Programmazione Avanzata 2008-2009

Civitas: un sistema sicuro per il voto elettronico

(M.R.Clarkson, S.Chong, A.C.Myers)

Silvia Messuti

Electronic Voting Systems

Hard to make trustworthy because of strong conflicting security requirements

- **Integrity**
  - All votes are counted correctly
  - Detect and correctly attribute any attempt to corrupt the election
- **Confidentiality**
  - Protect voters' privacy
  - Prevent selling of votes
  - Defend voters from coercion
- **Availability**

Civitas

Security properties

- Verifiability (integrity)
- Coercion resistance (confidentiality)

Assurance due to

- Security proofs
  - Based on JCJ scheme – satisfies formal definitions
- Secure information flow
  - Implementation in Jif

Tradeoffs between the level of security, the time required for tabulation and the monetary cost of the election

Civitas
Security Requirements

Security Model

Remote voting vs Supervised voting

- Voting could take place anywhere
- Trusted human supervision of the voters, procedures, hardware and software in polling places

Compromise between enabling remote voting and guaranteeing strong security properties

- In some circumstances voters must register at least partly in person
- Voters must trust the computational device they use to submit votes (not necessarily supplied by election authority)

Verifiability

The final tally is verifiably correct

- Voter verifiability
  - Anyone can check that their own vote is included in the tally
- Universal verifiability
  - Anyone can check that:
    - All votes cast are counted
    - Only authorized votes are counted
    - No votes are changed during counting

Coercion Resistance

Voters cannot prove whether or how they voted, even if they collude or interact with the adversary while voting

- Coercer can demand any behavior, remotely or in phisical presence of voters
- Coercer can observe and interact with voter during remote voting

Must prevent coercers from trusting their own observations

Threat Model

The adversary

- May corrupt a threshold of the election authorities
- May coerce voters (previous slide)
  - but may not control a voter throughout an entire election, otherwise the voter could never register or vote
- May control all public channel on the network
  - but we assume the existence of some anonymous channels on which the adversary cannot identify the sender
  - and some untappable channels which the adversary cannot use at all
- May perform any polynomial-time computation

Civitas
Design and Implementation

Agents

- The **supervisor** administers an election
  - Specifies the ballot design and the tellers
  - Starts and stops the election
- The **registrar** authorizes voters
- **Registration tellers** generate the credentials that voters use to cast their votes
- **Voters**
- **Tabulation tellers** tally votes

Log Service

- Publicly readable
- Insert-only
- Integrity ensured by digital signatures
  - Agents sign inserted messages → the log service cannot forge new messages
  - The log service signs its responses to reads → detection of attempts to present different contents to different readers

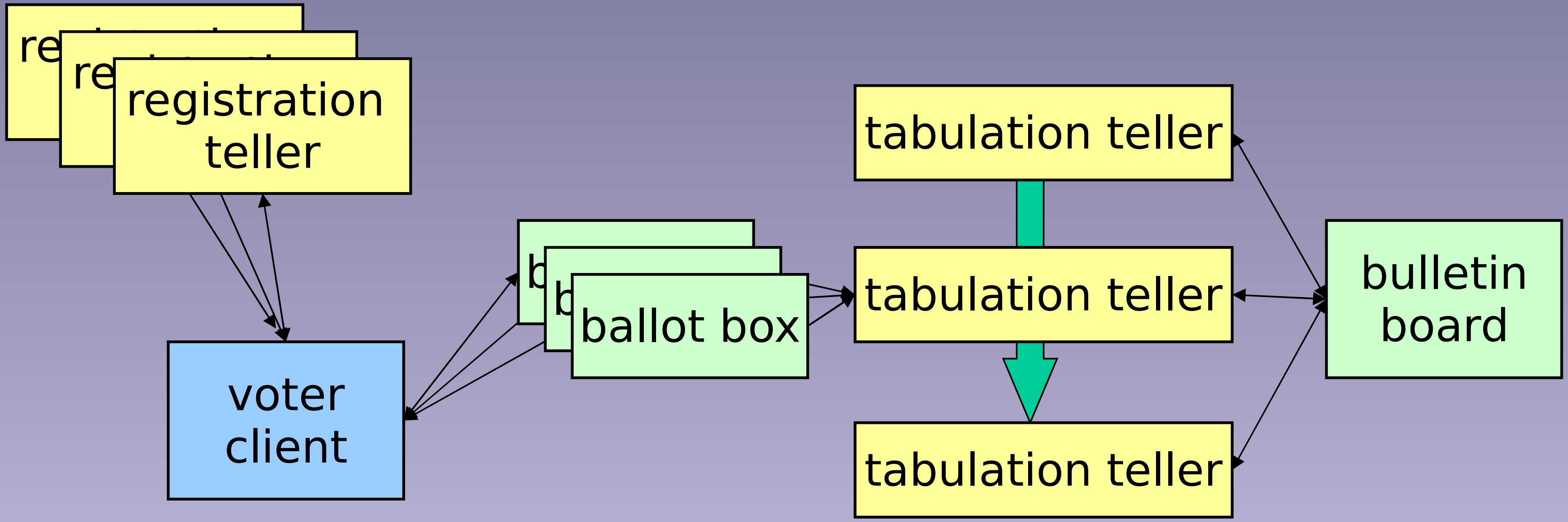
Bulletin board

- used by election authorities to record all the information needed for verifiability of the election

Ballot boxes

- used by voters to cast their votes

Civitas Architecture



Setup

The supervisor

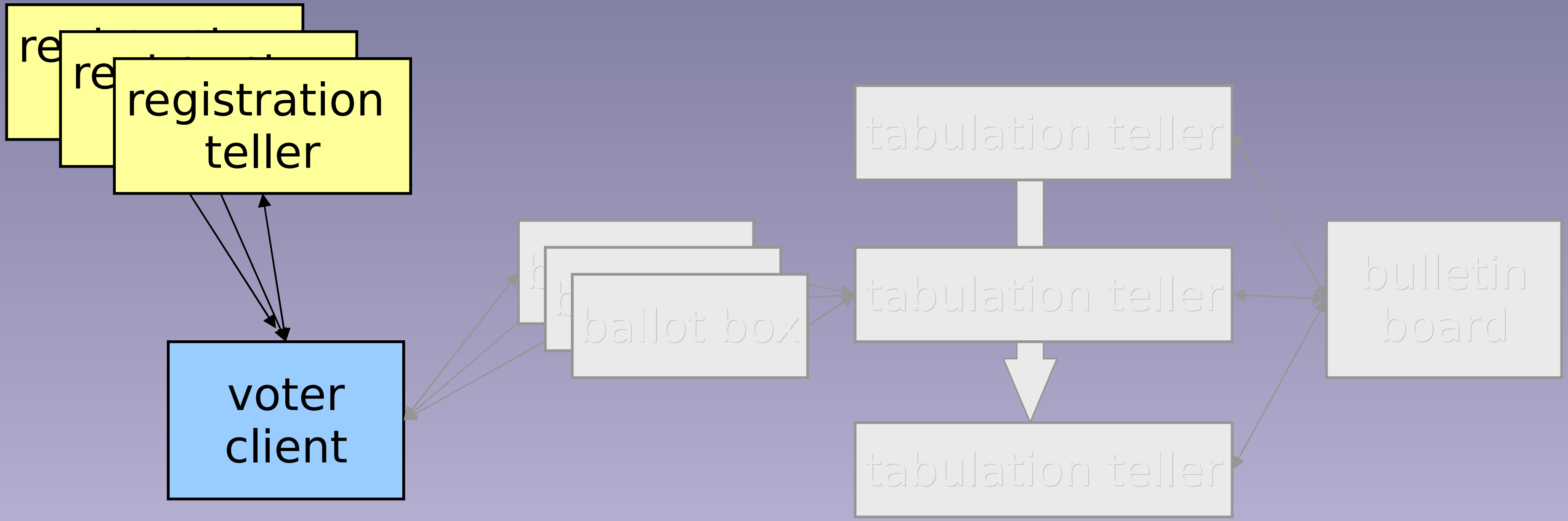
- creates the election by posting the ballot design on an empty bulletin board
- Identifies the tellers by posting their individual public keys

The registrar posts the electoral roll, containing identifiers and public keys for all authorized voters

Registration tellers

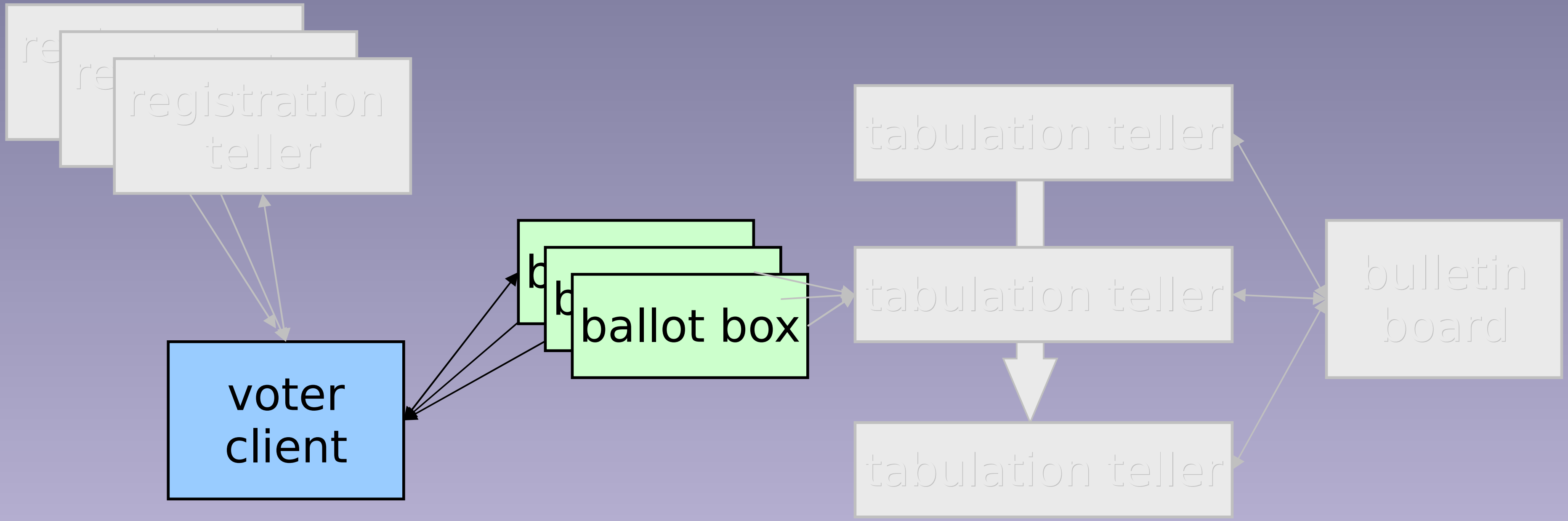
- Generate credentials for each voter
  - Public credentials are posted on the bulletin board
  - Private credentials share are stored
    - Can be forged or leaked only if all registration tellers collude

Registration



- Each registration teller authenticates a voter using the voter's registration key
- The teller releases his share of the voter's private credential to the voter, using the voter's designation key
- The voter combines shares to construct a private credential

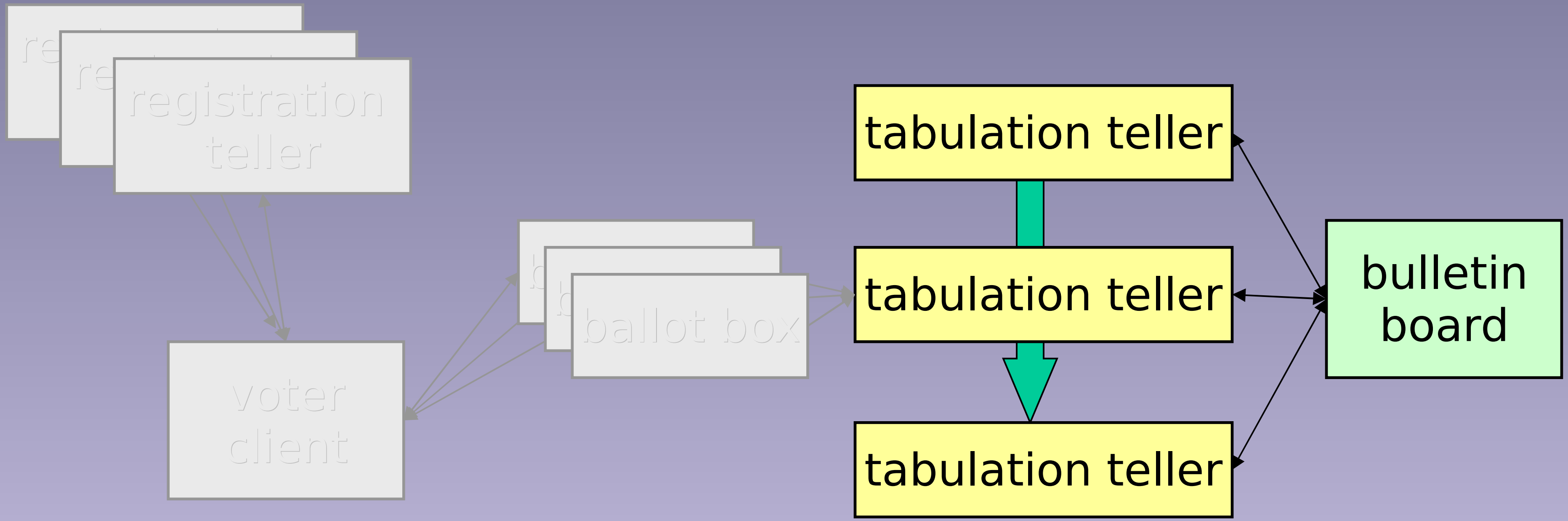
Voting



The voter submits to some or all of the ballot boxes

- A private credential
- A choice of a candidate
- A proof that the vote is well formed

Tabulation



Tabulation tellers

- Retrieve the votes and the public credentials from the bulletin board, eliminate duplicate and unauthorized votes
- Anonymize votes with mix network
- Decrypt the remaining choices (but not credentials)

Resisting Coercion

- Voters can substitute fake credentials for thei real credentials
  - Indistinguishable to an adversary
  - The faking algorithm requires the voter's private designation key
- Voters with fake credentials removed during tabulation
- Voters can vote again with real credentials

Verifying an election

- Tabulation tellers post proofs that they are honestly following the protocols
- An honest teller refuses to continue when it discovers an invalid proof
- Anyone can verify these proofs during and after tabulation (universal verifiability)
- A voter can verify that his vote is present in the set retrieved by the tabulation tellers (voter verifiability)

Secure Implementation

Jif (Myers 1999, Chong and Myers 2005, 2008)

- Security-typed language
- Types contain information-flow policies
  - Confidentiality, integrity

Jif$_E$ extends Jif with declassification and erasure policies

- Allow principals to state conditions on when the set of readers in a confidentiality policy may be expanded (declassification) or restricted (erasure)

If policies in code express correct requirements and Jif compiler is correct then **code is secure w.r.t. requirements**

Cryptographic Components

- RSA
- El Gamal
  - Distributed, non malleable
- Zero-knowledge proofs
  - Knowledge of a discrete logarithm
  - Equality of discrete logarithms
  - Designated-verified reencryption proof
  - Plaintext equivalence test
- Commitments
- Digital Signature
- Mix network

Civitas
Security Evaluation

Trust Assumptions

1. The adversary cannot simulate a voter during registration
2. Each voter trusts at least one registration teller and the channel from the voter to the trusted authority is untappable
3. Voters trust their voting clients
4. The channels on which voters cast their votes are anonymous
5. At least one of the ballot boxes to which a voter submits his vote is correct
6. There exist at least one honest tabulation teller
7. The Decision Diffie-Hellman (DDH) and RSA assumptions hold and SHA-256 implements a random oracle

Trust Assumptions

1. The adversary cannot simulate a voter during registration
2. Each voter trusts at least one registration teller and the channel from the voter to the trusted authority is untappable
3. Voters trust their voting clients
4. The channels on which voters cast their votes are anonymous
5. At least one of the ballot boxes to which a voter submits his vote is correct
6. There exist at least one honest tabulation teller
7. The Decision Diffie-Hellman (DDH) and RSA assumptions hold and SHA-256 implements a random oracle

Trust Assumptions

1. The adversary cannot simulate a voter during registration
2. Each voter trusts at least one registration teller and the channel from the voter to the trusted authority is untappable
3. Voters trust their voting clients
4. The channels on which voters cast their votes are anonymous
5. At least one of the ballot boxes to which a voter submits his vote is correct
6. There exist at least one honest tabulation teller
7. The Decision Diffie-Hellman (DDH) and RSA assumptions hold and SHA-256 implements a random oracle

Trust Assumptions

1. The adversary cannot simulate a voter during registration
2. Each voter trusts at least one registration teller and the channel from the voter to the trusted authority is untappable
3. Voters trust their voting clients
4. The channels on which voters cast their votes are anonymous
5. At least one of the ballot boxes to which a voter submits his vote is correct
6. There exist at least one honest tabulation teller
7. The Decision Diffie-Hellman (DDH) and RSA assumptions hold and SHA-256 implements a random oracle

Trust Assumptions

1. The adversary cannot simulate a voter during registration
2. Each voter trusts at least one registration teller and the channel from the voter to the trusted authority is untappable
3. Voters trust their voting clients
4. The channels on which voters cast their votes are anonymous
5. At least one of the ballot boxes to which a voter submits his vote is correct
6. There exist at least one honest tabulation teller
7. The Decision Diffie-Hellman (DDH) and RSA assumptions hold and SHA-256 implements a random oracle

Trust Assumptions

1. The adversary cannot simulate a voter during registration
2. Each voter trusts at least one registration teller and the channel from the voter to the trusted authority is untappable
3. Voters trust their voting clients
4. The channels on which voters cast their votes are anonymous
5. At least one of the ballot boxes to which a voter submits his vote is correct
6. There exist at least one honest tabulation teller
7. The Decision Diffie-Hellman (DDH) and RSA assumptions hold and SHA-256 implements a random oracle

Trust Assumptions

1. The adversary cannot simulate a voter during registration
2. Each voter trusts at least one registration teller and the channel from the voter to the trusted authority is untappable
3. Voters trust their voting clients
4. The channels on which voters cast their votes are anonymous
5. At least one of the ballot boxes to which a voter submits his vote is correct
6. There exist at least one honest tabulation teller
7. The Decision Diffie-Hellman (DDH) and RSA assumptions hold and SHA-256 implements a random oracle

Civitas
Cost Evaluation

Scalability

- Elimination of duplicate and invalid credentials take quadratic time
- Tabulation requires each teller to perform computation for each vote

Group voters into blocks

- The tally for each block can be computed independently (parallelizable)
- Each vote identifies in plaintext the block in which its credential resides (identifier made non-malleable by vote proof)
- Assignment into blocks need not be based on physical location
- Enables the production of early returns

Cost

Current real-world **total** cost is \$1-\$3/voter

Total cost for Civitas unknown

CPU cost for tabulation (for reasonable security parameters)

- CPU time: 39s/voter/authority
- If CPUs are bought, used (for 5 hours), then thrown away: \$1500/machine, \$12/voter
- If CPUs are rented: \$1/machine/hour, 4c/voter

Increased cost → increased security

Conclusion

Summary

Security

- Verifiability
- Coercion resistance

Assurance

- Security proofs
- Explicit trust assumptions
- Implementation in Jif

Open Issues

- Threshold cryptography
- Application-level denial of service
- Recovery of lost credentials
- Usability vs Security
- Distribute trust in voter client
- Eliminate in-person registration
- Acceptability of cryptography
- Access to computers