Anonymous Communication on Expander Networks

Megumi Ando [*] Hannah Lynn [†] Anna Lysyanskaya [‡] Eli Upfal [§]

Abstract

Who is talking to whom? Consider a group of users who wish to communicate anonymously via a network of intermediate relays. We study anonymous communication under two standard strong adversarial models. A *passive adversary* observes all network traffic and additionally views the internal states of a constant fraction of corrupted relays, while an *active adversary* may also control the behavior of these corrupted relays. The goal of an anonymous communication protocol is to ensure that the adversary cannot distinguish who is communicating with whom. One of the most practical and widely adopted approaches is *onion routing*, where messages are first wrapped in layers of encryption and anonymity emerges through repeated "shuffling" of onions at honest relays that peel a layer and randomly permute outgoing onions. In general, this approach may not achieve anonymity. The challenge is to rigorously quantify conditions for efficiently achieving anonymity, where efficiency is measured as a function of the protocol's *security parameter* λ, which we assume, w.l.o.g., is at least linear in the network size.

A well-known result (ICALP'18) shows that if each hop in a routing path is chosen uniformly at random from all relays, then onion routing achieves anonymity against a passive adversary whenever both the number of rounds and the server load grow faster than $\log \lambda$. In this setting, anonymity arises from the fact that every onion is repeatedly shuffled with a uniformly random subset of other onions. However, this assumption requires a fully connected network.

We generalize this result to sparse networks. We show that when routing paths are selected by performing independent random walks on a sparse, constant-degree expander graph, onion routing still achieves anonymity with the same asymptotic efficiency parameters as in the complete-network setting. In particular, this matches the optimal round-complexity bound known for complete networks, despite the fact that onions only shuffle within their local neighborhoods at each round, and an adversary may extract information from observing transitions between neighboring nodes.

We further extend our results to active adversaries. In the sparse-expander setting, we construct, under different conditions, (1) a *differentially private* protocol that achieves $(\epsilon, \mathsf{negl}(\lambda))$-DP, and (2) an anonymous protocol; both run efficiently in polylogarithmic rounds and incur polylogarithmic server load.

[*]Computer Science Department, Tufts University, mando@cs.tufts.edu
[†]Computer Science Department, Tufts University, hannah@cs.tufts.edu
[‡]Computer Science Department, Brown University, anna@cs.brown.edu
[§]Computer Science Department, Brown University, eli_upfal@brown.edu

1 Introduction

Privacy is a fundamental human right [Uni48], with a long history in legal thought [WB90] and modern protections in both Europe and the United States [Eur16, Cal18]. Anonymous communication is therefore essential for preserving privacy in a digital world connected through public networks such as the Internet.

In this paper, we study the problem of enabling peer-to-peer anonymous connections in the presence of a powerful adversary that can observe all network traffic and additionally monitor corrupted nodes. This model encompasses resourceful adversaries such as nation-states, autonomous systems (ASes), or Internet service providers (ISPs) and is a standard notion of adversary in cryptography fields such as multiparty computation.

As an example, suppose that Alice wishes to send a message to Bob such that an adversary observing at all network links and some of the nodes cannot determine that Alice is the sender. Simply encrypting the message before transmission is insufficient for anonymity, since the same sequence of bits originating from Alice's computer ultimately arrives at Bob's. Achieving anonymity, therefore, requires that the message changes its appearance before reaching Bob.

Onion encryption as an approach. A well-known approach to partially solving this problem is onion encryption [Cha81, GRS96]. To send the message m, Alice first selects a random route through the network consisting of intermediary relays R_1, R_2, and R_3. She then constructs a layered encryption structure called an *onion*.[1] The first intermediary, R_1, removes only the outermost encryption layer, revealing an inner onion O_2 together with the address of the next relay R_2. The second intermediary, R_2, removes another layer to obtain O_3 and the address of the next relay R_3. This process continues until Bob receives the final onion O_B and recovers the original message m. This approach prevents the attack described above: an adversary cannot directly link Alice to Bob based solely on packet appearance, since the onion is transformed at every hop along the route. The adversary, however, may still learn useful information from observing the numbers of onions entering and leaving a node in each step.

Onion encryption is also attractive for several practical reasons. It is conceptually *simple*: easy to implement, and requires little coordination among senders since each sender can choose routes for its own onions. It is also *scalable* as relay workloads grow modestly with network size, and supports *fault tolerance* since the loss of some packets does not necessarily affect the delivery of others.

These properties have contributed to the success of The Onion Router (Tor) [DMS04], which millions use daily[2] as well as Freenet [CSWH01] and Crowds [RR98]. However, while these solutions are practical, it is well known that the strategies underlying these networks often compromise anonymity as they are susceptible to traffic analysis attacks, particularly *timing analysis*, in which an adversary correlates packet timing and volume patterns observed at different points in the network to infer communication relationships [MD05, SEV+15, WSJ+18].

Motivation for shuffling. A major reason low-latency systems such as Tor fail to provide strong anonymity guarantees is they do not intentionally batch and shuffle onions before forwarding them. For example, if Alice's onion destined for Bob is the only onion traversing the network, then an adversary observing global traffic can simply trace the network flow from Alice to Bob.

To prevent such attacks, multiple onions must be batch-processed at honest relays. We say that onions are *batch-processed* at a relay if the relay first receives multiple onions, peels one encryption layer from each of them, and then forwards the resulting onions in random order. This local shuffling step prevents the adversary from determining which outgoing onion corresponds to which incoming onion. By repeat-

[1] Strictly speaking, an onion need not be constructed by layering encryption; this is just an example; see prior work [AL21, ACLM22].

[2] Data retrieved from the Tor Project.

edly performing such local shuffles throughout the network, one can obtain an onion routing protocol that achieves significantly stronger anonymity guarantees while retaining the original practical advantages of onion routing: simplicity, scalability, and fault tolerance.

1.1 Onion Shuffling Protocols

This observation motivates the central primitive underlying onion shuffling protocols. In such a protocol, N onions are routed through a communication graph $G = (\mathcal{R}, \mathcal{E})$ consisting of n relays, over L rounds. In each round, every relay $R \in \mathcal{R}$ batch-processes the onions it receives by peeling one encryption layer from each onion and forwarding the resulting onions in random order before the next round. If the protocol achieves *onion indistinguishability*, after the L rounds, the correspondence between input onions (at round 1) and output onions (at round L) becomes hidden from the adversary; see Section 2.4 for the formal definition. More precisely, the adversary should not be able to link onions observed in the final round to their corresponding onions in the first round with probability non-negligibly better than random guessing. (A function is *negligible*, denoted $\mathsf{negl}(\lambda)$, if it decays faster than the inverse of every polynomial in the security parameter λ.)

Prior result. A foundational result in this direction was given by Ando, Lysyanskaya, and Upfal [ALU18], who analyzed the simple protocol Π_p (here, p stands for passive adversary) over a complete relay network topology. In Π_p, each relay for an onion's path is chosen independently and uniformly at random from all n relays. They showed that Π_p is an onion shuffling protocol that achieves anonymity from the *passive* adversary (adversary that observes all network traffic and additionally monitors a constant fraction $\chi < 1$ of the relays), provided that both the number of rounds, L, and the server load (the expected number of onions processed by each relay per round), $\frac{N}{n}$, are polylogarithmic in the security parameter.

This result is significant because it provides a foundational protocol whose analysis abstracts anonymity into a clean combinatorial mechanism. As a result, it is widely used as a building block in the analysis of more complex protocols, including those designed for stronger adversary models (e.g., in the *active* adversary setting, in which corrupted relays may deviate arbitrarily from the protocol). Consequently, extending this result to new settings yields a *portability theorem* for a broad class of onion routing protocols.

1.2 The Sparse Network Setting

The analysis of [ALU18] critically relies on a complete network topology, requiring $\Omega\left(n^2\right)$ edges for n relays. In practice, however, many decentralized systems operate over sparse overlay networks in which nodes maintain connections to only $O(1)$ neighbors and global routing knowledge is often unavailable. Anonymous communication protocols are highly relevant in such environments, including peer-to-peer systems and blockchain infrastructures, yet it has remained unclear whether onion routing can achieve provable anonymity on realistic sparse topologies with locally constrained routing paths,[3] matching the near-logarithmic round complexity and server load achieved by the best protocols in the fully connected setting.

This motivates our study of sparse *expander graphs*, a widely used class of communication networks.[4] Expanders arise in high-performance datacenter fabrics such as *Xpander* [VSDS16] and *Jellyfish* [SHPG12], as well as in decentralized overlay networks designed to maintain connectivity under churn [APR+15].

The strong spectral expansion properties of these graphs make them both practically relevant and analytically tractable for studying anonymity in sparse communication networks. In particular, expander graphs provide a natural framework for investigating whether the anonymity and efficiency guarantees known for

[3] Previous works [CV14, ALU21] established anonymity guarantees for onion routing on highly structured multi-layered "butterfly" graphs. Although butterfly networks are useful in parallel computing, they do not naturally capture the loosely connected structure of WANs and other real-world distributed systems.

[4] For a comprehensive survey of expander constructions and applications, see [HLW06].

onion routing on complete graphs can be extended to more realistic sparse network topologies.

1.3 Our Contributions

Our main result shows that the protocol Π_p, modified so that onions perform random walks on a sparse expander graph rather than on a complete graph, still achieves onion indistinguishability against any passive adversary corrupting up to a constant fraction $\chi < 1$ of the relays, provided that both the number of rounds L and the server load $\frac{N}{n}$ grow strictly faster than logarithmic in the security parameter.

At a high level, this result is surprising because routing on sparse graphs permits only highly localized mixing in each round. Unlike the complete network setting, where an onion may move directly to any relay in a single step, an onion in a sparse graph can move only to one of a relay's constant-degree neighbors. Therefore, one might presume substantially more rounds to be necessary before onions become sufficiently mixed to guarantee anonymity. Nevertheless, we show that sparse expander topologies achieve the same asymptotic bounds on round complexity and server load as the complete graph setting.

Our proof proceeds through two main technical lemmas, which may be of independent interest.

Expectation bound (Lemma 1). We first show that, for any function $g(\lambda)$ satisfying $g(\lambda) \to \infty$, after $L = g(\lambda) \log \lambda$ rounds, the adversary's posterior distribution over the origin of any fixed round-L onion is negligibly close to the uniform distribution over the N initial onions. Informally, in expectation, every output onion appears almost equally likely to have originated from any input onion.

Concentration bound (Lemma 2). We next prove that, for any functions $g_1(\lambda), g_2(\lambda)$ satisfying $g_1(\lambda)$, $g_2(\lambda) \to \infty$, after $L = g_1(\lambda) \log \lambda$ rounds and with server load $\frac{N}{n} = g_2(\lambda) \log \lambda$, with overwhelming probability, the adversary's posterior distribution is tightly concentrated around its expectation. Thus, not only is the expected posterior distribution close to uniform, but substantial deviations from uniformity occur only with negligible probability.

Combining these lemmas yields our main theorem.

Main theorem (Theorem 1). Let G be an r-regular β-expander graph on n relays, where $r > 2$ and $\beta < 1$ are constants independent of the security parameter λ. Suppose N onions are routed on G according to $\Pi_p(G)$. If, for some functions $g_1(\lambda), g_2(\lambda)$ satisfying $g_1(\lambda), g_2(\lambda) \to \infty$, the number of rounds satisfies $L = g_1(\lambda) \log \lambda$, and the server load satisfies $\frac{N}{n} = g_2(\lambda) \log \lambda$, then $\Pi_p(G)$ achieves *onion indistinguishability*[5] on G against any passive adversary that corrupts up to a constant fraction $\chi < 1$ of the relays. Informally, this means that, after L rounds, the adversary cannot link a final onion to its initial form with better than negligible advantage over a random guess. *Combined with the lower bound of [Chr20], this establishes asymptotically optimal round complexity.*

Finally, we demonstrate the modularity of our result by showing how it can be plugged directly into existing onion routing constructions to extend their guarantees from complete networks to sparse expander networks with constant degree $r > 2$ and constant expansion parameter $\beta < 1$. Specifically, we prove the following:

Differential anonymity from active adversaries (Theorem 2). There exists an onion routing protocol in the sparse expander setting that achieves differential anonymity against any *active* adversary *controlling* a constant fraction $\chi < 1$ of the relays, for any server load and round complexity growing strictly faster than logarithmic in the security parameter λ. Here, *differential anonymity* means that for any two *neighboring* communication patterns differing in a single sender-receiver message pair, the adversary's resulting views are $(\epsilon, \mathsf{negl}(\lambda))$-indistinguishable[6] where $\mathsf{negl}(\lambda)$ denotes any negligible function in the security parameter.

[5] See Section 2.4 for the formal definition.

[6] See section 6.2 for the formal definition.

Anonymity from active adversaries (Theorem 3). We further obtain an onion routing protocol in the sparse-expander setting that achieves *anonymity* against active adversaries, leaking only the I/O-pattern, i.e., the number of messages sent and received by each participant. Concretely, for any two communication patterns with the same I/O-pattern leakage, the adversary cannot distinguish which one was executed except with negligible advantage.

Remark. While the *anonymity* guarantee of Theorem 3 is stronger in scope than *differential anonymity*, it explicitly permits the standard leakage of the number of messages sent and received by each participant.

1.4 Technical Challenges

Why the problem is nontrivial. We stress that our setting is fundamentally different from the classical mixing problem on expander graphs. Standard results imply that a random walk on an expander mixes rapidly: after any number of rounds growing asymptotically faster than $\log \lambda$, the location of an onion is negligibly close to uniformly distributed over the graph [HLW06]. However, anonymity requires a substantially stronger guarantee. The relevant quantity is not the unconditional distribution of an onion's location, but rather the adversary's *posterior* distribution conditioned on its observations (all traffic on network links together with the internal states of corrupted relays). Establishing anonymity therefore requires proving that these observations do not allow the adversary to meaningfully correlate input onions with output onions.

A straightforward way to obtain anonymity is to emulate the complete graph protocol of [ALU18] on top of the expander network. Specifically, each virtual edge of the complete graph can be implemented by routing onions along a random path of length $\Theta(\log n)$ in the expander. This immediately yields anonymity, but incurs an additional multiplicative $O(\log n)$ overhead in round complexity. Consequently, the resulting protocol requires $O(\log^a \lambda)$ rounds for some $a > 2$, rather than matching the asymptotically optimal guarantee that any round complexity growing faster than $\log \lambda$ suffices. Our main result shows that this overhead is unnecessary: sparse expander networks can achieve the same asymptotic anonymity guarantees as the complete graph setting without simulating global connectivity.

Thus, the main challenge in our setting lies in simultaneously achieving all three: (1) anonymity (against adversaries that corrupt any constant fraction < 1 of the relays), (2) negligible distinguishing advantage, and (3) asymptotically optimal round complexity.

Why the complete graph proof does not extend. The proof that Π_p achieves onion indistinguishability on the complete graph network does not extend directly to the sparse expander setting, as the original argument relies crucially on complete connectivity.

The proof proceeds inductively by tracking the adversary's posterior distribution over the location of a fixed target input onion at every round. In each round, the gap between the largest and smallest posterior probabilities shrinks by a constant factor. Intuitively, this contraction occurs because every relay receives a mixture of onions (both with high and low probabilities of being the target onion in the previous round) and then randomly shuffles them together. Probability concentration bounds imply that this averaging effect occurs across all relays, causing the probability gap to decrease geometrically over time.

This argument fundamentally depends on the fact that in a complete graph every onion can move independently and uniformly to any relay in a single step. In contrast to this, in the sparse expander setting, onions may move to one of a constant r neighbors, so mixing occurs much more locally. As a result, relays no longer receive globally representative mixtures of onions at every round, and the averaging phenomenon underlying the complete graph proof breaks down. Consequently, new techniques are required to analyze anonymity in the sparse expander setting.

Why naïve approaches are insufficient. Several natural approaches achieve anonymity in restricted settings, but fail to scale or provide robust guarantees against stronger adversaries.

A simple solution is to route all onions through a single trusted relay that batch-processes and randomly

permutes every message. While this immediately provides anonymity against the network adversary, it does *not scale*: the relay becomes both a computational bottleneck and a single point of failure since its workload grows linearly with the number of users. Moreover, the approach depends critically on the relay being fully trusted.

A more distributed approach is to route onions through highly structured permutation networks, such as butterfly networks, which spread the shuffling workload across many parties. However, these constructions are typically *not simple nor fault-tolerant*: for instance, the constructions of [TGL+17, KCDF17] rely on heavy cryptographic primitives (verifiable shuffling) that also make the constructions *fragile*, meaning that no message is delivered if even a single message packet is dropped. These limitations motivate the search for onion routing protocols that avoid centralized bottlenecks and use only lightweight cryptographic primitives that preserve the fault tolerance of the onion routing approach.

1.5 Key Insights Towards Main Result

The proof has two main parts. First, we analyze the adversary's uncertainty, on average, over a random execution of the protocol. A passive adversary can perfectly track onions through corrupted relays, so corrupted stretches of a route act as "transparent pass-throughs" between honest relays, possibly spanning multiple rounds. By collapsing these corrupted stretches into effective honest-to-honest transitions, we obtain a time-layered view in which the expected diffusion of certainty about where an initial onion last shuffled among the set of honest relays follows the same transition probabilities as a random walk on the underlying expander (projected onto the set of honest relays). This gives the expectation bound: on average, every final onion is (negligibly) close to equally likely to have originated from any initial onion.

Second, we show that the adversary's actual uncertainty in a concrete execution is close to this expectation, except with negligible probability. From the adversary's point of view, each honest shuffle creates several possible explanations for how an incoming onion became an outgoing onion. A full explanation for why a particular final onion may have come from a particular initial onion is a sequence of such candidate peelings through the protocol. Each honest shuffle splits probability mass across the onions in its batch, while corrupted relays add no uncertainty because the adversary sees the true input-output correspondence. Since the server load grows asymptotically faster than $\log \lambda$, and each route encounters a number of honest shuffles that grows asymptotically faster than $\log \lambda$ with overwhelming probability, each individual explanation has very small weight. A variance bound over the sum of all candidate explanations shows that the adversary's actual uncertainty is tightly concentrated around its expectation.

Together, these two bounds show that the adversary's belief about the origin of any final onion diffuses evenly across all initial onions by the end of the protocol.

2 Preliminaries

2.1 Notation

Negligibility and related terms. With respect to the security parameter λ, we say that a function $f : \mathbb{N} \mapsto \mathbb{R}$ is *negligible* (denoted $f(\lambda) = \mathsf{negl}(\lambda)$) if for every polynomial $p : \mathbb{N} \mapsto \mathbb{R}$, there exists some $\lambda_0 \in \mathbb{N}$ such that for all $\lambda > \lambda_0$, $f(\lambda) < \frac{1}{p(\lambda)}$. An event E occurs *with overwhelming probability* (w.o.p.) if its complement occurs with negligible probability; i.e., $1 - \Pr[E] = \mathsf{negl}(\lambda)$.

Super-logarithmic functions. We say that a function $f : \mathbb{N} \mapsto \mathbb{R}$ is *super-logarithmic* in the security parameter λ if $f(\lambda) = g(\lambda) \log \lambda$ for some function $g(\lambda) \to \infty$.

Statistical indistinguishability. Let $\mathbf{p}, \mathbf{q}$ be distributions over a finite state space $\mathcal{S}$. We say $\mathbf{p}$ and $\mathbf{q}$ are *statistically indistinguishable*[7] (written $\mathbf{p} \approx_\lambda \mathbf{q}$) if $d_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) = \mathsf{negl}(\lambda)$ where $d_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) = \frac{1}{2} \sum_{x \in \mathcal{S}} \left| \mathbf{p}_x - \mathbf{q}_x \right|$ denotes *total variation distance*.

2.2 Problem Setting

Network. We model the relay network as a graph $G = (\mathcal{R}, \mathcal{E})$, where $\mathcal{R}$ is a set of n relays and $\mathcal{E}$ is the set of communication links (edges) between relays. The graph topology constrains communication: in each round, a relay may send onions only to its neighbors in G. We assume that G is an r-regular β-expander. That is, if A denotes the normalized adjacency matrix of G, then the second-largest eigenvalue of A in magnitude is at most β.

Synchronous execution. The execution of a communication protocol proceeds in synchronized rounds $t = 0, 1, 2, \ldots, L$. At the beginning of each round, all parties simultaneously receive the traffic sent to them at the end of the previous round. They then perform the local computations specified by the protocol, and subsequently transmit their outgoing traffic for that round. Importantly, we assume that all transmissions sent at the end of round t are delivered without loss or delay at the beginning of round $t + 1$.

Passive adversary model. The passive adversary *non-adaptively corrupts* a constant fraction $\chi < 1$ of the relays in $\mathcal{R}$. Specifically, prior to protocol execution, the adversary selects a subset of relays to corrupt. During protocol execution, the adversary observes all network traffic (i.e., every onion transmitted across every edge in every round) as well as the internal states, randomness, and computations of all corrupted relays.

Remark. Corrupted relays do not contribute to shuffling, since the adversary observes the exact correspondence between incoming and outgoing onions at those relays.

2.3 The Onion Shuffling Protocol

The onion shuffling protocol proceeds in two phases: (1) the *onion forming phase*, in which the sender constructs all onions; and (2) the *routing phase*, in which the onions are routed through the relay network.

Onion forming phase. In this simplified setting, we assume a single honest sender constructs all N onions. Without loss of generality, the underlying messages are labeled $1, 2, \ldots, N$.

For each message $i \in [N]$, the sender samples a routing path $(P_i^{(1)}, P_i^{(2)}, \ldots, P_i^{(L)}, S)$, where L denotes the path length and S denotes the sender. The intermediary relays

$$P_i^{(1)}, P_i^{(2)}, \ldots, P_i^{(L)}$$

are sampled according to a length-L random walk on G. More precisely, the walk begins at a uniformly random relay $P_i^{(1)} \in \mathcal{R}$, and for each $t \in \{1, \ldots, L-1\}$, the next relay $P_i^{(t+1)}$ is chosen uniformly at random from the neighbors of $P_i^{(t)}$.

Using message i, the sampled routing path, and the public keys of the relays on the path, the sender constructs a layered onion $O_i^{(1)}$ The onion is constructed so that relay $P_i^{(t)}$ can peel exactly one encryption layer from onion $O_i^{(t)}$ to recover the next onion $O_i^{(t+1)}$ and the identity of the next relay, $P_i^{(t+1)}$.

Remark. For all of our analyses, we assume that onions are *perfect*. That is, from the adversary's perspective, peeling an onion at an honest relay produces a resulting onion that is information-theoretically independent of the original (unpeeled) onion.

We can make this simplifying assumption because ideal functionalities for onion encryption have been defined along these lines [CL05, AL21]; and Canetti's Universal Composability (UC) theorem [Can01]

[7] Unless stated otherwise, indistinguishability and negligibility are understood to be with respect to the security parameter λ.

allows us to analyze the security of the routing protocol separately from the onion encryption scheme, by assuming ideal onions.

Routing phase. After all N onions are constructed, the sender releases them into the relay network at round 1. Specifically, each onion $O_i^{(1)}$ is sent to the first relay on its path, $P_i^{(1)}$.

At each round $t \in [L]$, every relay *batch-processes* all onions received during that round. For each onion, the relay peels one encryption layer to obtain the peeled onion together with its next destination. The relay then forwards all peeled onions, in uniformly random order, to their respective next destinations. Consequently, each onion $O_i^{(t)}$ reaches relay $P_i^{(t)}$ at round t, and after processing by $P_i^{(t)}$, the peeled onion $O_i^{(t+1)}$ is forwarded to $P_i^{(t+1)}$ at round $t+1$.

At round L, the onions reach the final intermediary relays on their paths, and at the final round $(L+1)$, they are returned to the sender.

2.4 Goals

The goal of the protocol is to show that, after a "small" number of rounds L, the onions become sufficiently shuffled. Here, "small" means any super-logarithmic function of the security parameter λ.

By "sufficiently shuffled," we mean that the adversary cannot distinguish whether $(O_i^{(L)}, O_j^{(L)})$ corresponds to the peeled versions of $(O_i^{(1)}, O_j^{(1)})$ (case $b = 0$ in the game below) or to $(O_j^{(1)}, O_i^{(1)})$ (case $b = 1$), except with a negligibly small advantage.

The onion indistinguishability game. We formalize this notion of indistinguishability via a game between an adversary $\mathcal{A}$ and a challenger. The game

$$\mathsf{OnionIndistinguishabilityGame}(1^\lambda, \mathcal{A}, G, L)$$

is parameterized by the security parameter λ, the adversary $\mathcal{A}$, the communication graph G, and the number of rounds L. For now, we treat N and n as globally fixed parameters. The game proceeds as follows:

1. (*Setup*). The adversary $\mathcal{A}$ selects a set of relays $C \subseteq \mathcal{R}$ to corrupt and sends C to the challenger. The challenger provides $\mathcal{A}$ with the public parameters and the internal states of the corrupted relays.
2. (*Onion formation*). The challenger, acting as the honest sender, constructs N onions.
3. (*Routing phase*). The challenger and $\mathcal{A}$ jointly execute the routing phase of the protocol for L rounds. The adversary controls the corrupted relays, while the challenger controls all honest relays. Throughout the execution, $\mathcal{A}$ observes all information specified by the adversarial model.
4. (*Challenge query*). After observing the protocol execution, $\mathcal{A}$ selects two output onions observed at the end of round L. Let these output onions be denoted by $O_i^{(L)}$ and $O_j^{(L)}$.
5. (*Challenge response*). The challenger samples a uniformly random bit $b \leftarrow\!\$\ \{0,1\}$. If $b = 0$, the challenger sends $\mathcal{A}$ the corresponding initial onions in the correct order:
$$(O_i^{(1)}, O_j^{(1)}).$$
If $b = 1$, the challenger sends them in the swapped order:
$$(O_j^{(1)}, O_i^{(1)}).$$
6. (*Guess*). The adversary outputs a guess $b' \in \{0,1\}$ and wins if $b' = b$.

Onion indistinguishability. A protocol achieves *onion indistinguishability* against an adversary class $\mathbb{A}$ on the communication graph G within L rounds if, for every adversary $\mathcal{A} \in \mathbb{A}$,

$$\left|\Pr\left[\mathcal{A} \text{ wins } \mathsf{OnionIndistinguishabilityGame}(1^\lambda, \mathcal{A}, G, L)\right] - \frac{1}{2}\right| \leq \mathsf{negl}(\lambda),$$

where λ denotes the security parameter.

Intuitively, this captures the idea that even after conditioning on the adversary's entire view, the corre-

spondence between final onions and their initial forms should remain hidden.

Remark. An efficient onion shuffling protocol that achieves onion indistinguishability with any super-logarithmic round complexity and any super-logarithmic server load is a powerful routing primitive. In particular, it enables anonymous communication via onion routing and mixnets by ensuring that messages cannot be linked to their sources after a small number of hops, even under partial network corruption. We will explore applications to anonymous communication protocols in more detail in Section 6.

3 Our Main Result

In this section, we prove that our onion shuffling protocol defined in Section 2.3 achieves onion indistinguishability on sparse expander networks with no asymptotic overhead in round complexity or server load compared to the fully connected setting [ALU18]. In particular, the requirement for any super-logarithmic round complexity is *optimal*, matching the known lower bound [Chr20] for passive adversaries that corrupt a constant fraction of relays in the fully connected setting. Below we give the formal statement of our main result.

Theorem 1. *Let G be an r-regular β-expander graph on n relays, where $r > 2$ and $\beta < 1$ are constants independent of λ. Suppose N onions are routed on G according to our onion shuffling protocol. If, for some functions $g_1(\lambda), g_2(\lambda) \to \infty$, the number of rounds and server load satisfy*

$$L = g_1(\lambda) \log \lambda \qquad \text{and} \qquad \frac{N}{n} = g_2(\lambda) \log \lambda,$$

then our protocol achieves onion indistinguishability *on G within L rounds against the passive adversary corrupting up to a constant fraction $\chi \in [0, 1)$ of the relays.*

At first glance, one might hope that Theorem 1 follows directly from the standard mixing properties of random walks on expander graphs. However, this intuition is misleading because random-walk mixing and onion indistinguishability concern different conditional probability distributions.

To make this distinction precise, let $P^{(1)}, P^{(2)}, \ldots, P^{(L)}$ denote the (random) routing path of an onion. The standard mixing guarantee of a constant β ensures that if the onion starts at relay $u \in \mathcal{R}$, then after a super-logarithmic number of random-walk steps, its location is negligibly close to uniform over the n relays:

$$\Pr[P^{(L)} = v \mid P^{(1)} = u] \approx_\lambda \frac{1}{n}.$$

This is an unconditioned statement about the location of a single random walk. Equivalently, an observer sees that the onion starts at u, is then blindfolded for L rounds, and is finally asked where the onion is. The random-walk mixing guarantee implies that the observer's belief about the onion's final relay location is close to uniform.

Our security goal is much stronger, as the passive adversary is *not* blindfolded during the protocol. Instead, it observes the entire execution: the traffic volume on every edge in every round, the batches arriving at every relay, and the exact input-output correspondences at corrupted relays. We refer to this collection of observations as the adversary's *view*, denoted $\mathsf{View}_{\mathcal{A}}$. Thus, by the end of the protocol, the routing choices of all onions have already induced a network transcript of onion transitions through the network, and the adversary's uncertainty is conditioned on this view.

The belief matrix, Z. Therefore, the relevant question is: after observing the full view $\mathsf{View}_{\mathcal{A}}$, which final onion does the adversary believe originated as onion i? We capture this by the belief matrix Z, where

$$Z_{i,j} = \Pr[\text{final onion } O_j^{(L)} \text{ originated as initial onion } O_i^{(1)} \mid \mathsf{View}_{\mathcal{A}}].$$

Onion indistinguishability requires that, for every initial onion $O_i^{(1)}$, this posterior distribution is close to uniform over final onions:

$$d_{\mathrm{TV}}\left(Z_i, \frac{1}{N}\mathbf{1}\right) = \mathsf{negl}(\lambda).$$

The proof of our main result therefore reduces to showing that, with overwhelming probability, the belief matrix Z is negligibly close to the uniform matrix. We establish this through two ingredients, which are isolated and proved in the next two sections: an *expectation bound* (Lemma 1) showing that the adversary's belief is close to uniform on average, and a *concentration bound* (Lemma 2) showing that the realized belief matrix is tightly concentrated around this expectation. We now show how these two lemmas imply the main theorem.

Proof of Theorem 1. Our analysis focuses on the distribution Z (defined above) induced by the adversary's view $\mathsf{View}_{\mathcal{A}}$ over these possible correlations.

By Lemma 1, we have

$$\mathbb{E}[Z] = \frac{1}{N}\,\mathbf{1}\mathbf{1}^{\top},$$

where the expectation is taken over the randomness of the protocol execution, including the random-walk routing choices of all onions.

By Lemma 2, for every $i, j \in [N]$ and super-logarithmic m,

$$\mathrm{Var}\left[Z_{i,j}\right] \le \left(\frac{3n}{N}\right)^m \cdot \left(2^{1-m} + m\left(\frac{4N}{3n}\right)^2\right).$$

From these results, we can bound the expected ℓ_1-norm deviation of any row i:

$$\begin{aligned}\mathbb{E}\left[\left\|Z_i - \frac{1}{N}\mathbf{1}\right\|_1\right] &\le N \cdot \sqrt{\max_j \mathrm{Var}\left[Z_{i,j}\right]} \\ &\le N \cdot \sqrt{\left(\frac{3n}{N}\right)^m \cdot \left(2^{1-m} + m\left(\frac{4N}{3n}\right)^2\right)}\end{aligned}$$

Applying Markov's Inequality and taking a union bound over all $i \in [N]$ gives:

$$\begin{aligned}\Pr\left[\exists i \in [N] \mid \left\|Z_i - \frac{1}{N}\mathbf{1}\right\|_1 \ge \epsilon\right] &\le \frac{N}{\epsilon} \cdot \mathbb{E}\left[\left\|Z_i - \frac{1}{N}\mathbf{1}\right\|_1\right] \\ &\le \frac{N^2}{\epsilon} \cdot \sqrt{\left(\frac{3n}{N}\right)^m \cdot \left(2^{1-m} + m\left(\frac{4N}{3n}\right)^2\right)}\end{aligned}$$

Choosing $\epsilon = \left(\frac{1}{2}\right)^m$, we obtain:

$$\Pr\left[\exists i \in [N] \mid \left\|Z_i - \frac{1}{N}\mathbf{1}\right\|_1 \ge \left(\frac{1}{2}\right)^m\right] \le N^2 \cdot \sqrt{\left(\frac{12n}{N}\right)^m \cdot \left(2^{1-m} + m\left(\frac{4N}{3n}\right)^2\right)}$$

For super-logarithmic m, this probability is negligible in λ. Thus with overwhelming probability, for every i,

$$d_{\mathrm{TV}}\left(Z_i, \frac{1}{N}\mathbf{1}\right) = \frac{1}{2}\left\|Z_i - \frac{1}{N}\mathbf{1}\right\|_1 = \mathsf{negl}(\lambda).$$

Therefore, after observing the full protocol execution, the adversary's posterior belief over the origin of every final onion is negligibly close to uniform. In particular, for any two final onions chosen in the onion indistinguishability game, the adversary cannot distinguish which of the two corresponding round-1 onions generated which final onion except with negligible advantage. Hence the adversary wins the onion indistinguishability game with probability at most $1/2 + \mathsf{negl}(\lambda)$. □

4 Expectation Bound

In this section, we prove the first lemma used in the proof of Theorem 1. Intuitively, although the passive adversary can perfectly track onions through corrupted relays, those corrupted portions of the path only act as transparent channels between honest shuffles. Once these transparent channels are collapsed, the expected transitions of belief mass between honest relays behaves like a random walk (projected onto the set of honest relays) on the original expander graph.

Lemma 1 (Expectation)**.** *For every $i \in [N]$,*

$$d_{TV}\left(\mathbb{E}[Z_i], \frac{1}{N}\mathbf{1}\right) = \mathsf{negl}(\lambda).$$

The proof proceeds in two steps. First, we define a time-layered relay state space that captures the passive adversary's effective view of onion flows between honest relays. In this view, paths through corrupted relays are collapsed into single transparent channels. Second, we show that, in expectation over the routing randomness of the protocol, this collapsed process has the same transition probabilities as a simple random walk on G observed only at honest relays. Since G is an expander and L is super-logarithmic, this expected belief becomes negligibly close to uniform.

4.1 The Time-Layered Relay Model

Corrupted channels. Let $C \subseteq \mathcal{R}$ be the set of corrupted relays, and let $\mathcal{H} = \mathcal{R} \setminus C$ be the set of honest relays. Define a *corrupted channel* from an honest relay $u \in \mathcal{H}$ to an honest relay $v \in \mathcal{H}$ as a directed path in G that starts at u, ends at v, and has only corrupted relays in-between. We also treat a direct edge $u \to v$ between two honest relays as a degenerate corrupted channel of length 1.

From the passive adversary's perspective, corrupted channels act as transparent pass-throughs for onions: if an onion leaves an honest relay u, passes only through corrupted relays, and next reaches an honest relay v, then the adversary can track that onion perfectly along the entire path from u to v. Thus, the only points at which the adversary loses information about an onion's location are at honest relays. The effective view of the adversary can therefore be described as a sequence of onion flows between honest relays, where each flow may span one or more rounds.

Time-layered state space. We lift the graph-level view of $\mathcal{H}$ (the set of honest relays) into a time-layered state space

$$\mathcal{S} = \mathcal{H} \times [L],$$

where a state $(u, t) \in \mathcal{S}$ represents the batch of onions processed by honest relay u in round t.

Random transition kernel Q. For a fixed execution of the protocol, we define a matrix $Q \in [0,1]^{\mathcal{S}\times\mathcal{S}}$ that records the adversary's effective view of onion flows between honest relays. For states $(u, t), (v, t') \in \mathcal{S}$ with

$t' > t$, let

$$Q_{(u,t),(v,t')} \;=\; \text{the fraction of onions at } (u,t) \text{ whose next honest relay is } v \text{ at round } t'.$$

Equivalently, $Q_{(u,t),(v,t')}$ is the fraction of onions at (u,t) that are routed to (v,t') through a corrupted channel of length $t' - t$. If no such flow occurs, this entry is 0.

Thus, Q is the weighted adjacency matrix of a time-layered DAG. Each nonzero edge corresponds to an observed flow of onions through corrupted channels between honest relays. Since onions only move forward in time, Q is strictly upper triangular.

A path in this DAG represents a sequence of honest shuffling events. For example, a path

$$(h_1, t_1) \to (h_2, t_2) \to \cdots \to (h_k, t_k)$$

means that onions flow from honest relay $h_1 \in \mathcal{H}$ at round t_1 to honest relay $h_2 \in \mathcal{H}$ at round t_2, then to honest relay $h_3 \in \mathcal{H}$ at round t_3, and so on, where each transition may pass through zero or more corrupted relays.

Independence of rows. The matrix Q is random because it depends on the routing choices made by the onions in a particular execution. However, its rows are independent: the outgoing flows from a fixed time-layered state (u,t) depend only on the next routing choices of the onions in the batch at (u,t). In particular, if (u,t) and (v,t') are distinct time-layered states, then the random variables determining the outgoing row of Q from (u,t) are independent of those determining the outgoing row from (v,t').

Powers of Q. For $k \geq 1$, the matrix Q^k captures transitions involving exactly k honest shuffling events. Specifically, for $(u,t),(v,t') \in \mathcal{S}$ with $t' > t$, the entry

$$(Q^k)_{(u,t),(v,t')}$$

is the total weight of all time-respecting paths in the DAG that begin at (u,t), end at (v,t'), and use exactly k edges. Equivalently, it is the likelihood, from the adversary's perspective in this fixed execution, that an onion at (u,t) reaches (v,t') after exactly k honest shuffles.

Transitive closure of Q. Since Q is strictly upper triangular, its powers vanish after at most L time steps. Therefore,

$$\sum_{k=1}^{L-1} Q^k = (I - Q)^{-1} - I,$$

where I is the identity matrix. Here, the entry $((I - Q)^{-1} - I)_{(u,t),(v,t')}$ is the total probability, from the adversary's perspective in this fixed execution, that an onion at (u,t) eventually reaches (v,t') through any positive number of honest-to-honest transitions.

Importantly, this transitive closure is the relay-level analogue of the belief matrix Z: it tracks where a given initial onion is, while Z tracks which final onions may carry the identity mass of a given initial onion.

Deterministic kernel R. Let A be the normalized adjacency matrix of G, block-decomposed according to the honest and corrupted relays:

$$A = \begin{pmatrix} A_{\mathcal{H}\mathcal{H}} & A_{\mathcal{H}C} \\ A_{C\mathcal{H}} & A_{CC} \end{pmatrix}.$$

We define a deterministic kernel $R \in [0,1]^{\mathcal{S}\times\mathcal{S}}$ by setting, for $(u,t),(v,t') \in \mathcal{S}$,

$$R_{(u,t),(v,t')} = \begin{cases} (A_{\mathcal{H}\mathcal{H}})_{u,v} & \text{if } t' - t = 1, \\ \left(A_{\mathcal{H}C}(A_{CC})^{t'-t-2}A_{C\mathcal{H}}\right)_{u,v} & \text{if } t' - t > 1, \\ 0 & \text{otherwise.} \end{cases}$$

The first case corresponds to a direct transition from one honest relay to another. The second corresponds to a transition that leaves an honest relay, passes only through corrupted relays for the intermediate rounds, and next reaches an honest relay. Thus, R is the entrywise expectation of the random kernel Q.

4.2 Proof of Expectation Lemma

We now prove Lemma 1 in three steps. First, we show that the expected transitive closure of the random kernel Q is the transitive closure of the deterministic kernel R. Second, we show that this deterministic transitive closure exactly recovers the ordinary random-walk transition probabilities on G. Finally, we use the spectral expansion of G to conclude that the expected belief distribution is negligibly close to uniform.

Proof of Lemma 1. We first compare the random kernel Q to the deterministic kernel R. For every $k \geq 1$, an entry of Q^k is a sum over time-respecting paths of length k in the time-layered DAG. Each such path uses a distinct row of Q at each step, and by the row-wise independence of Q, the expectation of the product of the edge weights along the path factors into the product of their expectations. Since R was defined entrywise as the expectation of Q, summing over all such paths gives

$$\mathbb{E}\Big[Q^k\Big] = R^k.$$

Therefore, by linearity of expectation,

$$\mathbb{E}\Big[(I-Q)^{-1} - I\Big] = \mathbb{E}\left[\sum_{k=1}^{L-1} Q^k\right] = \sum_{k=1}^{L-1} \mathbb{E}\Big[Q^k\Big] = \sum_{k=1}^{L-1} R^k = (I-R)^{-1} - I.$$

It remains to identify the entries of $(I-R)^{-1} - I$. We claim that for all $u, v \in \mathcal{H}$ and all $t' > t$,

$$\big((I-R)^{-1} - I\big)_{(u,t),(v,t')} = (A^{t'-t})_{u,v},$$

meaning the transitive closure of R counts exactly the same length-$(t'-t)$ random-walk paths from u to v as the original walk on G, but viewed only at the times when the walk visits honest relays.

To prove the claim, let z be a formal variable and consider the block matrix

$$I - zA = \begin{pmatrix} I - zA_{HH} & -zA_{HC} \\ -zA_{CH} & I - zA_{CC} \end{pmatrix}.$$

Then its inverse is the resolvent $(I-zA)^{-1} \;=\; I + zA + z^2A^2 + z^3A^3 + \cdots$.

By construction, the coefficient of each z^m in this expansion is exactly A^m. (In particular, for $u, v \in \mathcal{H}$, the entry $(A^m)_{u,v}$ gives the likelihood that a random walk in G starting at u ends at v after m steps).

By the block inversion formula, the top-left block is

$$(I-zA)^{-1}_{HH} = \big(I - zA_{HH} - z^2A_{HC}(I-zA_{CC})^{-1}A_{CH}\big)^{-1}.$$

Define the generating function by

$$B(z) \;=\; zA_{\mathcal{H}\mathcal{H}} + z^2A_{\mathcal{H}C}(I-zA_{CC})^{-1}A_{C\mathcal{H}}.$$

so that we can write:

$$(I-zA)^{-1}_{\mathcal{H}\mathcal{H}} \;=\; (I-B(z))^{-1}.$$

On the left-hand side, we have

$$(I-zA)^{-1}_{\mathcal{H}\mathcal{H}} = I + zA_{\mathcal{H}\mathcal{H}} + z^2(A^2)_{\mathcal{H}\mathcal{H}} + z^3(A^3)_{\mathcal{H}\mathcal{H}} + \cdots,$$

so for all $u, v \in \mathcal{H}$, the (u,v)-entry of the coefficient of $z^{t'-t}$ is equal to $(A^{t'-t})_{u,v}$. On the right-hand side, we

expand

$$(I - B(z))^{-1} = I + B(z) + B(z)^2 + B(z)^3 + \cdots,$$

showing that each term corresponds to a concatenation of honest-to-honest hops (through corrupted channels). Every such hop contributes a power of z equal to the number of steps in the original graph: a direct $\mathcal{H} \to \mathcal{H}$ edge contributes z^1, while a corrupted channel $\mathcal{H} \to C \to \cdots \to C \to \mathcal{H}$ of length k contributes z^k. When several hops are multiplied together, the exponents add, so a sequence of hops of lengths $k_1, \ldots, k_j$ contributes a factor of $z^{k_1+\cdots+k_j}$. Thus, the coefficient of $z^{t'-t}$ in the (u, v)-entry of $(I - B(z))^{-1}$ corresponds exactly to $((I - R)^{-1} - I)_{(u,t),(v,t')}$. So $(I - B(z))^{-1}$ is the generating function whose coefficients reproduce the entries of $(I - R)^{-1} - I$.

Since the two generating functions are equal, their coefficients must match term-by-term. It follows that for all $u, v \in \mathcal{H}$ and $t' > t$,

$$\mathbb{E}\Big[((I - Q)^{-1} - I)_{(u,t),(v,t')}\Big] = (A^{t'-t})_{u,v}.$$

Since G is an r-regular β-expander with constant $\beta < 1$, for super-logarithmic $t' - t$,

$$\left|(A^{t'-t})_{u,v} - \frac{1}{n}\right| = \mathsf{negl}(\lambda).$$

Thus, in expectation, the relay-level belief mass induced by the adversary's view is negligibly close to uniform over relays.

Finally, by aggregating the contributions of $(A^{t'-t})_{u,v}$ across all batches at round $t' = L$, we recover that for every initial onion $i \in [N]$,

$$d_{\mathrm{TV}}\left(\mathbb{E}[Z_i], \frac{1}{N}\mathbf{1}\right) = \mathsf{negl}(\lambda).$$

□

5 Concentration Bound

We now show that the belief matrix arising in a concrete execution is tightly concentrated around its expectation.

Lemma 2 (Variance bound)**.** *With overwhelming probability, for every $i, j \in [N]$,*

$$\mathrm{Var}\Big[Z_{i,j}\Big] \le \left(\frac{3n}{N}\right)^m \cdot \left(2^{1-m} + m\left(\frac{4N}{3n}\right)^2\right),$$

for some super-logarithmic m.

Before proving Lemma 2, we introduce a path-based representation of the entries of Z.

5.1 Candidate Peeling Paths

We describe $Z_{i,j}$ as a sum over possible explanations, from the adversary's point of view, for how the input onion $O_i^{(1)}$ could have evolved into the output onion $O_j^{(L)}$.

Candidate peeling pairs. Consider a relay processing a batch of onions in some round t. From the adversary's point of view, each incoming round-t onion may have several possible outgoing successors after the relay peels one encryption layer and forwards the resulting round-t+1 onions. We call each such possible input-output correspondence a *candidate peeling pair*. We write such a pair as

$$(O_a^{(t)}, O_b^{(t+1)}),$$

where $O_a^{(t)}$ is an incoming onion at round t and $O_b^{(t+1)}$ is a possible peeled successor (from adversary's point of view) at round $t+1$.

We assign each candidate peeling pair a transition weight

$$w(O_a^{(t)}, O_b^{(t+1)}),$$

which represents the likelihood that $O_b^{(t+1)}$ is the *true* peeled successor of $O_a^{(t)}$, conditioned on the adversary's view. If the relay is *honest* and processes a batch of size s, then the adversary cannot distinguish among the s outgoing onions in the shuffled batch. Therefore, for any fixed incoming onion, each outgoing onion is assigned equal weight $1/s$ as a possible peeled successor. However, if the relay is *corrupted*, then the adversary observes the relay's internal computation and therefore learns the true input-output correspondence. In this case, among the s^2 possible input-output pairs for the batch, exactly s are true peeling pairs and carry a transition weight 1, while the remaining are assigned 0.

Candidate peeling paths. A *candidate peeling path* from $O_i^{(1)}$ to $O_j^{(L)}$ is a sequence of candidate peeling pairs connecting the round-1 onion $O_i^{(1)}$ to the round-L onion $O_j^{(L)}$. Equivalently, it is one possible explanation, consistent with the adversary's view, for how $O_i^{(1)}$ could have evolved into $O_j^{(L)}$. Figure 1 illustrates one such path in the time-layered onion state space.

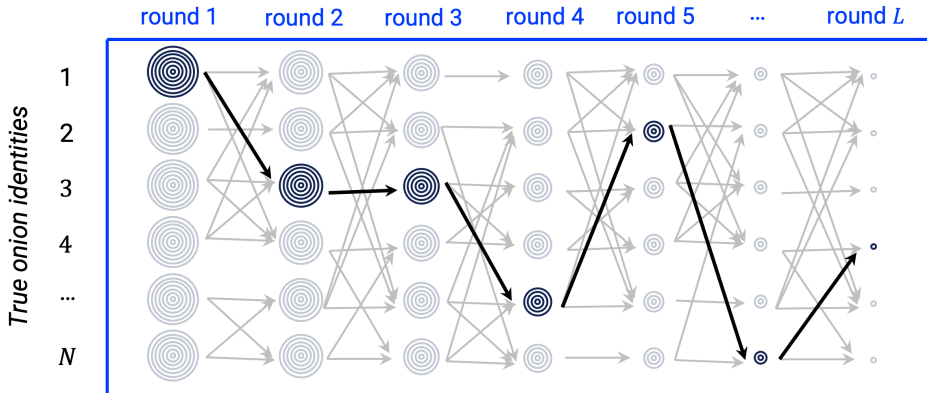


Figure 1: *An example candidate peeling path in the time-layered onion state space O. Each onion $O_i^{(t)}$ (represented at row i, column t) denotes the layer-t onion with (true) identity i. Arrows indicate non-zero transitions between candidate peeling pairs. The bold black arrows highlight one example of a candidate peeling path from a round-1 onion to a round-L onion.*

A path ρ may be written as

$$\rho = O_{a_1}^{(1)} \xrightarrow{w(O_{a_1}^{(1)}, O_{a_2}^{(2)})} O_{a_2}^{(2)} \xrightarrow{w(O_{a_2}^{(2)}, O_{a_3}^{(3)})} O_{a_3}^{(3)} \xrightarrow{w(O_{a_3}^{(3)}, O_{a_4}^{(4)})} \cdots \xrightarrow{w(O_{a_{L-1}}^{(L-1)}, O_{a_L}^{(L)})} O_{a_L}^{(L)}.$$

The weight $W(\rho)$ of a candidate peeling path is the product of the transition weights of its candidate peeling pairs:

$$W(\rho) = \prod_{t=1}^{L-1} w(O_{a_t}^{(t)}, O_{a_{t+1}}^{(t+1)}).$$

Batch-coupled paths. The only dependence between path weights arises when two paths use candidate peeling pairs from the same honest batch. To see this, consider an honest relay that shuffles a batch of size s in round t. From the adversary's point of view, each incoming onion has s possible outgoing successors, so this batch induces s^2 candidate peeling pairs, each with transition probability $1/s$. Therefore, the weights of any two paths that use candidate peeling pairs from this same batch both depend on the same random

variable s. We call such paths *batch-coupled*; see Figure 2 for an illustration. Paths that are not batch-coupled depend on disjoint honest batches and are independent.

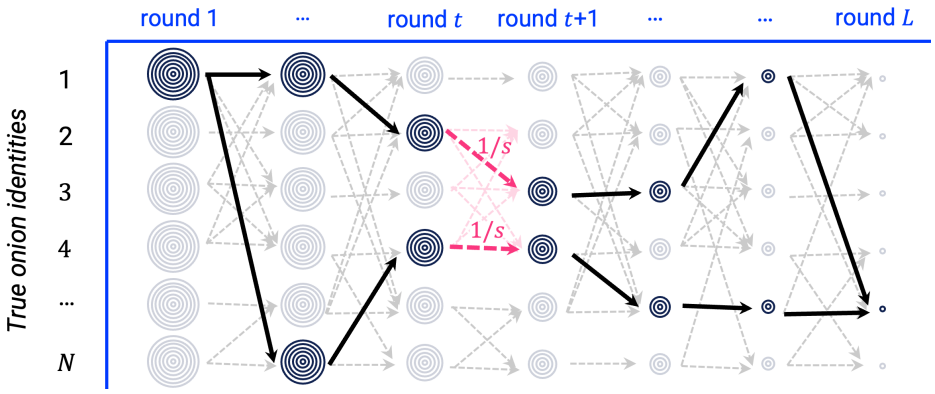


Figure 2: *An example of two batch-coupled candidate peeling paths, each containing at least one candidate peeling pair from the same honest batch at round t (highlighted in pink). The weights of these paths both depend on the same random variable s.*

Recovering $Z_{i,j}$. Although a single candidate peeling path is only one possible explanation for how onion i evolved into onion j, the adversary's total posterior belief is obtained by summing over all such explanations. Let $\mathcal{P}_{i\to j}$ denote the set of candidate peeling paths from round-1 onion i to round-L onion j. Then

$$Z_{i,j} = \sum_{\rho \in \mathcal{P}_{i\to j}} W(\rho).$$

Figure 3 illustrates this sum.

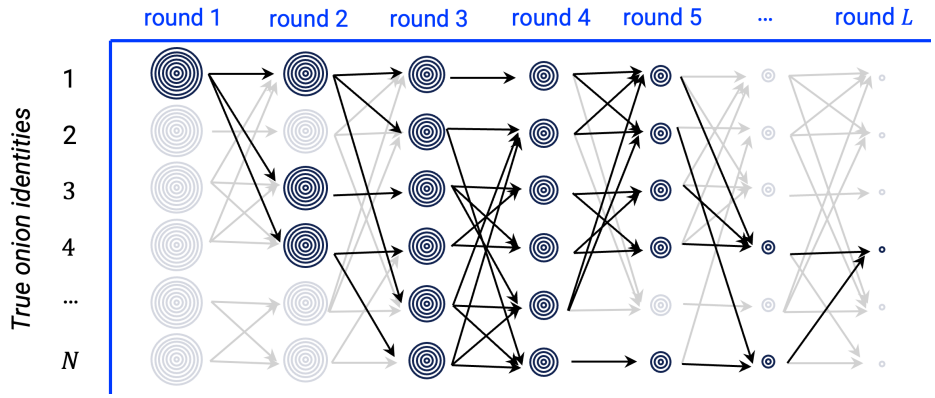


Figure 3: *All candidate peeling paths $\mathcal{P}_{i\to j}$ contributing to $Z_{i,j}$; here, $i = 1$ and $j = 4$. From the adversary's perspective, the event that $O_i^{(1)}$ evolves into $O_j^{(L)}$ decomposes as a disjoint union over all possible explanations (paths in $\mathcal{P}_{i\to j}$).*

Therefore, to bound the variance of $Z_{i,j}$, it suffices to understand the weights of candidate peeling paths and the dependence between different paths.

5.2 Proof of the Variance Bound

Proof of Lemma 2. By Lemma 5, with overwhelming probability, all but a negligible fraction of the belief mass originating from onion i passes through at least a super-logarithmic m number of honest relays be-

fore round L. Thus, up to negligible error, it suffices to bound the contribution of candidate peeling paths containing at least m honest candidate peeling pairs. For any longer path, we keep only its first m honest candidate peeling pairs; since each additional honest peeling step only multiplies the path weight by a factor < 1, the full path has weight at most the weight of its length-m honest prefix.

By applying Chernoff bounds (Lemma 9) and a union bound (Lemma 8), we ensure that with overwhelming probability, every relay in every round has batch size between $\frac{N}{n}(1 \pm \delta)$ for any arbitrarily small constant $\delta > 0$.

Thus w.o.p., every candidate peeling pair at an honest relay has transition probability at most $\frac{3n}{2N}$.

For any candidate peeling path ρ, the weight of ρ satisfies

$$w(\rho) \leq \left(\frac{3n}{2N}\right)^m.$$

Consequently,

$$\mathbb{E}\Big[w(\rho)^2\Big] \leq \left(\frac{3n}{2N}\right)^{2m}.$$

We now bound the variance of the path sum.

Let $\mathcal{P}^{(m)}_{i\to j}$ denote the set of length-m honest prefixes of candidate peeling paths from onion i to onion j.

Recall that two candidate peeling path prefixes $\rho, \rho' \in \mathcal{P}^{(m)}_{i\to j}$ are *batch-coupled* if they contain candidate peeling pairs from the same honest batch. Prefixes that are not batch-coupled depend on disjoint honest batches and are independent. Therefore, using the variance bound for sums of dependent random variables, we obtain

$$\begin{aligned}
\mathrm{Var}\Big[Z_{i,j}\Big] &= \mathrm{Var}\left[\sum_{\rho\in\mathcal{P}^{(m)}_{i\to j}} w(\rho)\right] \\
&\leq \sum_{\rho\in\mathcal{P}^{(m)}_{i\to j}} \mathrm{Var}[w(\rho)] + 2 \sum_{\substack{\rho,\rho'\in\mathcal{P}^{(m)}_{i\to j} \\ \text{batch-coupled}}} \mathrm{Cov}[w(\rho), w(\rho')] \\
&\leq (\#\{\text{length-}m \text{ prefixes } i \to j\} + \#\{\text{batch-coupled pairs of length-}m \text{ prefixes } i \to j\}) \cdot \max_{\rho\in\mathcal{P}^{(m)}_{i\to j}} \mathbb{E}\Big[w(\rho)^2\Big] \\
&\leq (\#\{\text{length-}m \text{ prefixes } i \to j\} + \#\{\text{batch-coupled pairs of length-}m \text{ prefixes } i \to j\}) \cdot \left(\frac{3n}{2N}\right)^{2m}.
\end{aligned}$$

It remains to bound the number of length-m candidate peeling paths from onion i to onion j, and the number of pairs of such paths that are batch-coupled. Since every honest batch has size at most $\frac{4N}{3n}$, each incoming onion has at most $\frac{4N}{3n}$ candidate successors at every honest peeling step. Therefore, the number of length-m candidate peeling path prefixes from i to j is at most

$$\#\{\text{length-}m \text{ prefixes } i \to j\} \leq \left(\frac{4N}{3n}\right)^m.$$

Each prefixed path passes through m honest batches. At any one of these batches, there are at most $\left(\frac{4N}{3n}\right)^2$ candidate peeling pairs where another path could overlap and be batch-coupled. Since there are at most

$\left(\frac{4N}{3n}\right)^m$ paths, the total number of batch-coupled pairs is bounded by

$$m\left(\frac{4N}{3n}\right)^2\left(\frac{4N}{3n}\right)^m = m\left(\frac{4N}{3n}\right)^{m+2}.$$

Combining these bounds, we obtain

$$\begin{aligned}\mathrm{Var}\left[Z_{i,j}\right] &\le \left(\left(\frac{4N}{3n}\right)^m + m\left(\frac{4N}{3n}\right)^{m+2}\right)\cdot\left(\frac{3n}{2N}\right)^{2m} \\ &\le \left(\frac{3n}{N}\right)^m\left(2^{1-m} + m\left(\frac{4N}{3n}\right)^2\right).\end{aligned}$$

□

6 Application to Anonymous Communication Protocols

We show how our main result (Theorem 1) can be leveraged to obtain new anonymity protocols for sparse expander networks that simultaneously achieve *anonymity* (an adversary cannot reliably link senders to recipients), *efficiency* (polylogarithmic communication complexity), and *availability* (honest messages are successfully delivered).

We first formalize the anonymous communication setting and the adversarial model considered in this section. We then show how two protocols originally analyzed in the complete network setting can be instantiated over sparse expander networks: one achieving differential anonymity against active adversaries, and one achieving "full" anonymity (albeit with standard leakages).

6.1 Problem Setting

Network. We model the relay network as a r-regular, β-expander graph $G = (\mathcal{R}, \mathcal{E})$ with $r > 2$ and $\beta < 1$, where $\mathcal{R}$ is a set of n relays and $\mathcal{E}$ is the set of communication links between relays. The topology of G constrains communication: in each round, a relay may send packets only to its neighbors in G.

Participants. Outside of the relay network, there are N participants $\mathcal{P} = \{1, 2, \ldots, N\}$ who wish to communicate anonymously by routing their messages through the relay network. We assume that each participant has at least one message to send.

Synchronous execution. We assume a *synchronous communication model* in which time proceeds in discrete rounds $t = 0, 1, 2, \ldots, L$. Onions sent at the end of round t are delivered at the beginning of round $t + 1$, without loss or delay.

Inputs. An execution of a communication protocol Π is parameterized by a security parameter λ, an adversary class $\mathbb{A}$, and an input vector $\sigma = (\sigma_1, \ldots, \sigma_N)$, where each σ_i specifies the messages sent by participant $i \in \mathcal{P}$:

$$\sigma_i \;=\; \{\,(m, j) \in \mathcal{M} \times \mathcal{P} \mid j \text{ is the recipient of a message } m \text{ from } i\,\}.$$

Here, $\mathcal{M}$ is a space of fixed-length messages.

Adversary model and adversarial view. In this section, we consider an *active adversary*, the strongest standard adversarial model. As in the passive setting, the adversary non-adaptively corrupts a set of relays and participants, denoted C, before the protocol begins and observes all network traffic, internal states, randomness, and computations of the corrupted parties. In addition, an active adversary may cause corrupted parties to deviate arbitrarily from the protocol, including by dropping, modifying, delaying, or injecting packets.

The *adversarial view*, denoted $\mathsf{View}_{\Pi,\mathcal{A}}(\sigma)$, is a random variable[8] consisting of all information available to adversary $\mathcal{A}$ during an execution of protocol Π on input σ. For an active adversary, this includes the full network transcript, as well as the complete execution state of every corrupted relay: its randomness, internal computations, received messages, outgoing messages, and any behavior specified by the adversary.

Remark. As in our analysis of onion shuffling in the expander setting, we continue to model onions as idealized objects. In particular, this abstraction suppresses the cryptographic assumptions required to realize onions that are computationally indistinguishable from ideal ones, including the existence of a public-key infrastructure (PKI) and standard cryptographic primitives commonly assumed in secure onion encryption schemes such as those of [CL05, AL21].

When our anonymity protocols are instantiated with concrete cryptographic constructions, we additionally assume whatever assumptions are required by those constructions. Under these assumptions, the anonymity guarantees we prove translate to their computational counterparts. In particular, the first protocol achieves computational differential anonymity, while the second achieves computational anonymity.

6.2 Definitions

Informally, a communication protocol achieves anonymity if no adversary can distinguish whether the protocol was executed on one input vector σ^0 or another input vector σ^1. However, without restrictions on the input vectors, this requirement is too strong in the presence of corrupted parties. In particular, the adversary necessarily learns the inputs of corrupted senders and the messages received by corrupted recipients, which, in a correct execution without adversarial interference, correspond exactly to the messages designated for the corrupted participants in the input vector. These leakages are unavoidable.

Leakage. More generally, a protocol can only be expected to hide information beyond some prescribed leakage. We formalize this using a leakage function $\mathcal{L}$ defined over the universe Σ of input vectors. The corrupted set C, together with $\mathcal{L}$, induces an equivalence relation over Σ: we write

$$\sigma^0 \equiv_{C,\mathcal{L}} \sigma^1$$

if the following conditions hold: (1) the corrupted parties in C have identical inputs and outputs in both σ^0 and σ^1 (capturing the unavoidable leakage); and (2) $\mathcal{L}(\sigma^0) = \mathcal{L}(\sigma^1)$. Security is required only for pairs of inputs that are equivalent in this sense.

I/O pattern leakage. In this work, we consider a generalization of a standard leakage notion from the provable anonymity literature, which we call the *I/O pattern* leakage and denote by $\mathcal{L}_{\mathsf{io}}$. Given an input vector

$$\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_N),$$

the leakage function outputs two sequences

$$(x_1, x_2, \ldots, x_N) \quad \text{and} \quad (y_1, y_2, \ldots, y_N),$$

where each $x_i = |\sigma_i|$ denotes the number of messages participant i sends, and each $y_i = \sum_{j=1}^{N} |\{(m, i) \in \sigma_j\}|$ denotes the number of messages participant i receives in a correct execution without adversarial message dropping.

Anonymity. A communication protocol Π is *anonymous* from adversaries in the class $\mathbb{A}$ with respect to leakage function $\mathcal{L}$ if, for every adversary $\mathcal{A} \in \mathbb{A}$, every corrupted set C chosen by $\mathcal{A}$, and every pair of input vectors $\sigma^0, \sigma^1 \in \Sigma$ satisfying

$$\sigma^0 \equiv_{C,\mathcal{L}} \sigma^1,$$

[8] Formally, $\mathsf{View}_{\Pi,\mathcal{A}}(\sigma)$ is taken over the joint randomness of the protocol and the adversary.

the corresponding adversarial views are statistically indistinguishable:

$$\mathsf{View}_{\Pi,\mathcal{A}}(\sigma^0) \approx_\lambda \mathsf{View}_{\Pi,\mathcal{A}}(\sigma^1).$$

Differential anonymity. Two inputs,

$$\sigma^0 = (\sigma^0_1, \sigma^0_2, \ldots, \sigma^0_N) \quad \text{and} \quad \sigma^1 = (\sigma^1_1, \sigma^1_2, \ldots, \sigma^1_N),$$

are *neighboring* if they are identical except that the inputs of two *honest* parties, each specifying a message to send to an *honest* recipient, are swapped. (As explained above, the input vectors σ^0 and σ^1 are required to agree on the inputs and "outputs" of the corrupted parties, since any discrepancy in this information would be trivially distinguishable by the adversary, rendering the anonymity notion unachievable.)

A communications protocol Π is (ϵ, δ)-*differentially anonymous* from adversaries in the class $\mathbb{A}$ if for all $\mathcal{A} \in \mathbb{A}$ and for any pair of neighboring inputs σ^0, σ^1,

$$\Pr[\mathsf{View}_{\Pi,\mathcal{A}}(\sigma^0) \in \mathcal{V}] \;\le\; e^\epsilon \cdot \Pr[\mathsf{View}_{\Pi,\mathcal{A}}(\sigma^1) \in \mathcal{V}] \;+\; \delta,$$

for any set $\mathcal{V}$ of views by $\mathcal{A}$ while participating in Π.

Efficiency. The efficiency of a protocol Π is characterized by: 1) its *round complexity* L, the number of synchronous rounds it takes for a message to reach its destination, and 2) its *onion cost*, the average number of packets processed per message (in the input vector) per round. We say that Π is *efficient* if both these properties are near-logarithmic in the security parameter, formally $O\left(\log^{1+\alpha} \lambda\right)$ for any arbitrarily small $\alpha > 0$.

Correctness (availability). Let the *drop rate* (γ) denote the fraction of message packets that adversarial parties fail to send as specified by the protocol. Let the *availability rate* (η) denote the fraction of messages from honest senders that are successfully delivered to their intended recipients. A communication protocol Π is said to be (η, γ)-*correct* against an adversary class $\mathbb{A}$ if it guarantees, with overwhelming probability, an availability rate of at least η for every input σ and every $\mathcal{A} \in \mathbb{A}$ whose drop rate is at most γ.

Remark. The special case $(\eta, \gamma) = (1, 0)$ corresponds to the standard notion of correctness in the passive adversary setting, where every honest recipient receives exactly the set of messages intended for it in σ.

6.3 A Differentially Private Protocol

Here and in the next subsection, we present new upper bounds for anonymity against active adversaries. Before doing so, however, we first observe that the onion shuffling protocol already achieves anonymity with only I/O-pattern leakage in the weaker passive adversary model.

The intuition is straightforward: in the synchronous communication model with a passive adversary, packets are never dropped, whether due to network failures or adversarial behavior. Consequently, once the numbers of messages sent and received by each participant are fixed, onion indistinguishability suffices to guarantee anonymity modulo the I/O-pattern leakage. This intuition can be formalized through a reduction argument.

However, the onion shuffling protocol no longer guarantees anonymity in the presence of adversarial packet drops. For example, suppose an honest participant i sends an onion that is routed through a corrupted party in the first round (an event that occurs with non-negligible probability). The adversary may simply drop this onion and later observe which participant fails to receive a message in the final round, thereby learning the recipient of i's message with non-negligible probability. Thus onion indistinguishability alone is insufficient in the active setting.

Nonetheless, we show that provable anonymity against active adversaries remains achievable in the expander setting.

Theorem 2. *Let G be an r-regular β-expander graph on n relays, where $r > 2$ and $\beta < 1$ are constants independent of the security parameter λ. There exists a communication protocol over G that simultaneously*

satisfies the following properties:

1. ***Anonymity.*** *For any arbitrarily small constant $\epsilon > 0$, the protocol achieves $(\epsilon, \mathsf{negl}(\lambda))$-*differential anonymity *against a non-adaptive active adversary corrupting up to any constant fraction $\chi \in [0, 1)$ of parties.*
2. ***Efficiency.*** *The round complexity L and onion cost x (i.e., the expected number of onions processed per message per round) are polylogarithmic in the security parameter λ:*

$$L = O(\log^a \lambda), \qquad x = O(\log^b \lambda),$$

   for constants $a, b > 1$ that may be chosen arbitrarily close to 1.
3. ***Correctness.*** *For any target delivery rate $\eta \in (0, 1)$, the protocol can be configured to guarantee (η, γ)-correctness for any constant drop rate $\gamma < \eta$.*

Proof of Theorem 2. We adapt the onion routing protocol Π_a of [ALU18] to operate over an r-regular β-expander relay network, and then prove that the resulting construction satisfies the required *efficiency*, *correctness*, and *anonymity* guarantees as stated above.

An onion encryption scheme consists of three algorithms: Gen generates key pairs for parties; FormOnion takes as input a message, a routing path, and metadata for each layer (e.g., address of the next hop), and outputs a layered ciphertext; and ProcOnion allows a relay to peel exactly one layer of the encryption to obtain that layer's metadata and the resulting peeled onion.

In Π_a's construction, we require each layer's metadata additionally include a unique nonce. To do this, we extend FormOnion so that it also takes as input a list of nonces $y = (y_1, \ldots, y_L)$, one per hop. The value y_t is embedded in layer t of the onion so that the relay at hop t recovers it when executing ProcOnion.

The protocol proceeds in two phases: an *onion forming phase* and a *routing phase*, described in detail below.

Onion forming phase. Each participant $i \in \mathcal{P}$ forms two types of onions. First, for each message-recipient pair in its input, i forms a *message-bearing onion* carrying the corresponding payload. Second, i forms a random number of *checkpoint onions*: cryptographically valid onions containing a dummy message $\perp$ at the innermost layer, terminating at independently sampled receivers, and embedding checkpoint markers (nonces) at pseudorandomly determined intermediate locations.

Let $L = \log^a \lambda$ be the length of each onion's routing path, for any constant $a > 1$.

Message-bearing onion. For each $(m, j) \in \sigma_i$ (specifying that participant i intends to send a message m to receiver j), the sender i samples a random walk $(T_1, \ldots, T_L, R)$ on the relay graph $G = (\mathcal{R}, \mathcal{E})$: T_1 is chosen uniformly from $\mathcal{R}$, and each subsequent hop is chosen uniformly from the neighbors of the previous one. The message-bearing onion is then formed by running

$$\textsc{FormOnion}\Big(m,\ (T_1, \ldots, T_L, j),\ (pk_{T_1}, \ldots, pk_{T_L}),\ (\perp, \ldots, \perp)\Big),$$

where $(\perp, \ldots, \perp)$ is a list of L empty nonces, since message-bearing onions carry no checkpoint markers.

Checkpoint onions. In addition to the message-bearing onion(s), sender i generates a pseudorandom number of *checkpoint onions*. Each checkpoint onion is "verifiable" by some intermediary relay u at a designated round $t = L/2$: if u receives the onion at round t, peeling the layer at that round reveals a pseudorandom nonce allowing u to verify that the onion has not been dropped earlier in the execution. This mechanism is realized using a PRF F as follows:

For each relay $u \in \mathcal{R}$:

1. Compute $b \leftarrow F(sk_{i,k}, 0)$, where $sk_{i,u}$ is the symmetric key shared between i and u.
2. If $b \equiv 1$ (an event occurring with frequency $\frac{\log^b \lambda}{N}$ for some constant $b > 1$) then proceed as follows:
   - i: Sample a random walk $T_1 = (T_{1,1}, \ldots, T_{1,\frac{L}{2}-1})$, where $T_{1,1}$ is chosen uniformly at random from the neighbors of u.
   - ii: Sample another independent random walk $T_2 = (T_{2,1}, \ldots, T_{2,\frac{L}{2}-1})$, where $T_{2,1}$ is also chosen uniformly from the neighbors of u.
   - iii: Reverse the order of nodes in T_1 to obtain T_1^{rev}.
   - iv: Choose a receiver k uniformly at random from the set of N participants.
   - v: Construct the path $(T_1^{\text{rev}}, u, T_2, k)$, which has length $L + 1$, with u appearing at position $\frac{L}{2}$ and k as the terminal node.
   - vi: Create a list of nonces $y = (y_1, \ldots, y_L)$, where $y_{\frac{L}{2}} = (\mathsf{checkpt}, F(sk_{i,u}, 1))$ and all other entries are set to $\perp$.
   - vii: Form a checkpoint onion using message $\perp$, the routing path T, the corresponding public keys, and the nonce list y.

Routing phase. All onions are released simultaneously at the start of round 1. In each subsequent round, every relay processes all onions it receives during that round, peels exactly one layer from each, and forwards the resulting onions to their next hops in uniformly random order (unless the protocol has been aborted).

Consider a checkpoint onion created by sender i with a nonce $y_{L/2}$ embedded in its $L/2$-th layer, and let u be the honest relay designated to receive this onion at round $L/2$. By construction, u should also receive a matching "symmetric" checkpoint onion from i that embeds the same nonce $y_{L/2}$ at that round. Upon peeling all onions it receives in round $L/2$, relay u checks how many expected nonces are missing. If more than a threshold τ are missing, u aborts; otherwise, it continues to forward all peeled onions to their next hops in uniformly random order starting in round $L/2 + 1$.

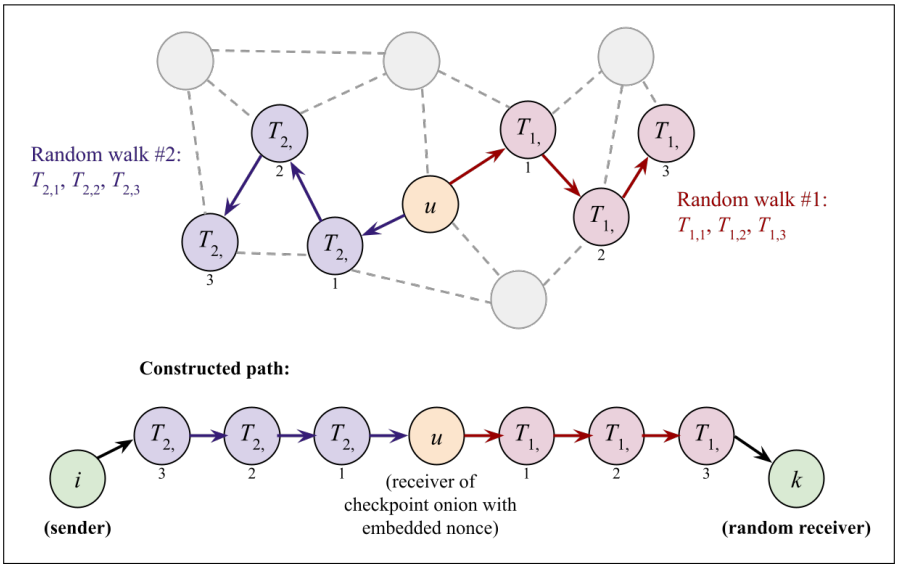


Figure 4: *Example construction of routing path for checkpoint onion intended to be received by relay u at round* 4.

Proof of correctness. For any target delivery rate $\eta \in (0, 1)$ and any constant drop rate $\gamma < \eta$, we can configure the abort threshold τ so that $\gamma x < \tau < \eta x$, where $x = \Theta(\log^b \lambda)$ is the server load. With this choice, we can show by a Chernoff bound that if the adversary drops fewer than a γ-fraction of onions, every honest

relay still receives at least τ checkpoints with overwhelming probability, and therefore the protocol does not abort. Moreover, conditioned on no abort, at least a η-fraction of checkpoint onions survive the first half of the execution, which in turn implies (from the security guaranteed by Lemma 3) that at least an η-fraction of honest parties' message-bearing onions are successfully delivered with overwhelming probability. Thus the protocol achieves (η, γ)-correctness.

Proof of differential anonymity. To prove that the protocol is secure, we require that the thresholding mechanism does its job:

Lemma 3. *Π_a (adapted to operate over a sparse expander graph) achieves $(\epsilon, \mathsf{negl}(\lambda))$-*differential anonymity*, for any arbitrarily small constant $\epsilon > 0$, against an active adversary that non-adaptively corrupts any constant fraction $\chi \in [0, 1)$ of parties.*

Proof of Lemma 3. Let σ, σ' be any neighboring input vectors. That is, σ and σ' are identical except on the inputs of two honest senders i and k and the "outputs" of two receivers j and l (the set of messages that j and l are slated to receive). Without loss of generality, assume that on input vector σ, i sends a message to j, and k sends to l; while in σ', this is swapped (i.e., i sends to l, while k sends to j).

We prove the theorem by case analysis.

Case 1: neither i's message nor k's message is delivered. The adversarial views for the two settings are perfectly indistinguishable since the adversary never observes the onions' layers for j and l, i.e., $\mathsf{View}_{\Pi_a,\mathcal{A}}(\sigma) = \mathsf{View}_{\Pi_a,\mathcal{A}}(\sigma')$.

Case 2: both i's message and k's message are delivered. In this case, $\mathsf{View}_{\Pi_a,\mathcal{A}}(\sigma)$ is statistically indistinguishable from $\mathsf{View}_{\Pi_a,\mathcal{A}}(\sigma')$, by Theorem 1.

Case 3: either i's or k's message is delivered, and the other is dropped. In this case, we will prove that $\mathsf{View}_{\Pi_a,\mathcal{A}}(\sigma)$ and $\mathsf{View}_{\Pi_a,\mathcal{A}}(\sigma')$ are differentially private. W.l.o.g., we assume that i's message makes it to its recipient, but k's does not. The only information the adversary has to help determine the input setting is the volumes of onions received by each recipient, j and l. Let X be the total number of checkpoint onions received by both j and l. As shown in [ALU18, Lemma 12], with $\mathbb{E}[X] = \Theta(\log^b \lambda)$ for any constant $b > 1$, Chernoff bounds imply that $X = \Theta(\log^b \lambda)$ except with negligible probability. Thus the number of (indistinguishable) onions going to j and l, denoted X_j and X_l, satisfies

$$X_j, X_l \sim \text{Binomial}(X, 1/2).$$

The true difference between the neighboring executions manifests only as a ± 1 shift (hidden inside binomial noise), and shown [DKM+06] to be $(\epsilon/2, \text{negl}(\lambda))$-differentially private for any positive constant ϵ. It follows from the composition theorem for differential anonymity that Π_a achieves $(\epsilon, \text{negl}(\lambda))$-differential anonymity for any positive constant ϵ. □

As shown above, the constructed protocol satisfies the three properties (correctness, efficiency, and anonymity) as required. □

6.4 A Protocol Leaking only I/O Pattern

Here, we present another positive result for anonymity in the active adversary setting. As is standard in the anonymity literature [ALU21, ACG+25], our notion of anonymity permits leakage of the I/O pattern. At present, no known techniques overcome this restriction, since controlling the number of onions received

by each participant appears difficult without requiring participants to reveal information about their secret inputs, namely, their intended recipients.

Note that the anonymity guarantees of the two protocols are not directly comparable. On the one hand, anonymity without any avoidable leakage is strictly stronger than differential anonymity. On the other hand, our full anonymity notion allows a limited and standard form of leakage, namely the I/O pattern.

Theorem 3. *Let G be an r-regular β-expander graph on n relays, where $r > 2$ and $\beta < 1$ are constants independent of the security parameter λ. There exists a communication protocol over G that simultaneously satisfies the following properties:*

1. ***Anonymity.*** *The protocol achieves* anonymity *(leaking only the I/O pattern) against an active adversary that non-adaptively corrupts any constant fraction $\chi \in [0, 1)$ of parties.*
2. ***Efficiency.*** *The round complexity L and onion cost x (i.e., the expected number of onions processed per message per round) are polylogarithmic in the security parameter λ:*

$$L = \Theta(\log^a \lambda \cdot \operatorname{loglog} \lambda), \qquad x = \Theta(\log^{2a} \lambda),$$

   for constant $a > 1$.
3. ***Correctness.*** *For any target delivery rate $\eta \in (0, 1)$, the protocol can be configured to guarantee (η, γ)-correctness for any constant drop rate $\gamma < \frac{1}{3}$.*

Proof of Theorem 3. We adapt $\Pi_\bullet$, pronounced "Pi-SEED" and short for *Protocol for the Synchronous Environment with Efficient Diagnostics*, from [ACG+25] to the sparse-expander setting. The original protocol is analyzed in the complete network model. Our adaptation replaces each uniformly random relay choice with a random-walk step on the r-regular β-expander graph G. The remaining diagnostic mechanisms are unchanged; their purpose is to prevent an active adversary from using selective dropping to create sender-dependent delivery patterns.

Epoch structure and diagnostics. $\Pi_\bullet$ consists of an initial onion forming phase, followed by a routing phase. $\Pi_\bullet$ divides the routing phase into $h = \Theta(\operatorname{loglog} \lambda)$ epochs of $T = \Theta(\log^a \lambda)$ rounds each, where $a > 1$ is any constant strictly larger than 1. So, the total number of rounds is $L = \Theta(\log^a \lambda \cdot \operatorname{loglog} \lambda)$.

Onion forming phase. During the onion forming phase, each party i forms three sets of onions: $y = \Theta\left(\log^{2a} \lambda \log\log \lambda\right)$ *checkpoint onions*, $\ell = \Theta(\log^a \lambda)$ loopback onions, and $z = \Theta\left(\log^{2a} \lambda\right)$ merging onions per message-recipient pair in σ_i.

Checkpoint onions. Checkpoint onions are exchanged between two parties i and k in round t depending on the output of a pseudorandom function F. As we want each party to be expecting y checkpoint onions over all the epochs, F needs to be chosen so that the probability that i forms a checkpoint onion for a random pair (k, t) is $\frac{y}{n \cdot hT}$. As i knows how many checkpoint onions to expect in a given round, it can determine if more than the fraction ρ_y is missing and signal to abort. The routing path of checkpoint onions are determined by stitching together two independent random walks, each originating at the intended receiver.

Loopback onions. In addition to checkpoint onions, each sender creates a polylogarithmic number of *loopback* onions that return to the sender after the first epoch. If more than a fraction ρ_x of loop back onions fail to return, the sender broadcasts an abort message. The routing path of each loop back onion is obtained by performing a random walk of length $T - 1$ before appending one's own address at the end.

Merging onions. Each message is replicated into a number of *merging onions* whose routing paths are arranged in a binary tree: the paths are disjoint until a merge point. If at least one path survives to the

merge point, the message is delivered exactly once (ensuring that the recipient receives the same number of messages regardless of which onions are dropped). The routing paths for each set of merging onions are constructed by first choosing a uniformly random merge endpoint. Starting from this endpoint, we generate a random-walk path backwards through the epochs, and at each epoch we branch the walk into two independent random-walk prefixes that continue the same process in the next epoch, recursively branching and eventually producing a depth-h binary tree of random-walk routes that (when reversed) appear to start out at random locations and repeatedly converge back to the same merge point over the course of h epochs.

Routing phase. All onions are released simultaneously at the start of the routing phase. In every round, each relay batch-processes all onions it receives, peels one encryption layer from each (unless it aborts), and forwards the resulting onions in uniformly random order to their next-hop addresses. Throughout execution, each honest party locally monitors the expected checkpoint and loopback onion counts: if in any epoch it observes more than a ρ_y-fraction of checkpoint onions missing, or if a party fails to receive at least a $(1 - \rho_x)$-fraction of its own loopback onions by the end of the first epoch, it broadcasts an abort message.

Proof of correctness. For any target delivery rate $\eta \in (0, 1)$ and any constant drop rate $\gamma < \eta$, we configure the checkpoint and loopback abort thresholds (ρ_x and ρ_y, respectively) so that $\gamma < \rho_x, \rho_y < \eta$. Each sender forms $\Theta(\log^a \lambda)$ loopback onions and each relay expects $\Theta(\log^{2a} \lambda)$ checkpoint onions per epoch. By a Chernoff bound, if the adversary drops fewer than a γ-fraction of onions, then with overwhelming probability every honest party receives at least $(1 - \rho_x)$ of its loopback onions and at least $(1 - \rho_y)$ of its checkpoint onions, and therefore the protocol does not abort.

Conditioned on no abort, at least an η-fraction of each honest sender's merging-onion copies survive to their merge points, and since each message is delivered if any one of its merging onions reaches the merge point, it follows that at least an η-fraction of honest parties' messages are delivered exactly once with overwhelming probability. Thus the protocol achieves (η, γ)-*correctness*.

Proofs of efficiency and anonymity. To prove that the protocol is secure for $L = h \cdot T = \Theta(\log^a \lambda \cdot \log\log \lambda)$ synchronous rounds and with an onion cost (i.e., number of onions expected per message per round) $\Theta(\log^{2a} \lambda \cdot \log\log \lambda)$, we require that the thresholding mechanisms do its job and that the appearance of each merging onion does not reveal any information about the sender set:

Lemma 4. *For any constant $a > 1$, with parameters $T = \ell = \sqrt{z} = \Omega(\log^a \lambda)$, and $h = \log(z) = \Theta(\log\log \lambda)$, the protocol is anonymous against active adversaries that corrupt up to a constant fraction $\chi < 1$ of the parties.*

Below, we sketch a security proof, which follows the same structure as the full analysis (for the complete-graph setting) that appears in [ACG+25, Theorem 2].

Proof sketch. We consider two cases.

Case 1: protocol aborts by the midpoint ($h/2$). In this case, anonymity is trivially preserved because no honest message is ever delivered and so no recipient learns anything about the sender set.

Case 2: protocol does not abort by the midpoint. Because each message-bearing onion is replicated along a binary tree of length z, the probability that all descendants are dropped in the first two epochs is negligibly small. A union bound then implies that with overwhelming probability, every message has at least one surviving descendant at the midpoint. After this point, since the adversary drops no more onions, the number of surviving onions for each sender remains balanced across senders for all subsequent epochs. Since all

message-bearing onions shuffle a polylogarithmic number of times, it follows from Theorem 1 that the adversary's view at delivery time is statistically indistinguishable across all sender inputs. □

Therefore the constructed protocol satisfies the three properties (correctness, efficiency, and anonymity) as required. □

7 Conclusion & Future Directions

We have established that random walk-based onion routing over sparse, constant-degree networks achieves anonymity against a passive adversary corrupting up to any constant fraction of nodes, with no asymptotic increase in efficiency parameters (any super-logarithmic round complexity and server load) compared to the fully connected network topology. Our analysis introduces two novel conceptual tools for reasoning about the passive adversary's view: (i) a *time-layered state space* that tracks flows between honest relays through corrupted channels, and (ii) a *time-inhomogeneous diffusion process* that captures how onions progressively lose correlations with their original senders. Together, these yield our main result: all onions become indistinguishable, even with a sparse network where up to any constant fraction (< 1) of the relays are passively corrupted and with no asymptotic increase in round complexity or server load compared to the complete network setting.

Technical challenges and how we overcame them. Because routing is constrained to local neighborhoods, the set of other onions an onion o can *next* shuffle with depends on the sets of onions o *recently* shuffled with, breaking the exchangeability assumption that underlies the simple contraction argument for the complete network setting. We address this challenge by reframing what the passive adversary sees during protocol execution: the time-layered state space allows us to recover uniformity in expectation of the L-step transition matrix of the adversary's belief, while the diffusion process allows us to bound how far the distributions deviate from uniform. This combination recovers the contraction effect necessary to prove anonymity.

Implications. Our result shows that anonymity is not solely an artifact of dense global connectivity (as in the complete graph setting) or even local symmetry (as in the network adversary model). Thus sufficient global mixing of onion identities emerges even in sparse, partially corrupted networks. This makes random walk-based onion routing both practical and provably secure in sparse, untrusted decentralized environments. More broadly, our two-model framework provides a new analytical lens for proving anonymity in settings where classical exchangeability-based assumptions (such as those of a fully connected mixnet) no longer hold.

By generalizing the [ALU18] result, which has served as an important primitive for analyzing more sophisticated random-walk–based onion routing protocols in the fully connected setting, we provide the analogous core component in the extension of these protocols to sparse expander networks. In particular, random walk-based onion routing protocols such as Π_a and $\Pi_\bullet$ (see Section 6) can be immediately extended to sparse networks by plugging in our main result in their existing anonymity analysis against an active adversary.

7.1 Future Work

The demand for anonymous communication is significant, as evidenced by the widespread use of existing networks aimed at anonymity like Tor [DMS04], which millions use daily,[9] as well as Freenet [CSWH01] and Crowds [RR98]. However, while these solutions are practical, it is well known that the strategies underlying these networks often compromise anonymity as they are susceptible to attacks such as traffic

[9] Data retrieved from the Tor Project.

correlation and timing analysis [MD05, SEV+15, WSJ+18].

In the last decade, numerous contributions have been made to the development of anonymous channels that are both practical and provable. These include definitions [CL05, BKM+13, ALU18, KBS+19, KBS20, ALU21, AL21, ACLM22]; protocols that offer provable guarantees against adversaries controlling a constant fraction $< \frac{1}{2}$ of the parties [vdHLZZ15, TGL+17, KCDF17, KLD20] with fault tolerance [ALU18, ALU21] and, additionally, in the presence of network churn [ACLM22] or in the asynchronous communication model [ACLM22, ACG+25]; and matching lower bounds on the round complexity [DMMK18, Chr20].

However, the positive results are based on much simpler settings than those found in reality, and significant work remains to bridge the gap between theory and practical application of these techniques. For example, they typically apply to *single-run* (the communication protocol runs once, allowing participants to send and receive one fixed-length message), *unidirectional* (Alice sends to Bob without allowing Bob to respond to Alice) communication over a *static, complete graph* network topology (simplifying networking issues).

Churn and locality. An interesting open problem is to design a way for clients to establish verifiable, anonymous random walks using only local neighborhood information (i.e., without requiring global knowledge of the network topology). Such a mechanism would be especially valuable for onion routing in sparse dynamic networks where relays may join and leave frequently and thus keeping all participants up-to-date with a global view of relay network is especially costly.

Beyond expanders. Another direction for future work is to extend our analysis beyond expander graphs to other sparse or structured classes of graphs, potentially revealing a more precise relationship between network expansion and anonymity. Understanding what efficiency parameters are possible for a given level of expansion would enable principled system design and provide insight into the minimal topological properties required to support efficient, anonymous, and robust communication.

Acknowledgments

This work was supported by NSF grants CCF-2312243 and CCF-2312241.

References

[ACG+25] M. Ando, M. Christ, K. Gupta, T. Malkin, and D. Smith. Full anonymity in the asynchronous setting from peony onion encryption. Cryptology ePrint Archive, Paper 2025/1067, 2025.

[ACLM22] M. Ando, M. Christ, A. Lysyanskaya, and T. Malkin. Poly onions: Achieving anonymity in the presence of churn. In *TCC 2022, Part II*, *LNCS* 13748. Springer, Heidelberg, November 2022.

[AL21] M. Ando and A. Lysyanskaya. Cryptographic shallots: A formal treatment of repliable onion encryption. In *TCC 2021, Part III*, *LNCS* 13044. Springer, Heidelberg, November 2021.

[ALU18] M. Ando, A. Lysyanskaya, and E. Upfal. Practical and provably secure onion routing. In *ICALP 2018*, *LIPIcs* 107. Schloss Dagstuhl, July 2018.

[ALU21] M. Ando, A. Lysyanskaya, and E. Upfal. On the complexity of anonymous communication through public networks. In *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[APR+15] J. Augustine, G. Pandurangan, P. Robinson, S. Roche, and E. Upfal. Enabling robust and efficient distributed computation in dynamic peer-to-peer networks. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 2015.

[BKM+13] M. Backes, A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi. AnoA: A framework for analyzing anonymous communication protocols. In *CSF 2013 Computer Security Foundations Symposium*. IEEE Computer Society Press, 2013.

[Cal18] California State Legislature. California consumer privacy act of 2018, 2018. Cal. Civ. Code §§1798.100 et seq.

[Can01] R. Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *42nd FOCS*. IEEE Computer Society Press, October 2001.

[Cha81] D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.

[Chr20] M. Christ. New lower bounds on the complexity of provably anonymous onion routing. *Undergraduate honors thesis, Brown University, Providence, RI*, 2912, 2020.

[CL05] J. Camenisch and A. Lysyanskaya. A formal treatment of onion routing. In *CRYPTO 2005*, *LNCS* 3621. Springer, Heidelberg, August 2005.

[CSWH01] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *Designing privacy enhancing technologies: international workshop on design issues in anonymity and unobservability Berkeley, CA, USA, July 25–26, 2000 Proceedings*. Springer, 2001.

[CV14] A. Czumaj and B. Vöcking. Thorp shuffling, butterflies, and non-Markovian couplings. In *ICALP 2014, Part I*, *LNCS* 8572. Springer, Heidelberg, July 2014.

[DKM+06] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT 2006*, *LNCS* 4004. Springer, Heidelberg, May / June 2006.

[DMMK18] D. Das, S. Meiser, E. Mohammadi, and A. Kate. Anonymity trilemma: Strong anonymity, low bandwidth overhead, low latency - choose two. In *2018 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2018.

[DMS04] R. Dingledine, N. Mathewson, and P. F. Syverson. Tor: the second-generation onion router. In *Proceedings of the 13th USENIX Security Symposium, August 9-13, 2004, San Diego, CA, USA*, 2004.

[Eur16] European Parliament and Council of the European Union. Regulation (eu) 2016/679 of the european parliament and of the council, 2016.

[Gil98] D. Gillman. A chernoff bound for random walks on expander graphs. *SIAM Journal on Computing*, 27(4):1203–1220, 1998.

[GRS96] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding routing information. In *International workshop on information hiding*. Springer, 1996.

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[KBS+19] C. Kuhn, M. Beck, S. Schiffner, E. A. Jorswieck, and T. Strufe. On privacy notions in anonymous communication. *PoPETs*, 2019(2):105–125, April 2019.

[KBS20] C. Kuhn, M. Beck, and T. Strufe. Breaking and (partially) fixing provably secure onion routing. In *2020 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2020.

[KCDF17] A. Kwon, H. Corrigan-Gibbs, S. Devadas, and B. Ford. Atom: Horizontally scaling strong anonymity. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*. ACM, 2017.

[KLD20] A. Kwon, D. Lu, and S. Devadas. {XRD}: Scalable messaging system with cryptographic privacy. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*, 2020.

[MD05] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of tor. In *2005 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2005.

[MU05] M. Mitzenmacher and E. Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge university press, 2005.

[RR98] M. K. Reiter and A. D. Rubin. Crowds: Anonymity for web transactions. *ACM transactions on information and system security (TISSEC)*, 1(1):66–92, 1998.

[SEV+15] Y. Sun, A. Edmundson, L. Vanbever, O. Li, J. Rexford, M. Chiang, and P. Mittal. {RAPTOR}: Routing attacks on privacy in tor. In *24th USENIX Security Symposium (USENIX Security 15)*, 2015.

[SHPG12] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey. Jellyfish: Networking data centers randomly. In *9th USENIX Symposium on Networked Systems Design and Implementation (NSDI 12)*, 2012.

[TGL+17] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich. Stadium: A distributed metadata-private messaging system. In *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*. ACM, 2017.

[Uni48] United Nations General Assembly. Universal declaration of human rights, 1948. Article 12.

[vdHLZZ15] J. van den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich. Vuvuzela: scalable private messaging resistant to traffic analysis. In *Proceedings of the 25th Symposium on Operating Systems Principles, SOSP 2015, Monterey, CA, USA, October 4-7, 2015*. ACM, 2015.

[VSDS16] A. Valadarsky, G. Shahaf, M. Dinitz, and M. Schapira. Xpander: Towards optimal-performance datacenters. In *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*, 2016.

[WB90] S. D. Warren and L. D. Brandeis. The right to privacy. *Harv. L. Rev.*, 4:193, 1890.

[WSJ+18] R. Wails, Y. Sun, A. Johnson, M. Chiang, and P. Mittal. Tempest: Temporal dynamics in anonymity systems. *PoPETs*, 2018(3):22–42, July 2018.

A Supplementary Proofs

Lemma 5. *Let $C \subset \mathcal{R}$ be a subset of corrupted relays, and let $\chi = |C|/n$ be a constant strictly less than* 1. *Let*

$$L = g(\lambda) \log \lambda$$

for some function $g(\lambda)$ satisfying $g(\lambda) \to \infty$. Then, with overwhelming probability in λ, a random walk of length L on G, starting from a uniformly random relay, visits honest relays $\mathcal{R} \setminus C$ for at least a nonzero constant fraction of its steps.

Proof for Lemma 5. Let c be the number of visits to C after L steps of the walk, where in expectation, $\mathbb{E}[c] = \chi \cdot t$.

[Gil98, Theorem 2.1] states that for every $\gamma \geq 0$:

$$\Pr\left[c - \mathbb{E}[c] \geq \gamma\right] \;\leq\; N_q \left(1 + \frac{\gamma(1-\beta)}{10L}\right) \exp\left(-\frac{\gamma^2(1-\beta)}{20L}\right).$$

where $N_q = 1$ accounts for the starting position being uniformly random.

Let $\gamma = \frac{L \cdot (1-\chi)}{2}$. Then $\frac{\gamma^2(1-\beta)}{20L}$ grows asymptotically faster than $\log \lambda$, hence the probability is negligible, and thus with overwhelming probability in λ

$$c \;<\; L\left(\frac{1+\chi}{2}\right).$$

Therefore the fraction of visits *not* in C is at least $\frac{1-\chi}{2}$, which is a constant fraction strictly greater than 0. □

Lemma 6. *Let X and Y be independent random variables, and suppose $\mathbb{E}[Y] = 0$. Then:*

$$\mathrm{Var}\left[XY\right] = \mathbb{E}\left[X^2\right] \cdot \mathrm{Var}\left[Y\right]$$

Proof of Lemma 6. Since X and Y are independent, by [MU05, Thm. 3.3], we have:

$$\mathbb{E}[XY] = \mathbb{E}[X] \cdot \mathbb{E}[Y] \quad \text{and} \quad \mathbb{E}\left[(XY)^2\right] = \mathbb{E}\left[X^2\right] \cdot \mathbb{E}\left[Y^2\right].$$

Substituting into the definition of variance:

$$\begin{aligned}\mathrm{Var}\left[XY\right] &= \mathbb{E}\left[(XY)^2\right] - \mathbb{E}[XY]^2 \\ &= \mathbb{E}\left[X^2\right] \cdot \mathbb{E}\left[Y^2\right] - (\mathbb{E}[X] \cdot \mathbb{E}[Y])^2\end{aligned}$$

Now suppose Y is zero-mean. Then:

$$\mathrm{Var}\left[XY\right] = \mathbb{E}\left[X^2\right] \cdot \mathbb{E}\left[Y^2\right] = \mathbb{E}\left[X^2\right] \cdot \mathrm{Var}\left[Y\right].$$

□

Lemma 7. *Let X be a random variable with expected value $\mathbb{E}[X] = \Omega(f(\lambda))$, where f denotes any function of λ that grows strictly faster than $\log^{\frac{1}{3}} \lambda$. Then with overwhelming probability w.r.t. λ,*

$$X = \Omega(f(\lambda)).$$

Proof of Lemma 7. From Chernoff bounds [MU05, Cor. 4.6] for all $0 < \delta < 1$:

$$\Pr\left(|X - \mathbb{E}[X]| \geq \delta \cdot \mathbb{E}[X]\right) \leq 2\exp(-\frac{\delta^2 \cdot \mathbb{E}[X]}{3}).$$

Solve for δ, requiring a negligible probability:

$$\delta = \sqrt{\frac{\Omega(\log^c \lambda)}{\mathbb{E}[X]}}, \quad \text{for any } c > 1$$

Substituting $\mathbb{E}[X] = \Omega(\log^\epsilon \lambda)$, for any $\epsilon > \frac{1}{3}$:

$$\delta = \Omega(\log^{\frac{c-\epsilon}{2}} \lambda)$$

Thus, with overwhelming probability (w.r.t. λ), we have:

$$\begin{aligned} X &> \Omega(\log^\epsilon \lambda - \log^{\tfrac{c-\epsilon}{2}} \lambda) \\ &= \Omega(\log^\epsilon \lambda), && \text{in particular for: } 1 < c < 3\epsilon \\ &= \Omega(f(\lambda)) \end{aligned}$$

□

Lemma 8. *Let X be an event that occurs with overwhelming probability with respect to λ. After N trials, if $N = O(\mathsf{poly}(\lambda))$, then with overwhelming probability in λ, X occurs in every trial.*

Proof of Lemma 8. We can write $N < \lambda^c$, for some $c > 1$. Taking the union bound over all N trials, the probability that X fails in at least one trial is bounded by:

$$\begin{aligned} \Pr[X \text{ fails in some trial}] &< N\,\lambda^{-\epsilon}, \quad \text{for every } \epsilon > 1. \\ &< \lambda^{c-\epsilon} \end{aligned}$$

Even though c is given, ϵ can be arbitrarily large. It follows that for all $k > 1$, there exists ϵ that satisfies $c - \epsilon = -k$. Thus

$$\begin{aligned} \Pr[X \text{ fails in some trial}] &< \lambda^{-k}, && \text{for every } k > 1 \\ &= \mathsf{negl}(\lambda) \end{aligned}$$

Therefore with overwhelming probability, X occurs in every trial. □

Lemma 9. *Let $b > 1$ and r be constants. Let $X_1, X_2, \ldots, X_r$ be independent random variables, each with expectation*

$$\mu = g(\lambda) \log \lambda$$

for some function $g(\lambda)$ satisfying $g(\lambda) \to \infty$. Then, for any arbitrarily small constant $\delta > 0$, the following holds with overwhelming probability in λ:

$$\forall i \in [r], \quad \frac{X_i}{\sum_{j=1}^{r} X_j} \in \left[\frac{1-\delta}{r}, \frac{1+\delta}{r}\right].$$

Proof of Lemma 9. Fix $i \in [r]$. From Chernoff bounds [MU05, Cor. 4.6], for any $0 < \delta < 1$,

$$\Pr\left[|X_i - \mu| \geq \delta\mu\right] \leq 2\exp\left(\frac{-\delta^2 \cdot \mu}{3}\right).$$

Applying a union bound over all $X_1, X_2, \ldots, X_r$ gives:

$$\Pr\left[\exists i \in [r] : |X_i - \mu| \geq \delta\mu\right] \leq 2r \cdot \exp\left(\frac{-\delta^2 \cdot \mu}{3}\right),$$

Since $\mu = g(\lambda)\log\lambda$ and $g(\lambda) \to \infty$, the exponent grows asymptotically faster than $\log\lambda$. Hence this

probability is negligible in λ for any arbitrarily small constant $\delta > 0$. Thus with overwhelming probability,

$$\forall i \in [r], \quad X_i \in [(1-\delta)\mu, (1+\delta)\mu], \quad \text{and} \quad \sum_{j=1}^{r} X_j \in [r(1-\delta)\mu, r(1+\delta)\mu].$$

Dividing, we obtain for each $i \in [r]$:

$$\frac{X_i}{\sum_{j=1}^{r} X_j} \in \left[\frac{1-\delta}{r(1+\delta)}, \frac{1+\delta}{r(1-\delta)}\right].$$

Define $\delta' = \frac{2\delta}{1-\delta}$, so that:

$$\frac{1-\delta}{1+\delta} = 1-\delta', \quad \text{and} \quad \frac{1+\delta}{1-\delta} = 1+\delta'.$$

Thus:

$$\frac{X_i}{\sum_{j=1}^{r} X_j} \in \left[\frac{1-\delta'}{r}, \frac{1+\delta'}{r}\right].$$

Since δ can be taken arbitrarily small, so can δ',. □