Understanding Statistical Disclosure: A Least Squares approach

Fernando Pérez-González[1][2][3] and Carmela Troncoso[4]

[1] Signal Theory and Communications Dept., University of Vigo
[2] Gradiant (Galician R&D Center in Advanced Telecommunications)
[3] Electrical and Computer Engineering Dept., University of New Mexico
fperez@gts.uvigo.es
[4] K.U. Leuven/IBBT, ESAT/SCD-COSIC
carmela.troncoso@esat.kuleuven.be

Abstract. It is widely accepted that Disclosure Attacks are effective against high-latency anonymous communication systems. A number of Disclosure Attack variants can be found in the literature that effectively de-anonymize traffic sent through a threshold mix. Nevertheless, these attacks' performance has been mostly evaluated through simulation and how their effectiveness varies with the parameters of the system is not well-understood. We present the LSDA, a novel disclosure attack based on the Maximum Likelihood (ML) approach, in which user profiles are estimated solving a Least Squares problem. Further, contrary to previous heuristic-based attacks, our approach allows to analytically derive formulae that characterize the profiling error of the LSDA with respect to the system's parameters. We verify through simulation that our predictors for the error closely model reality, and that the LSDA recovers users' profiles with greater accuracy than its predecessors.

1 Introduction

Mixes, relaying routers that hide the relation between incoming and outgoing messages [2], are one of the main building blocks for high-latency anonymous communications [4, 6, 9, 16]. A variety of Disclosure or Intersection Attacks [1, 3, 7, 8, 11, 12, 18, 19, 21] have been proposed to uncover persistent and repeated patterns of communication taking place through a mix. In a nutshell, these attacks find a target user's likely set of friends, also known as user profile, by intersecting the recipient anonymity sets of the messages this user sends.

Even though all attacks operate on the same principle they differ on how they exploit the observations in order to obtain user profiles. Statistical variants [3, 7, 21] rely on heuristics to operate, while the Bayesian inference-based method by Danezis and Troncoso [8] use Bayesian sampling techniques to recover accurately users' profiles in more complex systems than its predecessors. For any of these attacks it is difficult to obtain analytic results that characterize the dependence of their effectiveness on the parameters of the system, and hence they (as well as their sequels [5, 14, 15, 18]) have been mostly evaluated through simulation.

In this paper we propose a novel profiling attack based on the Maximum Likelihood (ML) approach. The attack estimates user profiles by solving a Least Squares problem, ensuring that the mean squared error between the real and estimated profiles is minimized. We empirically show that our attack indeed minimizes the mean squared error with respect to heuristic disclosure attack variants [3, 7, 21], although it performs slightly worse than the Bayesian approach [8] in the scenarios considered in this paper.

Nevertheless, we note that the most outstanding feature of the Least Squares approach is that, contrary to its predecessors, it allows us to derive analytical expressions that describe the evolution of the profiling error with the parameters of the system. This is a key property, as it allows designers to choose system parameters that provide a certain level of protection without the need to run simulations. We thoroughly validate our results through simulation, proving that our formulae reliably predict the evolution of our attack's error as the parameters of the system change.

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 in Sect. 4; where we derive equations that characterize its error with respect to the system parameters which we validate in Sect. 5. Finally, we discuss future lines of work in Sect. 6, and we conclude in Sect. 7.

2 Related work

We can find different flavors of disclosure attacks in the literature [1, 3, 7, 8, 10–13, 18, 19, 21] which we now proceed to revisit. A first family of intersection attacks are the so-called Disclosure Attack [1, 10] and its sequels [11–13, 18]. These attacks rely on Graph Theory in order to uncover the recipient set of a target user Alice. They seek to identify mutually disjoint sets of receivers amongst the recipient anonymity sets of the messages sent by Alice, which are intersected with the anonymity sets of Alice's sent messages to find her communication partners. The main drawback of the Disclosure attack 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, 12, 18] speeds up the search for Alice's messages recipients by looking for unique minimal hitting sets. An evaluation of this attack is provided in [18], where the relationship between the number of rounds the adversary needs to observe to uniquely identify the set of receivers is analyzed. The study by Pham et al. is similar to our work in spirit, but different in that they focus on attacks that unambiguously identifying recipient sets while our focus is on statistical attacks that only provide an estimation of such sets as the ones discussed below.

The series of statistical attacks was started by Danezis in [3] where he introduced the Statistical Disclosure Attack (SDA). Danezis observed that for a large enough number of observed mixing rounds the average of the probability distributions describing the recipient anonymity set [20] of Alice's messages

offers a very good estimation of her sending profile. Danezis considers that in each round where Alice sends a message, the recipient anonymity set of this message is uniform over the receivers present in the round (and zero for the rest of users). The SDA was subsequently extended to more complex mixing algorithms [7], to traffic containing replies [5], to consider other users in order to improve the identification of Alice's contacts [14], and to evaluate more complex user models [15].

Troncoso et al. proposed in [21] two attacks that outperform the SDA, the Perfect Matching Disclosure Attack (PMDA) and the Normalized Statistical Disclosure Attack (NSDA). Under the observation that in a round of mixing the relationships between sent and received messages must be one-to-one, the attacks consider interdependencies between senders and receivers in order to assign most likely receivers to each of the senders: the PMDA searches 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 each sender's message in a round is built taking into consideration the result of this assignment, instead of assigning uniform probability amongst all recipients.

Last, Danezis and Troncoso propose to approach the estimation of user profiles as a Bayesian inference problem [8]. They introduce the use of Bayesian sampling techniques to co-infer user communication profiles and de-anonymize messages. 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.

We note that previous authors evaluated the attacks either from mostly a de-anonymization of individual messages perspective (e.g., [8, 21]), or from the point of view of the number of rounds necessary to identify a percentage of Alice's recipients (e.g., [14, 15]). 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 an introduce the notation we use throughout the paper, summarized in Table 3.

System model. We consider a system in which a population of N_{users} users, designated by an index $i \in \{1, \ldots, N_{\text{users}}\}$, communicate through a threshold mix. This mix operates as follows. In each round of mixing it gathers t messages, transforms them cryptographically, and outputs them in a random order; hence hiding the correspondence between incoming and outgoing messages.

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 the column vectors 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 their recipients 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 $\sum_{i=1}^{N_{\text{users}}} p_{j,i}$.) 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. 5.

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. 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.

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 can be used to recover the users' sender and receiver profiles.

4 A Least Squares approach to Disclosure Attacks

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, which allow to simultaneously determine the sender and receiver profiles of all users.

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 amongst 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 station-

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 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, $\mathbf{p}_j = [p_{j,1}, p_{j,2}, \cdots, p_{j,N_{\text{users}}}]^T$
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 $\mathbf{p}_j$

ary (i.e., the parameters modeling their statistical behavior do not change with time).

4.1 Analysing one round of mixing

For simplicity of notation we will consider first a single round of observations, and later explain how to extend the derivation to an arbitrary number of rounds. Hence, for the moment, we will drop the superindex r. Let $Y_{j,i}$ be the random variable that models the number of messages received by user j that were sent by user i in the round under consideration. Then the number of messages that the jth user receives in this round can be computed as:

$$Y_j = \sum_{i=1}^{N_{\text{users}}} Y_{j,i} \, .$$

Recall that $p_{j,i}$ represents the probability that user i sends a message to user j. Then, the probability of user j receiving $y_{j,i}$ messages when the number of messages sent by user i is $X_i = x_i$ is given by a binomial distribution:

$$\Pr(Y_{j,i} = y_{j,i} | X_i = x_i) = \binom{x_i}{y_{j,i}} p_{j,i}^{y_{j,i}} (1 - p_{j,i})^{x_i - y_{j,i}} \, , \tag{1}$$

whose mean is $x_i \cdot p_{j,i}$ and variance $x_i \cdot p_{j,i}(1 - p_{j,i})$. This probability can be approximated by a Gaussian with the same mean and variance.

It is important to notice that the variables $Y_{j,i}$, $j = 1, \cdots, N_{\text{users}}$ are not independent, and rather they are jointly modeled by a multinomial distribution. However, the covariance $\text{cov}(Y_{j,i}, Y_{k,i}) = -x_i \cdot p_{j,i} \cdot p_{k,i}$, $k \neq j$, is small

(in comparison with diagonal terms of the covariance matrix) if the transition probabilities are also small. Moreover, in such case the variance of the binomial can be approximated by $x_i \cdot p_{j,i}$. Therefore, when the transition probabilities are small, and recalling that the sum of independent Gaussian random variables is itself Gaussian, we can approximate the conditional distribution of Y_j by a normal:

$$Pr(Y_j|\mathbf{X}=\mathbf{x}) \sim \mathcal{N}\left(\sum_{i=1}^{N_{\text{users}}} x_i p_{j,i}, \sum_{i=1}^{N_{\text{users}}} x_i p_{j,i}\right) ,$$

and consider that $\text{cov}(Y_j, Y_k) \approx 0$, whenever $k \neq j$.

Under the hypothesis above, since the random variables Y_j are approximately independent, we can write the joint probability of $\mathbf{Y}$ as

$$Pr(\mathbf{Y}|\mathbf{X}=\mathbf{x}) \sim \mathcal{N}\left(\mathbf{Hp}, \mathbf{\Sigma}_y\right) ,$$

where $\mathbf{p}^T \doteq [\mathbf{p}_1^T, \mathbf{p}_2^T, \cdots, \mathbf{p}_{N_{\text{users}}}^T]$, $\mathbf{\Sigma}_y \doteq \text{diag}(\mathbf{Hp})$, and $\mathbf{H}^T \doteq \mathbf{x} \otimes \mathbf{I}_{N_{\text{users}}}$. Here, $\mathbf{I}_n$ denotes the identity matrix of size $n \times n$, and $\otimes$ denotes the Kronecker product.

For a ML solution to the profiling problem, after observing $\mathbf{Y} = \mathbf{y}$, we seek that vector $\hat{\mathbf{p}}$ of probabilities that maximizes $Pr(\mathbf{Y}=\mathbf{y}|\mathbf{X}=\mathbf{x})$.[5] This can be explicitly written as follows:

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

where $\mathcal{P}$ denotes the set of valid probability vectors.[6]

For the unconstrained problem in (2) it is possible to uncouple the different terms and show that the solution must satisfy

$$\mathbf{x}^T \hat{\mathbf{p}}_j = \frac{1}{2}\left(\sqrt{1+4y_j^2} - 1\right), \quad j = 1, \cdots, N_{\text{users}} , \tag{3}$$

where $\hat{\mathbf{p}}_j$ is the estimated unnormalized receiver profile of user j.

The right hand side of (3) is smaller than y_j; however, it can be well approximated by y_j when the latter is large. Notice that (3) becomes an underdetermined linear system of equations.

4.2 Analysing ρ rounds

A different situation arises when the number of observed rounds is larger than the number of users. In this case, we form the following vectors/matrices:

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

[5] Notice that since the random variable $\mathbf{X}$ does not depend on the probabilities $\mathbf{p}$, the maximization of $Pr(\mathbf{Y}=\mathbf{y}|\mathbf{X}=\mathbf{x})$ is equivalent to that of $Pr(\mathbf{Y}=\mathbf{y}; \mathbf{X}=\mathbf{x})$.

[6] 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.

The ML solution must satisfy (2). However, notice that in the case of ρ rounds, the involved matrices and vectors are larger than those found in the case of a single observation.

Unlike (3), a closed-form solution seems not exist (even for the unconstrained case, i.e., when no constraints are imposed upon $\mathcal{P}$). We examine next some approximate solutions to the unconstrained problem that satisfy that $\hat{\mathbf{p}} \to \mathbf{p}$ as $\rho \to \infty$. To make this possible, we disregard the dependence of the covariance matrix $\mathbf{\Sigma}_y$ with $\mathbf{p}$ making the following approximation $\mathbf{\Sigma}_y \approx \mathrm{diag}(\mathbf{y})$.

In such case, the approximate ML estimator is given by

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}\in\mathcal{P}} ||\mathbf{\Sigma}_y^{-1/2}(\mathbf{y} - \mathbf{H}\mathbf{p})||^2 \,, \tag{4}$$

which is nothing but a constrained weighted least squares (WLS) problem.

For simplicity, we consider here the unweighted least squares (LS) case, i.e.,

$$\hat{\mathbf{p}} = \arg\min_{\mathbf{p}\in\mathcal{P}} ||\mathbf{y} - \mathbf{H}\mathbf{p}||^2 \,, \tag{5}$$

which, for the unconstrained case, has the well-known Moore-Penrose pseudoinverse solution:

$$\hat{\mathbf{p}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} \,. \tag{6}$$

At first sight, it might look that the matrix inversion needed in (6) is formidable: the matrix $\mathbf{H}^T\mathbf{H}$ has size $N_{\text{users}}^2 \times N_{\text{users}}^2$. However, its block-diagonal structure allows for a much more efficient solution; indeed,

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

and, hence,

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

where now $\mathbf{U}^T\mathbf{U}$ has size $N_{\text{users}} \times N_{\text{users}}$.

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

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

The decoupling above is possible only in the unconstrained case; this consideration, together with the simplicity of the performance analysis, make us focus 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 $\mathcal{P}$. In any case, the fact that the error $\mathbf{p} - \hat{\mathbf{p}}$ 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 $\hat{\mathbf{p}}$.

In any case, it is worth remarking that there are many iterative algorithms for solving (constrained) least squares problems, which do not require matrix

inversion. We leave the discussion on how they can be adapted to the problem as subject for future work. It is also worth stressing that we could have arrived at the LS estimate from the perspective of minimizing the mean square error between the observed $\mathbf{y}$ and a predictor based on a linear combination of the given inputs $\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^\rho$.

Finally, we note that the original Statistical Disclosure Attack (SDA) corresponds to a particular case of the proposed LS estimator. The SDA model assumes that the first user (Alice) sends only one message to an unknown recipient chosen uniformly from a set f friends. The other users send messages to recipients chosen uniformly from the set of all users $p_{j,i} = 1/N_{\text{users}}, \forall i \neq 1$. From this considerations, for a given round r where Alice does send a message, we have that $x_1^r = 1$ and $\sum_{j=2}^{N_{\text{users}}} x_i^r = (t-1)$, and all the transition probabilities of the form $p_{j,i}$, for $i \geq 2$ 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, $\mathbf{p}_1$ is unknown and all $\mathbf{p}_i$, $i = 2, \cdots, N_{\text{users}}$ are known. From here, it is possible to find that the LS estimate of the unknown probabilities is

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

which coincides with the SDA estimate. (We leave a more detailed derivation of this equation for an extended version of this paper [17].)

4.3 Performance analysis with respect to the system parameters

The Least Squares estimate in (6) 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

$$\mathrm{E}[(\mathbf{p} - \hat{\mathbf{p}})(\mathbf{p} - \hat{\mathbf{p}})^T] = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{\Sigma}_y\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\,. \tag{7}$$

Notice that the performance will depend on the actual input matrix $\mathbf{H}$; however, when the input process is wide-sense stationary, and $\rho \to \infty$ then $\mathbf{U}^T\mathbf{U}$ 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 $\mathbf{U}$. To this end, notice that if $\mathrm{Cov}(Y_i, Y_j) \approx 0$ for all $i \neq j$, then

$$\mathbf{\Sigma}_y \approx \mathrm{diag}(\xi_y) \otimes \mathbf{I}_{N_{\text{users}}}\,,$$

with $\xi_y = [\mathrm{Var}\{Y_1\}, \cdots, \mathrm{Var}\{Y_{N_{\text{users}}}\}]$.

In this case, (7) becomes

$$\mathrm{E}[(\mathbf{p} - \hat{\mathbf{p}})(\mathbf{p} - \hat{\mathbf{p}})^T] = \mathrm{diag}(\xi_y) \otimes (\mathbf{U}^T\mathbf{U})^{-1}\,. \tag{8}$$

Still we would 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

$$\mathbf{U}^T\mathbf{U} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^{-1}\,, \tag{9}$$

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

$$\mathrm{E}[(\mathbf{p}' - \hat{\mathbf{p}}')(\mathbf{p}' - \hat{\mathbf{p}}')^T] = \mathrm{diag}(\xi_y) \otimes \mathbf{\Lambda}^{-1} \tag{10}$$

A measure of the total error variance made with the proposed estimator is given by the trace. Notice that

$$\mathrm{E}\left[\mathrm{tr}\left((\mathbf{p}' - \hat{\mathbf{p}}')(\mathbf{p}' - \hat{\mathbf{p}}')^T\right)\right] = \mathrm{E}\left[\mathrm{tr}\left((\mathbf{p} - \hat{\mathbf{p}})(\mathbf{p} - \hat{\mathbf{p}})^T\right)\right] = \sum_{i=1}^{N_{\text{users}}} \sigma^2_{y_i} \cdot \sum_{j=1}^{N_{\text{users}}} \lambda^{-1}_{u,j} \tag{11}$$

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

Equation (11) 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

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

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,

$$\sum_{j=1}^{N_{\text{users}}} \lambda^{-1}_{u,j} = \frac{N_{\text{users}}}{\rho t}\left(N_{\text{users}} - 1 + \frac{1}{t}\right) \tag{12}$$

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

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

so the sum of variances becomes

$$\sum_{j=1}^{N_{\text{users}}} \sigma_{y_j}^2 = t \left(1 - \frac{\sum_{j=1}^{N_{\text{users}}} f_j^2}{f^2 N_{\text{users}}^2}\right) = t \left(1 - \frac{\tau_f}{N_{\text{users}}}\right) , \tag{13}$$

where we have used the fact that $\sum_{i=1}^{N_{\text{users}}} f_j = f \cdot N_{\text{users}}$.

Combining (12) and (13) we can write the MSE as

$$\mathrm{E}\left[\mathrm{tr}\left((\mathbf{p} - \hat{\mathbf{p}})(\mathbf{p} - \hat{\mathbf{p}})^T\right)\right] = \frac{1}{\rho}\left(N_{\text{users}} - 1 + \frac{1}{t}\right) \cdot (N_{\text{users}} - \tau_f) . \tag{14}$$

It is useful to interpret (14) 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 (14) 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 (14) as follows

$$\mathrm{E}\left[\mathrm{tr}\left((\mathbf{p} - \hat{\mathbf{p}})(\mathbf{p} - \hat{\mathbf{p}})^T\right)\right] \approx \frac{N_{\text{users}}^2}{\rho} . \tag{15}$$

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. 5.

5 Evaluation

5.1 Experimental setup

We evaluate the effectiveness of the Least Squares approach to Disclosure Attacks (LSDA) against synthetic anonymized traces created by a simulator written in

the Python language.[7] 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 $f_j = f$. In other words, every sender (receiver) profile has the same number of non-zero elements, and hence $\tau_f = 1$. Messages are anonymized using a threshold mix with threshold t, and we consider that the adversary observes ρ rounds of mixing. Table 5.1 summarizes the values of the parameters used in our experiments, where bold numbers indicate the parameters of the baseline experiment.

Table 2. System parameters used in the experiments.

Parameter	**Value**
N_{users}	$\{50, \mathbf{100}, 150, 200, 250, 300, 350, 400, 450, 500\}$
f	$\{5, 10, 15, 20, \mathbf{25}, 30, 35, 40, 45, 50\}$
t	$\{2, 5, \mathbf{10}, 20, 30, 40\}$
ρ	$\{\mathbf{10\,000}, 20\,000, \ldots, 100\,000\}$
τ_f	$\{\mathbf{1.0}, 1.76, 2.44, 3.04, 3.56, 4.0\}$

The parameters' 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 amongst them is preserved. Unfortunately, we cannot make a similar claim for the other attacks. Their heuristic nature makes it difficult to obtain analytical results that describe the dependence of their success on 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 to those obtained performing the Statistical Disclosure Attack (SDA) [3, 7], the Perfect Matching Disclosure Attack (PMDA) [21], the Normalized Statistical Disclosure Attack (NSDA) [21], and the Bayesian inference-based attack Vida [8].

5.2 Success metrics

We recall that the goal of the adversary is to estimate the values $p_{j,i}$ with as much accuracy as possible. The LSDA, as described in Sect. 4, is optimized to minimize the Mean Squared Error (MSE) between the actual transition probabilities $p_{j,i}$ and the adversary's estimated $\hat{p}_{j,i}$. We define two metrics to illustrate

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

the profiling accuracy of the attacks. The *Mean Squared Error per transition probability* (MSE_p) measures the average squared error between the elmements of the estimated matrix $\hat{\mathbf{p}}$ and the elements of the matrix $\mathbf{p}$ describing the actual behaviour of the users (see (6)):

$$\text{MSE}_p = \frac{\sum_{i,j}(\hat{p}_{j,i} - p_{j,i})^2}{N_{\text{users}}^2}.$$

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

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

which measures the average squared error between the probability of 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.

5.3 Results

Estimating sender and receiver profiles with the LSDA. We first illustrate how the LSDA can simultaneously estimate sender and receiver profiles. Traditionally, Disclosure Attacks focus in estimating the sender profiles; and receiver profiles can be inferred by resolving the inverse problem (i.e., performing the same attack inverting the role of senders and receivers). Even the Reverse Statistical Disclosure Attack [14], that explicitly requires receiver profiles to improve the estimation of the sender profiles, includes a step in which the SDA is applied in the reverse direction before results can be obtained.

The LSDA estimates the full matrix $\mathbf{p}$ in one go. By either considering the rows or columns of this matrix the adversary can recover the unnormalized receiver profile $\mathbf{p}_j = [p_{j,1}, p_{j,2}, \cdots, p_{j,N_{\text{users}}}]^T$, or the sender profile $\mathbf{q}_i = [p_{1,i}, p_{2,i}, \cdots, p_{N_{\text{users}},i}]^T$ without any additional operation. Fig. 1, left, shows box plots[8] describing the distribution of the MSE_p over 20 experiments for senders and receiver profiles, respectively. The right-hand side of the figure shows the sender and the receiver profiles computed performing the LSDA in the reverse direction, considering the receivers as senders, and vice versa.

[8] The line in the middle of the box represents the median of the distribution. The lower and upper limits of the box correspond, respectively, to the distribution's first (Q1) and third quartiles (Q3). We also show the outliers, represented with +: values x which are "far" from the rest of the distribution ($x > Q3 + 1.5(Q3 - Q1)$ or $x < Q1 - 1 : 5(Q3 - Q1)$).

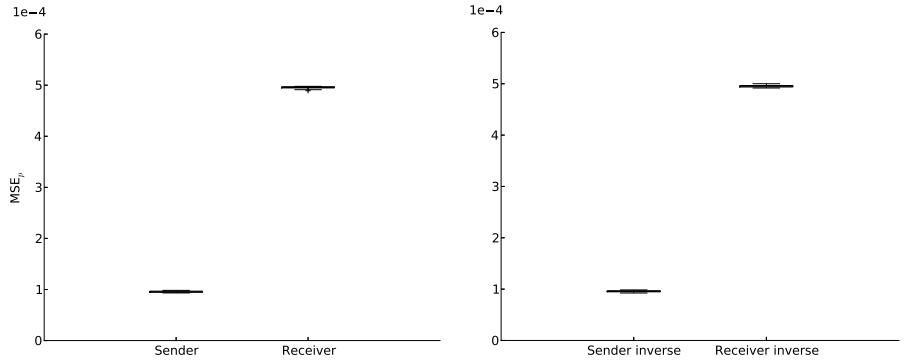

Fig. 1. LSDA's MSE_p per transition probability when inferring sender and receiver profiles in the forward (left) and reverse (right) directions ($N_{\text{users}} = 100$, $f = 25$, $t = 10$, $\rho = 10\,000$, $\tau_f = 1$).

We can see that the results obtained in the "forward" and reverse direction are not the same. In fact, we have observed that in each instance there is a direction that is better than the other in terms of MSE_p. While it is not possible to decide which side is going to provide better results, because a priori all profiles are equally likely, it is easy to see that the average of the estimations $\hat{\mathbf{p}}$ in both directions will yield a MSE per transition probability smaller than the worst case.

Performance with respect to the number of rounds ρ. As we discuss in Sect. 4.3, the number of observed rounds ρ has a dominant role in the estimation error incurred by the LSDA. We plot in Fig. 2, left, the MSE per transition probability MSE_p for the SDA, NSDA, PMDA and LSDA.

The LSDA, optimized to minimize the MSE_p, obtains the best results. Further, we can see how the approximation in Eq. (15), represented by $\bullet$ 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 [21]. 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 obtain 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 $\mathbf{U}$).

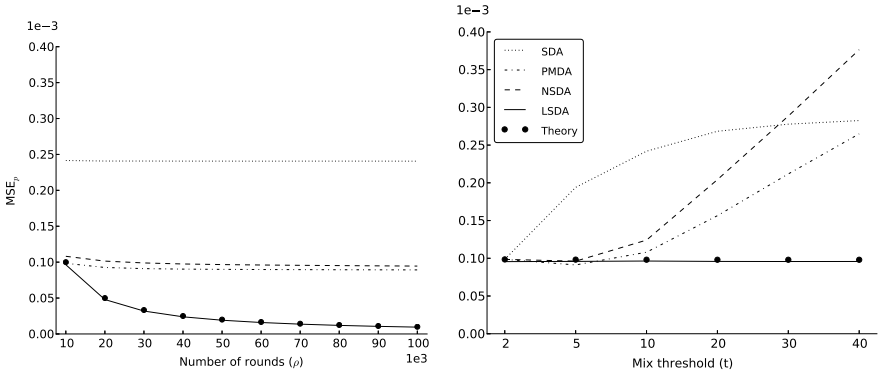

Fig. 2. MSE_p evolution with the number of rounds in the system ρ ($N = 100$, $f = 25$, $t = 10$, τ_f=1), and with the threshold mix t ($N = 100$, $f = 25$, $\rho = 10\,000$, $\tau_f = 1$) (left and right, respectively).

Performance with respect to the mix threshold t. By observing Eq. (14) 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 $t >> 1$. This is reflected by our experiments, shown in Fig. 2, left, where the error of the LSDA soon becomes stable as the threshold of the mix grows.

This desirable property does not hold for the other approaches. As expected, increasing the threshold has a negative effect on the three attacks. Nevertheless this effect differs depending on the approach used. The SDA's, surprisingly, seems to grow proportionally to $(1 - 1/t)$ and thus the increase of the error with the threshold is greatly reduced as t increases. This is not the case for the NSDA and PMDA, based on solving an optimization problem on the underlying bipartite graph representing a mix round. This problem becomes harder as the threshold grows, thus their MSE_p significantly increases with the number of messages processed in each mix round.

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. 3 for $\rho = 10\,000$ (left) and $\rho = 100\,000$ (right). As expected (see 15), 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. When the number of users increases, and the mix threshold does not vary, 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. 3 reinforce the idea that the number of rounds ρ is the main component

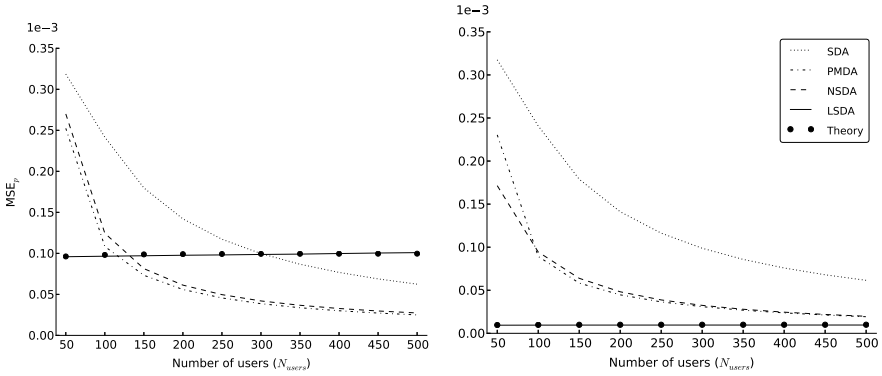

Fig. 3. MSE_p evolution with the number of users in the system N_{users} ($f = 25$, $t = 10$, $\rho = 10\,000, 100\,000$, $\tau_f = 1$) (left and right, respectively).

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}.

Performance with respect to the output variance $\boldsymbol{\sigma^2_{y_j}}$. 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 Eq. (13)). We first vary the number of friends of the senders f while keeping $f_j = f$ for all receivers j, ensuring that $\tau_f = 1$. We observe in Fig. 4, left, that the LSDA's MSE_p closely follows the prediction in formula (14).

In a second experiment, we fix the parameter f vary τ_f to represent different degrees of "hubness" in the population. We construct populations such in which 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. 4, right. It is worthy to note that the SDA significantly benefits from the hubness of the population. As some users concentrate the traffic, and 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 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.

Performance with respect to the user behaviour. Our experiments so far considered a very simplistic population in which users choose amongst their friends uniformly at random (which we denote as SDA). As it has been discussed

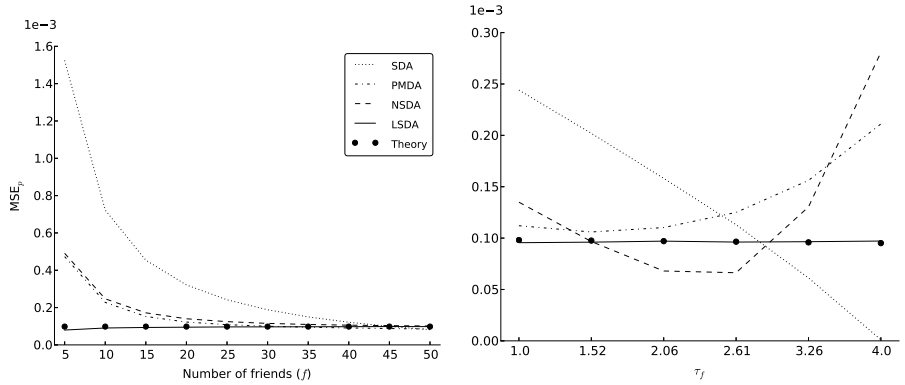

Fig. 4. MSE_p evolution with the number of friends f ($N = 100$, $f = 25$, $\rho = 10\,000$, τ_f=1), and with τ_f ($N = 100$, $f = 25$, $t = 10$, $\rho = 10\,000$) (left and right, respectively).

in the past [8, 21] 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. 5 (left) box plots representing the distribution of the MSE per sender profile MSE_{q_i} for all users in the population. We also plot the MSE_p for each attack in the figure, representing it with $\star$ (note that the MSE_p is also 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 worthy to notice that the user behaviour 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 attacks' estimations are poor. This explains the large variance in the SKW population with respect to the other population types.

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 [8]. We can see in Fig. 5 (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,

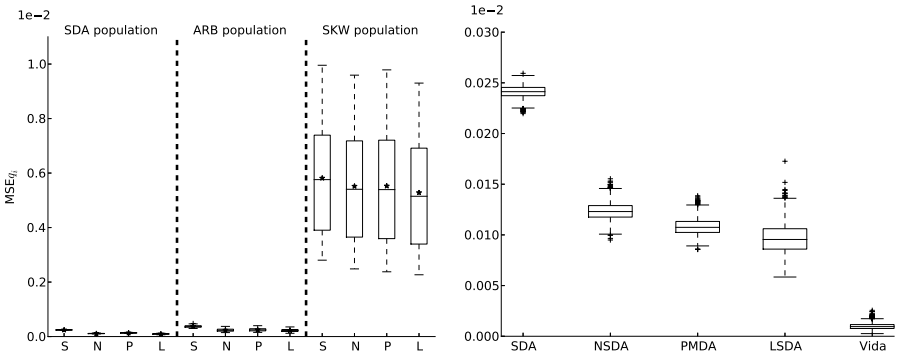

Fig. 5. MSE_{q_i} evolution with respect to the population type for all attacks (left) and only comparison between attack principles (right) ($N = 100$, $f = 25$, $t = 10$, $\rho = 10\,000$, $\tau_f = 1$). (We represent MSE_p with a $\star$.)

but does not account for the one-to-one relationship between send and received messages in the individual rounds of mixing. Vida, on the other hand, not only considers all rounds, but searches for perfect matchings in each round improving the profile estimation considerably. These results seemingly contradict the performance evaluation in [8]. This is because the comparison performed by Danezis and Troncoso was with respect to the message de-anonymization success rate, while we focus on the estimation of profiles. In fact, the results reported by Danezis and Troncoso show that when 512 rounds of mixing are observed the profiling accuracy of the algorithm is excellent.

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 note however that, as in [8], we have used the SDA's result as seed for the machine learning engine. Intrestingly, using the LSDA, which yields better estimation of the real profiles than the the SDA, instead may significantly speed up the learning time.

6 Discussion

We have shown that the LSDA is more effective than its statistical predecessors. Further, the matrix operations performed by the LSDA have much smaller computational requirements than the round-by-round processing carried out by the PMDA or the NSDA. This decrease in computation comes at the cost of memory: the LSDA operates with big matrices that have to be loaded to the RAM. The parameters we have used in this paper generated matrices that fitted comfortably in a commodity computer, but larger mix networks 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 reducing the computational requirements of the attack, that would deal with smaller ma-

trices. This iterative approach 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.

The fact that we have considered user profiling as an unconstrained problem (see Eq. (2)) resulted in some of the probabilities $\hat{p}_{j,i}$ estimated by the LSDA being negative, corresponding to receivers j that are not friends of user i. 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 $\hat{p}_{j,i} \geq 0$. One could reduce the error by just setting those probabilities to zero, disregarding that $\sum_j p_{j,i} = 1$ for all i. Alternatively, it is possible to establish constraints on Eq. (2) to ensure that the profiles recovered by the LSDA are well-defined. However, enforcing such constraints will no longer guarantee the decoupling of the unnormalized receiver profiles, and hence the solution is likely to be quite cumbersome. The development and analysis of such solution is left as subject for future research.

Threshold mixes are well fitted to analyse in theory, however deployed systems use pool mixes, which offer better anonymity. Up to know only the SDA has been adapted to traffic analysis of anonymous communications carried out through a pool mix [16]. This is because the internal mechanism of this mix, that may delay messages for more than one round, hinders the construction of a bipartite graph between senders and receivers. Hence, adapting the PMDA, the NSDA, or Vida to such scenario is non-trivial. The independence of the LSDA from the mix threshold makes it an ideal candidate for the analysis of pool mixes. In order to adapt the attack to this mix it is necessary to estimate the matrices $\mathrm{E}\{\mathbf{U}^T\mathbf{U}\}$ and $\mathrm{E}\{\mathbf{U}^T\mathbf{y}_j\}$, for all j.

Finally, in some cases it might be possible that some of the transition probabilities are known. It is possible to modify the machine learning approach [8] to account for this extra knowledge, but this is non-trivial for the SDA, PMDA or NSDA. The Least Squares formulation can be easily adapted to consider this additional information. Without loss of generality let us assume that $p_{1,1}$ is known. As this corresponds to the first element of $\mathbf{p}$, one can work instead with an equivalent problem in which we remove the first column of $\mathbf{H}$ and $p_{1,1}$ from $\mathbf{p}$; consequently, the observation vector $\mathbf{y}$ is replaced by $\mathbf{y} - p_{1,1}\mathbf{h}_1$. This procedure can be repeated for every known transition probability. Similar considerations can be made for the case where the transition probabilities $p_{j,i}$ depend on a smaller set of parameters (e.g., when some of the probabilities are known to be identical).

7 Conclusion

Since Kesdogan and Agrawal [1, 12] introduced the Disclosure Attack to profile users sending messages through an anonymous network, a stream of efficient statistical variants have been proposed [3, 7, 5, 8, 14, 15, 21]. Nevertheless, their heuristic nature hinders the search for analytical formulae describing the dependence of their success on the system parameters, which is difficult to characterize and predict as we have shown in our results.

We have introduced the LSDA, a new approach to Disclosure based on solving a Least Square problem, that minimizes the mean squared error between the estimated and real profiles. Further, the LSDA is the first disclosure attack able to simultaneously estimate sender and receiver profiles. The main advantage of our approach is that it allows the analyst to predict the profiling error given the system parameters. This capability is essential at the time of designing high-latency anonymous communication systems, as it permits the designer to choose the system 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 Vida. We have empirically evaluated the LSDA and we have proved that our formulae closely model its error.

Acknowledgements. Research supported by the European Regional Development Fund (ERDF); by the Galician Regional Government under projects Consolidation of Research Units 2010/85 and SCALLOPS (10PXIB322231PR); by the Spanish Government under project COMONSENS (CONSOLIDER-INGENIO 2010 CSD2008-00010); by the Iberdrola Foundation through the Prince of Asturias Endowed Chair in Information Science and Related Technologies; by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government; and by the IAP Programme P6/26 BCRYPT. C. Troncoso is a research assistant of the Flemish Fund for Scientific Research (FWO). The authors thank G. Danezis and C. Diaz for their comments on earlier versions of the manuscript.

References

1. D. Agrawal and D. Kesdogan. Measuring anonymity: The disclosure attack. *IEEE Security & Privacy*, 1(6):27–34, 2003.
2. D. Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM*, 24(2):84–90, 1981.
3. G. Danezis. Statistical disclosure attacks: Traffic confirmation in open environments. In Gritzalis, Vimercati, Samarati, and Katsikas, editors, *Proceedings of Security and Privacy in the Age of Uncertainty, (SEC2003)*, pages 421–426, Athens, May 2003. IFIP TC11, Kluwer.
4. G. Danezis, C. Diaz, and P. Syverson. Systems for anonymous communication. In B. Rosenberg, editor, *Handbook of Financial Cryptography and Security*, Cryptography and Network Security Series, pages 341–389. Chapman & Hall/CRC, 2009.
5. G. Danezis, C. Diaz, and C. Troncoso. Two-sided statistical disclosure attack. In N. Borisov and P. Golle, editors, *7th International Symposium on Privacy Enhancing Technologies (PETS 2007)*, volume 4776 of *LNCS*, pages 30–44. Springer-Verlag, 2007.
6. 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)*, pages 2–15. IEEE Computer Society, 2003.
7. G. Danezis and A. Serjantov. Statistical disclosure or intersection attacks on anonymity systems. In J. J. Fridrich, editor, *6th International Workshop on Information Hiding (IH 2004)*, volume 3200 of *LNCS*, pages 293–308. Springer, 2004.
8. G. Danezis and C. Troncoso. Vida: How to use Bayesian inference to de-anonymize persistent communications. In I. Goldberg and M. J. Atallah, editors, *9th Privacy*

Enhancing Technologies Symposium (PETS 2009), volume 5672 of *LNCS*, pages 56–72. Springer, 2009.

9. M. Edman and B. Yener. On anonymity in an electronic society: A survey of anonymous communication systems. *ACM Computing Surveys*, 42(1), 2010.
10. D. Kesdogan, D. Agrawal, and S. Penz. Limits of anonymity in open environments. In F. A. P. Petitcolas, editor, *5th International Workshop on Information Hiding (IH 2002)*, volume 2578 of *LNCS*, pages 53–69, 2002.
11. D. Kesdogan, D. Mölle, S. Richter, and P. Rossmanith. Breaking anonymity by learning a unique minimum hitting set. In A. Frid, A. Morozov, A. Rybalchenko, and K. Wagner, editors, *4th International Computer Science Symposium in Russia (CSR 2009)*, volume 5675 of *LNCS*, pages 299–309. Springer, 2009.
12. D. Kesdogan and L. Pimenidis. The hitting set attack on anonymity protocols. In J. J. Fridrich, editor, *6th International Workshop on Information Hiding (IH 2004)*, volume 3200 of *LNCS*, pages 326–339. Springer, 2004.
13. J. Liu, H. Xu, and C. Xie. A new statistical hitting set attack on anonymity protocols. In *Computational Intelligence and Security, International Conference (CIS 07)*, pages 922–925. IEEE Computer Society, 2007.
14. N. Mallesh and M. Wright. The reverse statistical disclosure attack. In R. Böhme, P. W. L. Fong, and R. Safavi-Naini, editors, *Information Hiding - 12th International Conference (IH 2010)*, volume 6387 of *LNCS*, pages 221–234. Springer, 2010.
15. N. Mathewson and R. Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In D. Martin and A. Serjantov, editors, *4th International Workshop on Privacy Enhancing Technologies (PET 2004)*, volume 3424 of *LNCS*, pages 17–34. Springer, 2004.
16. U. Möller, L. Cottrell, P. Palfrader, and L. Sassaman. Mixmaster Protocol — Version 2. IETF Internet Draft, July 2003.
17. F. Pérez-González and C. Troncoso. Understanding statistical disclosure: A least squares approach. *IEEE Transactions on Information Forensics and Security*, 2012. Under Submission.
18. D. V. Pham, J. Wright, and D. Kesdogan. A practical complexity-theoretic analysis of mix systems. In V. Atluri and C. Diaz, editors, *16th European Symposium on Research in Computer Security (ESORICS 2011)*, volume 6879 of *LNCS*, pages 508–527. Springer, 2011.
19. J.-F. Raymond. Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *LNCS*, pages 10–29. Springer-Verlag, July 2000.
20. A. Serjantov and G. Danezis. Towards an information theoretic metric for anonymity. In R. Dingledine and P. Syverson, editors, *2nd International Workshop on Privacy Enhancing Technologies (PET 2002)*, volume 2482 of *LNCS*, pages 41–53. Springer, 2002.
21. C. Troncoso, B. Gierlichs, B. Preneel, and I. Verbauwhede. Perfect matching disclosure attacks. In N. Borisov and I. Goldberg, editors, *8th International Symposium on Privacy Enhancing Technologies (PETS 2008)*, volume 5134 of *LNCS*, pages 2–23. Springer-Verlag, 2008.