Universally Verifiable Efficient Re-encryption Mixnet
July 2010

Jordi Puiggali
VP Research & Development
Jordi.Puiggali@scytl.com
• Auditability in e-voting
• Universal verifiable Mix-nets
• Building blocks
• Proposal description
• Properties
• Conclusions
Voter

Observers / auditors

Electoral board

Results

- Votes and processes (e.g., counting) are based on tangible elements
  - Audit can be done by observers and independents auditors by human means when the processes are carried out
  - Observers can monitor the behavior of other observers to detect any fraud practices
Auditability in e-voting

- Votes and processes are happening in a logical dimension
  - Audit cannot be done by human means
  - Difficult to monitor the behavior of other observers
Audit processes in e-voting

- Source code review/certification
  - Audit process done in advance
  - Does not allow to verify what is exactly happening when the voting platform is executed
- Audit logs
  - Allows to trace what is happening during the execution of the voting platform
  - Could be tampered with by the voting platform during the execution of the voting process
- Monitoring processes
  - Allows to monitor the behavior of the voting platform
  - Intrusive: Who monitors this processes? (Who watches the watchers?)

Is there any way to verify the behavior of the counting process when it is executed without being intrusive?

Solution: Universal verifiable cryptographic protocols
Objective
- Audit process based on the input and output data of the counting process
  - Inputs:
    - Encrypted votes
  - Outputs
    - Decrypted votes
    - Cryptographic proofs of correct behavior of the cryptographic processes (e.g., Zero Knowledge Proofs)
- Audit process shall preserve the privacy of voters and integrity of the election

Universal verifiable protocols
- Homomorphic tally
  - Flexibility problems: does not support all kinds of vote formats
  - Scalability problems: the amount of cryptographic operations depends on the number of candidates
- Universal verifiable Mix-nets
  - Does not have limitations on the format of the vote
  - More efficient for large candidate lists
• Auditability in e-voting
• Universal verifiable Mix-nets
• Building blocks
• Proposal description
• Properties
• Conclusions
Objectives

- Correctness
  - Probability of detecting a bad behavior in the Mixing process
- Voter privacy
  - Prevents the correlation between encrypted votes and decrypted ones
- Efficiency
  - Reduces the amount of cryptographic operations required to generate and verify the cryptographic proofs

Current proposals

- Sako and Kilian [SK95], Furukawa and Sako [FS01], and Neff [Ne01]
  - High correctness and preserve voter privacy
  - Low efficiency for large elections
- Random Partial Checking [JJR02]
  - Trade-offs correctness and some voter privacy to improve efficiency
- Optimistic Mixing [Go02]
  - Preserves voter privacy by sacrificing some correctness to improve efficiency.

Our proposal tries to solve the handicaps of Random Partial Checking and Optimistic Mixing proposals.
• Auditability in e-voting
• Universal verifiable Mix-nets
• Building blocks
• Proposal description
• Properties
• Conclusions
Building blocks from our proposal

- Re-encryption Mixnet
  - Preserves voter privacy in the decryption process

- Homomorphic operation of the votes
  - Allows to generate integrity proofs of sets of encrypted votes without knowing the contents

- Zero Knowledge proofs of:
  - Re-encryption
    - Allows to verify that the result of the re-encryption preserves the original encrypted vote without disclosing it.
  - Correct decryption
    - Allows to verify that a decrypted votes correspond to an encrypted one without disclosing the private key and randomization factor
• Re-encryption Mixing:
  – It uses the homomorphic and probabilistic properties of some algorithms: re-encrypting one vote with the same public key does not require multiple decryptions of the vote with the private key, only one.
  \[ c = c'. (1. h^{w''}, g^{w''}) = (m'.h^{w'}, g^{w'}) . (1. h^{w''}, g^{w''}) = (m'. h^{w+w''}, g^{w+w''}) \]
  – Votes are initially encrypted by voters using the Electoral Board public key
  – Each Mix-net node re-encrypts and shuffles the encrypted votes using the same Electoral Board private key
  – At the end of the Mix-net, the Electoral Board decrypts the votes using the private key only once
Shuffling and re-encrypting

Homomorphic Operation

Encrypted votes
Re-encrypted votes

Homomorphic Product

362880

==

362880
Zero Knowledge Proofs

- Zero Knowledge proof of re-encryption
  - Proofs that the Mix-node knows the re-encryption factor without disclosing it.
  - Based on Schnorr Identification Protocol

- Zero Knowledge proof of correct decryption
  - Proofs that the Mix-node knows the decryption factor without disclosing this factor or the private key.
  - Based on Schnorr Identification Protocol
• Auditability in e-voting
• Universal verifiable Mix-nets
• Building blocks
• Proposal description
• Properties
• Conclusions
Steps of our proposal

- Re-encryption of the votes
  - Encrypted votes are re-encrypted and shuffled using a re-encryption Mix-net

- Auditor/observer challenges the Mix-net
  - Auditor challenges the Mix-net defining a random vote grouping initialization vector
  - Mix-net nodes generates integrity proofs of votes groups

- Verification of the Integrity proofs
  - Integrity proofs are verified before proceeding with vote decryption

- Decryption of the votes
  - Proofs are verified before proceeding with vote decryption
  - Decryption process generates proofs of correct decryption

- Further audits
  - Input/output Mix-node information (encrypted votes) and integrity proofs can be securely stored to allow further audits
Grouping mechanism

Groups

Reencrypted votes

Groups

Groups

Observers / auditors
Grouping rationale

\[ n = \sqrt[\text{t}}]{m} \]

\[ t \text{ is number Mix-net nodes and } m \text{ is the total number of votes} \]
Integrity proofs

Mix-node Shuffling and reencryption

Groups

Reencrypted votes

Groups

Integrity proofs

ZKP and Verification

Groups

Integrity proofs
• Auditability in e-voting
• Universal verifiable Mix-nets
• Building blocks
• Proposal description
• Properties
• Conclusions
Robustness

The probability of detecting two modified votes is shown in the graph. The equation for the probability of success, $P_{success}$, is given by:

$$P_{success} = 1 - \frac{n - 1}{m - 1}$$

where $m$ is the total number of votes and $n$ is the number of votes in each group.
• Proofs are based on groups of votes
• Grouping mechanism prevents correlation between input and output Mix-net votes
Number of exponentiations at each mix-node

Number of exponentiations

- Neff proposal
- Optimistic Mixing
- RPC
- Boneh and Golle proposal
- Our proposal

Number of votes in the Mixnet
• Auditability in e-voting
• Universal verifiable Mix-nets
• Building blocks
• Proposal description
• Properties
• Conclusions
• From the point of view of efficiency, the computation cost of our proposal is close to the fastest one (Boneh and Golle [BG02]) the fastest one, and faster than Random Partial Checking [JJR02] for medium amounts of votes (more than 1500)

• In terms of privacy, it does not pose any privacy concerns as the other efficient methods.

• In terms of accuracy, our proposal achieves a high level of cheating detection compared to the other most efficient methods. This probability is closer to 100% when the number of votes is near 300 votes (99%). The other methods, except [Ne01], have similar or lower accuracy levels.

• In summary, compared with the current verification methods, our solution is the most well-balanced in terms of efficiency, privacy, and accuracy, while providing universal verification properties.
Any questions?