Electronic Voting System with Blockchain

Abraham P. M.
B.Sc. Information Technology Student
Department of CS&IT, JAIN (Deemed-To-Be University)
abraham.marshal@hotmail.com

Dr. Felix M Philip
Assistant Professor
Department of CS&IT, JAIN (Deemed-To-Be University)
m.felix@jainuniversity.ac.in

Abstract

Electronic voting or E-voting, is the process by which votes can be cast through electronic means, such as an Electronic Voting Machine or voter's personal device. This study aims to create a reasonably secure and hassle-free system through which votes can be cast by users through their phones with relative ease as compared to traditional means. Voting takes place through an app on the device, in which the user signs in with their ID and PIN. This lets the user establish secure means of communication with the server by exchanging encryption keys and authorization tokens, which in turn lets the user and server authenticate secure using various factors of authentication, such as a Time-based OTP system, to verify the user's identity and let the cast their vote securely onto the server's private blockchain. On the server side there will be "nodes" and an "owner" server that share a private blockchain, which handle counting the votes, authentication, user creation and managing the voting system and its configuration. After all the authentication is done, the user can cast their vote in the final part of the process. This will be verified using a token generated separately on both client and server side. If it matches then the user will get an appropriate message confirming their vote being cast, and upon failure the user can contact the appropriate authority conducting the poll for further action.

Keywords: *Blockchain, Electronic Voting, Cryptography*

1. Introduction

The system proposed in this paper aims to address the issue of voting through electronic means not being a practical method. There various security and privacy concerns that exist in current electronic voting deployments. One such example would be the Estonian Voting System - Security experts had found many flaws that made the system unfit for deployment [4]. A system for electronic voting could become a huge target depending on the scale that it's deployed and lead to attacks from all kinds of entities. However, there are obvious benefits of such a system being deployed, that make it necessary for continued innovation in this topic. It could lead to much larger participation from the public, infrastructure costs could drop as all the verification is done through computers remotely, make it easier for voting to be conducted during situations where the voter cannot leave their homes, e.g., The COVID-19 Pandemic. Blockchain-based electronic voting is not considerably secure

as the only guarantee that the blockchain provides, is to prevent tampering on previous blocks. Therefore, the blockchain only plays a minor role to act as a database unlike certain other systems [3]. A public blockchain would be a much bigger issue, hence the blockchain used in this project is private or permissioned [2] and split across multiple nodes in a centralized system.

2. Literature Review

New South Wales iVote System: The iVote system was a complex system of many components, some managed by the NSWEC and other administrators. Registration could be done through telephone, over the internet or from a NSWEC computer polling place. The voting process involves initially registering the voter with one of the aforementioned methods, and receiving an 8-digit ID and 6-digit PIN. The voter can use this ID and PIN to login to the voting server or telephone voting system and cast their vote, upon which they will receive a 12-digit receipt number. The vote is encrypted on the client and sent to a voting server and forwarded to a separate verification service. The vote could also be verified through the verification service through telephone by entering the ID, PIN and receipt number or visit an online receipt service to query if their vote was included in the final count. [5]

Estonian Internet Voting System: The I-voting system in Estonia was the first country to allow for completely online voting using a national identity card. The card provided cryptographic functions which make certain attacks significantly harder. The system uses public key cryptography to provide of "double envelope" ballots. The outer envelope being the voter's signature through the voter's private key and the inner envelope being public key encryption to protect the secrecy of the ballot. These ballots are moved to a physical machine that decrypts and counts the votes. At the start of the election the voting application can be downloaded onto the voter's system through https://valimised.ee. The client contains a TLS certificate for the server and public key for encryption. The voting process begins with the client launching the application and inserting their ID card and entering their PIN. The server confirms the voter's eligibility and returns a list of candidates for their respective district. The voter selects a choice and enters their signing key PIN. This pads and encrypts the vote with the public key and signs it with the voter's private key, this encrypted vote is sent to the server. The voting is allowed multiple times as a defense against coercion. The vote can also be verified by an application provided by the election authority. The votes are tabulated after the election ends by going through each vote and validating and reverifying them. During a public counting session, the votes are stripped off the signatures and exported to make them anonymous. [4]

3. Materials and Methods

The system featured in the paper uses standard cryptography combined with a private blockchain to ensure secure and anonymous voting. The voter goes through a series of stages to authenticate and prove their identity which at the final stage allows them to cast their vote.

3.1. Development

The development of both the APIs were done in Python using the Flask micro web framework and some libraries for cryptography, database communication and blockchain interaction. The servers use MongoDB for storing temporary data, and one of the servers posing as the "owner" also stores voter information and poll configuration on a MySQL Database. The smart contract for the blockchain

interaction is written in Solidity for the Ethereum blockchain and interfaced through Brownie. The voting application was written in Flutter with Android as the intended system for deployment, partly due to technical constraints. The interface for configuration was written in ReactJS with libraries used for designing the interface.

3.2. System

The system consists of two REST APIs at the server's side, i.e., the Voting API and the Configuration API, these are both run on different ports in the server, as shown in Figure 1. The Voting API is for clients to communicate with to cast their vote, while the Configuration API, which is ideally locked into a private network, is for the configuration of the system for system administrators. Since the APIs are REST-based, any application that allows for Secure HTTP requests can directly communicate with them for their appropriate purposes.

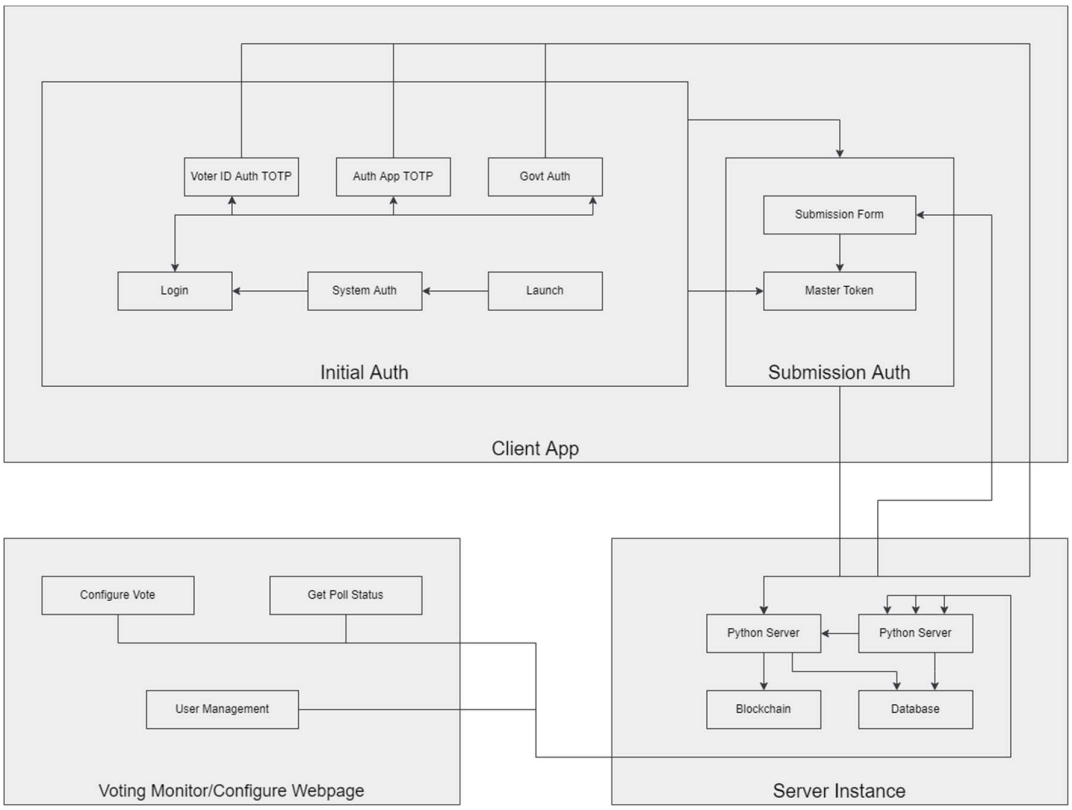

Figure 1. Voting System

3.3. Methods

The servers i.e., nodes will ideally be deployed at various locations connected by a secure private network. The locations these nodes are deployed at can also serve as registration centers for voters to come and register with the service. They will receive at ID, PIN and two Secret TOTP (Time-based One Time Password) generation secret QR Codes which they can scan. These codes should ideally be stored on two different devices, an intended use case would be for one code to be stored on the user's phone through the use of an authentication application, like Google Authenticator, while the other code would be stored on a dedicated device, preferably a device issued by the entity that is conducting

the polling, the device should be able to easily share the code share the code using RFID / NFC [1]. This process registers the user onto a MySQL database. The PIN is hashed and salted, and both TOTPs and the PIN are encrypted using the server's key.

After the voter's registration, they can cast their voting using the E-Vote Application through an Android Smartphone or some other application which can interface with the Voting API. The application should be able to fetch the address of the voting node from some trusted source which should be provided by the entity conducting the poll or hard-coded into the application.

The process starts by the voter signing in using their ID and PIN that was decided on registration. On signing in, a timestamp is stored to limit the time window for any potential attack, as the voter will need to sign in again if the vote is not cast before the expiry. The voter is also presented with all methods of authentication to complete one by one within the expiry, which includes the two TOTPs and also some form of Unique ID. The TOTPs are regenerated every 30 seconds on the respected devices and needs to be entered within that duration. The Unique ID can be a government ID or other verified ID which can be verified through a third-party API. If the verification is successful, then a response would be returned, allowing the system to confirm that the voter is an actual person.

Once the authentication process is done, the application lets the voter cast their vote through a form, fetched from the node. This sends a master token that gets generated using the tokens retrieved from all the authentication methods described above. The server also generates the same master token, as the server should have all the tokens stored as well. If the tokens match, then the vote is cast and the voter's ID is stored onto a temporary buffer list which is stored on "owner" machine. Once the list reaches a certain length, it is flushed into a map on the blockchain that marks the ID as having voted. This buffering is done to prevent any form of association with IDs and the vote count being incremented. The ID stored on the blockchain or in the temporary list will not be allowed to vote again. The vote can later be verified through an online interface, for instance, through the nodes, by checking for the existence of the ID of the voter on the blockchain [2].

All secretive data that is exchanged is encrypted with keys that are securely shared after signing in through envelope encryption. The system uses RSA-OAEP to handle key sharing. On logging in, the server generates an RSA Key with the client. This key is used to encrypt the AES keys that are shared during each authentication request. The server also ensures to encrypt all secretive permanent data on the server. However, HTTPS is still necessary to ensure that the initial login is completely secure and also as an added layer of security to the entire voting process.

3.4. Deployment

The servers are deployed as nodes with one of the nodes being the "owner" in a secure private network out of public access, with only the nodes' API endpoints being publicly visible. The "owner" server stores the MySQL Database for the voter's authentication data and for the poll's configuration. The table for voter information is modifiable by all nodes, however the configuration can only be read by other nodes while only the "owner" system can modify the table. Each server also has a MongoDB database for temporarily storing voter information while verifying.

Voters can connect to the system through any node's API endpoint depending on how the entity that is conducting the poll has decided to expose them. Ideally this would be through private infrastructure but should not pose any considerable risk if done through the internet.

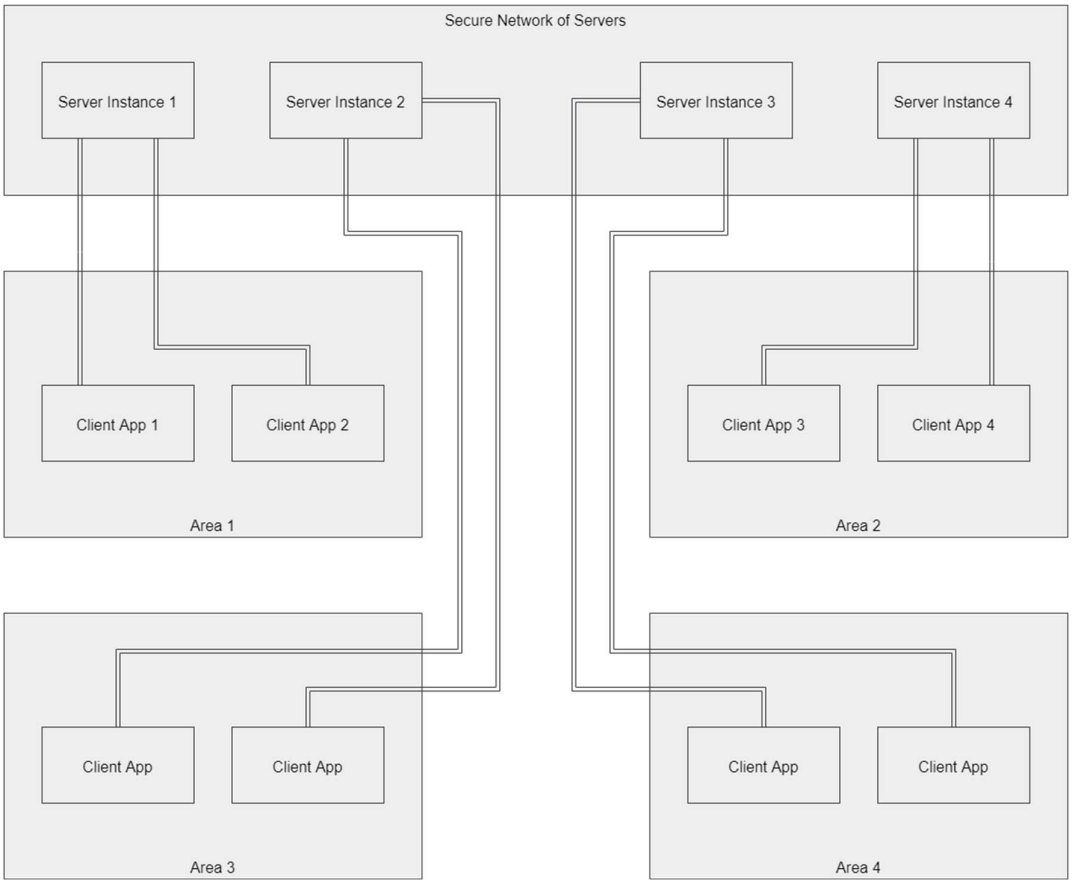

Figure 2. Deployment Example

4. Results and Discussion

The system is built with multiple failure points by using different factors of authentication to ensure that the voter is who they claim to be. And by limiting the time window along with using TOTPs, assuming that the authentication process itself isn't completely bypassed, should provide very little room for attacks. Though rigorous testing is yet to be done, and while the prototype still requires modifications to make it more secure, the system has held reasonably well in the minor testing that was conducted. The system itself is built as separate components and communicates with secure HTTP POST and GET requests. This makes it easy for any custom-made application to cast a vote with the server, provided that correct information is sent.

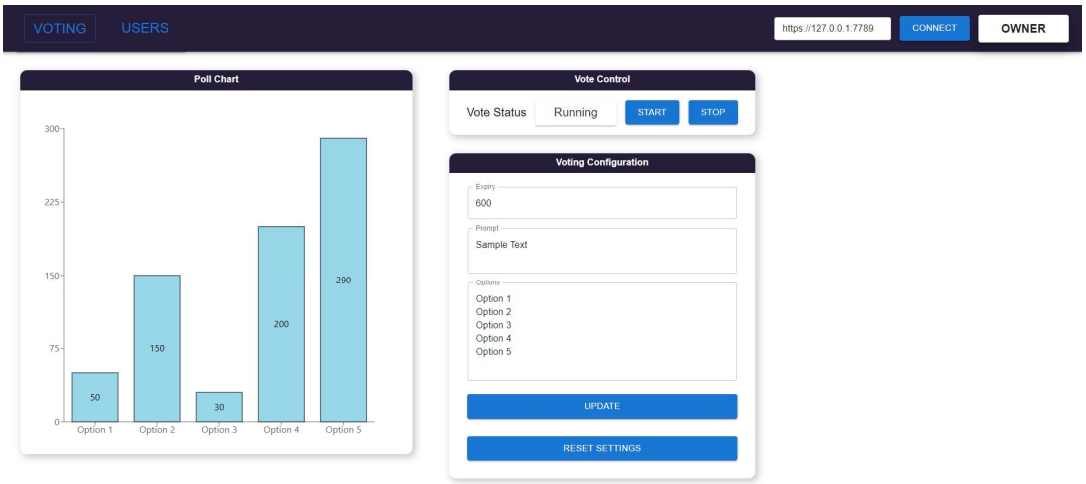

Figure 3. Voting Configuration Web UI (Owner)

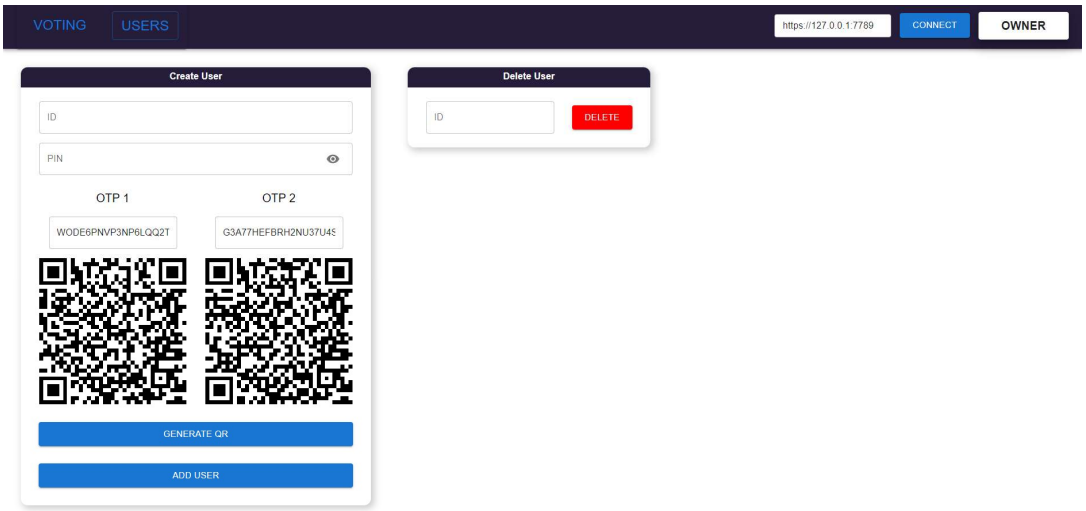

Figure 4. User Management Web UI

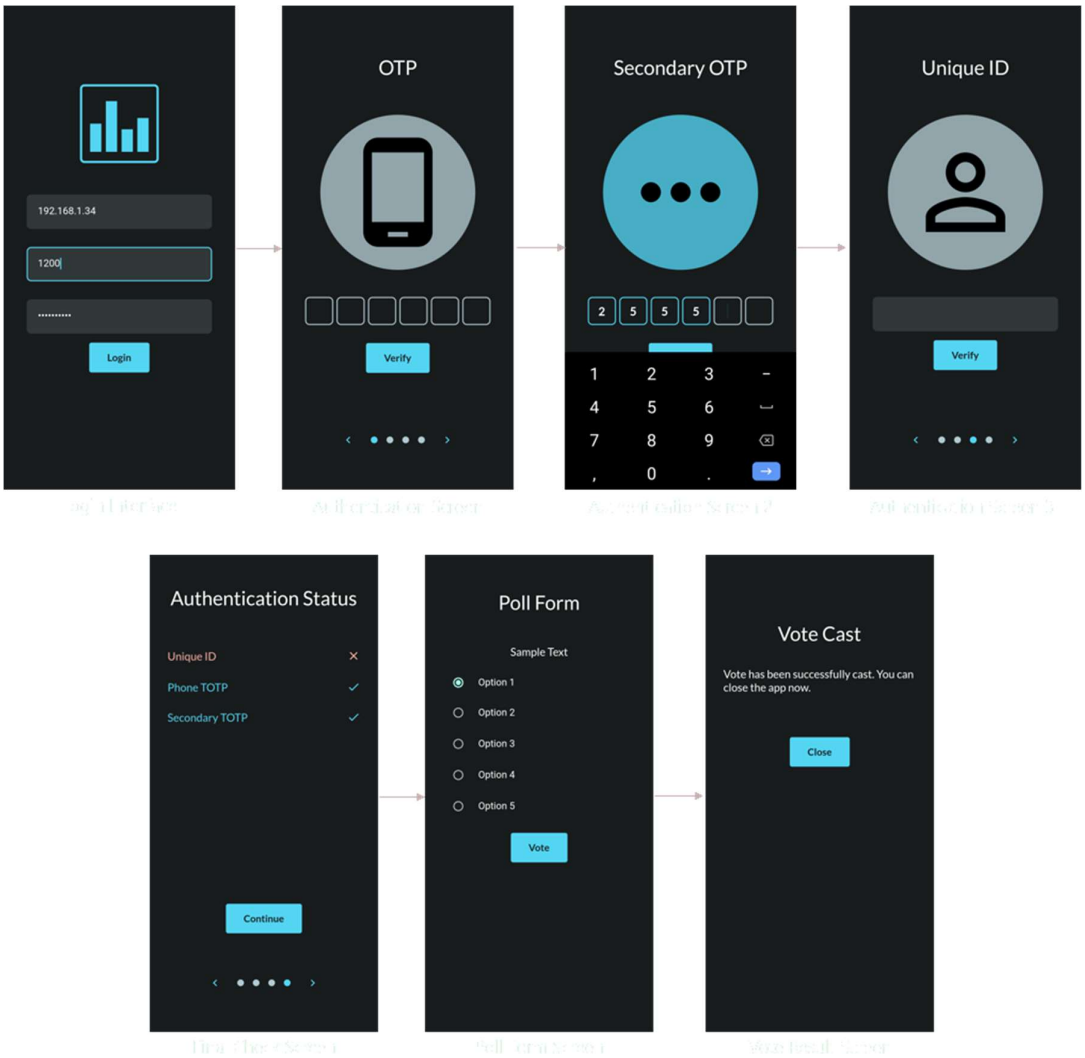

Figure 5. Voting Application Interface

5. Conclusion

The prototype of the system proposed in this paper works as intended in the limited testing that was done. Assuming that the application installed on the voter's device is trusted, which could be ensured by having the voter install the correct application on registration, can prevent attacks from installing untrusted applications that steal credentials or manipulate the process in any way. Disassociating the voter's votes from the voter themselves prevents the discovering who a voter has voted for. A simple and straightforward interface allows voters to easily traverse the application without assistance. Voter information is limited to just an ID, and all secretive information is encrypted, this ensures that even if the server's database is compromised, unless the key used to encrypt the data is also compromised, there is no risk of being able to impersonate a voter through the use of stolen information. The key can be made secure by storing the key on an external device or keeping the key out of the system and only using it when conducting a poll. The system should be much more secure after limiting login and authentication attempts, which will be added to the prototype in the future, will prevent any form brute forcing as well.

References

[1]. Hussien, H., & Aboelnaga, H. (2013, January). Design of a secured e-voting system. In 2013 International Conference on Computer Applications Technology (ICCAT) (pp. 1-5). IEEE.

[2]. Hjálmarsson, F. Þ., Hreiðarsson, G. K., Hamdaqa, M., & Hjálmtýsson, G. (2018, July). Blockchain-based e-voting system. In 2018 IEEE 11th international conference on cloud computing (CLOUD) (pp. 983-986). IEEE.

[3]. Ayed, A. B. (2017). A conceptual secure blockchain-based electronic voting system. International Journal of Network Security & Its Applications, 9(3), 01-09.

[4]. Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., & Halderman, J. A. (2014, November). Security analysis of the Estonian internet voting system. In Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (pp. 703-715).

[5]. Halderman, J. A., & Teague, V. (2015, September). The New South Wales iVote system: Security failures and verification flaws in a live online election. In International conference on e-voting and identity (pp. 35-53). Springer, Cham.