

Secure Recognition-Based Graphical Authentication Scheme Using Captcha and Visual Objects

Altaf Khan

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
July 2015
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Serhan Çiftçiođlu
Acting Director

I certify that this thesis satisfies the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Iřık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Assoc. Prof. Dr. Alexander Chefranov
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Alexander Chefranov

2. Asst. Prof. Dr. Gürcü Öz

3. Asst. Prof. Dr. Önsen Toygar

ABSTRACT

Graphical password is an alternative scheme of alphanumeric password that is very tiresome process to recall the complex password. Psychological studies of human mind argue that recalling of image is easier than alphabets or digits. In this thesis, recognition based authentication built on Captcha technology is proposed. I propose method "Click-on-Captcha-Objects", which contains Captcha based visual objects (letters of any language, digits, and user-defined images); it helps memorability of a strong password.

Proposed method was analyzed by two different approaches. One of them is usability and another one is security. For usability, 40 users participated in test to analyze how many users remember the complex password. Accuracy of the proposed scheme (percentage of users remembering the strong password) was 97.25%, in contrast to Captcha + text and Click-Text methods having accuracy of 88.75% and 93%, respectively. On the other hand, security analysis of the proposed system has been done with different types of attacks, popular four Captcha breakers software are used for Captcha images recognition. The proposed system resists in 98.5% cases against four Captcha breakers attacks. In contrary, Click-Text method resists in 95.5% cases. In addition, auto-mouse clicked attack analyzed; the performance against the attack of the proposed method was 97.66%, and Click-Text method performance was 95.41%. The results indicate that for proposed method it is easy to remember the strong password compared to alphanumeric and Click-Text based authentication schemes. Hence, performance of the proposed method is better than alphanumeric and Click-Text method. Traditional schemes of authentication mostly lead to guessable and unreliable password, but "Click-on-Captcha-Objects" provides

reasonable security and usability to authenticate a legitimate user. In order to check the time of generation of each image at server, an experiment has been performed at SAMSUNG (Core i5, RAM 4 GB, Processor 2.53 GHz) laptop and the result was approximately 40 milliseconds per "Click-on-Captcha-Objects" image.

Keywords: Graphical based authentication, secure password, Captcha based authentication, Click-on-Captcha-Objects

ÖZ

Grafiksel şifreleme; karmaşık şifrelerde yorucu bir işlem olan, alfanumerik şifrelemeye bir alternatiftir. İnsan zihni üzerine yapılan psikolojik çalışmalar, görüntü anımsamanın harf veya rakam anımsamaya kıyasla daha kolay olduğunu savunurlar. Bu çalışmada kimlik doğrulamaya dayalı Captcha teknolojisi geliştirilmiştir. Karmaşık şifreleri hatırlamaya yardımcı olan Captcha tabanlı görsel nesnelere (harf, rakam ve kullanıcı tanımlı görseller) içeren “Click-on-Capcha” tekniği önerilmiştir.

Önerilen bu teknik, iki farklı yöntem ile analiz edilmiştir. Bu yöntemlerden biri kullanılabilirlik, diğeri ise güvenlidir. Kullanılabilirlik analizi için, 40 farklı kullanıcı karmaşık şifreleri hatırlayabilmek adına bir teste katılmışlardır. Önerilen sistemin doğruluğu, diğere bir deyişle zor şifreleri hatırlayan kullanıcıların oranı %97.25’ dir. Bu, Captcha + Text (%88.75) ve Click - Text (%93) tekniklerine göre çok daha iyi bir orandır. Diğere yandan, önerilen sistemin güvenilirlik analizi, çeşitli saldırılar ve dört popüler Captcha kırıcı yazılımlar kullanılarak yapılmıştır. Önerilen bu sistem, %98.5 saldırı olayını engellerken, Click-Text tekniği %95.41’ini engelleyebilmiştir. Buna ek olarak, “auto-mouse clicked” saldırısı analizinde, önerilen tekniğin performansı %97.66 iken Clk-Text tekniğinin performansı %95.41’dir. Sonuçlar gösteriyor ki; önerilen teknik kullanılarak yapılan karmaşık bir parolayı hatırlamak, alfanumerik ve Clk-Text tabanlı kimlik doğrulamaya kıyasla daha kolaydır. Bunun sonucu olarak, önerilen bu tekniğin performansı bahsi geçen diğere metotlara nazaran daha iyidir. Geleneksel kimlik tanımlama, çoğunlukla tahmin edilebilir ve güvenli olmayan şifrelemeye sebebiyet verir fakat, “Click-on-Capcha-Object” kullanıcılar için makul bir güvenlik ve kullanılabilirlik sağlar. Her

görüntünün oluşturulma süresini kontrol etmek için deney, SAMSUNG (Core i5 4 GB RAM, 2.53 GHz İşlemci) dizüstü bilgisayar kullanılarak yapılmış ve her “Click-on-Captcha-Objects” görseli için sonuç yaklaşık olarak 40 milisaniye olarak gözlemlenmiştir.

Anahtar kelimeler: Grafik tabanlı kimlik doğrulaması, güvenli parola, Captcha tabanlı kimlik doğrulaması, Click-on-Captcha-Objects

DEDICATION

I commit my thesis work to my family and numerous companions. A unique feeling of appreciation to my cherishing folks, their inspirational statements and push for determination ring in my ears. My siblings who have never walked out on me and are extremely exceptional.

I likewise commit this paper to my numerous companions and their families who have bolstered me all through the procedure.

ACKNOWLEDGMENT

I might want to express my exceptional gratefulness and because of my supervisor Assoc. Prof. Dr. Alexander Chefranov, you have been a huge guide for me. I might want to thank you for empowering my exploration and for permitting me to develop as an examination researcher.

I would like to thank Prof. Dr. Işık Aybay, Chairman of Computer Engineering department, who provided us varies of research facilities. I shall be thankful to Asst. Prof. Dr. Önsen Toygar, Asst. Prof. Dr. Gürcü Öz and all those who have shared time with me for their understanding and companionship through the years, and special thanks to those who reviewed my thesis and provide me constructive criticism and feedback on my work over the years.

TABLE OF CONTENTS

ABSTRACT.....	iii
ÖZ.....	v
DEDICATION.....	vii
ACKNOWLEDGMENT.....	viii
LIST OF TABLES.....	xiii
LIST OF FIGURES.....	xiv
LIST OF SYMBOLS/ABBREVIATIONS.....	xvi
1 INTRODUCTION.....	1
1.1 Introduction to graphical password.....	1
1.2 Thesis statement.....	2
1.3 Main contribution.....	2
1.4 Document structure.....	3
2 REVIEW OF RECOGNITION-BASED GRAPHICAL AUTHENTICATION METHODS.....	4
2.1 What is authentication.....	4
2.1.1 Types of authentication.....	4
2.1.1.1 Physical trait-based authentication.....	4
2.1.1.2 Token-based authentication.....	5
2.1.1.3 Knowledge-based authentication.....	5
2.2 Graphical passwords.....	7
2.2.1 Types of graphical passwords.....	8
2.2.1.1 Recall-based graphical passwords.....	8
2.2.1.2 Cued-recall-based graphical passwords.....	9

2.2.1.3 Recognition-based graphical passwords.....	9
2.3 Review of recognition-based authentication schemes.....	10
2.4 Security issues of recognition-based graphical passwords.....	16
2.5 Summary of security and attacks.....	16
2.6 Motivation.....	17
2.7 Problem definition.....	17
3 DESCRIPTION OF SECURE RECOGNITION-BASED AUTHENTICATION ALGORITHM USING CAPTCHA AND VISUAL OBJECTS (CLICK-ON- CAPTCHA- OBJECTS).....	19
3.1 Definition of "Click-on-Captcha-Objects" algorithm.....	19
3.2 Theoretical concept of "Click-on-Captcha-Objects" algorithm.....	20
3.2.1 Captcha image of visual objects.....	21
3.2.2 Password complexity of "Click-on-Captcha-Objects" algorithm.....	32
3.2.3 Stages of " Click-on-Captcha-Objects" algorithm.....	33
3.2.3.1 Insertion of user defined objects in Captcha image.....	33
3.2.3.2 Register stage of "Click-on-Captcha-Objects" algorithm.....	34
3.2.3.3 Authentication stage of "Click-on-Captcha-Objects" algorithm.....	34
3.3 Structural representation of the proposed algorithm.....	35
3.3.1 Flowchart of the "Click-on-Captcha-Objects" algorithm.....	35
3.3.2 Description of the flowchart of "Click-on-Captcha-Objects" algorithm.....	37
4 DESIGN, IMPLEMENTATION AND TESTING OF "CLICK-ON-CAPTCHA- OBJECTS" ALGORITHM.....	42
4.1 Design of "Click-on-Captcha-Objects" algorithm.....	42
4.1.1 Interface of registration.....	43
4.1.2 Interface of authentication.....	43

4.2 Implementation of "Click-on-Captcha-Objects"	44
4.2.1 Implementation of registration phase.....	45
4.2.2 Implementation of authentication phase.....	47
4.3 Testing of "Click-on-Captcha-Objects" scheme.....	48
5 IMPLEMENTATION OF KNOWN METHODS AND COMPARISON WITH PROPOSED "CLICK-ON-CAPTCHA-OBJECTS" METHOD.....	50
5.1 Implementation of "Click-Text" and "Captcha + Text" schemes.....	50
5.2 Example of three password schemes.....	51
5.2.1 Example of "Click-on-Captcha-Objects" proposed scheme.....	51
5.2.2 Example of "Click-Text" scheme.....	51
5.2.3 Example of " Captcha+Text " scheme.....	51
5.3 Security analysis of Captcha generated images.....	52
5.4 Comparison of several attacks on known and proposed methods.....	56
5.4.1 Comparison results of Captcha breakers attacks.....	57
5.4.2 Auto mouse click attack.....	60
5.4.3 Guess-ability attacks.....	61
5.4.4 Brute force and dictionary attacks.....	62
5.5 Comparison of password complexity and memorability.....	62
5.6 Survey of same the complex password of three schemes.....	64
5.7 Comparison of convenience usability and time of authentication.....	66
6 CONCLUSION AND FUTURE WORK.....	70
REFERENCES.....	73
APPENDICES.....	79
Appendix A. "Click-On-Captcha-Objects" Image Generation Code.....	80
Appendix B. Selection of Object by Clicked on Image.....	81

Appendix C. Implementation of Hash Function.....	82
Appendix D. Add User Defined Images/ Objects.....	83
Appendix E. User Defined Function to Store User Information in Database.....	84
Appendix F. OCR Results of Captcha Image (Online i2ocr).....	86
Appendix G. Screenshot of Captcha Breaker Software	87
Appendix H. Interface of Captcha Alphabets Image as Graphical Password.....	88
Appendix I. Captcha Breaker Screenshot of Proposed Scheme Image.....	89
Appendix J. Implementation of "Captcha + Text" Registration and Authentication.....	90
Appendix K. Proposed Method Interface of Registration and Authentication.....	91
Appendix L. Implementation of Proposed System Registration and Authentication.....	93
Appendix M. Users Collection Data Related to Login Time against Three Schemes.....	94
Appendix N. Memorable Results of same Complex Passwords of Three Schemes.....	97

LIST OF TABLES

Table 4.1. C#.Net function for implementation of proposed method.....	44
Table 5.1. Captcha Breakers Software Results of Initial Generated Captcha Images of Figure 5.1 (a)-(d).....	54
Table 5.2. Results of Captcha Breaker Against Captcha Images of Figure 5.2(a)-(d).....	55
Table 5.3. Results of Captcha Breakers of Proposed Method Images of Figure 5.4..	58
Table 5.4. Captcha Breaker Results of Click-Text Method of Figure 5.5.....	59
Table 5.5. Comparison of Auto Mouse Click of Click-Text and Proposed Method..	60
Table 5.6. Comparison of Complex Password Memorability of Three-Schemes.....	63
Table 5.7. Ease Use of Click-on-Captcha-Objects Question to Users.....	66
Table 5.8. Authentication Time (s) of Three-Password Schemes.....	67
Table 5.9. Time (s) of Authentication of Known Methods (Click-Text and Captcha+ Text [1]).....	69

LIST OF FIGURES

Figure 2.1. DAS Grid of Graphical Password [15].....	8
Figure 2.2. The "PassPoint" Example of Cued- Recall Based Authentication [19]....	9
Figure 2.3. PassFaces Challenge Screen [20].....	10
Figure 2.4. Déjà VU Scheme of Authentication [21].....	11
Figure 2.5. Triangle Method of Graphical Password [26].....	12
Figure 2.6. Moveable Frame Schemes [26].....	12
Figure 2.7. Distorted Image Scheme [24].....	13
Figure 2.8. Example of Alie Algorithm[25].....	14
Figure 2.10. CAPTCHA of "smwm" Generated by Captcha[36].....	14
Figure 2.10. Captcha Generated Alphabets Image.....	14
Figure 2.11. Example of Click-Text [1].....	16
Figure 3.1. Initial Stage of Captcha Image Scheme Captcha based Image.....	23
Figure 3.2. Rectangle Object Coordinates	25
Figure 3.3. Representation of Objects (Wave Form) on Image.....	27
Figure 3.4. Two Triangle of One Rectangle to Recognize the Point.....	30
Figure 3.5. Two Point P1, P2 Lie on Line L	31
Figure 3.6. Request Response of User Authentication System	35
Figure 3.7. Flowchart of "Click-on-Captcha-Objects" Algorithm.....	36
Figure 3.8. Flowchart of Captcha Image Generation.....	37
Figure 3.9. Flowchart of How to Select Object.....	38
Figure 3.10. Flowchart to Check User-Name Already Exist or Not Exist.....	40
Figure 3.11. Flowchart to Get Image and Save Database.....	41
Figure 5.1. Captcha Generated Images with Different Parameters.....	53
Figure 5.2. Captcha Clickable Image	55

Figure 5.3. Proposed Scheme, "Click-on-Captcha-Objects" Output Image.....	56
Figure 5.4. Proposed Method Generated Images.....	57
Figure 5.5. Click-Text Scheme [1] Images Used in Captcha Breakers.....	58
Figure 5.6. Graph of Comparison of Captcha Attack Against Proposed and Click-Text Scheme.....	59
Figure 5.7. Graph of Memorable Password of Three Schemes in 3 Days Survey.....	65
Figure 5.8. Average Time(s) of Authentication of Three Methods.....	68
Figure 5.9. Authentication Time(s) of 40 Users Against Three Algorithms.....	68

LIST OF SYMBOLS/ABBREVIATIONS

DAS	Draw a Secret
ATM	Automated Teller Machine
Captcha	Completely Automated Public Turing test to tell Computers and Humans Apart
CaRP	Captcha as Graphical Password
RBGP	Recognition-Based Graphical Password
OCR	Optical Character Recognition
OS	Operating System
AI	Artificial Intelligence
SD	Standard Deviation

Chapter 1

INTRODUCTION

1.1 Introduction to graphical password

In practical life, everyone has resources; to make them secure, the locks and cabinets are used. Locks are used to hide secret resources, these all are physical resources and human used different ways to construct security for them. When user wants to cover out his/her private resources in form of electronics materials in the computer, then the authentication problem occurs. Therefore, the user has to know some characters, string or some digits to authenticate him/herself which should be kept secret from others. These characters or strings are conceived as a password. To authenticate a legitimate user, password recalled by user is to pass the security attempt. Password may be characters of any language like English alphabets or with some digits etc. In modern world, user name and password commonly are used during login process and to access the control of computer system, email, ATM machines, online money transfer etc.

Authentication can be categorized into three types, discussed in chapter 2. Most important authentication method is knowledge-based authentication scheme. In this scheme, user will remember the set of alphabets or digits to authenticate him/her to access the privacy profile.

In contrast, password breaker or brute force attack can easily access the simple selected password. Therefore, alphabetic password is not enough to make secure and reliable system. Recently, Graphical password scheme was proposed to enhance security and reduce attacks to crack the password. Graphical password scheme provides authentication of genuine claim of user using images, or visual pattern, which easily user can understand and can pass the challenge as compared to robots or system attacks. Number of researchers nominates the graphical password scheme with different angles, after usability and security study, Captcha-based authentication scheme provides more meaningful authentication for humans and reliable protection against online attacks. Recently, "Captcha as graphical password [1]" method was proposed, which is built at "Captcha is hard AI problem [2]", password is selected by click on correlated characters. The clicked characters or digits of Captcha image become password of the corresponding user. To reduce the guess-ability and enhance the level of security to construct strong password, proposed method is established. Details of the proposed scheme are explained in chapter 3.

1.2 Thesis statement

To reduce the guessing attacks and increase the memorability of password, secure graphical recognition based authentication scheme is proposed. The proposed algorithm is built on Captcha and visual objects. Visual objects are combination of alphanumeric characters, special symbols or user defined alphabets, and user defined images.

1.3 Main contribution

My main contribution includes a method, which analyzes the security measurements and makes it stronger using knowledge based user authentication. In addition, it illustrates how to make easy for user to remember and how it will be strong against

online attacks. "Click-on-Captcha-Objects" (Secure Recognition-based Authentication using Captcha and Visual Objects) also provides support of wide range length of password. I compared three passwords schemes, and their implementation. Previous graphical password method "Click-Text" [1], and alternative is proposed method "Click-on-Captcha-Objects" and third one is alphanumeric with Captcha. "Click-on-Captcha-Objects" avoids the weakness of existing method; limited number of alphanumeric set [1] and simple words based password (easy to guess)". In addition, I performed the real test bed for robotics attacks experiments, to recognize the 2D alphanumeric characters and 2D objects for analysis of graphical password.

1.4 Document structure

Whole document categorizes into five chapters. Chapter 2 is "related work of graphical recognition based authentication". Chapter 3 is "defining the problem and structure of proposed method". Chapter 4 represents "the design and implementation of "Click-on-Captcha-Objects"", chapter 5 concerns "implementation of previous method, performance of previous, and proposed method comparison results". Chapter 6 is "conclusion and results".

Chapter 2

REVIEW OF RECOGNITION-BASED GRAPHICAL AUTHENTICATION METHODS

2.1 What is authentication

Authentication is verification of user-ID to identify the legitimate user [3]. Hence, the authentication is the basic step of security, and authentication protocols are necessary part of entire secure systems. Authentication has two main aspects, one is security and another is usability. Both of them cannot be ignored to make reliable secure system. Although security researchers have made great paces against security threats to protect the system including individual traits of users (intrinsically), token based (User ID or passport) and knowledge based approaches. Authentication of user provides accessibility to users to their unique resources. So, is user genuine or not? It is very important to protect particular user information. To authenticate the user, there are different authentication methods discussed below.

2.1.1 Types of authentication

Generally, authentication methods are classified into three different categories: physical traits of user, extrinsic token based, and knowledge based authentication.

2.1.1.1 Physical trait-based authentication

It is biometric authentication, in which biometric information of a user is taken during the authentication to give access to the private system. An automatic system will verify the user on the base of his/ her physical behavior characteristics or traits. Every user has unique physical traits like iris pattern, face, fingerprints, and palm

prints, hand geometries and voice sound etc [4]. These traits are constant and it vary from user to user. Therefore, it is impossible that two user traits can match. It is assumed to be the best solution of secure system but in present, unfortunately user authentication can achieve exclusively through technical innovation.

2.1.1.2 Token-based authentication

Token-based authentication is based on tokens like a key, passport, smartcard and badge to identify the users. This information is used to give access to the secure system. The authentication is based on a key or id number, etc. Therefore if it is stolen or lost, the fake person can access the secure resources. User will always login with use of specific token.

2.1.1.3 Knowledge-based authentication

Knowledge based authentication is something you know and this secure information is used to grant access to user privileges. Like PIN, alphanumeric characters, digits, text in the form of a password, which is remembered by legitimate user. In this authentication, two authentication techniques are nominated, textual and graphical user authentication. Textual user authentication is based on digits or alphanumeric characters, graphical based authentication techniques, is based on image, graphical 2D objects like picture etc.

In addition, there are different authentication ways, in which a user can be authenticated. Location based authentication system insures that the same user ID or user Card login at one location at same time [7]. Moreover, another is time based authentication just allowing a user to access in specific time slots.

Since the most useful authentication is knowledge-based authentication, a mostly applications of internet, E-mail servers, social networks or distributed systems use

knowledge based authentications to verify the credential ID. Knowledge based password is strong as compared to token-based [6]. The main steps of knowledge based authentication are; user will enter his/her ID and secret key, may it be alphanumeric or digits, special character, etc., the secret key will be verified to decide the user that is genuine or imposter? Therefore, when he/she tries the system will decide that he/she is genuine or imposter. If genuine, the user can pass-out and access his/her privileges. There are many research studies, which represent that textual based password has limitation of memorability and it can be cracked [6]. To make good secure password, it should be at least 8 characters long with some digits and capital letters.

In contrast, there are many techniques proposed like graphical password. Graphical password corresponds to a text-based password. First graphical password concept was proposed by Blonder in 1996 [7]. The preference of graphical password is based on some psychological research results [8]. In addition, the graphical/visual object or image is easier to remember than a text based password [9].

The graphical password can be defined as an authentication system based on images or visual objects or visual characters. The graphical-based authentication is divided into three main types: *Recognition-based authentication*, *Recall-based* and *cued-recall based authentication*. By using recognition-based techniques, user shall recognize the visual objects and images to be authenticated. In recall-based techniques, a user has to re-produce something that he or she created during the registration stage. In cued-recall based authentication, the information is available in an image, which helps users to remember the selected password. First, I discuss the alphabetic password and its drawbacks; then, I focus on Graphical password

technique. There are many research studies, which represent the textual based password limitation of memorability and predictability [6]. In addition, the password should not have repeating characters, simple dictionary words, neither password should be string of sibling name, personal information, or home number etc., that is easily guessable and brute force can attack it easily [10]. It means, a password becomes strong but it is very hard to remember it for a user. "The survey of text based password revealed that current textual based password can be recovered about 80% in 30 second" (Xiaoyuan et al. 2005). Recently researchers invented new spread of a Keylogger spyware [11], which captures user's information during login and sends them to attacker; hence, the password should be extremely kept far away from this weakness. In addition, each user has multiple passwords and to remember each of them with unique account is contusive [12].

2.2 Graphical password

In contrary, there are many other techniques also proposed like graphical password. First graphical password concept was proposed by Blonder in 1996 [7]. The preference of graphical password is based on some psychological research results [8]. In addition, the graphical/ visual object or image is easier to remember than text based password [9].

Graphical password methods provide many ways of solution to user-caused vulnerabilities of text-password such as simple words, family name etc. User can choose his/her password by using images or visual objects. Graphical password is a knowledge-based protocol for user authentication. It enables user to remember the password in the form of graphical objects or images.

2.2.1 Types of graphical passwords

The common taxonomy of graphical password system is recall, cued-recall and recognition. This report focuses on Recognition based authentication because of my research domain.

2.2.1.1 Recall-based graphical passwords

In this scheme, user has to draw an image during registration, and this same process will be repeated at authentication time. It means user will draw some secret objects/ lines/ points on plain area; an example for this category is "Draw a Secret" (DAS) scheme which was proposed by Jermyn et al. (1999) [15]. Figure 2.1 gives an example of DAS. There are many new researches proposed related to Recall based authentication. They have a drawback to remember the password and security issues [16]. Another study report represents that DAS and Pass-Go (Pass-Go based on Draw a Secret methodology) password successfully access with guessing 2^{31} to 2^{41} entries [17, 18].

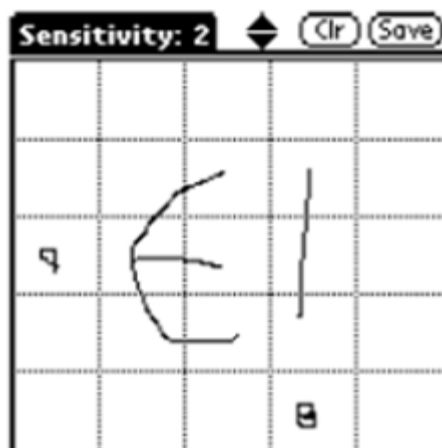


Figure 2.1. DAS Grid of Graphical Password [15]

2.2.1.2 Cued-recall graphical passwords

In this scheme, the user will choose some memorable point from the image and recall them at authentication time. Cued recall actually provides help to users to remember the password; the background image has many locations, points that users can easily remember. Hence, user will select some of them as his/her password. "PassPoint" is an example of a graphical password. Wiedenbeck et al. [19] proposed this method to authenticate a user. The figure 2.2 shown below illustrates it.

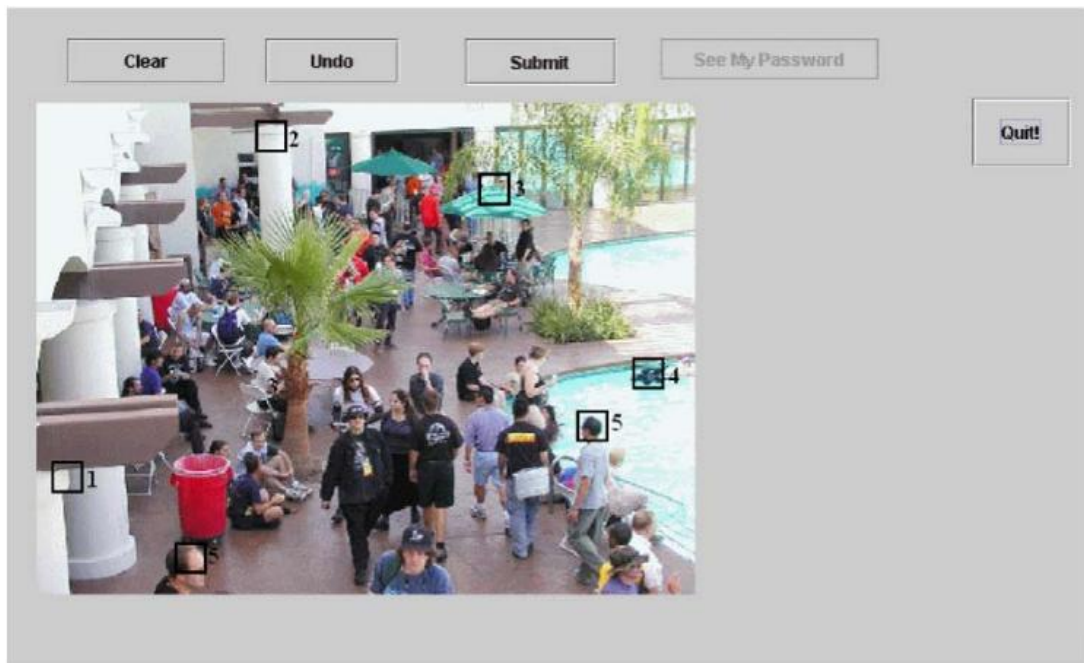


Figure 2.2. The "PassPoint" Example of Cued- Recall Based Authentication [19]

Unfortunately these techniques have own security issues like, PassPoint's image contains hotspot points which help to break significant part of a password with dictionary attack of 2^{26} to 2^{35} entries as compared to full length of 2^{43} [1,19].

2.2.1.3 Recognition-based graphical passwords

In recognition-based graphical password technique, a user interacts with a set of images or visual objects and selects images or objects. In registration phase, a user has to enter his/her name and then selects some images or objects as a password. At

the authentication step, a user will recognize the same objects or images, and the system will identify the user.

2.3 Review of recognition based authentication schemes

In recognition-based authentication scheme, there is a number of proposed methods to authenticate a user. Id Arts [20] exposed a technique, like pass images, based on PassFaces. In this method, a user selects a set of face images to authenticate, which he/she has selected at the registration time. In Figure 2.8, each panel has nine alternative different faces, and it consists of four challenge panels.



Figure 2.3. PassFaces Challenge Screen [20]

In 2000, Déjà vu [21] and Darren Davis [22], respectively proposed the new recognition authentication methods, Déjà vu scheme and story scheme. In Déjà vu scheme, the set of pass-images is generated randomly, and in story scheme (shown in Figure 2.4), a user selects a password as a sequence of unique images to make a story from the lots of images. These images may belong to food, animal or anything else

[22]. Nevertheless, the problem of both of them is taking too long time to identify the user as compared to text-based authentication.

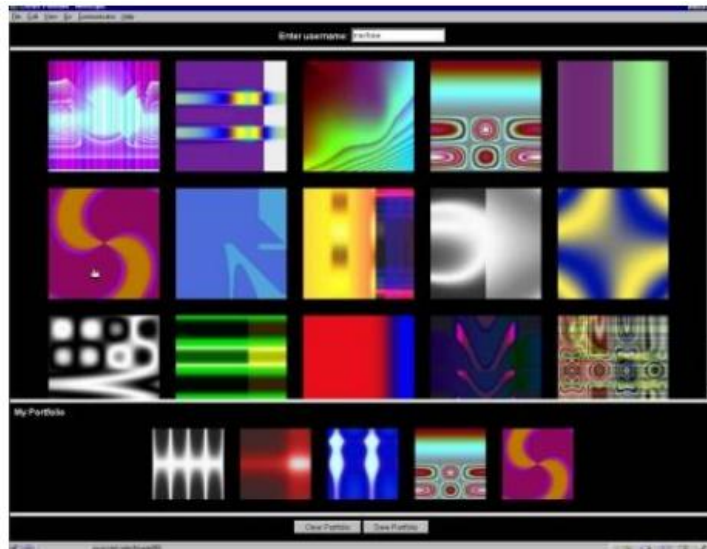


Figure 2.4. Déjà VU Scheme of Authentication [21]

Sobrado and Birget in 2002 [26], proposed a new method to avoid the shoulder surfing attacks, called *triangle algorithm*. In this method, during registration phase, a user chooses a number of objects from a given set of objects. In login step, the same objects are selected. The user repeats the same process several times, and each time the objects will be shuffled on the panel. There were 1000 proposed objects for a user to choose as a password; disadvantage was that to find the same object from a crowd is difficult, and it takes long time to authenticate a user [23]. On the other hand, this technique requires large display screen to find pass objects. The figure 2.5 shows the method of triangle.

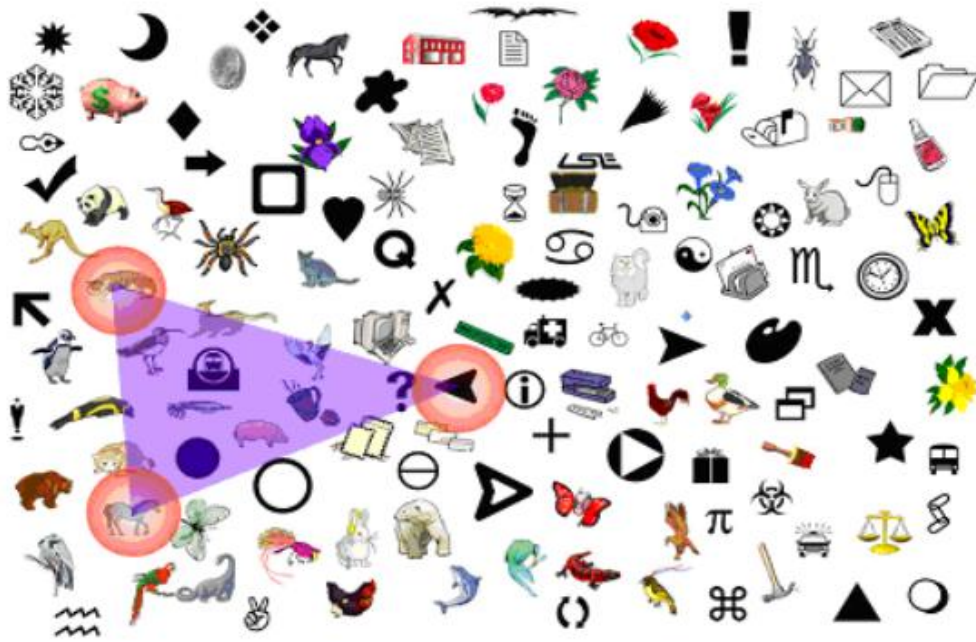


Figure 2.5. Triangle Method of Graphical Password [26]

Second enhancement in the triangle method was the 'moveable frame scheme'; it also requires a user to recognize the objects to pass the authentication. In this scheme, just "3 objects are displayed, and one of them is placed in a moveable frame ". The user has to rotate the frame until the other two objects get line up on frame. Example is shown in figure 2.6, but unfortunately, this process also required long time to achieve high security.



Figure 2.6. Moveable Frame Schemes [26]

In 2008, Eiji Hayashi [24] proposed a new graphical password technique called "your illusion". It has three stages, portfolio, creation practice and authentication. In portfolio step, a user generates a set of images, which he/she will use in the authentication. When images are produced, they shift to authentication step, and the images are distorted to resist the recognition attacks. Output images are known as portfolio. In practice stage, a set of portfolio images and decoy images are practiced and the system will give a feedback, whether it is correct? In the authentication stage, the user chooses the correct portfolio image from the given set. Decoy images are created from the original input images, and the noise level should be enough that details of original images are blotted out [24].



Figure 2.7. Distorted Image Scheme [24]

In 2012, Alia proposed new method of authentication [25]. In this method, a user has to select some shapes from side bar menu. At the registration step, the user has to select minimum of five shapes to select his/her password. During the authentication, it is necessary to select the same shapes for successful authentication. Problem of this method was easily guessable.

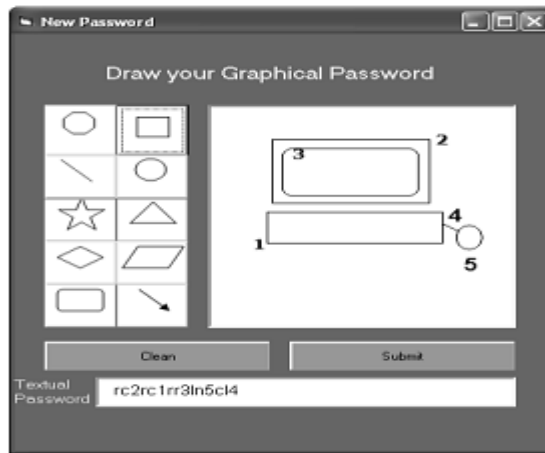


Figure 2.8. Example of Alie Algorithm[25]

In 2014, Captcha as Graphical Password (CaRP) method was proposed, method bases on Captcha as a hard AI problem [2]. Explanation of CaRP is illustrated below.

- **Captcha and security**

A Captcha (an acronym for "Completely Automated Public Turing test to tell Computers and Humans Apart") is a type of challenge-response test, which determines that a user is a human or not. Mark D. Lillibridge et al. [36]. This form of Captcha was distorted image containing alphanumeric characters and a user will type these characters or digits. Example of Captcha is shown below (figure 2.10).

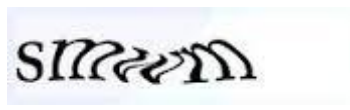


Figure 2.10. CAPTCHA of "smwm" Generated by Captcha[36]

Von Ahn, L., Blum, M. et al.[2] was proposed "Captcha using Hard AI problem for Security". "They introduced Captcha as automated test that humans can pass but computer program cannot pass it". Hence, this method is introduced to stop the bots attack on online websites and passwords. Like e-mail services (yahoo, Microsoft, Google, etc.), most of them were suffering from specific attacks. The bots create

thousands of e-mail accounts in every minute [2]. To avoid this problem the Captcha was best solution. Unluckily, there are some software (GSA breaker, Captcha Sniper, etc.), which can recognize captcha images up to 90%, and simple challenge of Captcha can be passed.

Recently Captcha as a Graphical Password method was proposed. CaRP method bases on Captcha using hard AI problem for security [2]. However, they use CaRP image instead of Captcha image, with more complex strategy. In CaRP method, a user can select any visual object by his/her own choice, but in Captcha image, a user has to follow the sequence of built-in characters or objects. CaRP is a click-based graphical password technique; it allows a user to select visual objects as a password. CaRP recognition based method "Text-Click" [1] is based on alphanumeric characters drawn randomly on image after rotating by an angle of 30° clockwise or anti clockwise, and overlapping of 3 pixels. Each character scales by 60 to 120%. It has two phases: the first one is registration, in which a user will register by selecting the characters, and the second one is authentication. The user will follow the same pattern to authenticate/ register him/ her. Upon the user's request, server generates CaRP image, and the object location changes randomly, and ground truth of each object will be saved to find the location of these objects when user wants to choose them as a password. An example of Click-Text is shown in Figure 2.11. The password will be the same in sequence of alphanumeric characters in authentication that user has selected during registration.



Figure 2.11. Example of Click-Text [1]

2.4 Security issues of recognition-based graphical passwords

As security point of view, each of these methods has some weaknesses as discussed above. Déjà vu [21] and Darren Davis [22] proposed method but problem was that the both take too much time for authentication [23]. Sobardo and Birget [26] scheme has the same disadvantage of long time [23]. Alia et al. [24] proposed the method having just five input symbols, which can easily be guessable. Recent proposed method is "Click-Text"[1] which has the limited number of characters which is easy to guess. If length of a password is 8, then total number of possibilities will be 2^{40} . If a user wants to fulfill the requirements to a complex password, it is tedious task for user to remember it. This method also has similar drawback like alphanumeric password.

2.5 Summary of security and attacks

To remember the password the most easy method/ technique is recognition-based authentication [1, 28]. On other hand, recall-based authentication is hard [16]. Mostly

recognition based authentication method has range 2^{13} to 2^{16} password combinations [16]. Another study reports that DAS (recall-based authentication) and Pass-Go password successfully access with guessing 2^{31} to 2^{41} entries [21, 23]. PassPoints (cued-recall based authentication) image contains hotspot points which help to break significant part of password [1,6] with dictionary of 2^{26} to 2^{35} entries as compared to full length of 2^{43} [1]. However, Click-Text is also easy to guess. It has 2^{40} possible entries, in survey, mostly, people use simple alphabetic password, and therefore, it is guessable. If strength of a password increases, it is a tedious task to remember it by a user.

2.6 Motivation

Graphical password is an alternative of text/alphanumeric based password, and the main motivation is that people can remember pictures and visual objects rather than text/words [8, 9]. Another side visual objects offer large set of usable passwords. Hence, it is indexed that a human would not be able to remember strong password. The contribution of my thesis is *the enhancement of security with new and secured alternative password scheme*. The proposed Scheme bases on Recognitions-based graphical authentication and will be explained in detail in chapter 4.

2.7 Problem definition

According to the above literature review, each of the available methods has some weakness and security issue. After surveying of the previously proposed method, I observed that all of the techniques are suffering from the various attacks, guessing attacks, and complex password issues (e.g. strength, complexity). In Text-Click method [1] and alphanumeric password scheme, when people set complex password, the percentage of the people remembering the password decreases. However, it performs strong resistance against Captcha breaker.

To solve the above problem, I choose recognition-based authentication due to some argues; Bin, Zhu et al. [1] took experimental result and observed that recognition-based authentication is easier for human memory. In contrast, the recall-based authentication is very hard to remember [16]. Zakaria, Zangooei et al. [28] also analyzed that the human performs better in recognition of visual objects as compared to other techniques. After analysis of these survey results, the RBGP has more potential variation as compared to Cued-Recall Based Graphical Password scheme. Most common problems of Recognition based graphical password are inconsistency and guess-ability effects [29]. To avoid these, new secure and efficient graphical password scheme is proposed which gives strong environment to authenticate the users.

Chapter 3

DESCRIPTION OF "CLICK-ON-CAPTCHA-OBJECTS" (SECURE RECOGNITION-BASED AUTHENTICATION ALGORITHM USING CAPTCHA AND VISUAL OBJECTS)

3.1 Definition of "Click-on-Captcha-Objects"

I can define my proposed method "Click-on-Captcha-Objects" in the following way:

"Secure Recognition-based Graphical Password using Captcha and visual objects to enhance the memorability and reduce the guess-ability"

I have three objectives of my proposed scheme.

Objective 1: Resist against the robots attack and achieve positive results for legitimate users.

Objective 2: Give the platform to gain strong password, which is easily memorable.

Objective 3: Measure the current security metrics against objectives 1 and 2.

To obtain stronger and reliable complex password, I employed Recognition-based authentication built on Captcha methodology, because of some evidence argue [1, 29]. Bin and Zhu et al [1]. proposed the method of Click-Text, built in Captcha with alphanumeric characters; the weakness of Click-Text was to set simple, usable characters as a password. Therefore, it is easily guessable, and with the limited number of alphanumeric characters. I analyzed that mostly people use simple string of characters like their name, family names or home address etc. Unlikely, people forget complex combination of characters. However, people prefer to set usual

alphabets and digits to build up their password. Hence, an attacker can guess it easily by user name, account information or family name etc. To handle this problem, new Captcha based authentication scheme "Click-on-Captcha-Objects" has been proposed herein. The purpose of this scheme is to boost memorability of complex password and cutback guess-ability attack. The proposed method is based on Captcha visual objects (visual objects are mixture of images or icons, and alphanumeric characters of any language). My proposed system generates an image based on Captcha, and the user will select password by clicking on corresponding object, which would be an element of password set. A user selects what he wants to set his password from the given Captcha based image. In addition, it provides an alternative way to a user to set a complex password easily. In other words, people select images by their own interest and they can remember these images for longer time compared to alphanumeric text. My proposed system is more complicated for robots and easy to remember the password for a human as compared to Click-Text and alphanumeric password schemes. Details of the comparison are described in chapter 5.

Click-on-Captcha-Objects description divides into two parts, first is relevant to description of the proposed method in informal way and algorithms. The second part focuses on data structure of proposed method.

3.2 Theoretical concept of "Click-on-Captcha-Objects" algorithm

Secure Recognition-Based Authentication scheme using Captcha and Visual Objects is based on Captcha technology. A method is proposed to reduce the guessing attacks and increases the memorability for a user to remember a strong password. Visual objects contain: images, special symbols, and alphanumeric characters. Visual objects are user-defined. System is basically using three different categories of

elements, the first is alphanumeric characters (any language), the second is some special symbols (after analysis of symbols suitable for my method), and the third is user-defined images (icon-sized) which would be embedded into the system by an administrator or user each time when a user wants to add a new image, the image will be resized into icon size and the name of the image will be set by auto unique numbering and stored in the database. A unique number will be obtained by combination of numbers and alphabets.

3.2.1 Captcha image of visual objects

How will Captcha image be generated and how will it work? I did analysis and observed that how it should be reliable and user can be comfortably authenticated. The proposed scheme is based on mouse-click instead of keyboard. Hence, a Captcha-generated image which contains visual objects, should be suitable for mouse clicking and easy for user to find and choose the object for the password. My Captcha image has the following parameters: each object will be rotated by -35° to 35° angle randomly, wrapped with 5 to 6 pixels (randomly) and each character/digits/ image will scale by 45% to 55% randomly, each visual object will be overlapped 3 to 5 pixels to each other. In addition, sine wave patterns are implemented to represent a character in the wave form, but these waves each time vary the wave-height from 8 pixels to 15 pixels. This is important to alter the position of objects randomly with random variation of the whole image structure to avoid the tracing pattern of the wave. How the parameter will be applied to draw an image is decided by the algorithms which are implemented to build image for selection by click to set a password.

The input string has N objects (mixed set of images (like icon), alphanumeric and special symbols). Unique characters or special symbols and repeating objects will be

skipped. After that, all the objects are randomly shuffled in the list. The list (list contains all the visual objects) will be divided into substrings/sub-lists (each sub-list can contain at most 15 objects). For example if length of substring is 15 and after 15th object, 16th will go to the next line and so on. However, my image height and width is varying because of number of objects is not fixed. Here, we assume that for one horizontal line suitable objects number is 15. For example, in English alphabet and 0 to 9 digits, the random string is generated excluding 0 (zero) and O. In addition, #, @, &, \$ special characters are added with special symbols and images. Let N be the number of symbols in the English language alphabet together with combination of above mentioned special characters, symbols, and images. The initial representation of objects is shown in Figure 3.1.

		Width														
		Wi	Wi+1	wi+2	wi+3	wi+4	wi+5	wi+6	wi+7	wi+8	wi+9	wi+10	wi+11	wi+12	wi+13	wi+14
Height	Hi	A	Q	Z	W	S	X	E	D	C	R	F	V	T	G	B
	Hi+1	Y	H	H	N	7	🍷	J	🐎	8	K	9	L	1	🍊	🐣
	Hi+2	☕	❤️	🐙	2	5	3	6	4	💖	🍋	🌕	🚰	📱	♣️	❤️
	Hi+3	🚫	✉️	★	🐦	🍉	🐯	🪄	🍦	U	M	★	☂️	🌸	#	&

Figure 3.1. Initial Stage of Proposed Scheme Captcha Based Image

While the string is being generated, all the objects are added one by one to a graphical bitmap from the sub-list (each object has its own bitmap graphical rectangular area). However, for drawing these objects in particular way, each object will get randomly scale by 45% to 55%. I did survey in which I observed that mostly people prefer the size of 45% to 55%. Hence, each time the size of an object will be

in range of 45% to 55% randomly. The font family is "Times New Roman", and font style is bold of each character. Each object will be represented in form of graphical rectangular area (H and W represent height and width respectively). So I consider $H = 25$ (range is 22 to 30) and $W = 25$ (range is 22 to 30) according to general review of people. After that, I perform rotation and shearing the objects. After that, each object scale by 45% to 55% randomly.

▪ **Rotation and wrapping of a rectangle**

Before performing wrap and rotate function on a rectangle, I generate a curve of objects, following the sine periodic function. It is important to construct complex structure for image segmentation (Captcha breaker, OCR). Hence, each object will change its position according to the following way.

When the object list will be called, random values will be in range 8 to 15 pixels (wave height) to build a horizontal line of objects. For example if the value is 10, then each objects row will be shifted 10 pixels down but after 1/3 part of the row passed, it will be again shifted up with 10 pixels to each object, once more, when 1/3 passed it is shifted down and so on. For each row, invariant amplitude of the wave will be applied for the current Captcha image, and it will become symmetrical, but for next Captcha image, it will vary randomly and the image shape will change.

During the object calling and building the image, each object is wrapped by m pixels to each coordinate of rectangle but randomly selected from 0 up to m , Here m is 5 or 6 pixels which is most suitable wrapping value, because if the value is increased to m , it creates confusion for user to recognize the characters. I did survey, 15 persons gave their comments, and, according to these comments, the significance range was set from 5 to 6 pixels. The new coordinate's values are to enhance the trouble for

machine to recognize the objects/ characters. At the same time when object will be wrapped, then to create more complexity for computer each object is rotated to add more confusion for a machine but easy for a human. Therefore, rotation has been done with clockwise or anticlockwise movement. To find suitable angle that is more reliable for user is 30° to 40° but mostly people prefer angles among 30° to 35° . There is no doubt for user to recognize the rotated objects. Therefore, this rotation is performed about the middle point of the rectangle, Coordinates of x and y are divided by 2 to obtain middle point that will be the origin of the rectangle. However, my object will not change the position because it rotates about center of x-axis and y-axis values, although it will change rotation of 30° to 35° angles.

Characters overlapping happened when each character rotate but overlapping boundary will not cover the main part of object. Thus, user can easily differentiate the objects. Overlapped area will not return any mouse coordinates if user clicks on this area by mistake.

- **Store coordinates of visual objects**

Consequently, each alphabet is stored into LIST against its coordinate's values, because the alphabets coordinates shall be required when I want to choose the object by click. To find the location of each object, I have to find the dynamic formula for finding the position of each object. Therefore, height and width of each object should be determined. According to current scenario, height and width are measured in the following way: W represents x-axis coordinate (horizontal coordinate from left to right of one object) value and H represents y-axis coordinate (vertical points from top to bottom of one object). The starting value will be zero for top-left corner of an image. Therefore, starting object top-left coordinate will be zero. For example, first object coordinate (upper-left upper-right , lower-left and right) are considered as

following: coordinate of upper-Left(0,0), the Upper-Right(Wi,0){i=1; addressed to position of object in the list} and lower coordinate of same character is; Lower-Left(0,Hi){i=1} and Lower-Right(Wi,Hi){i=1}. Since the all objects, coordinates are measured as this pattern. In Figure 3.2, character 'A' is illustrated with its dynamic coordinates. Since, for randomly selected coordinates are calculated by summation formula:

$$i) C_{tl} = (\sum_{m=1}^{i \% \text{substring}} W(m-1), (\sum_{m=1}^{\text{substring.count}} h(m-1)) \quad (3.1)$$

$$ii) C_{tr} = (\sum_{m=1}^{i \% \text{substring}} W(m), (\sum_{m=1}^{\text{substring.count}} h(m-1)) \quad (3.2)$$

$$iii) C_{ld} = (\sum_{m=1}^{i \% \text{substring}} W(m-1), (\sum_{m=1}^{\text{substring.count}} h(m)) \quad (3.3)$$

$$iv) C_{rd} = (\sum_{m=1}^{i \% \text{substring}} W(m), (\sum_{m=1}^{\text{substring.count}} h(m)) \quad (3.4)$$

Above-mentioned equations represent how to calculate coordinates of a character in an image. $i = 0,1,2,3 \dots N-1$, N is number of objects in image and substring (no. of characters in one row), $w(0)=h(0)=0$ and $w(m)=25$ and $h(m)=25$.}. Substring-Count represents height of each row. If number of objects exceeds substring, a new substring will be generated, and then sum of substrings will be counted. C_{tl} represents top left coordinates, C_{tr} represents top right coordinates, C_{ld} represents left down coordinates and C_{rd} represents right down coordinates.

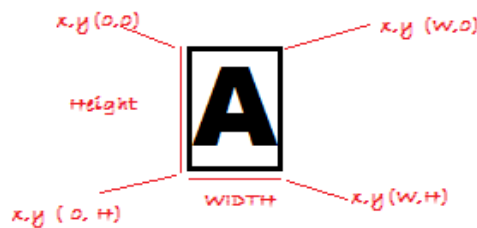


Figure 3.2. Rectangle Object Coordinates

▪ **Calculation of width and height of Captcha image**

When characters are drawn on image, width and height will be calculated as follows. It's clear that I can calculate each object's width by using above mentioned formula. If I know the width of one object then I can find the width of the whole image, first of all when the objects are called, each object has width w (consider $w = 25$). According to current scenario, I added 15 objects in each sub-list. Therefore, image width will be $25 * 15 = 375$ pixels. In same scenario, height can be calculated when whole list of objects will be called; I consider that each sub-list contains 15 objects, hence, the height will be summation of sub-list, for example if I have 45 objects and it's divided by 15 the sub-list of whole list is 3. Therefore, if height of one object considered as 25 then normal height (without sine wave) of image would be $25 * 3 = 75$. However, it is already mentioned above that I use sine wave pattern, shown in Figure 3.3. Since, total height after sine wave pattern will be " $75 + \text{wave height}/3$ ". Wave value can be calculated by using this formula. Wave-height (distance between crest to trough) value is randomly varying in the range of 8 to 15 for each object, and considered that W_h (wave-height) for one object is 9, and $W_v(\text{wavelength})/3$ is the number of objects between mid of line to crest/trough. Therefore, total addition variance in height will be $W_v/3 * (W_h)$.

If $W_v = 9$ objects;

$W_h = 8$; // for one object

Height of image = $75 + 9/3 * 8 = 99$ pixels.

In current scenario, total height of image will be 99 pixels. Height will always change when number of input objects will vary.

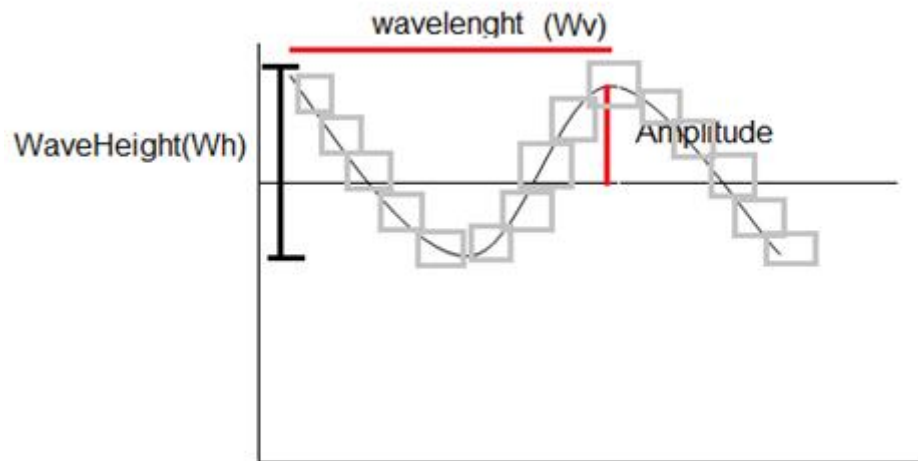


Figure 3.3. Representation of Objects (Wave Form) on Image

▪ **Interface of "Click-on-Captcha-Objects" image**

When upper labeled process finish, the image will be drawn on screen where the location is allocated. Actually when above process completed I got bitmap image and this bitmap image assigned to front-end screen image. The starting index of that image is (0, 0) coordinates to (n, n). Microsoft Visual Studio sets image coordinates (0, 0) to (n, n) automatically when Mouse click event on image is activated. Hence, bitmap image mapped on front-end image with own coordinates from (0, 0) to (n, n). Bitmap image coordinates values will not change because it already start from 0, 0 coordinates.

▪ **How can object be selected by click on image**

It can be determined by using the following way: each object has own rectangular area, and user will select them by click, if a user wants to choose character 'A', he will click on it. To clarify that click lies on which object boundary, here are basic mathematical equations to resolve this problem. To avoid the conflict of object selection, user should click on the main part of character, which is not overlapped with other. When user will click on one character, system will get x and y coordinates of clicked point, and character area is not simple rectangle but its

irregular Quadrilateral, because of some transformation of coordinates (wrapped/shearing and rotation). Each time when mouse click it returns x and y coordinates of that point. Therefore, when user clicks, system gets point value and matches it with object coordinates. Clicked x and y coordinates are compared with each coordinate of each object when it belongs to any character it will be selected. For example $x = 20$ and $y = 30$, first they match with top-left coordinate of first object and so on, if any object coordinates matches, it will be selected and process will stop. Now, how mouse click point is compared with object coordinates. First, corners of the rectangular coordinates are checked after it line equation is used to check the point to make it clear because of irregular shape of rectangle. The process is following to recognize each rectangular area of object, ms_X represents mouse clicked x value and ms_Y represents mouse clicked Y value, and rectangle coordinates are top_left_X , top_left_Y , top_right_X , top_right_Y , btm_left_X , btm_left_Y and btm_right_X , btm_right_Y . There are four steps to recognize the rectangle boundary, as explained below.

- i. First, top-left coordinates of character will be matched with x and y coordinates of mouse clicked point of x and y. If the point value of x is greater than existing character top-left coordinate of x value and mouse Y point value is also greater than coordinate of top-left Y value. $If((ms_X > top_left_x \ \&\& \ ms_Y > top_left_Y) == true)$. If it returns true, it will go to next step, otherwise, this area will be rejected, and next coordinates of rectangle will be fetched from the list.
- ii. If above process is true, then top-right coordinates of character will be matched with mouse-clicked point. If the mouse clicked point value of x is less than object rectangle top-right coordinate of X, and Y value of mouse clicked point is greater than top-right coordinate of Y ($if((ms_X < top_right_x \ \&\& \ ms_Y > top_right_Y) ==$

true)). If it is true, it will go to next step; otherwise, next object coordinates will be called.

iii. In the next, the bottom-left coordinate will match if above process will return true. If the mouse-clicked point x value is greater than x value of bottom-left coordinates and y value of mouse clicked point is less than y value of bottom-left coordinates, (if ((ms_X>btm-left_x && ms_Y<btm-left_Y)==true)), if this condition is true, the next step will be performed, in contrast, next rectangle area coordinates will be called.

iv. In last, bottom-right coordinates will match with x and y coordinate of mouse clicked point. If the x value of mouse-clicked point is less than bottom-right x coordinate value and Y value of mouse point is also less than coordinate value of bottom-right y, (if ((ms_X<btm-right_x && ms_Y<btm-right_Y) == true)), if this condition is true then this area object will be selected.

When system knows that a point lies between these character's coordinates, to prove this, there is method to use rectangular area for one character to check the point lies inside or outside. Rectangle is divided into two triangles; upper triangle and lower triangle. First, clicked point matches in upper triangle if it exists in the first part, no need to check next part of rectangle, in contrast, it will check the second part. Triangle method is more efficient as compared to rectangle (four lines comparison takes more time as compared to 3 lines of comparison). However, preference method is implemented and one rectangle partitioned into two parts, is shown in Figure 3.4 below. There are three points as A, B and C of upper triangle. Point lies inside the line or outside the line, there is mathematical equation to manipulate it. The three lines are analyzed A(x, y) to B(x,y), B(x,y) to C(x,y) and C(x,y) to A(x,y), it is shown in Figure 3.4. How can system know point is inside the line or outside the

line? A mathematical equation (3.9) returns that a point lies inside or outside the triangle. It will be decided after comparison of three lines of triangle from line A to B, line B to C and C to A. The returning value is greater than 0 for upper triangle. Because A to B the point will be left side if its returned value is positive, and the same for B to C and C to A then it means a point lies inside the triangle. In second lower triangle, the process will be inversed. If point lies inside, equation (3.9) will return negative value. Moreover, this selection pattern will address to selection of visual objects from Captcha image.

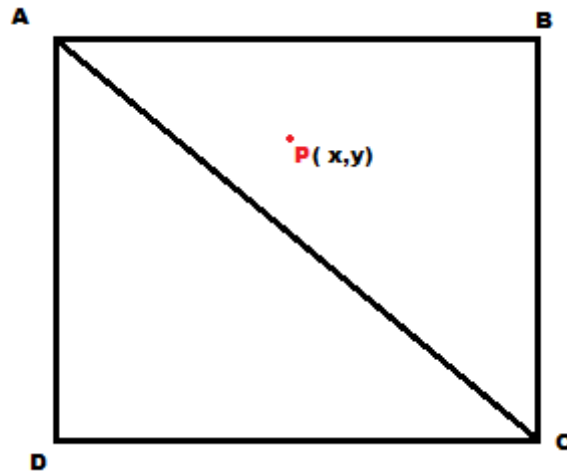


Figure 3.4. Two Triangles of One Rectangle to Recognize the Point

How to derive the equation, there is the following way to prove the equation.

Suppose that I am given two distinct points in the plane, (x_1, y_1) and (x_2, y_2) . There is a unique line

$$C_1X + C_2Y + C_3 = 0 \tag{3.5}$$

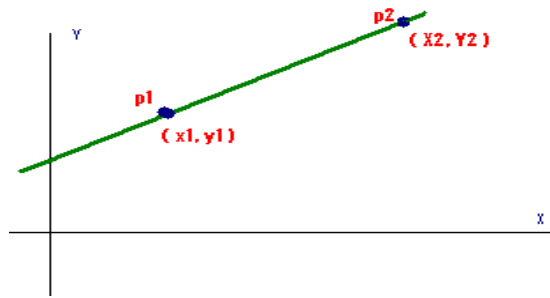


Figure 3.5. Two Points Lie on Line L, "p1 and p2".

Line L that passes through these two points p1 and p2 (Figure 3.5), c_1, c_2, c_3 are not zero generally, and these coefficients are unique only up to a multiplicative constant because (x_1, y_1) and (x_2, y_2) lie on the line, substituting them in (v).

$$C_1X_1+C_2Y_1+C_3=0 \quad (3.6)$$

$$C_1X_2+C_2Y_2+C_3=0 \quad (3.7)$$

So the three equations are grouped together and rewritten as

$$C_1X+C_2Y+C_3=0, C_1X_1+C_2Y_1+C_3=0, C_1X_2+C_2Y_2+C_3=0 \quad (3.8)$$

"Which is homogeneous linear system of three equations for c_1, c_2 , and c_3 . Because c_1, c_2 and c_3 all are not zero, this system has a nontrivial solution so that the determinant of the system must be zero, that is".[30]

$$\begin{vmatrix} x & y & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{vmatrix} = 0$$

Now, assume that $x = a_x$ and $x_1 = b_x$ and $x_2 = c_x$ and remaining b and c are similar, since by using above determinant of system to check the points lie left or right. Since c is point, which lie left/ right side of line and \vec{ab} is two point lie on the line so solution of above problem is;

$$\begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix} = 0$$

Using 3rd column to find the determinant,

$$(b_x c_y - b_y c_x) - (a_x c_y - a_y c_x) + (a_x b_y - a_y b_x) = 0$$

$$b_x c_y - b_y c_x - a_x c_y + a_y c_x + a_x b_y - a_y b_x = 0$$

$$b_x c_y - b_y c_x - a_x c_y + a_y c_x + a_x b_y - a_y b_x + c_y c_x - c_y c_x = 0 \quad // \quad c_y c_x \text{ add and subtract}$$

$$b_y (a_x - c_x) - c_y (a_x - c_x) - a_x (b_x - c_x) + C_y (b_x - c_x) = 0$$

Result is:

$$(a_x - c_x) * (b_y - c_y) - (a_y - c_y) * (b_x - c_x) > 0 \quad (3.9)$$

This equation returns the +ve, -ve or zero value. If result is positive, it means point is inside the line (left). If it is negative, it means point is outside the line (right), if zero it means point is lie on the line, the direction will be \vec{ab} [30].

As a reminder, system stores each object and its coordinates in list. Therefore, system takes one by one object coordinates from list, and matches its mouse clicked point. If the mouse clicked point belongs to near the boundary or overlapped area of object it will simply refuse because system can't decide that is it lie on current object or is it part of overlapped object? so, system will not return any character. If user will click three times outside the boundary of object or overlapped area, system will show message that click inside the object and new image will be generated with same objects but position and rotation will be changed, implementation details of this method are described in chapter 4.

3.2.2 Password complexity of "Click-on-Captcha-Objects" algorithm

On the other hand, proposed system has complex password strenght as compared to Click-Text and alphanumeric algorithm, they have problem of password complexity. Proposed method generates complex password easier compared to alphanumeric scheme. The review of password history illuminates that password length should be

at least 8 characters, if characters are not belonging to usual words, it becomes very hard to remember it. To avoid this problem, the proposed method is based on visual objects, which can be easily remembered by a human. Password is a mixture of alphanumeric character and special symbols and icon size images. Hence, my password is like this, suppose user clicks on alphabets P, A, K and after that user selects 3 symbols like ♣,♥,⚡ and after that three images selected like 🌟❤️☕.

Hence, my password length will be **PAK♣♥⚡[name of first image][name of second image][name of third image]**. Name of the image is generated automatically and is unique. However, the password will be so long as compared to Click-Text and alphanumeric and easily rememberable. User will have options to choose any object from the whole captcha image but at least 3 images in each password with some alphanumeric or special symbols and count of password should at least fulfill the rules of password security.

3.2.3 Stages of "Click-on-Captcha-Objects" algorithm

My proposed method is divided into three stages, the first stage is how the image object will be added and image generated, the second is related to registration, and the last one is authentication.

3.2.3.1 Insertion of user-defined objects in Captcha image

In this stage, user will input/upload set of images (transparent background) or alphabets to draw a Captcha image of these objects. Uploaded images can become a part of password. For example user wants to add Chinese alphabets and some images with some special symbols, but the length of total elements should not be less than 50-70 objects and not greater than 150 objects, otherwise the image becomes crowded, hence, user feels difficulty to find the objects on image.

3.2.3.2 Registration stage of "Click-on-Captcha-Objects" algorithm

In registration step, User will enter Username/e-mail id and will select password from image by click. User will click on main part of object, which clearly available for user and some part of object may be overlapped to another objects but main part will be available for clicking. User will repeat same password twice to make sure that which object he clicked and it is practice to remember the password. User can select some alphanumeric characters, and icon-images. When user will satisfy the password length, for example, selected objects of password are **PAK♣♥♠★❤☕**, then password p is hashed with salt value $\text{Hash}(p, \text{salt})$ and stored in database against new user record if corresponding name already not exist in database.

3.2.3.3 Authentication stage of "Click-on-Captcha-Objects" algorithm

In this stage, how user will be authenticated? The diagram is shown below. Basically how the system will work against any request to authenticate, the Flowchart 3.7 shows that first of all user will send request for authentication, then server will respond to the Captcha image. The user will enter user name and select objects on image by click. When user will submit request for login, server will hash his/her password and will compare it with already stored hashed passwords, then decision will be taken that user are legitimate or imposter?

Three times attempt will be allowed, after that system will not respond to the user. If user clicks on wrong position and if number of clicks will exceed from 3, system will automatically discard the current user. Following diagram, figure 3.6, represents the flow of authentication.

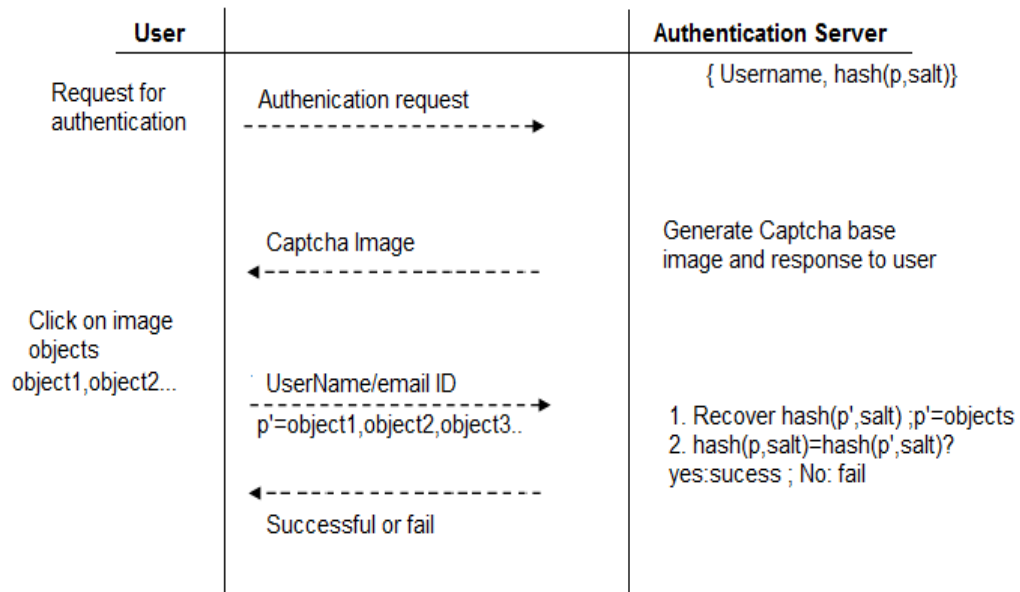


Figure 3.6. Request Response of User Authentication System

3.3 Structural representation of the proposed algorithm

Proposed "Click-on-Captcha-Objects" algorithm is divided into three main phases, add users defined objects, registration and authentication. Each part has sub-categories. Flowchart is at first place of whole algorithm and after that explanation represents the sub categories of each phase.

3.3.1 Flowchart of the "Click-on-Captcha-Objects" algorithm

Flowchart of "Click-on-Captcha-Objects" algorithm represents how user will act and system will react against it. System will follow the user behavioral information and set the direction of domain. For example if user wants to register, user can easily understand the process and can operate system by using given instructions (set control to registration), flowchart is shown below (figure 3.7). Each stage of a flowchart assigned a number; the details are represented in explanation sub categories. Start represents the start of program and stop shows the end of program.

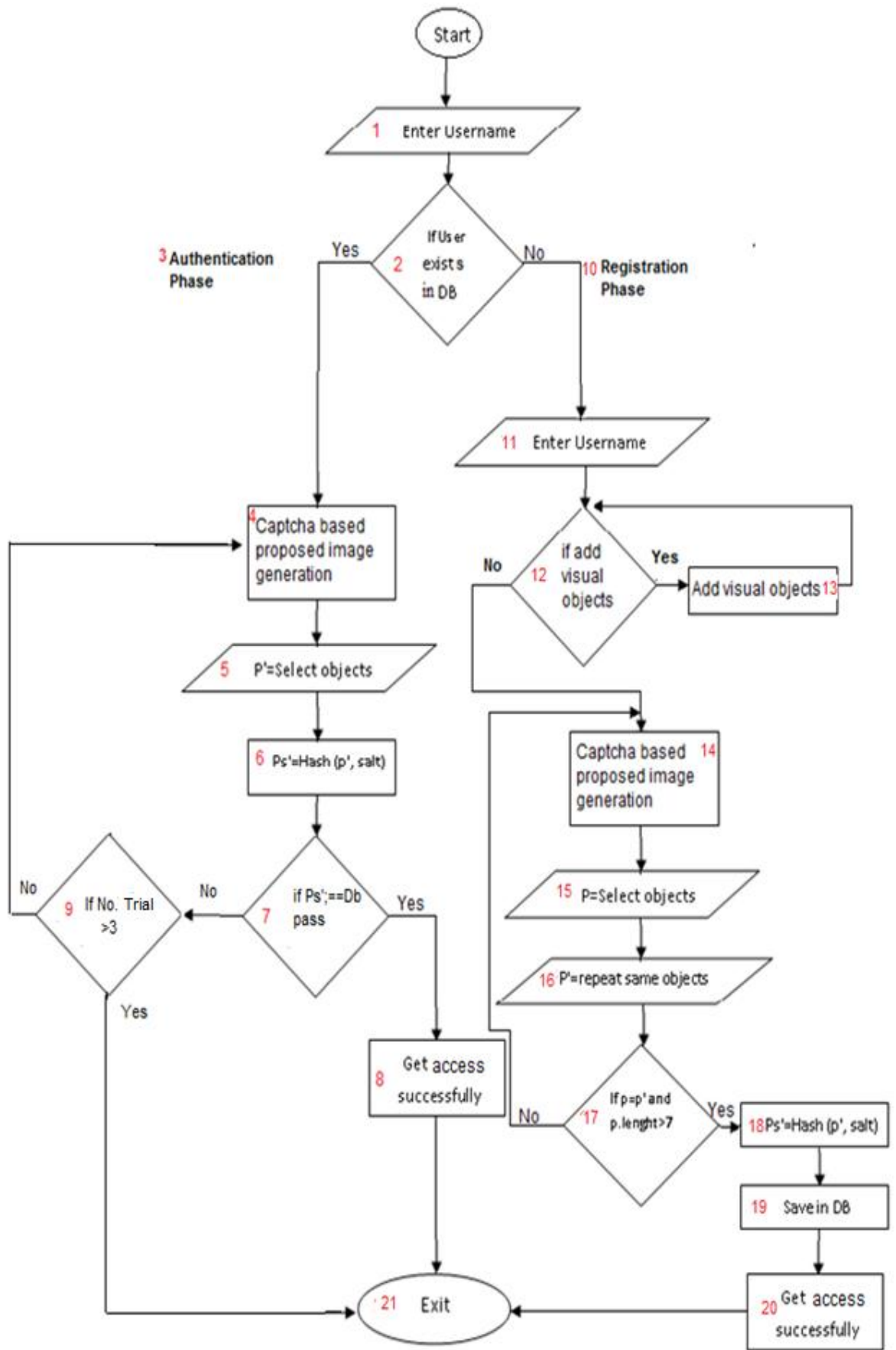


Figure 3.7. Flowchart of "Click-on-Captcha-Object" Algorithm

3.3.2 Description of the flowchart of "Click-on-Captcha-Objects" algorithm

To demonstrate the sub parts of above mentioned Flowchart (Figure 3.7), each of sub-steps is assigned a number. Steps of main Flowcharts are briefly discussed with sub-Flowcharts to more elucidate the concept of algorithm.

Step 1: "Enter user name ": It represents user name or email id.

Step 2: It verifies that if user already registered, if yes then user has to go for login process otherwise he/ she will select registration option.

Step 3: It represents that process of authentication will be called.

Step 4: Captcha-based image actually describes the proposed method image, Captcha based visual objects image has following flowchart (figure 3.8)

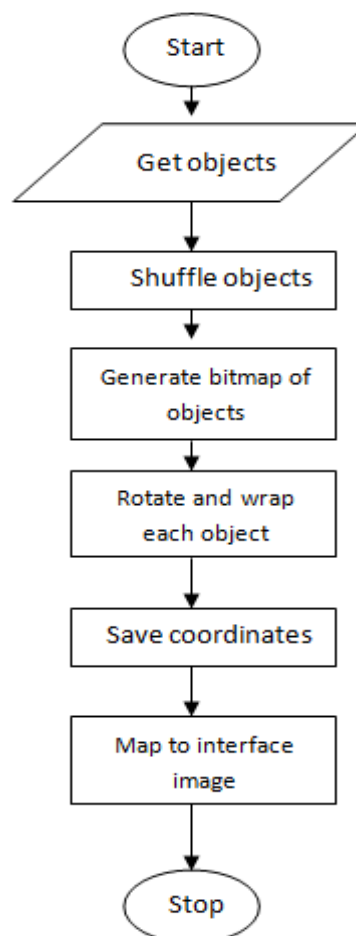


Figure 3.8. Flowchart of Captcha Based Image Generation

After it, program will get objects from database or strings and then shuffle them randomly. After that each object bitmap area will be selected and then rotated with 30° to 35° angles clockwise or anti-clockwise, and will wrap it, then the coordinates are stored into list and image is assigned to interface of the program. Code of generated image is shown in Appendix A.

Step 5: In this stage, user will select password by mouse-click on image. If the clicked point lies inside the corresponding object, that object will be selected, if point does not lie inside any character, it will be rejected, implementation code is shown in Appendix B and Flowchart 3.9 illustrates more details.

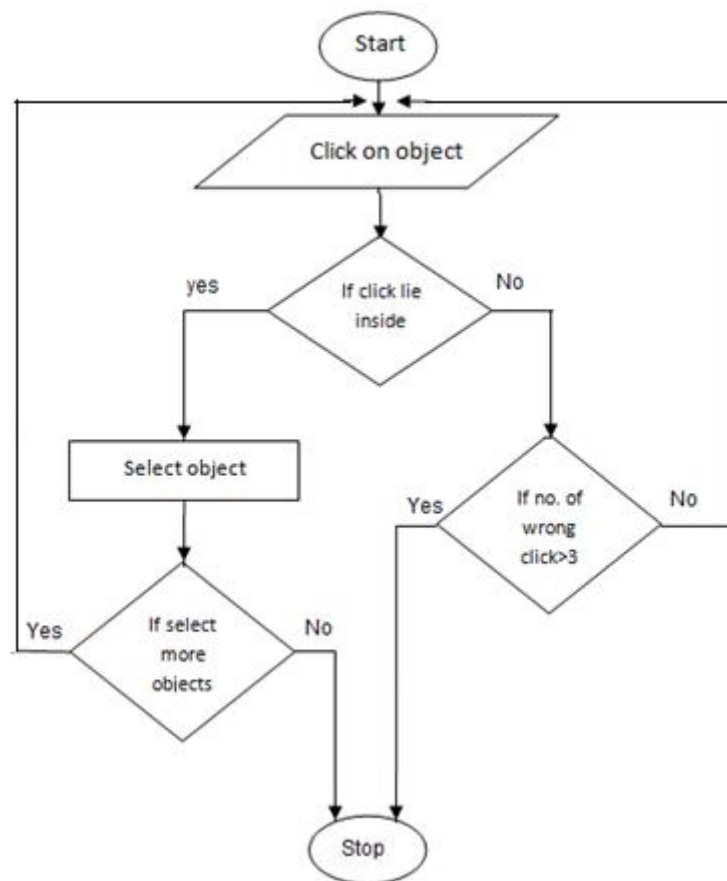


Figure 3.9. Flowchart of How to Select Object

Given diagram represents that when clicking is performed on welling object, the coordinates of mouse click are checked out that the point lies inside of boundary if point x , y coordinates values lie inside (according to equation (3.9)), the corresponding object will be selected. If the point is overlapped or outside of boundary, it will be rejected without any selection of object (how I select these objects? Description is available in 3.2.1). If limit of wrong clicking is more than three, program will be terminated and user has to call new session for authentication.

Step 6: In this step, password will be hashed with salt value which user will select. Hash is built-in function of C#.net. Therefore, Hash (salt, password) value perceived with current password, implementation code represented in Appendix C.

Step 7: At this stage, secure password will be compared with saved passwords in database, if current hashed password matches with store-hashed password against same User-ID, if both are equal, user will be allowed to access resources; otherwise, user will have to try again.

Step 8: If above step is true, user will get access to his/her profile and its secret documents.

Step 9: In this step it will be checked that after three wrong tries, user will be suspended, as like security requirements.

Step 10: (Registration phase): In this step, registration phase will be called for new user.

Step 11: (Enter name): It represents the first element of registration interface, when user will enter name, it will be verified that name is already exist or not, if it already exist new record will not be entered. Flowchart of username verification shown below (see Figure 3.10).

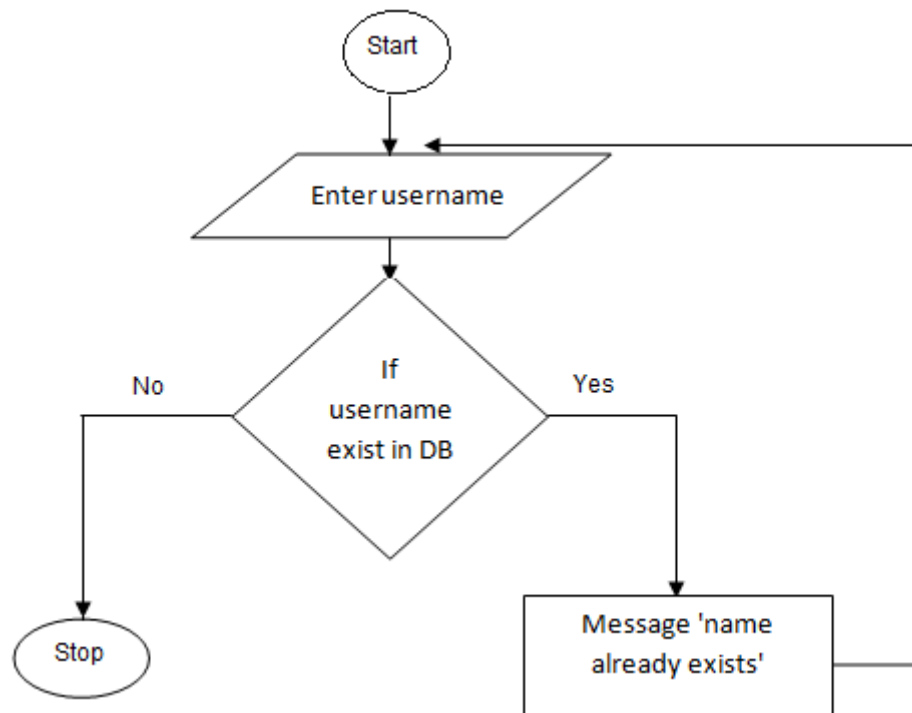


Figure 3.10. Flowchart to Check Username Exists or Not Exists

Step 12: In this step, before going to registration, if user wants to add images/ objects/characters then he/ she can add.

Step 13: Add image illustrates that user can add images and system will resize them, flowchart shown below, see Figure 3.11 In addition, implementation code is shown in Appendix D.

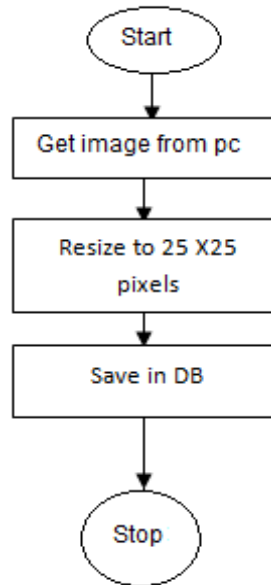


Figure 3.11. Flowchart to Get Image and Save in Database

Step 14, 15, 16, 17, 18 and 20: all are same like authentication, just it will be extra check if password length is less than 8 characters, it will show message to enhance the length. In addition, user has to repeat the same selection of character to verify the password, implementation code of this function is shown in Appendix E.

Step 19: In this step, all the information will be stored in database against new user, code and queries are shown in Appendix E.

Chapter 4

DESIGN, IMPLEMENTATION AND TESTING OF "CLICK-ON-CAPTCHA-OBJECTS" ALGORITHM

Click-on-Captcha-Visual-objects proposed system based on recognition based authentication of graphical password. In this method, above process is first analyzed, and then suitable interface is designed which fulfill the users requirements. The implementation has been done in `c#.net`.

System has been partitioned into several steps. System requirements, system design and Implementation (Coding) and last is Testing of proposed method.

System requirement means what resources will be required to implement the proposed system. Software, which necessary for implementation of proposed method is: OS window 7 or 8 (X86 or 64 bit), MS visual Studio 2012/2013, Language `C#.net` and MS-SERVER 2008 (Database). On other hand, to attempt test and practically evaluation, system needs users that evaluate the system to clarify that system has potential of strong security.

4.1 Design of "Click-on-Captcha-Objects" algorithm

In Designing phase, project has been divided into sub-categories to make easy to handle. The whole design is divided into two main parts, registration and authentication. Flowchart is representing my system model (see in Figure 3.7). Next session describes the design and development of both parts (registration and authentication).

4.1.1 Interface of registration

In registration phase, user can easily understand the procedure of proposed method. Figure is shown in Appendix K, Figure K(1). To implement this, Visual studio 2012, and c#.net are applied for back-end programming. First user-name text-box represents username or e-mail Id for registration, second is; add images / objects, here user can add images and alphabets/symbols. In third step, enter password: where user will select password by mouse click on image (generated by following above mentioned process discussed in 3.2.1) and repeat password shows that user will repeat again the same pattern. The password text box represents the number of objects set in the password. If user inputs wrong password he/ she can reset by using Reset button or can cancel it. After selection of objects, register button is available to register. If both of selected passwords are equal and their lengths are not less than eight objects (password should be contain at least 3 images), system interface will return message that user is successfully registered.

4.1.2 Interface of authentication

In login phase, system provides user to enter user name or ID to authenticate. After that, system will show the image, which contains all objects which user have set during the registration step (implementation is done in next 4.2.2). This image will be used for authentication to choose objects by clicking. Login, Reset, Cancel and registration buttons will help the user to understand the system and user can perform any action as per need. Interface of authentication is shown in Appendix K, figure K(2).

4.2 Implementation of "Click-on-Captcha-Objects"

In this step, the above design is implemented for the proposed algorithm. The tools which are required to code this system are as, Visual studio 2012, C#.net and SQLSERVER 2008. Main functions that are parts of both registration and authentication phases are described here. Each function name represents its functionality.

Table 4.1. C#.Net Functions for Implementation of Proposed Method.

Add_image();	In this function, new image will be added into the main image.
Store_image();	In this function, output is stored into database, implementation code is illustrated in Appendix D.
Get_objects();	Get_objects() function fetches objects/characters list from database or string and return this list to system.
Random_swap_object(List[] objects);	This function takes returned list of objects from control and performs objects shuffling.
draw_Captcha_image(List[] objects, scale, height, width);	It draws whole image with list concatenation of all objects and input parameters. Implementation code is shown in Appendix A.
List Store_coordinate_object(points[],string object);	This function is used to store the coordinates of each object.
get_clicked_object_location(Mouse_click_points[x,y]);	This function belongs to selection of objects, select an object and if it returns "yes" to system if click inside the rectangle to corresponding object.

Register_user();	Register function is used to save user information in database; it is simple insert the data into system. Code is illustrated into Appendix E.
compared_password(password, repeat_password);	It compares the password with the repeat_password.
hash(password,salt)	Hash function simply takes password and salt value (any key value) and encrypts it using hash algorithm, related code is implemented in Appendix C.
store_DB(name/id, hashpassword);	Store_DB function is used to store user information from interface to database, against user name/id and hashpassword,. Code of this function is represented in Appendix E.
fetch_pass(username);	This function uses to fetch password. It just returns the password corresponding to username if it exists.
compare_pass(pass', pass);	It takes two hashed passwords, current "pass" and database password "pass' " and compares them. If both are equal, it returns true value.

4.2.1 Implementation of registration phase

To implement the above methods, first database designed. The database contains related information about interface. There are two tables. One is related to add user-defined images or objects, and the second is for registration of user names and

passwords, date and active_id. Diagram of database is shown in Appendix K, figure K(3).

Related functions of registration defined in Table 4.1. In implementation of registration phase, user can add images by user choice, to add images, user has to click on “Add images” button, add_image() function will be called, it browses the documents folder and user can choose images by own choice. The name of the image will be generated with random unique name by newGuide() function. After this, store_image() function will be called to store in database. Same process will happen for characters and digits. "Generate image" button is used to draw image of input objects. However to draw image, proposed algorithm will be implemented. When clicks are placed on generate button, first, Get_object() function will be called and all objects (images and characters) will be loaded into List. After that, it is randomly swapped by Rand_swap_function(List[]), it takes list of all objects and randomly select any number from 0 to N objects and make new list and returns it back to control for next procedure. This function also counts the height and width of proposed method image (procedure is written in chapter 3) After that, draw_Captcha_image(list, scale, height, width) will be called, and each object will be called and convert it into a bitmap image, now rotation, sine waves and shear/wrapping of pixels equation will be executed inside of this function. Therefore, system will save the coordinates of each irregular rectangular areas using store_coordinate_object (points, object) against particular object. When it finishes, Clickable image will be generated and map it into front-end image location. Same image will be mapped at consecutive places, one for set password and second for confirm/repeat password. Now, the system is ready to enter the password. When user clicks on image objects, get_clicked_object_location (points) function will be called

and x and y coordinates of mouse clicked will be compared with all the objects rectangles until system found the object, otherwise no object will be selected. User will select objects then he/ she will repeats/confirm password on the next image. When user will click on register button, register_user () function will be called, password and repeat password will be compared by compared_password(password, repeat password) function and then will proceed if both are equal. The password will be hashed with some salt value using hash (password, salt) function. After that, user name and password will be stored in database using store() function. Code of main functions described in Appendices A-E. Interface of registration shown in Appendix L, figure L(1).

4.2.2 Implementation of authentication phase

In login phase, when user will request to system, system will call objects from database using function (mention in Table 1.4). System will generate same image with same parameters like registration phase, using above functions, but now just one image will map into interface location of image. On other hand, both registration and authentication phases images does not based on similar sequence of objects, both contain same objects but with randomly variant location on image. When image will be clicked and password will be set (description is in chapter 3, and code for selection of object is described in Appendix B), process will vary from registration phase. System will take user-name and password. Using hash function, password will be hashed by hash (pass, salt); it will be compared with stored password (password against input name). "fetch_pass(username);" function will be used to fetch the password of particular user. It returns hashed function against username, if name does not exist in database it will show message that user name does not exist. If name is wrong or password does not match with stored password, it will show the

message that password or user name is incorrect. Here is also an option for user i.e. if user wants to reset password, or new user can register to click on register button on authentication phase and system will redirect into register page. Implementation of authentication is shown in Appendix L, figure L(2).

4.3 Testing of "Click-on-Captcha-Objects" algorithm

After completing the implementation, I did different types of test to evaluate the proposed method. The purpose of the testing was to observe that it fulfills the requirements of proposed idea or not. Different types of testes were estimated and it helped me to make sure that proposed system is working accurate. First, input data functions and alternative function of system are checked that functions are working properly, otherwise it will be updated. In addition, check database connection and password hashed function to analyze that password is properly hashed. Another test, improvement in security test is performed in which password complexity is measured and graph (shown in chapter 5) shows that proposed "Click-on-Captcha-Objects" algorithm is stronger and easy to remember. Main idea of my thesis is to provide environment in which user can easily remember the password even with complex combination. On other side, most suitable combination and pattern of generation of image is finalized which is more comfortable for user and hard for robots to break it (analyses is discussed, chapter 5).

In load testing, Generation time of image was noted, method is implemented on SAMSUNG (4GB RAM, Core i5) laptop and conducted result was approximately 40 millisecond per "*Click-on-Captcha-Objects*" image. However nowadays, very high speeds of computers systems are available, so it can be applicable on single or clusters based network system. In addition, Captcha breaker and image segmentation

results represent that system is not easily breakable by Captcha attacks. Comparison results and graphs with traditional system explained in chapter 5.

Chapter 5

IMPLEMENTATION OF KNOWN METHODS AND COMPARISON WITH PROPOSED CLICK-ON- CAPTCHA-OBJECTS METHOD

In this chapter, first, previous method "Click-Text and alphanumeric + captcha " has implemented because I could not find source code. However, I did implementation on the same PC. Section 5.2, represents the security analysis of Captcha generated images. In remaining sections, experimental results conducted to compare the performance with proposed method "Click-on-Captcha-Objects" and already listed out methods (alphabets and Click-Text). Conducted results are divided into the following categories. Security based, performance based and convenience based. To achieve the performance and security comparison, I did survey of 40 people (five PhD, 12 master, five were jobholders and remaining were undergraduate students, average age was 24.5. However, four of them were females and remaining were males).

5.1 Implementation of "Click-Text" and "Captcha + Text" schemes

There are two main schemes, which are compared with proposed method. One is traditional alphanumeric password and other is Click-Text. Implementation of Click-Text technique has been represented in Appendix H. Traditional password method is implemented with alphanumeric and Captcha. The implementation and interface are available in Appendix J. Implementation has been done in c#.net by following Click-Text parameters.

5.2 Example of three password schemes

5.2.1 Example of "Click-on-Captcha-Objects" proposed scheme

In Click-on-Captcha-Objects proposed method, user will first register. User will select some characters with at least 3 user-defined images (user can add own user defined images). There are two panels (interface of proposed scheme is shown in Appendix L), one panel is to select password and the second is for repeat the same password to confirm that user have selected correct objects combination. For example, user selects AXY♥★❤☕ in the both panels. After selection of objects user will click on “register” button. If both passwords are equal and not less than required limit, then user will register successfully. After that, user can authenticate his/her account anytime. User will enter his/her name and after that, he/ she will repeat the same password by clicking on corresponding objects.

5.2.2 Example of "Click-Text" scheme (CaRP)

In Click-Text scheme, user will select password by click on alphabets or digits (interface of Click-Text scheme shown in Appendix H). User will select at-least eight alphanumeric characters. During the registration user has to select the same password two times in two separate panels. If both separate passwords are equal then user will be registered successfully. For example, user selects password NYTON#123. At authentication, user will select the same characters that he/ she have selected during registration, the hashed value will be the same, hence user will be authenticated successfully.

5.2.3 Example of "Captcha+ Text" scheme

In this scheme, user will register like alphabetic password but he/ she has to pass extra challenge of Captcha code. Each time user will face a random sequence of alphanumeric characters in range 6 to 12 (implementation detail shown in Appendix

J). When user registers, he/ she will enter password by using the keyboard, and character string length should not be less than 8 with at least two digits. User has to repeat the same password, after that he/ she will pass the Captcha challenge. For example, entered password is ABcde#123. At the authentication stage, user has to enter the same password. If the entered password will match with ABcde#123 then user will successfully login, otherwise he/ she has to try again and three tries will be possible; after three tries each scheme will not allow enter password for the current user.

5.3 Security analysis of captcha generated images

Captcha based challenge is a method of verification that user is a human. However, unfortunately simple Captcha is breakable. If a system can break Captcha, than the Captcha challenge is meaningless. However, proposed system targets to generate Captcha based image that contain visual objects, user has to choose some of them for password rather than enter all. Although, four OCRs (Optical Character Recognition) software applied on generated image which employ different set of parameters. First, simple Captcha image is generated and a test is performed to verify how much objects are recognizable. After that, complexity is increased with rotation by angle and waves to make complex for recognition of alphabets. List of software is as following:

- i. GSA Captcha breaker V2.97[31]
- ii. Captcha Sniper X3[32]
- iii. Free online OCR [33]
- iv. Online free OCR [34]

These four software, are applied to take experiments results. GSA is most popular and its performance is very high, "GSA Captcha breaker" Trail version 2.97 is available on Internet [31]. Second Captcha Sniper is also popular software of Captcha breaker. The performance of Captcha Sniper against simple Captcha is available at website [32] and trail version can be downloaded from Internet for one month [32]. Remaining two are online websites, which generate OCR of input images [33, 34]. Experimental results of above software are shown in Table 5.1. So many images created for experiments, some important experiments are discussed below.

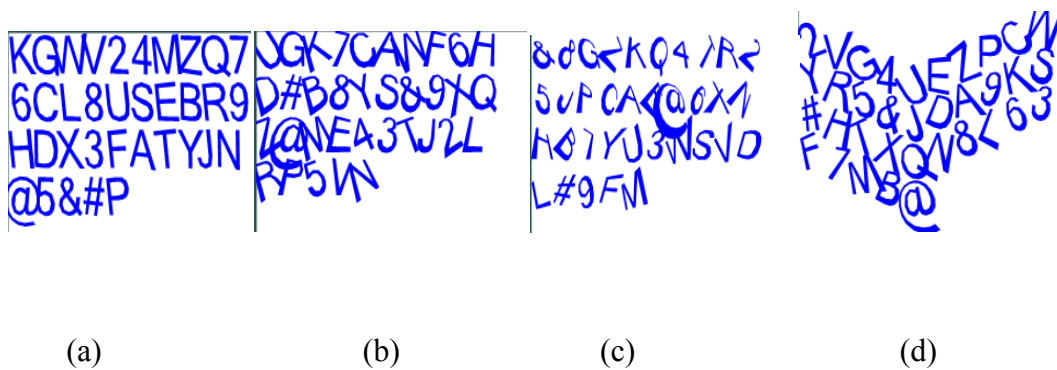


Figure 5.1. Captcha Generated Images with different parameters

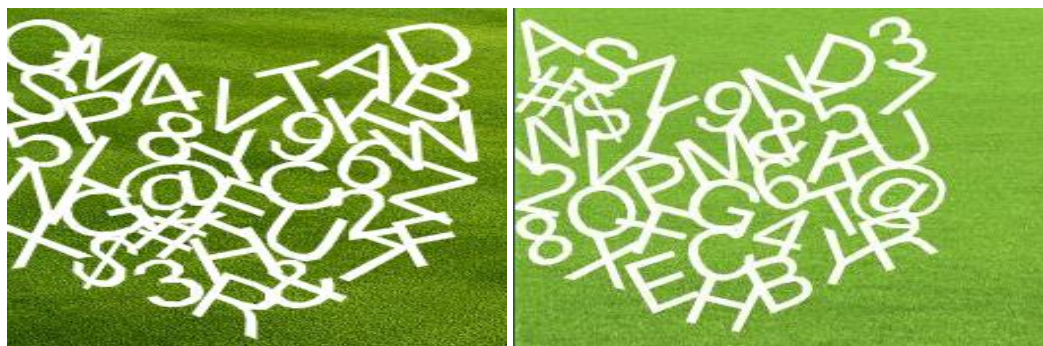
Figure 5.1(a) represents randomly shuffled alphabets and digits. Angle and wrapped pixels are set to zero. In figure 5.1 (b), angle is randomly rotated by 30° to -30° , wave height is consider zero and wrapped pixels are set 5 to 6 pixels randomly. In Figure 5.1 (c), the same above parameters but wrapped value is 10 pixels. In last Figure 5.1 (d), represents images with angle 30 to 32, and wave height is nine pixels. Above mentioned software are employed on these images and OCR result is shown in Table 5.1, and OCR results are shown in Appendix F.

Table 5.1. Captcha Breakers Software Results of Initial Generated Captcha Images of Figure 5.1 (a)-(d)

Sr.No	Captcha Breaker software	figure (a)	figure (b)	figure (c)	figure (d)
1	GSA Captcha breaker	97.10%	17.14%	40%	28%
2	Captcha sniper	71.42%	11.42%	34.28%	17%
3	www.free-ocr.com	70.40%	17.15%	20%	5.70%
4	www.i2ocr.com	94.28%	20%	37.14%	0%

This Table 5.1 represents the images recognition results by above software. Results of Table 5.1 indicate Figure 5.1[a, b, c, d] and four Captcha breaker software (Experimental screenshot is described in Appendix F). According to above Table 5.1, last Figure 5.1 (d) resultant percentage is lower than other three images. However, this recognition rate is 20%, which is very high.

To reduce the recognition percentage, new Captcha images are generated with random scale value from 40 to 60. In addition, sine wave function is performed to make complex image. There are following images that are generated by applying mentioned parameters (scale, angle, and wrapped parameter width and height of characters). Image Figures 5.2 (a) and (d) are scaled by 54 % to 57% and other images are generated under 52% to 54 % scale.



(a)

(b)

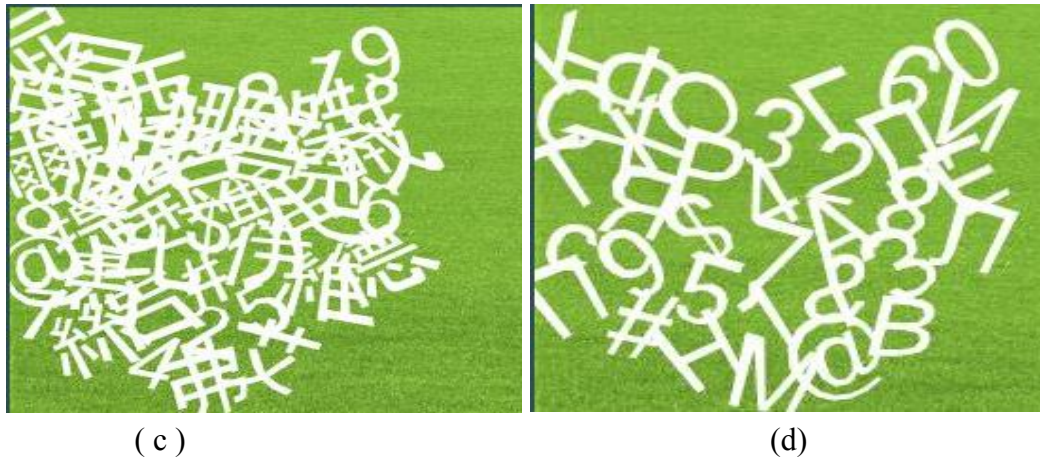


Figure 5.2 Captcha Clickable Image

Figure 5.2 images are scaled with 50% to 60%, angle 30 to 35 degree and waveform characters. Images Figure 5.2 'a' and 'b' represent English alphabets with digits, 'c' is Chinese language and 'd' is Russian language alphabets with digits. Generated images have been tested by above OCR software, recognition results of four software are shown in Table 5.2

Table 5.2. Results of Captcha Breaker against Captcha Images of Figure 5.2 (a)-(d)

Sr.No	Captcha Breaker software	figure (a) English	figure (b) English	figure (c) Chinese	figure (d) Russian
1	GSA Captcha breaker	5.50%	0.00%	0%	7%
2	Captcha sniper	5.50%	5.50%	0.00%	0%
3	www.free-ocr.com	2.50%	0.00%	0%	0.00%
4	www.i2ocr.com	0.00%	0%	0.00%	0%

Table 5.2 illustrates that two characters of English language are recognizable by GSA Captcha breaker and Captcha sniper, and free-ocr.com just recognize one character. In second test of image (b), which also based on English alphabets and digits, the recognition rate are, 0, 5.5, 0, and 0 % of Captcha breaker software according to the given table 5.2. The recognition rate of the Chinese alphabets

language (image Figure 5.2 c) was zero. In Last image Figure 5.2 (d), just GSA Captcha breaker recognized 7.10% of characters (two characters of whole image) and other software did not recognize any character. However, above results just represent the recognition of characters that belong to image. Captcha breaker did not return location or any image coordinate information against characters. Screenshots of several images recognizable results with same parameters shown in Appendix G.

However, proposed image contains several visual objects; proposed image is shown in figure 5.3. Captcha breaker software, which contains different types of filters (threshold, median, max, min, scale, normal etc, detail is available in links [25, 43]). These filters used to recognize and segment the image. Recognition results illustrated in Table 5.3 and screenshot of recognition results are mention in Appendix I.



Figure 5.3. Proposed Scheme "Click-on-Captcha-Objects" Output Image

5.4 Comparison of several attacks on known and proposed methods

There are different types of attacks possible on graphical password. I did analysis, which shows the comparative results of proposed scheme, Click-Text and alphanumeric password.

5.4.1 Comparison results of Captcha breakers attacks

Four above software measures Captcha breaker attacks. Results of the Captcha breaker against proposed "Click-on-Captcha-Objects" shown in Table 5.3.

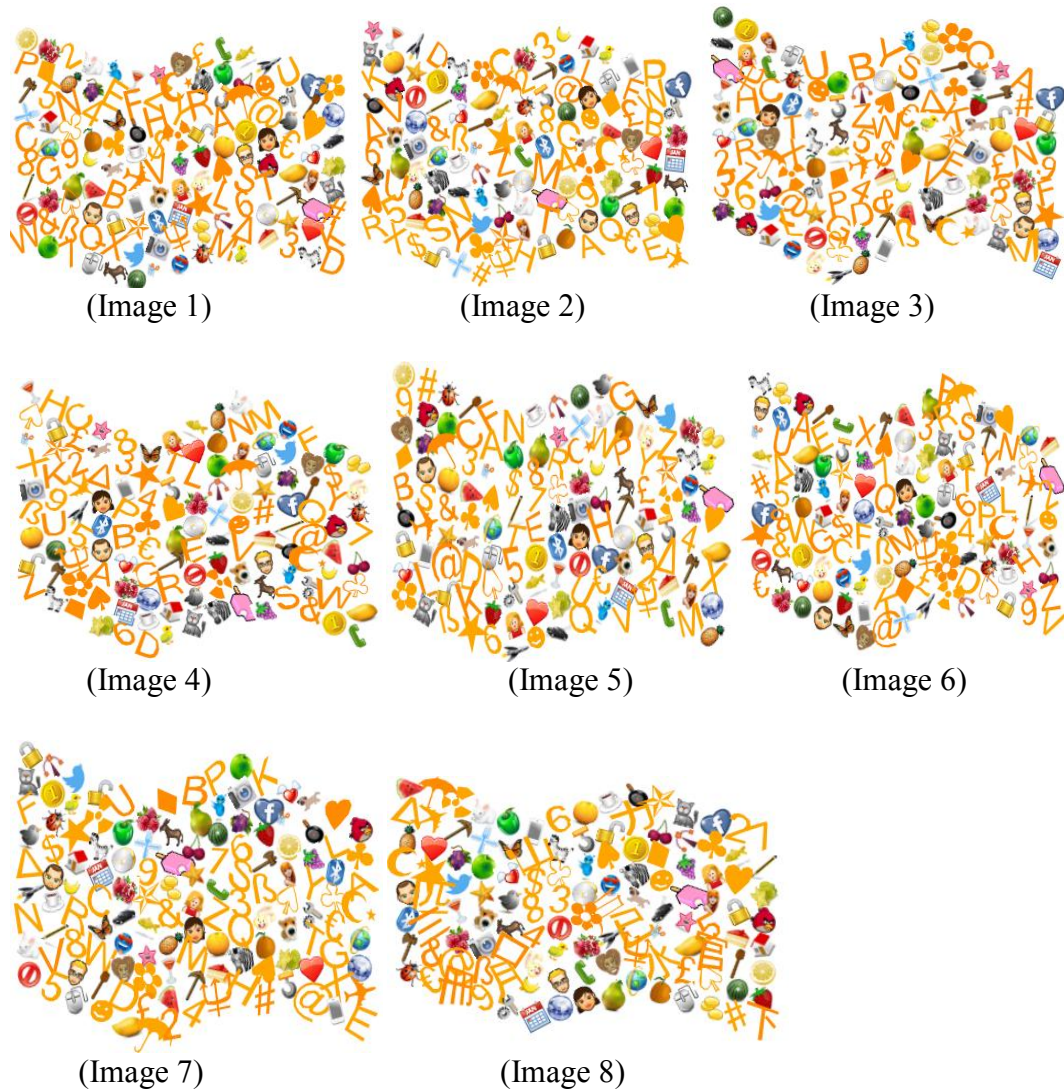


Figure 5.4. Proposed Method Generated Images

Table 5.3. Results of Captcha Breakers Recognition of Proposed Method Images of Figure 5.4

Click-on-Captcha-Objects								
Captcha breakers	Image test 1	Image test 2	Image test 3	Image test 4	Image test 5	Image test 6	Image test 7	Image test 8
GSA Captcha breaker	2.1	0.7	0	4.1	1.42	5	2.8	0
Captcha sniper	0	0	0	0	0	0	0	0
http://www.i2ocr.com	0	0	0	0	0	0	0	0
http://www.free-ocr.com/	2.14	6.24	5.71	5.71	3.5	4.28	2.85	0
Mean	1.06	1.73	1.42	2.45	1.23	2.32	1.41	0
Median	1.05	0.35	0	2.05	0.71	2.14	1.4	0
Variance	1.49	9.12	8.15	8.45	2.73	7.26	2.66	0
Standard deviation	1.22	3.02	2.85	2.90	1.65	2.69	1.63	0
Maximun STD	3.02							
Minimun STD	1.22							

Experimental result shows of Captcha breaker on proposed scheme shows that means of four Captcha algorithms was 1.45 and median was 1.86 and 3.02 and 1.22 is maximum and minimum value of standard deviation respectively. From the other hand, Click-Text method result against Captcha breaker is shown in Table 5.4

All the images, which are the part of these tests, are mentioned below.

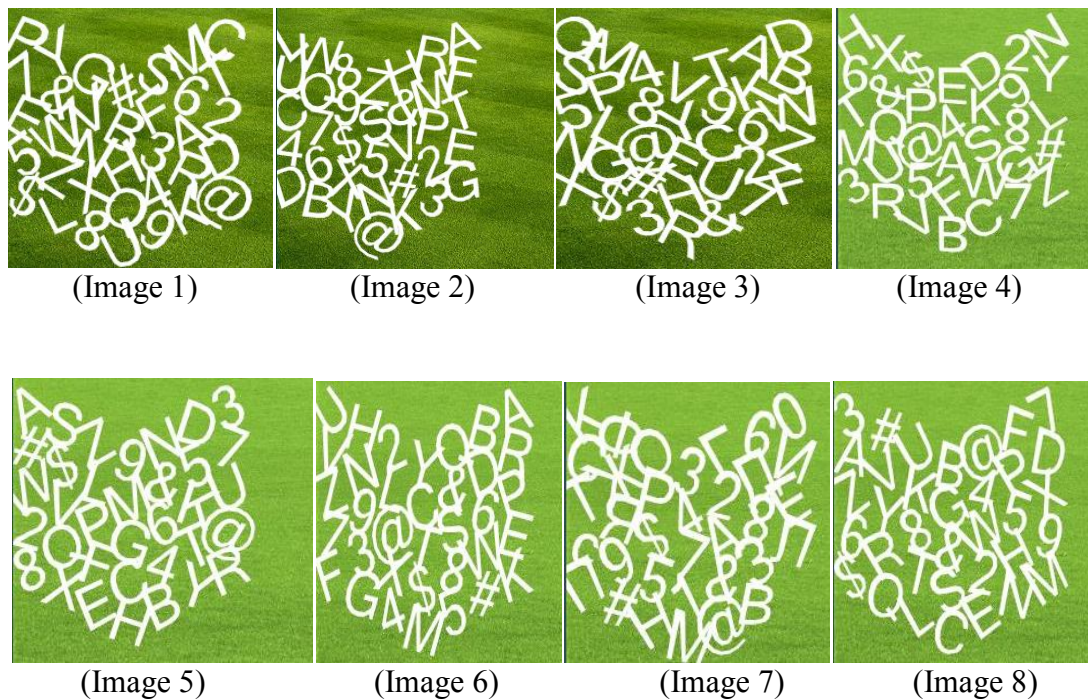


Figure 5.5. Click-Text Scheme [1] Images Used in Captcha Breakers

Table 5.4. Captcha Breaker Results of Click-Text Method of Figure 5.5

Click-Text method								
Captcha breakers	Image 1	Image 2	Image 3	Image 4	Image 5	Image 6	Image 7	Image 8
GSA Captcha breaker	2.7	8	5.5	6	2.7	2.7	10	21.16
Captcha sniper	4	5.5	8.3	8.3	8.3	5.5	5.5	6
Online captcha OCR	13.8	0	0	0	3	5.5	11.11	3
http://www.i2ocr.com	0	0	0	0	0	0	0	0
Mean	5.12	3.37	3.45	3.57	3.5	3.42	6.65	7.54
Median	3.35	2.75	2.75	3	2.85	4.1	7.75	4.5
Variance	3.64	16.22	17.17	17.92	12.06	6.95	25.55	88.44
standard deviation	6.018	4.028	4.14	4.23	3.47	2.63	5.055	9.40
Maximun STD	9.40							
Minimun STD	2.63							

Click-Text Captcha breaker algorithm has average mean value 4.58 and median values of tested images are 3.88. Maximum standard deviation is 9.40 and minimum standard deviation is 2.63. Comparative result shows that proposed method average mean value is less than Click-Text method. It means that the proposed system mean value 1.45 is less than 4.58 that is mean value of Click-Text method. Hence, recognition rate of Captcha breaker against proposed system is lower, the graph shown in Figure 5.6.

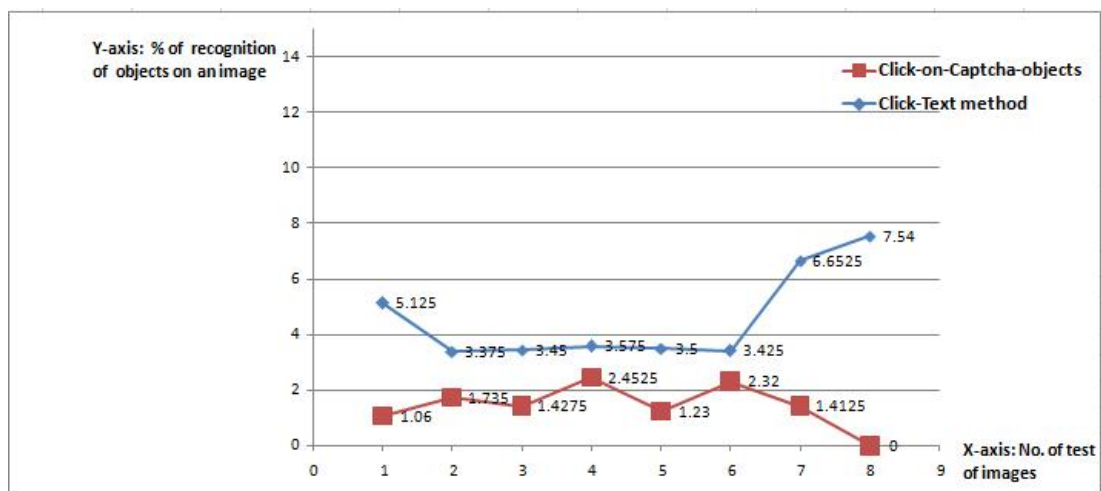


Figure 5.6. Graph of Comparison of Captcha Attack against Click-Text and Click-on-Captcha-Objects Scheme

Figure 5.6 y-axis represents the mean value of number of recognized character in per attempt (mean value is calculated by results of four Captcha breakers) and X-axis is shows the number of attempt on different images.

However, the resistance of proposed scheme is significantly high. The output of character recognition is reliable.

5.4.2 Auto mouse click attack

In this attack, mouse is controlled by a program and it works like human, the mouse motion is controlled by this program and it generates x, y coordinates values randomly to perform click operation on image. However, to reduce this attack, proposed system provides a platform in which clicked point placed more than three times on wrong position, program will close the account of current user. In addition, after third attempt, account will be locked. However, to make experimental approach, MurGee.com program has been downloaded [35]. MurGee.com is auto click software, which performs clicking automatically and is free available on Internet. The clicked location is randomly set and 30, 40, 50, 60, 80 continuous clicks performed on an image. Comparative results of proposed and Click-Text method shown in Table 5.5.

Table 5.5. Comparison of Auto Mouse Click of Click-Text and Proposed Method

Method	Mean value	SD value	Rejection rate
Click-Text	3.28	2.34	95.41%
Click-on-Captcha-Objects	2.55	2.38	97.66%

In above Table, means indicates the value of how much clicks are correctly lie on objects. In addition, rejection rate shows that proposed method rejects 97.66% of

attacks and Click-Text method rejects 95.41%. Both performances are relatively high. Auto mouse attack can be increased by finding specific location of image, but proposed scheme image will change order randomly each time.

5.4.3 Guess-ability attacks

Guessing attack is possible to any type of password. Guessing attack has two types, one is online guessing attack and other is human-based guessing attack. Online guessing attack is very hard to break graphical password, both Click-Text and proposed method have high resistance against online guessing attack but it is easy to break alphanumeric password. Online guessing attack is addressing to Captcha breaker results. If Captcha breaker can break the image, then online guessing attack is possible but according to above Captcha breaker results, it's not easy task to break Captcha image, and proposed system has more complex Captcha image structure. However, attackers can break image using high AI approaches, but proposed system contains user-defined images and system based on mouse-click approach, so it is again difficult task to get an object location.

In Human-based attack, eavesdropper collects the information related to user and try different possibilities to login his/ her account. Human-based guessing attack is possible to reduce but cannot ignore. However, in proposed system guessing attack is reduced by adding number of icon size images and symbols. The probability of human guessing attack is reduced as compared to Click-Text and alphanumeric password. Let consider 150 objects on image and password length is 8 objects, number of probability of guessing attack is $150^8 = \sim 2^{58}$ which is significantly higher than 2^{40} . To enter manually, it is very tedious long work to break graphical password. After survey of 40 people, most of them select simple useable password

characters, which are easily remember-able. Eavesdropper can easily access by guessing user name or family name, books etc. More than 50% password belongs to usual alphabets. To reduce this guessing attack, proposed method has strong ability. Proposed method has also alphabets but mixture with images, which reduces guessing attack.

5.4.4 Brute force and dictionary attacks

Brute force and dictionary attack are approaches in which all the possibilities of password are applied to get original password. In graphical password, it is very difficult task to recognize the object and then combine the objects for successfully login. According to Captcha breaker and auto mouse attacks results, brute force attack is negligible on proposed system.

Proposed system "clicked on Captcha objects" is not bullet proof for all attacks. Shoulder surfing attack is common attack for the entire graphical password scheme. However, it can be reduced by adding dual-view technology effect.

5.5 Comparison of password complexity and memorability

In this section, survey of 40 people has been taken. Three schemes are compared to find the memorability of password. One is "Click-on-Captcha-Objects" scheme, second is Click-Text and third is traditional alphanumeric password system with Captcha. First two methods are clicked-based schemes and last is traditional keyboard based password scheme. Complexity of password has been noted for entire method. To fulfill security needs, password should be at least 8 alphanumeric, which should not be repeatable characters or should not be name or any dictionary word and at least 2 digits should be part of password. On the other hand, the proposed system has been based on symbols, images, alphabets and digits. Therefore, I gave

instructions to users to follow all of above mentioned requirements. To measure the performance of each scheme, three-week experimental results have been taken. Users were authenticated by three schemes and they have to select complex password. In first attempt each user register and after a few times each user checked that he/ she remembered password or not. Each user attempted second and third week with the same user name and password. Analysis report shows that if password complexity increases, forget ratio of the password also increases. Table 5.6 shows the percentage of three methods that 40 people attempted to register and login, and how much people forget their password. In first attempt nobody forget password, and same was in first week, but in second and third week ratio is decreased, in Click-Text scheme, second week, one forgot full password and two persons forgot half passwords, and in third week, two users forgot full password and two of them remembered just 4 to 5 characters.

Table 5.6. Comparison of Complex Password Memorability of Three-Schemes

Method	First week	Second week	Third week
Click-Text	100%	95.00%	93.00%
Text+Captcha	98.75%	95.00%	88.75%
Proposed method	100%	98.75%	97.25%

Text + Captcha method has similar like text password but in Text + Captcha user has to pass extra Captcha challenge. The implementation details are available in Appendix J. According to survey report, the Table 5.6 second row shows Text + Captcha performance, in first week one person forgot half password, in second week, three students (users) forgot nearly 50% password and in third week, three persons

forgot full password and two person remembered 3 to 6 characters. Third scheme is proposed Click-on-Captcha-Objects scheme, first attempt was the same like above method, but in the second week, one user forgot half password and in the third week, two users forgot the half password. However, to remember the complex password proposed method performance was better than Click-Text and alphanumeric (text) based password. According to survey of Bin and Zhu et al. [1], the performance of Click-Text method was 97.5% of 39 users. On the other hand, the accuracy of Captcha + Text scheme was 85% and 34 users participated. However, results varied between proposed and already known results because each time users changed and each user has own memorability power.

Consequently, above results show that "Click-on-Captcha-Objects" stronger than Click-Text and alphanumeric password. The proposed system has alphanumeric, icon size images and symbols, which makes five time longer password as compared to Click-Text and alphanumeric password. In contrast, alphanumeric (text) passwords can be easily breakable according to Bin, Zhu et al. [1] implementation results; they break two passwords from 40 in 24 hours. On the other hand, Click-Text based password has limited number of characters that are easy to guess by human and according to survey alphanumeric and Click-Text password both are easy for user to remember, however it's hard and tedious task for user to remember complex password for long time and simple password is easily guessable. Hence, proposed method is more reliable than other two methods.

5.6 Survey of the same complex password of three schemes

In this survey, I conducted three days experimental results of 5 users (all the users were males and all were the students of bachelor and Master classes and average age

was 26.33). Constant and complex password is assumed for each scheme. However, password of proposed scheme was, "RED11🚩❤️🐶". Password of Click-Text scheme was, "NTMPK887" and password of Captcha + text password scheme was, "PYmbe389". Each user was authenticated two times a day and I conducted memorability results, shown in the graph below. Percentage of memorable password depends on number of characters which user recalls. If half is recalled, memorable password considered as 50%. Hence, accuracy of passwords of three schemes is, 75.33%, 84.33% and 91.66% of Captcha+Text, Click-Text and proposed method respectively. However, results represent that proposed method performance is better than that of Click-Text and Captcha + text method. Details of conducted results are mentioned in Appendix N. Memorability is increased for all schemes by taking practice of the same password. Graph in figure 5.7 illustrates the performance of the three methods.

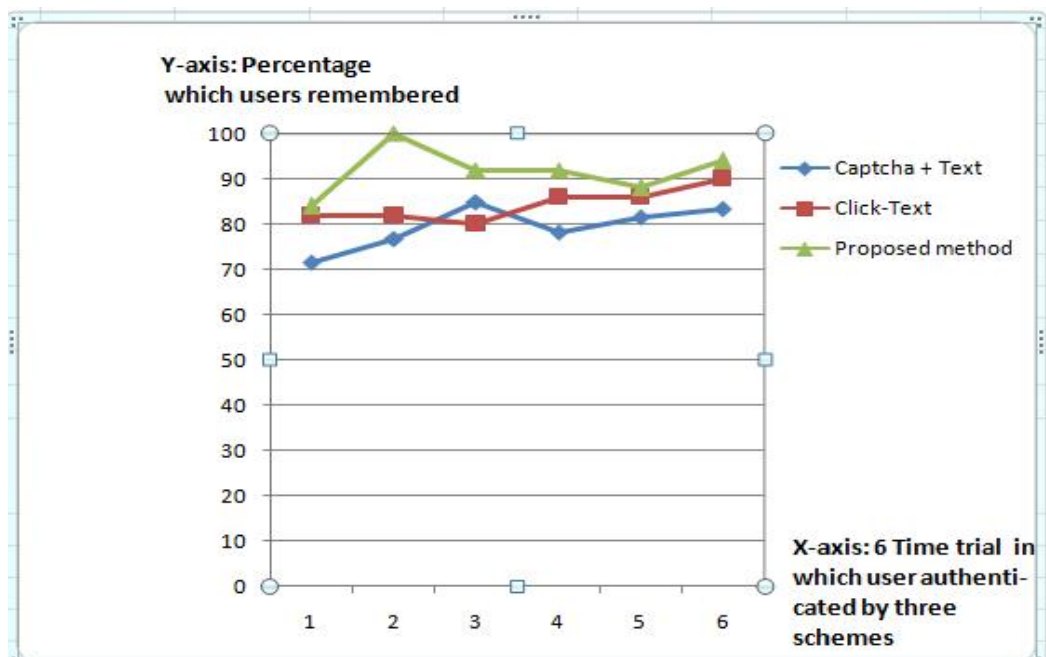


Figure 5.7. Graph of Memorable Password of Three Schemes in 3 Days Survey

5.7 Comparison of convenience usability and time of authentication

▪ Convenience usability

Usability of graphical password increases with number of trials, in first trial user takes much time and in second attempt user feels comfortable and login time decreases. However, during survey a number of questions were asked and Table 5.7 shows the comparison results of ease to use the proposed method with alphanumeric password. I assign value from 1 to 10 to each question, and 10 represents "much easy" or "much difficult".

Table 5.7. Ease Use of Click-on-Captcha-Objects Question to Users

Sr.No	Questions	Mean	Median	SD (σ)
1	How much easy to create Graphical password	7.91	8	0.56
2	How much easy to login	8.11	7.45	1.24
3	Text+ Captcha Password easier than Graphical Password	5.06	6	0.67
4	How much Graphical Password strong as compare to Click-Text and alphanumeric password	8.75	9.25	0.84
5	Other people will choose different Graphical password as compare to you	8.75	8.75	1.26
6	In Practice, User can enter	7.04	5.5	1.29
7	Proposed method is easy to remember	8.37	7.5	0.95

Table 5.7 shows the review of users against above three methods. How much easy to create graphical password means value is 7.9 and median is 8. The standard deviation is 0.56. How much easy for user to login mean value is 8.11, median value is 7.45 and SD is 1.24. Compare to strong password to alphanumeric and Click-Text, means 8.75, median 9.25 and SD is 0.84. Another important question is graphical password is "easy to remember" as compared to Click-Text and alphanumeric-based password,

user selection mean value is 8.37, median 7.5 and SD is 0.95. Another related question in practice and text + Captcha is "easier to set password", mean and median value shown in above Table. However, Questionnaires Table also nominates the performance of proposed system as compared to Click-Text method.

▪ **Authentication Time**

Authentication time of any password scheme, vary from user to user. Incremental number of trail establishes positive results of authentication time. Above three methods are employed by 40 people. The authentication time of alphanumeric password was shorter than both proposed and Click-Text method. Table 5.8 shows the three schemes login time means and SD value of 40 users, conducted experiments Table and people who involved and login time of three algorithms are described in Appendix M. Means value of Click-Text is 38.02 seconds, proposed method has means value 42.16 seconds. In contrast, alphanumeric means value is 26.63 seconds. Time of the login is very low as compared to both Click-Text and proposed method. However, alphanumeric password has other security issues. Standard deviation (SD σ) of Click-Text, alphanumeric and proposed method is 4.94 and 8.55 respectively.

Table 5.8. Authentication Time (s) of Three-Password Schemes

Methods	Mean time (s)	SD (σ)	Maximun(s)	minimun (s)
Click-Text	38.05	16.26	60.4	25.3
Text + Captcha	26.63	4.31	45.4	15
Proposed method	42.16	7.85	71.5	23.7

Maximum login attempt of Click-Text, Text + Captcha and proposed method is, 60.4, 45.4 and 71.5 respectively. Similarly, minimum values shown in Table 5.8.

Proposed method means value represents that user take more time as compared to Click-Text, reason is that proposed method has a number of visual objects as compared to Click-Text. Graph 5.8 represents the average time (s) for three schemes and Graph in Figure 5.9 shows the authentication time of three schemes. Y-axis shows the time (seconds) and x-axis represents that how many users tried for authentication against three algorithms.

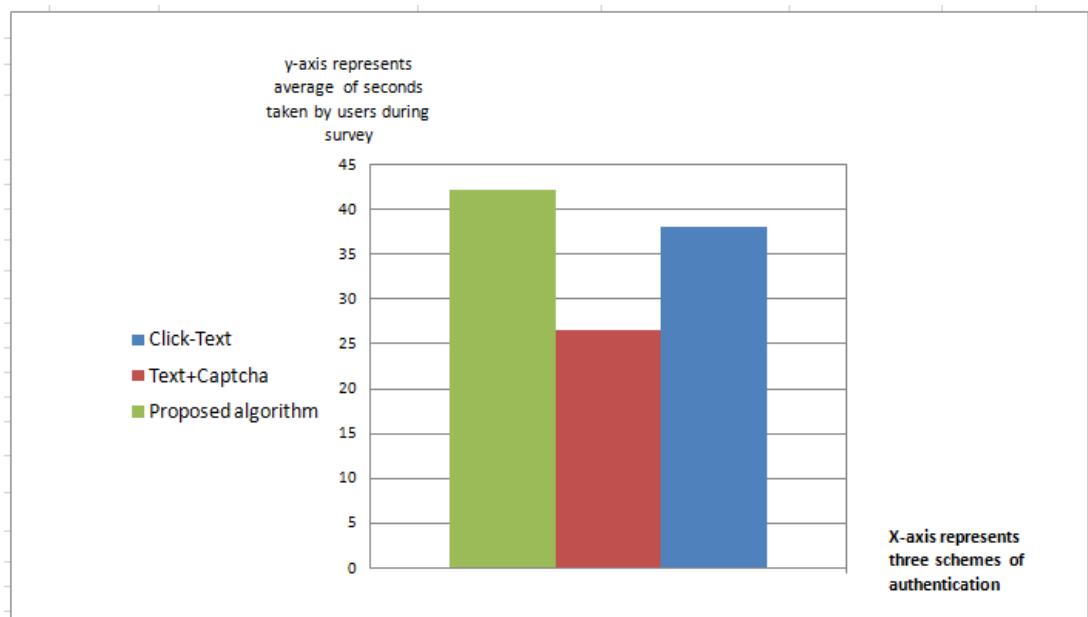


Figure 5.8. Average Time (s) of Authentication of Three Methods

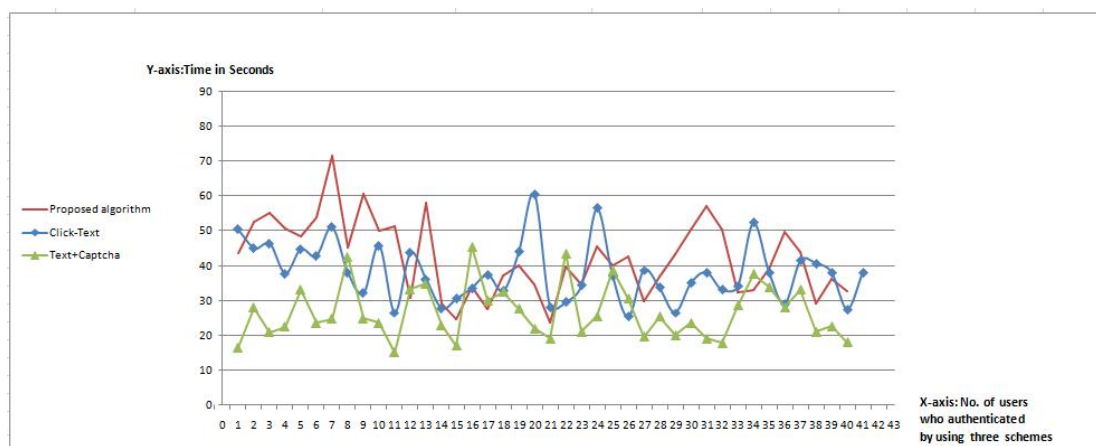


Figure 5.9. Authentication Time (s) of 40 Users Against Three Algorithms

However, already known method survey shows that they conducted results shown in Table 5.9, they implemented five schemes, but related schemes are Click-Text and P+C(Text+ Captcha). Both have time T (second) of authentication of 40 users are 27.22 and 28.24 respectively. However, in my survey the performance of authentication time is 38.05 and 26.63 respectively. May be, this difference is generated due to the variation of the code and people.

Table 5.9. Time (s) of Authentication of Known methods (Click-Text and Captcha+ Text [1])

Scheme	ClickText	Animal Grid	PassPoints	P+C	Text
T (s)	27.22	29.20	21.62	28.24	10.34
σ (s)	17.38	19.24	12.29	12.55	6.08
Max.(s)	65.62	88.51	45.17	50.84	31.25
Min.(s)	10.41	13.46	8.36	13.7	3.58

Chapter 6

CONCLUSION AND FUTURE WORK

Click-on-Captcha-Objects method is based on unsolvable Captcha "hard AI problem". Click-on-Captcha-Objects method contains alphanumeric, special symbols and user defined images. Authentication is done by clicking on some objects rather than to enter entire Captcha characters. Objects are easily recognizable by users but it is hard to be the recognized by system/robots.

Graphical password is option plan of alphanumeric password that is exceptionally tedious procedure to review the unpredictable secret word. Mentally investigations of human personality contend that reviewing of picture is simpler than letter sets or digits. I proposed a strategy for acknowledgment based confirmation, based on Captcha innovation. I indicated proposed strategy as "Click-on-Captcha-Objects", which contains Captcha based visual objects and it approaches to strong password.

Proposed strategy has been examined by two diverse methodologies. One of them is convenience and another one is security. For ease of use, 40 clients participated and report is prepared that how much client would recall the intricate password, accuracy of achievable results of strong password of proposed scheme is 97.25%, conversely, alphanumeric and Click-Text system has 88.75% and 93% respectively. In addition, 3 days survey of assuming same password for unique scheme represents that proposed scheme is easy to remember to complex password. On the other hand,

security investigation of proposed framework has been finished with diverse sorts of assaults, prominent four-Captcha breakers programming are executed to get Captcha acknowledgment results, then again, proposed framework opposes 98.5% against four Captcha breakers assaults, in opposite Click-Text technique opposes 95.5%. In addition, auto-mouse click attack performed; the accuracy of repulsion of proposed strategy was 97.66%, and Click-Text technique rejection rate was 95.41%. The outcomes demonstrate that there is generous variety to recollect the complex password compared to alphanumeric and Click-Text based validation plans. Consequently, execution of proposed system is superior to alphanumeric and Click-Text strategy. Traditional scheme of authentication mostly lead to guessable and unreliable password, "Click-on-Captcha-Objects" gives fair security and convenience to validate genuine client. In order to check the time of image at server, an experiment has been conducted at SAMSUNG (Core i5, RAM 4 GB, Processor 2.53 GHz) portable PC and conducted result was ~40 milliseconds for each "Click-on-Captcha-Objects" image. The execution time of proposed scheme would be reduced to get advantage of parallel processing technology. However, it offers reasonable security and convenience.

Consequently, proposed method can reduce online guessing-attack probability nearly to zero, and it can be applicable for online email services. Human guessing attack of graphical password reduced as compared to already proposed scheme, (Click-Text and PassPoint and alphabetic+ Captcha). On the other hand, it provides complex password, which is easy to remember and recognize for humans and hard to break by robots.

"Click-on-Captcha-Objects" can be refined with complex 3D structure of visual objects for useful future work. Moreover, proposed scheme provides reasonable security and usability for practical applications.

REFERENCES

- [1] Zhu, B. B., Yan, J. D., Bao, G., Yang, M., & Xu, N. (2014). Captcha as graphical passwords - a new security primitive based on hard AI problems. *IEEE Transactions on Information Forensics and Security*, 9(6), 891-904.

- [2] Von Ahn, L., Blum, M., Hopper, N. J., & Langford, J. (2003). CAPTCHA: Using hard AI problems for security. In *Advances in Cryptology - International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT)*, May 4–8, 2003 (pp. 294-311). Springer, Berlin, Heidelberg.

- [3] Adams, A., & Sasse, M. A. (1999). Users are not the enemy. *Communications of the ACM*, 42(12), 40-46.

- [4] Jain, A. K., Ross, A. A., & Nandakumar, K. (2011). *Introduction to biometrics*. Springer Science & Business Media, New York, ISBN 978-0-387-77325-4, p. 311.

- [5] Zhang, F., Kondoro, A., & Muftic, S. (2012). Location-based authentication and authorization using smart phones. In *Trust, Security and Privacy in Computing and Communications (TrustCom), 11th International Conference on Trust, Security and Privacy in Computing and Communications* June 25-27, 2012 (pp. 1285-1292). IEEE. Liverpool, United Kingdom.

- [6] Elftmann, P. (2006). Secure alternatives to password-based authentication mechanisms. *Lab. for Dependable Distributed Systems*, Diploma thesis,

Laboratory for Dependable Distributed Systems RWTH Aachen, University Aachen, Germany.

- [7] Blonder, G. E. (1996). *U.S. Patent No. 5,559,961*. Washington, DC: U.S. Patent and Trademark Office.

- [8] Shepard, R. N. (1967). Recognition memory for words, sentences, and pictures. *Journal of Verbal Learning and Verbal Behavior*, 6(1), 156-163.

- [9] De Angeli, A., Coventry, L., Johnson, G., & Renaud, K. (2005). Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. *International Journal of Human-Computer Studies*, 63(1), 128-152.

- [10] Jebriel, S. M. (2014). *Empirical approach towards investigating usability, guessability and social factors affecting graphical based passwords security* (Doctoral dissertation, University of Glasgow).

- [11] Suo, X., Zhu, Y., & Owen, G. S. (2005). Graphical passwords: A survey. In *Computer Security Applications Conference, 5-9 Dec. 2005, 21st Annual* (10 pp-472). IEEE.

- [12] Wiedenbeck, S., Waters, J., Birget, J. C., Brodskiy, A., & Memon, N. (2005). Authentication using graphical passwords: Effects of tolerance and image choice. In *Proceedings of the 2005 Symposium on Usable Privacy and Security*, July 6-8, 2005 (pp. 1-12). Pittsburgh, PA, USA.

- [13] Rundus, D. (1971). Analysis of rehearsal processes in free recall. *Journal of Experimental Psychology*, 89(1), 63-77.
- [14] Sasse, M. A., Brostoff, S., & Weirich, D. (2001). Transforming the ‘Weakest link’—a human/computer interaction approach to usable and effective security. *BT Technology Journal*, 19(3), 122-131.
- [15] Jermyn, I., Mayer, A. J., Monroe, F., Reiter, M. K., & Rubin, A. D. (1999). The design and analysis of graphical passwords. In *Usenix Security*. Washington, D.C., USA, 23-26 Aug. 1999.
- [16] Biddle, R., Chiasson, S., & Van Oorschot, P. C. (2012). Graphical passwords: Learning from the first twelve years. *ACM Computing Surveys*, 44(4), art. No. 19.
- [17] Zakaria, N. H., Griffiths, D., Brostoff, S., & Yan, J. (2011). Shoulder surfing defence for recall-based graphical passwords. In *Proceedings of the Seventh Symposium on Usable Privacy and Security*, July 20-22, 2011, (art. No. 6). ACM, Pittsburgh, PA, USA.
- [18] Tao, H., & Adams, C. (2008). Pass-Go: a proposal to improve the usability of graphical passwords. *International Journal of Network Security*, 7(2), 273–292.
- [19] Wiedenbeck, S., Waters, J., Birget, J. C., Brodskiy, A., & Memon, N. (2005). PassPoints: Design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies*, 63(1), 102-127.

- [20] PassFaces, <http://www.realuser.com/>, Last accessed Jun-07-2015.
- [21] De Angeli, A., Coventry, L., Johnson, G., & Renaud, K. (2005). Is a picture really worth a thousand words? Exploring the feasibility of graphical authentication systems. *International Journal of Human-Computer Studies*, 63(1), 128-152.
- [22] Davis, D., Monroe, F., & Reiter, M. K. (2004). On user choice in graphical password schemes. In *USENIX-Advance Computing System Association Security Symposium*, 9–13 Aug 2004. (Vol. 13, pp. 11-11).
- [23] Albayati, M. R., & Lashkari, A. H. (2014). A new graphical password based on decoy image portions (GP-DIP). In *2014 International Conference on Mathematics and Computers in Sciences and in Industry (MCSI)*, 13-15 Sept. 2014 (pp. 295-298). IEEE. Cyberjaya, Malaysia.
- [24] Hayashi, E., Dhamija, R., Christin, N., & Perrig, A. (2008). Use your illusion: secure authentication usable anywhere. In *Proceedings of the 4th Symposium on Usable Privacy and Security* July 23 - 25, 2008 (pp. 35-45). ACM, Pittsburgh, PA, USA.
- [25] Alia M., Hnaif, A., Al-Anie, H. & Tamimi. A. (2012). Graphical Password Based On Standard Shapes. *Science Series Data Report*, 22-24 Oct. 2014 Vol. 4(2). Al-Zaytoonah University of Jordan.

- [26] <http://rutgersscholar.rutgers.edu/volume04/sobrbirg/sobrbirg.htm/>, Last accessed Mar-15-2015.
- [27] Thorpe, J., & van Oorschot, P. C. (2007). Human-seeded attacks and exploiting hot-spots in graphical passwords. In *16th USENIX –Advance Computing Systems Association Security Symposium* Aug 6-10, 2007 (Vol. 7, pp. 103–118), Boston, USA.
- [28] Zakaria, O., Zangooui, T., & Mohd Shukran, M. A. (2012). Graphical password authentication: review and analysis. *Advances in Information Sciences & Service Sciences*, 4(15), 25-32.
- [29] Hlywa, M., Biddle, R., & Patrick, A. S. (2011). Facing the facts about image type in recognition-based graphical passwords. In *Proceedings of the 27th Annual Computer Security Applications Conference* 05-09, Dec. 2011 (pp. 149-158). ACM. Carleton University ,Ottawa, Canada.
- [30] Anton, H., & Rorres, C. (1994). *Elementary linear algebra: application version - 7th edition*, Howard, Drexel University, ISBN 0-471-58741-9.
- [31] <http://www.gsa-online.de/>, Last accessed Feb-21-2015.
- [32] <http://www.Captchasniper.com/>, Last accessed Feb-21-2015.
- [33] <http://www.i2ocr.com/>, Last accessed Feb-21-2015.

[34] <http://www.free-ocr.com/>, Last accessed Feb-21-2015.

[35] <http://www.murgee.com/auto-clicker/> , Last accessed Mar-10-2015.

[36] Lillibridge, M. D., Abadi, M., Bharat, K., & Broder, A. Z. (2001). Method for selectively restricting access to computer systems. *U.S. Patent No. 6,195,698*. Washington, DC: U.S. Patent and Trademark Office.

APPENDICES

Appendix A. "Click-on-Captcha-Objects" Image Generation Code

```
private void drawimage(List<string> str)
{
    str = regs.Generate_random(str);
    int counter = 0;
    for (int i = 0; i < str.Count(); i += 15)
    {
        counter++;
    }
    int height = counter * 50 + waveamp;
    int width;
    width = (str.Count()) * 30;

    try
    {
        Bitmap bm = MakeCaptchaImge(str, 50, 60, width, height);
        Random RNG = new Random();
        imgRegister1.Image = bm;
        // bm.Dispose();
    }
    catch (Exception x) { }
}
```


Appendix B. Selection of Objects by Clicked on Image

```
PointF[] RF = AuthList[i].pts;
try
{
    Random rd = new Random();
    int ind = rd.Next(2, 3);
    if ((RF[0].X + ind) < (float)(x) && (RF[0].Y + ind) < (float)(y) && (RF[1].X - ind)
        > (float)(x) && (RF[1].Y + ind) < (float)(y) && (RF[2].X + ind) < (float)(x) &&
        (RF[2].Y - ind) > (float)(y) && (RF[3].X - ind) > (float)(x) && (RF[3].Y - ind) >
        (float)(y))
    {
        stepcounter = 0;
        if (PointInTriangle(x, y, RF[0], RF[1], RF[2]) == true)
        {
            returnstring += AuthList[i].txt;
            realpassword += AuthList[i].txt;
            txtpassowrd.Text += passwordlenght++;
            break;
        }

        else if (PointInTriangle(x, y, RF[1], RF[2], RF[3]) == true)
        {
            returnstring += AuthList[i].txt;
            txtpassowrd.Text += passwordlenght++;
            realpassword += AuthList[i].txt;
            break;
        }

    }

}
catch (Exception ex) { }
```

Appendix C. Implementation of Hash Function

```
public string encrypted_function( string st)
{
    string secretKey = st;
    string salt = "encryptedkeyforgraphicalpassword";
    System.Security.Cryptography.SHA1 sha =
    System.Security.Cryptography.SHA1.Create();
    byte[] preHash = System.Text.Encoding.UTF32.GetBytes(secretKey + salt);
    byte[] hash = sha.ComputeHash(preHash);
    string password = System.Convert.ToBase64String(hash, 0, 15);
    return password;
}
```

Appendix D. Add User Defined Images/ Objects

```
OpenFileDialog open = new OpenFileDialog();
open.Filter = "Image Files(*.jpg; *.png; *.jpeg; *.gif; *.bmp)|*.jpg; *.jpeg; *.gif;
*.bmp";
if (open.ShowDialog() == DialogResult.OK)
{

Image yourImage = resizeImage(new Bitmap(open.FileName), new Size(26, 26));
Guid newGuid = Guid.NewGuid();
string imageName = newGuid.ToString();
yourImage.Save("~\\special image\\" + imageName + ".png");
Addin_ps(imageName);
MessageBox.Show("Image successfully added");
}
else { MessageBox.Show("please select image"); }
```

Appendix E. User Defined Function to Store User Information in Database

```
if (txtuseremail.Text != "" && txtpasswd1.Text != "" && realpassword = realpassword2)
{
if (passwordlength >= 4)
{
try
{
string txtpass = encrypted_function(realpassword);
SqlConnection conn = new SqlConnection(@"Data Source=AKN-LAPPY\SQLEXPRESS;Initial Catalog=password;Integrated Security=True");

conn.Open();

SqlCommand cmmd2 = new SqlCommand("SELECT * FROM gpasswordtest where Email='" + txtuseremail.Text + "'", conn);
cmmd2.CommandType = CommandType.Text;
SqlDataReader reader3 = cmmd2.ExecuteReader();
int id = 0;
while (reader3.Read())
{
id = Convert.ToInt16(reader3["Id"].ToString());
cmmd2.Connection = conn;

} reader3.Close();
conn.Close();
if (id != 0)
{
MessageBox.Show("Name is already exist!");
}
else
{
conn.Open();
SqlCommand cmd = new SqlCommand("INSERT INTO gpasswordtest(Email,Password,InputText) values(@email, @password, @inputtext)");
cmd.CommandType = CommandType.Text;
cmd.Connection = conn;cmd.Parameters.AddWithValue("@email", txtuseremail.Text.ToString());
cmd.Parameters.AddWithValue("@password", txtpass.ToString());
cmd.Parameters.AddWithValue("@inputtext", "0");
cmd.ExecuteNonQuery();
int return_nameid = 0;
SqlCommand cmd2 = new SqlCommand("SELECT * FROM gpasswordtest where Email='" + txtuseremail.Text + "'", conn);
cmd2.CommandType = CommandType.Text;
```

```
SqlDataReader reader = cmd2.ExecuteReader();
while (reader.Read())
{
return_nameid = Convert.ToInt16(reader["Id"].ToString());
cmd2.Connection = conn;

}
reader.Close();
```

```
MessageBox.Show("Saving is done!");
```

```
conn.Close();
successlogin ins = new successlogin();
ins.MdiParent = this.MdiParent;
this.Close();
ins.ShowDialog();
}
}
catch (Exception ex) { }
}
else { MessageBox.Show("please select at least 8 objects from image"); }}
```

Appendix F. OCR Results of Captcha Image (Online i2OCR)

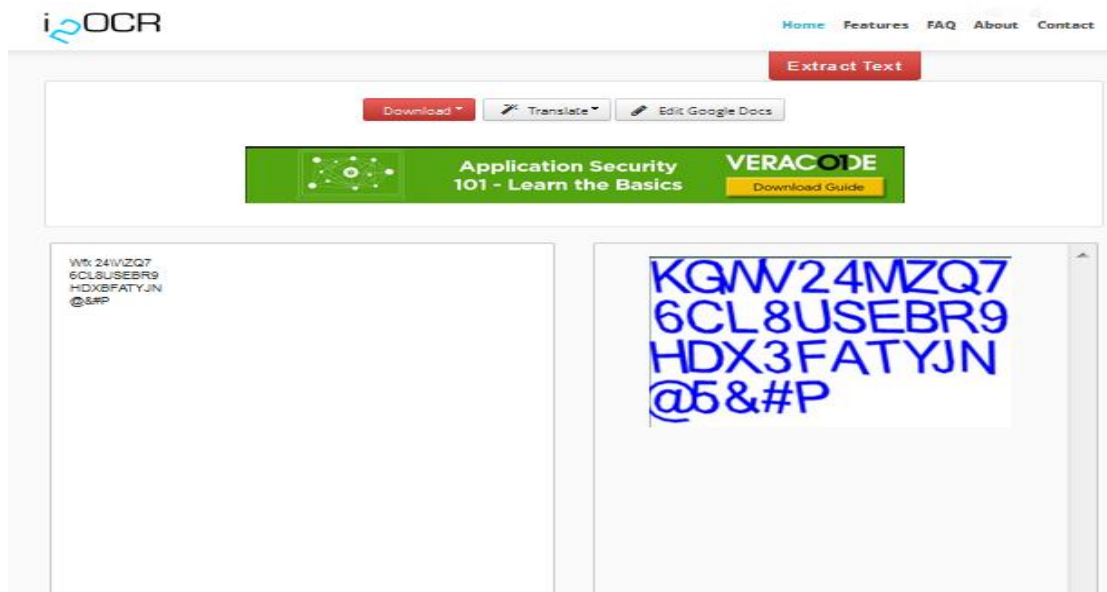


Figure F(1). Shows That How Did i2ocr Capture the Characters on Image

Appendix G. Screenshot of Captcha Breaker Software (GSA and Captcha Sniper)

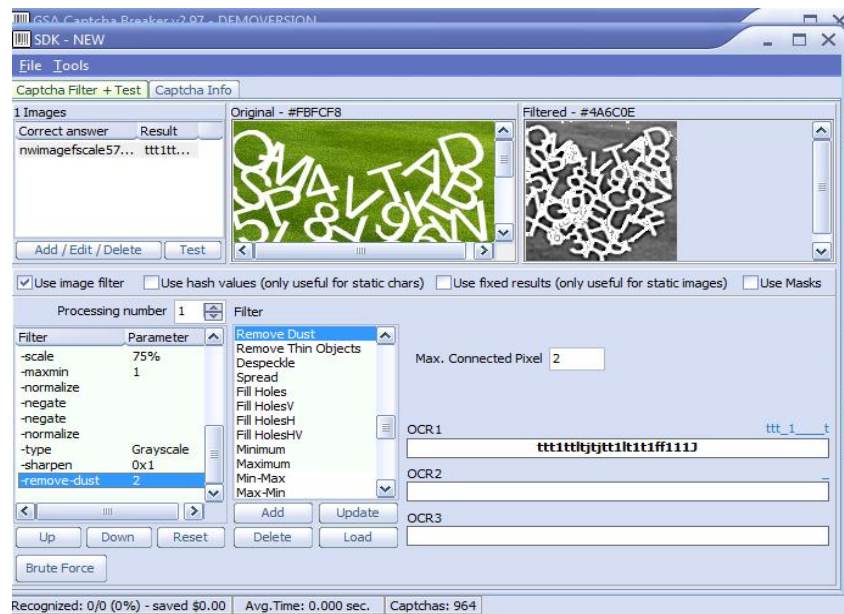


Figure G(1). GSA Captcha Breaker Screenshot.

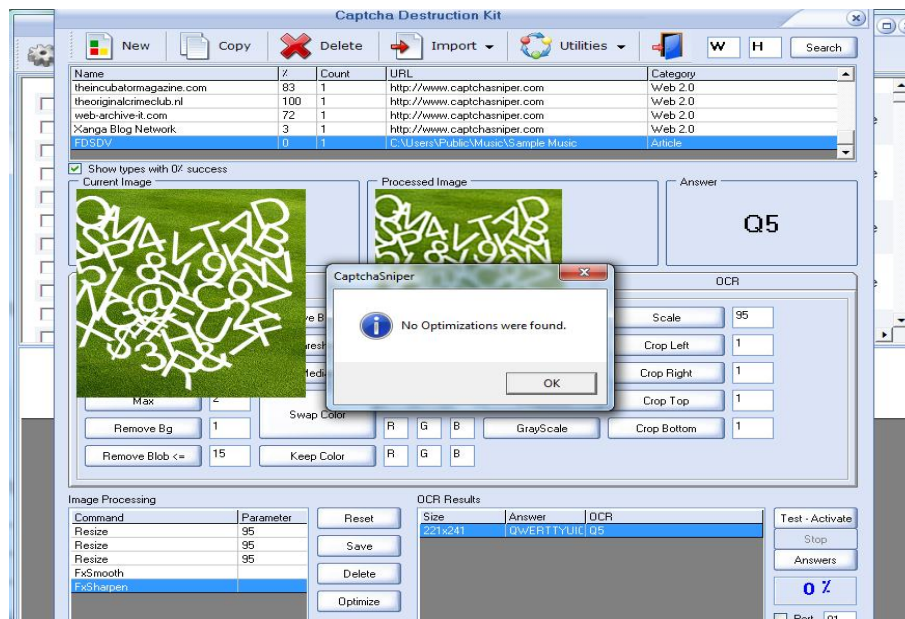


Figure F(2). Sniper Captcha Breaker Recognition Result.

Appendix H. Interface of Captcha Alphabets Image as Graphical Password

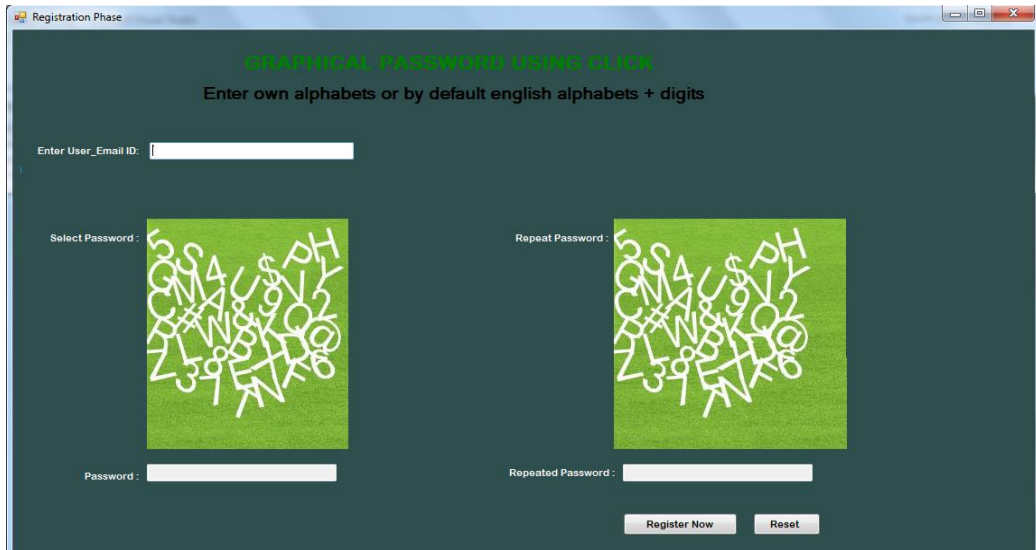


Figure H(1). Interface of Registration of Click-Text

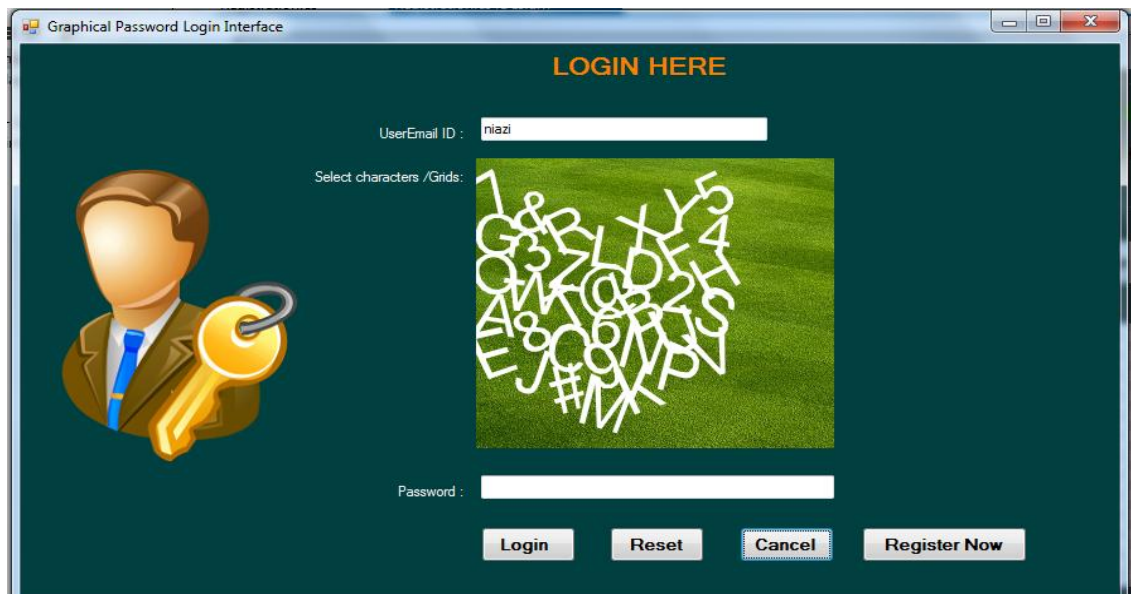


Figure H(2). Interface of Login of Click-Text

Appendix I. Captcha Breaker Screenshot of Proposed Scheme Image

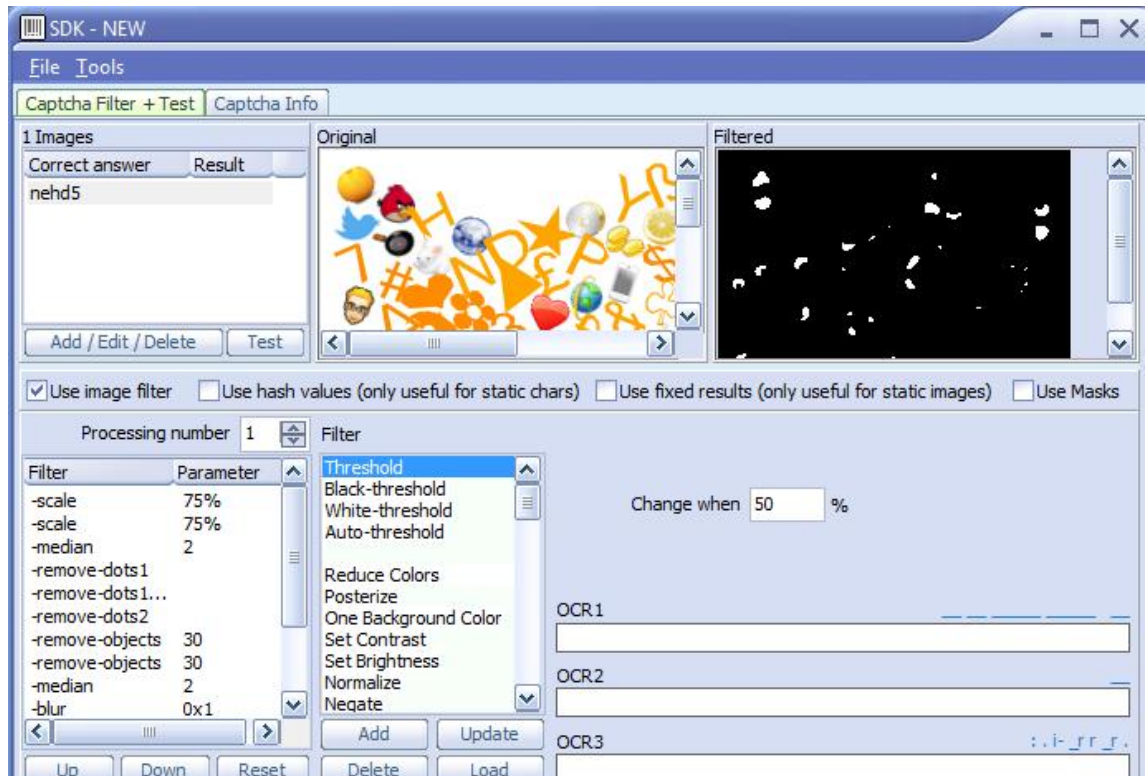


figure I(1). GSA Captcha Breaker Screenshot of Objects Recognition

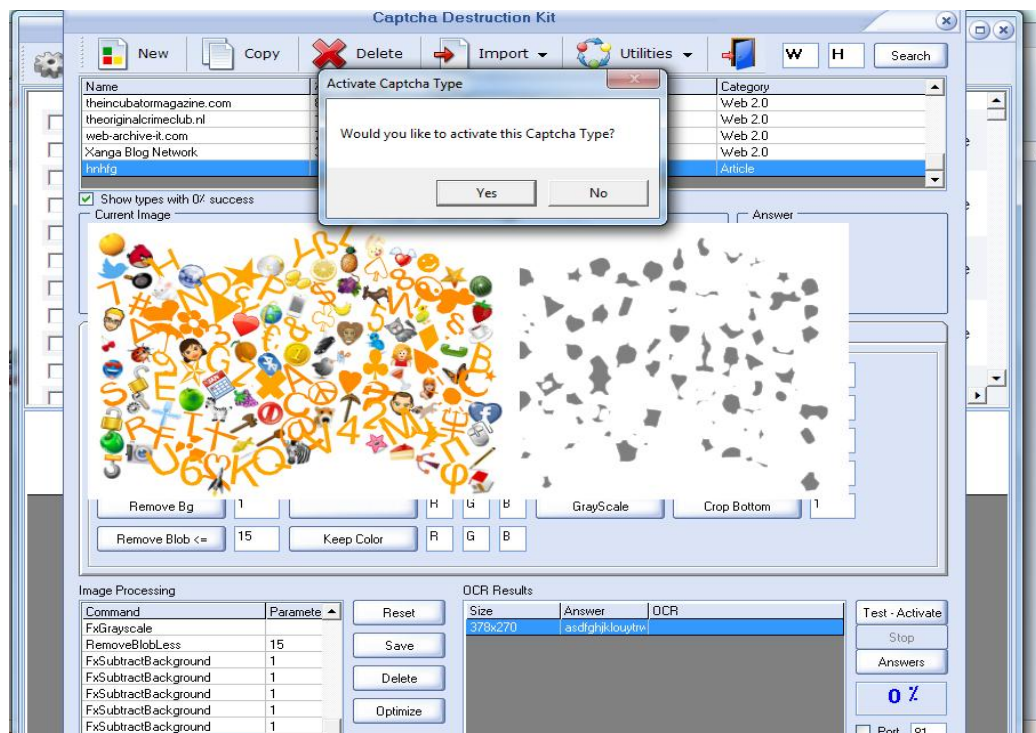
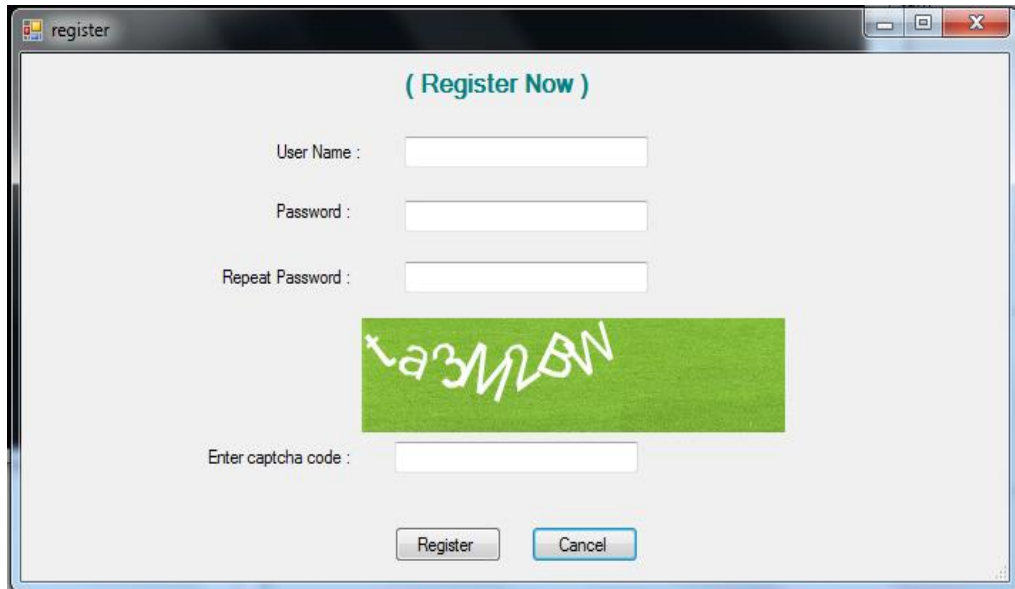


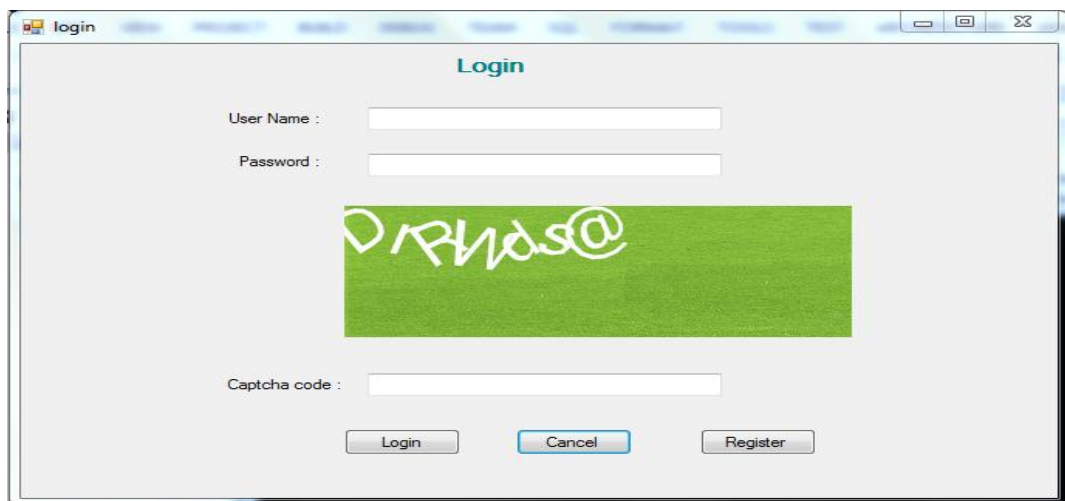
Figure I(2). Captcha Sniper Recognition Results of Objects.

Appendix J. Implementation of "Captcha + Text" Registration and Authentication



The screenshot shows a window titled "register" with a light gray background. At the top center, the text "(Register Now)" is displayed in a teal font. Below this, there are three input fields: "User Name :", "Password :", and "Repeat Password :". Each field is a simple white rectangle. Below the password fields is a green rectangular area containing a white captcha image with the text "1a3M2BN". Underneath the captcha is an input field labeled "Enter captcha code :". At the bottom of the window, there are two buttons: "Register" and "Cancel".

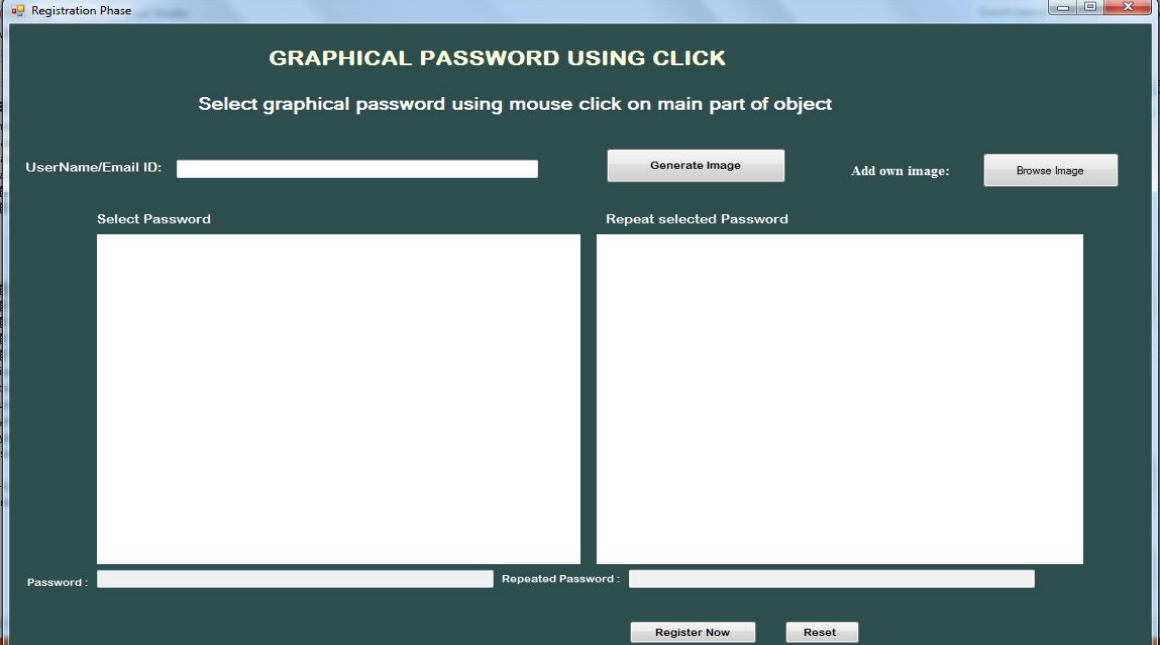
Figure J(1). Interface of Registration of Captcha + Text Scheme



The screenshot shows a window titled "login" with a light gray background. At the top center, the text "Login" is displayed in a teal font. Below this, there are three input fields: "User Name :", "Password :", and "Captcha code :". Each field is a simple white rectangle. Below the password field is a green rectangular area containing a white captcha image with the text "D/RHd5@". At the bottom of the window, there are three buttons: "Login", "Cancel", and "Register".

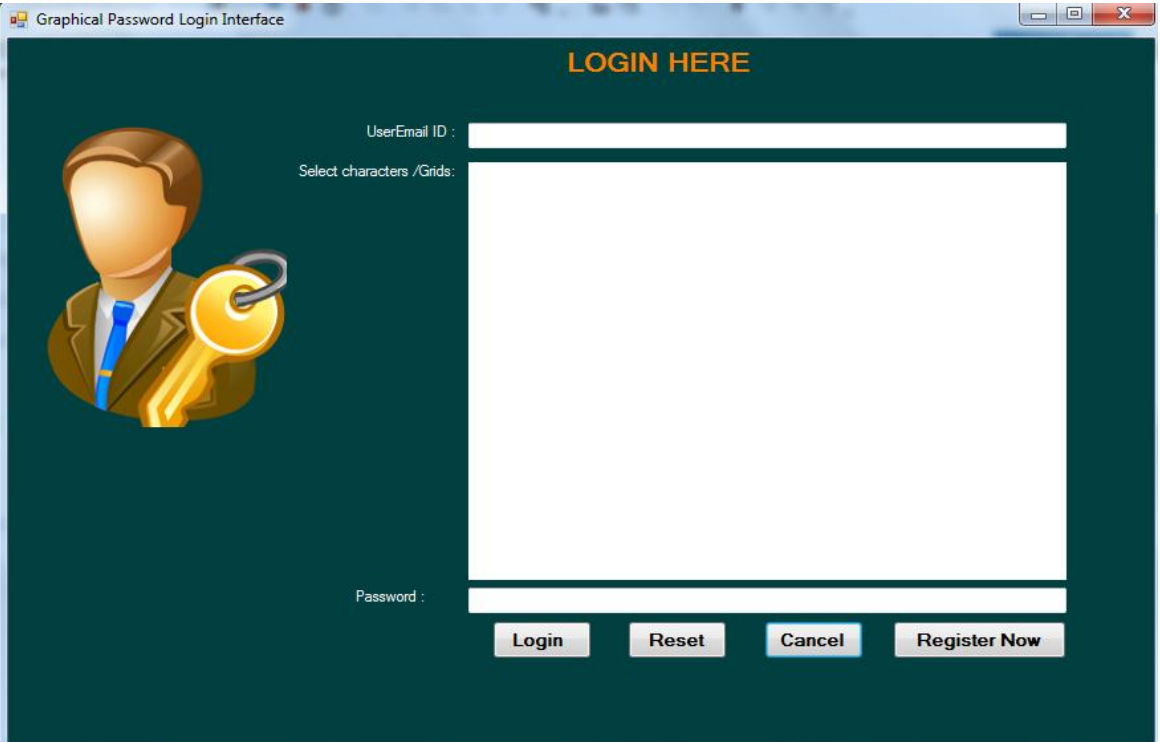
Figure J(2). Interface of Authentication Step of Captcha + Text Scheme

Appendix K. Proposed Method Interface of Registration and Authentication



The image shows a software window titled "Registration Phase" with a dark green background. At the top, it says "GRAPHICAL PASSWORD USING CLICK" and "Select graphical password using mouse click on main part of object". Below this, there is a text input field for "UserName/Email ID:" followed by a "Generate Image" button. To the right, there is an "Add own image:" label and a "Browse Image" button. The main area is divided into two large white rectangular boxes: "Select Password" on the left and "Repeat selected Password" on the right. At the bottom, there are two text input fields labeled "Password:" and "Repeated Password:", and two buttons: "Register Now" and "Reset".

Figure K(1). Registration Interface of Proposed System.



The image shows a software window titled "Graphical Password Login Interface" with a dark green background. At the top, it says "LOGIN HERE" in orange. On the left side, there is a 3D illustration of a man in a suit holding a large golden key. To the right of the illustration, there is a text input field for "UserEmail ID:" and a large white rectangular area labeled "Select characters /Grids:". Below this area is a text input field for "Password:". At the bottom, there are four buttons: "Login", "Reset", "Cancel", and "Register Now".

Figure K(2). Interface of Proposed System for Authentication

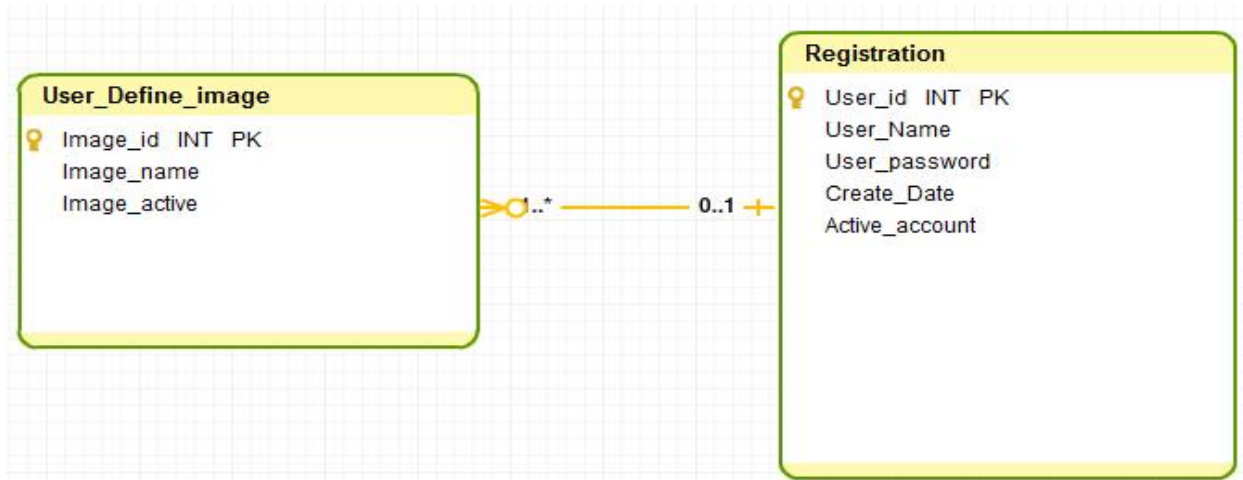


Figure K(3). Database Design of Proposed System

Appendix L. Implementation of Proposed System Registration and Authentication

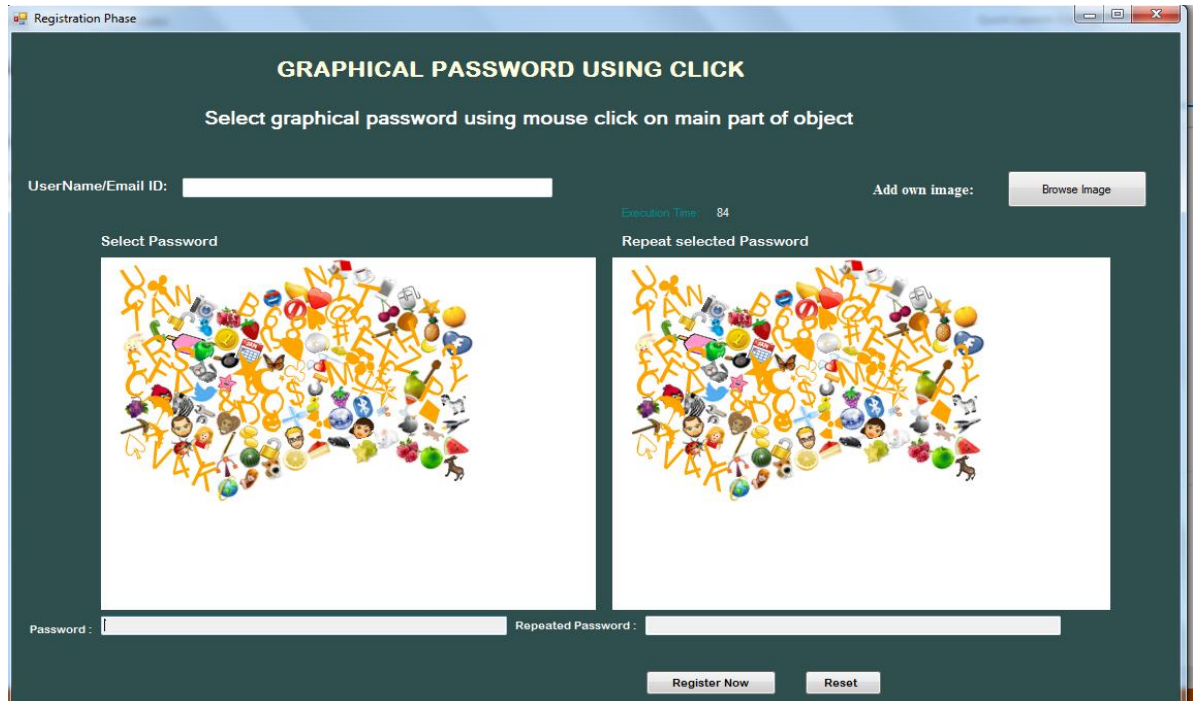


Figure L(1). Implementation of Interface of Registration of Proposed Scheme

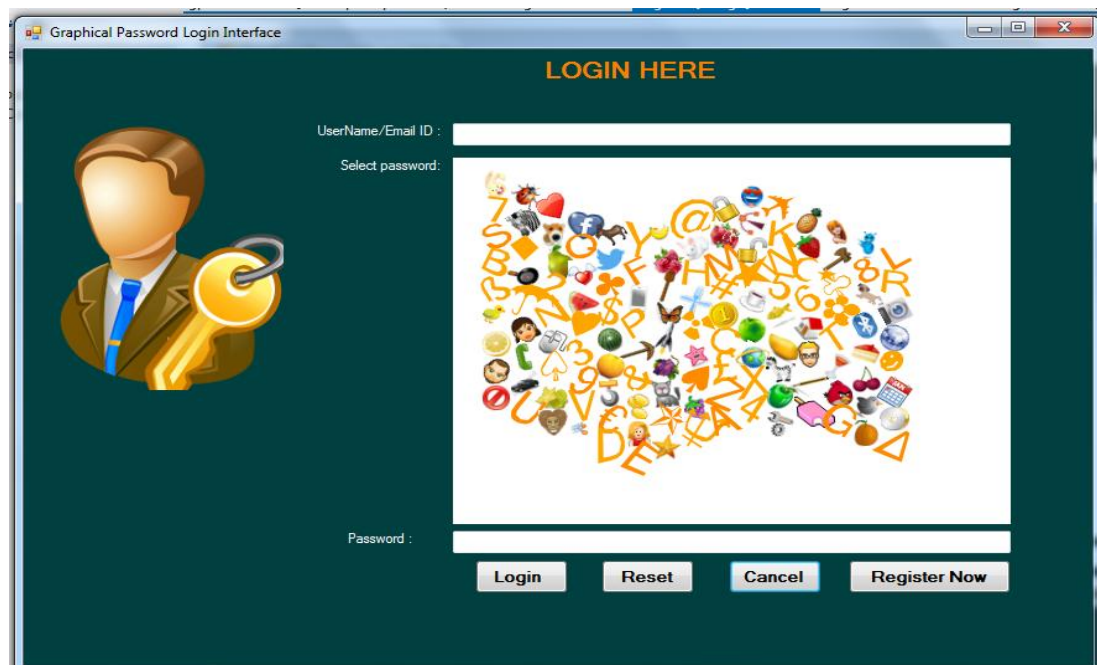


Figure L(2). Implementation of Authentication Phase of Proposed Scheme.

Appendix M. Users Collection Data Related to Login Time Against Three Schemes

Click-Text					
Sr. No	User name	First week	Second week	Third week	Authentication time (s)
1	Ibrahim ahmed	100	100	100	50.4
2	Jane Kumonova	100	100	100	46
3	Sauqir Nasir	100	100	100	46.3
4	Umar ali	100	100	100	37.7
5	Samir Khan	100	100	100	44.7
6	Zaheer	100	0	0	42.8
7	Muhammad maddiawan	100	100	100	51
8	Ahmed Ali	100	100	100	38
9	Ansar	100	100	100	32
10	allem	100	100	100	45.7
11	Suhir	100	100	100	26.5
12	Faheem	100	100	0	43.6
13	Redwan	100	50	50	36
14	Kheder Kasem	100	100	100	27.7
15	niazi	100	100	100	30.5
16	sajjad	100	100	100	33.5
17	poys	100	100	100	37.4
18	yukret	100	100	100	32.7
19	zabi	100	100	100	44
20	mehmet	100	100	100	60.4
21	khappi	100	100	100	23
22	Gohar	100	50	70	29.5
23	Jusaf	100	100	100	34.3
24	munir	100	100	100	56.4
25	John okula	100	100	100	37
26	kareem	100	100	100	25.3
27	aham	100	100	100	36.7
28	John ngr	100	100	100	33.6
29	salid	100	100	100	26.5
30	himad	100	100	100	35
31	husyn	100	100	100	38
32	salid Khan	100	100	100	33
33	alima	100	100	100	34
34	syaf	100	100	100	52.3
35	Jeff NG	100	100	100	38
36	hameed ulian	100	100	100	29
37	aslan	100	100	100	41.4
38	Bans	100	100	100	40.6
39	Jahan tk	100	100	100	38
40	rehab	100	100	100	27.4
Average		100	96	93	38.0476

Table M(1). Click-Text Login Time and Memorability Results

Proposed algorithm					
Sr. No	User name	First week	second week	Third week	Authentication time(s)
1	Ibrahim ahmed	100	100	100	43.6
2	Jane Kumonova	100	100	100	52.5
3	tauqir Nasir	100	100	100	55
4	Umar ali	100	100	100	50.7
5	aamir khan	100	100	100	48.5
6	zaheer	100	100	100	53.8
7	Muhammad makki awan	100	100	100	71.5
8	Ahmed Ali	100	100	100	45.1
9	Ansar	100	100	100	60.5
10	allem	100	100	100	49.8
11	zahir	100	100	100	51.4
12	Faheem	100	100	100	30.6
13	Redwan	100	100	100	58
14	Kheder Kasem	100	100	100	29
15	niazi	100	100	100	24.7
16	sajjad	100	100	100	33.8
17	poya	100	100	40	27.6
18	yukrat	100	100	100	37
19	zabi	100	100	100	40
20	mehmet	100	100	100	34.7
21	khapol	100	100	100	23.7
22	Gohar	100	50	100	39.6
23	jusaf	100	100	100	34.5
24	munir	100	100	100	45.6
25	john okula	100	100	100	40
26	kareem	100	100	100	42.6
27	anam	100	100	100	29.7
28	johan nag	100	100	100	36.9
29	said	100	100	100	43.2
30	hmad	100	100	100	50.3
31	husyn	100	100	100	57
32	said khan	100	100	50	50.4
33	alma	100	100	100	32.3
34	syaf	100	100	100	33
35	ieff NG	100	100	100	39
36	hameed ullah	100	100	100	49.5
37	aslan	100	100	100	44
38	Baris	100	100	100	29
39	Jahan trk	100	100	100	36
40	rehab	100	100	100	32.5
Average		100	98.75	97.25	42.165

Table M(2). Proposed Method Login Time (s) and Memorability Results

Text+Captcha					Authentication time (s)
Sr. No	User name	First week	second week	Third week	
1	Ibrahim ahmed	100	100	100	18.5
2	Jane Kumonova	100	100	100	27.9
3	taugir Nasir	100	100	100	20.8
4	Umar ali	100	100	100	22.5
5	aamir khan	100	50	100	33
6	zaheer	100	100	100	23.5
7	Muhammad makki awan	100	100	100	24.6
8	Ahmed Ali	100	100	100	42.5
9	Ansar	100	50	0	24.8
10	ailem	100	100	100	23.5
11	zuhir	100	100	100	15
12	Faheem	100	100	100	33.2
13	Redwan	100	100	100	34.8
14	Kheder Kasem	100	100	100	22.9
15	niazi	100	100	100	17
16	sajjad	100	100	0	45.4
17	poya	100	100	100	30
18	yukrat	100	50	0	32.3
19	zabi	100	100	100	27.5
20	mehmet	100	100	100	22
21	khapol	100	100	100	19
22	Gohar	100	100	100	43.5
23	iusaf	100	100	50	21
24	munir	100	100	100	25.5
25	john okula	100	100	100	38.6
26	kareem	100	100	100	30.5
27	anam	100	100	100	19.6
28	johan nag	100	100	50	25.3
29	said	100	100	100	20
30	hmad	100	100	100	23.5
31	husyn	100	50	50	19
32	said khan	100	100	100	17.7
33	alma	50	100	100	28.5
34	syaf	100	100	100	37.5
35	jeff NG	100	100	100	33.8
36	hameed ullah	100	100	100	28
37	aslan	100	100	100	33
38	Baris	100	100	100	21
39	Jahan trk	100	100	100	22.6
40	rehab	100	100	100	18
Average		98.75	95	88.75	26.6325

Table M(3). Captcha+ Text Method Login Time (s) and Memorability Results

Appendix N. Memorable Results of same Complex Passwords of Three Schemes

Captcha + Text							
Sr.	Name	First day		Second day		Third day	
		First time remembered password %	Second time remembered password %	First time remembered password %	Second time remembered password %	First time remembered password %	Second time remembered password %
1	Elif	40	60	50	50	40	50
2	Mehmmet	100	50	100	100	100	100
3	Aslan	100	100	100	100	50	50
4	Jahan trk	50	50	60	70	100	100
5	altaf	40	100	100	50	100	100
Average		71.66	76.66	85	78.33	81.66	83.33
SD		8.24					

Table N(1). Table of Memorability Results of Captcha +Text Scheme Against Constant Password

Click-Text							
Sr.	Name	First day		Second day		Third day	
		First time remembered password %	Second time remembered password %	First time remembered password %	Second time remembered password %	First time remembered password %	Second time remembered password %
1	Elif	50	100	100	60	70	100
2	Mehmmet	60	100	100	100	100	100
3	Aslan	100	60	100	70	60	50
4	Jahan trk	100	50	50	100	100	100
5	altaf	100	100	50	100	100	100
Average		82	82	80	86	86	90
SD		5.656854					

Table N(2). Table of Memorability Results of Click +Text Scheme Against Constant Password

Proposed method							
Sr.	Name	First day		Second day		Third day	
		First time remembered password %	Second time remembered password %	First time remembered password %	Second time remembered password %	First time remembered password %	Second time remembered password %
1	Elif	60	100	100	100	100	100
2	Mehmmet	100	100	100	100	100	100
3	Aslan	100	100	100	60	40	100
4	Jahan trk	60	100	100	100	100	70
5	altaf	100	100	60	100	100	100
Average		84	100	92	92	88	94
SD		7.071067812					

Table N(3). Table of Memorability Results of Proposed Method Scheme Against Constant Password