T2 Privacy in Communications: Mix and Mixnets

104369: Privacitat de Dades i Seguretat

G. Navarro-Arribas
UAB -- DEIC
v.20261014

©G. Navarro-Arribas – (2026)

Single Mix

Mixnet Architectures

Anonymity Properties and Metrics

Attacks

Defenses

Modern High-Latency Mixnets

©G. Navarro-Arribas – (2026)

Single Mix

Mixnet Architectures

Anonymity Properties and Metrics

Attacks

Defenses

Modern High-Latency Mixnets

©G. Navarro-Arribas – (2026)

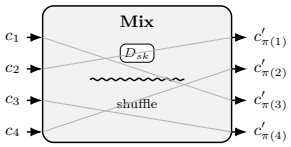
Chaum's mix (1981) [1]

Goal: unlink input messages from output messages.

- A Mix **decrypts** one layer, **reorders** (shuffles) a batch, and forwards.
- Single-hop encryption: sender A sends to mix M, then to B:

$$E_{PK_M}(B,\ E_{PK_B}(m))$$

- The Mix sees neither m nor the sender's identity.



Batching and permutation

- **Batch size** n: number of ciphertexts processed together.
- **Permutation** π chosen by the Mix.
- **Delays**: fixed window or randomized (discussed next).

E.g.

- Given $n = 4$ messages entering a single Mix, how many possible output orderings?
- If an attacker learns one input-output link, what's the anonymity set size for the remaining messages?

Batching and permutation

- **Batch size** n: number of ciphertexts processed together.
- **Permutation** π chosen by the Mix.
- **Delays**: fixed window or randomized (discussed next).

E.g.

- Given $n = 4$ messages entering a single Mix, how many possible output orderings? $4! = 24$
- If an attacker learns one input-output link, what's the anonymity set size for the remaining messages?

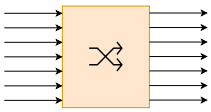
Batching and permutation

- **Batch size** n: number of ciphertexts processed together.
- **Permutation** π chosen by the Mix.
- **Delays**: fixed window or randomized (discussed next).

E.g.

- Given $n = 4$ messages entering a single Mix, how many possible output orderings? $4! = 24$
- If an attacker learns one input-output link, what's the anonymity set size for the remaining messages? ***3*** *(the 3 remaining messages could each be the source of any of the 3 remaining outputs)*

Threshold Mix

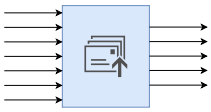


- Mix keeps messages; when it reaches T messages $\rightarrow$ all are forwarded at the same time (output queue randomly ordered).
- T is the **threshold**.
- All messages have the same size and are encrypted.

Threshold Mix properties

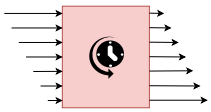
- Encryption avoids correlation at bit level.
- Correlation attack:
  - Passive and global attacker
  - Probability of success: $1/T$.
  - Anonymity set size: T.
  - Anonymity T/N (for N users in the system)

Pool Mix



- When T messages are received, the Mix outputs P ($P < T$);
  - the remaining $T - P$ messages stay in the pool for the next round.
- A message can be in the Mix for a long time (generally higher average delay than a Threshold Mix), but
- higher privacy!

Stop-and-go Mix



- Each message has a random delay.
- Average message delay is not affected by the number of messages.
- The delay is commonly modeled by an exponential distribution, $\sim \text{Exp}(\lambda)$.

Single Mix

Mixnet Architectures

Anonymity Properties and Metrics

Attacks

Defenses

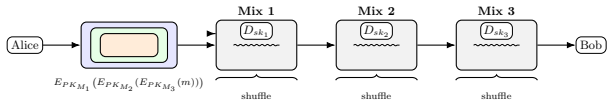
Modern High-Latency Mixnets

Layered encryption (onion)

To traverse mixes M_1, M_2, M_3 toward D:

$$C = E_{PK_{M_1}}\Big(M_2,\ E_{PK_{M_2}}\big(M_3,\ E_{PK_{M_3}}(D,\ E_{PK_D}(m))\big)\Big)$$

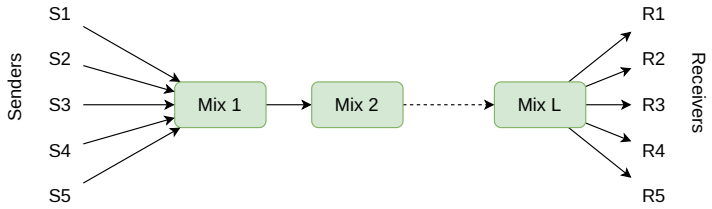
- M_1 decrypts, learns only next hop M_2, forwards inner ciphertext.
- No mix learns the full path or the plaintext.
- Cryptographic unlinkability if *at least one mix per path is honest* (cascade; in free-route, this must hold for the chosen path).



Architectures: cascade

Cascade

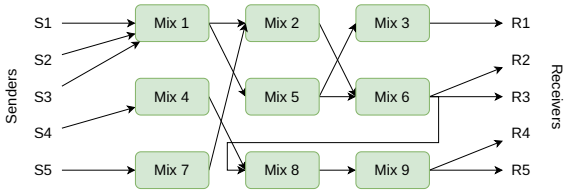
- All messages traverse the same fixed sequence of mixes.
- Easier to analyze; synchronized batches.



Architectures: free-route

Free-route

- Sender picks a path across a Mixnet graph.
- More flexible; heterogeneous paths and timings.



Single Mix

Mixnet Architectures

Anonymity Properties and Metrics

Attacks

Defenses

Modern High-Latency Mixnets

Unlinkability and anonymity sets

- **Sender (receiver) anonymity:** hide which sender (receiver) is linked to an observed output (input).
- **Unlinkability:** adversary cannot decide whether two items are related.

Anonymity set size (naïve): number of plausible candidates.

- If all candidates equally likely: size $= n$.
- If not uniform: we can use entropy.

Entropy-based anonymity

Let p_i be the posterior probability that user i is the actual sender (or receiver). Then

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i \quad \text{bits.}$$

- Maximum when $p_i = 1/n$ (uniform), then $H = \log_2 n$.
- Captures non-uniform beliefs from side-channels (timing, volume).

Effect of batching and delays on entropy

- Larger batches and randomized delays ⇒ flatter posterior p_i.
- Structured traffic (bursts, periodic sends) ⇒ skewed p_i.

E.g.

A threshold mix releases 4 messages from 4 senders, suppose posterior probabilities $p = \{0.4, 0.3, 0.2, 0.1\}$. Compute H and compare to uniform over 4.

Effect of batching and delays on entropy

- Larger batches and randomized delays ⇒ flatter posterior p_i.
- Structured traffic (bursts, periodic sends) ⇒ skewed p_i.

E.g.

A threshold mix releases 4 messages from 4 senders, suppose posterior probabilities $p = \{0.4, 0.3, 0.2, 0.1\}$. Compute H and compare to uniform over 4.
$H = -(0.4 \log_2 0.4 + 0.3 \log_2 0.3 + 0.2 \log_2 0.2 + 0.1 \log_2 0.1) \approx 1.85$ bits

Uniform over 4: $H_{\max} = \log_2 4 = 2$ bits.

Single Mix

Mixnet Architectures

Anonymity Properties and Metrics

Attacks

Defenses

Modern High-Latency Mixnets

Timing correlation & volume analysis

Attacker: observes input and output times/volumes.

- Correlate bursts and idle periods.
- Narrow candidate sets across multiple rounds.

Mitigations: randomized delays; cover traffic to smooth rates; batching policies.

Example: $n-1$ attacks

- Attacker manages to fill a Mix with its own messages except one (the victim).
  - Threshold Mix: $T-1$ messages belong to the attacker.
- Defense: use of dummy traffic (the Mix itself can generate traffic to itself)

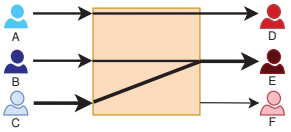
Statistical disclosure attacks

- Attacker can observe communication patterns over (a long) time.
- Obtain relative frequency of communication between pairs by intersecting different destination anonymity sets for a given victim.

E.g.

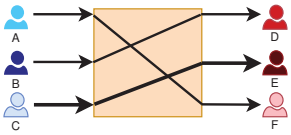
Attacker observes 100 cycles, in 25 of them, sender B and receiver E are present $\Rightarrow$ 0.25 is the relative frequency between B and E.

Example



0.4	0.25	0.35
0.45	**0.5**	0.05
0.1	**0.8**	0.1

Source: [5]



0.4	0.25	**0.35**
0.45	0.5	0.05
0.1	**0.8**	0.1

Source: [5]

Single Mix

Mixnet Architectures

Anonymity Properties and Metrics

Attacks

Defenses

Modern High-Latency Mixnets

Dummy traffic and message size

GOAL: Make traffic analysis harder [4].

- All messages have a **constant size** determined by the Mix. Padding is added to fit the required size.
- Clients and/or Mixes add **dummy traffic**: full messages without application purpose.
  - Example: make the system to have always a constant flow of messages.

Operational considerations

- Key management for mixes; rotating keys to limit linkability.
- Failure/malicious behavior: need at least one honest mix per path.
- Auditable shuffles (verifiable mix-nets) for bulletin boards/e-voting.

Single Mix

Mixnet Architectures

Anonymity Properties and Metrics

Attacks

Defenses

Modern High-Latency Mixnets

Loopix: stop-and-go mixing + cover traffic [6]

Idea:

1. each mix node is a **stop-and-go mix** ($\tau \sim \mathrm{Exp}(\lambda)$ per message)
2. *constant-rate* cover traffic end-to-end.

That is,

- Delays are exponential (memoryless) $\rightarrow$ the output of each node is a Poisson process.
  - This stop-an-go mix is also named a “Poisson mixing”.
- Clients inject two kinds of cover traffic:
  - *loop* messages (routed back to sender, detect disruption);
  - *drop* messages (sent to random recipients, pad volume).
- Latency/anonymity trade-off tuned by adjusting λ.

Nym: a production mixnet [3]

Nym is a production-grade privacy infrastructure built on Loopix [6].

Three-layer architecture:

1. **Mixnet layer** – stop-and-go mixes using Sphinx packet format [2]; same Loopix loop/drop cover traffic.
2. **Credential layer** – anonymous authentication via the Coconut credential scheme [7].
3. **Application layer** – clients (NymVPN, wallets, messaging) access the network through a gateway.

Nym credentials (Coconut)

Problem: how do clients authenticate to use the mixnet *without* revealing their identity?

Coconut credentials [7]:

- **Threshold blind issuance:** a set of validators jointly issue a credential; no single validator sees the full secret.
- **Selective disclosure:** the user can prove possession of a valid credential without revealing which credential (or to whom it was issued).
- **Unlinkability:** multiple uses of the same underlying credential are unlinkable to each other.

Net result: a client can prove "I have a valid subscription" to a gateway without leaking its identity or linking its sessions.

Nym incentive layer

Problem: why would anyone run a mix node? (volunteer-only networks are fragile)

Solution: token-based incentives.

- Mix operators **stake NYM tokens** to register as mix nodes.
- Nodes are rewarded for forwarding traffic, proportionally to their measured performance.
- **Slashing:** operators lose stake for misbehavior or prolonged unavailability.

Effect on anonymity:

- Larger, more stable node set $\Rightarrow$ larger anonymity sets.
- Incentives must be designed carefully not to leak traffic-volume information (reward signal can be an oracle for the attacker).

References i

[1] David L Chaum.
Untraceable electronic mail, return addresses, and digital pseudonyms.
Communications of the ACM, 24(2):5, 1981.

[2] G. Danezis and I. Goldberg.
Sphinx: A Compact and Provably Secure Mix Format.
In *2009 30th IEEE Symposium on Security and Privacy*, pages 269–282, May 2009.

[3] Claudia Diaz, Harry Halpin, and Aggelos Kiayias.
The Nym network: The next generation of privacy infrastructure.
Technical report, Nym Technologies SA, 2021.

[4] Claudia Diaz and Bart Preneel.
Taxonomy of Mixes and Dummy Traffic.
In Yves Deswarte, Frédéric Cuppens, Sushil Jajodia, and Lingyu Wang, editors, *Information Security Management, Education and Privacy*, IFIP International Federation for Information Processing, pages 217–232, Boston, MA, 2004. Springer US.

[5] Marc Juárez Miró.
Protocols de comunicació que preserven la privadesa.
Fundació Universitat Oberta de Catalunya, 2020.

[6] Ania M. Piotrowska, Jamie Hayes, Tariq Elahi, Sebastian Meiser, and George Danezis.
The Loopix Anonymity System.
In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1199–1216, 2017.

[7] Alberto Sonnino, Mustafa Al-Bassam, Shehar Bano, Sarah Meiklejohn, and George Danezis.
Coconut: Threshold issuance selective disclosure credentials with applications to distributed ledgers.
In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2019.

©G. Navarro-Arribas – (2026)

References ii

[8] Vicenç Torra.
Guide to Data Privacy: Models, Technologies, Solutions.
Undergraduate Topics in Computer Science. Springer International Publishing, Cham, 2022.

©G. Navarro-Arribas – (2026)