BLINDSKEEN: TOWARDS COORDINATION-STRUCTURE PRIVACY IN ATOMIC MULTICAST VIA ZERO-KNOWLEDGE MEMBERSHIP PROOFS

AUTHOR
Amanda Andree | a.andree@students.tudelft.nl

SUPERVISOR
Jérémie Decouchant

1. INTRODUCTION

Broadcast sends to all nodes. Multicast (Fig. 1) targets only relevant subsets. **Atomic multicas**t adds strict ordering guarantees on top:

- **Selective delivery:** only destination nodes deliver the message
- **Total ordering:** conflicting messages delivered in the same order across all shared recipients
- **Causal ordering:** message order respects happened-before relationships

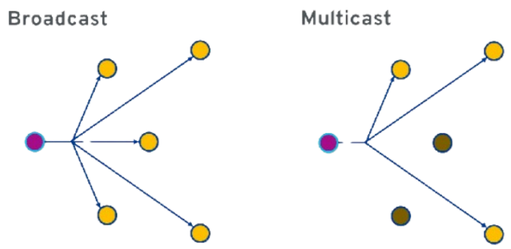


Fig. 1: Broadcast vs. multicast delivery

2. PROBLEM STATEMENT

- **The problem:** classical atomic multicast exposes destination sets, leaking participant relationships, access patterns, and workload structure.
- BlindSkeen hides (Fig. 2):
  - **Who receives:** nodes learn only their own group membership
  - **Who overlaps:** destination-set overlap is not observable across groups
  - **Who sends:** recipients cannot identify the coordinator

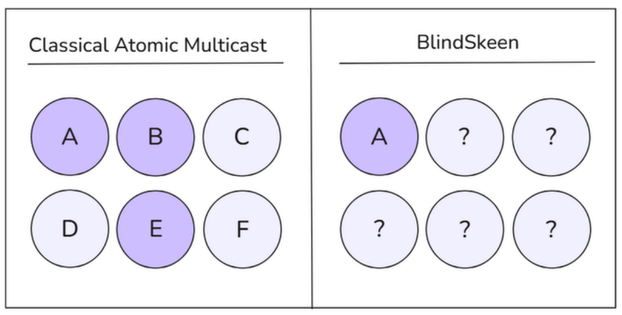


Fig. 2: In classical atomic multicast, all destination nodes are visible. In BlindSkeen, each node sees only its own membership.

3. RESEARCH QUESTIONS

RQ1: Can atomic multicast preserve ordering while hiding destination-set membership?
RQ2: What information is inherently leaked by BlindSkeen, and can we bound it?
RQ3: What is the performance overhead relative to standard atomic multicast?

4. DESIGN

1. **Commit & Prove:** per-group ZK proofs hide the full destination set
2. **Mixnet Routing:** onion-encrypted routing unlinking sender from destinations
3. **Selective Verification:** each group verifies only its own proof (Fig. 3)

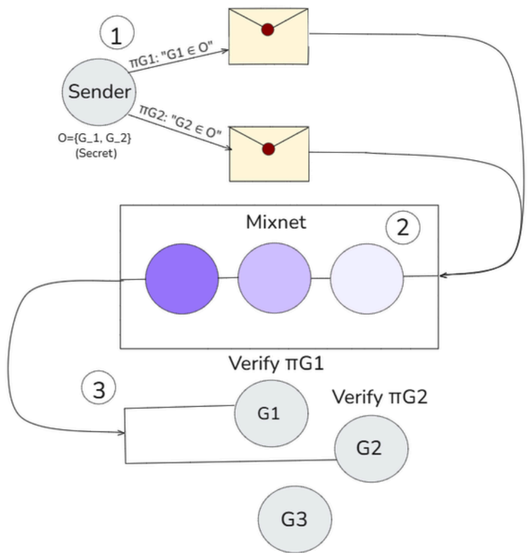
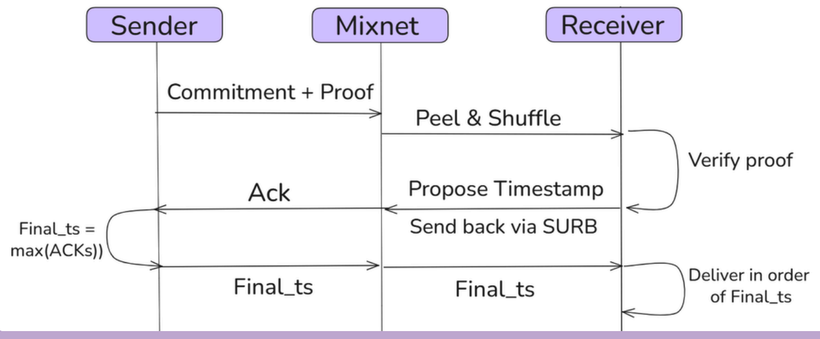


Fig. 3: BlindSkeen message flow . Acks and isFinal are omitted for clarity.

4. **Atomic** Ordering: recipient groups negotiate timestamps to enforce total-order delivery (Fig. 4)

Fig. 4: BlindSkeen Message-flow diagram.



5. RESULTS

This overhead buys formally proven privacy: **metadata and coordinator anonymity** hold with negligible adversarial advantage under standard cryptographic assumptions (proofs in the full paper).

Does privacy overhead grow with conflict rate? (Fig 5.)

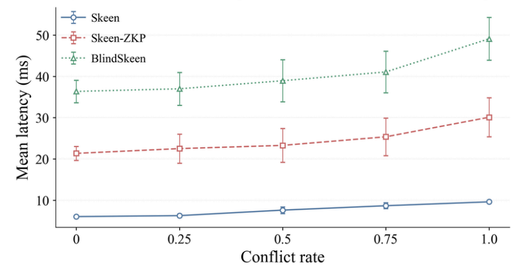


Fig. 5: Mean end-to-end latency vs. conflict rate (4 groups, 4 nodes/group, 30 reps).

→ Privacy does not cost more under contention: ZKP adds a fixed ~15 ms; mixnet adds ~13 ms, both independent of conflict rate.

How does latency scale with group count? (Fig. 6)

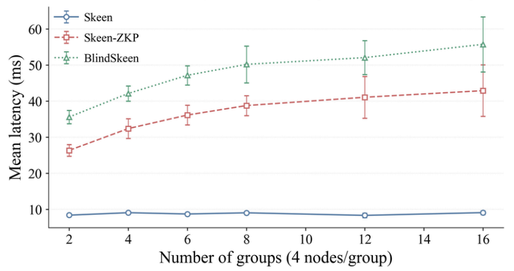


Fig. 6: Mean end-to-end latency vs. group count (4 nodes/group, ρ=0.5, 30 reps).

→ Overhead scales sub-linearly: BlindSkeen tracks Skeen-ZKP with a constant ~15 ms gap.

What drives end-to-end latency in BlindSkeen? (Fig. 7)

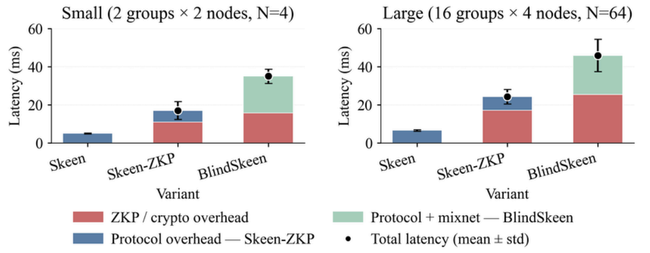


Fig. 7: Latency decomposition by topology: small (2 groups × 2 nodes) and large (16 groups × 4 nodes).

→ ZKP verification dominates (60–75%); mixnet adds a fixed, independent cost.

What is the bandwidth cost of privacy? (Fig. 8)

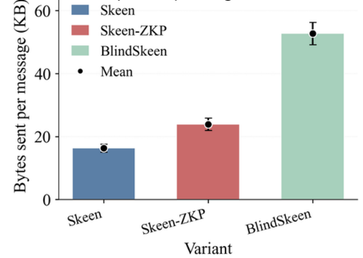


Fig. 8. Bytes sent per message across variants (4 groups, 4 nodes, ρ=0.5).

→ BlindSkeen uses 3.2× more bandwidth, but cost is per-message not per-byte.

6. CONCLUSION

1. Destination sets can be hidden without compromising ordering correctness, via per-group ZK proofs and mixnet transport.
2. Inherent leakage is limited to destination-set cardinality; all other coordination structure remains hidden.
3. Privacy overhead is bounded and predictable: 3.5× latency, stable across conflict rates and sub-linear in group count.

7. FUTURE WORK

- Replacing per-group commitments with **Merkle accumulators** without introducing cross-group correlation
- **Threshold constructions** as an alternative to membership proofs
- Extending blind ordering to **modern multicast protocols**