RESEARCH ARTICLE

A digital memories based user authentication scheme with privacy preservation

JunLiang Liu¹, Qiuyun Lyu^{1,2*}, Qiuhua Wang¹, Xiangxiang Yu¹

1 School of Cyberspace Security, Hangzhou Dianzi University, Hangzhou, ZheJiang, China, **2** School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, ZheJiang, China

* laqyzj@hdu.edu.cn

Abstract

The traditional username/password or PIN based authentication scheme, which still remains the most popular form of authentication, has been proved insecure, unmemorable and vulnerable to guessing, dictionary attack, key-logger, shoulder-surfing and social engineering. Based on this, a large number of new alternative methods have recently been proposed. However, most of them rely on users being able to accurately recall complex and unmemorable information or using extra hardware (such as a USB Key), which makes authentication more difficult and confusing. In this paper, we propose a Digital Memories based user authentication scheme adopting homomorphic encryption and a public key encryption design which can protect users' privacy effectively, prevent tracking and provide multi-level security in an Internet & IoT environment. Also, we prove the superior reliability and security of our scheme compared to other schemes and present a performance analysis and promising evaluation results.

OPEN ACCESS

Citation: Liu J, Lyu Q, Wang Q, Yu X (2017) A digital memories based user authentication scheme with privacy preservation. PLoS ONE 12 (11): e0186925. <https://doi.org/10.1371/journal.pone.0186925>

Editor: Kim-Kwang Raymond Choo, University of Texas at San Antonio, UNITED STATES

Received: March 9, 2017

Accepted: October 10, 2017

Published: November 30, 2017

Copyright: © 2017 Liu et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: This work was supported by The 2017 Discipline Construction of Zhejiang Provincial Key University—Cyberspace Security (No. GK178800225005), Natural Science Foundation of China (No. 61401128), Zhejiang Province Natural Science Foundation (No. LQ14F020010), Project of Zhejiang Provincial Key Enterprises Institute Construction and Project of Zhejiang Provincial Smart City regional synergy innovation center. The

Introduction

With the rapid growth of the Internet and the Internet of Things (IoT), a great number of personalized services are embedded in our daily lives, with many and increasing interactions with our personal devices. Unfortunately, many of the existing authentication mechanisms are considered weak, insecure or outdated, and these problems will continue to hamper the development of cyber space until a new, alternative method is developed.

Traditional knowledge-factor based authentication methods, for example, the text password, are based on letters, characters and numbers, which have long been considered insecure and vulnerable to guessing, dictionary attack, key-logger, shoulder-surfing and social engineering [1–5]. The fact users tend to choose very easy memorable passwords means they are often easily guessed. Worse still, a user has 26 online accounts on average [6,7], with each account demanding the user to set a password for logging in, so it's really unsurprising that they would use the same password for all these online accounts, even some important online accounts, such as for online banking. Thus, possession-factor based authentication methods, such as ID cards [8], smartphones or one-time password (OTP) generators are used as an additional authentication scheme in some important online account authentications [9]. The

funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

disadvantage of these schemes is that users have to carry these devices all the time, and some of this hardware is difficult or complex to use for some users. They are also vulnerable to man-in-the-middle attacks [2,3,5,9].

Considering the weakness of knowledge-factor based authentication methods and the complexity of possession-factor based authentication methods, biometric-factor based authentication methods have been introduced. These schemes rely upon unique inheritance factors such as finger prints, iris etc., which are unchanging during the lifetime of a human to achieve more security than the traditional schemes. But the major problem of adopting biometric-factor based authentication schemes is the high cost of additional devices needed for the identification process [10,11,12,13].

Graphical passwords have been a popular topic, and many practical ideas have been developed, like Passfaces™ [14] and Passpoint™ [15,16]. These systems require users to choose photos provided by the system as an authentication mechanisms. Nowadays, researchers have found that compared with static photos, it is easier for Users to recall their passwords by using their digital memories [7,17], and with the increasing number of devices within the Internet and IoT, Users' digital memories are more rich, colorful and available to be used as authentication mechanisms.

Based on this, Shone et al. [18] proposed a scheme named *Digital Memories Based Mobile User Authentication for IoT*, which required users to upload their digital memories and use them as authentication materials in IoT environments. However, there are imperfections in their scheme which is vulnerable to privacy disclosure attack and user tracking attack. In this paper, we enhanced Shone et al's [18] scheme to provide privacy preservation, anti-tracking and a multi-level security scheme for easing user authentication. In our scheme, we adopt homomorphic encryption to protect users' digital memories as well as users' privacy effectively, and apply public key encryption to improve data transmission security.

The rest of the paper is organized as follows: Section 2 provides background information and existing research on authentication and digital memories, and an analysis of Shone et al's [15] scheme. We propose our scheme in detail in Section 3. In Section 4 we test the security of our protocol and present our evaluation result. In section 5 we present the potential applications of our scheme and analyze performance. We draw conclusions in Section 6.

Related works

Digital memories

Human digital memory was first conceptualized in 1945 by Bush in his Memex system [18]. He described a device which was a place for storing books, records and communications and was envisioned as an "enlarged intimate supplement to memory" [19]. Nowadays, the rapid development of smart and wearable devices has made this idea a reality, and has paved the way for 'life-logging'. Users can use such devices to record aspects of their lives in digital form [20,21], such as the photos taken with their smartphone. These digital memories presents a highly recognizable identity to specific individuals and are much easier to remember than traditional passwords, so they have the potential to be used for authentication.

Recognition-based and Recall-based authentication

Recognition-based and Recall-based authentication stem from graphical passwords [3,22], and they are still effective when we apply them to digital memories, including video, audio, text, etc.

In recognition-based systems, a group of digital memories is displayed to the user and an accepted authentication requires correct digital memory objects (video, image, etc.) to be

clicked or touched in a particular order. An example of a recognition-based system is Passfaces™ developed by Real User Corporation [14]. In Passfaces™ the user is asked to choose four or more images of human faces from a face database provided by the system to use as their future password. In the authentication stage, the user sees a grid of nine faces, including one face chosen by the user and eight unfamiliar faces. The user recognizes and clicks on the familiar face. This process is repeated several times. The user is authenticated if he or she identifies the four faces they have previously chosen.

In recall-based systems, the user is asked to reproduce digital memories that he or she created or selected earlier during the registration phase. A typical recall-based authentication scheme is Passpoint™ [15,16]. In Passpoint™, the image can be an arbitrary photograph or paintings provided by the system with many clickable regions. The user is required to click on some predefined positions on the image in a particular order to be authenticated.

Analysis of Shone et al's scheme

Shone et al. [18] proposed a digital memories based user authentication scheme for mobile devices. There are three main actors in their scheme, including the user's smartphone (US), service provider (SP) and the digital memory authentication service (DMAS). The US serves as an independent platform allowing communications with other actors. As this is a mobile user authentication scheme, the physical presence of the US with the user also becomes an authentication factor. The SP provides a particular service to the user where the user must register their device with the SP. And the DMAS hosts the user's digital memories in a cloud environment and provides part of the authentication using their memory data. An overview of their scheme is shown in Fig 1.

Shone et al's. [18] scheme follows 6 steps, shown in Fig 1. Symmetric encryption is used to secure communications between the actors, as they all have pre-existing relationships.

Step 1: The user requests access to the SP's service, where the user's device has been registered. The hardware address of the US is first authenticated according to the SP's records.

Fig 1. Authentication mechanism overview of Shone et al's scheme.

<https://doi.org/10.1371/journal.pone.0186925.g001>

Step 2: Once the US is verified, the SP negotiates an authentication session with the DMAS for the user, specifying the comprehensiveness (e.g. number of questions) complexity (e.g. difficulty of the questions) of the desired authentication challenge.

Step 3: Upon the user requesting the challenge, it is computed by the DMAS to match the SP's requirements and is then sent to the user along with a session key (sKey). At this stage, all potentially useful meta-data has been stripped from any digital memory data used. The session key is used as a marker to prevent unauthorized replay attacks, in place of using nonces (arbitrary numbers that may only be used once). Each sKey is time-stamped, has a limited lifespan and is tied to the initiating IP address. A challenge/response mechanism is then used to authenticate the US against the DMAS. These challenges are computed by the DMAS using the user's digital memories and the SP's requirements.

Step 4: The user sends his response to the challenge back to the DMAS, and the DMAS validates the answers.

Step 5: The DMAS returns the authentication challenge result to the SP.

Step 6: If the authentication result is successful, the user will be granted access by the SP. And the user and the SP share the same sKey.

This scheme will provide challenges of different complexity in the authentication according to the required verification level; if the level is normal, the DMAS will give a challenge involving arranging several images in an ascending chronological order. As online banking accounts are considered high risk, this would require authentication challenges that match the higher complexity and comprehensiveness expected. For example, the DMAS may give a photo and ask about some details related to it, such as the device used to take the image and the year or month the photo was taken etc.

Although Shone et al's.[18] scheme made the authentication process easy for users as well as hard to attack with intelligence (challenges need user recognition or recall) and uncertainty (each authentication doesn't have the same answer), there exist some weakness or disadvantages. The main weakness or disadvantages of Shone et al's.[18] scheme are as follows:

- Digital memory objects stored in the DMAS without encryption will introduce potential privacy disclosure risk

The DMAS stores users' digital memories in a plaintext way; if the DMAS was attacked, there are potential privacy disclosure issues. For example, devices embed photo EXIF information when a user takes photos, and EXIF includes nearly all the details of the photo which are used by Shone et al.[18]'s scheme and are usually privacy-related. Worse still, if the DMAS is semi-trust (rigidly follows the protocol but curious) itself, it would misuse users' digital memories and may cause property damage to users.

- User's devID as the key factor of authentication stored in the SP will bring a single point failure attack and a user tracking attacks.

The SP doesn't provide a service to the user unless the user registers his device (devID) to it. That means the user's devID will be used and stored in each SP's database, which means that if any one of the SPs was attacked, the user's devID can be stolen, forged or misused. And also since the user's devID isn't changed, tracking attacks can be launched.

- Authentication process with whole symmetric encryption will result in man-in-the-middle attack. High database load on SP.

In Shone et al's.[18] scheme, they used symmetric encryption to secure communications among the actors, which means before the authentication stage the keys need to be transmitted among the actors, a situation which presents potential risks. This is especially vulnerable to

man-in-the-middle attack. Moreover, it increases the load on the actors' Web servers (the SP and the DMAS) since the servers need to receive, store and transmit the keys for each user when authenticating, which may reduce efficiency and cause a huge waste of network resources.

Proposed scheme

There are also three main actors in our scheme: The user, the service provider and the DMAS. The authentication process follows four phases: Phase 1: The SP registers to the DMAS; Phase 2: The user registers to the DMAS; Phase 3: The user logs in to the SP; Phase 4: The user logs in to the DMAS to maintain digital memory objects.

For clarity of explanation, notations adopted in this study are summarized in Table 1.

The SP registers to the DMAS

We can describe the process with the following formulas[23]:

$$SP \rightarrow DMAS : \{Application\}$$

$$DMAS \rightarrow SP : \{K_d, PID\}$$

As illustrated in Fig 2, the SP applies to the DMAS for cooperation, and the form of the application can be legal or authentic according to scenarios. The DMAS checks the strictness degree and anonymous degree the SP needs, then gives the SP an exclusive partner ID and distributes its public key K_d . This partner ID will be used when the SP requires authentication for the user. $Ans' = (c_1, c_2)(c_1, c_3) = (c_1 \bullet c_1, c_2 \bullet c_3) = (g^{2y}, m_1' \bullet m_2' \bullet h^{2y})$.

The partner ID has different levels, which represent different strictness degrees the SP requires. For example, a grid of eight pictures will be displayed for a normal degree partner ID in the authentication phase, while it will increase to twelve pictures for a high degree partner ID, which means more distractors.

Meanwhile, the partner ID also presents the SP's information requirement degree of the user. If the SP need not know what the user did during his/her previous entry, namely, if the user's current login has no relevance to his history, the DMAS will give a partner ID of anonymous degree to the SP, which means the user will log in to this SP with anonymous type; all his/her operations in this SP will not be recorded in order to prevent tracking. On the other hand if the SP needs to know the user's previous information and operations during their entry, such as with online bank and online shopping, the DMAS will give a partner ID of real

Table 1. Notations.

K_d	DMAS's public key
K_d^{-1}	DMAS's private key
K_{ud}	User's symmetric key
K_u	User's ElGamal public key, {G, q, g, h}
K_u^{-1}	User's ElGamal private key, {x}
TS_U	Timestamp from the User
TS_S	Timestamp from the SP
TS_D	Timestamp from the DMAS
Relation info	Related information of group digital media data
IP_U	User's IP address
DM	User's digital memory objects
EDMO	Encrypted digital memory objects
Ans	The answers of the challenge
PID	Partner ID

<https://doi.org/10.1371/journal.pone.0186925.t001>

Fig 2. UML sequence diagram of registration protocol between SP and DMAS.

<https://doi.org/10.1371/journal.pone.0186925.g002>

degree, which means the user will log in to this SP with real type; all his/her operations in this SP will be recorded and bound to this account in order to ensure the user can continue from the previous session.

The user registers to the DMAS

We can describe the steps with the following formulas:

$$U \rightarrow DMAS : \{DevID, UserID\}K_d$$

$$DMAS \rightarrow U : \{OK\}$$

$$U \rightarrow DMAS : \{\{DM\}K_{ud}, \{Relation_info\}K_u, DevID, UserID, TS_U\}K_d$$

$$DMAS \rightarrow U : \{h(DevID, UserID, \{Relation_info\}K_u), IP_U, TS_U\}K_d^{-1}$$

The user gets the public key of the DMAS, and then executes the following five steps to finish registering, as illustrated in Fig 3.

Step 1: The user randomly chooses a UserID for his/her DMAS account, and generates the unique devID, then he sends UserID and DevID encrypted with the DMAS's public key K_d . To ensure the uniqueness of devID, we first fetch the device's motherboard serial number, CPU serial number, hard disk serial number and other unique numbers on the hardware. We then combine them and use the SHA256 algorithm to generate a message digest to be used as the devID of this device.

Step 2: The DMAS checks if the UserID is available. If available, the DMAS stores the devID and bounds it to the UserID and takes it as the first factor, on the hardware level, in the authentication stage. Then the DMAS creates the account, and sends the DMAS's response to the user.

Step 3: The user generates a symmetric key K_{ud} and ElGamal [24] key pair K_u, K_u^{-1} , then encrypts and uploads the encrypted digital memories with related information.

Fig 3. UML sequence diagram of registration protocol between user and DMAS.

<https://doi.org/10.1371/journal.pone.0186925.g003>

The ElGamal key pair generation process is as follows:

- The user generates an efficient description of a cyclic group G of order q with generator g . See below for a discussion of the required properties of this group.
- The user chooses an x randomly from $\{1, \dots, q-1\}$, and computes $h := g^x$
- The user publishes h , along with the description of G , q , g , as his/her public key K_u . The user retains x as his/her private key K_u^{-1} , which must be kept secret. So the symmetric key K_{ud} and ElGamal private key K_u^{-1} will be kept and owned by the user only.

Then the user divides his/her digital memories which will be uploaded into several groups in which any digital memory objects can be included in many different groups on their device. The user then gives a name they like (even a code word or metaphor) to every group and encrypts the digital memory objects chosen with the symmetric key. The user then uploads the encrypted digital memories. While uploading, the DMAS will generate a feature code for every encrypted digital memory object and relate it to its groups. For example, the user creates a group named “food”, and selects five digital memory objects for this group, then, while uploading, the five digital memory objects’ feature codes will be encrypted separately and inserted into a database table whose name is the encrypted outcome of “food”. If other groups want to include these five digital memory objects, they will get a copy feature code of the digital memory object and insert into their database tables. We call these feature codes and group information “related information”. The related information’s database data is all encrypted with ElGamal public key K_u and is then named “relation_info” and sent with UserID and DevID, which are encrypted with the DMAS’s public key K_d before being transmitted.

The details of the process to encrypt related information group data using ElGamal are as follows (we suppose the group name is “food”, and it contains five digital memory objects):

- Use ElGamal public key K_u to encrypt “food”, and use the encrypted outcome as the newly created database table name.
- Use ElGamal public key K_u to encrypt the five digital memory objects’ feature codes separately, the steps are as follows:

- a). The user chooses a random y from $\{1, \dots, q-1\}$, then calculates $c_1 := g^y$ (note: one group's data shares the same y , and is owned by the user)
- b). The user calculates the shared secret $s := h^y$, and maps his/her digital memory objects' feature code m_1, m_2, m_3, m_4, m_5 onto an element $m_1', m_2', m_3', m_4', m_5'$ of G .
- c). The user calculates $c_2 := m_1' \cdot s, c_3 := m_2' \cdot s, c_4 := m_3' \cdot s, c_5 := m_4' \cdot s, c_6 := m_5' \cdot s$, and computes the ciphertext $(c_1, c_2) = (g^y, m_1' \cdot h^y), (c_1, c_3) = (g^y, m_2' \cdot h^y), (c_1, c_4) = (g^y, m_3' \cdot h^y), (c_1, c_5) = (g^y, m_4' \cdot h^y), (c_1, c_6) = (g^y, m_5' \cdot h^y)$
- c. Separately insert the outcome into the database table whose name is the encrypted outcome of "food".

Step 4: The DMAS decrypts and verifies the timestamp and checks if the DevID matches the UserID. If there is a match, the DMAS stores the encrypted relation information with the encrypted digital memories. Then the DMAS will send a data packet which contains the hash values of devID, UserID and the encrypted related information ($\{Relation_info\} K_d$). In addition, it also contains the user's IP address (IP_U) and the timestamp from the DMAS (TS_D). All this data is encrypted with the DMAS's private key K_d^{-1} .

Step 5: The user gets the response and verifies the data. If there is a match, it means the UserID is registered successfully.

The user logs in to the SP

We can describe the steps with the following formulas:

$$\begin{aligned}
 SP &\rightarrow DMAS : \{PID, TS_S\} K_d \\
 DMAS &\rightarrow U : \{OK\} \\
 U &\rightarrow DMAS : \{DevID, UserID, TS_D\} K_d \\
 DMAS &\rightarrow U : \{EDMO, h(UserID), TS_D\} K_d^{-1} \\
 U &\rightarrow DMAS : \{DevID, UserID, Ans, TS_U\} K_d \\
 DMAS &\rightarrow U : \{h(UserID), Pass_flag, TS_D\} K_d^{-1} \\
 DMAS &\rightarrow SP : \{h(PID), UserID/AnonymousID, TS_D\} K_d^{-1} \\
 SP &\rightarrow U : \{OK\}
 \end{aligned}$$

When the user wants to get service from the SP, he should be authenticated by the DMAS, the the login phase has 7 steps, as illustrated in Fig 4:

Step 1: When the user wants to log in to the SP, the SP will require DMAS for authentication. The SP encrypts its partner ID and the timestamp from the SP (TS_S) with K_d and sends them along with the user IP address to the DMAS.

Step 2: The DMAS decrypts the data and verifies if the TS_S is legal, then retrieves the strictness degree (number of digital memory objects) and anonymous degree (anonymous or not) for the desired authentication challenge according to the SP's partner ID, and redirects the DMAS's authentication page to the user.

Step 3: Upon receiving it, the user inputs his/her UserID of the DMAS account, computes the devID and then encrypts the UserID, the devID and the timestamp (TS_U) with K_d . Finally, the user sends them to the DMAS.

Step 4: The DMAS decrypts the data with K_d^{-1} and verifies if the devID matches the UserID. Then the DMAS, according to strictness degree and anonymous degree of the SP (see step 2), sends the challenge to the user which contains encrypted digital memory objects (EDMO) and

Fig 4. UML sequence diagram of login protocol between the user and the SP.

<https://doi.org/10.1371/journal.pone.0186925.g004>

encrypted group names for authentication, the hash value of UserID and the timestamp from the DMAS (TS_D). All this data is encrypted with K_d^{-1} .

Step 5: The user decrypts the data with K_d and verifies if the TS_D and hash value (UserID) are legal, then decrypts the encrypted digital memory objects and group names with K_{ud} , K_u^{-1} and answers the challenge. After finishing the answers, the user submits the response which contains the UserID, the devID, the answers of the challenge (Ans) and the timestamp from the user, this time (TS_U). All of this data is encrypted with K_d too. The Ans is made up of the feature code of the digital memory objects selected by the user, and is encrypted with the user's public key K_u .

Step 6: The DMAS decrypts the data and checks if the TS_U is legal and whether the devID matches the UserID. If all is matching, the DMAS then verifies the Ans. Because the Ans is encrypted with the user's K_u , the DMAS needs to verify it with relation_info using Elgamal homomorphic encryption.

The details of the DMAS verifying the Ans with Elgamal homomorphic encryption[25] are as follows:

- After the user gets the group name with K_u^{-1} , he chooses the right digital memory objects and generates their feature codes, then retrieves the parameter y , and carries out the encryption process. For example, with two digital memory objects, we do the following: First, map digital memory objects' feature codes m_1, m_2 onto an element m_1', m_2' of G . Second, calculate $g^y, h = g^y$, the shared secret $s := h^y$, and calculate $m_1' \cdot s = m_1' \cdot h^y, m_2' \cdot s = m_2' \cdot h^y$, Third, construct $Ans = (g^y \cdot g^y, m_1' \cdot h^y \cdot m_2' \cdot h^y) = (g^{2y}, m_1' \cdot m_2' \cdot h^{2y})$ When DMAS receives the Ans from the user, he chooses the encrypted feature codes (c_1, c_2) of m_1 , (c_1, c_3) of m_2 , then computes $Ans' = (c_1, c_2)(c_1, c_3) = (c_1 \cdot c_1, c_2 \cdot c_3) = (g^{2y}, m_1' \cdot m_2' \cdot h^{2y})$, since we have $(c_1, c_2) = (g^y, m_1' \cdot h^y), (c_1, c_3) = (g^y, m_2' \cdot h^y)$.
- If $Ans = Ans'$, then the user is authenticated.

Then, the DMAS will give a pass flag to the user which contains the hash value of the UserID, the Pass_flag and the timestamp from the DMAS, this time (TS_D). All this data is encrypted with K_d^{-1} .

Step 7: The user decrypts the data with K_d and verifies the hash value of the UserID and whether the TS_D is legal. If they are legal and the flag is a Pass_flag, it means the login is successful. Meanwhile, the SP will get the response from the DMAS, which contains the hash value of the partner ID. Also, the response contains the UserID (or anonymous ID according to the degree) and the timestamp from the DMAS, (TS_D) this time. All the data in the response is encrypted with K_d^{-1} . The SP decrypts the data and verifies the TS_D, then gets the UserID and responds to the user. If the degree of this SP is anonymous, the response UserID will be an anonymous ID, which means it has no relation to the real UserID. Otherwise, the response UserID will be the user's real UserID on the DMAS.

The user logs in to the DMAS to maintain digital memory objects

When the user needs to update the digital memory objects to achieve a more secure authentication result, he should first launch the login process, then follow step 3 to step 7 of Phase 3. For simplicity, we do not explain this in detail.

And then he follows step 3 and step 4 of Phase 2 to update or maintain his digital memory objects.

Security analysis & protocol evaluation

In this section, we choose several state-of-the-art schemes which are similar to our scheme to compare the security properties. The comparison can be seen Table 2.

Compared to Passpoints [16], IPAS [10], and Shone et al [18], our scheme has the following advantages:

1. Leak-proof: The DMAS stores users' encrypted digital memories, and the symmetric key is owned by the user only, which means the DMAS can't get the user's clear-text digital memories, which greatly protects the user's privacy.
2. Free from SP: We change the design of the register dependencies. In our mechanism, the user doesn't need to have any registration relationship with the SP. The user can log in to the SP as long as he/she has an DMAS account. When the user wants to log in to the SP, the SP will require a DMAS authentication site for authentication. This greatly simplifies the process of the authentication and reduces the possibility of divulging sensitive information, like devID.
3. Anti-tracking: In our scheme, the DMAS checks the anonymous degree the SP needs and gives anonymous ID to the SP who needn't know the user's operation history, which prevents unnecessary tracking of the user.
4. Different method: We use asymmetric encryption instead of symmetric encryption to secure the communications among the actors, which means in the authentication stage the session keys do not need to be transmitted among the actors by the network environment, which has potential risks. This will reduce the risk of suffering man-in-the-middle attack. Moreover, it

Table 2. Security comparison between state-of-the-art schemes and our scheme.

	Digital memory	Multi-level authentication	Privacy leak-proof	Semi-trust SP	User anti-tracking	Encryption method
PassPoints[16]	×	✓	×	×	×	N/A
IPAS[10]	✓	✓	×	×	×	N/A
Shone et al 's scheme [18]	✓	✓	×	×	×	symmetric
Our scheme	✓	✓	✓	✓	✓	Asymmetric

<https://doi.org/10.1371/journal.pone.0186925.t002>

reduces the load on the actors' Web servers (the SP and the DMAS) because the servers do not need to receive, store and transmit the keys for every user during authentication.

Also, we used the Scyther Tool [26], [27] to provide a formal analysis. Scyther is an automatic security protocol verification tool, which is used to identify potential attacks and vulnerabilities. It has been used to verify numerous security protocols. We used Scyther to evaluate the following properties of our proposed mechanism:

1. Secrecy
2. Replay attack resistance
3. Man-in-the-middle attack resistance
4. Reflection attack resistance

First we used Scyther to analyze the registration protocol between the user and the DMAS whose UML sequence diagram is illustrated in Fig 3. Fig 5 illustrates the results obtained from Scyther's analysis and we can see there is no attack present in our scheme. The Scyther testing code is in Appendices A.

Then we use Scyther to analyze the login protocol between the user and the DMAS. The Scyther testing code is in Appendices B. Fig 6 illustrates the results obtained from Scyther's analysis, and our scheme gets a good result.

Finally, we use Scyther to analyze the login protocol among the user, SP and DMAS whose UML sequence diagram is illustrated in Fig 4. The Scyther testing code is in Appendices C. Fig 7 illustrates the results obtained from Scyther's analysis.

As can be seen, Scyther has been unable to determine any weaknesses or feasible attacks against our proposed scheme. We repeated the Scyther verification ten times, using both the manually defined claims and Scyther's automatically generated claims, and the results remained the same.

Meanwhile, we also verified the security of our scheme from the point of view of cryptography information entropy.

The security of the password depends on the information entropy. The higher the information entropy, the higher the randomness, and the higher the security of the password. Suppose we set the number of pictures to n , the number of times the user needs to be challenged is t , the number of correct options in i round is k_i , so the information entropy is

$$H = \log\left(\prod_{i=0}^t C_n^{k_i}\right) \quad (1)$$

From the user experience point of view, the number of options should not be too many, 9–15 is appropriate. For security considerations, the correct number of options per round is at least 3, so you can get the minimum $H_{\min}$, and by probability, we can get the maximum $H_{\max}$ when $k_i = \lfloor \frac{n}{2} \rfloor$. The formulas are as follows:

$$H_{\min} = \log((C_n^3)^t) \quad (2)$$

$$H_{\max} = \log\left(\left(C_n^{\lfloor \frac{n}{2} \rfloor}\right)^t\right) \quad (3)$$

Since the optional range of Arabic numerals is 0 to 9, the information entropy H_{num} of each Arabic numeral is 1. And the known keyboard can knock out 98 different characters, so the

Claim				Status		Comments
User_DMAS_REG	User	User_DMAS_REG,User1	Secret P	Ok	Verified	No attacks.
		User_DMAS_REG,User2	Secret R_I	Ok	Verified	No attacks.
		User_DMAS_REG,User3	Secret DevID	Ok	Verified	No attacks.
		User_DMAS_REG,User4	Secret UserID	Ok	Verified	No attacks.
		User_DMAS_REG,User5	Alive	Ok	Verified	No attacks.
		User_DMAS_REG,User6	Weakagree	Ok	Verified	No attacks.
		User_DMAS_REG,User7	Niagree	Ok	Verified	No attacks.
		User_DMAS_REG,User8	Nisynch	Ok	Verified	No attacks.
DMAS	DMAS	User_DMAS_REG,DMAS1	Secret P	Ok	Verified	No attacks.
		User_DMAS_REG,DMAS2	Secret R_I	Ok	Verified	No attacks.
		User_DMAS_REG,DMAS3	Secret DevID	Ok	Verified	No attacks.
		User_DMAS_REG,DMAS4	Secret UserID	Ok	Verified	No attacks.
		User_DMAS_REG,DMAS5	Alive	Ok	Verified	No attacks.
		User_DMAS_REG,DMAS6	Weakagree	Ok	Verified	No attacks.
		User_DMAS_REG,DMAS7	Niagree	Ok	Verified	No attacks.
		User_DMAS_REG,DMAS8	Nisynch	Ok	Verified	No attacks.

Fig 5. Formal analysis results from Scyther for the the registration protocol. Between the user and the DMAS.

<https://doi.org/10.1371/journal.pone.0186925.g005>

information entropy of each text password H_{char} is 2, and through calculation, we can draw the outcome for reaching the same security, the length each Arabic password and text password needs. Table 3 shows the outcome for reaching the same security with Arabic passwords and text passwords, the specification our scheme needs.

After that, we validate the intensity enhancement trend curve of text passwords and our scheme in mathematical statistics, as illustrated in Fig 8:

As can be seen from Fig 8, the intensity enhancement trend of Arabic passwords, text passwords and our scheme increases with the increase of the order of magnitude, and the growth slope of our scheme is greater than the growth slope of the Arabic passwords and text passwords.

Because the Arabic number password's information entropy is very weak, it can be cracked very quickly through a brute force password attack. As for text passwords, there are many possibilities. If it is pure uppercase and lowercase letters, its security is also weak. If it is a combination of numbers and letters, the security is more general, because text passwords are logical reasoning. For example, if someone's name is Jack, his password is more likely to be like Jack,

Claim				Status		Comments
User_DMAS_LOGIN	User	User_DMAS_LOGIN,User1	Secret P	Ok	Verified	No attacks.
		User_DMAS_LOGIN,User2	Secret DevID	Ok	Verified	No attacks.
		User_DMAS_LOGIN,User3	Secret UserID	Ok	Verified	No attacks.
		User_DMAS_LOGIN,User4	Alive	Ok	Verified	No attacks.
		User_DMAS_LOGIN,User5	Weakagree	Ok	Verified	No attacks.
		User_DMAS_LOGIN,User6	Niagree	Ok	Verified	No attacks.
		User_DMAS_LOGIN,User7	Nisynch	Ok	Verified	No attacks.
DMAS	DMAS	User_DMAS_LOGIN,DMAS1	Secret P	Ok	Verified	No attacks.
		User_DMAS_LOGIN,DMAS2	Secret DevID	Ok	Verified	No attacks.
		User_DMAS_LOGIN,DMAS3	Secret UserID	Ok	Verified	No attacks.
		User_DMAS_LOGIN,DMAS4	Alive	Ok	Verified	No attacks.
		User_DMAS_LOGIN,DMAS5	Weakagree	Ok	Verified	No attacks.
		User_DMAS_LOGIN,DMAS6	Niagree	Ok	Verified	No attacks.
		User_DMAS_LOGIN,DMAS7	Nisynch	Ok	Verified	No attacks.

Fig 6. Formal analysis results from Scyther for the the login protocol. Between the user and the DMAS.

<https://doi.org/10.1371/journal.pone.0186925.g006>

or Jack plus a string of numbers (like a birthday). In our scheme, because of the User's own uploaded digital memory, there is no password to crack, because the brute force password attack requires a dictionary, and our scheme cannot create a crack dictionary because of the digital memory difference among users. Therefore, it is somewhat difficult to guess the answer through logical reasoning.

Furthermore, our scheme uses a large number of cryptographic protocols to ensure that attackers cannot enter the authentication page (such as DevID, timestamp and digital signature), so for attackers, obtaining the challenge and then cracking it is more difficult. Relative to the existing text password which anyone can access and attempt to crack, our security threshold is higher. Table 4 illustrates the password strength of Arabic number passwords, text passwords and our scheme.

Implementation and performance analysis

Implementation

Our scheme is extensively applicable to all kinds of SPs with different strictness degree and anonymous degree requirements. For example, if the user wants to log in to an SP with a

User_SP_DMAS_LOGIN	User	User_SP_DMAS_LOGIN,User1	Secret P	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,User2	Secret DevID	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,User3	Secret UserID	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,User4	Alive	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,User5	Weakagree	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,User6	Niagree	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,User7	Nisynch	Ok	Verified	No attacks.
SP		User_SP_DMAS_LOGIN,SP1	Secret UserID	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,SP2	Alive	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,SP3	Weakagree	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,SP4	Niagree	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,SP5	Nisynch	Ok	Verified	No attacks.
DMAS		User_SP_DMAS_LOGIN,DMAS1	Secret P	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,DMAS2	Secret DevID	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,DMAS3	Secret UserID	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,DMAS4	Alive	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,DMAS5	Weakagree	Ok	Verified	No attacks.
		User_SP_DMAS_LOGIN,DMAS6	Niagree	Ok	Verified	No attacks.

Fig 7. Formal analysis results from Scyther for the the login protocol. among the user, SP and DMAS.

<https://doi.org/10.1371/journal.pone.0186925.g007>

Table 3. The outcome for reaching the same security with Arabic passwords and text passwords, the specification our scheme needs. (n: the number of pictures, t: the number of times the User needs to be challenged).

n	t	H	Equivalent length of Arabic password	Equivalent length of text password
8	3	5.24~5.53	5	2
8	5	8.74~9.23	8~9	4
12	3	7.03~8.90	7~9	3
12	5	11.71~14.83	11~14	5~7
14	3	7.68~10.61	7~10	3~5
14	5	12.81~17.68	12~17	6~8
16	3	8.24~12.33	8~12	4~6
16	5	13.74~20.55	13~20	6~10

<https://doi.org/10.1371/journal.pone.0186925.t003>

Fig 8. The intensity enhancement trend curve of text passwords and our scheme in mathematical statistics.

<https://doi.org/10.1371/journal.pone.0186925.g008>

normal authentication strictness degree requirement, the DMAS is likely to give the user a challenge such as in Fig 9. The DMAS chooses a digital memory object group named “you~” from the user’s digital memories database, which is a proper subset of this group with random quantities displayed in a grid of eight digital memory objects. The name “you~” is a code word and only the user knows the meaning of the code word.

So in this example, the user uses the code word “you~” to refer to some pets pictures. Because of the random quantities, true answer digital memory objects are used and the distractors are also from the user’s digital memories, it’s difficult to guess the true answer.

Another potential application area is for the SPs with higher authentication strictness degree requirements, such as online banking. The DMAS is likely to give the user a challenge such as in Fig 10. The DMAS chooses a digital memory objects group named “chi” from the User’s digital memories database, which is a proper subset of this group with random quantities displayed in a grid of twelve pictures. The name “chi” is also a code word and only the user knows the meaning of the code word.

So in this example, the user uses the code word “chi” to refer to some food digital memory object. In this way, guessing and shoulder surfing attacks will be more difficult.

Performance analysis

The user registers with the DMAS. We test the registration phase with an Intel(R) Pentium(R) 4 [28] device as the user’s device, and the DMAS is a server equipped with Intel(R) Xeon(R) I5520[29]. We upload 10 digital memory objects as two groups, the outcome is as follows Table 5:

Table 4. The strength of Arabic number passwords, text passwords and our scheme.

	High	Medium	Weak
Arabic number password			✓
Text password		✓	
Our scheme	✓		

<https://doi.org/10.1371/journal.pone.0186925.t004>

Fig 9. Example challenge of normal strictness degree and the pictures in picture group “you~”.

<https://doi.org/10.1371/journal.pone.0186925.g009>

The user logs in to the SP. In this phase, we use the same devices as in the registration phase for the user and the DMAS, then we use a second server equipped with Intel(R) Xeon(R) I5520[29] as the SP. The challenge includes one group and 5 digital memory objects used in the authentication step. The outcome is as follows Table 6:

The tables show that the two phases cost little time, little enough to be acceptable in the authentication stage. Also, the CPUs used in our devices in the tests are now outdated, so we believe our scheme can perform much better with up to date devices.

Conclusion and future directions

In this paper, we have presented a digital memories based user authentication scheme with privacy preservation. In our scheme, we adopt homomorphic encryption and public key encryption to achieve leak-proof user privacy, untrackable user behavior, and to provide multi-level

Fig 10. Example challenge of higher strictness degree and the pictures in picture group “chi”.

<https://doi.org/10.1371/journal.pone.0186925.g010>

Table 5. Performance analysis of phase 2: The user registers with the DMAS (h-hash function; AE-Asymm Encrypt; SE-Symm Encrypt, AD-Asymm Decrypt, SD: Symm Decrypt).

Step	Time Used (User)	User	Time Used (DMAS)	DMAS	Total
1	0.5s	1AE	1s	1AD	1AE 1AD
2	2s	10SE 13AE 10h	1s	1AD	10SE 13AE 10h 1AD
3	3s	1AD	0.4s	1AE 1h	1AE 1AD 1h
Total	5.5s	10SE 14AE 10h 1AD	2.4s	2AD 1h 1AE	10SE 15AE 11h 3AD

<https://doi.org/10.1371/journal.pone.0186925.t005>

Table 6. Performance analysis of phase 3: The user logs in to the SP (h-hash function; AE-Asymm Encrypt; SE-Symm Encrypt, AD-Asymm Decrypt, SD: Symm Decrypt).

Step	Time Used (User)	User	Time Used (DMAS)	DMAS	Time Used (SP)	SP	Total
1			1s	1AD	0.3s	1AE	1AE 1AD
2	0.5s	1AE	1s	1AD			1AE 1AD
3	3.4s	2AD 5SD	0.4s	1AE 1h			1AE 1h 2AD 5SD
4	0.5s	1AE	1s	1AD			1AE 1AD
5	3s	1AD	0.8s	2h 2AE	1s	1AD	2AE 2AD 2h
Total	7.4s	2AE 3AD 5SD	4.2s	3AD 3h 3AE	1.3s	1AE 1AD	6AE 7AD 3h 5SD

<https://doi.org/10.1371/journal.pone.0186925.t006>

security in Internet & IoT environments. Also, we prove the reliability and security of our scheme compared to other schemes and present performance analysis and promising evaluation results.

However, there are still some details that can be further optimized, such as if the user forgets his/her digital passwords or wants to edit the picture groups and upload new digital memories, the authentications used nowadays are email authentication or SMS authentication which demand the user receive an email or a SMS and inputs the verification code to verify if the user is legal; these methods will not be appropriate in the future.

In our future work, we are hoping to expand our initial implementation into a fully working prototype and to optimize the details. We also aim to continue further study into digital memories based user authentication and come up with more constructive ideas.

We have uploaded the scyther protocol analysis files, the scyther analyzer, and all the figures that appear in our manuscript as our support file, and anyone can use it without restriction.

Appendices

A: code of Scyther for the the registration protocol between User and DMAS

```

hashfunction H1;
usertype Timestamp;
protocol User-DMAS-REG (User, DMAS)
{
    role User
    {
        const DevID: Nonce;
        const UserID: Nonce;
        const P: Nonce;
        const R-I: Nonce;
        fresh Ts: Timestamp;
        send_1 (User, DMAS, {DevID, UserID, User}pk (DMAS));
        send_2 (User, DMAS, {{P}k (User, DMAS), R-I, DevID, UserID, User,
Ts}pk (DMAS));
    }
}

```

```

recv_3(DMAS, User, {H1(UserID, DevID, R-I), Ts, DMAS}sk
(DMAS));
claim(User, Secret, P);
claim(User, Secret, R-I);
claim(User, Secret, DevID);
claim(User, Secret, UserID);
claim(User, Alive);
claim(User, Weakagree);
claim(User, Niagree);
claim(User, Nisynch);
};
role DMAS
{
    var DevID: Nonce;
    var UserID: Nonce;
    var R-I: Nonce;
    var P: Nonce;
    fresh Ts: Timestamp;
    recv_1(User, DMAS, {DevID, UserID, User}pk(DMAS));
    recv_2(User, DMAS, {{P}k(User, DMAS), R-I, DevID, UserID, User,
Ts}pk(DMAS));
    send_3(DMAS, User, {H1(UserID, DevID, R-I), Ts, DMAS}sk
(DMAS));
    claim(DMAS, Secret, P);
    claim(DMAS, Secret, R-I);
    claim(DMAS, Secret, DevID);
    claim(DMAS, Secret, UserID);
    claim(DMAS, Alive);
    claim(DMAS, Weakagree);
    claim(DMAS, Niagree);
    claim(DMAS, Nisynch);
};
};

```

B: code of Scyther for the the login protocol between User and DMAS

```

hashfunction H1;
usertype Timestamp;
protocol User-DMAS-LOGIN(User, DMAS)
{
    role User
    {
        const DevID: Nonce;
        const UserID: Nonce;
        const Ans: Nonce;
        fresh Ts: Timestamp;
        var Pass-flag: Nonce;
        var P: Nonce;
        send_1(User, DMAS, {UserID, DevID, User, Ts}pk(DMAS));
        recv_2(DMAS, User, {Ts, DMAS, H1(UserID), {P}k(User, DMAS)}sk
(DMAS));
        send_3(User, DMAS, {DevID, UserID, User, Ans, Ts}pk(DMAS));
        recv_4(DMAS, User, {H1(UserID), Pass-flag, Ts}sk(DMAS));
        claim(User, Secret, P);
        claim(User, Secret, DevID);
        claim(User, Secret, UserID);
        claim(User, Alive);
        claim(User, Weakagree);
        claim(User, Niagree);
    }
}

```

```

        claim(User, Nisynch);
    };
    role DMAS
    {
        const P: Nonce;
        const Pass-flag: Nonce;
        fresh Ts: Timestamp;
        var DevID: Nonce;
        var UserID: Nonce;
        var Ans: Nonce;
        recv_1 (User, DMAS, {UserID, DevID, User, Ts}pk(DMAS));
        send_2 (DMAS, User, {Ts, DMAS, H1 (UserID), {P}k (User, DMAS) }sk
(DMAS));

        recv_3 (User, DMAS, {DevID, UserID, User, Ans, Ts}pk(DMAS));
        send_4 (DMAS, User, {H1 (UserID), Pass-flag, Ts}sk(DMAS));
        claim(DMAS, Secret, P);
        claim(DMAS, Secret, DevID);
        claim(DMAS, Secret, UserID);
        claim(DMAS, Alive);
        claim(DMAS, Weakagree);
        claim(DMAS, Niagree);
        claim(DMAS, Nisynch);

    };
};

```

C: code of Scyther for the the login protocol among User, SP and DMAS

```

hashfunction H1;
usertype Timestamp;
protocol User-SP-DMAS-LOGIN (User, SP, DMAS)
{
    role User
    {
        const DevID: Nonce;
        const UserID: Nonce;
        const Ans: Nonce;
        fresh Ts: Timestamp;
        var Pass-flag: Nonce;
        var P: Nonce;
        send_2 (User, DMAS, {UserID, DevID, User, Ts}pk(DMAS));
        recv_3 (DMAS, User, {Ts, DMAS, H1 (UserID), {P}k (User, DMAS) }sk
(DMAS));

        send_4 (User, DMAS, {DevID, UserID, User, Ans, Ts}pk(DMAS));
        recv_5 (DMAS, User, {H1 (UserID), Pass-flag, Ts}sk(DMAS));
        claim(User, Secret, P);
        claim(User, Secret, DevID);
        claim(User, Secret, UserID);
        claim(User, Alive);
        claim(User, Weakagree);
        claim(User, Niagree);
        claim(User, Nisynch);

    };
    role SP
    {
        const PartnerID: Nonce;
        fresh Ts: Timestamp;
        var UserID: Nonce;
        send_1 (SP, DMAS, {SP, PartnerID, Ts}pk(DMAS));
        recv_6 (DMAS, SP, {H1 (PartnerID), UserID, Ts}sk(DMAS));
    };
};

```

```

        claim(SP, Secret, UserID);
        claim(SP, Alive);
        claim(SP, Weakagree);
        claim(SP, Niagree);
        claim(SP, Nisynch);
    };
    role DMAS
    {
        const P: Nonce;
        const Pass-flag: Nonce;
        fresh Ts: Timestamp;
        var DevID: Nonce;
        var UserID: Nonce;
        var Ans: Nonce;
        var PartnerID: Nonce;
        recv_1(SP, DMAS, {SP, PartnerID, Ts}pk(DMAS));
        recv_2(User, DMAS, {UserID, DevID, User, Ts}pk(DMAS));
        send_3(DMAS, User, {Ts, DMAS, H1(UserID), {P}k(User, DMAS)}sk
(DMAS));

        recv_4(User, DMAS, {DevID, UserID, User, Ans, Ts}pk(DMAS));
        send_5(DMAS, User, {H1(UserID), Pass-flag, Ts}sk(DMAS));
        send_6(DMAS, SP, {H1(PartnerID), UserID, Ts}sk(DMAS));
        claim(DMAS, Secret, P);
        claim(DMAS, Secret, DevID);
        claim(DMAS, Secret, UserID);
        claim(DMAS, Alive);
        claim(DMAS, Weakagree);
        claim(DMAS, Niagree);
        claim(DMAS, Nisynch);
    };
};

```

Supporting information

S1 File. Scyther protocol analysis file: User-DMAS-LOGIN.spdl.
(SPDL)

S2 File. Scyther protocol analysis file: User-DMAS-reg.spdl.
(SPDL)

S3 File. Scyther protocol analysis file: User-DMAS-SP LOGIN.spdl.
(SPDL)

S4 File. The scyther analyzer: scyther-w32-v1.1.3.
(ZIP)

Acknowledgments

We received helpful comments from multiple anonymous reviewers. We receive funding from the 2017 Discipline Construction of Zhejiang Provincial Key University—Cyberspace Security (No.GK178800225005), Natural Science Foundation of China (No.61401128), Zhejiang Province Natural Science Foundation (No. LQ14F020010), Project of Zhejiang Provincial Key Enterprises Institute Construction and Project of Zhejiang Provincial Smart City regional synergy innovation center. None of the authors declare any conflicts of interest. Data have been made available in the Supporting Information.

Author Contributions

Formal analysis: JunLiang Liu.

Investigation: Qiuhua Wang.

Resources: Xiangxiang Yu.

Software: Xiangxiang Yu.

Supervision: Qiuyun Lyu.

Validation: Qiuhua Wang.

Writing – original draft: JunLiang Liu.

Writing – review & editing: JunLiang Liu, Qiuyun Lyu.

References

1. Hu F., Jiang M., Wagner M., and Dong D. C., Privacy-preserving telecardiology sensor networks: Toward a low-cost portable wireless hardware/software codesign, *IEEE Trans. Inf. Tech. Biomed.*, vol. 11, no. 6, pp. 619–627, Nov. 2007.
2. Erkin Zekeriya, Veugen Thijs, Toft Tomas, and Lagendijk Reginald L., Generating Private Recommendations Efficiently Using Homomorphic Encryption and Data Packing, *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1053–1066, June 2012.
3. Xiaoyuan, S., Z. Ying, et al. Graphical passwords: a survey, *Proceedings of the 21st Annual Computer Security Applications Conference (ACSAC)*, 10 pp. - 472, 2005.
4. Wei Hu, Xiaoping Wu, Guoheng Wei. The Security Analysis of Graphical Passwords. *International Conference on Communications and Intelligence Information Security*, pp. 200–203, 2010.
5. Fu Zhangjie, Wu Xinle, Guan Chaowen, Sun Xingming and Ren Kui, Toward Efficient Multi-keyword Fuzzy Search over Encrypted Outsourced Data with Accuracy Improvement, *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.
6. Hayashi, E., and Hong, J., A Diary Study of Password Usage in Daily Life, *International Conference on Human Factors in Computing Systems (CHI)*, May 7–12, pp. 2627–2630, 2011.
7. Thomas S. Tullis and Donna P. Tedesco. Using Personal Photos as Pictorial Passwords. *Extended Abstracts Proceedings of the 2005 Conference on Human Factors in Computing Systems, (CHI)*, pp. 1841–1844, April 2–7, 2005.
8. Chan C.K. and Cheng L.M., Cryptanalysis of a remote user authentication scheme using smart cards, *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 992–993, November 2000.
9. Adomavicius G. and Tuzhilin A., Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.
10. Sadiq Almuairfi, Parakash Veeraraghavan and Naveen Chilamkurti. IPAS: Implicit Password Authentication System. *IEEE Workshops of International Conference on Advanced Information Networking and Applications*. pp. 430–435, 2011.
11. Haichang, G., L. Xiyang, et al. Design and Analysis of a Graphical Password Scheme. *Innovative Computing, Information and Control (ICICIC)*, 2009 Fourth International Conference on Graphical Passwords. pp. 675–678, 2009.
12. Li Jian, Li Xiaolong, Yang Bin, and Sun Xingming, Segmentation-based Image Copy-move Forgery Detection Scheme, *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 3, pp. 507–518, Mar. 2015.
13. Fu Zhangjie, Sun Xingming, Liu Qi, Zhou Lu, and Shu Jiangang, Achieving Efficient Cloud Search Services: Multi-keyword Ranked Search over Encrypted Cloud Data Supporting Parallel Computing, *IEEE Transactions on Communications*, vol. E98-B, no. 1, pp. 190–200, 2015.
14. RealUser [EB/OL]. [2007-11-1]. www.realuser.com.
15. Dirik, A., Memon, N., and Birget, J.-C. Modeling User Choice in the PassPoints Graphical Password Scheme, *Symposium on Usable Privacy & Security (SOUPS)*, pp. 20–28, 2007.
16. Wiedenbeck S., Waters J., Birget J., Brodskiy A., and Memon N. PassPoints: Design and Longitudinal Evaluation of a Graphical Password System. *Int. J. of Human-Computer Studies* 63, 1–2, pp. 102–127, 2005.

17. Takada T. and Koike H. Awase-E: Image-Based Authentication for Mobile Phones Using User's Favorite Images. *Human-Computer Interaction with Mobile Devices and Services*, Springer Berlin / Heidelberg. 2795: pp.347–351. 2003.
18. Nathan Shone, Chelsea Dobbins, William Hurst, Qi Shi. Digital Memories Based Mobile User Authentication for IoT. In *IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*. pp.1796-1802,2015.
19. V. Bush.As We May Think. *The Atlantic Monthly*. 1945.
20. Doherty A. R., Caprani N., Conaire C., Kalnikaite V., Gurrin C., Smeaton A. F., and O'Connor N. E., Passively Recognising Human Activities Through Lifelogging. *Comput. Human Behav.*, vol. 27, pp. 1948–1958.2011.
21. Fu Zhangjie, Ren Kui, Shu Jiangang, Sun Xingming, and Huang Fengxiao, Enabling Personalized Search over Encrypted Outsourced Data with Efficiency Improvement, *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.
22. Zhangjie Fu, Xingming Sun, Sai Ji, and Guowu Xie, Towards Efficient Content-aware Search over Encrypted Outsourced Data in Cloud, *Proceedings of the 35th Annual IEEE International Conference on Computer Communications (IEEE INFOCOM)*, San Francisco, CA, 2016, <https://doi.org/10.1109/INFOCOM.2016.7524606>
23. Sape J. Mullender, BAN Logic A Logic of Authentication <http://wwwhome.cs.utwente.nl/~sape/sse/ban.pdf>
24. ElGamal T., A public key cryptosystem and a signature scheme based on discrete logarithms, *IEEE Transactions on Information Theory*, vol. 31, no. 4, pp. 469–472, July 1985.
25. JH Cheon, J Coron, J Kim, et al. Batch Fully Homomorphic Encryption over the Integers, *Annual International Conference on the Theory and Applications of Cryptographic Techniques EUROCRYPT 2013: Advances in Cryptology–EUROCRYPT 2013* pp 315–335,2013.
26. C. Cremers, The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols, *20th International Conference on Computer Aided Verification (CAV)*, pp. 414–418,2008.
27. C.J.F. Cremers. Scyther: Unbounded Verification of Security Protocol. <http://e-collection.library.ethz.ch/eserv/eth:4952/eth-4952-01.pdf>
28. B. Bentley. Validating the Intel (R) Pentium (R) 4 microprocessor, *International Conference on Dependable Systems and Networks(DSN)*,pp.244-248,2001.
29. Mateusz Berezacki, Eitan Frachtenberg, Mike Paleczny, Kenneth Steele. Many-core key-value store, *2011 International Green Computing Conference and Workshops (IGCC)*,pp.1-8, 2011.