Overview

This project (accessible at https://github.com/andreadeoli/web-mpc) is a privacy-preserving online voting platform that uses secure multi-party computation. It considers a simple yes/no/abstain election. The administrator initiates the election with a title, a description, and an expiration time, then adds participants. Each willing participant submits a yes/no/abstain vote. After the election expires, the administrator is able to access the aggregate counts of yes/no/abstain votes, as well as the names and email addresses of participants who have submitted votes. Optionally, the administrator can distribute an email report on the election result through the platform. The implementation uses Web-MPC, a library for secure online data aggregation. For additional information on Web-MPC, visit the developer's website (https://multiparty.org/) or the GitHub repository (https://github.com/multiparty/web-mpc).
**Election Creation**

If running locally, the election creation page is accessible at localhost:8080/create. To initiate an election, the administrator must supply a session title, description, and submission deadline (EST).

After the administrator submits the necessary information, the platform provides a set of session details, including a unique election ID and password, both required for session management, and a private key, required to access the final result of the election. The platform automatically downloads the private key and a text file containing the election ID and password. Once the administrator confirms that they have saved the information in a secure location, they can proceed to session management.
Details about the session, including the title, description, submission deadline, election ID, password, and public key, are stored in a MongoDB collection called sessioninfos.
Election Management

If running locally, the election management page is accessible at localhost:8080/manage. To access management functionality, the administrator must first supply a valid election ID and password.

Afterwards, the administrator can add participants, view existing participants, access a submission history, and optionally stop or pause the election before the expiration deadline. To add participants, the administrator must supply a set of newline separated entries. Each participant must be specified by a name, followed by a comma, followed by an email address.
Upon submission, the platform trims additional whitespace and validates email addresses. If the administrator supplies invalid email addresses or specifies the participants using an invalid format, the administrator will receive an error modal and no emails will be sent. Otherwise, an email will be issued to each participant and will include the election title, description, deadline, and a button that links to the vote submission page with the participant’s unique ID.
A list of new participants, specified by names and email addresses, will appear. Details on each user, including session ID, user ID, name, and email address, are stored in a MongoDB collection called userkeys. If the moderator returns to the page at a later time, a list of existing participants, specified by names and email addresses, will appear at the bottom of the page.

Under the submission history section, the administrator can view timestamps of submitted votes. Information on who has voted, as well as the final election result, are only revealed at the end of the election. If necessary, the administrator can pause or stop the election prior to the specified endtime. Pausing the election temporarily prevents participants from submitting votes, while stopping the election prevents submission indefinitely.
If the administrator chooses to manually stop the election, a modal will prompt them to confirm the decision. Otherwise, the election will close automatically at the specified time. Every two seconds, the server checks whether there are any open sessions with deadlines that have passed. If so, the session status is updated and the UI changes to provide the moderator with a link to the unmasked result.
Vote Submission

If running locally, the vote submission page is accessible at localhost:8080. The participant must input the election ID and their unique participant code. If the participant accesses the page through the email, this information will be prefilled. To submit a vote, the participant must make their selection (yes, no, or abstain) and verify that all of the data is correct.

Once they submit their vote, they will receive a confirmation modal and their vote will populate the submission history.
If the participant decides to resubmit, their new vote will overwrite the previous one. The vote is secure under MPC using an additive scheme to determine the count of each vote type.

Additionally, the user ID of each participant that votes is encrypted using the session public key and added to a MongoDB collection called votingrecords. This collection is used during the unmasking process to determine the list of participants who submitted votes.
Result Access

To unmask the result, the administrator must drag and drop the private key and input the election ID and password.

Afterwards, the administrator will see a table that specifies the number of yes, no, and abstain votes, as well as a list of participants who voted, provided via name and email.
To gather the list of participants, encrypted user IDs are fetched from the votingrecords collection in the MongoDB database. On the client-side, these user IDs are decrypted and matched to entries in the userkeys collection. This allows the administrator to securely identify the names/emails of participants that voted. Optionally, the administrator can send information on the election result to all eligible participants through the platform. The website presents two options: sending only the counts of yes/no/abstain votes or sending both the counts and a list of participants who voted. If the administrator chooses to send an email report, the vote tallies and participant list are sent from the client to the server, which queries the userkeys collection to obtain all eligible participant email addresses and sends out the report.
Thank you for your participation in Test Election

The election results are as follows:
Number of Yes votes: 1
Number of No votes: 1
Number of Abstentions: 0

The specified participants submitted votes:
Andrea de Oliveira, andreadeoli@gmail.com
Jane Doe, andrea.doeliveira@yale.edu
Acknowledgements

I cannot express enough thanks for my advisor, Professor Joan Feigenbaum, who provided continuous guidance and support as I explored a new topic in Computer Science.

References

