JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Generation of Authentication String from pixel rearrangement to provide security to system.

AJINKYA VINODRAO BHATKAR,RAJAT SUNIL THOKAL, ADNAN SHAHA ASHFAQUE SHAHA, VINAY SUKHDEV LANDE
ASSISTANT PROFESSOR

ABSTRACT

The traditional authentication system used in technological applications is the well- known and widely spread user/password pair. This technology as proved itself as well acceptable by the users and quite safe when used according to good security practices, this is frequent change of the password, use of letters, number and symbols in the password, not revealing the password to others, not using the same password in more than one service, etc. But this is not what really happens, so we need to improve the protocol. Graphical secrets present lots of advantages and can increase the level of security without a significant change in the user's habits. For that, we need to possess strong ways to convert them into strings that will fed the implemented passwords systems. Graphical passwords are an alternative to alphanumeric passwords in which users click on images to authenticate themselves rather than type alphanumeric strings.

The traditional text passwords are often forgotten by users. In view of this, users often write them down on sheet of paper or any other surface for memo ability. Users tend to choose short and simple passwords in place of long and complex passwords. Graphical passwords have been introduced as an alternative to text passwords. This is because humans tend to remember visuals better than text.

In this project, Admin will create the user 's account and user can then login by clicking on images with a condition that user must remember the point on which he\she clicked when account was created. Also, it provides facility to generate the report of the user account, deletion and disabling the account.

- **Keywords:**

Pixcel,graphical,password etc.

- **Project methodology**

―Generation of Authentication Strings from Graphics Key for security in cloud computing‖ deals with Authentication. This software will help in providing the more security to user's account.

Generally, in the text-based password, the password is easy to guessing the others user. The one user is easily found out the password of second user and easily login her\his account. So, there is the need to finding the more secure password and to generate the graphical password.

In project, Admin will create the user's account and user can then login by clicking on images with a condition that user must remember the point on which he\she clicked when account was created. Also, it provides facility to generate the report of the user account, deletion and disabling the account.

Graphical secrets present lots of advantages and can increase the level of security without a significant change in the user's habits. For that, we need to possess strong ways to convert them into strings that will feed the implemented passwords systems.

- **Software requirements**

- Language: C#.Net
- Professional Environment: Visual Stdio,
- Database: MS-SQL

- **Hardware requirements**

- System Type: 64-bit or 32-bit
- Processor : Intel core i5,2GHz
- Random Access Memory(RAM): 8 GB
- Storage Capacity:1TB
- Device Name: Laptop or Computer

Chapter No 1 Introduction of project:

In real life, the user faces too many problems while handling the account, such as:

- In the text-based password, the one user will easily guessing the password of other user and easily login to first user account.
- Sometime the text-based password is difficult to remember and the problem is occurring for opening the account.
- An Account hacking problem is occurring text-based password.
- Also for pressing many times incorrect password, the user account will be locked.

The purpose of developing this software is to overcome above mentioned problems. So, for solving the above problems we create the graphical password system. In this, the user clicks on images and create the password. When the user login to her/his account the condition is that the sequence on clicking the images is correct

otherwise user do not login his/her account. This password is more secure than the text-based password.

1.1 Scope

This software has many uses in various companies. It can be used to handle small scale industries as well. The feature of this software is:

- It can maintain user account records.
- It can provide the more security to user account.
- It can store the user information.
- Store how many user can login.
- Disabling the account.
- Deleting the account.
- Generate the graphical password based on images.
- Creating new user.

In this way, this software allows us to create the more secure password for user account.

1.2 Objective

―To make software fast in processing, with good user interface so that user can change it and it should be used for a long time without error and maintenance‖.

The objective of this project is to easily remembering the password but not easily guessing the password. The objectives behind preparing this software are:

1. Avoid user account hacking process.
2. Increase the security.
3. Easily disabling the account.

2.1 Introduction

Chapter No 2 Literature Survey

Literature review and project methodology is important in a system that wants to develop. In this chapter, all task and modules that will be implementing will distribute into sub modules to make easy management. A details research about all information related to this project will implement to avoid possible problem happen during development phase. Besides, early preparation can be implement to avoid system failure risk.

Need of Graphical Password:

The text-Based password flaws in the aspects of usability and security issues that bring problems to users. Hence, there is a need for alternative mechanism to overcome these problems. The difficulty in remembering the text-Based password. To overcome this problem, we need the graphical password system. Passwords that are easily remembered for example pet's name, first name and street address. Unfortunately, these passwords can be easily guessed or broken. According to an article in Computerworld, the security team at a large company tested and ran a network password cracker and surprisingly within 30 seconds, they manage to crack approximately 80% of the passwords. Pictures are generally easier to be remembered or recognized than text, especially photos, which are even easier to be remembered than random pictures.

Why Graphical Passwords?

In user description of the concept an image would appear on the screen, and the user would click on a few chosen regions of it. If the correct regions were clicked in, the user would be authenticated.

Efficiency is important in password systems because users want to have quick access to systems. The time to input a

graphical password by a highly skilled, automated user can be predicted by Fitts' Law. The law states that the time to point to a target depends on the distance and size of the target – greater distance and smaller targets lead result in slower performance.

2.2 Facts and findings:

Existing System Case Study

- **Case Study 1:**

Greg Blonder was the first to describe graphical passwords, presenting in a United States Patent a system that would allow users to choose their picture, the number of regions to be clicked, their size and position. Since then, many variations of this system were presented and images have gained their way into the authentication processes.

Among the most popular graphical authentication systems we find Passfaces from the Passfaces Corporation, a commercial system were the user chooses a previously selected face from a set of faces and repeats this process for different faces in different sets for a defined number of times, but popular doesn't imply secure and a study of the users choices demonstrated that they are, in some cases, similar for all users. For instance, 10% of the passwords of males could have been guessed with only two attempts.

- **Case Study 2: Draw-A-Secret**

Draw-A-Secret (DAS) was the first recall-based graphical password system proposed. Users draw their password on a 2D grid using a stylus or mouse (see Figure). A drawing can consist of one continuous pen stroke or preferably, several strokes separated by ―pen-ups‖ that restart the next stroke in a different cell. To log in, users repeat the same path through the grid cells. The system encodes the user-drawn password as the sequence of coordinates of the grid cells passed through in the drawing, yielding an encoded DAS password. Its length is the number of coordinate pairs summing across all strokes. There is little information on either the usability or the practical security of the original DAS system, as to date it has only been user tested through paper prototypes (but see also the related Pass-Go system, below). Nali and Thorpe asked 16 participants to draw 6 ―doodles‖ and 6 ―logos‖ on 6 _ 6 grids. These drawings were visually inspected for symmetry and number of pen strokes. They found that participants tended to draw symmetric images with few pen strokes, and to place their drawing approximately in the center of the grid. Limitations of this preliminary study included: users were not told that their drawings were ―passwords‖, users did not have to later reproduce their drawings, and data was collected on paper (rather than users drawing using a computer). No usability data (login times, success rates, etc.) was collected. The size of the theoretical password space, that is, the number of all possible passwords regardless of how small their probabilities in actual practice, is related to the coarseness ofthe underlying 2D grid, and the maximum password length. For a 5 _ 5 grid and maximum length 12, the theoretical password space has cardinality 258. This is often stated as 58 bits for brevity, but should not be miss-interpreted as 58 bits of entropy, since passwords are far from equi-probable. To allow verification, the system must store the encoded DAS passwords. To avoid storing them cleartext, a one-way function of the password, or cryptographic hash, may be stored, as is done with text passwords (see Section VIII). Note that there is a many-to-one mapping from user-drawn passwords to encoded DAS

passwords; for example, all doodles drawn entirely within one grid square are equivalent to a dot.

In summary, the DAS design does offer a theoretical space comparable with text passwords, but the possibility that users will prefer predictable passwords such as symmetric passwordswith few strokes suggests that, as with text passwords, the effective space will be considerably smaller. Without an implementation and user studies, we can tell little more. Similarly, while a key motivation for DAS was the superior memorability associated with images, the lack of suitable user studies leaves as an open question how effectively this can be leveraged in graphical authentication.

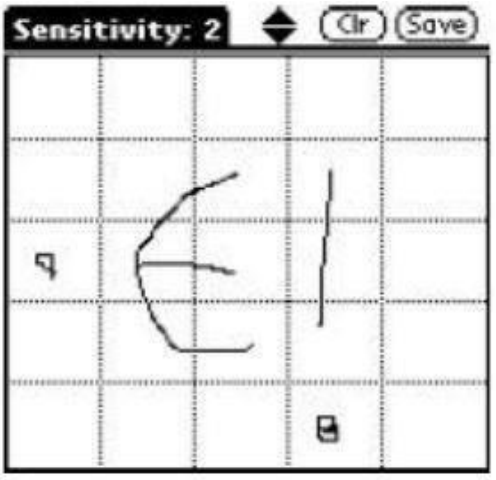


Figure.3.1:Sample Draw-A-Secret password

- **Case Study 3:**

Story (see Figure) was proposed by Davis et al. as a comparison system for PassFaces. Users first select a sequence of images for their portfolio. To log in, users are presented with one panel of images and they must identify their portfolio images from among decoys. Images in their user study contained everyday objects, places, or people. Story introduced a sequential component: users must select images in the correct order. To aid memorability, users were instructed to mentally construct a story to connect the images in their set. In the test system, a panel had 9 images and a password involved selecting a sequence of 4 images from this panel. Story was user-tested along with Faces in a field study. Davis et al. found that user choices in Story were more varied but still displayed exploitable patterns, such as differences between male and female choices. Users had more difficulty remembering Story passwords (85% success rate) and most frequently made ordering errors. Surveys with participants revealed that they were unlikely to have formulated a story as a memory aid, despite the designers' intentions; this may explain the high number of ordering errors. Different instructions or more user experience might possibly result in greater usage of a story strategy.

In D´ej`a Vu [56] (see Figure), users select and memorize a subset of ―random art‖ images from a larger sample to create their portfolio. To log in, users must recognize images belonging to their pre-defined portfolio from a set of decoy images; in the test system, a panel of 25 images is displayed, 5of which belong to the user's portfolio. Users must identify all images from their portfolio and only one panel is displayed.

Images of random art are used to make it more difficult for users to write down their password or share it with others by describing the images from their portfolio. The authors suggest that a fixed set of 10000 images suffices, but that ―attractive‖ images should be hand-selected to increase the likelihood that images

have similar probabilities of being selected by users.

- **Case Study 4: Passpoints**

The literature on cued-recall graphical password systems is dominated by PassPoints and its variations. During creation of a PassPoints password (see Figure), users are presented with an image. A password is a sequence of any n = 5 user-selected click-points (pixels) on this image. The user selects points by clicking on them using a mouse. During login, re-entry of the click-points must be in the correct order, and accurate within a system-specified tolerance. The image acts as a memory prompt of the location of the originally chosen click-points. Note that this is not an optimal cued- recall scenario: users are presented with only one cue, but must recall 5 pieces of information, in the correct order. The standard parameterization provides a theoretical password space of 243 conceivable passwords; this increases with larger n and smaller tolerance, though usability impacts are expected.

An important implementation detail is the type of discretization used — this is related to how the system determinesif entered click-points are acceptably close to the original points, and affects whether the system-side passwords stored for verification can be hashed. Robust discretization, centered discretization, and optimal discretization are possible alternatives. Kirovski et al. suggest how discretization could be implemented using Voronoi polygon tiling by analyzing image features and centering likely clickpoints within the polygons. Wiedenbeck et al. conducted three lab-based user studies of PassPoints. Users took 64 seconds to initially create a password, and required an additional 171 seconds of training time on average to memorize their password. Login took between 9 and 19 seconds on average. Login success rates varied from 55-90%, with users returning at different intervals to log in again. User performance was found to be similar on the four images tested, and it was recommended that tolerance areas around click-points be at least 14_14 pixels for acceptable usability. Chiasson et al. conducted a lab study and a large field study, finding that image choice does impact usability, that tolerance areas could be further reduced, and that memory interference from remembering multiple PassPoints passwords may be problematic. Later security analyses found it to be vulnerable to hotspots and simple patterns withiimages, as elaborated in Section VIII.

Bicakci et al. conducted a lab study where a PassPoints password was used as the master password for a web-based password manager and concluded that it was more usable than an alphanumeric master password. Their implementation used avisible grid dividing the image into discrete sections rather than any of the aforementioned discretization methods.The 5 numbered boxes (not ordinarily visible to users) illustrate the tolerance area around clickpoints.

Chapter No 3 Project Methodology

Development of this Generation of Graphical password is using System Analysis and Design methodology (SADM). The type of life cycle that will use is prototyping. This type is chosen because it has a few of benefits compare with another life cycle. One of the benefits is developer and user can avoid misunderstanding about the system because the developer can modify the system according to user requirements from time to time immediately. Other than that, system development process faster and more to prototype work. This technique also involves user at the early system development until the system finish.

There are five phase to develop this system. The phases are:

3.1 Planning

Project title is propose and all information about software and hardware also all items that suitable with the identify title.

3.2 Analysis

After the proposed title accepted, project will continue with analysis all problems that related with situation today. Besides, research for old method or already system will hold by interview session with retailers. Early overview about to be system will illustrate with data flow diagram (DFD). The purpose of the analysis phase is to build a logical model of the new system.

3.3 Design

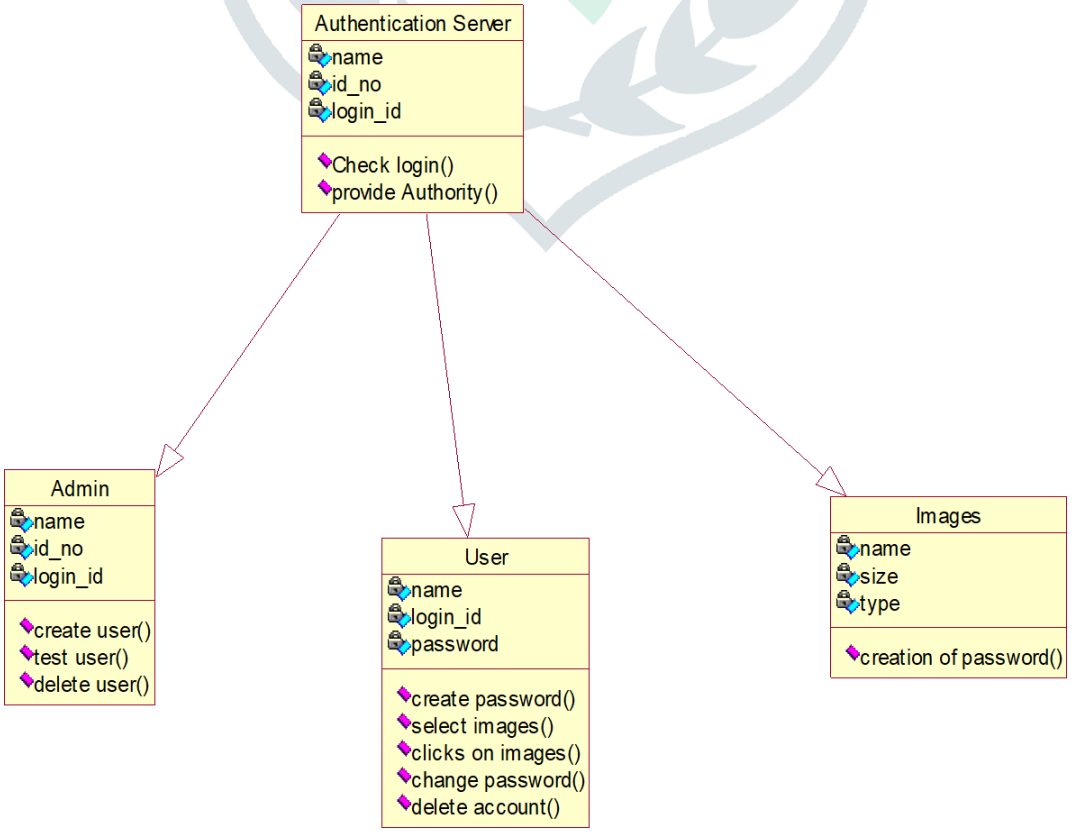
In this phase, overview of to be system will be explain more details. The purpose of this phase is to create a blueprint that will satisfy all documented requirements. Interface design and database will be design and suite based on user requirements and system logic.

3.4 Implementation

After design phase finished, system will be implement. This phase is a trial for the system and if the system does not have any problem, so the system can be declared as a successful system.

3.5 Operation and Support

This phase is implementing to avoid any problem happen from time to time. This phase also implement to support and enhances the system.



Chapter 4 Process Overview

4.1 Input Image:

Start with an image (user-provided or system-generated) that acts as a seed for authentication. This can be a profile picture, biometric image, or any predefined graphic.

4.2 Pixel Rearrangement Algorithm:

- **Step 1: Pixel Selection**: Define a selection mechanism (e.g., grid-based, random sampling, or using a hash function) to identify pixels for rearrangement.
- **Step 2: Rearrangement Rules**: Rearrange pixels based on predefined rules, such as:

Spatial Shuffling: Reorder pixels using a key or pseudorandom number generator.

Color Channel Manipulation: Alter RGB/HSV values based on a key or mathematical transformation.

Fractal Rearrangement: Rearrange pixels based on patterns derived from fractal geometries.

- **Step 3: Hashing**: Apply a cryptographic hash (e.g., SHA-256) to the rearranged pixel data to generate the authentication string.

4.3 Key Integration:

Incorporate a secret key (known only to the user/system) into the rearrangement or hashing process, ensuring that the same input image produces different authentication strings with different keys.

4.4 Output Authentication String:

The resulting hash serves as the authentication string, which can be stored securely or compared during login or verification processes.

Chapter 5 Security &Benefits

5.1 Enhanced Security & Reduced Risk of Replay Attacks:

Adds an additional layer of security by combining image-based and cryptographic methods. Authentication strings vary with each session if dynamic input images or keys are used.

5.2 Usability & Multifactor Compatibility:

Users can rely on memorable images or system-generated ones without the need to remember complex passwords.Can be paired with other authentication mechanisms (e.g., biometric or token-based) for multi-factor security.

5.3 Scalability:

Easily adaptable for use in various systems, from IoT to enterprise applications.

5.4 Uniqueness & Non-Reversibility :

The output is highly sensitive to even slight changes in the image or the key, ensuring uniqueness. The hashed authentication string cannot be reversed to recreate the original image or the key, safeguarding user data.

5.5 Dynamic Authentication & Key Dependency:

The pixel rearrangement process can be updated dynamically, preventing pattern recognition attacks.Without the secret key, it is computationally infeasible to replicate the authentication string.

Chapter No 6 Conclusion

In this project, we have conducted a comprehensive study of existing graphical password techniques. Our approach is to provide a scheme that will be able to satisfy the users need and requirements. To achieve such condition the usability and security features must be balanced. Also, in this topic we present a method to generate strong authentication strings, usable in common user/password authentication systems. Once again, we verify that the graphical secrets can be used in authentication with strong advantages to what concerns to security and without significant entropy to the users.

In future development we can also add challenge response interaction. In challenge response interactions, server will present a challenge to the client and the client need to give response according to the condition given. If the response is correct then access is granted. Also we can limit the number a user can enter the wrong password.

References

[1] Magalhães, S. T., Revett, K. and Santos, H. D.: Password Secured Sites - Stepping Forward With Keystroke Dynamics, Proceedings of the IEEE International Conference on Next GenerationWeb Services Practices, IEEE CS Press, Seoul, South Korea, 2005.

[2] Magalhães, S. T. and Santos, H. D.: An Improved Statistical Keystroke Dynamics Algorithm, Proceedings of the IADIS Virtual Multi Conference on Computer Science and Information Systems, 2005.

[3] Wiedenbeck, S., Waters, J., Birget, J.C., Brodskiy, A. and Memon, N.: Authentication using graphical passwords: Basic results, Human-Computer Interaction International (HCII 2005), Las Vegas, July 25-27, 2005

[4] Blonder, G. E.: Graphical password, U.S. Patent Number 5.559.961, 1996.

[5] The science behind PassfacesTM

[6] Davies, D., Monrose, F. and Reiter, M. K.: On User Choice in Graphical Password Schemes, 13th USENIX Security Symposium, 2004.

[7] Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K. and Rubin, A.: The Design and Analysis of Graphical Passwords, ??, 1999