

Counteracting Phishing Page Polymorphism: An Image Layout Analysis Approach

[†]Ieng-Fat Lam, [†]Wei-Cheng Xiao, [‡]Szu-Chi Wang, and [†]Kuan-Ta Chen

[†]Institute of Information Science, Academia Sinica

[‡]Institute of Computer Science and Information Engineering, National Ilan University
{iengfat, garry}@iis.sinica.edu.tw, wsc@niu.edu.tw, ktchen@iis.sinica.edu.tw

Abstract. Many visual similarity-based phishing page detectors have been developed to detect phishing webpages, however, scammers now create polymorphic phishing pages to breach the defense of those detectors. We call this kind of countermeasure *phishing page polymorphism*. Polymorphic pages are visually similar to genuine pages they try to mimic, but they use different representation techniques. It increases the level of difficulty to detect phishing pages. In this paper, we propose an effective detection mechanism to detect polymorphic phishing pages. In contrast to existing approaches, we analyze the *layout* of webpages rather than the HTML codes, colors, or content. Specifically, we compute the similarity degree of a suspect page and an authentic page through *image processing* techniques. Then, the degrees of similarity are ranked by a classifier trained to detect phishing pages. To verify the efficacy of our phishing detection mechanism, we collected 6,750 phishing pages and 312 mimicked targets for the performance evaluation. The results show that our method achieves an excellent detection rate of 99.6%.

1 Introduction

As network services become popular, people can send and retrieve emails, search for information, and conduct commodity trading, stockjobbing, and financial management tasks. However, as the functions of the Internet become more diverse and powerful, if user credentials, such as usernames and passwords, are leaked, the damage could be severe. For example, if a miscreant could obtain details of a user's online bank account, he could transfer the user's money to another account. Since such activities can be highly profitable, miscreants will try various methods to steal users' account details. One effective and popular way is called *phishing*.

Phishing is a kind of semantic attack whereby phishers send potential victims fake emails that purport to be from the account holders' banks or the banks' websites. Such emails may request updated information or password confirmation, and the phishers try to trick the recipients into providing their usernames, passwords, credit card numbers, or other personal information on the phishing pages. Phishers usually build phishing websites by faking the target pages. If recipients open the phishing pages, they may be deceived into thinking the pages are authentic and provide the requested information. If successful, phishers may

steal vast amounts of money themselves, or sell the users' information to other miscreants.

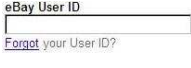
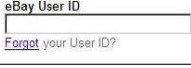
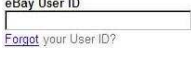
Phishing pages have to be quite similar to the authentic pages in order to deceive users. For this reason, phishers normally use a technique called *visual deception* [2]. Therefore, it should be possible to detect phishing pages by analyzing the visual similarity of a suspect page and the authentic webpage. However, to complicate and evade the detection process, phishers tend to use different representation techniques to create visually similar phishing pages. We call this kind of countermeasure *phishing page polymorphism*, after the polymorphism of computer viruses [3]. A polymorphic virus normally contains a variety of different encryption schemes that require different decryption routines. For example, virus programmers can add obfuscation codes to the original decryption function to alter the virus's signature so that it cannot be detected; or they can mutually reorder independent instructions to generate viruses with different binaries but identical functionalities. Moreover, a polymorphic virus may even change its own signature, i.e., its binary pattern, every time it replicates itself and infects a new file. These techniques greatly increase the level of difficulty for anti-virus programs to detect polymorphic viruses. Similarly, phishing page polymorphism applies different representation techniques for typesetting webpages with similar appearances. More specifically, a phisher can easily produce visually identical webpages by using different HTML tags, images, Flash, ActiveX, or some other dynamic components.

We use three examples to demonstrate phishing page polymorphism, as shown in Table 1. In case A, the phishing page is generated by copying the HTML codes of the sign-in page of eBay directly. The text on the page can be easily obtained by parsing the HTML codes. In case B, the text is replaced with images. Here, the HTML codes are quite different from those in case A. JavaScript is applied in case C to append content dynamically. Once the JavaScript function `show_content` is called, the HTML codes that have been created will be loaded into a DIV element. Because the HTML codes are usually constructed and loaded dynamically in case C, it is almost impossible to obtain text on the page by parsing the HTML codes. Although the HTML codes in these cases are clearly different, the pages appear to be identical on users' browsers. Therefore, a scheme based on HTML code similarity would fail to detect the polymorphic phishing pages in cases B and C, as they are visually similar to the genuine pages even though different representation techniques are used.

As mentioned above, text on a page can also be displayed through images or dynamic components, in this case, the text cannot be obtained by parsing the HTML codes. Images are the most common substitution for textual content. Besides, most browsers now support the embedding of dynamic components, such as Flash objects; hence, if all or part of a phishing page is displayed by Flash, to users, the page may still be the same as the mimicked target. Therefore, a visual similarity analysis technique which is not based on parsing HTML codes is required to fight phishing page polymorphism.

In this paper, we propose a layout-based mechanism for visual similarity analysis, and develop techniques for detecting polymorphic phishing pages based on

Table 1: HTML codes and screenshots of the sign-in page of eBay.com

Method	HTML Codes	Screenshot
A	<code>eBay User ID
<input type="text" name="userid" size="27">
Forgot your User ID?</code>	
B	<code>
<input type="text" name="userid" size="27">
</code>	
C	<pre>(Call function show_content() through body onload attribute) function show_content() { e.innerHTML = "eBay User ID
<input type=\"text\" name=\"userid\" size=\"27\">
Forgot your User ID?"; }</pre>	

the mechanism. During the detection process, we take *the whole page as an image instead of analyzing the HTML codes* [1]. Therefore, even if phishers adopt techniques like unicode homograph attacks, replace text with images, or use dynamic components, we can still detect a phishing page if it looks similar to the authentic one. In our experiments, we used the proposed detection mechanism to analyze phishing pages collected on the Internet. We fetched 312 pages and 1,531 corresponding phishing pages from 149 popular attack targets. The experiment results show that our mechanism achieves an accuracy rate of 99.6%, a false positive rate of 0.028%, and a false negative rate of 0.003%.

The contribution of this work is threefold:

1. We propose an image-based mechanism for detecting polymorphic phishing pages. Unlike current HTML-based methods, our mechanism analyzes web-pages as though they are images. Therefore, our mechanism can still achieve high detection accuracy, even if phishers create polymorphic pages by replacing text on the pages with non-HTML components, changing the structure of HTML codes, adding or removing some content, or applying unicode homograph attacks.
2. To analyze the visual similarity, we rely on the page layout rather than the page content; that is, our mechanism is robust to changes in textual content, colors, and images. This makes our mechanism more flexible than the HTML-based or other content-based methods.
3. We maintain a database of 6,750 phishing pages collected from 149 websites between August 2007 and January 2008. The results of experiments run on these polymorphic phishing pages and their mimicked targets demonstrate the efficacy of the proposed phishing detection mechanism.

The remainder of the paper is organized as follows. Section 2 contains a review of related works on phishing detection. In Section 3 we describe our layout-based phishing page detection mechanism in detail. Phishing detection experiments on extensive page samples are presented in Section 4 to evaluate the performance and accuracy of our mechanism. Then, in Section 5, we summarize our conclusions.

2 Related Work

Phishing pages and their targets are usually stored on different web servers. Therefore, we can protect users against phishing attacks if the authenticity of the web servers can be verified, and if users are allowed to sign in to the verified web servers only. Dhamija et al. [4] proposed Dynamic Security Skin, which enables web servers to prove their authenticity in a user-friendly fashion; while Tan et al. [5] proposed an ID-based SSL protocol based on the classic SSL protocol. Based on the protocol, users can verify the authenticity of a web server when establishing the SSL connections. Both of above the mechanisms allow users to determine whether the webpages they are browsing are genuine. Even so, these mechanisms are not popular, probably because some additional modules must be installed on both the server and the client.

On the other hand, since the content of most phishing pages is similar to that of the mimicked targets, some researchers have proposed content-based phishing detection methods. For example, Zhang et al. [6] apply TD-IDF analysis to the text on a page to extract keywords, which are input to a search engine. Then, phishing sites are detected based on the site ranking in the search results. Liu et al. [7] proposed a phishing detection mechanism based on visual assessment. They analyzed the DOM structure of a webpage to obtain its visual characteristics to detect phishing pages. These characteristics include block similarity, layout similarity, and style similarity. However, the mechanisms in [6, 7] can be rendered ineffective because phishers can easily rewrite the HTML codes to create phishing pages similar to the authentic pages. To address this problem, Fu et al. [8] proposed an image-based phishing detection method that applies the Earth Mover’s Distance (EMD) algorithm to calculate the similarity between the genuine page and the suspect page. However, the EMD-based method has some shortcomings: 1) it can only be applied to webpages with equivalent width and height ratios; that is, if a phisher creates a phishing page with a ratio different from that of the authentic page, the EMD-based method will be ineffective; 2) EMD may mistake a genuine page for a phishing page when its color disposition is similar to that of another genuine page; and 3) The accuracy of the EMD algorithm may be impacted if a phisher creates a phishing page by adding or removing content from the corresponding genuine page. Although our mechanism and that in [6] both apply image-based phishing detecting algorithms, there are some differences: 1) we analyze the similarity of page layouts rather than pixel information like colors and contrast; and 2) we take *partial similarity* into consideration; that is, even if some blocks have been replaced, added, or removed, or the colors have been changed, our mechanism can still maintain high accuracy.

3 Phishing Detection

In this section, we introduce the proposed phishing detection scheme. In our scheme, first the layout similarity between an authentic page and a suspect page is analyzed. Based on the similarity score, we then determine whether or not the suspect page is a phishing page. The detail steps are described in the following subsections.

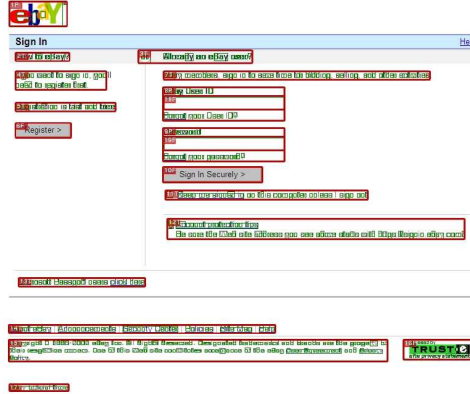


Fig. 1: A phishing page of eBay. Rectangles with bold edges are layout blocks obtained from the blobs.

3.1 Page Layout Analysis

First, we treat both the authentic page and the suspect page as images and apply Otsu’s thresholding method [9] to transform them into black-and-white images. We examine the pixels of the images, take all adjacent pixels with identical colors as a blob, and record their size and location information.

A blob is the most fundamental element in constructing a webpage, as the rectangular blocks with thin edges shown in Fig. 1. However, blobs may be very small that some of them even contain only a character, which is too small to be useful in layout analysis. Thus, based on the blob information, we divide the whole image into non-overlapped areas, called *layout blocks*, as the rectangles with bold edges shown in Fig. 1. The division process is performed as follows. For each blob, we find the minimum horizontal and vertical inter-space between the blob and all of its neighbors on the right-hand side. Next, the maximum of the minimal inter-spaces of all the blobs is selected. If it is larger than a threshold, the current block is divided into two sub-blocks at the midpoint of the maximum inter-space. Then, using the same procedure, we divide each sub-block into smaller blocks iteratively. If a sub-block cannot be divided any further, it is treated as a layout block.

The division process may cause severe fragmentation of layout blocks; for example, sometimes each line in a paragraph of text may form an individual layout block. Therefore, we apply the following heuristics in the division process: a) if the width or height of the inter-space is smaller than the average width or height of blobs in the sub-block, the division process will be terminated; and b) if the ratio of the size of the sub-image to that of the page is smaller than a threshold, the division process will also be terminated.

3.2 Layout Block Matching

Next, we compare the layout blocks of the suspect page with those of the authentic webpage to assess their similarity, as shown in Fig. 2. The matching process

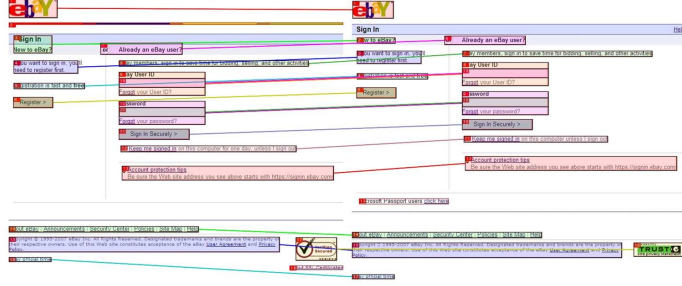


Fig. 2: The matched pairs of an authentic eBay webpage (left) and its corresponding phishing page (right)

is applied as follows. We assume there are a suspect webpage A and an authentic webpage B that contain layout blocks $a_1, a_2, a_3, \dots, a_m$ and $b_1, b_2, b_3, \dots, b_n$ respectively. First, we compare the sizes and locations of all block pairs (a_i, b_j) , $i = 1$ to m , and $j = 1$ to n . If the differences in the location, width, and height of a pair (a_i, b_j) are smaller than certain thresholds, the pair will be tagged as a candidate *matched pair*, and the block similarity degree D_{BS} will be calculated as the follows:

$$D_{BS} = \text{mean}\left(\frac{|w_{a_i} - w_{b_j}|}{T_w}, \frac{|h_{a_i} - h_{b_j}|}{T_h}, \frac{|x_{a_i} - x_{b_j}|}{T_x}, \frac{|y_{a_i} - y_{b_j}|}{T_y}\right),$$

where T_w and T_h are the respective thresholds of the width and height, T_x and T_y are the respective thresholds of the locations, i.e., the coordinates; and w , h , x and y are, respectively, the width, height, x -coordinate and y -coordinate of the top-left point of the layout block. Next, the block pair candidate with the highest block similarity degree is chosen as a matched pair. For a_1 , if its block pair candidates are (a_1, b_1) and (a_1, b_2) , and if (a_1, b_1) leads to a higher block similarity degree, (a_1, b_1) will be chosen as the matched pair of a_1 . However, if for a_2 , (a_2, b_2) is the pair with the highest block similarity degree, and if the degree is higher than that of (a_1, b_1) , then we will take (a_2, b_2) as the matched pair of a_2 ; and take (a_1, b_2) , which achieves the second highest degree of a_1 , as the matched pair of a_1 .

3.3 Similarity Score Computation

Having applied the layout block matching, we calculate the similarity scores of the matched pairs which will be used in the classification step. The similarity scores are defined by the following metrics:

1. The average differences between the minimal x-coordinates, maximal x-coordinates, minimal y-coordinates, and maximal y-coordinates of the layout blocks in each matched pairs.
2. The average differences between the widths, heights, and areas of each matched pairs.
3. The difference between the number of layout blocks on the authentic page and the number on the suspect page.

4. The differences between the total areas of the layout blocks on the authentic page and the suspect page.
5. The ratio $(0 - 1)$ of the number of blocks on the authentic page and the suspect page.
6. The ratio $(0 - 1)$ of the total areas of the layout blocks on the authentic page and the suspect page.
7. The ratio of matched blocks to unmatched blocks: Here, we consider several characteristics, which are ratios related to the number of matched blocks. We assume that there are two webpages, A and B, which contain N_A and N_B blocks respectively, and that there are N_M matched pairs between them. The following values are then used to calculate the layout similarity scores:
 - the match rate of webpage A: N_M/N_A ,
 - the match rate of webpage B: N_M/N_B ,
 - the match rate between the two pages: $(N_M \times 2)/(N_A + N_B)$.
8. The symmetry of the sizes and locations of the matched pairs: For each matched pair, we check whether the attributes, such as the size and location, of the pair and all the other pairs are *symmetric*. We use an example to illustrate the symmetry. Here we assume there are n matched pairs between webpages A and B, said $(M_{A_1}, M_{B_1}), (M_{A_2}, M_{B_2}), \dots$, and (M_{A_n}, M_{B_n}) . For each M_{A_i} , $i = 1$ to n , we compare its coordinates, width, height, and area with those of all the other matched blocks M_{A_k} , $k = 1$ to n , $i \neq k$. Our objective is to determine whether the comparison results are consistent with the results of the same comparison of each pair (M_{B_i}, M_{B_k}) . For example, if the width of M_{A_1} is larger than that of M_{A_2} and the width of M_{B_1} is larger than that of M_{B_2} , then the pairs (M_{A_1}, M_{B_1}) and (M_{A_2}, M_{B_2}) are symmetric in width. If for all $k = 1$ to n , $i \neq k$, the pair (M_{A_i}, M_{B_i}) and the pair (M_{A_k}, M_{B_k}) are symmetric in width, then we say that the pair (M_{A_i}, M_{B_i}) has symmetry in width. The truth values of the symmetry attributes are taken as the symmetry scores. The symmetry attributes include the area, width, height, the minimum and maximum of the x-coordinate and y-coordinate values.
9. Average of the block similarity degrees: Here, we take the average of the block similarity degrees of all matched pairs as a score.
10. Layout similarity degree: The layout similarity degree D_{LS} of two webpages is defined as the average of the block similarity degrees of all matched pairs multiplied by the match rate:

$$D_{LS} = \frac{\sum_{i=1}^{N_M} D_{BS_i}}{N_M} \times \frac{2N_M}{N_A + N_B},$$

where N_A and N_B are the numbers of blocks on webpages A and B respectively, and N_M is the number of matched pairs on the pages. D_{BS_i} is the block similarity degree of block i , $i = 1$ to N_M .

3.4 Phishing Page Classification

We use a supervised learning approach to determine whether a suspect page is indeed a phishing page. Since phishers try to create fakes of authentic pages,

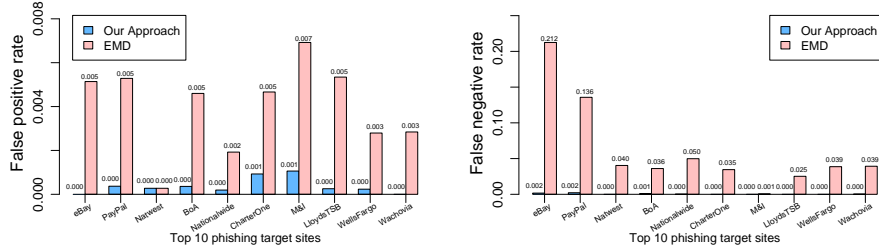
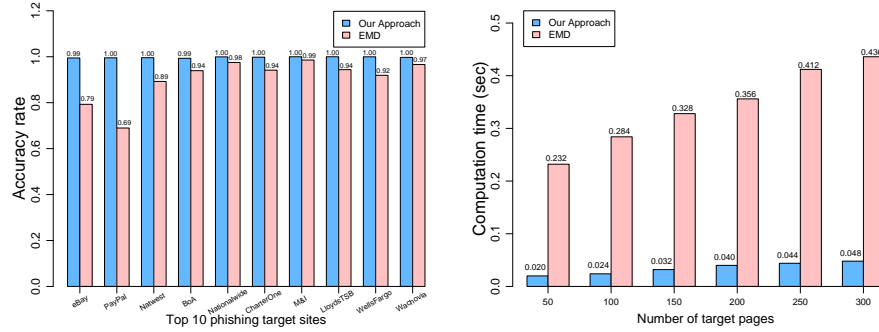


Fig. 3: Comparison of the false positive rate and the false negative rate of the top 10 phishing target sites



(a) Comparison of the accuracy rate of the top 10 phishing target sites (b) Relationships between the computation time and the number of target pages

Fig. 4: Performance of different phishing target sites and pages

there should be high degree of similarity between the layouts of a phishing page and its mimicked target. Conversely, the layout similarity of two non-phishing pages should be lower. Therefore, we use the layout similarity scores introduced in Section 3.3 as the feature vector for each page pair. Each page pair must be in one of the two categories “phishing” and “non-phishing”. We use a number of phishing page pairs and non-phishing page pairs to train the classifier. Once the classifier is trained correctly, for any suspect page, we can simply calculate the layout similarity degree between the suspect page and an authentic page which is likely to be the attack target using classification. If the classification result falls in the category “phishing”, the suspect page will be taken as a phishing page; otherwise, it will be considered a non-phishing page of the authentic page.

4 Performance Evaluation

4.1 Experiment Setup

From August 2007 to January 2008, we used an in-house automated program to fetch phishing pages from PhishTank and Anti-Phishing Working Group (APWG) every day. We then analyzed the URLs and text content of the collected 6,750 phishing pages to identify their mimicked targets. In addition, we manually collected 149 mimicked targets, which contained 312 authentic sign-in pages. The

6,750 phishing pages and the 312 authentic pages were used as sample pages in our experiment.

After extracting layout information from the collected pages, we computed the layout similarities of all the phishing pages and their mimicked targets. We obtained 1,531 layout similarity scores for the “phishing” category. From the 312 authentic pages, we calculated the layout similarity scores of all possible pairs of the pages and obtained 48,672 layout similarity scores for the “non-phishing” category. We set the thresholds to $T_w = 800, T_h = 600, T_x = 800, T_y = 600$ in the block matching processes, and then applied a naïve Bayesian classifier to categorize the pages based on their similarity scores. A ten-fold cross validation is performed in the evaluation of the phishing/non-phishing classification performance.

We compared the performance of our mechanism with that of the EMD algorithm. For the EMD algorithm, we followed the method proposed by Fu et al. [8] and set the width and height of a webpage to 100, the color degrading factor (CDF) to 32, and α (the amplifier) to 0.5. The parameter $|S_s|$, which is the number of samples for signatures, was set to 20. The weight, p , of the Euclidian distances between the RGB values of colors was 0.5, and the weight, q , of the Euclidian distance between the centers of color distribution was also 0.5. After re-sizing the phishing pages so that they were the same as the non-phishing pages, we took the number of pixels and the centre coordinates of the major colors as the signatures. Then, for each phishing page, we used the EMD algorithm to calculate the EMD distance between the phishing page and its mimicked target. This yielded 1,531 EMD distances for the “phishing” category and 48,672 EMD distances for the “non-phishing” category.

4.2 Evaluation Results

We define the accuracy rate as the ratio of successful classifications in the experiment; the false positive rate as the ratio of non-phishing pages misclassified as phishing pages; and the false negative rate as the ratio of phishing pages misclassified as non-phishing pages. The experiment results of our method show that the average accuracy rate was as high as 99.6% with a false positive rate of 0.028% and a false negative rate of 0.003%. The comparative rates derived by the EMD algorithm were 85%, 16%, and 31% respectively. Figure 4a shows the accuracy rates of our scheme and the EMD algorithm for various phishing target sites. From the figure, we observe that our mechanism achieves higher stability and accuracy. Moreover, Fig. 3 shows that our mechanism performs much better than the EMD algorithm for all of the ten different phishing targets in terms of both the false positive and false negative rates. The main reason is that the EMD algorithm is not robust to the change of image aspect ratio because it requires that all the images being compared have the same width and height. Therefore, if a phisher changes the aspect ratio of a fake page, the effectiveness of the EMD algorithm would be reduced due to the displacement of the centroid of important colors.

In addition, we compared the computation time of our mechanism with that of the EMD algorithm. The computation time is defined as the time used in

image capture, layout/image analysis, and page classification. Figure 4b shows that when the number of samples is 50, the computation time of our mechanism is only 1/10 that of the EMD algorithm. Moreover, as the number of authentic pages increases, the computation time of our mechanism only increases slightly. However, the increase in the computation time of the EMD algorithm is obvious. Therefore, our mechanism outperforms and is more efficient than the EMD algorithm.

5 Conclusion

In this paper, we have proposed a polymorphic phishing page detection mechanism based on layout similarity analysis. To cope with polymorphic counterattacks from phishers, we apply image processing techniques and analyze the layout of the page rather than the text content or the HTML codes. The image-based phishing detection mechanism is more robust than the HTML-based approach because it is more adaptable to phishing page polymorphism. In our experiments, 6,750 phishing pages and 312 authentic pages were analyzed and evaluated. The results show that our mechanism achieves an accuracy rate of 99.6%, a false positive rate of less than 0.028%, and a false negative rate of less than 0.003%.

Acknowledgements. This work was supported in part by Taiwan Information Security Center (TWISC), National Science Council under the grants NSC97-2219-E-001-001 and NSC97-2219-E-011-006. It was also supported in part by Taiwan E-learning and Digital Archives Programs (TELDAP) sponsored by the National Science Council of Taiwan under the grants NSC98-2631-001-011 and NSC98-2631-001-013.

References

1. K.-T. Chen, J.-Y. Chen, C.-R. Huang, and C.-S. Chen. Fighting Phishing with Discriminative Keypoint Features of Webpages. *IEEE Internet Computing*, 2009.
2. R. Dhamija, JD Tygar, and M. Hearst. Why phishing works. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 581–590, 2006.
3. P. Szor. *The Art of Computer Virus Research and Defense*. Addison-Wesley Professional, 2005.
4. J.A. Halderman, B. Waters, and E.W. Felten. A convenient method for securely managing passwords. *Proceedings of the 14th International Conference on World Wide Web*, pages 471–479, 2005.
5. C.H. Tan and J.C.M. Teo. Protection Against Web-based Password Phishing. In *Proceedings of the International Conference on Information Technology*, pages 754–759. IEEE Computer Society Washington, DC, USA, 2007.
6. Y. Zhang, J.I. Hong, and L.F. Cranor. Cantina: a content-based approach to detecting phishing web sites. *Proceedings of the 16th International Conference on World Wide Web*, pages 639–648, 2007.
7. W. Liu, X. Deng, G. Huang, and A.Y. Fu. An Antiphishing Strategy Based on Visual Similarity Assessment. *IEEE Internet Computing*, pages 58–65, 2006.
8. A.Y. Fu, L. Wenyin, and X. Deng. Detecting Phishing Web Pages with Visual Similarity Assessment Based on Earth Mover’s Distance (EMD). *IEEE Transactions on Dependable and Secure Computing*, pages 301–311, 2006.
9. N. Otsu et al. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.