Mixnets in Electronic Voting

Ben Adida
Cryptography and Information Security Group
MIT Computer Science and Artificial Intelligence Laboratory

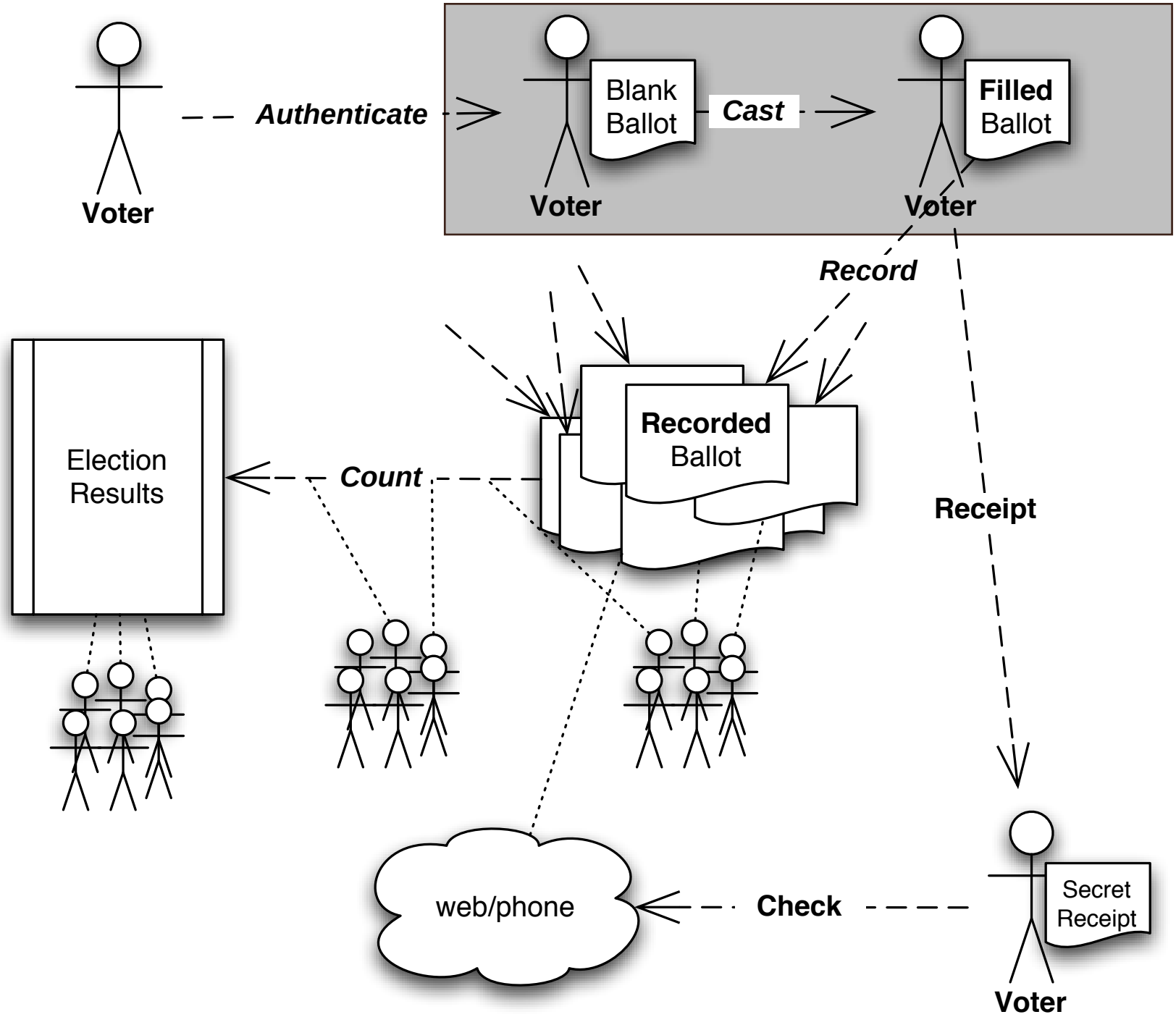
Cambridge University, 18 January 2005

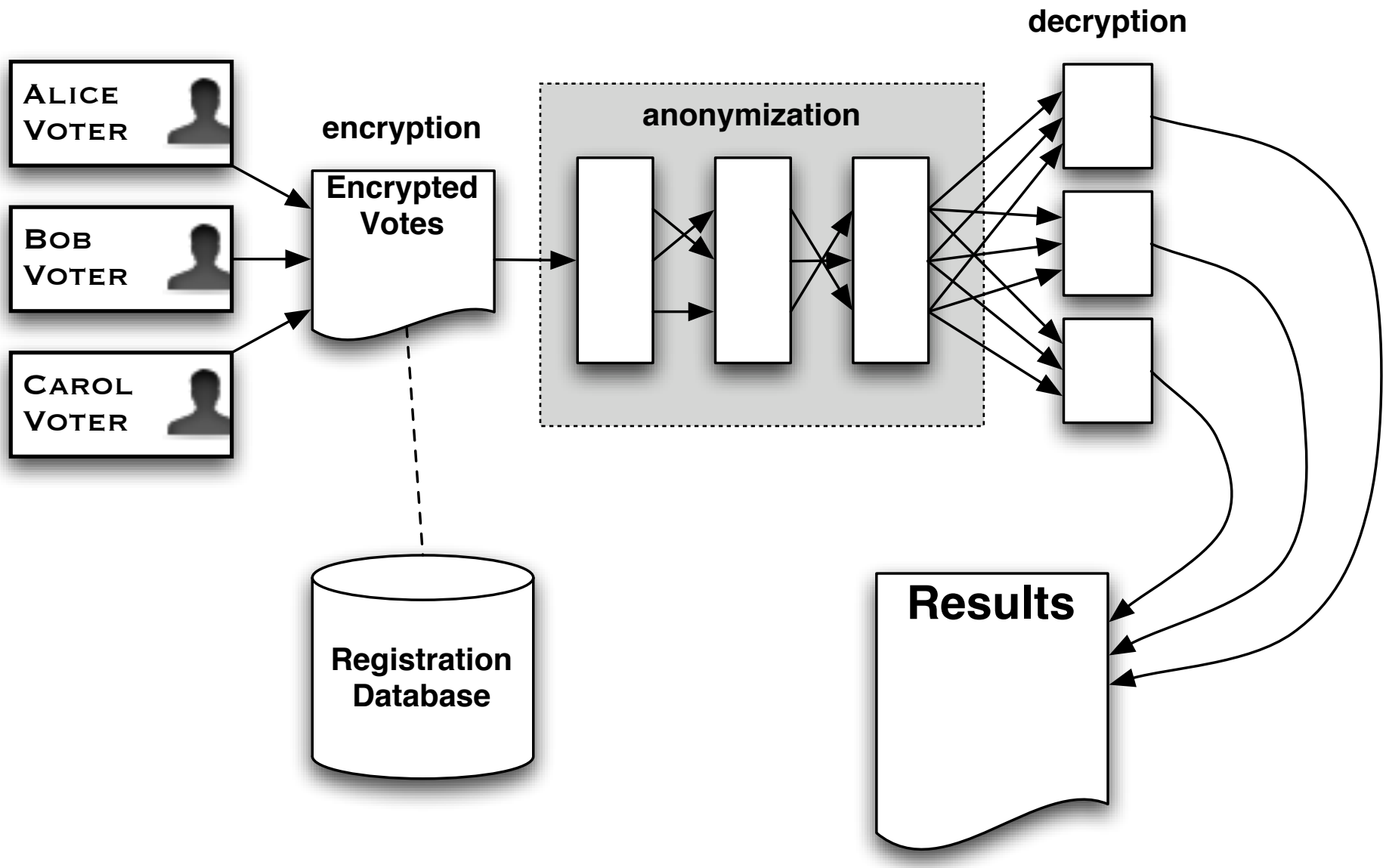
Overview

- Basics of Electronic Voting
- Reencryption Mixnets
- Millimix and Neff
- Almost-Entirely Correct Mixing
- Parallel Mixing
- Next Steps

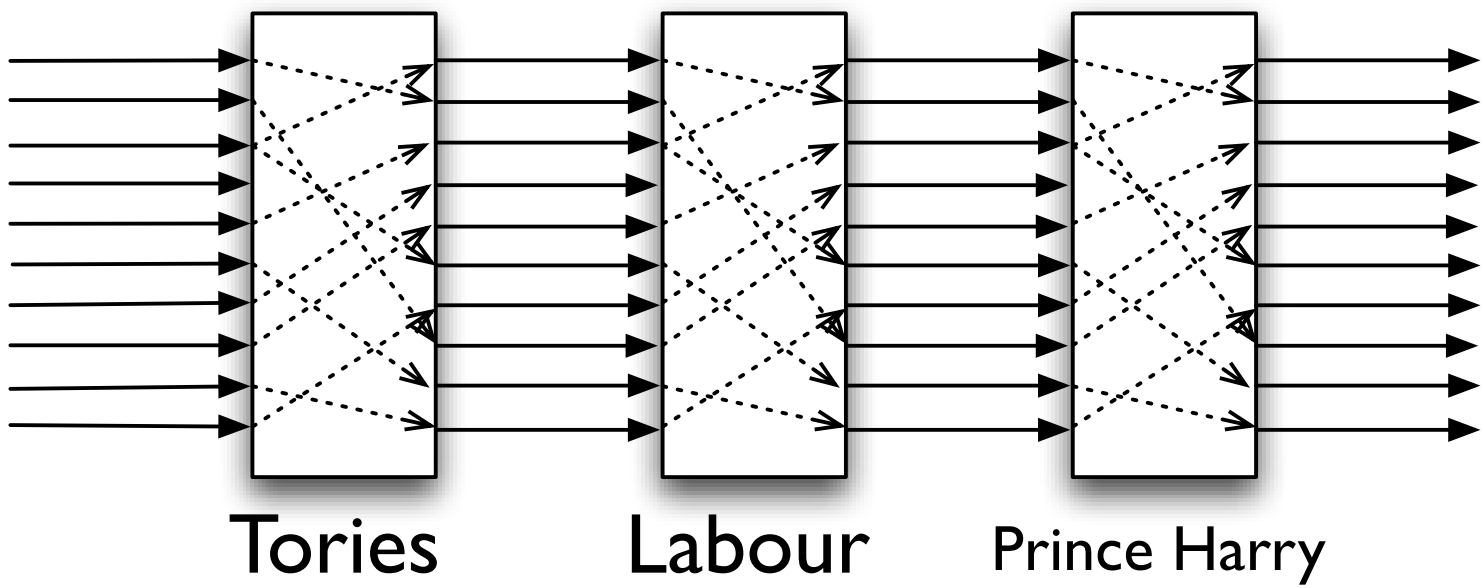
Electronic Voting

- Integrity Properties
  - Cast as Intended
  - Recorded as Cast
  - Tallied as Recorded
- Auditing with **Universal Verifiability**
- But... Coercion and Vote Selling


Voter
Authenticate
Blank Ballot
Voter
Cast
Filled Ballot
Voter
Record
Recorded Ballot
Election Results
Count
Receipt
web/phone
Check
Secret Receipt
Voter


decryption
Alice
Voter
Bob
Voter
Carol
Voter
encryption
Encrypted
Votes
anonymization
Registration
Database
Results

Robust Mixnet



Trust one mix server: the entire mixnet provides anonymity

Building Blocks

El-Gamal

- private key: $x \in \mathcal{Z}_q^*$
- public key: $y = g^x$

$$r \xleftarrow{R} \mathcal{Z}_q^*$$
$$Enc(m, r) = (\alpha, \beta) = (g^r, m * y^r)$$

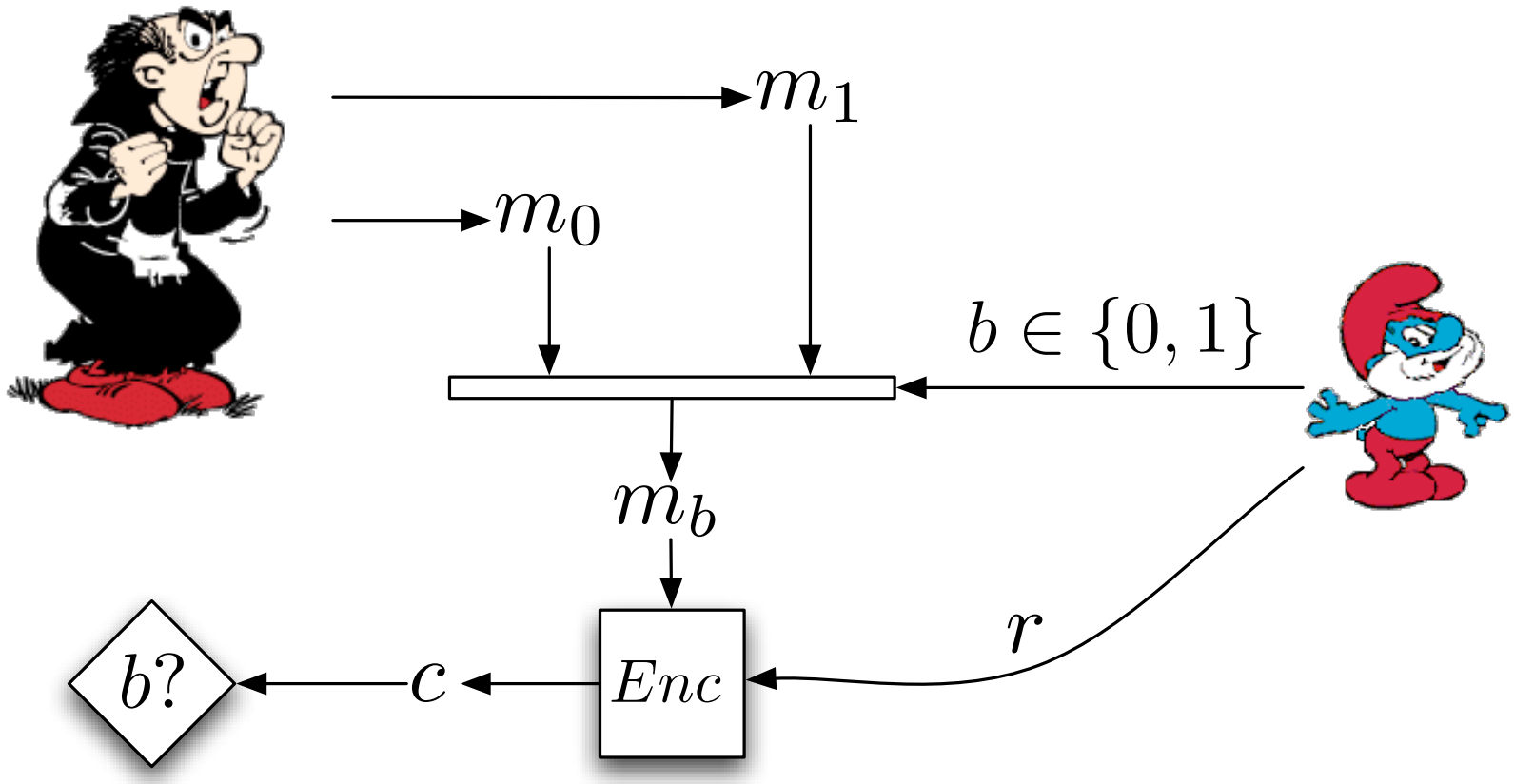
$$Dec(\alpha, \beta) = \frac{\beta}{\alpha^x}$$

Homomorphic Encryption & Reencryption

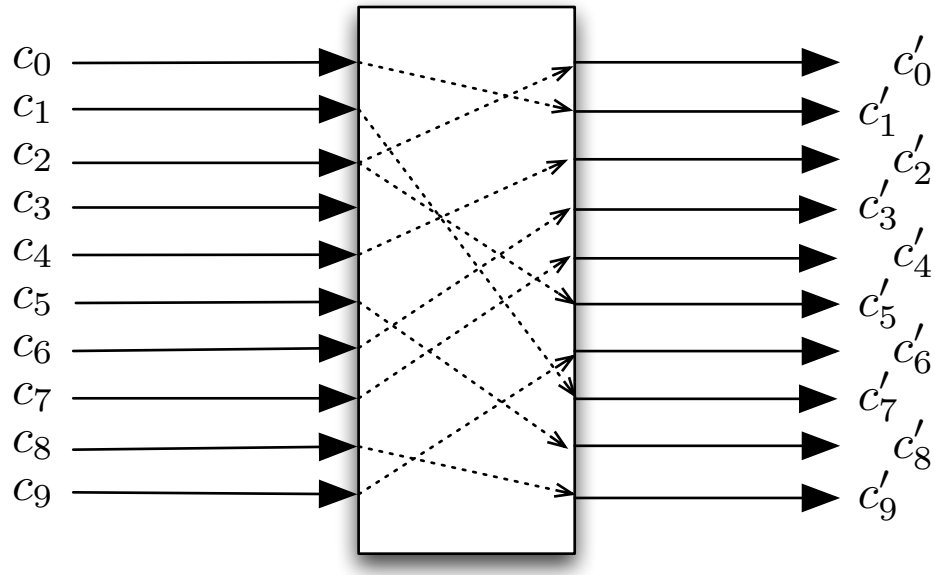
$$Enc(m_1, r_1) * Enc(m_2, r_2) =$$
$$Enc(m_1 * m_2, r_1 + r_2)$$

$$c = Enc(m, r)$$
$$Reenc(c, r') = c * Enc(1, r') = Enc(m, r + r')$$

Semantic Security



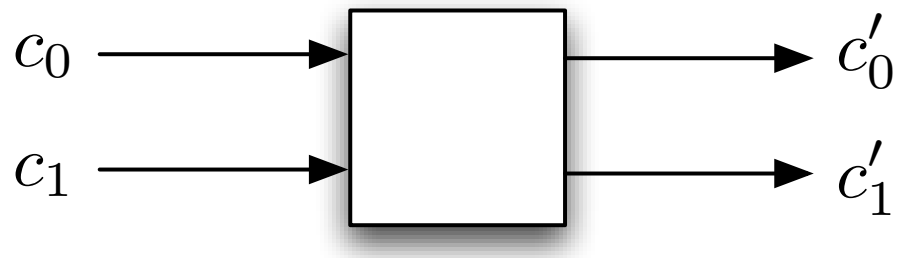
Reencryption Mixnet



$$\exists \pi, \exists r'_i, c'_{\pi(i)} = Reenc(c_i, r'_i)$$

We need a Proof of Knowledge
of the permutation and reencryption factors

Millimix (Juels 1999)



$$((m_0 = m'_0) \vee (m_0 = m'_1)) \wedge ((m_1 = m'_0) \vee (m_1 = m'_1)) \wedge$$
$$((m'_0 = m_0) \vee (m'_0 = m_1)) \wedge ((m'_1 = m_0) \wedge (m'_1 = m_1))$$

Using a Chaum/Pedersen Proof of Plaintext Equivalence and a Proof of Partial Knowledge of the **reencryption factors**

Millimix Expansion

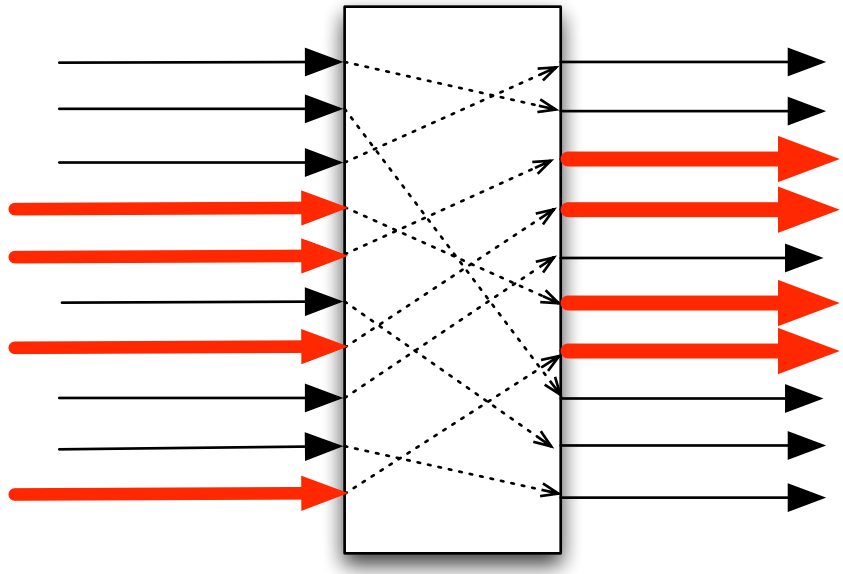
- Use a Sorting Network
  to expand Millimix to n inputs
- Proof Complexity = # of Comparators
- Time = $\Theta(n \log^2 n)$

Neff 2001/2003

- Proof Complexity: $8n$
- 100,000 votes in 10 hours on a PC
- used in VoteHere's software

Can We Make It Faster?

Almost-Entirely Correct Mixing
(Boneh/Golle 2002)



$$\text{Challenge: } B = \{i, b_i \xleftarrow{R} \{0,1\}, b_i = 1\}$$

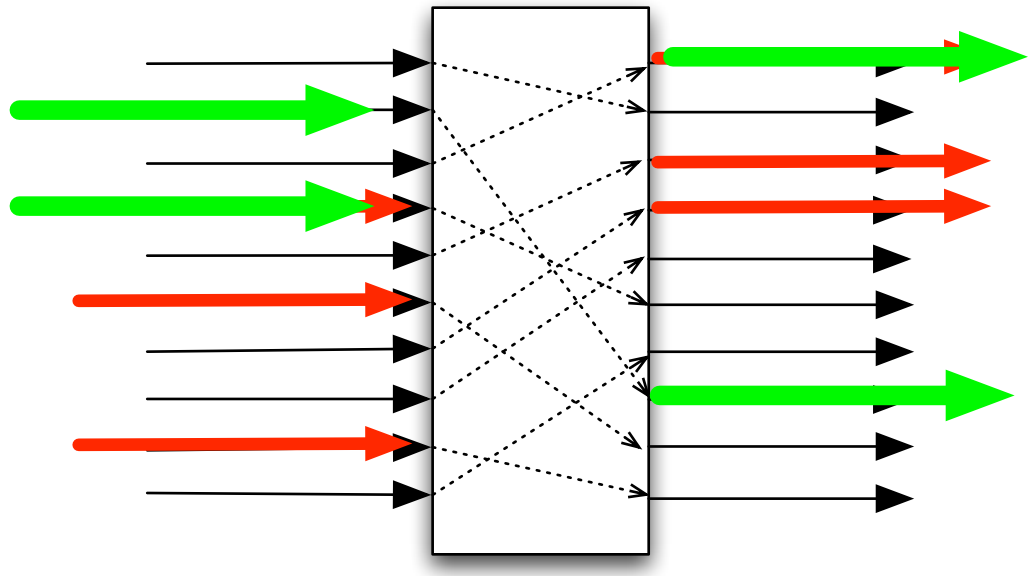
$$\text{Response: } B' = \{j, \prod_{i \in B} m_i = \prod_{j} m'_j\}$$

Proof of Knowledge of Block Reencryption Factor using Chaum Pedersen

Soundness

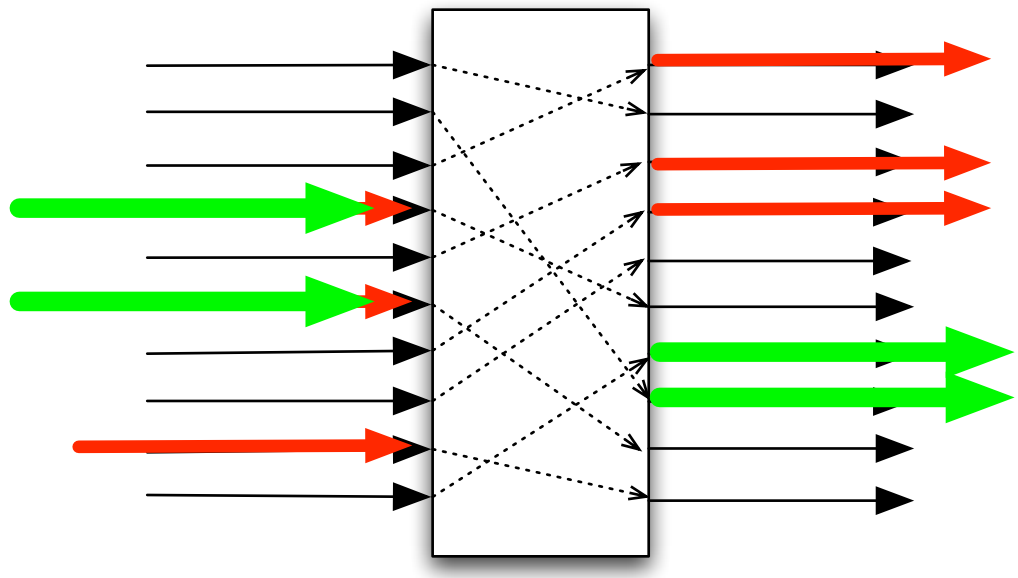
- If a mix server successfully answers a challenge with high enough probability...
  - it's honest, or
  - it can solve Discrete Log

Intuitively...



The Mix Server is Honest

Intuitively (II)...



Non-Trivial Identity on the Inputs

Non-Trivial Identity

$$\prod_{i=0}^{n} m_i^{e_i} = 1$$

Solving DLP

- Construct a Discrete Log Algorithm from non-trivial identity.
- Assume we are trying to compute $x = \log_g(h)$
- Construct message inputs as

$$r_i, s_i \xleftarrow{R} \mathcal{Z}_q^* \qquad m_i = g^{r_i} h^{s_i} = g^{r_i + x s_i}$$

Solving DLP (II)

- Compute $[e_0\ e_1\ \ldots e_{n-1}] \quad \prod_{i=0}^{n} m_i^{e_i} = 1$

$$\prod_{i=0}^{n} g^{e_i(r_i + x s_i)} = 1 \mod p$$

$$\sum_{i=0}^{n} e_i(r_i + x s_i) = 0 \mod q$$

$$\boxed{x = -\frac{\sum_{i=0}^{n} e_i r_i}{\sum_{i=0}^{n} e_i s_i}}$$

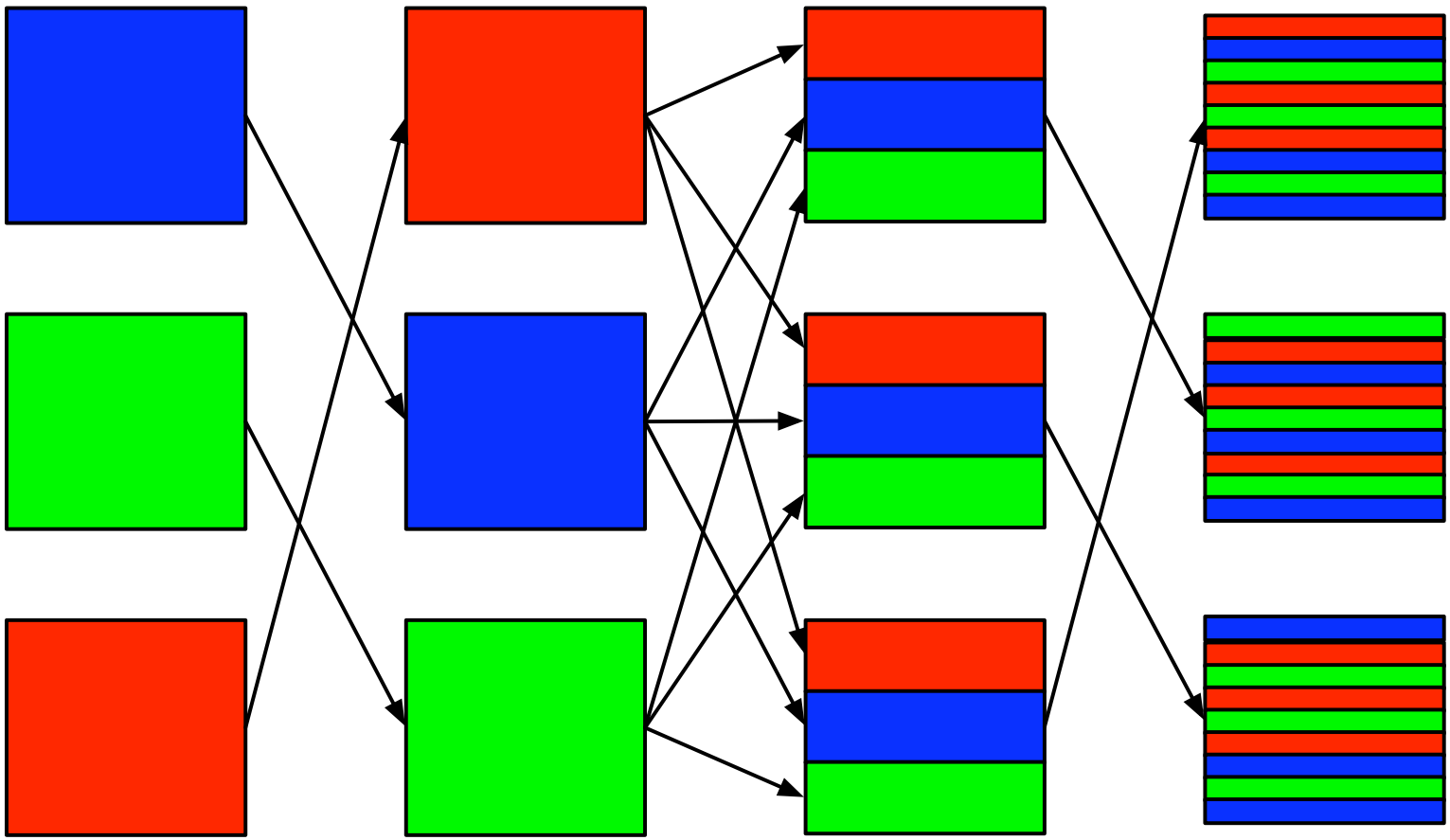
What About Privacy?

- In order to get 99% soundness, 10 challenges must be issued.
- Each challenge reduces the potential target space of some inputs by a factor of 2.
- Inputs are hidden among 1,000 times fewer outputs - ouch.

And the Mixing Step?

Parallel Mixing
(Golle/Juels 2004)

- Parallel Mixing
- Partition Votes into Batches
- More mixing time =
  Higher tolerance for bad mix servers



Future Work

- Lower Bound on Fully-Correct Mixing?
- Faster than Neff?
- More Anonymous than Boneh/Golle?

Questions?