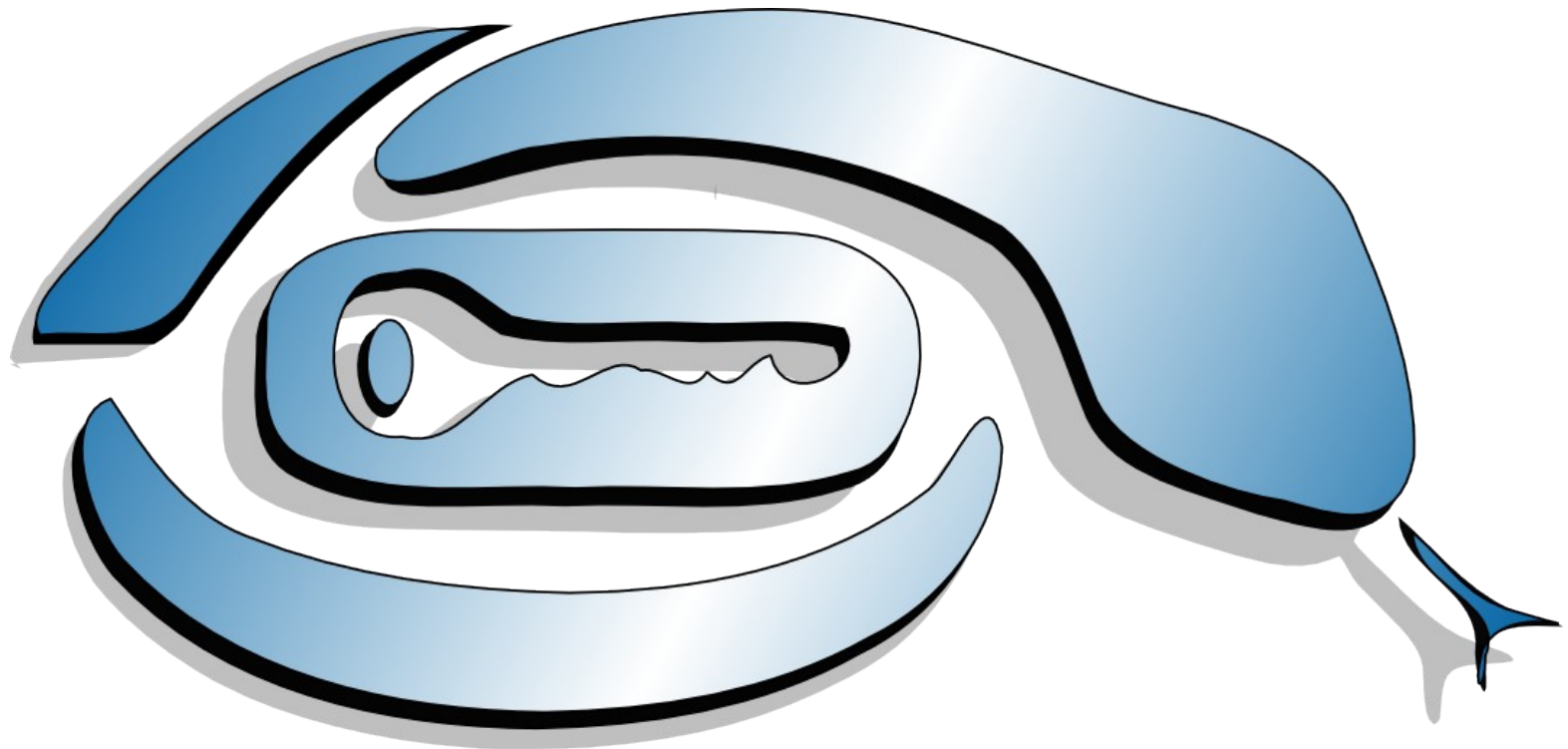


ADDER



The background of the image is a stylized American flag. It features a blue canton in the upper left corner filled with white stars, and the rest of the image is composed of horizontal stripes in red, white, and light blue. The text is overlaid on this background.

SECURE

ONLINE

VOTING

Presented by

Aggelos Kiayias
Michael Korman
David Walluck

University of Connecticut

For more information

<http://www.cse.uconn.edu/~adder>

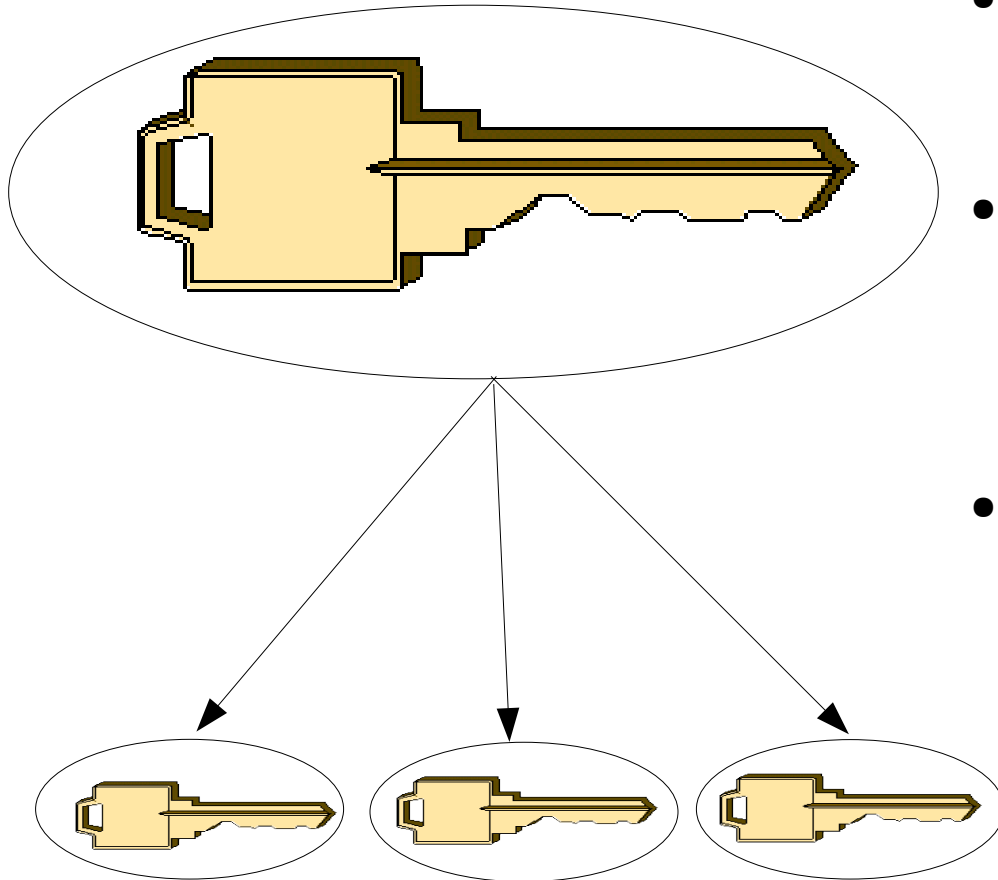
adder@cse.uconn.edu



Introduction

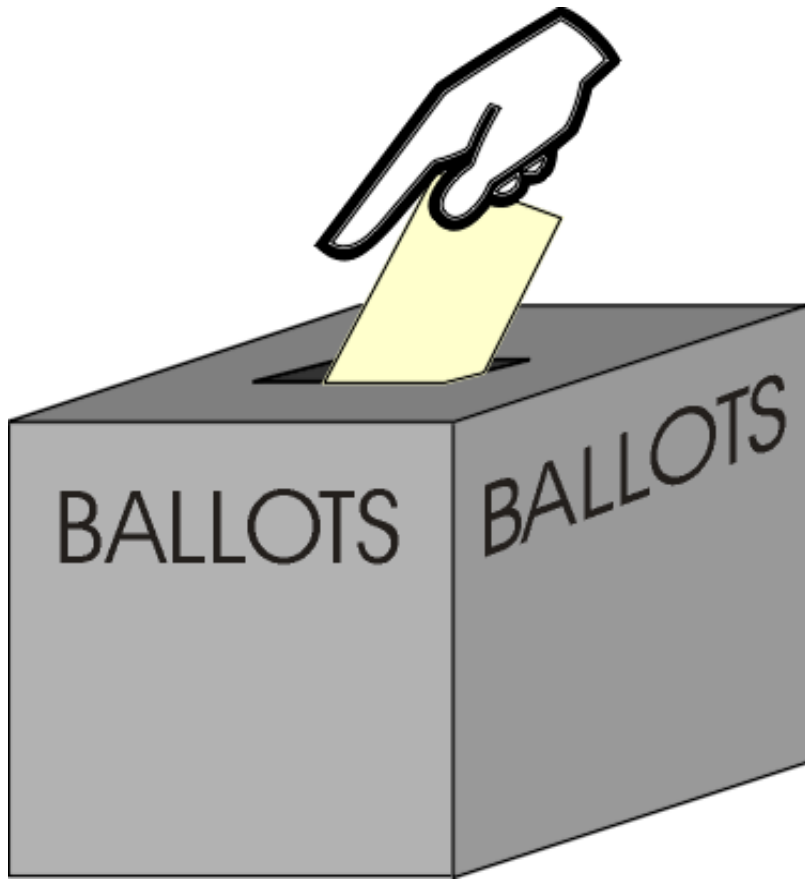
- Voting is a crucial element of democracy.
- Electronic voting systems must be secure against terrorist attacks.
- Systems must provide **robustness**, **auditability**, and **privacy**.
- **Adder** is a publicly available electronic voting system.

Stage 1: Key Generation



- **Authorities** log onto the system.
- They participate in a key-creation protocol.
- A **public encryption key** is created for the system, and **private decryption keys** are created for each authority.

Stage 2: Voting



- **Voters** log on to the system.
- Each voter encrypts a vote using the system's public key.
- Each vote is stored in a location designated for that particular user.

Stage 3: Vote Tallying



- All of the votes have now been submitted, and the election is closed.
- The server adds the encrypted votes, and published the encrypted total.

Stage 4: Sum Decryption



- Each authority downloads the encrypted total.
- Each authority decrypts the total and publishes this result.
- Once all authorities have submitted their decryptions, the server publishes the total.

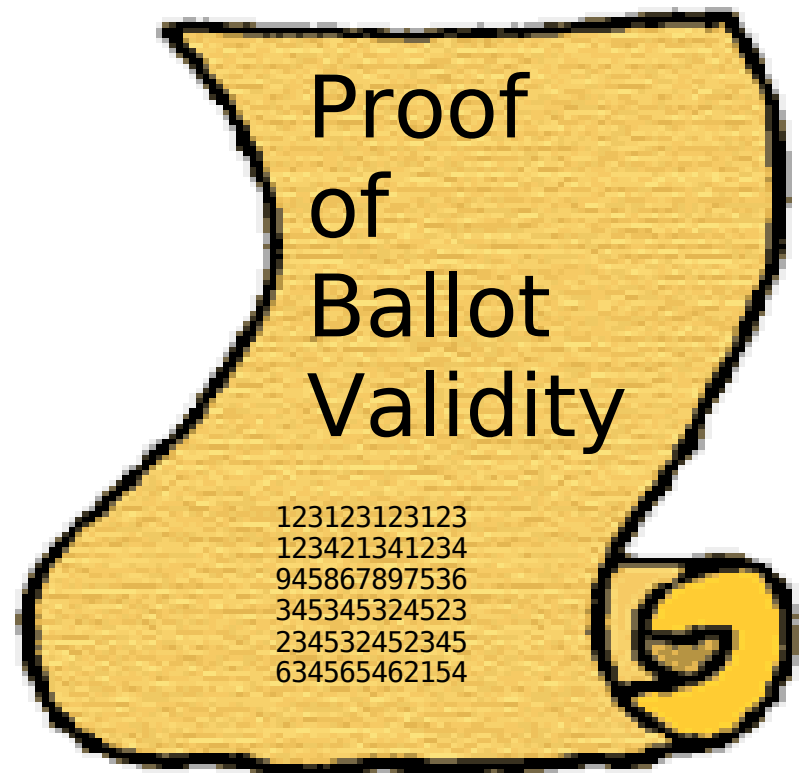
Zero-knowledge Proofs

- To prevent **ballot-stuffing**, when voters submit encrypted votes they also submit **proofs** that guarantee the vote is one of the allowed choices.

- Proofs are stored along with the vote in the database.
- If the user selects an option that is not an allowed choice, the server will recognize a faulty proof, return an error message and discard the vote.

Ballot

+



Homomorphic Encryption

- The server is able to add the votes while they are encrypted. This ensures that voters have complete privacy.
- Suppose we have one vote for candidate **A** and two votes for candidate **B**. The sum will be computed as follows:

$$\begin{aligned} & \text{encrypt}(1) + \text{encrypt}(0) + \text{encrypt}(1) \\ & = 123123 + 34508 + 234987 \\ & = 392618 \\ & = \text{encrypt}(2) \end{aligned}$$

Distributed Trust

- In order for an election to proceed, the number of participating authorities must be greater than or equal to a specified **threshold**.
- Decryption is handled jointly by the authorities. No one authority has the power to ruin the procedure.



Auditability

- **Adder** is designed to be completely **auditable**.
- All data present on the server are publicly viewable.
- Every computation performed by the server can thus be duplicated by an **independent auditor**.
- **Snapshots** can be made to ensure nothing is changed inappropriately, proofs can be checked, totals can be verified, etc.

Implementation

- **Main Server:** Written in C++ using the ACE networking library.
- **Libadder:** Cryptographic library written in C++ with the GMP multiple precision arithmetic library.
- **Client:** Java applet, and plugins for Mozilla and Internet Explorer.
- **Web Server:** Runs on Apache with PHP.
- **Database:** Currently uses MySQL.