An Internet Voting System Supporting User Privacy

Michael Korman
Joint work with Aggelos Kiayias and David Walluck

Computer Science & Engineering Department
University of Connecticut

December 12, 2006

Motivation

- Electronic voting is one of the most intensely debated subjects in information technology.
- It can be viewed as the holy grail of security!
- Compare this with other secure systems: ATMs, online credit card transactions, email, etc.
- E-voting systems in use have been shown inadequate.
- We are thus faced with the following questions.
  1. Should electronic voting be abandoned altogether?
  2. Can we build systems that are trustworthy?
  3. Under what conditions are electronic voting systems suitable?

In this talk, we introduce the ADDER system, an e-voting system designed to be run over the Internet.

Design & Security Goals

The following informal goals are desirable.

Transparency All data in the main server should be publicly accessible.

Universal Verifiability Any election result should be verifiable by any third party.

Privacy Individual votes should remain hidden. Only the final result is made public.

Distributed Trust Only a majority of authorities can obtain the final result.

The Need for Open-Source Voting Systems

- When the machinery used to manage an election runs inside a "black box," there is no way to verify the validity of the election.
- Current e-voting systems often do not have publicly available source code.
- Code may be concealed to avoid the discovery of embarrassing security flaws.
- Even proprietary systems that reveal source code often leave several critical components hidden.
- In order to ensure true democratic elections, voting software must be independently auditable and verifiable by any interested third party.

Cryptographic Tools

ADDER employs numerous cryptographic tools to achieve the security goals outlined previously.

- Homomorphic Elgamal encryption is used to protect voter privacy.
- Distribution of trust is used to ensure that no one party has access to the secret key of the election.
- Proofs of knowledge prevent voters from tampering with the election.

We will now discuss these tools, and explain how they work together to protect election integrity.

Simple Elections

In simple elections, voters have two options, 'yes' and 'no'.

Example

Suppose we have four voters:

Voter	Choice
V_1	0 (no)
V_2	1 (yes)
V_3	1 (yes)
V_4	1 (yes)

When the votes are tabulated, the result is $0 + 1 + 1 + 1 = 3$, which gives us the number of 'yes' votes. This, subtracted from the total number of votes, gives $4 - 3 = 1$, which is the number of 'no' votes.

Multi-way Elections

Multi-way elections require a more complex scheme.

Example

Imagine a three-way election, and four voters:

Voter	Choice
V_1	010 (candidate B)
V_2	100 (candidate C)
V_3	001 (candidate A)
V_4	001 (candidate A)

Tabulation is base-M addition, where M is greater than the number of voters. The result is $010 + 100 + 001 + 001 = 112$. Each position gives us the number of votes for each candidate: 2 for A, 1 for B, and 1 for C. A is the winner.

Homomorphic Encryption

- Privacy of voters is crucial; it should not be possible to determine any voter's ballot.
- ADDER uses homomorphic encryption to achieve ballot privacy:

$$\mathcal{E}(v_1, r_1) \oplus \mathcal{E}(v_2, r_2) = \mathcal{E}(v_1 + v_2, r_1 \otimes r_2).$$

- Thus, it is possible to add votes without ever revealing any individual ballot.

Bulletin Board Model

All communication in ADDER is done through a public bulletin board (introduced by Benaloh).

- Each user in the system has an area of memory in the bulletin board that it can write to.
- Access is limited to read and append.
- Every location in the bulletin board is readable by all users.
- To send a message from one user to another, the sender posts a message in the bulletin board, and the receiver reads it.

Thus, all communication is done in the open, and a public transcript can be maintained.

Distribution of Trust

ADDER uses a multi-authority system. Two sets of users (not necessarily disjoint): voters and authorities.
The secret key of the election must be distributed among the authorities. There are two problems:

- We do not trust any of the authorities, so we cannot hand out a complete copy of the key to each one.
- We do not want to give any authority the power to disrupt the procedure, so we cannot divide the key such that all authorities must be present to use it.

Thus, we need a way to break up the key such that only a certain threshold of shares must be present, while anything fewer than that is completely unusable.
ADDER uses secret sharing schemes based on polynomial interpolation to achieve this.

Ballot-stuffing

Let us look at a simple yes/no election scheme to see a hazard.

Example

Voter	Choice
V_1	0 (no)
V_2	0 (no)
V_3	1 (yes)
V_4	0 (no)
V_5	4 (???)

V_5 has an bogus ballot, but this will never be known, as it is encrypted. When tabulation occurs, there will be 5 'yes' votes, and 0 'no' votes!

Zero-knowledge Proofs

- We use zero-knowledge proofs to prevent ballot-stuffing.
- Each voter submits a proof that his vote is one of the given options.
- If a proof is rejected, the vote is refused.
- Authorities also use zero-knowledge proofs during the construction of the election secret key.

Stages of an ADDER Procedure

1. Administrator selects election parameters and creates voting procedure.
2. Authorities create public and private keys and publish their public keys.
3. Authorities jointly generate the election secret key, and distribute shares of this key among themselves.
4. Public key derived from the secret shares of each authority is published.
5. Voters cast their votes, encrypted with the public key.
6. The results are tabulated in public view, using the homomorphic property of the encryption.
7. Authorities each partially decrypt the encrypted sum.
8. Server combines the partial decryptions and publishes the result.

Electronic Voting Concerns

After the major controversies with electronic voting, several scientists came out harshly against current practices. We describe how ADDER addresses the most common concerns:

- User authentication.
- Ballot privacy.
- Universal verifiability.
- Voter verifiability.

User Authentication

- Authentication is performed by a Kerberos-like system called the gatekeeper.
- Gatekeeper has private table of users and credentials.
- The gatekeeper gives signed tickets to users to perform actions: `create procedure`, `delete procedure`, `reset procedure`, `vote`, and various authority actions.
- Bulletin board verifies gatekeeper's signature, and performs the requested action.
- Separation of trust is thus achieved, while being transparent to the user.

User Authentication, cont.

The question remains: how is the private gatekeeper user table populated?

- The administrator of the gatekeeper provides a list of names and email addresses.
- Gatekeeper then generates random passwords and emails credentials to users.
- Or, printed letters can be produced, and mailed to users.
- If anonymity is desired, identities can be assigned randomly.

Ballot Privacy

- The homomorphic encryption property previously described ensures voter privacy.
- Encrypted votes can be stored in plain view.
- Summation can be performed without knowledge of any secret key, as long as one knows the homomorphic operation to be performed.
- Thus, any third party can perform summation, and leave the decryption of the final total to the authorities.

Universal Verifiability

ADDER is designed to be universally verifiable.

- All data on the bulletin board are publicly viewable.
- The bulletin board has a special read-only account designed for auditors.
- A free and open source verification suite accomplishes:
  1. Tallying encrypted ballots.
  2. Verification of proofs.
  3. Decryption of the final tally.
  4. Verification of the hash chain.

Voter Verifiability

Voter verifiability is a common concern for electronic voting systems.

- Many direct recording electronic (DRE) systems use a voter verifiable paper audit trail (VVPAT).
- VVPAT has become a keyword in electronic voting over the past few years.
- Indeed, the presence of a paper audit trail is now law in several states.

The focus of ADDER is on providing a cryptographic solution to voting. VVPAT makes no sense in remote Internet voting, and we feel voter verifiability is an important, but orthogonal concern.

Ideal Technical Goals

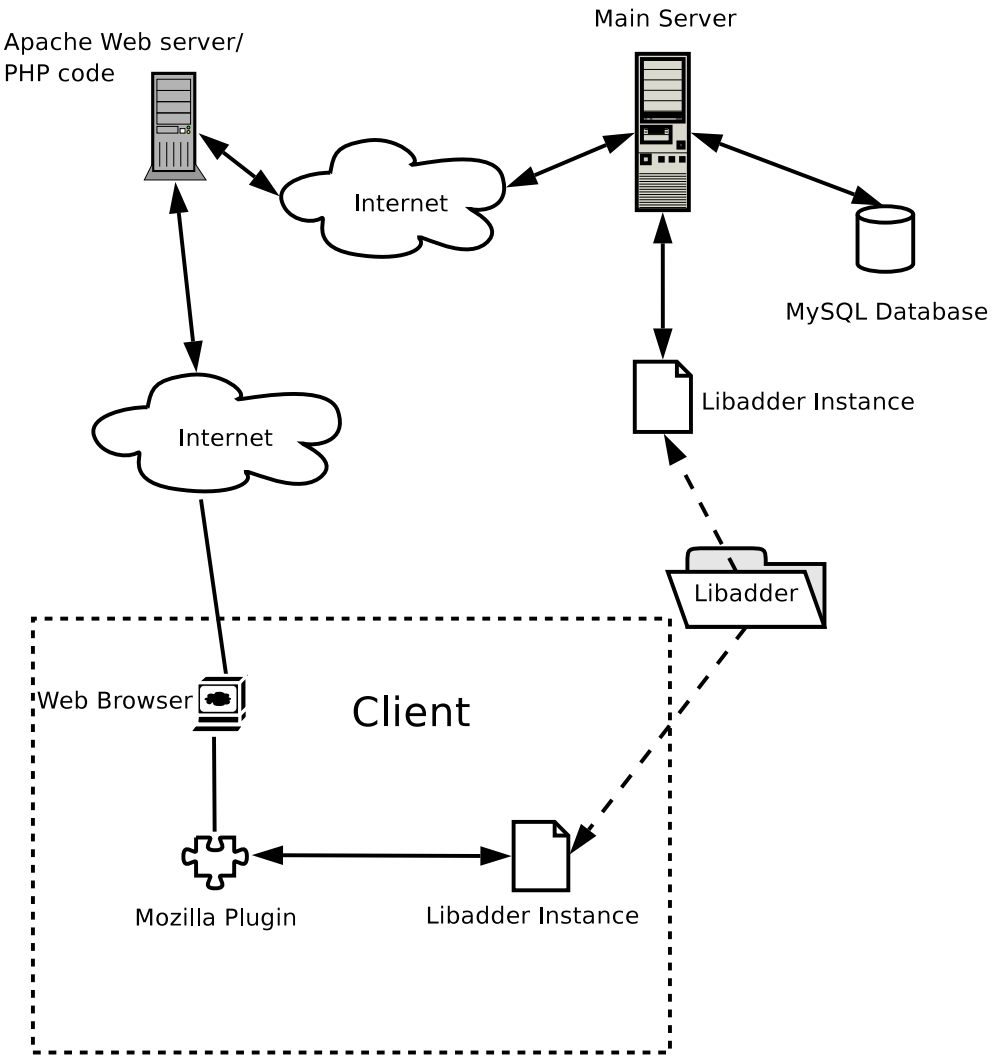
We designed ADDER with several goals in mind.

- Easy-to-use Web interface.
- Cross-platform.
- Must be secure on a standard PC.
- Fault-tolerant database and network.

Components of ADDER

- Cryptographic library.
  - C++ (with GMP) and Java
  - Shared by applet/plugin and main server.
- Java applet/browser plugin.
  - Plugins for Mozilla and IE.
- Qt GUI/Verification suite.
- Main server.
  - Written in C++ (with ACE).
  - Keeps track of running elections.
- Web server.
  - PHP on Apache.
- Database.
  - MySQL.

High-Level Diagram



Future Directions

ADDER makes good progress in the state-of-the-art of cryptographic voting systems, but there are several issues that need to be addressed:

- Transcript robustness
  - Possible solution: database replication.
- Vote buying and coercion
  - Possible solution: ciphertext re-randomization.
- Voter verifiability.
  - Possible solution: ???
- Viruses and other client-environment hazards.
  - Possible solution: custom boot CD-ROM.
- Denial-of-service attacks.
  - Possible solution: puzzle auctions.
- User interaction.

Further Information

- We encourage you to visit the ADDER Web site: `http://cryptodrm.engr.uconn.edu/adder/`
- Emails may be sent to `adder@cse.uconn.edu`.

Questions? Comments?