A Least Squares Approach to the Static Traffic Analysis of High-Latency Anonymous Communication Systems

Fernando Pérez-González*, *Senior Member, IEEE,* Carmela Troncoso and Simon Oya

***Abstract*—Mixes, relaying routers that hide the relation between incoming and outgoing messages, are the main building block of high-latency anonymous communication networks. A number of so-called disclosure attacks have been proposed to effectively de-anonymize traffic sent through these channels. Yet, the dependence of their success on the system parameters is not well-understood. We propose the Least Squares Disclosure Attack (LSDA), in which user profiles are estimated by solving a least squares problem. We show that 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 us to analytically derive expressions that characterize the profiling error of LSDA with respect to the system parameters. We empirically demonstrate that LSDA recovers users' profiles with greater accuracy than its statistical predecessors and verify that our analysis closely predicts actual performance.**

***Index Terms*—anonymity, mixes, disclosure attacks**

I. Introduction

COMMUNICATION confidentiality is traditionally 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-known 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 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 short, these attacks infer Alice's likely set of contacts, also known as her user profile, by observing the sets of possible receivers for each message Alice sends and processing this information.

F. Pérez-González and S. Oya are with the Signal Theory and Communications Dept., University of Vigo. F. Pérez-González and C. Troncoso are with Gradiant (Galician R&D Center in Advanced Telecommunications).

This work was supported in part by the Spanish Government and the European Regional Development Fund under project TACTICA, in part by the Galician Regional Government under project Consolidation of Research Units GRC2013/009, and by the EU 7th Framework Programme (FP7/2007-2013) under grant agreements 610613 (PRIPARE) and 285901 (LIFTGATE).

The 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 the analysis of how system parameters influence their success. Furthermore, the great majority of attacks have only been evaluated against simple threshold mixes, where 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 messages each user receives from the mix and a prediction based on the messages sent to the mix is minimized. We show that LSDA is an efficient estimator when users' behavior is static, i.e., the error incurred when estimating the user profiles asymptotically tends to zero as the number of observed mix rounds grows when user behavior does not change within the observation window, and that it is suitable to attack anonymous communication 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 [16].

We provide two variants of LSDA: a very efficient unconstrained profile estimator that outputs user profiles that may contain negative probabilities (usually corresponding to receivers that are not contacts of the target user) and a slower constrained version that further minimizes the error by ensuring that the output profiles are well-defined. We show through simulations that the latter indeed minimizes the mean squared error with respect to heuristic disclosure attack variants [7], [9], [15] although it performs slightly worse than the Bayesian approach [10] in the simple threshold mix scenario. In the pool mix scenario, however, applying the Bayesian inference techniques is computationally unfeasible and LSDA emerges as the best option to infer the user's behavior.

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 system parameters. This is a key property, as it permits designers to choose system

parameters that provide a certain level of protection without needing to run simulations. We empirically validate our results, showing that our formulas reliably predict the evolution of LSDA's error as the parameters of the system change when users' behavior is static (i.e., time-invariant), and show the usefulness of our methodology beyond this assumptions using real data.

We note that previous works 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 profile of Alice, i.e., we not only seek to identify Alice's messages receivers, but also to estimate the probability that Alice sends a message to them.

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. III. We introduce the least squares approach to disclosure applied to threshold mixes in Sect. IV, and extend it to account for the pool mix in Sect. V. In both sections we derive equations that characterize LSDA's error with respect to the system parameters which we validate in Sect. VI. We discuss the limitations of our model and explain how to extend our analysis to more realistic scenarios in Sect. VII, and we conclude in Sect. VIII.

II. An Overview of Disclosure Attacks

The first Disclosure Attack [6], [17] relies on graph theory to uncover the recipient set of a target user Alice. It identifies the set of Alice's contacts by seeking 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), originally proposed by Danezis [9], and its sequels [8], [12], [13], estimate Alice's sending profile by averaging the probability distributions describing the recipient anonymity set [18] of her messages. Mathewson and Dingledine improved Danezis' SDA by extending it to a more general scenario and to more complex mixing algorithms [12]. This improved version of the attack is able to isolate Alice's behavior by first estimating the behavior of all the remaining users, employing those observations where Alice has not participated.

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. PMDA accounts for this interdependency by searching for perfect matchings in the underlying bipartite graph representing a mix round, while 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 SDA does.

Last, Danezis and Troncoso propose to use Bayesian sampling techniques to co-infer users' profiles and de-anonymize messages [10]. The Bayesian approach outputs samples from the distribution of all possible sending profiles, which in turn allows to infer reliable error estimates. However, Vida requires the adversary to repeatedly seek for perfect matchings, increasing the computational requirements of the attack.

From all of the aforementioned attacks, only SDA has been extended to take into account pool mixes. The fact that a message can be delayed multiple rounds before being forwarded to its recipient largely increases the set of possible receivers of each message. This makes extending the Disclosure and Hitting Set attacks [6], [11] a non trivial task and finding correspondences between incoming and outgoing messages, such as in PMDA, NSDA [15] and Vida [10], computationally unfeasible. We will show that our least squares approach can be adapted to the pool mix probabilistic behavior without increasing significantly the computational resources needed.

III. System and Adversary 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 I. Capital letters denote random variables and lowercase letters denote realizations. Vectors are represented by boldface characters; thus, $\mathbf{x} = [x_1, \cdots, x_N]^T$ is a realization of random vector $\mathbf{X} = [X_1, \cdots, X_N]^T$, where T denotes the transposing operation. Matrices are represented by boldface capital characters; whether they contain random or specific values will be clear from the context. We 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 $N \times M$. Furthermore, $< . >$ represents the scalar product operation and $\otimes$ the Kronecker product.

1) System model: We study a system in which a population of N users, designated by an index $i \in \{1, \cdots, N\}$, exchange messages through a 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**: This pool mix collects t messages per round and alters their appearance to avoid bitwise linkability. However, instead of outputting them immediately, messages are placed in a pool and only leave the mix with probability α. Otherwise, they stay and get mixed with messages arriving in subsequent rounds.

We model the number of messages that user i sends in round r as the random variable X_i^r, and denote as x_i^r the actual

TABLE I: Summary of notation

Symbol	Meaning
N	Number of users in the population, denoted by $i \in \{1, \cdots, N\}$
t	Threshold of the threshold/pool mix
α	Firing probability of the binomial pool mix
f_i	Probability that a message arriving to the mix comes from user i
$p_{j,i}$	Probability that user i sends a message to user j
$\mathbf{q}_i$	Sender profile of user i, $\mathbf{q}_i \doteq [p_{1,i}, p_{2,i}, \cdots p_{N,i}]^T$
$\mathbf{p}_j$	Unnormalized receiver profile of user j, $\mathbf{p}_j \doteq [p_{j,1}, p_{j,2}, \cdots p_{j,N}]^T$
$\mathbf{p}$	Vector of transition probabilities, $\mathbf{p} \doteq [\mathbf{p}_1^T, \mathbf{p}_2^T, \cdots, \mathbf{p}_N^T]^T$
μ_i	$1 - \sum_{j=1}^{N} p_{j,i}^2$
$\bar{\mu}$	$\sum_{i=1}^{N} f_i \mu_i$
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), $i, j = 1, \cdots, N$
$\mathbf{y}_j$	Column vector containing elements y_j^r, $r = 1, \cdots, \rho$
$\mathbf{U}$	$\rho \times N$ matrix containing all the input observations $\mathbf{U} \doteq [\mathbf{x}^1, \cdots, \mathbf{x}^\rho]^T$
$\mathbf{H}$	$\mathbf{I}_N \otimes \mathbf{U}$
$\mathbf{y}$	$\rho N \times 1$ column vector containing all the output observations $\mathbf{y} \doteq [\mathbf{y}_1^T, \cdots, \mathbf{y}_N^T]^T$
$\hat{p}_{j,i}$	Adversary's estimation of $p_{j,i}$
$\hat{\mathbf{q}}_i$	Adversary's estimation of user i's sender profile $\mathbf{q}_i$
$\hat{\mathbf{p}}_j$	Adversary's estimation of user j's unnormalized receiver profile $\mathbf{p}_j$
$\hat{\mathbf{p}}$	Adversary's estimation of transition probabilities vector $\mathbf{p}$

number of messages user i sends in that round. Similarly, Y_j^r is the random variable that models the number of messages that user j receives in round r, and y_j^r the actual number of messages user j receives in that round. Let $\mathbf{x}^r$ and $\mathbf{y}^r$ denote column vectors that contain as elements the number of messages sent or received by all users in round r, i.e., $\mathbf{x}^r \doteq [x_1^r, \cdots, x_N^r]^T$, and $\mathbf{y}^r \doteq [y_1^r, \cdots, y_N^r]^T$, respectively. When it is clear from the context, the superscript r is dropped. We also group the messages received by user j up to round ρ in vector $\mathbf{y}_j \doteq [y_j^1, \cdots, y_j^\rho]^T$. Let $\mathbf{U}$ denote the $\rho \times N$ matrix containing all the input observations up to round ρ, $\mathbf{U} \doteq [\mathbf{x}^1, \mathbf{x}^2, \cdots, \mathbf{x}^\rho]^T$ and let $\mathbf{y}$ denote the $\rho N \times 1$ vector containing all the output observations up to round ρ, $\mathbf{y} \doteq [y_1^1, \cdots, y_1^\rho, y_2^1, \cdots, y_2^\rho, \cdots, y_N^1, \cdots, y_N^\rho]^T$. Lastly, we define the matrix $\mathbf{H} \doteq \mathbf{I}_N \otimes \mathbf{U}$ which we shall use when deriving the LSDA estimator in Sect. IV.

Users in our population send messages to their recipients according to two parameters:

- **Sender profile**: the sender profile of a user represents her communication preferences, i.e., what fraction of her messages is sent to each receiver. We denote the sender profile of user i by vector $\mathbf{q}_i \doteq [p_{1,i}, p_{2,i}, \cdots, p_{N,i}]^T$, where $p_{j,i}$ models the probability that user i sends a message to user j. We define $\mathbf{p}_j$ as the column vector containing the probabilities of those incoming messages to the jth user, i.e., $\mathbf{p}_j \doteq [p_{j,1}, p_{j,2}, \cdots, p_{j,N}]^T$. Let $\mathbf{p}$ be the vector containing all the transition probabilities, i.e., $\mathbf{p} \doteq [p_{1,1}, \cdots, p_{1,N}, p_{2,1}, \cdots, p_{2,N}, \cdots, p_{N,1}, \cdots, p_{N,N}]^T$. With the previous definitions, this vector can be written as $\mathbf{p} = [\mathbf{p}_1^T, \mathbf{p}_2^T, \cdots, \mathbf{p}_N^T]^T$. We make no assumptions on the shape of users' profiles (i.e., we impose no restrictions on the number of contacts a user may have, nor on how messages are distributed among them), other than $\mathbf{q}_i \in \mathcal{P}$, where $\mathcal{P}$ is the probability simplex in $\mathbb{R}^N$, i.e., $\mathcal{P} \doteq \left\{ \mathbf{r} \in \mathbb{R}^N : r_i \geq 0, \sum_{i=1}^{N} r_i = 1 \right\}$.
- **Sending frequency**: the sending frequencies model how often users participate in the system. We denote the sending frequency of user i by f_i, where f_i is the probability that a message arriving to the mix comes from user i. We make no assumptions on the values of the sending frequencies other than $0 \leq f_i \leq 1$ for $i = 1, 2, \cdots, N$ and $\sum_{i=1}^{N} f_i = 1$.

Finally, we define $\mu_i \doteq 1 - \sum_{j=1}^{N} p_{j,i}^2$ and $\bar{\mu} \doteq \sum_{i=1}^{N} f_i \mu_i$. The former, μ_i, represents the uniformity of the distribution of user i's sender profile. It ranges from 0, when user i always sends messages to the same user (i.e., $p_{k,i} = 1$ for a certain user $k \in \{1, 2, \cdots, N\}$, and $p_{j,i} = 0$ otherwise), to $\frac{N-1}{N}$, when user i sends messages to all the other users equiprobably (i.e., $p_{j,i} = \frac{1}{N}$ for $j = 1, 2, \cdots, N$). The parameter $\bar{\mu}$ represents the average uniformity of all users' sender profiles. These parameters shall come in handy in the performance evaluation in Sect. VI.

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/or α). As our objective is to illustrate the impact of disclosure attacks on anonymity, we assume that the cryptographic transformation performed by the mix 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 observations x_i^r and y_j^r, for $i, j = 1, \cdots, N$, and $r = 1, \cdots, \rho$, 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.

A. Working hypotheses

For the derivation of the LSDA estimator in the next section, we assume that users' choice of recipients is independent for

each input message, i.e., the recipient of each message sent by a user i is chosen randomly according to its sender profile $\mathbf{q}_i$. We also assume that the sender profiles are static, i.e., they do not change within the same round or between different rounds.

For the analysis in Sect. IV-A2, we further assume that the probability that a given message arriving to the mix comes from a certain user is independent for each incoming message, i.e., the number of messages sent by the users in each round can be modeled as a multinomial distribution whose parameters are the sending frequencies. Also, we consider that the sending frequencies are static, i.e., they do not change within the same rounds or between different rounds.

The implications of these assumptions not holding are discussed in Sect. VII.

IV. A Least Squares approach to Disclosure Attacks on Threshold Mixes

We aim here at deriving a profiling algorithm to recover the sending behavior of users anonymously communicating through a threshold mix based on a least squares approach, under the assumptions explained in Sect. III-A. Even though the mix output random variables are conditioned on the input matrix $\mathbf{U}$ (or, equivalently, $\mathbf{H}$), for the sake of notational simplicity, in this section we will not write such conditioning explicitly.

Our goal is to estimate the users' profiles given the input and output observations, x_i^r and y_j^r for $i, j = 1, \cdots, N$, and all rounds $r = 1, \cdots, \rho$. To derive our estimator, we first note that given the vector of probabilities $\mathbf{p}$ and the input samples in $\mathbf{U}$, the output process Y_j^r for $j = 1, \cdots, N$ can be modeled in each round $r \in \{1, \cdots, \rho\}$ as the sum of N multinomials:

$$\{Y_1^r, \cdots, Y_N^r\} \sim \sum_{i=1}^{N} \text{Multi}\left(x_i^r, \{p_{1,i}, \cdots, p_{N,i}\}\right) . \quad (1)$$

Recall that these output random variables Y_j^r, $j = 1, \cdots, N$, $r = 1, \cdots, \rho$ are collected in random vector $\mathbf{Y}$, of which we observe one realization $\mathbf{y}$. Given this realization $\mathbf{y}$ and the input observations $\mathbf{U}$, we want to estimate the probability vector $\mathbf{p}$. In order to do so, we look for the vector $\mathbf{p}$ that minimizes the Mean Squared Error (MSE) between $\mathbf{y}$ and $\mathbf{Y}$. Then, our estimator can be formulated as the following constrained least squares problem,

$$\hat{\mathbf{p}} = \underset{\mathbf{q}_i \in \mathcal{P},\, i=1,\cdots,N}{\arg\min} \; \mathrm{E}\left\{||\mathbf{y} - \mathbf{Y}(\mathbf{p})||^2\right\} \quad (2)$$

where we have written $\mathbf{Y}(\mathbf{p})$ to stress the fact that the output distribution actually depends on the probability vector $\mathbf{p}$ (cf. (1)). Notice that the constraints are enforced on each sender profile $\mathbf{q}_i$ to ensure that they are well-defined. Since there exists a one-to-one correspondence between each element in $\mathbf{q}_i$ for $i = 1, \cdots, N$ and each element in $\mathbf{p}$, imposing the restrictions over each $\mathbf{q}_i$ is equivalent to doing so in $\mathbf{p}$.

The estimator in (2) minimizes, on average, the squared error over the possible outputs of the system, but does not necessarily minimize the error in the estimation of $\mathbf{p}$. The estimator in (2) is actually biased, but expanding the formulation we can find an unbiased and asymptotically efficient estimator of $\mathbf{p}$. First, let $\mathbf{W}(\mathbf{p}) \doteq \mathbf{Y}(\mathbf{p}) - \mathrm{E}\{\mathbf{Y}(\mathbf{p})\}$, which is a vector containing zero-mean random variables and whose variance is equal to that of $\mathbf{Y}(\mathbf{p})$. Using the definition in (1), we can write $\mathrm{E}\{\mathbf{Y}_j(\mathbf{p}_j)\} = \mathbf{U} \cdot \mathbf{p}_j$, and therefore, $\mathrm{E}\{\mathbf{Y}(\mathbf{p})\} = (\mathbf{I}_N \otimes \mathbf{U})\,\mathbf{p} = \mathbf{H} \cdot \mathbf{p}$. Then,

$$\begin{aligned} \mathrm{E}\left\{||\mathbf{y} - \mathbf{Y}(\mathbf{p})||^2\right\} &= \mathrm{E}\left\{||\mathbf{y} - \mathbf{H}\mathbf{p} - \mathbf{W}(\mathbf{p})||^2\right\} \\ &= ||\mathbf{y} - \mathbf{H}\mathbf{p}||^2 + \mathrm{E}\left\{||\mathbf{W}(\mathbf{p})||^2\right\} \quad (3) \end{aligned}$$

where we have used that $\mathrm{E}\{< (\mathbf{y} - \mathbf{H}\mathbf{p}), \mathbf{W}(\mathbf{p}) >\} = \mathbf{0}$ since $\mathrm{E}\{\mathbf{W}(\mathbf{p})\} = \mathbf{0}$.

Removing the term $\mathrm{E}\left\{||\mathbf{W}(\mathbf{p})||^2\right\}$ from (3) leads to an estimator which is asymptotically *efficient* when users' behavior is static, in the sense that $\hat{\mathbf{p}}$ converges to the true profiles as $\rho \to \infty$. In that case, (2) becomes

$$\hat{\mathbf{p}} = \underset{\mathbf{q}_i \in \mathcal{P},\, i=1,\cdots,N}{\arg\min} \; ||\mathbf{y} - \mathbf{H}\mathbf{p}||^2 . \quad (4)$$

A. Unconstrained Least Squares Estimation

We first propose an unconstrained estimator which is still asymptotically efficient and amenable to an in-depth performance analysis, as we will show in Sect. IV-A2. It is well-known that, for the unconstrained case, the solution of (4) is provided by the Moore-Penrose pseudoinverse [19]:

$$\hat{\mathbf{p}} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} . \quad (5)$$

At first sight, it might look that the matrix inversion needed in (5) is formidable: the matrix $\mathbf{H}^T\mathbf{H}$ has size $N^2 \times N^2$. However, its block-diagonal structure allows for a more affordable solution:

$$\mathbf{H}^T\mathbf{H} = (\mathbf{I}_N \otimes \mathbf{U})^T \cdot \mathbf{I}_N \otimes \mathbf{U} = \mathbf{I}_N \otimes (\mathbf{U}^T\mathbf{U})$$

and, hence,

$$(\mathbf{H}^T\mathbf{H})^{-1} = \mathbf{I}_N \otimes (\mathbf{U}^T\mathbf{U})^{-1}$$

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

The decoupling above allows us to write a more efficient solution. The least squares estimate $\hat{\mathbf{p}}_j$ for the jth probability vector can be written as

$$\hat{\mathbf{p}}_j = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{y}_j, \quad j = 1, \cdots, N . \quad (6)$$

Notice that, as a consequence of removing the constraints, the obtained sender profiles are not guaranteed to lie in the probability simplex $\mathcal{P}$. In fact, some of the estimated probabilities $\hat{p}_{j,i}$ will often be negative, usually corresponding to receivers j that are not contacts of user i.

Note that the matrix operations performed by LSDA have much smaller computational requirements than the round-by-round processing carried out by previous attacks [15], [10]. This decrease in computation comes at the cost of memory: LSDA has to deal with large matrices. When memory is an issue, the Recursive Least Squares (RLS) algorithm [20], that computes the least squares solution by processing the observed rounds recursively, can reduce the requirements of the attack considerably.

1) 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 n_f contacts. 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 contacts of Alice.

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

$$||\mathbf{y}_j - \mathbf{U}\mathbf{p}_j||^2 \tag{7}$$

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

$$||\mathbf{y}_j - \mathbf{U}'\mathbf{p}'_j - p_{j,1}\mathbf{U}_1||^2 \tag{8}$$

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

$$\hat{p}_{j,1} = (\mathbf{U}_1^T\mathbf{U}_1)^{-1}\mathbf{U}_1^T(\mathbf{y}_j - \mathbf{U}'\mathbf{p}'_j)\,. \tag{9}$$

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 $1/N$ and the matrix $\mathbf{U}'$ is such that the sum of the elements in each row is $(t-1)$; therefore,

$$\hat{p}_{j,1} = \frac{1}{\rho}\sum_{r=1}^{\rho} y_j^r - \frac{(t-1)}{N}, \quad j = 1, \cdots, N \tag{10}$$

which coincides with Danezis' SDA estimate [7].

The LSDA estimator differs from the original SDA estimator 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.

2) Performance analysis with respect to the system parameters: Next, we assess the performance of our unconstrained solution in (6) for the working hypothesis explained in Sect. III-A. This will serve to understand the influence of the system parameters on the knowledge that an adversary can gain by applying our algorithm. To the best of our knowledge, this is the first in-depth analysis of how the system parameters affect the performance of an attack on mixes. We aim at deriving a theoretical expression for the MSE in the estimation of the sender profile of user i, which we define as

$$\text{MSE}_i \doteq \text{E}\{||\mathbf{q}_i - \hat{\mathbf{q}}_i||^2\} = \sum_{j=1}^{N} \text{E}\left\{(p_{j,i} - \hat{p}_{j,i})^2\right\}\,. \tag{11}$$

We carry out this derivation in Appendix A, obtaining

$$\text{MSE}_i \approx \frac{1}{\rho}\left\{\left(f_i^{-1} - 1\right)\left(1 - \frac{1}{t}\right)\bar{\mu} + \frac{f_i^{-1}}{t}\cdot\mu_i\right\}\,. \tag{12}$$

It is useful to interpret (12) in terms of the users' sending frequency f_i and the profile uniformity μ_i. First, the error of LSDA increases with f_i^{-1}, since it becomes harder to estimate the behavior of a user when that user rarely participates in the system. The MSE_i also increases with μ_i and $\bar{\mu}$. This also makes sense, since given a certain amount of observations it is harder to infer the sending profiles when the behavior of the users in the system is highly random (large μ_i or $\bar{\mu}$) than when the users are more predictable (low μ_i or $\bar{\mu}$). The MSE also decreases as $1/\rho$ with the number of rounds ρ; this implies that the unconstrained LS estimator is asymptotically efficient as $\rho \to \infty$. Even though this is somewhat to be expected, notice that other estimators might not share this desirable property, as we will experimentally confirm in Sect. VI.

Note that our expression for the theoretical MSE of LSDA in (12) is an approximation due to the simplification (30) made during its derivation in Appendix A. Therefore, there will be a discrepancy between our MSE formula and the real MSE of the LSDA estimator which increases with the number of users in the system N and decreases with ρ.

To better understand the performance of the estimator, we now derive a rough approximation of (12). Normally, when the number of users is large, we can expect the sender frequencies of the users to be low, i.e., $f_i^{-1} \gg 1$. Also, if we assume that users have many contacts without any specific preference to any of them, then $\sum_{j=1}^{N} p_{j,i}^2 \ll 1$ or, equivalently, $\mu_i \approx 1$. In this case, (12) can be approximated as

$$\text{MSE}_i \approx \frac{f_i^{-1}}{\rho}\,. \tag{13}$$

Although this is a rough approximation, it shows the dominant parameters that affect the performance of the attack. Furthermore, it shall be useful when comparing the performance of LSDA for threshold mixes with that of pool mixes.

B. Constrained Least Squares Estimation

We now derive an estimator for the constrained problem in (4). Note that one could reduce the error of the unconstrained estimator (6) by just setting the negative probabilities to zero. The $\sum_j p_{j,i} = 1$ constraint could be later ensured by normalizing the profile, but this normalization has to be performed without information and hence the estimation is not guaranteed to be optimal.

We recall that the constraints are $0 \leq p_{j,i} \leq 1$, for all $i, j = 1, \cdots, N$, and $\sum_{j=1}^{N} p_{j,i} = 1$, for all $i = 1, \cdots, N$. One might think of imposing such constraints to the decoupled optimization problems (7) 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.

An alternative solution consists in solving the problem in an iterative fashion. If we define $\hat{\mathbf{P}} = [\hat{\mathbf{q}}_1, \hat{\mathbf{q}}_2, \cdots, \hat{\mathbf{q}}_N]$ and

$\mathbf{V} \doteq [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N]$, then the *unconstrained* solution in (6) can be written in a more compact form as

$$\hat{\mathbf{P}}^T = (\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{V}\,. \tag{14}$$

Using a gradient descent algorithm, it is possible to solve (14) by iteratively updating matrix $\hat{\mathbf{P}}$ as

$$\hat{\mathbf{P}}_{new} = \hat{\mathbf{P}}_{old} - \tau \cdot \mathbf{U}^T\left(\mathbf{U}\cdot\hat{\mathbf{P}}_{old} - \mathbf{V}\right)\,. \tag{15}$$

Here, the stepsize τ is chosen such that $0 < \tau < 2/\lambda_{max}$, where λ_{max} is the largest eigenvalue of matrix $\mathbf{U}^T\mathbf{U}$. The *constrained* solution can be obtained by projecting each column of $\hat{\mathbf{P}}$ onto the probability simplex $\mathcal{P}$ after each iteration. This solution, which we will denote C-LSDA, outperforms the unconstrained one as we will show in Sect. VI.

As a final remark, the constraints make a performance analysis similar to that in Section IV-A2 much more cumbersome. The analysis of such solution is left as subject for future research, but we note that the MSE for the unconstrained version approximated in (12) constitutes an upper bound on the MSE of the constrained variant.

V. A Least Squares approach to Disclosure Attacks on Pool Mixes

In this section, we show how to extend the Least Squares Disclosure Attack and the analysis of its performance to the threshold binomial pool mix, which was described in Sect. III, working under our hypotheses in Sect. III-A. We note, however, that the principles behind the attack make it easily adaptable to other mixing strategies (e.g., timed mixes), as shown in [16]. The main difference of the pool mix with respect to the threshold mix arises from the fact that some messages stay in the pool so it is no longer possible for the adversary to know how many messages from user i leave the mix in round r.

In order to make the derivation of the estimator easier, we abstract the threshold binomial pool mix as a combination of a pool block followed by a mixing block, as depicted in Fig. 1. The pool block stores messages until it has received t of them, and then outputs each with probability α, leaving the remaining for subsequent rounds. The messages that leave the pool block traverse the mix block, which changes their appearance and forwards them to their receivers. Note that the pool mix always receives t messages each round, while the number of messages that leave, denoted by t_s in the figure, is variable. To distinguish between the number of messages from user i that enter and leave the pool block in round r, we will respectively use x_i^r and $x_{s,i}^r$, as shown in Fig. 1. The adversary is only able to observe vectors $\mathbf{x}^r$ and $\mathbf{y}^r$, while the number of messages from each sender that leave the mix in each round, modeled by the random vector $\mathbf{X}_s^r$, is unknown to the attacker. We let $\mathbf{U}_s^T \doteq [\mathbf{X}_s^1, \mathbf{X}_s^2, \cdots, \mathbf{X}_s^\rho]$.

We assume that at the time the adversary starts her observation the pool contains m messages whose sender is unknown. Then, the messages in $X_{s,i}^r$ may come from two sources: the initial m messages in the pool and the messages sent by user i in the current or earlier rounds. We will use N_i^r to model the number of messages from user i that were initially in the pool

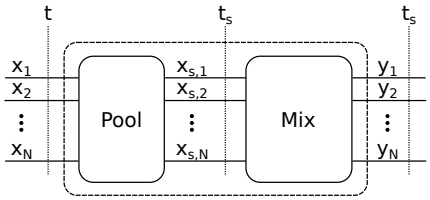


Fig. 1: Abstract model of the pool mix, represented as a combination of two blocks: the pool block, which delays the messages, and the mix block, which operates as a standard threshold mix.

and leave the mix in round r, and $X_{s,i}^{r,k}$ to model the number of messages sent by user i in round k that leave the mix in round r. Thus, the total number $X_{s,i}^r$ of messages from user i that leave the mix in round r can be written as

$$X_{s,i}^r = \sum_{k=1}^{r} X_{s,i}^{r,k} + N_i^r \tag{16}$$

where each of the contributions $X_{s,i}^{r,k}$ for $k = 1, \cdots, r$ and N_i^r are independent. The random variables $\{X_{s,i}^{k,k}, X_{s,i}^{k+1,k}, \cdots, X_{s,i}^{k+l,k}, \cdots\}$ can be modeled together as a multinomial distribution with x_i^k trials and probabilities $\{\alpha, \alpha(1-\alpha), \cdots, \alpha(1-\alpha)^l, \cdots\}$.

We derive the LSDA estimator for the pool mix problem as we did in Sect. IV for the threshold mix. Again, for notational simplicity, we will not write the conditioning of $\mathbf{U}_s$ and $\mathbf{Y}$ on $\mathbf{U}$. Given the vector of probabilities $\mathbf{p}$ and the messages that leave the pool in each round $\mathbf{U}_s$, we can model the output process as

$$\{Y_1^r, \cdots, Y_N^r\} \sim \sum_{i=1}^{N} \text{Multi}\left(x_{s,i}^r, \{p_{1,i}, \cdots, p_{N,i}\}\right)\,. \tag{17}$$

In this case, however, the values in $x_{s,i}^r$ are not observable. To derive our estimator following the approach in Sect. IV, we need to compute the expected value of $\mathbf{Y}(\mathbf{p})$. In this case,

$$\mathrm{E}\{\mathbf{Y}_j(\mathbf{p}_j)\} = \mathrm{E}\{\mathrm{E}\{\mathbf{Y}_j(\mathbf{p}_j)|\mathbf{U}_s\}\} = \mathrm{E}\{\mathbf{U}_s\}\cdot\mathbf{p}_j = \hat{\mathbf{U}}_s\cdot\mathbf{p}_j \tag{18}$$

where each element of $\hat{\mathbf{U}}_s \doteq \mathrm{E}\{\mathbf{U}_s\}$ can be computed, using (16), as

$$\begin{aligned}\hat{x}_{s,i}^r &= \mathrm{E}\{X_{s,i}^r\} = \sum_{k=1}^{r}\mathrm{E}\{X_{s,i}^{r,k}\} + \mathrm{E}\{N_i^r\} \\ &= \sum_{k=1}^{r} x_i^k\alpha(1-\alpha)^{r-k} + m\hat{f}_i\alpha(1-\alpha)^{r-1}\,. \end{aligned}\tag{19}$$

Here, $\hat{f}_i$ represents the adversary's estimation of the probability that each of the m messages in the initial pool corresponds to user i. An attacker can estimate those values by observing the system for a while. Anyhow, the influence of the initial m messages in the estimation of the profiles diminishes quickly as the number of rounds observed increases. For implementation purposes, a more convenient way of writing (19) is the following recursive equation

$$\hat{x}_{s,i}^{r+1} = (1-\alpha)\hat{x}_{s,i}^r + \alpha x_i^{r+1}, \quad r = 1, \cdots, N \tag{20}$$

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

For compactness, we will find it useful to define the following *convolution matrix*

$$\mathbf{B} \doteq \begin{bmatrix} \alpha & 0 & 0 & \cdots & 0 \\ \alpha(1-\alpha) & \alpha & 0 & \cdots & 0 \\ \alpha(1-\alpha)^2 & \alpha(1-\alpha) & \alpha & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \alpha(1-\alpha)^{\rho-1} & \alpha(1-\alpha)^{\rho-2} & \alpha(1-\alpha)^{\rho-3} & \cdots & \alpha \end{bmatrix} \tag{21}$$

Then, we can write

$$\hat{\mathbf{U}}_s = \mathbf{B} \cdot (\mathbf{U} + \mathbf{N}_0) \tag{22}$$

where the matrix $\mathbf{N}_0$, which accounts for the average initial state of the mix, is such that the i-th entry in the first row takes the value $m\hat{f}_i$, while all the remaining elements are zero.

Our LSDA estimator in the pool mix, following the derivations given in the threshold mix scenario (4), can be formulated as the following constrained problem:

$$\hat{\mathbf{p}} = \underset{\mathbf{q}_i \in \mathcal{P},\, i=1,\cdots,N}{\arg\min} ||\mathbf{y} - \hat{\mathbf{H}}_s \mathbf{p}||^2 \tag{23}$$

where $\hat{\mathbf{H}}_s \doteq \mathbf{I}_N \otimes \hat{\mathbf{U}}_s$.

A. Unconstrained Least Squares Estimation on Pool Mixes

The solution for the unconstrained case is given by

$$\hat{\mathbf{p}}_j = (\hat{\mathbf{U}}_s^T \hat{\mathbf{U}}_s)^{-1} \hat{\mathbf{U}}_s^T \mathbf{y}_j, \quad j = 1, \cdots, N \tag{24}$$

where $\hat{\mathbf{U}}_s = \mathbf{B} \cdot (\mathbf{U} + \mathbf{N}_0)$. 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.

1) Performance analysis with respect to the system parameters: The performance analysis of the LSDA estimator in the pool mix for static user behavior is carried out in Appendix B, where it is shown that

$$\begin{aligned} \text{MSE}_i \quad \approx \quad & \frac{1}{\rho}\left\{(f_i^{-1}-1)\left[\bar{\mu}\left(\frac{1}{\alpha_r}-\frac{1}{t}\right)+\left(\frac{1}{\alpha_q}-\frac{1}{\alpha_r}\right)\right]\right. \\ & + \quad \left.\frac{f_i^{-1}}{t}\cdot\mu_i\right\} \end{aligned} \tag{25}$$

where $\alpha_q \doteq \dfrac{\alpha}{2-\alpha}$ and $\alpha_r \doteq \dfrac{\alpha(2-\alpha)}{2-\alpha(2-\alpha)}$. This approximation is asymptotically tight as $\rho \to \infty$. Moreover, when $\alpha = 1$ we recover (12).

Comparing the MSE estimator in (25) with the threshold mix one (12), we can conclude that the difficulty of learning the profiles is always larger in the pool mix, since $1/\alpha_r \geq 1$ and $(1/\alpha_q - 1/\alpha_r) \geq 0$. In the particular case that we have analyzed for the threshold mix, where we have assumed that $f_i^{-1} \gg 1$ and $\mu_i \approx 1$, (25) can be approximated by

$$\text{MSE}_i \approx \frac{f_i^{-1}}{\rho} \cdot \frac{2-\alpha}{\alpha}. \tag{26}$$

Comparing this approximation with (13) we can conclude that the pool mix requires *approximately* $(2-\alpha)/\alpha$ times more rounds to achieve the same MSE. 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 $(1-\alpha)/\alpha$.

TABLE II: System parameters used in the experiments.

Param	Value
ρ	$\{\mathbf{10\,000}, 20\,000, \ldots, 100\,000\}$
N	$\{25, 50, 75, \mathbf{100}, 200, 300, \ldots, 1\,000\}$
n_f	$\{10, 20, \mathbf{25}, 30, 40, 50, \cdots, 100\}$
f_i	$\{\mathbf{uniform}(f_i = 1/N), Zipf(f_i = i^{-1}/\sum_{k=1}^{N} f_k)\}$
t	$\{1, 2, 5, \mathbf{10}, 20, 30, 40, 50\}$
α	$\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1\}$

B. Constrained Least Squares Estimation on Pool Mixes

We remark that it is also possible to implement a constrained version of the estimator in the pool mix which forces the profiles to lie in the feasible set $\mathcal{P}$ by just replacing matrices $\mathbf{U}$ by $\hat{\mathbf{U}}_s$ in (15). As we will confirm in the evaluation section, the constrained version outperforms the unconstrained one.

VI. Evaluation

A. Experimental setup

We evaluate the effectiveness of LSDA against synthetic anonymized traces created by a simulator written in the Matlab language.[1] We simulate a population of N users with n_f contacts each to whom they send messages following a Zipf distribution. We have chosen the Zipf distribution as a particular case of the power law probability distribution often used to model social networks [21]. We note that both the LSDA estimator and the theoretical approximation of its performance (12) are not restricted to any specific shape of sender profiles and therefore they would be applicable in any other case (e.g., [22]). In our baseline experiment, users send messages with the same frequency ($f_i = 1/N$ for all i) although we also simulate the case where users do not participate evenly in the system. For all the experiments in this section, we keep the sender profiles and sending frequencies constant between rounds, as assumed by our model in Sect. III-A.

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 in each round t messages arrive to the mix and each message in the pool has a probability α of leaving the mix. We evaluate the results for the case that the adversary observes ρ rounds of mixing. Table II summarizes the values of the parameters used in our experiments, where bold numbers indicate the parameters of the baseline experiment.

We compare the effectiveness of the unconstrained (LSDA) and constrained (C-LSDA) versions of our attack when profiling users with respect to the following attacks:

- SDA-MD: we have implemented Mathewson and Dingledine's version of the Statistical Disclosure Attack as explained in [12] for the threshold mix. However, we have found that the extension of the attack to the pool mix in [12] returned incorrect profile estimations due to

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

scaling issues in its formulation. We have corrected the formula so it does estimate the profiles properly while keeping the philosophy of the attack (we have used the formula for `SDA1` in [23] using matrix $\hat{\mathbf{U}}_s$ instead of $\mathbf{U}$).

- PMDA: the Perfect Matching Disclosure Attack, proposed by Troncoso et al. in [15], estimates the users profiles in two steps: first, starting with an initial estimation of the profiles (e.g., SDA), it computes the most probable correspondence between input and output messages in each round. Then, using that information, it builds a new estimation of the profiles based on a weight parameter z, which is $z = 0.5$ in [15]. We have implemented the attack using $z = 1$, since we have observed that this yields the best results. Note that, because of this, the results of PMDA in this paper differ from those in [22].
- Vida: the Bayesian inference attack proposed by Danezis and Troncoso in [10] allows to draw samples from the distribution of the sender profiles given the observations. We have fixed errors in the implementation of the algorithm in [10], [22]. Also, in order to compare with the rest of the attacks, we have decided to compute the average of the samples obtained using the Bayesian inference attack, since the MSE of this average profile is always lower than the average MSE of the individual samples.

B. Performance 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 sender profile* (MSE_i), previously defined in (11), measures the squared error between the estimated sender profile of user i, $\hat{\mathbf{q}}_i$, and the real sender profile $\mathbf{q}_i$. Secondly, the *average Mean Squared Error per transition probability* measures the average squared error in each transition probability $\hat{p}_{j,i}$,

$$\text{MSE}_p \doteq \frac{1}{N^2}\sum_{i=1}^{N}\text{MSE}_i = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N}\text{E}\left\{(p_{j,i}-\hat{p}_{j,i})^2\right\}. \tag{27}$$

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 (ρ, N, μ_i, f_i, t, α) we record the sets of senders and receivers during ρ rounds and compute the MSE_p (or MSE_i) for each of the attacks. We repeat this process 100 times and plot the average of the results in our figures.

C. Results: Threshold mix

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

1) Performance with respect to the number of rounds ρ: As we discuss in IV-A2, the number of observed rounds ρ has a dominant role in the estimation error incurred by LSDA. We plot in Fig. 2 the MSE per transition probability MSE_p for LSDA, C-LSDA, SDA-MD and PMDA.

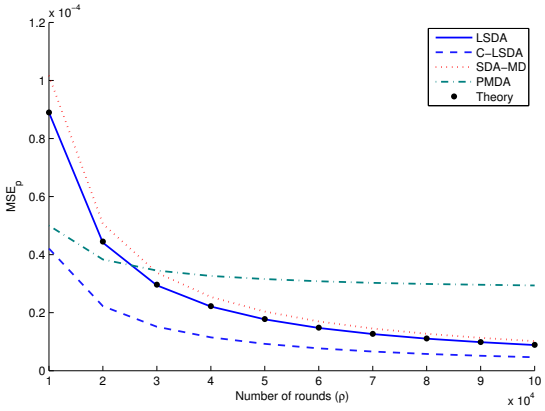


Fig. 2: MSE_p evolution with the number of rounds in the system ρ ($N = 100$, $n_f = 25$, $f_i = 1/N$, $t = 10$).

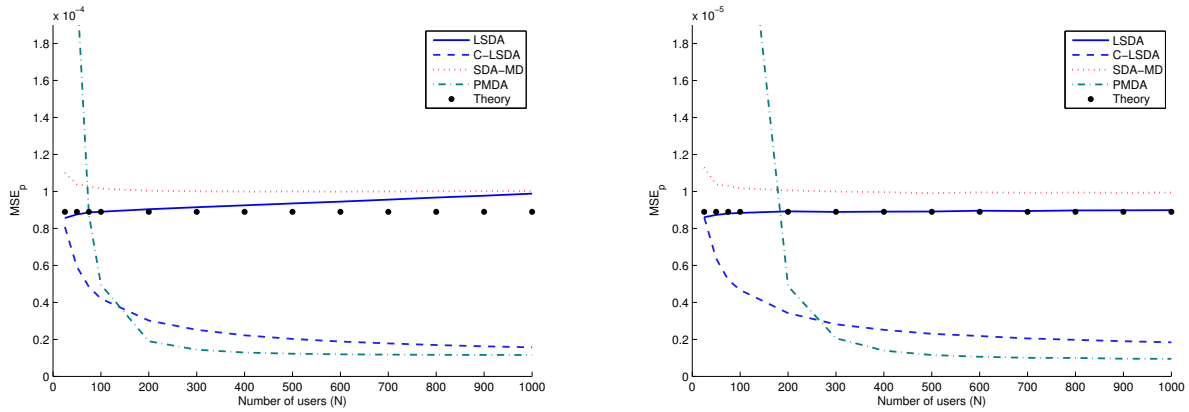
The constrained LSDA obtains the best results, although the unconstrained variant and Mathewson and Dingledine's version of SDA follow up closely. These three estimators are unbiased and asymptotically efficient, i.e., $\text{MSE}_p \to 0$ when $\rho \to \infty$. Furthermore, we can see how the approximation in (12), represented by • in the figure, reliably describes the decrease in the profile estimation error as more information is made available to the adversary. On the other hand, PMDA relies on an initial estimation of the profiles which, for $N = 100$, is far from reality. Therefore, even when the number of rounds observed increases, PMDA is not able to improve as effectively as the other attacks.

2) Performance with respect to the number of users N: Next, we study the influence of the number of users in the system on the estimation error. The results are shown in Fig. 3a for $\rho = 10\,000$ and Fig. 3b for $\rho = 100\,000$. Note that we have adjusted the vertical axis of Fig. 3b so that it is 10 times smaller than that in Fig. 3a.

The MSE_p of LSDA increases with N in Fig. 3a in a way that it is not predicted by our estimation (12). This difference is due to the approximation taken in (30) when deriving the MSE formula and is reduced as the number of rounds observed increases, as shown in Fig. 3b.

Both C-LSDA and PMDA improve their result as the number of users increases. The initial estimation of PMDA improves with N, which allows the attack to achieve better results also when ρ grows (Fig. 3b). However, we note that C-LSDA eventually outperforms PMDA when increasing only N or ρ. On the one hand, the advantage PMDA gains by increasing N does not improve constantly if ρ is kept fixed, while the MSE of C-LSDA keeps decreasing with N. Also, C-LSDA is asymptotically efficient while PMDA is not, which means that C-LSDA will eventually outperform PMDA when increasing ρ regardless of the number of users in the system.

3) Performance with respect to $\bar{\mu}$ and μ_i: In this section, we first analyze the performance of the attacks with $\bar{\mu}$ by increasing the number of friends n_f assigned to all users in steps of 10, as shown in Table II. In this experiment, $\mu_i = \bar{\mu}$ for all i. The results are displayed in Fig. 4a. As we already hinted in Sect. IV-A2, user profiles are harder to estimate in LSDA (C-LSDA) when $\bar{\mu}$ is closer to 1, due to the sending



(a) $\rho = 10000$ (b) $\rho = 100000$

Fig. 3: MSE_p evolution with the number of users in the system N ($\rho = 10\,000, 100\,000$, $n_f = 25$, $f_i = 1/N$, $t = 10$).

behavior of the users becoming more random. PMDA is able to guess the correspondence between the input and output messages more often when users focus their traffic on few others, which happens at lower values of $\bar{\mu}$. The results of SDA-MD, however, are almost independent of $\bar{\mu}$ and μ_i since it groups the sending behavior of the users and thus is not able to exploit the hubness of the traffic, as explained in [23].

Figure 4b shows the MSE per sender profile, MSE_i, in a scenario where the parameters μ_i are different for every user. The same reasons outlined above explain the performance of the attacks in this case.

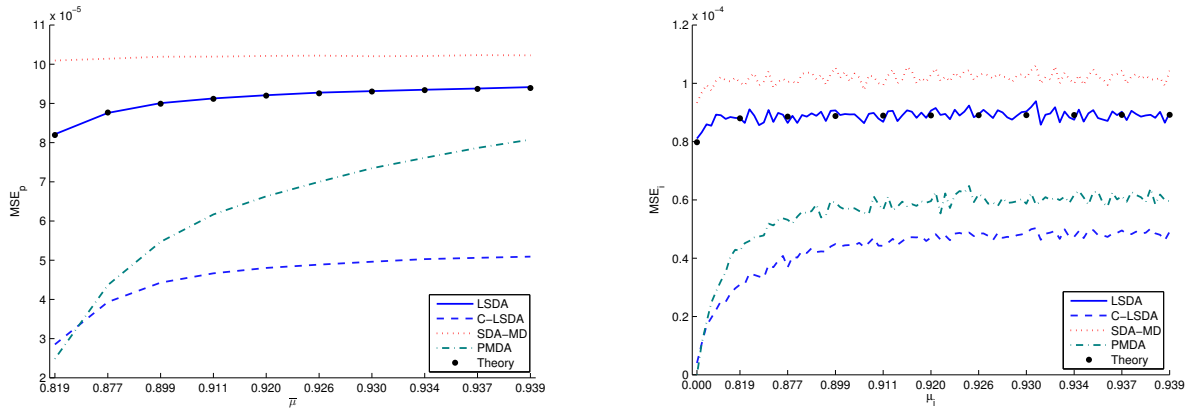
4) Performance with respect to the sending frequencies f_i: So far, we have assumed that every user participates in the system equally often. We now study what happens when this is not the case. Figure 5 shows the MSE_i in a scenario where f_i varies among users. As we have outlined in Sect. IV-A2, the error in the estimation of a user's profile is tied to the participation of that user in the system. It is important to note that, for users who rarely send messages, PMDA seems to achieve better results than C-LSDA. This is because, for those users, the one-to-one correspondence of the messages sent and received in each round is very valuable information and LSDA does not exploit it. Nevertheless, if more round observations are available to the attacker, LSDA (C-LSDA) will eventually outperform PMDA.

5) Performance with respect to the mix threshold t: By observing (12) one can see that the threshold t of the mix has little influence on the MSE_p of LSDA, becoming negligible as $t \gg 1$. This is reflected by our experiments, shown in Fig. 6, where the error of 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. Hence, although the error is constant with t, increasing the threshold delays the obtaining of accurate user profiles. Note that this protection the mix offers when increasing t comes at the cost of an increased delay in the communications.

As expected, increasing the threshold has a negative effect on the other two attacks. The error in SDA-MD grows slightly with t, since increasing this parameter decreases the number of rounds where user i does not participate and this makes harder to estimate the behavior of the messages not sent by i. On the other hand, as the threshold grows the number of plausible matchings increases and thus the likelihood that the most probable matching is the real one decreases. This in turn worsens PMDA's performance significantly.

6) Comparison between attack principles: Throughout the evaluation section we have studied three 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]. For the sake of comparison, we also test the efficacy of simply setting the negative probabilities output by the unconstrained LSDA and SDA-MD to zero (denoted as Z-LSDA and Z-SDA-MD, respectively). In order to illustrate the differences in computational load of the attacks, we have measured their running-time, carrying the experiment in a server with a Core2 Quad Q8300 2.5GHz processor, 8GB of RAM and Matlab version 7.13.0.564 (64-bit) on Ubuntu 12.04.3 LTS.

Figure 7 shows box plots representing the distribution of the MSE_p obtained after 20 repetitions of our baseline experiment. We have already discussed that LSDA obtains an advantage over SDA-MD by solving the problem for all users simultaneously, but does not account for the one-to-one relationship between sent and received messages in the individual rounds of mixing as PMDA does. Recall, however, that both LSDA and SDA-MD outperform PMDA when the number of rounds observed is sufficiently large. When not considering the negative probabilities, both LSDA and SDA-MD improve their performance. However, only LSDA can be formulated as a constrained optimization problem (C-LSDA) and, when doing so, it achieves results really close to Vida's performance. This improvement comes at an increase in the computational cost: in this particular experiment, implementing LSDA as in (14) and C-LSDA following (15), the latter was on average about 25 times slower than LSDA (in each realization, C-LSDA took always less than 2 seconds to finish). The approach followed in Vida improves the profile estimations considerably with respect to the other attacks. While the effectiveness of Vida is desirable, it comes at a huge computational cost because each



(a) (b)

Fig. 4: MSE_p evolution with $\bar{\mu}$ in a scenario where μ_i is constant (a) and MSE_i's in an experiment where each user has a different value of μ_i (b) ($\rho = 10\,000$, $N = 100$, $f_i = 1/N$, $t = 10$).

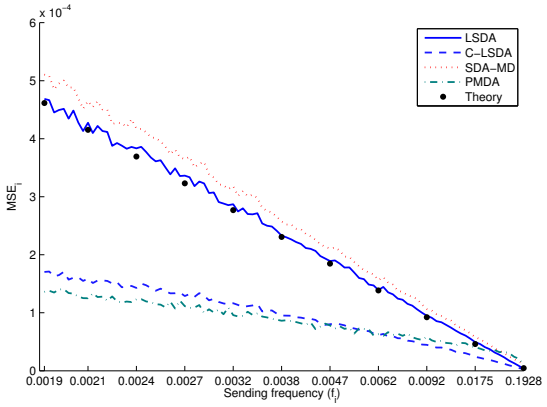


Fig. 5: Evolution of the MSE per sender profile with the sending frequency of the users f_i ($\rho = 10\,000$, $N = 100$, $n_f = 25$, $t = 10$).

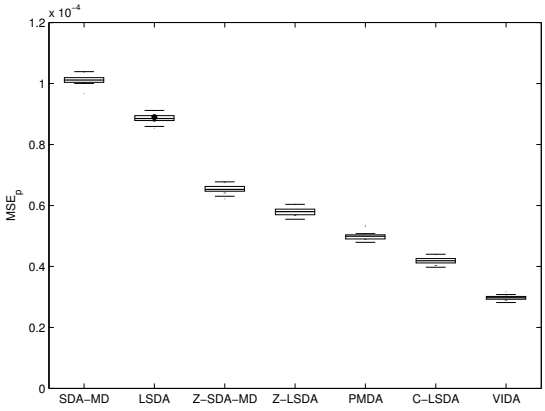


Fig. 7: Comparison between attack principles in a threshold mix ($\rho = 10\,000$, $N = 100$, $n_f = 25$, $f_i = 1/N$, $t = 10$).

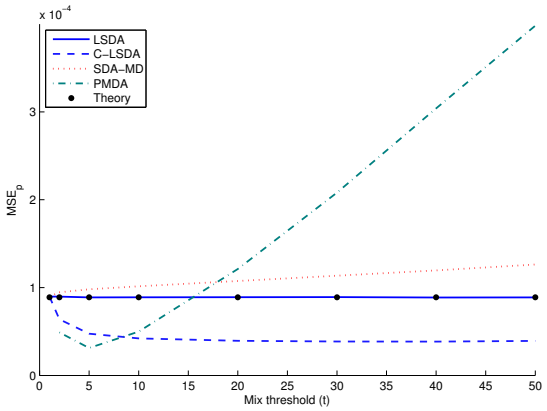


Fig. 6: MSE_p evolution with the threshold t ($\rho = 10\,000$, $N = 100$, $n_f = 25$, $f_i = 1/N$).

iteration of the algorithm requires finding a perfect matching in all the ρ rounds observed. In this case, Vida was on average $280\,000$ times slower than LSDA (each realization took always more than 4 hours to finish).

D. Results: Pool mix

We now proceed to evaluate LSDA's profiling performance when messages are anonymized using a threshold binomial pool mix. We recall that, in a threshold binomial pool mix, arriving messages are stored in a pool and leave the mix each round (i.e., when t messages are received) with probability α. Otherwise, messages stay in 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 LSDA with SDA-MD, the most effective attack in the literature that has been applied to pool mixes, which we have implemented as explained in Sect. VI-A.

In this case, we only analyze the performance of the attacks with the firing probability α since, as we show in Appendix B with (25), the behavior of LSDA in the pool mix with all the other system parameters (i.e., ρ, N, μ_i, f_i, t) does not change with respect to the threshold mix case.

1) Performance with respect to the firing probability α: 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 $(1 - \alpha)/\alpha$.

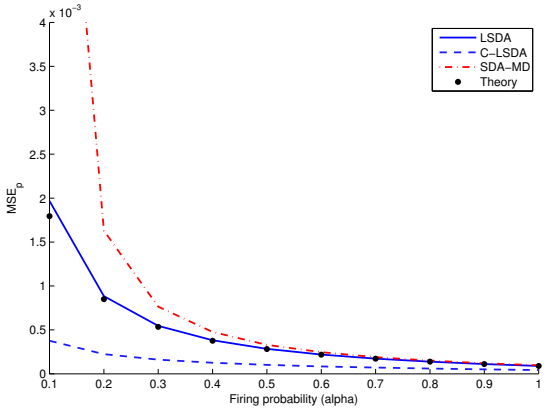


Fig. 8: MSE_p evolution with the firing probability α ($\rho = 10\,000$, $N = 100$, $n_f = 25$, $f_i = 1/N$, $t = 10$).

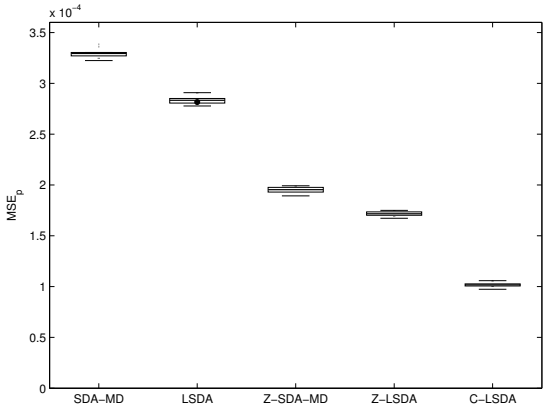


Fig. 9: Comparison between attack principles in a pool mix ($\rho = 10\,000$, $N = 100$, $n_f = 25$, $f_i = 1/N$, $t = 10$, $\alpha = 0.5$).

In Fig. 8, we illustrate the tradeoff between the profiling accuracy of 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 (the mean size of the pool is $\frac{t-\alpha t}{\alpha}$), 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 (25).

The figure also shows the evolution of the MSE per transition probability MSE_p of Mathewson and Dingledine's SDA-MD. When the firing probability is low, messages from users are carried further to the next rounds. This, in time, reduces the number of rounds that can be used to estimate the behavior of all users but Alice (similarly to what happens when increasing t), making the estimation of the profiles increasingly difficult.

2) Comparison between attack principles: We show, in Fig. 9, box plots representing the distribution of the MSE_p for LSDA, C-LSDA, SDA-MD, Z-LSDA and Z-SDA-MD in a pool mix with $\alpha = 0.5$. These results emphasize that LSDA (C-LSDA) is the attack which performs better in the pool mix.

VII. Extensions and Limitations of the Least Squares Disclosure Attack

A. Real-world behavior

In a realistic scenario, the frequency with which users send messages to the anonymous communication system, as well as their choices of recipients (i.e., their sending profiles), are likely to evolve over time. However, the analysis developed in this paper assumes that profiles and sending frequencies are static and hence it is only strictly applicable to observed time windows in which users' behavior does not change. As the length of the window increases, users' behavior evolution causes a deviation of the adversary's observation from these assumptions, and the accuracy of the error prediction diminishes accordingly. We now discuss how our analysis can be applied and extended to take into account behavior evolution, and empirically evaluate the performance of the static and non-static error estimation against real data.

Evolving behavior can be modeled by a stochastic process describing the collection of random variables that represent potential users' profiles. Hence, the behavior leading to a particular observation can be modeled by the random processes $\{F_i^r\}$ and $\{P_{j,i}^r\}$ whose respective realizations $\{f_i^r\}$ and $\{p_{j,i}^r\}$ represent the actual values of the sending frequencies and transition probabilities in each round. We note that such a model can accommodate temporal changes of the profiles due to the evolution of users' individual preferences, as well as changes due to interactions between users (e.g., replies to messages or groups of receivers often contacted simultaneously).

In this scenario, LSDA can be used to estimate the average value of the random process $\{P_{j,i}^r\}$ representing the users' *average sending profiles* in the observed period, i.e., the average proportion of messages sent in this period to each of the possible recipients of the users. Notice that this corresponds to a frequentist interpretation of the user profiles. In fact, it can be shown that LSDA is an *unbiased and efficient estimator* of the users' average sending profile as long as the sending profiles in each round $p_{j,i}^r$ can be modeled as a realization of a wide-sense stationary process $\{P_{j,i}^r\}$ [24].

Following the approach carried out in the Appendices for the static case, it is possible to characterize the estimation error for the case of time-varying profiles [24]. This analysis reveals that time-varying profiles increase the variance of the outputs given the input observations, which in turn slightly increases the adversary's estimation error. On the other hand, time-varying sending frequencies increase the variance of the input process, providing the attacker with a wider variety of input observations than in the static case. This in turn reduces LSDA's MSE and allows the adversary to obtain better estimates of the user profiles [24].

In order to show the usefulness of our methodology in presence of real traffic, we evaluate the performance of our error estimation analysis using the public Enron dataset.[2] This dataset consists of $N = 294$ senders from this database (we have removed 11 users that send less than 20 messages during the collection period), which send a total of 220 032

[2] http://www.cs.cmu.edu/~./enron/

messages to 17 009 receivers. In this experiment, messages are used as input to an anonymous channel implemented as a threshold mix with threshold $t = 10$, obtaining 22 000 mixing rounds. Note that, although the model in Sect. III assumes that the number of senders and receivers is the same, it is straightforward to extend the attack to the case where these numbers do not coincide by redefining the sizes of the involved vectors and matrices. Since we do not have access to the real profiles and sending frequencies of the users that generated the trace, for our experiments we compute the real probabilities $p_{j,i}$ as the proportion of messages from user i sent to j in the observed period, and the parameters f_i as the fraction of all incoming messages to the mix that were sent by user i.

We estimate the sender profiles of the users that participate in the system using LSDA for $\rho = \{2\,200, 4\,400, \ldots, 22\,000\}$ rounds, and compute the MSE for each estimated transition probability $\hat{p}_{j,i}$. We plot in Fig. 10 the average MSE per transition probability (MSE_p) of 100 users (thick straight line) that send messages throughout the full observation period. The figure also shows the theoretical estimation of the error using (12) (dotted line), and using the error prediction for non-static cases provided in [24] for the cases when sending frequencies vary over time while profiles stay invariant (dashed lines), and when profiles vary over time while sending frequencies remain fixed (thin straight line).

As expected, real traffic does not behave according to the assumptions made in the error estimation performance analysis, and hence the results are less accurate than in the simulations in Sect. VI. Yet, the predictor based on static parameters is not far from the empirical result (the predicted transition probabilities are off by at most 1.5×10^{-5}), and correctly follows the trend of the error as more information is made available to the adversary. In fact, in [24] we show that our formula in (12) practically gives an upper bound to the empirical MSE when the number of users in the system is relatively large. Therefore, the results provided in this paper conversely serve as a lower bound to the *privacy loss* of mixes as rounds of observations become available to the attacker, even when users' profiles and sending frequencies evolve with time. This highlights the usefulness of the theoretical results provided in this paper, pointing at a fundamental weakness of existing mix-based anonymous communication systems. Such results generalize the findings of Kesdogan et al [17], proving that anonymity protection limits are caused by diversification in user behavior rather than by the observation of changing anonymity sets.

As a final remark, note that LSDA can be used to infer how fast the sending behavior of the users evolves with time. To this end, the adversary can split the observation in shorter time windows (which may overlap) and use LSDA to obtain the users' average sending profile within each subset of consecutive rounds of mixing. By varying the size of the window the adversary can detect when the profiles change.

B. Adversarial prior knowledge

In some cases it might be possible that the adversary has some prior information on the transition probabilities. While it is possible to modify the machine learning approach [10] to account for this extra knowledge, this is non-trivial for SDA or PMDA. In contrast, the least squares formulation can be easily adapted to consider this additional information: if the values of some probabilities are known, the attack can be extended in a similar way as we did to show that the original SDA is a particular, but largely suboptimal, instance of LSDA. On the other hand, the prior knowledge may be given as a set of constraints on the transition probabilities (e.g., the adversary knows that user i contacts user j at least half of the times). In that case, our iterative implementation of C-LSDA in Sect. IV-B could be adapted to such scenario by just projecting the solution onto the new set of constraints.

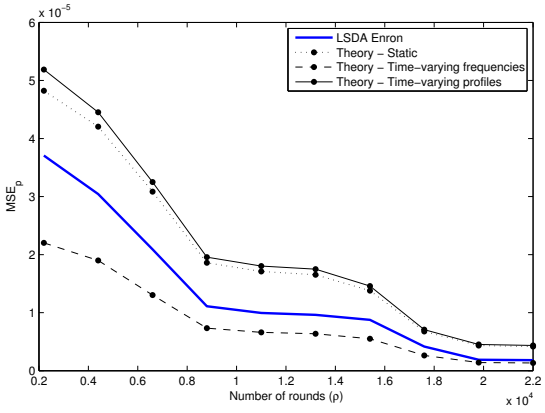


Fig. 10: Empirical and theoretical MSE_p of LSDA against the Enron dataset.

C. User and background cover traffic

Cover traffic, i.e., fake messages generated by the users and/or the mix, makes harder for the attacker to infer the users' sending profiles [25]. In this scenario, dummy messages can be regarded as noise in the input and output observations. In order to derive the LSDA estimator in this case, we would look for the profiles which minimize the error between the observed and the expected output, now considering that this output consists of real and dummy messages and that the input observations are noisy.

VIII. Conclusions

We have introduced the Least Squares Disclosure Attack, which estimates user profiles by 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 prediction is very accurate when users' behavior is static, and when this assumption does not hold it gives a good approximation that correctly follows the trend of the error as more information is made available. 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 expensive simulations [10], [15].

Under the hypotheses of static users' behavior, we have proven that, contrary to other approaches [15], this least squares estimator is asymptotically efficient, meaning that if the attacker is given enough observations of the system, the error on the estimation of the user profiles will approach zero. This finding generalizes previous results on the limits of the protection of mix-based anonymous communication systems, proving that such limits are imposed by varying user behavior rather than changing anonymity sets.

Our experiments confirm that, in simple threshold mixes, LSDA outperforms the state-of-the-art version of the Statistical Disclosure Attack [12] and that a more sophisticated implementation of LSDA gives results close to those of the computationally expensive Bayesian inference approach [10] at an affordable cost. Our attack is not limited to the analysis of threshold mixes but can be easily extended to more complex mixing strategies such as pool mixes [16]. In these complex scenarios, the Bayesian approach is untractable and LSDA yields the best results.

APPENDIX A
DERIVATION OF MSE_i FOR THE THRESHOLD MIX

Our goal is to derive an expression for the Mean Squared Error per user, $\mathrm{MSE}_i \doteq \sum_{j=1}^{N} \mathrm{E}\left\{(p_{j,i} - \hat{p}_{j,i})^2\right\}$, when using the LSDA estimator in (6) in a threshold mix scenario. To this end, we first remark that the unconstrained least squares estimate (6) is unbiased: it is straightforward to show that $\mathrm{E}\{\hat{\mathbf{p}}_j\} = \mathbf{p}_j$ by using the law of total expectation and $\mathrm{E}\left\{\mathbf{Y}_j|\mathbf{U}\right\} = \mathbf{U}\cdot\mathbf{p}_j$:

$$\begin{aligned}\mathrm{E}\{\hat{\mathbf{p}}_j\} &= \mathrm{E}\left\{\mathrm{E}\left\{\hat{\mathbf{p}}_j|\mathbf{U}\right\}\right\} = \mathrm{E}\left\{\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T\mathrm{E}\left\{\mathbf{Y}_j|\mathbf{U}\right\}\right\} \\ &= \mathrm{E}\left\{\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T\mathbf{U}\cdot\mathbf{p}_j\right\} = \mathbf{p}_j\,.\end{aligned} \tag{28}$$

On the other hand, we can use the law of total variance together with the fact that, from (28), $\mathrm{Var}\{\mathrm{E}\{\hat{p}_{j,i}|\mathbf{U}\}\} = 0$ and $\mathrm{Cov}\{\mathrm{E}\{\hat{p}_{j,i}|\mathbf{U}\}, \mathrm{E}\{\hat{p}_{j,k}|\mathbf{U}\}\} = 0$ for $k \neq i$, to express the covariance matrix of $\mathbf{p}_j$ as

$$\mathbf{\Sigma}_{\mathbf{p}_j} = \mathrm{E}\{\mathbf{\Sigma}_{\mathbf{p}_j|\mathbf{U}}\} = \mathrm{E}\{(\mathbf{U}^T\mathbf{U})^{-1}\mathbf{U}^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\mathbf{U}(\mathbf{U}^T\mathbf{U})^{-1}\} \tag{29}$$

where $\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}} = \mathrm{E}\{(\mathbf{Y}_j - \mathrm{E}\left\{\mathbf{Y}_j|\mathbf{U}\right\})(\mathbf{Y}_j - \mathrm{E}\{\mathbf{Y}_j|\mathbf{U}\})^T|\mathbf{U}\}$.

We model $\{X_1^r, \cdots, X_N^r\}$ jointly as a multinomial distribution with t trials and probabilities $\{f_1, \cdots, f_N\}$. Also, we note that X_i^r and X_k^s are independent when $r \neq s$. In order to compute (29), we first point out that, since the input process is stationary and memoryless, then using the Law of Large Numbers we can write

$$\lim_{\rho\to\infty} \mathbf{U}^T\mathbf{U}/\rho \to \mathbf{R}_x \tag{30}$$

where $\mathbf{R}_x$ is the autocorrelation matrix of the input process. We can write this autocorrelation matrix, using

$$\mathrm{E}\{X_i^2\} = \left(t^2 - t\right) f_i^2 + t f_i \tag{31}$$

$$\mathrm{E}\{X_i X_k\} = \left(t^2 - t\right) f_i f_k \qquad \text{for } i \neq k \tag{32}$$

as

$$\mathbf{R}_x = t\left[\mathbf{\Lambda_F} + (t-1)\,\mathbf{f}\cdot\mathbf{f}^T\right] \tag{33}$$

where $\mathbf{f} \doteq [f_1, \cdots, f_N]^T$ and $\mathbf{\Lambda_F} \doteq \mathrm{diag}\{\mathbf{f}\}$. Applying the Sherman-Morrison formula [26], the inverse of this autocorrelation matrix can be written as

$$\mathbf{R}_x^{-1} = \frac{1}{t}\left[{\mathbf{\Lambda_F}}^{-1} - \left(1 - \frac{1}{t}\right)\mathbf{1}_{N\times N}\right]. \tag{34}$$

Assuming that the number of observed rounds ρ is large, we can approximate (29) as

$$\mathbf{\Sigma}_{\mathbf{p}_j} \approx \frac{1}{\rho^2}\mathbf{R}_x^{-1}\mathrm{E}\{\mathbf{U}^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\mathbf{U}\}\mathbf{R}_x^{-1}. \tag{35}$$

We now focus on the term $\mathrm{E}\{\mathbf{U}^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\mathbf{U}\}$. We model $Y_j^r|\mathbf{U}$ as the sum of N binomial processes with x_i^r trials and probabilities $p_{j,i}$, for $i = 1, 2, \cdots, N$. Note that Y_j^r and Y_j^s are independent for $r \neq s$. Let $s_{j,i} \doteq p_{j,i}\cdot(1 - p_{j,i})$ and $\mathbf{S}_j \doteq \mathrm{diag}\{s_{j,1}, \cdots, s_{j,N}\}$. Then, $\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}$ is a diagonal matrix whose (r,r)-th element is

$$\left(\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\right)_{r,r} = \sum_{i=1}^{N} x_i^r s_{j,i}\,. \tag{36}$$

Operating, we get that the (m,n)-th element of $\mathrm{E}\{\mathbf{U}^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\mathbf{U}\}$ is

$$\left(\mathrm{E}\{\mathbf{U}^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\mathbf{U}\}\right)_{m,n} = \rho\sum_{i=1}^{N} s_{j,i}\cdot\mathrm{E}\{X_i X_m X_n\} \tag{37}$$

and therefore we can write this term as

$$\begin{aligned}&\mathrm{E}\{\mathbf{U}^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\mathbf{U}\} = \\ &\rho\left[\mathbf{\Lambda_F}\left(\eta_j t^{(3)}\mathbf{1}_{N\times N} + \mathbf{S}_j\mathbf{1}_{N\times N}t^{(2)} + \mathbf{1}_{N\times N}\mathbf{S}_j t^{(2)}\right)\mathbf{\Lambda_F}\right] \\ &+ \rho\left[\left(\eta_j t^{(2)}\mathbf{I}_{N\times N} + t\mathbf{S}_j\right)\mathbf{\Lambda_F}\right]\end{aligned} \tag{38}$$

where $\eta_j \doteq \sum_{i=1}^{N} f_i s_{j,i}$ and $t^{(n)} \doteq t\cdot(t-1)\cdots(t-n+1)$.

Plugging (38) into (35) we get an approximation of $\mathbf{\Sigma}_{\mathbf{p}_j}$. Now, taking each of the diagonal elements of this matrix, which are $\mathrm{Var}\{\hat{p}_{j,i}\}$ for $i = 1, \cdots, N$ and adding them along j to obtain $\mathrm{MSE}_i \doteq \sum_{j=1}^{N}\mathrm{Var}\{\hat{p}_{j,i}\}$, we finally get

$$\mathrm{MSE}_i \approx \frac{1}{\rho}\left\{\left(f_i^{-1} - 1\right)\left(1 - \frac{1}{t}\right)\bar{\mu} + \frac{f_i^{-1}}{t}\cdot\mu_i\right\}. \tag{39}$$

APPENDIX B
DERIVATION OF MSE_i FOR THE POOL MIX

We aim here at deriving an expression for the MSE in the estimation of the sender profile of user i, previously defined as $\mathrm{MSE}_i \doteq \sum_{i=1}^{N}\mathrm{E}\{(p_{j,i} - \hat{p}_{j,i})^2\}$, for the LSDA estimator in the pool mix (24). To this end, we follow the same approach as for the threshold mix. We will assume that $\mathbf{N}_0 = \mathbf{0}$, as the impact of the initial conditions can be neglected for large ρ.

We start by showing that this estimator is unbiased. Following (18), we can write $\mathrm{E}\{\mathbf{Y}_j|\mathbf{U}\} = \hat{\mathbf{U}}_s\cdot\mathbf{p}_j$. Then,

$$\begin{aligned}\mathrm{E}\{\hat{\mathbf{p}}_j\} &= \mathrm{E}\left\{\mathrm{E}\left\{\hat{\mathbf{p}}_j|\mathbf{U}\right\}\right\} = \mathrm{E}\left\{\left(\hat{\mathbf{U}}_s^T\hat{\mathbf{U}}_s\right)^{-1}\hat{\mathbf{U}}_s^T\mathrm{E}\left\{\mathbf{Y}_j|\mathbf{U}\right\}\right\} \\ &= \mathrm{E}\left\{\left(\hat{\mathbf{U}}_s^T\hat{\mathbf{U}}_s\right)^{-1}\hat{\mathbf{U}}_s^T\hat{\mathbf{U}}_s\cdot\mathbf{p}_j\right\} = \mathbf{p}_j\,.\end{aligned}$$

Using the law of total variance, we can now write

$$\mathbf{\Sigma}_{\mathbf{p}_j} = \mathrm{E}\{(\hat{\mathbf{U}}_s^T\hat{\mathbf{U}}_s)^{-1}\hat{\mathbf{U}}_s^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\hat{\mathbf{U}}_s(\hat{\mathbf{U}}_s^T\hat{\mathbf{U}}_s)^{-1}\} \tag{40}$$

where $\hat{\mathbf{U}}_s = \mathbf{BU}$. As in the threshold mix case, we approximate (40), assuming that the number of observed rounds ρ is large enough, as

$$\mathbf{\Sigma}_{\mathbf{p}_j} \approx \frac{1}{\rho^2}\mathbf{R}_{xs}^{-1}\mathrm{E}\{\hat{\mathbf{U}}_s^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\hat{\mathbf{U}}_s\}\mathbf{R}_{xs}^{-1}\,. \tag{41}$$

The (m,n)-th element of the autocorrelation matrix in the pool mix case, $\mathbf{R}_{xs}$, is

$$(\mathbf{R}_{xs})_{m,n} = \frac{1}{\rho}\sum_{k=1}^{\rho}\sum_{r=1}^{k}\sum_{s=1}^{k}\mathrm{E}\{X_m^r X_n^s\}\alpha^2(1-\alpha)^{2k-r-s}\,. \tag{42}$$

In order to get an expression for this matrix, we can use (31) and (32) since the distribution of the input process in the pool mix is the same as in the threshold mix. If we assume that $\rho \gg 1/\alpha$, and define $\alpha_q = \alpha/(2-\alpha)$, then we can approximate this autocorrelation matrix by $\mathbf{R}_{xs} \approx \alpha_q t\mathbf{\Lambda_F} + \left(t^2 - \alpha_q t\right)\mathbf{f}\cdot\mathbf{f}^T$, whose inverse is

$$\mathbf{R}_{xs}^{-1} \approx \frac{1}{\rho\alpha_q t}\left[{\mathbf{\Lambda_F}}^{-1} - \left(1-\frac{\alpha_q}{t}\right)\mathbf{1}_{N\times N}\right]\,. \tag{43}$$

We now focus on $\mathrm{E}\{\hat{\mathbf{U}}_s^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\hat{\mathbf{U}}_s\}$ in (41). In this case, the random variables Y_j^r in different rounds are not independent and therefore we cannot use the $\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}$ that we derived for the threshold mix scenario. Using the law of total variance, it can be shown that

$$\begin{aligned}\mathrm{Var}\{Y_j^r|\mathbf{U}\} = \sum_{k=1}^{r}\sum_{i=1}^{N} x_i^k \quad &\big(p_{j,i}\alpha(1-\alpha)^{r-k} \\ &-p_{j,i}^2\alpha^2(1-\alpha)^{2(r-k)}\ \big)\end{aligned}$$

$$\mathrm{Cov}\{Y_j^r, Y_j^s|\mathbf{U}\} = -\sum_{k=1}^{\min(r,s)}\sum_{i=1}^{N} x_i^k p_{j,i}^2\alpha^2(1-\alpha)^{r+s-2k} \tag{44}$$

which equals, in matricial form, to

$$\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}} = \mathrm{diag}\{\mathbf{BUP}_j\mathbf{1}_N\} - \mathbf{B}\cdot\mathrm{diag}\{\mathbf{UP}_j^2\mathbf{1}_N\}\cdot\mathbf{B}^T \tag{45}$$

where $\mathbf{P}_j = \mathrm{diag}\{\mathbf{p}_j\}$.

The following steps are similar to those performed in the threshold mix case and consist only of laborious matrix multiplications. We omit the full description of these steps for practicality issues and outline the remaining process: using (45), we compute $\mathrm{E}\{\hat{\mathbf{U}}_s^T\mathbf{\Sigma}_{\mathbf{Y}_j|\mathbf{U}}\hat{\mathbf{U}}_s\}$. We multiply the resulting matrix left and right by (43), and then take the diagonal elements of the resulting matrix, which are an approximation of $\mathrm{Var}\{\hat{p}_{j,i}\}$. Adding these elements along j and defining $\alpha_r \doteq \alpha(2-\alpha)/(2-\alpha(2-\alpha))$, we finally obtain

$$\mathrm{MSE}_i \approx \frac{1}{\rho}\left\{(f_i^{-1}-1)\left[\bar{\mu}\left(\frac{1}{\alpha_r}-\frac{1}{t}\right)+\left(\frac{1}{\alpha_q}-\frac{1}{\alpha_r}\right)\right]+\frac{f_i^{-1}}{t}\cdot\mu_i\right\} \tag{46}$$

References

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

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

[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," *Commun. 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*, Gritzalis, Vimercati, Samarati, and Katsikas, Eds., IFIP TC11. Athens: Kluwer, May 2003, pp. 421–426.

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

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

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

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

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

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

[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*, ser. LNCS, V. Atluri and C. Diaz, Eds., vol. 6879. Springer, 2011, pp. 508–527.

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

[16] F. Pérez-González and C. Troncoso, "A least squares approach to user profiling in pool mix-based anonymous communication systems," in *IEEE Workshop on Information Forensics and Security*, 2012, pp. 115–120.

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

[18] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *2nd Workshop on Privacy Enhancing Technologies*, ser. LNCS, R. Dingledine and P. Syverson, Eds., vol. 2482. Springer, 2002, pp. 41–53.

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

[20] S. Haykin, *Adaptive Filter Theory, 4/e*. Prentice Hall, 2002.

[21] A. Clauset, C. R. Shalizi, and M. Newman, "Power-law distributions in empirical data," *SIAM Review*, vol. 51, no. 4, pp. 661–703, 2009.

[22] F. Pérez-González and C. Troncoso, "Understanding statistical disclosure: A least squares approach," in *Privacy Enhancing Technologies - 12th Symposium*, ser. LNCS, vol. 7384. Springer-Verlag, 2012, pp. 38–57.

[23] S. Oya, C. Troncoso, and F. Pérez-González, "Meet the family of statistical disclosure attacks," *IEEE Global Conference on Signal and Information Processing*, p. 4p, 2013.

[24] F. Pérez-González, C. Troncoso, and S. Oya, "Technical report tsc/fpg/20032014: Performance analysis of the least squares disclosure attack with time-varying user behavior," http://webs.uvigo.es/gpscuvigo/sites/default/files/publications/techrep.pdf.

[25] N. Mallesh and M. Wright, "Countering statistical disclosure with receiver-bound cover traffic," in *12th European Symposium on Research in Computer Security*, ser. LNCS, J. Biskup and J. Lopez, Eds., vol. 4734. Springer, 2007, pp. 547–562.

[26] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *The Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 124–127, 1950.