Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.

ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-139

Shape grammar model generating secure visual passwords

The move towards completely grammar based images

[Mokgadi Rasekgala, Ian Sanders, Sigrid Ewert, Thomas Fogwill]

***Abstract*—Computer security is largely supported by passwords forming the principal part of the authentication process. Visual passwords have been proposed as a more secure alternative to alphanumeric passwords but have not as of yet been readily adapted due to the problems associated with current implementations of visual passwords. One of the main issues in the implementation of visual password schemes is the type of password images used. To address the issue of designing and using secure password images, we present a model of a shape grammar supported visual password scheme generating password images that adhere to a specific syntax or set of rules. We present a design of a grammar model which is well adaptable for implementations of visual password scheme images and images with a rule based structure.**

Keywords—grammars, shape grammars, visual password authentication, theoretical computer science

I. Introduction

For the majority of computer systems, passwords are the method of choice for authenticating users because they are easier and cheaper to implement compared to token based and biometric authentication methods[1]. The most popular of these password based systems are alphanumeric password systems which use a string of a selection of alphabets and numbers as the password[2].

Alphanumeric passwords have however been shown to have significant disadvantages. For example, users cannot remember secure alphanumeric passwords which should be a meaningless string with a good mix of numbers and alphanbets in both upper and lower case [2] . Users therefore resort to choosing predictable passwords which are prone to guessing attacks and brute force attacks. In attempt to address the problems associated with alphanumeric passwords, visual passwords are introduced[3]. The move towards visual passwords was mainly supported by the ability of people to remember pictures more easily than they can words referred to as the picture superiority effect[4].

Visual passwords are not still a new idea but have not as of yet been widely used due to problems associated with implementation of visual passwords. The work of [7],[8] and [9] describe and analyse design considerations that can be taken in designing efficient adaptable visual password schemes and the work of [10], presenting exact rules that can be used to generate secure visual password schemes. These works present a high level conceptual design of secure graphical passwords and note that the type of password images used are of hifgh importance. There has been, to our knowledge, no work addressing the exact implementation of visual password schemes that address the current problems with visual password schemes. This work takes a step in that direction. We adopt the definition of a safe password set in the work of [10] and design a shape grammar based visual password scheme enforcing only the generation of safe password images to be used in the scheme described by [10].

Shape grammars are a formalism used for rule based designs. They have previously been used in the field of architectural design to generate houses and different styles of buildings. This work takes inspiration from matching the rule based nature of shape grammars and the rule specification of secure visual passwords presented in the work of [10] and designs a grammar-based password scheme for the generation of secure password images.

The contribution of this paper is :

- A grammar model allowing for a procedural method to designing a shape grammar which will only generate images adhering to a specific password syntax.
- *PassImages*, a grammar which is designed as an instantiation of the grammar model designed . *PassImages* generates images meeting requirements of the password imgages described in [10].
- Introduction of the idea of grammar designed images for easier storage and processing of images for general design.

The paper is structured as follows. Section II presents related work giving context of using shape

Mokgadi Rasekgala
University of the Witwatersrand and CSIR

Prof Ian Sanders
University of South Africa

Prof Sigrid Ewert
University of the Witwatersrand

Thomas Fogwill
Council for scientific and industrial research

grammars as well as work presenting the definition of secure visual passwords by refferring to literature. The next section, Section III presents the method used in generating the grammar model and an instantiation of the model resulting in *passImages.* Section IV gives the significance of the work and the last section, Section V concludes the work by bringing together the contributions of the work and future work that could be derived from this work.

II. Literature Review

A. Shape grammars

Shape grammars were first introduced by Stiny and Gips for use in painting and sculpting[5]. They are a formalism that can be used for creating and describing designs. Shape grammars are a formal rule based system, consisting of an alphabet of shapes , a starting shape and shape rules defining the spatial relations between the shapes [6]. The primary goal of a shape grammar, is to provide a simple way to fully describe a language.

While some languages of designs can be described using simple grammars, more complex designs require grammars which are able to see state and need rules that are dependent on the elements in the current picture generation as the design emerges. Literature shows that the combination of attribute grammars[11] and set grammars[12] enable the design of grammars which can generate more complex designs. Set grammars simplify shape grammars making them more ameanable to computer implementations[13][12]. Attribute grammars define static and dynamic sematics of a language. The work of [12] uses both set and attribute grammars to design split gramars[12]. The objects of the split gramar are attributed, parametrized , labelled shapes wich are reffered to as basic shapes. The object is a three tuple <s,P,V> giving s the label representing the simple shape associated with the object, P the position of the simple shape and V a set of attributes of the shape. The split grammar generated designs by continously splitting the area of a shape into a number of shapes. The work of [13] extends the split grammar and designs the CGA shape grammar which has the ability to generate detailed buildings from complex mass models. This is done by producing context sensitive shape rules suitable for computer graphics architecture. The rules in the grammar are of the form:

id: predecessor : cond → successor : prob

id refers to a unique identifier for the rule, *predecessor* to a left hand side shape that can be converted to the *successor* when the rule is applied, if the condition *cond* is met. The probability of the rule being applied is given by prob.

Even with the CGA shape grammar being more general there seems to be a gap in the type of images designed by current grammars. There seems to be no grammars designing images that simply abide to a drawing area with no specific sequence to the grammar.

B. Visual passwords and requirements for secure visual passwords

Thorpe and Oorschot[14] present a study presenting parametres and guidelines to be considered in designing secure graphical passwords. Most of the literature covered shows that visual authentication is dependent on the type of images that a passwords cheme uses and how those images are encoded and retreived when required. The works of [16], [17] and [4] all present guidelines to be used in setting requirements.

Our earlier work in [10] presents a comprehensive study on the requirements of a secure visual password scheme and the images such a scheme uses by studying the literature. The work examined current and anticipated challenges in implementing secure visual password schemes. The work notes three major problems of impementing visual passwords. The first is that visual passwords are more vulrenabe to shoulder surfing than conventional alphanumeric passwords as pictures are larger and easier to see from a distance[7]. Due to the picture superiority effect, images can also be remembered more easily than secure alphanumeric passwords. Another disadvantage is the reduction in the number of possible passwords which is reffered to as the password space due having to store all possible passwords in searchmetric type password schemes[8]. The work of [10] results in a specification for a visual password scheme which allows for a large password space, low suscepetibility to shoulder surfing and one eliminating the challenges associated with drawmetric schemes associated with redrawing passwords exactly as they were originally drawn on registration.

The work of [10] describes a secure visual password. The figure below, Figure 1, presents the workflow which describes the scheme. A user creates a password image by selecting a number of shapes from the set of four shapes (square, rectangle, traingle and a circle) and placing them placed on a 5 x 5 grid. To reduce storage memory, the password image is stored as a set of grammar rules instead of an image data file. The password is saved as a set of grammar rules that can be used to generate the image. The issue was now in implementing a grammar that allows only for such a generation and restricts all images not conforming to the password syntax. According to the requiremnts stated in [10] a user should be guided by the scheme embedded rules to generate a password consisting of atleast three and at most twelve shapes. The number of

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-139

shapes also reffered to as the shape count of the password image. The length of the password, which constitutes the number of lines making up the password is bounded at a maximum of twenty and a minimum of eight[10].

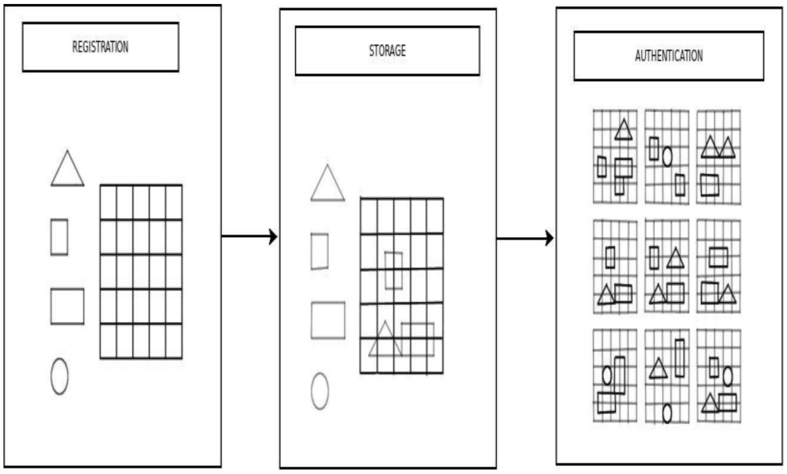

Figure 1: *Flow diagram depicting the system outline of secure password scheme described in [10]. A user creates a password image from a set of shapes being a square,rectangle, triangle and a circle. The sheme stores the password image as a set of password rules generating the password image. On authentication the user is presented a with a set of images similar to the password image reffered to as decoys.*

III. Shape grammar design

A. Grammar model

While most shape grammars describe designs by relating each of the shapes within the design to one another, we note that this is not the general make up of all images. Some designs consist simply of a collection of similar objects placed on a surface with each object having its own position and attributes. The elements of the design or picture are not necessarily related to each other, either than the fact that they all form part of the same image. There is no real relation stipulating where another shape should be placed dependet on the position or attributes of another shape in the setential form of the generation. The only condition all shapes should abide to is that they should fit within the design area and are therefore bounded by the same design restrictions.

We design a grammar $G=(V_N, V_T, R, I)$ generating only the language of allowed passwords. The grammar model describes a grammar which allows for the sequential generation of a password image by the addition of shape objects at each step.

Grammar objects

We refer to the definition of a shape given by [19] which describes a shape as an arrangement of straight lines in a 2-D euclidean space. The terminal and non-terminal objects of our grammar both have different representations.

A terminal is inspired from the shape objects in the CGA grammar. The terminal shape is an attributed paremetrized shape $t=\langle s,P,V\rangle$ where s is a symbol representing a simple shape, at position P . V contains a number of attributes which are represented as a set of name-value pairs. An example is defined as follows:

Square shape=<S ,(1,3) , {(color:blue) ,(orientation:90)} > . The square shape representing a square at x position 1 and y position 3 in the color blue and rotated 90 degrees. S being the symbol used to represent the shape.

A non-terminal shape is a label $n=N$ where N represents the non-terminal object label. The non-terminal shape is used as a place holder in the grammar to eventually generate one of the terminal shapes or an empty shape.

An image forming part of the grammar language is obtained when there are no non-terminal labels in the sentential string.

Grammar rules

The grammar is designed for images generated from a set of primitive shapes on a $n\times n$ grid. The image abiding to a set of rules describing the structure of the password .

The initial shape/s I , are the objects all the images forming part of the language of the grammar are generated from. TThe production rules are similar to those used in the CGA shape. We however eliminate the use of $prob$ in the rules as the grammar is completely user based and does not depend on scheme for the decision on the rules to be used to generate a user's password image. The gramma rules are therefore defined as

$$(id)\, predecessor \rightarrow successor : cond$$

The rules of the grammar are dependent on a number of factors that each password image should put a value to in

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-139

defining the syntax of an allowed password image. The factors being:

- Primitive elements making up the password image
- Password drawing area or grid
- Length of the password
- Attributes of the grammar objects

Elements used to make up the password image: The primitive shapes of the password image forms the terminal objects of the password image. The terminal set defined as:

$$V_T = \{\langle t_{1,} P_{1,} V_1 \rangle, \langle t_{2,} P_{2,} V_2 \rangle, \ldots, \langle t_n, P_n, V_n \rangle\}$$

$t_{1,} t_{2,} \ldots, t_n$ are symbols for all the simple shapes used to generate the password image. The attributes (V_i)and position (P_i) of the shapes are set when a user selects where the shape is placed and the specific attributes respectively.

Password drawing area: The password image (P) is to be presented on a $n \times n$ grid. For every password image abide to the grid bounds, there is a need for conditional checks within the grammar rules when placing a shape to ensure the shape stays within the grid boundary. For example, a password scheme with a square of length one grid unit, as one of the simple shapes, can only allow for the square to be placed at position (x,y) on cond:

$$1 \leq x \leq 4 \text{ and } 1 \leq y \leq 4$$

with the reference point of the shape being the left most top most point.

Allowed Length of the password image: The length of a password image can be measured in two different ways. We can restrict the length by reffering to the shapecount (X) of the password or the sum of all lines making up the shapes in the password image. The shape count of an image refers to the number of shapes making up an image.

To restrict the shape count, there must be a set structure representing holders for the allowed shape count number of objects. For a minimum of X_{min} shapes there must be X_{min} object holders of type S_1 that can only generate terminal shapes. For the maximum shape count (X_{max}), there must be an extra $X_{max} - X_{min}$ object holders of a different type, say type S_2 that will each generate a terminal shape or an empty shape.

For a password of length between L_{min} and L_{max} the terminals generated from the S_1 and S_2 objects. This is achieved by having state attributes for the grammar which can be checked before applying each rule for shape placement. To ensure that the minimum length (L_{min}) is met, the place holder S_2 can only generate an empty shape once the length of the password in the sentential form is greater or equal to L_{min} .

The full password image PI therefore looks like :

$$PI = S_{1_1} S_{1_2} \ldots S_{1_{Xmin}} S_{2_1} S_{2_2} \ldots S_{2_{Xmax - Xmin}}$$

Where:

- S_{1_i} generates a terminal shape

- S_{2_i} represents a non terminal which generates a terminal or empty shape

- PI fits onto the password grid

- $L_{min} \leq length(PI) \leq L_{max}$

-No element of the password shapes should be idetical in position and shape to avoid exact overlaps.

The grammar $G = (V_N, V_T, R, I)$ described by the grammar model is presented as:

$$V_N = S, S_{1,} S_2$$

$$V_T = \{<t_{1,} P_{1,} V_1>, <t_{2,} P_{2,} V_2>, \ldots <t_n, P_n, V_n>\}$$

$$R = (0) S \rightarrow S_{1_1} S_{1_2} \ldots S_{1_{Xmin}} S_{2_1} S_{2_2} S_{2_{Xmax - Xmin}}$$

$$R = (1) S_2 \rightarrow \varepsilon \quad : length(PI) \geq L_{min}$$

$$R = (2) S_2 \rightarrow S_1 \quad : length(PI)$$

$$+ minLength(t_{1,} t_{2,} \ldots t_n) \leq L_{max}$$

$$R = (3) S_1 \rightarrow <t_i, P_i, V_i> \text{¿}$$

$$\text{¿.} \quad :$$

$$PboundsMet(t_i)$$

$$I = S$$

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-139

The position P_i and attributes V_i of the shape t_i are set when the user places the shape on the grid. P_i is set by the position the shape is placed and V_i by the thinngs such as color and size that the user selects on placing the the shape on the grid.

Length(PI) refers to the current length of the password in the sentential form and *PboundsMet(**shape**)* is a boolean check determining whether the boundary of **shape** is within the bounds of the grid and that there is not an exact copy of the shape on that position.

Attributes of the password objects: To add image complexity depending on the design of the password images, there should be a way to cater for the addition of attributes to the shapes within the passwod image. The atributes also help to say more about the shapes that can not be easily stipulated by a simple shape grammar.

B. Grammar instantiation

The design of the grammar described above can be used for the generation of secure password images. As proof of concept we use the grammar model to design a shape grammar enforcing a secure password image as defined by the requirements in [10].

Elements used to make up the password image: The easiest part of the grammar to be defined are the terminal objects which are made up of all the shapes used to create the password image. The password scheme described in [10] presents a scheme using a square, rectangle, triangle and circle as the primitive shapes. The terminal set (V_T) is represented bnelow:

$$V_T = \{S, R, T, C\}$$

Where S is the symbol representing a square, R representing a rectangle shape, T, a triangle and C a circle. The non-terminal objects are made up of the objects defining the structure of the password image. These are the start symbol S , holders generating the terminal objects S_1 and non terminal object generating either a terminal or empty shape S_2 . The rest of the sections design the grammar using these objects.

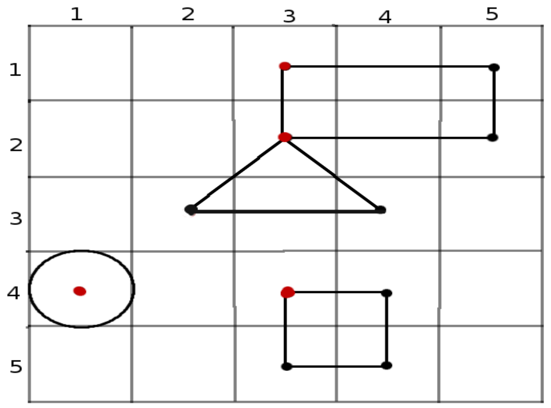

Figure 2: *Illustration of the reference points of each of the terminal shapes in grammar PassImages in the color red.*

Password drawing area: For PI, the password image to fit onto the drawing grid, every element of PI has to be at position (x, y) that does not have the shape boundary outside of the grid. We therefore define *PboundsMet()* for each terminal shape. *PboundsMet()* for any shape is dependent on the reference point of a shape. For each of the terminals, the reference point will be reffered to as the top most left most grid block the shape is drawn on. Figure 2 shows the reference points of each of the terminal shapes in red.

Analysing the reference points and size of each terminal shape as well as the size of the grid we can determine the points which will result in the terminal shape still, staying within bounds of the grid.

For reference point (x, y) , a square can be placed on the grid if there is a space for one unit to the right one unit down of the point (x, y) . Using the axixs system presented in Figure 2, a square can be placed at position (x, y) if $x+1\leq 5$ and $y+1\leq 5$ and there is no square at the same position. To place a circle at position (x, y) the point should not contain another circle at the same position. For the triangle, $(x+2)\leq 5$ and $(y-1)\geq 1$. The point (x,y) is suitable for a rectangle if $x+2\leq 5$ and $y+1\leq 5$. This is what is used to define the PboundsMet(shape) for each shape as it holds true if the conditions above are met.

Allowed Length of the password image: Using the grammar model and the length restriction in the requirements given by [10] we come to the passImage grammar:

Proc. of the Second Intl. Conf. on Advances In Computing, Communication and Information Technology- CCIT 2014.
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-051-4 doi: 10.15224/ 978-1-63248-051-4-139

$$passImages=\{V_N,V_T,R,I\}$$

$$V_N=\{S,S_{1,}S_2\}$$

$$V_T=\{<S,P,V>,<C,P,V>,<T,P,V>,<R,P,V>\}$$

$$R=(0)S\rightarrow S_{1_1}S_{1_2}S_{1_3}S_{2_1}S_{2_2}\dots S_{2_{12-3}}$$

$$(1)S_2\rightarrow\varepsilon \quad : \quad length(PI)\geq 8$$

$$(2)S_2\rightarrow S_1 \quad : \quad length(PI)+1\leq 20$$

$$(3)S_1\rightarrow<S,P,V> \quad : \quad PboundsMet(S)$$

and

$$length(PI)+length(S)\leq 20$$

$$(4)S_1\rightarrow<C,P,V> \quad : \quad PboundsMet(C)$$

and

$$length(PI)+length(C)\leq 20$$

$$(5)S_1\rightarrow<R,P,V> \quad : \quad PBoundsMet(R)$$

and

$$length(PI)+length(T)\leq 20$$

$$(6)S_1\rightarrow<T,P,V> \quad : \quad PboundsMet(T)$$

and

$$length(PI)=length(T)\leq 20 \quad I=\{S\}$$

PassImages is instantiated from the grammar model by replacing L_{min} and L_{max} with 8 and 20 respectively. X_{min} and X_{max} are also substitutted by 3 and 12 respectively in the grammar model. The position(P) and attributes(V) are set by the user when selecting the position and color of the shape as well as the rotation.

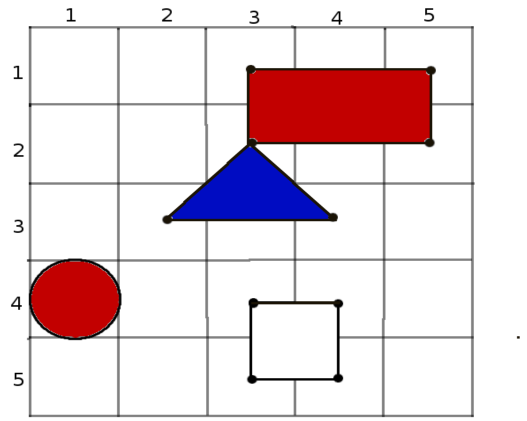

Figure 3: *An example of a password image generated by PassImages.*

Attributes of the grammar objects: We add attributes relating to the colour of the shape. This means addition of rules that can change the colour of the shape. We set the possible shape colours to red, blue and white. These are selected merely as a proof of concept to show application of the grammar passImages.

For colour, we add the following rules:

tiV.colour=white → tiV.colour=white

tiV.colour=blue → tiV.colour=blue

tiV.colour=red → tiV.colour=red

Figure 3 presents an example of an image generated by passImages. The generation process that generates the password image is starts from the start symbol S and progresses as shown below:

$$S\rightarrow S_{1_1}S_{1_2}S_{1_3}S_{2_1}\dots.S_{2_9}$$

$$S_{1_1}\rightarrow<C,(1,4),\{(colour=r)\}>$$

$$S_{1_2}\rightarrow<T,(3,2),\{(colour=b)\}>$$

$$S_{1_3}\rightarrow<R,(3,1),\{(colour=r)\}>$$

$$S_{2_1}\rightarrow S_1$$

$$\rightarrow<R,(3,4),\{(colour=w)\}>$$

$$S_{2_2}\rightarrow\varepsilon$$

$$S_{2_3}\rightarrow\varepsilon$$

...

$$S_{2_9}\rightarrow\varepsilon$$

The generation is not unique and a usercan very generate the same password image in a different order of placement of the shapes.

IV. Significance

Visual passwords have been shown to be lighter on memory strain than alphanumeric passwords due to the picture superiority effect. They have however not as of yet been predominantly used due to difficulty in implementing secure and efficient visual password schemes. Our work pushes in the direction of supporting visual passwords. We present a grammar which can generate password images that follow a syntax. This also opens up a new authentication mthod who may find it difficult to read of write due to being able to identify text symbols.

The shape grammar also improves on the user experience with the grammar enforciong the syntax of the password to be generated. Unlike some password schemes which let you enter a password and then let you know if it is correct, the grammar only allows at every step of design images that will be allowed in the password scheme.

While our work presents the grammar for authentication schemes purposes, this work can be generallized into other areas such as picture generations and graphics helping in the design of images . This allows for a new method of generating and storing images. Any image can be represented by a set of rules placing primitive objects at some position on a grid. This will allow for easier processing and lighter memory storage of images as only the rules storing the image are to be stored instead of a graphic file. For instance object recognition would be easier as all that would be required is analysing the text for an object as opposed to current image manipulation methods.

V. Conclusion

Our work looks into the body of knowledge and highlights the problems associated with implementation of visual password schemes. The literature shows that one of the major concerns, is the type of images that visual password schemes use.

From looking into the existing literature work that has been done in analyzing design of secure password images is adopted. We then present a grammar that enforces the generation of only the secure password images. We present a context sensitive grammar using attributed shape objects to design a grammar model used for generating images following a specific syntax. As proof of concept, *passImages* is instantiated from the model to show use of the grammar model.

This grammar model also introduces a new method that can be used to store images that make object recognition a easier as the anylysis is done through the rules instead of the actual image. An image is looked at as a collection of objects on a grid space.

The grammar model is general enough such that a grammar can be defined by describing the primitive shapes of the language of images and the rules that all images must follow.

Future work can look into a general control grammar which selects the attributes and positions of the shapes while still staying true to the syntax the grammar defines. FUture work could also be targeted at coming up with image analysis methods which take into account the structure of an image being a set of rules.

Acknowledgment

The authors appreciate and send a thank you to the technical advise and direction from Ishmael Makitla having always being the person to run ideas through.

References

[1] I. Jermyn, A. Mayer, F. Monrose, M. Reiter and A. Rubin. “The design and analysis of graphical passwords.” In Proceedings of the 8th usenix Security Symposium, pp. 1-14, *1999*

[2] *K. Renaud and A. De Angeli. “Visual Passwords: cure-all or snake oil?” Communications of the ACM, 2009, vol. 52, no. 12, pp. 135-140*

[3] R. Shepard. “Recognition memory for words, sentences, and pictures” Journal of Verbal Learning and Verbal Behaviour, 1967, vol. 6, no. 1, pp. 156-163

[4] X. Suo, Y. Zhu and G.DS Owen. “Graphicall passwords: a survey” Computer Security Applications Conference. 21St Annual, 2005

[5] G, Stiny,J Gips . “Shape Grammars and the Generative Specification of Painting and Sculpture”. In Information Processing ,1971, Vol. 71, pp 1460–1465

[6] T. Schnier, and J. Gero. ”Learning genetic representations as alternative to hand-coded shape grammars.” In Artificial Intelligence in Design, 1996, pp. 39–57. Springer Netherlands.

[7] A. Lashkari, D. Zakaria, S. Farmand, O. Bin, and R. Saleh, “Shoulder Surfing attack in graphical password authentication. “ International Journal of Computer Science and Information Security, 2009, vol. 6, no. 2, pp. 145–154.

[8] A. De Angeli, L. Coventrya, G. Johnsona, and K. Renaud, “ Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems.” International Journal of Human-Computer Studies, 2005, vol. 63, no. 1, pp. 128–152.

[9] I.E, Liao, C_C. Lee and M.S. Hwang, (2006). “A password authentication scheme over insecure networks.” Journal of Computer and System Sciences, 2006, vol.72, no. 4, pp. 727–740.

[10] M. Rasekgala, S. Ewert, I. Sanders and T Fogwill, “Requirements for secure graphical password schemes” In IST-Africa Conference Proceedings, 2014, pp.1-10

[11] D.E. Knuth, “ Semantics of context-free languages. Mathematical” Systems Theory, 1968, vol. 2 , no. 2 , pp.127–145.

[12] P. Wonka, M. Wimmer, F. Sillion and W. Ribarsky, “Instant architecture.” ACM Trans. Graph., 2003, vol.22 , no.3, pp. 669–677.

[13] P. Muller, P. Wonka, S. Haegler, A. Ulmer and L. Van Gool, “Procedural modeling of buildings.” ACM Transactions on Graphics- Proceedings of ACM SIGGRAPH , 2006, vol.25, no. 3, pp.614–623

[14] J. Thorpe, and P. van Oorschot . “Towards secure design choices for implementing graphical passwords.” In Computer Security Applications

Conference , 2004, pp. 50–60. IEEE.

[15] dd

[16] C. Tsai, C. Lee, and M.S. Hwang, “Password Authentication Schemes: Current Status and Key Issues.” Network Security, 2006, vol. 3, no.2, pp.101–115.

[17] fdsf

[18] B. Colakoglu, “ Design by Grammar. In Predicting the Future” 25th eCAADe Conference Proceedings, 2007, Vol.22, pp.919–925 .

G. Stiny, G. “Introduction to shape and shape grammars.” Environment and Planning B , 1980, vol. 7, no.3, pp. 343-351.