Advanced Collage Steganography

Sajad Shirali-Shahreza and Mohammad Shirali-Shahreza

***Abstract*— Establishing hidden communication is an important subject that nowadays has gained increasing importance. Embedding a secret message into a cover media without attracting any attention, known as steganography, is one of the methods used for hidden communication. In this paper, we present an advanced version of "Collage Steganography" method which is robust against various attacks. In this method, an image (for example a view of a room) is created which contains some coded information. This method is robust against attacks such as adding noise, blurring, JPEG compression, and print-scan attacks.**

***Index Terms*— Image Steganography, Information Hiding, Print-Scan Attack, Collage, Multi-Resolution Processing.**

I. INTRODUCTION

By development of computer and expansion of its usage in different aspects of daily life and work, the issue of information security has gained special significance. One of the main concerns in the field of information security is the hidden exchange of information. For this purpose, various methods including cryptography, steganography, watermarking and coding have been used. Steganography is a method which has attracted more attention during the recent years.

The word "steganography" is a Greek word that means 'hidden writing'. While implementing this method, the main purpose is to hide data in a cover media so that other persons will not notice the existence of such data. This is a major distinction of this method with the other methods of hidden data exchange. For instance, in cryptography methods the individuals receiving the encoded data notice that such secret data exists but they cannot comprehend it. However, in steganography, the individuals will not notice the existence of such data in the sources [1]. It is also worth mentioning the difference between steganography and watermarking. The watermarking methods are used in applications such as copyright protection, preventing e-document forging [2] where the main goal is to protect/fingerprint the cover multimedia (image, video, speech, music, text, graphic, medical images) and thus one should make sure that the quality of the cover source is preserved during its transfer or possible attacks, while in steganography the main goal is the transfer of the hidden data in the cover source and thus its quality should be preserved during the subsequent processes. Most steganography methods have been performed on images [3], video clips [4], text [5], music, and sound [6]. It has also been implemented on various systems such as computers and mobile phones [7].

There are three important parameters in designing steganography methods: perceptual transparency, robustness, and hiding capacity. These requirements are known as "The Magic Triangle". It is well-known that these factors are contradictory [8].

Among steganography methods, the most common approach is to use images as the cover media for applying steganography (because the redundancy of information in images is high). In these methods, features such as pixels of image are changed in order to hide the information so as not to be perceived by human observers.

Various image steganography methods reported in the literatures can be classified into three major groups: temporal methods, transform domain methods, and fractal methods.

In temporal methods, the data in question is added to quantities of luminosity of pixels in the image. One of the common methods of temporal method is the least significant bit (LSB) method, in which the information is hidden in one to four least significant bits [9].

In transform domain methods, the image is first transferred into another domain. The information is hidden in that domain and finally the image is transferred back to the spatial domain using the related inverse transform. Some well-known transform tools which are used in steganography are discrete Fourier transform (DFT), discrete cosine transform (DCT) [10], and discrete wavelet transform (DWT).

In fractal methods, blocks of image that contain repeated patterns are selected and the information is embedded in them [11].

In general, steganography methods which are using image as cover media are usually applied on structural features of images. But, the method which we propose in this paper is fundamentally different from the other approaches. This is, in fact, an advanced version of our "Collage Steganography" [12] method. The general idea used in this work is embedding relevant pictures into a scene so as not to be noticeable. Then, depending on the location of the embedded object on the image and also the mode of the object, the desired data is hidden in the image.

As mentioned in [12], the "Collage Steganography" method has the ability to become resistant to different attacks. In this paper, we have designed an advanced version of the "Collage Steganography" which is resistant to JPEG compression, additional noise, blurring, and print-scan processes which are known as complex and difficult tasks [13]. While most of previous print-scan resistant methods try to hide information in any image and have

Computer Engineering Department, Sharif University of Technology, Azadi Street, Tehran, IRAN, shirali@ce.sharif.edu

Computer Science Department, Sharif University of Technology, Azadi Street, Tehran, IRAN, shirali@cs.sharif.edu

1999-4974

applications in copyright protection and authenticity verification [14-16], our method is designed to exchange information through the cover media. Therefore, in our method, an image is created which contains information, while other methods get an image and hide information in that.

The organization of the paper is as follows. In Section 2, the original "Collage Steganography" method [12] is described. The hiding phase of the new method is very similar to the original "Collage Steganography" method. Our main contributions are in the extracting phase. Our new proposed method is explained in Section 3. The experimental results, including the test which has been done to measure the robustness of the method are mentioned in Section 4. Finally, Section 5 concludes the paper.

II. Original Collage Steganography Method

In this section, we will describe the original "Collage Steganography" method. For more details refer to [12].

In this method, first a picture is selected as the background image (e.g., the picture of an airport runway). Then, the images of a number of appropriate objects related to the background are selected (e.g., birds in the sky, an airplane on the ground and a guide car). Each of these objects is selected from various types of that object. For example for the airplane, several types of airplanes such as training airplanes, passenger airplanes, military airplanes, and jet airplanes form the set of available objects to choose from.

Each of the selected objects can be placed only in a certain area, for example a car cannot be placed in the sky. Rather, a permissible area in the bottom left hand of the screen is given to the car. For instance, if a background scene is 480×680 pixels, the permissible area for the position of the airplane image with dimensions of 100×200 can be a rectangular area with apexes [(0,0), (0,600), (600,400), (0,400)] to the rectangular area with apexes [(20,50), (620,50), (620,450), (20,450)]; i.e., the image of the airplane can be displaced up to 50 pixels to the right and 20 pixels to the bottom.

Considering the above mentioned factors (existing objects, type of objects and their locations), one can create pictures with different views. For example, for the object of airplane in the above example, there are 4 types of airplane (training, passenger, military, or jet airplane) and 1,000 different positions (20×50=1,000). These create 4,000 modes in total (4×1,000=4,000). In other hand, there are two other objects (bird and car), which has 2000 different modes. In general, in this image the number of different modes is equal to 16×10^9 ($4000\times2000\times2000=16\times10^9$).

To determine the type and location of each object in the image (in order to hide information), first we convert the input text to arrays of bits. Indeed in the beginning of the array, we also put the size of input information so that the information can be extracted properly from the image.

Now, we calculate the number of possible modes for the first object; e.g., there were 4,000 modes for the airplane. Then, we calculate the greatest power of 2 which is less than the number of obtained modes. Equal to the number of this power, we read bits from input array. For example, the greates power of 2 less than number 4,000 is $2^{11}=2048$. Then, we read 11 bits of the input array. Now, we find the corresponding number of these bits at the base of 10. For example, if the 11 obtained bits are in the form of 00001100010, the corresponding number is 98. Now, we find the location and appropriate type of the object which corresponds to this number. To do this, first we divide the obtained number by the number of modes of the object. For example, in this example, we divide 98 by 4 (4 types of airplanes). The remainder of this division determines the type of the object. In this example, the remainder is 2. Thus, the airplane type is selected as military. Now, we divide the quotient of this division by the number of possible columns in which the object can be displaced. For example, here we divided 24, which is the quotient of division of 98 by 4, by 20. The remainder of this division shows the amount for which the object has to be displaced in the horizontal direction and the quotient of this division shows the amount for which the object has to be displaced in the vertical direction. For example, for the airplane, we have:

Horizontal displacement: 24%20 = 4
Vertical displacement: 24÷20 = 1

By adding these two quantities with the base location of the object, the image location is determined. For example, for the airplane, the base location is (600,400). So, the location of airplane is:

Horizontal position: 600+4=604
Vertical position: 400+1=401

In a similar manner, the type and location of the other objects are found according to the remaining bits. Thus, the information is hidden in the image. Now, the embedded image is sent to the individual in question. Indeed, one has to consider that the information of objects that consists of the object name, different object types, object location and permissible object displacement acts as a key and both sender and receiver must have them. Without this data, steganography and extraction of information is not possible.

For extracting the hidden information, the program is using the key file including the objects information and searches for different types of objects in the permissible area of each object to find the type and location of the objects. For example, from the rectangular area with apexes [(0,0), (600,0), (600,400), (0,400)] to the rectangular area with apexes [(20,50), (620,50), (620,450), (20,450)], it searches airplanes of the training, passenger, military, and jet types.

Now, considering the type and location of the object, we can extract the hidden data. The formula for obtaining the corresponding data of this image is briefly as follows:

Y = (object column number – base column of the object) + [length of permissible object displacement rectangle × (object row number – primary object row)] (1)

Fig.1. A sample of the Collage Steganography [12].

Corresponding number of the mode = number of present picture type + (number of picture types × Y)

Finally, by applying the inverse of the actions performed in the steganography process, the corresponding bits of this number are extracted. By implementing the above mentioned actions for all objects of the image and putting the extracted bits next to each other, the information hidden in the image is obtained. Fig. 1 shows an example of this method.

III. Our Proposed Method

In this section, our new method, named "Advanced Collage Steganography", is explained. This method has the same basic idea as "Collage Steganography". But, its implementation process is entirely changed; especially the extracting phase is redesigned, in order to make the method robust to different attacks. Therefore, the main contributions are in the extracting phase.

One of the design goals of the new method is to be robust against Print-Scan attacks. To the best knowledge of the authors, all of the reported steganography and watermarks methods that are robust to Print-Scan attacks, work on grayscale images. It seems that designing a steganography (or watermarking) method for color images which is robust to Print-Scan attack is very difficult. May be the main reason of this difficulty is that printers and scanners usually have different color gamuts. Therefore, we use grayscale images, although in the original "Collage Steganography" color images where used.

Except using grayscale images instead of color images, the overall hiding phase is similar to the hiding phase of the original "Collage Steganography" method.

At first, the type of objects and their locations are calculated. This is done by dividing the data by the number of types. The reminder is the object type. The quotient is then divided by the number of available places for the x-coordinate of object. The reminder plus the base point of object is the x-coordinate of the object location. The quotient is then divided by the number of available places for the y-coordinate of object and thus the y-coordinate of the object location is calculated. The quotient is used as data in continuance for finding the location and type of other objects.

In the original "Collage Steganography" method, to check whether the object is presented at any location, the number of pixels with the same color in the object and the region was counted. Although it was allowed that a small number of pixels differ, but it was prone to different attacks. For example, even simple attacks such as adding noise, could break that method.

To overcome this problem, here we have improved the method used to find the objects. In fact, instead of counting the number of pixels which are equal in region and object, here we use the correlation between the object and the region. Note that the correlation must be calculated between the foreground pixels of the object and equivalent pixels of the region.

Correlation is a good measure to assign the similarity of two images. But, the main problem in using the correlation is its computational complexity. If we intend to check all the locations which an object can occur on them using correlation measure, the program needs several hours to extract information. This is because each object may have about 100000 locations (for a 300×300 permissible box), normal size for the objects is about 400×500, and there are many objects to check for.

To overcome this shortcoming, here we use a multiresolution search approach. As such, we first downsample the image and the object with a big factor (20 in our implementation), and apply a quick search. After that, we select coefficients with high correlation. If the correlation of all coefficients is less than a threshold (0.4 in our implementation), then we conclude that the object does not exist in the image. Now, we apply a second quick search in regions around points which were chosen in previous search, at a higher resolution (5 in our implementation). Similar to first quick search, we choose coefficients with high correlation (bigger than 0.7). Finally, we apply an exact search in regions around points which are found in the second quick search.

This approach makes the extracting process fast. For most objects which are not in the image, the search is finished after the first quick search. In addition, the first quick search is very fast. For a small number of objects, the second quick search is performed. The exact search is only applied for objects which are in the image and objects are similar to them.

When the image is printed and scanned (Print-Scan attack), the image is usually displaced by a number of pixels. Although the displacement is usually small, but if the location in which the objects can be placed have a distance lower than this placement (like original "Collage Steganography" method which objects places have distance of 1), then this small movement will distort the hidden information.

To solve this problem, we segment the permissible box of each object to some nonoverlapping blocks where the objects can only be placed in the center of the blocks. In the extracting phase, even if the object moves a bit and is still in its block, the information can be retrieved correctly.

The size of the block is a parameter for robustness of our

Fig.2. Sample background used in our method.

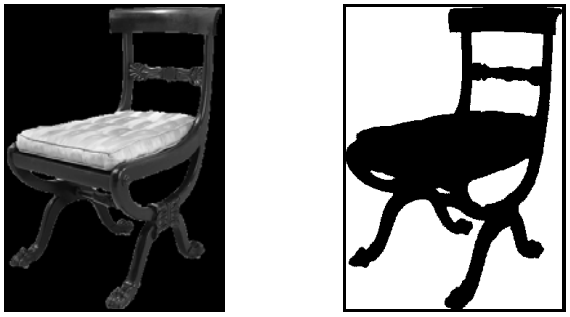
method. With increasing the block size, the method becomes more robust to displacements, but the hiding capacity is lowered and vice versa. This is consistent with the “Magic Triangle” phenomenon.

IV. Experimental Result

We implemented our new method using Matlab software. For the scene, we used the indoor view of a living room. We used the photo of an empty room as the background (Fig. 2). For objects, we choose four categories: clocks, chairs, dressers, and tables. In our database there are 5 types of clocks, 7 types of chairs, 24 types of dressers, and 16 types of tables. With this set, we can hide up to 74 bits in each image.The background image is a 2048×1536 image. Objects have different sizes. For example the size of the chair which is shown in Fig.3 is 300×418.

For each object, there are two images, one image contains object with black background (Fig. 3a) and another image shows the foreground of the image (Fig. 3b). To embed an object onto an image, that region is multiplied by the foreground image. Because the foreground pixels are zero and background pixels are 1, the pixels of region in which the object must be embedded become zero, while the background pixels do not change. Then, the image is added to the foreground object image. Because the background object pixels are zero (black), the background pixels of the image do not change, while foreground pixels are set to pixels of object image.

The hiding process is very quick and takes less than a second to be completed. But the extracting process is not so fast. In current implementation, it requires about 5 minutes to extract the information from an input image on a system with AMD 3200+ CPU and 1 GB of RAM.



(a) Object with black background (b) Foreground of the object

Fig.3. A sample object in our database.

Our method is robust to a large number of attacks. Here, we will describe some obtained results. The results are obtained for the sample image shown in Fig. 4 as the original image which contains hidden information.

First we tested the robustness of our method to JPEG compression. We tested different JPEG qualities, from 100 to 5. Even with JPEG quality 5, our method can extract all of the hidden information. Fig. 5 is the compressed version of Fig. 4 using JPEG with quality 5.

Fig.4. A sample output of our steganography method.

Fig.5. JPEG compressed image of Fig. 4 with quality 5.

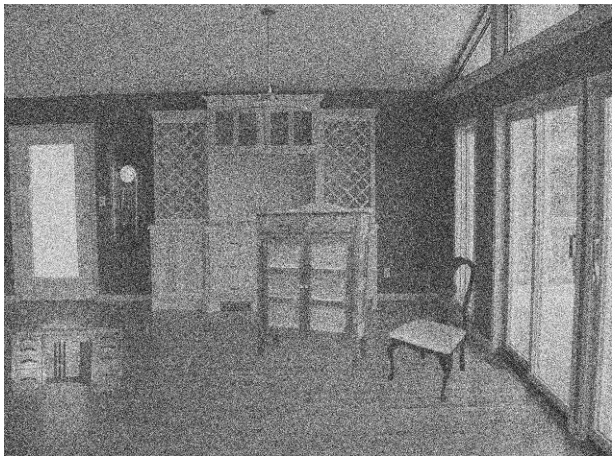


Fig.6. Gaussian added noise of Fig. 4 (SNR = 3.2dB).

Our method is also robust to Gaussian noise and blurring. Fig. 6 is the Fig. 4 with Gaussian noise. Fig. 7 is the blurred version of Fig. 6. The data was correctly extracted from these images. Fig. 7 is the blurred version of Fig. 6. The data was correctly extracted from these images.

Finally, we tested the Print-Scan attack. Unlike the methods such as [17] which print image with high quality (for example the image is 60 pixels-per-inch (ppi) and printed with a printer which print 600 dots-per-inch (dpi), then a square of 10×10 dots in printed image in dedicated to one pixel in the original image [13]), we printed the image at 300 ppi and 600 ppi with a 600 dpi laser printer. We then scanned the image with 600 dpi resolution. Note that the 600 dpi is usually used as the default printer quality for printing documents.

Then, we tested the robustness of our method to noise. We tested different amounts of random Gaussian noise. Even in presence of heavy noise (such as Fig. 6 which have signal-to-noise ratio (SNR) of 3.2 dB), our method can correctly extract the hidden information.

Fig. 8 shows the scanned result of the image which was printed using 600 dpi. This means that each dot printed on paper is dedicated to one pixel of image. The data extracted correctly with block size bigger than 8.

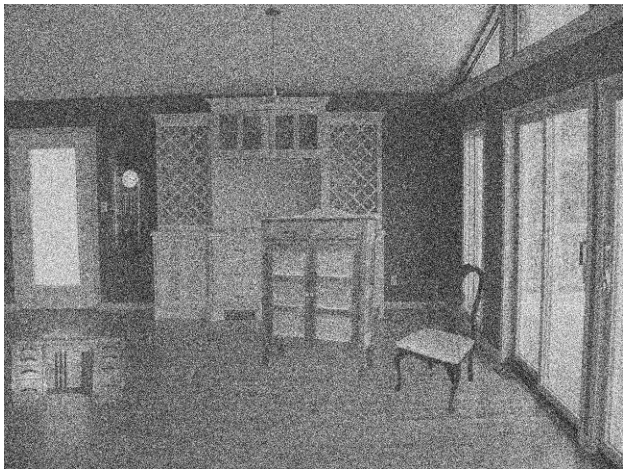


Fig.7. Blurred image of Fig. 6 (SNR = 4.7 dB).

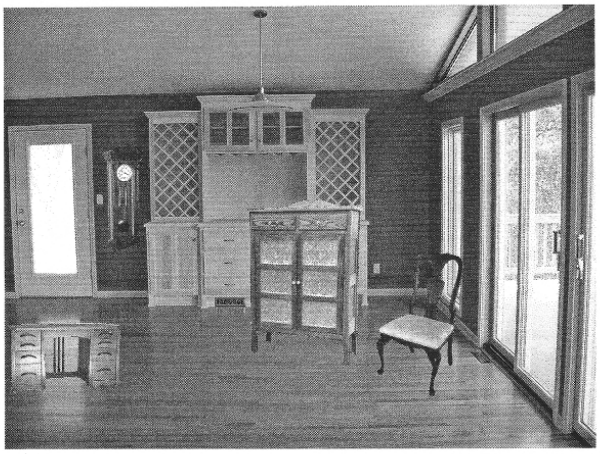


Fig.8. Scanned image of Fig. 4 printed using 600 ppi

We also did a more destructive test. We crushed the paper which was printed at 300 ppi and 600 dpi. This shows the extreme results of paper folding. Fig. 9 shows the scanned version of this image using 600 dpi scanning. The data was also extracted correctly from this image with block size greater than 8. This shows that our proposed method can be used to send information by using normal printers, send them by post and finally scan images by normal scanners.

To show the effect of block size, we have listed the location of the object founds in different conditions in Table I. Because the type of all objects was found correctly, we did not mention the type of each found object and only mention the lower right point of each found object. Table 1 shows that for JPEG compressed data and additional noise, the method can found the exact location of the objects. In the case of blurring, it found objects with a maximum of one pixel error. For the case of image printing and scanning, we found the location with less than 5 pixels error. The sources of this error are small rotations added during scans or displacements during selecting the bound of image in scanned images.

Fig.9. Scanned image of Fig. 4 printed using 300 ppi with 600 dpi resolution, crushed and scanned at 600 dpi.

TABLE I
LOCATION OF OBJECTS FOUND IN DIFFERENT CONDITIONS.

Condition	Chair	Clock	Dresser	Chair
Original (Fig. 4)	(1355,1589)	(859,489)	(1245,1243)	(1339,413)
JPEG quality 5 (Fig. 5)	(1355,1589)	(859,489)	(1245,1243)	(1339,413)
Heavy Noise (Fig. 6)	(1355,1589)	(859,489)	(1245,1243)	(1339,413)
Blurring (Fig. 7)	(1355,1589)	(858,490)	(1245,1243)	(1339,414)
Print-Scan (Fig. 8)	(1353,1585)	(856,485)	(1242,1239)	(1337,410)
Print-Scan with small rotation	(1357,1593)	(862,492)	(1247,1247)	(1343,418)
Crushed Printed Image (Fig. 9)	(1359,1590)	(860,487)	(1249,1246)	(1342,409)

V. CONCLUSION

In this paper, we proposed an advanced version of our "Collage Steganography" method. The idea of this method is to create a scene with different objects according to information which we intend to hide. Our method is fundamentally different from other methods reported in the literatures, because we create an image which contains the information, instead of hiding information in input images.

In this advanced version, in order to make the method more robust to the possible attacks we used the correlation measure to extract objects from the image. In addition, we used a multiresolution search framework to make the search process more robust to noise and also to reduce the computational cost.

Our tests showed that our method is robust to different attacks, especially Print-Scan attack. It is also robust against defects which may happen to printed images such as folding. In the case of additional noise or blurring, our method did not require any further processes such as Wiener Filters. It also tolerates small rotations.

REFERENCES

[1] J.C. Judge, "Steganography: Past, Present, Future," *SANS white paper*, November 2001, http://www.sans.org/rr/papers/index.php?id=552.

[2] F. Hartung and B. Girod, "Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video," in *Proc. of European Conf. on Multimedia Applications, Services and Techniques (ECMAST '97), Springer LNCS*, vol. 1242, 1997, pp. 423-436.

[3] R. Chandramouli and N. Memon, "Analysis of LSB based image steganography techniques," in *Proc. of 8th Int. Conf. on Image Processing*, 2001, vol. 3, pp.1019-1022.

[4] G. Doërr and J.L. Dugelay, "A Guide Tour of Video Watermarking," *Signal Processing: Image Communication*, vol. 18, no. 4, 2003, pp. 263-282.

[5] A.M. Alattar and O.M. Alattar, "Watermarking electronic text documents containing justified paragraphs and irregular line spacing," in *Proc. of the Security, stenography, and watermarking of multimedia contents Conference, SPIE proceedings series*, vol. 5306, January 2004, pp. 685-695.

[6] K. Gopalan, "Audio steganography using bit modification," in *Proc. of 28th IEEE Int. Con. on Acoustics, Speech, and Signal Processing*, 2003, vol. 2, pp. 421-424.

[7] M. Shirali-Shahreza, "An Improved Method for Steganography on Mobile Phone," *WSEAS Transactions on Systems*, vol. 4, no. 7, July 2005, pp. 955-957.

[8] N. Cvejic and T. Seppanen, "A wavelet domain LSB insertion algorithm for high capacity audio steganography," in *Proc. of 10th Digital Signal Processing Workshop and 2nd Signal Processing Education Workshop*, 2002, pp. 53-55.

[9] K. Curran and K. Bailey, "An Evaluation of Image Based Steganography Methods," *International Journal of Digital Evidence*, vol. 2, no. 2, Fall 2003, pp. 1-40.

[10] N. Provos and P. Honeyman, "Hide and Seek: An Introduction to Steganography," *IEEE Security & Privacy Magazine*, vol. 3, no. 1, May/June 2003, pp. 32-44.

[11] L.M. Marvel, C.G. Boncelet, and C.T. Retter, "Spread spectrum image steganography," *IEEE Transactions on Image Processing*, vol. 8, no. 8, 1999, pp. 1075-1083.

[12] M. Shirali-Shahreza and S. Shirali-Shahreza, "Collage Steganography," in *Proc. of the 5th Int. Conf. on Computer and Information Science*, 2006, pp. 316-321.

[13] K. Solanki, U. Madhow, B.S. Manjunath, and S. Chandrasekaran, "Modeling the Print-Scan Process for Resilient Data Hiding," in *Proc. of SPIE Security, Steganography, and Watermarking of Multimedia Contents VII, 5681*, (March 2005), 2005, pp. 418-429.

[14] P.J.Chiang, et.al., "Extrinsic signatures embedding and detection for information hiding and secure printing in electrophotography," in *Proc. of the 2006 American Control Conference*, Minneapolis, June 14-16, 2006, pp. 2539-2544.

[15] A. F.Martone, A. K. Mikkilineni, and E.J. Delp, "Forensics of things," in *Proc. of the 2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, Denver, Colorado, March 2006, pp. 149-152.

[16] P.W. Wong, and N.D. Memon, "Image processing for halftones," *IEEE Signal Processing Magazine*, vol. 20, no. 4, pp. 59-70, July 2003.

[17] K. Solanki, O. Dabeer, U. Madhow, B.S. Manjunath, and S. Chandrasekaran, "Robust Image-Adaptive Data Hiding: Modeling," Source Coding, and Channel Coding, in *Proc. of 41st Allerton Conference on Communications, Control, and Computing*, October 2003.