التعريف الشخصي باستخدام تقنية (hash) الصورية

د. عبدالرحمن حامد الحسيني* د. نضال السيد** اخلاص عباس **

المستخلص

هناك عوامل بشرية عديدة تؤثر سلبا على أنظمة الحماية وبضمنها امنية تحقيق عضوية المستفيد. ان العوامل البشرية تتضمن، اولا ان الاشخاص بطيئين وغير معول عليهم وغير قادرين على التعامل مع كلمات المرور، وثانيا ان قابلية الاشخاص محدودة بتذكر كلمة المرور المستخدمة في التعريف الشخصي.

في هذا البحث تم التطرق الى نقاط الضعف الاساسية في اساليب التعريف الشخصي المعتمدة على المعرفة وكيفية تحسين استخدام وأمنية انظمة التريف الشخصي باستخدام (hash Visualization) بديلا عن سلسلة الاحرف مع الصور المهيكلة (structured). ان الحل الريادي المقترح لتقنية (hash Visualization) هو مولد الصور العشوائي (RIG). لقد تم بناء (RIG) بتحويل سلسلة الاحرف (التي لها معنى باستخدام صيغة رياضية عشوائية، من خلال قيمة اللون لكل (Pixel).

ان الصور المولدة يمكن تمييزها ولكن لا يمكن وضعها للآخرين.

ان طريقة التعريف المقترحة تعتمد على التمييز بدلا عن الاستدعاء. تم فحص متطلبات نظام التعريف المعتمدة على التمييز وتم اقتراح النظام (SIAS) والذي من خلاله يتم تعريف المستفيد (المستخدم) من خلال الصور السابقة له.

ان نظام (SIAS) قد حسن الامنية بسبب اعتماد التعريف على التمييز وكذلك كان النظام اكثر اعتمادية (معول عليه) وسهل الاستخدام مقارنة مع الطرق الاخرى. اضافة الى ميزة منع المستخدمين من اختيار كلمات المرور الضعيفة وتقليل صعوبات كتابة كلمات المرور ومشاركتها مع الاخرين.

* وزارة التعليم العالي والبحث العلمي

** الجامعة التكنولوجية

The SIAS has the advantage that the authentication task is shown to be more reliable, easier to use. In addition, the system prevents the users from choosing weak passwords and makes it difficult for users to write passwords down and to communicate them to others. A number of possible attacks against the SIAS are identified and implemented, which strengthen the system and improve its usability.

References

1.Scott E Umbaugh. "Computer Vision And Image Processing", A Simon And Schuster, 1998.

2.Seberry. J. And Pieprzyk. J., "Cryptography, And Introduction To Computer Security", Prentice Hall Of Australia Ptd, Ttd, 1989.

3.Rachna Dhamija. Hash Visualization In User Authentication. In Proceedings Of The Computer Human Interaction 2000 Conference, April 2000.

4.Kenneth R. Boff, Lloyd Kaufman, And James P. Thomas. "Handbook Of Perception And Human Performance". John Wily And Sons, 1989.

5.Internet Communication "Authentication, Hash Function And Digital Signature", Http://W.W.W.Adfa.Edu.Au.

6.Internet Communication "Biometrics User Authentication" Http://W.W.W.Dell.Cz.Html.

7.Internet Communication "What Is A Hash Function?", At Http://W.W.W.Rsasecurity.Com.

8.Http://W.W.W.Oga.Co.Th/Syncom/Securid/Resource/Fags /Indes.Html.

9.FIBSPUB 180-1, Supersedes. "Secure Hash Standard", Http://W.W.W.Itl.Nist.Gov/Div897/Pubs/Fip. 180-1.Htm.

10.Andrej Bauer. "Gallery Of Random Art", Http://W.W.W.Cs.Cmu.Edu/~Andrej/Art/1998.

11.Carl. H. Meyer And Stephen M. Matyas "Cryptography: Anew Dimension In Computer Data Security". John Wiley & Sons.

12.Dhamigja, R., And Pering, A., "Déjà Vu: A User Study Using Image For Authentication.

performing repeated impersonation to discover the entire password images.

6. Conclusions

long random strings, such as hash values, are difficult to remember. It is much easier for people to deal with images instead of meaningless strings. Therefore a new user authentication system, which is called *structure image authentication system (SIAS)*, has been developed. The requirements of a recognition-based system are examined and the SIAS method is used to authenticate people using the structured images. The key insight SIAS is to base the person authentication on a human strength instead of on a human weakness. Most currently used authentication schemes authenticate the user through precise recall, a task that people are not particularly good at.

The prototype solution of hash visualization technique is developed, it is called *Random Image Generation (RIG)*. RIG has been constructed to converts alphabetical meaning or meaningless string into structured random image. Each image is described by a random mathematical formula, which defines the color value of each pixel. These generated images can be recognized but can not be really described to others. RIG has been constructed to satisfy all the properties of the *hash visualization technique*, which are *image-generation, ease of computation, near-one-way property, regularity property, complexity property, and entropy property.* RIG has three modules, which are: secure hash function-1 (SHF-1), generating the random mathematical formula, and the image generation. SHF-1 satisfies the near one-way property. The other two successive modules satisfy the ease of computation and the image generation properties. Every generated image is transferred to the frequency domain, to test it's the regularity and complexity properties. The amount of information in the generated image is calculated using the entropy equation.

Person 1 observes *Person 2* during multiple authentication's, he can know *Person2's* password perfectly. In SIAS, to prevent this type of attack, the size of password images is larger than the number of password images, which are presented in the three levels. If the number of password images is 5 (which are selected at the setting password images phase), and the number of the displayed password images is 3 (one at each level), the probability that an observer sees the same password images after one observation is:

$$=1/\binom{5}{3} = 1\backslash 10 = 0.1$$

The setting password images phase, where the users select their password images, is done in a way that only *person2* can see them clearly. So that the observer gains no knowledge of the password images by observing which images he selects, since the position of the displayed password images on the screen is located randomly.

Another possible solution to prevent this attack is that the setting of the password images can be changed in each authentication. The goal is that a legitimate user can still recognize the password images, while filtering less information about the password images to the observer.

6.4 Intersection Attacks

If all password images are part of the three levels images, and all the images in the three levels are changed in each trail, *person 1* can use the intersection of two trails to reveal the password images. This is a serious problem, and it is taken into account in the SIAS. The SIAS consists of multiple levels. Each level presents ten images, nine of them are selected images from the decoy database, and one image is selected randomly from the password images. If a user makes a mistake in any level, all subsequent levels are only selected from the decoy database without displaying any of the password images. This prevents an adversary from

6.Attacks and Countermeasures

a number of possible attacks are identified, which serve to impersonate the user. In the following scenarios, *person 1* an attacker who wants to impersonate *person 2.*

6.1 Brute-Force Attacks

person 1 attempts to impersonate by picking random images in the challenge set of the images, hoping that they are part of *Parson 2's* password. The probability that *person 1* succeeds is $1/(^{n}_{m})$, which depends on the choice of $n = 30$ and $m = 3$, the probability to succeed is:

$$=1/\binom{30}{3}=1/4060=0.000246$$

To prevent brute-force attacks, the system may deny assess after a small number of trials.

In SIAS, to prevent the brute-force attacks, the system will be presented with a multi-stage authentication (three levels). If the user correctly identifies the entire password image in each level, the user is authenticated, otherwise, the SIAS will reveal only decoy images after the user makes an error in any stage. The user finally will fail to login to the system application. This means that the user has actually one trail to authenticate.

6.2 Educated Guess Attacks

if *person 1 knows Person 2's* taste in images, he might be able to predict which images are in *person 2's* password.

In SIAS, the RIG is used, which makes it hard for *person 1* to predict *Person2's* password images, even if he knows her preference. Since users tend to pick appealing pictures for their passwords. It will be clear, which images in the challenge set, are the password if they are not all equally appealing. Therefore, the password images are tested quantitatively to ensure that no weak images are used.

6 .3 Observer Attacks

Ross Anderson shows that observation of PIN codes on ATMs has been used to impersonate users [23]. Similarly, if

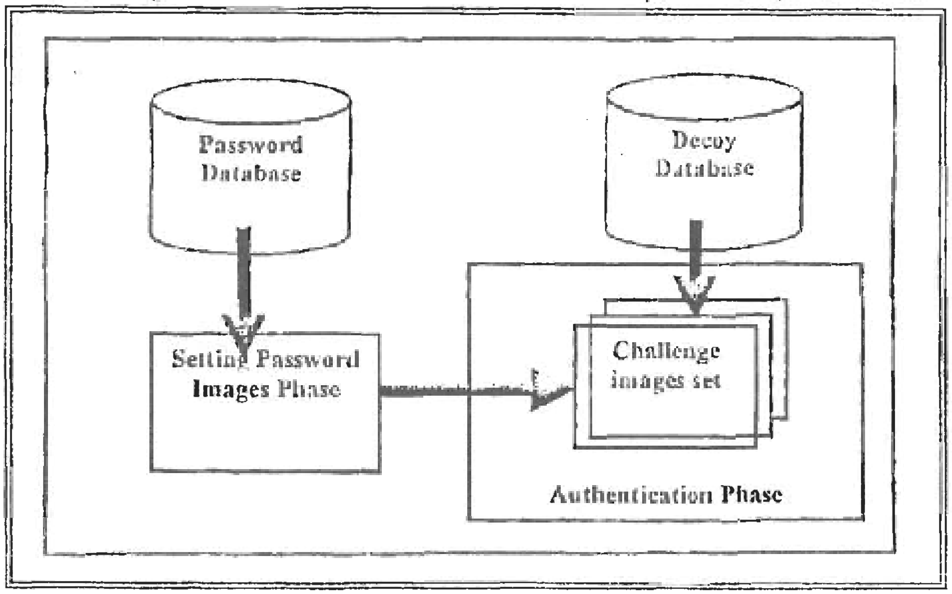


Figure (11): the SIAS architecture, which is consists from setting phase and authentication phase.

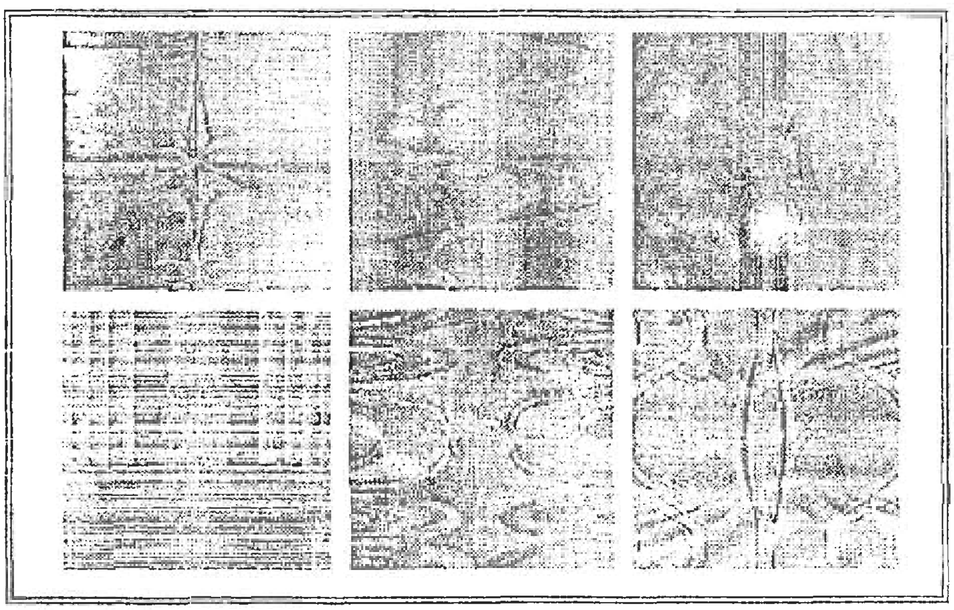


Figure (12): Six sample images of password images.

screen, and picked by the end users through the authentication phase;

b.create a new set of quantitative password images, and select any five images to be used in the authentication phase.

In figure (12), six samples of password images are selected out of password database. These password images are hand selected from the quantitatively tested images.

5.2 The Authentication Phase

At the authentication phase, the system has three hierarchical levels to strengthen the degree of security. Each level displays a challenge set that contains ten images. Nine images are selected randomly from the *"decoy database"* and one image is selected randomly from the subset of the *"password images"*, which are chosen by the SIAS administrator.

At this phase, the ten images of the challenge set are scattered on the screen randomly and the user should pick up three password images, one at each level. If the user correctly identifies the entire password image, the user is authenticated, otherwise the subsequent levels will only display ten images all of them are selected form *decoy database* and the user can not be authenticated.

The quantitative images that are generated from the RIG algorithm and quantitatively tested are stored in two particular databases, which will be used in the SIAS. The set of images, which some of them are selected to be used as a password images, is stored in a particular database that is called *"a password database"*. Initially, the *"password database"* consists of sixteen-images. Another set of images, some of, which are selected to be used as a non-password images, is stored in another database that is called *"a decoy database"*.

Using SIAS, the user creates a password images, by selecting a subset of p images out of sixteen-images stored in the password database.

To authenticate the user, the SIAS system presents a *challenge set* to be displayed on the screen, consisting of n images. This challenge set contains m images out of password images. The remaining n-m images from the decoy images. To authenticate, the user must correctly identify the images which are part of his password images. In SIAS, the password images are distributed among three levels, and each level can contribute a part of the challenge set for the authentication.

The SIAS has two phases, which are setting password image phase and authentication phase. The SIAS architecture can be shown in figure (1).

5.1 Setting Password Images Phase

The SIAS administrator is responsible on the setting password images. SIAS administrator can be one person, or a group in a network. He has two alternative decisions:

a. selects and chooses only five images, which are called *"password database"* where the password database contains sixteen passwords images. These selected images can be displayed on the computer

For example, the figure (10a) and (10c) shows two generated images and the corresponding frequency spectrum, which are satisfying all the properties of IIVF.

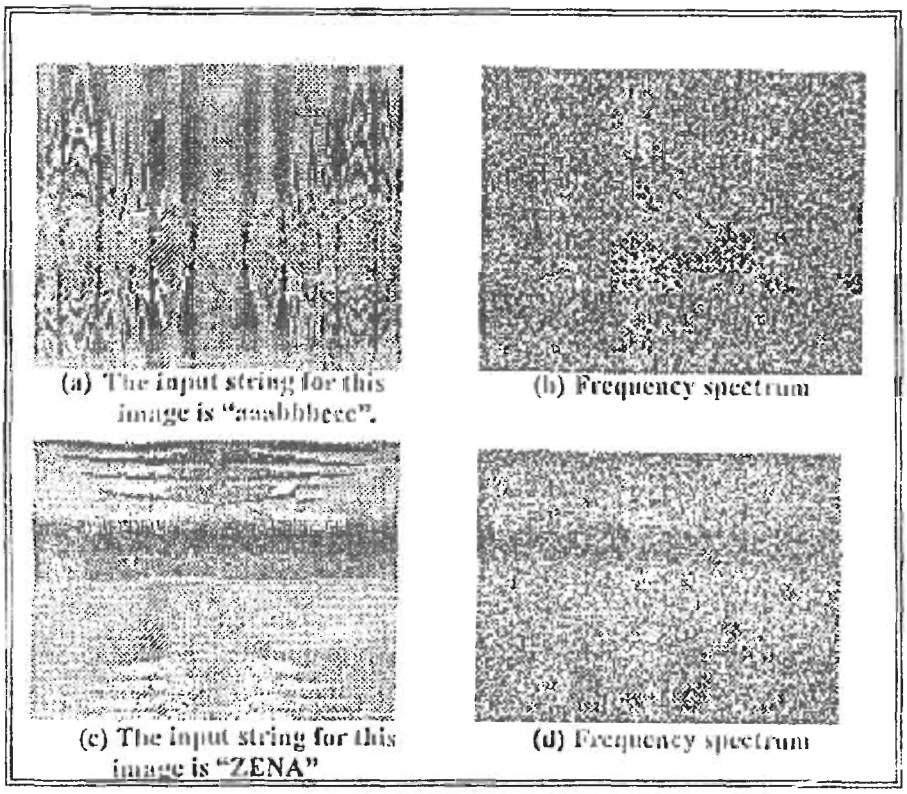
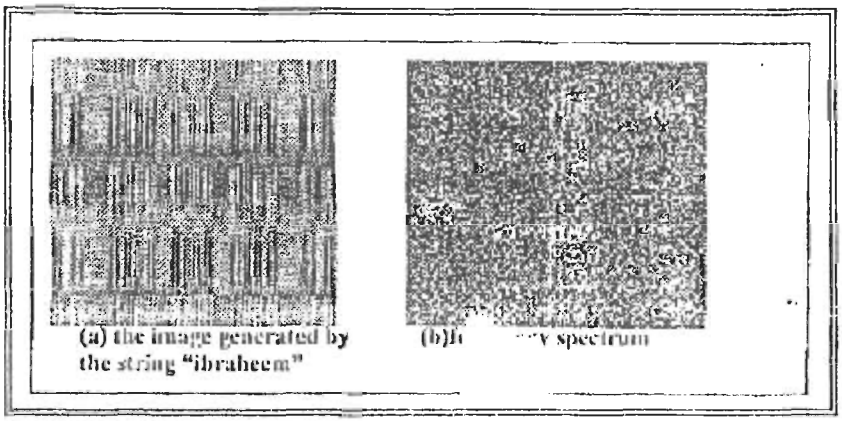


Figure (10): Example of accepted images

5. Structured Image Authentication System (SIAS)

We proposed *structured Image Authentication System* (SIAS) for user authentication. SIAS is a user authentication system, which authenticates people through image recognition. The SIAS system is based on the observation that people are extremely good at pointing out which images they have been seen [4] previously.

hand the image, and its corresponding frequency spectrum, which are shown in figure (8a), are accepted because the frequency spectrum does not have too much energy in high frequencies.


(a) the image generated by the string "ibraheem"

(b)f [illegible]y spectrum

Figure (9): An example of irregular image, a. shows the irregular generated image and b. shows the noisy frequency spectrum for image in (a) with energy in high frequencies.

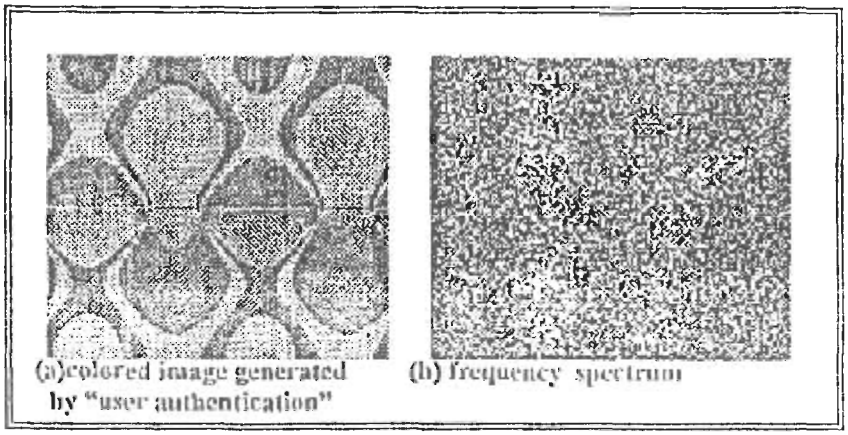
c. The Complexity Property

An immediate implication of this property is that an image can not be too simplistic in shapes and patterns or rely on subtle color differences. Just like for to the regularity property, we could use the frequency spectrum to detect images that are simplistic. For example, the frequency spectrum of such a simplistic picture had all the energy in the lowest frequency components [10]. For example, the image in the figure (8a) does not have much energy in low frequency.

Figure (9): an example of simplest image, a. the simplest generated image and b. shows the frequency spectrum for image in (a) with energy in low frequencies.

L represents the total number of gray levels (e.g., 256 for 8 bits).

The entropy for the image that is shown in the figure (7) is 7.2144 bits per pixel.



(a)colored image generated by "user authentication"

(b) frequency spectrum

b. Regularity Property

Humans are good at identifying geometric objects (such as circles, rectangles, triangles, and lines), and shapes in general. We call images, which contain mostly recognizable shapes, regular image. If an image is not regular, i.e. does not contain identifiable objects or patterns, or is too chaotic (such as white noise), it is difficult for humans to compare or recall it [10].

Non-regular images tend to have wide frequency spectra. Noisy images contain a high percentage of the energy in high frequencies. Hence we can transform an image to the Fourier domain and compute the magnitude spectrum. If the magnitude spectrum dose not have too much energy in the high frequencies, then the image is regular [10].

For example, the image, which is shown in figure (7a), is irregular because its frequency spectrum as shown in figure (7b) has too much energy in high frequencies. On the other

image database. The tests include test them *image regularity, image complexity, and the amount of information in the image using the entropy property.*

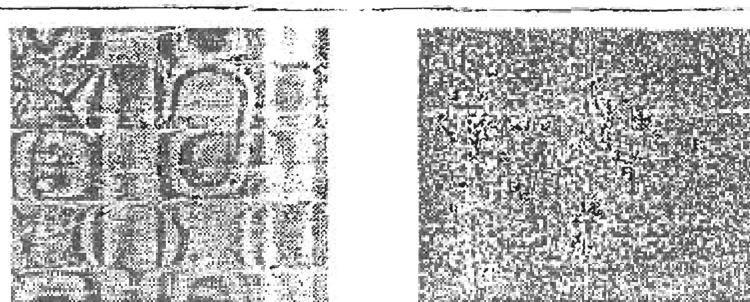
Every image generated by **RIG** must satisfy all the properties of the **HVF.** The **RIG** algorithm satisfies the image-*generation, ease of* computation *near preimage resistance, near* 2^{nd}*-preimage resistance, and near collision resistance properties*. But the regularity, complexity and entropy properties are not satisfied yet. So every image generated by **RIG** must be tested for its regularity, complexity and the amount of information (entropy).

a. Entropy Property

The average of information in an image is calculated by the entropy. The entropy for an **N x N** image can be calculated by the equation [1].

$$entropy = -\sum_{i=0}^{l-1} p_1 * \log_2(p_2), \ldots\ldots\ldots(1)$$

where p_i represents the probability of the ith gray level = n_k/N^2.



(a)the image generated by the string "Nethal"

(b) frequency spectrum

Figure(7)

n_k represents the total number of pixels with gray value k.

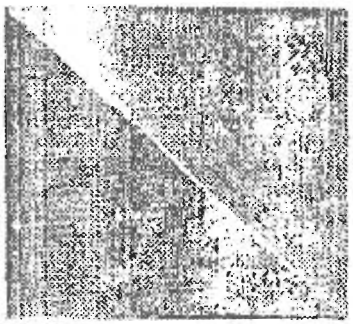


Figure (4): Image generated by the random formula
RGB[sin (X[]),div (X[],Y[]),cos (Y[])]

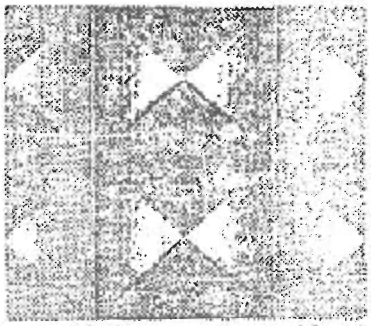


Figure (5): Image generated by the formula
RGB[mod(sin(Y[]),cos(X[])),rgb(X[],Y[],cos(X[])),

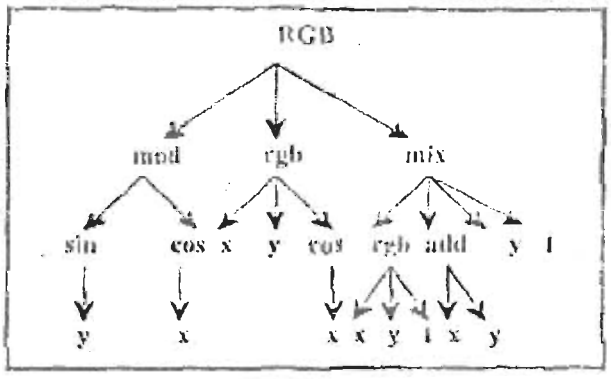


Figure (6): the tree representation of formula in figure (5).

4. Testing the Generated Images

The generated images from implementing Random Image Generating (RIG) need to be tested before storing them in

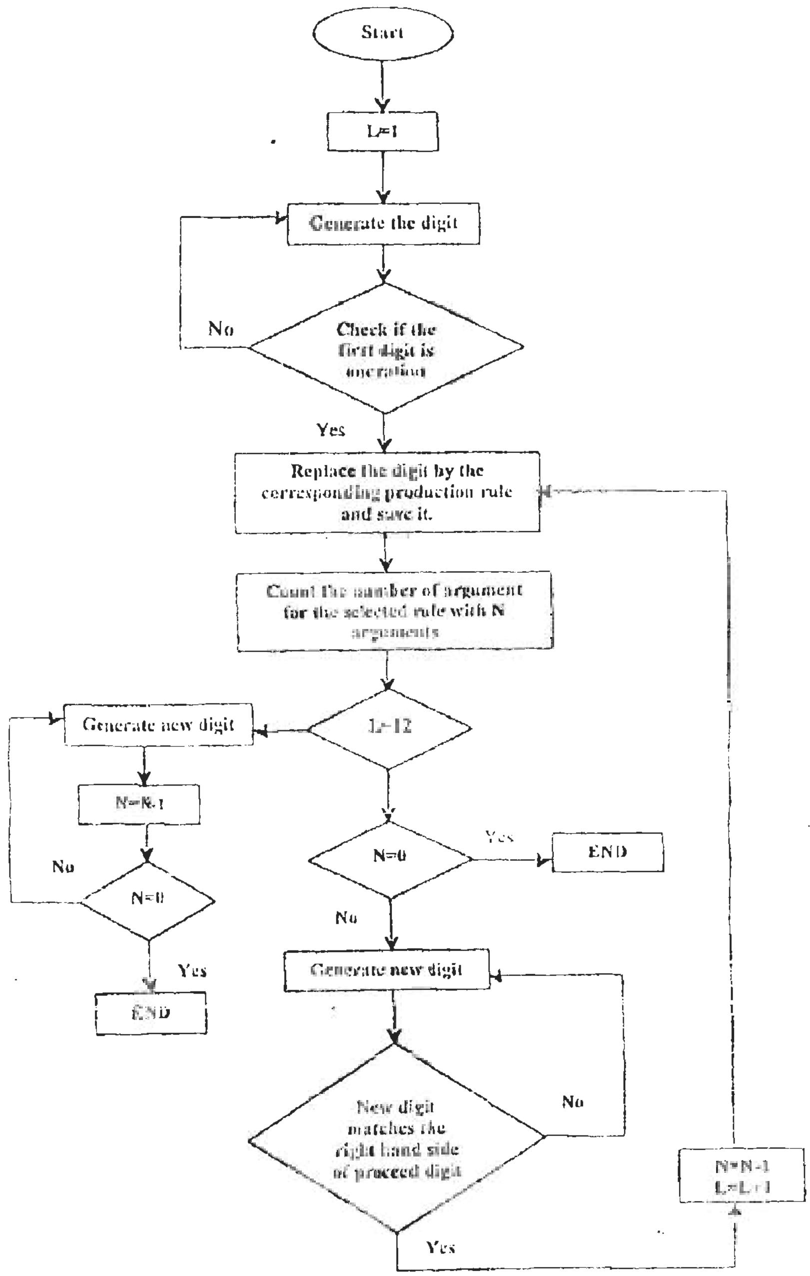


Figure (3): The Flowchart Of GRMF Algorithm.

repeated for all pixels in the two-dimensional image and generates the final colored image.

All functions in RIG algorithm are scaled to the interval [-1,1]. It means that the input and output for every function is in the interval [-1,1]. This condition ensures that all randomly generated expressions are valid. For example, the scaling for add function is achieved by defining.

Add (X, Y) = (X + Y)/ 2

And the scaling for sin function is achieved by defining

Sin (X) = Sin ((X+1) * 2π

And so on for all others functions. A very simple formula gives a very simple picture. For example, the three formula:

Red formula = sin (X[]), Green formula = div (X[], Y[]), Blue formula = cos (Y[]).

Give a picture that is shown in figure (4) and corresponding formula. Other functions are used to make more complicated formulas, more complex pictures are expected. If the formula is complicated enough, it is very difficult to see how it corresponds to the picture. For example, the three formula:

Red formula = mod (sin (Y[]), cos (X[]), Green formula = rgb (X[],Y[],cos(X[]))

Blue formula = Mix (rgb(X[],Y[],1) add (W[],Y[],-1)

Gives a picture that is shown in figure (5).

In order to increase the complexity of the image, the number of the function in each formula is relatively high Each formula has a number of functions less than or equal to 13, Each formula can be represented as tree with depth of 13. For example, the tree representation for the formula in the previous example represent as a tree structured as it is shown in figure (6).

the process is terminated. Otherwise, if the number of arguments of first digit is not equal to zero and the number of functions is greater than 13 then generate random number and decrement one from the number of argument of first digit. This process is iterated until the number of argument of first digit is equal to zero. 6927233.

In the process of converting the stream of digit to the corresponding mathematical functions, each generated rule number digit is substituted by its corresponding right-hand side. Function of the applied rule, and the resulted mathematical formula is represented as prefix notation:

Mult (Sin (X), Div (X,Y)).

The GRMF procedure generates the random mathematical formula for each of the three seeds (red seed, green seed, and the blue seed).

The generated formulas for red, green and blue can be represented as tree with the depth of 13. The length of formula is chosen to be long in order to increase the randomness and the complexity for the generated image.

3.1.1.4 Image Generation

The result of FRMF algorithm is three random formulas; one for the red component, one for the green component, and one for the blue component. In this procedure, each of the three formulas is evaluated to get a pixel (x,y) and each formula gives one color value. Red formula represents the intensity, for red color, green formula represents the intensity for green color, and blue formula represents the intensity for blue color. The red, green and blue values are then scaled from the interval [-1,1] to the interval [0,256] which represent the color value. The three colors then will be mixed in RGB function and produce one value that represents the intensity color for that pixel. This operation

case, number 6 matches one of the specified first generated digits so it is stored in a temporary array.

4-The production rules are searched to find the matching rule with the left-hand side equal to 6. The rule <operation>::= mult (<Exp>.<Exp>) is the applicable production rule which has the rule number equal to6.

5-Count the number of the arguments of the function, (Which defines the input value, and is equal to zero). The number of argument for mult (<Exp>,<Exp>) is equal to 2.

6-If the number of the arguments is equal to zero, then the process is terminated. In this example, the number of arguments is two and increment the number of functions by one. Then repeat the following steps:

a.Generate the next random number and suppose the number and suppose the number is 9.

b.Tee production rules are searched to find the matching rule with the rule number equal to 9. The <operation>::= Sin (<Exp>) is the production rule which has number 9.

c.Count the number of the arguments of the function. In this example, the number of arguments for sine function is equal to1.

d.If the number of arguments of the sine function is equal to zero. Then decrement one from the number of arguments of the mult function. Otherwise generate new random number from right shifted of LFSR five times. The number of arguments sine function is not zero so a new random number is generated and the previous steps are repeated until the number of argument of first digit is equal to zero or the number of functions is greater than 13. If the number of arguments of first digit is equal to zero,

3.1.1.3.1 GRMF Algorithm

Each seed is input to the LFSR to generate the stream of digits in the GRMF algorithm. The *first generated digits* are all the rule numbers that the right-hand side of them that is equal to *<exp>* or *<operation>* and which the left-hand side is equal to *<operation>*. The left-hand side in the rule of *the first generated digit* can be *<operation>* and the right hand side can be any defined mathematical function. These declarations are shown previously in the RIG grammar in table (1), where the rule number '0', the rule number '2', the rule number '3', and the rule number '4' can not be an applicable rule specifically to the first generated digit. So the reminder rules number can be an applicable rules to the first generated digit.

The flowchart of the GRMF algorithm can be seen in the figure (3), where L represents the length of the generated formula (the number of nested functions in the generated formula).

To trace the FRMF algorithm:-

1-Let the first generated number be generated by LFSR is 3.

2-If number 3 matches one of the specified first generated digits, then create the first digit otherwise, it is rejected and LFSR generates a new random number. In this case, the number 3 does not match one of the specified first generated digits because 3 is the number of rule whose left-hand side is <Input> and the resulting formula is prefix expression, which must be started by either *<exp>* or *<operation>*, so that the LFSR will generate a new random number let the number be 6.

3-If number 6 matches one of the specified first generated digits, then create the first digit otherwise, it is rejected and LFSR will generate a new random number. In this

Table (1): The RIG grammar.

Production Rule number	The production rule
0	<Exp>::=<Input>
1	<Exp>::=<operation>
2	<Input>::= X
3	<Input>::= Y
4	<Input>::=R
5	<operation>::=Add(<Exp>,<Exp>)
6	<operation>::= mult(<Exp>,<Exp>)
7	<operation>::= Div(<Exp>,<Exp>)
8	<operation>::= Mod(<Exp><Exp>)
9	<operation>::= Sin(<Exp>)
10	<operation>::= Tan(<Exp>)
11	<operation>::= Arctan(<Exp>)
12	<operation>::= Cos(<Exp>)
13	<operation>::= Reverse(<Exp>)
14	<operation>::= Bw(<Exp>)
15	<operation>::= Expf(<Exp>)
16	<operation>::=rgb(<Exp>,<Exp>,<Exp>)
17	<operation>::= If(<Exp>,<Exp>,<Exp>)
18	<operation>::=Mix(<Exp>,<Exp>,<Exp>,<Exp>)

Table (2): The list of functions that are used in RIG

Function number	Function name	Function meaning
1	X[]	Identify the X coordinate.
2	Y[]	Identify the Y coordinate.
3	R[]	Generate random number in range [-1,1].
4	Bw[G]	A shade of gray, G=-1 is black, G= 1 is white.
5	Expf[a]	The exponential function.
6	Sin[a]	The sine function.
7	Cos[a]	The cosine function.
8	Tan[a]	The tangent function.
9	Arctan[a]	The arctangent function.
10	Reverse[a]	The reverse color of a.
11	Add[a,b]	The sum of two colors.
12	Mult[a,b]	The product of two colors.
13	Div[a,b]	The division of two colors.
14	RGB[r,g,b]	A color specified by the RGB component.
15	If[cond,T,E]	The "if-then-else" function: if cond is positive, then the value is T, else the value is E.
16	Mix[a,b,c,d]	The mixing of two colors a and b depending on the value of c and d.

The grammar rules, which are used in the *GRMF*, are *Backus Normal Form (BNF)*. The BNF grammars that are used in the proposed RIF are defined as it is shown in table (1).

The pseudo random number generator is a linear feedback shift register (LFSR). In RIG algorithm, the length of LFSR is equal to the length of the input seed, which is 10 bits. In addition the feedback function in RIG is XOR function of the first and the last bit of the LFSR. Every time the random number is needed, the LFSR generates a binary string of 5-bits and translated to decimal number. Since the total number of the production rules in the grammar are nineteen rule [0..18] as shown in the grammar in table (1), so each random number, which is grater than 19 is modulated (mod to 19) to this valid range. So if the generated number is greater than 18 then it is normalized into the rang [0..18] by suing the *mod 19* to that number. The functions, which are used in the production rules in the grammar rule, are shown in the table (2).

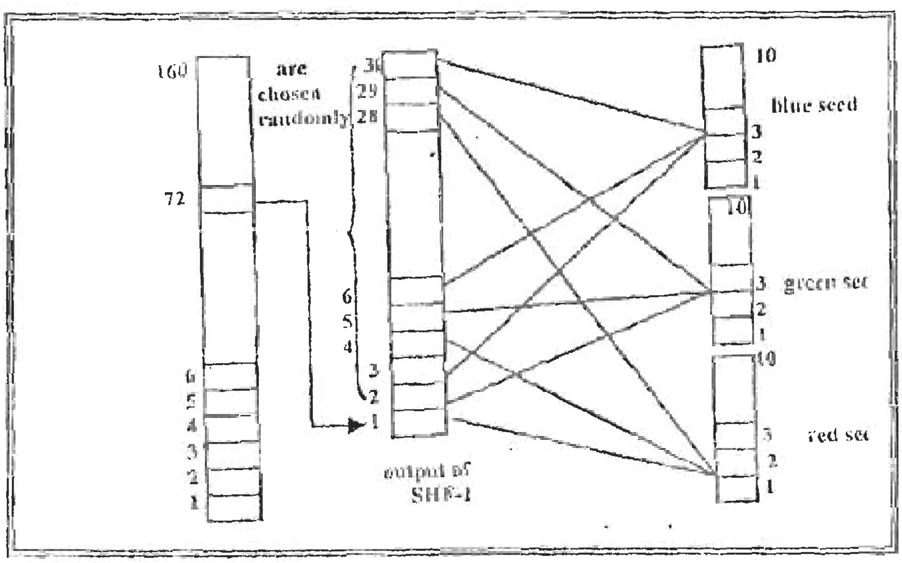


Figure (2): The Choosing Method Of The Three Random Seeds.

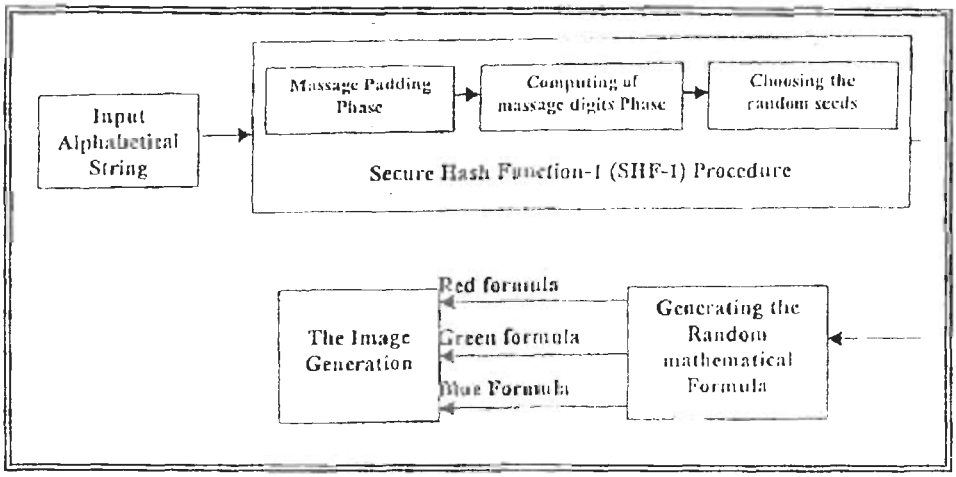


Figure (1): Block Diagram of RIG Architecture.

3.1.1.3 Generating the Random Mathematical Formula

The resulting 30-bit string forms the output of SHF-1 and it is divided into three 10-bit strings for red, green, and blue seeds figure (2) shows the method of choosing the red, green and blue seeds. The first bit in the output of SHF-1 is selected to be the first bit in the red seed, the second bit in the output of SHF-1 is selected to be the first bit in the green seed, and the third bit in the output of SHF-1 is selected to be the first bit in the blue seed. The rest of the bits in the output of SHF-1 are selected as a pre-defined sequence to be the bits of red, green, and blue seed respectively.

Each of the resulting seeds (red, green and blue seed) is use in the *generating the random mathematical formula (GRMF)* to produce the mathematical random formula. Each mathematical random formula is constructed by choosing applicable *grammar rules* depending on the output values of the *pseudo random number generator*.

with SHF-1 to seed a cryptographically secure random number generator. Hence, the pre image resistance property is achieved. SHF-1 procedure is used in the RIG in order to make it one-way technique and to make it computationally infeasible to find two different strings, which generate the same image [9].

When the message of any length $<2^{64}$ bits is binary input string, the SHF-1 produces a 160-bits output called a massage digest. The message or data file should be considered to be a binary string. The SHF-1 sequentially processes blocks of 512 bits when computing the message digest. The message or data file should be considered to be a binary string. THE SHF-1 sequentially processes blocks of 512 bits when computing the message digest [9]. The SHF-1 algorithm can be divided two-phases: *message padding phase* and computing of *massage digest phase.*

After the SHF-1 produces 160-bits string output, a string of 30-bits are selected randomly from 160-bit string. The method of *choosing the 30-bits string* depends on the length of the input alphabetical string 's'. Each character in the input alphabetical string 's' is represented by 1-byte. The position of the first selected bit is calculated by multiplying the length of the input alphabetical string 's' by 8. The rest bits are chosen randomly.

For example, if the input string is "My Mother", the length of this string is 9, each character is represented by 8 bits. The position of the first selected bit is the bit at position 72, and the rest bits are chosen randomly.

into RGB [-1,-1,-1], RGB [255,0,0] is red which normalized into RGB [1,-1,-1] and son on.

Random image generation (RIG) is an algorithm that is given an alphabetical string as input, to generate a function $F:[-1,1]^2 \rightarrow [-1,1]^3$, which defines an image. It is based on the idea of the *Random Art technique* [10].

Given an alphabetical string, *RIG* generates a random mathematical formula, which defines the color value for each pixel on the image plan. The image generation process is deterministic and the image depends only on the input string.

3.1.1 RIG Architecture

The RIG algorithm consists of four components, which are the input alphabetical string, secure hash function-1 (SHF-1), generating the random mathematical formula, and the image generation. Figure (1) shows the block diagram of RIG algorithm.

3.1.1.2 Secure Hash Function (SHF)

A one-way hash function is a function, mathematical function of otherwise, that takes a variable-length input string (called *pre-image*) and converts it to a fixed-length (generally smaller) output string (called a *hash value*) [7]. One important type of one-way hash function is *Secure hash function (SHF)* that provides relatively high security and most useful type of SHF is found to be *SHF-1* [9]. The is called secure because it is computationally infeasible to find a message, which corresponds to a given message digest, or to find two different messages, which produce the same message digest.

Any modification in a message in the transmitted will, with very high probability, give a new different message digest [9].

RIG needs to satisfy the near one-way requirements of hash visualization. It is achieved by hashing the input string

Visualization technique. In particular, we aim to satisfy the following requirements:

1. *The system should be based on image recognition, to make the authentication task reliable and easier for the users.*
2. *The system should prevent users from choosing weak passwords.*
3. *The system should make it difficult to write passwords down and to share them with others.*

To explore these requirements and improve the security of these systems, the *Hash Visualization Technique* that replaces meaningless string with structured images, is used. And the proposed a prototype solution for hash visualization technique is *Random Image Generation (RIG)* that is based on a cognitive science.

Instead of having a user memorize a password, the user is able to create a password images, by selecting some desired number of images, which the user must memorize for recognition. During authentication. The user is presented with different set of images, some of which are chosen, from the password images and others, which are chosen randomly. To login, the user must correctly identify all of the images form user's password images.

3.1 Random Image Generation (RIG):-

Generally, each pixel in plan image can be represented by x and coordinates, and the pixel value is evaluated from the three components red, Green, and Blue, which can be denoted ad [R, G, B]. While in the proposed prototype solution, the pixel coordinates rang continuously from –1 to 1, in both x, y dimensions. The pixel components (red, green, blue) should also have a value between [-1,1], since its value is dependent on a mathematical functions (such as sin, cos, etc.). For example, RGB [0,0,0,] is black which is normalized

b. Simple meaningful passwords or that are associated with the user are easier to remember, but are vulnerable to attack.

c. Password that are complex and arbitrary are more secure, but difficult to remember. The users tend to write them down. This compromises security since the user might forget, lose or leave the paper in an insecure place.

d. The numbers of applications and services, which require password, have dramatically increased. Since users can only remember a limited number of passwords, they tend to write them down or will use similar or even identical Passwords for different purpose. Both options increase the chance of a security compromise.

e. The people have difficulties with choosing and memorizing secure passwords and personal identification numbers (PIN) [5]. Most security systems still suffer from the fact that they fail to account for human factors. Humans are slow and unreliable at processing long and meaningless strings, yet many security applications depend on this skill. These humans' factors negatively affect many security systems, including the security of user authentication.

3. Proposed Solution

in fact, classic cognitive science experiments have shown that humans have a vast and limitless memory for pictures and images in particular. The experiments show that humans can remember and recognize hundreds to thousands of pictures in fractions of a second. Therefore, by replacing precise recall of the password with image recognition. The user cognitive load can be minimized, and the mistakes done by the users can be minimized.

This work is to explore the user authentication aspects more thoroughly, and design a prototype system that perform and improve user authentication using *Hash*

most popular, and the ATM card that needs the memorized password (PIN) [8]. There are another forms for token authentication, it may be hand-held calculator. Diskette like cartridges. Modems and son on.

c. *Biometrics-based system (biometrics authentication)* is based on physical features such as a fingerprint. Retina, iris, hand or face. Although voice and signature identification do not involve physical characteristics, they are usually included with biometrics user authentication [6].

In today's security systems, knowledge-based schemes are predominantly used for user authentication because of these reasons [12]:

First, although biometrics can be useful for user identification. One problem with these systems is the difficult tradeoff between impostor pass rate and false rate. In addition many biometrics systems require specialized devices. And some can be unpleasant to use.

Second, most token authentication systems also use knowledge-based authentication to prevent impersonation through theft or loss of the token. An example is ATM authentication, which requires a combination of token (a bankcard) and secret PIN.

2. Shortcomings of Password Authentication

Despite their wide usage, password and PINs have a number of shortcomings [12,3]:

a. the main weakness of knowledge-based system authentication is that it relies on precise recall of the secret information. If the user makes a small error in entering the secret information, the authentication fails. Unfortunately, precise recall is not a strong point of human cognition.

unique secret piece of information is provided to be in the user's possession. As a matter of fact the indirect user authentication is equivalent to massage authentication which relies upon imposing a prearranged structure for the message [2].

There are three types of information that a computer can usefully process to independently authenticate, that a person at a remote terminal is, in fact, whom he or she claims to be. These are typically referred to as the *"Three Factors of Identification"* [8].

i. *"something known"*:- evidence that a person knows a secret (e.g., a memorized password) that has been given to a specific person.

ii. *"Something hold"*:- evidence that the person holds a token (e.g., a credit card, or an Automatic Teller Machine (ATM) card) that is given to a specific person.

iii. *"Something one is"*:- a biometrics; some human attribute, difficult to counterfeit. It can scanned and digitally documented (e.g., a fingerprint or retina scan) so that it can be compared to previously recorded scan, taken from a specific person.

Depending on this information, user authentication can be distinguished into three min techniques:

a. *Knowledge- based system (password authentication)* which relies upon comparing is secret, they must be enciphered before sending via transmission channels [11].

b. *Token-based system (token authentication)* is based on possession of some object that is combined with password [5]. It also called two-factor authentication because it is require at least two of the three primary factors to use to authenticate an individual [8]. Token authentication system comes in different forms. The credit card is the

image is described by a random mathematical formula, which defines the color value of each pixel.

These generated images can be recognized but can not be really described to others.

Our approach relies on *recognition-based,* rather than *recall-based* authentication. We examine the requirement of recognition-based authentication system and proposed the *structured image authentication system (SIAS),* which authenticates a user through recognizing the previously seen images. The *SIAS* improves the security, since it relies on the recognition-based authentication. The SIAS is more reliable and easier to use than the traditional recall-based schemes. Furthermore, it has the advantage that it prevents users from choosing weak passwords and limited the difficulties to write down the passwords or sharing them with others.

Keywords: *User authentication, hash visualization, human factors, password authentication system, genetic programming, structured images.*

Introduction

User authentication is the necessary foundation for all computer and data security. It is a central component of currently deployed security infrastructures. Several secure technologies haved to offer ways for a computer to authenticate that a specific electronic identity (an on-line personal) actually represents the human being that was originally assigned specific rights and authority by a trusted administrator. It can be made either *directly* when the useir's specific characteristics (e.g. finger print, retina patterns, digital signature flow..etc.) are checked, or *indirectly* when a

User Authentication Using Hash Visualization Technique

Dr. Abdul Rahman H. Alhusainy*
Dr. Nidhal Al Saied**
Ikhlas Abbas **

Abstract

Some of human factors negatively affect many security systems, including the security of user authentication. These human factors are: first, people are slow and unreliable at processing and comparing long meaningless password string; and second, people have limitation difficulties in remembering secure passwords for Personal Identification Number (PIN).

In this paper, fundamental weaknesses of knowledge-based authentication schemes are addressed and how the usability and security of the user authentication systems can be improved using *hash visualization* that replaces alphabetical string with structured images. Our prototype solution of hash visualization technique is *Random Image Generation (RIG)*. RIG has been constructed to converts alphabetical meaning or meaningless string into structured random image. Each

* Mins. Of Higher Education & Scientific Research
** Univ. Of Technology