6.897: Advanced Topics in Cryptography Apr 23, 2004

Lecture 22: Voter Verification in Mix-net Voting Systems

Scribed by: Yael Tauman Kalai

1 Introduction

Any voting system is required to be verifiable. Namely, it is required that each voter can verify that his vote was counted correctly. On the other hand, it is also required that a voter cannot produce a receipt that allows him to prove to others how he voted. Thus, a voting system is required to be *voter* verifiable but not *publicly* verifiable. Our goal is to construct a voting system that satisfies both, seemingly contradictory, requirements.

In previous lectures, the notion of a *mix-net voting system* was presented. Recall that such a system consists of an initial encryption phase, followed by several mix phases and a final decryption phase. So far, we have focused on constructing verifiable mix phases. Note however, that all the mixing protocols that we considered were *publicly* verifiable. Namely, each mix server produced a receipt that proves the fact that a legal mix occurred.

Today, we are going to focus on constructing the encryption phase. This phase, as opposed to the mix phases, cannot be publicly verifiable, because if it were publicly verifiable then it could have been used as a receipt to a vote. Thus, this phase should be *voter* verifiable yet not *publicly* verifiable. Note that the voter is a human, and thus the encryption phase is required to be verifiable by a human. Constructing any protocol that a human can verify is very tricky, as humans, as opposed to servers, are very limited computationally.

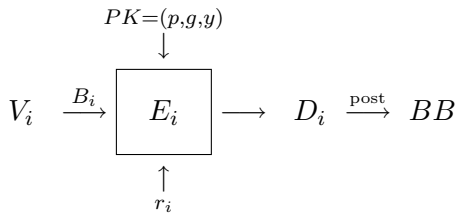
Chaum proposed an encryption phase which is *voter* verifiable yet not *publicly* verifiable. Two main tools are used in his construction: *visual cryptography* and the *cut and choose* technique. In this lecture we present a variant of Chaum's scheme.

2 Goals

Throughout this lecture we use the following notation.

- V_i denotes the i'th voter.
- B_i denotes V_i's ballot.
- E denotes the encryption box at the polling station, which encrypts by applying the El-Gamal encryption scheme.
- D_i denotes the encryption of B_i. That is, $D_i = (g^{r_i}, B_i y^{r_i})$.

Recall that for every i, (V_i, D_i) is posted on a public bulletin board. Thus, the encryption process is of the following form:



Our goal is to construct an encryption phase that satisfies the following properties.

- The voter has confidence that D_i represents B_i, i.e., that E is acting faithfully on his behalf.
- The voter is not given a receipt that allows him to prove to others how he voted.
- If E is misbehaving then the voter has an indisputable proof of its misbehavior (and the voter cannot produce such a proof if E is *not* misbehaving).

3 Construction

In what follows we present a construction of an encryption phase which is voter verifiable yet not publicly verifiable. The construction is a variant of Chaum's construction. Two main tools are used:

- **The cut & choose technique.** E generates a receipt which can be cut into two links. The voter can see both links but can keep only one link, of his choice.

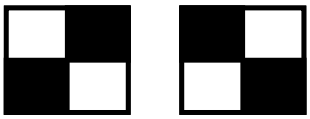
  This idea is carried out as follows: E, on input a ballot B_i, first encrypts B_i to obtain a cipher-text $C_i = (g^{r_i}, B_i y^{r_i})$, and then re-encrypts C_i to obtain a cipher-text $D_i = (g^{r_i+s_i}, B_i y^{r_i+s_i})$. E creates a receipt which consists of two links; the first link is the triplet (B_i, C_i, r_i) and the second link is the triplet (C_i, D_i, s_i). The voter keeps one link of his choice. The idea is that if E is misbehaving then, with probability $\frac{1}{2}$, the voter has a proof of this misbehavior.

  There are several problems with this approach.

  1. The triplet (B_i, C_i, r_i) is essentially a receipt that voter V_i used ballot B_i.
  2. In order to verify that, with probability $\frac{1}{2}$, his vote was counted correctly, the voter needs to check that the C_i in the triplet (B_i, C_i, r_i) is the same as the C_i in the triplet (C_i, D_i, s_i). This gives rise to the issue that humans are very limited computationally, and so it is not clear how a human can test whether two cipher-texts are equal.
  3. Any disruptive voter can create any triplet of his choice, and then complain that this was the triplet given to him by E.

There are several ways to overcome these problems. In particular, the first problem can be solved by having a bucket filled with triplets of the form (B_i, C_i, r_i) in the polling station, and so any voter can take any such triplet, regardless of his vote. Another solution to this problem is to program E to give the voter a triplet (B_i, C_i, r_i), for any B_i of his choice, regardless of his vote. The second problem, as we shall see, can be solved using visual cryptography, and the third problem can be solved by having E sign all the triplets that it generates. (Visual cryptography will also be used, as we shall see, make the receipts look more random.)

- **Visual cryptography.** This tool is used to help the voter carry out computations that seem too difficult for him to do. More specifically, using visual cryptography, a voter can compute the XOR function. This is done by transforming each string of bits into a transparency, where each bit is represented by a 2×2 matrix:



0 1

Therefore, if $B_i = B'_i \oplus B''_i$, then when aligning the two transparencies, corresponding to B'_i and B''_i, one on top of the other, we get B_i. Thus a voter can use his visual sense to verify that $B_i = B'_i \oplus B''_i$.

In what follows we give a more detailed description of the protocol. We omit the use of visual cryptography in this description. This part will be elaborated on in the next lecture.

3.1 The Protocol

Throughout the protocol E uses the El-Gamal encryption algorithm and two signature algorithms; an initial signature algorithm σ_I, and a final signature algorithm σ_F.[1]

1. Voter V_i submits a ballot B_i.
2. E chooses at random r_i and s_i, and computes $C_i = (g^{r_i}, B_i y^{r_i})$ and $D_i = (g^{r_i+s_i}, B_i y^{r_i+s_i})$. Let
   - $R_i = (V_i, D_i)$
   - $S_i = (B_i, C_i, r_i)$
   - $T_i = (C_i, D_i, s_i)$.
3. The voter gets R_i, S_i, T_i all signed with σ_I (the signatures come to assure that R_i, S_i, T_i were really given by the machine).
4. The voter checks that $B_i(S_i)$ is as intended. If it isn't the voter's intended vote, then the voter simply restarts the voting process.

[1] E can use the same signature scheme with two different keys.

5. The voter should also check that the following two equalities hold:

   - $C_i(S_i) = C_i(T_i)$
   - $D_i(T_i) = D_i(R_i)$

   If one of these equalities does not hold then the voter complains. Otherwise, the voter says "submit."

   Note that this is hard for a human to check. This will be improved later using visual cryptography; we can also make the receipts look more random this way.

6. $\sigma_F(R_i)$ is given to voter, and $(R_i, \sigma_F(R_i))$ is posted on the bulletin board. $(R_i, \sigma_F(R_i))$ is the official ballot.

7. Voter chooses whether he wants to keep (R_i, S_i) or (R_i, T_i) (he cannot keep both pairs!).

8. Voter leaves the voting booth, and checks off-line that all the signatures he received are valid signatures (with respect to σ_I and σ_F).

9. Voter can check that his ballot R_i appears on the bulletin board.

Remarks:

- *Correctness.* If $B_i \to C_i \to D_i$ is not a correct El-Gamal encryption and re-encryption, this will be detected with probability at least $\frac{1}{2}$.
- *The voter has no receipt.* As was mentioned earlier, the S_i's (even signed by σ_I) can be obtained easily. This can be done by having a bucket of many S_i's, or by programming E to give the voter a triplet (B_i, C_i, r_i), for any B_i of his choice, regardless of his vote. Thus, S_i is not necessarily related to the voter's vote. Given T_i, there is no way to verify (without knowledge of the El-Gamal secret key) that B_i is the plain-text corresponding to T_i. Thus, neither S_i nor T_i can be used as receipts.

Next lecture, we will see how to use visual cryptography to get rid of the bucket of S_i's and to check equalities of cipher-text by humans.

References

[NS94] M. Naor and A. Shamir. Visual Cryptography. Eurocrypt 94: pages 1-12. 68-77.

[CP92] D. Chaum. Secret Ballot Receipts: True Voter Verifiable Elections. IEEE Security and Privacy, January-February 2004: pages 3847.