New Random Generator of a Safe Cryptographic Salt Per Session

Younes Asimi[1], Abdallah Amghar[2], Ahmed Asimi[1], and Yassine Sadqi[1]
(Corresponding author: Ahmed Asimi)
Departments of Mathematics and Computer Science & Information Systems and Vision Laboratory, Ibn Zohr University[1]
Department of Physic & Information Systems and Vision Laboratory, Ibn Zohr University[2]
B. P. 8106, City Dakhla, Agadir, Morocco.
(Email: *asimiahmed2008@gmail.com*)
(Received Nov. 14, 2013; revised and accepted Sept. 5 & Dec. 16, 2014)

Abstract

Nowadays, client authentication in Web applications for each user based on passwords and a statically salts [11, 13, 18, 19]. The aim of this article is to propose random generator of a safe cryptographic salt per session ($RGSCS$). The interest to introduce this regenerator is to contribute to the evolution of the cryptographic quality of the systems of strong zero knowledge authentication based on passwords. In Section 3, we propose a model for regeneration a SOTS based on random functions and on CRC code. To study the behavior of the $RGSCS$, which is the objective of Section 4, we have, in one hand, defined and proved a metric on the finite set of periodic binary sequences not necessarily the same period, the uncorrelation, the impact of the distribution of lengths and the unpredictability of primitive signals and in the other hand, evaluated the performance of our purpose by using several tests. The outcome showed that $RGSCS$ has a chaotic behavior. As for Section 5, is devoted to the implementation of our $RGSCS$ algorithm under PHP5. This article is finished by a conclusion.

Keywords: *CRC code, passwords, random generator RGSCS, safe one time salt, strong zero knowledge authentication*

1 Introduction

Design methods of passwords are the first authentication techniques in the web, which is based in one hand on hash functions for example MD5 [23] (complete collisions) and $SHA-1, 2$ [17, 22] (theoretical collisions) and in the other hand on statically salts.

The objective of this paper is to improve the authentication mechanism by preposition and behavioral study of a new model that regenerates a safe one time salt for each session successfully connected. This system is based on pseudo-random functions and the error detection code (CRC) [21] with a variable length to ensure the integrity of the generated binary sequences.

This paper is organized as follows: In Section 3, we propose a design of a new model to regenerate a safe one time salt ($RGSCS$) composed by three processes. The first process consists to regenerate the one time salt from random functions defined in PHP, denoted by OTS. To increase the security level of OTS, which is the goal of the process II, we apply the CRC of variable lengths on primitive signal associate to OTS for regenerate a safe one time salt, denoted by SOTS. Hence the authentication parameter for each user is ($SOTS$, N) where N is the number of bits equal to one in a primitive signal of SOTS. The process III consists to check the integrity of SOTS for each attempt to connect and to update the authentication parameter after successful connection. The Section 4 studies the behavior of $RGSCS$. Therefore we define and prove a metric on the finite set of periodic binary sequences not necessarily the same period. And we finished this section by the evaluation the performances of $RGSCS$ by using several tests according the length, period and distribution of primitive signals of $SOTS$. As for the fifth section, we realized an implementation of our $RGSCS$ algorithm which reassures its cryptographic nature and its capacity to detect any unexpected perturbations of OTS and the some conclusion are draw in final section.

In this section, we introduce the notations that will be used throughout this paper in Table 1.

2 Related Work

The concept of salts was introduced by Morris and Thompson [11] as another alternative of one time passwords OTP to ensure the security password on $UNIX$. They are based on storing passwords salted and hashed to reduce the risk of password file compromise [1]. We also underline that several extensions have been proposed to

Table 1: Notations

$\mathbb{N}$:	Set of natural numbers.
$\mathbb{R}$:	Set of real numbers.
F_K:	Set of periodic binary functions of same period K.
Γ:	Set of periodic binary functions not necessarily the same period.
$L(S)$:	Length of binary sequence S.
$S_P(F)$:	Primitive signal of binary function F.
$P(x)$:	Probability of event x.
$Lmc(K_1, \ldots, K_r)$:	Lowest common multiple of positive integers $K_1, \cdots, K_r$.
CRC:	Cyclic Redundancy Check.
$SOTS$:	Safe One Time Salt.
OTS:	One Time Salt.
$NIST$:	National Institute of Standards and Technology.
$\ll$:	Inferior.
$\gg$:	Superior.

evolve the security of the password against multiple attacks specifically against Phishing and Spyware attacks. The technical of SpoofGuard [3] 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 [7] 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, the technique PwdHash [13] was developed for Internet Browsers Explored and Mozilla Firefox. It allows to evolve the security of the passwords in the 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 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). In addition, neither the robustness and nor the integrity of this salt are verified. Indeed, this salt allows to extend the length of passwords chosen by the user. Yet, it is incapable to touch at the bottom the cryptographic quality of the passwords. More critical, for the users who have the same original passwords will have the same final password. In general, all the 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 [2, 4, 6, 12, 20]. It is necessary to note also that numerous studies on the JavaScript attacks showed that the implementation in complete safety of the hashing in the browser is rather difficult on the modern Web applications [9].

At that time, the $HTTPS$ protocol was the only way to ensure the confidentiality and the integrity of data which transit on the network. But, thanks to an analytical study made by American researchers [10], the monitoring of the Web traffics leaves enough information even if the data which transit are encrypted. However, the security of the authentication systems based on the passwords represents a big challenge to the development of the digital enterprises. The interest to introduce this $RGSCS$ regenerator is to contribute to the evolution of the cryptographic quality of the passwords to meet the requirements of the IT security and also push aside the limits and the concerns of the users which are unable to maintain complex passwords. In our proposal, following to the cryptographic nature of the OTS, it is almost impossible to find the same final password for two users with the same original passwords.

3 RGSCS Algorithm

A salt is a safe one time (SOTS) if it's specific for each user session, regenerated by a pseudo-random and unfalsifiable regenerator.

- **Specific to each session:** After the opening of each session a new salt will be regenerated. Therefore, the decrease in the probability of attacking users.
- **Pseudo-random:** Its aim is to produce dynamics OTS with uncorrelated primitive signals.
- **Unfalsifiable:** The regenerated binary sequences are protected with a mechanism for errors detecting CRC with variables lengths to check their integrity.

We refer to [11, 13, 18, 19], to get the following results:

- **A global salt:** Consists to add the only salt for all sites and for all users (equivalent to the domain name of remote site). This is easy technique to perform.

Furthermore, this salt is not secret, which explains that the use of this technique is just for increase the complexity of time. Because only one dictionary necessary to attack all members of the site.

- **One salt for each user:** This technique is similar to the previous one. Except in this case, we have a user-specific salt. This is the most common technique used so far due to the following factors: The simplicity of programming and the level of protection against dictionary attacks.

- **A salt per session:** This is a technique requires the handling of twice salts: A global salt and a salt regenerated for each session. A global salt used for password deformation to register before you encrypt with a cryptographic hash function. The other salt is used to protect all stored passwords. This technique is very difficult to implement yet.

In all cases, the regeneration of these salts based on a random strings or on a random number generated by the function rand(). Also, the implementation of these techniques is based on AJAX and JavaScripts that generate the following drawbacks [9, 21].

- **The function Rand():** Uses a linear congruential regenerator and generates a sequence of integers. Hence, the interval of numbers introduced by this function is limited. In fact, we can test all possible numbers with a simple script.

- **Scripts AJAX:** Checks the existence of an identifier of a user after each entry of a character in the login field to return the salt that is transmitted in clear text. This facilitates dictionary attacks and brute force attacks.

- **The JavaScripts:** The client-side security is not assured in spite of the use of CryptJS.

To remedy the problems of static salt and salt per session, we propose a new conception of random generator of a safe cryptographic salt ($RGSCS$). This algorithm allows, from three functions: Rand(), Microtime() and mcrypt_create_iv(), to regenerate a safe one time salt. It consists of three processes. The first aims to regenerate a dynamic salt for each successful connection. The second applies the CRC of variable lengths (that we call CVL) on primitive signal associate to OTS for regenerate a safe and one time salt ($SOTS$). The third checks the integrity of SOTS and updates the authentication parameters ($SOTS$, N).

3.1 Process I

The main objective of this process is the regeneration of OTS, by using three functions Rand(), Microtime() and mcrypt_create_iv(), as follows:

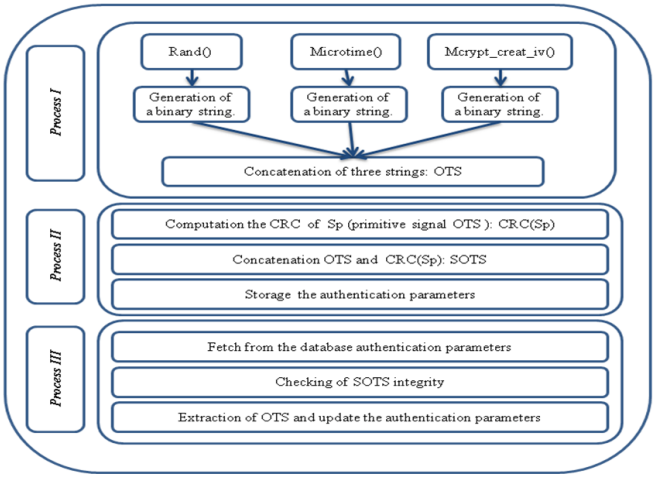

Figure 1: RGSCS algorithm

- Let S, R and T be three strings regenerated respectively by Rand(), Microtime() and mcrypt_create_iv().

- Let $S2$, $R2$ and $T2$ be three binary representations of S, R and T respectively.

- OTS is the concatenation of $S2$, $R2$ and $T2$.

- OTS is seen as a concatenation of primitive signals.

3.2 Process II

This process improves the security level of any dynamics salt created in process I, specifically the OTS integrity, as flows:

- Let S_P be the primitive signal of OTS and G a polynomial generator of CRC.

- K is the number of S_P bits set to 1.

- $M = max(K, strelen(S_P) - K)$.

- $N = strelen(S_P) moduloM$.

- The polynomial G is associated to binary representation of N.

- Compute $R = CRC(S_P)$.

- $SOTS$ is the concatenation of S_P and R.

- Store $SOTS$ and N in database.

- The authentication parameters per session are SOTS and N.

3.3 Process III

This process occurs for each new connection. It builds on the previous two processes. It will verify the integrity of $SOTS$ through each attempt to connect and update authentication parameters $(SOTS, N)$ after each successful connection. For that we proceed as follows:

- Fetch the authentication parameters $SOTS$ and N.
- Compute G associate to N.
- Check of $SOTS$ integrity.
- If this verification is successful, then we deduce OTS of $SOTS$.
- Otherwise the validation is failed.
- In the favorable case, we use the previous two processes to update $SOTS$ and N.

4 Behavioral Study of $RGSCS$ Algorithm

To estimate the complexity of the $RGSCS$ algorithm, a behavioral study is dedicated to analysis of the generated primitive signals. However, the testing of these classes of binary functions shows that not necessarily the same period. Hence, the difficulty of computing their Hamming distances and analyze the results. Therefore, we are reduced to define and prove a distance which is an extension of a Hamming distance of sets of periodic strings that are not necessarily the same period.

4.1 Metric on the Set of Periodic Binary Strings

From [15, 16], we deduce some results:

Definition 1. *We call a binary function, all function defined from $\mathbb{N}$ into $\{0,1\}$.*

Definition 2. *For each binary function F, we associate the only binary string f defined by $f = F(0)\ F(1)\ F(2)\ \cdots\ F(n)\ \cdots$. And if there is an integer k such that $f = F(0)\ F(1)\ F(2)\cdots F(k-1)\ F(0)\ F(1)\ F(2)\cdots$, therefore F is periodic with period k, and if more k is the smallest integer, then the sequence $F(0)\ F(1)\ F(2)\cdots F(k-1)$ is called primitive signal of f, which denotes by $S_P(F)$. In this case, $F(n) = F(n \bmod L(S_P(F)))$ for all $n \in \mathbb{N}$.*

And if f is a finite sequence, we extended to a unique periodic infinite sequence with a length of its primitive signal is a divider of $L(f)$.

We call regenerative signal of F, that we denote by $S_R(F)$, a concatenation of the its primitive signal.

Definition 3. *Let S and S' be two elements of F_K. S and S' are equal and we denote $S = S'$ if and only if $S(n) = S'(n)$ for all $n \in \mathbb{N}$.*

Theorem 1. *Let S and S' be two elements of F_K. The following conditions are equivalent:*

1) $S = S'$.

2) $S_P(S) = S_P(S')$.

4.1.1 Metric on the Finite Set of Periodic Binary Sequences of Same Period

In this section, we focus on the definition of the distance between binary sequences with same period K.

Definition 4. *[16] A metric space is a nonempty set E together with a function d called a metric, denoted by (E, d).*

Definition 5. *[16] Let E be a metric space. The metric d on E is a function defined from $E\times E$ into $\mathbb{R}^+$ and satisfied the following axioms for all x, y, z in E:*

1) $d(x,y)\geq 0$ et $d(x,y) = 0 \Longleftrightarrow x = y$.

2) $d(x,y) = d(y,x)$.

3) $d(x,y)\leq d(x,z) + d(z,y)$.

Lemma 1. *[16] Let S, S' and S'' be three elements of F_K. We consider the following sets:*
$T = \{i \in \{0,...,J-1\}/S(i) \neq S'(i)\}$,
$H = \{i \in \{0,...,J-1\}/S(i) \neq S''(i)\}$
and $G = \{i \in \{0,...,J-1\}/S''(i) \neq S'(i)\}$.
we have $T \subset H \cup G$.

Proposition 1. *Let S and S be two elements of F_K. The function D:*

$$\begin{array}{rccc} D & : & F_K\times F_K & \longrightarrow & \mathbb{N} \\ & & (S, S') & \longmapsto & \displaystyle\sum_{i=0}^{K-1}((S(i)+S'(i))\%2). \end{array}$$

Proof. We have $D(S,S') = \displaystyle\sum_{i=0}^{K-1}((S(i)+S'(i))\%2) \geq 0$ for all $(S,S') \in F_K^2$.

$$\begin{array}{rcl} D(S,S') = 0 & \Leftrightarrow & \displaystyle\sum_{i=0}^{K-1}((S(i)+S'(i))\%2) = 0 \\ & \Leftrightarrow & (S(i)+S'(i))\% = 0\ \forall i \in \{0,...,K-1\} \\ & \Leftrightarrow & S(i) = S'(i)\ \forall i \in \{0,...,K-1\} \\ & \Leftrightarrow & S = S'\ (Theorem\ 1). \end{array}$$

$$\begin{array}{rcl} D(S,S') & = & \displaystyle\sum_{i=0}^{K-1}((S(i)+S'(i))\%2) \\ & = & \displaystyle\sum_{i=0}^{K-1}((S'(i)+S(i))\%2) \\ & = & D(S',S). \end{array}$$

We have $T \subset H \cup G$ (Lemma 1).

$$\begin{aligned} D(S,S') &= \sum_{i=0}^{K-1}((S(i)+S'(i))\%2) \\ &= \sum_{i\in T}((S'(i)+S(i))\%2) \\ &\leq \sum_{i\in H}((S(i)+S''(i))\%2) \\ &\quad +\sum_{i\in G}((S''(i)+S'(i))\%2) \\ &\leq D(S,S'')+D(S'',S'). \end{aligned}$$

Therefore D is a metric on F_K. □

4.1.2 Metric on the Finite Set of Periodic Binary Sequences not Necessarily the Same Period

In this section, we denote by Γ a finite set of periodic binary sequences, not necessarily the same period and the lowest common multiple of their periods.

Proposition 2. *The function D': $\Gamma\times\Gamma\to[0,1]$ defined by:*

$$D'(S,S') = \frac{\sum_{i=0}^{T-1}((S(i)+S'(i))\%2)}{T}$$

is a distance on Γ.

The proof of this proposition is similar to the proof of Proposition 1.

Corollary 1. *Let S and S' be two elements of Γ of periods k and k' respectively and $K = Lmc(k,k')$.*

The function D': $\Gamma\times\Gamma\to[0,1]$ defined by:

$$D'(S,S') = \frac{\sum_{i=0}^{K-1}((S(i)+S'(i))\%2)}{K}$$

is a normalized distance of Γ.

Proof. It suffices to see that:

$$\sum_{i=0}^{T-1}((S(i)+S'(i))\%2) = \frac{\sum_{i=0}^{K-1}((S(i)+S'(i))\%2)}{K}\times T$$

Thus, from proposition 2, we deduce that:

$$D'(S,S') = \frac{\sum_{i=0}^{K-1}((S(i)+S'(i))\%2)}{K}.$$

□

Definition 6. *[16] A square matrix $H=(d_{ij})$ is called metric matrix if it satisfies the following properties:*

1) $d_{ij} = d_{ji}$ for all i and j (symmetric).
2) $d_{ij} = 0$ for all $i = j$ (diagonalized).
3) $d_{ij} \geq 0$ for all $i \neq j$.

Proposition 3. *For all S and S' in Γ, the normalized distance D' satisfies the following equality:*

$$D'(S,S') = 1 - D'(S,\overline{S'}).$$

Proof. Let M be the cardinal number between two bits strings S and S' such that have the same period. Then we get:

$$M = \sum_{i=0}^{K-1}((S(i)+S'(i))\%2).$$

Hence

$$D'(S,S') = \frac{M}{K}$$

or

$$\begin{aligned} \sum_{i=0}^{K-1}((S(i)+\overline{S'(i)})\%2) &= K - \sum_{i=0}^{K-1}((S(i)+S'(i))\%2) \\ &= K - M. \end{aligned}$$

Then

$$D'(S,\overline{S'}) = \frac{K-M}{K} = 1 - \frac{M}{K}.$$

We also deduce

$$D'(S,S') = 1 - D'(S,\overline{S'}).$$

□

Definition 7. *Two binary strings of the same period are called strongly correlated if the knowledge of one, within reasonable time, determines the other. The opposite case, they are said to be weakly correlated.*

Definition 8. *We say that two binary strings S and S' of the same period K are weakly correlated if:*

$$D'(S,S') \simeq D'(S,\overline{S'}).$$

Propriety 1. *For all S and S' in Γ, we say that two binary strings are weakly correlated.*

The proof of this proposition relies on Proposition 3 and on Definition 6.

Corollary 2. *If $D'(S,S') \ll 0.5$, we say that S and S' are highly correlated. If $D'(S,S') \gg 0.5$, then $D'(S,\overline{S'}) \ll 0.5$, we say S and $\overline{S'}$ that are highly correlated.*

Proposition 4. *Let $S_{m,N}$ be the set of binary strings such that its waist is between m and $m+N$.*

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

Proof. We know that the number of binary strings of length k is 2^k, therefore:

$$\begin{aligned} \#S_{m,N} &= \sum_{k=m}^{m+N} 2^k \\ &= 2^m \sum_{S=0}^{N} 2^S \\ &= 2^m(2^{N+1}-1). \end{aligned}$$

We hence get Proposition 4. □

4.2 Behavioral Study of RGSCS Algorithm

After explaining the principals and the advantages of each component process of the *RGSCS* algorithm, a behavioral study dedicates to highlight its characteristics: The distribution of lengths of primitive signals and distances of the regenerated binary sequences.

4.2.1 The Lengths Distribution of Primitive Signals

In this section, we study the components functions of *RGSCS* according to the lengths of their primitive signals for one hundred, two hundred and three hundred iterations.

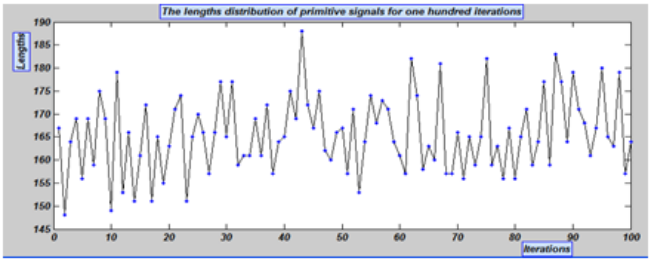

Figure 2: The lengths distribution of primitive signals for one hundred iterations

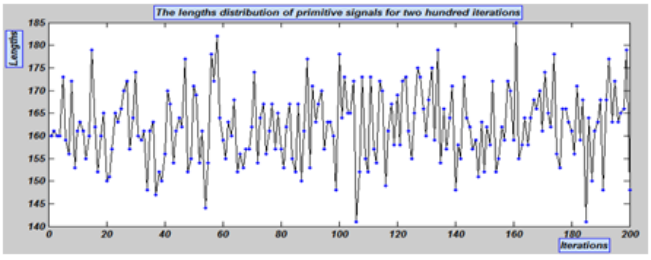

Figure 3: The lengths distribution of primitive signals for two hundred iterations

From the Figures 2, 3, 4 and Proposition 4, we deduce that the lengths distribution of primitive signals generated is random and unpredictable over time. The range of lengths of sequences is enough large and more subtle (between 140 and 185 bits).

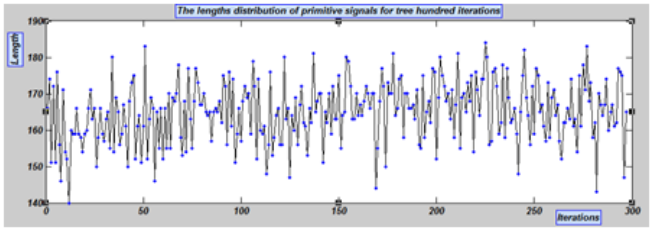

Figure 4: The lengths distribution of primitive signals for three hundred iterations

4.2.2 The Distances Distribution Between Binary Sequences

In this section, we examine the distribution of standardized three classes of distance sequences, a class of one hundred, two hundred and three hundred observers by computing the normalized distance between these sequences. Let S_i and S_j be tow elements of a given class. Set $d_{ij} = D(S_i, S_j)$ for $i, j \in \{1, ..., m\}$. The symmetric square matrix $(d_{ij})_{1 \leq i,j \leq m}$ called distance matrix for its class.

The analysis of Hamming distance matrix [8] associated to each given class will give an estimation of the complexity, correlation and coverage of its sequences. The above figures show the histograms of distance matrix of three classes: One hundred, two hundred and three hundred iterations.

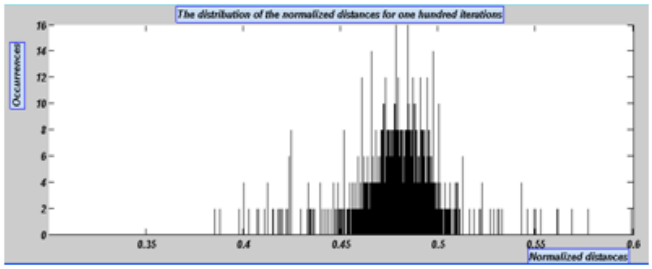

Figure 5: The distribution of normalized distances for one hundred iterations

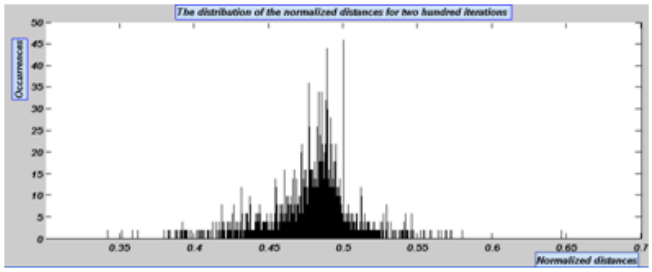

Figure 6: The distribution of normalized distances for two hundred iterations

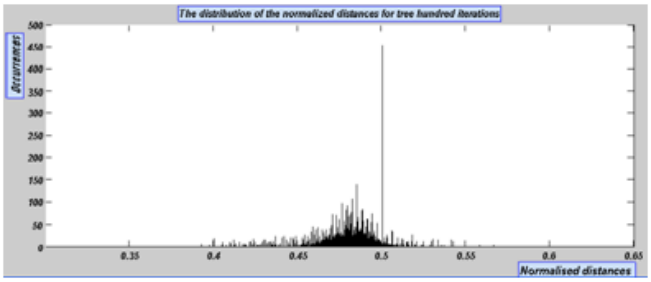

Figure 7: The distribution of normalized distances for three hundred iterations

From these histograms (5, 6, 7), we can divide regions of interest in three periods:

- From 0.35 to 0.45: in this portion, the distribution of the normalized distances phenomenon seems chaotic.
- Between 0.45 and 0.52: In this portion, we have an accumulation of normalized distances. But with a distribution seems a bit like Gaussian curve followed by small peaks. So, we do have the not correlation of generated primitive signals able to withstand the collision problem.
- Between 0.52 and 0.6: almost the same as the first portion.

The results obtains are almost identical in all three histograms. The only difference is the apparition of a peak nearest 0.5 percent for the three hundred iterations. This is normal because we have normalized between binary functions and the theory of distances required to have this peak. Hence, our purpose has unpredictable characteristics, witch is recommended by $NIST$ [14]. This enables us to ensure the cryptographic nature of the $RGSCS$ algorithm. Finally, we can summarize these features as follows:

- The distribution of lengths and periods are random.
- The primitive signals are unpredictable.
- The integrity of all OTS is provided by CRC of variable lengths.

5 Implementation of RGSCS Algorithm

Our $RGSCS$ algorithm can be executed in different types of authentication system especially banking systems and Web applications, more generally, in all the systems of cyberspace. We aim, in this work, to evolve the cryptographic quality passwords against various types of attacks. In particular, the attacks which found on the usurpation of the private data during their transmissions or their storages or on the limits of the users related to choices, memorization and storage of the passwords [1, 5]. The robustness of an authentication system is the measure of its ability to deal with all vulnerabilities, to resist against various types of attacks degrading the level of security and also to innovate an authentication system that meets the limits user. Thus, according to the theoretical and behavioral study of our $RGSCS$ algorithm, the cryptographic quality of the primitive signals regenerated is assured. Likewise, the originality and validity of any regenerated salt is provided to avoid any falsifications or perturbations unexpected of OTS primitive signals during execution. The execution of our model is done in a transparent manner. Furthermore, the portability is ensured to facilitate the movement of the internet users to a specific browser (Portability of authentication system) and avoid the risks related with the problems of storing sensitive data on the client side. For greater security, the integrity of salts exchanged between the communicating entities is also insured by the integration of a technique of errors detection CRC of variables lengths which adapts itself with all polynomials generator regenerated during any session. The interest to introduce this control mechanism of integrity aims at avoid the problem of collision of code CRC of fixed length (two primitive signals giving the same checksum), also, to meet the needs of our architecture which regenerates polynomials generator of the variables lengths.

The implementation of our proposed scheme to regenerate safes one time salts $SOTS$ specific any session opened by a user. The regenerated salts cannot be guess by the previous values. They are unfalsifiable, uncorrelated random and unpredictable.

In the following example, we have regenerated three safe one time salts by using the programming language PHP.

We aim by this work to evolve at the authentication systems based on the virtual passwords. For this interest, we have checked during the conception of our $RGSCS$ algorithm on the cryptographic quality and the integrity control of salts OTS regenerated. A priori, this mechanism is designed to preserve the validity of salts against any modifications or perturbations unexpected. Figure 8 shows three safe one time salt regenerated for three different successive sessions.

- **The binary representation of one time salts OTS:** It is the binary representation of the salts regenerated. According to these results, the one time salts regenerated are neither periodic nor the same length.
- **The real representation of one time salts OTS:** It is the $ASCII$ code representation of the primitive signals of any salt regenerated. The chain of the characters returned consists of very difficult random characters which can be memorized or guessed. They exceed the capacity of encoding information of the browsers. For this, we rewrote the characters in hexadecimal seen that the most supported by mod-

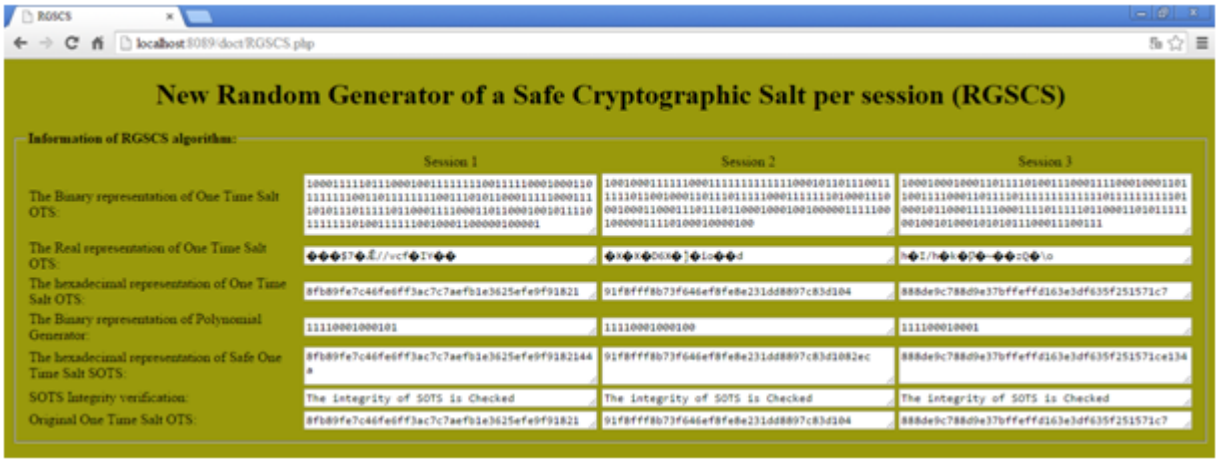

Figure 8: Implementation of RGSCS algorithm

ern browsers. This result reassures once again the cryptographic quality of our regenerator.

- **The hexadecimal representation of one time salts** OTS**:** It is the hexadecimal representation of primitive signals of any salt regenerated.
- **The binary representation of polynomials generator:** It is the binary representation of the polynomials generator which will be used to calculate the cyclic checksum CRC specific to any primitive signal OTS and to verify their integrity.
- **The hexadecimal representation of safe one time salts SOTS:** It is concatenation of the one time salt and its calculated cyclic checksum CRC.
- **SOTS integrity verification:** The primitive signals specific to any salts can undergo to falsifications or perturbations unexpected during their transitions on the thread. For this, we have to reapply this mechanism of errors detection CRC on the $SOTS$ primitive signal. If the cyclic checksum CRC is zero, then the integrity of $SOTS$ is checked, otherwise, the validity of $SOTS$ has been altered.
- **Original one time salts** OTS**:** If the integrity verification of $SOTS$ is successful, then, we deduct the original primitive signal. For this, we should remove the cyclic checksum CRC of SOTS. This OTS will be used for the regeneration of a new virtual password.

Mathematically, the CRC code is a surjective function, which means we can have the same checksum for several different primitive signals. Whence, an attacker can seek to change a primitive signal in order to have same checksum without need to modify the polynomial generator. But, further to the dynamic cryptographic nature of the $SOTS$ and to the polynomials generative which depend on the complexity of the SOTS, this attack remains very distant especially for the most connected users.

6 Conclusion

Awareness about the impact of computer sciences security on the quality of applications and websites, has leaded us to the development of a new $RGSCS$ algorithm. This paper has come in order to strengthen and improve user authentication based on passwords and a safe one time salts.

Certainly in terms of security, authentication, integrity, simplicity, predictability, transparency and complexity all play an important role. Subjectively, our purpose based on simple and programmable operations in most programming languages. Hence, we associate a random primitive signal to a salt. Then there are almost impossible to divine its through successive iterations. And to make sure of their integrities, we adopt an error detection code mechanism CRC that can adapt with all polynomials generator. However, $RGSCS$ algorithm is able to detect any changes on any primitive signal constituting its salt. Thus, we develop new authentication architecture that can completely deform the password or digital signatures in general and improve the level of security against multiple types of attacks: dictionary, brute force, phishing, collision, spyware, and rainbow table attacks. Finally, we summarize its characteristics as follows:

- The length and period of generated binary sequences are random.
- The nature of the generated primitive signal is pseudo-random and in some situations seems chaotic.
- The integrity of OTS is ensured by integration mechanism CRC error detection of variables lengths.
- The complexity of the $RGSCS$ algorithm comes from the unpredictable nature of any generated primitive signal. However, for an attack, the divination of the following strings becomes very complicated or impossible.

References

[1] 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*, pp. 1–48, 2010.

[2] D. Boyd, *Answers to Questions from Twitter on Teen Practices*, Technical Report, Apophenia, 2009.

[3] 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.

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

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

[6] S. Gaw and E. W. Felten, "Password management strategies for online accounts," in *Proceedings of the Second Symposium on Usable Privacy and Security*, pp. 44–55, 2006.

[7] 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*, pp. 471–479, 2005.

[8] R. W. Hamming, "Error-detecting and error-correcting codes," *Bell System Technical Journal*, vol. 29, no. 2, pp. 147–160, 1950.

[9] Matasano, Javascript Cryptography Considered Harmful, 2011. (`http://www.matasano.com/articles/javascript-cryptography`)

[10] B. Miller, L. Huang, A. D. Joseph, and J. D. Tygar, *I Know Why You Went to the Clinic: Risks and Realization of https Traffic Analysis*, Technical Report, arXiv preprint arXiv: 1403.0297, 2014.

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

[12] G. Notoatmodjo and C. Thomborson, "Passwords and perceptions," in *Seventh Australasian Information Security Conference*, pp. 71–78, 2009.

[13] B. Ross, C. Jackson, N. Miyake, D. Boneh, and J. C. Mitchell, "Stronger password authentication using browser extensions," in *Proceedings of the 14th conference on USENIX Security Symposium (SSYM'05)*, vol. 14, pp. 2, 2005.

[14] A. Rukhin, et al., *A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications*, Technical Report, NIST Special Publication - 800-22 (Revision 1a), 2010.

[15] A. Sabour and and A. Lbekkour A. Asimi, "Genetic regenerator of pseudo-random sequences RA NMJ," *International Journal of Computer Science and Network Security*, vol. 7, no. 1, pp. 594–597, 2007.

[16] A. Sabour, A. Asimi, and A. Lbekkouri, "The three states functions: Theoretical foundations and estimated complexity," in *The 3rd International Conference on Information Technology*, pp. 1–9, 2007.

[17] S. K. Sanadhya and P. Sarkar, "New collision attacks against upto 24-step sha-2," in *Advances in Cryptology (Indocrypt'08)*, LNCS 5365, pp. 91–103, Springer, 2008.

[18] D. Seguy and P. Gamache, *Security PHP 5 et MySQL*, Eyrolles, 2007.

[19] C. Shiflett, *Essential PHP Security*, O'Reilly, 2005.

[20] S. Singh, A. Cabraal, C. Demosthenous, G. Astbrink, and M. Furlong, "Password sharing: Implications for security design based on social practice," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 895–904, 2007.

[21] W. Steinmetz and B. Ward, *PHP Clés En Main*, 76 scripts efficaces pour enrichir vos sites web, 2008.

[22] X. Wang, Y. L. Yin, and H. Yu, "Finding collisions in the full SHA-1," in *Advances in Cryptology (Crypto'05)*, LNCS 3621, pp. 17–36, Springer, 2005.

[23] X. Wang and H. Yu, "How to break MD5 and other hash functions", in *Advances in Cryptology (EUROCRYPT'05)*, LNCS 3494, pp. 19–35, Springer, 2005.

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.