Artemis

Mike Boss

October 5, 2022

Participants.

1. Registrar (R): Generates cast and lock-in codes for each user.
2. Election authority (EA): Manages the bulletin board as well as its voter's login credentials. Tallies the cast ballots from the bulletin board.
3. Bulletin board (BB): A secure append-only-authenticated-write and public-read access bulletin board available to all participants.
4. Voter (V): A maybe malicious voter who can read, write as well as generate and remember short strings.
5. Voting terminal (VT): The device which generates the initial encryptions of the ballots.
6. Active voting assistant (AVA): A secondary device on which the choice is made used for privacy.
7. Voting assistant(s) (VA): Additional devices used for checking correctness of the VB, AVA and BB.

Note. Actions by the R and EA may be done in a shared fashion for more security. For simplicity only one actor for each is assumed in the following protocol.

Notation. Homomorphic re-encryption / re-randomization is denoted by $\{e\}_k^n$ for a ciphertext e encrypted n times using key k. Initial encryption is denoted by $\{e\}_k^1$. Any re-encryption increases n by one.

1 Re-encryption mix-net

The protocol.

1. **Election Setup.**
   (a) EA generates a public / private key pair and publishes the public key $(_{EA})$ on the BB.

    (b) R sends casting and lock-in codes to all eligible voters.

    (c) Voter's create login credentials with the bulletin board.

2. **Session Setup.**

    (a) VT generates a public / private key pair and publishes the public key on BB.

    (b) V enters a short string on the VT which is appended to the *sessionID*.

    (c) VT starts a session using a new *sessionID* appended with the short string.

    (d) VT displays the *sessionID* as well as a QR code containing a secret symmetric key k_{secret} and the *sessionID*.

    (e) V checks that the short string is appended to the *sessionID*.

    (f) V scans the QR code with the AVA and optionally with additional VAs.

    (g) VAs find the session corresponding to the *sessonID* on the BB, obtaining the public key of VT. The VAs display the received *sessionID*.

    (h) V checks that all the *sessionID*'s match with the original.

3. **Ballot-color-correspondence Setup.**

    (a) VT obtains the list of ballots from the BB.

    (b) VT encrypts all the ballots with the EA's public key ($\{B_i\}^1_{EA}$).

    (c) VT generates a correspondence ($\{\{B_i\}^1_{EA}, c_i\}$) between a permutation of the list of encrypted ballots and a set of colors (c_i).

    (d) VT publishes the correspondence encrypted by the pre-shared secret symmetric key (k_{secret}).

4. **Ballot Choice.**

    (a) The AVA obtains the ballot-color-correspondence from the BB, decrypts it using the pre-shared key and display the colors to V.

    (b) The V chooses the color corresponding to the choice they want to make displayed on the VT.

    (c) The AVA re-randomizes the encryption of the chosen ballot ($\{B_i\}^2_{EA}$) and publishes it on the BB.

    (d) The BB re-randomizes the received ballot ($\{B_i\}^3_{EA}$) before making it publicly visible.

5. **Audit. (Optional)**

    (a) The V initiates auditing from either the VT or the AVA. The respective device sends a message to the BB that the ballot is used for auditing. The BB informs the other device of the audit.

(b) The VT, AVA and BB reveal their randomness used for encryption or re-randomization respectively on the BB.

(c) The VT and AVA compute the actual choice made by the V, check if all steps were done correctly and finally display the choice to the user. The BB is unable to learn the choice as the initial correspondence was encrypted with the pre-shared key.

(d) The V checks if their choice was correctly recorded. If one of the devices is at fault they may change devices else they may report the BB. If correct they go to step 3.

6. **Cast.**

(a) The V enters their login credentials on the VT and a cast-code. The VT publishes the received ballot from the BB using this information on the BB.

(b) The VT and VA's display a message that a new ballot using a cast-code was published on the BB. If the ballot is not the same as the previous ballot they warn the V.

(c) The V enters their lock-in code on the VT. The VT publishes the ballot again using this lock-in code.

(d) The VT and VA's display a message that the final ballot was published on the BB.

7. **Tally.**

(a) The EA uses a re-encryption mix-net to obfuscate any relation between the published ballots and the newly re-encrypted permuted ballots after mixing. The EA generates proofs for correct mixing and re-encryption of the used mix-net and publishes them on the BB.

(b) The EA decrypts the re-encrypted permuted ballots revealing the decrypted ballots publicly on the BB.

(c) The EA invalidates any ballot that does not correctly encode a valid voting choice.

(d) The EA tallies all available valid votes and reveals the final tally on the BB.

2 Homomorphic tallying

Assumptions. All ballot encryptions are homomorphic and re-randomizable. The zero-knowledge proof belonging to the encrypted ballot can be adjusted to still be valid after re-randomization. This can be achieved using Groth-Sahai proofs.

The protocol.

1. **Election Setup.**

   (a) EA generates a public / private key pair and publishes the public key ($_{EA}$) on the BB.

   (b) R sends casting and lock-in codes to all eligible voters.

   (c) Voter's create login credentials with the bulletin board.

2. **Session Setup.**

   (a) VT generates a public / private key pair and publishes the public key on BB.

   (b) V enters a short string on the VT which is appended to the *sessionID*.

   (c) VT starts a session using a new *sessionID* appended with the short string.

   (d) VT displays the *sessionID* as well as a QR code containing a secret symmetric key k_{secret} and the *sessionID*.

   (e) V checks that the short string is appended to the *sessionID*.

   (f) V scans the QR code with the AVA and optionally with additional VAs.

   (g) VAs find the session corresponding to the *sessonID* on the BB, obtaining the public key of VT. The VAs display the received *sessionID*.

   (h) V checks that all the *sessionID*'s match with the original.

3. **Ballot-color-correspondence Setup.**

   (a) VT obtains the list of ballots from the BB.

   (b) VT encrypts all the ballots with the EA's public key ($\{B_i\}^1_{EA}$) and generates the needed zero-knowledge proofs.

   (c) VT generates a correspondence ($\{\{B_i\}^1_{EA}, c_i\}$) between a permutation of the list of encrypted ballots and a set of colors (c_i).

   (d) VT publishes the correspondence encrypted by the pre-shared secret symmetric key (k_{secret}).

4. **Ballot Choice.**

   (a) The AVA obtains the ballot-color-correspondence from the BB, decrypts it using the pre-shared key and display the colors to V.

   (b) The V chooses the color corresponding to the choice they want to make displayed on the VT.

   (c) The AVA re-randomizes the encryption of the chosen ballot ($\{B_i\}^2_{EA}$) and publishes it on the BB.

   (d) The BB re-randomizes the received ballot ($\{B_i\}^3_{EA}$) before making it publicly visible.

5. **Audit. (Optional)**

   (a) The V initiates auditing from either the VT or the AVA. The respective device sends a message to the BB that the ballot is used for auditing. The BB informs the other device of the audit.

   (b) The VT, AVA and BB reveal their randomness used for encryption or re-randomization respectively on the BB.

   (c) The VT and AVA compute the actual choice made by the V, check if all steps were done correctly and finally display the choice to the user. The BB is unable to learn the choice as the initial correspondence was encrypted with the pre-shared key.

   (d) The V checks if their choice was correctly recorded. If one of the devices is at fault they may change devices else they may report the BB. If correct they go to step 3.

6. **Cast.**

   (a) The V enters their login credentials on the VT and a cast-code. The VT publishes the received ballot from the BB using this information on the BB.

   (b) The VT and VA's display a message that a new ballot using a cast-code was published on the BB. If the ballot is not the same as the previous ballot they warn the V.

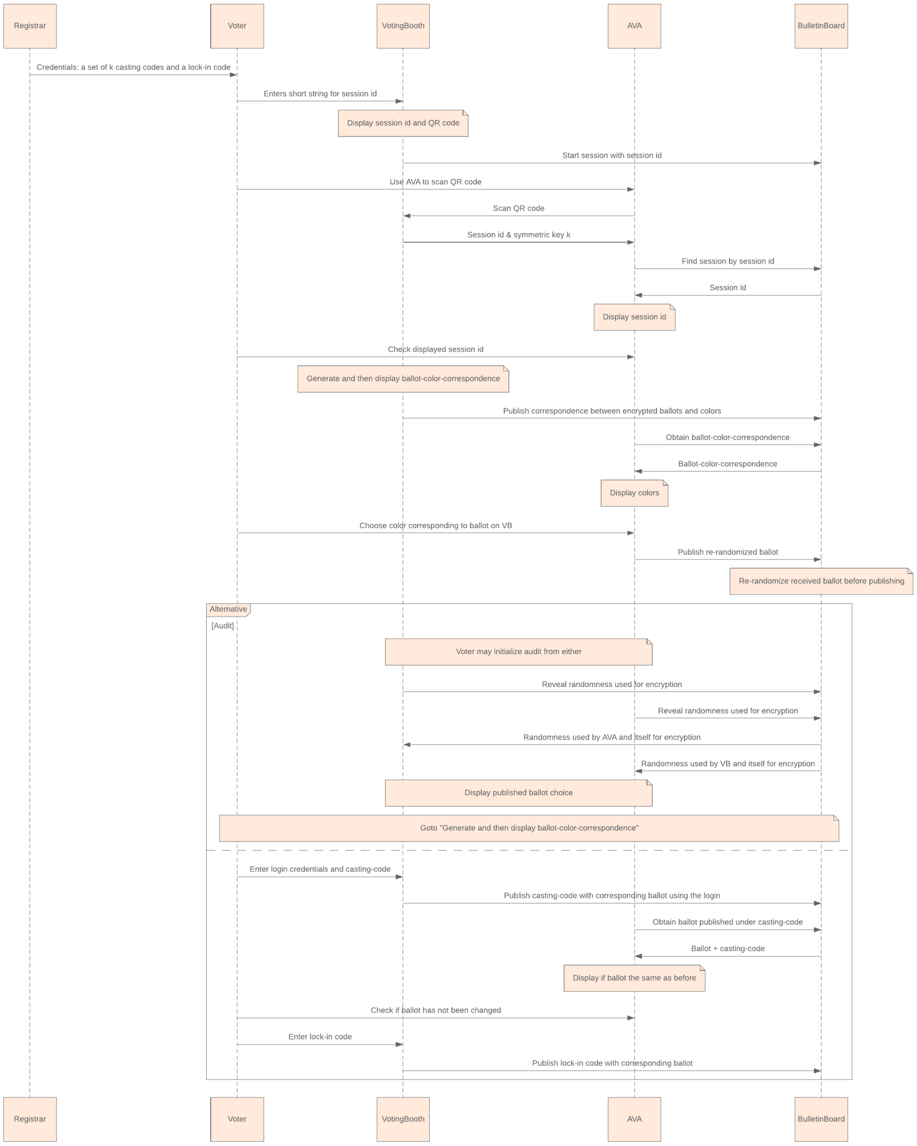
   (c) The V enters their lock-in code on the VT. The VT publishes the ballot again using this lock-in code.

   (d) The VT and VA's display a message that the final ballot was published on the BB.

7. **Tally.**

   (a) The EA gathers all locked-in ballots and sums or multiplies them together as they are encrypted using homomorphic encryption the result is a representation of the final tally.

   (b) The EA decrypts the final encrypted tally, providing proof of correct decryption and revealing the final verifiable count publicly.


Registrar
Voter
VotingBooth
AVA
BulletinBoard
Credentials: a set of k casting codes and a lock-in code
Enters short string for session id
Display session id and QR code
Start session with session id
Use AVA to scan QR code
Scan QR code
Session id & symmetric key k
Find session by session id
Session id
Display session id
Check displayed session id
Generate and then display ballot-color-correspondence
Publish correspondence between encrypted ballots and colors
Obtain ballot-color-correspondence
Ballot-color-correspondence
Display colors
Choose color corresponding to ballot on VB
Publish re-randomized ballot
Re-randomize received ballot before publishing
Alternative
[Audit]
Voter may initialize audit from either
Reveal randomness used for encryption
Reveal randomness used for encryption
Randomness used by AVA and itself for encryption
Randomness used by VB and itself for encryption
Display published ballot choice
Goto "Generate and then display ballot-color-correspondence"
Enter login credentials and casting-code
Publish casting-code with corresponding ballot using the login
Obtain ballot published under casting-code
Ballot + casting-code
Display if ballot the same as before
Check if ballot has not been changed
Enter lock-in code
Publish lock-in code with corresponding ballot
Registrar
Voter
VotingBooth
AVA
BulletinBoard