Dissent: Accountable Anonymous Group Messaging

Sitora Salaeva

Reza Ghasemi

UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO MATEMATICA

A.Y. 2023/24

Definition: Anonymity

It refers to unknown authorship or origin.

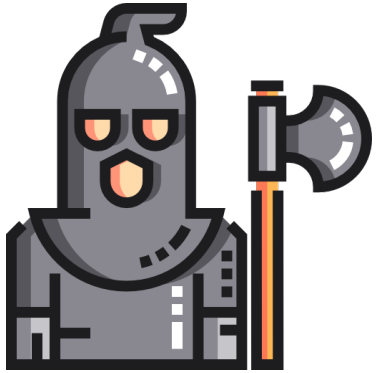
The term **anonymous** has Greek origins coming from "anonymos".

Identity ≠ Anonymity

Whereas **identity, ties a name and face to an individual**, **anonymity tries to remove them**.

Motives: Anonymity

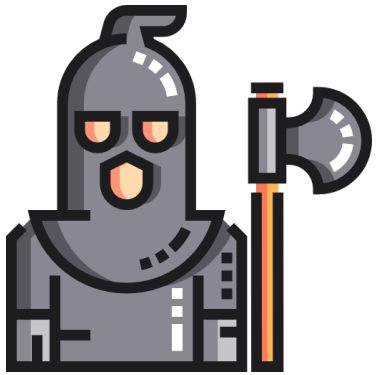
Anonymity has been mainly used as a tool for self protection.



Examples include:

- to avoid being called a heretic
- to fear for their lives due to a statement or an action

Motives: Anonymity (cont.)



Anonymity has been mainly used as a tool for self protection.

Examples include:

- to avoid being called a heretic
- to fear for their lives due to a statement or an action

Venetian masks allowed individuals in the past to cover their identities, giving them safe space to explore personal, romantic relationships outside of social restrictions.

Motives: Anonymity (cont.)

More specifically, In the 18th Century, the **Bauta mask** was mandatory during political decision making for acting anonymously.

Motives: Anonymity (cont.)

Anonymity and privacy are crucial to discuss topics which are considered **taboo** today.

Many positive social effects of today are results of having a private sphere to discuss what was considered unacceptable at some point:

- Legalization of pot
- Same-sex marriage
- Abolition of slavery

Motives: Anonymity (cont.)

Not just political changes, but for other motives as well.

Art:

- Banksy
- Elena Ferrante

Man is least himself when he talks in his own person.
Give him a mask, and he will tell you the truth. - Oscar Wilde

Motives: Anonymity (cont.)

Not just political changes, but for other motives as well.

Art:

- Banksy
- Elena Ferrante

Technology:

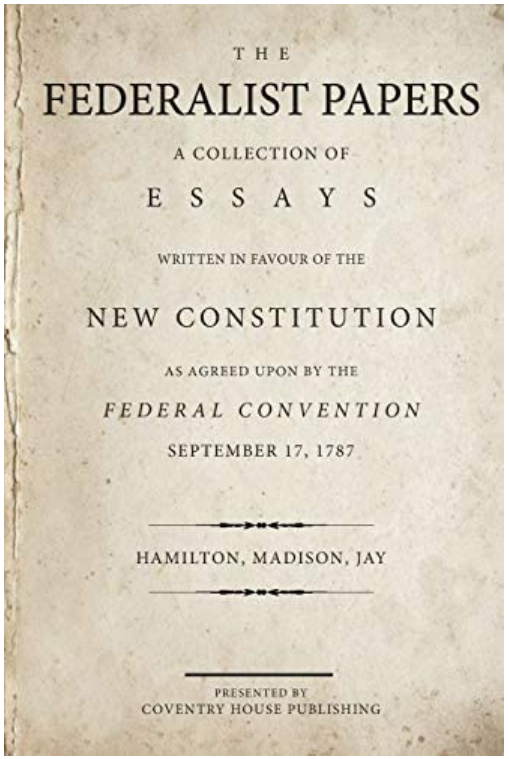
- Satoshi Nakamoto
- Nicolas van Saberhagen
- Creators of TrueCrypt

Historical Example

Collection of 85 political papers published under the pseudonym “Publius”

Three men wrote the papers:

- Alexander Hamilton
- James Madison
- John Jay



DC-Net

The original model for **anonymous group communication** was proposed by **David Chaum** in **1988** composed of **three participants** and messages were **single bits**.

He proved that it is possible to achieve **unconditional sender and recipient untraceability**.

DC-Net (cont.)

The original model for **anonymous group communication** was proposed by **David Chaum** in **1988** composed of **three participants** and messages were **single bits**.

He proved that it is possible to achieve **unconditional sender and recipient untraceability**.

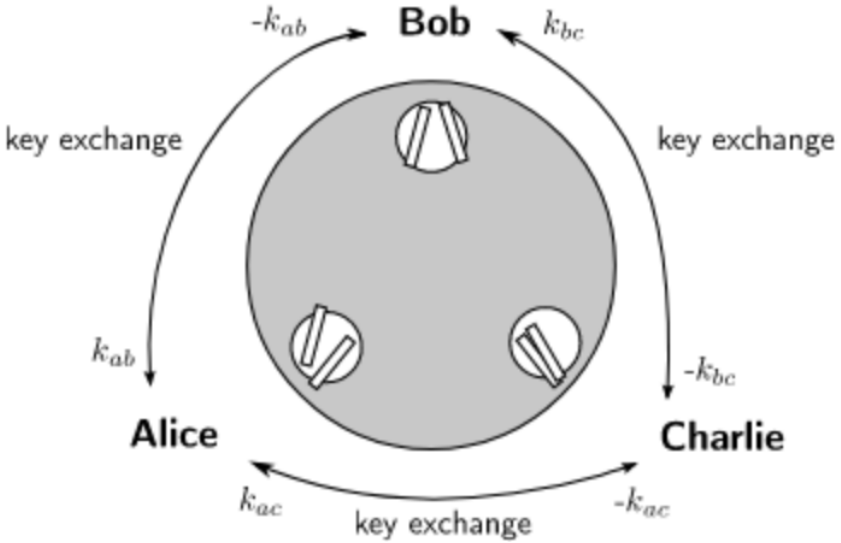
DC-Net is the generalized model which can have more number participants and larger length of messages.

Dining Cryptographers Problem

The Problem: Three cryptographers want to identify whether it was one of them or it was the NSA who paid for the dinner.

A two stage protocol is initiated to identify (while respecting privacy of cryptographers) the payer.

Dining Cryptographers Problem (cont.)



Stage 1: A coin is tossed between two cryptographers to determine a one bit shared secret key among pairs

K_{AB} : Secret key of Alice and Bob

K_{AC} : Secret key of Alice and Charlie

K_{BC} : Secret key of Alice and Bob

Dining Cryptographers Problem (cont.)

Stage 2: Each cryptographer must announce a bit.

If a cryptographer paid for the dinner, the opposite of XOR is announced.
Otherwise XOR of secret keys must be announced.

If the final result is one, the payer was one of the cryptographers.
If zero, it was the NSA.

Dining Cryptographers Problem (cont.)

Example: Alice paid for the dinner

K_{AB} : 0

K_{AC} : 1

K_{BC} : 1

Alice $K_{AB} \oplus K_{AC} = 0 \oplus 1 = 1 \rightarrow 0$

Bob $K_{AB} \oplus K_{BC} = 0 \oplus 1 = 1 \rightarrow 1$

Charlie $K_{AC} \oplus K_{BC} = 1 \oplus 1 = 0 \rightarrow 0$

$$0 \oplus (1 \oplus 0) = 1$$

We identified who payed for the dinner without revealing his identity!

DC-Net: Limitations

Collision: Due to nature of XOR operator, if two cryptographer paid for the dinner, result would cancel out (meaning zero is announced).

Disruption: Attacker could send false bits and disrupt the network.

Complexity: Each pair of cryptographers must generate a secret key among themselves, hence for a **network of size n** we require **n(n-1)/2 secret keys**

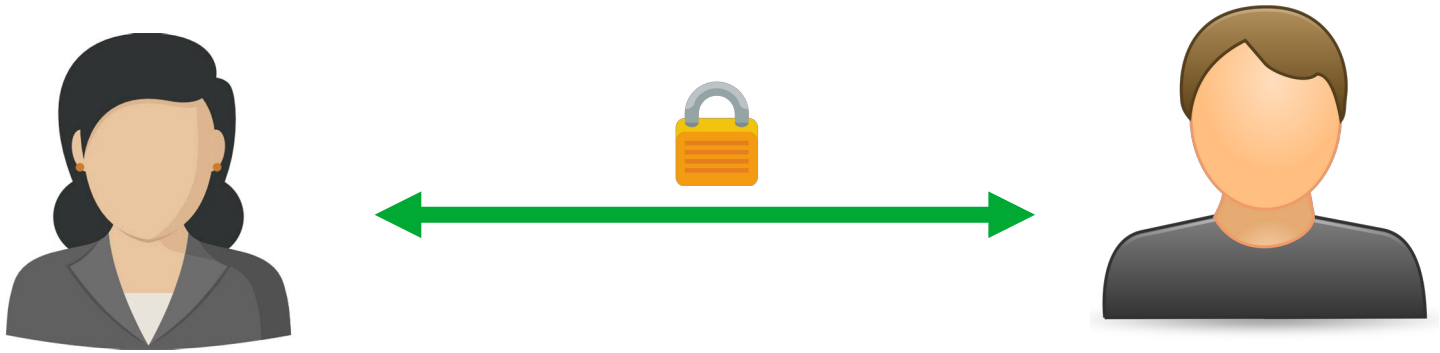
Possible solution to collision: Re-transmit once collision is detected.

Mixnets

Definition: A form of network architecture aiming to protect sender's location.

The idea was originally introduced in **"Untraceable electronic mail, return addresses, and digital pseudonyms"** by **Chaum** in **1981**.

Issue with encryption: Even though communication is secure, Eve still knows there is a communication happening between Alice & Bob!



Mixnets (cont.)

Definition: A form of network architecture aiming to protect sender's location.

The idea was originally introduced in **"Untraceable electronic mail, return addresses, and digital pseudonyms"** by **Chaum** in **1981**.

Issue with encryption: Even though communication is secure, Eve still knows there is a communication happening between Alice & Bob!

Mixnets (cont.)

Mixnets **aim to hide the correspondence** between items. To achieve untraceability, mixnet **stores n messages**, **shuffles them, then sends them out**.

Additionally, to prevent loss of confidentiality, message is encrypted twice:

$$E_{key2}(\ E_{key1}(input)\ ) \rightarrow \boxed{\text{mix node}} \rightarrow E_{key1}(input)$$

In this manner, we avoid loss of confidentiality by not allowing the mix node and Eve to see the plaintext.

Dissent Protocol

Dissent (Dining-Cryptographers Shuffled-Send Network) tries to solve issues which mixnets have with timing attacks and DC-nets have with Sybil attacks and DoS attacks.

Characteristics:

- Usable on medium-scale groups (prototype worked in a group of 40)
- Preserves message integrity and one-to-one correspondence
- Efficient handling of large message loads

Dissent operates in two phases: **1) Shuffle** **2) Bulk transfer**

Dissent Protocol (cont.)

Built on the existing data mining protocol of Brickell/Shmatikov.

A set of fixed length of messages is permuted and broadcasted with cryptographic strong anonymity.

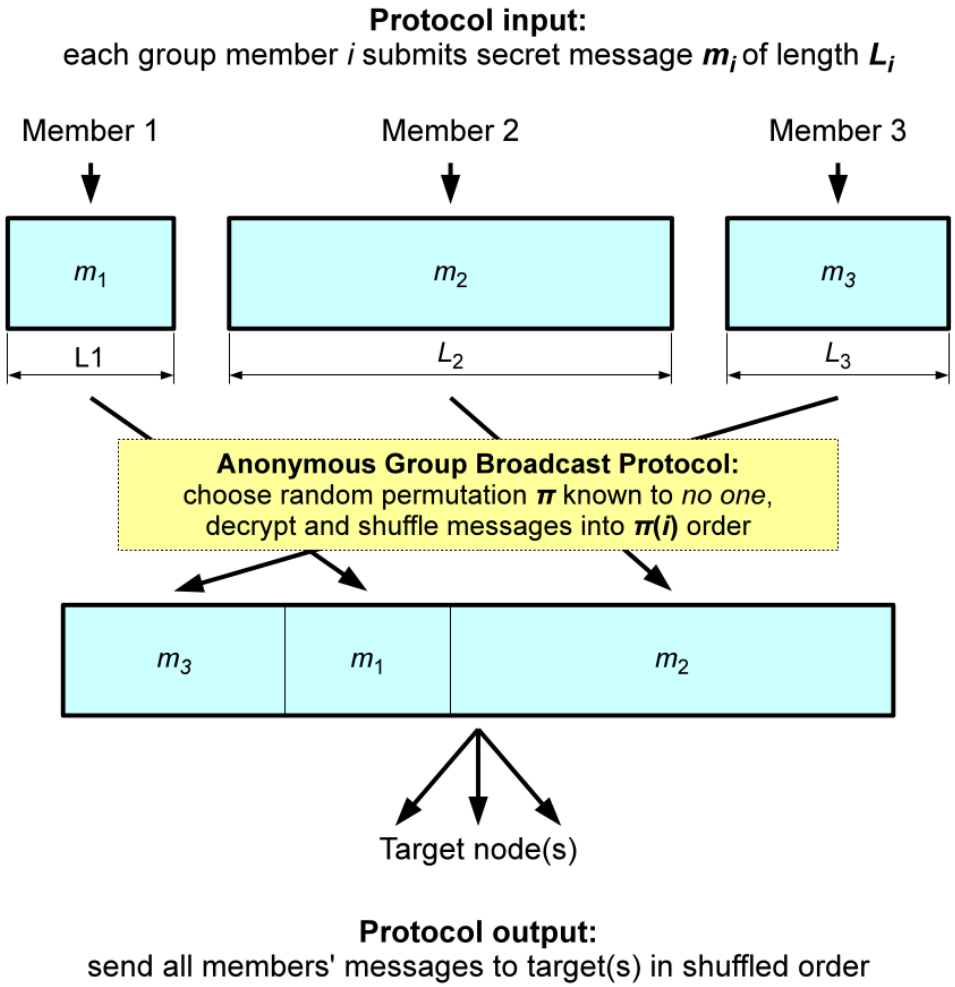
Dissent uses go/no go and blame phases to trace malicious users and prevent them from distrupting the network. (existing issue in DC-net is solved here)

It has **two limitions**:

1) All messages must be of equal length L → incurring O(NL) extra messages required even if only a single user wants to send a message

2) Decrypt and shuffle is serial → If N or L is large, decryption time will be long

Dissent Protocol (cont.)



Dissent: Signature Scheme

We have the following for the signature scheme:

- **Key generation algorithm**: public and private key is generated

$$(u, v)$$

- **Signing algorithm**: a signature (σ) is generated.

$$\sigma = Sig_u(m) \qquad u: \text{private key}$$

- **Verification algorithm**: checking whether σ is correct signature of m given v

Dissent: Cryptosystem

We have the following public-key cryptosystem:

- **Key generation algorithm**: public and private key is generated

$$(x, y)$$

- **Encryption algorithm**: it produces a ciphertext given public key, some random bits and a message.

$$C = \{m\}^R_y$$

- **Deterministic decryption algorithm**: given private key (x) and ciphertext (c), plaintext (m) is returned.

RSA-OAEP using a **pseudo-random number generator** could be used.

IND-CCA2 Notation

The public-key cryptosystem must be **IND-CCA2 secure**.

Algorithm:

i. Challenger: KG = (K_E, K_D)

ii. Adversary: may call encryption and decryption oracle as many times as he wishes

iii. Adversary: *m0*, *m1* = two messages of same length are chosen

iv. Challenger: *b* = chooses between 0 and 1 randomly

v. Challenger: C = E(K_E, m_b) and sends C to adversary

vi. Adversary may perform additional operations (including calling to oracles) and outputs ***guess***

Vii. If guess = b then adversary wins

Dissent: Shuffle sub-protocol

Steps:

n_R: nonce

u_i: private key

$h_{i\varphi}$: hash of member I at current log in phase φ

$\mu_{i\varphi}$ = unique message at phase φ

Phase 1: Generate secondary keypair (w_i, z_i) and each member (i) broadcasts the following:

$$\mu_{i1} = \{z_i, n_R, h_{i1}\}\mathrm{SIG}_{u_i}$$

Phase 2: Each member will encrypt her datum (m_i) with all members public keys:

$$C_i' = \{m_i\}_{z_N:z_1}$$

Member will store this for later, it will continue to encrypt it with public keys of other members and stores random bits:

$$C_i = \{C_i'\}_{y_N:y_1}^{R_{iN}:R_{i1}}$$

If encryption fails (for any reason), we jump to blame phase. The following will be sent to member 1 by member i:

$$\mu_{i2} = \{C_i, n_R, h_{i2}\}\mathrm{SIG}_{u_i}$$

Dissent: Shuffle sub-protocol

Steps:

If C_i' is included in C_N then GO will have the flag = true, otherwise it will be false

Phase 3: For anonymization, member 1 collects all ciphertexts into $\vec{C}_0 = C_1, \ldots, C_N$ and randomly permutes them.

Then member 1 builds $\vec{C}_1$ by stripping one encryption layer using private key x_1 and sends the following to member 2:

$$\mu_{13} = \{\vec{C}_1, n_R, h_{13}\}\text{SIG}_{u_1}$$

Member i will wait to receive Go_j to be True from every member j for HASH{B}, if not, it will go to blame phase.

Dissent: Shuffle sub-protocol

Steps:

Phase 4: To verify, each member will verify her own C_i' is in $C_N^{\rightarrow}$ which is a permutation of $C_1', \ldots, C_N'$.

Then it will create **vector B** of all messages that have been broadcasted and member i will broadcast the following:

$$\mu_{i4} = \{\mathrm{GO}_i, \mathrm{HASH}\{\vec{B}\}, n_R, h_{i4}\}\mathrm{SIG}_{u_i}$$

member i will wait to receive Go_j to be True from every member j for HASH{B}, if not, it will go to blame phase.

Dissent: Shuffle sub-protocol

Steps:

Phase 5a (Decryption): To decrypt, each members destroys saved random bits and C_i' and broadcasts secondary private key w_i to all members :

$$\mu_{i5} = \{w_i, n_R, h_{i5}\}\mathrm{SIG}_{u_i}$$

After receiving all keys, member(i) will check if each w_j corresponds to z_j. If not, she exposes it. Otherwise levels of encryption are removed and the protocol is completed with success.

Phase 5b (Blame): secondary private keys (wi) are destroyed. Then all random bits stored (Rij) are announced to all members.

If member j signed a faulty or incorrectly encrypted or when the hash was wrong, … member I will expose j.

BULK PROTOCOL

Protocol Description

- Builds on DC-net
- Transmits messages of variable length anonymously
- Operates in 5 phases:
  - **Phase 1. Message Descriptor Generation**

$$d_i = \{L_i, HASH\{m_i\}, \vec{H}_i, \vec{S}_i\}$$

L_i - pseudorandom bits, m_i - message,
$\vec{H}_i$ - vector of cipher text hashes, $\vec{S}_i$ - vector of encrypted seeds

  - **Phase 2. Message Descriptor Shuffle**

    Runs Shuffle Protocol and broadcasts all descriptors

Protocol Description

- **Phase 3. Data Transmission**

  Signs and sends each ciphertext to the target

- **Phase 4. Message Recovery**

  Checks each ciphertext hash and recovers a message

- **Phase 5. Blame**

  Checks corrupted message

Protocol Correctness

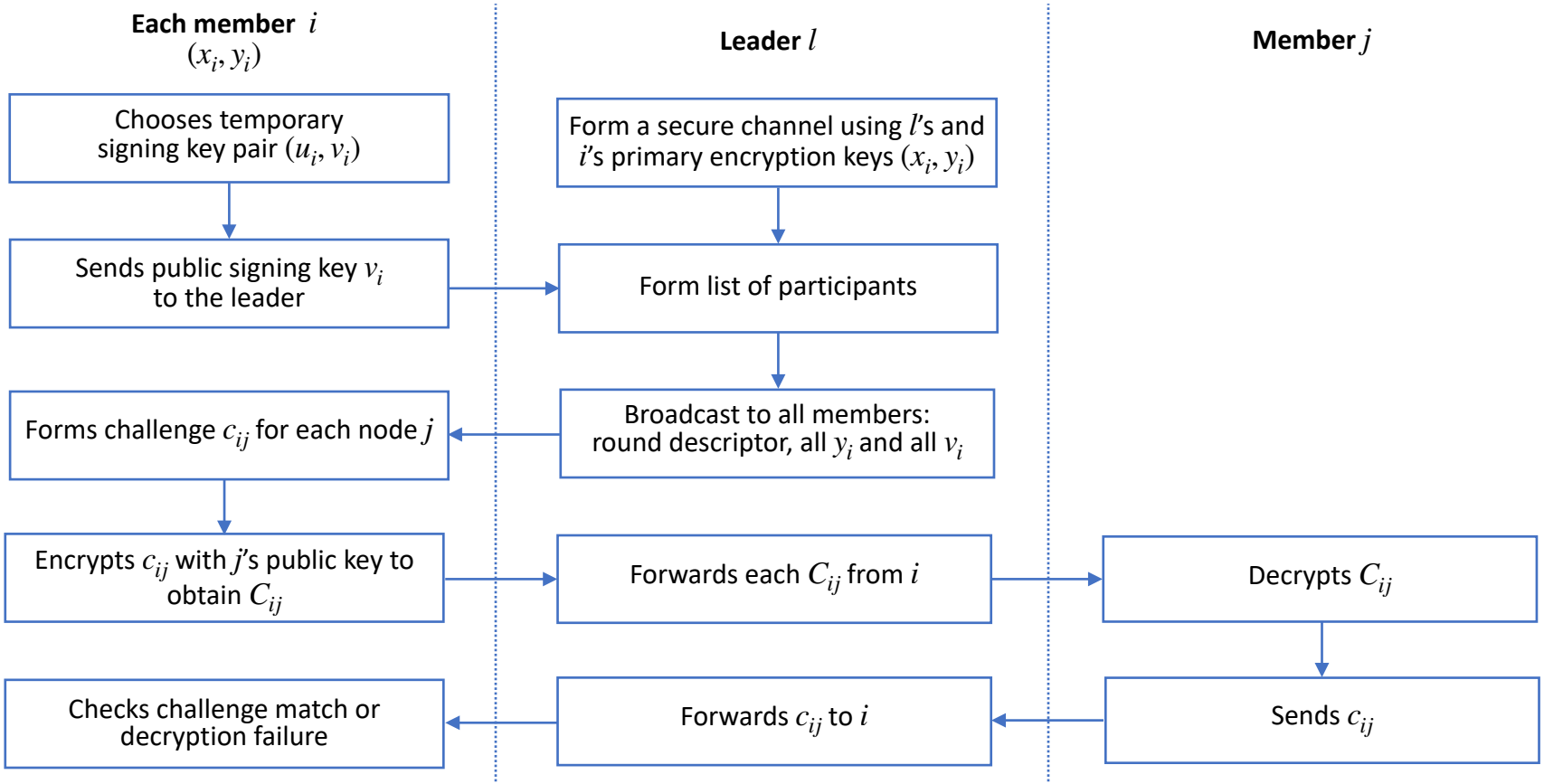
- **Integrity**
  - Uses H_{ij} or $HASH\{m_i\}$ from d_i
- **Anonymity**
  - Shuffle protocol preserves anonymity
  - Shuffled message descriptor depends only on random bits
- **Accountability**
  - Dishonest member cannot expose honest member
  - Correctly computed hashes and encrypted seeds can reveal message

Wrapper protocol

- **Protocol initiation**
  - each member of the group periodically initiates the execution of the protocol
- **Member selection**
  - the leader of run is responsible for detecting which members are available now to agree on a set of members

Wrapper protocol

- **Deniable keying**



Wrapper protocol

- **Liveness assurance**
  - member j suspects member i and inform leader l
  - l demands a signed copy of i's message for j:
    - No response:
      - l notifies all members
      - start protocol without i
    - Otherwise:
      - i sends the message

Wrapper protocol

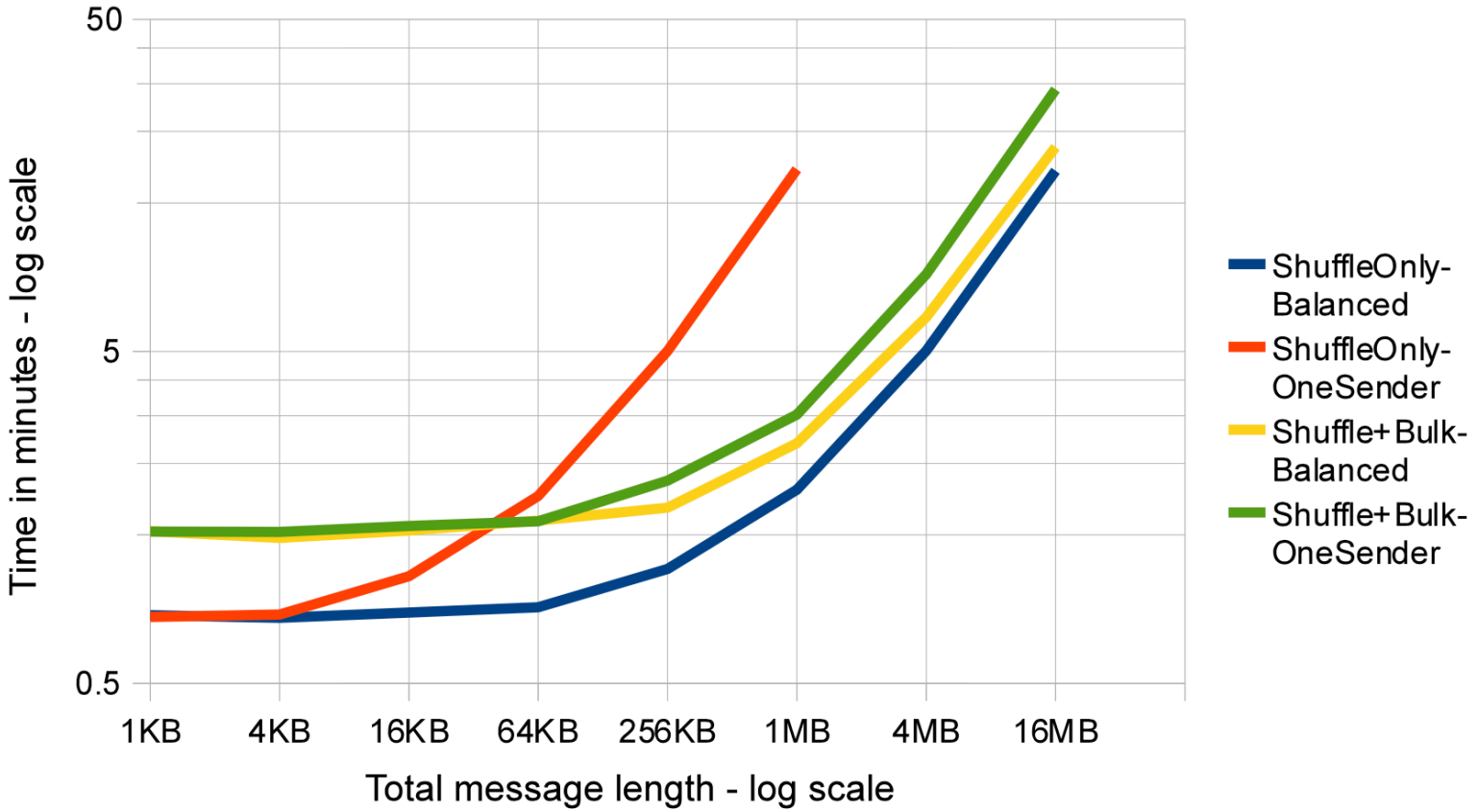
- **End-to-End reliability**
  - Solving the problem of closing the protocol
  - Message acknowledgement by other members
  - Acknowledgement using keys or signatures (public or pseudonymous keys, or group or ring signatures)

Prototype implementation

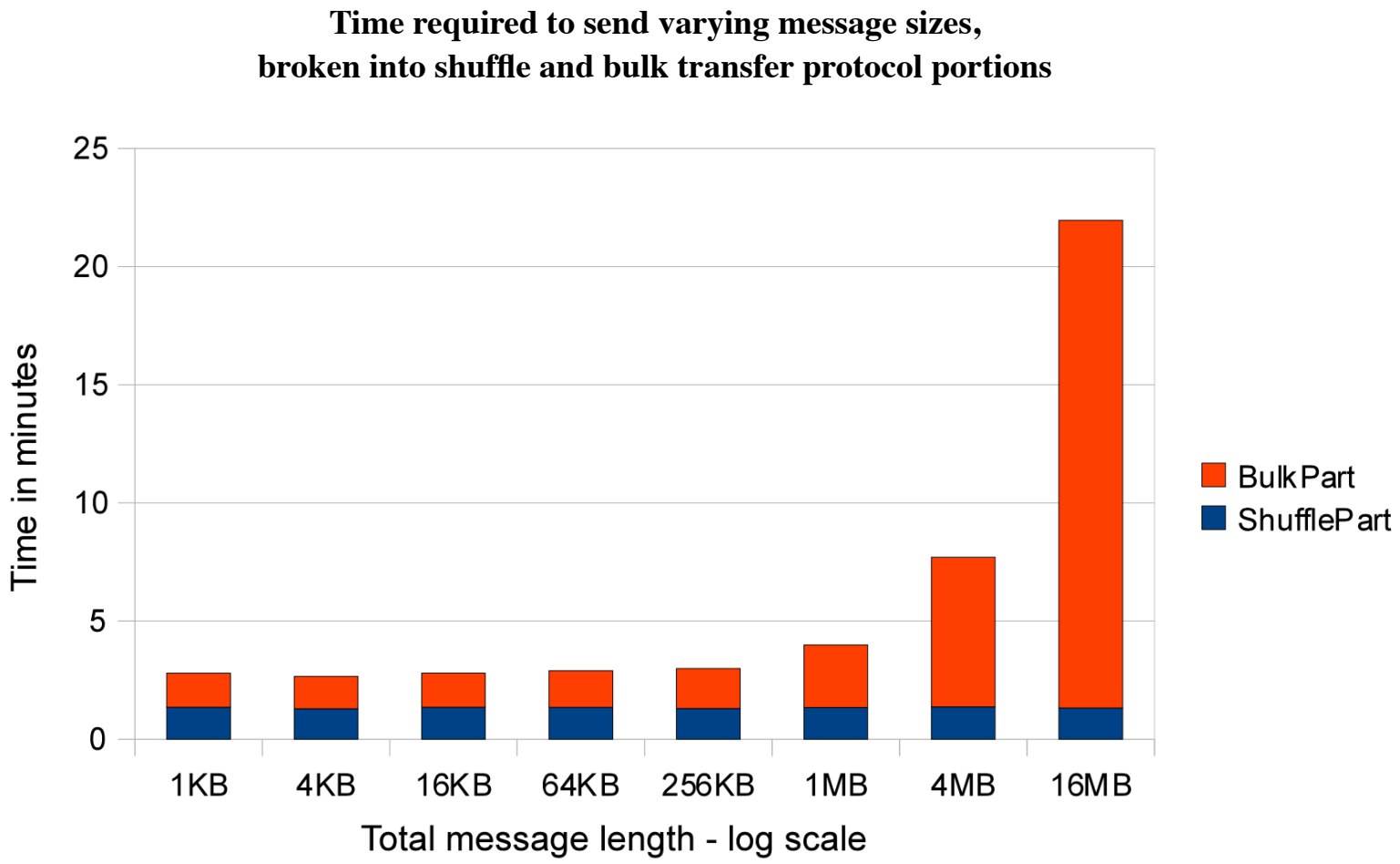
- Python
- 1014-bit RSA-OAEP with AES-256 -> **public key encryption and signing**
- AES-256 in counter mode -> **PRNG**
- Hash algoritm -> **SHA-1**
- Emulab testbed
- Star topology
- TCP protocol

Performance Evaluation

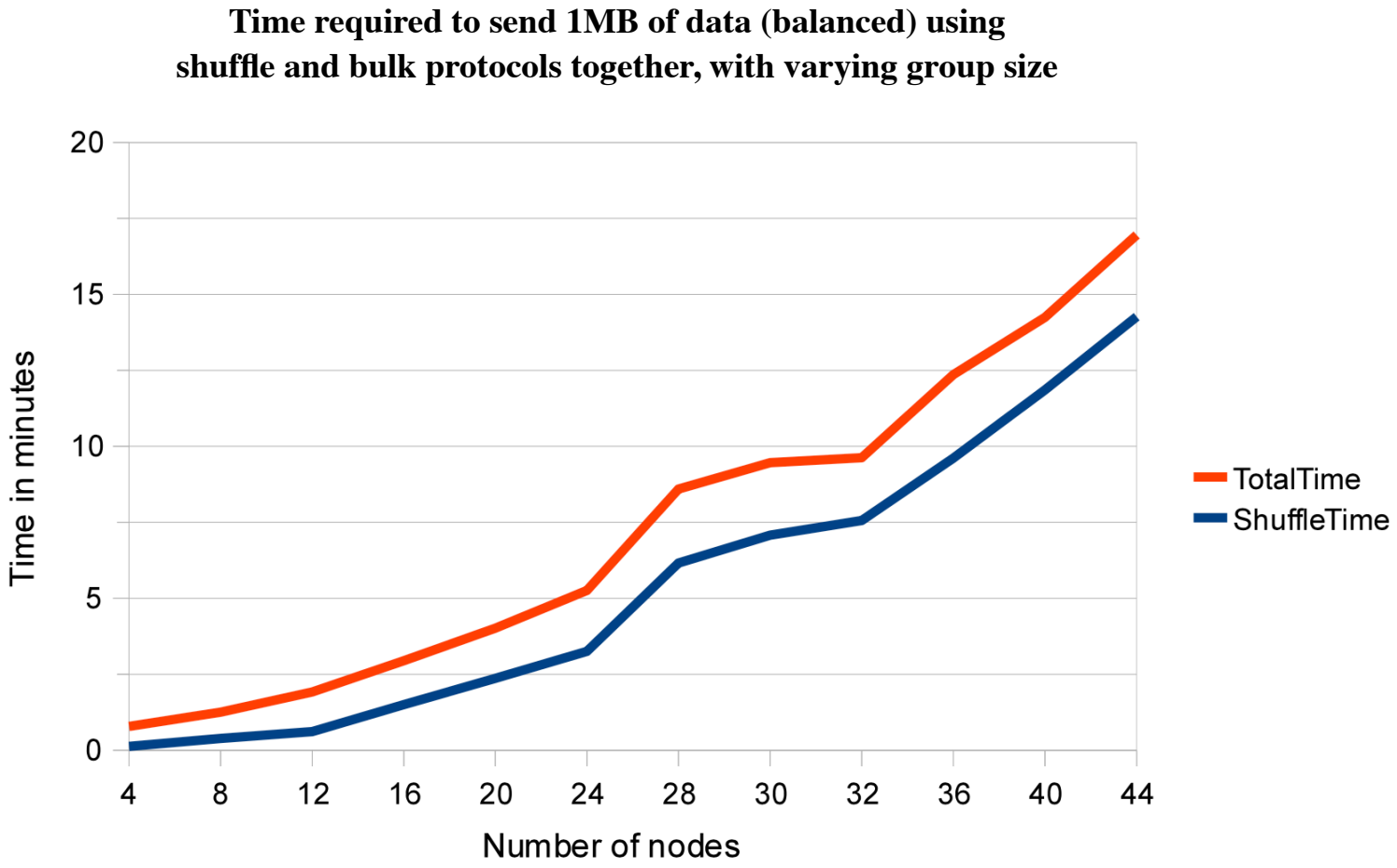
Time required for anonymous broadcast of balanced and unbalanced message loads among 16 nodes, via shuffle alone or full Dissent protocol



Performance Evaluation



Performance Evaluation



DCN-based Anonymous Communication Systems

UNIVERSITÀ DEGLI STUDI DI PADOVA

ACS	Comput. cost	Transmission overhead	Latency	Disadvantages	Main Feautures	Desired Applications	Attacks
Basic DCN	Medium	High	High	• Disruptions • Only supports transmission of one-bit messages per round	Information theoretically secure		
Dissent	Medium	High	High	• Linear increase of overhead with anonymity set size • Not intended for large scale • Per round star-up delay due to serialised shuffle protocol	• Client-server architecture • Scheduling via secret shuffling • Support variable-size message transmissions • Minimum participating users threshold (min anonymity set) • Leaves out slow users	• Latency tolerant messaging • File sharing	Intersection Attacks

DIPARTIMENTO MATEMATICA

DCN-based Anonymous Communication Systems

UNIVERSITÀ DEGLI STUDI DI PADOVA

ACS	Comput. cost	Transmission overhead	Latency	Disadvantages	Main Feautures	Desired Applications	Attacks
Riposte	Medium	High	High	• Trust model • Long epochs to ensure enough user participations • Uploading limit for message sizes • Identity of active users in each epoch is known	• Secret sharing of write request • Bandwith-efficiency	Anonymity message broadcasting	Intersection Attacks
Shared-Dining	High	High	High	• Supports group with small number of participants • Fixed-length messages	• Combination of secret sharing with classical DCN • Incentivise nodes to participate in the protocol • Prevents privacy breach by considering a threshold for participating users in a round	Applications with high privacy requirements, e.g. financial systems	N/A

DIPARTIMENTO MATEMATICA

References

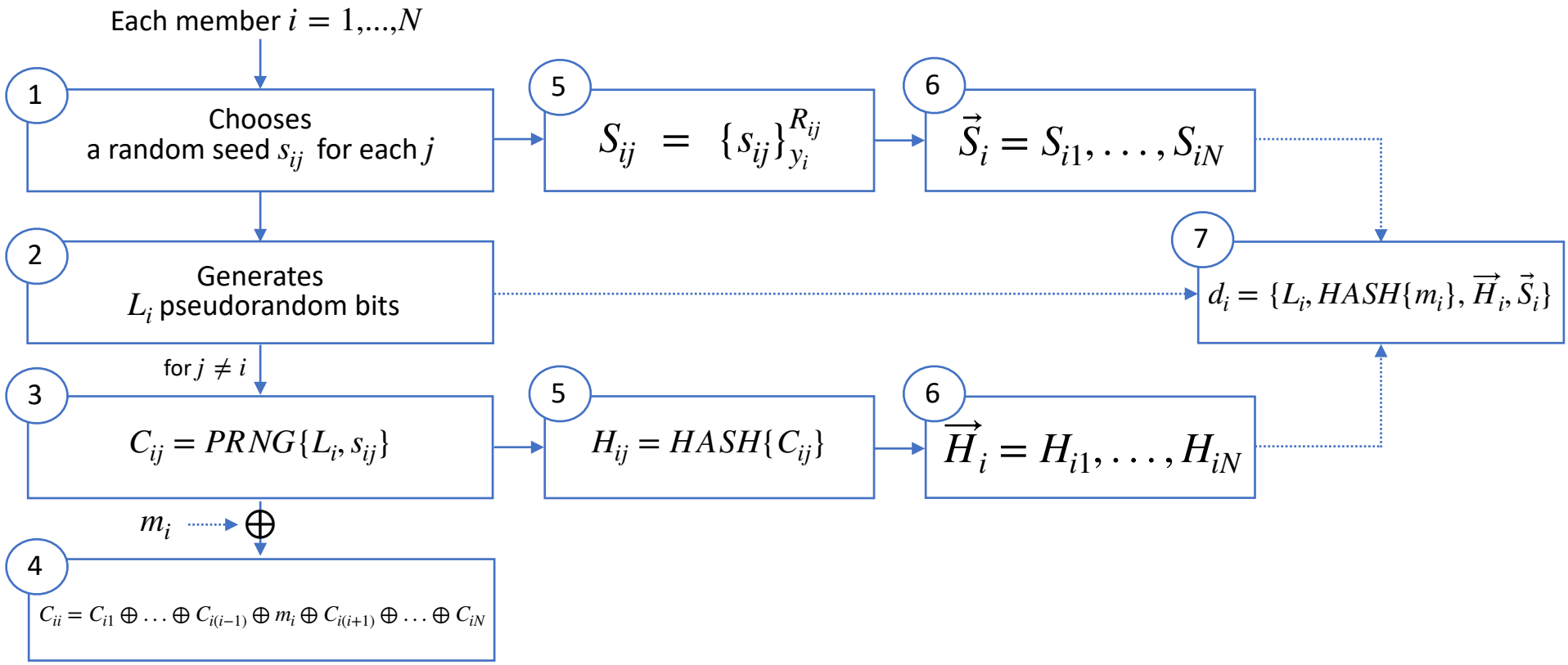
- Shirali, Mohsen & Tefke, Tobias & Staudemeyer, Ralf & Poehls, Henrich. (2022). A Survey on Anonymous Communication Systems with a Focus on Dining Cryptographers Networks. 10.48550/arXiv.2212.08275.

APPENDIX

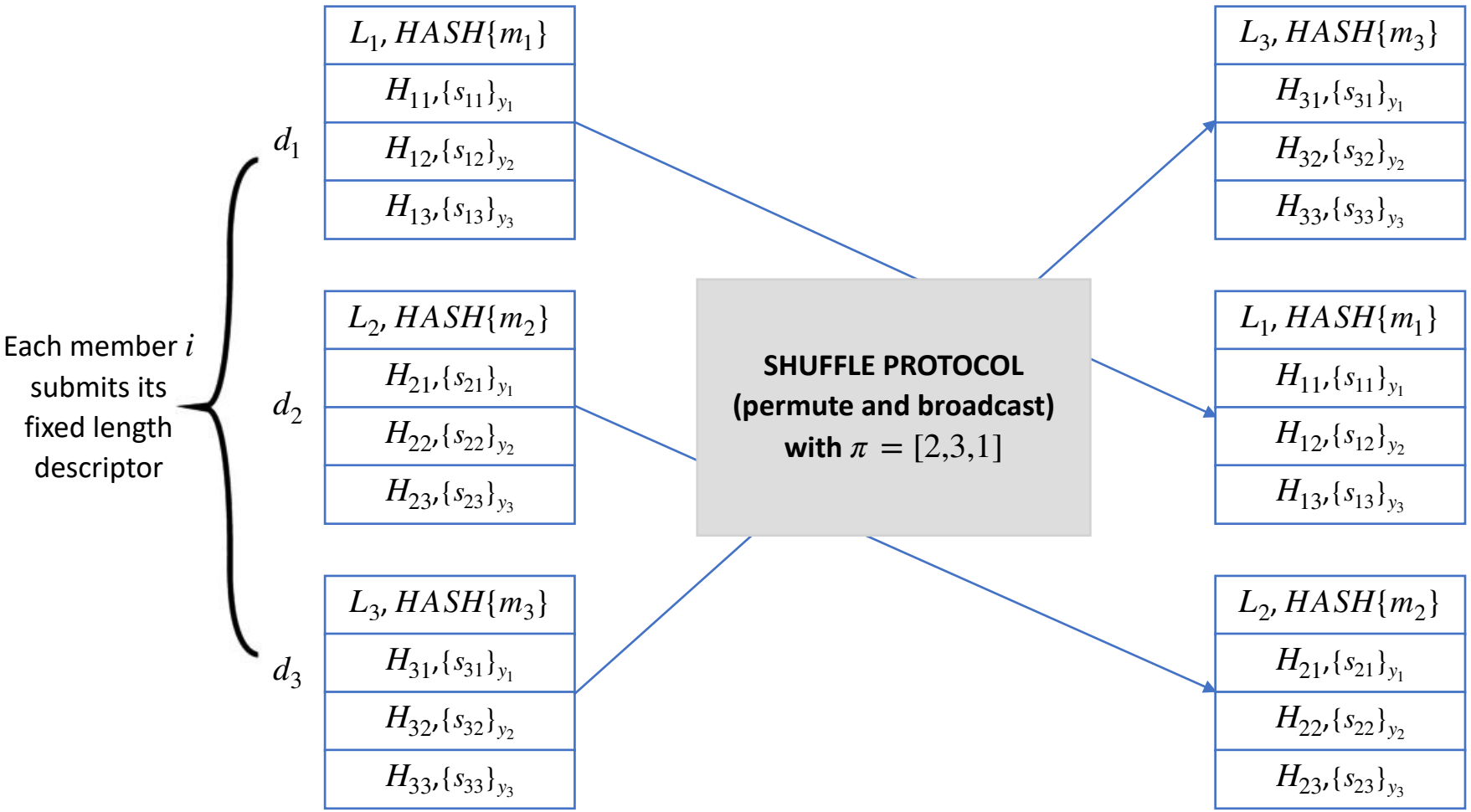
Protocol Description

- Phase 1. Message Descriptor Generation
- Phase 2. Message Descriptor Shuffle
- Phase 3. Data Transmission
- Phase 4. Message Recovery
- Phase 5. Blame

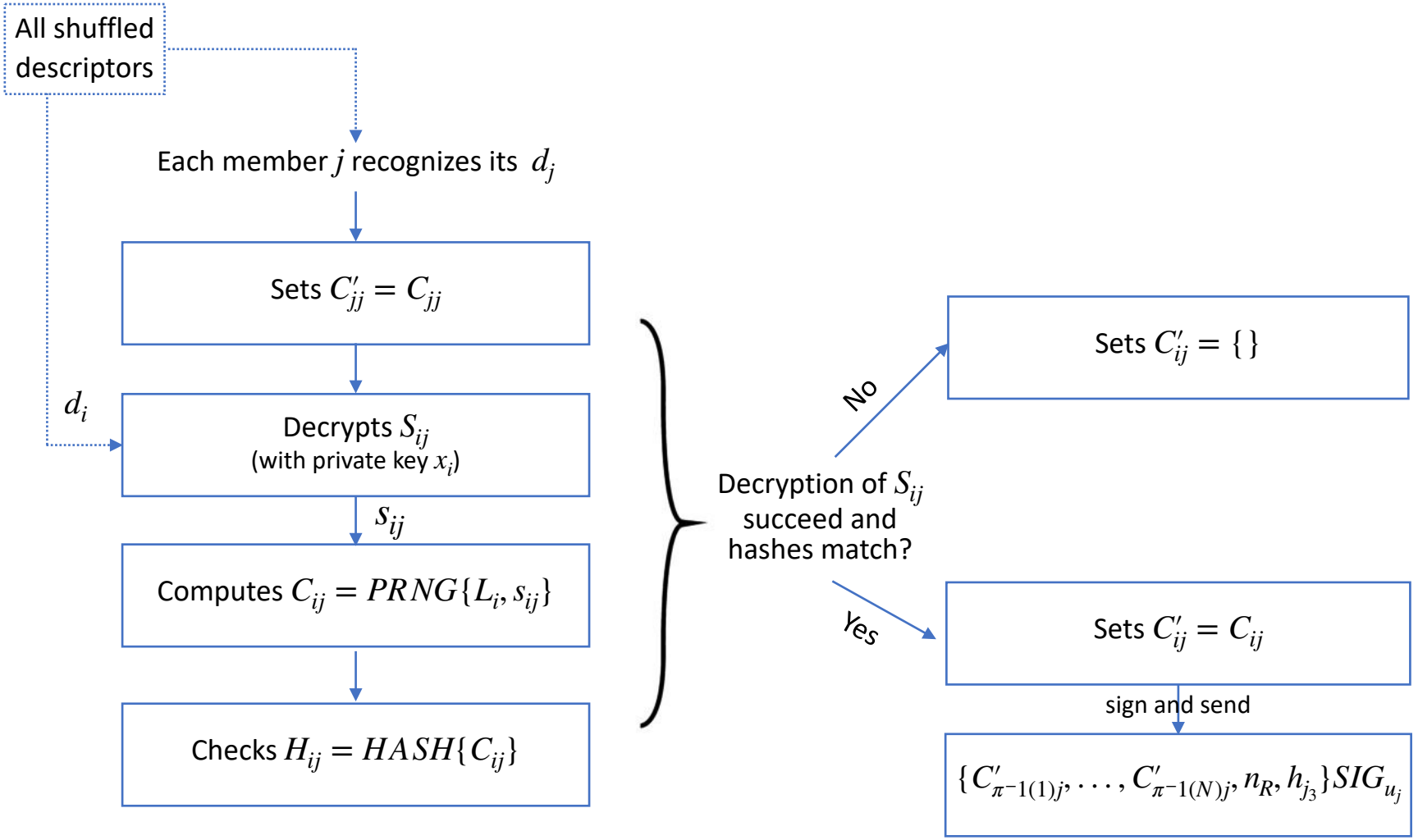
Phase 1. Message Descriptor Generation



Phase 2. Message Descriptor Shuffle

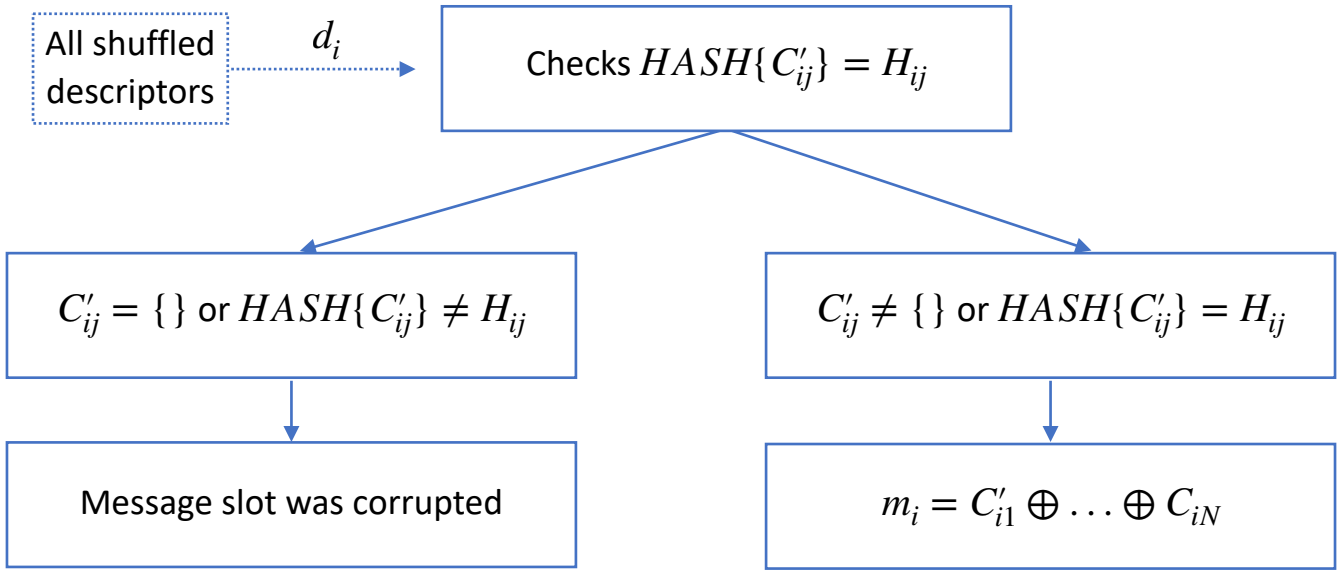


Phase 3. Data Transmission



Phase 4. Message Recovery

Receiver



Phase 5. Blame

- All members run the shuffle protocol again
- Each member i with corrupted message anonymously broadcasts:

$$A_i = \{j, S_{ij}, s_{ij}, R_{ij}\}$$

- Each member k verifies the revealed seed by:

$$S_{ij} = \{s_{ij}\}_{y_i}^{R_{ij}}$$

$$H_{ij} = HASH\{PRNG\{L_i, s_{ij}\}\}$$