Catch Me If You Can Privacy-Preserving Dissemination in Micro-Blogging

George Giakkoupis, Rachid Guerraoui, Arnaud Jégou, Anne-Marie Kermarrec, Nupur Mittal

Technical Report: Catch Me If You Can Privacy-Preserving Dissemination in Micro-Blogging

George Giakkoupis
INRIA Rennes, France

Rachid Guerraoui
EPFL, Swizerland

Arnaud Jégou
INRIA Rennes, France

Anne-Marie Kermarrec
INRIA Rennes, France

Nupur Mittal
INRIA Rennes, France

ABSTRACT

Online micro-blogging services and social networks, as exemplified by Twitter and Facebook, have emerged as an important means of disseminating information quickly and at large scale. A standard mechanism in micro-blogging that allows for interesting content to reach a wider audience is that of *reposting* (i.e., *retweeting* in Twitter, or *sharing* in Facebook) of content initially posted by another user. Motivated by recent events in which users were prosecuted merely for reposting anti-government information, we present RIPOSTE, a randomized reposting scheme that provides privacy guarantees against such charges.

The idea is that if the user likes a post, RIPOSTE will repost it only with some (carefully chosen) probability; and if the user does not like it, RIPOSTE may still repost it with a slightly smaller probability. These probabilities are computed for each user as a function of the number of connections of the user in the network, and the extent to which the post has already reached those connections. The choice of these probabilities is based on results for branching processes, and ensures that interesting posts (liked by a large fraction of users) are likely to disseminate widely, whereas uninteresting posts (or spam) do not spread. RIPOSTE is executed locally at the user, thus the user's opinion on the post is not communicated to the micro-blogging server.

We quantify RIPOSTE's ability to protect users in terms of differential privacy and provide analytical bounds on the dissemination of posts. We also report on experimental results based on topologies of real networks, including Twitter, Facebook, Renren, Google+ and LiveJournal.

1. INTRODUCTION

Micro-blogging platforms and online social networks are becoming a main source of news dissemination in the world. The open nature of such platforms provides unique opportunities for so-called *web activists* to denounce despotic activities and corrupted regimes [1, 15].[1] Not surprisingly, secret (and not so secret) police services have been carefully following Internet traffic, and there has been an increasing number of arrests of such web activists. In many of these cases, people have been convicted merely for *contributing* to the dissemination of a post initially posted by someone else, a.k.a. *reposting* the post.[2,3]

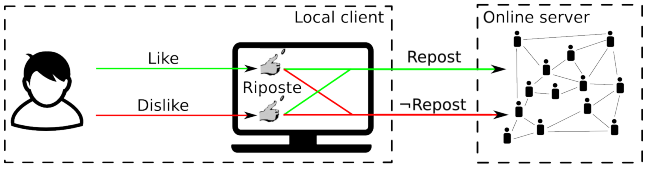


Figure 1: In RIPOSTE, the user's opinion is locally randomized, and only the output of RIPOSTE is exposed to the online service. The ¬Repost arrow is represented for illustration purposes and is not an explicit action of RIPOSTE.

The motivation of this work is to devise a dissemination protocol for a micro-blogging or other social networks that would protect its users from such charges and hence conviction. Consider, for example, the setting where the content to disseminate is an article or a video from a media edited in some country X, and a link to that content was posted to the social network. We then seek to protect the privacy of an activist who wishes to contribute to the spread of that post by reposting it to her followers/friends in a country Y, e.g., *retweet* it on Tweeter, or *share* it on Facebook. The *attacker* here is any individual/group/external agency trying to identify the users involved in the spread of a specific post, by observing the dissemination, or even accessing the servers of the network.

We present RIPOSTE,[4] a privacy-preserving reposting protocol. RIPOSTE achieves privacy by randomizing the

[1] http://www.huffingtonpost.com/2011/02/11/egypt-facebook-revolution-wael-ghonim_n_822078.html

[2] http://newsfeed.time.com/2012/11/19/indian-woman-arrested-over-facebook-like/

[3] http://www.npr.org/2011/12/01/142998183/in-south-korea-old-law-leads-to-new-crackdown

[4] Riposte is the French word for counter-attack.

user's repost actions: If a user likes a post, RIPOSTE will repost it with some (carefully chosen) probability; otherwise, RIPOSTE may still repost it, but with a smaller probability, equally carefully chosen. This decision is made locally, i.e., by RIPOSTE's code at the user's browser, and is not communicated to the central service (Figure 1). This way, it is ensured that when an attacker observes a repost action, she cannot know with certainty whether it reflects the user's opinion, or whether RIPOSTE enforced that action, thus providing *plausible deniability* to the user.

The main challenge in our approach stems from the trade-off between guaranteeing privacy on one hand, and ensuring that the spread of a post reflects the overall opinion of users, on the other. The latter means that posts liked by many users should spread a lot, whereas non-interesting posts (or spam) should not spread. The requirement for privacy suggests that the two probabilities RIPOSTE uses to decide for the repost when the user likes/does not like the post should be close to each other. Whereas the requirement that the size of dissemination should reflect users' opinion suggests the opposite: the two probabilities cannot be "too" close to each other, otherwise the user's opinion is not taken into account.

RIPOSTE computes these probabilities, at each user for each post, based on the number of connections of the user in the micro-blogging network, and the extend to which the post has already reached those connections. Our choice of these probabilities draws from the theory of branching processes [3] and ensures the following properties.

- *Privacy:* By observing the reposts of a user, an attacker cannot tell (with sufficient confidence) if that user likes the post or not.
- *Dissemination:* Posts liked by a large fraction of users are likely to disseminate to a large fraction of the network, whereas posts that only few people like (or spam posts) do not spread.

RIPOSTE shall be seen as a novel dissemination scheme that seeks to protect its users with respect to the privacy of their repost actions. We should stress that RIPOSTE diverges from the standard repost practice: A user may not repost all the content she would like to, and may repost content she does not like. This change in the dissemination semantics is the price to pay for users to see their privacy preserved. Also, we believe that the social nature of such platforms can be leveraged by users themselves. They can use their real relationship with followers or friends to implement a human filter on the reposting actions: Alice might realize that the reposting of X does not resemble Bob's habits and therefore be the consequence of running RIPOSTE.[5] The recent success of social applications such as Snapchat or Whisper shows an increased concern of users for privacy and therefore their willingness to change their habits and adopt new privacy preserving schemes.

This paper makes the following contributions:

1. We present a simple yet powerful privacy-preserving reposting protocol, called RIPOSTE.
2. We provide a theoretical analysis of RIPOSTE. We express the privacy properties of RIPOSTE in terms of ε-differential privacy [6]. For the dissemination, we provide analytical bounds on the spread of posts (under certain modeling assumptions).
3. We complement the analysis by extensive experimental results on real micro-blogging and social networks topologies, from Twitter, Facebook, LiveJournal, Google+ and Renren.

The rest of the paper is organized as follows. In Section 2 we describe RIPOSTE and discuss its properties. In Section 3 we analyze RIPOSTE's privacy guarantees. Section 4 provides an analysis of the dissemination of posts. In Section 5 we describe our experimental setup followed by the experimental results in Section 6. Section 7 discusses some implementation details. Section 8 gives an overview of related work, and Section 9 concludes the paper.

2. THE RIPOSTE PROTOCOL

In this section we describe RIPOSTE and discuss its properties.

2.1 Protocol Description

RIPOSTE is a randomized dissemination scheme with a repost mechanism.We assume that each user in this system has a set of *followers* (or friends), and when a user posts or reposts some content, this content is sent to *all* its followers.

RIPOSTE decides for each post the user receives, whether to repost it or not. This decision is local to the user: it is made by RIPOSTE's code at the user's machine. RIPOSTE solicits the opinion of the user about the post, i.e., if the user likes it or not, and then decides whether or not to repost it. If it decides to repost, then a repost request is communicated to the system server. We stress that RIPOSTE does not reveal the user's opinion on the post, to the server or to any other user.

RIPOSTE does not repost every post that the user likes, and it may repost content that the user does not like. If the user likes a post then it is reposted with some carefully computed probability; and if the user does not like it, it is reposted with another carefully chosen probability, which is slightly smaller. These two probabilities

[5]Note that such a human filter is first difficult to reproduce for an attacker and second can never lead to a clear evidence.

are computed based on (1) the number of the user's followers that have not received the post yet; and (2) two global parameters of RIPOSTE, λ and δ, called *spreading* and *blocking* factor respectively . These parameters are fixed and are the same for all users—they are not chosen by each user. The values of these parameters impact the dissemination of posts, and determine also the privacy guarantees the protocol provides, as explained in detail later. Both parameters are real numbers with

$$0 < \delta < 1 \quad \text{and} \quad \lambda > 1.$$

RIPOSTE decides whether or not to repost as follows: Consider a user u who receives a new post, and suppose that $s > 0$ of its followers have not yet received the post at the time. If u likes the post, it is reposted with probability

$$r_{\text{like}}(s) := \begin{cases} \lambda/s, & \text{if } s \geq \lambda + \delta, \\ 1 - \frac{\delta(s-\delta)}{\lambda s}, & \text{if } 0 < s < \lambda + \delta; \end{cases}$$

while if u does not like the post, it is reposted with probability

$$r_{\text{dis}}(s) := \delta/s.$$

These random decisions are independent for each user and for each post.

Remark 1. The second formula for $r_{\text{like}}(s)$, for the case of small s, will be justified when we analyse the privacy properties of the protocol, in Section 3. Until then we can assume the following simpler definition for all s,

$$r_{\text{like}}(s) := \min\{\lambda/s,\, 1\}.$$

We assume that RIPOSTE knows the number s of followers that have not yet received a given post. This information is available and can be easily obtained in many existing platforms, including Twitter. Nevertheless, at the end of this section we provide a simple variant of RIPOSTE that uses the *total number* of followers in place of s; we show that this variant has very similar properties as the RIPOSTE protocol presented above.

We give now an informal overview of RIPOSTE's properties, and provide some intuitive explanation. The formal analysis will be provided in the next sections. First we discuss privacy properties, and then the guarantees provided with respect to the dissemination of posts.

2.2 Privacy Overview

With respect to privacy, our goal is that an attacker, who observes all the reposts of the users, cannot tell whether or not a particular user indeed likes a given post—unless the post originated at that user.[6]

More specifically, observing that the post was reposted (or not) from the user, may change only slightly the prior knowledge that the attacker has about the user's opinion on the post (e.g., by observing previous posts by that user). In other words, if before having observed reposting actions, the attacker believed that with probability q the user likes the post, then learning whether or not the post was reposted from the user may change this probability to $\hat{q}$, such that this new probability $\hat{q}$ will be close to q.

How close these two probabilities are in RIPOSTE depends on the choice of parameters δ and λ. Since $r_{\text{like}}(s)$ and $r_{\text{dis}}(s)$ are respectively (at most) λ/s and δ/s, it is intuitive that the closer the values of δ and λ, the less information the attacker gains about the user's opinion by observing his reposts, thus the better the privacy guarantee. Consider, for example, the extreme case in which $\delta = 0$ and $\lambda = n$ (n being the size of the network). I.e., every post that the user likes is reposted, and no other posts are reposted, as, e.g., in Twitter. In this case, if the post is reposted (resp. not reposted), an attacker can conclude with certainty that the user likes it (resp. does not like it). On the other hand, in the opposite extreme case in which $\delta = \lambda = 1$, reposting reveals no information at all. However, choosing δ and λ to be equal is not very useful, as the dissemination is then independent of the user's opinion.

In Section 3, we show that our protocol ensures ϵ-differential privacy for $\epsilon = \ln(\lambda/\delta)$. Thus, the closer the ratio λ/δ to one, the better the achieved privacy. We will see that if q is the probability for a given user u to like the post, as perceived by the attacker, and $\hat{q}$ is the same probability after observing that the post was reposted from u then

$$\hat{q} = \frac{q}{q + (1-q)\delta/\lambda}.$$

E.g., for typical parameter values, $\delta = 3/4$ and $\lambda = 3$, we have that if $q = 0.01$ then $\hat{q} \approx 0.04$; if $q = 0.1$ then $\hat{q} \approx 0.3$; and if $q = 0.9$ then $\hat{q} \approx 0.97$.

2.3 Dissemination Overview

With respect to dissemination, our goal is that the fraction of users reached by a post should reflect the user's overall opinion on the post. In particular, we want that interesting posts, i.e., posts many users like, to typically spread to a lot of users, while not interesting posts should not reach many users. The main difficulty in this goal lies in the fact that the users' opinions on a post are not known in advance, and thus we cannot tell beforehand if a post is interesting or not. Indeed RIPOSTE does not rely on any prior knowledge of how interesting the post may be. Furthermore, we are constrained by the requirement that the users' opinions must remain private.

The idea behind RIPOSTE's dissemination scheme is simple, and draws from the theory of branching processes [3]. A branching process is a random process

[6] RIPOSTE does not protect the privacy of the source of a post. This is a reasonable assumption since a lot of posts originate from media.

modeling a population, which starts with one or more individuals, and at each step a single individual produces zero of more offsprings and then dies. In the most basic model, the number of offsprings of an individual follows a fixed probability distribution that does not vary between individuals. Let μ be the expected number of offsprings of an individual. It is a well-known fact then that if $\mu < 1$ the population dies quickly, while if $\mu > 1$ it survives forever with some positive probability.

To see the connection of RISPOSTE to branching processes, let us compute the expected number of new users that learn the post from a given user u who has received the post. Let $s > 0$ be the number of u's follower who have not already received the post from some other user before u'action is decided.

If u likes the post, then RISPOSTE will repost it with probability $r_{\text{like}}(s)$, and thus, with this probability, s new users will learn the post. It follows that $r_{\text{like}}(s) \cdot s$ users will learn it *in expectation*. Further, if $s \geq \lambda + \delta$, then $r_{\text{like}}(s) = \lambda/s$ and thus $r_{\text{like}}(s) \cdot s = \lambda > 1$ new users receive the post from u in expectation.

By a similar reasoning, if the user does not like the post, then RISPOSTE forwards it to at most $\delta < 1$ other users in expectation. Depending now on the *popularity* of the post, that is, the fraction of users who like it,[7] we have more than 1 new users in expectation that learn the post from the average user if the post is sufficiently popular, or fewer than 1 if the post is less popular. An analysis then, using standard results from branching processes, shows that if the expected number of users to receive the post from the average user is even slightly larger than 1 then the post is likely to reach a significant fraction of the network; while if this expectation is even slightly smaller than 1 then the post is unlikely to spread.

More precisely, in Section 4 we prove the following bounds on dissemination, under some independence assumption on the opinion of different users. For any parameters $0 < \delta < 1 < \lambda$, there is a *popularity threshold*

$$p^* = \frac{1-\delta}{\lambda-\delta},$$

such that:

(a) A post with popularity smaller than p^* (*unpopular* post), spreads to an expected number of users that is *at most* a constant factor larger than the number of followers of the user who sent the original post. This result holds for any network topology.

(b) A post with popularity greater than p^* (*popular* post), spreads to *at least* some constant fraction of the network, with at least a constant probability, provided that the number of followers of the user who sent the original post is not much smaller than the average. This result is shown for a specific random graph model for social networks.

[7]We stress that the popularity of posts is not known in advance.

2.4 DB-RISPOSTE: Counting All Followers

RISPOSTE needs to know the number of the user's followers who have not yet received the post. In most platforms, this information is readily available, as the default setting is that a user can access the list of posts each of its followers has received. However, followers may have the option to hide that information.

DB-RISPOSTE (Degree-Based Riposte) is a simple variant that accounts for these concerns. This protocol is identical to RISPOSTE, except that in the definition of probabilities r_{like} and r_{dis}, we replace the number of followers s that have not already received the post, by the *total number* d of followers.

DB-RISPOSTE provides the same privacy guarantees as RISPOSTE, and similar dissemination guarantees except that the spread achieved for popular posts may be smaller by some small factor.

We will also provide an analysis and experimental evaluation of this variant of RISPOSTE.

3. ANALYSIS OF PRIVACY

In this section we show that RISPOSTE is ε-differentially private.

3.1 Differential Privacy

We start by recalling the definition of an ε-differentially private algorithm. Suppose we have a randomized algorithm A that takes as input a collection of m values, $x_1, \ldots, x_m$ from some domain D_{in}, and returns a value from some domain D_{out}. We denote by $A(x_1, \ldots, x_m)$ the output value of the algorithm. Since the algorithm is randomized, it may give different outputs for the same input. Thus, for a fixed input $x_1, \ldots, x_m$, the output $A(x_1, \ldots, x_m)$ is a random variable, with some distribution over D_{out}. Suppose now that the input to A is not known to us (is private), and by observing just the output of A we want to find out the value of some of the inputs. More generally, we may have some information about the input, i.e., a distribution over the possible combinations of input values, and we want, by observing A's output, to improve this information, i.e., obtain a distribution that is closer to the true input values. We can quantify the extend to which this is possible in terms of ε-differential privacy. Algorithm A is ε-differentially private if changing exactly one of it inputs $x_1, \ldots, x_m$ changes the distribution of the output only by at most an e^{ε} factor.

Definition 1 (ε-differential privacy)**.** *A randomized algorithm A with inputs $x_1, \ldots, x_m$ from some finite domain D_{in} and output $A(x_1, \ldots, x_m)$ on some domain*

D_{out}, is ε-differentially private if for any two set of inputs $x_1, \ldots, x_m$ and $x'_1, \ldots, x'_m$ that differ in exactly one value, and for any subset of outputs $S \subseteq D_{\text{out}}$,

$$\Pr\big(A(x_1, \ldots, x_m) \in S\big) \leq e^{\varepsilon} \cdot \Pr\big(A(x'_1, \ldots x'_m) \in S\big).$$

In our setting, algorithm A is the reposting protocol, which takes a single binary input: the opinion of the user, and has a binary output: repost or not-repost.

Theorem 1. *Both* RIPOSTE *and* DB-RIPOSTE *are ε-differentially private for $\varepsilon = \ln(\lambda/\delta)$.*

Proof. Below we use k to denote the number of followers of the user considered if DB-RIPOSTE is used, or the number of followers that have not received the post yet if RIPOSTE is used.

We must show that (1) the probability of reposting when the user likes the post is no larger than $e^{\varepsilon} = \lambda/\delta$ times the probability of reposting when the user does not like the post, i.e.,

$$r_{\text{like}}(k) \leq (\lambda/\delta) \cdot r_{\text{dis}}(k); \tag{1}$$

and (2) the probability of not reposting when the user does not like the post is no larger than λ/δ times the probability of not reposting when the user likes the post,

$$1 - r_{\text{dis}}(k) \leq (\lambda/\delta) \cdot (1 - r_{\text{like}}(k)). \tag{2}$$

Ineq. (1) holds because: $r_{\text{like}}(k) = \delta/k$ for all $k > 0$, $r_{\text{dis}}(k) = \lambda/k$ if $k \geq \lambda + \delta$, and $r_{\text{dis}}(k) < \lambda/k$ if $0 < k < \lambda + \delta$. For Ineq. (2), we have for $0 < k < \lambda + \delta$,

$$(\lambda/\delta) \cdot (1 - r_{\text{like}}(k)) = (\lambda/\delta) \cdot \frac{\delta(k-\delta)}{\lambda k} = 1 - r_{\text{dis}}(k);$$

and for $k \geq \lambda + \delta$, we must show

$$\begin{aligned} & (\lambda/\delta)(1 - \lambda/k) \geq 1 - \delta/k \\ \Leftrightarrow \quad & \lambda(1 - \lambda/k) \geq \delta(1 - \delta/k) \\ \Leftrightarrow \quad & \lambda - \delta \geq (\lambda^2 - \delta^2)/k \\ \Leftrightarrow \quad & 1 \geq (\lambda + \delta)/k \\ \Leftrightarrow \quad & k \geq \lambda + \delta, \end{aligned}$$

which holds. □

We explain now what this property implies for the information an attacker can gain for the opinion of a user, by learning whether or not it reposted a post.

Suppose the attacker has some prior knowledge on the opinion of the user on the post, i.e., the attacker believes that with some probability q the user likes the post. We argue that the new probability $\hat{q}$ with which the attacker believes that the user likes the post, after it has learned whether or not the user has reposted, is

$$\frac{q}{q + (1-q)(\lambda/\delta)} \leq \hat{q} \leq \frac{q}{q + (1-q)(\delta/\lambda)}. \tag{3}$$

Let $\mathcal{L}$ be the event that the user likes the post, and R the binary random variable that is 1 if the user reposts and 0 if it does not repost. Then the probabilities q and $\hat{q}$ can be expressed as $q = \Pr(\mathcal{L})$ and $\hat{q} = \Pr(\mathcal{L} \mid R)$. Suppose that the user has at least one follower who does not know the post yet and thus the probability of repost is not zero. From Bayes' Rule,

$$\begin{aligned} \Pr(\mathcal{L} \mid R) &= \frac{\Pr(R \mid \mathcal{L}) \cdot \Pr(\mathcal{L})}{\Pr(R \mid \mathcal{L}) \Pr(\mathcal{L}) + \Pr(R \mid \neg\mathcal{L}) \Pr(\neg\mathcal{L})} \\ &= \frac{\Pr(\mathcal{L})}{\Pr(\mathcal{L}) + \frac{\Pr(R|\neg\mathcal{L})}{\Pr(R|\mathcal{L})} \cdot \Pr(\neg\mathcal{L})}. \end{aligned} \tag{4}$$

Also from Theorem 1, we have $\delta/\lambda \leq \frac{\Pr(R|\neg\mathcal{L})}{\Pr(R|\mathcal{L})} \leq \lambda/\delta$. Applying this to the expression for $\Pr(\mathcal{L} \mid R)$ above proves Ineq. (3).

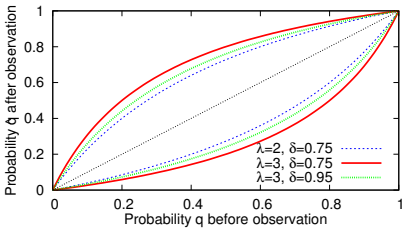


Figure 2: Illustration of Ineq. (3) that provides upper and lower bounds on $\hat{q}$ in terms of q. Probability $\hat{q}$ diverges more from q as λ increases or δ decreases.

3.2 Correlated Posts

We have seen that the observance of an individual post reposted by a user only marginally changes the attacker's confidence about the user's opinion on that post. However, if a user receives several *correlated* posts, e.g., posts supporting a particular anti-government view, then the reposts by the user can be used to compute an accurate estimate of whether the user supports the view. The larger the set of correlated posts, the higher the accuracy of the estimation. In this setting, we argue that the attacker cannot identify a large set of users such that with significant probability all the users in the set support the view.

Suppose there is a set of t posts on some anti-government view. Considering the attacker's point of view, we call a user *guilty* if she likes a post and *innocent* otherwise. The goal for the attacker is to identify the largest subset L of users that like at least one of those t posts, such that the probability that all users in L are guilty is at least $1/2$. This goal makes sense if, for example, the government wants to charge the largest possible number of users, without charging innocent users. For simplicity we will assume that each user likes either all the t posts or none. We also assume that the expected fraction p of users that like the posts is known, but no additional information is available in advance about users. So, we can assume that each user likes the posts with probability p independently of the other users. We

are interested in the size $\ell = |L|$ of L. For simplicity, we assume DB-Riposte is used, but similar reasoning applies also to Riposte.

We observe the dissemination of the t anti-government posts. For a user u with degree d, who has received i of those posts and has reposted r, the probability of disliking the posts can be computed similar to Equation 4, which gives the probability for liking, and is given by:

$$\theta(i,r,d) := \frac{1-p}{(1-p) + \frac{\Pr(B(i,\delta/d)=r)}{\Pr(B(i,\lambda/d)=r)} \cdot p},$$

where $B(i,q)$ is the binomial distribution counting the number of successes among i independent trials with success probability q. Substituting the definition for the binomial distribution yields:

$$\theta(i,r,d) = \frac{1-p}{(1-p) + \left(\frac{\lambda}{\delta}\right)^r \left(\frac{d-\lambda}{d-\delta}\right)^{i-r} \cdot p}.$$

It is easy to verify that the following strategy for choosing the elements of L maximizes the size of L. Let θ_j be the value for the j-the user in a list of users with increasing values of θ. We choose L to be the set containing the first ℓ users in the list, for ℓ the largest index such that

$$1 - \prod_{j \le \ell} (1-\theta_j) < 1/2.$$

The quantity on the left is the probability that not all users like the posts, and is approximately $\sum_{j \le \ell} \theta_j$.

We evaluate ℓ numerically in the following setting. We assume that the attacker fixes in advance a set of m users, then each of these users is given all the t posts, and the attacker must choose a subset L of these m users after it observes their reposts.

Table 1 shows the average (over 10000 runs) of the maximum number ℓ of users that can be identified as guilty. We chose the fraction p of guilty users to be 1%, 10% and 30%, the number of users to be $m = 100, 1000, 10000, 100000$, and the number of posts to be $t = 5, 10, 20$. We assume $\delta = 0.75$, $\lambda = 3$ and $d = 40$. For example, in a group of 1000 users, 100 of which are guilty in expectation ($p = 10\%$), and for a set of 10 correlated posts, an attacker can only identify a set of at most 3 guilty users on average.

As expected, the number of users that can be convicted increases with the number of correlated posts, as the more a user expresses her opinion about a topic, the easier it becomes to estimate her opinion and hence convict her. Overall, the number of users that can be convicted is very small with respect to the number of guilty users.

4. ANALYSIS OF DISSEMINATION

In this section we provide analytical bounds on the dissemination of posts using Riposte.

p	users	guilty	convicted		
			5 posts	10 posts	20 posts
0.01	100	1	0.0	0.0	0.1
	1,000	10	0.0	0.1	0.7
	10,000	100	0.0	0.5	1.8
	100,000	1,000	0.2	1.5	5.0
0.1	100	10	0.6	1.1	1.8
	1,000	100	1.4	3.0	6.5
	10,000	1,000	3.5	6.8	19
	100,000	10,000	5.3	17	54.1
0.3	100	30	1.9	3.7	6.2
	1,000	300	4.9	10.8	21.1
	10,000	3,000	12.7	23.3	63.9
	100,000	30,000	20.2	63.4	190.9

Table 1: Number of users convicted such that the probability to convict a innocent user is at most 0.5.

4.1 Preliminaries

Let G denote a *directed* network of users. A connection from user u to user v denotes that v follows u. Let n denote the total number of users.

For the analysis we make the assumption that all users are equally likely to like a given post, independently of their position in the network and the opinion of other users. Precisely, we assume the following probabilistic *uniform opinion model*.

Definition 2 (Uniform Opinion Model)**.** *In this model, each post is associated with some probability* p*, called the* popularity *of the post (different posts may have different popularity). The probability that any given user* u *likes a post is equal to the post's popularity* p*, and is independent of the set of users* $v \neq u$ *that like the post.*

Suppose that user u receives a post of popularity p. Let s be the number of u's followers that have not received the post yet, and suppose $s > 0$. If we assume the uniform opinion model, then u has probability p of liking the post, and thus the probability of reposting for Riposte is

$$p \cdot r_{\text{like}}(s) + (1-p) \cdot r_{\text{dis}}(s).$$

If $s \geq \lambda + \delta$ then $r_{\text{like}}(s) = \lambda/s$ and the probability above is

$$\frac{p\lambda + (1-p)\delta}{s}.$$

Since s followers of u do not know the post yet, the expected number of users that will learn the post from u is

$$\frac{p\lambda + (1-p)\delta}{s} \cdot s = p\lambda + (1-p)\delta.$$

If this quantity is greater than 1, we say that the post is *popular*; if it is smaller than 1, we say it is *unpopular*. The formal definition is as follows.

Definition 3 (Popular/Unpopular posts)**.** *For given* λ *and* δ*, we define the popularity threshold*

$$p^* := \frac{1-\delta}{\lambda - \delta},$$

and we call a post popular *if its popularity is* $p > p^*$*, and* unpopular *if* $p < p^*$.[8]

We define now a simple random network model of a typical micro-blogging network, in which there is a large variation in the number of followers between users, while there is little variation in the number of followees (i.e., the users that one user follows). The model takes as a parameter a distribution ϕ on the number of followers.

Definition 4 (G_ϕ network model)**.** *Let ϕ be a probability distribution on the integers $0, \ldots, n-1$. Then G_ϕ is an n-user random network constructed as follows. Independently for each user u, we first choose the number d of followers that u will have, according to distribution ϕ; and then we choose d uniformly random users among all the other users (excluding u) to be u's followers.*

Finally, we note that the probability functions $r_{\text{like}}(k)$ and $r_{\text{dis}}(k)$ have been defined only for $k \geq 1$. To simplify notation, we define also $r_{\text{like}}(0) = r_{\text{dis}}(0) = 0$.

4.2 Bounds on Dissemination

We begin with the simple observation that RIPOSTE achieves at least as large dissemination as DB-RIPOSTE, in the following sense.

Observation 1. For a network G, consider the spread of a post when RIPOSTE is used, and the spread of the same post (originated at the same user) if DB-RIPOSTE is used instead. We assume that the opinion of each user is the same in both cases. Let N and N_{db} denote the number of users that receive the post in each case. Then for any k, the probability that $N \geq k$ is at least equal to the probability that $N_{db} \geq k$.

The reason is that for a user u, the number of followers s that have not yet received the post is smaller or equal to the total number d of u's followers, and from this it follows that $r_{\text{like}}(s) \geq r_{\text{like}}(d)$ and $r_{\text{dis}}(s) \geq r_{\text{dis}}(s)$, if $s > 0$. The complete proof can be found in the appendix.

Remark 2. From Observation 1 it follows that any upper bound on the dissemination that holds for RIPOSTE, applies also to DB-RIPOSTE; and any lower bound for DB-RIPOSTE applies also to RIPOSTE.

Next we establish an upper bound on the spread of unpopular posts, and a lower bound on the spread of popular posts.

Unpopular posts. We present now an upper bound on the expected dissemination of unpopular posts. We show that the expected number of users who receive a given unpopular post is by at most a constant factor larger that the number of followers of the user who started the post. The constant factor depends on the popularity of the post and parameters δ and λ. This bound holds for any network G, assuming the uniform opinion model. Recall that a post is unpopular if its popularity is smaller than $p^* = (1-\delta)/(\lambda - \delta)$.

Theorem 2 (Upper bound for unpopular posts)**.** *For any G, and under the uniform opinion model, both* RIPOSTE *and* DB-RIPOSTE *guarantee that a post with popularity $p < p^*$ started by a user with d followers is received by an expected total number of at most d/β users, where $\beta = (p^* - p)(\lambda - \delta)$.*

Observe that as p approaches the popularity threshold p^*, factor β decreases, and thus the bound on the expected spread increases. Further, substituting the definition of p^* gives $\beta = 1 - \delta - p(\lambda - \delta)$, which implies that increasing either λ or δ increases the expected spread. All these observations are consistent with the intuition.

From Remark 2, it follows that it suffices to prove the upper bound of Theorem 2 just for RIPOSTE, as then the same bound holds for DB-RIPOSTE. The proof for RIPOSTE is a bit technical and can be found in the appendix. Instead, we provide below a simpler proof that holds only for DB-RIPOSTE.

Proof of Theorem 2 for DB-RIPOSTE. In DB-RIPOSTE, the set of all users that receive the post does not depend on the order in which reposts take place. Thus we can assume that dissemination proceeds in rounds in a breadth-first manner, as follows: In round 0, the source user posts the content; then in each round $t > 0$, every user that learned the post in the previous round $t-1$, either reposts or decides it will not repost. (We say a user *learns* a post the first time it receives it.)

Let Z_t, for $t \geq 0$, denote the number of users that learn the post in round t; so $Z_0 = d$. The total number T of users that learn the post (in any round) is then

$$T = \sum_{t \geq 0} Z_t.$$

We bound now the expectation of each Z_t. The probability for a user u with k followers to repost when it receives the post, is

$$p \cdot r_{\text{like}}(k) + (1-p) \cdot r_{\text{dis}}(k) \leq \frac{p\lambda + (1-p)\delta}{k} = \frac{1-\beta}{k},$$

where the last equation is obtained using the equations $\beta = (p^* - p)(\lambda - \delta)$ and $p^* = (1-\delta)/(\lambda - \delta)$. Thus the expected number of users that receive the post from u is at most

$$\frac{1-\beta}{k} \cdot k = 1 - \beta.$$

Some of these users may have already received the post from a different user, in the same or a previous round, thus $1-\beta$ is just an upper bound on the expected number of users that learn the post from u. Given now the

[8]For the asymptotic bounds we show later, we assume for a popular post that $p > p^* + \epsilon$, and for an unpopular post that $p < p^* - \epsilon$, for some arbitrary small constant $\epsilon > 0$.

number of users that learned the post in round $t-1$, it follows from the linearity of expectation that the expected number of users that learned the post in round t is

$$\mathbf{E}[Z_t \mid Z_{t-1}] \leq Z_{t-1} \cdot (1-\beta).$$

Taking the unconditional expectation on both sides yields $\mathbf{E}[Z_t] \leq \mathbf{E}[Z_{t-1}] \cdot (1-\beta)$. Applying this inequality iteratively and using that $\mathbf{E}[Z_0] = Z_0 = d$ gives

$$\mathbf{E}[Z_t] \leq (1-\beta)^t d.$$

Since $T = \sum_{t \geq 0} Z_t$, it follows from the linearity of expectation and the above bound that

$$\mathbf{E}[T] = \sum_{t \geq 0} \mathbf{E}[Z_t] \leq \sum_{t \geq 0} (1-\beta)^t d = d/\beta.$$

This completes the proof of Theorem 2 for DB-RIPOSTE. □

Popular posts. Next we study the dissemination of popular posts on the G_ϕ network model, for an arbitrary distribution ϕ for the followers (under a mild constraint on the min number of followers). We establish a lower bound on the probability of a popular post to be received by a constant fraction of users. The probability and the size of the fraction grow respectively with the number d of followers of the source, and the popularity of the post. In particular, the probability converges to 1 for d sufficiently large relative to the average number of followers.

Theorem 3 (Lower bound for popular posts)**.** *Consider the network model G_ϕ, for a distribution ϕ such that the minimum number of followers of any user is at least $\lambda + \delta$, and the average number is μ. Let $\epsilon, \epsilon' > 0$ be arbitrary small constants. Suppose that a post with popularity $p \geq p^* + \epsilon$ is posted by a random user, and this user has d followers. Under the uniform opinion model, both* RIPOSTE *and* DB-RIPOSTE *guarantee that with probability at least $1 - e^{-\Omega(d/\mu)}$ the total number of users that receive the post is at least*

$$(1-\epsilon') \cdot \frac{\beta n}{\beta + 1},$$

where $\beta = (p - p^*)(\lambda - \delta)$.

Note that the same constant $\beta = |p - p^*| \cdot (\lambda - \delta)$ appears in both Theorems 2 and 3. Unlike the d/β bound of Theorem 2, the threshold spread of $(1-\epsilon') \cdot \frac{\beta n}{\beta+1}$ predicted by Theorem 2 is independent of d, and depends only on λ and δ: the larger their value the larger the spread. The independence from d is intuitively justified, because as soon as the the post reaches a "critical mass" of users it will almost surely spread to a constant fraction of the network. However, d determines the probability with which such a critical mass will be reached. For d close to the average number of followers, this probability is at least some constant.

The proof of Theorem 3 uses a coupling between the dissemination process and an appropriate branching process, to show that the probability of the event we are interested in, that at least a certain fraction of users receive the post, is lower-bounded by the survival probability of the branching process. Then we bound this survival probability using a basic result for branching processes. The proof can be found in the appendix.

5. EXPERIMENTAL SETUP

This section presents the experimental setup of RIPOSTE on real topologies from micro-blogging platforms and social networks.

5.1 Data-sets

We use data-sets from several online services as the underlying topology for our experiments.

Twitter is arguably the most popular micro-blogging platform. We use a snapshot of the complete Twitter network from 2009 containing 42 million users and 1.5 billion edges [14].

Facebook is the largest online social network, with more than a billion active users. We use a sample of Facebook from 2009 containing 3 million users and 28 million links [21].

Renren is an online social network similar to Facebook, that is is popular in China. We use a sample of Renren from 2013 containing 1 million users and 57 million links [5].

LiveJournal is a combination of a blogging service and a social network. We use a sample of LiveJournal containing 4.8 million users and 69 million edges.[9]

Google+ is another popular social network. We use a sample containing 107 thousands users and 14 millions edges.[10]

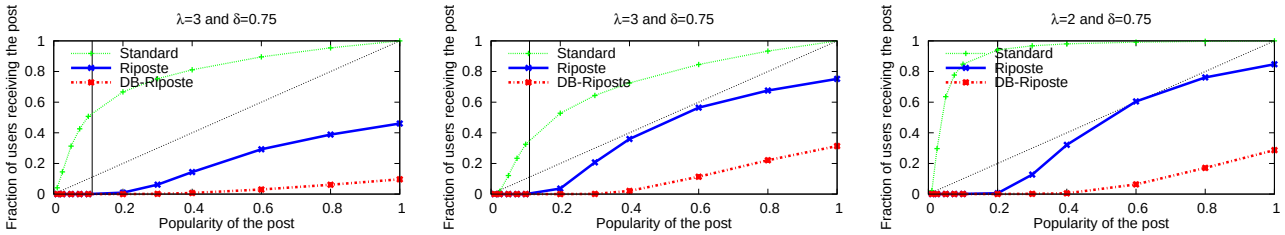
A summary of these topologies is given in Table 2.

Name	# Users	# Links	Average # Followers
Twitter	42M	1468M	35
Renren	1M	58M	58
LiveJournal	4.8M	69M	14
Facebook	3M	47M	16
Google+	107K	14M	130

Table 2: Data-Sets characteristics

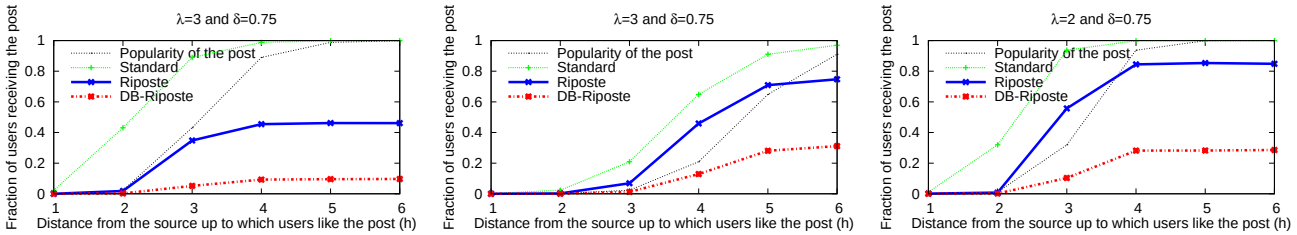
[9] http://snap.stanford.edu/data/soc-LiveJournal1.html

[10] http://snap.stanford.edu/data/egonets-Gplus.html



(a) Twitter data-set (b) LiveJournal data-set (c) Renren data-set

Figure 3: Spreading patterns under RIPOSTE, DB-RIPOSTE and STANDARD protocols (Uniform opinion model)



(a) Twitter data-set (b) LiveJournal data-set (c) Renren data-set

Figure 4: Spreading patterns under RIPOSTE, DB-RIPOSTE and STANDARD protocols (Distance-threshold opinion model)

5.2 User Opinion Model

While data on the topology of online services are available, access to the history of users about the posts they received or liked is severely restricted. Hence, we rely on synthetic models to generate the users opinions. We consider two user opinion models: the uniform model and the distance-threshold model.

Uniform model. In this model, every user has the same probability to like a given post. Precisely, each post is assigned a popularity in $[0, 1]$, which is the probability with which any user will like this post. Under this model, the opinion of a user is independent of her position in the network. This model is the same as the one of Definition 2 on which our analysis of RIPOSTE in Section 4 is based.

Distance-threshold model. This model accounts for the fact that in social networks, users are more likely to be interested in the same posts when they are closer to each other in the network. In the distance-threshold opinion model, the opinion of a user about a post depends on her shortest-path distance from the source of the post. If this distance is below some threshold h, which is a parameter of the model, the user likes the post with probability 1, whereas if the distance is larger than h then the user dislikes it with probability 1.

5.3 Evaluation metrics

We evaluate the dissemination properties of RIPOSTE along the following metrics:

Fraction of users that receiving the post. This metric is the number of users who receive a post over the total number of users.

Recall.

$$recall = \frac{|\text{users who like the post} \cap \text{users who receive the post}|}{|\text{users who like the post}|}$$

This metric measures how well the dissemination algorithm is able to disseminate the post among users who like it.

Precision.

$$precision = \frac{|\text{users who like the post} \cap \text{users who receive the post}|}{|\text{users who receive the post}|}$$

This is an indication of how many users receive posts they are not interested in.

Spam rate.

$$spam = \frac{|\text{users who dislike the post} \cap \text{users who receive the post}|}{|\text{users who dislike the post}|}$$

5.4 Dissemination model

To evaluate the performances of RIPOSTE, we simulate the dissemination of posts over several networks we selected and with the two user opinion models. For each simulation we select the source of the post at random among the users having at least 30 followers (30 is the average number of followers for different data-sets). To start, we add all the followers of the source to a receivers list and then we iterate over each of the user in this list. For each user, we decide if she likes the post

according the opinion model, and then if she forwards the post to her followers according to the spreading algorithm. If the post is forwarded to the user's followers, all the followers that receive the post for the first time are added to the receivers list.

In addition to RIPOSTE and DB-RIPOSTE, we simulate the spreading with a third dissemination algorithm that we call STANDARD. STANDARD is a non-privacy preserving dissemination protocol that forwards with probability 1 when the user likes the post, and with probability 0 when the user dislikes the post (this is equivalent to DB-RIPOSTE with $\delta = 0$ and $\lambda = n$ where n is the number of nodes in the network).

6. EXPERIMENTAL RESULTS

Our experimental evaluation shows that the dissemination patterns generated by RIPOSTE match the theoretical analysis.

6.1 Fraction of users receiving the post

Figure 5 displays the fraction of users who received the disseminated posts with RIPOSTE over different data-sets. We see that RIPOSTE behaves consistently with different data-sets. The posts with a popularity smaller than 0.2 never spread, while for the more popular posts the spreading increases with the popularity.

For space reasons, we consider only three data-sets. Given that Twitter, Facebook and Goggle+ have similar results, we keep only Twitter which is the largest one. Thus in the following we will only show the results for Twitter, LiveJournal and Renren.

Figure 3 displays the average fraction of users who received the posts in the considered data-sets under the uniform opinion model, for various popularity values. The straight line represents an ideal system, e.g., a post liked by 40% of the users reach nearly 40% users on an average.

We observe that the STANDARD algorithm is far from achieving this objective and clearly over-disseminates the posts. In particular, unpopular content is widely disseminated, for instance posts with a popularity of 0.1 reaches over 50% of the network in Twitter and over 80% in Renren. On the contrary, RIPOSTE achieves a dissemination much closer to the ideal one in all data-sets and particularly in LiveJournal and Renren.

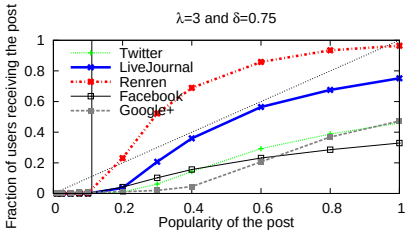


Figure 5: Spreading patterns under RIPOSTE, all data-sets (Uniform opinion model)

The popularity threshold p^*, introduced in Definition 3, is displayed in Figure 3 by a vertical line. The experiments validate that, for both RIPOSTE and DB-RIPOSTE, posts with popularity below threshold spread to a very small fraction of users, whereas when popularity grows above the threshold, the posts are disseminated to a large fraction of the users.

Figure 4 presents the same experiments with the distance-threshold model. As expected, we observe similar results as for uniform model. Not only RIPOSTE ensures privacy, but the spreading achieved by the protocol matches the expected spreading and clearly outperforms STANDARD in this respect.

6.2 Dissemination of unpopular posts

According to Theorem 2, the spreading of unpopular posts is bounded by:

$$\frac{|\text{users who like the post}|}{\text{source degree}} \leq \frac{1}{\beta} = \frac{1}{(p^* - p) \times (\lambda - \delta)}$$

Figure 6 depicts the spreading of unpopular posts achieved by RIPOSTE and DB-RIPOSTE when compared to this theoretical bound. We observe that the experiments match the analytical results: with both RIPOSTE and DB-RIPOSTE the spreading of unpopular posts is indeed lower than the theoretical bound.

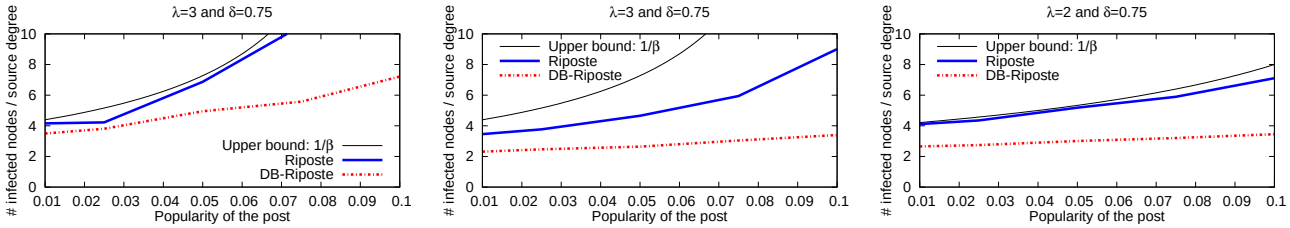
6.3 Precision and recall of the spreading

We now do a qualitative evaluation of the spreading by comparing the sets of users receiving the posts and the sets of users interested in these posts.

Due to lack of space, we do not provide the results for all data-sets but only for LiveJournal in Figure 7. For this part of the evaluation we use only the distance-threshold model as the uniform model is irrelevant in this configuration.[11]

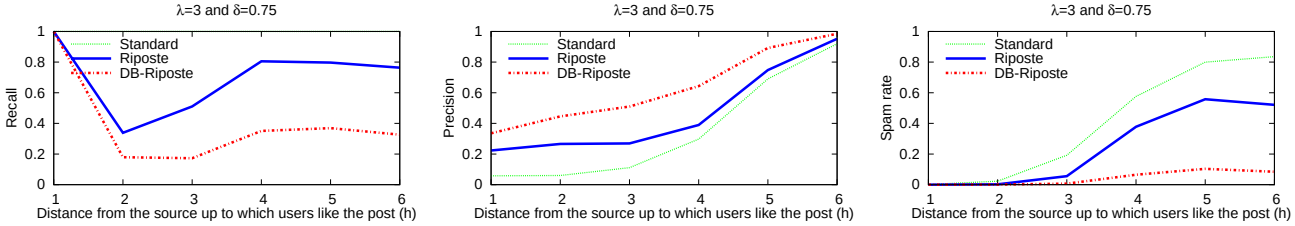
In Figure 7a we observe that the recall for the STANDARD protocol is always 1. This is because in the distance-threshold model for every user who likes the post there exists a path in the network between her and the source of the post containing only users who like the post. Thus, since in STANDARD the users always forward posts they like, all the users liking the post end up receiving it in the distance-threshold model. In contrast, since with RIPOSTE, a posts is never reposted with probability 1, the recall is always less than 1. However RIPOSTE still spreads the interesting posts to a very large fraction of the interested users (up to 80%) while DB-RIPOSTE achieves at best a recall of 35%.

[11]In the uniform model, the opinion of the user is independent from her position in the network, thus the precision depends only on the popularity of the post, and not on the spreading algorithm



(a) Twitter data-set (b) LiveJournal data-set (c) Renren data-set

Figure 6: Spreading patterns of unpopular posts in RIPOSTE and DB-RIPOSTE when compared to the theoretical bound (Uniform opinion model)



(a) Recall (b) Precision (c) Spam rate

Figure 7: Spreading characteristics under RIPOSTE, DB-RIPOSTE and the STANDARD protocol, LiveJournal data-set (Distance-threshold model)

In Figure 7b, we can see that RIPOSTE achieves a better precision than STANDARD. It may seem surprising since because of the randomization performed by RIPOSTE, many users could repost the posts they do not like which could strongly reduce the precision. However, we can see that this does not happen as the precision of RIPOSTE is significantly larger than the one of STANDARD, especially for small values of h. We can draw the same conclusion from Figure 7c as it shows that RIPOSTE spams spread much less than STANDARD.

6.4 Effect of λ and δ on spreading

In the experiments shown so far, we set the values of λ and δ to values that match a level of privacy that we believe is reasonable, with respect to the desired dissemination. Now we evaluate the effect of varying values of λ and δ on the dissemination. Results are shown only for LiveJournal for space reasons. To isolate the impact of each parameter, we first vary the value of δ with $\lambda = 3$ and then vary the value of λ with δ set to 0.75. Results are reported in Figure 8a and Figure 8b respectively. Figure 8a shows that varying the value of δ has a strong impact on the popularity threshold after which the posts start to spread, but has little effect on the spread of posts with a popularity greater than 0.4. Figure 8b, where the value of λ varies between 2 and 4, shows that λ strongly impacts both the popularity threshold after which the posts starts to spread and the number of users receiving the post.

7. IMPLEMENTATION DETAILS

In this section we discuss RIPOSTE's underlying implementation issues and its interaction with the social network or the micro-blogging platform.

As we have mentioned earlier, RIPOSTE does not require any involvement from the server: it fits within the user's web browser. In this case, the randomization of RIPOSTE is done locally on the user's machine, and the only information accessible by the server is the output of RIPOSTE, i.e., the randomized decision. Thus the users' privacy is protected not only against an attacker, but also against the service provider.

We should stress that RIPOSTE diverges from the standard repost practice: A user may not repost all the content he would like to, and may repost content he does not like. The goal of our protocol, however, is to complement rather than replace the standard mechanism where each user reposts what he likes. Although not addressed in this paper, it is possible to have a hybrid version of RIPOSTE where a user can choose to "force" a repost action on a specific post, if the user is not concerned about revealing its opinion on that post. However, we cannot allow a user to force that a post is *not* reposted, otherwise RIPOSTE's privacy guarantees no longer hold. Indeed, in that situation a user could be convicted simply because she did not force the non-reposting of a post, whatever her true opinion is. The information of whether a repost was done by RIPOSTE or forced by the user can be communicated to the other

users. So, for example, a user may choose to ignore all reposts that were not forced, in which case the hybrid protocol behaves like a standard micro-blogging service.

However, for RIPOSTE to achieve the desired dissemination, a user should always express her opinion for the posts she receives. In order to achieve this, we propose to display one post at a time to the user and let her take a decision to either repost it or to move to another post, which implicitly implies her decision not to repost it. Another solution to deal with this is to impose a time-out, after which the post cannot be disseminated using RIPOSTE.

8. RELATED WORK

Dwyer et al. [7] highlighted the weak privacy safeguards in micro-blogging and social-networking services. Not surprisingly, a lot of work has been devoted to the problem, including studies on how micro-blogging and social network services fail to give complete privacy guarantees to the user [23], characterization of privacy threats to facilitate their intensive study [13], as well as solutions dealing with individual threats [12, 19, 24].

Anonymization (i.e., replacing names with meaningless unique identifiers [4]) is one of the most studied privacy-preserving mechanisms for users in social networks, and for releasing data from these networks for analysis and research. Anonymization does not always suffice as the original data-sets can sometimes be reconstructed [4, 9, 16]. K-anonymity, introduced in [20], protects tabular micro-data against linking attacks. The concept has been extended for social network data-sets in [25] . Clearly, there is an information loss that comes with anonymization in social-networks and there is a trade-off between the increased anonymity and the loss of information as a result of anonymization [9].

The authors of [2] have recently proposed a system, *anonyLikes*, that keeps the actual like count of a post without revealing the names of the users who like it. The adoption of *pseudonyms* has also been a common practice by activists to hide their identity while contributing to the spread of sensitive information [22]. This promises privacy to the user's original identity at the cost of the trust that their followers would put in them should they knew the original identity of that user. A similar issue concerns the hiding of IP addresses [17].

RIPOSTE preserves the identity of the users while providing privacy guarantees that can be quantified in terms of differential privacy guarantees. Although there has been a considerable work to bringing the concept of differential-privacy to social-networks and micro-blogging systems [10,11,18], to the best of our knowledge, our approach of disseminating information without anonymizing the users by giving differential privacy guarantees is novel.

9. CONCLUSION

Privacy is becoming a prominent and continuously raising concern in micro-blogging systems and is somewhat incompatible with the social nature of such platforms, e.g., the identities of followers as well as the information they reposted is known. These systems expect the interesting content to spread and spam to die quickly. In RIPOSTE we take on this challenge and provide a privacy-preserving reposting protocol that ensures that an external attacker cannot tell with certainty whether a user contributed or not to the dissemination of some specific content. This is a powerful property as social platforms play a major role in many countries and may lead to actual arrests. Beyond privacy, RIPOSTE also sustains the expected spreading patterns of posts in a micro-blogging system by achieving a wide spread dissemination of popular posts and limited spread of unpopular posts.

RIPOSTE achieves this trade-off between privacy and dissemination by probabilistically switching the reposting actions of a user. We quantify the privacy property of RIPOSTE analytically and provide bounds on the dissemination patterns. Our experimental results based on a wide range of real topologies (Twitter, Google+, Facebook, Renren and LiveJournal) show that the dissemination patterns indeed reflect the overall interests of posts in the system. We believe that such a protocol could lead to novel kinds of privacy-preserving micro-blogging platforms or be widely adopted by existing ones.

(a) $\lambda = 3$, variation of δ

(b) $\delta = 0.75$, variation of λ

Figure 8: Effect of the variations of λ and δ, LiveJournal (Uniform model)

10. REFERENCES

[1] B. Al-Ani, G. Mark, J. Chung, and J. Jones. The Egyptian blogosphere: A counter-narrative of the revolution. In *CSCW*, pages 17–26, 2012.

[2] P. Alves and P. Ferreira. Anonylikes: Anonymous quantitative feedback on social networks. In *Middleware*, pages 466–484, 2013.

[3] K. Athreya and P. Ney. *Branching processes*. Dover Publications, 2004.

[4] L. Backstrom, C. Dwork, and J. Kleinberg. Wherefore art thou r3579x?: Anonymized social networks, hidden patterns, and structural steganography. In *WWW*, pages 181–190, 2007.

[5] C. Ding, Y. Chen, and X. Fu. Crowd crawling: towards collaborative data collection for large-scale online social networks. In *Proceedings of the first ACM conference on Online social networks*, pages 183–188. ACM, 2013.

[6] C. Dwork. Differential privacy: A survey of results. In *TAMC*, pages 1—19, 2008.

[7] C. Dwyer, S. R. Hiltz, and K. Passerini. Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace. In *AMCIS*, page 339, 2007.

[8] P. Haccou, P. Jagers, and V. A. Vatutin. *Branching processes: Variation, growth, and extinction of populations*. Cambridge Univ. Press, 2005.

[9] M. Hay, G. Miklau, D. Jensen, D. Towsley, and P. Weis. Resisting structural re-identification in anonymized social networks. *PVLDB*, 1(1):102–114, 2008.

[10] S. P. Kasiviswanathan, K. Nissim, S. Raskhodnikova, and A. Smith. Analyzing graphs with node differential privacy. In *TTC*, pages 457–476, 2013.

[11] D. Kifer and A. Machanavajjhala. No free lunch in data privacy. In *SIGMOD Conference*, pages 193–204, 2011.

[12] A. Korolova, R. Motwani, S. U. Nabar, and Y. Xu. Link privacy in social networks. In *CIKM*, pages 289–298, 2008.

[13] B. Krishnamurthy and C. E. Wills. Characterizing privacy in online social networks. In *WOSN*, pages 37–42, 2008.

[14] H. Kwak, C. Lee, H. Park, and S. Moon. What is Twitter, a social network or a news media? In *WWW*, pages 591–600, 2010.

[15] G. Lotan, E. Graeff, M. Ananny, D. Gaffney, I. Pearce, and D. Boyd. The revolutions were tweeted: Information flows during the 2011 Tunisian and Egyptian revolutions. *IJoC*, 5:1375–1405, 2011.

[16] A. Narayanan and V. Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, pages 173–187, 2009.

[17] O. Olaore. Politexting: Using mobile technology to connect the unconnected and expanding the scope of political communication. In *ISECON*, pages 1–8, 2011.

[18] C. Task and C. Clifton. A guide to differential privacy theory in social network analysis. In *ASONAM*, pages 411–417, 2012.

[19] A. Tootoonchian, S. Saroiu, Y. Ganjali, and A. Wolman. Lockr: Better privacy for social networks. In *CoNEXT*, pages 169–180, 2009.

[20] J. R. Ullmann. An algorithm for subgraph isomorphism. *JACM*, 23(1):31–42, 1976.

[21] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao. User interactions in social networks and their implications. In *EuroSys*, pages 205–218, 2009.

[22] V. Wulf, K. Misaki, M. Atam, D. Randall, and M. Rohde. 'on the ground' in Sidi Bouzid: Investigating social media use during the tunisian revolution. In *CSCW*, pages 1409–1418, 2013.

[23] E. Zheleva and L. Getoor. Privacy in social networks: A survey. In *Social Network Data Analytics*, pages 277–306, 2011.

[24] B. Zhou and J. Pei. Preserving privacy in social networks against neighborhood attacks. In *ICDE*, pages 506–515, 2008.

[25] B. Zhou and J. Pei. The k-anonymity and ℓ-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowl. Inf. Syst.*, 28(1):47–77, 2011.

APPENDIX

A. OMITTED PROOFS

A.1 Proof of Observation 1

If a user u likes the post it is reposted with probability $r_{\text{like}}(d)$ by DB-RIPOSTE, and with probability $r_{\text{like}}(s)$ by RIPOSTE, where d is the total number of u's followers and s is the number of those followers that do not know the rumor yet. Since $s \leq d$ it follows that $r_{\text{like}}(s) \geq r_{\text{like}}(d)$, if $s > 0$. Similarly, if u does not like the post it is reposted with probabilities $r_{\text{dis}}(s)$ and $r_{\text{dis}}(d)$, respectively, and $r_{\text{dis}}(s) \geq r_{\text{dis}}(d)$. The claim now follows by a standard coupling argument. Since each user has the same opinion in both cases, we can couple the random choices of the two protocols so that DB-RIPOSTE does a reposts from u *only if* RIPOSTE also reposts from u. This coupling ensures that the set of users who receive the post when DB-RIPOSTE is used is a subset of those who receive the post with DB-RIPOSTE. This implies that $\Pr(N \geq k) \geq \Pr(N_{db} \geq k)$, for all k.

A.2 Proof of Theorem 2 for RIPOSTE

We consider the following representation of the random process underlying the dissemination of the post. At each point in time users are divided into three sets: (1) the set D of those who have received the post and either have reposted it, or it has been decided they will not repost; (2) the set N of users who have received the post but it has not yet been decided whether they will repost or not; and (3) the set S of the remaining users, who have not received the post yet.[12] We assume that dissemination proceeds in steps: At each step, a single user u from set N is considered, and this user either reposts the post or it is decided to not repost. As a result, u is moved from set N to set D, and if u reposts then all its followers from set S are moved to N. The dissemination is completed when the set N becomes empty. For our analysis, the order in which the users from N are considered can be arbitrary.

We denote by D_t and N_t the values of the corresponding sets defined above after the first t steps; we assume that D_0 contains just the source of the post, and N_0 contains the d followers of the source.

Let $n_t = |N_t|$. The total number T of users that receive the post (excluding the source) is then

$$T = \min\{t \colon n_t = 0\}.$$

Suppose that $t \leq T$ (and thus $n_{t-1} > 0$), and consider the expected change on n_t in round t: If the user u considered in step t has s followers that have not received the post yet, then the probability u reposts is

$$p \cdot r_{\text{like}}(s) + (1-p) \cdot r_{\text{dis}}(s) \leq \frac{p\lambda + (1-p)\delta}{s} = \frac{1-\beta}{s}.$$

[12]The source user belongs to set D.

Thus, the expected number of new users that learn the post at step t is at most $1-\beta$, and the expected change in n_t is

$$\mathbf{E}[n_t - n_{t-1} \mid n_{t-1}] \leq (1-\beta) - 1 = -\beta, \qquad (5)$$

where the '-1' in the middle expression accounts for the removal of u from N_{t-1}.

It is now easy to understand the intuition behind the bound we must show: At each step, n_t drops *in expectation* by at least β. If, instead, the *actual* drop were at least β, then it would follow that the number of steps until n_t becomes zero would be at most $n_0/\beta = d/\beta$, which is equal to the bound we must show.

The formal proof is by a standard martingale argument. For $0 \leq t \leq T$, let $X_t = n_t + \beta t$. Then for $0 < t \leq T$, we have $X_t - X_{t-1} = n_t - n_{t-1} + \beta$, and

$$\mathbf{E}[X_t - X_{t-1} \mid X_0 \ldots X_{t-1}]$$
$$= \mathbf{E}[n_t - n_{t-1} + \beta \mid X_0 \ldots X_{t-1}] \overset{(5)}{\leq} 0.$$

Thus the sequence $X_0, X_1, \ldots, X_T$ is a super-martingale, and T is a stopping time for this sequence. Since the random variable T is bounded (it is at most equal to the total number $n-1$ of users, excluding the source), we can apply the martingale stopping theorem to obtain

$$\mathbf{E}[X_T] \leq \mathbf{E}[X_0].$$

Substituting the values $X_T = n_T + \beta T = \beta T$, as $n_T = 0$ by T's definition, and $X_0 = n_0 = d$, yields $\mathbf{E}[T] \leq d/\beta$. This completes the proof of Theorem 2 for RIPOSTE.

A.3 Proof of Theorem 3

We consider just DB-RIPOSTE; the claim for RIPOSTE follows then from Remark 2.

The proof uses a standard coupling argument to show that the probability of the event we are interested in, that at least a certain fraction of users receive the post, is lower-bounded by the survival probability of an appropriate branching process. Then we bound this survival probability using a result for branching processes.

Recall that a (Galton-Watson) branching process is a random process starting with one or more individuals, and in each step of the process a single individual produces zero of more offsprings and then dies. The number of offsprings of an individual follows a fixed probability distribution that does not vary between individuals. The process either finishes after a finite number of steps, when there are no individuals left, or continues forever. The probabilities of these two complementary events are called *extinction* and *survival probability*, respectively.

We compare now the dissemination of a post in G_ϕ, with a branching process we specify. First we compute the distribution of the number of new users that learn the post from a user u, at a point in time when fewer than ℓ users have learned the post, for some ℓ to be

exposed later. We have that the probability of u having i followers is $\phi(i)$; and if u has i followers, then the probability it reposts is $(p\lambda + (1-p)\delta)/i = (\beta+1)/i$. Further, if u decides to repost, we can assume that only then its i followers are chosen, and that they are chosen sequentially, one after the other. Then the probability that the j-th follower of u does not already know the post, and thus learns it from u, is at least $1 - \ell/n$, provided that at most ℓ users already know the post (including the first $j-1$ followers of u, and the source).

Consider now the branching process in which d individuals exist initially, and the number X of offsprings of an individual is chosen as follows. First we choose some integer $i \geq 0$ from distribution ϕ. If $i = 0$ then $X = 0$. If $i > 0$ instead, then with probability $(\beta+1)/i$ we let again $X = 0$; while with the remaining probability, we let X be a binomial random variable $B(i, q)$, counting the number of successes among i independent identical trials with success probability $q := 1 - \ell/n$.

Consider now a coupling of the two processes above, the dissemination process and the branching process, until the earliest point in the dissemination process when (at least) ℓ users have received the post, or the dissemination has finished. The coupling is such that the number N_t of new users that learn the post in a step t of the dissemination process is greater or equal to the number X_t of offsprings produced in the corresponding step of the branching process, for all steps t after which the total number of users that have learned the post is still smaller than ℓ. Such a coupling exists since, by construction, N_t dominates stochastically X_t.

From the coupling above, the probability that at least $\ell := (1-\epsilon') \cdot \beta n/(\beta+1)$ users receive the post in total, is lower-bounded by the probability that the total progeny of the branching process (i.e., the total number of individuals that ever existed) is at least ℓ. Further, the latter probability is lower bounded by the survival probability of the branching process; we denote this by ζ_d. Thus to prove the theorem it suffices to show that

$$\zeta_d = 1 - e^{-\Omega(d/\mu)}.$$

We do so in the remainder of the proof.

By the definition of the branching process, the expected number of offsprings of an individual is

$$\begin{aligned}\mathbf{E}[X] &= \sum_i \phi(i) \cdot \frac{\beta+1}{i} \cdot \mathbf{E}[B(i,q)] = \sum_i \phi(i) \frac{\beta+1}{i} \cdot iq \\ &= \sum_i \phi(i) \cdot (\beta+1) \cdot q = (\beta+1) \cdot q.\end{aligned}$$

We observe that $\mathbf{E}[X] > 1$, as

$$(\beta+1) \cdot q = (\beta+1) \cdot \left(1 - \frac{(1-\epsilon')\beta}{\beta+1}\right) = 1 + \epsilon'\beta. \quad (6)$$

Further, we have

$$\begin{aligned}\mathbf{E}[X^2] &= \sum_i \phi(i) \cdot \frac{\beta+1}{i} \cdot \mathbf{E}[(B(i,q))^2] \\ &= \sum_i \phi(i) \cdot \frac{\beta+1}{i} \cdot (i^2q^2 + iq(1-q)) \\ &= \sum_i \phi(i) \cdot (\beta+1) \cdot (iq^2 + q(1-q)) \\ &= (\beta+1) \cdot (\mu q^2 + q(1-q)),\end{aligned}$$

where $\mu = \sum_i \phi(i) \cdot i$ is the mean of ϕ. We will use the following standard lower bound on the survival probability ζ_1, when there is just one individual initially (see, e.g., in [8, Section 5.6.1]),

$$\zeta_1 \geq \frac{2(\mathbf{E}[X]-1)}{\mathbf{E}[X^2] - \mathbf{E}[X]}.$$

Substituting the values for $\mathbf{E}[X]$ and $\mathbf{E}[X^2]$ computed above yields

$$\begin{aligned}\zeta_1 &\geq \frac{2(q(\beta+1)-1)}{(\beta+1)(\mu q^2 + q(1-q)) - q(\beta+1)} \\ &= \frac{2(q(\beta+1)-1)}{q^2(\beta+1)(\mu-1)} = \frac{2(q(\beta+1)-1)(\beta+1)}{q^2(\beta+1)^2(\mu-1)} \\ &\overset{(6)}{=} \frac{2\epsilon'\beta(\beta+1)}{(1+\epsilon'\beta)^2(\mu-1)} = \Omega(1/\mu),\end{aligned}$$

where the last equation holds because $\beta = (p-p^*)(\lambda-\delta) \geq \epsilon(\lambda-\delta) = \Omega(1)$.

We can now express ζ_d in terms of ζ_1, by observing that the branching process starting with d individuals can be viewed as d independent branching processes starting with a single individual each. The former branching process survives if and only if at least one of the latter ones survives, thus,

$$\zeta_d = 1 - (1-\zeta_1)^d \geq 1 - e^{\zeta_1 d} = 1 - e^{\Omega(d/\mu)}.$$

This completes the proof of Theorem 3.