Password Authentication Using Context-Sensitive Associative Memory Neural Networks: A Novel Approach

P.E.S.N. Krishna Prasad[1], B.D.C.N. Prasad[2],
A.S.N. Chakravarthy[3], and P.S. Avadhani[4]

[1] Department of CSE,
Aditya Engineering College, Kakinadas, India
[2] Dept. of Computer Applications,
P V P Siddhartha Institute of Technology, Vijayawada, India
[3] Department of CSE & IT, Sri Aditya Engineering College, Kakinada, India
[4] Dept. of CS & SE, Andhra University, Visakhapatnam, India
{1surya125,2bdcnprasad}@gmail.com,
{3asnchakravarthy,4psavadhani}@yahoo.com

Abstract. Passwords are the most widely used form of authentication. In many systems the passwords, on the host itself, are not stored as plain text but are encrypted. However, conventional cryptography based encryption methods are having their own limitations, either in terms of complexity or in terms of efficiency. The conventional verification table approach has significant drawbacks and storing passwords in password table is one of the drawbacks.

In the present paper, we propose a cognitive neural model using Context-Sensitive Associative Memory Model(CSAM) for password authentication, which is derived from cognitive domain and vector logic. According to the model, the product of two vectors is an associative memory(context-dependent) that plays critical role in the neural networks domain. In this model the output (encrypted password) is associated with the Kronecker Product of an input (key) and a context (password). The encrypted password is decoded with key and the context-dependent memory (Krnocker product) to get the original password. The proposed system provides better accuracy and quicker response time to authenticate the password but this model requires more space for holding context-dependent associative memory.

Keywords: Password Authentication, Cryptography, Associaitive neural memory, Kronecker Product, context-sensitive memory models.

1 Introduction

Security is a broad topic which covers many issues. Security is essential for data operation today. Information or commerce exchanges need security and reliability. Authentication is a critical part of any network security policy.In fact,

N. Meghanathan et al. (Eds.): CCSIT 2012, Part II, LNICST 85, pp. 454–468, 2012.

Authentication is the act of confirming the truth of an attribute of a datum or entity. This might involve confirming the identity of a person, tracing the origins of an artifact, ensuring that a product is what it's packaging and labeling claims to be, or assuring that the computer program is a trusted one. One familiar use of authentication and authorization is access control. Authentication validates the identity of a user or device. When using a mutual authentication scheme, not only the client is authenticated, but also the network itself. In remotely accessed computer systems the user authenticates/identifies himself to the system by sending a secret password.

Process of authentication can be defined as developing a unique mapping process from given secret password to some other unique information in a defined domain. The guarantee of security doesn't only depend on unique mapping but greatly depend upon difficulties associated with getting back the password from the mapped formation. Password authentication is the foremost mechanism for verifying the identity of computer users, even though it is well known that people frequently choose passwords that are vulnerable to dictionary attacks. The motivation for addressing the security and shortcomings of traditional password-based authentication is that users tend to choose passwords that are easy to remember, which in the case of textual passwords usually implies that they are easy to obtain by searching through a carefully formed dictionary" of candidate passwords[1].

A computer system that is supposed to be used only by those authorized must attempt to detect and exclude the unauthorized. Its access is, therefore, usually controlled by insisting on the authentication procedure to establish, with some degree of confidence, the identity of the user, hence granting those privileges as may be authorized to that identity[2].

Common examples of access control involving authentication include:

1. A captcha is a means of asserting that a user is a human being and not a computer program.
2. A computer program using a blind credential to authenticate another program
3. Entering a country with a passport
4. Logging in to a computer
5. Using a confirmation E-mail to verify ownership of an e-mail address
6. Using an Internet banking system
7. Withdrawing cash from an ATM.

1.1 Password

Use of strong passwords lowers overall risk of a security breach, but strong passwords do not replace the need for other effective security controls. The effectiveness of a password of a given strength is strongly determined by the design and implementation of the authentication system software, particularly how frequently password guesses can be tested by an attacker and how securely information on user passwords is stored and transmitted. Risks are also posed by several means of breaching computer security which are unrelated to password strength.

1.2 Password Authentication

The idea of password assignment is to base the authentication of an identity on something the user knows. In other words, the distinguishing characteristic is knowledge. In a security perspective it should be seen as a user-remembered key. Password should ideally be a random string of letters, numbers and other symbols, which is far from reality in most of the systems. The whole notation of passwords is based on an oxymoron. The idea is to have a random string that is easy to remember.

Drawbacks with traditional password authentication:

1. User password is difficult to memorize.
2. User cannot freely choose the password
3. User cannot change his password
4. It cannot stand with forgery attack.

Our proposed method can stand with Replay and forgery attacks. Many of the deficiencies of password authentication systems arise from the limitations of human memory. If human were not required to remember the password, a maximally secure password would be one with maximum entropy: it would consist a string as long as the system allows, with characters selected from all those allowed by the system. Some passwords are very easy to remember, but also very easy to guess with dictionary searches. In contrast, some passwords are very secure against guessing but difficult to remember[1].

The biggest problem with the above systems is that they are vulnerable to "over the shoulder" attacks where someone can simply observe the combination and ordering of images used to login to the system. This problem is not unique to graphical authentication systems as it is still possible to observe finger movement of a slow typist to discern a password in a traditional password system although the traditional password model avoids the direct over the shoulder attack by hiding the entered password behind asterixes or a similar typographical symbol or glyph which is not possible when the user needs to click on an images. The traditional password model also fails in this respect as key presses can be logged by specialized hardware or software to reconstruct the password.

An additional problem is the small number of combinations available. If you have 4 stages of 16 images each there ares 65536 combinations which are trivial for a computer to brute force if no restriction is placed on maximum number of trails. It is therefore required to protect the system further using either delays between attempts (with a maximum number of attempts) or increasing delays between attempts to discourage automated brute forcing of the combination.

Password strength is a measure of the effectiveness of a password in resisting the guessing and brute-force attacks. In its usual form, it estimates how many trials an attacker who does not have direct access to the password would need, on average, to guess it correctly. The strength of a password is a function of length, complexity, and unpredictability[3].

1.3 Alphanumeric Password (Textual)

Alphanumeric password is derived from a Character Set. There are so many types of Character sets depending upon the application where we need authentication. One of the well known Character Set is the American Standard Code for Information Interchange (ASCII). It is a character-encoding scheme based on the ordering of the English alphabet. ASCII includes definitions for 128 characters: 33 are non-printing control characters (now mostly obsolete) that affect how text and space is processed; 94 are printable characters, and the space is considered as an invisible graphic. A common attack against password authenticated system is the dictionary attack. An attacker can write a program that, imitating a legitimate user, repeatedly tries different passwords, say from a dictionary, until it gets the correct password. We present an alternative defence against dictionary attacks by using Graphical password [2].

1.4 Graphical Password

The most common computer authentication method is to use alphanumerical usernames and passwords. This method has a significant drawbacks. For example, users tend to pick passwords that can be easily guessed. On the other hand, if a password is hard to guess, then it is often hard to remember. To address this problem, we a developed authentication method that use pictures as passwords.

Usage of graphics (images) instead of alphanumerical passwords is based on two neglected facts:

1. A picture is worth a thousand words.
2. Humans remember pictures better than words.

Although, the main argument for graphical passwords is that people are better at memorizing graphical passwords than text-based passwords, the existing user studies are very limited and there is not yet convincing evidence to support this argument. Our preliminary analysis suggests that it is more difficult to break graphical passwords using the traditional attack methods such as brute force search, dictionary attack, or spyware. However, since there is not yet wide deployment of graphical password systems, the vulnerabilities of graphical passwords are still not fully understood [2].

Conversion of image to matrix (or text). By using the following procedure it can be converted any image into a matrix, consisting of a set of numbers representing all the pixels of the image. After converting image[2] into a matrix consisting of set of numbers that gives as input to the neural network and the network train with image matrix servers as a training sample.

One ought to read color of each pixel of the image, and to convert the color into red, green and blue (RGB) parts, as each color can be produced using these colors.

$$\begin{pmatrix} 135 & 206 & 235 & 154 & 85 & 25 & 69 & 158 & \ldots\ldots\ldots\ldots.196 \\ 148 & 58 & 157 & 35 & 154 & 129 & 35 & 78\ldots\ldots\ldots\ldots\ldots\ldots.254 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.45 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots.255 \\ 148 & 58 & 157 & 35 & 154 & 129 & 35 & 78\ldots\ldots\ldots\ldots\ldots\ldots.148 \end{pmatrix}$$

Fig. 1. Matrix representation of an image

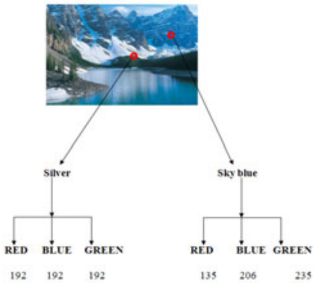


Fig. 2. Reading pixel values

2 Cognitive Domain and Vector Logic

Cognitive functions[10, 13, 14] rely on the extensive use of information stored in the brain, and the searching for the relevant information for solving some problem is a very complex task. Human cognition largely uses biological search engines, and assuming that to study cognitive function need to understand the way these brain search engines work. The approach is to study multi modular network models, able to solve particular problems that involve information searching. The building blocks of these multi modular networks are the context dependent memory models.

These models work by associating an output to the Kronecker product[3, 10] of an input and a context. Input, context and output are vectors that represent cognitive variables[3, 10–12]. The present model constitute a natural extension of the traditional linear associator, showing that coding the information in vectors that are processed through association matrices, allows for a direct contact between these memory models and some procedures that are now classical in the Information Retrieval field. One essential feature of context-dependent models is that they are based on the thematic packing of information, where by each context points to a particular set of related concepts. The thematic packing can be extended to multi modular networks involving input-output contexts, in order to accomplish more complex tasks. Contexts act as passwords that elicit the appropriate memory to deal with a query. We also show toy versions of several "neuromimetic" devices that solve cognitive tasks as diverse as decision making or word sense disambiguation. The functioning of these multi modular networks can be described as dynamical systems at the level of cognitive variables.

Vector logic is a mathematical model of logic in which the truth values are mapped on elements of a vector space. The binary logical functions are performed by rectangular matrices operating on the Kronecker product of their vectorial arguments. The binary operators acting on vectors representing ambiguous (fuzzy) truth values generate many-valued logics. "Eduardo Mijraji[10, 13]" showed that, within the formalism of vector logic, it becomes possible to obtain truth-functional definitions of the modalities "possibility" and "necessity". These definitions are based on the matrix operators that represent disjunction and conjunction respectively, and each modality emerges by means of an iterative process.

The mathematical representations of logic have opened illuminating perspectives for the understanding of the logical constructions created by the humans. These representations have also provided us with powerful technical instruments that present a wide spectrum of applications. Recently, an algebraic representation of the propositional calculus in which the truth values are mapped on the elements of a vector space has been described. In this representation, the logical computations are performed by matrix operators[6]. In particular, binary operations are executed by rectangular matrices that act over the Kronecker product of their vectorial arguments. This algebraic model of logic has been denominated "vector logic", and has been discovered for investigating a neural model.

3 Context-Sensitive Auto-associative Memories

Memory plays a major role in Artificial Neural Networks[10, 13]. Without memory, Neural Network cannot be learned itself. One of the primary concepts of memory in neural networks is Associative neural memories[25]. It accesses the memory by its contents, not by where it is stored in the neural pathways of the brain. This is very powerful; given even a poor photograph of a person we are quite good at reconstructing the persons face quite accurately. This is very different from a traditional computer where specific facts are located in specific places in computer memory. If only partial information is available about this location, the memory cannot be recalled at all.

Traditional measures of associative memory performance are its memory capacity and content-addressability. Memory capacity refers to the maximum number of associated pattern pairs that can be stored and correctly retrieved while content addressability is the ability of the network to retrieve the correct stored pattern. Obviously, the two performance measures are related to each other.

Associative neural memories[7] are concerned with associative learning and retrieval of information (vector patterns) in neural networks. These networks represent one of the most extensively analyzed classes of artificial neural networks. One of the primary functions of the brain is associative memory. It associate the faces with names, letters with sounds, or we can recognize the people even if they have sunglasses or if they are somehow elder now.

Associative memories can be implemented either by using feed forward or recurrent neural networks. Such associative neural networks are used to associate one set of vectors with another set of vectors, say input and output patterns. The aim of an associative memory is, to produce the associated output pattern whenever one of the input patterns is applied to the neural network. The input pattern may be applied to the network either as input or as initial state and the output pattern is observed at the outputs of some neurons constituting the network.

According to the way that the network handles errors at the input pattern, they are classified as interpolative and accretive memory. In the interpolative memory it is allowed to have some deviation from the desired output pattern when added some noise to the related input pattern. However, in accretive memory, it is desired the output to be exactly the same as the associated output pattern, even if the input pattern is noisy. The memory in which the associated input and output patterns differ is called hetero-associative memory, and called auto-associative memory if they are the same - another classification.

Context-sensitive auto-associative memories[5, 8] are models that allow the retrieval of different vectorial responses given the same vectorial stimulus, depending on the context presented to the memory. The contextualization is obtained by the Kronecker product between two vectorial entries to the associative memory: the key stimulus and the context. These memories are able to display a wide variety of behaviors that range from all the basic operations of the logical calculus (including fuzzy logics) to the selective extraction of features from complex vectorial patterns.

In the present contribution, we show that a context-dependent memory matrix stores a large amount of possible virtual associative memories that awaken in the presence of a context. It shows how the vectorial context allows a memory matrix to be representable in terms of its singular-value decomposition. We describe a neural interpretation of the model in which the Kronecker product is performed on the same neurons that sustain the memory.

The present investigation explores, with numerical experiments, the reliability of chains of contextualized associations. In some cases, random disconnection produces the emergence of oscillatory behaviors of the system. The current task results in that associative chains retain their performances for relatively large dimensions. Finally, it analyze the properties of some modules of context-dependent auto associative memories inserted in recursive nets: the perceptual auto organization in the presence of ambiguous inputs (e.g. the disambiguation of the Necker's cube figure), the construction of intersection filters, and the feature extraction capabilities.

A system of networks consisting of first net which constructs the Kronecker product between two vectors and then sends it to the second net that sustains a correlation memory, defines a context-sensitive associative memory. In the real nervous system of higher mammals, the anatomy of the neural connections surely exhibits a considerable amount of local imprecision superimposed on a regular global layout. In order to evaluate the potentialities of the multiplicative devices to constitute plausible biological models, we analyze the performances of a context-sensitive memory when the multiplicative net, responsible for the construction of the Kronecker product, presents an incomplete connectivity. The investigation shows that a large dimensional system is able to support a considerable amount of incompleteness in the connectivity without a great deterioration of the memory. It establishs a scaling relationship between the degree of incompleteness, the capacity of the memory, and the tolerance threshold to imperfections in the output. And then it analyzes some performances that show the versatility of this kind of network to represent a variety of functions. These functions include a context-modulated novelty filter, a network that computes logical modalities and an adaptive searching device.

3.1 Kronecker Product and the Vector Operator

Let A be an $m \times n$ matrix and B an $p \times q$ matrix. The $m.n \times p.q$ matrix is called the Kronecker product[30] of A and B.

The resultant matrix is represented as

$$A \bigotimes B = \begin{pmatrix} a_{1,1}B & a_{1,2}B & \dots & a_{1,n}B \\ a_{2,1}B & a_{1,2}B & \dots & a_{1,n}B \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & a_{n,p}B \\ a_{m,1}B & a_{m,2}B & \dots & a_{m,n}B \end{pmatrix}$$

Some Properties of Kronecker Product: It is also called the tensor product. Some properties of the Kronecker product[30] as:

- for a scalar a, $a \bigotimes A = A \bigotimes a = a.A$,
- for scalars a and b, $aA \bigotimes bB = abA \bigotimes B$,
- for conforming matrices, $(A \bigotimes B)(C \bigotimes D) = AC \bigotimes BD$,
- $(A \bigotimes B)^T = A^T \bigotimes B^T, (A \bigotimes B)^H = A^H \bigotimes B^H$,
- for vectors a and b, $a^T \bigotimes b = ba^T = b \bigotimes a^T$,
  (Note: $a.a^T = a \bigotimes a^T$),
- for square nonsingular matrices A and B:
  $(A \bigotimes B)^{-1} = A^{-1} \bigotimes B^{-1}$
- $tr(A \bigotimes B) = tr(A).tr(B)$,
- $rank(A \bigotimes B) = rank(A).rank(B)$.

In addition to these properties, some more properties can be attributed to kronecker product, the present task needs these properties which are included in the list mentioned.

3.2 Numerical Example for Encoding and Decoding

A simple example is illustrated with a mathematical model, how the encoding and decoding mechanism applied on textual and graphical password for encoding and decoding approaches.

Let A and B be the two matrices of sizes 3 × 3 and 2 × 2, given as

$$A = \begin{pmatrix} 1\,1\,0 \\ 0\,1\,1 \\ 1\,0\,1 \end{pmatrix}_{3\times3} \quad \& \; B = \begin{pmatrix} 1\,0 \\ 1\,1 \end{pmatrix}_{2\times2}$$

Encoding Matrix as Memory matrix. The resultant of matrix is represented as a memory matrix M, as encoded matrix of $A \bigotimes B$:

$$E(M) = A \bigotimes B = \begin{pmatrix} 1\,0\,1\,0\,0\,0 \\ 1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,0\,1\,0 \\ 0\,0\,1\,1\,1\,1 \\ 1\,0\,0\,0\,1\,0 \\ 1\,1\,0\,0\,1\,1 \end{pmatrix}$$

Decoding Matrix as original input A. Let the matrix B is supplied as one of the input to decoding matrix and the second one as the encoded matrix M and then to get the resultant matrix as original matrix A, on needs yo follow as:

$$M_1 = E(M).B_1 \text{ where } B_1 \text{ is } 1^{st} \text{ column vector}$$

$$M_1 = \begin{pmatrix} 1\,0\,1\,0\,0\,0 \\ 1\,1\,1\,1\,0\,0 \\ 0\,0\,1\,0\,1\,0 \\ 0\,0\,1\,1\,1\,1 \\ 1\,0\,0\,0\,1\,0 \\ 1\,1\,0\,0\,1\,1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

$$M_1 = \begin{pmatrix} 1\,1\,0 \\ 2\,2\,0 \\ 0\,1\,1 \\ 0\,2\,2 \\ 1\,0\,1 \\ 2\,0\,2 \end{pmatrix}$$

$$M_1^T = \begin{pmatrix} 1\,2\,0\,0\,1\,2 \\ 1\,2\,1\,2\,0\,0 \\ 0\,0\,1\,2\,1\,2 \end{pmatrix}$$

$$M_2 = M_1^T.B_2$$

(Note: 2 denotes second column vector of B)

$$M_2 = M_1^T. \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 2\,0\,2 \\ 2\,2\,0 \\ 0\,2\,2 \end{pmatrix}$$

$$\text{Finally, } A = \lambda.M_2^T = \begin{pmatrix} 1\,1\,0 \\ 0\,1\,1 \\ 1\,0\,1 \end{pmatrix}$$

where $\lambda = 0.5$, here A is a Password matrix and B is a Key matrix.

4 Algorithm for Encoding and Decoding the Password

4.1 Password Encoding

The procedure for encoding a password P with the given key K is as follows:

A context-dependent associative memory M acting as encoding model is a matrix

$$M = \sum_{i=1}^{m} P_i [P_i \bigotimes \sum_{j(i)} K_j]^T \tag{1}$$

where P_i are column vectors password (the set P is chosen orthonormal), and $K_{j(i)}$ are column vectors mapping to Key accompanying the i^{th} password (also an orthonormal set).

By feeding the context-sensitive associative module M with Key K, the system retrieves the possible password associated with the key.

At resting conditions the system is grounded in an indifferent state g. The mathematics of the model implies the priming of the memory with a linear combination which has an equal weight

$$M(g \bigotimes I_{m\times n}) = \sum_i < P_i, g > P_i (\sum_{j(i)} K_j^T) = \sum_i P_i (\sum_{j(i)} K_j)^T \tag{2}$$

where $g = \sum_i P_i$ and I is the $n \times n$ identity matrix. From (2) it is evident that, after the priming, the context-dependent memory becomes a classical memory associated password with the specified key.

4.2 Password Decoding

The procedure for decoding matrix is as follows:

Step 1: Initial Memory matrix:

$$M_0 = E(M) \tag{3}$$

Step 2: For each column vector of K_j, where j^{th} column
do

$$M_j = M_{j-1}^T.K_i \tag{4}$$

```
Step 3:
                    $M_j = M_j^T$                    (5)
    repeat the steps 2 and 3 until for all $j$ columns of $K$.
Step 4: if $P == \lambda.M_j^T$ then
    the decoded matrix is the original matrix $P$
      else
    the decoded matrix is not an original matrix $P$
      where $\lambda$ is a learning constant.
```

The structural representation of Context-sensitive associative memory model (CSAM) for password authentication is represented in Figure 3, Figure 3(a) represents encoding process and Figure 3(b) represents decoding process

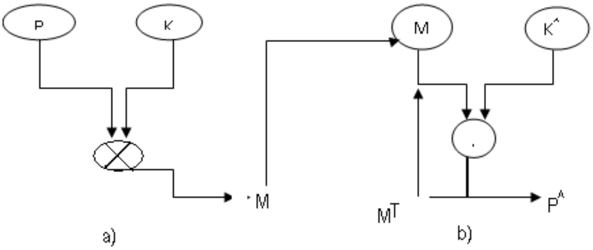
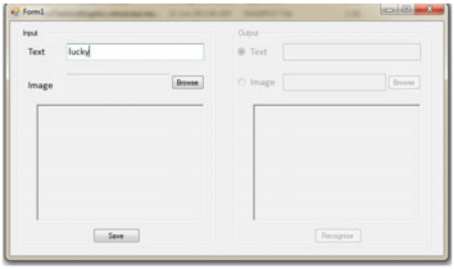


Fig. 3. Password authentication using CSAM

5 Result Analysis

Authentication process can be done through both normal text as one input and image as another input. Using both input the chosen model encoding into memory model and then decoding the memory model through either normal text password as one input to get the image or chosen image as another input to get the normal password. The process is presented below:

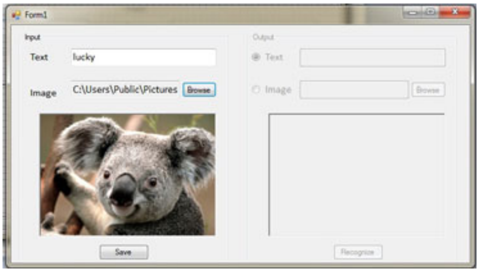
1. Text password is given by the user. Any password using a particular character set can be used. Suppose to enhance the security then the given text password can be normalized using any available normalization process. The input of textual password is shown in Figure 4
2. Now an image can be given as input by choosing any image from the available source or from data repository. Then the result process is shown in Figure 5
3. When the text and image are given as input to the system, then system immediately encodes these two passwords by applying encoding technique of the chosen model and then the result is stored onto the database in the form of encoded representation. The process is shown in Figure 6



Text: lucky

● Text

Fig. 4. Text as one input

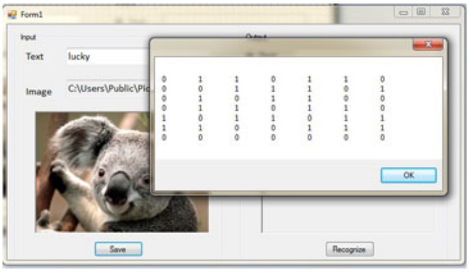


Text: lucky

Image: C:\Users\Public\Pictures

● Text

Fig. 5. Image as another input



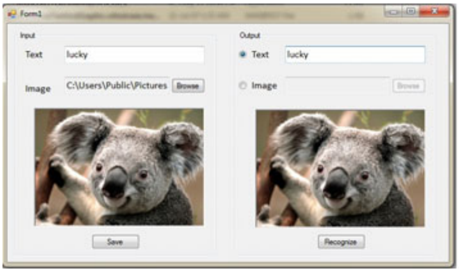
Text: lucky

Image: C:\Users\Public\Pic

```
0    1    1    0    1    1    0
0    0    1    1    1    0    1
0    1    0    1    1    0    0
0    1    1    0    1    1    0
1    0    1    1    0    1    1
1    1    0    0    1    1    1
0    0    0    0    0    0    0
```

Fig. 6. Encryption

4. In the decoding process, choose text as one input and the encoded data as another input. By applying decoding mechanism of the associative memory model to get the chosen image as output to the environment. The result status is shown in Figure 7
5. In the decoding process choose image as one input and the encoded data as another input. By applying decoding mechanism of the associative memory model to get the original text password as output to the environment. The result status is shown in Figure 8



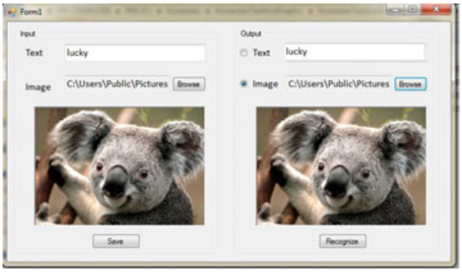
Text lucky

Image C:\Users\Public\Pictures

● Text lucky

○ Image

Fig. 7. Decryption through text



Text lucky

Image C:\Users\Public\Pictures

○ Text lucky

● Image C:\Users\Public\Pictures

Fig. 8. Decryption through image

6 Conclusion

Memory is not just a passive store for holding ideas without changing them; it may transform those ideas when they are being retrieved. There are many examples showing that what is retrieved is different from what was initially stored. Simple Associative memories are static and contain very low memory so that they cannot be applied in the applications where high memory is required. A simple model describing context-dependent associative memories generates a good vectorial representation of basic logical calculus. One of the powers of this vectorial representation is the very natural way in which binary matrix operators are capable to compute ambiguous situations. This fact presents a biological interest because of the very natural way in which the human mind is able to take decisions in the presence of uncertainties. Also these memories could be used to develop expert agents to the recent problem domain.

Converting text password to binary form and converting image to binary form is a big issue here in this method. If we choose images which are having more pixel density the encoding and decoding processes will take more time. To overcome this problem less density images can be used or changing image from one format to other format will also reduce the image density value.

Context-sensitive associative neural memory model is more powerful and more secure model for authentication both textual passwords and image passwords.

For normal case of textual passwords,either private key, public key using some available mechanisms or any user defined key is used as another input to this model to encrypt and decrypt into original one.

Using the present model,some applications like banking, email, mobile based devices, to provide better and better security than the currently available mechanisms like DES and AES algorithms.

References

1. Doja, M.N., Kumar, N.: User Authentication Schemes For Mobile And Handheld Devices. INFOCOMP - Journal of Computer Science 7(4) (December 2008)
2. Krishna Prasad, P.E.S.N., Chakravarthy, A.S.N., Avadhani, P.S.: A Probabilistic Approach For Authenticating Text Or Graphical Passwords Using Back Propagation. IJCSNS International Journal of Computer Science and Network Security 11(5) (May 2011)
3. Mizraji, E., Pomi, A., Valle-Lisboa, J.C.: Dynamic searching in the brain. Cogn. Neurodyn. 3, 401–414 (2009)
4. Cyber Security Tip ST04-002. Choosing and Protecting Passwords. US CERT (retrieved June 20, 2009)
5. Pomi, A., Olivera, F., Mizraji, E.: Context-sensitive auto-associative memories as expert systems in medical diagnosis. BMC Medical Informatics & Decision Making (2006)
6. Hugues, G.E., Cresswell, M.J.: An Introduction to Modal Logic. Methuen, London (1972)
7. Prasad, B.D.C.N., Krishna Prasad, P.E.S.N.: A Study on Associative Neural Memories. (IJACSA) International Journal of Advanced Computer Science and Applications 1(6) (December 2010)
8. Prasad, B.D.C.N., Krishna Prasad, P.E.S.N., Sagar, Y.: A Comparative Study of Machine Learning Algorithms as Expert Systems in Medical Diagnosis (Asthma). In: Meghanathan, N., Kaushik, B.K., Nagamalai, D. (eds.) CCSIT 2011, Part I. CCIS, vol. 131, pp. 570–576. Springer, Heidelberg (2011)
9. Horng, G.: Password authentication without using password table. Inform. Processing Lett. 55, 247–250 (1995)
10. Mizraji, E.: Modalities in Vector Logic. Notre Dame Journal of Formal Logic 35(2) (Spring 1994)
11. Mizraji, E.: Reasoning with associative memories. Biological Complexity (1997)
12. Udi, M.: A simple scheme to make passwords based on one-way function much harder to crack. Computer Security 15(2), 171–176 (1996)
13. Pomi, A., Eduardo, M.: Semantic Graphs & Associative memories. Physical Review (2004)
14. Pomi, A., Eduardo, M., Alvarez, F.: Multiplicative contexts in associative memories. Biosystems 32, 145–161 (1994)
15. Eduardo, M.: Context-dependent associations in Linear distributed Memories. Bulletin Math., Biol., 195–205 (1989)
16. Pomi, A., Eduardo, M., Valle-Lishoa, C.J.: Dynamic searching in the Brain. Cognitive Neurodynamics, 401–414 (2009)
17. Mello, S.D., Franklin, S., Ramamurthy, U., Buars, B.: A cognitive Science based Machine Learning Architecture. AAAI (2006)

18. Mizraji, E.: Neural Memories and Search Engines. International Journal of General Systems 37(6), 715–732 (2008)
19. Mizraji, E., Lin, J.: A dynamical approach to logical decisions. Complexity 2, 56–63 (1997)
20. Mizraji, E., Lin, J.: Fuzzy decisions in modular neural networks. Int. J. Bifurcation and Chaos 11, 155–167 (2001)
21. Pomi, A., Mizraji, E.: A cognitive architecture that solves a problem stated by Minsky. IEEE on Systems, Man and Cybernetics B (Cybernetics) 31, 729–734 (2001)
22. Biddle, R., Chiasson, S., van Oorschot, P.C.: Graphical Passwords: Learning from the First Twelve Years. Technical Report TR-11-01, School of Computer Science, Carleton University (January 4, 2011)
23. Ku, W.-C.: Weaknesses and Drawbacks of a Password Authentication Scheme Using Neural Networks for Multiserver Architecture. IEEE Transactions on Neural Networks 16(4) (July 2005)
24. Obaidat, M.S., Macchiarolo, D.T.: An on-line neural-network system for computer access security. IEEE Trans. Ind. Electron. 40, 235–242 (1993)
25. A multilayer neural-network system for computer access security. IEEE Trans. Syst., Man, Cybern. 24, 806–813 (1994)
26. Schmidt, T., Rahnama, H., Sadeghian, A.: A Review of Applications of Artificial Neural Networks in Cryptosystems. In: Seventh International Symposium on Neural Networks, Shanghai, China, June 6-9 (2010)
27. Agarwal, G., Shukla, R.S.: Security Analysis of Graphical Passwords over the Alphanumeric Passwords. Int. J. Pure Appl. Sci. Technol. (2010)
28. Suo, X., Zhu, Y., Scott Owen, G.: Graphical Passwords: A Survey. In: Proceedings of 21st Annual Computer Security Applications Conference, Tucson, Arizona, December 5-9 (2005)
29. Pomi-Brea, A., Mizraji, E.: Memories in context. BioSystems 50, 173–188 (1999)
30. van Loan, C.F.: The ubiquitous Kronecker Product. JCAM (Elsevier), 85–100 (1999)