Strong Zero-knowledge Authentication Based on Virtual Passwords

Younes Asimi[1], Abdallah Amghar[2], Ahmed Asimi[1] and Yassine Sadqi[1]

(Corresponding author: Ahmed Asimi)

Departments of Mathematics and Computer Sciences, Ibn Zohr University[1]

Department of Physic, Ibn Zohr University[2]

Information Systems and Vision Laboratory (LabSiV), B. P. 8106, City Dakhla, Agadir, Morocco

(Email: *asimiahmed*2008@*gmail.com*)

(Received May 23, 2014; revised and accepted Sept. 23 & Nov. 8, 2014)

Abstract

Currently, the security of the users' privacy in public spaces has more concerns especially in web applications. Also, the unconsciousness of users by the importance of the quality cryptographic of these authentication parameters makes their commoditized accounts. Hence, investment in the computer discipline becomes more demanding to prevent potential attacks. In this paper, we introduce a new strong zero knowledge authentication system based on virtual passwords (SAVP). Its objective of this paper is to ensure the identification of users on the network by ensuring intractability, portability, unpredictability, integrity and reusability of their authentication settings. In the second section, we study the difficulties and users habits followed in the selection, storage or memorizing passwords, as well, the evolution and the limits of all categories of texture password authentication. Also, we locate the importance of integration of salts in authentication mechanisms and their impacts on the robustness of passwords regenerated. As for the third section, we start with a detail description of all mechanisms and component contributing to the robustness of our mutual authentication system. Our goal is to provide a strong zero knowledge authentication system based on salts generated by a cryptographically secure random regenerator, algorithm for dynamic rotation of binary strings, symmetric cryptography primitive, one-way hash function and random nonce to provide mutual authentication. The security analysis of our proposal, which is the goal of the fourth section, shows their ability to resist against multiple types of attacks.

Keywords : *Dynamic rotation of binary strings, mutual authentication, one-way hash function, strong zero knowledge authentication, virtual password*

1 Introduction and Notations

Authentication systems have highly evolved in recent years, particularly in public environments especially in web applications. Also, activities and government enterprises rely increasingly on these technologies. This protocol requires identification by username/password and monitoring states of sessions and cookies. In addition, their facilities implementation and deployment have made omnipresent and unavoidable. Their seductive and opportunities in the evolution of companies encourage more attackers to re-evolve their ways of attacks.

Attacks against this protocol affect, in general, the confidentiality of data exchanged between the client and the server. In particular, authentication settings and states monitoring. For this, it requires the use of Secure Socket Layer (SSL/TLS) when registering or logging on to the internet via HTTPS. That seeks to have a valid digital certificate and a browser be able to manage the public key infrastructure (PKI). In case of sensitive data, instead of sending passwords in clear, they introduced one way hash functions to make the password hash. But with the variety of types of attacks that fit with any situation, this system is unable to ensure the privacy of the users. In particular, if we note that static passwords gen-

eration is totally breakable. At present, they are three strong alternative password authentication systems: One-time Passwords [2, 13], Object Passwords [16] and Virtual Passwords [14, 29].

The security of web applications is one of the areas that generate more concerns within research laboratories and companies [3, 7, 8, 9, 13, 19, 20, 21, 22, 26, 28, 31]. In particular, the transmission and storage of highly sensitive data like passwords. Certainly the experts have sacrificed more time to certify the objectives of computer security [31]. The emergence of new vulnerabilities related to cryptographic hash functions, the JavaScript programming language and existing authentication systems, we addressed in this paper to design a new of strong authentication system for remedy these problems. We focused on strong zero-knowledge authentication based on the virtual passwords be able of withstanding data theft attacks on the client or server side, such as Phishing, Shoulder surfing, SQL injection, collision, the man in the middle, brute force, dictionary and spyware. Thus, we rely our proposal on one-way hash function, symmetric cryptographic primitive, salts per user generated by a cryptographic random regenerator sure [1] and random nonce to ensure mutual authentication.

Our work is divided into four sections. In the second section, we study the difficulties and habits followed by users in passwords selection and storage, also, the evolution and the limits of all texture password authentication categories. And, we locate the importance of integration of salts in user authentication mechanisms and their impact on the robustness of the generated passwords. In the third section, we start by describing in detail the objectives of our proposal such as: zero-knowledge, untraceability, portability, integrity and authentication settings. Then, we study the regenerator user-specific random salts and their impact on the quality of cryptographic passwords regenerated. For dealing with problems of static salts, we propose an algorithm for dynamic rotation of binary strings and study its impact on the unpredictability and non-traceability of original passwords totally breakable for minimal disruption. The results obtained show the random nature of passwords generated for the minimum conditions of security. The security analysis of our system, which is the objective of the fourth section, shows the ability of our system to resist against multiple types of attacks.

In the rest of this paper for each user U_i, we denote by:

ID_i	: The user identifier U_i.
PW_i	: A valid original password.
PWV_i	: The virtual password.
HPW_i	: The final password.
RS_i	: Random salt.
$CSRS_i$	: Cryptographically secure random salt.
CRC	: Cyclic redundancy check.
CVL	: CRC code of variables lengths.
DR	: Dynamic rotation.
E	: Symmetric cryptographic primitive.
H	: One-way hash function.
Tb_i, Ts_i	: Random nonce.
$<<<$	: Rotation left without loss of information.
$>>>$	: Rotation right without loss of information.
$\oplus$	: XOR operation.
$==$	: Comparison.
$\|\|$	: Concatenation.
$P(x)$	: Probability of event x.
$NIST$	: National Institute of Standards and Technology.
CC_i	: Challenge of server calculated by the client.
RCS_i	: Challenge and response of client calculated by the server.
RC_i	: Client's response to server's challenge.
RCC_i^{new}	: Response and challenge server's calculated by the client.
X_i^{new}	: Renewal of the parameter X.

2 Related Work

The improvements proposed to evaluate their level of security remain unable to overcome all these weaknesses [26, 31]. In particular, if we note that the design architectures provide static passwords engendering more security concern. Similarly, habits followed by users to select and maintain passwords of many online accounts are courageous for attackers. It should also be noted that all studies in this field confirm that the great challenge among the users is the difficulty to remember a password for each online account [3, 7], which generate the following habits:

- Users choose passwords that are easy, memorizable and guessable.
- They reuse the same password on multiple accounts despite their consciences by the risks.

- They resort to share these passwords with other individuals.
- They often forget their passwords.
- They store them in plain text in the browser.
- They use personal information to build these passwords.

In general, all studies in this field have shown that the problem of memorization and storage is among the major causes of the inability of users to respond to recommendations of the computer security related to passwords [8, 9, 19, 22, 28, 31]. But at the university level, a survey realized by Shay et al. [27] showed that the majority of the users are aware by the impact of the requirements of the computer security on their accounts. Moreover, they found users who can memorize complex passwords. In parallel, other alternatives were proposed to replace the architectures of authentication by texture passwords. Conlan et al. [5] confirmed that the only alternative which entered in significant competition with this technique is the one which based itself on graphic indicators to calculate the passwords of every user. But, the technique of passwords textures stays the most usable, profitable and attractive [32].

2.1 Evolution of Texture Password Authentication

The robustness of a password is the measure of its capacity to resist against various types of attacks. It estimates the average degree of necessary attempts (for every type) to an attacker to discover any original password. The robustness is a function of the length, the range of lengths, the period, the unpredictability, the untraceability, the reusability and the complexity of distribution of the random characters of the password.

2.1.1 Static Passwords

At the time of the computing, this technique was the simplest method of authentication to implement, efficient and secure to protect the accesses to sensitive data. The evolution and the opening of the computer systems on the network have made the static passwords ineffective to assure the privacy of users. Also, the evolutions of the techniques of attacks have trivialized them especially in public environments. In this case, a password remains identical for several connections commonly met under Windows and Unix. The current recommendation is to limit their uses for the local authentication.

2.1.2 One-Time Passwords

To push aside the risk theft of static passwords in insecure channels, Lamport [13] described a scheme of a one-time password (OTP) which based on the repetitive hashing. It generates a different password for every connection more strong than the static password. The inconveniences of this technique come from the dependence of the generated passwords, the listening of doors, the stealing of the passwords and the time required executing N times the hash function. Several variants studies were developed to evolve the level of security of this protocol. Bellovin and Merrit [2], proposed a protocol for exchange of encrypted keys (EKE) and then its extension, which allows preventing the dictionary attacks and the compromise of password files. This extension is based on a one-way hash function to hash passwords, nonce for mutual authentication and Diffie-Hellman to compute a session key. Morris and Thompson [18] introduced another alternative of OTP to ensure password security on UNIX. They are based on storing passwords salted and hashed to reduce the risk of password file compromise [3]. This technique has been improved by Feldmeier and Karn [32].

2.1.3 Objects Passwords

The systems of alphanumeric passwords are easily attacked by shoulder-surfing and Spyware, in which the adversary can record users' movements by a hidden camera when the user tapes the password or with a Trojan Horse. In order to meet the recommendations of the security related to the choice of passwords that have high entropy. Also, to help users who are unable to store random passwords generated by the machine. ObPwd [16] is another alternative system to generate the strong's enough passwords based on digital objects. The user does not need to remember a very complex password. But, just for him to remember a password object locally or in the web. When the user points at an object, this system takes care to recover its signature (SHA-1) as being a password of strong entropy. The choice of objects digital as passwords is an interesting alternative to be explored. Because, in addition to the cryptographic quality of passwords created and maintained by the users, it is very sophisticated against Spyware and shoulder-surfing attacks. Especially, the software that are based on the recording of keystrokes

on local machines.

2.1.4 Virtual Passwords

Another alternative for traditional password was proposed by Lei et al. [14] in 2008. It is based on a virtual password system. Its objective was for them to have a mechanism of authentication capable of withstanding the theft attacks, phishing and the keylogger and shoulder-surfing attacks. They used a linear random function, a salt generated by the random server, a fixed password and a random number selected by the user. This virtual system has been modified by [30] in order to minimize processing time by the server. This system is theoretically breakable because all keys $\{0, ..., Z-1\}$ are finished. In 2011, Sandeep Kumar Sood et al. [31] proposed a Inverse Cookie-based Virtual Password Authentication Protocol. This authentication protocol is based on the storage of cookies on the client computer and the Secure Socket Layer protocol (SSL) to protect the advantages of authentication by password and to evolve its complexity against multiple attacks including dictionary attacks online. But, according to an analytical study made on SSL protocol by American researchers, monitoring of web traffics leaves sufficient information even if the data that transit are encrypted [17]. It also presents a very important evolution for passwords authentication systems, because it allows to regenerate different virtual passwords for every user. But, it does not manage to push aside SSL vulnerabilities. In addition, it does not ensure the quality of the encrypted passwords.

2.1.5 Evolution of Salts

The salt was introduced by Morris and Thompson [18] as another alternative of OTP to ensure the password security on UNIX. We note that several extensions have been proposed to develop the security of the password against multiple attacks specifically against Phishing and Spyware attacks. The technical of SpoofGuard [4] is a browser extension that examines Web pages and notifies the user when data requests may be part of a spoof attack (Phishing). Halderman et al. [11] proposed a mechanism operates entirely on the client. This extension allows the reassurance of the passwords against the attacks of dictionary by means of a hash function. We are stretching the hash function it can complicate the calculation of the original password. More critically, it generates the static passwords unable to resist against multiple attacks (Phishing or Replay attack). In 2005, PwdHash [23] was developed for Internet Browsers Explored and Mozilla Firefox. It allows improving the security of passwords in Web applications. It generates a different password for each site seamlessly. This extension applies a cryptographic function on a password in clear and its private salt stored in the client computer. In general, this extension allows you to generate a global salt (equivalent to the domain name of remote site) specific to each site. This technique helps to prevent Phishing attack but remains unable to resist against network attacks (Man in the middle, Replay attack) and attacks against servers (brute force attack, dictionary attack, theft of the database). Numerous studies on JavaScript attacks showed that the implementation in complete safety of the hashing in the browser is rather difficult on the modern Web applications.

3 Our Proposal

The studied systems of authentication are divided on three categories: virtual passwords, object-based passwords and one-time passwords. The robustness of passwords of all these proposals on one hand is expressed according to the length, to the plage, to the random nature and to the unpredictability and on the other hand is related to the behavior of users which has a very important impact on the cryptographic quality of their passwords, and that it is impossible to control, but can be evolved through the sensitization. The aim of our proposal is to strengthen the users' authentication by virtual passwords. We therefore propose a system be able of withstand multiples types of attacks including Phishing, dictionary attack, brute force, Spyware, man in the middle and also the problem of collision [31]. It minimizes the number of passwords memorized by users. It's based on a salt per user generated by a cryptographically secure random regenerator [1], one-way hash functions, a symmetric cryptographic primitive, nonce to ensure the mutual authentication and the updating of authentication settings during the phase renewal.

Random passwords are difficult to remember. Thus, the interest to introduce this new proposal of a zero-knowledge authentication system based on virtual passwords (SAVP). The users don't need to remember a password for each account and can use it for more than one account. Because, the cryptographic quality of our system is related to the random nature of regenerating of salt used to ensure untraceability of passwords on the network [1].

Our proposal is characterized by:

- A random salt appropriate to each user [1] to avoid the problem of change of domains.
- The integrity of this salt is assured by CRC code of variables lengths [1].
- The space of keys is unlimited and the primitive signals constituting the generated keys meet of the following conditions [1]:
  - Their length and period are variable and unpredictable.
  - Their distributions will also be unpredictable.
  - The untraceability of the keys.
- The users are free to choose the way of seizing words pass by keyboard or to use the passwords objects that have a great ability to counter spyware attacks.
- The users do not need to make calculations. The regeneration of the virtual passwords is made transparently.
- The use of a strong cryptographic hash function (SHA-224).
- The update of the authentication settings collaborates to protect servers against the potentials types of attacks.
- It is almost impossible to find the same virtual password for two users who have the same original passwords.

3.1 Zero-Knowledge Proof

The concept of a proof of zero-knowledge was introduced in the firstly by Goldwasser, Micali and Racko [10]. It is used in cryptography to ensure the identity of users. It appears in the mutual authentication protocols without disclosure of secret data in the form of challenges and responses. The entities must authenticate without needing to reveal the accuracy of their secrets.

3.2 Reuse of Passwords

Users are unable to memorize a complex password for every account. Thus, the majority of them reuse a single password in several accounts, share with others and also store it clear in the browsers [3, 7]. To cope with these difficulties, in our system, we melt the security level passwords regenerated on the cryptographic quality of our regenerator of the salts used [1]. The goal is to have passwords able to resist the network and server's attacks. Therefore, the users will not need to change them to make sure on their cryptographic qualities. But, they have to cope with Spyware attacks.

3.3 Untraceability of Passwords

In internet, the traceability of connection data (logs) is a solution to monitor users and conducting surveys. It also serves to follow their activities to create profiles in the semantic case of Web: the movements, the consulted sites, the exchanges and the sharing. And it can become a cause of mistrust and disclosure of confidential data. Since most attacks are based on spying sensitive data on the web especially the passwords, thus our objective is to propose and study a strong authentication system based on the regeneration of virtual passwords to guarantee their untraceability.

3.4 Portability of Our Authentication System

In addition to security in web applications, it adds another very important characteristic: the portability of an authentication system. Indeed, most authentication systems offer very complex architectures to gain the trust of users. Generally, they base on the capacity of modern Web browsers to memorized the parameters of authentication to simplify the users experience. But, they forget the risks bound to the problem of not standardization of browsers and the security of the files of storage of these parameters client side. More critical, they impede the movement of internet users to a specific browser. For that purpose, in our proposal, all authentication parameters will be stored on the server side to assure the portability of our system. Besides, the passwords will be strengthened by safe cryptographic salts to have more security, simplicity, safety and trust of the users.

3.5 Controls of Integrity

The majority of authentication architectures leave out the control of the integrity of data exchanged between the server and the client during authentication. They can be the cause of failure of authentication because attacks do not always have an intention to have the access to

your account; but they can try just to damage the validity of your parameters of authentication. Consequently, the corruption can be involuntary. For this interest, we propose a dynamic system that ensures the integrity and authenticity of the parameters of authentication of data exchanged between the communicating entities. Thus, we introduce a technique for error detection (CRC) of variables lengths which adapts with any polynomial generator (Noted CVL) [1] to ensure the integrity of messages exchanged between the client and the server salt and an one-way hash function to generate very strong passwords which will be used as encryption keys and decryption.

3.6 Random Generator of a Safe Cryptographic Salt

We refer to [1], the salts regenerated by our regenerator $RGSCS$ have unpredictable primitive signals, pseudo-random and in certain cases seems chaotic. That is to say, their divinations by the successive iterations are almost impossible. The interest to introduce this system is to meet the requirements of computer security and also to solve the problems of storage and memorization of complex passwords of the users in Web applications. It is built on salts appropriate to every user generated by a secure cryptographic random regenerator [1]. The purpose, is to contribute to the level improvement of security of the passwords against multiple types of attacks.

The regenerator $RGSCS$ consists of three processes. For details see [1]:

- The regeneration of random salts.
- The calculation of a CRC of variable length on any primitive signal to assure the integrity of regenerated salts.
- The check of the integrity of salts and the update of the authentication settings.

According to $NIST$ [25], the length and the range of lengths are among the key factors of the robustness of generated passwords. To test the impact of this regenerator on their cryptographic quality, we have to calculate minimal and maximal complexity ($\mathbf{S_{m,N}}$). Then the probability to have such a primitive signal for minimal passwords. According to [1], we have:

1) The cardinal of $\mathbf{S_{m,N}}$ is $\#\mathbf{S_{m,N}} = \mathbf{2^m(2^{N+1} - 1)}$.
2) If the elements of $\mathbf{S_{m,N}}$ are equiprobable then for all $\mathbf{S} \in \mathbf{S_{m,N}}$ we get $\mathbf{P(x) = 1/\#S_{m,N}}$.

The recommendation of the information security is to have a password that consists of at least eight characters. In the table below, we studied the complexity and the probability of the virtual passwords according to a password and salt regenerated by our algorithm [1].

Table 1: The complexity and the probability of the virtual passwords regenerated

The length of the salt (bit)	Complexity	Probability
without	$\mathbf{1.845\ 10^{19}}$	$\mathbf{5.422\ 10^{-20}}$
140	$\mathbf{2.572\ 10^{61}}$	$\mathbf{3.890\ 10^{-62}}$
150	$\mathbf{2.663\ 10^{64}}$	$\mathbf{3.799\ 10^{-65}}$
160	$\mathbf{2.696\ 10^{67}}$	$\mathbf{3.710\ 10^{-68}}$
...	...	...
180	$\mathbf{2.827\ 10^{68}}$	$\mathbf{3.538\ 10^{-74}}$
185	$\mathbf{9.047\ 10^{74}}$	$\mathbf{1.106\ 10^{-75}}$

According to these results (Table 1), we notice that the complexity and the probability to have such a primitive signal are strongly evolved and compared with the original passwords (without salt). Thus, the key space has increased by $\mathbf{1.394\ 10^{42}}$ for a salt of a minimum length and by $\mathbf{4.904\ 10^{55}}$ for a salt of a maximum length by report an original password. Also, the probability of such a primitive signal has decreased by $\mathbf{7.175\ 10^{-43}}$ for a salt of a minimum length and $\mathbf{2.040\ 10^{-56}}$ for a salt of a maximum length by report an original password. Of course, the keys space is very important for evolving their level of security against multiple attacks, but, this is not sufficient to speak about the random complexity of the passwords which meet the requirements of the computer security. For that purpose, we have to estimate the impact of this regenerator on the unpredictability of the regenerated primitive signals.

3.7 Dynamic Rotation of Binary Strings

Knowing that, if an attacker manages to find the static salt associated with a password, their mission to find the original password in clear rest to build a dictionary contains all possible combinations. Indeed, the concatenation has no influence on the level of security, it can extend only its length. And if of more the integration of this technique in the systems of authentication remains in a static way [4, 11, 18, 23], then to strengthen the level of security of a system of authentication based on

passwords, we thought of proposing a virtual system of authentication based not only on blocks of data, but on their binary parts. Hence, we propose a new mechanism of regeneration virtual passwords by basing itself on salts by user [1] and on algorithm of dynamic rotation of the binary strings before the hashing. The goal is to have passwords which have the recommended characteristics in current authentication systems namely: untraceability, randomness, virtual and also reusable.

In our approach, the objective is not to complicate the existing proposals. For that purpose, we build our proposal on simple and practicable operations in most of the programming languages namely:

- The concatenation of a password PW_i and an unpredictable salt RS_i appropriate to every user U_i.
- The regeneration of a binary sequence $S = x_{nb} \dots x_1$ from $PW_i \| RS_i$, with $x_i \in \{0, 1\}$.
- The ordinary sum of the bits positioned in one in S to determine the dynamic position of the rotation P_i: $P_i = \sum_{i=1}^{nb} x_i$.
- The dynamic rotation depends on the parity of P_i, as follows:
  - If P_i is even, we shall have a circular rotation to the right with P_i position.
  - If P_i is odd, we shall have a circular rotation to the left with $nb - P_i$ position.
- Hence, the regeneration of the virtual passwords $PWV_i = DR(PW_i \| RS_i)$.

To estimate the complexity of the virtual passwords generated in our system, a behavioral study is dedicated to the analysis of these generated primitive signals. For this, we will study the divergence of Hamming distances between the primitive signals [1] after minimal internal disturbances (a single bit) on a totally breakable password for the same salt and its impact on the robustness of these passwords regenerated.

3.7.1 Impact of Minimal Perturbations of an Original Password on Virtual Passwords

To estimate the impact of this algorithm of dynamic rotation over the complexity of the regenerated virtual passwords, we will study the distribution of distances of the primitive signals regenerated by minimal disturbances (only bit by iteration) on the initial condition ($PW_i \| RS_i$). For this, we take the original totally breakable password "*aaaaaaaa*" concatenated with a given salt. The perturbations will be only made on the original password.

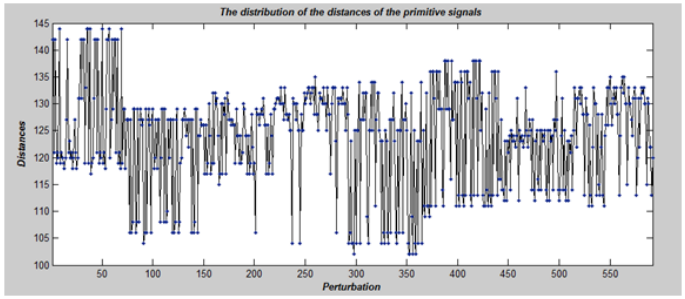

Figure 1: The distribution of distances of primitive signals according to the minimal perturbations on the initial condition

According to this histogram (Figure 1) we notice that, for any iteration, the range of lengths of the distances is more important and more subtle (between 100 and 145 bits), and their distribution seems chaotic. Therefore, our system assures the untraceability in spite the reuse of a same password. This algorithm will thus have a very remarkable contribution on the complexity of the virtual passwords regenerated. However, if an attacker manages to find the final plaintext passwords, it will be painful for him the exact localization of the password entered by the user.

3.7.2 Impact of Salts on the Robustness of Passwords

In order to argue the impact of this dynamic rotation algorithm on the robustness of passwords, we study the correlation of primitive signals regenerated for original password concatenated with two hundred salts. More critically, we chose a password that meets the minimum recommendation of computer security.

Original password: $a * 7F\_eW5$.

According to this histogram (Figure 2), we can split the zones of interest into three portions:

- Between 0.3 and 0.42: the distribution of the normalized distances [1] seems to a chaotic phenomenon.
- Between 0.42 and 0.52: we have an accumulation of the normalized distances. But, with a distribution seems a bit like Gaussian curve followed by small peaks.

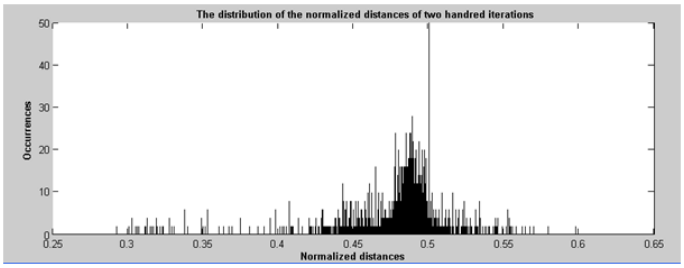

Figure 2: The distribution of normalized distances

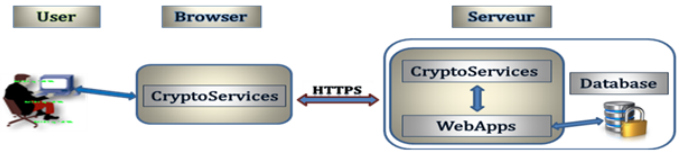

Figure 3: Model of SAVP

- Between 0.52 and 0.6: almost the same as the first portion.

We refer to (Figures 1, 2), $NIST$ [25] and [1], our system has filled the characteristics recommended by computer security. It enables us to make sure well over the cryptographic nature of the virtual passwords. Finally, we can summarize these internal characteristics as follows:

- The distribution of lengths and periods is random.
- The passwords are unpredictable.
- The untraceability of the original passwords.

Consequently, we assure the uncorrelated, the untraceability and unpredictable of the regenerated primitive signals can withstand the multiple types of attacks such as: dictionary attack, brute force attack, phishing attack, man in the middle attack (MIM) and also in the collision problem. Therefore, the robustness and the complexity of the virtual passwords regenerated are assured.

3.8 Strong Authentication by Virtual Passwords

3.8.1 General SAVP Scheme

Figure 3 is a model of strong authentication by virtual passwords (SAVP). The scheme of our proposal is composed of the following items:

The browsers. They would support the protocol $HTTPS$ to guarantee more confidentiality of data exchanged between the customer and the server.

Extension CryptoServices. It must provide, in both sides, the following features:

- The hash functions.
- The symmetric cryptographic primitives.
- The dynamic rotation of binary strings.
- The CRC code of a variable length.
- Regeneration random salts RS_i specific to each user U_i.

WebApps. It is web application usually placed on a web server and handles by pressing of widgets using a web browser via a computer network. It can be a system of content management, search engine, an e-commerce software, a social network, etc.

Database. Each user U_i is characterized by four authentication settings, which will be created during the recording phase. These settings are used to identify users during the authentication phase (See Table 2). They can be changed easily during the renewal phase:

- **Identifier (ID_i):** Only identifier (ID_i) for each user (U_i).

Table 2: Users authentication settings

ID_i	HPW_i	$CSRS_i$	N_i

- **Password (HPW_i):** In our proposal, the password will be used as an encryption key and decryption to ensure:
  1) The identification of users during authentication and renewal phases.
  2) The confidentiality of nonce exchanged between the client and the server to assure the mutual authentication.
  3) The confidentiality of the new passwords chosen by the users during the renewal phase.
- **Salt ($CSRS_i$)**: In the registration phase, a random regenerator handles to regenerate $CSRS_i$ for each user who has a chaotic behavior [1]. It will be associated with the original password to ensure its robustness and its complexity.
- **A positive integer (N_i):** It corresponds to the sum of the bits positioned in one in a primitive

signal RS_i. It will be used to generate a polynomial generator to make sure on the integrity of the salts ($CSRS_i$) [1].

3.8.2 Conception of SAVP

Our mutual authentication system SAVP consists of three phases: the registration phase, the identification and authentication phase and the renewal phase.

3.8.2.1 Registration Phase

This phase, allows any new user to register with the Web application. Each user should have a unique representation within server. The data exchanged very sensitive require a level of confidentiality and integrity quite high (See Figure 4). Hence, the necessity to recommend the use of $HTTPS$ protocol to ensure the confidentiality and integrity of the authentication settings.

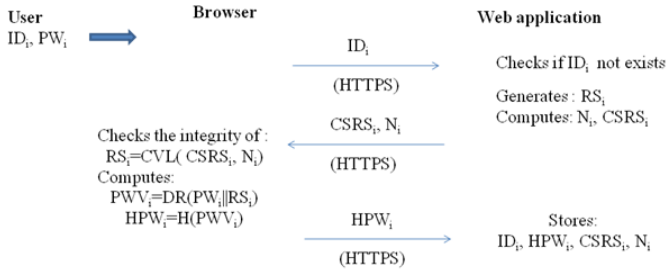

Figure 4: Registration phase

This registration process can generate for each user U_i itself authentication settings. It is based on random salts regenerated by a cryptographically safe regenerator and original password chosen by the user U_i as follows:

- The user U_i must have a valid password PW_i and an unique identifier ID_i that does not exist in the server database. If it exists, the server sends back a message of exception informing the user to choose other identifier.
- The browser sends the identifier ID_i entered by the user U_i to the server.
- The server checks the existence of the user U_i, otherwise:
  - Generates a random salt RS_i.
  - Calculates a number N_i and $CSRS_i$.
  - Sends a salt $CSRS_i$ and a number N_i to the browser.
- The browser:
  - Calculates $RS_i = CVL(CSRS_i, N_i)$.
  - Calculates virtual password, by using the Dynamic Rotation (DR) on the concatenation of an original password and a random salt: $PWV_i = DR(PW_i \| RS_i)$.
  - Calculates the final password by hashing of the virtual password with a one-way hash function H: $HPW_i = H(PWV_i)$.
  - Sends the final password HPW_i to server.
- The server:
  - Saves the authentication parameters associated to the user U_i: ID_i, HPW_i, $CSRS_i$, N_i.

3.8.2.2 Identification and Authentication Phase

In this phase each user U_i must provide a proof of its identity (username/password) to the server. Obviously, authentication systems based on a simple password do not meet the demanding requirements of computer security. For this, we integrated several parameters of authentication to assure strong authentication of the users. The goal is to establish a secured session with the Web server by using the $HTTPS$ protocol and the authentication service. In this phase, we have to make sure on (See Figure 5):

- Identity of the users.
- Integrity and confidentiality of exchanged random salts.
- Validity of recalculated passwords.
- Mutual authentication.

The identification and authentication process allows verifying well the identity and authenticity of the users and the server. The aim is to provide mutual authentication between communicating entities without disclosure the originals parameters of authentication. Also, for more confidence and seductive, we ensure over the untraceability and portability in our system.

- The browser:
  - Sends the identifier ID_i of a user U_i to the server.
  - Generates a nonce Tb_i.
- The server checks the existence of the user:

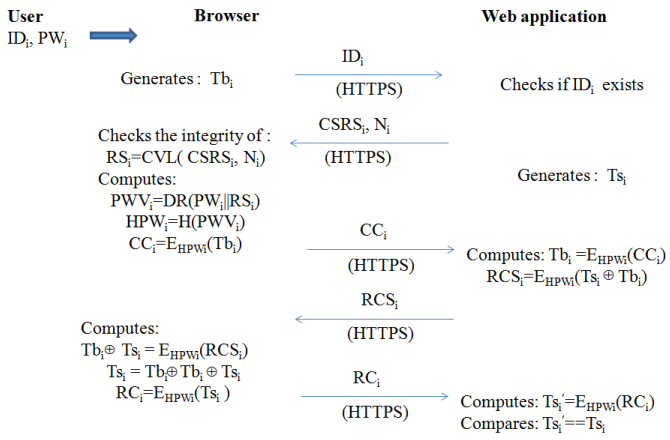

Figure 5: Identification and authentication phase

- If yes, then the server:
  - ∗ sends a cryptographically secure random salt $CSRS_i$ and N_i number.
  - ∗ Generates a nonce Ts_i.
- Otherwise, it returns an error message.

- The browser:
  - Checks the integrity of $CSRS_i$ by calculating $RS_i = CVL(CSRS_i, N_i)$.
  - Calculates the virtual password of a user U_i by Dynamic Rotation applied to the concatenation of its original valid password PW_i and its random salt RS_i : $PWV_i = DR(PW_i \| RS_i)$.
  - Calculates the final password of the user U_i by hashing the virtual password PWV_i with a one-way hash function H: $HPW_i = H(PWV_i)$.
  - Encrypts the nonce Tb_i by the final password HPW_i as a symmetric encryption key: $CC_i = E_{HPW_i}(Tb_i)$.
  - Sends CC_i as an authentication challenge to the server.

- The server:
  - Decrypts the received message: $Tb_i = E_{HPW_i}(CC_i)$.
  - Calculates a challenge for the browser: $RCS_i = E_{HPW_i}(Tb_i \oplus Ts_i)$.
  - Sends as an authentication challenge to the browser : RCS_i.

- The browser:
  - Decrypts the received message: $Tb_i \oplus Ts_i = E_{HPW_i}(RCS_i)$.
  - Calculates: $Ts_i = Tb_i \oplus Tb_i \oplus Ts_i$.
  - Calculates: $RC_i = E_{HPW_i}(Ts_i)$.
  - Sends as a response to the authentication challenge to the server: RC_i.

- The server:
  - Decrypts the received message : $Ts_i^{'} = E_{HPW_i}(RC_i)$.
  - Compares the received nonce of mutual authentication $Ts_i^{'}$ with one who sent Ts_i: $Ts_i^{'} == Ts_i$.
  - If comparison is successful, then:
    - ⋆ Mutual authentication is assured between the browser and the server.
    - ⋆ Successful Connection.

3.8.2.3 Renewal Phase

This phase is very interested and recommended especially for newly registered users. Because, it allows renewing all authentication settings in a more secure environment than registration phase. Also, it offers a higher level of protection of sensitive authentication settings (See Figure 6). In this phase, we must ensure:

- The identity of users.
- The integrity and confidentiality of regenerated salts and passwords.
- The validity of recalculated passwords.
- The mutual authentication.
- Updating of the authentication settings.

This process allows to renew the authentication settings safely. It gives another chance to users for strengthen these authentication settings.

- The browser sends the identifier ID_i of a user U_i to the server.
- The server checks the existence of the user:
  - If it exists, then:
    - ∗ Generates a new random salt RS_i^{new} and calculates a new number N_i^{new}.

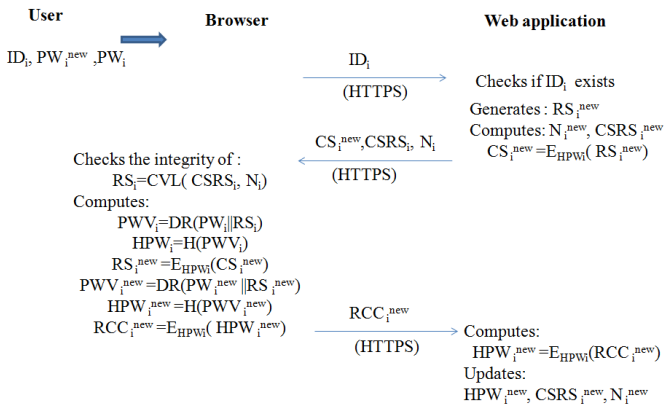

Figure 6: Renewal phase

  - ∗ Encrypts the random salt RS_i^{new} generated by the final password HPW_i of user U_i: $CS_i^{new} = E_{HPW_i}(RS_i^{new})$.
  - ∗ Sends CS_i^{new}, $CSRS_i$ and N_i to the browser.
- Otherwise, returns an error message.

- The browser:
  - Checks the integrity of $CSRS_i$ by the calculation of $RS_i = CVL(CSRS_i, N_i)$.
  - Calculates:
    - ∗ The virtual password of a user U_i by the Dynamic Rotation exercised on the concatenation of its valid original password PW_i and its random salt RS_i: $PWV_i = DR(PW_i \| RS_i)$.
    - ∗ The final password of the user U_i by hashing the virtual password PWV_i with a one-way hash function H: $HPW_i = H(PWV_i)$.
    - ∗ Decrypts the received message CS_i^{new} by the final password HPW_i calculated as a key of symmetric encryption in order to have the new random salt generated for the user U_i: $RS_i^{new} = E_{HPW_i}(CS_i^{new})$.
  - If the decryption is successful, then the server is authenticated by the client and also the confidentiality and integrity of RS_i^{new} are ensured.
  - Then, calculates:
    - ∗ The new virtual user U_i password by the dynamic rotation (DR) applied on the concatenation of its new original password valid PW_i^{new} and new random salt RS_i^{new}: $PWV_i^{new} = DR(PW_i^{new} \| RS_i^{new})$.
    - ∗ The new final password of the user U_i by hashing of the new virtual password PWV_i^{new} with an one-way hash function H: $HPW_i^{new} = H(PWV_i^{new})$.
    - ∗ The new encryption password calculated with the ancient password as symmetric encryption key: $RCC_i^{new} = E_{HPW_i}(HPW_i^{new})$.
  - Sends RCC_i^{new} as a challenge of authentication and a new final valid password to the server.

- The server:
  - Decrypts the received message in order to have the new password passes final calculated by the browser : $HPW_i^{new} = E_{HPW_i}(RCC_i^{new})$. If the decryption is successful, then, the browser is authenticated by the server. Thus, the mutual authentication of the server and of the browser is guaranteed.
  - Updates of authentication settings: HPW_i^{new}, $CSRS_i^{new}$ and N_i^{new}.

4 Security Analysis

In order to prove the degree of adaptation and robustness of our strong zero knowledge authentication proposal, we have to estimate their capacity to resist various attacks. The majority of attacks against Web applications are relied on the theft, the traceability and the weaknesses of critical data.

4.1 Defends Against Theft of Data

In companies the confidence is a range of users to preserve. More critically, the protection of data exchanged must be well protected against the theft or any other leak. In general, the space of attacks is a vast environment affects all web applications. In our proposal, we must estimate their impacts on the physical and digital security of data exchanged between the communicating entities in the following three subspaces: Client, Server and Network.

4.1.1 Client Side

In order to prevent attacks by Spyware and also to help the users who are unable to memorize of the random pass-

words. We recommend the use of ObPwd [16] that generate strong enough passwords based on the digital objects.

4.1.2 Server Side

The passwords stored in a server are strengthened by safe cryptographically random salts, are totally deformed by the dynamic rotation and their cryptographic qualities, and are assured by an one-way hash function: $HPW_i = H(DR(PW_i \| RS_i))$.

4.1.3 On the Network

In most of the existing systems of authentication passwords submission is done in plaintext or hashed. Also, for a server these passwords play the role of an original password. More critically, this process encourages attacks to sniff the network. In this situation, the attacker does not need to find the original password entered by the user. But, it suffices for him to build a script which contains the passwords intercepted on the network. Thus, to cope with this situation, we add $HTTPS$ in our proposal, and we use the passwords as key of encryption/decryption mutual authentication messages between the client and the server. Therefore, we assure the undisclosure, the untraceability and the confidentiality of the passwords that transit on the wire.

4.2 Defends Against Phishing Attacks

The phishing attack is a set of very effective attempts to data theft online. To cope with this attack, we propose the following technique:

4.2.1 A Cryptographically Safe Random Salts

To prevent the weaknesses and the problems of a salt generated from a given domain name. We propose this solution which allows to have and to verify the integrity of a random salt appropriate to each user: $CVL(CSRS_i, N_i)$. This verification can be taken to ensure the origin authentication settings. In addition, the recovery of these parameters is conditioned on the existence of a given user in order to prevent the falsification of the original sites. In this case, the attacker does not only need to create a site for acquire deceitfully the sensitive information from users, but it must answer their challenges which are impossible.

4.2.2 Mutual Authentication

In case of success of the check of the integrity of a salt. This attack rests on the hypothesis: "The password stored in the database during the recording phase will retransmit on the network". What is wrong in our proposal. The recalculated passwords never will retransmit on the wire. But, they will be used as keys of encryption of the messages of mutual authentication $E_{HPW_i}(RS_i), E_{HPW_i}(RS_i^{new}), E_{HPW_i}(Ts_i), \ldots$. For this interest, we have watched over the complexity and untraceability of generated passwords based on the cryptographic nature of regenerated random salts, dynamic rotation of binary strings generated in order to break the link between the original and virtual password and one-way hash function: $PWV_i = DR(PW_i \| RS_i)$, $HPW_i = H(PWV_i)$.

4.3 Defends Against the Shoulder Surfing

This attack is strongly related to consciousness and habits followed by users to protect their privacy especially in public spaces. But, in the case of highly sensitive web applications, we recommend the integration of the technical of password object [16]. Because, this technique allows to hide all the movements of the users and also to have very complicated passwords meeting the requirements of the computer security.

4.4 Defend Against SQL Injection

This attack presents a serious threat for the security of the dynamics of Web sites. To check well the validity and the robustness of the parameters of authentication chosen by the user. It is recommended to use the grey list and the methods of filtering (validation and cleaning) to make sure on the reliability of data. It is very effective in standard architectures which are based on a positive answer to a given request. In our proposal, we introduce a process of identification that can eliminate this problem. In reality, we propose an interactive system of authentication. More critically, the communicating entities must verify and respond to authentication challenges (for encryption / decryption nonce) to assure mutual authentication. Specifically, all responses must be confirmed by the previous challenge and accompanied by a new challenge: $Tb_i \oplus Ts_i = E_{HPW_i}(RCS_i)$, $Ts_i = Tb_i \oplus Tb_i \oplus Ts_i$, $RC_i = E_{HPW_i}(Ts_i)$. And taking into account the internal characteristic of our system, the first request allows only to verify the existence of user U_i and to get back

its own random salt. Where from, the inclusion of metacharacters in username/password fields will have no influence on the safety of user accounts. Otherwise, it will generate error messages at verification or identification of users. Consequently, our proposal resists against this attack.

4.5 Defends Against the Collisions

The proof of security of any hash functions (compression function) is measured by these capacities to resist collisions attacks (pseudo-collisions exist on the compression function in certain iteration). The domain extender algorithm defined by Merkle Damgard has known a wide range of collision attacks. The attack of extension of length which was remedied by Coron et al. [6]. Also, Joux [12] discovered the multicollision attack which looks for k internal collisions from k different messages. This vulnerability affects almost at the bottom the security of any domain extender algorithm whose internal state length equal to that cadence. But, according to Lucks [15] recommendation to remedy this problem is to increase the internal state length of the compression function to $N \geqslant 2n$ (with n is the length of the hash). In general, there are two types of attacks affecting the quality of cryptographic hash functions, namely: the probabilistic and structural attacks. In this article, we interest a improving the robustness of hash functions against probabilistic attacks. These types of attacks are based on the inability of users to choose passwords that can meet the requirements of computer security. Hence, the interest to introduce our system that is able to extend the length and to evolve the cryptographic quality of passwords. Thus, through [1] and the results obtained in Section 3.7, we deduce, on one hand, that the regenerated virtual passwords are of cryptographic nature, and on the other hand the uncorrelation, the untraceability and unpredictable of the primitive signals regenerated are assured for a weak original password. In addition, if we combine the different chosen passwords, the cryptographic quality of random salts [1], the algorithm of dynamic rotation and the robustness of an one-way hash function ($HPW_i = H(DR(PW_i \| RS_i))$ this collision problem will be actually very far.

4.6 Defends Against Man in the Middle Attack

In this technique, the attacker should be able to observe and intercept (Sniffing) the encrypted data exchanged between two victims in a valid time. It is particularly applicable in the original protocol of exchange of keys Diffie-Hellman, when it is used without authentication. In this proposed protocol, for more complexity against the attacks, we exploit the symmetric cryptographic primitives. Consequently, the attacker should intercept the connection request messages $RCC_i^{new} = E_{HPW_i}(HPW_i^{new})$ sent by a user U_i to the server and to replay the responses to the challenges of mutual authentication such as $CC_i = E_{HPW_i}(Tb_i)$,$RCS_i = E_{HPW_i}(Ts_i \oplus Tb_i)$ and $RC_i = E_{HPW_i}(Ts_i)$. But, as the attacker does not have value of HPW_i^{new}, it will be unable to replay nor connect messages nor responses to mutual authentication challenges. Thus, the resistance of our protocol is assured against this attack.

4.7 Defends Against Brute Force Attack

According to [1, 24], the resistance of the passwords against this attack is strongly bound to their complexities. The attacker should get back the file of the passwords then launch a software of brute forces "cracking" in order to test in a exhaustive way all the possible combinations of the passwords. In our proposal, the attacker does not only need to find the virtual password hashed by an one-way hash function $HPW_i = H(PWV_i)$, nevertheless, he should extract the original password which has been totally deformed in a random salt by a dynamic rotation of their concatenation $PWV_i = DR(PW_i \| RS_i)$. In addition to the cryptographic quality of the salts used [1] and according to the part 3.7, the dynamic rotation allows to break any correlation between the original and virtual passwords. Hence, we confirmed the unpredictable nature of virtual passwords generated. Therefore, the proposed protocol is secure against the attacks of brute forces.

4.8 Defends Against the Dictionary Attack

This type of attack is very effective in case of passwords with weak entropy or of authentication systems based on breakable hash functions. In our system, the cryptographic quality of the passwords is strongly bound to salts generated appropriate to every user, the dynamic rotation and the one-way hash function: $RS_i = CVL(CSRS_i, N_i)$, $PWV_i = DR(PW_i \| RS_i)$, $HPW_i = H(PWV_i)$. According to Subsection 3.6, the range of lengths of the complexity of the generated virtual passwords is $[\mathbf{2.572\ 10^{61}}, \mathbf{9.047\ 10^{74}}]$, and to Subsection 3.7,

the uncorrelation and the unpredictability of the passwords are assured for a minimal original password. Therefore, our system is actually protected against this attack.

5 Conclusion

In computer environment, the security or rather the privacy of users is the heritage of any company or organization on the wire. According to all the studies on user habits have shown their limits to meet the requirements of computer security in this discipline. In particular their incapacities to memorize random passwords. Then, they resort to habits facilitate attacks. More critical, it is impossible to rely on the users as key factors of the computer security. All these difficulties push us to the conception of a new system of authentication SAVP. This work comes in the optics to strengthen and to improve the mutual authentication of web users based on virtual passwords.

Taking account of the evolution of attacks and constraints of user systems, the cryptographic quality authentication system should not be linked to their ability to meet of the recommendations of computer security. Strongly, their consciences, their behaviors and passwords choices have very remarkable influences on the survival of their accounts. For this, web application security must be seen as an inter-connected environment requiring input from all entities constituting our system. Therefore, in our proposal, the security is required for all elements of our system. The interest is to have a system of mutual authentication based on virtual passwords capable of resisting multiple types of attacks in particular phishing, dictionary, brute force, spyware, man in the middle and replay attacks. So, we propose a strong system of authentication with zero knowledge based on:

- Salts generated by a cryptographically secure regenerator.
- An algorithm for the dynamic rotation of binary strings in order to ensure uncorrelation, unpredictability and untraceability of passwords for minimal disturbance to the initial condition.
- The symmetric cryptographic primitives for more privacy authentication settings.
- An one-way hash function, random nonce to ensure mutual authentication of communicating entities and the updating of the parameters of authentication during the renewal phase.

Generally, we can quote its characteristics as follows:

- The distribution of lengths and periods of virtual passwords generated are random and unlimited.
- The nature of virtual password generated is pseudo-random and in some situations seems chaotic.
- The untraceability and the reuse of original passwords are handled securely by integrating cryptographically secure salts algorithm and dynamic rotation of binary strings to withstand multiple types of attacks.
- The integrity of salts is assured by integration of the mechanism of CRC code of variables lengths.
- The transparency and portability of our system in all steps of executions to ensure non-occupation of the users and to hide any sensitive information can help an attacker to attack our application.
- The complexity of our system SAVP comes from the unpredictable nature of any regenerated salt.
- The simplicity in all operations building our proposal to be feasible in all programming languages.

References

[1] Y. Asimi, A. Asimi, Y. Sadqi, "New random generator of a safe cryptographic salt per session (RGSCS)," *International Journal of Network Security*, vol. 18, no. 3, pp. 445–453, May 2016.

[2] S. M. Bellovin and M. Merritt, "Encrypted key exchange: Password-based protocols secureAgainst dictionary attacks," in *Proceedings of IEEE Symposium on Security and Privacy (SP'92)*, pp. 72, Washington, DC, USA, 1992.

[3] J. Bonneau and S. Preibusch, "The password thicket: technical and market failures in human authentication on the web," in *The Ninth Workshop on the Economics of Information Security (WEIS'10)*, pp. 1–49, 2010.

[4] N. Chou, R. Ledesma, Y. Teraguchi, and J. Mitchell, "Client-side defense against web-based identity theft," in *Proceedings of Network and Distributed Systems Security (NDSS'04)*, pp. 1–16, 2004.

[5] R. M. Conlan and P. Tarasewich, "Improving interface designs to help users choose better passwords",

in *Extended Abstracts on Human Factors in Computing Systems (CHI'06)*, pp. 652–657, New York, USA, 2006.

[6] J. S. Coron, Y. Dodis, C. Malinaud and P. Puniya, "Merkle-damgard revisited: How to construct a hash function," in *Advances in Cryptology (CRYPTO'05)*, LNCS 3621, pp. 430–448, Springer-Verlag, 2005.

[7] D. Florncio and C. Herley, "A large-scale study of web password habits," in *Proceedings of the 16th ACM International Conference on World Wide Web*, pp. 657–666, 2007.

[8] P. Dourish, E. Grinter, J. D. de la Flor, and M. Joseph, "Security in the wild: User strategies for managing security as an everyday, practical problem," *Personal Ubiquitous Computing*, vol. 8, no. 6, pp. 391–401, 2004.

[9] S. Gaw and E. W. Felten, "Password management strategies for online accounts," in *Proceedings of the Second ACM Symposium on Usable Privacy and Security (SOUPS'06)*, pp. 44–55, New York, USA, 2006.

[10] S. Goldwasser, S. Micali, and C. Racko, "The knowledge complexity of interactive proof-systems," in *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing (STOC'85)*, pp. 291–304, 1985.

[11] J. A. Halderman, B. Waters, and E. Felten, "A convenient method for securely managing passwords," in *Proceedings of the 14th International World Wide Web Conference (WWW'05)*, pp. 471–479, 2005.

[12] A. Joux, "Multi-collisions in iterated hash functions. Application to cascaded constructions", in *Advances in Cryptology (CRYPTO'04)*, LNCS 3152, pp. 306–316, Springer-Verlag, 2004.

[13] L. Lamport, "Password authentication with insecure communication," *Communications of ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[14] M. Lei, Y. Xiao, S. V. Vrbsky, C. C. Li, and L. Liu, "A virtual password scheme to protect passwords," in *Proceedings of IEEE International Conference on Communications (ICC'08)*, pp. 1536–1540, 2008.

[15] S. Lucks, "A failure-friendly design principle for hash functions," in *Advances in Cryptology (ASIACRYPT'05)*, LNCS 3788, pp. 474–494, Springer-Verlag, 2005.

[16] M. Mannan and P. C. van Oorschot, *Digital Objects as Passwords*, Carleton University, Canada, July 14, 2008.

[17] B. Miller, L. Huang, A. D. Joseph, J. D. Tygar, "I know why you went to the clinic: risks and realization of https traffic analysis," in *14th International Symposium on Privacy Enhancing Technologies (PETS'14)*, pp. 143–163, 2014.

[18] R. Morris and K. Thompson, "Password security: A case history," *Communications of ACM*, vol. 22, no. 11, pp. 594–597, 1979.

[19] G. Notoatmodjo and C. Thomborson, "Passwords and perceptions," in *Seventh Australasian Information Security Conference (AISC'09)*, pp. 71–78, Wellington, New Zealand, 2009.

[20] N. Ojha and S. Padhye, "Cryptanalysis of multi prime RSA with secret key greater than public key", *International Journal of Network Security*, vol. 16, no. 1, pp. 53–57, Jan. 2014.

[21] A. Prakash, "A biometric approach for continuous user authentication by fusing hard and soft traits", *International Journal of Network Security*, vol. 16, no. 1, pp. 65–70, Jan. 2014.

[22] S. Riley, "Password security: What users know and what they actually do," *Usability News*, vol. 8, no. 1, pp. 2833–2836, 2006.

[23] B. Ross, C. Jackson, N. Miyake, D. Boneh, J. C. Mitchell, "Stronger password authentication using browser extensions", in *Proceedings of Usenix Security*, pp. 17-32, 2005.

[24] T. Rowan, "Password protection: The next generation," *Network Security*, vol. 2009, no. 2, pp. 4–7, Feb. 2009.

[25] A. Rukhin, et al., *A Statistical Test Suite for Random and Pseudorandom Number Generators For Cryptographic Applications*, NIST Special Publication 800-22, Apr. 2010.

[26] Y. Sadqi, A. Asimi, A. Younes, "Short: A lightweight and secure session management protocol", in *The Second International Conference of NETworked sYStems (NETYS'14)*, LNCS 8593, pp. 319–323, Springer-Verlag, 2014.

[27] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, "Encountering stronger password requirements: User attitudes and behaviors," in *Proceedings of the Sixth ACM Symposium on Usable privacy and Security (SOUPS'10)*, Article no. 2, 2010.

[28] S. Singh, A. Cabraal, C. Demosthenous, G. Astbrink, and M. Furlong, "Password sharing: Implications for security design based on social practice," in *Proceedings of ACM SIGCHI Conference on Human Factors*

in Computing Systems (CHI'07), pp. 895–904, New York, USA, 2007.

[29] S. K. Sood, A. K. Sarje, and K. Singh, "Inverse cookie-based virtual password authentication protocol", *International Journal of Network Security*, vol. 13, no. 2, pp. 98–108, Sept. 2011.

[30] B. Tanti, N. Doshi, "A secure email login system using virtual password," Cryptology ePrint Archive, Report 1009.5729, 2010. (`http://eprint.iacr.org/2010/481.pdf`)

[31] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password authentication schemes: Current status and key issues", *International Journal of Network Security*, vol. 3, no. 2 , pp. 101–115, Sept. 2006.

[32] D. Weirich and M. A. Sasse, "Pretty good persuasion: A first step towards effective password security in the real world," in *Proceedings of the 2001 ACM Workshop on New Security Paradigms*, pp. 137–143, New York, NY, USA, 2001.

Younes ASIMI received his Master's degree in Computer Science and Distributed Systems in 2012 from Departments of Mathematics and Computer Sciences, Faculty of Science, University Ibn Zohr, Agadir,Morocco. He is currently pursuing Ph.D in Departments of Mathematics and Computer Sciences, Information Systems and Vision Laboratory,Morocco. His research interests include Authentication Protocols, Computer and Network Security and Cryptography.

Abdallah AMGHAR is a Professor in the Physics Department, Faculty of Science, University Ibn Zohr,Morocco. He received his DEA and DES degree in 1994 from Department of Physics, Faculty of Science, University Hassan II, Morocco. In January 2002, he has Ph.D degree in microelectronic from Department of Physics, Faculty of Science, University Ibn Zohr, Morocco. His areas of research interests include Cryptography, DNT, embedded systems and microelectronic.

Ahmed ASIMI received his PhD degree in Number theory from the University Mohammed V - Agdal in 2001. His research interest includes Number theory, Code theory, and Computer Cryptology and Security. He is a full professor at the Faculty of Science at Agadir since 2008.

Yassine SADQI received his Master in the field of Computer Science and Distributed Systems at the Ibn Zoher University in 2012. He is currently a Ph.D. candidate of the Ibn Zoher University, Agadir, Morocco. His main field of research interest is computer security, cryptography and authentication in Web applications.