arXiv:1902.07138v1 [cs.DC] 19 Feb 2019

Who started this rumor? Quantifying the natural differential privacy guarantees of gossip protocols

AURÉLIEN BELLET, INRIA Lille, France
RACHID GUERRAOUI, EPFL, Switzerland
HADRIEN HENDRIKX, MSR - INRIA Joint Centre, France

Abstract: Gossip protocols, also called rumor spreading or epidemic protocols, are widely used to disseminate information in massive peer-to-peer networks. These protocols are often claimed to guarantee privacy because of the uncertainty they introduce on the node that started the dissemination. But is that claim really true? Can one indeed start a gossip and safely hide in the crowd? This paper is the first to study gossip protocols using a rigorous mathematical framework based on differential privacy to determine the extent to which the source of a gossip can be traceable. Considering the case of a complete graph in which a subset of the nodes are curious sensors, we derive matching lower and upper bounds on the differential privacy parameters. Crucially, our results show that gossip protocols can naturally guarantee some privacy without the need for additional perturbations, and reveal that asynchronous protocols provide a different and stronger type of privacy guarantees than their synchronous counterparts. Furthermore, while the optimal privacy guarantees are attained at the cost of a drastic reduction of the dissemination speed, we show that one can devise gossip protocols achieving both fast spreading time and near-optimal privacy.

1 INTRODUCTION

Peer-to-peer networks enable people to share information without the need for any central authority. Some of this information may be sensitive, and people sharing it may not want to be identified, for instance because of copyright infringement when sharing music, or in the case of whistle-blowers. Anonymous sharing platforms can also help people exercise their right to freedom of expression in totalitarian regimes. Conversely, it may be important to locate the source of a (computer or biological) virus, or fake news, spreading in a network. Therefore, it is crucial to understand the fundamental limits on privacy and anonymity in information dissemination. *Gossip* protocols (also called *rumor spreading* or *epidemic protocols*), in which nodes *randomly* choose a neighbor to exchange information, are both simple and efficient [1, 6, 22, 29, 37]. They can be used to spread and aggregate information in distributed databases [2, 7, 12, 30, 31] and social networks [13], as well as to optimize cost functions involving distributed datasets in machine learning [10, 14, 26, 38]. A folklore belief is that gossip protocols guarantee *source anonymity* because users cannot know who issued the information in the first place [23]. Although a lot of work has been devoted to assessing how efficiently one could locate the source of a gossip in specific settings [28, 36, 41], the general anonymity claim has never been studied from a pure *privacy* perspective, independently of any particular attack model. Intuitively indeed, random and local exchanges improve privacy, but to what extent? Given the importance of privacy and peer-to-peer information dissemination, it is crucial to study the limitations of this claim through a principled approach. This is the challenge we take up in this paper for the classic case of a complete network graph.

Our first contribution is an information-theoretic model of anonymity in gossip protocols based on an adaptation of (ϵ, δ)-*differential privacy* (DP) [15]. Originally introduced in the database community, DP is a precise mathematical framework recognized as the gold standard for studying the privacy guarantees of information release protocols. In our proposed model, the information to protect is the source of the gossip, while the attackers are a subset of curious nodes monitoring the communications they receive. We make an intuitive, yet crucial distinction between the *synchronous* and *asynchronous* settings: in the former, the attackers observe a global timestamp along with each communication they receive, while in the latter they only know the relative order between the communications. Our notion of DP then requires that the probability of any possible observation of the attackers is almost the same regardless of which node is the source. A key novel aspect of our model is that the mechanism that seeks to ensure DP comes only from the *natural* randomness and partial observability of gossip protocols, not from additional perturbation or noise as generally needed to guarantee DP [17]. We believe our adaptation of DP to be of independent interest. We also complement it with a notion of *prediction uncertainty* which guarantees that even unlikely events do not fully reveal the identity of the source under a uniform prior. This property gives an upper bound on the probability of success of any source prediction protocol, including the maximum likelihood estimate.

Based on our proposed model, we then establish *matching upper and lower bounds* on the privacy guarantees of gossip protocols. Essentially, our upper bounds on differential privacy are derived from the fact that (i) it is quite likely that the node starting the rumor discloses it to an attacker during the first rounds, whereas (ii) this is extremely unlikely to happen for a random node fixed in advance. Interestingly, our results highlight a fundamental qualitative difference in the privacy guarantees of the synchronous and asynchronous settings: unlike their synchronous counterparts, asynchronous gossip protocols can satisfy prediction uncertainty and fulfill a strict version of differential privacy. We show that the upper bounds on privacy are matched by a gossip protocol which has very slow spreading time (*linear* in the number of nodes), highlighting an interesting tension between privacy and *dissemination speed*.

To capture this trade-off between speed and privacy, we introduce a *parameterized* gossip protocol in which nodes have a fixed probability of forgetting the rumor after each communication. This gives the protocol the ability to forget initial conditions, thereby ensuring the privacy of the source. The standard "push" gossip protocol [37], as well as the optimally private but slow protocol we previously introduced, can both be derived from our parameterized scheme with specific choices of the parameter. We show that the standard gossip protocol is inherently not differentially private for arbitrarily large graphs, but that it is possible to devise gossip protocols that are *near-optimally private* with spreading time *logarithmic* in the size of the graph. We prove the protocol speed by analyzing the mean dynamics of gossip and leveraging concentration inequalities. The privacy results are obtained by showing that only a small fraction of the possible outcomes have different probabilities when two different nodes initially have the gossip. This requires to precisely evaluate the probability of well-chosen worst-case sequences, which is generally hard as randomness is involved both when nodes decide to stop sending messages as well as when they choose who to send messages to.

The rest of the paper is organized as follows. We discuss related work in Section 2. In Section 3, we formally introduce our model and privacy definitions. In Section 4, we give matching upper and lower bounds on the privacy guarantees of gossip protocols, and present a privacy-optimal but slow protocol. Section 5 studies how to control the trade-off between speed and privacy. Finally, we conclude in Section 6 by discussing open questions.

2 RELATED WORK

2.1 Gossiping

The idea of disseminating information in a distributed system by having each node *push* messages to a randomly chosen neighbor, initially coined the *random phone-call model*, dates back to even before the democratization of the internet [22, 37]. Such protocols, later called *gossip*, *epidemic* or *rumor spreading*, were for instance applied to ensure the consistency of a replicated database system [2, 12]. Indeed, gossip protocols scale very well with the size of the system while deterministic consistency becomes too expensive to ensure among the replicas. They have gained even more importance when argued to model information spreading in social networks [13]. Gossip protocols can also be used to compute aggregate queries on a database distributed across the nodes of a network [7, 30, 31], and have recently become popular in machine learning for optimizing cost functions involving distributed datasets [10, 14, 26, 38]. Gossip protocols differ according to their interaction schemes, i.e., *pull* or *push*, sometimes combining both [29]. In this work, we focus on the classical push form in the case of a *complete* graph. Gossip protocols have also been generalized for graphs of any given conductance [24, 40]. Differences between *synchronous* and *asynchronous* gossip protocols in general graphs have also been investigated in [1, 25]. The trade-off between spreading time and number of messages exchanged was studied in [6].

2.2 Locating the gossip source

Determining the source of a gossip has been an active research topic, especially given the potential applications to social networks (see [28] for a recent survey). Existing approaches have focused so far on building protocols to compute or approximate the maximum likelihood estimate of the source given some observed information. Each approach typically assumes a specific kind of graphs (e.g., trees, small world, etc.), dissemination model and observed information. In *rumor centrality* [41–43], the gossip communication graph is assumed to be fully observed and the goal is to determine the *center* of this graph in order to deduce the node that started the gossip. Another line of work studies the setting in which some nodes are *curious sensors* that inform a central entity whenever they receive a message [36]. Gossiping is assumed to happen at random times and the source node is estimated by comparing the different timings at which the information reaches the sensors. The specific attack considered in this work is very natural in trees but does not generalize to highly connected graphs in which all nodes are approximately at the same distance (in which other attacks may have a higher probability of success). The work of [21] focuses on the problem of hiding the source instead of locating it. The observed information is a snapshot of who has the rumor at a given time. A specific dissemination protocol is proposed to hide the source but the privacy guarantees they obtain only hold for tree graphs.

We stress the fact that the privacy guarantees that can be derived from the above work (i.e., the probability not to be detected) only hold under the specific attacks considered therein. Furthermore, all approaches rely on maximum likelihood and hence assume a uniform prior on the probability of each node to be the source. The guarantees would thus break in case the protocol was run twice from the same source, or if the attacker knew that some of the nodes could not have started the rumor.

2.3 Differential privacy

While we borrow ideas from the approaches mentioned above (e.g., we assume that a subset of nodes are curious sensors as in [36]), our work differs fundamentally for we aim at studying the fundamental privacy limits of any gossip source location protocol, independently of any specific attack, by evaluating the amount of information that is released during a gossip scheme. For this purpose, a general and robust notion of privacy is required. Many privacy definitions exist but *differential privacy* [15, 17] has emerged as a gold standard for it holds independently of any assumption on the model, the computational power, or the background knowledge that the attacker may have. Differentially private protocols have been proposed for numerous problems in the fields of databases, data mining and machine learning: examples include computing aggregate and linear counting queries [17, 33], releasing and estimating graph properties [11, 34], clustering [27] and recently deep learning [44].

In this work, we consider the classic relaxed version of differential privacy which involves two parameters $\epsilon, \delta \geq 0$ that quantify the privacy guarantee [16]. More precisely, given any two databases $\mathcal{D}_1$ and $\mathcal{D}_2$ that differ in at most one row (all the rows are the same except for one),[1] a (randomized) information release protocol $\mathcal{P}$, and the set $\mathcal{S}$ of all possible outputs of $\mathcal{P}$, protocol $\mathcal{P}$ is said to guarantee (ϵ, δ)-differential privacy if for any $S \subset \mathcal{S}$:

$$p(\mathcal{P}(\mathcal{D}_1) \in S) \leq e^{\epsilon} p(\mathcal{P}(\mathcal{D}_2) \in S) + \delta. \tag{1}$$

Parameter ϵ places a bound on the change in output distribution when changing one entry of the database, while parameter δ is assumed to be small and allows the bound to be violated with small probability. When $\delta = 0$, we recover the strict ϵ-differential privacy. The above privacy guarantees hold for any attack and are robust against strong background knowledge that the attacker may have about the records of the database (in particular, the attacker may know all records in $\mathcal{D}_1$ and $\mathcal{D}_2$ except the differing ones). In our context, the background information could be the knowledge that the source is among a subset of k nodes. Robustness against such background knowledge is crucial in some applications, for instance when sharing secret information that few people could have known and leaked in the first place. Another important feature of differential privacy is *composability*: if (ϵ, δ)-differential privacy holds for a release protocol, then querying this protocol two times about the same dataset satisfies $(2\epsilon, 2\delta)$-differential privacy. This is crucial in our context for it enables to quantify privacy when the source propagates multiple messages and the adversary is able to link them to the same source (e.g., due to the content of the message). This happens for instance when breaking big messages into multiple pieces, which is known to drastically improve spreading time for sharing large files [39].

Existing differentially private protocols typically introduce additional *perturbation* (also called *noise*) to hide critical information [17]. In contrast, an original aspect of our work is to solely rely on the *natural* randomness and limited observability brought by gossip protocols to guarantee differential privacy.

3 PROPOSED MODEL

Our first contribution is a precise mathematical framework for studying the fundamental privacy guarantees of gossip protocols. We define the family of protocols we consider, their inputs and the outputs observed by the attackers during the execution of a protocol, as well as the privacy notions we consider. In the following, we consider a complete graph with n nodes labeled from 0 to $n-1$.

3.1 Gossip protocols

To specify the class of protocols we consider in this paper, we first define a key communication primitive. Denoting by I the set of informed nodes, $\texttt{tell\_gossip}(i, I)$ allows an informed node $i \in I$ to tell the information to another node $j \in \{0, \ldots, n-1\}$ chosen uniformly at random. $\texttt{tell\_gossip}(i, I)$ returns j (the node that received the message) and the updated I (the new set of informed nodes that includes j). Equipped with this primitive, we can now define gossip protocols as follows.[2]

Definition 3.1 (Gossip protocols). A gossip protocol on a complete graph is one that (a) terminates, (b) ensures that at the end of its execution, the set of informed nodes $I = \{0, \ldots, n-1\}$, and (c) can modify I only through calls to the `tell_gossip` primitive.

[1] Strictly speaking, the above definition should be called (ϵ, δ)-*indistinguishability* [16], while *differential privacy* refers to the case where $\mathcal{D}_1$ is obtained by either adding or removing a row from $\mathcal{D}_2$. We thus reproduce the slight abuse of terminology commonly found in the literature.

[2] Definition 3.1 is stronger than the one of *address independent protocol* introduced in [29]: it enforces the communication protocol, thus enforcing address independence.

3.2 Inputs and outputs

As recalled in Section 2.3, differential privacy is a probabilistic notion that evaluates a protocol based on the variations of the *output* distribution for a change in the *input*. In this paper, we adapt it to our gossip context. We first formalize the *inputs* and *outputs*, in the case of a *single piece of information to disseminate* (multiple pieces can be addressed through composition, see Section 2.3). A single node has the information (the gossip, or rumor) at the beginning of the protocol. This node defines the input of the gossip protocol, and it is the actual "database" that we want to protect. In this sense, the source node is a database with n rows, each with a binary attribute which is 1 for the source node and 0 elsewhere.

We define the output of a gossip protocol as the information disclosed to some attackers during the execution of the protocol. In this work, we focus on attackers that can monitor a set of *curious nodes* C of size f, i.e. they observe all communications involving a curious node. More formally, a gossip protocol generates an ordered sequence S_{omni} of triplets (t, i, j) of executions of `tell_gossip` where t counts the number of times the `tell_gossip` primitive has been called, i is the node on which `tell_gossip` was used and j the node that was told the information. This sequence corresponds to the output that would be observed by someone who could eavesdrop on all communications. Through the (random) execution of the protocol, the attackers we consider gather a (random) subsequence $S \subset S_{\text{omni}}$ since it only monitors a subset of the nodes.

We define two settings depending on the timing assumption that we make. In the synchronous setting, the attackers have access to the global counter telling them how many times the `tell_gossip` primitive has been invoked by the protocol. In the weaker asynchronous setting, curious nodes only have access to the relative order in which the information has been disclosed to them.

Definition 3.2. In the synchronous setting, a gossip protocol outputs the sequence $S = ((t, i, j) \in S_{\text{omni}} | j \in C)$. In the *asynchronous* setting, it outputs $S = ((i, j) | (t, i, j) \in S_{\text{omni}}, j \in C)$.

This natural distinction reflects the influence of the time model used by the protocol. In the synchronous setting, there is a discrete global clock and we assume that nodes can concurrently perform a `tell_gossip` operation in one unit of time (round). This means that nodes know the global timestamp for the messages they receive, and can thus know how many rounds (and therefore have an estimate of how many communications) have happened in the network before they receive a given message. In contrast, time is continuous in the asynchronous setting: each node is equipped with an internal clock that ticks at the times of a rate 1 Poisson process [7]. Nodes can perform a `tell_gossip` operation at each of their internal clock ticks. Consequently, curious nodes know the relative order of communications they receive but not the global timestamps. Note that because we focus on complete graphs, knowing which curious node received the rumor gives no information on the starting node. For a given output sequence S, we therefore write $S_t = i$ to denote that `tell_gossip` has been used by node i at time t, when time is relative or absolute depending on the context.

The ratio f/n of curious nodes in the graph determines the probability of the attacker to gather information. Unless otherwise noted, we assume this ratio to be constant. In particular, we see it as a quantity independent of n, otherwise the attacker would only become weaker as the graph grows bigger.

3.3 Privacy definitions

Now that we have precisely defined the inputs and outputs of the "release protocols" that we consider, we can formally introduce privacy definitions for the gossip problem. To ease notations, we denote by I_0 the source of the gossip (the set of informed nodes at time 0), and for any given $i \in \{0, \ldots, n-1\}$, we denote by $p_i(E) = p(E | I_0 = 0)$ the probability of event E if node i is the source of the gossip. The release protocol is therefore abstracted in this notation. Recalling that $\mathcal{S}$ is the set of all possible outputs of the information release procedure, we say that a gossip protocol is (ϵ, δ)-differentially private if:

$$p_i(S) \le e^{\epsilon} p_j(S) + \delta, \quad \forall S \subset \mathcal{S}, \ \forall i, j \in \{0, \dots, n-1\}, \tag{2}$$

where $p(S)$ is the probability that the output belongs to the set S. This formalizes a notion of *source indistinguishability* in the sense that, with high probability, any output is almost as likely to be observed by the attackers regardless of who started the gossip ("nothing bad will happen with high probability"). Note however that when $\delta > 0$, this definition allows a protocol to release the identity of the source with small probability. To capture the fact that "nothing too bad will ever happen", we favor differentially private protocols that also guarantee the complementary notion of *prediction uncertainty.*

Definition 3.3 (Prediction uncertainty). A gossip protocol is said to guarantee c-prediction uncertainty if there exists a constant $c > 0$ such that for a uniform prior $p(I_0)$ on source nodes and any $i \in \{0, \dots, n-1\}$:

$$\frac{p(I_0 \neq \{i\} | S)}{p(I_0 = \{i\} | S)} \ge c, \quad \forall S \subset \mathcal{S}, \ p_i(S) > 0. \tag{3}$$

Prediction uncertainty guarantees that no observable output S can identify a node as the source with large enough probability, ensuring that the probability of success of any source prediction protocol is upper bounded by $1/(1+c)$. This holds in particular for the maximum likelihood estimate. Prediction uncertainty does not have the same robustness against background knowledge as differential privacy, as it assumes a uniform prior on the source. While it can be shown that $(\epsilon, 0)$-DP with $\epsilon > 0$ implies prediction uncertainty, the converse is not true. Indeed, prediction uncertainty is satisfied as soon as no output identifies any node with enough probability, without necessarily making all pairs of nodes indistinguishable as in DP. As we show later, prediction uncertainty pinpoints a key advantage of asynchronous gossip protocols. Thanks to the symmetry of our problem, we can consider without loss of generality that node 0 starts the rumor ($I_0 = \{0\}$) and verify Equation 2 and Equation 3 only for $i = 0$ and $j = 1$.

Remark 3.1 (Model extensions). *We have kept our model relatively simple to avoid unnecessary technicalities in the derivation and presentation of our results. For completeness, we discuss the impact of some possible extensions (e.g., information observed by attackers, malicious behavior, termination criterion) in Appendix A.*

4 OPTIMAL PRIVACY

In this section, we study the fundamental limits of gossip protocols in terms of privacy. Our main result is a set of tight bounds on the privacy guarantees that can be achieved by gossip protocols. We state and discuss these bounds in Section 4.1. We then present the optimally private gossip protocol that matches our bounds and discuss its properties in Section 4.2.

4.1 Main result: matching upper and lower bounds on privacy

We now state upper bounds on the differential privacy and prediction uncertainty that hold for any gossip protocol in the sense of Definition 3.1.

Theorem 4.1. *If a gossip protocol satisfies (ϵ, δ)-differential privacy and c-prediction uncertainty then we have $\delta \ge \frac{f}{n}$ and $c = 0$ in the synchronous setting, and $\delta \ge \frac{f}{n}\left(1 - \frac{e^{\epsilon}-1}{f}\right)$ and $c \le \frac{n}{f+1} - 1$ in the asynchronous setting. Furthermore, these bounds are tight and matched by Algorithm 1 when its parameter is set to $s = 0$.*

Sketch of proof. (The complete proof of the statement can be found in Appendix B). To prove the lower bounds on the parameter δ of differential privacy, we upper bound the probability that the first node that communicates with a curious node is the source of the rumor. Then, we lower bound the probability that another node fixed in advance communicates with a curious node. While the synchronous case is straightforward, the asynchronous setting is more subtle. We heavily rely on the observation that all nodes are equally likely to be the first to disclose information to curious nodes after the first message has been sent. Regarding prediction uncertainty, it is easy to show that $c = 0$ for synchronous gossip protocols by observing that outputs S such that $S_0 = 0$ (i.e., node 0 communicates with a curious node at time 0) have nonzero probability only if node 0 started the rumor. In contrast, in asynchronous protocols the event $S_0 = 0$ (i.e., node 0 is the first to communicate with a curious node) has nonzero probability even if 0 did not start the rumor. Then, we use the same set of sequences that we used to prove the lower bound on differential privacy to get the lower bound on prediction uncertainty. □

Theorem 4.1 shows that gossip protocols are able to *naturally* provide privacy guarantees. It also reveals a *fundamental qualitative difference* between the synchronous and asynchronous settings. For differential privacy in the synchronous setting, δ cannot be smaller than the proportion of curious nodes. This is rather intuitive since the source node is revealed with probability f/n (when the first message is sent to a curious node). In contrast, it is possible to break this limit in the asynchronous case, where one can achieve δ smaller than f/n by setting $\epsilon > 0$. One can even satisfy the *strict* version of differential privacy ($\delta = 0$) by setting $\epsilon \approx \log f$, which provides good privacy guarantees when the number of curious nodes is not too large. We see an even more striking difference in terms of prediction uncertainty, which is not satisfied by any protocol in the synchronous setting while strong guarantees are achievable in the asynchronous case. Combined with the differential privacy result, this means that the most private synchronous protocol reveals the identity of the source with probability at least f/n. On the other hand, even though the probability of disclosing *some* information is still of order f/n in the asynchronous setting, the attackers always have a high probability of making a mistake in their attempt to locate the source.

4.2 Optimally private protocol

It turns out that the bounds of Theorem 4.1 are matched by a very simple protocol: nodes forward the message to exactly one random neighbor when they receive it and then stop emitting until they receive the message again. This protocol, which we refer to as *private gossip*, corresponds to a special case of the more general protocol described in Algorithm 1 when its parameter is $s = 0$ (see discussion in Section 5). Private gossip is similar to the protocol introduced by [21] in the sense that at each time step, the source changes and it is quickly impossible to recover which node started the gossip (as initial conditions are quickly forgotten). Private gossip ensures that (i) the gossip does not die before all nodes are informed, (ii) the state of the system (the set informed nodes I) after round 0 is completely independent from the source node, and (iii) all nodes follow the same behavior. The first property ensures that the protocol falls within Definition 3.1, the second one is key to match the optimal privacy parameters stated in Theorem 4.1 (see Appendix B.2 for the proof), and the third one prevents the source to be identified based on its special behavior, as in the naive alternative described below.

Remark 4.1 (Delayed start protocol). *A naive alternative to the private gossip is as follows: the source node transmits the rumor to a random node and forgets it, then a standard gossip protocol (such as Algorithm 1 with $s = 1$) may start normally from the node that received the information. While this* delayed start *protocol guarantees optimal differential privacy in the synchronous setting, it is fundamentally flawed. In particular, even its asynchronous version does not guarantee prediction uncertainty in the sense that $c \longrightarrow 0$ as the size of the graph increases. This is because attackers can identify the source with high probability by detecting that it communicated only once and then stopped emitting for many rounds. We refer to Appendix D for details.*

An obvious drawback of the private gossip protocol is that it is very slow, since only one node sends the rumor at any given time. To precisely quantify its dissemination speed, one can observe that it performs a random walk on the complete graph. Therefore, the number of gossip operations needed to inform all

```
Algorithm 1 Fast Private Gossip
Require: $n$ {Number of nodes}, $k$ {Source node}, $s$
    {Probability for a node to remain active}
Ensure: $I = \{0, \dots, n-1\}$ {All nodes are informed}
  $I \leftarrow \{k\}, A \leftarrow \{k\}$
  while $|I| < n$ do
    for each $i \in A$ do
      $j, I \leftarrow$ tell_gossip($i, I$), $A \leftarrow A \cup \{j\}$
      $A \leftarrow A \setminus \{i\}$ with probability $1 - s$
    end for
  end while
```

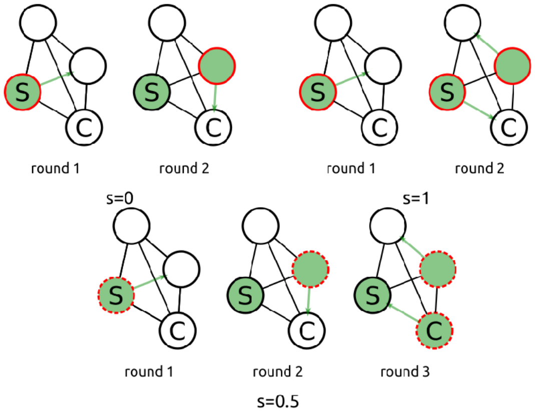

Fig. 1. *Left:* Fast Private Gossip protocol (synchronous setting). *Right:* Illustration of the role of parameter s. S indicates the source and C a curious node. Green nodes know the rumor, and red circled nodes are active. When $s = 0$, there is only one active node at a time, which always stops emitting after telling the gossip. The resulting protocol (private gossip) is private but slow. In the case $s = 1$, nodes always remain active once they know the rumor, leading to a fast but non private protocol (this is the standard push gossip protocol [37]). When $0 < s < 1$, each node remains active with probability s at each round, providing a trade-off between privacy and speed.

nodes can be reduced to the time needed for the classical coupon collection problem: it takes $O(n \log n)$ communications to inform all nodes with probability at least $1 - 1/n$ [18]. As the private gossip protocol performs exactly one communication at each round, it needs $O(n \log n)$ rounds to inform all nodes with high probability. This is much slower than the standard "push" gossip protocol, which requires only $O(\log n)$ rounds [22], motivating the exploration of the privacy-speed trade-off. In the next section, we introduce gossip protocols with $O(\log n)$ speed and nearly optimal privacy.

5 FASTER PRIVATE GOSSIP PROTOCOLS

In this section, we study faster variants of the private gossip protocol in which nodes do not necessarily stop emitting after they first transmit the information. Algorithm 1 describes a class of (synchronous) gossip protocols parameterized by $s \in [0, 1]$, which fits Definition 3.1. Unlike the private gossip protocol, more than one node can spread the rumor at each round as the protocol maintains a set A of active nodes (initialized to the source node). At each round, each active node $i \in A$ invokes the `tell_gossip` primitive to send the information to another node (which in turn becomes active), while i also stays active with probability s. This protocol, illustrated in Figure 1, can be understood as a gossip protocol with a randomized version of *fanout* [20].[3] Intuitively, the set of active nodes will grow until the probability of spawning an additional source (which is exactly s) is equal to the probability of losing a source (i.e., $1 - s$ times the probability of sending a message to a node that was already active). Parameter s controls the trade-off between privacy and speed: in particular, $s = 0$ recovers the private gossip protocol (optimal privacy) and $s = 1$ recovers the standard gossip protocol (optimal speed).

Remark 5.1 (Asynchronous protocol). *Algorithm 1 can be made asynchronous by having a single node i sampled uniformly from A instead of iterating over the full set A.*

In the rest of this section, we study how parameter s of Algorithm 1 impacts its privacy guarantees and its dissemination speed. Section 5.1 establishes that the privacy guarantees of the standard gossip protocol

[3] Unlike in classic fanout, nodes start to gossip again each time they receive a message instead of deactivating permanently.

($s = 1$) must be arbitrarily bad for large graphs. Then, we show in Section 5.2 that nearly optimal privacy can be achieved for smaller s. Finally, Section 5.3 studies the dissemination speed and shows that the known logarithmic diffusion time of the standard gossip protocol also holds for $s > 0$, leading to a sweet spot in the privacy-speed trade-off.

5.1 Standard gossip is not differentially private

Section 4 hints at the fact that gossip protocols need to forget initial conditions quickly in order to be private. In this section, we strengthen this intuition by showing that the differential privacy guarantees of the standard gossip protocol (corresponding to Algorithm 1 with $s = 1$) become arbitrarily bad as the size of the graph increases (keeping the fraction of curious nodes constant).

Theorem 5.1. *If Algorithm 1 with $s = 1$ guarantees (ϵ, δ)-differential privacy in the synchronous setting for all values of n, then $\delta = 1$. The same result holds for the asynchronous version defined in Remark 5.1.*

The proof of this result can be found in Appendix C. Essentially, it comes from the fact that the event "node 0 communicates with a curious node before node 1 gets the message" becomes more and more likely as n grows, hence preventing any meaningful differential privacy guarantee when n is large enough. Theorem 5.1 is actually derived from a stronger result establishing lower bounds on the δ achievable by Algorithm 1 that depend on s (see Theorem C.1 in Appendix C). This motivates our interest for protocols with parameter $s < 1$.

5.2 Privacy guarantees of the fast private gossip protocol

The previous section clearly highlights the fact that the standard gossip protocol ($s = 1$) is not differentially private. We now show that giving nodes the possibility to stop emitting by setting $s < 1$ is enough for the protocol to have non-trivial privacy guarantees.

Theorem 5.2. *For $0 < s < 1$, Algorithm 1 with parameter s guarantees $(0, \delta)$-differential privacy in both the synchronous and asynchronous settings with for any fixed $r \in \mathbb{N}^*$:*

$$\delta = 1 - (1 - s) \sum_{k=0}^{\infty} s^k \left(1 - \frac{f}{n}\right)^{k+1} \leq 1 - (1 - s^r) \left(1 - \frac{f}{n}\right)^r . \tag{4}$$

Proof. See Appendix E.1. □

Theorem 5.2 proves a $(0, \delta)$-differential privacy result, which means that apart from some unlikely outputs that may disclose the identity of the source node, most outputs actually have the same probability regardless of which node started the diffusion. The guarantee we obtain here holds for any graph with fixed proportion f/n of curious nodes. Figure 2 (left) shows the gap between the differential privacy guarantees given by Theorem 5.2 and the optimal guarantees of Theorem 4.1 (i.e., the ratio between the upper bound and lower bound on δ). We can see that both bounds are of the same order of magnitude when s is not too large: in particular, the ratio is less than 2 for all $s \leq 0.5$. This indicates that the privacy guarantees are very tight in this regime. Note that setting $r = 1$ in Theorem 5.2 leads to an additive gap of $s(1 - f/n)$ between the privacy of Algorithm 1 and the optimal guarantee, showing that one can be as close as desired to the optimal privacy as long as s is chosen close enough to 0. We also recover exactly the optimal guarantee of Theorem 4.1 in the case $s = 0$ (without the ability to control the trade-off between ϵ and δ).

Remark 5.2 (Fixed-size graphs). *When $s = 1$, the bound of Equation 4 gives $\delta = 1$ (trivial guarantees). This is expected because our bound does not depend on the size of the graph and Theorem 5.1 states that $\delta = 1$ is needed for arbitrarily large graphs. However, it is still possible to have some privacy guarantees when n is*

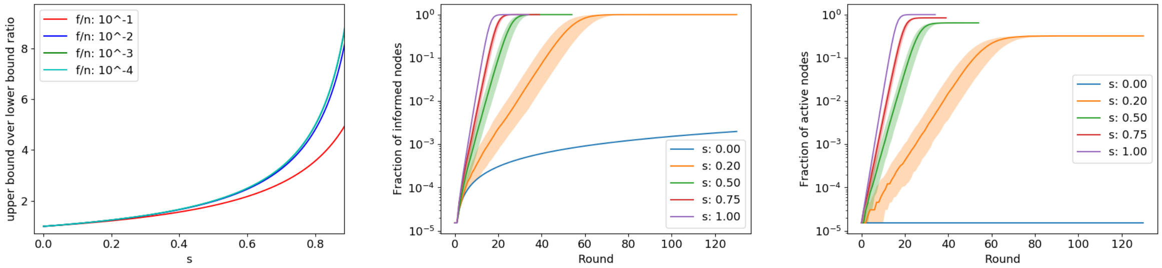

Fig. 2. Effect of parameter s of Algorithm 1 on the privacy guarantees and spreading speed. *Left:* Ratio between the upper bound and lower bound on the differential privacy parameter δ. *Middle, Right:* Fractions of informed and active nodes along rounds on a network of $n = 2^{16}$ nodes. The lines stop when all nodes are informed (and so the protocol terminates), except for $s = 0$ since the protocol is very slow in this case. The curves represent median values and the shaded area represents the 10 and 90 percent confidence intervals computed over 100 runs.

not too large. We characterize these guarantees for fixed values of n in Appendix F. They confirm that setting $s < 1$ allows to obtain much stronger privacy.

Importantly, we also prove that the asynchronous version of Algorithm 1 with $s < 1$ satisfies prediction uncertainty, unlike the case where $s = 1$.

Theorem 5.3. *In the asynchronous setting, Algorithm 1 guarantees prediction uncertainty with $c = (1 - \frac{f+1}{n})(1 - s)$.*

Proof. See Appendix E.2. □

5.3 Dissemination speed

We have shown in the previous section that parameter s has a significant impact of privacy, from optimal ($s = 0$) to very weak ($s = 1$) guarantees. Intuitively, s also impacts the dissemination speed: the larger s, the more active nodes at each round. This is highlighted by the two extreme cases, for which the speed is known and exhibits a large gap: $O(\log n)$ for $s = 1$ [22] while it is only $O(n \log n)$ when $s = 0$. To establish whether we can obtain a protocol that is both private and fast, we need to characterize the dissemination speed for the cases where $0 < s < 1$.

The key result of this section is to prove that the logarithmic speed of the standard gossip protocol holds more generally for all $s > 0$. This result is derived from the fact that the ability to forget does not prevent an *exponential growth* phase. What changes is that the population of active nodes takes approximately $1/s$ rounds to double instead of 1 for standard gossip.

Theorem 5.4. *For a given $s > 0$, there exists $\alpha > 0$ such that for all $C > 0$, there exists n large enough such that the synchronous version of Algorithm 1 with parameter s sends at least $Cn \log n$ messages in $C\alpha^{-1} \log n$ rounds with probability at least $1 - 1/n$.*

Sketch of proof. The key argument of the proof is that after a transition phase of a logarithmic number of rounds, a constant fraction of the nodes (depending on s) remains active despite the probability to stop emitting after each communication. The proof builds on ideas presented in Lemma 15 in [39], and relies on mean-field equations and concentration inequalities. The details can be found in Appendix G.1. □

A similar result (with an appropriate notion of rounds) can be obtained for the asynchronous version of Algorithm 1 (see Appendix G.2 for details). These results show that our parameterized gossip protocol with

$s > 0$ still has a logarithmic spreading time even if nodes can stop transmitting the message. Note that the constant α depends on s and will go to infinity as $s \to 0$ because $1/s$ rounds are needed in expectation to double the population of active nodes (even without taking collisions into account). Simulations shown in Figure 2 (right) confirm that the fraction of active nodes grows exponentially fast for all values of s and then reaches a plateau when the probability of creating a new active node is compensated by the probability of message collisions (informing an already active node). Empirically, this happens when the fraction of active nodes is of order s, meaning that the last phase (during which the remaining uninformed nodes need to be reached by a stable number of $s \times n$ active nodes) remains short. This highlights the fact that Algorithm 1 remains significantly faster than the slow private gossip: for instance, dissemination speed for $s = 0.5$ is very close to the fastest case $s = 1$ (see Figure 2, middle).

Combining with our previous results, we have thus shown that one can achieve both fast spreading and near-optimal privacy.

6 CONCLUDING REMARKS

This paper initiates the study of privacy in gossip protocols to determine the extent to which the source of a gossip can be traceable. Our contributions are the following. (1) We proposed a formal model of anonymity in gossip protocols based on an adaptation of differential privacy. (2) We established tight upper bounds on the privacy of gossip protocols, highlighting the natural privacy guarantees brought by gossip protocols as well as a fundamental difference between the synchronous and the asynchronous cases. (3) We precisely captured the trade-off between privacy and speed with a parameterized gossip protocol allowing nodes to stop gossiping after some time, showing that we can design gossip protocols that are both fast and near-optimally private.

Our results leave open the question of whether the gap in prediction uncertainty between the cases $s = 0$ and $s > 0$ can be reduced or is in fact unavoidable. The analysis for $s = 0$ heavily relies on the fact that for any output sequence S, $p_i(S|S_0)$ does not depend on i and that $p_i(S_0)$ is rather easy to evaluate. These properties break when $s > 0$, leading to a substantially more involved analysis.

More broadly, our work opens several interesting perspectives. First, it paves the way to the study of the privacy guarantees of gossip protocols in *arbitrary graphs*. In this setting, the desired notion of privacy should allow two nodes to become more and more distinguishable as their distance in the graph increases, and could also depend on the distance to the closest curious node. This could be formalized by considering metric-based relaxations of differential privacy [3, 8], or Pufferfish privacy [32]. The latter is a flexible framework that introduces the notion of secrets to protect, allowing for instance to encode the fact that two distant nodes in the network do not need to be indistinguishable.

Another exciting avenue for future research is motivated by some recent work showing that hiding the source of a message can significantly amplify differential privacy guarantees for the *content* of the message [9, 19]. However, primitives to hide the source of messages such as onion routing [45] can be difficult and costly to deploy. Showing that gossiping can *naturally* amplify differential privacy for the message contents would make gossip protocols very desirable for privacy-friendly distributed applications and privacy-preserving decentralized machine learning [5].

ACKNOWLEDGMENTS

This research was partially supported by grants ANR-16-CE23-0016-01 and ANR-18-CE23-0018-03, by a grant from CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020 and by European ERC Grant 339539 - AOC.

REFERENCES

[1] Huseyin Acan, Andrea Collevecchio, Abbas Mehrabian, and Nick Wormald. On the push&pull protocol for rumor spreading. *SIAM Journal on Discrete Mathematics*, 31(2):647–668, 2017.

[2] Divyakant Agrawal, Amr El Abbadi, and Robert C Steinke. Epidemic algorithms in replicated databases. In *Proceedings of the 16th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 161–172. ACM, 1997.

[3] Miguel E. Andrés, Nicolás Emilio Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. Geo-indistinguishability: differential privacy for location-based systems. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2013.

[4] Richard Arratia and Louis Gordon. Tutorial on large deviations for the binomial distribution. *Bulletin of mathematical biology*, 51(1):125–131, 1989.

[5] Aurélien Bellet, Rachid Guerraoui, Mahsa Taziki, and Marc Tommasi. Personalized and Private Peer-to-Peer Machine Learning. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2018.

[6] Petra Berenbrink, Jurek Czyzowicz, Robert Elsässer, and Leszek Gąsieniec. Efficient information exchange in the random phone-call model. *Automata, Languages and Programming*, pages 127–138, 2010.

[7] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE transactions on information theory*, 52(6):2508–2530, 2006.

[8] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. In *Proceedings of the International Symposium on Privacy Enhancing Technologies (PETS) Year = 2013*.

[9] Albert Cheu, Adam D. Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. Distributed Differential Privacy via Shuffling. Technical report, arXiv:1808.01394, 2018.

[10] Igor Colin, Aurélien Bellet, Joseph Salmon, and Stéphan Clémençon. Gossip dual averaging for decentralized optimization of pairwise functions. *arXiv preprint arXiv:1606.02421*, 2016.

[11] Wei-Yen Day, Ninghui Li, and Min Lyu. Publishing graph degree distribution with node differential privacy. In *Proceedings of the 2016 International Conference on Management of Data (SIGMOD)*, pages 123–138, 2016.

[12] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. Epidemic algorithms for replicated database maintenance. In *Proceedings of the sixth annual ACM Symposium on Principles of distributed computing*, pages 1–12. ACM, 1987.

[13] Benjamin Doerr, Mahmoud Fouz, and Tobias Friedrich. Social networks spread rumors in sublogarithmic time. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 21–30. ACM, 2011.

[14] John C. Duchi, Alekh Agarwal, and Martin J. Wainwright. Dual Averaging for Distributed Optimization: Convergence Analysis and Network Scaling. *IEEE Transactions on Automatic Control*, 57(3):592–606, 2012.

[15] Cynthia Dwork. Differential Privacy. In *Proceedings of the 33rd International Colloquium on Automata, Languages and Programming*, pages 1–12, 2006.

[16] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503, 2006.

[17] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[18] Paul Erdős. On a classical problem of probability theory. 1961.

[19] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, and Ananth Raghunathan abd Kunal Talwar. Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity. Technical report, arXiv:1811.12469, 2018.

[20] Patrick T Eugster, Rachid Guerraoui, Anne-Marie Kermarrec, and Laurent Massoulié. Epidemic information dissemination in distributed systems. *Computer*, 37(5):60–67, 2004.

[21] Giulia Fanti, Peter Kairouz, Sewoong Oh, Kannan Ramchandran, and Pramod Viswanath. Hiding the rumor source. *IEEE Transactions on Information Theory*, 2017.

[22] Alan M Frieze and Geoffrey R Grimmett. The shortest-path problem for graphs with random arc-lengths. *Discrete Applied Mathematics*, 10(1):57–77, 1985.

[23] Mohsen Ghaffari and Calvin Newport. How to discreetly spread a rumor in a crowd. In *International Symposium on Distributed Computing*, pages 357–370. Springer, 2016.

[24] George Giakkoupis. Tight bounds for rumor spreading in graphs of a given conductance. In *Symposium on Theoretical Aspects of Computer Science (STACS2011)*, volume 9, pages 57–68, 2011.

[25] George Giakkoupis, Yasamin Nazari, and Philipp Woelfel. How asynchrony affects rumor spreading time. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, pages 185–194. ACM, 2016.

[26] Hadrien Hendrikx, Francis Bach, and Laurent Massoulié. Accelerated decentralized optimization with local updates for smooth and strongly convex objectives. *arXiv preprint arXiv:1810.02660*, 2018.

[27] Zhiyi Huang and Jinyan Liu. Optimal differentially private algorithms for k-means clustering. In *Proceedings of the 37th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 395–408, 2018.

[28] Jiaojiao Jiang, Sheng Wen, Shui Yu, Yang Xiang, and Wanlei Zhou. Identifying propagation sources in networks: State-of-the-art and comparative studies. *IEEE Communications Surveys & Tutorials*, 19(1):465–481, 2017.

[29] Richard Karp, Christian Schindelhauer, Scott Shenker, and Berthold Vocking. Randomized rumor spreading. In *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*, pages 565–574. IEEE, 2000.

[30] Srinivas R. Kashyap, Supratim Deb, K. V. M. Naidu, Rajeev Rastogi, and Anand Srinivasan. Efficient gossip-based aggregate computation. In *Proceedings of the 25th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 308–317, 2006.

[31] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-Based Computation of Aggregate Information. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 482–491, 2003.

[32] Daniel Kifer and Ashwin Machanavajjhala. Pufferfish: A framework for mathematical privacy definitions. *ACM Transactions on Database Systems (TODS)*, 39(1):3, 2014.

[33] Chao Li, Michael Hay, Vibhor Rastogi, Gerome Miklau, and Andrew McGregor. Optimizing linear counting queries under differential privacy. In *Proceedings of the 29th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 123–134, 2010.

[34] Wentian Lu and Gerome Miklau. Exponential random graph estimation under differential privacy. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD)*, pages 921–930, 2014.

[35] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[36] Pedro C Pinto, Patrick Thiran, and Martin Vetterli. Locating the source of diffusion in large-scale networks. *Physical review letters*, 109(6):068702, 2012.

[37] Boris Pittel. On spreading a rumor. *SIAM Journal on Applied Mathematics*, 47(1):213–223, 1987.

[38] S Sundhar Ram, A Nedić, and Venugopal V Veeravalli. Asynchronous gossip algorithms for stochastic optimization. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 3581–3586. IEEE, 2009.

[39] Sujay Sanghavi, Bruce Hajek, and Laurent Massoulié. Gossiping with multiple messages. *IEEE Transactions on Information Theory*, 53(12):4640–4654, 2007.

[40] Devavrat Shah et al. Gossip algorithms. *Foundations and Trends in Networking*, 3(1):1–125, 2009.

[41] Devavrat Shah and Tauhid Zaman. Rumors in a network: Who's the culprit? *IEEE Transactions on information theory*, 57(8):5163–5181, 2011.

[42] Devavrat Shah and Tauhid Zaman. Rumor centrality: a universal source detector. In *ACM SIGMETRICS Performance Evaluation Review*, volume 40, pages 199–210, 2012.

[43] Devavrat Shah and Tauhid Zaman. Finding rumor sources on random trees. *Operations Research*, 64(3):736–755, 2016.

[44] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, 2015.

[45] Paul F. Syverson, David M. Goldschlag, and Michael G. Reed. Personalized and Private Peer-to-Peer Machine Learning. In *Proceedings of the IEEE Symposium on Security and Privacy (S&P)*, 1997.

A DISCUSSION ON MODEL EXTENSIONS

We have kept our model of Section 3 relatively simple to avoid unnecessary complexity in the notations and additional technicalities in the derivation and presentation of our results. In this section, we briefly discuss some possible extensions. Our main point is to illustrate the fact that they will generally lead to some technical complications without impacting the privacy guarantees significantly.

A.1 Pull and push-pull protocols

Our study focuses on the classic *push* form of gossip protocols. This can be justified by the fact that, for regular graphs, synchronous push has asymptotic spreading time guarantees that are comparable with the push-pull variant [25]. Besides, the differential privacy guarantees of any gossip protocol are limited by the probability that the first node informed by the source is a curious node, and we show this bound can be matched with push protocols. Nevertheless, extensions of our results to pull and push-pull variants of gossip protocols [29] are possible. Forgetting mechanisms similar to the ones in Algorithm 1 can be introduced for these protocols, i.e. nodes would have a probability $1 - s$ to stop disclosing information after each time they are pulled (if they do not pull someone with the information in between). Although slightly different, the optimal privacy guarantees would remain of the same order of magnitude. Yet, we expect pull guarantees to be much worse in the case $s = 1$ because curious nodes could stop suspecting all nodes that they have pulled and that did not have the rumor. Besides, the pull protocol for $s = 0$ would be even slower than its push counterpart.

A.2 Eavesdropping adversary

Since we consider a complete graph, our formalization of the attackers as a fraction f/n of curious nodes is closely related to an eavesdropping attacker who would observe each communication with probability f/n. Indeed, both models consider that each communication has a probability f/n of being disclosed to the attacker. Most of our results are thus easily transferable to this alternative setting. The only difference would be that all nodes can be suspected in the eavesdropping model, thus introducing a $(1 - f/n)^{-1}$ factor each time we consider the population of non-curious nodes.

A.3 Information observed by attackers

We discuss two possible generalizations of the output observed by the attackers (Definition 3.2).

Messages sent by curious nodes. For simplicity of exposition, Definition 3.2 considers that curious nodes only observe messages that are sent to them and not the messages that they send. However, including the messages sent by curious nodes in their observed output would not impact the bounds on privacy (i.e., the guarantees for the algorithms). For the optimal algorithm of Section 4.2, we only consider what happens during the first round, so including the messages sent by curious nodes does not change the result. This in particular implies that the fundamental limits of Theorem 4.1 remain the same (since the attackers observe strictly more information). Similarly, for the parameterized algorithm of Section 5, Theorem 5.2 is obtained by bounding the probability of a set $\hat{S}$. Then, we have $p(\hat{S}, S_{\text{out}}) \leq p(\hat{S})$ where S_{out} is the sequence of messages sent by the curious nodes. In general, adding the messages sent by curious nodes to the output sequences has little or no impact on the results.

Message ordering in the asynchronous setting. We assume in Definition 3.2 that the relative order of messages is preserved in the output sequence observed by curious nodes. This could be relaxed in the asynchronous setting, as in practical scenarios a message sent before another may well be received after it. One could for instance introduce a random swapping model to take this into account and investigate whether this weaker output leads to an improvement in the privacy guarantees. However, we argue that this improvement would be quite limited. First of all, it would not affect the privacy guarantees of Section 4: since there is a single active node able to send a message at any given time, swapping is not possible. Furthermore, even when several nodes are active at the same time (e.g., in Algorithm 1 with $s > 0$), the proofs can be adapted to work with counting the messages *received* instead of the messages *sent*. In this case, swapping is as likely to expose the source (making its messages arrive earlier) than to hide it (delaying the messages it sends). Therefore, privacy would not improve substantially.

A.4 Malicious behavior

In this work, we have assumed for simplicity that nodes are *curious* but not *malicious*, i.e., they follow the protocol. This is motivated by a practical scenario where a subset of nodes are simply being monitored by a curious entity. If curious nodes can also act maliciously, they have three possible ways to affect the protocol: emitting more, emitting less, or not choosing neighbours uniformly at random. If they emit more, they will inform more nodes, which makes it more difficult for them to locate the source. If they emit less (potentially not at all), then in the case $s < 1$ the protocol could stop before all nodes are informed. Yet, the privacy bounds are derived from the fact that the source forgets the information before communicating to a curious node. If they choose the neighbors they send the messages to, it reduces to the case in which they emit less (because they do not send messages to uninformed nodes) but without affecting protocol speed or termination (because it does not reduce the number of active nodes). Thus, the impact on the observed output and therefore on the privacy would be minimal. In the case $s = 1$, malicious nodes have slightly more impact but remain quite small as it only makes the set of informed nodes grow slightly slower.

A.5 Termination criterion

For simplicity, in all our gossip protocols we have used a global termination criterion (the protocol terminates when all nodes are informed). Termination without using global coordination is a problem in its own right that has been extensively studied (see for instance [29]). Although some termination criteria could have a great impact on privacy, we argue that termination can be handled late in the execution so as to reveal very little about the beginning, hence avoiding any significant impact on privacy. For instance, it is possible to design a variant of Algorithm 1 in which nodes only flip a coin with probability s for a fixed number of times, and then stop emitting completely. This fixed number would have to depend on s, but then if it is large enough, it would guarantee both termination and privacy. Indeed, nodes would not communicate with curious nodes each time they are activated with high probability so this counter would actually provide very little information to the curious nodes. Determining how large this number of iterations should be, and the exact impact on privacy (which we argue is very small), is beyond the scope of this paper.

B PROOFS OF THE OPTIMAL PRIVACY RESULTS

In this section, we prove Theorem 4.1. To do so, we start by deriving the lower bounds on the privacy parameters and then show that they are matched by the private gossip protocol.

B.1 Lower bounds

To this end, we introduce a useful technical lemma which directly follows from the definition of differential privacy and is at the heart of our lower bound proofs. Lemma B.1 means that proving a lower bound on the differential privacy parameters can be achieved by finding a set of possible outputs S (here, a set of ordered sequences) that is more likely if node 0 starts the gossip than if node 1 does. It is a direct application of the definition of differential privacy.

Lemma B.1. *Given any gossip protocol, let $S \subset \mathcal{S}$ and $w_0, w_1 \in \mathbb{R}$ be such that $w_0 \leq p_0(S)$ and $p_1(S) \leq w_1$. If the protocol satisfies (ϵ, δ) differential privacy then $\delta \geq w_0 - e^{\epsilon} w_1$.*

The lower bound result we would like to prove is the following:

Theorem. *If a gossip protocol satisfies (ϵ, δ)-differential privacy and c-prediction uncertainty then we have $\delta \geq \frac{f}{n}$ and $c = 0$ in the synchronous setting, and $\delta \geq \frac{f}{n}\left(1 - \frac{e^{\epsilon}-1}{f}\right)$ and $c \leq \frac{n}{f+1} - 1$ in the asynchronous setting.*

Proof. We start with the synchronous case. Since `tell_gossip` requires that the input node i is in I and that at the beginning, $I = \{0\}$, the first time the procedure is called must be on node 0. The procedure is called at least once otherwise the protocol terminates with $I = \{0\}$, violating the conditions of Definition 3.1. We denote by $S^{(0)}$ the set of output sequences such that $S_0 = 0$ (i.e., 0 communicated with a curious node at time 0). Since the protocol is run on the complete graph, the node selected by `tell_gossip` is chosen uniformly within $\{0, \ldots, n-1\}$, so a curious node is selected with probability $\frac{f}{n}$. We thus have $p_0(S^{(0)}) = \frac{f}{n}$. Besides, node 0 cannot communicate with a curious node at time 0 if node 1 starts the rumor so $p_1(S^{(0)}) = 0$. We conclude by Lemma B.1. For prediction uncertainty, using the same sequence $S^{(0)}$ yields $\frac{p_i(S^{(0)})}{p_0(S^{(0)})} = 0$ for all $i \neq 0$ and therefore $c = 0$.

We now prove the result for the asynchronous setting. By the same reasoning as before, `tell_gossip` is called at least once and is first called on node 0. Unlike in the synchronous case, recall that the event $S_0 = 0$ means that 0 was the first to communicate with a curious node (which is not necessarily at time 0).

We denote by T_0^c the event such that the starting node does not communicate with a curious node for its first communication. Conditionally upon T_0^c, the node that started the gossip is at least as likely as any other node to emit the second message, because with probability $\frac{1}{n}$ it is the only node with the rumor after the first message is sent. Since the probability to hit a curious node is the same regardless of who sends the message, we have for all $i, j \notin C$: $p_j(S_0 = i|T_0^c) \leq p_j(S_0 = j|T_0^c)$. From this inequality we get

$$\sum_{i \notin C} p_0(S_0 = 0|T_0^c) \geq \sum_{i \notin C} p_0(S_0 = i|T_0^c) = 1 \geq \sum_{i \notin C} p_0(S_0 = 1|T_0^c),$$

where the equality comes from the fact that S_0 is the first node that communicates with a curious node (and the curious nodes do not start with the information). The second inequality comes from the fact that $p_j(S_0 = i|T_0^c) = p_j(S_0 = k|T_0^c)$ for all i, k different from j. Therefore, we have $p_0(S_0 = 0|T_0^c) \geq \frac{1}{n-f}$ and $p_0(S_0 = 1|T_0^c) \leq \frac{1}{n-f}$.

Combining the above expressions, we derive the probability of $S^{(0)}$ when 0 started the diffusion:

$$\begin{aligned} p_0\left(S^{(0)}\right) &= p_0\left(S^{(0)}, t_c = 0\right) + p_0\left(S^{(0)}, T_0^c\right) = p_0\left(t_c = 0\right) p_0\left(S^{(0)}|t_c = 0\right) + p_0\left(S^{(0)}|T_0^c\right) p_0\left(T_0^c\right) \\ &\geq \frac{f}{n} \times 1 + \frac{1}{n-f}\left(1 - \frac{f}{n}\right) = \frac{f}{n} + \frac{1}{n}. \end{aligned}$$

We can then do the same split if node 1 initially has the message, but in this case $p_1\left(S^{(0)}|t_c = 0\right) = 0$ and we get $p_1(S^{(0)}) = p_1(T_0^c)p_1(S^{(0)}|T_0^c) \leq \frac{1}{n}$. We conclude again by Lemma B.1.

The upper bound on prediction uncertainty is derived using the same quantities. More precisely:

$$\frac{p(I_0 \neq 0|S^{(0)})}{p(I_0 = 0|S^{(0)})} = \sum_{i \notin C \cup \{0\}} \frac{p_i(S^{(0)})}{p_0(S^{(0)})} \leq (n - f - 1)\frac{p_1(S^{(0)})}{p_0(S^{(0)})} \leq (n - f - 1)\frac{1}{f+1} = \frac{n}{f+1} - 1 \qquad (5)$$

□

B.2 Optimal protocol

We now proceed to proving the second part of Theorem 4.1, *i.e.* that these bounds are matched by the private gossip protocol.

Theorem. *For all $\epsilon \geq 0$, Algorithm 1 with $s = 0$ guarantees (ϵ, δ)-differential privacy and c-prediction uncertainty, with $\delta = \frac{f}{n}$ and $c = 0$ in the synchronous setting, and $\delta = \frac{f}{n}\left(1 - \frac{e^{\epsilon}-1}{f}\right)$ and $c = \frac{n}{f+1} - 1$ in the asynchronous setting.*

Proof. For this protocol, the only outputs that have a different probability if node 0 starts (compared to the case when 1 starts) are those in which 0 (or 1) communicates with a curious node at time 0. This is true in both the synchronous and the asynchronous settings. Following our previous notations, we write these two events $S_0 = 0$ and $S_0 = 1$ and further denote by $S^{(0)}$ (resp. $S^{(1)}$) the set of output sequences such that $S_0 = 0$ (resp. $S_0 = 1$).

For the synchronous setting, we have $p_0(S_0 = 0) = p_1(S_0 = 1) = \frac{f}{n}$ and $p_0(S_0 = 1) = p_1(S_0 = 0) = 0$. This ensures that $p_0(S^{(0)}) \leq p_1(S^{(0)}) + \frac{f}{n}$ (similarly for $S^{(1)}$), and the result follows.

We now turn to the asynchronous setting. We denote by T_0 the event such that node 0 communicates with a curious node (and T_0^c the negation of this event). We have:

$$p_0(S_0 = 0) = p(T_0)p_0(S_0 = 0|T_0) + p(T_0^c)p_0(S_0 = 0|T_0^c). \tag{6}$$

For any $i \notin C$ where C is the set of curious nodes, we have that $p_0(S_0 = 0|T_0^c) = p_0(S_0 = i|T_0^c) = \frac{1}{n-f}$. Indeed, given T_0^c, the node that received the first message was selected uniformly at random among non-curious nodes, and have the same probability to disclose the gossip at future rounds. Plugging into (6), we obtain:

$$p_0(S_0 = 0) = \frac{f}{n} + \left(1 - \frac{f}{n}\right)\frac{1}{n-f} = \frac{f+1}{n}.$$

For any other node $i \neq 0$, $p_0(S_0 = i) = p_0(T_0^c)p_0(S_0 = i|T_0^c) = \frac{1}{n}$ because $p_0(S_0 = i|T_0) = 0$. Combining these results we get $p_0(S^{(0)}) \leq e^{\epsilon}p_1(S^{(0)}) + \delta$ for any $\epsilon > 0$ and $\delta = \frac{f}{n}(1 - \frac{e^{\epsilon}-1}{f})$. By symmetry, we can make a similar derivation for $S^{(1)}$, which concludes the proof.

To prove the prediction uncertainty result, we use the differential privacy result with $e^{\epsilon} = f + 1$ (and thus $\delta = 0$ and write that for any $S \in \mathcal{S}$:

$$\frac{p(I_0 \neq 0|S)}{p(I_0 = 0|S)} = \sum_{i \notin C \cup \{0\}} \frac{p_i(S)}{p_0(S)} \geq (n - f - 1)e^{-\epsilon} = \frac{n}{f+1} - 1 \tag{7}$$

□

C LIMITS ON THE DIFFERENTIAL PRIVACY OF PROTOCOL ??

In this section, we prove a general result in the form of lower bounds on parameters ϵ and δ of differential privacy that can be achieved by Algorithm 1, which then imply Theorem 5.1 as a corollary.

C.1 Synchronous setting

For simplicity, we first focus on the synchronous setting.

Theorem C.1. *If Algorithm 1 satisfies (ϵ, δ)-differential privacy in the synchronous setting, then the parameters (ϵ, δ) need to satisfy the following relationships:*

$$\delta \geq \max\left[\frac{f}{n}\left(1 + s\left(1 - \frac{f}{n}\right) - \frac{e^{\epsilon}}{n}\right), \max_{r \in \mathbb{R}}\left(s^r\left(1 - \left(1 - \frac{f}{n}\right)^r\right) - e^{\epsilon}\frac{f}{n}\frac{2^{r+1}}{n}\right)\right]. \tag{8}$$

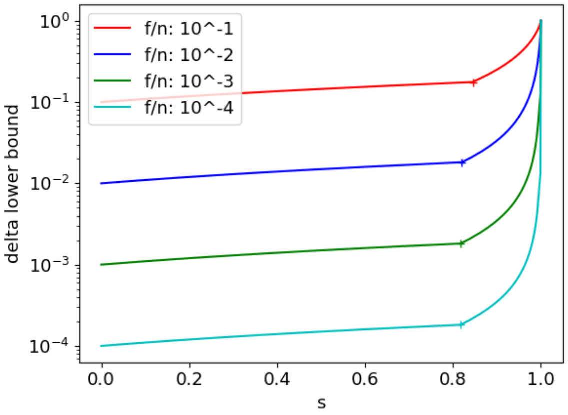

Fig. 3. Lower bounds on δ with respect to s for different fractions of curious nodes in the graph with $\epsilon = 0$ and n arbitrarily large. The left part of the curve corresponds to the first term of Theorem C.1 and the right part to the second term.

Proof. The first part of the lower bound of Equation 8 is proven by exactly evaluating the probability of $S_1^{(0)}$, the set of output sequences for which node 0 communicates with a curious node before round 1. We get that $p_0(S_1^{(0)}) \geq \frac{f}{n}(1 + s(1 - \frac{f}{n}))$ and $p_1(S_1^{(0)}) \leq \frac{f}{n}\frac{1}{n}$. We conclude by the usual Lemma B.1. For s close to 1, one would obtain tighter bounds by considering more rounds, but the expressions quickly become very complex.

For the second part, we fix $r \in \mathbb{N}$ and we denote by t_f the time at which node 0 stops emitting for the first time (with probability $1 - s$ at each round). Then, we condition $p_0(S_r^{(0)})$ on $t_f \geq r$ to get:

$$p_0(S_r^{(0)}) \geq p_0(t_f \geq r) p_0(S_r^{(0)} | t_f \geq r) \geq s^r \big(1 - \big(1 - \frac{f}{n}\big)^{r+1}\big).$$

On the other hand, we can write $p_1(S_r^{(0)})$ as follows:

$$p_1(S_r^{(0)}) = p_1(\cup_{k=0}^{r} \{0 \in S_k\}) \leq \sum_{k=0}^{r} p_1(0 \in I_k) p_1(0 \in S_k | 0 \in I_k).$$

By using the fact that $p_1(0 \in I_k) \leq \frac{|I_k|}{n} \leq \frac{2^k}{n}$, we get $p_1(S_r^{(0)}) \leq \frac{2^{r+1}}{n}\frac{f}{n}$, and we again conclude by Lemma B.1. □

The first bound in Theorem C.1 is tighter for small values of s whereas the second one is more precise as s becomes closer to 1. This is because the first bound ignores the communications occurring after round 1, whose impact is negligible only when s is small enough. A closer look at Equation (8) for a fixed ratio $\frac{f}{n}$ of curious nodes indicates that the term involving ϵ reduces with n. This means that the effect of ϵ becomes negligible as the network grows larger, allowing us to obtain Theorem 5.1. We give an illustration of Theorem C.1 in Figure 3, which shows that the lower bound given by disclosure at the first round is rather precise for many values of s. Yet, the second term clearly shows the explosion of the δ parameter when s is close to 1.

Proof of Theorem 5.1 (synchronous setting). This is a direct corollary of Theorem C.1. Indeed, for any $\Delta > 0$ then we can set r such that $(1 - \frac{f}{n})^r < \frac{\Delta}{2}$ and then n large enough so that $e^{\epsilon} \frac{f}{n} \frac{2^{r+1}}{n} < \frac{\Delta}{2}$ so that we obtain $\delta \geq 1 - \Delta$. □

C.2 Asynchronous setting

For the asynchronous setting, we do not have a lower bound as precise as the one in Theorem C.1. Therefore, we directly prove the asynchronous part of Theorem 5.1.

Proof of Theorem 5.1 (asynchronous setting). We derive a lower bound in a similar manner as for the synchronous setting, but we focus on the case $s = 1$ only. To do so, we will study $S_r^{(0)}$, the set of output sequences such that the rank of node 0 in the sequence is less than r. For a specific sequence not in $S_r^{(0)}$, there must have been at least r communications (because r nodes must have communicated with curious nodes), and none of them involved 0 and a curious node. Therefore, if we note $n_c(r)$ the number of communications that actually happened before the output sequence reached size r, we have $n_c(r) \geq r$. Then, denoting by $C(t)$ the node that communicated with a curious node at time t (with $C(t) = -1$ when the communication did not involve a curious node):

$$p_0(S_r^{(0)}) = 1 - p\left(\cap_{t=0}^{n_c(r)} C(t) \neq 0\right) = 1 - \prod_{t=0}^{n_c(r)} p\left(C(t) \neq 0\right) \geq 1 - \prod_{t=0}^{r} \left(1 - p\left(C(t) = 0\right)\right) \geq 1 - \prod_{t=0}^{r} \left(1 - \frac{f}{n}\frac{1}{t}\right),$$

where the last step comes from the fact that the probability of node 0 to be selected at time t is $\frac{1}{|I_t|} \geq \frac{1}{t}$ because at most one node is informed at each step and the active node is selected uniformly among nodes that have the information. We use the fact that $\log(1 + x) \leq x$ for any $x > -1$ on $x = -\frac{f}{n}\frac{1}{t}$ to show that:

$$\prod_{t=0}^{r} \left(1 - \frac{f}{n}\frac{1}{t}\right) = e^{\sum_{t=0}^{r} \log\left(1 - \frac{f}{n}\frac{1}{t}\right)} \leq e^{-\frac{f}{n}\sum_{t=0}^{r}\frac{1}{t}}. \tag{9}$$

Therefore, $p_0(S_r^{(0)})$ goes to 1 as r goes to infinity.

Then, for a given r and for any $k > 0$, $p(n_c(r) \leq k)$ is equal to $p(Binom(k, \frac{f}{n}) \geq r)$ where $Binom(k, \frac{f}{n})$ is the binomial law of parameters k and $\frac{f}{n}$. This is because it is the probability of having exactly r successes with the sum of less than k Bernoullis of parameter $\frac{f}{n}$, which is equal to the probability of having more than r successes with the sum of k Bernoullis of the same parameters. Therefore, $p(n_c(r) \leq k)$ is independent of n and we can choose k^* independently of n such that $p(n_c(r) > k^*) \leq \frac{1}{n}$. Then, we write that

$$p_1(S_r^{(0)}) = p_0(S_r^{(0)}, n_c(r) \leq k^*) + p_0(S_r^{(0)}, n_c(r) > k^*) \leq p_0(S_r^{(0)} | n_c(r) \leq k^*) + \frac{1}{n}.$$

This implies $p_0(S_r^{(0)} | n_c(r) \leq k^*) \leq p_0(0 \in I_r | n_c(r) \leq k^*) \leq 1 - p(0 \notin I_r | n_c(r) \leq k^*)$. We know that only r communications have reached curious nodes but the others have reached a random node in the graph, and there is at most k^* of them, so finally:

$$p_1(S_r^{(0)}) \leq 1 - (1 - \frac{1}{n})^{k^*} + \frac{1}{n}.$$

We immediately see that $p_1(S_r^{(0)})$ goes to 0 as n grows because k^* is independent of n, and we have shown above that $p_0(S_r^{(0)})$ goes to 1 as n grows. Since we must have that $p_0(S_r^{(0)}) \leq e^{\epsilon} p_1(S_r^{(0)}) + \delta$, we must have $\delta = 1$ if we want δ to be independent of n. □

D DELAYED START GOSSIP

Consider the protocol described in Remark 4.1, that we call *delayed start gossip*:

1. The source calls `tell_gossip` once to transmit the rumor to an arbitrary node, say node j.
2. Node j then starts a standard gossip (Algorithm 1 with $s = 1$).

This simple protocol is optimal from the point of view of differential privacy because if the first communication does not hit a curious node then the probability of a given output when two different nodes start the gossip is the same. It is also fast since it runs the standard gossip after the first round.

Yet, this naive protocol has a major flaw. Indeed, when the first communication hits a curious node, the attackers can monitor whether the sender communicates with them again in the next rounds. If it does not, they can guess that the node is the source, and they will in fact make a correct guess with probability arbitrarily close to 1 for large enough graphs. On the other hand, when the sender communicates again with a curious node shortly after, they can be very confident that this node is not the source. Hence, it is possible to design a very simple attack with a very high precision (almost always right) and almost optimal recall (finds the source when the information is released, i.e. with probability $\frac{f}{n}$).

Making sure that the attacker is uncertain about its prediction is therefore a desirable property. This is captured by the notion of *prediction uncertainty* that we introduced in Section 3.3. The following proposition formalizes the above claims and motivates the need for more involved protocols such as the faster private gossip protocol presented in Section 5.

Proposition D.1. *We call c_{ds} the prediction uncertainty constant of the asynchronous delayed start protocol and we assume the ratio of curious nodes f/n to be constant. Then $c_{ds} \to 0$ when $n \to \infty$.*

More generally, it is in principle possible to prove similar results for any protocol in which the source node does not behave like other nodes. Indeed, if the special behaviour can be detected, then attackers can know for sure the source of the rumor.

Proof of Proposition D.1. The proof reuses some elements of the proof of Theorem 5.1 for the asynchronous setting (see Appendix C.2). We consider the sequence $S_r^{(0)}$ such that node 0 is the first node to communicate with a curious node ($S_0 = 0$) and then r other nodes communicate with curious nodes before 0 does ($S_i \neq 0$ for $i \in \{1, ..., r\}$). We denote by t_0 the time at which node 0 gets the message and becomes active again (we refer here to the global order, although of course the curious nodes do not have access to it). Then, with the usual notations we have:

$$p_0\left(S_r^{(0)}\right) = p_0(S_0 = 0)p_0\left(S_r^{(0)}|S_0 = 0\right) \geq \frac{f}{n}p_0\left(\cap_{i=1}^r S_i \neq 0|S_0 = 0\right) \geq \frac{f}{n}p_0(t_0 \geq r)$$
$$\geq \frac{f}{n}p_0(n_c(r) \leq k^*)p_0(t_0 \geq r|n_c(r) \leq k^*).$$

Then, we recall from the proof of Theorem 5.1 (Appendix C.2) that

$$p_0(n_c(r) \leq k) = p\left(Binom(k, \frac{f}{n}) \geq r\right) = p\left(Binom(k, 1 - \frac{f}{n}) < k - r\right) = 1 - p\left(Binom(k, 1 - \frac{f}{n}) \geq k - r\right),$$

so if we set $k = \frac{2n}{f}r$ and use tail bounds on the binomial law (Theorem 1 of [4]) then there exists a constant H (only depending on $\frac{f}{n}$) such that:

$$p_0(n_c(r) \leq r\frac{2n}{f}) \geq 1 - e^{-rH}.$$

Therefore, we have:

$$p_0\left(S_r^{(0)}\right) \geq \frac{f}{n}\left(1 - e^{-rH}\right)\left(1 - \frac{1}{n}\right)^{r\frac{2f}{n}} \geq C_1(r, n). \tag{10}$$

The last line comes from calculations done in the proof of Theorem 5.1 (Appendix C.2).

We now study $p_1(S_r^{(0)})$. Since node 1 started the protocol then it means that no other node (and in particular 0) will stop emitting the message. Therefore, if node 0 is the first to communicate with a curious node then it will remain active for the whole duration of the protocol. Consider that the first disclosure happens after T_f communications. We can write:

$$\begin{aligned} p_1\left(S_r^{(0)}\right) \leq\ & p_1(S_0 = 0)p_1\left(\cap_{i=1}^r S_i \neq 0 | S_0 = 0, T_f \leq t_f\right) \\ & + p_1(T_f > t_f). \end{aligned}$$

Since the fraction of curious nodes is constant, we can choose t_f independently of n or r such that $p(T_f > t_f) \leq e^{-\frac{f}{n}t_f} \leq \frac{\epsilon}{4(n-f)}$ if $t_f = \frac{n}{f}\log\left(\frac{4(n-f)}{\epsilon}\right)$ in order to control the second term. Then,

$$p_1\left(\cap_{i=1}^r S_i \neq 0 | S_0 = 0, T_f \leq t_f\right) \leq \prod_{t=t_f}^{t_f+r}\left(1 - \frac{f}{n}\frac{1}{t}\right) \leq e^{-\frac{f}{n}\sum_{t=t_f}^{t_f+r}\frac{1}{t}}.$$

A series-integral comparison yields that if $r = \log^2(n)$ then $e^{-\frac{f}{n}\sum_{t=t_f}^{t_f+r}\frac{1}{t}} \leq \frac{\epsilon}{4}$ for n large enough. Finally, we use the fact that $p_1(S_0 = 0) \leq \frac{1}{n-f}$ to write that:

$$p_1\left(S_r^{(0)}\right) \leq \frac{\epsilon}{2(n-f)}. \tag{11}$$

Finally, we observe that $C_1(\log^2 n, n) \longrightarrow \frac{f}{n}$ when $n \longrightarrow \infty$ where C_1 is defined in Equation 10. In particular, $C_1(\log^2 n, n) \geq \frac{f}{2n}$ for n large enough, so we have

$$\frac{p(I_0 \neq 0|S_r^{(0)})}{p(I_0 = 0|S_r^{(0)})} = \sum_{i \notin C\cup\{0\}}\frac{p_i(S_r^{(0)})}{p_0(S_r^{(0)})} \leq \frac{n}{f}\epsilon. \tag{12}$$

Since ϵ can be picked arbitrarily small and $\frac{n}{f}$ is assumed to be constant then the previous ratio can be made arbitrary small. □

E PRIVACY GUARANTEES OF FAST PRIVATE GOSSIP

E.1 Differential privacy

We prove here the differential privacy guarantees for Algorithm 1 when $0 < s < 1$ (Theorem 5.2).

Proof of Theorem 5.2. Let us define the event F which denotes the fact that the source stops emitting before communicating with a curious node, and $\bar{F}$ its negation. In particular, $p_0(S, F) = p_i(S, F)$ for all i because the source never directly communicates with a curious node so it does not impact the output sequence. Then, we can write:

$$p_0(S) = p_0(S, F) + p_0(S, \bar{F}) \leq p_1(S, F) + p_0(\bar{F}) \leq p_1(S) + p_0(\bar{F}).$$

Then, noting T_f the number of messages after which the source stops emitting, we write for $s > 0$:

$$p_0(\bar{F}) = \sum_{k=1}^{\infty} p_0(T_f = k)p_0(\bar{F}|T_f = k) = \sum_{k=0}^{\infty}(1 - s)s^k\left(1 - \left(1 - \frac{f}{n}\right)^{k+1}\right).$$

Note that we can also write for $k \geq 1$:

$$p_0(\bar{F}) = p_0(\bar{F}, T_f \leq k) + p_0(\bar{F}, T_f > k) \leq (1 - s^k)\left(1 - \left(1 - \frac{f}{n}\right)^k\right) + s^k = 1 - (1 - s^k)\left(1 - \frac{f}{n}\right)^k.$$

Note that in particular, the proof holds for both the synchronous and asynchronous versions of Algorithm 1. □

E.2 Prediction uncertainty

We prove here that the asynchronous version of our parameterized protocol (Algorithm 1) guarantees prediction uncertainty when $s < 1$ (Theorem 5.3).

Proof of Theorem 5.3. For any set of sequences $S \subset \mathcal{S}$ such that $p_0(S) > 0$, we have:

$$\frac{p(I_0 \neq 0|S)}{p(I_0 = 0|S)} = \sum_{i \notin C \cup \{0\}} \frac{p_i(S)}{p_0(S)} \geq \sum_{i \notin C \cup \{0\}} \frac{p_i(A_1 = \{0\}) p_i(S|A_1 = \{0\})}{p_0(S)},$$

where A_1 is the set of active nodes at round 1. Because the state of the system (active nodes) is the same in both cases we can write that $p_i(S|A_1 = \{0\}) = p_0(S)$. Besides, $p_i(A_1 = \{0\})$ corresponds to the probability that node i sends a message to node 0 and then stops emitting. Therefore:

$$\frac{p(I_0 \neq 0|S)}{p(I_0 = 0|S)} \geq \left(1 - \frac{f+1}{n}\right)(1 - s) > 0.$$

This shows that the fast private gossip guarantees prediction uncertainty and concludes the proof. □

Remark E.1 (Background knowledge). *As stated in Section 3.3, prediction uncertainty guarantees are not robust to background knowledge. However, the definition can be adapted to include such knowledge, for example to model the fact that attackers may know that some nodes did not start the rumor. This corresponds to changing the $p(I_0 \neq \{0\}|S)$ to $p(I_0 \in J|S)$, where J is the set of nodes that can actually be suspected.*

F PRIVACY OF THE STANDARD GOSSIP PROTOCOL FOR FIXED SIZE GRAPHS

Although the standard gossip protocol obtained by taking $s = 1$ guarantees arbitrarily bad privacy in the limit case of large n, the balancing term $e^{\epsilon \frac{f}{n} \frac{2^{r+1}}{n}}$ fades quite slowly with n so we can hope to get acceptable guarantees for some values of n. The following theorem deals with the case of the standard gossip protocol for fixed size graphs.

Theorem F.1. *For all $r, R_1, R_2 \in \mathbb{N}$ such that $R_1 \leq R_2 \leq r$, the standard synchronous gossip protocol is $(0, \delta_{r,R_1,R_2})$-differentially private with:*

$$\begin{aligned}\delta_{r,R_1,R_2} =& 1 - \left(1 - \frac{f}{n}\right)^{r+\tilde{r}_n+1} \prod_{k=1}^{r-\tilde{r}_n}\left(1 - \frac{f}{n}\frac{2^k}{n-f}\right) + 1 - \left(1 - \left(1 - \frac{M_{R_2-R_1} - 1}{n}\right)\left(1 - \frac{1}{n}\right)^{(r-R_2)M_{R_2-R_1}}\right) \\ &\times \frac{(n-1)!}{(n-2^{R_1})! n^{2^{R_1}-1}} \prod_{k=0}^{R_2-R_1-1} \left(1 - f(\eta_k)^{\mu_k}\right),\end{aligned}$$

with $\tilde{r}_n = \max(0, r - \lfloor \log_2(n - f) \rfloor)$, $M_0 = 2^{R_1}$, $\mu_k = 2M_k \frac{M_k - 1}{n}$ *and*

$$M_{k+1} = 2M_k\left(1 - (1 + \eta_k)\frac{M_k}{n}\right),$$

where $f(\eta) = \frac{e^\eta}{(1+\eta)^{(1+\eta)}}$.

Proof. To prove this theorem, we will use the set S_r that contains all outputs such that neither 0 nor 1 communicate with curious nodes before round r. We call $\hat{S}_r = \mathcal{S} \backslash S_\tau$. We can write for any set of sequences $S \in \mathcal{S}$:

$$\begin{aligned} p_0(S) &= p_0(S \cap \hat{S}_r) + p_0(S \cap S_r) = p_0(S \cap \hat{S}_r) + p_0(S \cap S_r, 1 \notin I_r) + p_0(S \cap S_r, 1 \in I_r) \\ &\leq p_0(\hat{S}_r) + p_0(1 \notin I_r) + p_0(S \cap S_r, 1 \in I_r) \\ &= \delta_r + p_0(S \cap S_r, 1 \in I_r) = \delta_r + p_0(1 \in I_r) p_0(S \cap S_r \mid 1 \in I_r), \end{aligned}$$

with $\delta_r = p_0(\hat{S}_r) + p_0(1 \notin I_r)$. Here, we can use the fact that neither 0 nor 1 communicates with a curious node before round r so $p_0(S \cap S_r \mid 1 \in I_r) = p_1(S \cap S_r \mid 0 \in I_r)$, and because the graph is complete we have $p_0(1 \in I_r) = p_1(0 \in I_r)$. Therefore,

$$p_0(S) = \delta_r + p_1(0 \in I_r) p_1(S \cap S_r \mid 0 \in I_r) = \delta_r + p_1(S \cap S_r, 0 \in I_r) \leq \delta_r + p_1(S \cap S_r) \leq \delta_r + p_1(S).$$

Since $p_0(\hat{S}_r) = p_1(\hat{S}_r)$ and $p_0(1 \notin I_r) = p_1(0 \notin I_r)$ we conclude that the same result holds if we invert the roles of 0 and 1. The probability $p_0(\hat{S}_r)$ that 0 or 1 transmit the information before r rounds is computed in Lemma F.2.

For the second part, we have to bound $p(1 \notin I_r)$. The idea of the proof is to split the analysis into 3 steps. At the beginning (until round R_1) there are very few informed nodes so their number doubles with high probability. Then, (until round R_2), the number of informed nodes does not exactly double at each round but still increases very fast (we use a Chernoff bound to evaluate the probability). Finally, between rounds R_2 and r, we consider that the number of informed nodes does not increase anymore but at each step, there is a high probability that node 1 gets the value.

More formally, we write:

$$\begin{aligned} p(1 \in I_r) &= \sum_{k=1}^{2^{R_1}} p(|I_{R_1}| = k) p(1 \in I_r \mid |I_{R_1}| = k) \geq p(|I_{R_1}| = 2^{R_1}) p(1 \in I_r \mid |I_{R_1}| = 2^{R_1}) \\ &\geq p(|I_{R_1}| = 2^{R_1}) p(|I_{R_2}| \geq M \mid |I_{R_1}| = 2^{R_1}) p(1 \in I_r \mid |I_{R_2}| \geq M). \end{aligned}$$

We can then conclude by using the Lemmas F.3, F.4 and F.5 to bound each quantity. □

The complexity of this result is due to the extensive use of Chernoff bounds. It is also necessary to define an appropriate sequence (η_k) associated with each bound. Optimal parameters r, R_1 and R_2 can be found by an extensive grid search. Approximations using the fact that n is big and $\frac{f}{n}$ is small give good indications on where to look for optimal values.

Note that when s is very close to 1, the δ given by Theorem F.1 is better than the one given by Theorem 5.2. This is due to the fact that the bound in Theorem 5.2 is not uniform in s. Theorem F.1 could be adapted to give slightly better results for s close to 1 in the case of non arbitrarily large graphs. However, we have already discussed the fact that taking s very close to 1 dramatically decreases privacy with no real boost in terms of speed (see Figure 4).

Figure 4 shows the value of the bounds for $n = 2^{16}$ and different fractions of curious nodes. We see that the upper bound we get is relatively tight (same order of magnitude as the lower bound). However, taking a value $s = 0.5$ yields better privacy than the lower bound of the protocol for $s = 1$.

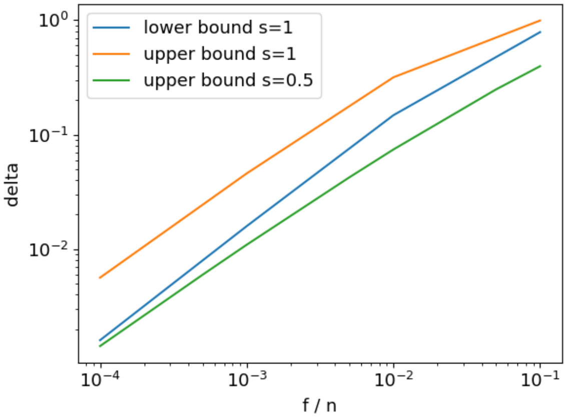

Fig. 4. Upper and lower bounds on δ for the standard gossip protocol ($s = 1$) for $\epsilon = 0$, and upper bound for $s = 0.5$ with respect to the fraction of curious nodes in the graph for a graph of size $n = 2^{16}$.

Lemma F.2. *Denote $\tilde{r}_n = \max(0, r - \lfloor \log_2(n - f) \rfloor)$. We have:*

$$p_0(\hat{S}_r) \leq 1 - \left(1 - \frac{f}{n}\right)^{r+\tilde{r}_n+1} \prod_{k=1}^{r-\tilde{r}_n} \left(1 - \frac{f}{n}\frac{2^k}{n-f}\right). \tag{13}$$

Remark F.1 (Finer result). *For an arbitrarily large graph and a fixed r, the right term goes to 0 and the optimal result is always for $r < \log n$, so the result simplifies to $p_0(\hat{S}_r) \leq 1 - \left(1 - \frac{f}{n}\right)^r$.*

Proof. We note $C(k)$ the set of nodes that communicated with a curious node at round k. Then:

$$\begin{aligned} p_0(\hat{S}_r) &= 1 - p_0(t_0 \geq r, t_1 \geq r) = 1 - p_0(t_0 \geq r)p(t_1 \geq r \mid t_0 \geq r) = 1 - \left(1 - \frac{f}{n}\right)^r p(t_1 \geq r \mid t_0 \geq r) \\ &= 1 - \left(1 - \frac{f}{n}\right)^r \prod_{k=1}^{r-1} p(1 \notin C(k) | \, t_0 \geq r) = 1 - \left(1 - \frac{f}{n}\right)^r \prod_{k=1}^{r-1} (1 - p(1 \in C(k) | \, t_0 \geq r)) \\ &= 1 - \left(1 - \frac{f}{n}\right)^r \prod_{k=1}^{r-1} \left(1 - \frac{f}{n} p(1 \in I_k | \, t_0 \geq r)\right) \leq 1 - \left(1 - \frac{f}{n}\right)^{r+\tilde{r}_n+1} \prod_{k=1}^{r-\tilde{r}_n} \left(1 - \frac{f}{n}\frac{2^k}{n-f}\right), \end{aligned}$$

where the last line comes from the fact that $p(1 \in I_k | \, t_0 \geq r) \leq \frac{|I_k|}{n-f}$ because we assumed that 0 did not communicate with a curious node so non curious nodes have higher chances of having been informed. □

Lemma F.3. *We have:*

$$p(|I_t| = 2^t) = \frac{(n-1)!}{(n-2^t)! n^{2^t-1}}.$$

Proof. We denote by m_k the event such that message k is not "lost" (i.e., hit a node that was not yet informed) and $m_{0,k}$ the event such that no message until k has been lost.

We have:

$$p(|I_t| = 2^t) = p\left(\cap_{k=1}^{2^t-1} m_k\right) = \prod_{k=1}^{2^t-1} p\left(m_k | m_{0,k-1}\right) = \prod_{k=1}^{2^t-1} \left(1 - \frac{n-k}{n}\right) = \frac{1}{n^{2^t-1}} \frac{(n-1)!}{(n-2^t)!}.$$

where the third line comes from the fact that the probability of losing a message is simply the number of informed nodes over the total number of nodes. $\square$

Lemma F.4. *We have:*

$$p(1 \in I_r | I_{R_2} \geq M) \geq 1 - (1 - \frac{M-1}{n})(1 - \frac{1}{n})^{(r-R_2)M}.$$

Proof.

$$\begin{aligned} p(1 \in I_r | I_{R_2} \geq M) &= 1 - p(1 \notin I_r | I_{R_2} \geq M) = 1 - p(1 \notin I_{R_2} | I_{R_2} \geq M) p(1 \notin I_r | I_{R_2} \geq M, 1 \notin I_{R_2}) \\ &\geq 1 - (1 - \frac{M-1}{n})(1 - \frac{1}{n})^{(r-R_2)M}. \end{aligned}$$

$\square$

Lemma F.5. *For $R_1 \leq R_2$ and a sequence $\eta_k > 0$, we have:*

$$p(|I_{R_2}| \geq M_{R_1 - R_2} \mid |I_{R_1}| = 2^{R_1}) \geq \prod_{k=R_1}^{R_2-1} (1 - f(\eta_k)^{\mu_k}),$$

with $M_{r_1} = 2^{R_1}$, $\mu_k = 2M_k \frac{M_k - 1}{n}$ and

$$M_{k+1} = 2M_k \left(1 - (1 + \eta_k) \frac{M_k - 1}{n}\right),$$

where $f(\eta) = \frac{e^{\eta}}{(1+\eta)^{(1+\eta)}}$.

Proof. The bound is obtained by recursively applying a Chernoff bound to control the number of messages that are lost at each step. We can write:

$$\begin{aligned} p(|I_{R_2}| \geq M_{R_2} \mid |I_{R_1}| = 2^{R_1}) &= p(|I_{R_2}| \geq M_{R_2} | |I_{R_2-1}| \geq M_{R_2-1}) p(|I_{R_2-1}| \geq M_{R_2-1} | |I_{R_1}| = 2^{R_1}) \\ &\geq p(|I_{R_2}| \geq M_{R_2} | |I_{R_2-1}| = M_{R_2-1}) p(|I_{R_2-1}| \geq M_{R_2-1} \mid |I_{R_1}| = 2^{R_1}) \\ &\geq \prod_{k=R_1}^{R_2-1} p(|I_{k+1}| \geq M_{k+1} \mid |I_k| = M_k). \end{aligned}$$

To obtain these inequalities, we used Bayes rule and the fact that for all M, r, $p(|I_{r+1}| \geq M_{r+1} \mid |I_r| \geq M_r) \geq p(|I_{r+1}| \geq M_{r+1} \mid |I_r| = M_r)$. This comes from the fact that at step r, exactly $|I_r|$ messages are sent. Consider two set of nodes of cardinal $A < B$. Then, if $M_{r+1} \leq B$ the inequality is directly true since $p(|I_{r+1}| \geq M_{r+1} \mid |I_r| = B) = 1$. Now consider that $M_{r+1} > B$. Consider $I_{r,k}$ the set of informed nodes at round r after k messages are sent during this round ($0 \leq k \leq |I_r|$). Then to have $|I_{r+1}| \geq M_{r+1}$, there must exist some $k^* \leq A$ such that $|I_{r,k^*}| = B$. In the second setting, this $k^* = 0$ so B messages are yet to be sent whereas only $A - k^*$ can still be sent if $|I_r| = A$. Since the probability of wasting a message only depends on the number of informed nodes at the time it was sent, and more messages will be sent in the second setting than in the first one with same initial conditions, $p(|I_{r+1}| \geq M_{r+1} \mid |I_r| = B) \geq p(|I_{r+1}| \geq M_{r+1} \mid |I_r| = A)$.

We will now compute $p(|I_{k+1}| \geq M_{k+1} \mid |I_k| = M_k)$. For that, we write W_r the number of messages that are wasted at step r. Then, we have for all M_{r+1}, M_r by using the fact that each informed node sends a message:

$$p(|I_{r+1}| \leq M_{r+1} \mid |I_r| = M_r) = p(|I_r| + |I_r| - W_r \leq M_{r+1} \mid |I_r| = M_r) = p(W_r \geq 2M_r - M_{r+1} \mid |I_r| = M_r).$$

Unfortunately, the event such that the message is lost depends on how many messages have been lost in the past. We note $V_{r,k}$ the random variable such that $V_{r,k} = 1$ with probability $p = 2\frac{M_r - 1}{n}$ and 0 otherwise, and $V_r = \sum_{k=1}^{M_r} V_{r,k}$. Then, for all $M \in \mathbb{R}$, we have $p(W_r \geq M) \leq p(V_r \geq M)$ because p is an upper bound on the probability of wasting a message no matter how many messages have been lost before. We fix $\eta_r > 0$ and $M_{r+1} = 2M_r - (1+\eta_r)p) = 2M_r \left(1 - (1+\eta)\frac{M_r-1}{n-1}\right)$. Applying a Chernoff bound to V_r leads to:

$$p(W_r \geq 2M_r - M_{r+1} \mid |I_r| = M_r) \leq p(V_r \geq 2M_r - M_{r+1} \mid |I_r| = M_r) = p(V_r \geq (1+\eta_r)\mathbb{E}[V_r]) \leq f(\eta_r)^{\mathbb{E}[V_r]},$$

where $f(\eta) = \frac{e^\eta}{(1+\eta)^{1+\eta}}$ results from the use of the Chernoff bound. □

G DISSEMINATION SPEED OF FAST PRIVATE GOSSIP

In this section, we prove Theorem 5.4 and discuss extensions of this result to the asynchronous setting.

G.1 Proof of Theorem 5.4

Proof of Theorem 5.4. This proof builds on ideas from Lemma 15 in [39] and uses Azuma inequality [35], for example to obtain Equation 16. We start by showing that if more than $k(s)$ nodes are informed at a given time, then with very high probability the number of informed nodes will never drop below this fraction. Therefore, a number of messages proportional to the size of the graph will be sent at each round. The condition on s for this to happen is written in Equation 18. More formally, we fix $s \in (0,1]$ and denote by A_t the number of nodes that are active at round t, which is such that $A_t = \alpha_t n$. Then, we note

$$f : \alpha \longrightarrow 1 - p_u(\alpha)(1-\alpha s), \tag{14}$$

where $p_u(\alpha) = (1-\frac{1}{n})^{\alpha n}$. Note that $f(\alpha)$ can be rewritten $f(\alpha) = \frac{1}{n}\mathbb{E}[A_{t+1} - A_t | A_t = \alpha n]$. As a matter of fact, for each node, the probability of getting the message is exactly $1 - p_u(\alpha)$ so $n(1-p_u(\alpha))$ nodes get the message in expectation. The rest of the active nodes at the following round is made of the nodes that were active, did not receive the message and did not deactivate, which represents a portion $n\alpha p_u(\alpha)s$ of the nodes. Then, one can see that the function f is simply the sum of these 2 terms. We show by using that $(1-x)^y \leq e^{-xy} \leq 1 - xy + \frac{x^2y^2}{2}$ that for $\alpha \leq \alpha_s = \frac{s}{1+2s}$, we have:

$$f(\alpha) \geq \left(1 + \frac{s}{2}\right)\alpha. \tag{15}$$

Then, we follow the same steps as in Lemma 15 in [39]. We call A_t the number of active nodes at round t, and $A_{t,m}$ the number of active nodes at round t after m messages have been sent (so during the round). Then, we can define $X_i = A_{t,i+1} - A_{t,i}$. $A_{t,i+1}$ only depends on $A_{t,i}$ and so does X_i:

$$X_i = \begin{cases} 1 & \text{with proba} \quad s(1 - \frac{|A_{t,i}|}{n}) \\ -1 & \text{with proba} \quad (1-s)\frac{|A_{t,i}|-1}{n} \\ 0 & \text{otherwise} \end{cases}$$

Then, we define the martingale $Z_i = \mathbb{E}[\sum_{i=1}^{A_t} X_i | X_1, \cdots, X_i, A_t]$. This allows us to write $A_{t+1} - nf(\alpha) = Z_{A_t} - Z_0$. If we call $S_{k,t} = \sum_{i=k}^{A_t} X_i$ then for any $d \in \{-1, 0, 1\}$:

$$\mathbb{E}[S_{1,t} | X_1, , X_i, X_{i+1} = 1, A_t] \geq \mathbb{E}[S_{1,t} | X_1, \cdots, X_i, X_{i+1} = d, A_t] \geq \mathbb{E}[S_{1,t} | X_1, \cdots, X_i, X_{i+1} = -1, A_t],$$

because the distribution of X_i only depends on $A_{t,i}$. Therefore, $|Z_{i+1} - Z_i| \leq (1 + \mathbb{E}[S_{i+1,t} | A_t + 1]) - (\mathbb{E}[S_{i+1,t} | A_t - 1] - 1)] \leq 2$. Azuma's inequality then gives:

$$p\left(A_{t+1} - nf(\frac{A_t}{n})\right) \leq -\lambda A_t | A_t = k) \leq e^{-\frac{\lambda^2 k}{8}}. \tag{16}$$

We also have that $p(A_{t+1} < k | A_t \geq k) \leq p(A_{t+1} \leq k | A_t = k)$. Then, for any $\alpha \leq \alpha_s$ we have that $p(A_{t+1} < k | A_t \geq k) \leq p(A_{t+1} - nf(\frac{A_t}{n}) \leq -\frac{s}{2} A_t | A_t = k)$ by using Equation 15. We can then bound this expression by using Equation 16 with $\lambda = \frac{s}{2}$. Therefore, we need that:

$$\alpha \leq \alpha_s. \tag{17}$$

Denoting by $N_{k,j}$ the number of messages sent between rounds k and j, we can decompose over $C\alpha^{-1} \log n$ rounds so that if m is such that there are at least α active nodes at round m then:

$$p(N_{m, m+C\alpha^{-1} \log n} \geq Cn \log n) \geq (1 - e^{-\frac{s^2 \alpha n}{32}})^{C\alpha^{-1} \log n},$$

because it is equal to the probability that the fraction of active nodes never goes below α for $C\alpha^{-1} \log n$ rounds. Therefore, if

$$s^2 \geq \frac{32}{\alpha n} \log \frac{3Cn \log n}{\alpha}, \tag{18}$$

then $p(N_{m, m+C\alpha^{-1} \log n} \geq Cn \log n) \geq 1 - \frac{1}{3n}$.

Equation 18 gives a lower bound on the value of α. Note that this lower bound goes to 0 as n grows so in particular, Equation 18 is satisfied for $\alpha = \alpha_s$ if n is large enough. It now remains to show that such a fraction α of active nodes can be reached in logarithmic time. Usual gossip analysis takes advantage of the exponential growth of the informed nodes during early rounds for which no collision occur. We have to adapt the analysis to the fact that nodes may stop communicating with some probability and split the analysis into two phases.

In the rest of the proof, we prove that a constant fraction of the nodes (independent of n) can be reached with a logarithmic number of rounds. We first analyze how long it takes to go from $O(\log n)$ to $O(n)$ active nodes and then from 1 to $O(\log n)$.

Equation 15 along with Equation 16 with $\lambda = \frac{s}{4}$ give that as long as $A_{t_0}(1 + \frac{s}{4})^t \leq \alpha_s n$ then

$$p\left(A_{t+t_0+1} \geq A_{t_0}(1 + \frac{s}{4})^{t+1} | A_t = A_{t_0}(1 + \frac{s}{4})^t\right) \geq 1 - e^{-\frac{\alpha n s^2}{128}} \tag{19}$$

for any $t \geq t_0$ such that $A_{t_0}\left(1 + \frac{c}{2}\right)^t \leq n\alpha_s$. Therefore, if we do this for all $t \leq t_\alpha = \frac{\log(\alpha n)}{\log(1+\frac{s}{4})}$ rounds (so for a logarithmic number of rounds) then $p\left(A_{t_\alpha + t_0} \geq n\alpha | A_{t_0}\right) \geq \left(1 - e^{-\frac{A_{t_0} s^2}{128}}\right)^{t_\alpha}$ because in this case, $A_t \geq A_{t_0}$ for $t \geq t_0$. Therefore, if

$$A_{t_0} \geq -\frac{128}{s^2} \log\left(1 - \left(1 - \frac{1}{3n}\right)^{\frac{1}{t_\alpha}}\right), \tag{20}$$

then $p(A_{t_\alpha + t_0} \geq n\alpha) \geq 1 - \frac{1}{3n}$. If we use the fact that $(1-x)^y \leq e^{-xy} \leq 1 - xy + \frac{x^2 y^2}{2}$ to simplify Equation 20, we can show that if A_{t_0} satisfies:

$$A_{t_0} \geq \frac{128}{s^2}\left(\log n + \log(3t_\alpha) - \log\left(1 - \frac{1}{6nt_\alpha}\right)\right), \tag{21}$$

then it also satisfies Equation 20. Since the terms on the right hand side are dominated by $\log n$, for n large enough, a sufficient condition for Equation 20 to hold is:

$$A_{t_0} \geq \frac{256}{s^2} \log n. \tag{22}$$

It only remains to prove that a logarithmic number of nodes can be reached in logarithmic time. For this, we use again Azuma inequality but we start from the very beginning of the protocol ($|A_0| = 1$) and for a fixed sequence of m messages. This time, we write $S_n = \sum_{i=1}^{m} X_i$ with the exact same notations, and by calling $A_{0,m}$ the number of actives nodes after m messages (without taking rounds into account) then

$p(A_{0,m} - a \leq -\lambda) \leq e^{-\frac{\lambda^2}{8m}}$ for any $a \leq \mathbb{E}[S_n]$. Then, we denote E_i the event such that $X_i \geq 0$ for all i and write that $\mathbb{E}[S_n] \geq p(E_i)\mathbb{E}[S_n|E_i]$. Considering that $A_{0,m} \leq m$, we can write that $\mathbb{E}[S_n] \geq (1-(1-s)\frac{m}{n})^m ms(1-\frac{m}{n})$. Therefore, we can apply Azuma inequality with $\lambda = ms\left(\frac{1}{2} - \frac{m}{n}\left[(1-s)m-1\right]\right)$, which yields:

$$p(A_{0,m} \leq \frac{ms}{2}) \leq e^{-\frac{ms^2}{8}\left(\frac{1}{2}-\frac{m}{n}[(1-s)m-1]\right)^2}. \tag{23}$$

The number of messages sent during rounds 1 to t_0 is at least equal to $m \geq t_0$. We set $t_0 = \frac{512}{s^3}\log(3n)$, and since for n large enough we have $\frac{m}{n}\left[(1-s)m-1\right] \leq \frac{1}{4}$, then

$$p\left(A_{t_0} \geq \frac{256}{s^2}\log(n)\right) \geq 1 - e^{-\frac{\tilde{m}s^2}{128}} \geq 1 - \frac{1}{3n}. \tag{24}$$

We conclude the proof by noting that

$$p\left(N_{0,\, t_0+t_\alpha+C\alpha^{-1}\log n} \geq Cn\log n\right) \geq p\left(A_{t_0} \geq \frac{256}{s^2}\log n\right) p\left(A_{t_\alpha+t_0} \geq n\alpha | A_{t_0} \geq \frac{256}{s^2}\log n\right)$$

$$p\left(N_{t_\alpha+t_0,\, t_\alpha+t_0+C\alpha^{-1}\log n} \geq Cn\log n | A_{n_\alpha} \geq \alpha\right) \geq \left(1 - \frac{1}{3n}\right)^3 \geq 1 - \frac{1}{n}.$$

The number of rounds is logarithmic since both t_0 and t_α depend logarithmically on n. □

G.2 Extension to the asynchronous setting

The first part of the proof directly extends to the asynchronous algorithm by simply considering slices of time during which a set of αn nodes send αn messages, which essentially means constant time. Then, we consider a logarithmic number of slices. The phase from 1 to $O(\log n)$ active nodes requires sending a logarithmic number of messages and can thus be done in logarithmic time. Finally, phase 2 (going from $O(\log n)$ to $O(n)$ active nodes) consists in evaluating a logarithmic number of rounds during which a logarithmic number of nodes are active. Again, the only important thing is the number of messages sent (and not which node sent them) so using constant time intervals ensures that enough messages are sent between each pseudo-rounds with high probability.

Therefore, it is possible to prove a statement very similar to that of Theorem 5.4 in the asynchronous setting, where the notion of rounds is replaced by constant time intervals. We omit the exact details of this alternative formulation.