CHAPTER 4

DEFENSE MECHANISM AT THE APPLICATION LAYER

4.1 INTRODUCTION

Defense mechanism at the Application Level concentrates on preventing automated tools access to web resources based on Referral Architecture and Picture based CAPTCHA Application. The objective is to filter numerous illegitimate accesses to web resources and provide access to the legitimate clients. In this chapter, an architecture based on Trusted Referral Server is proposed which provides secure access to the web server resources to the legitimate clients. A Picture based CAPTCHA technique is presented which provides improved security over text based CAPTCHA and other picture based CAPTCHA techniques at the same time ensuring the user friendliness of the system.

4.2 REFERRAL ARCHITECTURE

The major challenge of defending DDoS attacks is to tackle its distributed nature. Attackers are distributed and so attack agents. Millions of zombies are there in the attacker’s army to target the victim. The proposed Trusted Referral Server based architecture is a divide and conquer strategy to distribute the http request among the trusted referral servers and valuating the legitimacy of the request. As the illegitimate request to the web servers are infinite during a DDoS attack, providing a distributed defense mechanism
with the help of Trusted Referral Servers will make the defense mechanism as simple and easy. The architecture mitigates the DDoS attacks at the server side with a minimal overhead on the Trusted Referral Servers, at the same time ensures secure communication.

4.2.1 Trusted Referral Server Based Architecture for Mitigation of DDOS Attacks

The central idea of our referral architecture is to let the trusted referral servers on the website refer legitimate clients to it, while the web server is under a DDoS attack. Attackers are widely distributed and they attack the web server through millions of zombies. The proposed method redirects the incoming HTTP requests to one of the Trusted Referral Servers for validation. The TRS validates the client by issuing a Picture based CAPTCHA mechanism. The picture based CAPTCHA mechanism is discussed in detail in section 4.3. The requests of the clients who fail to clear the picture based CAPTCHA for three consecutive times are discarded. A privilege token is issued to those clients who clear the picture based CAPTCHA. The privilege token is transferred to the IPS through the client via secure cookies. Only clients trusted by the referrers are given the privilege tokens for accessing the target host. The IPS reads the privilege token via secure cookies and verifies the content of the cookie before accessing the web page from the web server. The IPS gain confidence as the qualified clients is those who are authenticated by the referrer website. Confidentiality, Integrity and Authentication of the privilege token is ensured. The communication flow between the client (C), Intermediary Proxy Server (IPS) and Trusted Referral Server (TRS) are depicted below.
Figure 4.1 Communication flow between Client, Intermediary Proxy Server and Trusted Referral Server

1. C → IPS : Http Request
2. IPS → TRS : Http redirect during DDoS scenario
3. TRS → C : Pic\textsubscript{CAPTCHA}
4. C → TRS : Solution for Pic\textsubscript{CAPTCHA} + IP\textsubscript{C}
5. TRS → C : Cookie\textsubscript{c}
6. C → IPS : Cookie\textsubscript{c} + IP\textsubscript{C} + TS\textsubscript{2}

where,

**Pic\textsubscript{CAPTCHA}** Picture based CAPTCHA

**Cookie\textsubscript{c}**

\[ E_{K_{IPS}}[IP_{TRS} + K_{TRS,IPS} + Token_{c} + H(Token_{c})] \]

**Token\textsubscript{c}**

\[ E_{K_{TRS,IPS}}[IP_{c} + TS_{1} + Lifetime_{1}] \]

**H(Token\textsubscript{c})** Token Digest

**IP\textsubscript{c}** IP address of the client

**IP\textsubscript{TRS}** IP address of the TRS

**TS\textsubscript{1}, TS\textsubscript{2}** Timestamps
The communication steps are described as follows:

1. The client access the web server through Intermediary Proxy Server. All the incoming packets are captured for packet analysis which is done by Botnet Analysis Engine.

2. During DDoS attack, the firewall redirects the request to the Referral server once the packets are captured for analysis.

3. Referral web server does a Picture Based CAPTCHA test with the client.

4. If the client fails the Picture Based CAPTCHA test, the referral server generates another Picture Based CAPTCHA test. If the client fails for more than three attempts, the request is dropped, suspecting a bot.

5. If the client clears the Picture Based CAPTCHA test, the referral server creates a cookie, which contains a privileged token inside and the request is redirected to the Intermediary Proxy Server. The cookie contains the IP address of the TRS, session key shared between TRS, IPS and privilege Token of the client encrypted by the public key of the Intermediary proxy server.

6. The Intermediary Proxy server reads the cookie and the verifier validate the origin of the cookie (created by the referral server). The IPS decrypts the Cookie_c and using the $K_{TRS,IPS}$ it reads the Token_c. The token contains the IP address of the client which the IPS uses to permit the http request to access the web servers. Once validated, the application
processes the client request w.r.to response from any one of the Virtual Machines (VMs).

4.2.2 Performance Analysis

Performance analysis focuses on two issues. Firstly, a security analysis is done on the secure communication between Client, IPS and the TRS server. Secondly, experiments are conducted to calculate the overhead of the referral server by forwarding 100, 200 client requests simultaneously. The time taken for successful connection by our proposed method is compared with WRAPS method and the results show slight increase in overhead in the proposed method. Considering the security benefits offered by the proposed method, the increase in overhead is negligible.

4.2.2.1 Security analysis of the communication between Client, IPS and the TRS Server

- *Secrecy, authenticity and Integrity of the privilege token*

  CIA parameters are ensured in the privilege token. The privilege token is created by the TRS server on successful clearance of the Picture based CAPTCHA test by the client. The token is encrypted by session key between TRS and IPS. Hence, client cannot open the token information. Moreover, the hash code of the token ensures integrity of the privilege token. In case of any modification attacks to the privilege token, the hash code doesn’t match at the IPS server and hence the client requests are rejected.

- *Not prone to Man-In-Middle attacks*

  Man in Middle attacks may happen by some intruders compromising the token and trying to flood the network. As
the privilege token contains the information about the client, any kind of Man-In-Middle attack is very well known by the IPS server and hence rejects the requests.

- **Not prone to Replay attacks**

The lifetime of the tickets plays a vital role in identifying replay attacks. The privilege token issued by the TRS server is subjected to a life time. Beyond the lifetime, the privilege token get invalid. Hence, any kind of replay attacks using the expired privilege token is not possible.

**4.2.2.2 Experimental setup**

The overall experiment was done on an Intel Core 2 Duo CPU with 3 GB RAM. Linux operating system with kernel version 2.6.24.7 is used. The simulations are done using network simulator NS-2.34, Click version 1.8.0 with packet size of 500 bytes. Dumbell topology is chosen as simulation topology with droptail management policy.

**4.2.2.3 Results**

Here we calculate the overhead on the referral server with the help of clients who are assumed to be legitimate by all means. These clients requests for forwarding requests to the referral server by sending 100,200 referral requests. The graph depicts, as the connection number of requests increases, so does the connection time as well as the percentage of successful connections. The X axis is the percentage of the successful connections and the Y axis is the connection time to the referral server in milliseconds. The evaluation for the scenario with the proposed referral architecture for a mean of 100 and 200 referrals is also given in the Figure 4.2 and Figure 4.3.
Figure 4.2 Comparison of connection time between WRAPS and proposed method for 100 referrals

![Comparison of Connection Time between WRAPS and Proposed Method for 100 referrals](image1)

Figure 4.3 Comparison of Connection time between WRAPS and proposed method for 200 referrals

In case of 100 referrals, the average time difference is 1.51ms and in case of 200 referrals, the average time difference is 4.805ms which seems
to be negligible considering the security benefits. Similar pattern continues as the number of referral requests increases.

4.2.2.4 Comparison of the existing architecture with the proposed architecture

Here we present the different approaches to mitigate the DoS attacks based on standard design criteria as given in the table 4.1. The table provides a subjective characterization of the various approaches in terms of deployment time, router overhead, network overhead and the end to end latency.
Table 4.1  Comparison of existing architectures with the proposed architecture

<table>
<thead>
<tr>
<th>Approaches parameters</th>
<th>Capability Approach</th>
<th>Secure Overlay Services</th>
<th>Pushback mechanism</th>
<th>Network Ingress Filtering</th>
<th>Web Referral Architectures</th>
<th>Trusted Referral Architecture</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic Architecture</td>
<td>Capability Distribution</td>
<td>Overlay Architecture</td>
<td>Aggregate based congestion control</td>
<td>Filtering Mechanism</td>
<td>Referral Architecture (WRAPS)</td>
<td>Trusted Referral Architecture</td>
</tr>
<tr>
<td>Deployment Time</td>
<td>Less</td>
<td>Moderate</td>
<td>Moderate</td>
<td>High Degree</td>
<td>Less</td>
<td>Less</td>
</tr>
<tr>
<td>Network Overhead</td>
<td>Moderate</td>
<td>Less</td>
<td>High</td>
<td>Less</td>
<td>Less</td>
<td>Moderate</td>
</tr>
<tr>
<td>Router Overhead</td>
<td>High</td>
<td>Moderate</td>
<td>High</td>
<td>Moderate</td>
<td>High</td>
<td>Moderate</td>
</tr>
<tr>
<td>End To End Latency</td>
<td>Moderate</td>
<td>High</td>
<td>High</td>
<td>Moderate</td>
<td>Moderate</td>
<td>Moderate</td>
</tr>
</tbody>
</table>
• **Basic Architecture**

Each approach follows an exclusive and unique architecture. The capability based mechanism involves the use of a capability distribution scheme which involves the distribution of tokens. This capability distribution scheme is again aggregated with the web referral architecture in the case of WRAPs. Secure overlay services are constructed using an overlay network which consists of selected routers which act as overlay nodes. Trusted Referral Server incorporates secure cookies to referral architecture. The pushback scheme works under the basic principle of aggregate based congestion control which helps in the identification of aggregates. Network ingress filtering takes into account the basic filtering mechanism based on ingress rules which are predetermined.

• **Deployment Time**

Considering the deployment time required for making an enhancement available, is highest for ingress filtering and path identification, where it requires a source address which provides proof of authorship if every node in the network is part of the trusted computing base. It can be observed that it is moderate for pushback mechanisms and secure overlay services as it needs modification to protocols and client side software and also supports strong incremental deployment properties. The deployment time is less in case of Trusted Referral server as the tokens are generated by the script running on the TRS server. The web referral architectures supports low deployment properties as it is built on an already existing architecture of web referrals. The capability
mechanism also has low deployment time as it require only basic modification either on server side.

- **Network Overhead**
  
The next design parameter is that of the network overhead, where the secure overlay services, network ingress filtering and the web referral architecture enjoys the least additional network overhead as it has an ability to recover from random or induced failures. The pushback has a high overhead as it does not scale well to multiple paths of attack in that it requires an exponentially increasing number of packets to accurately judge the attacking paths.

- **Router Overhead**
  
The next design parameter is the router overhead which takes into account the overhead caused due to the router modifications. Here we see that the router overhead is the highest for the capability mechanism, pushback, path identification and the web referral architecture as all the above mechanisms impose high level modifications to the routers involved thereby increasing the overhead. In case of Trusted Referral Server architecture, the overhead is moderate.

- **End to end latency**
  
The last important parameter is the end to end latency where the pushback mechanism seems to have the highest latency as it is effective only when an attack is isotropic that is, when the routers are fairly close to the target where most of the attack traffic will be arriving from a subset of the input links. The other mechanisms seem to have a moderate level of latency.
4.3 PICTURE BASED CAPTCHA APPLICATION

Automated network attack such as denial-of-service (DoS) leads to significant wastage of resources, which is a common threat to network security. To prevent these automated network attacks CAPTCHA based security mechanism is to be adopted so that it will differentiate humans from machines. Optical Character Recognition (OCR) based CAPTCHAs are more vulnerable to automated attacks due to the existence of correlation algorithms and direct distortion estimation techniques. The illegibility of the text CAPTCHA makes the user difficult to read it and thus they feel uncomfortable. In order to overcome these difficulties a new type of CAPTCHA, that is, picture based CAPTCHAs came into existence, which are more efficient and secure than the existing text based CAPTCHAs. We propose a new architecture for the generation of picture based CAPTCHA, which is resistant to segmentation through edge detection and thresholding, shape matching and random guessing. Our security analysis shows that the proposed architecture is showing 107.2 ms increase in Canny edge detection time, 91.2 ms increase in Zero Crossing Edge Detection time, 537.8 ms increase in time taken for segmentation through adaptive thresholding (with mean n=7 and varying threshold reduction r), 4263.8 ms increase in time taken for segmentation through adaptive thresholding (with median n=14 and varying threshold reduction r) in comparison with MOSAHP picture based CAPTCHA. Increase in the time taken for edge detection and segmentation indicates that the automated tools finds more difficult to pass the picture based CAPTCHA test.

The proposed scheme works on the current difficulty of image segmentation algorithms in the presence of a complex background.
4.3.1 Novel Architecture for Generation of Picture Based CAPTCHA

Automated web tools are used to achieve a wide range of different tasks, some of which are legal activities, whilst others are considered attacks to the security and data integrity of online services. Effective solutions to counter the threat represented by such programs are therefore required. The proposed scheme is able to prevent massive automated access to web resources. Properties of the proposed solution are that it is resistant to automated attacks as well as it has a good visual impact factor. That means automated tools find difficulty to pass the test but human can easily identify the image. Compared to common text-based HIPs which involve a check just once, this approach allows a more accurate control over the whole browsing process and a consequent higher level of security. The proposed scheme exploits the current computer’s difficulty in performing:

- Image segmentation in regions of interests, in presence of complex background.
- Recognition of specific concepts from background clutter.
- Shape matching when specific transformations are applied to pictures.

The proposed scheme challenges the user with a single large image composed of smaller and partially overlapping pictures.

These are taken from two different categories:

- Images representing real, existing concepts.
- Fake images, expressing artificially created, non-existent concepts.
Only small subsets of pictures, labeled real, belong to the first category and are those the user is required to identify. They are pseudo-randomly positioned within the mosaic image and are also partially overlaying each other, so that separating them into sub images is not an easy task for a computer. The other pictures, the fake ones, are created by using randomly chosen colors from the color histogram of the real pictures. Moreover, a mix of randomly created shapes, lines and effects are added to each of them. They are used only to generate background clutter, whose purpose is to make difficult the recognition of images expressing real concepts to non-human users.

The users are required to identify the picture indicated by the CAPTCHA and contained within the mosaic image. If this is done correctly, the resource is sent to the web browser. Otherwise, a new test is presented to the user, with a different set of images. In case the number of failed attempts from a single IP address exceeds a maximum, specific countermeasures are taken to slow down or stop the activity of a possible attacker.

Figure 4.4 depicts the architecture for generation of picture based CAPTCHA.
Our proposed technique is classified into three phases namely,

1. Foreground Image Formation
2. Background Image Formation
3. Picture based CAPTCHA Image Formation

### 4.3.1.1 Foreground image formation

The objective of this phase is to form a foreground image with the help of random set of real images. One of the images is chosen as the reference image from the random set. This reference image is prompted to
the end user for selection. For each of the random set of k images the following operations are performed.

i) Scaling

ii) Rotation

iii) Transparency

A scaling factor $S_j$ is randomly chosen for each images and a scaling function is applied. The reference image is scaled to keep the size of the reference image to a minimum. A rotation angle $\theta_j$ is randomly chosen and each image is rotated by this angle. A transparency factor $T_j$ is randomly selected and transparency function is applied. Create the composite image $C_{img}$ with transparent background and randomly place the resultant images.

4.3.1.2 Background image formation

The objective of this phase is to create a complex background image with fake images which makes it difficult for image segmentation algorithms. To create such a complex background image, the original input (random k images) except the reference image $I_{r}$ is taken and is subjected to certain atomic distortions. The atomic distortion operations used in this approach are Color Quantization, Dithering and Distortion.

Color Quantization

Color quantization or color image quantization is a process that reduces the number of distinct colors used in an image, usually with the intention that the new image should be as visually similar as possible to the original image. In the proposed architecture, the color spaces for the image
representations are quantized. The image pixels are transformed from RGB to CIE-LUV color space. The color points are subject to k-means clustering algorithm. The number of color clusters is controlled by a parameter. All the colors are then mapped to this reduced set of colors. When the number of color clusters is lower, it translates to loss of information and therefore the recognizability of the image is less.

**Dithering**

Full-color photographs may contain an almost infinite range of color values. Dithering is the most common means of reducing the color range of images down to fewer colors seen in 8-bit GIF images. Dithering is the process of juxtaposing pixels of two colors to create the illusion that a third color is present. Here, Floyd-Steinberg dithering algorithm is used in the proposed method. The algorithm provides an attractive distortion as it affects the low-level feature extraction. Randomly selected ‘n’ colors are used for dithering the region. The automatic image segmentation becomes difficult due to this dithering effect.

**Distortion**

In this phase, swirling distortion is applied to the images selected. Swirling distortion makes the image twisted or spiral pattern. The key nature of this distortion is that the image will become rotated in the center by the specified angle while the circular edge remains unaffected. After the swirling distortion, the image has more complex pattern.
Adding Noisy Lines

Noisy lines are added to the image in order to make it difficult for machine reading. But it also affects the human recognisability. Hence there is a trade off between human recognisability and machine reading. Here, thick lines, sinusoids are added as noisy lines as the pixel-wide noise are reversible by the median filtering.

4.3.1.3 Picture based CAPTCHA image formation

The foreground image is merged with the background image to form the final picture based CAPTCHA Image. Since the foreground image is formed with a transparent background, the background image created forms the background image. Figure 4.5 shows a sample picture based CAPTCHA image.

![Image](image.png)

**Figure 4.5** Picture based CAPTCHA image
4.3.2  Algorithm

The algorithm for the formation of Picture based CAPTCHA is summarized as follows:

**Algorithm**: Picture based CAPTCHA Image formation algorithm

**Input**: Set of real images

**Output**: Picture based CAPTCHA image

**Steps**:

1. Select k images randomly from the database of input images. Now we have a set of random images which is used to generate our final CAPTCHA image.

   \[ R = \{I_1, \ldots, I_k\} \]

2. Select a reference image \( I_r \) from the set \( R \).

   \( I_r \subseteq R \)

3. For each image \( I_j \in R \) except \( I_r \):
   
   a) Choose a scaling factor \( S_j \) randomly and apply a scaling function to \( I_j \)
   
   b) Choose a rotation angle \( \theta_j \) randomly and apply rotation to \( I_j \)
   
   c) Choose a transparency factor \( T_j \) and apply transparency function to \( I_j \)

4. Create the composite image \( C_{img} \) of size m by n with transparent background.
5. Randomly place each image $I_j \in \mathcal{R}$ into the composite image $C_{\text{img}}$. Note down the coordinate positions of $I_r$ in the composite image.

6. Create a background image $B_{\text{img}}$ of the same size of composite image $C_{\text{img}}$.

7. Choose each original image $I_j \in \mathcal{R}$ except the $I_r \in \mathcal{R}$ and perform color quantization, dithering and distortion. Add noisy lines.

8. Place the images of Step 7 into the background image $B_{\text{img}}$.

9. Merge the Composite Image $C_{\text{img}}$ into the background image $B_{\text{img}}$ to form the final picture based CAPTCHA image $P_{\text{img}}$. Since $C_{\text{img}}$ has a transparent background, now $B_{\text{img}}$ contains real and fake images.

10. Return the picture based CAPTCHA image $P_{\text{img}}$ along with the coordinates of the reference image $I_r$.

4.3.3 Performance Analysis

In the experiments, security analysis is done based on edge detection and thresholding. Edge detection is based Canny Edge Detection and Zero Crossing Edge Detection. Adaptive thresholding is done with varying mean and median. The time taken for edge detection is compared with the MosaHIP and our results are recorded.
4.3.3.1 Experimental setup

Experimentation is done on Intel Core i3 CPU M370@2.40GHz processor with 2 GB RAM. The software used are PHP, Graphic Library GD and Image Magick.

4.3.3.2 Results

Segmentation through Edge Detection

Current research in the image retrieval field shows that one of the most important challenges yet to be solved is visual concept detection in the presence of complex background or clutter. Therefore, an important security measure adopted by the proposed scheme exploits the difficulties of Content Based Image Retrieval (CBIR) methods to extract the semantic content expressed by a picture in the presence of complex background. Segmenting an image in regions of interest is indeed required to attempt an identification procedure in order to discriminate between images containing real concepts from fake ones. A possible method for executing such a task is to employ an edge detection algorithm, in order to delimit the contours of subpictures which compose the mosaic image. The procedure of contour separation is intended for fast reduction of image data volume and simplification of subsequent recognition steps. Image segmentation of final CAPTCHA images are tested using two effective and well-known techniques to detect edges: the zero-crossing Laplacian operator and the Canny edge detection. Figure 4.6 represents the final CAPTCHA image after canny edge detection and Figure 4.7 represents the final CAPTCHA image after zero crossing laplacian operator. The application of an edge detection algorithm has reduced the amount of information present in the image and extracted relevant features of each subpicture. However, the number of connected components within the
resulting image is still very high and does not allow to easily detect the shapes belonging to real images, which are needed to perform a CBIR process. The time taken for the edge detection is also a very important factor. Figure 4.8 gives the details of comparison of edge detection time taken between MosaHIP and the proposed method using canny edge detection and zero crossing edge detection. It is evident from Figure 4.8 that the segmentation time taken by the proposed method is higher than the existing method and hence the time taken by the attacker to compromise the CAPTCHA image proposed is higher.

Figure 4.6 The final CAPTCHA image after canny edge detection

Figure 4.7 The final CAPTCHA image after zero crossing edge detection
**Figure 4.8** Comparison of edge detection time between MosaHIP and proposed method

**Resistance to Segmentation through Shape Matching**

In computer vision, thresholding provides an easy and convenient way to separate regions of an image corresponding to objects of interest from the background. This segmentation process can be performed on the basis of different intensities of colors in the foreground and background regions of an image. The segmentation through thresholding is determined by comparing each pixel in the image with an intensity threshold and, if its intensity is higher than the threshold, the pixel is set to white in the output, otherwise it is set to black. Not all images can be neatly segmented into foreground and background using simple thresholding. Adaptive thresholding changes the threshold dynamically over the image by selecting an individual threshold for each pixel based on the range of intensity values in its local neighborhood. This allows for thresholding of an image whose intensity histogram does not contain distinct peaks. Images generated by the proposed scheme are not easily segmentable by means of color thresholding. Figures 4.9, 4.10, 4.11 represents the segmentation of the final CAPTCHA image through adaptive thresholding with mean n=7 and varying threshold. Figure 4.12 gives the
details of comparison of segmentation time taken between MosaHIP and the proposed method using adaptive thresholding with mean n=7 and varying threshold reduction r. It is evident from Figure 4.12 that the segmentation time taken by the proposed method is higher than the existing method and hence the time taken by the attacker to compromise the CAPTCHA image proposed is higher.

Figure 4.9  The final CAPTCHA image after adaptive thresholding, mean n=7, r=7

Figure 4.10  The final CAPTCHA image after adaptive thresholding, mean n=7, r=20
Figure 4.11 The final CAPTCHA image after adaptive thresholding, mean n=7, r=30

Figure 4.12 Comparison of segmentation time between MosaHIP and proposed method with mean n=7, varying threshold reduction r

Figure 4.13 gives the details of comparison of segmentation time taken between MosaHIP and the proposed method using adaptive thresholding with median n=14 and varying threshold reduction r. Figure 4.13 gives a graphical representation of the comparison. It is evident from Figure 4.13 that the segmentation time taken by the proposed method is
higher than the existing method and hence the time taken by the attacker to compromise the CAPTCHA image proposed is higher.

![Comparison of Segmentation time between MosaHIP and Proposed method](image)

**Figure 4.13** Comparison of segmentation time between MosaHIP and proposed method with Median n=14, r=30

**Visual Impact Factor**

The basic guideline for any CAPTCHA is that it must be easily recognizable to human. The visual impact factor gives the difficulty of human visibility and this factor varies from 1 to 5. If the visual impact factor is one, then it shows very easy for human to guess the reference image. Visual Impact Factor analysis with 100 users have been done and the results show that the proposed scheme has a very good visual impact factor of 2.

**4.3.3.3 Comparison of existing CAPTCHAs with the proposed architecture**

Table 4.2 gives a comparison of exisiting CAPTCHAs with the proposed architecture. The factors considered for comparison are:
• **Automation and gradability**

The test should be automatically generated and graded by a machine. This is the same as the old guideline and is the minimum requirement of a HIP system.

• **Easy to human**

The test should be quickly and easily taken by a human user. Any test that requires longer than 30 seconds becomes less useful in practice.

• **Hard to machine**

The test should be based on a well-known problem which has been investigated extensively, and the best *existing* techniques are far from solving the problem. An example problem that satisfies our requirement is “automatic image understanding” which is well known and has been investigated for more than three decades but is still without success. On the other hand, printed clean text OCR is not a hard problem, as today’s existing techniques can already do a very good job. HIP has an analogy to cryptography: in cryptography it is assumed that the attacker cannot factor 1024-bit integer in reasonable amount of time. In HIP, we assume that the attacker cannot solve a well known hard AI problem.

• **Universality**

The test should be independent of user’s language, physical location, and education background, among others. This guideline is motivated by practical considerations, and is
especially important for companies with international customers, e.g., Yahoo and Microsoft. It would be a nightmare for Yahoo or Microsoft if they had to localize a HIP test to 20 different languages. As an example, any digits based audio HIP tests are not universal because there is no universal language on digits. A different HIP test would have to be implemented for each different language, thus not cost effective. Strictly speaking, no HIP test can be absolutely universal, as there are no two humans that are the same in this world. However, we can make reasonable assumptions. For example, we can consider EZ Gimpy as universal because if a user can use a computer, it is reasonable to assume he or she knows the 10 digits and the 26 English alphabets. In contrast, Gimpy is not as universal as EZ Gimpy because users who know English have much better chance to succeed. Gimpy is quite difficult for non-English speakers.

- **Resistance to no-effort attacks**

The test should survive no effort attacks. No-effort attacks are the ones that can solve a HIP test without solving the hard AI problem. Here is an example. Bongo is a two-class classification challenge. To attack Bongo, the attacker needs no effort other than always guessing LEFT. This will guarantee the attacker to achieve 50% accuracy. Even if Bongo can ask a user to solve 4 tests together, that still gives no-effort attacks 1/16 accuracy. Animal Pix is another example that will not survive no-effort attack. Because there are 12 predefined animal labels, a no-effort attack can achieve 1/12 accuracy without solving the animal recognition problem.
The HIP tests that cannot survive no-effort attacks do not have practical usefulness and cannot advance AI research.

- **Robustness when database publicized.**

  The test should be difficult to attack even if the database, from which the test is generated, is publicized. For example, both Pix and Animal Pix would be very easy to attack once the database is publicly available. They therefore are not good HIP tests.

- **Blind-voting attack**

  Blind-voting attack executes repeated attempts to vote by subsequently dragging the chosen option on the first of the two pictures. Repeating the procedure for n times would allow the attacker to successfully cast an average of n/2 votes and fail the other half. All these wrong attempts would lead to an increasing temporary lock of the attacker's IP address, making the blind-voting attack too slow.

- **Similarity-based attack**

  Similarity-based attack tries to guess the correct binding between the indicated category and one of the displayed pictures by exploiting visual image similarities. First the bot builds up a repertoire of reference images for each category, retrieving several pictures from the page, which are then categorized by a human being. Then the real attack and works by repeatedly comparing the two candidate images against previous instances of pictures.
### Table 4.2 Comparison of existing CAPTCHAs with the proposed architecture

<table>
<thead>
<tr>
<th>Guidelines</th>
<th>Automation and gradability</th>
<th>Easy to human</th>
<th>Hard to machine</th>
<th>Universality</th>
<th>Resistance to no-effort attacks</th>
<th>Robustness when database publicized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gimpy</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>But the partially overlapped text can be hard to recognize</td>
<td></td>
<td></td>
<td>People who know English have much more advantages</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EZ Gimpy</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>It has been broken</td>
<td></td>
<td></td>
<td>Success rate(83%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bongo</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>A machine can randomly guess an answer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 4.2 (Continued)

<table>
<thead>
<tr>
<th>Guidelines</th>
<th>Automation and gradability</th>
<th>Easy to human</th>
<th>Hard to machine</th>
<th>Universality</th>
<th>Resistance to no-effort attacks</th>
<th>Robustness when database publicized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pix</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td>But the labels can be ambiguous (cars vs. Whitecars)</td>
<td></td>
<td></td>
<td>Some objects do not exist in some countries.</td>
<td></td>
<td>With the database, it becomes simple image matching</td>
</tr>
<tr>
<td>ESP-PIX</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Some animals are only popular in a few countries.</td>
<td></td>
<td>With the database, it becomes simple image matching</td>
</tr>
<tr>
<td>Pessimal</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>People who know English have much more advantages</td>
<td></td>
<td>Has only 70 words</td>
</tr>
<tr>
<td>Guidelines</td>
<td>Automation and gradability</td>
<td>Easy to human</td>
<td>Hard to machine</td>
<td>Universality</td>
<td>Resistance to no-effort attacks</td>
<td>Robustness when database publicized</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------</td>
<td>---------------</td>
<td>-----------------</td>
<td>--------------</td>
<td>---------------------------------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>BaffleText</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>ARTi FACIAL</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MOSAHP</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Proposed scheme</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
4.4 Conclusion

This chapter focuses on the Application Layer Defense Mechanism for mitigating DDoS attacks. Two architectures are proposed:

- Referral Architecture

- Picture based CAPTCHA application.

Firstly, a referral architecture is proposed where privilege tokens through secure cookies are given to legitimate clients. The security analysis ensure the CIA parameters of the privilege token. In case of 100 referrals, the average time difference is 1.51ms and in case of 200 referrals, the average time difference is 4.805ms which seems to be negligible considering the security benefits. Similar pattern continues as the number of referral requests increases. Secondly, A novel architecture of image CAPTCHA is proposed, which is more resistant to attacks from automated web tools. Since we perform a combination of atomic distortions in our system, it will reduce the machine recognisability. The security analysis of the proposed technique was discussed in terms of segmentation, shape matching and blind attack. The outcome of the security analysis shows that the proposed architecture is showing 107.2 ms increase in Canny edge detection time, 91.2 ms increase in Zero Crossing Edge Detection time, 537.8 ms increase in time taken for segmentation through adaptive thresholding (with mean n=7 and varying threshold reduction r), 4263.8 ms increase in time taken for segmentation through adaptive thresholding (with median n=14 and varying threshold reduction r) in comparison with MOSAHIP picture based CAPTCHA. Increase in the time taken for edge detection and segmentation indicates that the automated tools finds more difficult to pass the picture based CAPTCHA test.