Volume 2, Issue 6, June 2012 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Remote Voting System for Corporate Companies using Visual Cryptography

Anusha MN
ISE & VTU, INDIA.
Email id: m.anusha539mail.com

Srinivas B K.
ISE & VTU,INDIA.
bksrinivas@rvce.edu.in

ABSTRACT: **The project titled "Remote Voting System for Corporate Using Visual Cryptography" aims at providing a facility to cast vote for critical and confidential internal corporate decisions. It has the flexibility to allow casting of vote from any remote place, even when key stakeholders of election process are not available at workplace. This is enabled by leveraging the features that are provided by visual cryptography that are implemented in Remote Voting System. The election is held in full confidentiality by applying appropriate security measures to allow the voter to vote for any participating candidate only if he logs into his login by entering the correct password which is generated by merging the two shares.**

Keywords: Visual Cryptography, Steganography, Cheat-detection, Digital Signature, Authentication.

I . INTRODUCTION

Trustworthy elections are essential to democracy. Elections are complex and involved processes that involve many components including voter registration, ballot preparation and distribution, voter authentication, vote casting, tabulation, result reporting, auditing, and validation. Either a technical or a human factors flaw in any part of the system can lead to an incorrect election result or reduce public confidence in an election. We are concerned primarily with the security and trustworthiness of the system, but realize that absolute security is not attainable. Convenience and security are often at odds, and we must consider practical and political realities in designing for security. In this paper, we only consider authentication. We realize there are many other important security issues to address before Internet voting could be adopted in governmental elections such as database security and denial-of-service attacks on the Internet, but do not consider those issues in this paper.

A. Voting over the Internet

When the term Internet voting is used, it generally refers to remote Internet voting, where the client software communicates over the Internet to the server software, say, from a voter's PC. However, there are at least two other ways to implement voting over the Internet: kiosk voting and poll-site voting. Each of these three ways has its own particular security requirements. Remote. In this scenario, a third party, or the voter himself (rather than election officials) has control over the voting client and operating environment. Kiosk. In this scenario, the voting client may be installed by election officials, but the voting environment is out of election officials' control. Poll-site. In this scenario, election officials have control over the voting client and the operating environment. Although the ADDER system was designed especially for remote Internet voting, nothing prevents it from being deployed for poll-site or kiosk voting, depending on the security requirements. ADDER also has the ability to carry out small-scale and large-scale election procedures, or even surveys where strong security may be less of a concern. It is not unreasonable to ask that remote Internet voting be as secure as voting by mail. We note that although remote Internet voting opens itself up to a wide range of attacks that may not be applicable to poll-site or kiosk Internet voting , it at least reduces the threat of insider attacks and allows less trust to be placed in the election officials. In many cases, voting machines arrive at polling places days or weeks early, making the threat of an on-site attack a real concern.

B. System overview

An ADDER election procedure is initiated through an interface which allows the administrator to provide the candidate list and specify the eligible users. Such users are voters and authorities. An ADDER election procedure progresses in the following manner. The authorities log into the system and participate in a protocol that results in the creation of a public encryption key for the system, and a unique private decryption key for each authority. Next, each voter logs on, downloads the public key of the system,

and uses that to encrypt the ballot, which is placed in an area of public storage specifically reserved for that voter. When the election is over, the server tallies the votes (using special encryption properties) and posts the encrypted result. Subsequently, the authorities provide some decoding information based on the encrypted result and their private keys. When enough such decoding information has been collected, the server combines the individual pieces to form the election result, which is then published. We note that ADDER does not employ any user-to-user communication; instead, users of the system (in particular, the authorities) communicate indirectly through the public bulletin board that is maintained by the system. Voters are only active in one round throughout the system's operation (unless they are also playing the role of the authorities, which is possible in our architecture). The ADDER system is implemented as a bulletin board server, an authentication server (the gatekeeper), and client software (either a Java applet or a stand-alone program).

C. Design & security goals

In creating the ADDER system, we adhered to the following design goals.

1) Transparency. All of the data on the bulletin board should be accessible to the public. This includes the encrypted votes, public encryption keys, and final tallies. The bulletin board does not store secrets.

2) Universal Verifiability. Any election result obtained by the system should be verifiable by any third party. By inspecting the election transcript, it should be possible to perform a complete audit of any procedure.

3) Privacy. All voters in an election should be confident that their individual choices will remain hidden. Only the total is made available to the public.

4) Distributed Trust. Each procedure is "supervised" by multiple authorities, and the final sum cannot be

revealed without the cooperation of a given number of authorities. Any attempt to undermine the procedure will require the corruption of a large number of authorities. Authorities and voters may overlap arbitrarily. Thus, it is possible for the voters themselves to ensure trustworthiness (or have an active role in it)

II. LITERATURE REVIEW

There are number of visual cryptography schemes in existence. Some of them are described below.

A. 2 out of 2 Visual Cryptography Scheme

In this type of visual cryptography scheme, the secret image is divided into exactly two shares. This is the simplest kind of visual cryptography. The major application of this scheme is found with remote voting system that uses 2 out of 2 secret sharing schemes for authentication purpose. To reveal the original image, these two shares are required to be stacked together. Figure 1 represents the division of black and white pixel in this scheme.

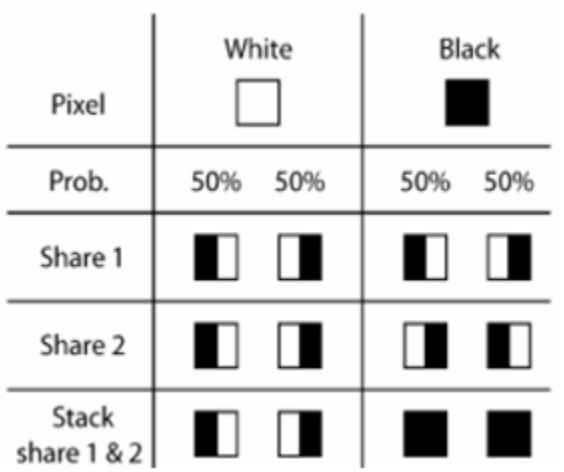

Figure 1. Basic concept of 2 out of 2 scheme

B. K out of N Visual Cryptography

This kind of scheme allows dividing a secret into K number of shares. Then the secret can be revealed from any N number of Shares among K. The major problem associated with this scheme is that the user needs to maintain many shares which may result into loss of shares. Also more number of shares means more memory consumption. The application of this scheme is found with banking system. For the joint accounts, three shares are generated. One is kept with bank's server, second is delivered to the one customer for the joint account and third share is delivered to the second customer. Hence both customers are able to access the account .

C. K out of K Visual Cryptography

Here original secret is divided into K number of shares and for reconstruction of the secret, all K shares are necessary. This scheme is not so popular because managing k number of shares is difficult task and it also increases time complexity

D. Visual Cryptography Background

Visual cryptography was first introduced in 1994, and provides provable secrecy in a way similar to a one-time pad. The simplest form of visual cryptography separates an image into two layers so that either layer by itself conveys no information, but when the layers are combined the image is revealed. One layer can be printed on a

transparency, and the other layer displayed on a monitor. When the transparency is placed on top of the monitor and aligned correctly, the image is revealed. For each image pixel, one of the two encoding options is randomly selected with equal probability. Then, the appropriate colorings of the transparency and screen squares are determined based on the color of the pixel in the image. This scheme provides theoretically perfect secrecy. An attacker who obtains either the transparency image or the screen image obtains no information at all about the encoded image since a black-white square on either image is equally likely to encode a clear or dark square in the original image. Another valuable property of visual cryptography is that we can create the second layer after distributing the first layer to produce any image we want. Given a known transparency image, we can select a screen image by choosing the appropriate squares to produce the desired image.

E. Generating Transparencies

Before an election, the election officials need to generate and mail image transparencies to eligible voters. To generate them, they need a secure symmetric key (hereafter, *Kg*). The election officials generate *n* random symmetric keys, *Ki*, where *n* is the number of eligible voters. A transparency is generated for each voter, using the result of encrypting *Ki* with key *Kg* as the seed to a cryptographic random number generator used to generate the transparency image. In addition to the image, the transparency includes the key *Ki* in a human-readable and typeable format. Note that there is no mapping between voter identities and the transparency they receive, and the corresponding screen image for *Ki* is yet to be generated. After the generation of transparencies, the election officials send the generated transparencies and an address list of eligible voters to a third party who sends each eligible voter a randomly selected transparency along with a voter information packet including voting instructions. We rely on the integrity of the U.S. mail as does absentee ballots. Anyone intercepting a transparency in the mail could cast an extra vote, but there are already well-established severe penalties for mail tampering to deter this. As with traditional absentee ballots, there is nothing to prevent voters from selling their votes. An opportunistic voter could sell the transparency to another voter, who can then use it to cast the desired vote. Without identity-based authentication in the voting process, it is unlikely that vote selling can be prevented. Our design assumes that the election officials generating the transparencies do not collaborate with the third party sending out voting packets. This property could be guaranteed by requiring an open process. For instance, the placing of transparencies in envelops could be conducted in public where voters could observe that the transparencies are selected randomly.

III. METHODOLOGY

A voter visits the election web site and enters the typeable version of the key *Ki* found on the transparency. We can encode a 64-bit key in 12 characters selected from lowercase letters and numbers. Many software packages require much longer input strings for their installation, so voters should not mind typing 12 characters. The election web site maintains a list of the *Ki* values used to generate the transparencies and checks that the entered key is on the list and has not been used already (extensions that would allow a voter to change a previously cast vote are possible but not considered here). If the entered *Ki* is valid, the election server (which has access to *Kg*) can calculate the corresponding transparency image. The election server then generates a random string to use as a password, and generates an image containing that string rendered as a bitmap image. The complementary image to the password image for the voter's transparency is generated and displayed on a web page returned to the voter. After the web server displays the corresponding image generated from *Kg*, the voter holds the transparency up to the screen to reveal the password. To continue the voting process, the voter enters the revealed password. This protocol serves to both authenticate the voter to the election server and the election server web site to the voter. Only someone with the correct *Ki* transparency could decode the password in the generated image; only something with knowledge of the transparency sent to the voter could generate a sensible password image. This process is more cumbersome, but provides substantially better security, than alternatives such as expecting a user to check a SSL certificate. In addition, we suspect from anecdotal evidence (but no scientific user studies yet) that nearly everyone will find the process of revealing a secret by holding a transparency up to an image on a monitor to be a satisfying and reassuring experience (some even find it magical!). Previous studies have analyzed how much a user needs to know in order to make rational decisions in the security of computer services, and the users showed they did not have a solid grasp on the security aspects of the system. With our system, voters do not need to understand how visual cryptography works, but are directly involved in performing the decryption in an intuitive and physical way. Our authentication scheme ensures that the voter cannot continue with the voting process without also verifying the server is legitimate.

A. Comparison to the present work

ADDER is an Internet-based e-voting system based on a strong voting-oriented cryptographic primitive (homomorphic encryption). ADDER is free software released under the GNU GPL. Anyone can create his own installation of ADDER for testing or general usage. To the best of our knowledge, ADDER is the first system of this kind. Moreover, ADDER compares particularly favorably against commercial Internet voting systems (e.g., SERVE).

For instance, ADDER supports large-scale trust distribution for voter privacy. As a large number of keyshare- holding authorities is supported, elections can essentially be run by the community. In addition, ADDER employs state-of-the-art encryption methods and puts forth the very attractive design principle of transparency: the bulletin board is publicly readable and holds no secrets. Thus, even if it is compromised, the privacy of the voters cannot be violated. Additionally, the whole election process is universally verifiable. Admittedly, ADDER has many limitations; nevertheless these are shared by all systems of the same kind. While there exist serious and justified security concerns regarding the employment of Internet-based voting for sensitive election procedures such as Presidential elections, we believe the existence of free and open source system like ADDER will motivate further testing and development, and will be a step forward in the development of truly robust and trustworthy e-voting procedures

IV . EXPERIMENTAL RESULTS

The theory proposed is verified in this section by experiments through different types of images. Simulation result on the black and white Lena image of 256× 256 size for 2-out-of-2 VC is reported here. The digital signature of the black and white Lena image is C7F2F183C2BB2F0195CDBFB9C71C0D80, which is generated by MD5, a freeware available for generating the digital signature of any type of file. The original image contains 55.17% white pixels. The size of the decrypted image is found to be four times that of the original image in 2-out-of-2 VC scheme for $m1=2$ and $m2=2$. The decrypted image contains 27.59% white pixels, because it is dominant with black pixels and so there is contrast loss in the decrypted image. Original image, two shares with embedded signature, decrypted image and reconstructed image are shown in Figure 2(a,b,c,d,g). Fake share s2, decrypted image with fake share and reconstructed image with fake share are shown in Figure 2(e,f,h). There is no different between decrypted image in Figure 2(d) and decrypted image with fake share in Figure 2(f) visually. Even the two reconstructed images in Figure 2(g) and Figure 2(h) are identical visually. Hence it cannot be concluded visually whether the fake share is produced by the cheater. The digital signature of the reconstructed image with the fake share is C&F2B181C BB2B0095B@_B18A__B0D09 that is different from the original signature, which shows that a fake share is produced by the cheater. So it can be verified in the presence of all shares by comparing the new digital signature generated by using MD5 from the reconstructed image and the extracted digital signature, when there is doubt that the cheater has produced the fake share.

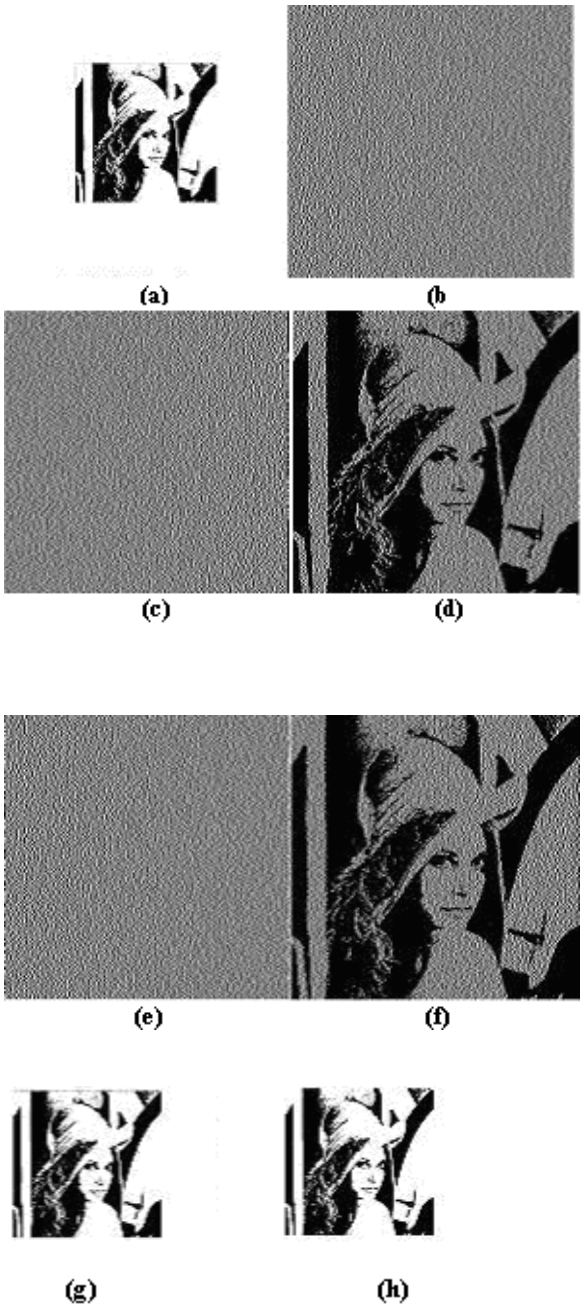

Figure 2. Experimental results: (a) Original image, (b) Share 1 with signature, (c) Share s2 with signature, (d) Decrypted image, (e) Fake share s2, (f) Decrypted image with fake share s2, (g) Reconstructed image and (h) Reconstructed image with fake share.

V. CONCLUSION

Internet-based voting offers many benefits including low cost and increased voter participation. Voting systems must consider security and human factors carefully, and in particular make sure that they provide voters with reliable and intuitive indications of the validity of the voting process. The system we propose uses visual cryptograph to

provide mutual authentication for voters and election servers.

ACKNOWLEDGMENT

The satisfaction and euphoria that accomplishes the successful completion of any task would be incomplete without the mention of people who make it possible. My project was the result of the encouragement of many people who helped in shaping it and providing feedback and guidance. It is with hearty gratitude that I acknowledge their contributions to my project. I avail this opportunity to express profound sense of gratitude and thank my internal guide **Srinivas B K,** Assistant Professor, Department of Information science and Engineering, RVCE for the constant guidance and feedback provided to me during the entire course of the project.

REFERENCES

[1] M. Naor and A. Shamir, Visual Cryptography, in "Advances in Cryptology – Eurocrypt '94", A. De Santis ed., Vol. 950 of Lecture Notes in Computer Science, *Springer-Verlag, Berlin*, pp. 1–12, 1995.

[2] M. Naor and A. Shamir, Visual Cryptography II: improving the constrast via the cover base, in "Security Protocols", M. Lomas, ed., Lecture *Notes in Computer Science* 1189 (1997), 197-202.

[3] E.R. Verheul and H.C.A. Tilborg van, "Construction and Properties of *k* out of *n* Visual Secret Sharing Schemes", Designs, Codes and Cryptography, vol. 11, pp. 179-196, 1997.

[4] C. Blundo, A Santis De and M. Naor, " Visual Cryptography for Gray Level Images", Image Processing Letters, vol. 75, pp. 255-259, 2000.

[5] H. Koga and H. Yamamoto, "Proposal of a lattice-based visual secret sharing scheme for color and gray-scale images", *IEICE Trans. Fundam.*, vol. E81-A, no. 6. pp. 1263-1269, 1998.

[6] Holmström, U. User-Centered Design of Security Software. *Proceedings of 17th InternationaSymposium, Human Factors in Telecommunications*, pp. 49-57, 1999.

[7] Karvonen, K. Creating Trust. *Proceedings of the 4th Nordic Workshop on Secure IT systems*, Nov. 1999.

