Towards Provably-Secure Scalable Anonymous Broadcast

Mahdi Zamani
Department of Computer Science
University of New Mexico

Joint work with
Jared Saia†, Mahnush Movahedi†, and Joud Khoury‡

†University of New Mexico
‡Raytheon BBN Technologies
Anonymity

- Anonymity enables *freedom of speech*
  - One can speak without fear of consequences

- Political discussions can result in torture, imprisonment, and death!

- Theoretical guarantees are critical
Our Goal

- Provably-anonymous and fully-decentralized broadcast
- Ten-year vision
  - Fully decentralized anonymous Twitter
Our Model

- *n* parties, all pairs connected by private channel
- Up to a *1/3 fraction* are *bad*
- **Byzantine** adversary: *Bad* parties can deviate arbitrarily from protocol; the rest are *Good*
- **Static** adversary: takes over bad parties at the beginning of protocol
- **Synchronous** communication
Anonymous Broadcast Problem

- Each party has a message to broadcast

- Goals:
  - All parties receive the messages broadcast
  - No coalition of parties can do better than uniform guessing to map messages to senders
Our Results

- Best known protocol for this model is [GJ ’04]
  - Sends $O(n)$ bits per player per anonymous bit
  - Constant latency
  - Cryptographic

- Our Algorithm
  - Sends $\tilde{O}(1)$ bits per player per anonymous bit
  - $O(\log n)$ latency
  - Probability that a party’s message is broadcast is $1 - \frac{1}{k}$
  - Unconditionally secure
Secure Multi-Party Computation

- Each party knows a global function and a private input

- Goals:
  - Each party learns the output of the function over the private inputs
  - Keep inputs private: nobody learns anything about the inputs other than what is revealed by the output
Simple **DC-Net** [Chaum ’88]

- One party has input $m$, remaining have input zero
- Compute sum via MPC
- All parties learn $m$, sender identity remains hidden

Pros and cons

- Secure against traffic analysis
- Vulnerable to *jamming attacks*
- Poor scalability
Our Protocol

- Create a **circuit** to enable anonymous broadcast with jam prevention
- Prevent jamming by using **virtual channels**
- Use scalable MPC scheme of [DKMS ’12] to securely evaluate this circuit
- Trade off latency to decrease bandwidth cost
Quorums [DKMS ’12]

- A set of $\log(n)$ parties
- 2/3 fraction of them are good
- Each gate is mapped to a quorum
- This quorum computes the gate output securely
- Distributed quorum creation is done via [KLST ’10]
The Circuit
The Circuit

Left input if right input is 0
0 otherwise

All 0 if \( \leq 1 \) channel used
all 1 otherwise

Virtual
channels

\( p_1 \)

\( p_2 \)

\( p_n \)
JAM-Detector
0 if 0 channel used
1 if 1 channel used
2 otherwise
Analysis

- Let $r$ be the number of channels
- Each party sends $\tilde{O}\left(\frac{r + \sqrt{n}}{n}\right)$ bits for sending one anonymous bit
- If $r = kn$, probability of collision per party is $1/k$
- Each party sends $\tilde{O}(1)$ bits per anonymous bit.
Simulation Results

Number of bits sent per player per anonymous bit

- **This paper**
- **Ongoing work**
- **[GJ '04]**

Log number of players vs. Log number of bits sent.
Simulation Results

Number of bits sent per player per anonymous bit

Log number of players

Log number of bits sent

This paper
Ongoing work
[GJ ’04]
Conclusion

- Anonymous broadcast against a powerful adversary
- Asymptotically-efficient
  - Reduces bandwidth cost by $n$
  - Increases latency by $\log n$
- We proved
  - No coalition of parties can do better than uniform guessing
data mapping for messages to senders
Future Work I

- Ten-year vision: decentralized anonymous Twitter
- MBs per player is too high for an anonymous tweet!
- Three orders of magnitude improvement plan
  - Use cryptographic methods
    - Fully homomorphic encryption
    - Threshold cryptography for Byzantine agreement
  - Blacklist bad parties over time
Future Work II

- Our model was defined 25 years ago [Chaum ’88]
- Modern networks are more dynamic and flexible
- How to handle modern networks and still give theoretical guarantees?
- Also
  - Asynchronous communication
  - Adaptive adversary
  - Churn
Thank you!

Questions?