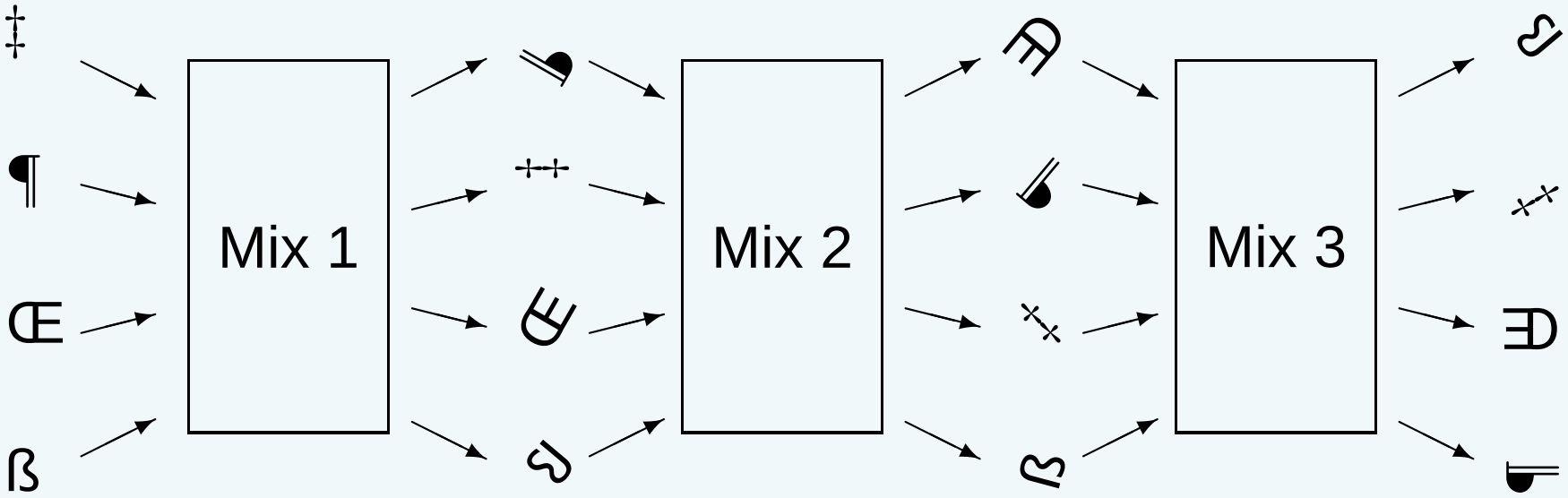
Random Block Verification:

Improvements to the Norwegian electoral mix net

Denise Demirel, *Hugo Jonker*, Melanie Volkamer

What is mixing?

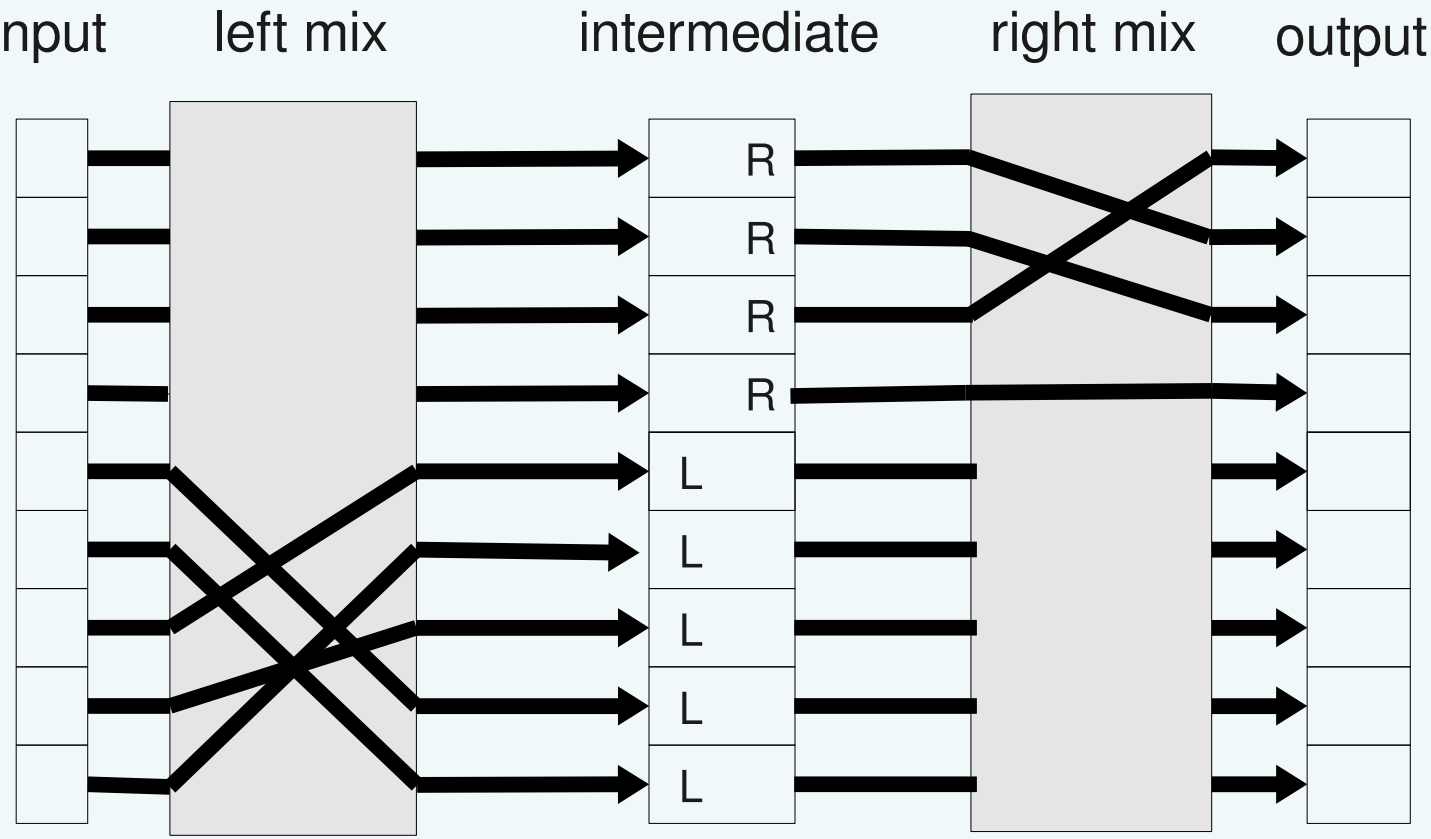


How to verify correctness of a mix operation?

- Using zero knowledge proofs
  efficiency problems, hard to understand, detects all fraud, no privacy loss

Trading detection for efficiency

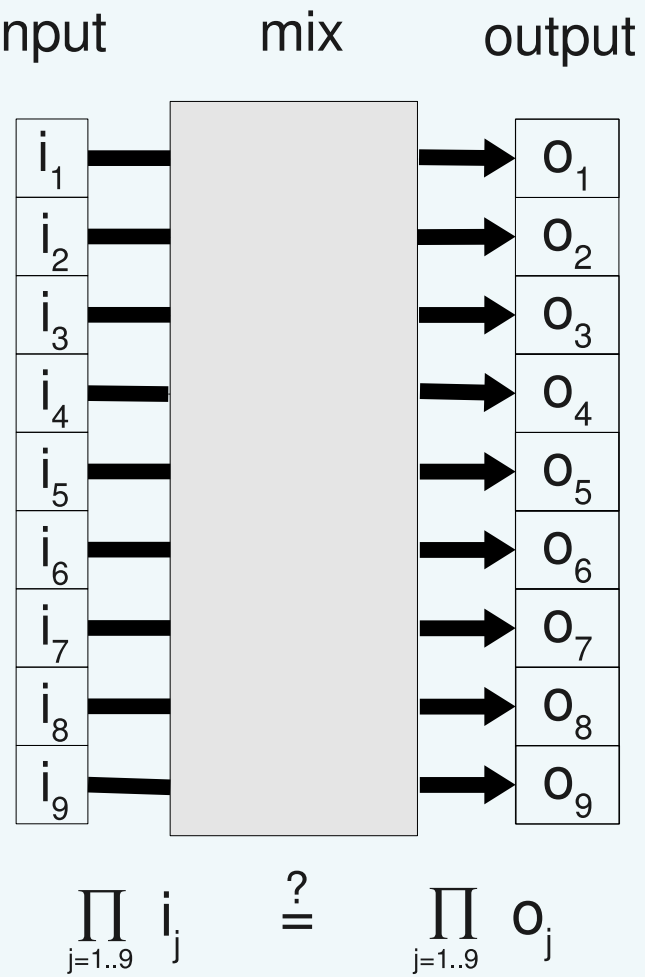
- Randomized Partial Checking [JJR02] (RPC)



doubles # operations, not all cheating detected, less privacy.

Trading detection for efficience (2)

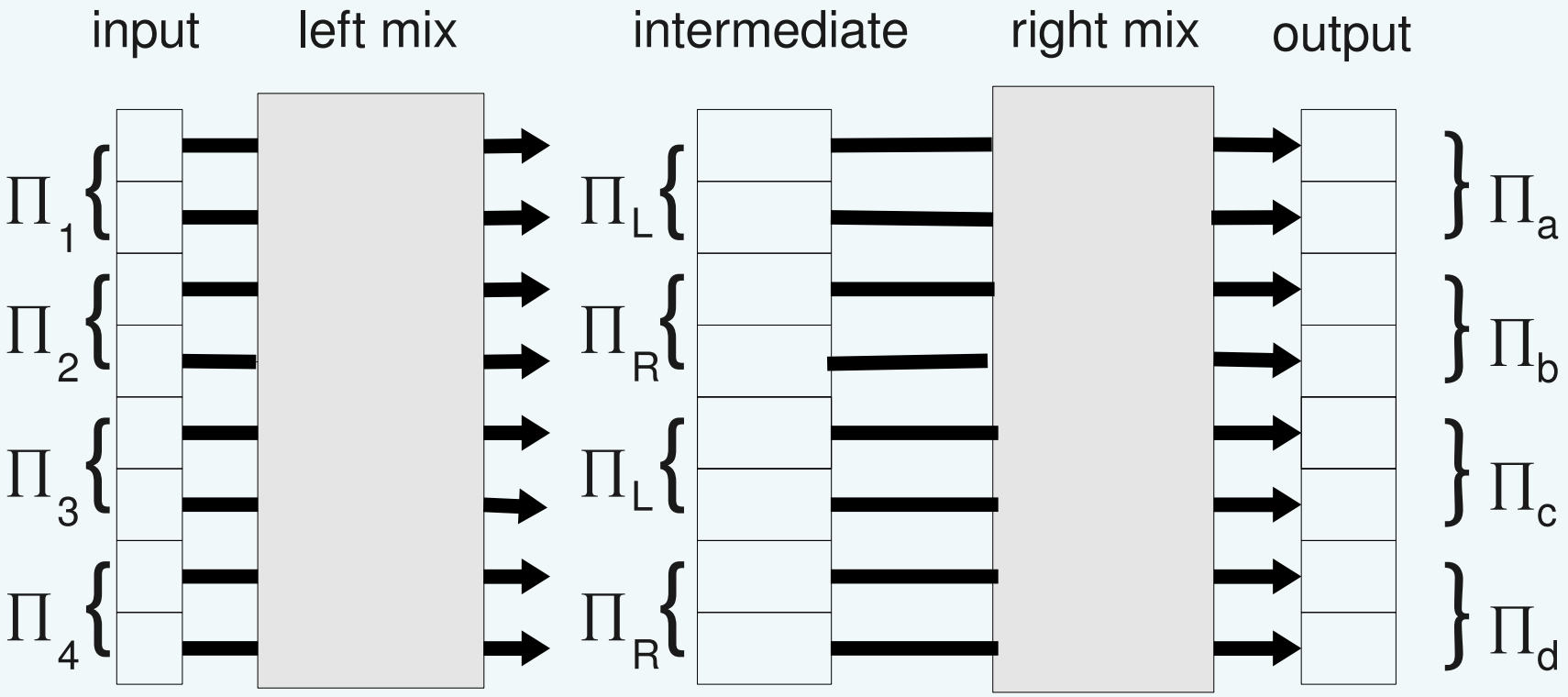
- Optimistic mixing [GZBJJ02] (OM)



Needs crypto, not all fraud detected, privacy loss.

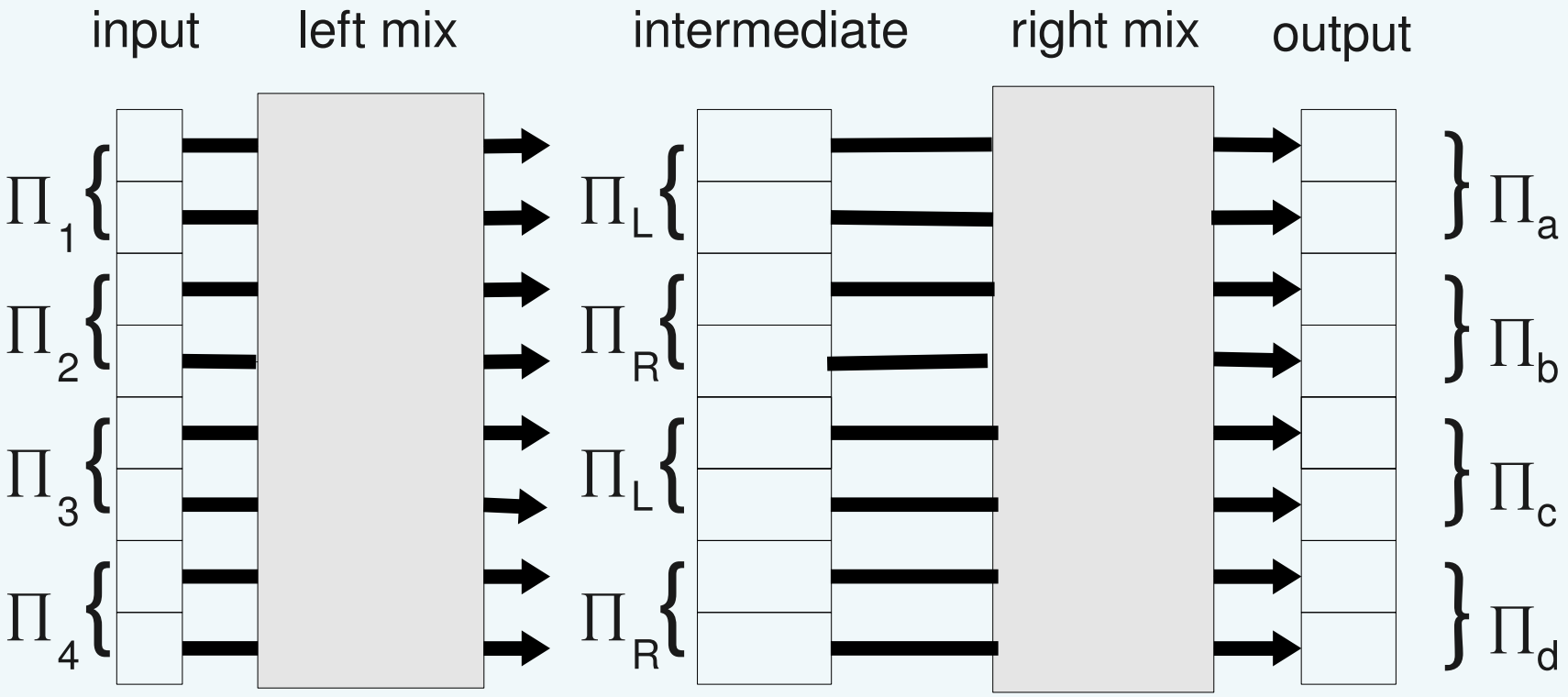
Combination: RPC+OM

Can a combination improve the trade-off?



Combination: RPC+OM

Can a combination improve the trade-off?



Puiggalí Allepuz and Guasch Castelló: [PG10].

Verification of the [PG10] mix net

1. Votes divided into $l = \sqrt[m]{n}$ blocks, for m mix nodes and n votes.
2. For every block:
   - identify corresponding group of outputs
   - publish products of input and output blocks
   - publish zero knowledge proof of equality of products
3. votes in output blocks are somewhat dispersed over the input blocks of the next mix.

Remarks on Norwegian mix net

- efficiency of proofs.
  - reveal re-encryption randomness
  - use more efficient proofs [JJ99]

Remarks on Norwegian mix net

- efficiency of proofs.
  - reveal re-encryption randomness
  - use more efficient proofs [JJ99]
- speedup via parallelisation.
  parallel mixing (of blocks) vs sequential mixing.
  used in Norway too.

Remarks on Norwegian mix net

- efficiency of proofs.
  - reveal re-encryption randomness
  - use more efficient proofs [JJ99]
- speedup via parallelisation.
  parallel mixing (of blocks) vs sequential mixing.
  used in Norway too.
- Reducing trust assumptions.
  - privacy in [PG10] preserved if all mix nets are honest (“*gradual* dispersal”).
  - Use Fiat-Shamir to prevent first mix from predicting block assignment.

Randomized Block Verification

Mixing in *RBV*

1. Divide the n votes into m subsets (for m mix nodes).

2. Mix i mixes subset i twice, and publishes intermediate and final results.

3. Next, mix i takes the final result of mix $i-1$ as input.
   Repeat m times.

Verification of *RBV*

- Divide input into $l = \lfloor \sqrt{n} \rfloor$ blocks.
  - $r = n - l \cdot l$ blocks with $l + 1$ elements, and
  - $l - r$ blocks with l elements.

Verification of *RBV*

- Divide input into $l = \lfloor \sqrt{n} \rfloor$ blocks.
  - $r = n - l \cdot l$ blocks with $l + 1$ elements, and
  - $l - r$ blocks with l elements.

- Determine blocks: input blocks $=$ output blocks of previous mix.

Verification of *RBV*

- Divide input into $l = \lfloor \sqrt{n} \rfloor$ blocks.
  - $r = n - l \cdot l$ blocks with $l + 1$ elements, and
  - $l - r$ blocks with l elements.

- Determine blocks: input blocks $=$ output blocks of previous mix.

- Prove correspondence of input blocks with intermediate blocks (first mixing operation).
  Efficient ZK proof or reveal re-encryption randomness.

Verification of *RBV*

- Divide input into $l = \lfloor \sqrt{n} \rfloor$ blocks.
  - $r = n - l \cdot l$ blocks with $l + 1$ elements, and
  - $l - r$ blocks with l elements.

- Determine blocks: input blocks $=$ output blocks of previous mix.

- Prove correspondence of input blocks with intermediate blocks (first mixing operation).
  Efficient ZK proof or reveal re-encryption randomness.

- Redistribute votes over blocks. l blocks with l elements $\implies$ perfect redistribution. We're close.

Verification of *RBV*

- Divide input into $l = \lfloor \sqrt{n} \rfloor$ blocks.
  - $r = n - l \cdot l$ blocks with $l + 1$ elements, and
  - $l - r$ blocks with l elements.
- Determine blocks: input blocks $=$ output blocks of previous mix.
- Prove correspondence of input blocks with intermediate blocks (first mixing operation).
  Efficient ZK proof or reveal re-encryption randomness.
- Redistribute votes over blocks. l blocks with l elements $\implies$ perfect redistribution. We're close.
- Prove correspondence intermediate blocks with output blocks.

Verification of *RBV*

- Divide input into $l = \lfloor \sqrt{n} \rfloor$ blocks.
  - $r = n - l \cdot l$ blocks with $l + 1$ elements, and
  - $l - r$ blocks with l elements.

- Determine blocks: input blocks $=$ output blocks of previous mix.

- Prove correspondence of input blocks with intermediate blocks (first mixing operation).
  Efficient ZK proof or reveal re-encryption randomness.

- Redistribute votes over blocks. l blocks with l elements $\implies$ perfect redistribution. We're close.

- Prove correspondence intermediate blocks with output blocks.

Done.

Comparison

Detecting fraud

mix	*chance of not detecting fraud*
RPC [JJR02]	$P(k\ undetected\ changes) = 2^{-k}$.
PoS [GZBJJ02]	$\max(P(k+1\ undetected\ changes)) = \frac{5}{8}$.
Norway [PG10]	$P(k+1\ undetected\ changes) = \left(\frac{\sqrt[m]{n}-1}{n-1}\right)^k$.
RBV	$P(k+1\ undetected\ changes) = \left(\frac{\sqrt{n}-1}{n-1}\right)^k$.

Note: RBV not depending on # mix nodes.

Privacy

mix	*size of anonymity group*
RPC	$\left\|AG\right\| = \frac{n}{2}$.
PoS	$\left\|AG\right\| = \frac{n}{2^{\alpha}}, (0 < \alpha \leq 5)$.
Norway	$\left\|AG\right\| = \frac{n}{\sqrt[m]{n}}$.
RBV	$\left\|AG\right\| = n$.

Efficiency of RBV

mix	*#exp per mix node*
RPC	$2n$.
PoS	$2\alpha \cdot (2m-1), \qquad 0 < \alpha \leq 5.$
Norway	$6 \cdot \frac{n}{\sqrt[m]{n}}$.
RBV	$6 \cdot \frac{n}{\lfloor \sqrt{n} \rfloor}$.

Conclusions

- Improved privacy of [PG10].
- Some efficiency improvements.
- Main cost: fraud detection.
  However, still too good for practical attacks.

- For future: mix net verification using ZK proofs more and more efficient.

Conclusions

- Improved privacy of [PG10].
- Some efficiency improvements.
- Main cost: fraud detection.
  However, still too good for practical attacks.

- For future: mix net verification using ZK proofs more and more efficient.

Thanks for your attention.

Questions?

References

[JJR02] Jakobsson, M., Juels, A., Rivest, R.L.: Making mix nets robust for electronic voting by randomized partial checking. In: Proc. USENIX'02 (2002)

[PG10] Puiggal í Allepuz, J., Guasch Castelló, S.: Universally verifiable efficient re-encryption mixnet. In: Proc. EVOTE 2010. LNI, vol. P-167, pp. 241–254. GI (2010)

[GZBJJ02] Golle, P., Zhong, S., Boneh, D., Jakobsson, M., Juels, A.: Optimistic mixing for exit-polls. In: Asiacrypt 2002, LNCS 2501. pp. 451–465. Springer-Verlag (2002)

[JJ99] Jakobsson, M., Juels, A.: Millimix: Mixing in small batches. Tech. rep., Center for Discrete Mathematics #38; Theoretical Computer Science (1999)