I.J. Computer Network and Information Security, **2014, 7, 11-18**
Published Online June 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2014.07.02

ECCO Mnemonic Authentication

Two-Factor Authentication Method with Ease-of-Use

Saman Gerami Moghaddam, Amin Nasiri, Mohsen Sharifi
Computer Engineering Department Iran University of Science and Technology Tehran, Iran
{samangerami, a_nasiri}@comp.iust.ac.ir, msharifi@iust.ac.ir

***Abstract*—**Not very long ago, organizations used to identify their customers by means of one-factor authentication mechanisms. In today's world, however, these mechanisms cannot overcome the new security threats at least when it comes to high risk situations. Hence, identity providers have introduced varieties of two-factor authentication mechanisms. It may be argued that users may experience difficulties at time of authentication in systems that use two-factor authentication mechanisms for example because they may be forced to carry extra devices to be authenticated more accurately. This is however the tradeoff between ease-of-use and having a secure system that may be decided by the users and not the security providers. In this paper we present a new two-factor authentication mechanism that secures systems and at the same time is easier to use. We have used mnemonic features and the cache concept to achieve ease-of-use and security, respectively. Also, we have tested our method with almost 6500 users in real world using The Mechanical Turk Developer Sandbox.

Index Terms—Security, authentication, identification, privacy, cache-based, mnemonic.

I. Introduction

It goes without saying that information can protect only when it is accessed just by authorized users. Gain or revoke access to a group of people can be achieved merely with a proper authentication method. Nowadays, text-based passwords are the most common mechanism for authenticating humans to computers. Providers, however, are worried about complexity of those passwords users pick. So, they have to use password-composition policies in order to make users choose a strong password. These policies become stricter and stricter over years and especially in sensitive situation [1]. Undoubtedly, keeping complex passwords in mind in not easy and users are not willing to follow these rules [2]. Hence, most of them choose one and reuse it in all situations. This is one of the serious problems have to cope with in authentication mechanisms called password reuse [3].

Weakness of password based authentication methods cannot be simply narrowed down to password reuse. When exploring the facts causing text passwords unreliable, password lost, password sharing, steeling passwords, and cracking passwords are merely a few instances of reasons that show the weakness of text passwords [4].

Also there are services that require a higher level of assurance such as electronic banking that need to use more secure mechanism. Taking everything into consideration, provides can't trust on text passwords authentication or even any one-factor authentication.

One of the alternative methods is biometric authentication. But, acquiring biometric system is neither convenient nor inexpensive. Moreover, biometric mechanism cannot ensure a completely secure system, since, obtaining a copy of an individual's biometrics is needed to make the process of authentication reliable. Nonetheless, major problem in biometric mechanisms is the uniqueness of biometric attributes [2]. Namely, if an individual's biometric is stolen, it means that it is stolen for the entire victim's life and could not be trusted anymore

In the past one-factor authentication has been good enough security, but the modern connected world needs more security. This leads to using of multi-factor authentication. Nowadays, there are a number of organizations who have developed two factor authentication systems in order to identify their customers [5, 6, 7]. Among implemented methods, combination of password and token is the most common approach [8]. Another common problem is the fact that they are often single-purpose solution (e.g. they can only be used for one bank). Also, we should keep in mind that users desire a solution which gives them little control on level of assurance.

In this paper we will discuss our new two-factor authentication method which is based on ECCO and mnemonic features. The remainders of this paper are organized as follows: Section II is an overview of two factor authentication methods. In Section III some challenges of two-factor authentication mechanisms will be discussed. Section IV presents our new authentication method. In this section the terminology of mnemonic and ECCO parameters will be introduced. At the end of this section evaluation of our proposed method is presented and section V concludes the paper.

II. Related works

A. Introduction

In this section a brief description of the mnemonic approach and cache-based features will be discussed. As a target for compression, an overview of two-factor authentication mechanisms will be presented.

B. Two-factor authentication

Three factors of authentication are: what you know, what you have, and what you are. Any dual combination of these factors is called "two-factor authentication" [9]. Between existing two-factor methods, OTP tokens, Challenge/ response tokens, PKI tokens, and SMS OTP are the major ones. One-Time password or OTP tokens are devices that generate passwords based on time-synchronization or mathematical algorithm. Challenge/response tokens are similar to OTP tokens in which the tokens rely on mathematical algorithms for their operation. However, in OTPs a new password is generated based on the previous one, while in challenge/response tokens the new password is generated based on a challenge. The previous two solutions rely on symmetric cryptography which means that the sender and the receiver of a message have to share a single, common key in order to encrypt and decrypt their message. In contrast, PKI tokens rely on public key cryptography. Public key cryptography utilizes a public key to encrypt messages and a private key to decrypt them. SMS OTP or one-time password is sent via SMS text message. Mobile phones are one of the very few devices most people already carry. SMS OTP requires an authentication server sending one-time password by SMS text message to the user. SMS OTP can be used as mutual authentication in which the server and the user authenticate themselves [10].

III. Challenges with two-factor authentication

A. Introduction

In this section, challenges of existing two-factor authentication methods will be discussed. At first, we present our criteria for the assessment of two-factor authentication methods.

B. Criteria

There are people believing and insisting that authentication methods have two basic requirements of security and usability. These people, however, are wrong when it comes to deployment and ease of use. If a particular approach provides improved security but is cumbersome to use or unaffordable to deploy, then it defeats its own purpose. Being difficult to use will lead to users either bypassing the mechanism or electing not to perform the online transaction. Being too expensive will lead to organizations not being able to afford to protect users' online identities. Criteria for evaluating effective strong authentication techniques are listed below:

- Security: Shows how a method can be secure against different attacks.
- Flexibility: Authentication solutions now leverage risk assessment to determine the appropriate level of authentication. For example, a user checking their account balance from home has a different risk profile than attempting an interbank transfer from a foreign country. Of course, this mechanism must be done with identity manager not authentication method but, we believe that a strong method must have the capacity that can give identity manager an ability to leverage risk assessment.
- Easy to use: If a particular approach is too cumbersome or confusing, users will either turn to expensive alternative channels or disengage completely.
- Easy and low cost to deploy: Any feasible solution must be affordable to deploy to millions of users, including fitting easily into existing infrastructures.
- Time: Means how long it takes for a user to authenticate in his account.

The most controversial issue in current methods is about being single purpose. For instance, an OTP device offered by a bank cannot be used for the other ones because of the shared secret technology. This problem makes the users carry more than one device. Also, it is difficult for organizations to integrate their old systems with the new one. The most vulnerable point in SMS OTP is its weak protection against stealing of phone.

A comparison among existing two-factor authentication methods is shown in Table 1.

Table 1. Comparison of two-factor authentication methods

	Security	Flexibility	Easy to Use	Easy & Low Cost to Deploy	Time
Usern./pwd	Low	low	high	high	high
OTP token	High	low	low	low	mid
C/R token	High	low	low	low	low
PKI token	High	low	low	low	mid
SMS OTP	Mid	mid	low	mid	mid

IV. Ecco Mnemonic Method

A. Introduction

In our proposed method we suppose that the user has private devices and he will access websites using them. Though, this assumption may count as a negative point for the proposed method, but, in today's world most of the people use their smart phones or laptops to do their transaction. Moreover, as our proposed method can be used in high risk situations, so, users usually prefer to use their private device.

Of course, we also consider those situations in which the user changes his device. To do this, a one factor mechanism based on Mnemonic will be used.

Furthermore, user's new device's info will be gathered in order to help the user uses his both devices. In sum, our proposal is secure and usable even for the users who prefer to change their devices.

B. Mnemonic

Mnemonic is an approach to human memory usage which can be considered as a memory trick. In fact mnemonic is introduced to be used as a replacement for complex passwords [11]. There are providers introduced an approach which is closed to the concept of mnemonic in which user builds his password from the first letters of easily to remember phrases [12], such as a poem, or a song lyrics. Say, at first the user chooses a sentence like: "David is coming home after two years" which would result in a password such as: "Dichaty". This kind of password is easy to remember and is difficult to crack.

Next generation of memory based password was a two-step password selection. At first the user chooses a simple and memorable password, and then in the next step he will choose a mnemonic substitution for the selected password [13]. In this way a more complex password will be achieved. Inserting numbers or using leet-speaking are the techniques which can be used. However, these techniques do not work for every password, and a unique mechanism is needed to transform each one.

Fastword is another approach proposed by Jakobsson and Akavipat with the purpose of improving previous methods which is considered faster and easier to remember than regular passwords [8]. In his method users are asked to choose a memorable story, and then pick three words from that story. These words have to indicate the story.

C. ECCO

In today's world cookie is integral part of most of authentication methods and many web sites will not work with cookie not allowed. However, cookie is not strong enough to ensure website administrators to track their users especially those who can be considered as attackers in their system [14, 15].

The first reason why cookie per se cannot be considered as a parameter to authenticate users is its lifetime. We did a multi-phase experiment in which we ask about 717 people to visit our website during about 25 days in order to fill out some forms. In background of each form we set a cookie on each visitor's device to find out how long a cookie will be kept by each user.

The contributor to our belief of weakness of cookie cannot be simply narrowed down to cookie's lifetime. There are viruses try to copy cookies stored on victim's computer and send them to someone else. If a virus manages to steal the cookies, it is then possible for it to use it to login to websites pretending to be the victim. This is the reason why many websites do log you off automatically just to be safe. To solve these problems we started an effort to find a solution which recovers cookie's problems.

ECCO stands for Extended Cache-Cookie Object that can be used instead of cookie. In this object, we put a set of parameters beside cookie to make it stronger. These parameters contain Cache-cookie, Etag, browser user agent and some other parameters which will be discussed in continue. In aforementioned experiment we also examined ECCO by setting it's parameters in user's device.

D. The concept

Our proposed method is combination of EECO and mnemonic feature which pre-sented as a two factor authentication mechanism. Let introduce the method using an example. At first, user is asked to create a memorable story but not completely by his own words. To do this, a table with 20 words will be shown him. These words are categorized in 4 groups. User has to choose one word from each group and finally add their own words to. On the other hand, a table like figure 2 will be shown the user:

I	ate	its	cat
He	wrote	her	pen
Jesus	tossed	his	apple
Mum	used	that	form
We	took	my	book

Fig 1. An example of mnemonic table

Now, user has to create a memorable story using 4 words from the table. The pass-phrase can be something like "I was hungry and mum tossed me that apple". It can be concluded that the user chose "mum", "tossed", "that", and "apple" from the table and add his own words to make a meaningful sentence. Unlike the other mnemonic methods, here, user can choose the whole pass-phrase as his password and there is no limitation for length of password.

While the user is choosing his password and filling out the forms, process of creating EECO is being done. An object like an image plays the role of cache-object in our method. When the user opens the webpage for the first time, a hidden image will be cached in his browser along with an id indicating user's identification number. Now, registration process is finished.

Next steps are about user's next visits that we need to authenticate him. In authentication phase, first the user will be identified using ECCO which was created at the time of registration. If identification process was successful, then he needs only to type the 4 words that he chose at the time of registration. These table, although, has the same words, but the words within may ordered in a different way.

E. How it works

As mentioned before, ECCO uses a set of parameters beside cookie to authenticate user. The first parameter we want to introduce is Cache-Cookie. First time, Markus Jakobsson introduced Cache-Cookie to recover cookie's problem [16]. Cache-cookie works like cookie but does not use the same memory to keep its value. Websites store their cookies in a secure area on client's computer which is known as the 'cookie jar'! This is the only area that cookies can be stored and retrieved. However, Cache-Cookie uses browser's cache memory to keep its value. Cache-Cookie object can be anything that has the ability to be stored in browser's cache memory. For example an image can play Cache-cookie's role when it stores in user's cache memory. The main concept behind Cache-cookie is to retrieve a value which is already put in user's browser cache. To do this we put user's id in an object and force the object to cache in client side. Then, in the next visits we will check whether any object is cached or not. If there is any in cache then we will retrieve user's id from. In this way even if we the cookie is lost then we can recognize the user from Cache-Cookie object.

Our experiment showed that Cache-cookie can improve up to above 2% in recognizing our visitors. It is crystal clear that it is not enough yet. So we have to use other parameters to. Etag is a http header which is used to find that if the current version of an object in user's cache is up to date or not. Etag helps clients to download only new objects in a website and decrease download time. But, we have used this value to keep user's id.

The next parameter we want to discuss is flash cookie. Flash cookie is an Adobe object which works like cookie but stores in Adobe space in client's device. Flash cookie would not help us enough because it needs the users have flash player installed on their browser while most of the users may visit websites through their cell phones without flash player installed.

Now that's the time to introduce next parameter which is browsers fonts that is a part of browser user agent. We know that fonts installed on user's device can be different from each other. But how can we use browser fonts to make the users unique? In the experiment we also collected user's device fonts. If we find a set of fonts in one user device which is unique between all visitors and in user's next visits see the same set of fonts then this parameter can be consider as a value to identify the user. It worth remembering that the font sequence received in the server side can be different from user to user even if they have the same set of fonts and they will keep their sequence.

Another parameter is browser installed plug-ins which can be used like browser font. Flash capability string is also another parameter which can be retrieved from flash player on user's device and can be use like browser fonts. Another one is http request header that is a set of http headers. Now, we want to add user's system fonts as an identifier to our method. But how can we achieve this? To do this we consider three conditions to count a set of fonts as an identifier: 1) if the user has the same set of fonts as its set of fonts in his last visit, 2) if the current set is unique through other user's set, 3) if other mentioned parameters such as cookie, and Cache-cookie did not work.

If we consider all ECCO's parameters then we would reach to about 47% identification which shows about 15% improvement of cookie method.

As we would like to propose our method for smart phones, it is needed to see the outcome when we only consider visits with smart phones. The outcome is illustrated in Figure 2 which shows that the progress is from 24% to 35%, namely, about 11%.

We call all these parameters as ECCO. Server read ECCO from user's first request for a page. So user has not involved with this step of authentication at all and any attacker or untrusted third-party couldn't place in middle of transaction.

Now, let talk about the step user has to involve. In our database there are four categories of words. In registration phase, we show the user a table with 20 words that includes 5 words from each category. These categories contain words that are picked at random from large database of suitable words. User has to choose 4 words, one word from each category, to make a meaningful or unmeaningful sentence. According to our experience, both approaches can be helpful, because, some people can better memorize meaningful sentences while the others can better memorize unmeaningful sentences. However, we don't want them to memorize a sentence, we only want them to pick 4 words and make a sentence in a way that it reminds them an event or something that is important to them. Hence, we don't have any limitation to add other words to this sentence for making appropriate and meaningful sentences.

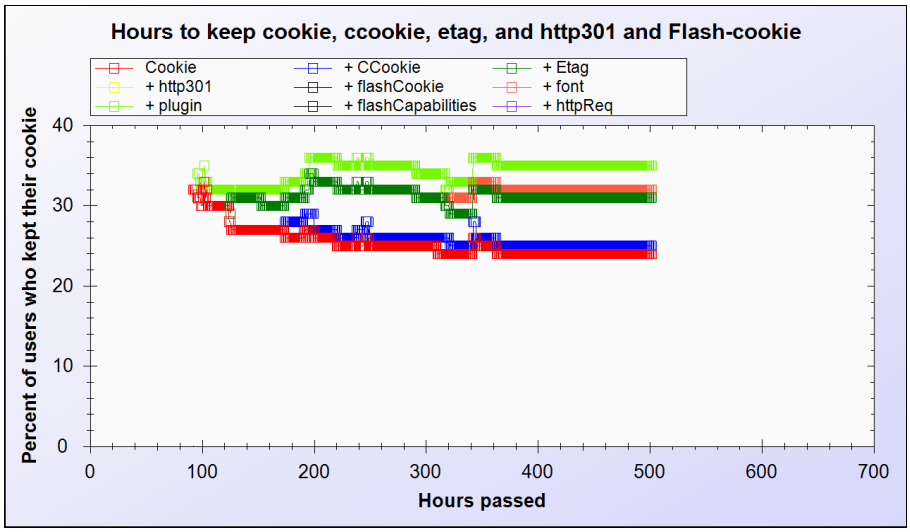

Fig 2. ECCO outcome for mobile devices

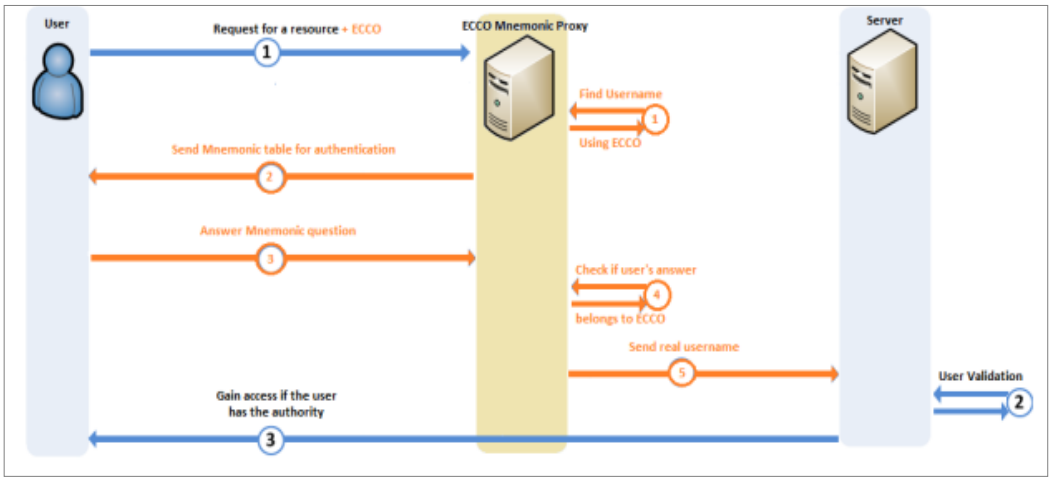

Fig 3. Proposed interaction

In authentication phase, we show the user the same table as the one in registration phase. But, whenever the user opens the webpage the order of the words in each category would be different. Showing the same table in both registration phase and authentication phase will help the user to remember what sentences he had used. Moreover, random order of words helps us to avoid attacks looking at intersections between subsequent collections.

Then, the user will be asked to click on these four words in the right order. An adversary would neither know what the sentence structure is, nor what four words the account owner selected. If there are, say, two possible sentence structures, and there are five words of each type, there would be (54)2 ways of selecting these words.

What was written till now tried to introduce ECCO and mnemonic separately. Now, we would like to show how ECCO can be used as an authentication factor beside Mnemonic. Consider an organization which uses an identity validation mechanism and we want to replace our method with their existing mechanism or add beside it. Figure 4 shows how the organization interacts with its users:

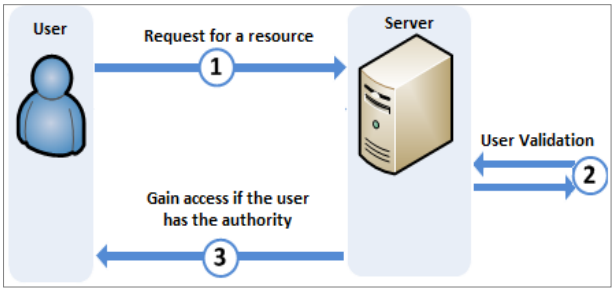

Fig 4. Traditional interaction

Now, if the organization wants to add ECCO Mnemonic method to its current method or as replacement, it just needs to add a proxy server between the user and its server. In this way, neither user nor server will change and consequently the organization should not pay for anything to change its current authentication

mechanism. Figure 3 shows the flows when ECCO Mnemonic proxy is added:

Of course, we need to do some operations in the user's registration time. When the user opens the registration page the ECCO Mnemonic proxy will set ECCO parameters on the user's device in order to identify him in next visits. Figure 5 shows the registration process:

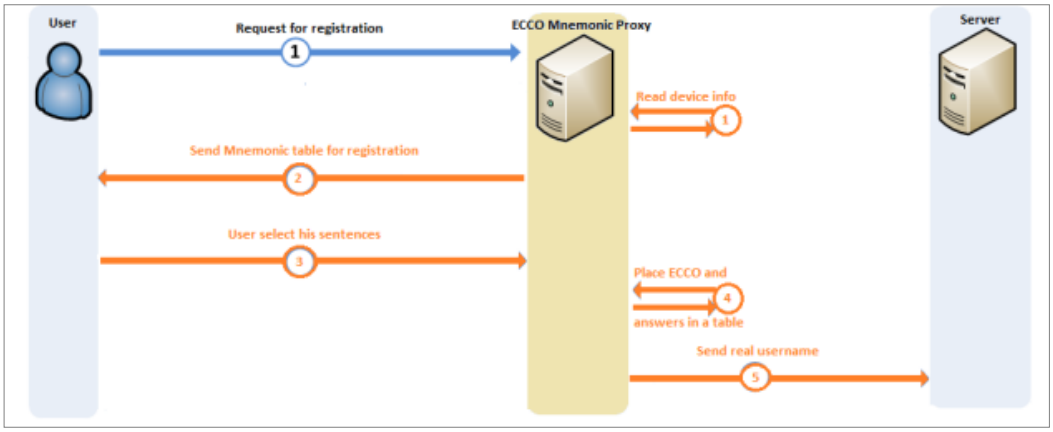

Fig 5. Registration process

F. ECCO mnemonic on new devices

As we mentioned before, our method is secure and usable for transactions which perform from private devices. When user wants to access his account form different devices which are new, then ECCO Mnemonic proxy can't match the new ECCO from the user's device with any records of database. In this case, the proxy tries to authenticate user with just Mnemonic feature. But the Mnemonic feature that is used is different from when it is used with two-factor authentication. In this case, user needs to type his pass-phrase completely. Of course, typing the whole sentences that he chose in registration phase is difficult. To make it simple for the user, our proxy will check similarity of the current sentences entered by the user with the ones entered in the registration phase. If similarity degree satisfies a threshold, then authentication has done successfully. Process of similarity consists of three parts listed below:

1. Elimination of stop –words and auxiliary words
2. Stemming
3. Similarity check with Jakard algorithm
4. Evaluation

In previous sections, we mentioned most of our evaluation outcomes in order to describe how we reach ECCO. Now, we want to introduce entropy that will give us an insight of how useful each parameter in ECCO could be. To calculate the entropy we have used Shannon entropy.

$$H(X) = \sum_{i=1}^{n} p(x_i)\, I(x_i) = \sum_{i=1}^{n} p(x_i) \log_b \frac{1}{p(x_i)} = -\sum_{i=1}^{n} p(x_i) \log_b p(x_i)$$

The entropy is used to measure the uncertainty associated with the visitors.

In our cache-cookie experiments there were 56 different parameters extracted from the users' device. Between these values, some of them are enough to make the user unique. To see this, Shannon entropy of those parameters which can be considered more important are calculated and also written in Table 2. To make it easy to understand, entropy of 'id' which is unique between all visitors is written at the top of the table. This value is the maximum possible value for the Shannon entropy.

Table 2. Shannon entropies for ECCO parameters

Parameter	Shannon Entropy
ID	10.781359713524594
Plug-ins	7.7256481258104923
Flash capabilities	7.3253969299604487
Fonts	6.934254559379756
Browser User agent	6.5844026662766568
Http request	6.5019839140945725
Connection Speed	6.423731208382681
IP Address	5.9911314672122726
Aspect ratio	4.2825846595683386

Some more information about our experiment is shown in Figure 6:

Beside of the experiment for ECCO, we performed an experiment to figure out how usable and memorable is our Mnemonic approach. We performed an experiment in which we asked users to set up 1 to 3 different pass-phrases, and to attempt to authenticate between 2-3 weeks later. Our main sample consists of 30 users. There were 24 male and 6 female respondents. All participants were in their 20's. These participants created 56 accounts and we asked them not to write down any pass-phrases in the duration of experiment.

There were 3 levels for authentication. First user writes his pass-phrase in empty box without any help if he can

remember it. If it failed in next level we show him a table with 2 columns which contains half of the words in setup phase. And in 3th level we show the user whole table to help him for remembering. From total number of subjects that had enrolled in this study; 25 ones complete it. Also 20 subjects have accomplished correct authentication. From total number of 56 accounts which were created by users, 38 accounts were completed. Moreover, 6 accounts were completed correctly in first level. 2 accounts were completed correctly in second level and 18 accounts were completed correctly in third level.

Table 3. Recall rate of mnemonic experiment

Level	Recall Rate
First Level	15%
Second Level	21%
Third Level	68%

The recall rates of the various levels are shown in Table 3. Recall rate of each level is summation of that level and previous levels.

G. Security analysis

The main contribution of this method is its power to cover organization's vulnerability against online attacks. One of the major attacks that existing methods cannot cope with is man in the middle attack. In our method as ECCO is not provided by the user per se, so there is no way to still the ECCO information. In fact, as our proxy will fetch ECCO so the attacker cannot impersonate someone else. In the same way, phishing and pharming attacks would be overcome. As the user does not have any control on ECCO and the word in shown in the table, so he cannot reuse his credential in other situations. Consequently, we can overcome password reuse problem.

VI. Conclusions

Our method prevents users from reusing their password in other systems in how user should make story based on the words in the table and for each session these words are different. Also with this feature, phishing attack isn't possible anymore. Also, cache-based authentication helped us overcome new security threats, such as, phishing, man in the middle, and pharming. However, the only limitation in cache-based authentication is that it is related to a device, and any changes in a user's device will cause to lose cache memory. To solve this problem the mnemonic will help us.

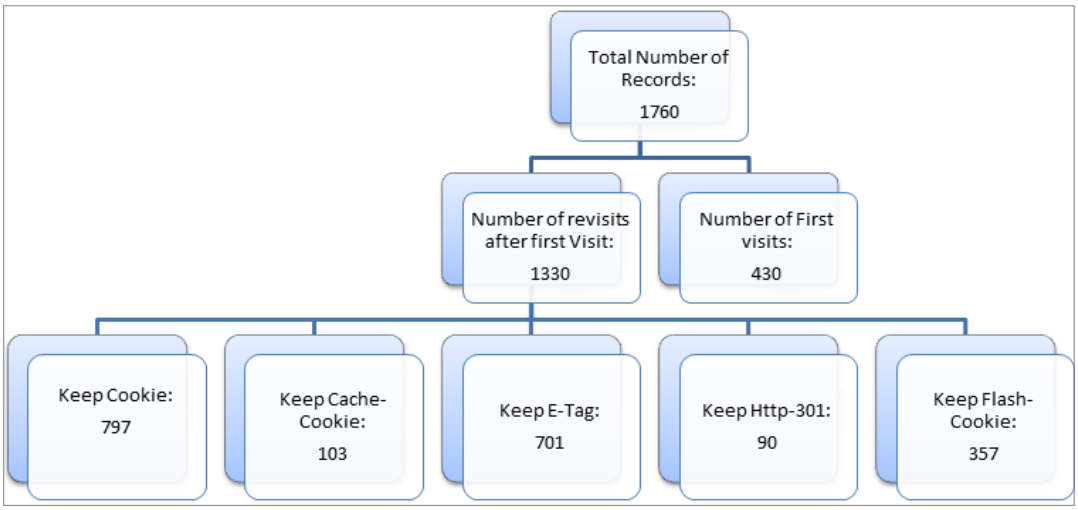

Fig 6. ECCO experiment information

VII. Acknowledgment

The authors would like to thank Dr. Markus Jakobsson, without whom writing this paper was almost impossible. It was a really good learning experience working under him. Also, a very special thanks to Mr. Hossein Siadati, for providing us a valuable information in this area.

References

[1] Markus Jakobsson, Saman Gerami Moghaddam and Mohsen Sharifi, "Mobile Authentication", Book Chapter in: Computer Science, Springer, 2012

[2] K. AltinKemer, and T. Wang, "Cost and benefit analysis of authentication systems", journal decision support systems, vol. 51, issue 3, June 2011.

[3] D. Florêncio, and C. Herley, "A large scale study of web password habits", the 16th interna-tional conference on World Wide Web, New York 2007.

[4] B. Ives,K. R. Walsh, and H. Schneider," The domino effect of password reuse", communications of the ACM, 2004.

[5] Y. Bang, D. Lee, Y. Bae, and J. Ahn, "Improving information security management: an analysis of ID-Password usage and a new login vulnerability measure", international journal of information management, February 2012.

[6] Chaudhari,S, Rawat, A, "Design, Implementation and Analysis of Multi-Layer, Multi-Factor Authentication (MFA) Setup for Webmail Access in Multi-Trust Networks", IEEE 10th ACIS International Conferences on Software.
[7] A. Sabzevar, and P. Sousa, "Improving the Security of Mobile-Phone Access to Remote Personal Computers", International Journal of Software and Data Technologies, Springer, 2009.
[8] P. Eckersley, "How Unique is Your Web Browser?" 10th International Conference on Privacy Enhancing Technologies, Springer-Verlag Berlin, Heidelberg, 2010.
[9] L. Gong, J. Pan, B. Liu, and S. Zhao, "A novel one-time password mutual authentication scheme on sharing renewed finite random sub-passwords" journal of computer and system sciences, vol. 79, issue 1, February 2013.
[10] D. Pavlovic and C. Meadows, "Deriving Authentication for Pervasive Security"; ACM Proceeding of the ISTPS, Texas, USA, 12-16 Juan, 2008.
[11] D. Thanh, I. Jorstad, and T. Jonvik, "Strong Authentication with Mobile Phone as Security Token", IEEE 6th International Conference on Mobile Adhoc and Sensor Systems, Norway, 12-15 November, 2009.
[12] K. L. Vu, R. W. Proctor, A. Bhargav-Spantzel, B. Tai, J. Cook, and E. Schultz, "Improving password security and memorability to protect personal and organization information", international journal of human-computer studies, volume 65, issue 8, August 2007.
[13] K. P. L. Vu, J. Cook,, A. Bhargav, and R. W. Proctor, "Short-term and longterm retention of passwords generated by first-letter and entire-word mnemonic methods", the 5th annual security conference, April 2006.
[14] M. Jakobsson, L. Yang, and S. Wetzel, "Quantifying the security of preference-based authentication", the 4th ACM workshop on digital identity management, Virginia, USA, October 2008.
[15] M. Jakobsson, Shi E and Golle P, "Implicit Authentication for Mobile Devices", 4th USENIX Workshop on Hot Topics in Security, Montreal, Canada, August 2009.
[16] M. Jakobsson, "Web Camouflage: Protecting Your Clients from Browser-Sniffing Attacks", IEEE Symposium on Security and Privacy, California, USA, 2007.
[17] A. Juels, M. Jakobsson and T. N. Jagatic, "Cache Cookie for Browser Authentication", IEEE Symposium on Security and Privacy, California, USA, 2006.

Authors' Profiles

Saman Gerami Moghaddam, is a master of science student at the Department of Computer Engineering, Iran University of Science and Technology, Tehran. His research interest is in the area of network security that focuses on web space vulnerability, he is also an IT Department Manager at Iran Science and Technology Museum where he implement his new ideas.

Amin Nasiri, is a master of science student at the Department of Computer Engineering, Iran University of Science and Technology, Tehran. His research interest is in the area of network security that focuses ease of use, he is also an working as a cell phone programmer.

Mohsen Sharifi, is a Professor at the Department of Computer Engineering, Iran University of Science and Technology, Tehran. His research interest is in the area of distributed system.

How to cite this paper: Saman Gerami Moghaddam, Amin Nasiri, Mohsen Sharifi,"ECCO Mnemonic Authentication—Two-Factor Authentication Method with Ease-of-Use", IJCNIS, vol.6, no.7, pp.11-18, 2014. DOI: 10.5815/ijcnis.2014.07.02

