to Biometrics*, 2012. http://imaging.utwente.nl/open/courses/intro_biometrics/121090_Lecture08.pdf. [Accessed on 18-01, 2013]

[17] B. Yang, D. Hartung, K. Simoens, C Busch, “Dynamic random projection for biometric template protection,” *Proc of the 4th IEEE Int Conf on Biometrics: Theory, applications and systems* (BTAS'10), 2010.