Voting Criteria

Requirements for electronic voting include:

- Only authorized voters can vote.
- ≤ 1 ballot for each voter.
- Ballots counted as cast.
- Ballots are **secret** or "receipt-free": there's no way to prove to a third party that you voted a particular way (to prevent voter coercion).

There are logically two steps:

1. Cast ballot in a "ballot box".
2. Tally "ballot box" to get result.

Paper Ballots

If you think about old-fashioned paper ballots, they have the following properties:

- Cheap to operate.
- Easy to understand.
- Problematic if the ballot is long or complex.
- Subject to trickery, e.g., with chain-voting, where a goon hands you a filled-out ballot and demands that you bring out an empty ballot.
- Chain of custody: what happens with the ballot box between the time votes are cast and counted?

End-to-End (E2E) Cryptographic Voting

The basic outline is as follows:

1. Voter encrypts their ballot.
2. Voter casts the ciphertext vote.

3. System publishes the ciphertext ballots on a public bulletin board.

4. End of election tally:

   (a) Shuffle and re-encrypt ciphertext ballots.

   (b) Trustees collectively decrypt the ciphertext ballots, producing plaintext ballots.

   (c) Publish everything, so anyone can do the tally.

"Re-encryption" implies randomized encryption: one plaintext maps to many possible ciphertexts. This is an operation that, given a ciphertext C, computes $reenc(C)$ using a randomized algorithm with the property that $decrypt(reenc(C)) = decrypt(C)$. Essentially, the re-encryption still allows you to decrypt to the original plaintext.

El Gamal Encryption

For El Gamal encryption, you have public parameters g, p (Diffie-Hellman) and a private key x. The public key is computed as $g^x \bmod p$.

To encrypt a message M:

1. Pick a random r.

2. Compute $(g^r \bmod p, mg^{rx} \bmod p)$. Notice that this is computable knowing only the public key.

To decrypt a ciphertext (A, B), compute $A^{-x}B \bmod p$. Notice that this is only computable if you know the private key x.

El Gamal allows for re-encryption: given (A, B), you can generate a new random value r' and compute:

$$\begin{aligned}(Ag^{r'} \bmod p, Bg^{r'x} \bmod p) &= (g^r g^{r'} \bmod p, mg^{rx} g^{r'x} \bmod p) \\ &= (g^{r+r'} \bmod p, mg^{(r+r')x} \bmod p)\end{aligned}$$

Notice that the second line is just El Gamal encryption with random value $(r + r')$, i.e., it is a valid re-encryption, as it will decrypt to the same plaintext.

As an added bonus, you can prove to challengers that you performed a valid re-encryption by revealing the value r'.

Shuffling

We start with a "ballot box" B and end with B', which should be a reordering of B. To prove that the shuffler didn't cheat, we'll need to use a (zero-knowledge) proof protocol:

1. Prover (shuffler) produces B_1. B_1 should be equivalent to B and B'.

2. Prover knows the correspondence.

3. If B is *not* equivalent to B', then B_1 cannot be equivalent to both of them.

4. To challenge the prover, we have the challenger flip a coin and ask the prover to show the equivalence between B and B_1 or B' and B_1 (depending on the result of the coin flip).

5. Prover unwraps the equivalence by demonstrating the correspondence: "You can get from B' to B by a re-encryption with random value r."

6. The prover will fail at least half the time if he or she doesn't actually have a valid correspondence.

7. To increase our guarantee, run the proof protocol over and over.

Lower-Tech Voting

Type of record	Paper	Electronic
Example	Paper-ballots in ballot box	Voting machines
Counting	Slow, expensive	Fast, cheap
Main threat	Tampering afterward	Tampering beforehand

Paper with Electronic Records

As an example, consider optical-scan voting:

- The voters fills out a paper ballot.
- The voter feeds the ballet into a scanner.
- The scanner records an electronic record.
- The paper ballot is fed into a ballot box.

This produces both a paper and electronic ballot.