Authentication for Remote Voting

Nathanael Paul **David Evans**
University of Virginia
Department of Computer Science
151 Engineer's Way
Charlottesville, VA 22903 USA
[nrpaul, evans]@cs.virginia.edu

Avi Rubin
Johns Hopkins University
Department of Computer Science
3400 N. Charles Street
Baltimore, MD 21218 USA
rubin@jhu.edu

Dan Wallach
Rice University
Department of Computer Science
6100 Main Street
Houston, TX 77005 USA
dwallach@cs.rice.edu

ABSTRACT

Authentication is an important part of the voting process, both for the voting system authenticating the human as a legitimate voter without sacrificing secret balloting, and for the voter authenticating the vote recorder. Voters want the capability to vote remotely, but this makes both directions of authentication more difficult. Human factors are a crucial part of the authentication process. In particular, the system for authenticating the vote recorder must be designed in a way that ensures the human cannot be easily tricked into trusting an illegitimate recorder and so that the voter has confidence in the integrity of the voting process. In this paper, we discuss some of the issues associated with Internet-based remote voting and argue that visual cryptography offers a promising way to provide both satisfactory authentication and secret ballot guarantees.

Keywords

Authentication, human factors, human-computer interaction, voting, security.

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 [6], but do not consider those issues in this paper.

AUTHENTICATION

Establishing trust is one of the most important human-human and human-computer interactions. Authentication can be done using something you know (for example, a password), something you have (for example, a mechanical key), or something you are (a living, breathing human). These may be combined to provide stronger authentication. For example, a vigilant bartender may authenticate a customer by asking for a driver's license that has a birth date at least 21 years old (something you have), observing if the customer looks like the picture on the license (something you are), and asking for the zip code (something you know).

With traditional poll site voting, voters authenticate themselves by providing identification or an affirmation to a trusted poll worker; a poll site authenticates itself to a voter by being at a well-publicized physical location and having officials representing several different organizations present (including police and political party representatives). Internet-based voting offers great convenience, but does not offer such obvious authentication methods. Today, remote voting in governmental elections is done through absentee ballots that offer little security, and are slow and expensive to tabulate. Voters vote with their feet (and votes), of course, and remote voting is becoming increasingly accepted and popular. In the 2001 general election in Washington State, 69% of votes cast were cast by mail [9].

In voting, an additional concern is supporting secret ballots. The amount of trust needed before a voter is willing to cast a vote depends on the voter and the election. In typical United States elections, most voters are willing to place a high amount of confidence in the integrity of the process without needing definite mechanisms to guarantee their vote and identity cannot be linked. In other countries voters may not have such confidence. This is probably one of the reasons why every one of the 11,445,638 eligible voters voted "Yes" for Saddam Hussein in Iraq's 2002 presidential referendum [2] (the vote recording process is also suspect).

In the most recent election in Albemarle County, Virginia, voters enter a poll site and are given a numeric PIN by the poll worker after they check the voter's identity and inclusion on the list of eligible voters. The voter enters this PIN into the machine on which the vote is entered. There is no reason for the voter to be confident the assigned PIN will not be associated with the vote that was cast and the identity verified when the PIN was provided. With traditional voting, most voters are willing to accept this because they believe the poll workers are trustworthy. With remote voting, stronger measures are required, and it is important that the existence of those measures is clearly conveyed to voters in a way that establishes an appropriate level of trust [4].

APPROACH

We propose to provide remote authentication for both voters and voting systems using visual cryptography. David Chaum first proposed applying visual cryptography to elections to allow voters to verify their votes are included correctly in the final tabulation [1]. Using Chaum's secret-ballot receipts, a machine prints a receipt showing the results of a cast vote. The voter chooses to keep the top or bottom layer, each being unreadable without the other layer. Upon leaving the polling place, each voter can check to make sure the layer is correct and the vote was counted by visiting an official website that has a listing of all voter receipts used for tabulation. By itself, the voter's receipt does not reveal any information about the actual vote. In our work we do not consider ways for voters to verify their vote was recorded correctly in the final tabulation, but rather focus on the authentication process. By using visual cryptography, we believe it will be possible to establish authentication in a way that is satisfying to voters and resistant to large-scale fraud.

Our authentication scheme assumes the elections office mails each voter a voting packet including a printed transparency. This would not dramatically increase the cost of conducting an election, since many jurisdictions (including the state of Washington [8]) already mail any voter requesting a packet before each election. The voter will use this transparency to authenticate herself to the voting system and to verify that the election server is legitimate.

Visual Cryptography Background

Visual cryptography was first introduced in 1994 [5], and provides provable secrecy in a way similar to a one-time pad [7]. 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.

Figure 1 illustrates how an image is divided into layers. 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.

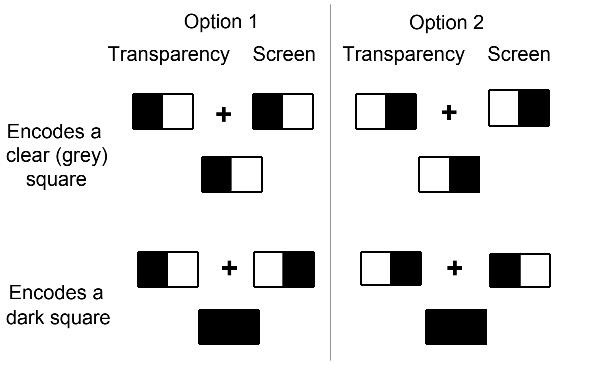

Figure 1. Visual Cryptography.

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, K_g). The election officials generate n random symmetric keys, K_i, where n is the number of eligible voters. A transparency is generated for each voter, using the result of encrypting K_i with key K_g as the seed to a cryptographic random number generator [12] used to generate the transparency image. In addition to the image, the transparency includes the key K_i 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 K_i 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.

Voting Process

A voter visits the election web site and enters the typeable version of the key K_i 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 K_i 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 K_i is valid, the election server (which has access to K_g) 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 K_g, the voter holds the transparency up to the screen to reveal the password (see Figures 2 and 3). 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 K_i 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 [3, 4]. 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.

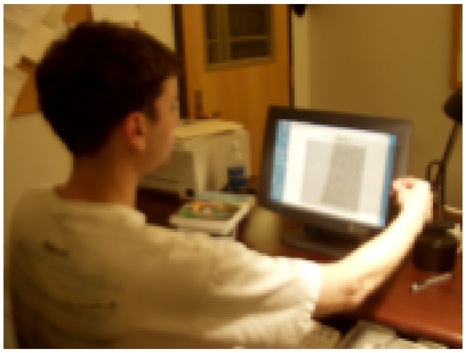

Figure 2. The screen image and transparency are random dots before aligning.

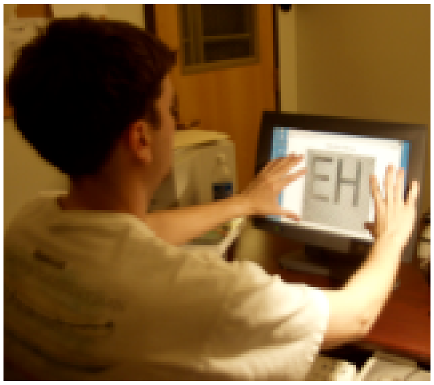

Figure 3. After aligning the transparency and screen image, the voter password (EH) is revealed.

ACCESSIBILITY

It is important that voting is accessible to all eligible voters, and some issues must be addressed to ensure this. The first problem deals with the usability of the security of the software [10, 11]. The user may find it difficult to go to the extra trouble of determining what is wrong if an incorrect image is seen. An incorrect image would just come up as garbage on the computer screen, which could be improper settings of a voter's monitor. The security holds, since the image will not be seen with an incorrect layer on the server. However, the problem cannot be corrected if the user does not notify anyone, but instead just assumes something is wrong and gives up on casting a vote. The user must know what action to perform in case an image does not appear. Designing and implementing intuitive mechanisms for sizing and aligning the image and transparency is a challenging problem. Our scheme provides a tradeoff between usability and security: larger pixels make it easier to align and view the image, but also decrease the length of the revealed password.

Our approach is not a good option for voters who are visually impaired, dyslexic, or with limited motor control.

Some voters may not be able to size and align the screen image, so an Internet-based voting system needs to be accessible to all voters. An alternative would need to be provided for those voters who cannot view images well enough to use our system.

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 cryptography to provide mutual authentication for voters and election servers.

ACKNOWLEDGMENTS

David Evans is supported by an NSF CAREER and NSF ITR Awards. David Elliot, Assistant Director of Elections for Washington State, provided useful insight into how elections operate.

REFERENCES

1. Chaum, D. Secret-Ballot Receipts and Transparent Integrity. *Palo Alto Workshop on Information Dynamics in the Networked Economy*, 2002.
2. CNN.com News. *Saddam gets perfect poll result.* 16 Oct. 2002. http://www.cnn.com/2002/WORLD/meast/10/16/iraq.vote/
3. Holmström, U. User-Centered Design of Security Software. *Proceedings of 17th International Symposium, Human Factors in Telecommunications*, pp. 49-57, 1999.
4. Karvonen, K. Creating Trust. *Proceedings of the 4th Nordic Workshop on Secure IT systems*, Nov. 1999.
5. Naor, M., Shamir A. Visual Cryptography. *Eurocrypt*, 1994.
6. Rubin, A. Security Considerations for Remote Electronic Voting. *Communications of the ACM 45*, 2, pp. 39-44.
7. Shannon, C. Communication Theory of Secrecy Systems. *Bell System Technical Journal*, *28,* pp. 656-715, 1949.
8. Washington, Secretary of State, Elections & Voting: Fraud Squad. http://www.secstate.wa.gov/elections/voterguide/.
9. Washington, Secretary of State, News Release. http://www.secstate.wa.gov/office/news.aspx?news_id=139.
10. Whitten, A., Tygar, J.D. Usability of Security: A Case Study. Technical Report CMU-CS-98-155, Carnegie Mellon School of Computer Science, Dec. 1998.
11. Whitten, A., Tygar, J.D. Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0. *Proceedings of the 8th USENIX Security Symposium*, Aug. 1999.
12. Yarrow PRNG. http://www.counterpane.com/yarrow.html.