Corey Proscia

CSC 290A Fall 2006

An Implementation of Chaum's Voter Verifiable Voting System

Background and Motivation

Electronic voting machines are in use around the world. One goal of developing an electronic voting system is to allow voters to verify that their votes are counted without revealing or giving them proof of who they voted for. This second condition is to prevent vote buying, where a voter is paid to vote for a particular candidate. Electronic systems are prone to failure and many don't have a way for a voter to be sure that their vote is counted. It may not be possible to correct a mistake or even detect if one was made. Another vulnerability of electronic voting machines is that an error, either intentional or unintentional, can cause damage on a large scale. Since electronic systems are more prone to failure either accidentally or on purpose and this failure can happen on a large scale, it is preferable to use one where the results can be verified by voters. Since electronic machines are being used, they can perform mathematical computations, for example using public key cryptography, to ensure that votes are only readable by a set of Trustees, but pieces of the vote can be published later. A voter may compare a receipt or piece of her vote to information published on a website to be sure that her vote was counted.

Prior Work

Rivest describes a system where the voter casts three ballots, voting for her candidate twice and for everyone else once (2006). The voter leaves with a copy of one ballot, which cannot be used to determine who she voted for, so all of her ballots can be posted on a website. Even though she does not have direct evidence of who she voted for, there may be enough ways to fill out her ballot that all three ballots posted on the website are unique. She can agree ahead of time how to fill out the ballots and use the posted results as evidence of her votes. This demonstrates the difficulty of designing a system that keeps votes private, while still being verifiable. A solution to this problem is to have the voter vote on an electronic machine that prints out the three ballots with marks distributed randomly. The voter can verify that there are two marks for her candidate and one for the others as before.

If an electronic voting machine must be used, it may be better to use one that gives some computational security to the voters' votes. Chaum describes a verifiable voting system where each

vote is printed on two layers (2004). The voter can see that the layers create a readable vote when viewed together, and she destroys one layer. The other layer is retained electronically and by the voter. Its information is secure because it is encrypted by a onetime pad. This half of the ballot can be decrypted securely because the seeds to a random number generator used to generate the onetime pad are encrypted used the public keys of a set of trustees. These trustees must decrypt the ballot in a particular order using their private keys. The interesting part of Chaum's system is how an error in generating the initial two layers can be detected, and half of the process of the trustees decrypting the receipts is revealed along with all of the partially decrypted receipts at each step. The revealed links between the ballots at each decryption step can verify the decryption, but that ballot cannot be traced further in either direction.

Vora explains Chaum's system with examples of how receipts are verified and decrypted (2004).

Voting Procedure

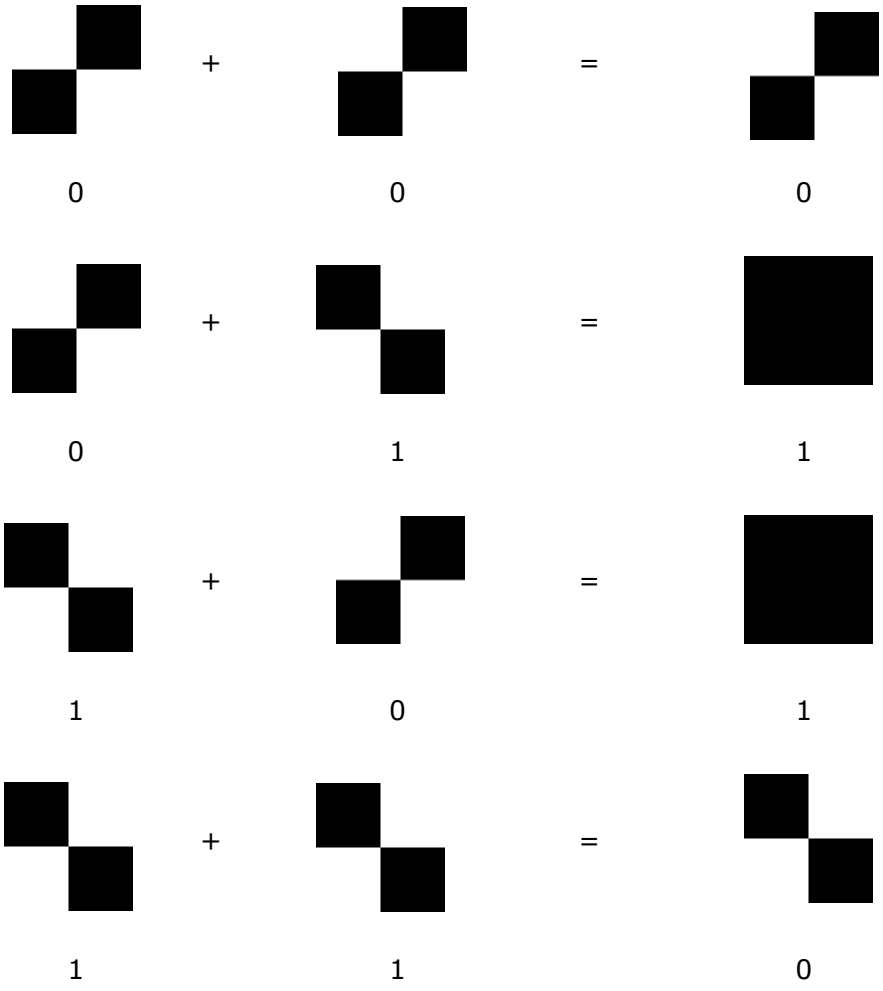
The people involved in the voting procedure are the Voter and Trustees. The Voter casts a vote and verifies that it is correctly computed. The Trustees maintain the privacy of the Voter's vote. Each trustee has a public and private key. There is also a voting machine, which has two public and private key pairs.

The ballot is printed as two layers, and the optical sum of these layers form the ballot. One of these two layers, which the voter chooses after they are printed, becomes the voter's receipt. The voter leaves the polling station with the receipt. After the voter chooses which layer is her receipt, additional information is printed on it. This additional information can be used to verify that the layer was generated correctly, but it cannot be used to decode the vote. Before Alice leaves the polling station, she destroys the layer she didn't choose using a paper shredder. Any electronic copy of this layer is also destroyed.

The voter, Alice, walks up the the voting machine and enters her vote. She votes for Lincoln. The machine then prints out the two transparent layers together. Alice can read the name Lincoln through the two layers. The machine then asks Alice which layer she would like to keep. Alice may choose either the top or bottom layer. Additional information is printed on the layer she chooses. Some of this information is encrypted so the Trustees can decrypt the receipt into the ballot, and other information allows Alice or someone else to verify that her layer of choice was generated correctly.

Sum of the top and bottom layers

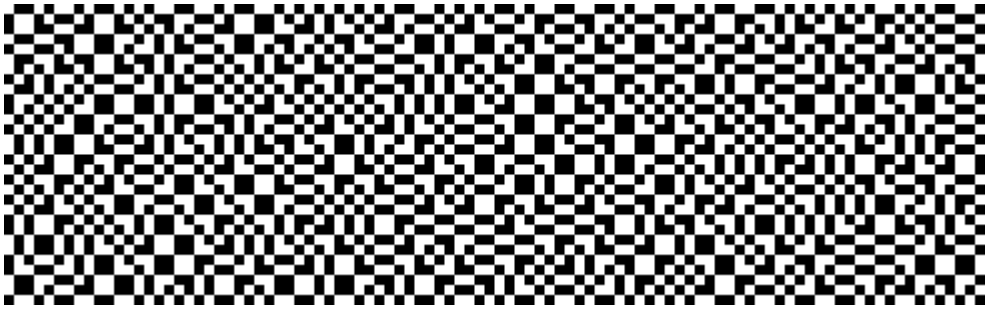
Between the two layers, half of the pixels represent Alice's vote and the other half are random. This is also true of the pixels on each individual layer. This forms the equivalent of a one-time pad since, given one layer, all possible sums are equally likely. Half of the pixels on each layer are generated by a pseudo-random number generator from two seeds, one seed for each layer. The non-random pixels on each layer are determined by the corresponding random pixel on the opposite layer and by the voter's ballot. As with a one-time pad, a pixel on the top layer “added” to a pixel on the second layer to form the voter's original ballot. Normally, the XOR operation is used to add each half of a one-time pad together. The works fine for a computer, and it is how the Trustees will later decrypt the ballot electronically, but for Alice to be sure that the layers sum to her ballot, they must be added optically, when 1 + 1 = 1. Visual cryptography can be used to visually implement the XOR operation.



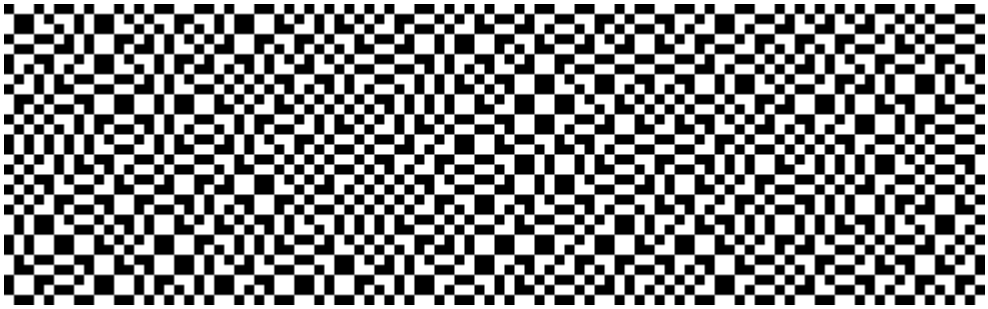
Visual Cryptography Between 2 Layers

Visual cryptography was introduced in 1994 by Naor and Shamir. Each pixel is represented as four subpixels in a square. Two of the subpixels are clear, and the other two are black. The combination of subpixels used to represent a white pixel are the opposite of the subpixels used to represent a black pixel. When two of the same pixels are overlaid, the result is a *clear* pixel made up of two clear subpixels and two black subpixels, and when two different pixels are overlaid, the result is a *black* pixel made up of four black subpixels. With this implementation, 1 + 1 = 0 because each 1 is represented by the same four subpixels.

Visual cryptography was easy to implement because it relied mostly on the abstraction of a Layer. Layers can be combined through either an XOR or an OR operation. Converting a layer that can be used in visual cryptography from a normal layer just involves creating four new pixels for each existing pixel.



Top Layer



Bottom Layer

Overview of Decrypting the Receipt

The random pixels on each layer are arranged in a checkerboard pattern. The Trustees decrypt the receipt by decrypting the destroyed layer's seed and generating the destroyed layer's random pixels. The Trustees combine this information with the non-random pixels on the receipt to produce half of the original ballot. This half of the original ballot is in the checkerboard pattern; every other pixel is recovered. The other half of the pixels cannot be recovered because that half's non-random pixels were on the destroyed layer.

Recovered ballot; half of the original

Since half of the pixels of each letter are lost, a special redundant font must be used. No two letters can differ by only one pixel because that pixel will be lost in half of the decrypted ballots. Any pair of letters must differ by at least two pixels that are an odd Manhattan distance away from each other. The letters in my implementation are based on an example ballot by Chaum. An example of redundancy is the *dot* at the top of letter “i”, which is represented by two adjacent pixels. At least one of these pixels will appear in every decrypted ballot.

Overview of Verifying the Accuracy of the Receipt

After Alice leaves the polling place with her receipt, she or a third-party can verify that it was generated correctly. Information on the receipt allows Alice to verify that the random numbers were generated correctly and that the information given to the Trustees was encrypted correctly. If the voting machine made a single error, Alice has a ½ chance to detect it because there is a ½ chance that the mistake is on the layer she chose.

For every error that is made by the voting machine, there is a ½ chance that it will be detected. If two errors are made, there is a 75% chance that at least one of them will be detected. Errors will be detected even if only a small fraction of the receipts are checked because each checked receipt halves the chance of all errors going undetected. To make receipt-checking easier, third-party groups can collect receipts from voters as they leave the polling station. The additional information on each receipt would be stored as a bar code. The public keys of the Trustees and the voting machine are needed to check all of data available.

When implementing receipt checking, I checked that the machine's signature of the receipt's id number was valid. I was not able to check that the seed corresponding to the chosen layer was encrypted correctly because of random padding used when encrypting with RSA. The two encryptions were always different. One other check that is possible to perform is checking that the random pixels on the chosen layer where correctly generated from the hashed signature.

Details on Decrypting the Receipts

To decrypt a receipt, a Trustee needs only the encrypted seed and the graphical part of the receipt (the pixels). Before the seed is encrypted, it is generated by the voting machine using an id number on the receipt. This id number is signed by the voting machine and run through a hash function: seed = hash(sig_{chosen}(id)). This hash allowed me to encrypt the seed because the length of the output of the hash function is smaller than the length of the signature. Java uses 1024 bit keys for RSA, but only allows encryption of up to 936 bits. It provides a separate abstraction for signatures. The signature is used to sign the id so that only the voting machine can encrypt ballots. This is important because the receipts and decrypted ballots are published and someone should not be able to encrypt and trace ballots backwards through the decryption process. There should be no way to link the ballots

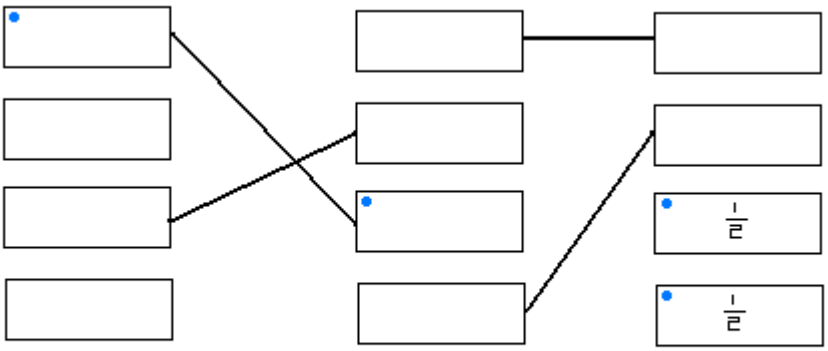
with the receipts. To encrypt ballots, you need the voting machine's private key. To decrypt receipts, you need the Trustee's private key.

Although my implementation uses only one Trustee, Chaum's voting procedure allows for multiple Trustees. The voting machine generates one seed for each Trustee. The receipt is the result of adding the random pixels generated from each seed together along with the ballot, so each trustee can add his decryption to the previous trustee's partial decryption before passing his partial decryption onto the next Trustee. Each Trustee has a different public and private key.

Chaum ensures that the seeds are decrypted in a particular order by encrypting the last Trustee's seed with the second to last, with the third to last, and so on. The result is a single “Doll” that can be given to the first Trustee. Chaum uses an analogy of Russian dolls to describe the encryption because the first doll contains both the first seed and the second doll. Within the second doll is the second seed and the third doll and so on. By ensuring an ordered decrypted process, parts of the decryption process can be randomly revealed to prove that no trustee is changing the results. It would otherwise be easy for a trustee to change the results because the decrypted ballot is the result of a simple XOR operation, which is very predictable and any Trustee would have full control over the results. It is fine to assume that the Trustees know the private key that the voting machines use to sign the id numbers of each receipt.

Auditing the Decryption Process

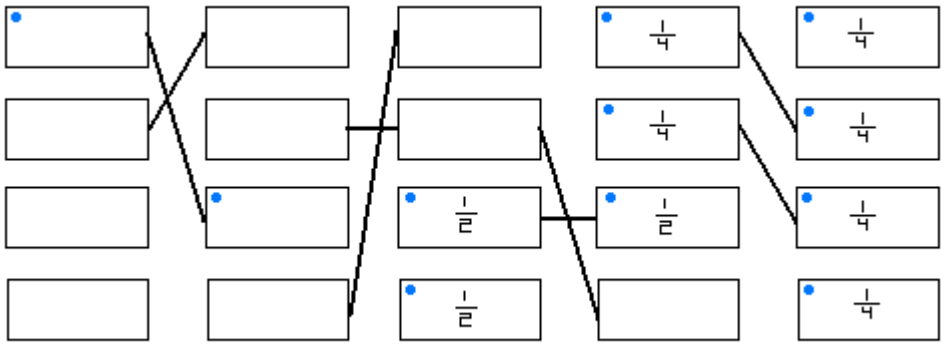
Chaum describes how half of the decryption process can be revealed without revealing the path of any one vote from start to finish. Each Trustee can process two adjacent seeds and shuffle the order of the receipts for each processing operation. After the decryption process is complete, a third-party can specify half of the links between the first and second to be revealed. The opposite set of links are revealed for the second operation.



Original Intermediate Sent to next Trustee

Two Decrypting Operations per Trustee

By revealing some links between stages of decryption in pairs, some ballots are still more likely to be associated with some receipts. To counter this, Chaum explains how each Trustee can perform four decryption operations to preserve the privacy of all of the ballots. A third-party specifies which links between the first and second set (the first decryption operation) should be revealed and the rest follows. The links revealed for the second operation are the complement of the first. The links for the third operation are half of those from the first and half of those from the second. The links for the fourth and last operation are the complement of those revealed for the third.



Original Sent to Next Trustee

Four Decrypting Operations per Trustee

Details on Verifying the Receipt

Information on the receipt:

Chosen Layer (pixels)

id

Encrypted hash(sig_{top}(id))

Encrypted hash(sig_{bottom}(id))

sig_{chosen}(id)

sig_{chosen}(everything above)

The receipt contains these 6 pieces of information. The first four pieces of information are printed before the voter has chosen which layer to keep, so the voting machine is committed to their values. sig_{chosen}(id) is the voting machine's signature of the id with one of its two private keys. When hash(sig_{chosen}(id)) is encrypted with the Trustee's public key, it should equal one of the two seeds that the voting machine encrypted. hash(sig_{chosen}(id)) can also be used to generate the random pixels on the chosen layer, which should match the random pixels on the receipt. With multiple trustees, one of the two Dolls would be computed instead of the encrypted hash(sig_{chosen}(id)).

Analysis

An advantage of Chaum's voting procedure is that the Trustees are accountable for decrypting the votes. The voter can also check to see if her receipt was among those decrypted. It would appear in the first set of receipts to be decrypted. The procedure balances voter privacy against accountability using probability. Every error has a ½ chance of being detected.

A possible disadvantage of Chaum's voting system is that errors may not be correctable if they are caught once decryption has begun. Key management is another issue. After the private keys on the voting machine are used they are no longer needed, but the Trustees must still maintain their private keys from when the voting machines are initialized to when the votes are decrypted. Chaum states that using secret sharing, it is possible for decryption to proceed without all of the Trustees present. Another issue is that voters must believe that their receipt does not contain enough information for anyone but the Trustees to determine their ballot, but only a small percentage must be willing to give up their receipts for the verification process to be successful.

Conclusion

Chaum's voting procedure succeeds at balancing accountability with voter privacy. It is not

possible for a voter to prove who she voted for, but any error in either generating the receipt or decrypting the receipt can be detected with a ½ probability. The procedure's use of a one-time pad means that some parts of the voting system would remain secure as technology improves. The part of the system's security that is based on computation uses public key encryption, which is well-understood and used in many applications today. Most importantly, Chaum shows that a voter does not have to trust either the voting machine or even the Trustees when using electronic voting machines.

References

Chaum, David. "Secret-Ballot Reciepts: True Voter-Verifiable Elections." *IEEE Security & Privacy*, May 2004: 38-47.

Naor M., Shamir A. "Visual Cryptography." *Eurocrypt*. 1994: 1-12.

Rivest Ronald. "The ThreeBallot Voting System." 2006. http://theory.lcs.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf

Vora, Poorvi. "David Chaum's Voter Verification using Encrypted Paper Receipts". 2004. http://www.seas.gwu.edu/~poorvi/Chaum/chaum.pdf