Visual Cryptography for Authentication Using CAPTCHA

A. Vinodhini¹ and L. Jani Anbarasi²

¹Lecturer, GKM College of Engineering, Tambaram, Chennai
E-mail:vino.aug@gmail.com
²Assistant Professor, Jerusalem College of Engineering, Pallikkaranai, Chennai-600100
E-mail:janitherbert79@gmail.com

Abstract

Authenticity of the user is the major issue in today’s internet applications such as core banking. Password has been the most used authentication mechanism which is subjected to offline and online dictionary attacks. Today hacking of the databases on the internet, is unavoidable. It is difficult to trust the information on the internet. To solve this problem this paper proposes a CAPTCHA based Visual Cryptography scheme to address the authentication issues. This methodology generates a unique CAPTCHA image for users which in turn is divided into two shares. One share is stored in the bank database and the other share is provided to the customer. Hash code is generated for the customer share and it is stored in the bank database. The customer has to present the share during all of his/her transactions. When the customer presents his share the hash code is generated and compared with the database value. If it matches the shares are stacked to get the original CAPTCHA image which authenticates the user.

Introduction

Trusting information on the internet is quite difficult due to hacking of database on the internet. So it is nearly impossible to be sure whether a computer that is connected to the internet can be trustworthy and secure or not. In a core banking system there is a possibility of encountering forged signature for transaction. In the net banking system, the password of the customer can be hacked and misused. This paper proposes a technique to improve security and to prevent hacking of database. An idea based on image processing and Visual Cryptography is used. Visual Cryptography introduced by Naor and Shamir[1] is a method used for encrypting a secret image into
shares, such that stacking the shares reveals the secret image. The main advantage of Visual Cryptography is the decryption of the message which does not involve more process. The decryption time is very less when Visual Cryptography technique is used. Visual Cryptography is used to check a person for his/her authentication.

Visual Cryptography provides a very powerful technique by which one secret can be distributed in two or more shares. When the shares on transparencies are superimposed exactly together the original secret can be discovered without computer participation.

Completely Automated Public Turing test to tell Computers and Human Apart (CAPTCHA) are systems which are used to tell human and machine apart automatically. It is a type of challenge response test used in computing to ensure that the response is not generated by a computer. A common type of CAPTCHA requires that the user type letters/digits from a distorted image that appears on the screen.

A hash function is any well-defined procedure or mathematical function that converts a large, possibly variable-sized amount of data into a small datum, usually a single integer that may serve as an index to an array. The values returned by a hash function are called hash values, hash codes, hash sums, or simply hashes.

In cryptography, MD5 (Message-Digest algorithm 5) is a widely used cryptographic hash function with a 128-bit hash value. As an Internet standard (RFC 1321), MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files. However, it has been shown that MD5 is not collision resistant as such, MD5 is not suitable for applications like SSL certificates or digital signatures that rely on this property. An MD5 hash is typically expressed as a 32-digit hexadecimal number.

Related Work
This section provides a brief description of Visual Cryptography and its applications. Although introduced and studied in the late 1970’s and early 1979’s Visual Cryptography have become increasingly popular in the last several years. Visual Cryptography schemes were independently introduced by Shamir[2] and blakley[3].

In Visual Cryptography each pixel appears in n modified shares. The shares are a collection of m black and white sub pixels arranged closely together. This is described as an n x m Boolean matrix. When the shares are superimposed and the sub pixels are correctly aligned the original image is obtained. Since the individual shares gives no idea of whether a specific pixel is black or white it become impossible to decrypt the shares, no matter how much computational power is available.

Visual Cryptography proposed by Chetana Hedge et al.,[4] can be applied to avoid hacking of database on the internet. This paper gives details about image processing and Visual Cryptography. They proposed an algorithm of secure authentication for banking applications where the user signature is converted to image which is used for generating the shares and these shares are stacked to get the original signature image. Visual Cryptography scheme introduced by Naor and Shamir, explains about secret sharing. Secret sharing is an algorithm in cryptography where a secret is
divided into n parts, giving each participants a unique part, where some of the parts or all of them are needed to reconstruct the secret.

A segment based Visual cryptography proposed by Borchert [5] is used to encrypt messages consisting of symbols especially bank account numbers and symbols which can be represented by a segment display. This segment display is easier to adjust the secret images and the symbols are potentially easier to recognize for the human eye.

Visual Cryptography scheme proposed by W-Q Yan et al., [6] can be applied only for printed text or image. The shares of Visual Cryptography are printed on transparencies which needs to be superimposed. A Visual Cryptography method proposed by T. Monoth et al.,[7] uses random basis column pixel expansion technique. The encoded shares are further encoded into number of sub shares recursively which is computationally complex. Similarly a technique proposed by H.J. Kim et al., [8] explains an algorithm for secret sharing scheme that allows a group of participants to share a secret among them. In this paper we propose a Visual Cryptography scheme based on CAPTCHA image. Shares are created for the CAPTCHA image generated using the pin number. (2,2) Visual Cryptography is used to create the shares. One share is in the server database and the other is with the user. When shares are superimposed, original CAPTCHA image is recovered which authenticates the user. Hash code is generated to check the integrity of the share since the share is in the network.

**Architecture and Modelling**

Visual cryptography is a cryptographic technique which allows visual information (pictures, text, etc.) to be encrypted in such a way that the decryption can be performed by humans (without computers).

It involves breaking up the image into n shares so that only someone with all n shares could decrypt the image by overlaying each of the shares over each other. In this technique n-1 shares reveals no information about the original image. We can achieve this by using one of following access structure schemes [8].

1: (2, 2) – Threshold VCS: This is a simplest threshold scheme that takes a secret image and encrypts it into two different shares that reveal the secret image when they are overlaid. No additional information is required to create this kind of access structure.

2: (2, n) – Threshold VCS: This scheme encrypts the secret image into n shares such that when any two (or more) of the shares are overlaid the secret image is revealed.

3: (n, n) – Threshold VCS: This scheme encrypts the secret image into n shares such that only when all n of the shares are combined the secret image will be revealed.

4: (k, n) – Threshold VCS: This scheme encrypts the secret image into n shares such that when any group of at least k shares are overlaid the secret image will be revealed.

Basic visual cryptography is based on breaking of pixels into some sub pixels or we can say expansion of pixels. There are two approaches for (2, 2) –Threshold VCS.
In this first approach shows that each pixel is broken into two sub pixels. Let B shows black pixel and T shows Transparent (White) pixel. Each share will be taken into different transparencies. When we place both transparencies on top of each other we get following combinations, for black pixel BT+TB=BB or TB+BT=BB and for white pixel BT+BT=BT or TB+TB=TB. Similarly second approach is given where each pixel is broken into four sub pixels. We can achieve 4C2 =6 different cases for this approach.[8].

The model for creating shares[9] is explained here.

Let \( P = \{1, \ldots, n\} \) be a set of elements called participants, and let \( 2^P \) denote the set of all subsets of \( P \). Let \( \Gamma_{\text{Qual}} \subseteq 2^P \) and \( \Gamma_{\text{Forb}} \subseteq 2^P \), where \( \Gamma_{\text{Qual}} \cap \Gamma_{\text{Forb}} = \emptyset \). The members of \( \Gamma_{\text{Qual}} \) are qualified sets and members of \( \Gamma_{\text{Forb}} \) are forbidden sets. The pair \((\Gamma_{\text{Qual}}, \Gamma_{\text{Forb}})\) is called the access structure of the scheme.

Let \( \Gamma_0 \) consists of all the minimal qualified sets:

\[ \Gamma_0 = \{ A \in \Gamma_{\text{Qual}} : A' \in \Gamma_{\text{Qual}} \text{ for all } A' \subseteq A \}. \]

A participant \( P \in P \) is an essential participant if there exists a set \( X \subseteq P \) such that \( X \cup \{P\} \in \Gamma_{\text{Qual}} \) but \( X \in \Gamma_{\text{Qual}} \). If a participant \( P \) is not essential then we can construct a visual cryptography scheme giving him a share completely white or even nothing as his share. In fact, a non-essential participant does not need to participate actively in the reconstruction of the image since the information he has is not needed by any set in \( P \) in order to recover the shared image. In any VCS having non-essential participants, these participants do not require any information in their shares. We assume all participants as essential.

In the case where \( \Gamma_{\text{Qual}} \) is monotone increasing, \( \Gamma_{\text{Forb}} \) is monotone decreasing, and \( \Gamma_{\text{Qual}} \cup \Gamma_{\text{Forb}} = 2^P \), the access structure is said to be strong, and \( \Gamma_0 \) is termed a basis. In a strong access structure,

\[ \Gamma_{\text{Qual}} = \{ C \subseteq P : \exists B \subseteq C \text{ for some } B \in \Gamma_0 \}, \]

and we say that \( \Gamma_{\text{Qual}} \) is the closure of \( \Gamma_0 \).

We assume that the message consists of black and white pixels. Each pixel appears in \( n \) versions called shares, one for each transparency. Each share is a collection of \( m \) black and white subpixels. The resulting structure can be described by an \( n \times m \) Boolean matrix \( S = \{s_{ij}\} \) where \( s_{ij} = 1 \) iff the \( j \) th subpixel in the \( i \) th transparency is black. Therefore the grey level of the combined share, obtained by stacking the transparencies \( i_1, \ldots, i_s \), is proportional to the Hamming weight \( w(V) \) of the \( m \)-vector \( V = \text{OR}(r_{i1}, \ldots, r_{is}) \) where \( r_{i1}, \ldots, r_{is} \) are the rows of \( S \) associated with the transparencies we stack. This grey level is interpreted by the visual system of the users as black or as white in according with some rule of contrast.

Definition : Let \((\Gamma_{\text{Qual}}, \Gamma_{\text{Forb}})\) be an access structure on a set of \( n \) participants. Two collections (multisets) of \( n \times m \) Boolean matrices \( b_0 \) and \( b_1 \) constitute a visual cryptography scheme \((\Gamma_{\text{Qual}}, \Gamma_{\text{Forb}}, m)\)-VCS if there exist the value \( \alpha(m) \) and the set \( \{ (X, t_X) \} \) satisfying:

1. Any (qualified) set \( X = \{i_1, \ldots, i_s\} \subseteq \Gamma_{\text{Qual}} \) can recover the shared image by stacking their transparencies. Formally, for any \( M \in b_0 \), the “or” \( V \) of rows \( i_1, \ldots, i_s \) satisfies \( w(V) \leq t_X - \alpha(m) \cdot m \); whereas, for any \( M \in b_1 \) it results that \( w(V) \geq t_X \).
2. Any (forbidden) set \( X = \{i_1, i_2, \ldots, i_p\} \in \Gamma_{Forb} \) has no information on the shared image. Formally, the two collections of \( p \times m \) matrices \( D_t \), with \( t \in \{0, 1\} \), obtained by restricting each \( n \times m \) matrix in \( b_t \) to rows \( i_1, i_2, \ldots, i_p \) are indistinguishable in the sense that they contain the same matrices with the same frequencies.

The first property is related to the contrast of the image. It states that when a qualified set of users stack their transparencies they can correctly recover the shared image. The value \( \alpha(m) \) is called relative difference, the number \( \alpha(m) \cdot m \) is referred to as the contrast of the image, the set \( \{(X, tX)\}_{X \in \Gamma_{Qual}} \) is called the set of thresholds, and \( t_X \) is the threshold associated with \( X \in \Gamma_{Qual} \). We want the contrast to be as large as possible and at least one, that is, \( \alpha(m) \geq 1/m \). The second property is called security, since it implies that, even by inspecting all their shares, a forbidden set of participants cannot gain any information useful in deciding whether the shared pixel was white or black.

**out of 2 Scheme : (2 subpixels)**

In black and white image each pixel is divided into two sub-pixels. Randomly pixels are chosen between black and white. If white, then randomly choose one of the two rows for white.

<table>
<thead>
<tr>
<th>Pixel</th>
<th>Share 1</th>
<th>Share 2</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( p = \frac{1}{2} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( 1 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td></td>
<td>( p = \frac{1}{2} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

If black, then randomly choose between one of the two rows for black.

<table>
<thead>
<tr>
<th>Pixel</th>
<th>Share 1</th>
<th>Share 2</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( p = \frac{1}{2} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( 0 )</td>
<td>( 1 )</td>
</tr>
<tr>
<td></td>
<td>( p = \frac{1}{2} )</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>( 1 )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

**Our Work**

Overview of this paper consists of three processes.
- Share creation process.
- Hash code generation
- Authentication process.

**Share Creation Process**

Overview of share creation process is shown in Figure 4.1. User registers to the server by providing their credentials such as Name, Date of birth, Occupation, Address and
Pin number which will be stored in the server database. The secret pin number is the vital attribute that the user should provide. The secret pin number will be the basis for the CAPTCHA image generation. Server generates a unique CAPTCHA image using the customer pin number. The generated CAPTCHA image is divided into two shares. One share is stored in the server database and the other is given to the customer.

Client Server

A. Captcha Image Generation Process
CAPTCHA is a cryptographic protocol whose underlying hardness assumption is based on an AI problem. It will be more secure to add noisy backgrounds, colors and increasing the level of distortion against character recognition programs. It is difficult for them to read low-quality text and the manuscripts. The method used for generating CAPTCHA[10] splits the image into several parts in random width and height values. Additionally, it rotates the split characters in random rotation angles that yield a particular distortion in the image. It is very difficult for OCR softwares to find out where characters are split and the end points of each image because of the random rotation. The pin number is taken as the input for generating CAPTCHA which is then used for creating shares.
Creation Of Shares

The basic assumption here is that the image is a collection of black and white pixels and each pixel is handled separately. Each original pixel appears in $n$ modified versions, one for each share. Each share is a collection of $m$ black and white sub-pixels that are printed in close proximity to each other so that the human visual system averages their individual black/white contributions. The resulting structure can be described by an $nxm$ Boolean matrix $S = [s_{ij}]$ where $s_{ij} = 1$ iff the $j^{th}$ sub-pixel in the $i^{th}$ share is black. $s_{ij} = 0$ iff the $j^{th}$ sub-pixel in the $i^{th}$ share is white. Figure 4.3 shows the shares of an image.

![Image of two shares](image.png)

**Figure 4.3**: Two shares obtained for a image in 2 out of 2 Scheme.

The algorithm for creation of shares is the CAPTCHA image is converted to its Boolean matrix values. This matrix shows the black pixel and white pixel of the CAPTCHA image. Initialize the $S_0$ matrix value and $S_1$ matrix values in which $S_0$ shows white pixel and $S_1$ shows black pixel.

\[
S_0 = \begin{bmatrix}
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\]

\[
S_1 = \begin{bmatrix}
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}
\]

If the pixel value of the image is white then, consider the first row of $S_0$ value to share1 and second row of $S_0$ to share2. Continue the process for all pixels of the CAPTCHA image. Each pixel is represented using four sub pixels which represents two shares. Thus the (2,2) shares are generated.

Hash Code Generation

MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit little endian integers); the message is padded so that its length is divisible by 512. The padding works as follows: first a single bit, 1, is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits fewer than a multiple of 512. The remaining bits are filled up with a 64-bit integer representing the length of the original message, in bits.

The main MD5 algorithm operates on a 128-bit state, divided into four 32-bit words, denoted $A$, $B$, $C$ and $D$. These are initialized to certain fixed constants. The main algorithm then operates on each 512-bit message block in turn, each block modifying the state. The processing of a message block consists of four similar stages,
A. Vinodhini et al

termed *rounds*; each round is composed of 16 similar operations based on a non-linear function $F$, modular addition, and left rotation. There are four possible functions $F$.

\[
F(X, Y, Z) = (X \land Y) \lor (\neg X \land Z)
\]

\[
G(X, Y, Z) = (X \land Z) \lor (Y \land \neg Z)
\]

\[
H(X, Y, Z) = X \Theta Y \Theta Z
\]

\[
I(X, Y, Z) = Y \Theta (X \lor \neg Z)
\]

$\Theta, \land, \lor, \neg$ denote the XOR, AND, OR and NOT operations respectively. For each share hash code is generated and stored in the server database.

**Authentication Process**

When the customer presents his share for any transaction hash code is generated for that share and compared with the database value. If it matches then the customer share is stacked with the share present in the database server. The stacked image is post processed to remove the noises. Finally authentication testing is done to accept or reject the user.

**A. Stacking**

The original image is reconstructed by stacking the transparencies[1]. When transparencies $i_1, i_2, \ldots, i_r$ are stacked together in a way which properly aligns the sub-pixels, one can see a combined share whose black sub-pixels are represented by the Boolean $OR$ of rows $i_1, i_2, \ldots, i_r$ in $S$.

- The grey level of the combined share is interpreted by the visual system:
  1. as black if $H(V) \geq d$
  2. as white if $H(V) < d - \alpha m$

- $1 \leq d \leq m$ is some fixed threshold and $\alpha > 0$

  Is the relative difference.

- $H(V)$ is the hamming weight of the “$OR$”

Combined share vector of rows $i_1, \ldots, i_n$ in $S$ vector.

when we are stacking the shares the following procedure is followed for each sub-pixels of share1 and share2. If the value of share1 is black then have the value as 1 in an output matrix1. If the value of share1 is white then have the value as 0 in an output matrix1. Do the process for share2 in output matrix2. Finally XOR the output matrix1 and matrix2 which gives the original image.

**B. Post-Processing**

The stacked shares results in an image where the black pixels will yield the required information. But other pixels which represents the grey image are randomly distributed. To avoid this noise, the technique of post-processing is applied on this image.
Visual Cryptography for Authentication Using CAPTCHA

Conclusion
In this paper CAPTCHA based Visual Cryptography scheme for secure authentication in internet applications has been proposed. Earlier approaches uses the signature of the customer for creation of shares. This involves manual intervention and increases the computation due to preprocessing. This approach is efficient by utilizing the CAPTCHA image (generated from the pin number) for the generation of shares which requires limited processing power. The future enhancement can be generation of hash code to check the integrity of the shares which is in the network.

Implementation and Performance Analysis
The implementation of generation of CAPTCHA image ,Share creation and stacking of shares is done using matlab7. Generation of hash code and verification of hash value of the user share is done using jdk1.5. We observed that any shares other than original can be identified using this process.

References

