DISSENT: Accountable, Anonymous Communication

Joan Feigenbaum
http://www.cs.yale.edu/homes/jf/

Joint work with Bryan Ford (PI), Henry Corrigan-Gibbs, Ramakrishna Gummadi, Aaron Johnson (NRL), Vitaly Shmatikov (UT Austin), Ewa Syta, and David Wolinksy

Supported by DARPA

Problem Statement

- A group of $N \geq 2$ parties wish to communicate anonymously, either with each other or with someone outside of the group.
- They have persistent, "real-world" identities and are known, by themselves and the recipients of their communications, to be a group.
- They want a protocol with four properties:
    - ✓ Integrity
    - ✓ Anonymity
    - ❖ Accountability
    - o Efficiency

Accountability

- Group member *i* **exposes** group member *j* if *i* obtains proof, verifiable by a third party (not necessarily in the group), that *j* **disrupted** a protocol run.
- The protocol maintains **accountability** if no honest member is ever exposed, and, after every run, either:
    - o every honest member successfully receives every honest member's message, or
    - o every honest member exposes at least one disruptive member.

Need for Anonymity (1)

- Communication in hostile environments

  From the BAA: “The goal of the program is to develop technology that will enable safe, resilient communications over the Internet, particularly in situations in which a third party is attempting to discover the identity or location of the end users or block the communication.”

Need for Anonymity (2)

- Cash transactions
- Twelve-step programs (pseudonymy)
- Law-enforcement “tip” hotlines
- Websites about sensitive topics, e.g., sexuality, politics, religion, or disease
- Voting
- . . .

Need for Accountability

- Authoritative, credentialed group, e.g.:
  - Board of Directors of an organization
  - Federation of journalists (… think Wikileaks)
  - Registered voters
- Internal disagreement is inevitable.
- Infiltration by the enemy may be feasible.
- ➢ Disruption is expected and must be combated.

? It's not clear that "accountability" is the right word to use here (… and that's part of a longer story).

Outline

- Prior work on anonymous communication
- Basic DISSENT protocol (ACM CCS 2010)
- Results to date

Outline

- ***Prior work on anonymous communication***
- Basic DISSENT protocol (ACM CCS 2010)
- Results to date

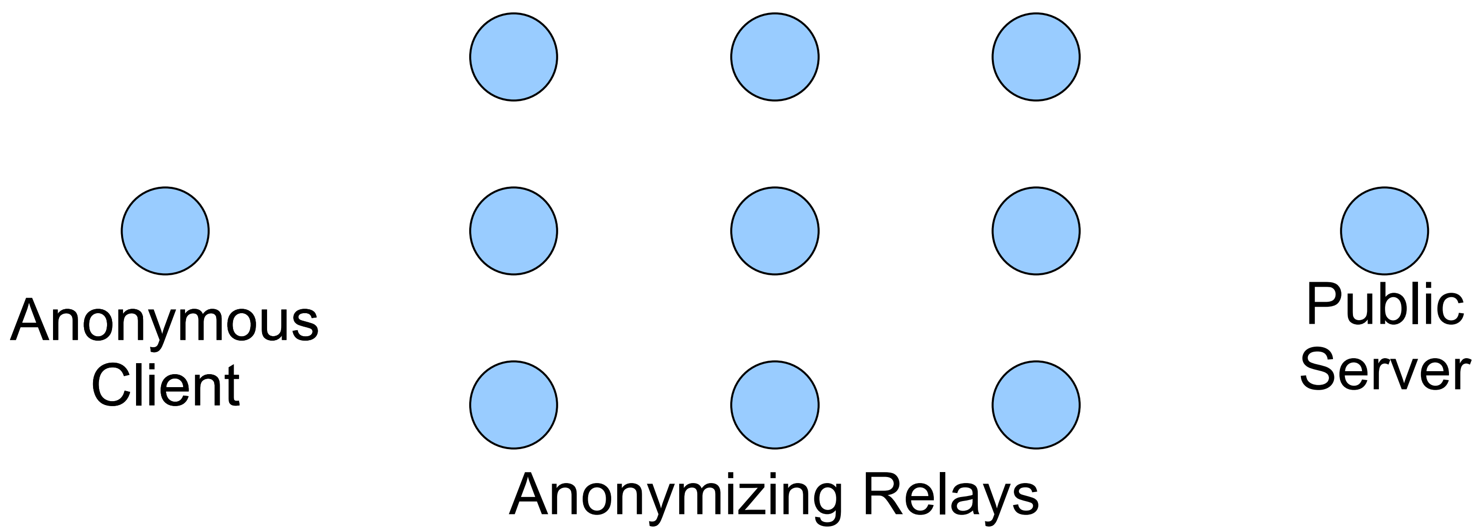
Major Themes in Prior Work

- General-purpose anonymous-communication mechanisms
  - MIX networks and Onion Routing (OR)
  - Dining-Cryptographers networks (DC-nets)

- Special-purpose mechanisms, e.g.:
  - Anonymous voting
  - Anonymous authentication, e.g., group or ring signatures
  - E-cash

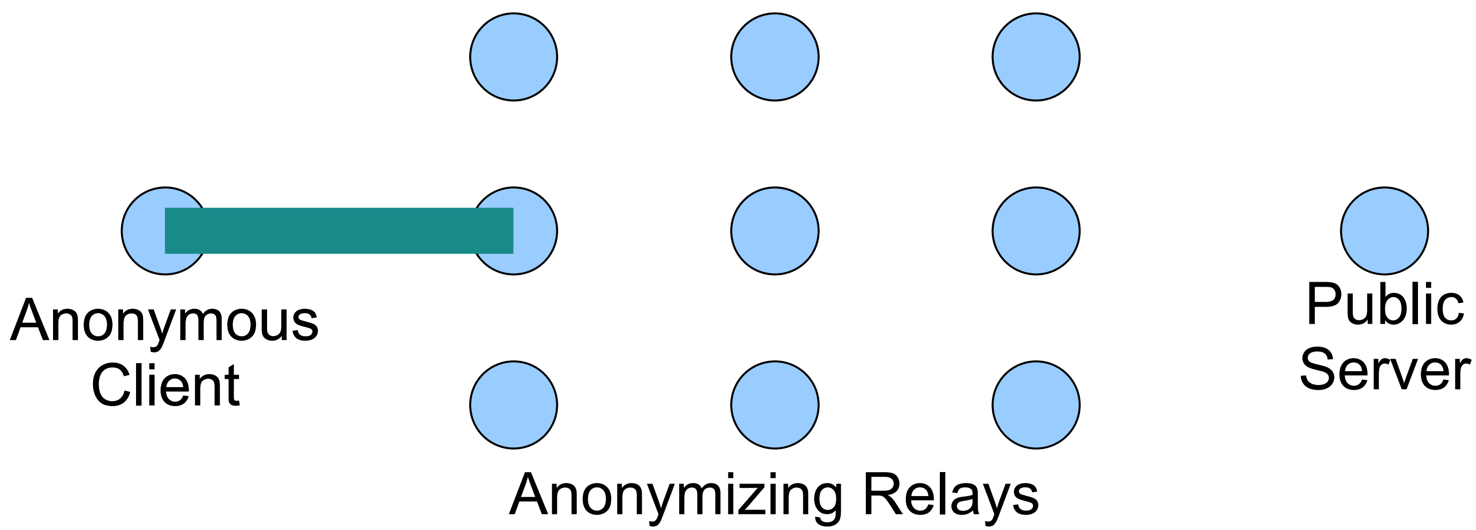
Basic Operation of Onion Routing

- Client picks a few (e.g., three) **anonymizing relays** from a cloud of available relays.
- He then builds and uses an **onion** of cryptographic tunnels through the relays to his communication partner.



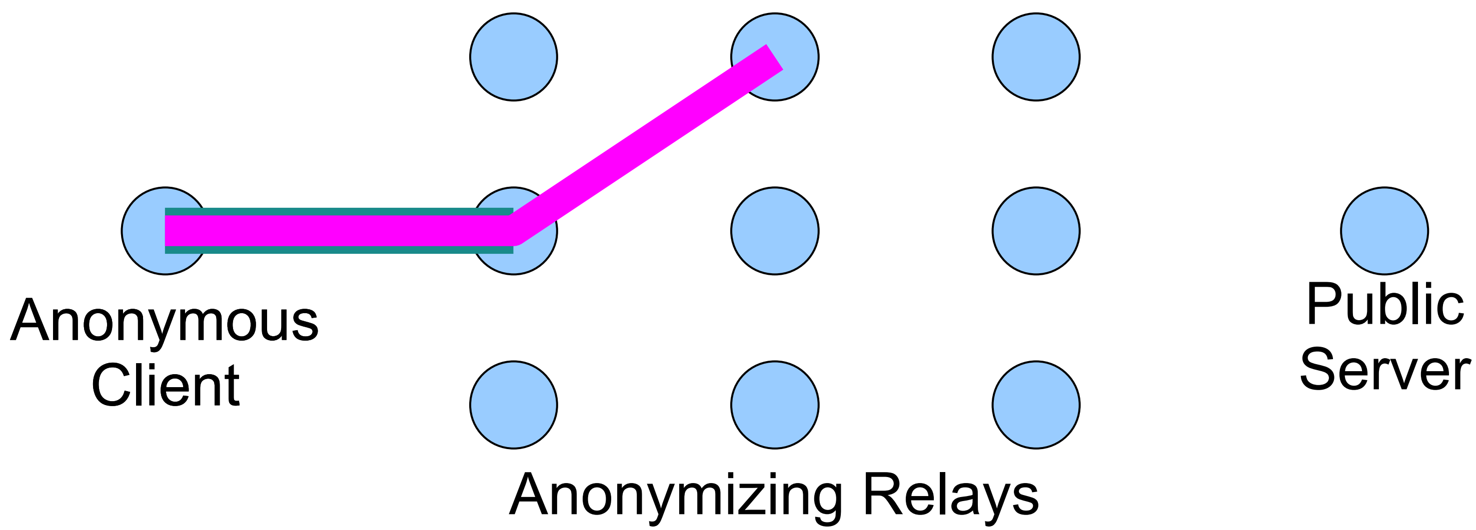
Basic Operation of Onion Routing

- Client picks a few (e.g., three) **anonymizing relays** from a cloud of available relays.
- He then builds and uses an **onion** of cryptographic tunnels through the relays to his communication partner.



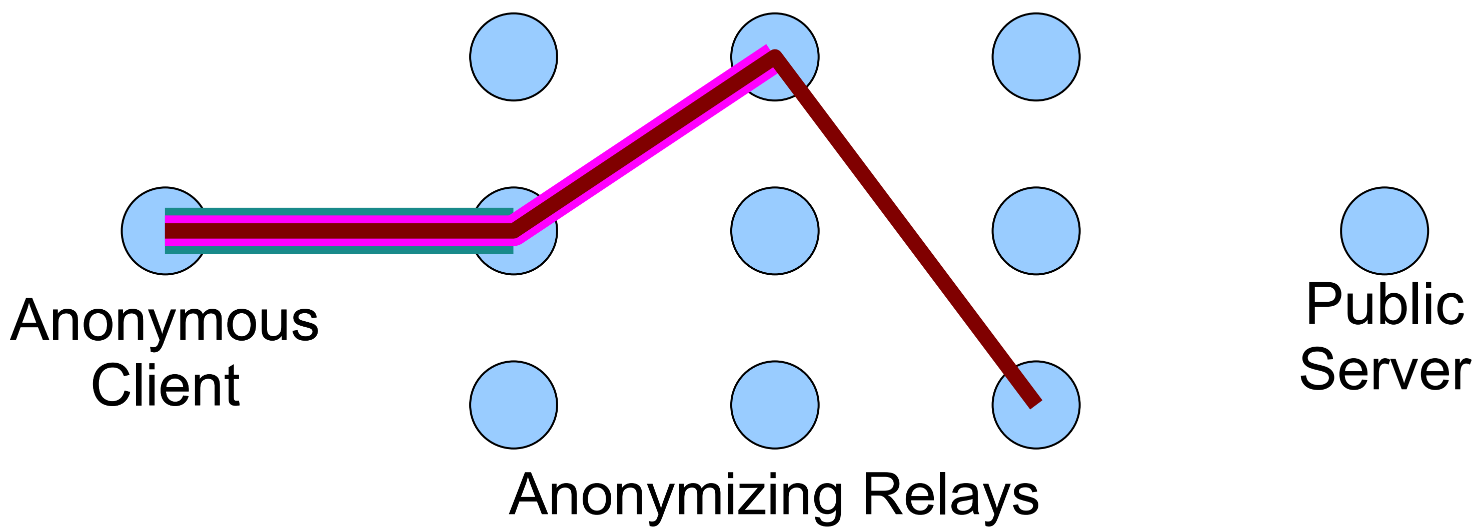
Basic Operation of Onion Routing

- Client picks a few (e.g., three) **anonymizing relays** from a cloud of available relays.
- He then builds and uses an **onion** of cryptographic tunnels through the relays to his communication partner.



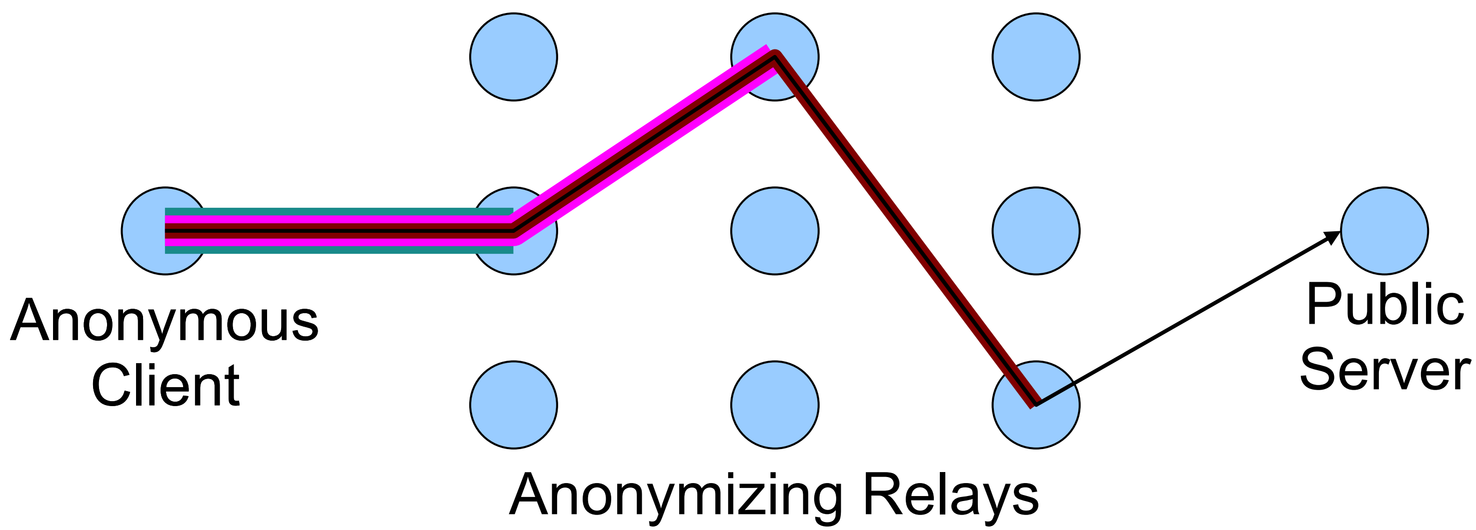
Basic Operation of Onion Routing

- Client picks a few (e.g., three) **anonymizing relays** from a cloud of available relays.
- He then builds and uses an **onion** of cryptographic tunnels through the relays to his communication partner.



Basic Operation of Onion Routing

- Client picks a few (e.g., three) **anonymizing relays** from a cloud of available relays.
- He then builds and uses an **onion** of cryptographic tunnels through the relays to his communication partner.



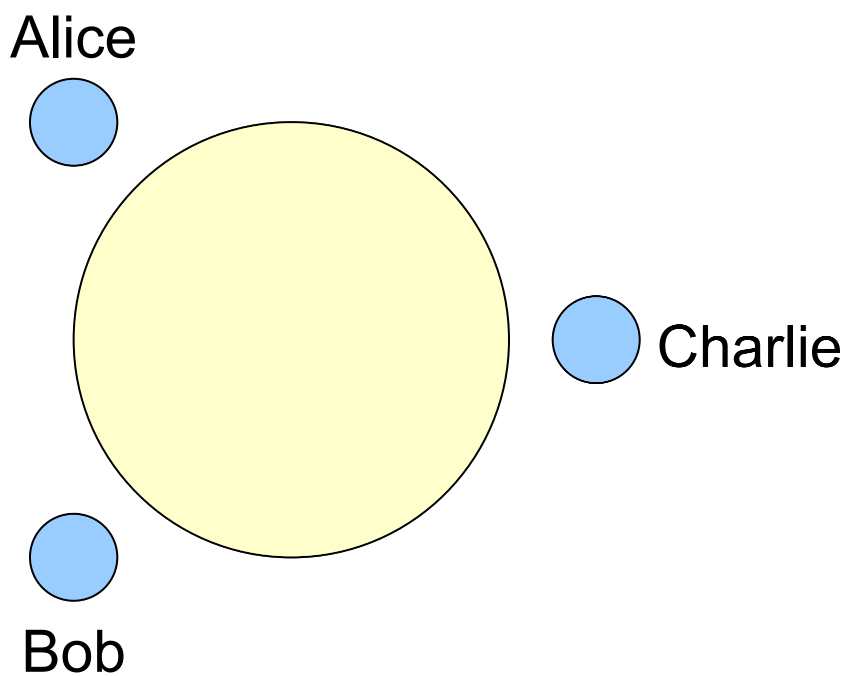
Properties of Onion Routing

- Key advantages:
  - Scalable to large groups of clients and relays
  - Can be made interactive (e.g., Tor)
  - Widely deployed (e.g., Tor)

- Key disadvantages:
  - Many vulnerabilities to traffic analysis
  - No accountability: Anonymous disruptors can
    - Spam or DoS-attack relays or innocent nodes
    - Compromise other users' anonymity [Borisov *et al.* '07]

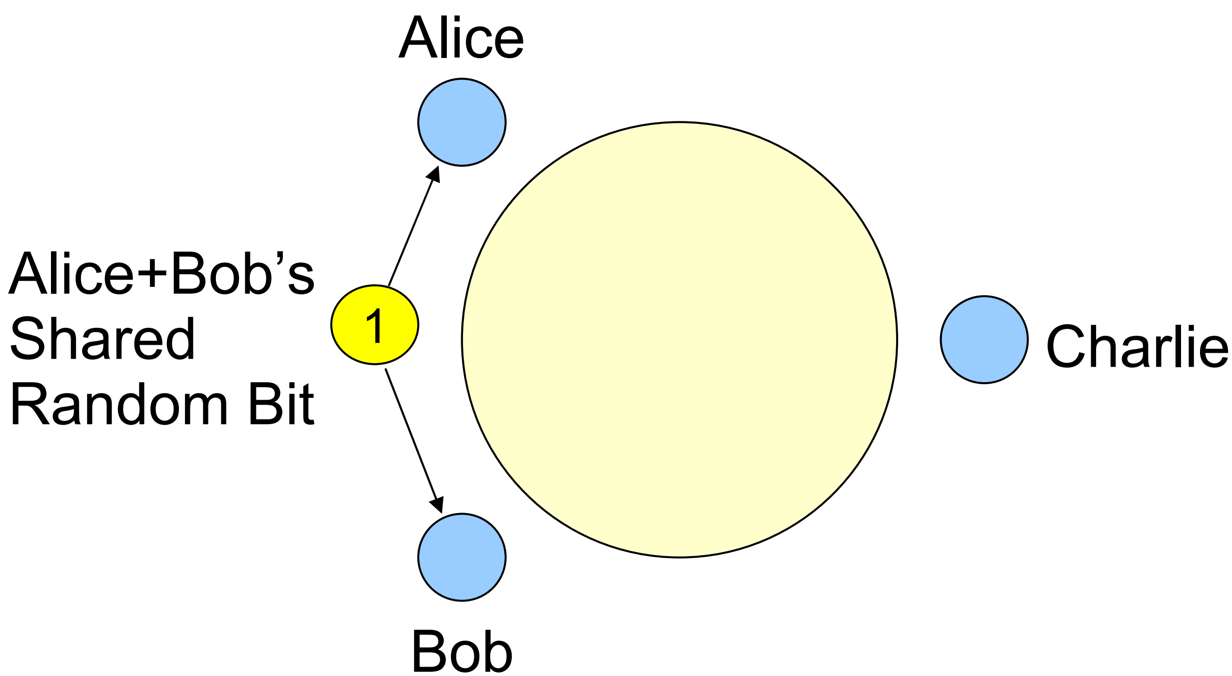
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 1: “Alice+Bob” sends a 1-bit secret to Charlie.



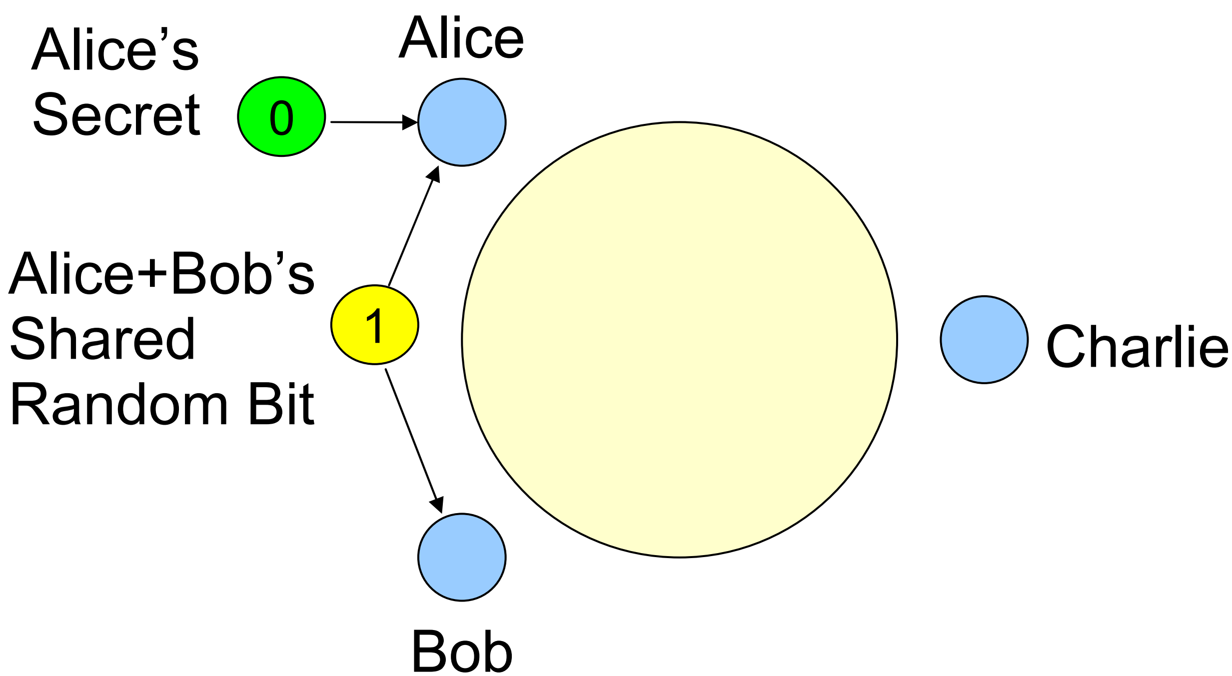
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 1: “Alice+Bob” sends a 1-bit secret to Charlie.



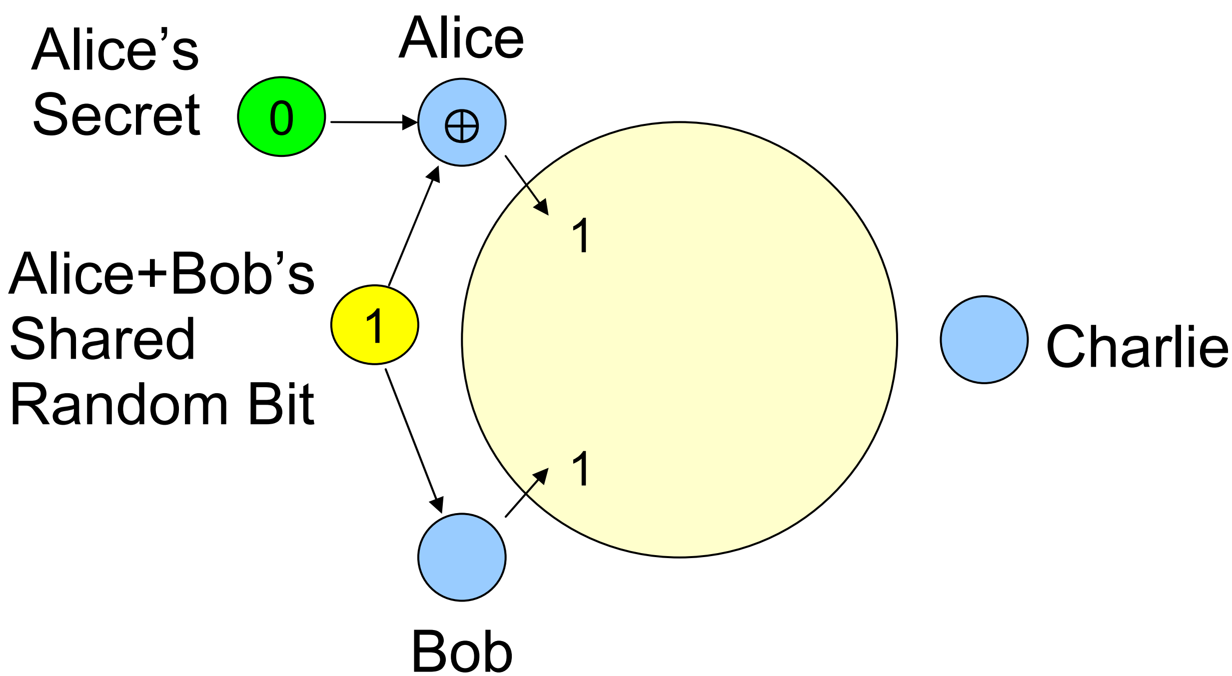
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 1: “Alice+Bob” sends a 1-bit secret to Charlie.



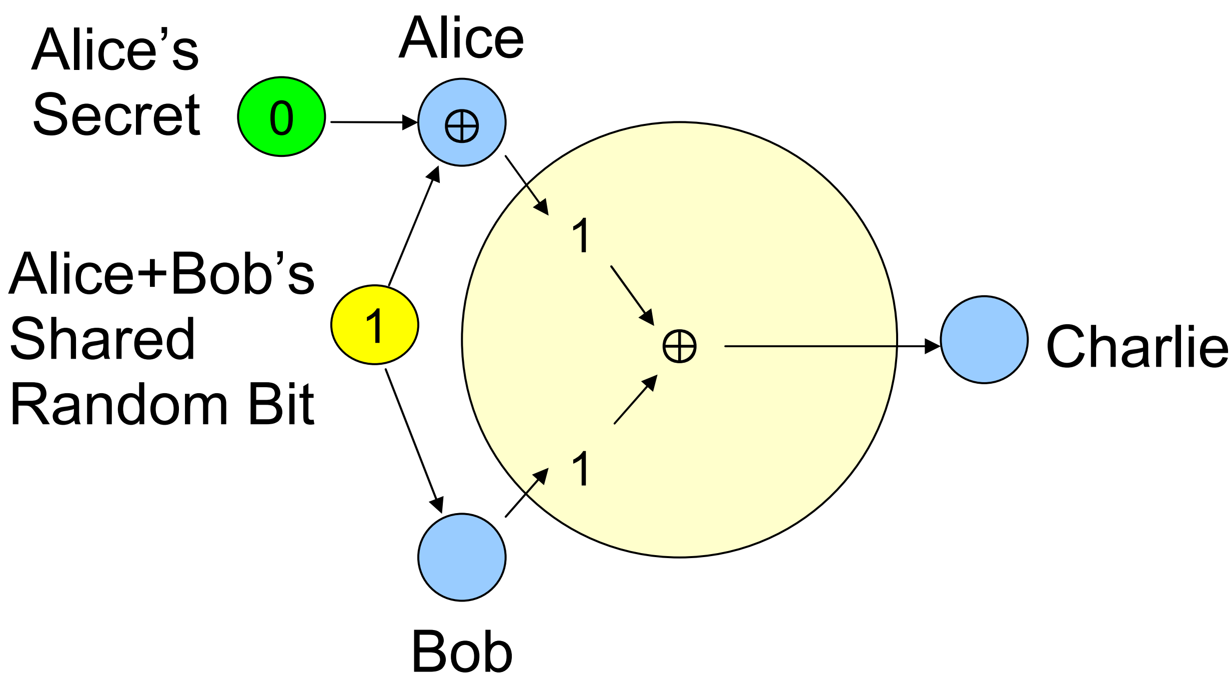
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 1: “Alice+Bob” sends a 1-bit secret to Charlie.



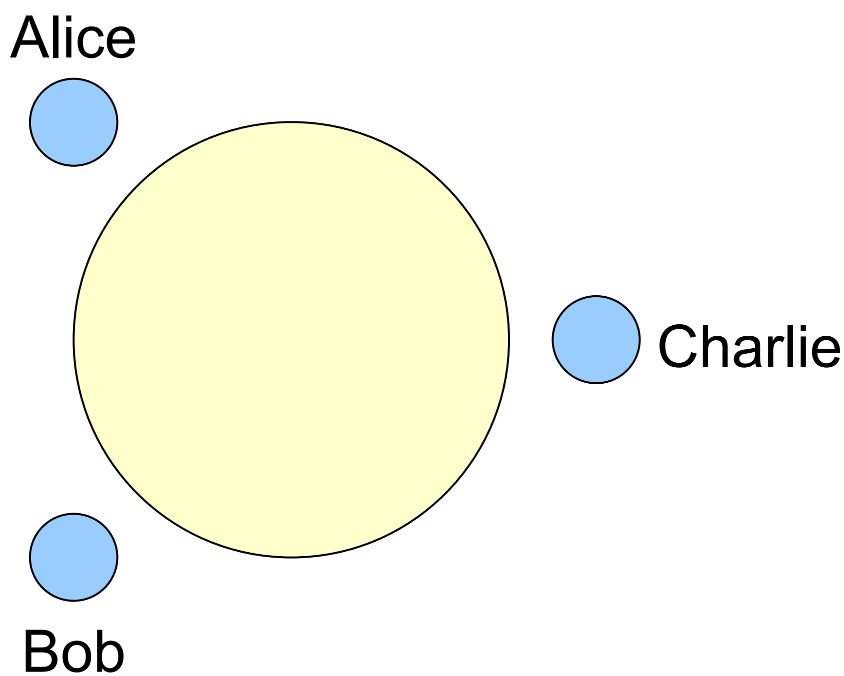
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 1: “Alice+Bob” sends a 1-bit secret to Charlie.



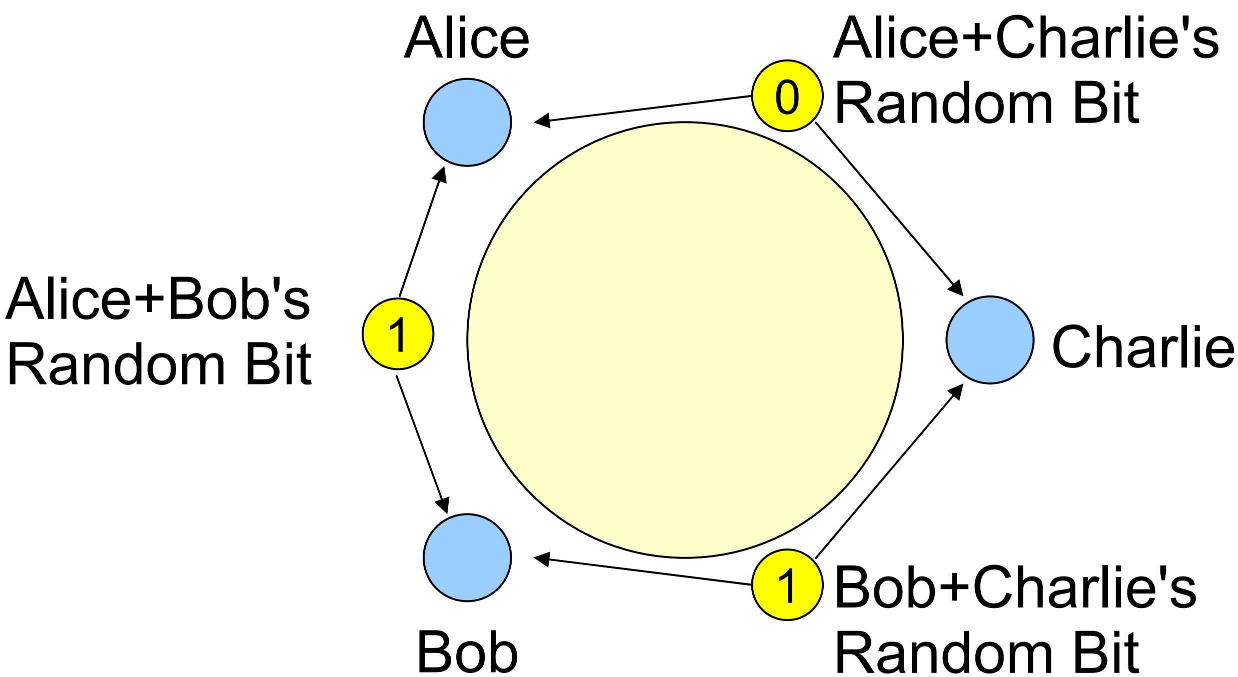
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 2: Homogeneous 3-member anonymity group



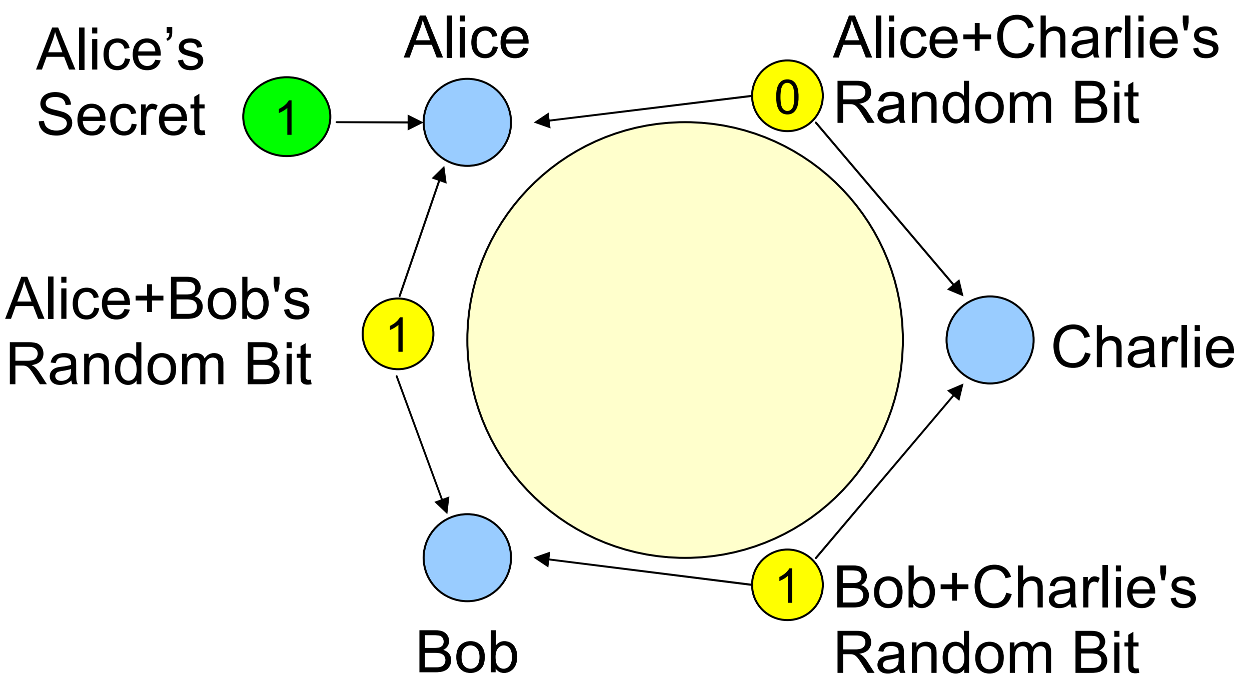
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 2: Homogeneous 3-member anonymity group



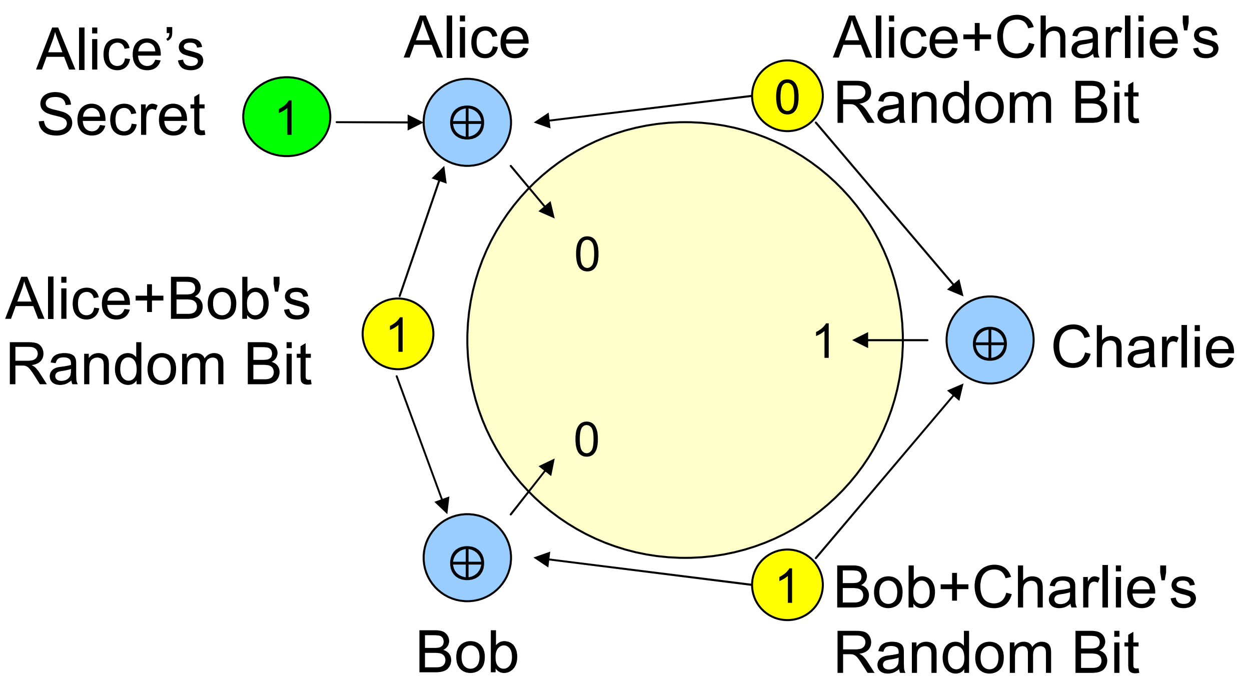
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 2: Homogeneous 3-member anonymity group



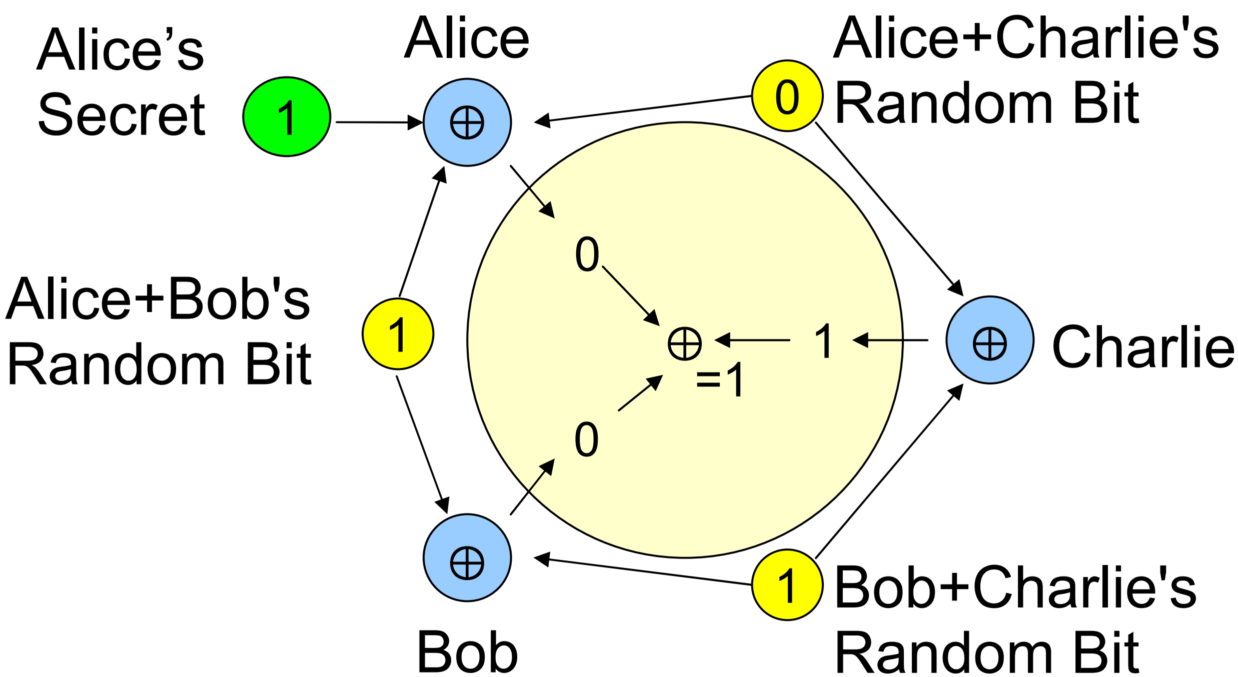
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 2: Homogeneous 3-member anonymity group



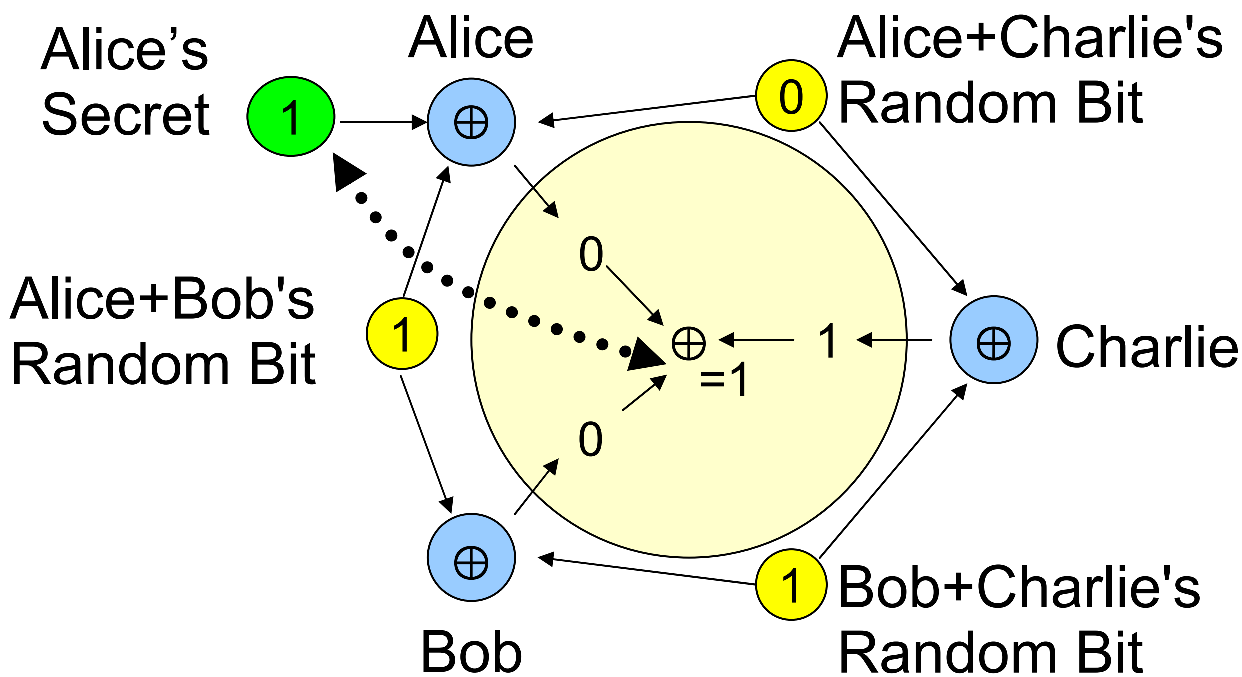
Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 2: Homogeneous 3-member anonymity group



Dining Cryptographers (DC-nets)

- Information-theoretic group anonymity
- Ex. 2: Homogeneous 3-member anonymity group



Properties of DC-nets Schemes

- Key advantages:
  - Provable, information-theoretic anonymity
  - Resistence to traffic analysis and collusion

- Key disadvantages:
  - Not easy to scale up or implement efficiently
  - Not widely deployed
  - No accountability: Anonymous disruptors can
    - Spam or DoS-attack the group without discovery
    - Force group reformation without being eliminated

Outline

- Prior work on anonymous communication
- ***Basic DISSENT protocol (ACM CCS 2010)***
- Results to date

Starting Point: Verifiable, Anonymous Shuffling [Brickell and Shmatikov '06]

- N parties with equal-length messages $m_1, \ldots, m_N$ send $m_{\pi(1)}, \ldots, m_{\pi(N)}$ to a data collector.
- The protocol provably provides
  - Integrity: $\{m_1, \ldots, m_N\} = \{m_{\pi(1)}, \ldots, m_{\pi(N)}\}$
  - Anonymity: π is random and not known by anyone.
  - Resistance to traffic analysis and collusion
- DISSENT adds accountability and the ability to handle variable-length messages efficiently.

Basic DISSENT Protocol: Overview

- Assumptions:
  - Equal-length messages
  - Each group member has a signature key pair; all messages are signed.
- Phase 1: Setup
  - Each member chooses two encryption key pairs for this run.
- Phase 2: Onion encryption
  - Each member encrypts his message with *everyone's* encryption keys.
- Phase 3: Anonymization
  - Each member applies a random permutation to the set of messages.
- Phase 4: Validation
  - Each member *i* checks that (uncorrupted) m_i is in the permuted set.
- Phase 5: Decryption or Blame
  - If all phase-4 checks succeed, decrypt all of the messages.
  - Else, honest members run a protocol that allows each of them to expose at least one disruptive member.

Phase 1: Setup

- Recall that
  - Members know each others' public verification keys.
  - Members sign (and verify signatures on) *all* messages.
- Each group member *i* chooses:
  - Secret message m_i (and pads it if necessary)
  - Outer encryption key pair (O_i, O'_i)
  - Inner encryption key pair (I_i, I'_i)
- Each group member *i* broadcasts public encryption keys O_i, I_i

Phase 2: Onion Encryption

Each group member *i*:

- Encrypts m_i with inner keys $I_N,\ldots,I_1$ to create m'_i
- Encrypts m'_i with outer keys $O_N,\ldots,O_1$ to create m''_i

Phase 2: Onion Encryption

Each group member *i*:

- Encrypts m_i with inner keys $I_N,\ldots,I_1$ to create m'_i
- Encrypts m'_i with outer keys $O_N,\ldots,O_1$ to create m''_i

Example with N = 3:

Phase 2: Onion Encryption

Each group member *i*:

- Encrypts m_i with inner keys $I_N,\ldots,I_1$ to create m'_i
- Encrypts m'_i with outer keys $O_N,\ldots,O_1$ to create m''_i

Example with N = 3:

m_1

m_2

m_3

Phase 2: Onion Encryption

Each group member *i*:

- Encrypts m_i with inner keys $I_N,\ldots,I_1$ to create m'_i
- Encrypts m'_i with outer keys $O_N,\ldots,O_1$ to create m''_i

Example with N = 3:

$$m'_1 = \{\ \{\ \{\ m_1\ \}I_3\ \}I_2\ \}I_1$$

$$m'_2 = \{\ \{\ \{\ m_2\ \}I_3\ \}I_2\ \}I_1$$

$$m'_3 = \{\ \{\ \{\ m_3\ \}I_3\ \}I_2\ \}I_1$$

Phase 2: Onion Encryption

Each group member *i*:

- Encrypts m_i with inner keys $I_N,\ldots,I_1$ to create m'_i
- Encrypts m'_i with outer keys $O_N,\ldots,O_1$ to create m''_i

Example with N = 3:

$$m''_1 = \{\ \{\ \{\quad m'_1 = \{\ \{\ \{\quad m_1 \quad \}I_3\ \}I_2\ \}I_1 \quad \}O_3\ \}O_2\ \}O_1$$

$$m''_2 = \{\ \{\ \{\quad m'_2 = \{\ \{\ \{\quad m_2 \quad \}I_3\ \}I_2\ \}I_1 \quad \}O_3\ \}O_2\ \}O_1$$

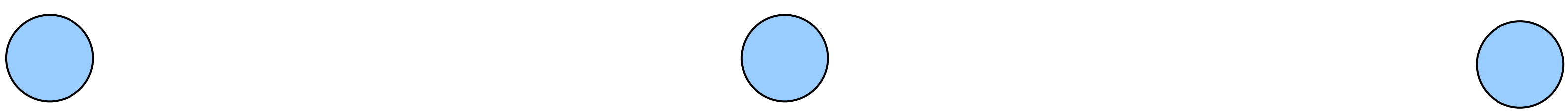
$$m''_3 = \{\ \{\ \{\quad m'_3 = \{\ \{\ \{\quad m_3 \quad \}I_3\ \}I_2\ \}I_1 \quad \}O_3\ \}O_2\ \}O_1$$

Phase 3: Anonymization (1)

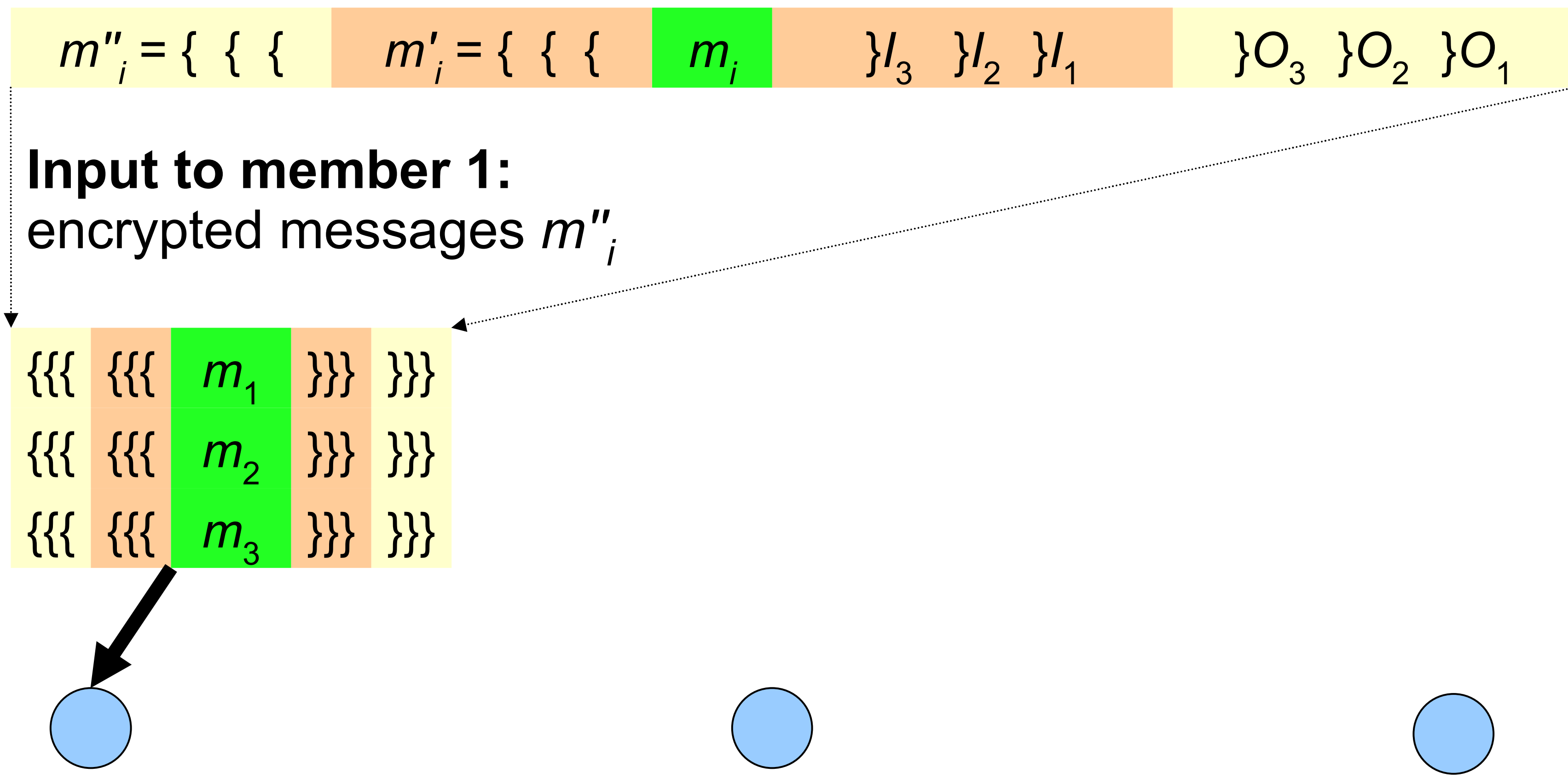
- Member 1 collects (m''_1, …, m''_N).
- For $i \leftarrow 1$ to N, member i
  - Decrypts the i^{th} layer of outer-key encryption
  - Randomly permutes the resulting list (of partially decrypted messages) and (temporarily) saves the random permutation
  - Forwards the permuted list to member $i+1$ (if $i < N$)
- Member N broadcasts the permuted m'_i list.

Phase 3: Anonymization (2)

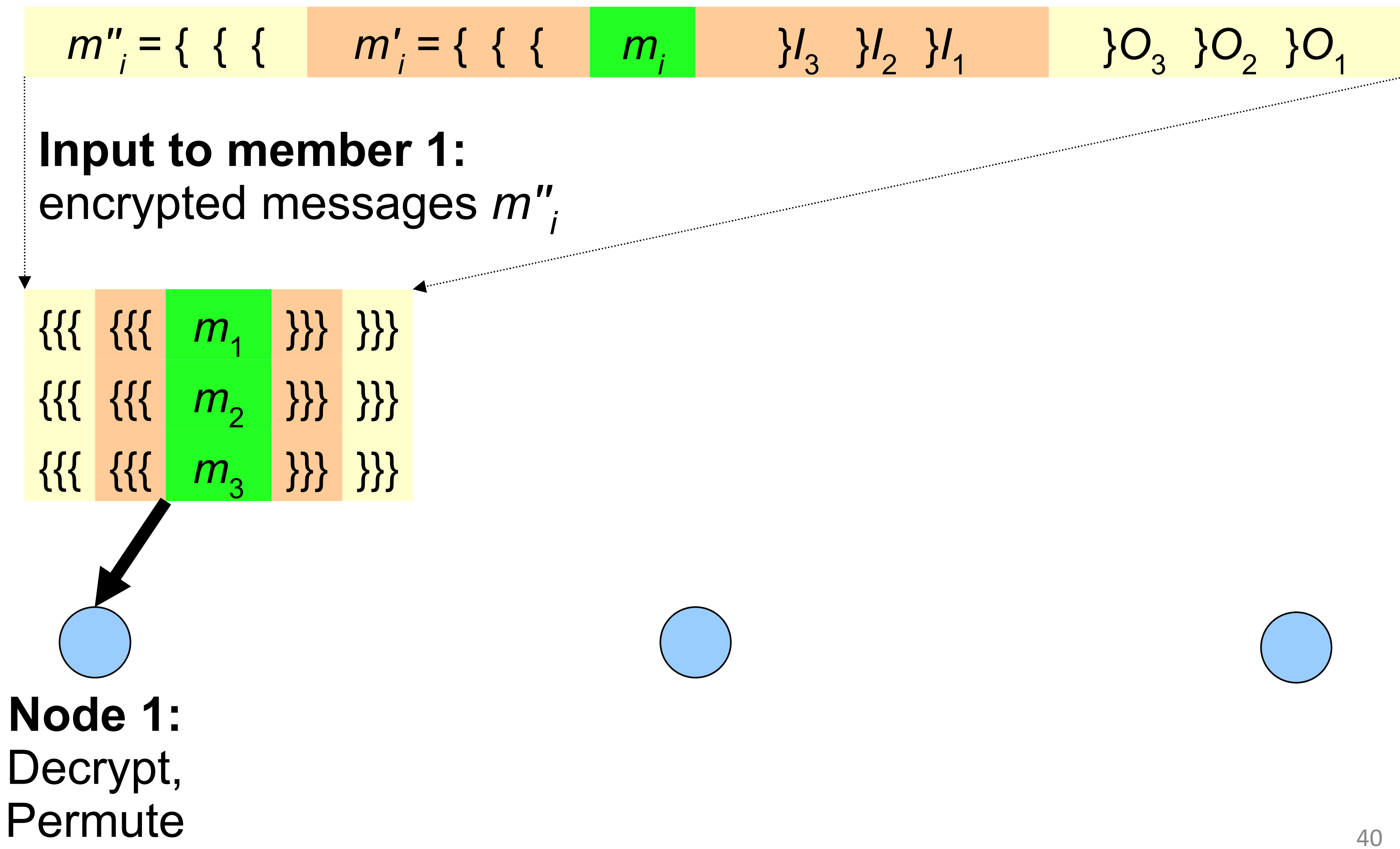
$m''_i = \{\ \{\ \{\quad m'_i = \{\ \{\ \{\quad m_i\quad \}I_3\ \ \}I_2\ \ \}I_1\quad \}O_3\ \ \}O_2\ \ \}O_1$



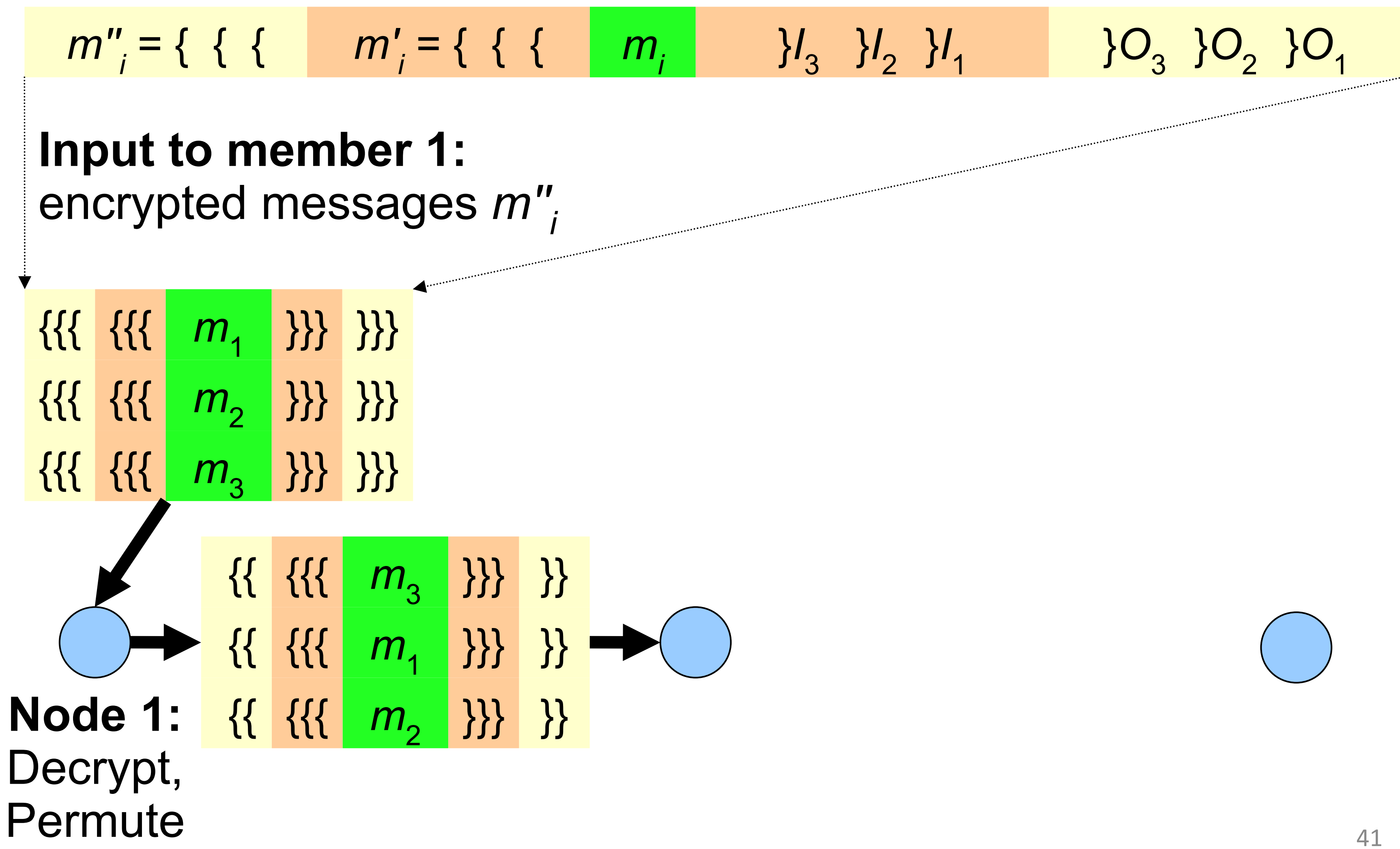
Phase 3: Anonymization (2)



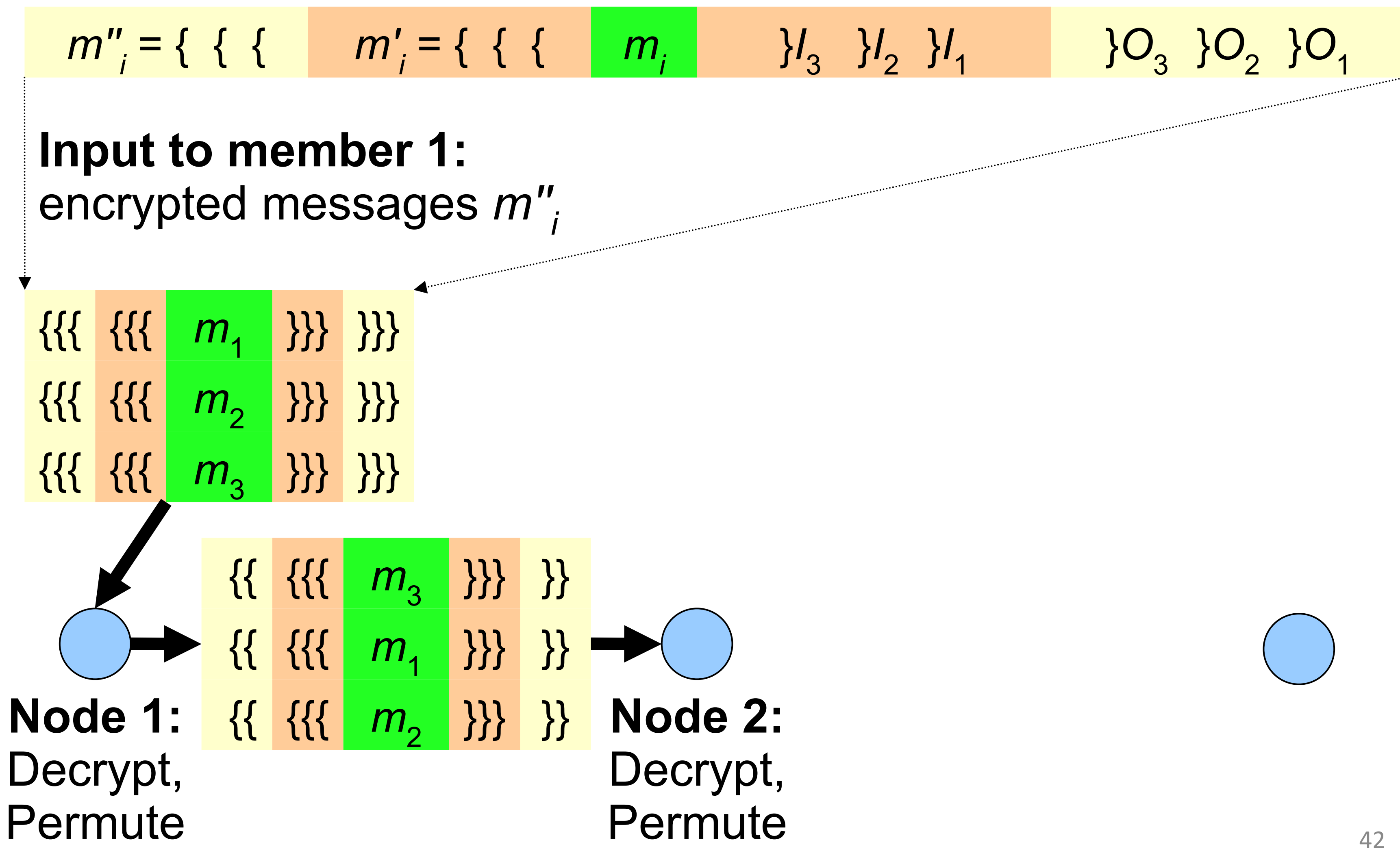
Phase 3: Anonymization (2)



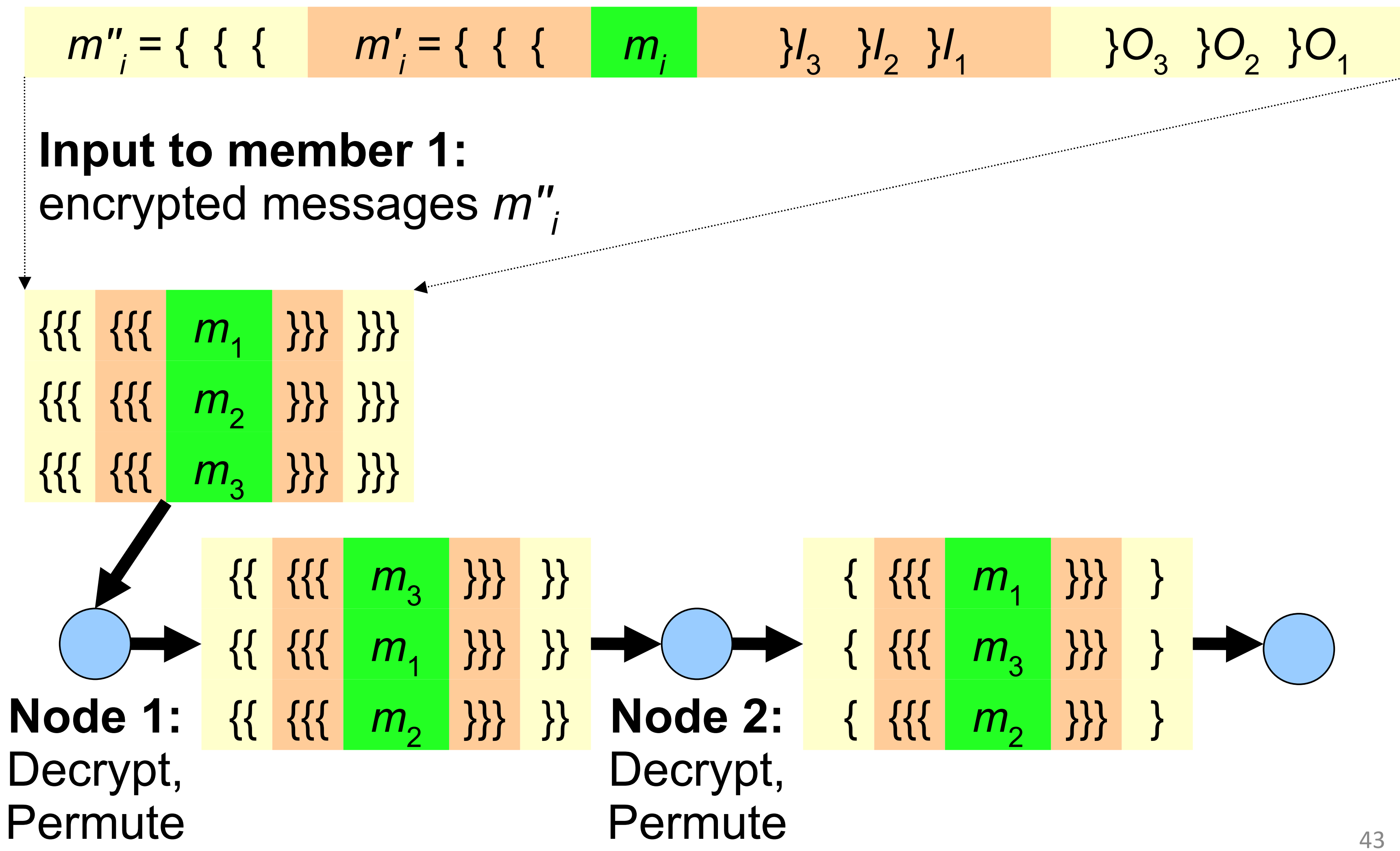
Phase 3: Anonymization (2)



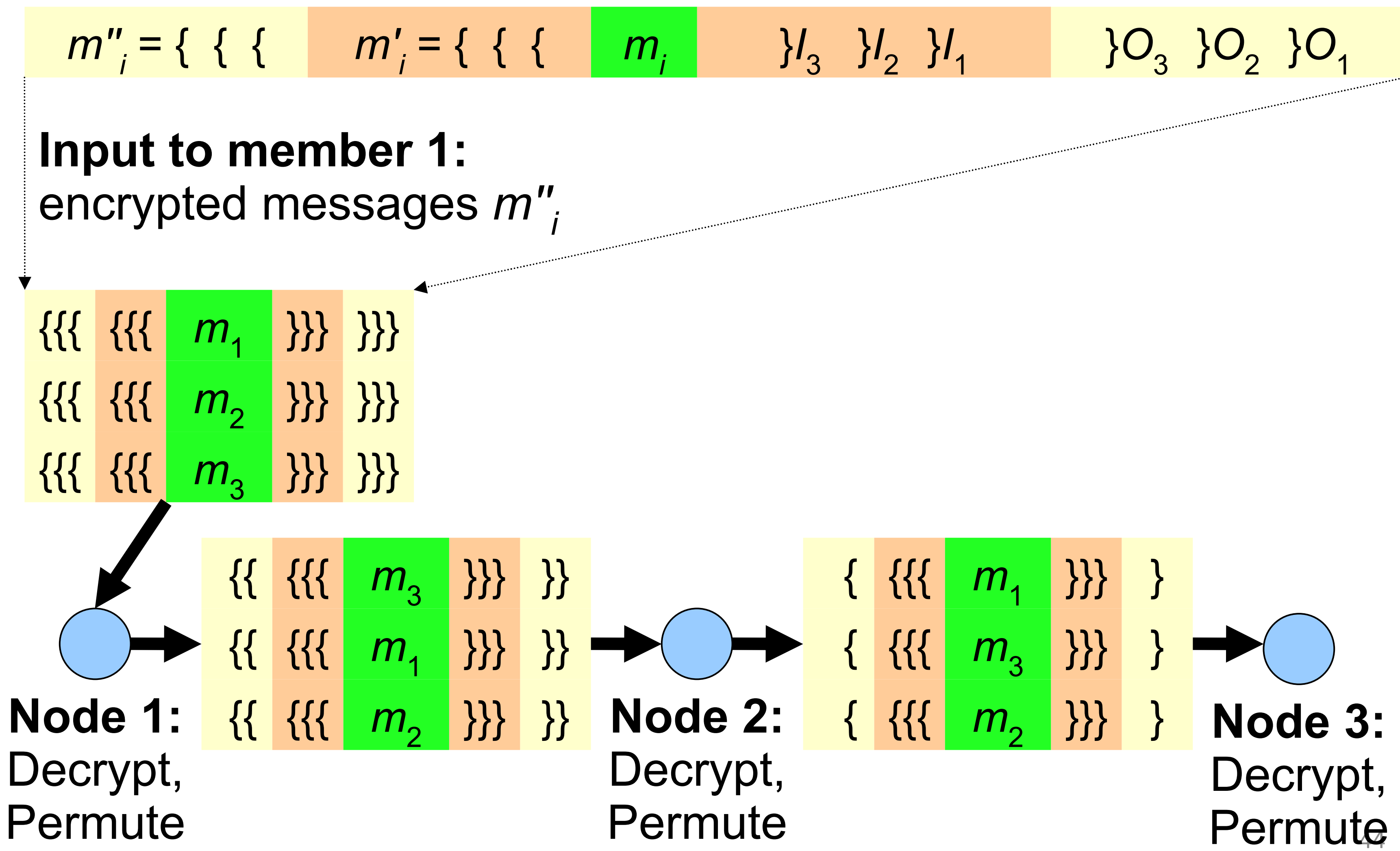
Phase 3: Anonymization (2)



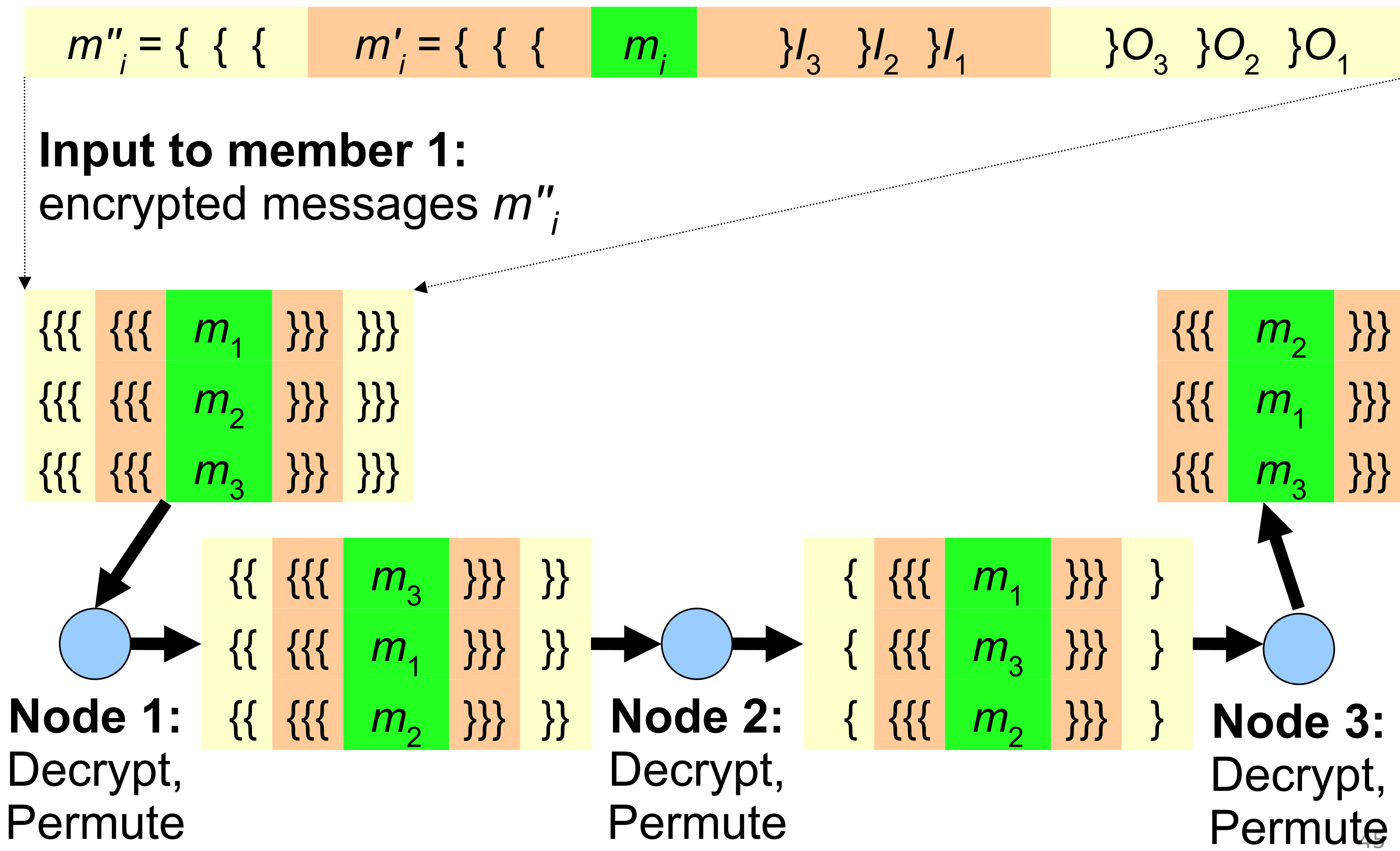
Phase 3: Anonymization (2)



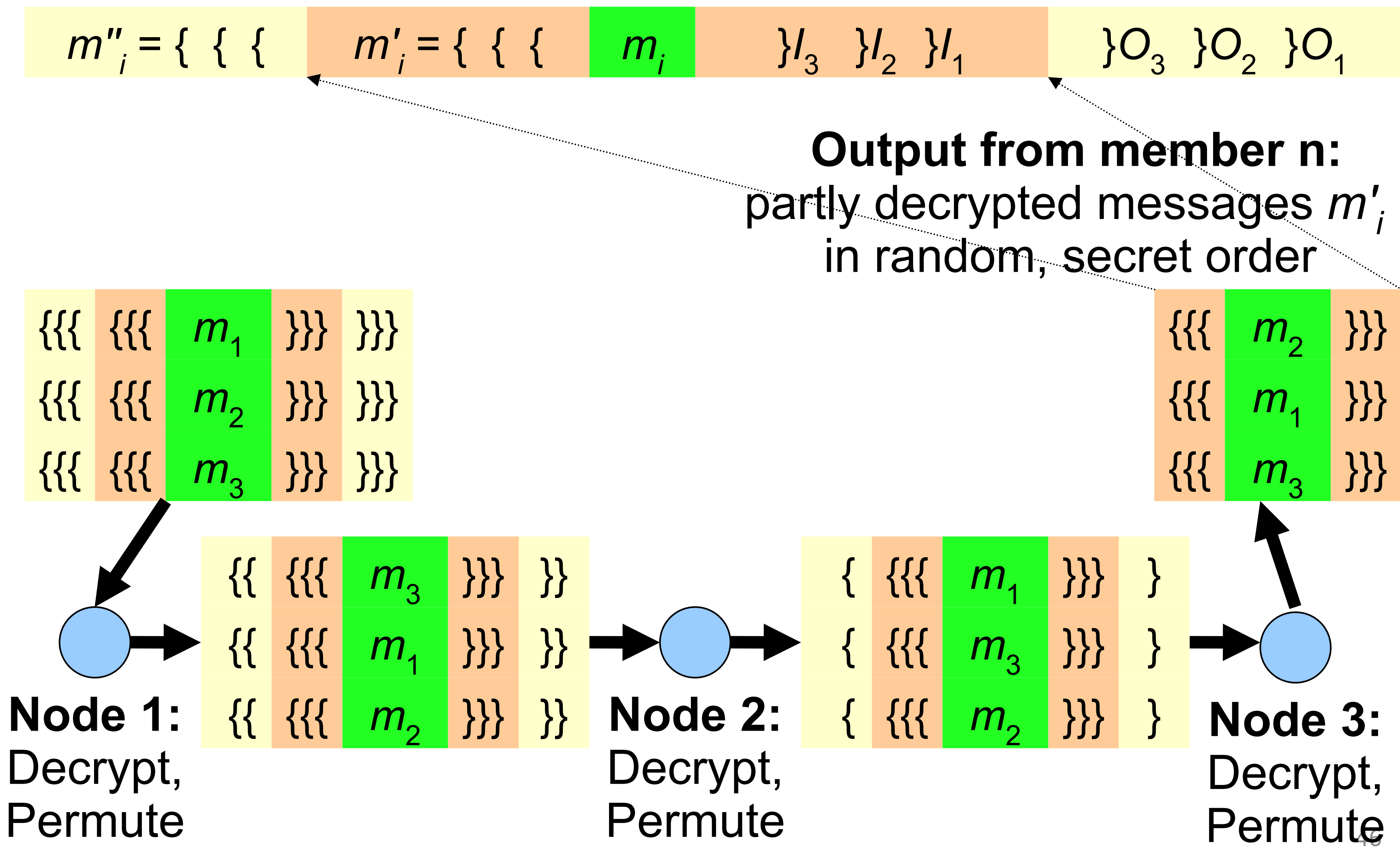
Phase 3: Anonymization (2)



Phase 3: Anonymization (2)



Phase 3: Anonymization (2)



Phase 4: Validation

After the anonymization phase, no member knows the final permutation, but every member *i* should see his own m'_i in the list!

Each member *i* looks for m'_i in the permuted list.

- **Present** → member *i* broadcasts "GO".
- **Absent** → member *i* broadcasts "NO-GO" and destroys his inner decryption key I'_i .

Phase 5: Decryption or Blame

- Each member *i* collects all GO/NO-GO messages.
- **GO messages from *all* nodes (including self):**
  - Each member *i* broadcasts his own inner decryption key I'_i .
  - All members use keys $I'_1,\ldots, I'_N$ to decrypt all the m'_j, revealing all the cleartext messages m_j.
- **NO-GO message from *any* node:**
  - Each member *i* broadcasts the proof that he decrypted and permuted properly in Phase 3.
  - All members use these proofs to expose disruptor(s).

How DISSENT Provides Accountability

- Any NO-GO message obliges *all* members to “prove their innocence,” i.e., that they:
  - correctly encrypted messages in Phase 2
  - correctly decrypted/permuted in Phase 3
  - correctly validated the final list in Phase 4
- This process reveals the “secret” permutation but leaves the permuted cleartexts m_j undecipherable: They are protected by all honest nodes' inner decryption keys, which have not been revealed.

Handling Variable-Length Messages

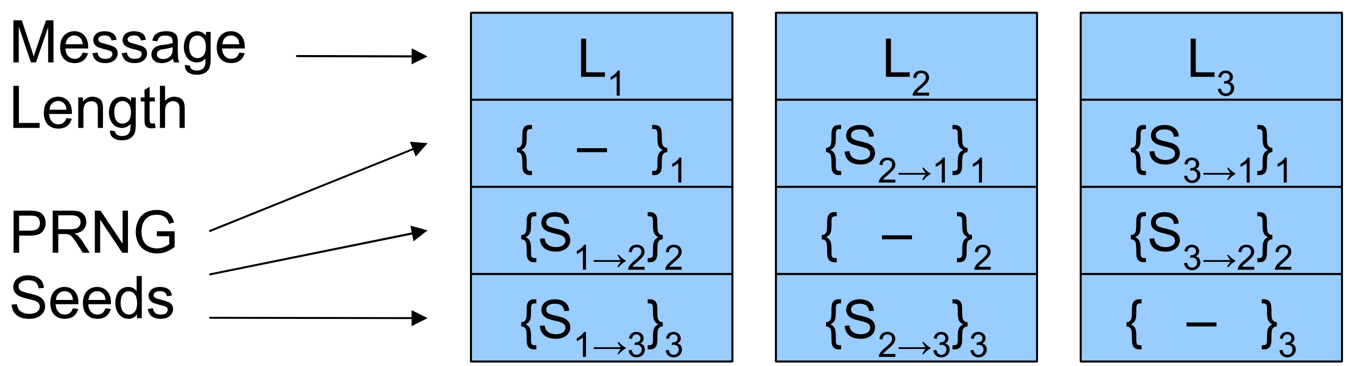
- Anonymous-shuffle protocols pad all messages to a common length in order to resist traffic analysis.
- What if the message load is unbalanced, e.g.:
  - Member *i* wants to send an L=646MB video.
  - Members $j \neq i$ have nothing to send in this run of the protocol.
- The group must shuffle the video and *N-1* 646MB padded cleartexts, resulting in O(NL) bits per node and O(N^2L) bits total.

Basic "Bulk Send" variant

- Use the (slow) accountable-shuffle protocol to exchange randomly permuted metadata.
- Interpret the random permutation as a "schedule" for exchange of data, which is done using DC-nets.
- Accountability of the DISSENT shuffle allows each group member to verify that all members transmitted the correct data in the proper DC-nets "timeslot."
- Cost of the case in which just one member wants to send *L*=646MB drops to O(*L*) bits per node and O(*NL*) bits total.

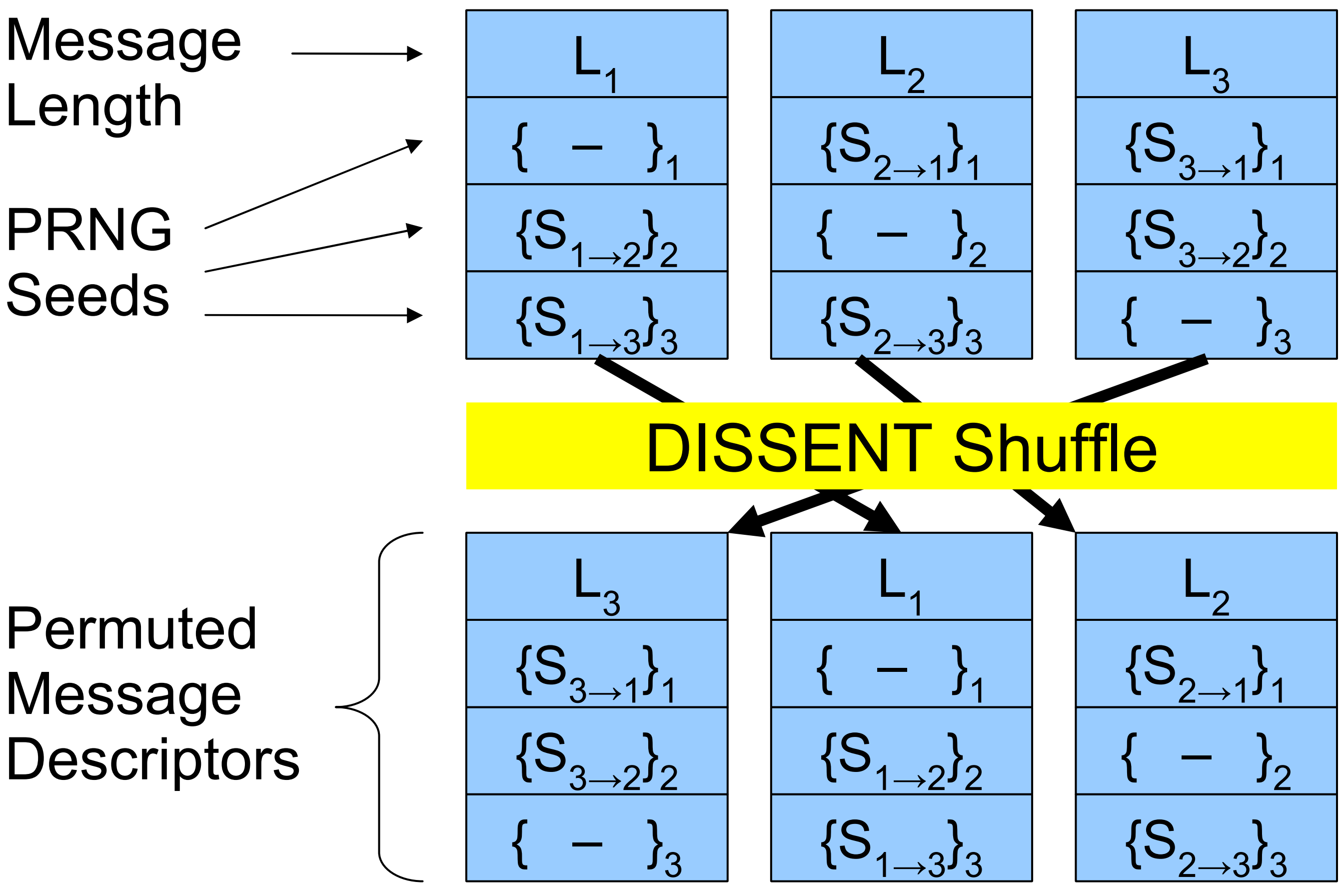
Basic Bulk Send (1)

Shuffle **metadata** describing the messages that the nodes want to send.



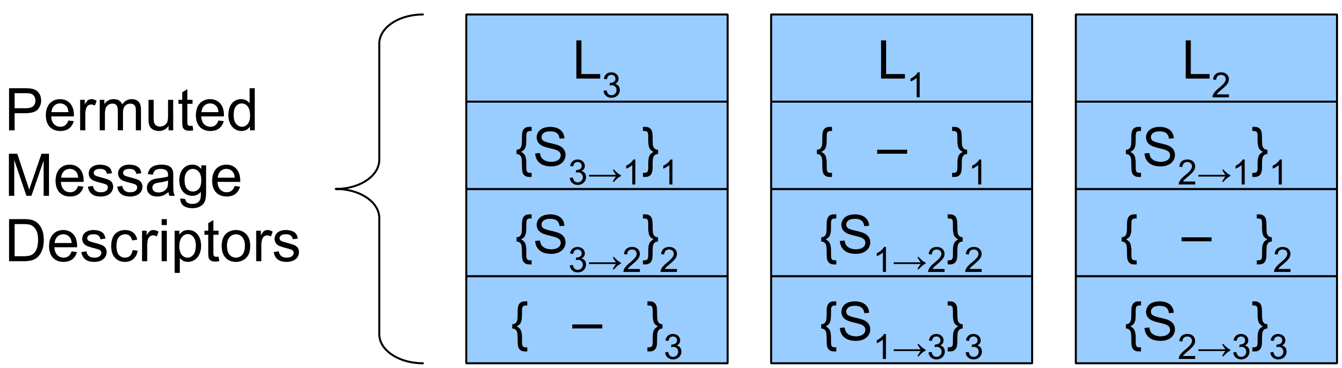
Basic Bulk Send (1)

Shuffle **metadata** describing the messages that the nodes want to send.



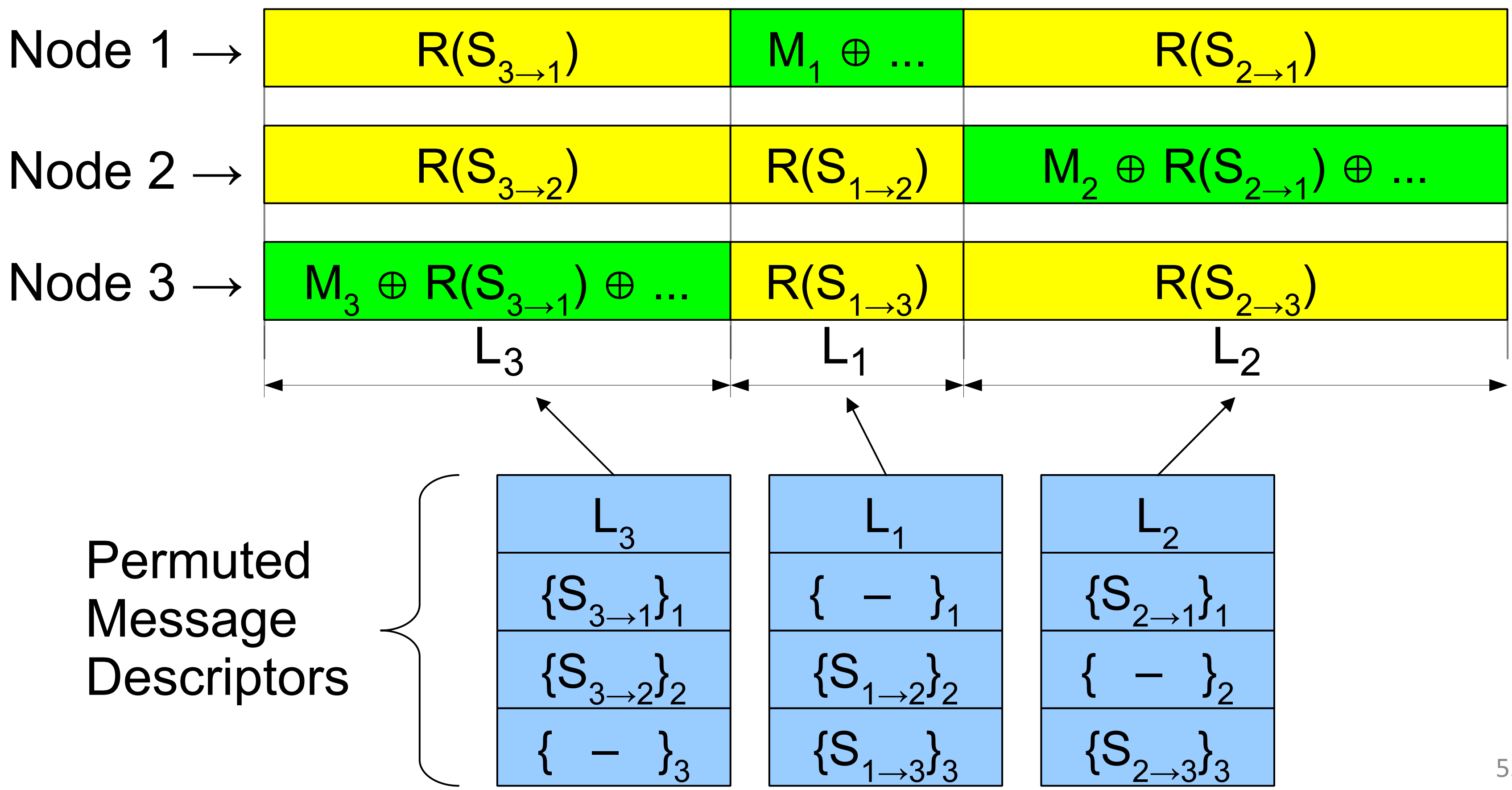
Basic Bulk Send (2)

The shuffled message descriptors form a **schedule** for a DC-nets transmission.



Basic Bulk Send (2)

The shuffled message descriptors form a **schedule** for a DC-nets transmission.



Outline

- Prior work on anonymous communication
- Basic DISSENT protocol (ACM CCS 2010)
- ***Results to date***

Results to Date (1)

- Reduced latency
  - Multiple bulk sends per shuffle
- Increased scalability (OSDI 2012)
  - Groups with 5000+ members
  - ***N*** clients, ***M*** servers
  - Secure against both active disruption by up to ***N-2*** clients and liveness attacks by a (tunable) constant fraction of clients. This enables ``churn tolerance.''
  - Secure against active disruption by up to ***M-1*** servers (but not against liveness attacks by servers).

Results to Date (2)

- Applications
  - "Anonymity scavenging" for wide-area microblogging
  - WiNon: DISSENT-based Web Browsing
    - ✓ "Strong, small" anonymity sets instead of the "large, weak" sets offered by Tor-based browsing tools
  - WiNon + Tor
    - ✓ Diverse, wide-area anonymity against weak attacker
    - ✓ Local-area anon./deniability if attacker can defeat Tor
- Formal proofs that basic DISSENT satisfies
  - Integrity
  - Anonymity
  - Accountability

Ongoing and Future Work

- Protection against ``intersection attacks’’
- Protection against liveness attacks on servers
- Formal security proofs for enhanced DISSENT protocols
- Integration with other anonymity protocols