Intersection Attacks in Non-Uniform Setting

Dongchen Zou under the instruction of Simon Langowski

January 2024

Abstract

Recently consumer demand for privacy has spurred growth in private messaging systems. However, formally, privacy degrades in such systems when users log on and off: this change of status exposes the ongoing conversations. Intersection attacks (also known as statistical disclosure attacks) use messaging patterns or liveness information to reconstruct relationships, deanonymize users, and track user behaviors. Prior attacks assume users have an underlying uniform communication pattern for simplicity, leaving the question open of how effective such attacks would be in a non-uniform real world. We observe that effects like clustering in real social graphs and correlation between repeated conversations change the behavior and potential of such attacks. This paper provides a new approach that can consider some of these additional factors by constructing a polynomial to determine the social graph. We provide an analysis of the performance, accuracy, and convergence rate of our attack. Our attack applies to many existing anonymous communication systems, and our technique can be extended to incorporate additional factors.

1 Introduction

Anonymity ensures that two parties can converse in a network without leaking information to the network provider or third parties. Most research papers that look into anonymity base themselves on the setting that David Chaum proposed: the mix network[1]. It is a network system that hides the correspondence between the message sender and receiver and attains some degree of privacy. Efforts to take advantage of repeating patterns in the mix network to uncover user relationships and degrade anonymity are called intersection attacks. An example of the aforementioned repeating pattern that the adversary can exploit is the online status. To increase user activity, many platforms willingly display whether users are online, with a green dot, for example. Therefore, the adversary can easily know when a person is online or offline. This information, on its own, might not be a security issue. But when the eavesdropper integrates the activity patterns of many people, they can learn about underlying user relationships. For example, seeing person A frequently online with person B strongly implies that they are friends.

Previous papers developed different attacks to utilize this easy-to-obtain information. They use mathematical tools like Bayesian inference [2] and maximum likelihood estimation [3]. However, they all worked with a uniform mix in which every user is identical. This assumption of uniformity is not essentially true in real-world social media where everyone is so different (some are talkative, some are quiet, some like talking about movies, and some like to talk in the middle of the day) and understates the difficulty of extracting information from the patterns. We aim to provide a variant of mix network that is closer to real-life social media while maintaining the original properties. We then provide an intersection attack that works on the variant. Our contributions are

- Our model of social media better describes real-world users by considering factors like clustering (people tend to talk in groups) and correlation between repeated conversations (people tend to keep talking if they have talked before).
- Our innovative approach can consider these factors by constructing a special polynomial. By solving for the root of the polynomial, we can learn about the social network and its user behaviors.

This paper is structured as follows.

1. Setting: The variant of the mix and how it is generated
2. Attack: The description and implementation of the intersection attack
3. Analysis: Analysis of the validity and performance of the attack
4. Result: The output of the attack and comparison to theoretical analysis.

2 Setting

2.1 General Mix Network

In the mix that David Chaum proposed, there is a chain of proxies that takes in messages from multiple senders, shuffles them, and sends them back out in random order to the next destination (possibly another mix node) until they reach their final destinations: the intended recipients. This process of sending messages, shuffling, and reaching the recipients happens in one epoch, a time period where the messages are sent through the mix and anonymized. Although this is only a simplified description, it is apparent that, for each epoch, the eavesdropper only gets to see who sent messages but doesn't know to whom the messages go. If we assume that every person sends messages whenever they are online, the adversary only knows who is online for each epoch.

Let's there be N users in the mix network, and for any epoch, the probability that user i and j talk with each other is denoted as $0 \leq \mathsf{A}[\mathsf{i},\mathsf{j}] \leq 1$. As we can see, the $N \times N$ matrix A perfectly describes the mix network for this specific problem, so we will only refer to A in the future. The information that the adversary has for each epoch is simply a list of all people who are online.

Most papers work with a uniform mix network, which includes uniform A and uniform epochs. We will introduce and analyze both the uniform case and our variant in the following subsections.

2.2 Previous Uniform Mix Networks

When the mix network is uniform, each entry of A can only be two possible values: 0 or some fixed constant. In other words, two users either talk with a uniform probability or they don't. In addition to a uniform A, the epochs are also uniform, meaning that it's simply a coin flip of all $\mathsf{A}[\mathsf{i},\mathsf{j}]$ and if the coin flips to head, user i and j are put into the list.

This uniform mix network is easy to analyze but very different from the real world. In the real-world social media, people are different and they prefer to talk to some people more than others. The uniform epoch generation also ignores the tendency of people to continue their conversations. For example, if user i and j talk in the previous epoch, the probability that they talk for the current epoch should be larger than $\mathsf{A}[\mathsf{i},\mathsf{j}]$. Such a uniform network oversimplifies the real world and understates the complexity of this problem.

2.3 Our Non-Uniform Variant

We offer a variant of the mix network that should model the real world better. We observe that it is the common interests that initiate the conversation. That is, it is more likely for someone to talk with someone with more common interests and vice versa. Let Int be a finite set of all the interests. Each user in our variant has a list of interests $\mathsf{Int}_k \subseteq \mathsf{Int}$ and $\mathsf{A[i,j]}$ is directly proportional to $|\mathsf{Int}_i \cap \mathsf{Int}_j|$. The following is the implementation.

As we discussed in the previous subsection, people tend to keep talking. If user i and j talked in the previous epoch, the probability that they talk during this epoch should be greater than $\mathsf{A[i,j]}$. Therefore, we introduce the idea of correlation. We have a correlation factor $0 \leq \gamma \leq 1$. If user i and j talked in the previous epoch, the probability that they talk for the current epoch will be adjusted to $(1-\gamma)\mathsf{A[i,j]} + \gamma$. If $\gamma = 0$, the new probability will be $\mathsf{A[i,j]}$; if $\gamma = 1$, the new probability will be 1.

3 Intersection Attack

The adversary wants to find out A, the matrix that describes the mix network, because A tells him who is friends with whom (the larger $\mathsf{A[i,j]}$, the closer user i and user j are), which allows him to break the anonymization of the mix.

After getting t epochs of information, they can compile an array Count_t in which $\mathsf{Count}_t\mathsf{[i,j]}$ is the number of times where user i and user j both appear online (note that this is not necessarily talking to each other because both of them can be online talking to different people) in the first t epochs. They will use this information to put out a guess, denoted by $\mathsf{A'}_t$. This is their guess for how the mix network looks, and they want to get it close to A.

3.1 Description of the Attack

The attack will revolve around one question: what is the probability that user i and j appear online together?

Empirically, the adversary observes the probability to be $\frac{\mathsf{Count}_t\mathsf{[i,j]}}{t}$. They can compute this probability with $\mathsf{A'}$ as well. Note that user i and user j appearing online together does not necessarily imply that they are talking during that epoch. This can also occur when i talks to someone not j and j talks to someone not i. The probability i is talking to someone not j can be expressed via

$$\mathsf{Pr}\left[i \text{ talking to someone not } j\right] = 1 - \prod_{k \neq i}(1 - \mathsf{A'}[k,j])$$

because $\prod_{k\neq i}(1-\mathsf{A'}[k,j])$ is the probability that i is not talking to anyone and $1-\prod_{k\neq i}(1-\mathsf{A'}[k,j])$ is then the probability that i is talking to at least one person who is not j. We can now write out this probability, denoted by $\mathsf{F_{[i,j]}(A'')}$. If user i and user j are taking directly with each other (probability $\mathsf{A'[i,j]}$), they definitely appear online together; if user i and user j are not talking directly with each other (probability $1 - \mathsf{A'[i,j]}$), i is talking to someone not j (probability $1 - \prod_{k\neq i}(1 - \mathsf{A'}[k,j])$, and j is talking to some not i (probability $1 - \prod_{k\neq j}(1 - \mathsf{A'}[i,k])$), they can also appear online together. Hence,

$$\mathsf{F}_{[\mathsf{i},\mathsf{j}]}(\mathsf{A}') = \mathsf{A}'[\mathsf{i},\mathsf{j}] + (1 - \mathsf{A}'[\mathsf{i},\mathsf{j}])\left(1 - \prod_{k\neq i}(1 - \mathsf{A}'[k,j])\right)\left(1 - \prod_{k\neq j}(1 - \mathsf{A}'[i,k])\right)$$

The adversary wants this theoretical probability $\mathsf{F}_{[\mathsf{i},\mathsf{j}]}(\mathsf{A}')$ to match the empirical probability $\frac{\mathsf{Count}_t[\mathsf{i},\mathsf{j}]}{t}$ for every $[\mathsf{i},\mathsf{j}]$ because this tells them that his guess A' is matching the observations. Thus, they will try to minimize the difference between the two probabilities. Mathematically, the adversary is trying to minimize the polynomial

$$G(\mathsf{A}') = \sum_{[\mathsf{i},\mathsf{j}]}\left(\mathsf{F}_{[\mathsf{i},\mathsf{j}]}(\mathsf{A}') - \frac{\mathsf{Count}[\mathsf{i},\mathsf{j}]}{t}\right)^2$$

and the adversary tries to find the A' that minimizes $G(\mathsf{A}')$. Since $G(\mathsf{A}')$ is nonnegative, they are finding the root of G.

3.2 Validity

It is expected that

$$\lim_{t\to\infty}\frac{\mathsf{Count}[\mathsf{i},\mathsf{j}]}{t} = \mathsf{F}_{[\mathsf{i},\mathsf{j}]}(\mathsf{A})$$

This is because of the law of large numbers, which states that the empirical probability of success in a series of Bernoulli trials, which is $\frac{\mathsf{Count}[\mathsf{i},\mathsf{j}]}{t}$, will converge to the theoretical probability $\mathsf{F}_{[\mathsf{i},\mathsf{j}]}(\mathsf{A}')$. This means that

$$\lim_{t\to\infty} G(\mathsf{A}) = 0$$

However, how do we know that there isn't another $\mathsf{B} \neq \mathsf{A}$ such that

$$\lim_{t\to\infty} G(\mathsf{A}) = \lim_{t\to\infty} G(\mathsf{B}) = 0$$

? In other words, how do we know that A is the only solution to $G = 0$ and minimizing G will indeed give us the right A, instead of just some random B that accidentally satisfies $\lim_{t\to\infty} G(\mathsf{B}) = 0$? We observe in practice that F is sufficiently smooth such that there is a large region that converges to the true A for large enough t. This is because F is a polynomial in terms of the entries of A, and so there is a region that will be close (in the uniform continuity sense) to each 0. Mathematically speaking, we have observed that a significant portion of the hypercube region $[0,1]^N$ will be covered by the region converging to A for large t.
We have shown that $\lim_{t\to\infty} G(\mathsf{A}) = 0$ for A and only for A.

4 Analysis

In the previous section, we stated the attack and showed that the adversary's guess A' will eventually converge to A. In this section, we offer an analysis of how fast the convergence happens. In our setting of the mix network, the epochs are not independent because of the correlation. To make the analysis easier, we will assume they are independent. We will later see that this is a reasonable assumption. If the epochs are independent, then $\mathsf{Count}_t[\mathsf{i},\mathsf{j}]$ is approximately binomial

– with probability $\mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A})$ and t flips. We have that

$$\mathsf{E}\left[\mathsf{Count}_t[\mathsf{i},\mathsf{j}]\right] = \mathsf{E}\left[\mathsf{Binomial}(t, \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}))\right] = t \cdot \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A})$$

and hence

$$\mathsf{E}\left[\frac{\mathsf{Count}_t[\mathsf{i},\mathsf{j}]}{t}\right] = \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A})$$

The variance of the binomial distribution is

$$\mathsf{Var}\left[\mathsf{Binomial}(\mathsf{t}, \mathsf{p}[\mathsf{i},\mathsf{j}])\right] = t \cdot \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}) \cdot (1 - \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}))$$

Writing out the same logic with standard deviation, we have

$$\mathsf{StdDev}\left(\mathsf{Binomial}(\mathsf{t}, \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}))\right) = \sqrt{t \cdot \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}) \cdot (1 - \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}))}$$

and

$$\mathsf{StdDev}\left(\frac{\mathsf{Binomial}(\mathsf{t}, \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}))}{t}\right) = \frac{1}{t}\sqrt{t \cdot \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}) \cdot (1 - \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}))} = \frac{\sqrt{\mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}) \cdot (1 - \mathsf{F}_{[\mathsf{i,j}]}(\mathsf{A}))}}{\sqrt{t}}$$

Therefore, the standard deviation of $\frac{\mathsf{Count}_t[\mathsf{i,j}]}{t}$ is directly proportional to $\frac{1}{\sqrt{t}}$ because $\mathsf{F}_{[\mathsf{i,j}]}(A)$ is a constant. The standard deviation measures how much variation the random variable will have about its mean. This variation shrinks proportionally to $\frac{1}{\sqrt{t}}$ so we can say that the difference between the guess A' and the real A shrinks with speed $\frac{1}{\sqrt{t}}$
We can prove that this is indeed close to the best we can do using Central Limit Theorem. Let $\mathbf{C}$ be a random variable representing the observations with $\mathbf{C}_i$ the sample for epoch i. In other words, $\{\mathbf{C}_1, \mathbf{C}_2, \ldots, \mathbf{C}_t\}$ are independent observations of an unknown distribution $\mathsf{Dist}(\mu = \mathsf{F}(\mathsf{A}), \sigma^2)$. We are looking at the sample average which is

$$\left(\bar{\mathbf{C}} = \frac{\mathbf{C}_1 + \mathbf{C}_2 + \ldots + \mathbf{C}_t}{t}\right) \to \mathsf{F}(\mathsf{A}) \text{ when } t \to \infty$$

The central limit theorem states that

$$(\bar{\mathbf{C}} - \mathsf{F}(\mathsf{A})) \sim \frac{\mathcal{N}(0, \sigma^2)}{\sqrt{t}}$$

It essentially says that $\bar{\mathbf{C}} - \mathsf{F}(\mathsf{A})$ converges to 0 at the speed of $\frac{1}{\sqrt{t}}$ which proves that the attack is optimal asymptotically within a constant factor.
If the adversary gets two times as many epochs, they should expect to be better by a factor of $\frac{1}{\sqrt{2}} \approx 0.707$ so they improve by 29%.

4.1 Experiment

We will measure the accuracy of A' with the L-2 norm

$$d_t = \sqrt{\sum_{[\mathsf{i,j}]} (\mathsf{A}[\mathsf{i},\mathsf{j}] - \mathsf{A'}_t[\mathsf{i},\mathsf{j}])^2}$$

where A'_t is the best guess after integrating information for t epochs. The graphs of d_t versus t for different correlation factors γ are shown on the bottom. The blue-purple line is the actual graph and the orange line, which graphs our expectations, is

$$y = \frac{d_1}{\sqrt{t}}$$

The first two graphs are for $\gamma = 0.001$ and $\gamma = 0.002$. As we can see, a slight correlation does not perturb the result. For both cases, the two lines match with each other despite initial perturbation.

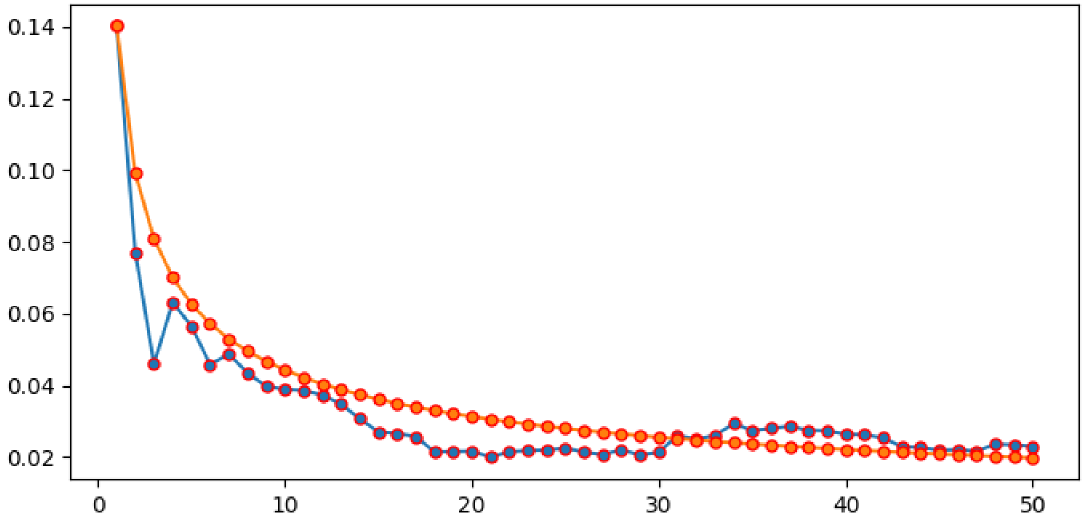

Figure 1: $\gamma = 0.001$

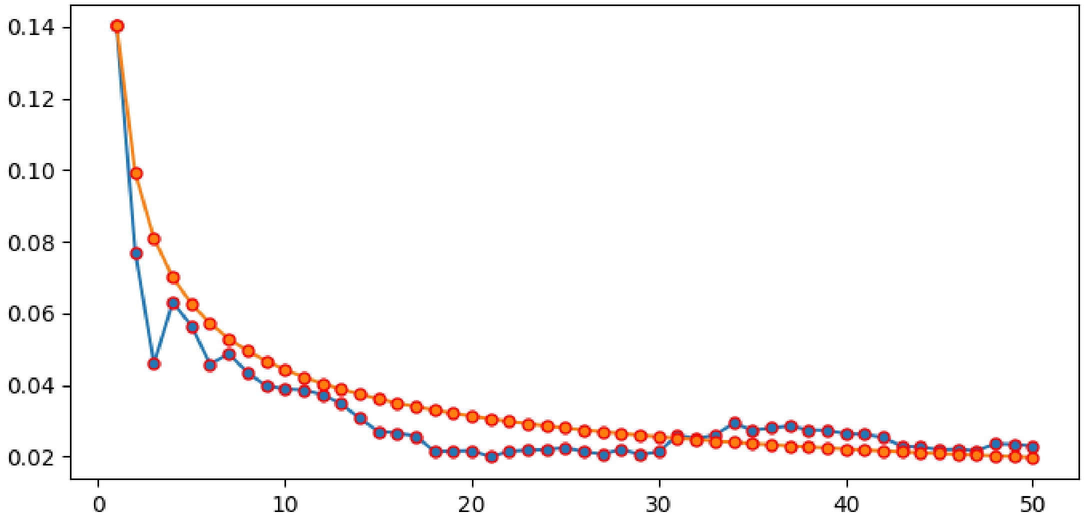

Figure 2: $\gamma = 0.002$

However, when $\gamma = 0.1$, the two lines still basically match, but the purple lines stay on top of the orange line. This is because a decent degree of correlation hides information from the adversary as he can't differentiate whether user i and j are talking because A[i, j] is big or they were simply talking the previous epoch.

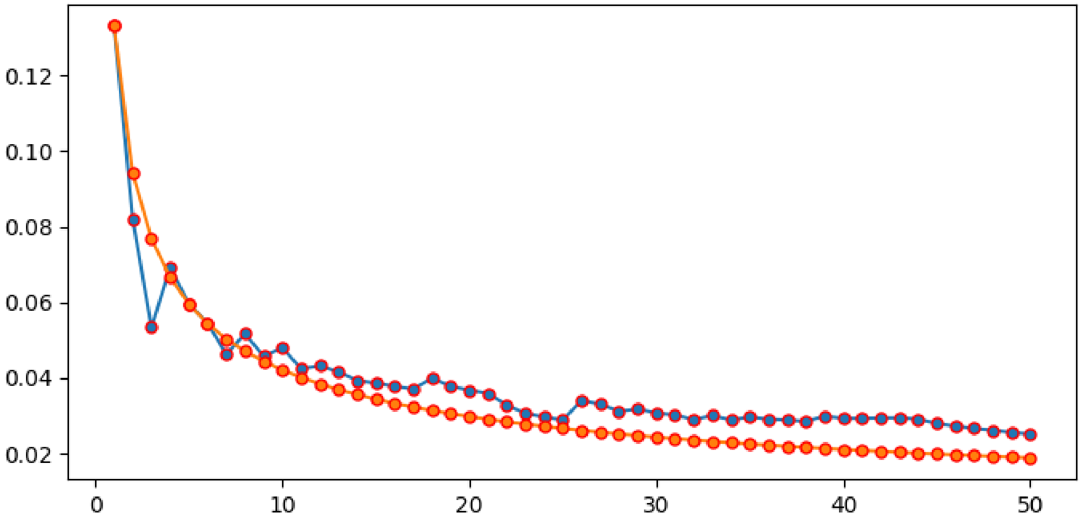

Figure 3: $\gamma = 0.1$

Finally, the attack just completely breaks down when $\gamma = 0.8$ as the adversary gets overwhelmed by repeating information. The programmed result strongly implies that our analysis is correct for small γ values.

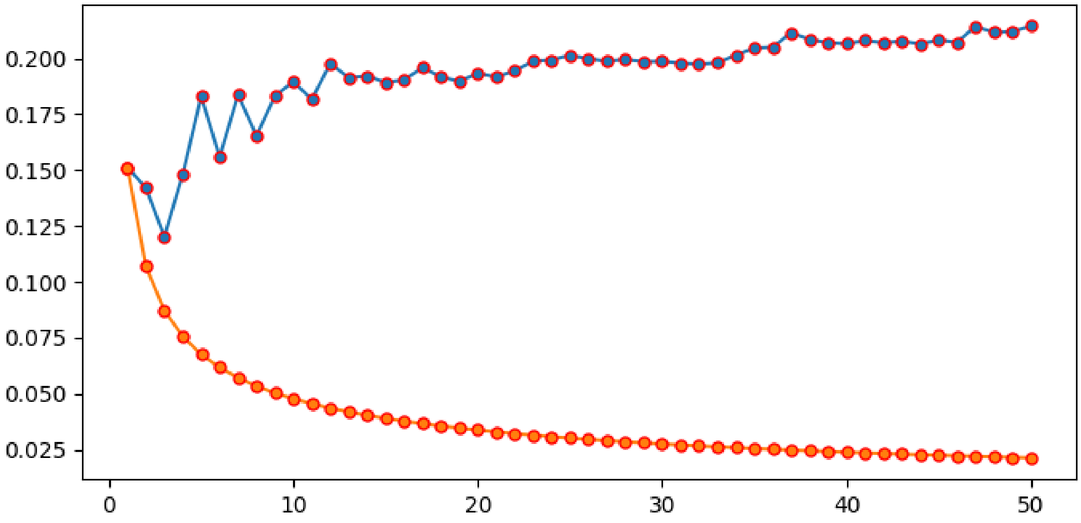

Figure 4: $\gamma = 0.8$

5 Acknowledgement

Finally, I would like to thank (the list is not ordered)

- my mentor Simon Langowski for his guidance on this topic. This project would not be possible without him.
- my family for being by my side.
- my friends like Celine, Michael, and Sufian (ordered alphabetically), to name a few, for their support.
- games like Baldur's Gate 3, Civ 6, GTA V (also not ordered) for they give me happiness
- the MIT PRIMES program for offering such a wonderful opportunity

6 Reference

[1] Chaum, David L. ”Untraceable electronic mail, return addresses, and digital pseudonyms.” Communications of the ACM 24.2 (1981): 84-90.
[2] Danezis, George, and Carmela Troncoso. ”Vida: How to use bayesian inference to de-anonymize persistent communications.” International Symposium on Privacy Enhancing Technologies Symposium. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009.
[3] Pérez-González, Fernando, and Carmela Troncoso. ”Understanding statistical disclosure: A least squares approach.” International Symposium on Privacy Enhancing Technologies Symposium. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.