# Cloud Password Manager Using Privacy-preserved Biometrics

Huiguang Chu

Master's Thesis
Master of Applied Computer Science
30 ECTS
Department of Computer Science and Media Technology
Gjøvik University College,

Avdeling for
informatikk og medieteknikk
Høgskolen i Gjøvik
Postboks 191
2802 Gjøvik

Department of Computer Science
and Media Technology
Gjøvik University College
Box 191
N-2802 Gjøvik
Norway

# Cloud Password Manager Using Privacy-preserved Biometrics

Huiguang Chu

2014/06/01

# 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 paradox 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 manage 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.

Therefore, our goal in this thesis work is to design a privacy preserved and security enhanced password manger by using the human unique biometrics attributes. Based on the purpose, several technical aspects i.e., authentication schemes, existing password manager taxonomy, biometrics template protection, offline storage techniques, encryption and decryption algorithms and so on have been surveyed in this thesis. A novel scheme for password manager authentication, password binding, releasing and protecting is proposed. On the basis of the proposed scheme, a global structure is designed for a real password manager named NBLpass, which is implemented as well. NbLpass password manager uses the proposed privacy-preserved and security-enhanced scheme through combining facial features with plain text password, and it is capable of working locally and being synchronized with a cloud database. By using the NBLpass password manager, a user needs only to login to the password manager using one password (called the master key) and his / her freshly captured biometric data prior to the authentication of a web service.

# Preface

I would like to thank Bian Yang as my supervisor and help me in the whole master thesis process.

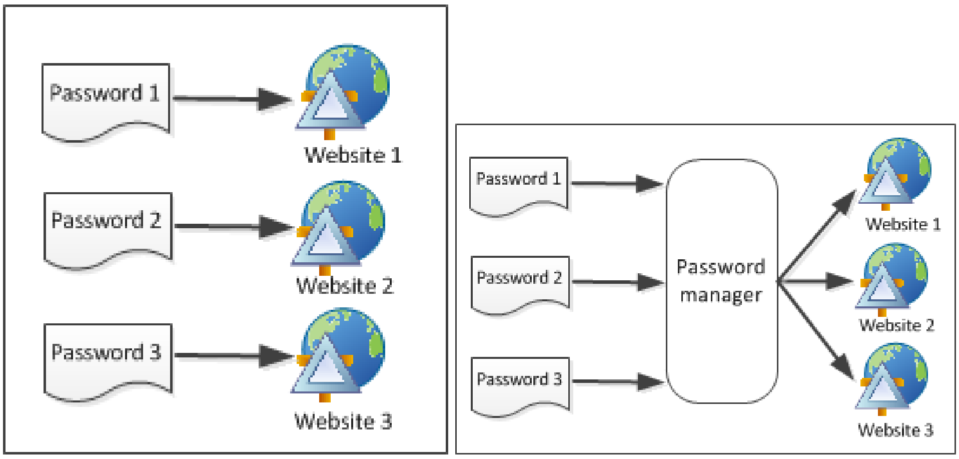
# Contents

# List of Figures

# List of Tables

# 1 Introduction

Utilizing password for identity verification is a common authentication method for website login. Many people prefer using one password for several web services, which is insecure because the leaked password can compromise all web service accounts. While, 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.



(a) Non password manager authentication (b) Password manager based authentication

Figure 1: Difference between authentication with and without password manager

## 1.1 State of The Arts

In the real web world, there are several methods to implement a 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 [4] [5] [6], or using two-factor authentication by means of a master key and biometrics [7]. 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 if we allow Chrome to remember the password. While, when user delete the data the plaintext password will be removed as well. 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 [8]. 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.

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 [8]. 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 [9] 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 leakage is unknown to public. 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-based 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 if the biometric characteristics (such as fingerprint, iris, etc.) are hacked [10]. That is also a reason, among others, why people concerned about the privacy issues towards using biometric data for identity authentication. Therefore, it is desirable to find a solution to address this privacy concern when it comes to a biometrics based password manager.

## 1.2 Motivation

In order to better balance the convenience, security, and privacy issues mentioned above, we proposed a novel solution to a cloud password manager with privacy-preserved biometrics in this thesis work. 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 prior to the authentication of a web service.

Compared with a traditional password manager, several benefits of using NBLpass can be summarized as follows. (1)More convenient: the users only have to remember the password manager login password (master key) and the cloud database synchronization password. All of the other passwords those users owned will be remembered by the password manager. In additional, users do not need to recall a password when he/she register a website, the password manager will help them to generate a strong password. (2)Securer: compared with the only text based authentication and the only biometrics based authentication, we use the fuzzy commitment [11] to integrate the text password and the unique biometric information. (3)Flexible: The user passwords can be synchronized with the cloud database and the password manager works as a browser extension, capable of being installed and used anywhere. (4)Privacy preserved and security enhanced: all of the plain biometrics data cannot be obtained by the attacker due to the biometrics information has been well protected and the protected template makes exclusive OR operation with a random number which is immediately discarded after use.

The general purpose of the thesis work is to design a privacy preserved and security enhanced password manger by using the human unique biometrics attributes. Based on the purpose, several technical aspects i.e., authentication schemes, existing password manager taxonomy, biometrics template protection, offline storage techniques, encryption and decryption algorithms and so on have been surveyed in this thesis. A novel scheme for password manager authentication, password binding, releasing and protecting is proposed. On the basis of the proposed scheme, a global structure is designed for the NBLpass, which is implemented as well. NbLpass password manager uses the proposed privacy-preserved and security-enhanced scheme through combining facial features with plain text password, and it is capable of working locally and being synchronized with a cloud database.

The rests of this paper are allocated as follows: Chapter II gives the preliminary knowledge that is related to the whole system; Chapter III presents the details of the system design; Chapter IV gives the implementation details of the system. Further work is proposed in chapter V followed by Conclusion in Chapter VI.

# 2 Preliminaries

## 2.1 Web-Based Identity Authentication Schemes

In today's web-enabled world, web services are being improved rapidly and capable of providing a variety of services for users exchanging information over the network. However, the privacy and confidentiality are vulnerable to attacks. Therefore, how to provide a secure and convenient web service for users is important and security solutions are needed to protect users' privacy. In current server-client based web architecture, remote authentication is regarded as the most important security service existing in many products and applications services. Many applications in our daily life also require user authentication as well. For example, online shopping, travel booking, online bank, social benefit claiming, *etc*. Such online applications require users be authenticated before they are granted with the access to login.

In fact, various schemes with a different degrees of reliability and different level of security has been utilized to perform a remote authentication. The current schemes can be categorized into three types: knowledge-based, token-based and biometrics-based [12]. Knowledge-based schemes such as text passwords and graphical passwords are the most widely used authentication schemes. Token-based scheme such as a smart card is widely used. Many token-based authentication systems also use knowledge-based techniques to enhance the security. For instance, ATM cards are generally used together with a PIN number. Biometrics-based schemes such as fingerprints, iris recognition are not yet widely utilized in the commercial area. However, they are gaining people's attention and increasing market currently.

### 2.1.1 Web-based authentication

According to the definition [13], a web-based application means that a program use HTTP/HTTPS connection to access over a network; While, non web-based application refers to a program running within a device's own memory. The web-based application might run in web browsers or run independently but have an http/https connection to the server. By understanding the web-based application, it is not difficult to understand what web-based authentication is. Take a simple example of use, when a user initiates an http/https session through a web-based application, the web-based application intercepts http/https packages from the host and may send a login page/form to the user. The user type in his/her credentials, and then the web-based application sends it to the server for authentication. If the identity is authenticated, the web-based application sends a login-successful page/form to the user and applies the access policies returned by the server. If the identity authentication fails, the web-based application may send a login-failure page/form to the user, which also may prompt the user to try it again. This simple process can be regarded as a web-based authentication process.

It is known from the studies [14] [15], there are several benefits of web-based authentication as follows. (1)No area limitation: web-based authentication is typically implemented via a network connection or a web browser, which allows audiences in a wider geographical area for identity authentication as long as users can access the network; (2)Familiarity: web-based authentication is extensively available in daily life such as those social websites(facebook, linkedin), and most end users are familiar with them. However, there are also some disadvantages of the web-based authentication such as :( 1) Security: when it comes to the web, security is a vital problem; (2)Not transparent: the web-based authentication may occur in a virtual network which is not transparent for end users.

### 2.1.2 Classification of identity authentication approaches

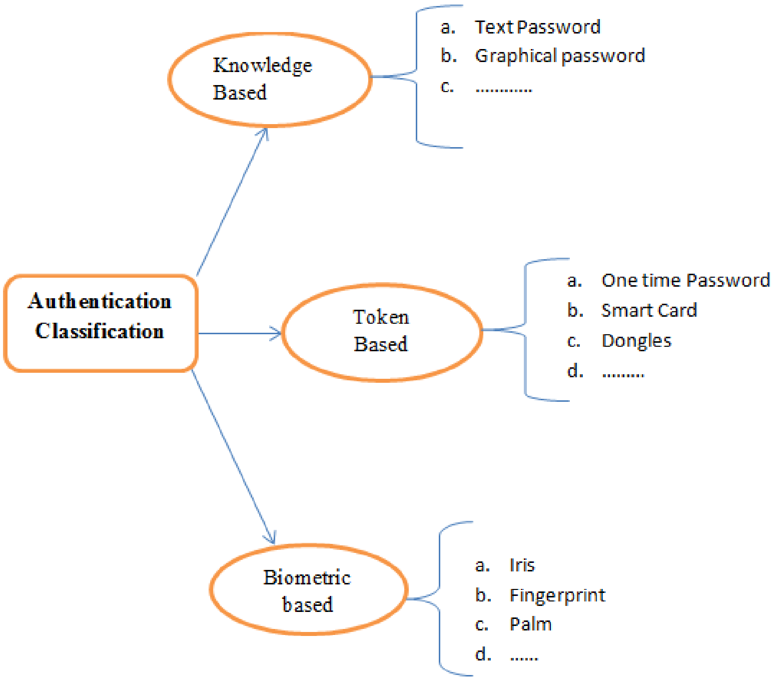


Figure 2: Authentication approaches classification

#### (1) Knowledge-based Authentication

Knowledge-based authentication means that the web services seek to prove the identities of the users who want to access the resources of the websites or applications. The oldest and most primitive scheme is the text password based authentication [16]. In this scheme, when user login to a remote server, they should submit a user name and a text password (in order to secure the user's password, the password stored in the remote server database is encrypted with some algorithms rather than plain text [16])to the remote server via a remote authentication web service. While receiving the login information, the remote server will check if the submitted information can be matched in the verification table. If the submitted information can be matched, the access will be granted (Figure 3). However, this method is widely known that if the password is long, it is harder to be remembered by the users; if the password is short it is easier to be guessed by an attacker [17].

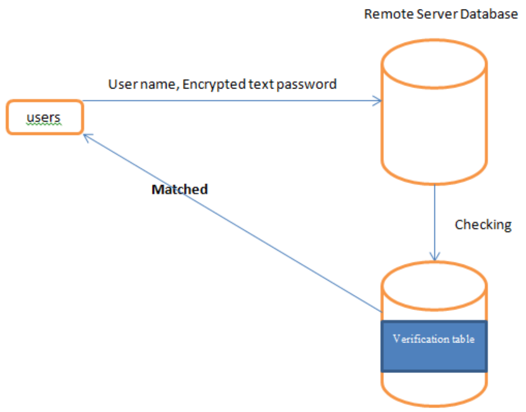


Figure 3: Plain text password system workflow

Considering the problems in the text password scheme, graphical password scheme is proposed as a possible alternative to text based schemes. Studies have shown that humans can remember pictures easier than strings [18]  [19]. So far, there are a lot of graphical password-based schemes such as Jensen et al. model [20], Passfaces model  [21] and other schemes mentioned in [22]. Take a look at a study the "Deja Vu" system [19],including three phases(Figure 4). Firstly,

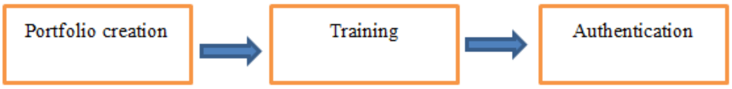


Figure 4: Graphical password scheme workflow

select a subset of p images from a set of sample images. In order to improve user's memorability, a training is needed after the creating the portfolio. And then, to authenticate the user, the system presents a challenge set, consisting of n images. This challenge contains m images out of the portfolio. Moreover, remaining n-m images decoy images. If the user correctly identifies the images, which are part of her portfolio, the user will be authenticated and granted (Figure 5). While, there still exist some possible attacks, for example, closed circuit hidden cameras can be placed for tracking the entry made into the system and the screen can be recorded by Spyware (even though there is no such Spyware so far, it is an potential one) [22]. Therefore, passwords can be stolen by making observation on the user.

### (2) Token-based Authentication

Token-based authentication is a security technique that allows the users to enter their username and password in order to get a token which grants them the access to fetch resources in the secure server or network without using their user name and password [12]. Due to the mobile banking

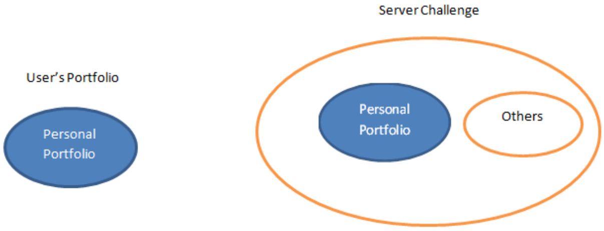
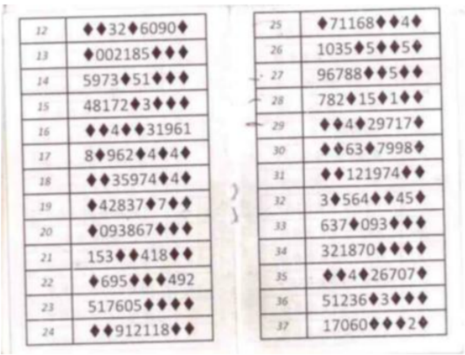


Figure 5: Graphical password scheme authentication

is popular and the security problem, saurap, prasadn, and rbhaska proposed two schemes based on a research with 50000 customers in India. One old scheme is based on a secret 4-digit PIN code and a codebook(Figure 6)[23].



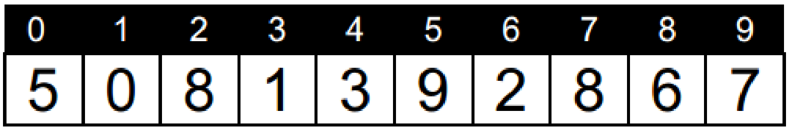
| | |
|---|---|
| 12 | ♦♦32♦6090♦ |
| 13 | ♦002185♦♦♦ |
| 14 | 5973♦51♦♦♦ |
| 15 | 48172♦3♦♦♦ |
| 16 | ♦♦4♦♦31961 |
| 17 | 8♦962♦4♦4♦ |
| 18 | ♦♦35974♦4♦ |
| 19 | ♦42837♦7♦♦ |
| 20 | ♦093867♦♦♦ |
| 21 | 153♦♦418♦♦ |
| 22 | ♦695♦♦♦492 |
| 23 | 517605♦♦♦♦ |
| 24 | ♦♦912118♦♦ |
| 25 | ♦71168♦♦4♦ |
| 26 | 1035♦5♦♦5♦ |
| 27 | 96788♦♦5♦♦ |
| 28 | 782♦15♦1♦♦ |
| 29 | ♦♦4♦29717♦ |
| 30 | ♦♦63♦7998♦ |
| 31 | ♦♦121974♦♦ |
| 32 | 3♦564♦♦45♦ |
| 33 | 637♦093♦♦♦ |
| 34 | 321870♦♦♦♦ |
| 35 | ♦♦4♦26707♦ |
| 36 | 51236♦3♦♦♦ |
| 37 | 17060♦♦♦2♦ |

Figure 6: Codebooks used in Eko's current scheme (Adapted from [23])

As shown in the Figure 6, the codebook contains 10 digit length string including 6 digit random and 4 blank spaces. Before starting the transaction, the users need to authenticate him/her, firstly, the user create a formatted transaction message with appending a 10 digit numeric signature. Examples in Figure 7, each signature are formed by looking up the first unused string in the codebook and placing the PIN code in 4 blank spaces position. For authentication, the server will check if t a user must place his PIN code in the blank spaces for the current nonce correctly. For example, if user uses the 15th nonce in the codebook, and his pin is 5432, his signature for the current transaction is 48172**53432**. However, there is an problem in this scheme is that the PIN code is possible to suffer recovery attacks.

Considering this, they came up the new scheme and the format of the codebook was changed. In the new scheme, each entry is a 10-digit nonce with the position labeled from 0 to 9. The user authenticate himself/herself by looking up the first unused nonce in codebook with a PIN code, for example 6391, and forming a 4-digit number consisting of the 6th, 3rd, 9th, and 1st digits in the nonce(**2170**). Then the bank recomputes the signature using the locally stored PIN and codebook. While there still exists one security issue, i.e., the threat from the man in the middle

(MITM) [24] attacks. The adversary can intercept communication between the user and server and then can modify the message. The bank will still regard the modified message come from the legitimate user as the signature will be a valid PIN encoding [23]. In addition, there are a lot of token-based authentication schemes such as using mobile phone [25], using smart card with image encryption [26] and some others [27] [28],etc.



| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 0 | 8 | 1 | 3 | 9 | 2 | 8 | 6 | 7 |

Figure 7: Nonce in new scheme (Adapted from [23])

### (3) Biometrics-based Authentication

Biometrics-based authentication refers to identify the identity of one person by recognizing humans based on the one or more unique physical attributes or behaviour [29] [12]. There is a large scale of systems use the unique characteristics like fingerprint, iris or voice and face recognition to identify users. The biometric device capture a user's biometric data such as iris or fingerprint and convert it to digital information that computer can interpret and recognize [30]. This is illustrated clearly in Figure 8.

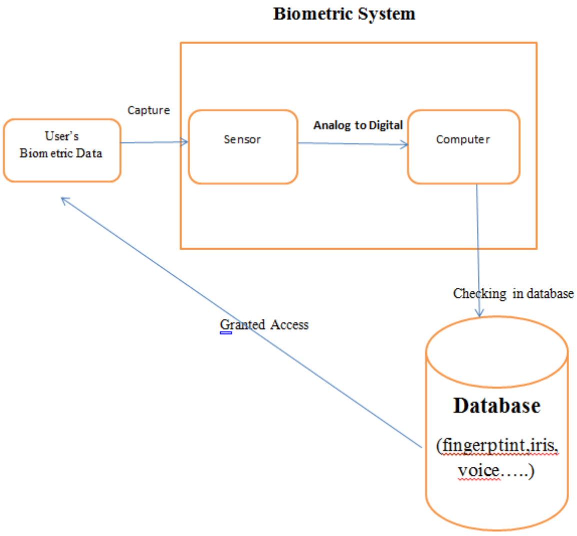


Figure 8: Biometrics-based authentication structure

Take one existing scheme as example [10], in this scheme, the authors analyzed the performance and feasibility of biometrics-based authentication scheme which relies on a robust hash function(one-way function designed as a sum of many Gaussian functions) and a cryptographic hash function. Authentication process is performed as follows: Firstly, user's biometric data will be acquired with a sensor and his/her feature vector will be extracted. Secondly, one-way transformation will be generated, and it will be evaluated at the extracted feature vector component values. Finally, values obtained after quantization will be concatenated together to form a string and then hashed. This scheme is illustrated in Figure 9. However, this scheme is not 100 percentages perfect, even though that Gaussians value cannot be guessed easily, the transformation are not used in an efficient way [10]. If attackers are the expert of the mathematics, he/she can reduce the brute force guessing space.

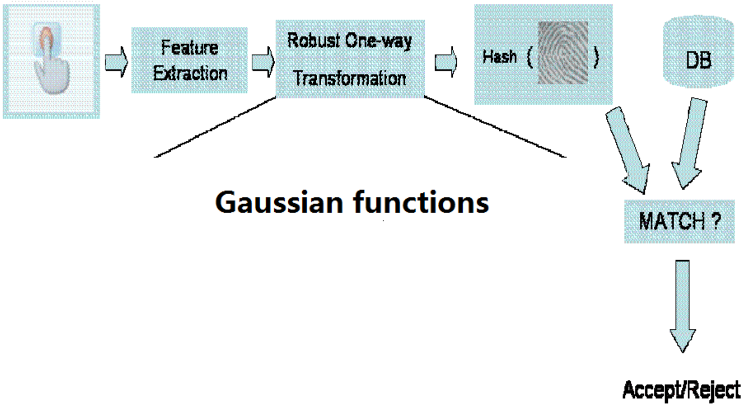


Figure 9: One-way transformation scheme (Adapted from [10])

Another scheme using Iris recognition to do the authentication has been studied by ARWA and DR.lehab [31], they proposed a biometric encryption to mobile web services authentication. The user iris will be used to regenerate the user's encryption key on the fly on each time the user needs to be authenticated. The iris features is extracted and encrypted by using the fuzzy commitment scheme (which is described in section 5), this is illustrated in Figure 10. In details, extracting iris features and then binding it with a random generated key. Then the biometric encryption template ((BE template) and key hash value will be stored in user's mobile devices. The purpose of storing hashed key value is that it can be capable to reject incorrect keys before starting the remote authentication process. For the authentication, the service decrypts the tickets from the Key distributed center and exacts the session key and time stamp, if the time stamp is matched, the user will be authenticated. Security issue in this scheme seems to be solved well, while there is a problem is that iris recognition's accuracy and has a high dependency on the sensor.

### 2.1.3 Comparison

The advantages and disadvantages of the various authentication approaches are summarized in Table 1. For the basic authentication, password is the simplest way to be implemented and this is why password schemes are so common. Unfortunately, we can see from the Table 1, password is the weakest authentication scheme because it is possible to be guessed and stolen easily. Moreover, users often choose weak password. Therefore, it seems the weakest way to do the authentication.

Some systems commonly use tokens to do the authentication and they are devices (e.g. Don-

Table 1: Advantages and disadvantages of each authentication schemes

| | **Pros** | **Cons** |
|---|---|---|
| **Knowledge-based authentication** | a. graphical passwords are hard to guess by the attacker<br>b. easy to communicate<br>c. pictures are easier to remember | a. text password is difficult to remember<br>b. easily Predictable<br>c. prone to dictionary attacks, brute force attacks, observation attacks |
| **Token-based authentication** | a. It increases security compared to passwords<br>b. simple and cheap to be produced | a. accessibility<br>b. observable and possible to be replicated<br>c. can be lost and stolen<br>d. simple and cheap to reproduce |
| **Biometrics-based authentication** | a. highest level of security<br>b. positive and accurate identification<br>c. safe and user friendly | a. susceptible to replay-capture of data and reuse<br>b. biometrics systems need some special hardwares which are expensive<br>c. unreliable<br>d. perceived privacy threats |

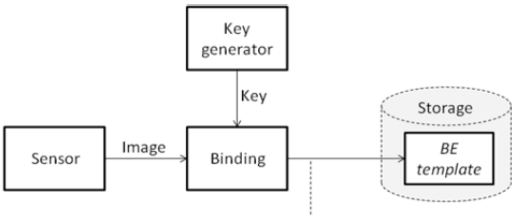


Figure 10: Iris-based authentication (Adapted from [31])

gles) or objects (Smart card, student card...) that can authenticate users. Current examples include credit card, university student card and physical keys, which are widely supported and cheap to produce and apply. However, token has its own disadvantages as well. For example, token is simple, cheap and easy to be made. This makes it vulnerable to be reproduced by the attackers. In addition, typically token is a physical object or device, they can be stolen more easily.

Biometric systems according to humans have individual specific physical characteristics, which can be identified uniquely to do the authentication. Humans are conditioned to recognize these attributes and utilize them for authentication. Obviously, because the unique individual biometric attributes the authentication is relatively secure. However, it is more difficult for the digital devices to recognize the analog characteristics such as voice, fingerprint directly. Therefore the processing of the biometric attributes is very complicated, requiring some specific hardware and commonly this hardware are expensive. Moreover, in the open environment, it is possible threats are that the biometric information could be regenerated and misused by the attacker.

## 2.2 Password Managers

There are several sorts of password managers are studied in this paper, such as storing the plain text password [1], encrypting the master key with specific algorithms [2] [3], using biometrics authentication [4] [5] [6], utilizing two factors authentication [7]. Commonly, the password manager in the browsers such as Google Chrome [1], enable users to save the user name and password when the users login some websites first time. However, there is one striking problem, looking at the Figure 11, a screenshot from Google Chrome browser's password manager, the password is stored in plain text. The problem that major browsers store password in plain text is also uncovered in [8]. KeePass [2] and LastPass [3] are the commonly used independent password manager. KeePass is a locally installed password manager while the LastPass is a cross-browser's extension with cloud synchronization. They have one common point where the users' passwords are stored in an encrypted format by using the Advanced Encryption Standard (AES) 256 algorithm. However, the AES-256 is reported to be not secure and the attacks on AES-256 algorithm has been studied in [32] [33]. In additional, passwords leakage already happened in the real world [8] and study in [9] shown that biometric based authentication systems can be used to improve the owner authentication process security and make agent and owner reputations strictly related. Therefore, some biometric password managers has been come up to

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

Figure 11: Password manager in Chrome browser

prevent the accidental or corrupt data leakage. For instance, M2SYS [5], allowing businesses to implement a centralized biometric password management repository for single sign-on and secure access to applications and web sites. However, you need to install on the PC, inconvenient. Some other removable biometric password such as myIDkey [4], APC Touch Biometric Pod [6], become more convenient but they might be lost easily. Despite using biometrics is more secure and we do not have to remember password, they have some weakness as well. Unlike the password, once the biometrics information is comprised, it is almost impossible to assign a new one [10]. That is also why some people concern the privacy issues when using the biometrics to do the authentication and it is not widely used in our daily routines nowadays.

## 2.3 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[34]. Using SSO can reduce the inconvenience of having an individual username and password for specific web site[35]. 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[36] discovered 8 confirmed security flaws. In short, SSO provides a solution to the challenge of managing multiple passwords, but it needs strengthening in security.

## 2.4 Biometrics for Authentication

Biometrics is a field of technology which has been and is becoming the foundation of identification of individuals based on some physical attribute or behavioral characteristic [37]. Generally, there are several biometrics features have been extensively used in the real world. For example, facial feature, fingerprint, hand geometry, iris, retinal, signature and voice, etc. Through the existed study in [38], Speaker and fingerprint were the first explored. The application was in early 1960s. In the 1970s, a hand geometry system has been developed. Retinal and signature verification came in 1980s and the Iris recognition system came in the 1990s. Each of them is briefly described as follows.

### 2.4.1 Taxonomy of biometrics attributes

**(1) Fingerprint**

Fingerprint has been widely used for many years and it is regarded as a reliable and unique identity verification solution [39]. It might be a quite good choice for in-house system due to we can give users adequate explanation and training and the system could be controlled. Since the low cost,small size and ease of integration of fingerprint verification system/devices, it seems to be extensively used in the workstation access.

**(2) Face**

Facial recognition analyzes facial features to to the identity verification. It requires a digital system to capture the users facial image, and it need extra peripheral equipment rather than the basic PCs.

**(3) Iris**

Iris scans are used to do the identity authentication by analyzing the features of color patterns in the Iris.Due to it is not easy for using and system integration, it has a exception of making big improvements.

**(4) Retina**

Retina based biometrics are used to implement the identity verification by analyzing the arrangement of the subject's blood vessels which exist at the back of the eye. It is not widely accepted by the users due to it requires the user to look into an receptacle and focus on a given point.

**(5) Voice**

Voice verification uses a voice print, which is a text transformed from the voice, to analyze how a user speak a particular word or sentence unique to him/her. However, poor quality and even the small noise in the environment can affect the authentication accuracy, therefore, it is not user friendly.

**(6) Signature**

Signature verification analyzes how the users sign her/his name, belonging to the behavior feature recognition. The features could include signing speed, pressure and the shape. Even though signature verification are reasonably accurate in operation, few signature applications have emerged compared to the other biometrics characteristics.

Whereas, when it comes to the biometrics authentication, there are three issues need to be concerned, biometrics privacy issue, biometrics security issue and biometrics efficiency issue. They are also the main focuses in the biometrics system implementation in this thesis.

## 2.5 Biometrics Protection Template

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

Some template protection approaches are proposed[40], such as salting, which transforms biometrics features by using a function defined by a user-specific key or password; noninvertible transform, which employs one-way transformation function; key-generating biometrics, which generates cryptographic key from biometrics directly; and key-binding biometrics cryptosystem, which binds the biometrics template with a key within a cryptographic framework, e.g., fuzzy commitment[11] shown in Figure 12. The idea of fuzzy commitment scheme is using Error-Correcting Code (ECC) to tolerate the intrinsic fuzziness of biometrics signals and using

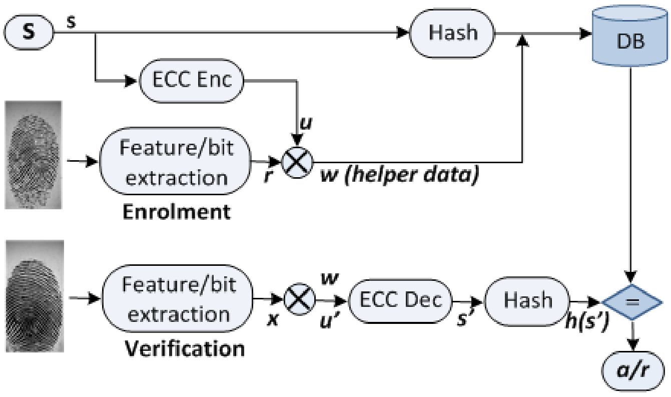


Figure 12: Fuzzy commitment scheme for biometrics template protection

exclusive-or to combine a biometrics 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 biometrics template with a password which is to be saved by the password manager.

## 2.6 Honeywords and Passwords Cracking Detection

Ordinarily, some websites will block the users account after we have several failed login attempts. Some attackers are using this kind of policy to get the user credentials by making the user account blocked and changing the new password. In order to address this issue,"Kamouflag" architecture was proposed in [19], where the honeywords comes at the first time. They defined the honey words as decoy passwords sets. Furthermore, Ari Juels and Roanld proposed a simple method to employ the honeywords In order to make the correct password cracking detectable[41] .

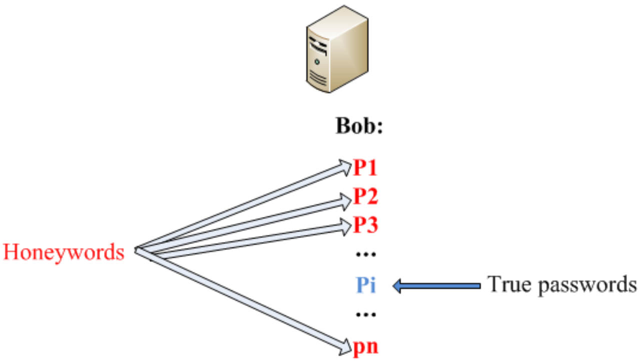


Figure 13: Honey words sample

Take a look at the Figure 13, the authentication system contains a set of passwords including honeywords and the true password. In one set of passwords, only one password is true for the user's identity authentication, and the others are called honeywords.

## 2.7 Client-side Storage

Client-side storage is also can be named as offline storage which is about capturing the resources the users are interested in or the specific data produced by the users [42]. The offline storage techniques can be divided into two groups. One is older offline storage technologies, including Cookies, Plugin based storage, and another one one is HTML5 featured storage, containing web storage, web SQL Database, IndexedDB.

### 2.7.1 Older offline technologies

**Cookies**

Cookies have been there since the web came. Nowadays, they are only used by the most applications to store the identifying information and the rest of the users data on the server, though they were intended to a little data related with user at the beginning. Cookies have an extremely limited storage capacity, as low as 20 cookies, and 4KB for each according to the specification [43]. In additional, cookies slow down the network speed due to that they come from and go to the server by the HTTP headers.

### 2.7.2 HTML5 featured offline storage

**(1) Web storage**

Web Storage includes two objects; first, one is called localStorage, which is a single key/value persistence object. And the second one is sessionStorage, which works the same way as the localStorage except that the storage clears when the window is closed.

The Web Storage is a real simple API and it is available in all major browsers. It is the most compatible storage in the current browsers. However, the data is not structures, there is no query language, schemes therefore it is scaled not very well when users have a large data set. Another defect is that it lacks the transactions, which means that it might race conditions and have the potential risk of data corruption.

**(2) Web SQL storage**

Web SQL storage is just as the name, an offline SQL database. It comes all features with the typical SQL relational database. It is a fast and rich sql implementation, we can do the basic SQL query such as select, insert, update and delete. Furthermore, we can do joins, inner selects, etc, those advanced operation. Besides, there is transaction's transportation and database is protected from race conditions which might arise in web storage.

Whereas, The Web SQL is not available in all web browsers, it is only supported in the Chrome and webkit-based browsers right now. Therefore, it is not supported in the Firefox and IE. Furthermore, it will be deprecated soon and will not be supported from all the browsers.

**(3) IndexedDB**

IndexDB is basically a simple flat-file database with hierarchical key/value persistence and basic indexing [44] [45]. It can be regarded as a compromise between Web storage and Web SQL database. Like the web storage, a simple key/value pair; and like Web SQL database, it is capable of being fast operated.

It has indices on the certain fields inside the stored data and we can make fast query with indices. Then we do not need to search through all of the data structure. In addition, it supports the transactions and the data is capable of being protected against the race conditions [46]. However, It is not available in all the new browsers.

### 2.7.3 Pros and cons

After reviewing the client-side storage technologies, in Table 2, we give a comparison about the pros and cons of them.

Table 2: Comparison of various client-side storage technologies

| | Pros | Cons |
|---|---|---|
| **Cookies** | • simple, easy to learn.<br>• extensively supported by all the browsers. | • limited storage capacity, 20 cookies and 4kb for each<br>• transferred inside HTTP headers, slow down the network. |
| **Web storage** | • supported by all modern browsers, as well as on mobile devices).<br>• simple and a synchronous API<br>• sost compatible. | • data is not structures, consistency and integrity might be an issue.<br>• without indices, complicated for large data storage.<br>• need to be serialized and deserialized String values.<br>• no transaction, might suffer race conditions. |
| **Web SQL database** | • structured data, no worry about the integrity and consistency<br>• an asynchronous API.<br>• data can be indexed according to search keys.<br>• robust, due to is it is a typical SQL model.<br>• support transactions, protect database against race conditions. | • deprecated. Will not be supported by the web browsers.<br>• requiring knowledge of relational databases and SQL. |

*Continued on next page*

| | | |
|---|---|---|
| **IndexedDB** | • an asynchronous API.<br>• data is indexed by the indices, fast query with index<br>• various versions supportable.<br>• transaction support, no worry about suffering from race conditions.<br>• simple data model, easy to learn. | • not widely supported for all the web browsers.<br>• complicated callbacks. |

## 2.8 Communication Technologies between Client and Server

Traditionally, the web design only allows the one-way communication in the client server communication model, which is client initialized communication. The database is hosted on the server and queried and updated by the client in a specific time interval. This original client server one-way communication technique is known as polling [47], which cause the browser refreshing and waiting for the update from the server. However, nobody is willing to wait or to do the refresh manually and most people expect to automatically acquire the update as soon as possible. In additional, the mobile web is becoming more and more prevalent but the critical limitation of the mobile devices is the battery capacity [48]. Therefore, if still utilizing poor polling technique to query and update the server, it inevitably consumes a lot of resources on the mobile devices.

How to get the automatic update? Commonly, a solution which is used to emulate the server to client communication is known as long polling [49]. In this approach, the client sends the request to server and server has no response unless there is update on it. As long as the client gets the response, it will send new request and wait for new response. Therefore, long polling only can be regarded as near pushing, because the new request makes some delays and the blackout period between sending and response.

With the advent of HTML5 technologies, HTML5 working group came up with two powerful pushing technologies, Server-Sent Events [50] and Web Sockets [51]. Server-sent Events allows delivery of data in real time uni-bidirectionally and the Web Sockets allows delivery of data in real time bidirectionally. Both of them have been designed for maximum efficiency, high scalability and to be compatible with current web infrastructures.

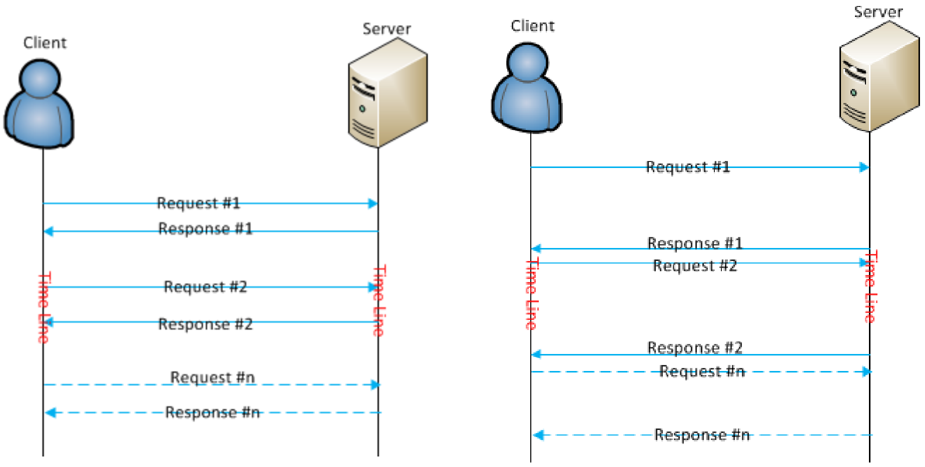
### 2.8.1 Polling

Polling is the most common and conventional communication way used in web world [52]. The illustration is shown in Figure 14(a), client sends requests to the server continually at a fixed time interval(for example 5 seconds) and receive immediately no matter the server has the update or not. This approach is obviously not efficient due to the response need to be generated even though there is no real content. In other word, the result of polling is usually a large amount of extra overhead, which leads to decreased overall network throughput [53]. Usually, the overhead can be decreased by increasing the polling requests interval. However, it increases the delay of the new server update to the client in turn.

### 2.8.2 Long polling

Due to the inefficient of the polling technique, long polling was introduced. In this approach, the server keeps the request open if the data is not available right now [54], which means that the response is allowed to be postponed until there is actual data available. The process is illustrated in Figure 14(b), Client sends the request to server and server does not send the response until there is new information available. If new data is received within the fixed time interval, the current request connection is closed, restarting the new request. Otherwise, if new update is not received within the time interval, the server will notify the client to close current request and open a new connection, which results in server respond with actual content constantly as soon as the new data is available.

While the long polling is more efficient than polling, it is still inefficient in terms of the overhead when the request is time out. If the request is time out, it requires a restarting the connection similar to the polling loop. In additional, during the request time interval, the client cannot initial another request to the server over the same HTTP connection [54]. For example, the Facebook would require one connection to acquire the live updates and another one to post changes/updates triggered by the users.



(a) Polling model

(b) Long polling model

Figure 14: Polling and long polling models

### 2.8.3 Pushing

#### (1) Server-Sent events

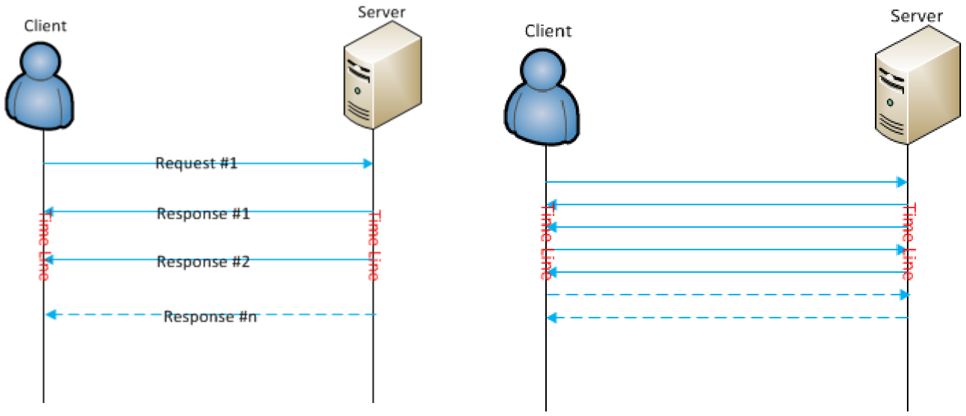
Server-Sent Events is originally developed by Opera [55] and it is suitable for the one-way communication from server to client. The Specification defines it as an API for opening an HTTP connection for receiving push notifications from a server in the form of DOM events. The API is designed such that it can be extended to work with other push notification schemes such as Push

SMS [54]. In other word, it (SSE) is a standard describing how servers can transmit towards clients once an initial client connection has been established and it is commonly used to send message updates or continuous data streams to a browser client and designed to enhance native, cross-browser streaming through Event Source [50]. As example is illustrated in Figure 15(a), the client requests a particular URL in order to receive an event stream and the connection between server and client will be kept persistently, and server will send an event to the client as long as there is new information available.

Server-Sent Events are promising for one-way server push applications because its simplicity, widespread compatibility, high scalability and automatic reconnection [54] .

**(2) Web sockets**

Compared with Server-Sent Events, Web Sockets provides a richer protocol to perform bidirectional and full-duplex communication between client and server [56]. It is very similar to TCP Sockets [57] in terms of both of them provides the persistent and duplex connection. Illustration in Figure 15(b) shows that how the Web Sockets works: A client requests server to open a connection. Once the connection is established, the client and server can send data with each other when new data on either side is available. However, in some scenarios new data does not need to be sent form the client to server, such as news feeds, stock tickers, *etc* or some other automated data pushing mechanism(for example updating a client side web SQL database or index DB which will be used in my master thesis). When the client data is needed to be sent to the server, XmlHtttpRequest(XHR) [58] is also a good choice. In additional, compared to Server-sent Events, Web sockets requires the full-duplex connections and a new Web Sockets server to handle the protocol and it lacks automatic reconnection and the capability to send arbitrary events [56].



(a) Server-sent events model

(b) Web sockets model

Figure 15: Server-Sent events and web sockets models

### 2.8.4 Summary

In Table 3, the differences between polling, long polling, server-sent Events and web sockets and the each feature of them are summarized.

Table 3: Comparison of different communication technologies

| | **Polling** | **Long polling** | **Server-Sent events** | **Web sockets** |
|---|---|---|---|---|
| **Timeliness** | not real time, fixed time interval | near real time, request delay | real time | true real time |
| **Browser support** | supported by the most browser | supported by the most browser | supported by Chrome 29+, Firefox 24+, Opera 17+, Safari 5.1+; not supported by IE [59] | supported by Chrome 29+, Firefox 24.0+, Opera 17+, Safari 5.1+, IE 10.0+ [60] |
| **Communica -tion features** | client to server one-way communication | client to server one way communication | server to client unidirectional communication | bidirectional communication |
| **Advantages** | mature<br>easy to be implemented | mature<br>easy to be implemented | real-time traffic from server to client, mostly what we need | real-time traffic from the server to the client and from the client to the server<br>capable of crossing domain [61] |
| **Disadvantages** | empty response creates overhead<br>inefficient | cannot initial another request to the server over the same HTTP connection | not all browser support (i.e. does not support)<br>cannot cross domain | requires a special protocol and server implementation<br>lack automatic reconnection, event IDs,<br>cannot send arbitrary events [56] |

# 3 System Design

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

- Biometrics is incorporated as the second factor into the password manag-er design, in addition to a master key.
- 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.
- 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[62].
- 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.
- 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.
- 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.

## 3.1 Global Structure of The System

As it is shown in the Figure 16, the whole system contains three main parts, a biometrics information capture/detection program, the NBLpass password manager part, the cloud database. The Biometrics information detection part is used to capture the user's biometrics characteristics as we described before, fingerprint, face, voice, Iris, etc., and then interpret the biometrics information to a machine-readable data, and then the data can be used for the NBLpass account registration and identity verification process. The NBLpass password manager part is an chrome extension, including two main components, one is local indexedDB where the records are stored offline. And another one is the interface where user can take options such as adding a new record, deleting an existing record, updating a record, uploading records to the cloud database.

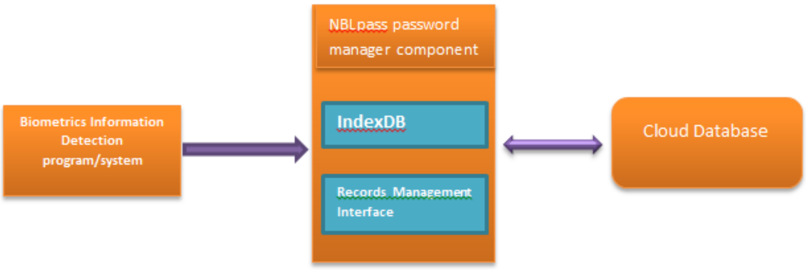

NBLpass password manager component
Biometrics Information Detection program/system
IndexDB
Records Management Interface
Cloud Database

Figure 16: Global structure of the system

The last part is the cloud database where all of the users' records are stored. Cloud database is used by users to back up the local records and synchronize the records between different devices.

## 3.2 Definition of Functionalities of The System

In this section, the functionalities of the NBLpass are described from user perspective and system itself perspective. In simple, the system we designed enables users to manage/generate passwords by using a biometric attribute and a plain text password for the system identity authentication. In details, users are capable of using this system to register a new account for a web service provider, auto fill login for for a website, generate a new password, store the passwords and share passwords between various devices.

From the system itself perspective, system can do the biometrics attributes capturing and interpret that kind of attribute to a machine readable data format, make the user's records more secure by combining the biometrics attribute and a plain text password to do the identity authentication, backup and share user's a variety of records.

## 3.3 Component Functionality Description

### 3.3.1 Password binding (PB)

Password binding is a process which generates a password vault $\texttt{Ws}$, $\texttt{Wp}$) by taking as input a master key $\texttt{Km}$, the biometric feature $\texttt{B}$, and the password $\texttt{PSW}$ to be saved by the password manager. The whole process is presented in Figure 17:

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

- Step 2: $\texttt{s}$ 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 $\overrightarrow{\texttt{BTP}}$) to generate a protected template $\texttt{PT}$;

- 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 Wp. The ECC adds some robustness to a PT, which is generated by those distance-preserving transformation algorithms such as the dynamic random projection and thus modulates some fuzziness inherent in B.

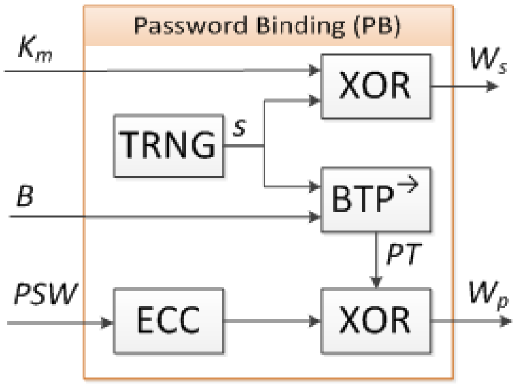


Figure 17: Password binding process

Note that after the password binding process, both the randomly generated number s and the generated protected template PT are discarded. The fact is that only Ws and Wp are saved in the password vault which requires that the user simultaneously give correct Km 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 18.

### 3.3.2 Password releasing (PR)

Password releasing is a process which releases a password PSW' from the password vault Ws , Wp by taking as input a master key K'm and the bio-metric feature B'. The whole process is presented in Figure 18:

- Step 1: A secrets' is recovered by exclusive-or (XOR) operation of K'm and Ws;
- Step 2: The recovered s' is used as an external parameter to the rigidly-irreversible biometric template protection method $\overrightarrow{\mathrm{BTP}}$ to generate a protected template PT';
- Step 3: A password PSW' is derived by firstly performing exclusive-or (XOR) operation of PT' and Wp, and secondly Error-Correction-Code decoding ($\mathrm{ECC}^{-1}$) the XOR result.

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 Km or the

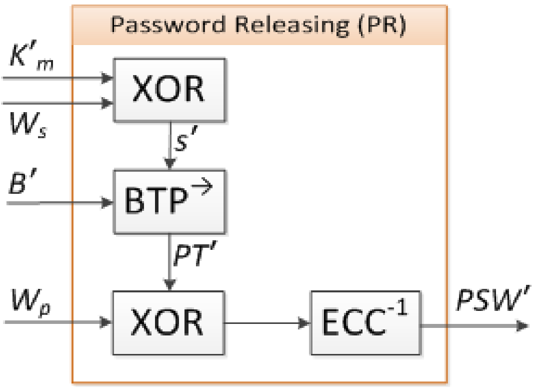


Figure 18: Password releasing process

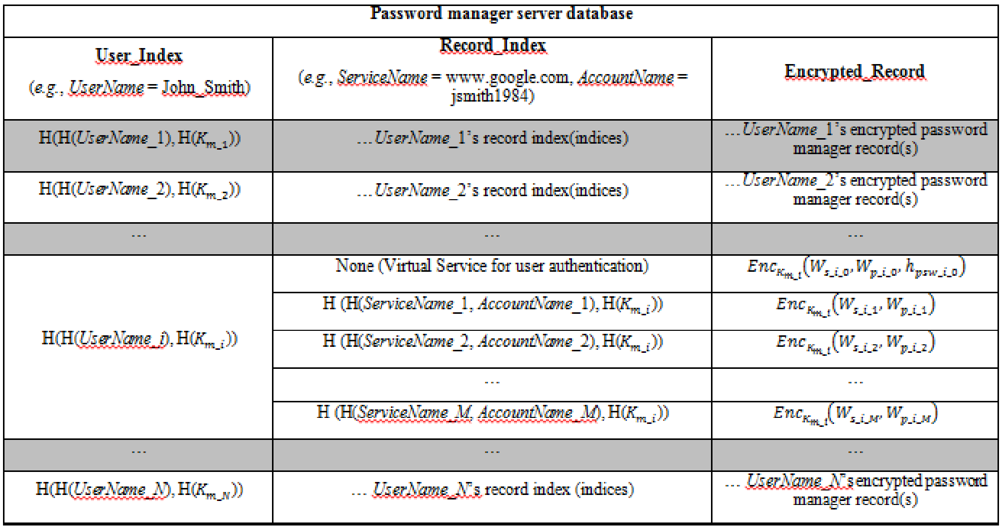
biometric input B separately).

### 3.3.3 Password manager maintenance

Figure 19 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 is 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 user name UserName_i and master key Km_i are required to generate an index by a mathematical hash function H(.), e.g., SHA-2/3: H(H(UserName_i),H(Km_i)), where H(Km_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(Km_i) to generate a record index H(H(ServiceName_t, AccountName_t),H(Km_i) used to locate the t-th record for user_i. The t-th record is encrypted by advanced standard encryption(aes) [63] algorithm, e.g., AES-256, into an encrypted record EncKm_i(Ws_i_t , Wp_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. 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 Figure 19) EncKm_i(Ws_i_t , Wp_i_t, hpsw_i_t) for user authentication is first used to verify



| **Password manager server database** | | |
|---|---|---|
| **User_Index**<br>(*e.g.*, *UserName* = John_Smith) | **Record_Index**<br>(*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) | $Enc_{K_{m\_i}}(W_{s\_i\_0}, W_{p\_i\_0}, h_{psw\_i\_0})$ |
| | H (H(*ServiceName*_1, *AccountName*_1), H($K_{m\_i}$)) | $Enc_{K_{m\_i}}(W_{s\_i\_1}, W_{p\_i\_1})$ |
| | H (H(*ServiceName*_2, *AccountName*_2), H($K_{m\_i}$)) | $Enc_{K_{m\_i}}(W_{s\_i\_2}, W_{p\_i\_2})$ |
| | ... | ... |
| | H (H(*ServiceName*_*M*, *AccountName*_*M*), H($K_{m\_i}$)) | $Enc_{K_{m\_i}}(W_{s\_i\_M}, W_{p\_i\_M})$ |
| ... | ... | ... |
| H(H(*UserName*_*N*), H($K_{m\_N}$)) | ... *UserName*_*N*'s record index (indices) | ... *UserName*_*N*'s encrypted password manager record(s) |

Figure 19: An example of data structure of the proposed password manager in the cloud server

if the hash value h'psw_i_t = H(PSW') of the released PSW' can match hpsw_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 Km and biometric input B as in Figure 17. 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 the client end. The synchronized data in the client database can be decrypted by the master key Km to obtain the password vault elements Ws and Wp.

## 3.4 Security Analysis

### 3.4.1 System level analysis

When it comes to passwords, security is always a sensitive and serious issue. Therefore, it is necessary to analyze the security models in this part. In this section, several situations are assumed from the attacker's perspective. We assume that attackers are capable of cracking the password, user name from the third party web service, capable of comprising the data we upload/download from cloud database, capable of attacking the cloud database and capable of acquiring NBLpass records from local IndexedDB. The analysis for each situation is given as follows.

**(1) Given user name and password are compromised from third party web service provider**

In fact, this is a very common problem in the web world. User name and password are compromised when users are attempting to log in the website or from the third party web service provider's database. It is also reasonable to imaging what will happen if this situation comes. However, under this situation NBLpass can do nothing but help user to generate a new password.

**(2) Given encrypted data are compromised during the transmission**

It is not surprised if this situation happen due to the fact that we only use hypertext transfer protocol secure protocol for the communication layer. Technically, according to the specification [64], it is not a protocol itself, it is a combination of HTTP(Hypertext Transfer Protocol) and SSL protocols. Even though, it is extensively adopted by many web service provider besides in our NBLpass password manager, which does not mean it is completely secure. In [65], the author proposes and analyzes the serious problems with https on the web. Therefore, the data might be compromised during the transmission from client to server or form server to client.

What will happen if the data are compromised during the synchronization process? Nothing will happen, although attackers get those data, they can do nothing. For the transmitted data, Advanced Encryption Standard (AES) [63] algorithm is utilized. Dive into AES algorithm, no matter to encrypt plaintext to ciphertext or to decrypt the ciphertext to plaintext, a cipher is needed. In our NBLpass model, we employ the NBLpass login password to be the secret key to that cipher. Thereby, even though the transmitted data are intercepted by the attacker, the original data cannot be cracked as long as they do not know the NBLpass authentication password.

**(3) Given cloud database is attacked**

Under this assumption, there is no threats come from the data compromising. As it is described in the previous subsection, the data are encrypted by using AES algorithms before transmitting to the cloud database. Therefore, the data in cloud database are still in AES algorithm encryption format. As long as the NBLpass login password is not acquired by the attacker, there could not be any threats. However,there might be some attackers are not intended to compromise the user information but make the database corrupted. Under this situation, the data can not be synchronized between client and server. The potential threat is that NBLpass might not work if there is no existing records in local indexedDB of the website where user are going to login .

**(4) Given attack in local IndexedDB**

Records in NBLpass local indexDB are encrypted by using AES algorithm as well. Therefore, referring to the analysis from the previous two subsections, as long as the NBLpass authentication password is not compromised, there will no threats from the records uncompromising in indexDB. As it is said, an attacker may make the indexeDB corrupted. When this occasion comes, then the NBLpass will not work.

### 3.4.2 Risk analysis from algorithm perspective

The security issues have been analyzed in terms of the system level. In this section, risk analysis is given from the algorithm perspective. The risk analysis mainly focuses on the elements used in the mathematical operation in the password binding and password releasing algorithm. Km, a plain text password used as one of the NBLpass system authentication factors; B, a kind of biometrics attribute used as another factor for NBLpass system authentication and used as a factor for PSW encryption. In addition, there are another two generated input elements, a random integer (s) and PT(an invertable biometrics template). Two generated outputs Ws and Wp through the NBLpass system.

**(1) Given any of the elements mentioned above is compromised**

From the proposed scheme, we can see that each of the elements mentioned above has a straightforward or mediate relationship with each other, thus once any one of them is compromised will not bring security problems for users. For example, Km and B are both the authentication factors for the NBLpass password manager, even though attackers intercept one of them, they will not login; Ws, Wp is the generated data, if Km and B is not know for attackers, there is no worry about the security.

**(2) Given any two of the elements mentioned above are compromised**

Under this situation, we need to apply the variable control method for the risk analysis. For instance, assume a precondition that the Km is compromised already, once the B is compromised as well, attacker will take control the NBLpass password manager, at least they can log in the third party website and change the user's passwords even though they do now know the old passwords; Once the integer s is guessed out by the attacker, there is no threat due to they do not know the B.

According to that kind of variable control analysis approach, assuming each of them is compromised, two of them, three of them and even all of them. Conclusion is that as long as Km and B are not compromised at the same time, the NBLpass system is secure and PSW can not be compromised from NBLpass password manager.

When use the same way to analysis the password releasing process. The conclusion is that as long as the B' and Km' are not compromised at the same moment, the NBLpass password management system is secure, and the PSW' can not be intercepted by the attacker.

### 3.4.3 Password manager security enhancement on the system level honey templates

In order to enhance the security level of the password manager, we introduce the honey words checker template from [41] for the master key(Km) and biometrics attribute(B) cracking check. While, it is not implemented in the system, only an security analysis. In the description, the Km is taken as the example, the same working mechanism for B cracking check. Note from Figure 20, when user input the master key(Km), the Km will be checked by the honey word checker; As it is shown in the figure, if Km is the true passwords, the password manager will continue the next authentication step; If Km is justified as a honey word, then the password manager will set off an alarm to user reminding that the password might be compromised by the adversary; If Km is not in the passwords set(neither a correct password nor honey words ), the password manager will

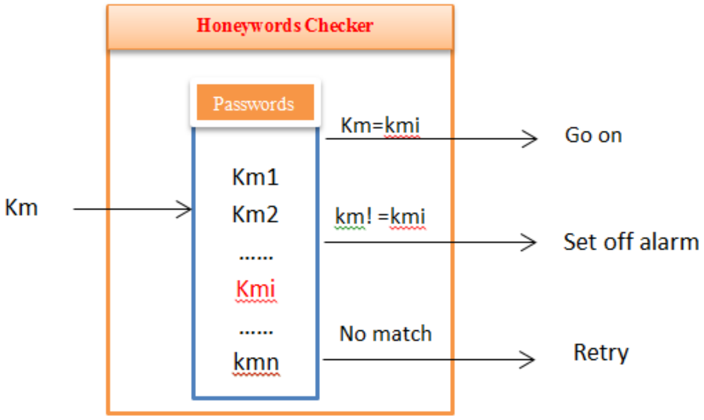


Figure 20: System level honey checker template

require user to type in the Km again.

# 4 System Implementation

## 4.1 Application Environment and Use Cases

First and the most important before we describe the usage scenario, the password manager account creation need to be explained. Taking a look at Figure 21, we name the password manager as NBLpass. For the new user, three areas need to be filled up in the new account registration interface/dialogue. User name could be the user email account; Password, a text plain passwords come form the users and Biometrics attributes are the subjects' biometrics features input, can be the features which were described in the Section 4, Chapter 2, i.e., Fingerprint, Iris, Face, Retina, Signature, etc.

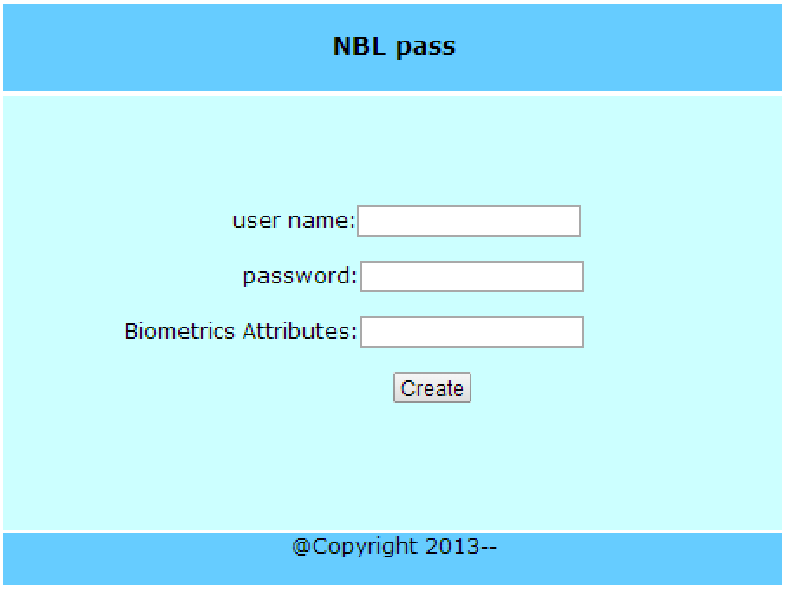


Figure 21: Password manager new user account registration

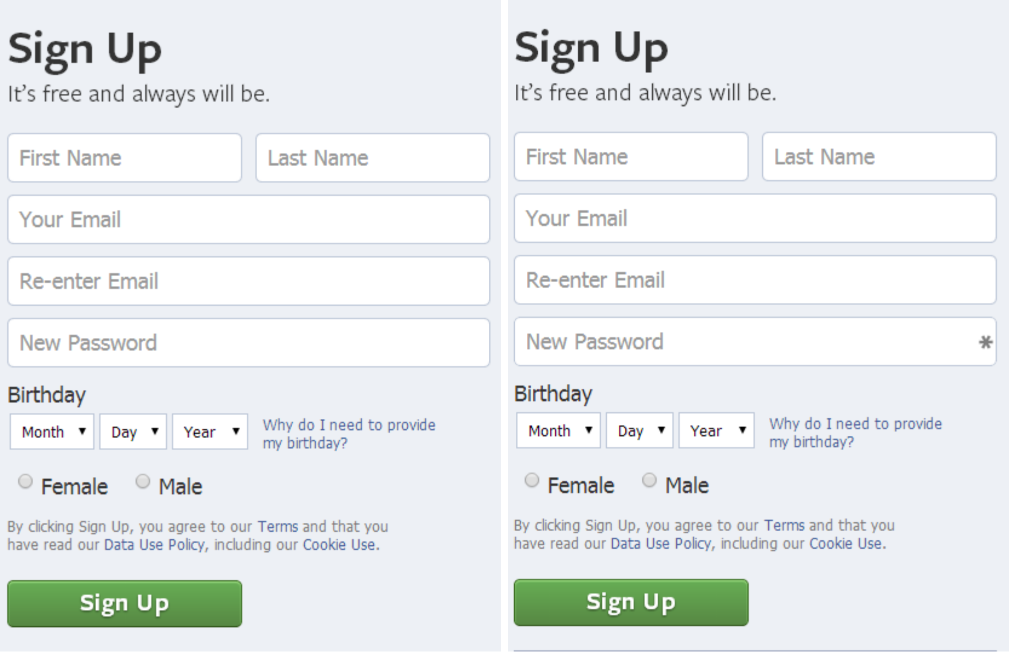
In this section, the password manager usage environments and scenarios are assumed. The environments can be classified as an public place such as laboratory, Internet cafes, airports and some other untrusted area. A relatively trusted environment such as home, office, etc. Application scenarios could be offline and online, when user has the network connection, they can creating new accounts for third-party web services, log in the third-party web service, and manage passwords. While, when the user is offline, only the passwords management can be available. Details are explained as follows.

### 4.1.1 Working online

Under this case, users might take and be allowed to take the following activities: firstly, users are going to register himself/herself to the third-party websites; secondly, the user might want to log in the third-party website and at the last, users are going to manage their passwords. Details are described as follows.

**(1) Creating new account for the third-party web service.**

This is a very common situation when we do not have the account for the website which we want to log in and take some activities on it. For instance, we want to find friends and share moments with friend in some social website where we have no user account. Therefore, before we have the login access we have to register ourself to the website. In order to elaborate the usage of our password manager we take face book registration as an example. Figure 22 show the difference between when users create new account with NBLpass and without NBLpass. As it is showed in Figure 22(b), users are using the NBLpass, there will be an dialogue come when user click the '*', where users have options to ask NBLpass to help them generate password or save the record to the password manager database.



Sign Up

It's free and always will be.

By clicking Sign Up, you agree to our Terms and that you have read our Data Use Policy, including our Cookie Use.

Sign Up

It's free and always will be.

By clicking Sign Up, you agree to our Terms and that you have read our Data Use Policy, including our Cookie Use.

(a) Facebook registration without NBLpas

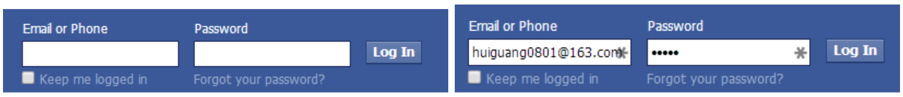
(b) Facebook registration with NBLpas

Figure 22: Facebook registration sample

**(2) Log in the third-party web service**

In this part, we assume users are going to perform the login process with NBLpass. As normally, for a website, if we want to log in, we need to type into the user name (usually it is an email address) and password. There are two possible situations, one is there exist the records in the

NBLpass for website which users are going to log in. In this situation, the users do not need to input the user name and password manually, NBLpass can help users fill up the login form automatically. Another situation is that there is no record in NBLpass for the website which users are going to log on. Under this situation, users get the option to save the passwords and user name to the NBLpass database when user click the '*' sign and then the dialogue will be triggered. In this part, facebook is taken as an example as well, Figure 23(a) shows normally login, and Figure 23(b) is employing the NBLpass to implement the login process.



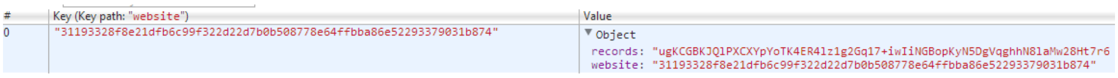
Email or Phone: huiguang0801@163.com
Password: •••••

(a) Facebook login without NBLpass (b) Facebook login with NBLpass

Figure 23: Facebook login sample

### (3) Managing passwords

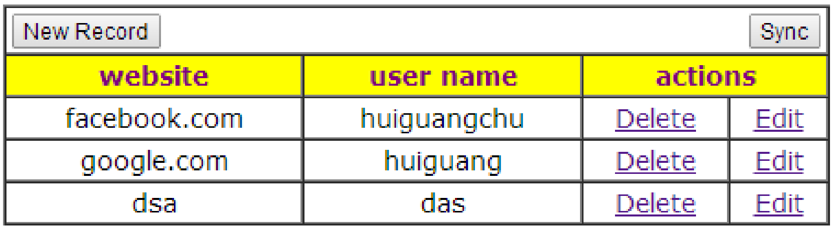
In this part, there two conditions. First situation is that there is no record in the NBLpass database. Under this case, users are required to authenticate her/him in order to download the records from the cloud database. Second situation is that there are already records in NBLpass local indexedDB database (see Figure 24).



```
#  Key (Key path: "website")                                                     Value
0  "31193328f8e21dfb6c99f322d22d7b0b508778e64ffbba86e52293379031b874"            ▼ Object
                                                                                   records: "ugKCGBKJQ1PXCXYpYoTK4ER41z1g2Gq17+iwIiNGBopKyN5DgVqghhN81aMw28Ht7r6
                                                                                   website: "31193328f8e21dfb6c99f322d22d7b0b508778e64ffbba86e52293379031b874"
```

Figure 24: Records in local IndexDB example

As shown in Figure 25, the data table includes two columns website, user name can indicate the user the specific website and the relative record. It enable user to take adding new record action, deleting existed records action, changing existed records action. In addition, the NBLpass can be used in different devices as long as there is Chrome browser. As long as the update happens in the cloud database, then the local data records in IndexDB can be updated automatically.



| website | user name | actions | |
|---|---|---|---|
| facebook.com | huiguangchu | Delete | Edit |
| google.com | huiguang | Delete | Edit |
| dsa | das | Delete | Edit |

Figure 25: NBLpass records management interface

### 4.1.2 Working offline

In this case, only the passwords management is available. Preconditions are that there are existed passwords in the NBLpass local database and there are the records in the IndexedDB where the users' records are stored locally in the browser. Under the offline case, the users are only allowed to add new record, deleting existed records, changing existed records. When the users have network connection, the records can be synchronized to the cloud database via HTTPS communication.

## 4.2 Implementation

As aforementioned, the complete system are composed by three parts, biometrics attributes acquisition parts, NBLpass password manages which is the main focus in the system design and implementation, and the cloud database. This section introduces the system in a product development way, including development environment, developing tools, and diving into the details of each part development.

### 4.2.1 Development Environment

**(1) Hardware**

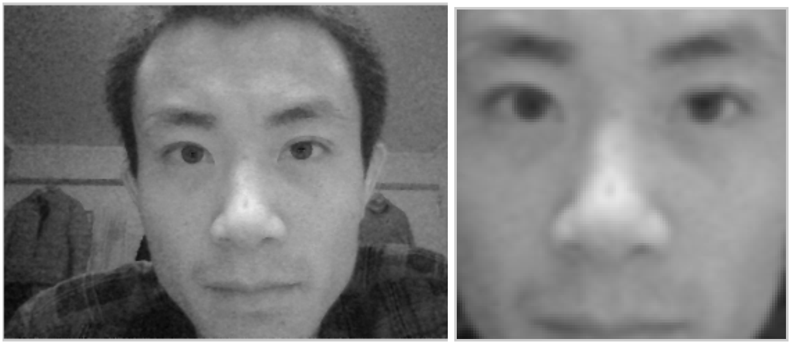
- Laptop Thinkpad s431
- Laptop Integrated Camera

**(2) Software**

- MATLAB, a developing platform used for face detection.
- Eclipse, a developing platform used for service development.
- Chrome, used for system testing and debugging.
- Tomcat, used as server.
- MySQL, cloud side data storage.

### 4.2.2 Biometrics characteristic acquisition Program

This is a MATLAB program to acquire the subject's facial attribute and interpret the data to a binary string which can be read and processed by the NBLpass password manager part. In practical, there is no requirement for the operation system as long as there is MATLAB installed and the program can be compiled successfully. Diving into this program, when the program is compiled successfully then the camera will be invoked and the face will be captured and processed with a special algorithm which is from NISLab. The capture result of normal face sample and processed face sample are shown in Figure 26. The data of the processed face picture can be obtained from the MATLAB program and it is transformed to be an NBLpass password manager readable binary string.



(a) Normal face sample (b) Processed face sample

Figure 26: Normal face sample and processed face sample

**PCA face recognition**

When diving into the face recognition program, principle component analysis (PCA) [66] [67] is needed to be introduced due to the fact is that it plays an important role in the face recognition process. PCA is a very useful statistical technique for finding pattern in high dimensional data, expressing the data by highlighting the differences and similarities in much reduced data dimensions, and it has been widely used in the face recognition and image compression [66]. Details about the PCA technique in the program are described as follows:

- Creating the data set.
  We have 50 images in the training data set. Those images come from 10 subjects and 5 images for each subject. And each of them are 120 pixels high by 120 pixels wide.
- Creating image vector for each image.
  Each of the 120 pixels by 120 pixels images can be expressed as a $120^2$ dimensional vector.
- Composite image-matrix.
  We put all the images in one big image matrix which is a starting point of PCA.
- Measure the difference and similarities.
  The program measures the difference between the new face image with the original images in the 50 images set. We need to note that it is the new axes derived from the PCA rather than the original axes. The new axes works much better in term of the user's face recognition since PCA has given the original images in terms of the difference and similarities between them [66].

### 4.2.3 NBLpass password manager

It is developed as a chrome extension and it is the main focus of the thesis. In this part, there are several web pages including backend are designed. For example, login page which is shown in Figure 27, new user registration page(Figure 15), records management page(Figure 19 ), etc. The proposed scheme is interpreted to the javascript code and compiled as a chrome extension application. In this part, various algorithms are utilized such as AES [63] which used to encrpted each record. Error Code Correction (ECC) which is used to detect the specific number of errors of the biometrics attribute and correct them.

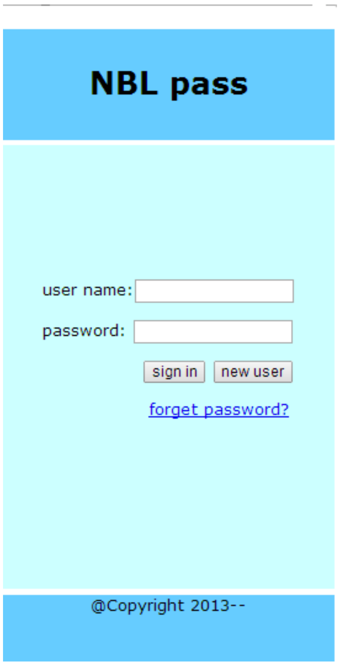


Figure 27: NBLpass login interface

### 4.2.4 Cloud database

In this part, a service needs to be provided for mutual communication between NBLpass password manager and cloud database. The developing language is in Java and Jersy [68] framework is used to implement a RESTful web service [69]. When it comes to the cloud database structure, it was shown in Figure 23, with each user name hashed and each record encrypted.

# 5 Further Work

In the future, the NBLpass password management system is aimed to be a generic biometrics based ID authentication platform. Therefore, the biometrics acquisition part is needed to be redesigned and redeveloped. At least, it can be integrated into the NBLpass password manager or an independent web application which can be invoked by the NBLpass password manager. The cloud database needs to be redesigned as well, because there is only one table is used for all the users in our testing and debugging phase. If it is aiming at a widely used system, each user should have one separate table.

# 6 Conclusion

In principle, if people have a password manager, people do not need to remember passwords anymore and the password manager can help users manage them. Compared with the conventional password manager installed on PC, the token-based biometric password managers described in the introduction part have no location limitation and promote the security and privacy. However, the token-based password manager might be lost more easily. Cloud based password manager in this thesis just provides a service, which is not only has no location limitation but also has no possibility to get lost since cloud services can penetrate every corner in the digital era. In addition, the privacy-preserved biometrics (using rigidly-irreversible template protection method) and password leakage detection(honeywords) based password manager scheme proposed in this thesis enables people to use two factors (a master key and the biometrics) for password binding and releasing in a secure way and enables people to know whether his or her passwords have been cracked or not.

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 and the passwords cracking detection take 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.

# 7 Paper Published During the Master Degree Program Period

Bian Yang, Huiguang Chu, Guoqiang Li, Slobodan Petrovic, Christoph Busch. *Cloud Password Manager Using Privacy-Preserved Biometrics*. IC2E 2014 : 2014 IEEE International Conference on Cloud Engineering, Boston, MA,USA.

# Bibliography

[1] Manage your website passwords, google chrome help. `https://support.google.com/chrome/answer/95606`. "[Online; Visited 12-December-2013]".

[2] . Detailed information about the security of keepass, keepass help center. `http://keepass.info/help/base/security.html`. "[Online; Visited 12-December-2013]".

[3] LastPass. . The last password you have to remember. `https://lastpass.com/`. "[Online; Visited 12-December-2013]".

[4] Knight, S. February 20 2013. myidkey biometric password manager seeks kickstarter funding. `http://www.techspot.com/news/51697-myidkey-biometric-password-manager-seeks-kickstarter-funding.html`. "[Online; Visited 12-December-2013]".

[5] Software, B. S. S. S.-O. Enterprise password management and network security software with seamless interface to active directory. `http://www.m2sys.com/EBS.htm`. "[Online; Visited 12-December-2013]".

[6] Security, B. Apc touch biometric pod password manager. `http://www.apc.com/resource/include/techspec_index.cfm?base_sku=BIOPOD`. "[Online; Visited 12-December-2013]".

[7] Two-factor authentication, password manager pro. `http://www.manageengine.com/products/passwordmanagerpro/two-factor-authentication.html`. "[Online; Visited 12-December-2013]".

[8] Gaw, S. & Felten, E. W. 2006. Password management strategies for online accounts. In *Proceedings of the second symposium on Usable privacy and security*, 44–55. ACM.

[9] Vitabile, S., Pilato, G., Conti, V., Gioè, G., & Sorbello, F. 2004. Biometric features for mobile agents ownership. *IPSI, Trans. on Internet Research, Issues in Computer Science and Engineering*, 1(1), 81–89.

[10] Sutcu, Y., Sencar, H. T., & Memon, N. 2005. A secure biometric authentication scheme based on robust hashing. In *Proceedings of the 7th workshop on Multimedia and security*, 111–116. ACM.

[11] Juels, A. & Wattenberg, M. 1999. A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security*, 28–36. ACM.

[12] Khan, W. Z., Aalsalem, M. Y., & Xiang, Y. 2011. A graphical password based system for small mobile devices. *arXiv preprint arXiv:1110.3844*.

[13] Janssen, C. Web-based application. http://www.techopedia.com/definition/26002/web-based-application. "[Online; Visited 27-May-2014]".

[14] Ibns: Web authentication deployment and configuration guide, cisco white paper. http://www.cisco.com/c/en/us/products/collateral/ios-nx-os-software/identity-based-networking-services/app_note_c27-577494.html. "[Online; Visited 27-May-2014]".

[15] Larry Simmons, D. M. The disadvantages of building a non-web-based application. http://smallbusiness.chron.com/disadvantages-building-nonwebbased-application-40347.html. "[Online; Visited 27-May-2014]".

[16] Lamport, L. 1981. Password authentication with insecure communication. *Communications of the ACM*, 24(11), 770–772.

[17] Suo, X., Zhu, Y., & Owen, G. S. 2005. Graphical passwords: A survey. In *Computer Security Applications Conference, 21st Annual*, 10–pp. IEEE.

[18] Elftmann, P. 2006. Secure alternatives to password-based authentication mechanisms. *Lab. for Dependable Distributed Systems, RWTH Aachen Univ*.

[19] S.K., S., R.L., P., & Kumar, A. 10.2012. Graphical password authentication scheme based on color image gallery. *International Journal of Engineering and Innovative Technology (IJEIT)*, 2(4).

[20] Jansen, W. A. 2003. *Picture password: a visual login technique for mobile devices*. US Department of Commerce, National Institute of Standards and Technology.

[21] October 2013. Two factor authentication for the enterprise. http://www.realuser.com. "[Online; Visited 10-May-2014]".

[22] Sarohi, H. K. & Khan, F. U. 2013. Graphical password authentication schemes: Current status and key issues.

[23] Panjwani, Saurabh, Naldurg, Prasad, Bhaskar, & Raghav. Analysis of two token-based authentication schemes for mobile banking. Technical report, Microsoft Research Technical Report, 2010.

[24] Man in the middle attack. http://en.wikipedia.org/wiki/Man-in-the-middle_attack. "[Online; Visited Macrh 2014]".

[25] Tanvi, P., Sonal, G., & Kumar, S. M. 2011. Token based authentication using mobile phone. In *Communication Systems and Network Technologies (CSNT), 2011 International Conference on*, 85–88. IEEE.

[26] Pippal, R. S., Gupta, P., & Singh, R. June 2013. Article: A novel smart card authentication scheme using image encryption. *International Journal of Computer Applications*, 72(9), 8–14. Published by Foundation of Computer Science, New York, USA.

[27] Peng, D., Li, C., & Huo, H. 8-11 Aug 2009. An extended usernametoken-based approach for rest-style web service security authentication. *International Journal of Computer Applications*, 582–586.

[28] Song, R. 2010. Advanced smart card based password authentication protocol. *Computer Standards & Interfaces*, 32(5), 321–325.

[29] 2008. Authentication. `http://www.cs.bham.ac.uk/~mdr/teaching/modules/security/lectures/biometric.html`. "[Online; Visited April 2014]".

[30] Weigold, T., Kramp, T., & Baentsch, M. 2008. Remote client authentication. *Security & Privacy, IEEE*, 6(4), 36–43.

[31] Al-Hussain, A. & Al-Rassan, I. 2010. A biometric-based authentication system for web services mobile user. In *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, 447–452. ACM.

[32] Bhargavan, K. & Delignat-Lavaud, A. 2012. Web-based attacks on host-proof encrypted storage. In *WOOT*, 97–104.

[33] Biryukov, A. & Khovratovich, D. 2009. Related-key cryptanalysis of the full aes-192 and aes-256. In *Advances in Cryptology–ASIACRYPT 2009*, 1–18. Springer.

[34] De Clercq, J. 2002. Single sign-on architectures. In *Infrastructure Security*, 40–58. Springer.

[35] Tsyrklevich, E. & Tsyrklevich, V. 2007. Single sign-on for the internet: a security story. *BalckHat USA*.

[36] Wang, R., Chen, S., & Wang, X. 2012. Signing me onto your accounts through facebook and google: a traffic-guided security study of commercially deployed single-sign-on web services. In *Security and Privacy (SP), 2012 IEEE Symposium on*, 365–379. IEEE.

[37] Yeung, B. Biometrics. `http://www.tns.com/biometrics.asp`. "[Online; Visited 10-May-2014]".

[38] Wayman, J., Jain, A., Maltoni, D., & Maio, D. 2005. An introduction to biometric authentication systems. In *Biometric Systems*, 1–20. Springer.

[39] Zimmerman, M. 2002. Biometrics and user authentication. *SANS Institute InfoSec Reading Room*.

[40] Anil K, J., Karthik, N., Abhishek, N., et al. 2008. Biometric template security. *EURASIP Journal on Advances in Signal Processing*, 2008.

[41] Juels, A. & Rivest, R. L. 2013. Honeywords: Making password-cracking detectable. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 145–160. ACM.

[42] Mahemoff, M. October 29th 2013. "offline": What does it mean and why should i care? http://www.html5rocks.com/en/tutorials/offline/whats-offline/. "[Online; Visited 10-May-2014]".

[43] Kristol, D. M. February 1997. Http state management mechanism. http://www.ietf.org/rfc/rfc2109.txt. "[Online; Visited 10-May-2014]".

[44] 05.18 2011. Html5 storage wars - localstorage vs. indexeddb vs. web sql, christian simms's weblog. http://csimms.botonomy.com/2011/05/html5-storage-wars-localstorage-vs-indexeddb-vs-web-sql.html. "[Online; Visited 10-May-2014]".

[45] Pilgrim, M. Dive into html5: No.7 the past, present & future of local storage for web applications. http://diveintohtml5.info/storage.html. "[Online; Visited 10-May-2014]".

[46] Calvert, C. April 27 2010. What is a race condition? http://codingclues.eu/2008/preventing-database-race-conditions/. "[Online; Visited 28-May-2014]".

[47] Sharma, N. May 2013. Push technology–long polling. *International Journal of Computer Science and Management Research*.

[48] Bajad, R. A., Srivastava, M., & Sinha, A. 2012. Survey on mobile cloud computing. *International Journal of Engineering Sciences & Emerging Technologies*, 1(2), 8–19.

[49] Pautasso, C. & Wilde, E. 2011. Push-enabling restful business processes. In *Service-Oriented Computing*, 32–46. Springer.

[50] W3C. 11 December 2012. Server-sent events. http://www.w3.org/TR/eventsource/. "[Online; Visited 12-December-2013]".

[51] W3C. 20 September 2012. The websocket api. http://www.w3.org/TR/websockets/. "[Online; Visited 12-December-2013]".

[52] Boon, M., Van der Mei, R., & Winands, E. 2011. Applications of polling systems. *Surveys in Operations Research and Management Science*, 16(2), 67–82.

[53] Lubbers, P. & Greco, F. 2010. Html5 web sockets: A quantum leap in scalability for the web. *SOA World Magazine*.

[54] Estep, E. Mobile html5:efficiency and performance ofwebsockets and server-sent events. Master's thesis, School of Science Double Degree Programme NordSecMob, Aalto University, June 28, 2013.

[55] Bersvendsen, A. September 1 2006. Event streaming to web browsers. http://dev.opera.com/articles/view/labs-event-streaming-to-web-browsers/. "[Online; Visited 12-December-2013]".

[56] Bidelman, E. November 30th 2010. Stream updates with server-sent events. http://www.html5rocks.com/en/tutorials/eventsource/basics/. "[Online; Visited 12-December-2013]".

[57] Sinha, S. November 30th 11/98. A tcp tutorial. http://www.ssfnet.org/Exchange/tcp/tcpTutorialNotes.html. "[Online; Visited 12-December-2013]".

[58] W3C. 6 December 2012. Xmlhttprequest, w3c working draft. http://www.w3.org/TR/XMLHttpRequest/. "[Online; Visited 12-December-2013]".

[59] Server-sent dom events - candidate recommendation, html5 - can i use. http://caniuse.com/#feat=events. "[Online; Visited 12-December-2013]".

[60] Web sockets - candidate recommendation, html5 - can i use. http://caniuse.com/#feat=websockets. "[Online; Visited 12-December-2013]".

[61] Oct 12 2012. What are long-polling, websockets, server-sent events (sse) and comet?, stack overflow. http://stackoverflow.com/questions/11077857/what-are-long-polling-websockets-server-sent-events-sse-and-comet. "[Online; Visited 12-December-2013]".

[62] Yang, B., Hartung, D., Simoens, K., & Busch, C. 2010. Dynamic random projection for biometric template protection. In *Biometrics: Theory Applications and Systems (BTAS), 2010 Fourth IEEE International Conference on*, 1–7. IEEE.

[63] November 26 2001. Advanced encryption standard (aes), national institute of standards and technology (nist). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf. "[Online; Visited 12-May-2014]".

[64] May 2000. Http over tls, eric rescorla rtfm, inc. http://tools.ietf.org/html/rfc2818. "[Online; Visited 12-May-2014]".

[65] Hoffman, C. 02.13 2014. 5 serious problems with https and ssl security on the web. http://www.howtogeek.com/182425/5-serious-problems-with-https-and-ssl-security-on-the-web/. "[Online; Visited 12-May-2014]".

[66] Smith, L. I. 2002. A tutorial on principal components analysis. *Cornell University, USA*, 51, 52.

[67] Wood, F. 2009. Principal component analysis.

[68] 2014. Restful web services in java. https://jersey.java.net/. "[Online; Visited 28-May-2014]".

[69] Tyagi, S. August 2006. Restful web services. http://www.oracle.com/technetwork/articles/javase/restful-142517.html. "[Online; Visited 28-May-2014]".