Least Squares Disclosure Attacks: User Profiling for Mix-based Anonymous Communication Systems
Tech Report UV/TSC/FPG/30052012

Fernando Pérez-González, and Carmela Troncoso

January 2013

Abstract

Mixes, relaying routers that hide the relation between incoming and outgoing messages, are the main building block of high-latency anonymous communication systems. A number of so-called Disclosure Attacks have been proposed that effectively de-anonymize traffic sent through these systems. Yet, the dependence of their success on the system parameters is not well-understood. We propose the Least Squares Disclosure Attack (LSDA), an approach to disclosure rooted in Maximum Likelihood parameter estimation, in which user profiles are estimated solving a Least Squares problem. We show that the LSDA is not only suitable for the analysis of threshold mixes, but can be easily extended to attack pool mixes. Furthermore, contrary to previous heuristic-based attacks, our approach allows to analytically derive expressions that characterize the profiling error of the LSDA with respect to the system parameters. We empirically demonstrate that the LSDA recovers profiles users with greater accuracy than its predecessors, and verify that our analysis closely predicts actual performance.

1 Introduction

Computer security traditionally focuses on ensuring the confidentiality, integrity and availability of information; properties that are mostly achieved through cryptographic means. This protection, however, usually targets communication content and leaves network information accessible to potential adversaries. These traffic data, such as the identities of the participants in the communication (e.g. IP addresses), their location, or the amount and timing of data transferred, can be exploited by a passive observer to infer sensitive private information about the communication.

A well-studied countermeasure against traffic analysis for high-latency anonymous communications, i.e., communications that tolerate delay (e.g, e-mail), is the use of mix networks [1, 2, 3, 4]. Mixes are routers that prevent an observer from tracking communications by hiding the correspondence between inputs and outputs [5]. However, it is known that persistent and repeated communication patterns can be uncovered by means of a Disclosure Attack [6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. In a nutshell, this attack finds a user's likely set of friends, also known as user profile, by intersecting the recipient anonymity sets of the target user's messages.

The different variants of the disclosure attack differ on the technique used to infer user profiles from the observed communications. Even though all of them have been proven effective at the time of de-anonymization/profiling, their heuristic nature and/or their complexity hinders a possible analysis of how system parameters influence their success. Furthermore, the great majority of attacks have only been evaluated against simple threshold mixes (mixing occurs only between messages in a given round), and only the Statistical Disclosure Attack has been extended to attack pool mixes in which messages can be delayed for more than one round [9].

In this paper we propose a profiling approach based on solving a Least Squares problem, the Least Squares Disclosure Attack (LSDA). This approach ensures that the error between the actual number of output messages and a prediction based on the input messages is minimized. We show that the LSDA is suitable to attack anonymous communications through both threshold and pool mixes. In particular, in this paper we consider threshold binomial pool mixes, in which messages are individually selected to stay in the mix or to be sent to their receiver according to a binomial distribution. We note, however, that the choice of this mix is arbitrary and our approach can be adapted to many other probabilistic mixing strategies.

A remarkable feature of the Least Squares approach is that it allows for the derivation of analytical expressions that describe the evolution of the profiling error with the parameters of the system. This is a key property, as it permits designers to choose system parameters that provide a certain level of protection without the need to run simulations. We empirically validate our results, proving that our formulas reliably predict the evolution of the LSDA's error as the parameters of the system change, and that this error asymptotically tends to zero as the number of observed mix rounds grows.

The proposed profile estimator relies only on first and second order statistics. Hence the complexity of the LSDA is smaller than that of the most accurate profiling algorithms up to date, the Perfect Matching Disclosure Attack [15] and Vida [10], which are based on finding perfect matchings on a bipartite graph. We provide two variants of the LSDA: a very efficient unconstrained profile estimator that outputs user profiles that may contain negative probabilities (usually corresponding to receivers that are not friends with the target user), and a slower constrained version that further minimizes the error by ensuring that the output profiles are well-defined. We demonstrate through simulations that both versions indeed minimize the mean squared error with respect to heuristic disclosure attack variants [7, 9, 15], although they perform slightly worse than the Bayesian approach [10] in the scenarios considered in this paper.

The rest of the paper is organized as follows: in the next section we revisit previous work on Disclosure Attacks and we describe our system and adversarial models in Sect. 3. We introduce the Least Squares approach to disclosure applied to threshold mixes in Sect. 4, and extend it to account for the pool mix in Sect. 5. In both sections we derive equations that characterize the LSDA's error with respect to the system parameters which we validate in Sect. 6. We discuss future lines of work in Sect. 7, and we conclude in Sect. 8.

2 An overview of Disclosure Attacks

The first Disclosure Attack [6, 16] relies on Graph Theory to uncover the recipient set of a target user Alice. It identifies the set of Alice's friends by seeking for mutually disjoint sets of receivers among the recipient anonymity sets of the messages sent by Alice. The main drawback of this approach is that it is equivalent to solving a Constraint Satisfaction Problem which is well-known to be NP-complete.

The subfamily of Hitting Set Attacks [11, 14] speeds up the search for Alice's messages recipients by restricting the search to unique minimal hitting sets. Pham et al. studied the relationship between the number of observed rounds to uniquely identify the set of receivers and the parameters of the system [14]. This evaluation is similar to our work in spirit, but it focuses on attacks that unambiguously identify recipient sets while we deal with statistical attacks that only provide an estimation of such sets as the ones discussed below.

The Statistical Disclosure Attack (SDA) and its sequels [8, 9, 12, 13] estimate Alice's sending profile by averaging the probability distributions describing the recipient anonymity set [17] of her messages. These distributions are computed considering that the recipient anonymity set of a message is uniform over the receivers present in a round (and zero for the rest of users).

Troncoso et al. proposed in [15] two attacks: the Perfect Matching Disclosure Attack (PMDA) and the Normalized Statistical Disclosure Attack (NSDA). These attacks exploit the fact that the relationship between sent and received messages in a round must be one-to-one to improve the accuracy of the estimated profiles. The PMDA accounts for this interdependency by searching for perfect matchings in the underlying bipartite graph representing a mix round, while the NSDA normalizes the adjacency matrix representing this graph. The recipient anonymity set of a message is built based on the result of this assignment, instead of assigning uniform probabilities among all recipients as the SDA does.

Last, Danezis and Troncoso propose to use of Bayesian sampling techniques to co-infer user communication profiles and de-anonymize messages [10]. The Bayesian approach can be adapted to analyze arbitrarily complex systems and outputs reliable error estimates, but it requires the adversary to repeatedly seek for perfect matchings increasing the computational requirements of the attack.

From all of the aforementioned attacks, only the SDA has been extended to take into account pool mixes. The probabilistic behavior of these mixes makes it difficult to unambiguously identify sets of receivers, and to define a bipartite graph describing the relationship between inputs and outputs in a round. Hence, it is not trivial to extend the Disclosure and Hitting Set attacks [6, 11], or the PMDA, NSDA [15], and Vida [10] to analyze pool mixes. We will show that our Least Squares approach can be easily adapted to the pool mix probabilistic behavior, and that it obtains much better results than the SDA.

We note that previous authors evaluated the attacks either from mostly a de-anonymization of individual messages perspective (e.g., [10, 15]), or from the point of view of the number of rounds necessary to identify a percentage of Alice's recipients (e.g., [13, 12, 14]). In this work we are interested in the accuracy with which the adversary can infer the sender (respectively, receiver) profile of Alice, i.e., we not only seek to identify Alice's messages receivers, but also to estimate the probability that Alice sends (or receives) a message to (from) them.

3 System model

In this section we describe our model of an anonymous communication system and introduce the notation we use throughout the paper, which we summarize in Table 3.0.2. Throughout the text, we will use capital letters to denote random variables and lowercase letters to denote realizations. Vectors will be represented using boldface characters; thus, $\mathbf{x} = [x_1, \cdots, x_N]^T$ denotes a realization of random vector $\mathbf{X} = [X_1, \cdots, X_N]^T$. Matrices will be represented by boldface capital characters; whether they contain random or specific values will be clear from the context. We will use $\mathbf{1}_N$ to denote the column vector whose N elements are 1; similarly, $\mathbf{1}_{N\times M}$ denotes the all-ones matrix of size $N \times M$.

3.0.1 System model.

We study a system in which a population of N_{users} users, designated by an index $i \in \{1, \ldots, N_{\text{users}}\}$, communicate through an high-latency anonymous communication channel. We consider two types of mixes:

- **Threshold Mix**: This mix gathers t messages each round, transforms them cryptographically, and outputs them in a random order; hence hiding the correspondence between incoming and outgoing messages.

- **Binomial Threshold Pool Mix**: Similarly to the threshold mix, the pool mix collects t messages per round and alters their appearance to avoid bitwise linkability. However, instead of outputting them immediately, messages are placed on a pool and only leave the mix with probability α. Otherwise, they stay and get mixed with messages arriving in subsequent rounds. This behavior increases the adversary's confusion on the correspondence between inputs and outputs.

We model the number of messages that the ith user sends in round r as the random variable X_i^r; and denote as x_i^r the actual number of messages i sends in that round. Similarly, Y_j^r is the random variable that models the number of messages that the jth user receives in round r; and y_j^r the actual number of messages j receives in that round. Let $\mathbf{x}^r$ and $\mathbf{y}^r$ denote column vector that contain as elements the number of messages sent or received by all users in round r: $\mathbf{x}^r = [x_1^r, \cdots, x_{N_{\text{users}}}^r]^T$, and $\mathbf{y}^r = [y_1^r, \cdots, y_{N_{\text{users}}}^r]^T$, respectively. When it is clear from the context, the superindex r is dropped.

Users in our population choose the recipients of their messages according to their sending profile $\mathbf{q}_i \doteq [p_{1,i}, p_{2,i}, \cdots, p_{N_{\text{users}},i}]^T$; where $p_{j,i}$ models the probability that user i sends a message to user j. We consider that a user i has f friends to whom she sends with probability $p_{j,i}$, and assign $p_{j,i} = 0$ for each user j that is not a friend of i. Conversely, $\mathbf{p}_j$ is the column vector containing the probabilities of those incoming messages to the jth user, i.e., $\mathbf{p}_j \doteq [p_{j,1}, p_{j,2}, \cdots, p_{j,N_{\text{users}}}]^T$. (This vector can be related to the receiving profile of user j through a simple normalization, i.e., by dividing its components by $\sum_{i=1}^{N_{\text{users}}} p_{j,i}$.) Let $\mathbf{P}$ be the matrix of transition probabilities whose j, ith element is $p_{j,i}$; with the previous definitions, this matrix can be written as $\mathbf{P} = [\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{N_{\text{users}}}]$ or, equivalently, $\mathbf{P}^T = [\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{N_{\text{users}}}]$. We denote as f_j the number of senders that send messages to receiver j (i.e., the cardinality of the set $\mathcal{F}_j = \{i | p_{j,i} > 0, \, p_{j,i} \in \mathbf{p}_j\}$); and define $\tau_f \doteq \sum_{i=1}^{N_{\text{users}}} f_i^2/(f^2 N_{\text{users}})$, which shall come handy in the performance evaluation performed in Sect. 6.

3.0.2 Adversary model.

We consider a global passive adversary that observes the system during ρ rounds. She can observe the identity of the senders and receivers that communicate through the mix. Furthermore, she knows all the parameters of the mix (e.g. t and α). As our objective is to illustrate the impact of disclosure attacks on the anonymity provided by the mix we assume that the cryptographic transformation performed during the mixing is perfect and thus the adversary cannot gain any information from studying the content of the messages.

Table 1: Summary of notation

Symbol	Meaning
N_{users}	Number of users in the population, denoted by $i = \{1, \cdots, N_{\text{users}}\}$
f	Number of friends of each sender i
t	Threshold of the threshold/pool mix
α	Firing probability of the binomial pool mix
f_j	Number of senders sending messages to receiver j
τ_f	$\sum_{j=1}^{N_{\text{users}}} f_j^2/(f^2 N_{\text{users}})$
$p_{j,i}$	Probability that user i sends a message to user j
$\mathbf{q}_i$	Sender profile of user i, $\mathbf{q}_i = [p_{1,i}, p_{2,i}, \cdots, p_{N_{\text{users}},i}]^T$
$\mathbf{p}_j$	Unnormalized receiver profile of user j, $\mathbf{p}_j = [p_{j,1}, p_{j,2}, \cdots, p_{j,N_{\text{users}}}]^T$
$\mathbf{P}$	Transition probabilities matrix.
ρ	Number of rounds observed by the adversary
x_i^r (y_j^r)	Number of messages that the ith (jth) user sends (receives) in round r
$\mathbf{x}^r$ $(\mathbf{y}^r)$	Column vector containing elements x_i^r (y_j^r), $i = 1, \cdots, N_{users}$
$\hat{p}_{j,i}$	Adversary's estimation of $p_{j,i}$
$\hat{\mathbf{q}}_\mathbf{i}$	Adversary's estimation of user i's sender profile $\mathbf{q}_i$
$\hat{\mathbf{p}}_\mathbf{j}$	Adversary's estimation of user j's unnormalized receiver profile $\mathbf{p}_j$
$\hat{\mathbf{P}}$	Adversary's estimation of transition probabilities matrix.

The adversary's goal is to uncover communication patterns from the observed flow of messages. Formally, given the observation $\mathbf{x}^r = \{x_i^r\}$ and $\mathbf{y}^r = \{y_j^r\}$, for $i, j = 1, \ldots, N_{\text{users}}$, and $r = 1, \ldots, \rho$, the adversary's goal is to obtain estimates $\hat{p}_{j,i}$ as close as possible to the probabilities $p_{j,i}$, which in turn allow her to recover the users' sender and receiver profiles.

4 A Least Squares approach to Disclosure Attacks on Threshold Mixes

We aim here at deriving a profiling algorithm based on the Maximum Likelihood (ML) approach to recover the communication patterns of users anonymously communicating through a threshold mix. The general idea is to be able to estimate the probabilities $p_{j,i}$ that user i sends a message to user j.

We make no assumptions on the user's profiles (i.e., we impose no restrictions on the number of friends a user may have, nor on how messages are distributed among them). Nevertheless, we follow the standard assumptions regarding users' behavior and consider that they are memoryless (i.e., for a user the probability of sending a message to a specific receiver does not depend on previously sent messages), independent (i.e., the behavior of a certain user is independent from the others), with uniform priors (i.e., any incoming message to the mix is a priori sent by any user with the same probability), and stationary (i.e., the parameters modeling their statistical behavior do not change with time).

4.1 A Least Squares estimator

Our aim is to estimate the users' profiles given the observed vectors $\mathbf{x}^r$ and $\mathbf{y}^r$, $r = 1, \cdots, N_{\text{users}}$. Instead of focusing on a single user at a time, as the SDA does, we want to simultaneously find the profiles for all users. To this end, we form the following vectors/matrices:

$$\begin{aligned}
\mathbf{Y}^T &\doteq [Y_1^1, Y_1^2, \cdots, Y_1^\rho, Y_2^1, Y_2^2, \cdots, Y_2^\rho, \cdots, Y_{N_{\text{users}}}^1, Y_{N_{\text{users}}}^2, \cdots, Y_{N_{\text{users}}}^\rho] \\
\mathbf{U}^T &\doteq [\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^\rho] \\
\mathbf{H} &\doteq \mathbf{I}_{N_{\text{users}}} \otimes \mathbf{U}
\end{aligned}$$

Vector $\mathbf{Y}$ basically stacks all the outputs from the mix, while matrix $\mathbf{U}$ contains all the observed inputs. In [18] we discuss how the Maximum Likelihood (ML) principle can be applied to our problem: we want to find the vector $\mathbf{p}^T \doteq [\mathbf{p}_1^T, \mathbf{p}_2^T, \cdots, \mathbf{p}_{N_{\text{users}}}^T]$ containing all the profiles, such that the probability of observing the output given the input, i.e., $\Pr(\mathbf{Y} = \mathbf{y}|\mathbf{x}^1, \cdots, \mathbf{x}^\rho)$, is maximum. The probability distribution of $\mathbf{Y}$ given the input, which is given by a vector of multinomial distributions, depends on $\mathbf{p}$, so the maximization can be carried out with respect to the transition probabilities. However, as discussed in [18], this dependence is highly nonlinear, thus complicating the search for a solution even for simple cases. A further step is then taken in [18], where it is shown that $\Pr(\mathbf{Y}|\mathbf{x}^1, \cdots, \mathbf{x}^\rho)$ approximately follows a Gaussian distribution with mean $\mathbf{Hp}$ and a covariance matrix $\mathbf{\Sigma}_y$ which also depends on $\mathbf{p}$.

After this approximation, the optimization problem can be explicitly written as follows:

$$\hat{\mathbf{p}} = \arg\max_{\mathbf{p} \in \mathcal{P}} \frac{1}{\sqrt{\det(\mathbf{\Sigma}_y)}} \cdot \exp\left(-\frac{1}{2} (\mathbf{y} - \mathbf{Hp})^T \mathbf{\Sigma}_y^{-1} (\mathbf{y} - \mathbf{Hp}) \right), \qquad (1)$$

where $\mathcal{P}$ denotes the set of valid probability vectors.[1]

The dependence of $\mathbf{\Sigma}_y$ with $\mathbf{p}$ still makes the solution to (1) very complicated, so one simplification consists in assuming $\mathbf{\Sigma}_y \approx \text{diag}([\text{Var}\{Y_1^1\}, \cdots, \text{Var}\{Y_{N_{\text{users}}}^\rho\}])$, that is reasonable because the covariance between Y_j^r and Y_k^s is zero for $r \neq s$, and small (compared to diagonal terms) when $r = s$ and $k \neq j$ (see [18] for more details). With this simplification, the maximization in

[1] Without further constraints, that may be furnished when there is partial knowledge about the transition probabilities, $\mathcal{P}$ is simply given by the constraints $0 \leq p_{j,i} \leq 1$ for all j, i, and $\sum_{j=1}^{N_{\text{users}}} p_{j,i} = 1$, for all i.

(1) is equivalent to minimizing the exponent; this leads to a *weighted* Least Squares problem. When the variances of all the elements in $\mathbf{Y}$ are similar, it is simpler to minimize just $(\mathbf{y}-\mathbf{Hp})^T(\mathbf{y}-\mathbf{Hp})$, which can be summarized in the following constrained Least Squares problem

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}\in\mathcal{P}} ||\mathbf{y}-\mathbf{Hp}||^2\,, \tag{2}$$

Notice that the constraints come from the fact that $\mathbf{p}$ must contain valid probability profiles. Interestingly, removing the constraints leads to a solution that is not only amenable to an in-depth performance analysis, as we will confirm in Sect. 4.2, but is also asymptotically *efficient*, in the sense that $\hat{\mathbf{p}}$ converges to the true profiles $\mathbf{p}$ as $\rho \to \infty$, .

It is well known that for the unconstrained case the solution is provided by the Moore-Penrose pseudoinverse [19]:

$$\hat{\mathbf{p}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}\,. \tag{3}$$

At first sight, it might look that the matrix inversion needed in (3) is formidable: the matrix $\mathbf{H}^T\mathbf{H}$ has size $N^2_{\text{users}} \times N^2_{\text{users}}$. However, its block-diagonal structure allows for a more affordable solution; indeed,

$$\mathbf{H}^T\mathbf{H} = (\mathbf{I}_{N_{\text{users}}} \otimes \mathbf{U})^T \cdot \mathbf{I}_{N_{\text{users}}} \otimes \mathbf{U} = \mathbf{I}_{N_{\text{users}}} \otimes (\mathbf{U}^T\mathbf{U})$$

and, hence,

$$(\mathbf{H}^T\mathbf{H})^{-1} = \mathbf{I}_{N_{\text{users}}} \otimes (\mathbf{U}^T\mathbf{U})^{-1}$$

where $\mathbf{U}^T\mathbf{U}$ of size $N_{\text{users}} \times N_{\text{users}}$ is assumed to have full rank.

The decoupling above allows us to write a more efficient solution as follows. Let $\mathbf{y}_j \doteq [y_j^1, y_j^2, \cdots y_j^\rho]^T$. Then, the LS estimate $\hat{\mathbf{p}}_j$ for the jth probability vector can be written as

$$\hat{\mathbf{p}}_j = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{y}_j, \quad j = 1, \cdots, N_{\text{users}}\,. \tag{4}$$

The decoupling is possible only in the unconstrained case; this consideration, together with the simplicity of the performance analysis, make us focus mostly on the unconstrained LS approach. Notice, however, that, as a consequence, the obtained solution is not guaranteed to meet the constraints on the transition probabilities. This can be overcome by projecting the solution onto the set $\mathcal{P}$, as it will be discussed later. In any case, the fact that, as we will later prove, the error between the actual and estimated values $\mathbf{p}-\hat{\mathbf{p}}$ tends to zero as $\rho \to \infty$, ensures that $\hat{\mathbf{p}}$ can be made arbitrarily close to $\mathcal{P}$ by increasing the number of observed rounds. Finally, note that when $\hat{\mathbf{p}}_j$ is computed for all users, it is also possible to recover the sender profiles $\mathbf{q}_i$ by taking the rows of the matrix $\hat{\mathbf{P}}$.

We point out that the LS estimate can be interpreted as a linear predictor: given the inputs $\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^\rho$, we construct a linear predictor for the observations $\mathbf{y}$ that minimizes the mean squared prediction error. We will call the attack described throughout this section Least Squares Disclose Attack (LSDA).

The Statistical Disclosure Attack as an LS estimator. We now show that the original Statistical Disclosure Attack [7] in fact corresponds to a particular case of the proposed LSDA estimator. Here, the first user (Alice) is supposed to send only one message to an unknown recipient chosen uniformly from a set of f friends. The other users are assumed to send messages to recipients chosen uniformly from the set of all users. The target is to determine the set of friends of Alice.

From these considerations, for a given round r where Alice does send a message, we have that $x_1^r = 1$ and $\sum_{i=2}^{N_{\text{users}}} x_i^r = (t-1)$, and all the transition probabilities $p_{j,i}$, for $i \geq 2$, $j = 1, \cdots, N_{\text{users}}$, are known to be equal to $1/N_{\text{users}}$. If we suppose that in all rounds Alice transmits a message, we will have a vector $\mathbf{y}$ which contains the $\rho \cdot N_{\text{users}}$ observations, $\mathbf{q}_1$ is unknown, and all $\mathbf{q}_i$, $i = 2, \cdots, N_{\text{users}}$ are known. The unconstrained unweighted LSDA estimator can be broken down into subproblems in which we seek $\mathbf{p}_j$, for all $j = 1, \cdots, N_{\text{users}}$, such that

$$||\mathbf{y}_j - \mathbf{U}\mathbf{p}_j||^2 \tag{5}$$

is minimized. Noticing that for each $\mathbf{p}_j$ only $p_{j,1}$ is unknown, we can write the equivalent problem of finding $p_{j,1}$ such that

$$||\mathbf{y}_j - \mathbf{U}'\mathbf{p}'_j - p_{j,1}\mathbf{U}_1||^2 \tag{6}$$

is minimized, where $\mathbf{U}'$ is obtained from $\mathbf{U}$ by deleting its first column, itself denoted by $\mathbf{U}_1$, and where $\mathbf{p}'_j$ is obtained from $\mathbf{p}_j$ after deleting its first element.

Then, the LS solution is

$$\hat{p}_{j,1} = (\mathbf{U}_1^T\mathbf{U}_1)^{-1}\mathbf{U}_1^T(\mathbf{y}_j - \mathbf{U}'\mathbf{p}'_j) \tag{7}$$

From the fact that $\mathbf{U}_1 = \mathbf{1}_\rho$ (as Alice sends one and only one message per round), it follows that $\mathbf{U}_1^T\mathbf{U}_1 = \rho$. On the other hand, all elements in $\mathbf{p}'_j$ take the value $1/N_{\text{users}}$ and the matrix $\mathbf{U}'$ is such that the sum of the elements in each column is $(t-1)$; therefore,

$$\hat{p}_{j,1} = \frac{1}{\rho}\sum_{r=1}^{\rho} y_j^r - \frac{(t-1)}{N_{\text{users}}}, \quad j = 1, \cdots, N_{\text{users}} \tag{8}$$

which coincides with the SDA estimate.

The LSDA estimator differs from the SDA one in that it does not make any underlying assumption on the transition probabilities and that it simultaneously solves for the entire matrix of transition probabilities.

4.2 Performance analysis with respect to the system parameters

Next, we assess the performance of our solution. This will serve to understand the influence of the system parameters on the knowledge that can be gained by an adversary applying our algorithm. To this end, we first remark that the Least Squares estimate in (3) is unbiased: it is straightforward to show that $\mathrm{E}[\hat{\mathbf{p}}] = \mathbf{p}$. On the other hand, the covariance matrix of $\hat{\mathbf{p}}$, for a fixed matrix $\mathbf{H}$, is given by [19]

$$\mathrm{E}[(\mathbf{p}-\hat{\mathbf{p}})(\mathbf{p}-\hat{\mathbf{p}})^T] = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{\Sigma}_y\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\,. \tag{9}$$

From this covariance matrix it is immediate to calculate the total Mean Square Error (MSE) in the estimation of the profiles, since

$$\mathrm{MSE} \doteq \mathrm{E}\sum_{i=1}^{N_{\mathrm{users}}}\sum_{j=1}^{N_{\mathrm{users}}} |p_{j,i}-\hat{p}_{j,i}|^2 = \mathrm{E}[\mathrm{tr}\left((\mathbf{p}-\hat{\mathbf{p}})(\mathbf{p}-\hat{\mathbf{p}})^T\right)] \tag{10}$$

where $\mathrm{tr}(\mathbf{M})$ denotes the trace of matrix $\mathbf{M}$.

Due to the decoupling for the estimation of each $\mathbf{p}_j$ discussed above, we can write a similar equation to (9) for each unnmormalized receiver profile, that is,

$$\mathrm{E}[(\mathbf{p}_j-\hat{\mathbf{p}}_j)(\mathbf{p}_j-\hat{\mathbf{p}}_j)^T] = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{\Sigma}_{y_j}\mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}, \text{ for all } j=1,\cdots,N_{\mathrm{users}} \tag{11}$$

keeping in mind that

$$\mathrm{MSE} = \sum_{j=1}^{N_{\mathrm{users}}} \mathrm{tr}\left(\mathrm{E}[(\mathbf{p}_j-\hat{\mathbf{p}}_j)(\mathbf{p}_j-\hat{\mathbf{p}}_j)^T]\right) \tag{12}$$

Notice from (11) that the performance will depend on the actual input matrix $\mathbf{U}$; however, since the input process is wide-sense stationary and ergodic, when $\rho \to \infty$ the block $\mathbf{U}^T\mathbf{U}$ will converge to the input autocorrelation matrix $\mathbf{R}_x$. Then, when the number of observations is large, approximating $\mathbf{U}^T\mathbf{U} \approx \mathbf{R}_x$ will allow us to extract some quantitative conclusions that are independent of $\mathbf{U}$. We further point out that since $\mathrm{Cov}(Y_i^r, Y_i^s) = 0$ for all i and $r \neq s$, then $\mathbf{\Sigma}_{y_j} = \sigma_{y_j}^2\mathbf{I}_\rho$, where $\sigma_{y_j}^2 \doteq \mathrm{Var}\{Y_j\}$.

In this case, (11) becomes

$$\mathrm{E}[(\mathbf{p}_j-\hat{\mathbf{p}})(\mathbf{p}-\hat{\mathbf{p}}_j)^T] = \sigma_{y_j}^2(\mathbf{U}^T\mathbf{U})^{-1}\,. \tag{13}$$

We still need to quantify how large $(\mathbf{U}^T\mathbf{U})^{-1}$ is. Since $\mathbf{U}^T\mathbf{U}$ is symmetric, we can write the following eigendecomposition

$$\mathbf{U}^T\mathbf{U} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}\,, \tag{14}$$

where $\mathbf{Q}$ is orthonormal and $\mathbf{\Lambda}$ is diagonal. In this case, $(\mathbf{U}^T\mathbf{U})^{-1} = \mathbf{Q}^{-1}\mathbf{\Lambda}^{-1}\mathbf{Q}$. Then, if we define the *transformed probability space* where $\mathbf{p}'_j \doteq \mathbf{Q}\mathbf{p}_j$ and $\hat{\mathbf{p}}'_j \doteq \mathbf{Q}\hat{\mathbf{p}}_j$ we have

$$\mathrm{E}[(\mathbf{p}_j - \hat{\mathbf{p}}_j)(\mathbf{p}_j - \hat{\mathbf{p}}_j)^T] = \mathrm{E}[(\mathbf{p}'_j - \hat{\mathbf{p}}'_j)(\mathbf{p}'_j - \hat{\mathbf{p}}'_j)^T] = \sigma^2_{y_j}\mathbf{\Lambda}^{-1} \tag{15}$$

Then, computing the trace of (15) and substituting into (12) we can express the total MSE as

$$\mathrm{MSE} = \sum_{j=1}^{N_{\text{users}}} \sigma^2_{y_j} \cdot \sum_{i=1}^{N_{\text{users}}} \lambda^{-1}_{u,i} \tag{16}$$

where $\lambda_{u,j}$, $j = 1, \cdots, N_{\text{users}}$ denote the eigenvalues of $\mathbf{U}^T\mathbf{U}$.

Equation (16) can be interpreted as having two terms that depend on the output covariance and input autocorrelation, respectively. In fact, for some cases of interest, it is possible to derive explicit expressions, as we discuss next.

Consider the case where each user has exactly the same probability $1/N_{\text{users}}$ of sending a message to one of her friends and that each message is sent independently. Then, if t messages are sent per round, the observed input vector at the jth round $\mathbf{x}^j$ will follow a multinomial distribution for which

$$\mathrm{E}\{X_i^2\} = t^2p_x^2 + tp_x(1-p_x), \;\text{ and }\; \mathrm{E}\{X_iX_k\} = t^2p_x^2 - tp_x^2, \;\; i \neq k$$

where $p_x = 1/N_{\text{users}}$. Then, the autocorrelation matrix $\mathbf{R}_x$ can be shown to have $(N_{\text{users}} - 1)$ identical eigenvalues which are equal to $\rho \cdot t \cdot p_x$ and the remaining eigenvalue equal to $\rho \cdot t \cdot p_x + \rho \cdot t \cdot p_x^2(t-1)N_{\text{users}}$. Therefore,

$$\sum_{j=1}^{N_{\text{users}}} \lambda^{-1}_{u,j} = \frac{N_{\text{users}}}{\rho t}\left(N_{\text{users}} - 1 + \frac{1}{t}\right) \tag{17}$$

Next we focus on the output variance. We consider the case where each user has f friends in her sending profile to whom she sends messages with probability $1/f$ each. Let $\mathcal{F}_j$ be the set of users that send messages to the jth user with non-zero probability, and let f_j be its cardinality. Then, for the input conditions discussed in the previous paragraph (i.e., i.i.d. uniform users), the probability that one given message is sent by one user in $\mathcal{F}_j$ is f_j/N_{users}. In turn, the probability that one message originating from a user in $\mathcal{F}_j$ is sent to the jth user is $1/f$. Therefore, we can see Y_j^k as the output of a binomial process with probability

$$p_{y_j} = \frac{f_j}{fN_{\text{users}}},$$

and with t messages at its input. Hence, the variance of Y_j is

$$\sigma^2_{y_j} = t \cdot p_{y_j}(1 - p_{y_j}) = \frac{t \cdot f_j}{f \cdot N_{\text{users}}} \cdot \left(1 - \frac{f_j}{f \cdot N_{\text{users}}}\right),$$

so the sum of variances becomes

$$\sum_{j=1}^{N_{\text{users}}} \sigma_{y_j}^2 = t\left(1 - \frac{\sum_{j=1}^{N_{\text{users}}} f_j^2}{f^2 N_{\text{users}}^2}\right) = t\left(1 - \frac{\tau_f}{N_{\text{users}}}\right) , \tag{18}$$

where we have used the fact that $\sum_{j=1}^{N_{\text{users}}} f_j = f \cdot N_{\text{users}}$, and τ_f is defined in Table 3.0.2.

Combining (17) and (18) we can write the MSE as

$$\text{MSE} = \frac{1}{\rho}\left(N_{\text{users}} - 1 + \frac{1}{t}\right) \cdot (N_{\text{users}} - \tau_f) . \tag{19}$$

It is useful to interpret (19) in terms of the number of friends of each receiver. We will consider two particular cases of interest: 1) If each receiver has exactly f friends, then $\tau_f = \tau_{f,1} = 1$; 2) If only f receivers have N_{users} friends, and the remaining $N_{\text{users}} - f$ receivers have no friends, then $\tau_f = \tau_{f,2} = N_{\text{users}}/f$. The second case models a situation where f receivers act as hubs (i.e., f users concentrate the traffic of all the population), while in the first there is absolutely no skew in the distribution. In fact, using the Lagrange multipliers technique, it can be shown that for all other cases, including random connections (but always keeping the constraint that each sender has exactly f friends), the parameter τ_f satisfies that $\tau_{f,1} \leq \tau_f \leq \tau_{f,2}$. Since (19) monotonically decreases with τ_f, we can conclude that for the symmetric case (i.e., $\tau_f = 1$) the MSE is larger, revealing that it will be harder to learn the transition matrix.

When N_{users} is large, we can approximate (19) as follows

$$\text{MSE} \approx \frac{N_{\text{users}}^2}{\rho} . \tag{20}$$

If we recall that there are N_{users}^2 probabilities to estimate from the transition matrix, we can conclude that the variance *per transition element* $p_{j,i}$ is approximately $1/\rho$. The total MSE decreases as $1/\rho$ with the number of rounds ρ; this implies that the unconstrained, unweighted LS estimator is asymptotically efficient as $\rho \to \infty$. Even though this is somewhat to be expected, notice that other simpler estimators might not share this desirable property, as we will experimentally confirm in Sect. 6.

4.3 Constrained Least Squares Estimation

One interesting property of the unconstrained LS estimator proposed above is that the estimated sender profiles satisfy that $\sum_{j=1}^{N_{\text{users}}} \hat{p}_{j,i} = 1$, for all i, or in short, $\mathbf{1}_{N_{\text{users}}}^T \hat{\mathbf{q}}_i = 1$, for all i. An even more compact form is

$$\mathbf{1}_{N_{\text{users}}}^T \hat{\mathbf{P}} = \mathbf{1}_{N_{\text{users}}}^T . \tag{21}$$

To prove this property, recall that $\hat{\mathbf{P}}^T \doteq [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, \cdots, \hat{\mathbf{p}}_{N_{\text{users}}}]$, so (4) can be rewritten as

$$\hat{\mathbf{P}}^T = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{V}^T, \tag{22}$$

where $\mathbf{V}^T \doteq [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_{N_{\text{users}}}]$. For convenience, let $\mathbf{z}^T \doteq \mathbf{1}^T_{N_{\text{users}}}\hat{\mathbf{P}}$; then, we want to show that $\mathbf{z}^T = \mathbf{1}^T_{N_{\text{users}}}$. To this end, we write $\mathbf{z}^T = \mathbf{1}^T_{N_{\text{users}}} + \mathbf{z}'^T$ and derive the following chain of identities starting from (21) after substituting (22):

$$
\begin{align}
& (\mathbf{1}^T_{N_{\text{users}}} + \mathbf{z}'^T)\mathbf{U}^T\mathbf{U} = \mathbf{1}^T_{N_{\text{users}}}\mathbf{V}^T\mathbf{U} \tag{23}\\
\Rightarrow\quad & \mathbf{1}^T_{N_{\text{users}}}\mathbf{U}^T\mathbf{U} + \mathbf{z}'^T\mathbf{U}^T\mathbf{U} = t\mathbf{1}^T_{\rho}\mathbf{U} \tag{24}\\
\Rightarrow\quad & t\mathbf{1}^T_{\rho}\mathbf{U} + \mathbf{z}'^T\mathbf{U}^T\mathbf{U} = t\mathbf{1}^T_{\rho}\mathbf{U} \tag{25}\\
\Rightarrow\quad & \mathbf{z}'^T\mathbf{U}^T\mathbf{U} = \mathbf{0} \tag{26}\\
\Rightarrow\quad & \mathbf{z}'^T = \mathbf{0} \tag{27}
\end{align}
$$

which yields the desired proof. In the previous chain, we have used the fact that $\mathbf{U}^T\mathbf{U}$ is non-singular and that $\mathbf{1}^T_{N_{\text{users}}}\mathbf{U}^T = \mathbf{1}^T_{N_{\text{users}}}\mathbf{V}^T = t\mathbf{1}^T_{\rho}$. Therefore, this property holds if the number of messages per round is constant, but not necessarily in more general cases.

While the unconstrained LS estimator yields sender profiles satisfying $\sum_{j=1}^{N_{\text{users}}} \hat{p}_{j,i} = 1$, for all i, the output of the estimator is not guaranteed to be a proper probability distribution representing a user's profile. The fact that we have considered user profiling as an unconstrained problem will often result in some of the estimated probabilities $\hat{p}_{j,i}$ being negative, usually corresponding to receivers j that are not friends of user i. Thus, when $p_{j,i} = 0$ the algorithm returns $\hat{p}_{j,i}$ that lie near zero, but as the solution is unconstrained, it is not guaranteed that $\hat{p}_{j,i} \geq 0$ as one would desire.

One could reduce the error by just setting those probabilities to zero, disregarding that $\sum_j p_{j,i} = 1$ for all i. This constraint could be ensured by normalizing the profile, but this normalization has to be performed without information and hence the estimation is not guaranteed to be optimal.

Alternatively, it is possible to recover the constrained problem in (2). We recall that the constraints are $1 \geq p_{i,j} \geq 0$, for all $i, j = 1, \cdots, N_{\text{users}}$, and $\sum_{j=1}^{N_{\text{users}}} p_{j,i} = 1$, for all $i = 1, \cdots, N_{\text{users}}$. One might think of imposing such constraints to the decoupled optimization problems (5) for each j. Unfortunately, while the optimization is performed with respect to $\mathbf{p}_j$, each of the previous sum constraints is given in terms of $\mathbf{q}_i$. Hence, if those constraints are to be enforced, then the optimization problems can no longer be decoupled. In such case, to reduce the complexity of the search, it is possible to use an *alternating projection* strategy which succesively fixes all but the ith sender profiles, and then solves the resulting constrained least squares problem for the estimator $\hat{\mathbf{q}}_i$, where now there are only N_{users} unknowns, one equality constraint (i.e., $\mathbf{1}^T_{N_{\text{users}}}\hat{\mathbf{q}}_i = 1$), and N_{users} inequality constraints (i.e., $\hat{p}_{j,i} \geq 0$, for all $j = 1, \cdots, N_{\text{users}}$). Once the optimization for the ith sender is completed, the estimator $\hat{\mathbf{P}}$ is modified by updating its ith row with the newly obtained $\hat{\mathbf{q}}_i$, and then the process continues with the $(i+1)$th user until all the N_{users} users have been swept.

As a final remark, we note that the constraints make a performance analysis similar to that in Section 4.2 much more cumbersome. The analysis of such solution is left as subject for future research.

5 A Least Squares approach to Disclosure Attacks on Pool Mixes

In this section we show how to extend the derivations given for the threshold mix case to the case of a pool mix. The main difference with respect to the threshold mix arises from the fact that the output of the mix is now a probabilistic function of the input observations, so it is no longer possible for the adversary to know how many messages from the ith user are sent in the rth round. To distinguish between the number of messages from the ith sender that enter and leave (note that some messages may stay in the pool) the mix in round r, we will respectively use vectors $\mathbf{x}^r$ and $\mathbf{X}_s^r$. Thus, the vector $\mathbf{x}^r$ is observable, while $\mathbf{X}_s^r$ is not. We let $\mathbf{U}_s^T \doteq [\mathbf{X}_s^1, \mathbf{X}_s^2, \cdots, \mathbf{X}_s^\rho]$.

We focus our analysis on a threshold binomial pool mix, where each message in the pool has a probability α of leaving and $(1-\alpha)$ of remaining in the pool. We assume that at the time the adversary starts her observation the pool contains m messages whose sender is unknown. Then, the messages in X_s^r may come from two sources: the initial m messages in the pool, and the messages observed at the input of the mix in the current or earlier rounds. We will use vectors $\mathbf{N}^r$ and $\mathbf{X}_e^r$ to model these two contributions, and write $\mathbf{X}_s^r = \mathbf{X}_e^r + \mathbf{N}^r$. Notice that $\mathbf{N}^r$ can be seen as noise, as only $\mathbf{X}_e^r$ contains information regarding the observed input messages.

Our approach to solving the pool mix problem is to substitute $\mathbf{U}$ in (4) by a predictor of matrix $\mathbf{U}_s$. To this end, the minimum mean squared predictor of $X_{s,i}^r$, given the matrix of observations $\mathbf{U}$, is

$$\begin{aligned} \hat{x}_{s,i}^r &= \mathrm{E}\{X_{s,i}^r|\mathbf{U}\} = \mathrm{E}\{X_{e,i}^r|\mathbf{U}\} + \mathrm{E}\{N_i^r\} \\ &= \alpha \sum_{k=0}^{r-1} (1-\alpha)^k x_i^{r-k} + \alpha(1-\alpha)^{r-1} m/N_{\text{users}} \end{aligned} \tag{28}$$

where we have assumed that each of the m messages in the initial pool may correspond to any of the N_{users} users with uniform probability. For implementation purposes, a more convenient way of writing (28) is the following recursive equation

$$\hat{x}_{s,i}^{r+1} = (1-\alpha)\hat{x}_{s,i}^r + \alpha x_i^{r+1}, \quad r = 1, \cdots, N_{\text{users}} \tag{29}$$

where $\hat{x}^1_{s,i}$ is initialized to $x^1_i + m/N_{\text{users}}$.

The solution to the unconstrained optimization problem is obtained by replacing $\mathbf{U}$ in (4) by $\hat{\mathbf{U}}_s$, where $\hat{\mathbf{U}}_s^T \doteq [\hat{\mathbf{x}}^1_s, \hat{\mathbf{x}}^2_s, \cdots, \hat{\mathbf{x}}^\rho_s]$. For compactness, we will find it useful to define the following *convolution matrix*

$$\mathbf{B} \doteq \begin{bmatrix} \alpha & 0 & 0 & \cdots & 0 \\ \alpha(1-\alpha) & \alpha & 0 & \cdots & 0 \\ \alpha(1-\alpha)^2 & \alpha(1-\alpha) & \alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \alpha(1-\alpha)^{\rho-1} & \alpha(1-\alpha)^{\rho-2} & \alpha(1-\alpha)^{\rho-3} & \cdots & \alpha \end{bmatrix} \tag{30}$$

Then, we can write

$$\hat{\mathbf{U}}_s = \mathbf{B}(\mathbf{U} + \mathbf{N}_0) \tag{31}$$

where the matrix $\mathbf{N}_0$, which accounts for the average initial state of the mix, is such that all entries in the first row take the value m/N_{users}, while all the remaining elements are zero. Then, the solution is

$$\hat{\mathbf{p}}_j = (\hat{\mathbf{U}}_s^T \hat{\mathbf{U}}_s)^{-1} \hat{\mathbf{U}}_s^T \mathbf{y}_j, \quad j = 1, \cdots, N_{\text{users}}\,. \tag{32}$$

Notice that for the standard threshold mix, which corresponds to $\alpha = 1$, $m = 0$, we have that $\mathbf{B} = \mathbf{I}_\rho$, $\mathbf{N}_0 = \mathbf{0}$, so $\hat{\mathbf{U}}_s = \mathbf{U}$, and both solutions coincide.

The performance analysis of this estimator in the case of a pool mix and $\tau_f = 1$ is carried out in Appendix 9, where it is shown that

$$\text{MSE} \approx \frac{N_{\text{users}}(N_{\text{users}} - 1 + \alpha_q/t)}{\rho\alpha_q} - \frac{(N_{\text{users}} - 1)/(2-\alpha) + 1/t}{\rho} \tag{33}$$

where $\alpha_q \doteq \alpha/(2-\alpha)$. This approximation is asymptotically tight as $\rho \to \infty$. Moreover, when $\alpha = 1$ we recover (19). When N_{users} is large, (34) can be approximated as

$$\text{MSE} \approx \frac{N^2_{\text{users}}}{\rho\alpha_q} \tag{34}$$

Comparing the approximation above with (20) we can conclude that the pool mix requires $(2-\alpha)/\alpha$ times more rounds to achieve the same MSE. For instance, for $\alpha = 0.5$, three times more rounds are needed to achieve the same MSE as in the threshold mix. Since α_q monotonically increases with α, the difficulty of learning the profiles is always larger in the pool mix compared to the threshold mix. Of course, this comes at the price of an increased delay; in fact, it can be shown that the *average* delay for a message introduced by the pool, measured in rounds, is $(1-\alpha)/\alpha$.

Finally, we remark that it is also possible to derive a constrained version of the estimator which forces the estimated profiles to lie in the feasible set $\mathcal{P}$. As we will confirm in the evaluation section, the constrained version outperforms the unconstrained one.

Table 2: System parameters used in the experiments.

	Parameter	**Value**
Population	N_{users}	$\{20, 50, \mathbf{100}, 150, 200, 250, 300, 350, 400, 450, 500\}$
	f	$\{5, 10, 15, 20, \mathbf{25}, 30, 35, 40, 45, 50\}$
	τ_f	$\{\mathbf{1.0}, 1.76, 2.44, 3.04, 3.56, 4.0\}$
	ρ	$\{\mathbf{10\,000}, 20\,000, \ldots, 100\,000\}$
Threshold mix	t	$\{2, 5, \mathbf{10}, 20, 30, 40\}$
Binomial Pool mix	α	$\{0.1, 0.2, 0.3, 0.4, \mathbf{0.5}, 0.6, 0.7, 0.8, 0.9\}$
	t	$\{5, \mathbf{10}\}$

6 Evaluation

6.1 Experimental setup

We evaluate the effectiveness of the LSDA approach against synthetic anonymized traces created by a simulator written in the Python language.[2] We simulate a population of N_{users} users with f contacts each, to whom they send messages with equal probability (i.e., $p_{j,i} = 1/f$ if i is friends with j, zero otherwise). In order to easily study the influence of the system parameters on the success of the attack, in our simulations we further fix the senders that send messages to each receiver to be such that $f_j = f$. In other words, every sender (receiver) profile has the same number of non-zero elements, and hence $\tau_f = 1$. In the first part of the evaluation messages are anonymized using a threshold mix with threshold t; and in the second part using a binomial pool mix where each round t messages arrive to the mix[3], and each message in the pool has a probability α of leaving the mix. We consider that the adversary observes ρ rounds of mixing.

Table 6.1 summarizes the values of the parameters used in our experiments, where bold numbers indicate the parameters of the baseline experiment. The values used in our experiments, though rather unrealistic, have been chosen in order to cover a wide variety of scenarios in which to study the performance of the attack while ensuring that experiments could be carried out in reasonable time. We note, however, that the results regarding the LSDA can be easily extrapolated to any set of parameters as long as the proportion among them is preserved. Unfortunately, we cannot make a similar claim for other attacks: their heuristic nature makes it difficult to obtain analytical results that describe the dependence of their success on

[2] The code will be made available upon request.

[3] We fix the number of messages arriving in each round in order to simplify our simulations, but we recall that the method can be adapted to a variable number of arriving messages (see Sect. 5).

the system parameters, and the evolution of their error difficult to predict as we will see throughout this section.

Besides testing the effectiveness of the LSDA when profiling users, we also compare its results with those obtained performing the Statistical Disclosure Attack (SDA) [7, 9], the Perfect Matching Disclosure Attack (PMDA) [15], the Normalized Statistical Disclosure Attack (NSDA) [15], and the Bayesian inference-based attack Vida [10].

6.2 Success metrics

We recall that the goal of the adversary is to estimate the values $p_{j,i}$ with as much accuracy as possible. We define two metrics to illustrate the profiling accuracy of the attacks. The *Mean Squared Error per transition probability* (MSE_p) measures the average squared error between the elements of the estimated matrix $\hat{\mathbf{p}}$ and the elements of the matrix $\mathbf{p}$ describing the actual behavior of the users (see (3)), and is simply the total MSE normalized by the number of elements:

$$\mathrm{MSE}_p = \mathrm{MSE}/N_{\text{users}}^2.$$

Secondly, we define the *Mean Squared Error per sender profile* (MSE_{q_i}):

$$\mathrm{MSE}_{q_i} = \frac{\sum_j (\hat{p}_{j,i} - p_{j,i})^2}{N_{\text{users}}}\ ,\ \ i = 1, \ldots, N_{\text{users}}$$

which measures the average squared error between the the estimated $\hat{\mathbf{q}}_i$ and actual $\mathbf{q}_i$ user i's sender profiles. Both MSEs measure the amount by which the values output by the attack differ from the actual value to be estimated. The smaller the MSE, the better is the adversary's estimation of the users' actual profiles.

For each of the studied set of parameters (N_{users}, f, t, ρ, τ_f) we record the sets of senders and receivers during ρ rounds and compute the MSE_p (or the MSE_{q_i}) for each of the attacks. We repeat this process 20 times and plot the average of the result in our figures.

6.3 Results: Threshold mix

We first study the effectiveness of the LSDA in profiling messages anonymized using a threshold mix in different scenarios.

6.3.1 Performance with respect to the number of rounds ρ

As we discuss in Sect. 4.2, the number of observed rounds ρ has a dominant role in the estimation error incurred by the LSDA. We plot in Fig. 1, left, the MSE per transition probability MSE_p for the SDA, NSDA, PMDA and LSDA.

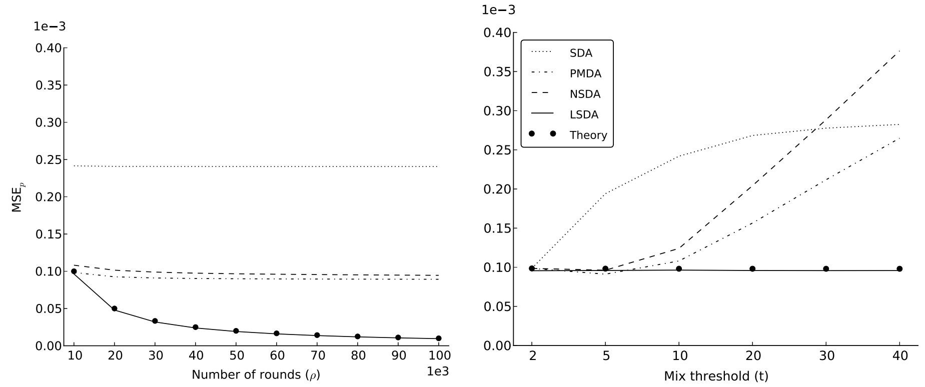

Figure 1: MSE$_p$ evolution with the number of rounds in the system ρ ($N = 100$, $f = 25$, $t = 10$, τ_f=1), and with the threshold mix t ($N = 100$, $f = 25$, $\rho = 10\,000$, $\tau_f = 1$) (left and right, respectively).

The LSDA obtains the best results. Furthermore, we can see how the approximation in (20), represented by • in the figure, reliably describes the decrease in the profile estimation error as more information is made available to the adversary.

It is also interesting to notice how the different attacks take advantage of the information procured by additional rounds. The naive approach followed by the SDA soon maxes out in terms of information extracted from the observation and its MSE$_p$ does not decrease significantly as more rounds are observed, confirming the results in [15]. The NSDA and PMDA perform slightly better in this sense, although their MSE$_p$ also decreases slowly. The LSDA, on the other hand, is able to extract information from each new observed round reducing significantly the MSE$_p$, that tends to zero as $\rho \to \infty$. This is because, as opposed to its predecessors which process the rounds one at a time, the LSDA considers all rounds simultaneously (by means of the matrices $\mathbf{Y}$ and $\mathbf{U}$).

6.3.2 Performance with respect to the mix threshold t

By observing (19) one can see that the threshold t of the mix has little influence on the MSE$_p$ of the LSDA, becoming negligible as t increases and $t \gg 1$. This is reflected by our experiments, shown in Fig. 1, left, where the error of the LSDA soon becomes stable as the threshold of the mix grows. We must note that the time necessary to observe ρ mixing rounds grows with the size of the threshold mix. Hence, although the error is constant with t, increasing the threshold delays the obtention of accurate user profiles.

This property does not hold for the other attacking approaches. As expected, increasing the threshold has a negative effect on the other three attacks. The SDA's error, surprisingly, seems to grow proportionally to

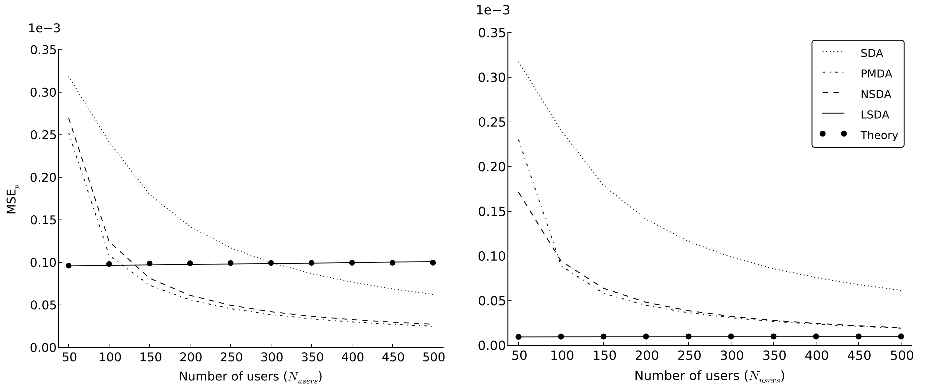

Figure 2: MSE$_p$ evolution with the number of users in the system N_{users} ($f = 25$, $t = 10$, $\rho = 10\,000, 100\,000$, $\tau_f = 1$) (left and right, respectively).

$(1-1/t)$ and thus it is greatly reduced as t increases. This is not the case for the NSDA and PMDA, that rely on solving an optimization problem on the underlying bipartite graph representing a mix round. As the threshold grows this problem becomes harder and hence their MSE$_p$ significantly increases.

6.3.3 Performance with respect to the number of users N_{users}

Next, we study the influence of the number of users in the system on the estimation error. The results are shown in Fig. 2 for $\rho = 10\,000$ (left) and $\rho = 100\,000$ (right). As expected, see (20), the LSDA's MSE$_p$ grows slowly with the number of users. The other three attacks, on the other hand, improve their results when the number of users increase. In this case, if the mix threshold is kept fixed, the intersection between the senders of different mixing rounds becomes smaller, and thus the SDA can better identify their sender profiles. The PMDA and the NSDA use the result of the SDA as attack seed. Hence, the better estimations output by the SDA, the better results obtained by the PMDA and the NSDA.

Even though N_{users} has some effect on the MSE$_p$ of the LSDA the results in Fig. 2 reinforce the idea that the number of rounds ρ is the main component of the error. When $\rho = 10\,000$ rounds are observed, the LSDA does not provide better results than the other attacks. Nevertheless, as the number of rounds increases, the LSDA outperforms the other attacks regardless of the growth of the MSE with N_{users}.

6.3.4 Performance with respect to the output variance $\sigma^2_{y_j}$

The influence on the LSDA's MSE of the output variance $\sigma^2_{y_j}$ can be studied by varying the value of the parameters f and τ_f, while maintaining N_{users} and t constant, see (18). We first vary the number of friends of the senders

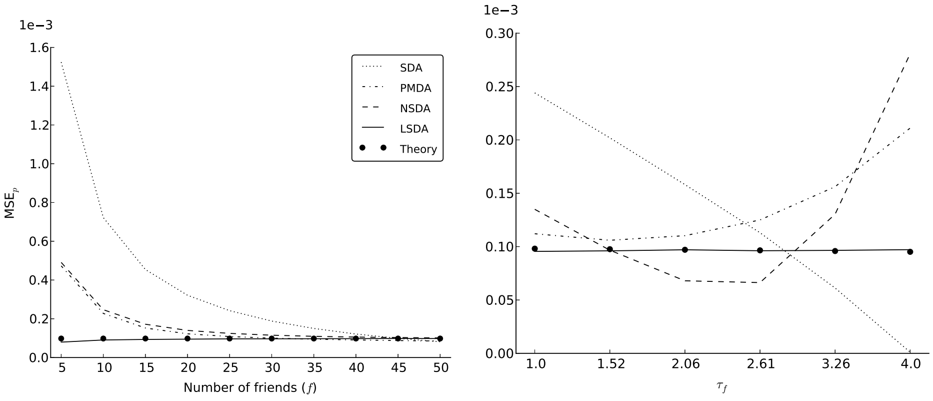

Figure 3: MSE$_p$ evolution with the number of friends f ($N = 100$, $f = 25$, $\rho = 10\,000$, τ_f=1), and with τ_f ($N = 100$, $f = 25$, $t = 10$, $\rho = 10\,000$) (left and right, respectively).

f while keeping $f_j = f$ for all receivers j, ensuring that $\tau_f = 1$. We observe in Fig. 3, left, that the LSDA's MSE$_p$ closely follows the prediction given in (19).

In a second experiment, we fix the parameter f and vary τ_f to represent different degrees of "hubness" in the population. We construct populations such that there are $\alpha = 0, \cdots, f$ hub receivers that have N_{users} friends, while the remaining $N_{\text{users}} - \alpha$ receivers are assigned small amounts of friends in order to obtain different τ_f arbitrarily chosen between $\tau_{f,1} = 1$ and $\tau_{f,2} = N_{\text{users}}/f$. The result is shown in Fig. 3, right. It is worthy to note that the SDA significantly benefits from the hubness of the population. As some users concentrate the traffic the sending profiles become more uniform: all users tend to send their messages to the same set of receivers. In this scenario the strategy of the SDA, that assigns equal probability to every receiver in a mix batch, closely models reality and thus the error tends to zero. While the error of the SDA is very small, the estimated profiles still have small biases toward some users. This effect is amplified by the NSDA and PMDA, significantly increasing their estimation error.

6.3.5 Performance with respect to the user behavior

Our experiments so far considered a very simplistic population in which users choose among their friends uniformly at random (which we denote as SDA). As it has been discussed in the past [10, 15], this population is unlikely to represent real users. We now evaluate the four attacks against two more realistic populations in which users choose the recipients according to an arbitrary multinomial distribution, more (SKW) or less (ARB) skewed depending on the experiment.

We show in Fig. 4 (left) box plots representing the distribution of the MSE per sender profile MSE_{q_i} for all users in the population. We also plot MSE_p for each attack in the figure, representing it with $\star$ (note that the MSE_p is the mean of MSE_{q_i} for all i). We recall that, as the PMDA and NSDA, the LSDA makes no assumptions on the users' profiles, while the SDA assumes uniform behavior. Hence, as expected, when the profiles become increasingly skewed the SDA performs the worst, obtaining the LSDA the smallest MSE_p. Furthermore, it is worth noticing that the user behavior has a strong influence on the variance of the MSE_{q_i}. The fact that users have favorite friends who receive a large fraction of their messages makes the probability of these receivers easy to estimate, while for receivers that are not often chosen the estimates are poor. This explains the large variance in the SKW population with respect to the other population types.

6.3.6 Comparison between attack principles

Throughout the evaluation section we have considered four disclosure attacks that estimate users profiles using statistics and optimization techniques. We now compare these attacks to Vida, the Bayesian inference-based machine learning algorithm proposed by Danezis and Troncoso in [10]. Additionally, we also test the efficacy of simply setting the negative probabilities output by the unconstrained LSDA to zero (denoted as Z-LSDA), and of the constrained LSDA version, that we call C-LSDA.

We implement the alternating projection strategy described Sect. 4.3 for the constrained version. While this approach is much faster than solving the constrained problem in (2), due to memory limitations we had to reduce the number of rounds analyzed in order to obtain results at the time of submission.[4] In order to make our results comparable to that in previous sections we choose the following parameters $N_{\text{users}} = 20$, $t = 5$, $f = 5$, $\rho = 1000$, and $\tau_f = 1$. These parameters ensure that the average number of messages observed by the adversary from each sender to each sender's friend is the same as in the baseline example above.

We can see in Fig. 4 (right), which shows box plots representing the distribution of the MSE_{q_i} for all users under observation, that Vida outperforms the statistical variants. In order to simplify the figure, we have not plotted the the MSE_p, that lies extremely close to the median in all cases.

We have already discussed that the LSDA obtains an advantage over the SDA, PMDA, and NSDA by considering all observed rounds simultaneously, but does not account for the one-to-one relationship between send and received messages in the individual rounds of mixing. As we see in the figure, this advantage can be increased by not considering the negative probabilities (Z-LSDA). The best results, close to Vida's performance, are obtained when

[4] We plan to include results for $\rho = 10000$ in the final version of the paper.

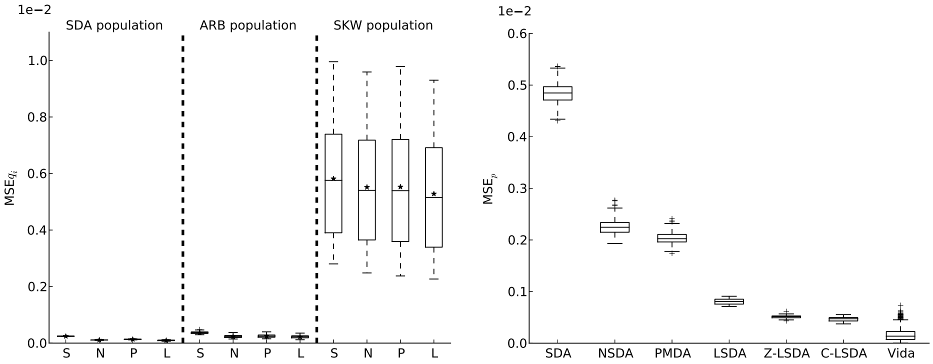

Figure 4: MSE$_{q_i}$ evolution with respect to the population type for all attacks ($N = 100$, $f = 25$, $t = 10$, $\rho = 10\,000$, $\tau_f = 1$) (left) and only comparison between attack principles (right) ($N = 20$, $f = 5$, $t = 5$, $\rho = 1000$, $\tau_f = 1$). (We represent MSE$_p$ with a $\star$.)

constraints ensuring the estimated profiles are well-defined are imposed on the solution (C-LSDA).

The approach followed in Vida, not only considers all rounds, but searches for perfect matchings in each round improving the profile estimation considerably with respect to the other attacks. While the effectiveness of Vida is desirable, it comes at a high computational cost because each iteration of the algorithm requires finding a perfect matching in all the ρ rounds observed. We have also noticed that when the population characteristics ease the profile estimation (few users with few friends) the performance of Vida is significantly affected. This is because in this case the set of possible matchings is reduced and finding them becomes increasingly difficult.

6.4 Results: Pool mix

We now proceed to evaluate the LSDA profiling performance when messages are anonymized using a threshold binomial pool mix. We recall that in such mix arriving messages are stored on a pool, and each round (i.e., when t messages are received) leave the mix with probability α. Otherwise, messages stay on the pool until the next round, when they are mixed with the arriving fresh messages and again probabilistically selected to be fired or not. Additionally, we compare the LSDA with the SDA, the only attack in the literature that has been applied to pool mixes.

6.4.1 Performance with respect to delay

Given the operation of the mix, the delay (in rounds) suffered by messages traversing the mix follows a geometric distribution with parameter α and hence its mean is $(1-\alpha)/\alpha$. In Fig. 5, left, we illustrate the tradeoff between

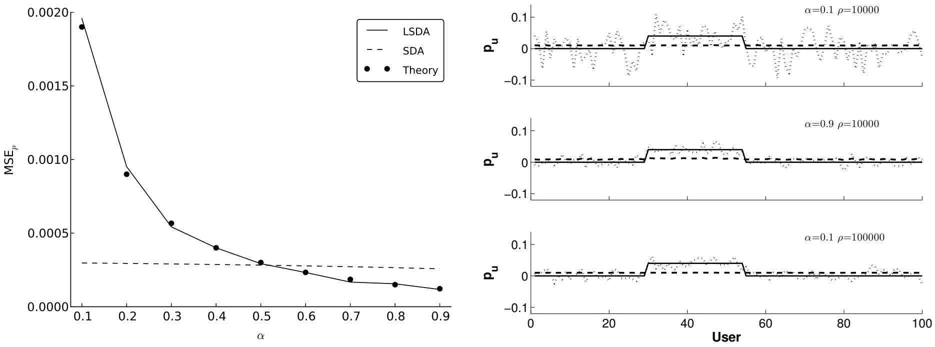

Figure 5: MSE$_p$ evolution with α ($N = 100$, $f = 25$, $t = 10$, $\rho = 10\,000$, τ_f=1) (left). Real (—), LSDA ($\cdots$), and SDA (- -) profiles for a given user depending on α and ρ (right).

the profiling accuracy of the LSDA and α. As expected, small values of α (i.e., large delays) result in larger error than when messages abandon the mix very fast. The longer the delay, the more messages participate in the mixing (recall that the mean size of the pool is $m = \frac{t-\alpha t}{\alpha}$), and the more difficult it is to estimate relations between senders and receivers. One can also see that the empirical error closely follows the prediction given by (34).

The figure also shows the evolution of the MSE per transition probability MSE$_p$ of the SDA. Perhaps surprisingly, it seems that the SDA outperforms the LSDA for small values of α, and that further its MSE$_p$ is independent from the pool mix firing probability. Fig. 5 (right) shows a comparison between the real profile of a given user (—), and the estimations output by the LSDA ($\cdots$) and SDA (- -). A closer look at the estimated profiles reveals that actually the SDA's output resembles noise with mean $1/N_{\text{users}}$. Only when α takes values closer to 1, i.e., when the pool mix operation is most similar to that of a threshold mix, the user's friends stand out in the profile output by the SDA. Thus, if we measure the percentage of friends correctly identified by both attacks, i.e., the percentage of real friends contained in the f users assigned the largest probabilities in each of the user profiles, we find that when least information is available ($\alpha = 0.1$,$\rho = 10\,000$), the LSDA correctly identifies 47% of the users' friends, while the SDA only uncovers 37%. This difference is reduced when α is increased to 0.9, and the LSDA and SDA correctly identify 93% and 91% of friends, respectively.

6.4.2 Performance with respect to the number of rounds ρ

Fig. 6, left shows the evolution of the MSE$_p$ with the number of rounds for two firing probabilities: $\alpha = 0.5$, and $\alpha = 0.9$. Similarly to the case of the threshold mix (see Fig. 1, left), and as predicted by (34), augmenting the number of observed rounds decreases significantly the error incurred by the LSDA. We can see that the empirical results closely follow our MSE_p prediction (represented by • in the figure).

As we have explained, the larger the firing probability α, the more similar to the threshold mix is the pool mix behavior, and hence the better is the estimation of the LSDA. The SDA, however, does not take much advantage of this fact, hence the difference between the MSE_p of both attacks increases with α. Moreover, observing more rounds does not significantly improve the SDA's performance, as illustrated by the first and second rows in Fig. 5 (right). The SDA's naive approach takes little advantage of the information procured by additional observations, and its output error remains virtually constant as the number of rounds grows. For $\alpha = 0.1$ the LSDA correctly identifies 47% of the users' friends when 10 000 rounds are observed, and 75% when ρ is increased to 100 000. The SDA's performance in identifying friends, however, only improves from 37% to 55%.

6.4.3 Comparison between attack principles

We show in Fig. 6 (right), box plots representing the distribution of the MSE_{q_i} for all users under observation for the SDA, the LSDA, the Z-LSDA (profiles obtained setting the negative probabilities output by the unconstrained LSDA to zero), and the C-LSDA (profiles obtained constraining the solution of the LSDA, Sect. 4.3). We use the parameters in Sect. 6.3.6 for the reasons outlined above.

The SDA results reflect the fact that the attack output is virtually independent from the input, and hence its error in profiling has almost no variance. We also observe that the Z-LSDA obtains much better results in terms of MSE_{q_i} than the bare bones LSDA. This is because the unconstrained LSDA's estimate includes negative probabilities that increase the mean squared error (cf. Fig. 5 (right)), while in practice gives perfect information to the adversary who infers that the corresponding users cannot be friends. Similarly to the threshold mix, the error is further reduced when constrains are imposed to ensure that the estimated profiles are well-defined.

7 Discussion and Future Work

We have shown that the Least Squares based approach is significantly more effective than its statistical predecessors in learning the users' profiles. Moreover, in the unconstrained case the matrix operations performed by the LSDA have much smaller computational requirements than the round-by-round processing carried out by the PMDA, NSDA or Vida. This decrease in computation comes at the cost of memory: the LSDA has to deal with large matrices. The parameters we have used in this paper generated matrices that fitted comfortably in a commodity computer, but larger mix networks

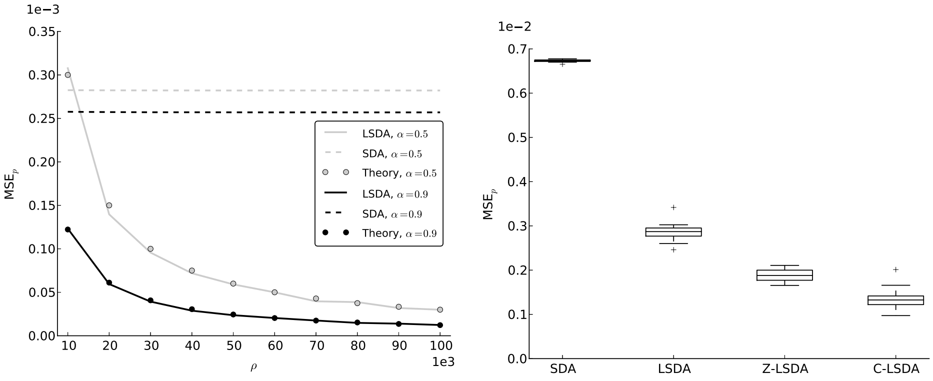

Figure 6: MSE$_p$ evolution with the number of pool mix rounds ρ ($N = 100$, $f = 25$, $t = 10$, $\alpha = 0.5$, τ_f=1), comparison between MSE$_p$ output by the different attack principles (right) ($N = 20$, $f = 5$, $t = 5$, $\rho = 1000$, $\tau_f = 1$, $\alpha = 0.5$).

may need extra memory. When memory is an issue, a gradient-based approach can be used to iteratively process the rounds obtaining the same result while considerably reducing the computational requirements of the attack.

The constrained LSDA considers all users simultaneously with a much higher computational load, which can be reduced again with iterative approaches as the *alternate optimization* solution that we have discussed. Moreover, our results show that the setting to zero the negative probabilities output by the unconstrained LSDA yields results close to the constrained solution at a minimal cost.

A common limitation of previously proposed disclosure attacks is that they are designed considering that user profiles are static. This assumption is rather unrealistic, as user's friendships are not guaranteed to be stable over time. The aforementioned iterative approach can not only be used to reduce the LSDA's computational requirements, but can be further adapted to account for temporal changes in the profiles. Extending the LSDA to accommodate such evolution is a promising line of future work.

In some cases it might be possible that some of the transition probabilities are known. It is possible to modify the machine learning approach [10] to account for this extra knowledge, but this is non-trivial for the SDA, PMDA or NSDA. In contrast, the Least Squares formulation can be easily adapted to consider this additional information, in a similar way as we did to show that the SDA is a particular, but largely suboptimal, instance of LSDA.

Finally, we have presented the attack against the threshold binomial pool mix assuming that the number of messages arriving to the mix is constant every round and that the firing probability is constant. However, the principles behind the attack make it easily adaptable to other mixing strategies where these conditions do not hold, for example, timed mixes (where the pool mix fires periodically regardless of the number of messages it has received) by adapting the construction of the matrices $\mathbf{U}_s^T$ and $\mathbf{B}$ to account for the specific system parameters and mixing algorithm. It must be noted that such adjustments would require adapting the derivations in the appendix; however, the methodology will still be valid. In fact, equipped with these tools, we plan to develop a framework to compare different mixing strategies and design new ones.

8 Conclusion

We have introduced the Least Squares Disclosure Attack, that estimates user profiles minimizing the prediction error of the output given the input. By modeling the estimation of profiles as a Least Squares problem, we are able to obtain analytic results that predict the profiling error for a given set of system parameters. This feature permits the designer of a high-latency anonymous communication system to choose parameters that provide a desired level of protection depending on the population characteristics without the need to perform simulations which may require a large computational effort as in the case of the matching-based approaches [10, 15].

Moreover, our attack is not limited to the analysis of threshold mixes but, contrary to most previous proposals, can be easily extended to more complex mixing strategies such as pool mixes. We have empirically evaluated the LSDA's performance in a wide variety of scenarios proving that it is superior to previous proposals, and that our formulas closely model its error.

9 Derivation of MSE for the pool mix.

First, we introduce the following three lemmas:

Lemma 1 *Let $\mathbf{A} = d\mathbf{I}_n + c\mathbf{1}_{n\times n}$, with d and c two real numbers. Then*

$$\mathbf{A}^{-1} = d^{-1}\left(\mathbf{I}_n - \frac{c}{nc+d}\mathbf{1}_{n\times n}\right)$$

Proof: See [20].

Lemma 2 *We will also need the following result. Let $\mathbf{M}$ be an arbitrary matrix of size $\rho \times \rho$. Then, for large ρ*

$$\mathbf{U}^T\mathbf{M}\mathbf{U} \approx tr(\mathbf{M})tp_x\mathbf{I}_{N_{users}} + \left(sum(\mathbf{M})t^2p_x^2 - tr(\mathbf{M})tp_x^2\right)\mathbf{1}_{N_{users}\times N_{users}} \quad (35)$$

where $tr(\mathbf{M})$ stands for the trace of $\mathbf{M}$ and $sum(\mathbf{M})$ is the summation of all the elements of $\mathbf{M}$.

Proof: The input sequences X_i^r, X_k^r, for all $i,k = 1,\cdots,N_{\text{users}}$ are ergodic in the first and second order joint moments, so the product $\mathbf{U}^T\mathbf{M}\mathbf{U}$ will converge to its expected value. Let $\mathbf{X}_j$, $\mathbf{m}_j$ denote the jth column of $\mathbf{U}$ and $\mathbf{M}$, respectively. Then, the (i,j)th element of $\mathbf{U}^T\mathbf{M}\mathbf{U}$ can be written as

$$E\{(\mathbf{U}^T\mathbf{M}\mathbf{U})_{i,j}\} = E\{\mathbf{X}_i^T\mathbf{M}\mathbf{X}_j\} = \sum_{r=1}^{\rho} E\{X_j^r\mathbf{X}_i^T\}\mathbf{m}_r \tag{36}$$

We must distinguish the cases $i \neq j$ and $i = j$. For the first, we use the facts that $\mathrm{E}\{X_i^r X_j^l\} = t^2p_x^2$ when $r \neq l$, and $\mathrm{E}\{X_i^r X_j^l\} = t^2p_x^2 - tp_x^2$ when $r = l$, to write

$$E\{(\mathbf{U}^T\mathbf{M}\mathbf{U})_{i,j}\} = t^2p_x^2\sum_{r=1}^{\rho}\mathbf{1}_\rho^T\mathbf{m}_r - tp_x^2\sum_{r=1}^{\rho}M_{r,r} = t^2p_x^2\mathrm{sum}(\mathbf{M}) - tp_x^2\mathrm{tr}(\mathbf{M}),\ \ i \neq j \tag{37}$$

For the case $i = j$ we use the fact that $\mathrm{E}\{(X_i^r)^2\} = t^2p_x^2 + tp_x(1-p_x)$, so now

$$E\{(\mathbf{U}^T\mathbf{M}\mathbf{U})_{i,i}\} = t^2p_x^2\sum_{r=1}^{\rho}\mathbf{1}_\rho^T\mathbf{m}_r + tp_x(1-p_x)\sum_{r=1}^{\rho}M_{r,r} = t^2p_x^2\mathrm{sum}(\mathbf{M}) + tp_x(1-p_x)\mathrm{tr}(\mathbf{M}), \tag{38}$$

Combining (37) and (38) we obtain (35). This completes the proof.

We can start now to derive an expression for the MSE. From (12) we can write the MSE as the sum of the traces of the covariance matrix for the estimated profile error corresponding to each user. Thus, we focus first on determining such covariance matrix, which for the jth user, and similarly to (11) becomes

$$\mathrm{E}[(\mathbf{p}_j - \hat{\mathbf{p}}_j)(\mathbf{p}_j - \hat{\mathbf{p}}_j)^T] = (\hat{\mathbf{R}}_{xs})^{-1}\hat{\mathbf{U}}_s^T\mathbf{\Sigma}_{y_j}\hat{\mathbf{U}}_s(\hat{\mathbf{R}}_{xs})^{-1} \tag{39}$$

where

$$\hat{\mathbf{R}}_{xs} \doteq \hat{\mathbf{U}}_s^T\hat{\mathbf{U}}_s = (\mathbf{U}^T + \mathbf{N}_0^T)\mathbf{B}^T\mathbf{B}(\mathbf{U} + \mathbf{N}_0) \tag{40}$$

is an estimate of the correlation matrix of $\hat{\mathbf{x}}_s^r$.

First, we compute the covariance matrix $\mathbf{\Sigma}_{y_j}$ of $\mathbf{Y}_j$, whose entries are $\mathrm{Cov}\{Y_j^r, Y_j^l\}$, for all $r,l = 1,\cdots,\rho$. Under the assumption that each sender and receiver have exactly f friends, we can see Y_j^r as the sum of r independent binomial processes with t trials and probabilities $p_x\alpha(1-\alpha)^k$, $k = 0,\cdots,r-1$. Then,

$$\begin{aligned}
\mathrm{Cov}\{Y_j^r, Y_j^l\} &= -tp_x^2\alpha^2\sum_{m=0}^{r-1}\sum_{k=0}^{l-1}(1-\alpha)^m(1-\alpha)^k,\ \ r \neq l \\
\mathrm{Var}\{Y_j^r\} &= tp_x\alpha\sum_{m=0}^{r-1}(1-\alpha)^m - tp_x^2\alpha^2\sum_{m=0}^{r-1}(1-\alpha)^{2m}
\end{aligned} \tag{41}$$

For large ρ, most of the main diagonal terms can be well approximated by making $r \to \infty$. In such case, we can write

$$\mathbf{\Sigma}_{y_j} \approx tp_x \mathbf{I}_\rho - tp_x^2 \mathbf{B}\mathbf{B}^T \tag{42}$$

We also need to write the estimated correlation matrix $\hat{\mathbf{R}}_{xs}$ in a way that does not depend on the particular input. We will assume that $\mathbf{N}_0 = \mathbf{0}$, as the impact of the initial conditions can be neglected for a large number of rounds. Therefore, $\hat{\mathbf{R}}_{xs} = \mathbf{U}^T\mathbf{B}^T\mathbf{B}\mathbf{U}$, so from Lemma 2 we need to obtain $\mathrm{tr}(\mathbf{B}^T\mathbf{B})$ and $\mathrm{sum}(\mathbf{B}^T\mathbf{B})$. To this end, and for large ρ, we can neglect border effects and approximate $\mathbf{B}$ by a doubly infinite lower triangular matrix whose columns contain the filter samples $b_k \doteq \alpha(1-\alpha)^k u_k$, where u_k is the unit-step function, and then truncating it to a $\rho \times \rho$ lower triangular matrix. Now, if $*$ denotes convolution, we can write

$$\mathrm{tr}(\mathbf{B}^T\mathbf{B}) \approx \rho(b_k * b_{-k})|_{k=0}; \quad \mathrm{sum}(\mathbf{B}^T\mathbf{B}) \approx \rho \sum_{k=-\infty}^{\infty} b_k * b_{-k} \tag{43}$$

From the definition, we find that

$$b_k * b_{-k} = \frac{\alpha}{2-\alpha}(1-\alpha)^{|k|} \tag{44}$$

from which it follows that $\mathrm{tr}(\mathbf{B}^T\mathbf{B}) = \rho\alpha/(2-\alpha) \doteq \rho\alpha_q$ and $\mathrm{sum}(\mathbf{B}^T\mathbf{B}) = \rho$. Then, we can write $\hat{\mathbf{R}}_{xs} = d_{xs}\mathbf{I}_{N_\mathrm{users}} + c_{xs}\mathbf{1}_{N_\mathrm{users}\times N_\mathrm{users}}$, where

$$c_{xs} \doteq \rho t p_x^2 (t - \alpha_q); \quad d_{xs} \doteq \rho\alpha_q t p_x \tag{45}$$

On the other hand, application of Lemma 1 and some algebra allows us to write

$$\begin{aligned}
\hat{\mathbf{R}}_{xs}^{-1} &= d_{xs}^{-1}\left(\mathbf{I}_{N_\mathrm{users}} - \theta\mathbf{1}_{N_\mathrm{users}\times N_\mathrm{users}}\right) && (46)\\
\hat{\mathbf{R}}_{xs}^{-2} &= d_{xs}^{-2}\left(\mathbf{I}_{N_\mathrm{users}} - (2\theta - N_\mathrm{users}\theta^2)\mathbf{1}_{N_\mathrm{users}\times N_\mathrm{users}}\right) && (47)
\end{aligned}$$

where $\theta \doteq p_x(1 - \alpha_q/t)$.

With the previous derivations, the trace of (39) can be expanded as follows

$$\mathrm{tr}(\hat{\mathbf{R}}_{xs}^{-1}\mathbf{U}^T\mathbf{B}^T\mathbf{\Sigma}_{y_j}\mathbf{B}\mathbf{U}\hat{\mathbf{R}}_{xs}^{-1}) = tp_x\mathrm{tr}(\hat{\mathbf{R}}_{xs}^{-1}) - tp_x^2\mathrm{tr}(\mathbf{U}^T(\mathbf{B}^T\mathbf{B})^2\mathbf{U}\hat{\mathbf{R}}_{xs}^{-2}) \tag{48}$$

We develop next the two summands in (48). From (46) we have

$$\mathrm{tr}(\hat{\mathbf{R}}_{xs}^{-1}) = N_\mathrm{users}d_{xs}^{-1} - N_\mathrm{users}\theta = \frac{N_\mathrm{users}}{\rho t\alpha_q}\left(N_\mathrm{users} - 1 + \alpha_q/t\right) \tag{49}$$

which coincides with (17) for $\alpha = 1$.

For the second summand in (48) we use (47) together with Lemma 2 and simplify terms to show

$$\begin{aligned}\mathbf{U}^T(\mathbf{B}^T\mathbf{B})^2\mathbf{U}\hat{\mathbf{R}}_{xs}^{-2} &= d_{xs}^{-2}\mathrm{tr}((\mathbf{B}^T\mathbf{B})^2)tp_x\mathbf{I}_{N_{\text{users}}} \\ &+ d_{xs}^{-2}\left(\mathrm{sum}((\mathbf{B}^T\mathbf{B})^2t^2p_x^2(1-N_{\text{users}}\theta)^2 - \mathrm{tr}((\mathbf{B}^T\mathbf{B})^2)tp_x^2\right)\mathbf{1}_{N_{\text{users}}\times N_{\text{users}}}\end{aligned} \quad (50)$$

Following the same reasoning as above, for large ρ we can write

$$\mathrm{tr}((\mathbf{B}^T\mathbf{B})^2) \approx \rho(b_k{*}b_{-k}{*}b_k{*}b_{-k})|_{k=0};\ \ \mathrm{sum}((\mathbf{B}^T\mathbf{B})^2) \approx \rho\sum_{k=-\infty}^{\infty} b_k{*}b_{-k}{*}b_k{*}b_{-k} \quad (51)$$

These two quantities can be evaluated using the Z-Transform and the Residue Theorem, to show that

$$\mathrm{tr}((\mathbf{B}^T\mathbf{B})^2) \approx \frac{\rho\alpha}{(2-\alpha)^3};\ \ \mathrm{sum}((\mathbf{B}^T\mathbf{B})^2) \approx \rho \quad (52)$$

In any case it is interesting to note that without any approximation both the trace and the sum of $(\mathbf{B}^T\mathbf{B})^2$ can be bounded from above by ρ and from below by 0.

Now, we can plug (52) in (50) to find

$$\mathrm{tr}(\mathbf{U}^T(\mathbf{B}^T\mathbf{B})^2\mathbf{U}\hat{\mathbf{R}}_{xs}^{-2}) \approx d_{xs}^{-2}N_{\text{users}}\rho\left(\frac{\alpha}{(2-\alpha)^3}tp_x(1-p_x) + \frac{\alpha_q^2}{t}t^2p_x^2\right) \quad (53)$$

Substituting (49) and (53) in (48) we find that

$$\mathrm{tr}(\hat{\mathbf{R}}_{xs}^{-1}\mathbf{U}^T\mathbf{B}^T\mathbf{\Sigma}_{y_j}\mathbf{B}\mathbf{U}\hat{\mathbf{R}}_{xs}^{-1}) \approx \frac{N_{\text{users}}-1+\alpha_q/t}{\rho\alpha_q} - \frac{(N_{\text{users}}-1)/(2-\alpha)+1/t}{\rho N_{\text{users}}} \quad (54)$$

where $\alpha_q = \alpha/(2-\alpha)$.

Finally, from (11)

$$\mathrm{MSE} = \sum_{j=1}^{N_{\text{users}}} \mathrm{tr}\left(\mathrm{E}[(\mathbf{p}_j-\hat{\mathbf{p}}_j)(\mathbf{p}_j-\hat{\mathbf{p}}_j)^T]\right) = N_{\text{users}}\mathrm{tr}(\hat{\mathbf{R}}_{xs}^{-1}\mathbf{U}^T\mathbf{B}^T\mathbf{\Sigma}_{y_j}\mathbf{B}\mathbf{U}\hat{\mathbf{R}}_{xs}^{-1}) \quad (55)$$

where the last equality is a consequence of (54) being independent of j.

References

[1] G. Danezis, C. Diaz, and P. Syverson, "Systems for anonymous communication," in *Handbook of Financial Cryptography and Security* (B. Rosenberg, ed.), Cryptography and Network Security Series, pp. 341–389, Chapman & Hall/CRC, 2009.

[2] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: Design of a Type III Anonymous Remailer Protocol," in *IEEE Symposium on Security and Privacy (S&P 2003)*, pp. 2–15, IEEE Computer Society, 2003.

[3] M. Edman and B. Yener, "On anonymity in an electronic society: A survey of anonymous communication systems," *ACM Computing Surveys*, vol. 42, no. 1, 2010.

[4] U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman, "Mixmaster Protocol — Version 2." IETF Internet Draft, July 2003.

[5] D. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[6] D. Agrawal and D. Kesdogan, "Measuring anonymity: The disclosure attack," *IEEE Security & Privacy*, vol. 1, no. 6, pp. 27–34, 2003.

[7] G. Danezis, "Statistical disclosure attacks: Traffic confirmation in open environments," in *Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003)* (Gritzalis, Vimercati, Samarati, and Katsikas, eds.), (Athens), pp. 421–426, IFIP TC11, Kluwer, May 2003.

[8] G. Danezis, C. Diaz, and C. Troncoso, "Two-sided statistical disclosure attack," in *7th International Symposium on Privacy Enhancing Technologies (PETS 2007)* (N. Borisov and P. Golle, eds.), vol. 4776 of *LNCS*, pp. 30–44, Springer-Verlag, 2007.

[9] G. Danezis and A. Serjantov, "Statistical disclosure or intersection attacks on anonymity systems," in *6th International Workshop on Information Hiding (IH 2004)* (J. J. Fridrich, ed.), vol. 3200 of *LNCS*, pp. 293–308, Springer, 2004.

[10] G. Danezis and C. Troncoso, "Vida: How to use Bayesian inference to de-anonymize persistent communications," in *9th Privacy Enhancing Technologies Symposium (PETS 2009)* (I. Goldberg and M. J. Atallah, eds.), vol. 5672 of *LNCS*, pp. 56–72, Springer, 2009.

[11] D. Kesdogan and L. Pimenidis, "The hitting set attack on anonymity protocols," in *6th International Workshop on Information Hiding (IH 2004)* (J. J. Fridrich, ed.), vol. 3200 of *LNCS*, pp. 326–339, Springer, 2004.

[12] N. Mathewson and R. Dingledine, "Practical traffic analysis: Extending and resisting statistical disclosure," in *4th International Workshop on Privacy Enhancing Technologies (PET 2004)* (D. Martin and A. Serjantov, eds.), vol. 3424 of *LNCS*, pp. 17–34, Springer, 2004.

[13] N. Mallesh and M. Wright, "The reverse statistical disclosure attack," in *Information Hiding - 12th International Conference (IH 2010)* (R. Böhme, P. W. L. Fong, and R. Safavi-Naini, eds.), vol. 6387 of *LNCS*, pp. 221–234, Springer, 2010.

[14] D. V. Pham, J. Wright, and D. Kesdogan, "A practical complexity-theoretic analysis of mix systems," in *16th European Symposium on Research in Computer Security (ESORICS 2011)* (V. Atluri and C. Diaz, eds.), vol. 6879 of *LNCS*, pp. 508–527, Springer, 2011.

[15] C. Troncoso, B. Gierlichs, B. Preneel, and I. Verbauwhede, "Perfect matching disclosure attacks," in *8th International Symposium on Privacy Enhancing Technologies (PETS 2008)* (N. Borisov and I. Goldberg, eds.), vol. 5134 of *LNCS*, pp. 2–23, Springer-Verlag, 2008.

[16] D. Kesdogan, D. Agrawal, and S. Penz, "Limits of anonymity in open environments," in *5th International Workshop on Information Hiding (IH 2002)* (F. A. P. Petitcolas, ed.), vol. 2578 of *LNCS*, pp. 53–69, 2002.

[17] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *2nd International Workshop on Privacy Enhancing Technologies (PET 2002)* (R. Dingledine and P. Syverson, eds.), vol. 2482 of *LNCS*, pp. 41–53, Springer, 2002.

[18] F. Pérez-González and C. Troncoso, "Understanding statistical disclosure: A least squares approach," *IEEE Transactions on Information Forensics and Security*, 2012. Under Submission.

[19] L. Scharf, *Statistical signal processing: detection, estimation, and time series analysis*. Addison-Wesley Publishing Company, 1991.

[20] K. Miller, "On the inverse of the sum of matrices," *Mathematics Magazine*, vol. 54, no. 2, pp. 67–72, 1981.