Design of Pool Mixes Against Profiling Attacks in Real Conditions

Simon Oya, *Graduate Student Member, IEEE,* Fernando Pérez-González, *Fellow, IEEE,* and Carmela Troncoso

***Abstract*—Current implementations of high-latency anonymous communication systems are based on pool mixes. These tools act as routers that apply a random delay to the messages traversing them, making it hard for an eavesdropper to guess the correspondences between incoming and outgoing messages. This hides the identities of communicating partners in the network, but it does not prevent an adversary continuously monitoring the network from unveiling the communication profiles of the users. In this work, we tackle the problem of designing the delay characteristic of pool mixes so as to maximize the protection of the users against profiling attacks. First, we propose a theoretical model for users' sending behavior which we validate using three real datasets of different nature. Then, we use this model to perform a privacy analysis of the system and obtain the delay function of the mix which is optimal in the sense of protecting the users. Since computing the delay characteristic of this optimal pool mix requires information about the users' behavior, we also propose a user-independent but less effective mix design. We evaluate these pool mixes, comparing them with one of the most studied existing designs, the binomial pool mix. Our experiments show that an adversary against our optimal design may need up to 30 times as long to achieve the same level of disclosure as for a binomial pool mix.**

***Index Terms*—anonymity, optimization, pool mixes**

I. Introduction

The introduction of mixes by Chaum back in 1981 [1] paved the way to the development of high-latency anonymous communication systems [2], [3], [4]. Mixes can be seen as communication channels that provide unlinkability between the messages they receive and the messages they output. This, in turn, prevents an external observer from inferring who communicates with whom. Mixes provide unlinkability by performing two basic operations: changing the appearance of the messages to avoid bit-wise correlations, which can be done through encryption, and breaking the timing information of the messages to avoid time correlations, which is done by delaying and reordering the messages.

One widely studied type of the mix is the so-called pool mix [3], [5], [6], [7]. This mix stores the messages it receives in a *pool* and applies a random delay to each before forwarding them to their corresponding recipients. This randomness in the delay makes it hard to infer for an external eavesdropper, such as the ISP provider, who is the sender of a given message leaving the mix. However, when the communications take place over a sufficiently long period of time, it is known that an adversary observing the flow of messages traversing the mix can learn information about the communication preferences of the users by means of a disclosure attack [8], [9], [10], [11], [12].

One key factor that determines the degree of protection provided by the pool mix is the so-called *delay characteristic* of the mix, i.e., the function from which the random delays of the messages are drawn. Previous works show that, for a given distribution on the message delay (e.g., geometric distribution), higher average delays provide better protection to the users [7], [11], [13]. However, two delay distributions that produce the same average delay may yield different protection properties. It is thus important to understand how the messages inside the pool should be delayed so as to maximize the anonymity of the users. The search for the optimal delay characteristic of the pool mix has been previously carried out in [7], [13] from an information-theoretic point of view and assuming that the user traffic follows statistical models that are far from being realistic.

In this work we adopt an estimation-theoretic approach to the analysis of pool mixes, studying how to optimize their the delay characteristic so as to maximize the privacy of the users. This complements the information-theoretical approach of [7], [13] and allows us to obtain results in complex and realistic scenarios. We are interested in understanding how to protect the users against *profiling attacks*, i.e., attacks that aim at revealing the long-term communication profiles of the users rather than finding the sender and recipient of a particular message. Our work shows that the optimal design of the delay characteristic actually depends on how users behave in the system, and therefore a user-independent solution is not optimal. We start by presenting a novel theoretical study of mix-based systems that help us to better understand how the behavior of the users affects their privacy. Based on this model, we obtain the delay function that maximizes our anonymity metric, namely the adversary's mean square error. This optimal pool mix design allows users communicating for almost three years with a global adversary eavesdropping the communications to achieve the same level of protection as users communicating for one month through a binomial pool

S. Oya and F. Pérez-González are with the Signal Theory and Communications Dept., University of Vigo (e-mail: simonoya@gts.uvigo.es, fperez@gts.uvigo.es). C. Troncoso is with The IMDEA Software Institute, (e-mail: carmela.troncoso@imdea.org).

This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund (ERDF) under projects TACTICA and COMPASS (TEC2013-47020-C2-1-R), by the Galician Regional Government and ERDF under projects ”Consolidation of Research Units” (GRC2013/009) and AtlantTIC, and by the EU H2020 Framework Programme under project WITDOM (project no. 644371). The work of S. Oya was also supported by the predoctoral grant by the Galician Regional Government, by the Spanish Ministry of Education, Culture and Sport under the grant FPU and by the Fundación Barrié under Programa de Becas de Posgrado en el Extranjero. C. Troncoso is also supported by EU H2020-ICT-10-2015 NEXTLEAP (GA n 688722).

mix [6], one of the state-of-the-art designs. This highlights the importance of optimizing the delay characteristic in pool mixes. We validate our findings with real data, and discuss why previous theoretical analyses are not suitable in practice. The approach we follow in the paper can be summarized in the following steps:

1) We find a theoretical model for the behavior of the users that suits real behavior.
2) We derive a formula that predicts the performance of the system in real scenarios.
3) We study which delay characteristic optimizes this formula from the defender's point of view.
4) We evaluate the designs obtained with real data and compare with the literature.

The rest of the document is structured as follows. In the next section, we introduce the system model and notation used throughout the paper, explain how we measure the privacy of the users and describe the real data we use to evaluate our findings. We propose a theoretical model for user behavior in Section III, which we then use to obtain a mathematical expression that models the degree of protection of the users in the system. With this expression, we solve in Section IV the problem of building an optimal delay characteristic for the pool mix and propose quasi-optimal and sub-optimal variants of this design. We evaluate our solutions and compare them with the binomial pool mix in Section V, and discuss the differences between our estimation-theory approach and the information-theory approach taken in previous analysis in Section VI. We conclude in Section VII.

II. Preliminaries

In this section, we introduce our system and adversary model, together with an explanation of the notation used in the paper. We then explain how we measure privacy and describe the data we use to validate the models and results proposed throughout this work.

A. System Model and Notation

Our system consists of N senders that communicate with M receivers through a mix-based anonymous communication system implementing a pool. The system operates in batches that we call *communication rounds*. The operation of the mix in each of these rounds is described by the following *batching strategy*:

1) The mix gathers messages from the senders, assigns to each of them a waiting time (in rounds) chosen according to a *delay characteristic*, and stores them in its pool.
2) When a certain *flushing condition* triggers (e.g., a timer expires), the mix selects from the pool the messages whose waiting time has expired, changes their appearance using encryption techniques, and forwards them to their corresponding recipients.
3) The mix decreases in one unit the waiting time of the messages that remain in the pool. These messages will be mixed with the ones arriving in subsequent rounds.

Our adversary is a passive eavesdropper that observes all the messages being sent and received in the system during ρ communication rounds. She cannot see the contents of the messages entering and leaving the mix nor establish any bit-wise linkability between them, but she is aware of all the system parameters and knows how the system operates (i.e., the batching strategy). The aim of the attacker is to reconstruct the *sending profiles* of the users, denoted as $\mathbf{q}_i \doteq [p_{1,i}, p_{2,i}, \cdots, p_{M,i}]^T$ for each sender i, where $p_{j,i}$ represents the *probability* that a given message sent by sender $i \in \{1, \cdots, N\}$ is addressed to receiver $j \in \{1, \cdots, M\}$. These profiles represent the intensity with which each sender communicates with each receiver.

The notation we use throughout the paper is illustrated in Fig. 1 and summarized in Table I. We use upper case characters to denote random variables, and lower case characters to denote their realizations. Vectors are denoted by upper-case boldface characters when they contain random variables, and by lower-case boldface characters when they are realizations of random vectors or when they contain constant parameters. Matrices are represented by upper-case boldface characters; whether the values inside them are random variables or realizations will be clear from the context. Matrix $\mathbf{A}^T$ is the transpose of $\mathbf{A}$ (same for vectors), diag$\{\mathbf{a}\}$ is a diagonal matrix whose main diagonal contains the elements of the vector $\mathbf{a}$, and Tr$\{\mathbf{A}\}$ is the trace of matrix $\mathbf{A}$. Matrix $\mathbf{I}_{N\times N}$ is the $N \times N$ identity matrix, $\mathbf{0}_{N\times N}$ and $\mathbf{1}_{N\times N}$ are the $N \times N$ zero and ones matrices and $\mathbf{1}_\rho$ is the $\rho\times 1$ vector of ones. The Euclidean norm of vector $\mathbf{a}$ is denoted by $||\mathbf{a}||$, and the operator $\circ$ is the entrywise or Hadamard product of matrices. Also, $\hat{\mathbf{A}}$ is the adversary's estimation of $\mathbf{A}$ (the same applies to vectors and scalar values).

The random variable that models the number of messages sender i sends in round r is denoted by X_i^r (then, x_i^r is one realization of this variable). The delay characteristic of the mix is defined by the *probability mass function* of the delay, measured in rounds. The probability that a message is delayed k rounds inside the pool is denoted by d_k ($k \geq 0$). The random variable that models the amount of messages from each sender i that leave the pool in round r is Z_i^r, and $Y_{j,i}^r$ models the number of those messages that are addressed to receiver j (note that $Z_i^r = \sum_{j=1}^M Y_{j,i}^r$). The total number of messages leaving the pool for receiver j in round r, from all senders, is $Y_j^r \doteq \sum_{i=1}^N Y_{j,i}^r$.

From these basic random variables, we now form the following vectors and matrices: the vector $\mathbf{X}_i \doteq [X_i^1, \cdots, X_i^\rho]^T$ contains the input process for user i, and the matrix $\mathbf{X} \doteq [\mathbf{X}_1, \cdots, \mathbf{X}_N]$ contains all the observed inputs. Matrix $\mathbf{Z}$ is defined in the same way for the number of messages from each sender that leave the pool in each round, i.e., Z_i^r. Likewise, for the outputs we define vector $\mathbf{Y}_j \doteq [Y_j^1, \cdots, Y_j^\rho]^T$ and matrix $\mathbf{Y} \doteq [\mathbf{Y}_1, \cdots, \mathbf{Y}_M]$. We group the values that form the delay characteristic, d_k, in vector $\mathbf{d} \doteq [d_0, \cdots, d_{\rho-1}]^T$. We also define the convolution matrix $\mathbf{D}$, which is a $\rho \times \rho$ matrix whose r, s-th element is d_{r-s} if $r \geq s$ and 0 otherwise. This matrix is depicted in (2) and will come in handy later. With the probabilities $p_{j,i}$ we define the vector that represents the receiver profile for each receiver j, i.e., $\mathbf{p}_j \doteq [p_{j,1}, \cdots, p_{j,N}]^T$, and the matrix containing all probabilities $\mathbf{P} \doteq [\mathbf{p}_1, \cdots, \mathbf{p}_M]$ (the sending profile $\mathbf{q}_i$ defined before is the i-th row of this

TABLE I: Summary of notation

Symbol	Meaning
N	Number of senders, denoted by $i \in \{1, \cdots, N\}$.
M	Number of receivers, denoted by $j \in \{1, \cdots, M\}$.
ρ	Number of rounds observed by the adversary, $r \in \{1, \cdots, \rho\}$.
$p_{j,i}$	Probability that sender i sends a message to receiver j.
X_i^r	Number of messages sent by sender i in round r.
Z_i^r	Number of messages sent by i leaving the pool in round r.
$Y_{j,i}^r$	Number of messages from i leaving for j in round r.
Y_j^r	Number of messages from all users leaving for j in round r.
d_k	Probability that a message is delayed k rounds in the pool.
$\mathbf{q}_i$	Sending profile of user i, $\mathbf{q}_i \doteq [p_{1,i}, p_{2,i}, \cdots, p_{M,i}]^T$.
$\mathbf{p}_j$	Vector of probabilities per receiver, $\mathbf{p}_j \doteq [p_{j,1}, \cdots, p_{j,N}]^T$.
$\mathbf{P}$	Matrix of all probabilities, $\mathbf{P} \doteq [\mathbf{p}_1, \cdots, \mathbf{p}_M]$.
$\mathbf{X}_i$	Input process for sender i, $\mathbf{X}_i \doteq [X_i^1, \cdots, X_i^\rho]^T$.
$\mathbf{X}$	Matrix with all the inputs, $\mathbf{X} \doteq [\mathbf{X}_1, \cdots, \mathbf{X}_N]$.
$\mathbf{Z}$	$\rho \times N$ matrix containing Z_d^r in its (r, i)-th entry.
$\mathbf{Y}_j$	Output process for receiver j, $\mathbf{Y}_j \doteq [Y_j^1, \cdots, Y_j^\rho]^T$.
$\mathbf{Y}$	Matrix with all the outputs, $\mathbf{Y} \doteq [\mathbf{Y}_1, \cdots, \mathbf{Y}_M]$.
$\mathbf{d}$	Delay characteristic of the mix, $\mathbf{d} \doteq [d_0, \cdots, d_{\rho-1}]^T$.
$\mathbf{D}$	Convolution matrix of the delay characteristic, shown in (2).
$\mathbf{E}$	Estimation error of the adversary, $\mathbf{E} \doteq \hat{\mathbf{P}} - \mathbf{P}$.
$\mathbf{C}_e$	Covariance matrix of the estimation error, $\mathbf{C}_e \doteq \mathrm{E}\left\{\mathbf{E}\mathbf{E}^T\right\}$.
$\mu(i)$	Avg. No of mes. sent by user i per round, $\mu(i) \equiv \mathrm{E}\left\{X_i^r\right\}$.
$\mathbf{M}$	Diagonal matrix $\mathbf{M} \doteq \mathrm{diag}\{[\mu(1), \cdots, \mu(N)]\}$.
ξ_i	Average estimation error on i's sending profile.
ξ_T	Total average estimation error of the LSDA attacker.

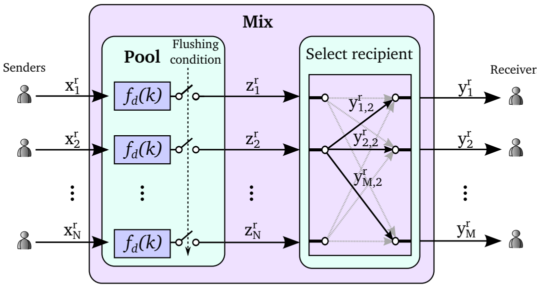


Fig. 1: System model during the communication *round* r. A global passive adversary is only able to see the messages arriving and leaving the mix (i.e., $x_1^r, x_2^r, \cdots, x_N^r$ and $y_1^r, y_2^r, \cdots, y_M^r$) but is not aware of what happens inside of it.

matrix).

B. Privacy Metrics

We measure the privacy of the users in our system as the attacker's estimation error. As we have mentioned in the introduction, we are interested in *profiling attacks*, i.e., attacks that aim at estimating the average sending behavior of the users in the long term, represented by the sending profiles $\mathbf{q}_i$, rather than attacks that aim at de-anonymizing each particular message. The suitability of the estimation error as a privacy metric is thoroughly discussed in [14], but the intuition is simple: a larger estimation error means that the adversary's estimation of the sending profiles $\hat{\mathbf{q}}_i$ is further from the real ones $\mathbf{q}_i$, and therefore users enjoy a better protection. The long-term disclosure attacks proposed in the literature that are applicable to the general scenario we have presented are the attacks belonging to the so-called Statistical Disclosure Attack (SDA) family [10], [11], [15], [16], the Perfect Matching Disclosure Attack (PMDA) [9] and the Bayesian inference attack (Vida) [8]. We do not consider other attacks such as the Disclosure Attack [17] or the Hitting Set Disclosure Attack [18], since they estimate the exact set of contacts of each sender instead of the intensity of the communications of such sender with each of those contacts. We also leave the Two-Sided SDA [19] out of our study, since it is only applicable under some assumptions on how users reply to messages.

The SDA family is a set of efficient profiling attacks that work by solving a *linear problem* that is built using the observations. PMDA and Vida work by finding *matchings* in the system, i.e., studying the possible correspondences between all messages entering and leaving the mix. PMDA is based on looking for the most probable matching, while Vida iterates by sampling matchings given the observations. In this sense, these two attacks follow a message-based approach, which they then use to estimate the sending profiles. From all these attacks, only some members of the SDA family have been applied to pool mixes. In principle, we could think of extending PMDA and Vida to work in pool mixes. However, finding matchings in a pool mix requires processing the whole trace *at once*, since the pool introduces dependencies between rounds. This renders PMDA and Vida computationally prohibitive against pool mixes. We therefore limit our choice to the attacks of the SDA family. From this family, the Least Squares Disclosure Attack (LSDA) has been proven to outperform all its relatives [15], so we use the performance of LSDA as our metric for anonymity. We note that, even though it outperforms any known feasible attack, LSDA is not necessarily the optimal attack against pool mixes and better non-linear attacks may appear in the future. Nevertheless, this is the first work to study the optimal delay characteristic of the pool mix against profiling attacks and, hence, our results shall serve as baseline for future proposals.

1) Description of LSDA: The Least Squares Disclosure Attack [16], [20] takes the count of messages that arrive to the mix from each sender and that leave the mix to each receiver in each round, and employs a least-squares algorithm to estimate the sending profiles $\mathbf{q}_i$ of each user i. The LSDA algorithm for the pool mix can be explained in two steps. First, the attacker estimates the number of messages from each sender that leave the pool in each round given the input messages she observes (i.e., $\mathbf{x}_i$) and the delay characteristic (i.e., $\mathbf{d}$), following the equation

$$\hat{z}_i^r \doteq \mathrm{E}\left\{Z_i^r | \mathbf{X}_i = \mathbf{x}_i\right\} = \sum_{k=1}^{r} x_i^k \cdot d_{r-k} \,. \qquad (1)$$

Here, we are assuming that, by the time the adversary starts observing the system, there are no messages in the pool. This assumption is reasonable, since the effect of the initial number of messages in the pool decreases rapidly as the attacker observes more communication rounds [16]. Using the convolution matrix

$$\mathbf{D} \doteq \begin{bmatrix} d_0 & 0 & 0 & \cdots & 0 \\ d_1 & d_0 & 0 & \cdots & 0 \\ d_2 & d_1 & d_0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{\rho-1} & d_{\rho-2} & d_{\rho-3} & \cdots & d_0 \end{bmatrix}, \tag{2}$$

the operation in (1) can be written in matricial form as $\hat{\mathbf{Z}} = \mathbf{DX}$. With this estimation of $\mathbf{Z}$, the sending probabilities can be inferred by solving

$$\hat{\mathbf{P}} = (\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\mathbf{Y}\,. \tag{3}$$

2) Performance of LSDA: We measure the performance of LSDA as the Mean Squared Error (MSE) of the estimator. We define the *average estimation error per sending profile* as

$$\xi_i \doteq \mathrm{E}\left\{||\hat{\mathbf{q}}_i - \mathbf{q}_i||^2\right\}\,. \tag{4}$$

This quantity represents how far the adversary's estimation of the sending profile is from the real profile, on average. The further the adversary is from the real user profile, the more privacy the user enjoys.

We also define a global measure of the privacy of the system by combining the individual errors ξ_i. In order to produce a fair combination of the individual MSE's, we first note that the product $\rho \cdot \mu(i) \cdot \hat{p}_{j,i}$, where $\mu(i)$ is the average number of messages sent by user i per round, can be seen as an estimation of the number of messages user i sends to j during the ρ observed rounds. The MSE of this estimation can then be written as $\rho^2\mu(i)^2\mathrm{E}\left\{(\hat{p}_{j,i} - p_{j,i})^2\right\}$. Now, adding along i and j we obtain the *total* MSE of the estimated number of messages each sender sends to each receiver. Normalizing this quantity to make it comparable to (4), and using $\xi_i = \sum_{j=1}^{M}\mathrm{E}\left\{(\hat{p}_{j,i} - p_{j,i})^2\right\}$, we obtain the *total average estimation error*:

$$\xi_T \doteq \sum_{i=1}^{N} \frac{\mu(i)^2}{\sum_{k=1}^{N}\mu(k)^2} \cdot \xi_i\,. \tag{5}$$

This parameter is an global metric of the level of protection of all the users against the LSDA attacker. We will use this metric to assess the performance of a pool mix with a given delay characteristic.

This metric can be expressed in a more convenient way by using the error matrix $\mathbf{E} \doteq \hat{\mathbf{P}} - \mathbf{P}$. We build the MSE matrix $\mathbf{C}_e \doteq \mathrm{E}\left\{\mathbf{E}\mathbf{E}^T\right\}$ and use the fact that the diagonal entries of this matrix correspond to ξ_i for $i = 1, \cdots, N$ to rewrite (5) as

$$\xi_T \doteq \mathrm{Tr}\left\{\mathbf{M}\mathbf{C}_e\mathbf{M}\right\}/\mathrm{Tr}\left\{\mathbf{M}^2\right\}\,, \tag{6}$$

where $\mathbf{M} \doteq \mathrm{diag}\left\{[\mu(1), \cdots, \mu(N)]\right\}$.

C. Real Datasets

In this work, we use real datasets to validate our theoretical study of pool mixes and to assess empirically the performance of the pool mix designs. Each dataset consists of a collection of messages exchanged in a communications system, from which we know the sending time, the sender, and the recipient. In order to work with them, we perform the following preprocessing steps:

1) We select the flushing condition of our mix, i.e., the condition that triggers the end of a round, from the two we contemplate. We consider *threshold pool mixes*, in which the end of the round is determined by the arrival of t messages to the system, and *timed pool mixes*, that wait τ units of time before triggering the end of the round. We choose values of t and τ that provide a reasonable anonymity/delay trade-off [21]: we pick $t = 100$ in the threshold pool mix in all datasets, and a value of τ in the timed pool mix that ensures that approximately 100 messages are mixed each round, but also guaranteeing that a round does not last more than 24 hours.
2) We fit our user behavioral model to the information in the datasets. The full list of parameters we use to model the sending behavior of the users and how we compute them from the datasets is explained in Section III-A.
3) We simulate the mixing process as explained in Section II-A, generating the observations that would be available to the adversary: $\mathbf{X}$ and $\mathbf{Y}$.

The three datasets we use, along with the values of time τ we use for the timed mix in each case, are the following:

- **Email:** this dataset contains about 220 000 emails sent by the employees of the Enron company.[1] We treat each of the 294 email addresses sending emails as the senders of our system, and consider that messages with multiple recipients are different messages sent simultaneously to each recipient. The aim of the anonymous communication system is to hide who sends emails to whom. We use a value of $\tau = 12$ hours for the timed mix in this dataset.
- **Location:** this dataset is a collection of around 400 000 location check-ins which were carried out by the 500 most active users of the Gowalla social network.[2] Each check-in can be seen as a message sent by the sender to the location the user is checking-in, and the aim of the anonymous communication system is to hide who checks-in where. The timed mix operates with $\tau = 1$ hour.
- **MailingList:** this dataset contains almost 180 000 posts to the public mailing lists of Indymedia[3] made by the 500 most active posters. The anonymous communication system is used to hide which user posts to which thread. We use $\tau = 24$ hours.

By combining the 3 real datasets and the 2 types of flushing conditions, we get 6 sets of observations, which we use in Sects. III and V.

III. Theoretical Study of Pool Mix-based Systems

In this section we set the theoretical grounds that we later use to improve the design of the pool mix. We start by proposing a behavioral model for the users of the mix, and then use this model to develop a formula that establishes a relation between the delay characteristic of the mix, along with

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

[2] http://snap.stanford.edu/data/loc-gowalla.html

[3] http://lists.indymedia.org/

the statistics of the input and output processes, and the privacy of the system.

A. Behavioral Model

We aim at proposing a statistical model that characterizes real user behavior with respect to

(a) How and when users send messages, which is determined by the random process that models the number of input messages sent by each user i in each round r, i.e., $\{X_i^r\}$.
(b) How senders choose the recipients of their messages, which is characterized by the random process that models the number of messages at each output j in round r given all the inputs, i.e., $\{Y_j^r|\mathbf{X}\}$.

1) Input process: For the first of these problems, we assume that the input processes $\{X_i^r\}$ for $i = 1, \ldots, N$ are stationary and ergodic, i.e., their statistical moments do not change with the rounds r, and we can compute these moments from a sufficiently large realization of the process. We do not assume that the input processes follow any specific probability distribution, which allows us to obtain distribution-independent results. We assume stationarity and ergodicity in order to be able to carry out our theoretical analysis afterwards. Nevertheless, as we will see in the next section, it is enough to assume that these properties hold up to fourth order moments since these are the moments we handle. We note that, although these assumptions limit the applicability of our results, we are able to obtain accurate results for the real data we use in this paper and, hence, we consider these assumptions reasonable for a range of realistic scenarios as the ones we study.

2) Output process given the inputs: The problem with $\{Y_j^r|\mathbf{X}\}$ is different, as we need to have expressions for $\mathrm{E}\left\{Y_j^r|\mathbf{X}\right\}$ and $\mathrm{Cov}\left\{Y_j^r, Y_j^s|\mathbf{X}\right\}$ relating the inputs and the outputs to perform the analysis. We therefore need a model that assigns the input messages to the outputs.

We propose a model that considers that the messages sent by the users in each round belong to one of two types of conversations: sporadic conversations and dedicated ones. The messages that belong to sporadic conversations are sent to a recipient chosen independently for each message. The messages that belong to a dedicated conversation are all sent to the same recipient, and this recipient may be the same across several rounds. With this model, we accommodate different sending behaviors that were considered in the literature. The independent choice of recipient, which is an appropriate model in those communication scenarios where users contact multiple receivers at once or just hold sporadic communications with different users (e.g., *Email* dataset), has been assumed in most of the previous works [8], [9], [11], [15], [16], [20]. On the other hand, the model that considers dedicated conversations, more appropriate in systems where users hold long conversations with a single receiver before switching to another one (e.g., *Location* and *MailingList* datasets), was only used in [21], although the authors of that work did not consider that users focused on a certain recipient are more likely to keep sending messages to that same recipient in consecutive rounds. We now describe into detail how our model works.

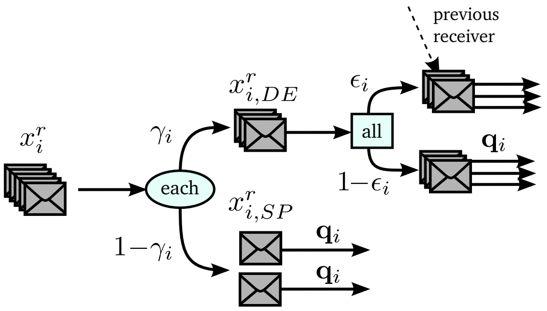


Fig. 2: Representation of how the receivers are assigned to the messages sent by user i in round r in the proposed behavioral model.

Model description: there are three parameters that model the sending behavior of each user i: the sending profile $\mathbf{q}_i$, which was defined before, the *focus* γ_i and the *persistence* ϵ_i. Each round r, the number of messages each user i sends, x_i^r, is assigned independently to the dedicated conversation group $x_{i,DE}^r$ with probability γ_i, and to the sporadic conversation group $x_{i,SP}^r$ otherwise. Then, all the messages in $x_{i,DE}^r$ are assigned a single recipient: this recipient is the same as the one chosen for the messages in the previous round (i.e., $x_{i,DE}^{r-1}$) with probability ϵ_i, and a new one following $\mathbf{q}_i$ otherwise. The recipient of each of the messages in $x_{i,SP}^r$ is chosen independently and according to the sending profile $\mathbf{q}_i$. This model is depicted in Fig. 2. Table II summarizes the new notation introduced in this section.

The rationale behind this model is the following. The focus γ_i is a probability that allows us to model users that tend to focus in a single receiver per round (γ_i close to 1), or users that are more likely to send sporadic messages to different contacts (γ_i close to 0). Intermediate values allow us to model hybrid users. The persistence ϵ_i allows us to model how likely the user is to focus on the same receiver during consecutive rounds. This value will be closer to 1, for example, for users that tend to keep long conversations with others, while it will be close to 0 for users that keep short but dedicated conversations with their recipients. This model does not account for interrelations between users, i.e., the fact that a user choosing a certain receiver affects the choice of other users' receivers (as opposed to users choosing their recipients independently of each other). Including this feature in the system would require many additional parameters (N^2), which has two disadvantages: it would substantially increase the difficulty of the privacy analysis, and obtaining these parameters given the observations would likely cause overfitting problems.

We note that, although the model does not capture scenarios where users send messages to a group of receivers (e.g., broadcast messages or dedicated conversations with multiple receivers), we obtain accurate results in presence of such traffic (e.g., results on the *Email* dataset [21]). We conjecture that these results are due to the effect of the pool, that delays messages independently and therefore group messages can be treated as sporadic messages in our analysis. In presence of more complex user sending behavior, the model should be modified by the system designer and validated following the methodology explained below.

Fitting the model to real data: We now explain how we compute the values of the parameters of our model (i.e., $\mathbf{q}_i$, γ_i and ϵ_i for all $i \in \{1, \cdots, N\}$) for each dataset and flushing condition of the mix described in Section II-C. The sending profile $\mathbf{q}_i$, defined in Section II-A, contains the probabilities $p_{j,i}$ that sender i sends a message to each receiver $j \in \{1, \cdots, M\}$. We compute these probabilities by counting the total number of messages user i sends to j and dividing between the total number of messages sent by user i.

Regarding the choice of γ_i and ϵ_i, we pick them so as to accurately fit the variance (and covariance) of the outputs given the inputs. First, we take into account the type of mix used and generate samples from the number of messages sent by sender i in each round r: x_i^r. Then, we store the number of messages from x_i^r that go to each receiver j in $\tilde{y}_{j,i}^r$ (note that this process is different from $y_{j,i}^r$ because it does not take the delaying in the pool into account). Let $\bar{\sigma}_i^l$ be the total sample output covariance with l rounds of difference, i.e.,

$$\bar{\sigma}_i^l \doteq \sum_{r=1}^{\rho-l} \sum_{j=1}^{M} (\tilde{y}_{j,i}^r - x_i^r \cdot p_{j,i})(\tilde{y}_{j,i}^{r+l} - x_i^{r+l} \cdot p_{j,i}) \,. \tag{7}$$

Likewise, let σ_i^l be the value of the output covariance given by our model, i.e.,

$$\sigma_i^l \doteq \sum_{r=1}^{\rho-l} \sum_{j=1}^{M} \text{Cov}\left\{ \tilde{Y}_{j,i}^r, \tilde{Y}_{j,i}^{r+l} | X_i^r = x_i^r, X_i^{r+l} = x_i^{r+l} \right\} \,. \tag{8}$$

This value is computed using

$$\sum_{j=1}^{M} \text{Var}\left\{ \tilde{Y}_{j,i}^r | X_i^r \right\} = \left(X_i^r + X_i^r (X_i^r - 1) \gamma_i^2 \right) v_i \,, \tag{9}$$

and

$$\sum_{j=1}^{M} \text{Cov}\left\{ \tilde{Y}_{j,i}^r, \tilde{Y}_{j,i}^{r+l} | X_i^r, X_i^{r+l} \right\} = X_i^r X_i^{r+l} \gamma_i^2 \epsilon_i^{|l|} v_i \,, \tag{10}$$

which are the theoretical expressions for the variance and covariance of our model, derived from the formulas (26) and (27) in the Appendix. Here, v_i represents the *uniformity* of the sending profile $\mathbf{q}_i$, and is defined as $v_i \doteq 1 - ||\mathbf{q}_i||^2$. The uniformity ranges from 0, when the profile contains one value equal to 1 and all the other values are 0, to $(N-1)/N$, when it is uniform, i.e., $p_{j,i} = 1/M$, $\forall j$. The first block of Table II contains a summary of the parameters that affect the variance of the outputs.

We compute γ_i for each sender i as the value that minimizes the mean squared error between the total sample variance and the variance of the model, i.e.,

$$\gamma_i = \underset{\gamma_i}{\text{argmin}} \left(\bar{\sigma}_i^0 - \sigma_i^0 \right)^2 \,. \tag{11}$$

TABLE II: Notation developed in Section III.

Symb.	Meaning
v_i	Uniformity: $v_i \doteq 1 - \lVert\mathbf{q}_i\rVert^2$.
γ_i	Focus: prob. of sending each message to the focused receiver.
ϵ_i	Persistence: prob. of keeping the focused receiver between rounds.
$x_{i,DE}^r$	Mes. from x_i^r assigned to the dedicated conv. group.
$x_{i,SP}^r$	Mes. from x_i^r assigned to the sporadic conv. group.
σ_i^l	Total output covariance for user i with l rounds of difference.
$\bar{\sigma}_i^l$	Total sample output covariance for user i with l rounds of diff.
$r_1(i)$	Combination of v_i and γ_i; $r_1(i) \doteq (1 - v_i) + \gamma_i^2 v_i$.
$r_2(i)$	Combination of v_i and γ_i; $r_2(i) \doteq \gamma_i^2 v_i$.

Similarly, we obtain the values of ϵ_i as those that minimize the error between the total sample covariance and the covariance of the model, using the γ_i obtained in (11), and considering only the covariance up to R rounds of difference, i.e.,

$$\epsilon_i = \underset{\epsilon_i}{\text{argmin}} \sum_{l=1}^{R} \left(\bar{\sigma}_i^l - \sigma_i^l \right)^2 \,. \tag{12}$$

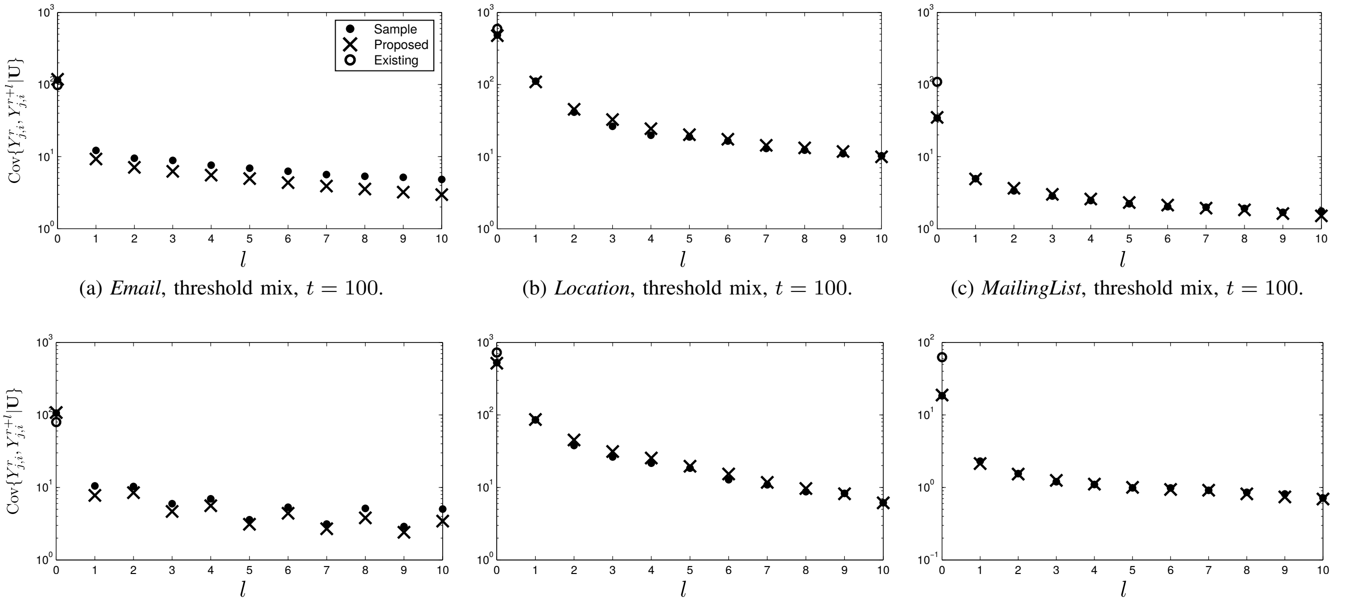
In this work, we set $R = 20$ because we have validated empirically that considering more than 20 rounds of difference does not provide extra accuracy in our analysis.

Validation of the model: Figure 3 shows how accurate this model is: we plot the sample covariance $\text{Cov}\left\{ Y_{j,i}^r, Y_{j,i}^{r+l} | \mathbf{X} \right\}$ averaged over all senders i, receivers j, and rounds r, for each of the real datasets and the different mixing scenarios described in Section II-C, for different values of the distance between rounds l. We also plot the average variance estimated given the inputs with the proposed model, as well as the variance predicted with the models in [21]. Note that, in the existing models in [21], it was assumed that $\text{Cov}\left\{ Y_{j,i}^r, Y_{j,i}^{r+l} | \mathbf{X} \right\} = 0$ for $l \neq 0$, and therefore we can only observe this value for $l = 0$ in the logarithmic plot. In all the figures, the covariance decreases as we consider rounds that are more separated. In Fig. 3d the covariance also oscillates. This is because the activity of the users in *Email* dataset presents a strong dependency on the time of the day (note that in this case the duration of the round is $\tau = 12$ hours, so the periodicity in the figure makes sense). The results of this figure confirm that, with the sending profile $\mathbf{q}_i$ and only two additional parameters per user (γ_i and ϵ_i), our model does not only outperform the prediction of existing models for $l = 0$, but it is also able to predict the real covariance accurately for multiple values of l.

B. Privacy Analysis

We aim at assessing the privacy of the system based on the behavioral model we have introduced. Our goal is to obtain an expression for the MSE matrix $\mathbf{C}_e$, since this can then be used to compute the privacy metrics presented in Section II-B (the average estimation errors per sending profile, ξ_i, are the diagonal elements of $\mathbf{C}_e$, and the total average estimation error, ξ_T, can be computed performing (6)).

We start by showing that the LSDA estimator in (3) is unbiased. In the Appendix A, equation (31), we show that



(a) *Email*, threshold mix, $t = 100$.

(b) *Location*, threshold mix, $t = 100$.

(c) *MailingList*, threshold mix, $t = 100$.

(d) *Email*, timed mix, $\tau = 12h$.

(e) *Location*, timed mix, $\tau = 1h$.

(f) *MailingList*, timed mix, $\tau = 24h$.

Fig. 3: Average output covariance Cov $\left\{Y^r_{j,i}, Y^{r+l}_{j,i}|\mathbf{X}\right\}$ for each of the datasets as a function of l, obtained with the real data (●), predicted by the proposed model (×), and predicted by the existing models (○). The covariance for values $l \neq 0$ in the existing models [21] is 0, and therefore it is only observable when $l = 0$.

$\mathrm{E}\left\{\mathbf{Y}|\mathbf{X}\right\} = \hat{\mathbf{Z}} \cdot \mathbf{P}$, which allows us to write

$$\begin{aligned}\mathrm{E}\left\{\hat{\mathbf{P}}\right\} &= \mathrm{E}\left\{(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\mathrm{E}\left\{\mathbf{Y}|\mathbf{X}\right\}\right\}\\ &= \mathrm{E}\left\{(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\hat{\mathbf{Z}}\cdot\mathbf{P}\right\} = \mathbf{P}\,. \end{aligned}\qquad (13)$$

Therefore, using the law of total covariance we can write $\mathbf{C}_e$ as

$$\begin{aligned}\mathbf{C}_e \doteq \mathrm{E}\left\{\mathbf{E}\mathbf{E}^T\right\} &= \mathrm{E}\left\{(\hat{\mathbf{P}}-\mathbf{P})(\hat{\mathbf{P}}-\mathbf{P})^T\right\}\\ &= \mathrm{E}\left\{(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\hat{\mathbf{Z}}^T\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}\hat{\mathbf{Z}}(\hat{\mathbf{Z}}^T\hat{\mathbf{Z}})^{-1}\right\}\,, \end{aligned}\qquad (14)$$

where $\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}$ is a $\rho \times \rho$ matrix whose (r, s)-th entry is $\sum_{j=1}^{M} \mathrm{Cov}\left\{Y^r_j, Y^s_j|\mathbf{X}\right\}$. We now simplify the computation of $\mathbf{C}_e$ by considering that the adversary observes the system for a *sufficiently large* amount of rounds. We note that matrices $\hat{\mathbf{Z}}^T\hat{\mathbf{Z}}/\rho$ and $\hat{\mathbf{Z}}^T\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}\hat{\mathbf{Z}}/\rho$ contain sample averages of up to fourth order moments of the input processes. Since we are assuming that these processes are ergodic and that ρ is sufficiently large, we can approximate those matrices by their expected values. Although we could write these expected values as an expression independent from ρ, in order to reduce the notational complexity of our analysis we find it convenient to define them as $\mathbf{R}_{xx} \doteq \mathrm{E}\{\hat{\mathbf{Z}}^T\hat{\mathbf{Z}}\}/\rho$ and $\mathbf{R}_{xyx} \doteq \mathrm{E}\{\hat{\mathbf{Z}}^T\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}\hat{\mathbf{Z}}\}/\rho$, and write

$$\mathbf{C}_e \approx \frac{1}{\rho}\mathbf{R}_{xx}^{-1}\mathbf{R}_{xyx}\mathbf{R}_{xx}^{-1}\,. \qquad (15)$$

Matrix $\mathbf{R}_{xx}$ depends only on the input process ($\mathbf{X}$) and the delay characteristic (given by $\mathbf{D}$), and can be written as

$$\mathbf{R}_{xx} = \frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\mathbf{D}\mathbf{X}\right\}\,. \qquad (16)$$

Matrix $\mathbf{R}_{xyx}$ also depends on the relations between the inputs and the outputs, represented by the covariance matrix $\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}}$. A closed-form expression of this latter matrix can be found in (32) in Appendix A. Plugging this formula into the definition of $\mathbf{R}_{xyx}$ above allows us to write

$$\begin{aligned}\mathbf{R}_{xyx} = &\frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\cdot\mathrm{diag}\left\{\mathbf{D}\mathbf{X}\cdot\mathbf{1}_N\right\}\cdot\mathbf{D}\mathbf{X}\right\}\\ &-\frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\mathbf{D}\cdot\mathrm{diag}\left\{\mathbf{X}\cdot\mathbf{r}_1\right\}\cdot\mathbf{D}^T\mathbf{D}\mathbf{X}\right\}\\ &+\frac{1}{\rho}\mathrm{E}\left\{\mathbf{X}^T\mathbf{D}^T\mathbf{D}\left[\sum_{i=1}^{N}\left(\mathbf{X}_i\mathbf{X}_i^T\circ\boldsymbol{E}_i\right)r_2(i)\right]\mathbf{D}^T\mathbf{D}\mathbf{X}\right\}\,. \end{aligned}\qquad (17)$$

For readability, we have grouped the effects of v_i and γ_i in the functions $r_1(i) \doteq (1 - v_i) + \gamma_i^2 v_i$ and $r_2(i) \doteq \gamma_i^2 v_i$. We also use $\mathbf{r}_1 \doteq [r_1(1), \cdots, r_1(N)]^T$. For users that send messages independently to their contacts (i.e., $\gamma_i = 0$), $r_1(i) = 1 - v_i$ and $r_2(i) = 0$. In contrast, users that always focus on a certain receiver (i.e., $\gamma_i = 1$) get $r_1(i) = 1$ and $r_2(i) = v_i$. Note that if $\gamma_i = 0$ for a certain user i, then $r_2(i) = 0$ and the contribution of that user to the last summand in (17) is zero. In that case, we can compute $\mathbf{R}_{xx}$ and $\mathbf{R}_{xyx}$ with only the first, second and third order moments of the input process of that user. However, in most scenarios this will not be the case, and we would also need the fourth order moments to compute the last summand of (17). Note that, although we have assumed strong stationarity and ergodicity, it is enough for our analysis to assume stationarity and ergodicity up to order four, since these are the largest order moments we handle.

We can compute our error metrics ξ_i and ξ_T to assess the privacy of the users by plugging (16) and (17) into (15). The complexity of this formula is considerable, and simplifying it yields much less accurate results. Fortunately, when our goal is to solve the problem of finding the delay characteristic that maximizes ξ_T, we can find an alternative objective function relating ξ_T and $\mathbf{D}$ that is more amenable to analysis and yields a solution close to the optimal one.

C. Evaluation

We evaluate the performance of our formula in a binomial pool mix scenario [6]. The delay characteristic of this mix follows a geometric distribution $d_k = \alpha(1-\alpha)^k$, where α is the probability that a message stored in the pool leaves in each round.

Figure 4 represents the overall error (6) predicted by our formula, together with the real error of the attack. We also plot the most accurate expressions found in the literature [21] to model the adversary's error in these datasets, which we have adapted to pool mixes. We can see that our formula clearly follows the trend of the real MSE as the delay characteristic varies, while the ones in [21] are coincidentally accurate when $\alpha = 1$ (in this case, $d_0 = 1$ and $d_k = 0$ for $k > 0$, so it is equivalent to having no pool), but are not valid to predict the error for other pool mix designs ($\alpha < 1$).

IV. Optimizing the Design of Pool Mixes

In this section, we address the problem of optimizing the performance of the pool mix with respect to its delay characteristic, i.e., finding the delay characteristic $\mathbf{d}_{opt}$ that maximizes our global privacy metric. We start by setting an optimization problem whose solution is the optimal one, although its complexity makes it hard to study. In order to shed some light into how $\mathbf{d}_{opt}$ depends on the users' behavior, we set an alternative optimization problem which is much more amenable to analysis and whose solution is remarkably close to the optimal one. Using this alternative formulation of the problem, we study the optimal mix designs under different assumptions on the users' behavior, and come up with a user-independent albeit sub-optimal design, that is useful when no a priori information about the users is available.

A. Optimal Pool Mix Design

The optimal delay characteristic can be obtained by looking for the vector $\mathbf{d} \doteq [d_0, d_1, \cdots, d_{\rho-1}]^T$ that maximizes the overall protection of the users in the system, defined in (6). The problem is formally stated as

Optimal Pool Mix Design Problem:

$$\begin{aligned} \mathbf{d}_{opt} = \underset{\mathbf{d}}{\text{argmax}} \quad & \text{Tr}\left\{\mathbf{M}\mathbf{C}_e\mathbf{M}\right\} \\ \text{subject to} \quad & \sum_{k=0}^{\rho-1} d_k = 1, \quad d_k \geq 0, \ \forall k \\ & \sum_{k=1}^{\rho-1} k \cdot d_k \leq \bar{\delta} \end{aligned} \tag{18}$$

We have disregarded the normalization by $\text{Tr}\left\{\mathbf{M}^2\right\}$ in (6), since this normalization does not affect the maximum of the function with respect to $\mathbf{d}$. The first constraint ensures that the delay characteristic obtained constitutes a valid probability mass function, and the second one is a constraint on the maximum average delay in rounds that the messages suffer inside the pool, where $\bar{\delta}$ denotes this maximum average delay. This formulation can also be accommodated to obtain the optimal delay function given different constraints, for example, a different bound on the maximum delay in rounds tolerated for the messages (i.e., L_{max} such that $d_k = 0$ for $k > L_{max}$).

Solving the problem in (18) is not straightforward: we need to know the values of a huge amount of input moments (or make assumptions on them) and all the parameters that model the sending behavior of the users, namely $\mathbf{q}_i$, γ_i and ϵ_i for $i = 1, \ldots, N$. It is also very hard to get an intuitive idea of how the shape of the optimal delay characteristic $\mathbf{d}_{opt}$ relates to these parameters. Motivated by this, in the next section we look for an alternative formulation of this problem that is more amenable to analysis.

B. Alternative Formulation of the Optimal Pool Mix Design: Quasi-Optimal Pool Mix

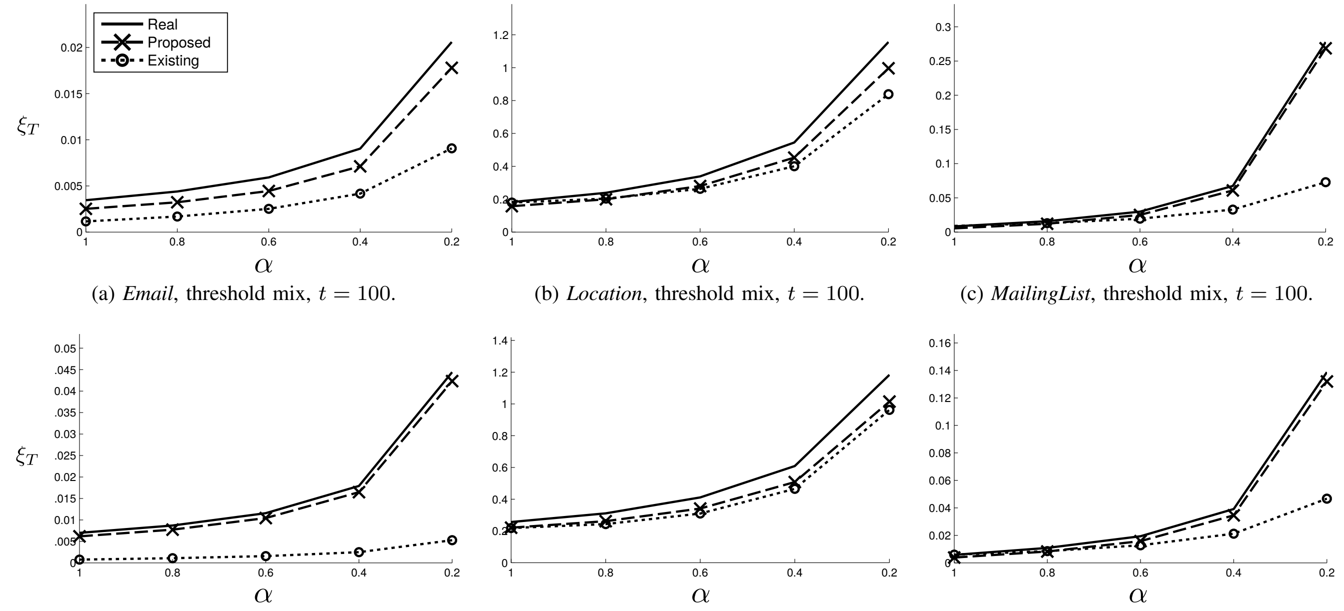
In [22], we show that when the number of users in the system N is comparable to ρ as $\rho \to \infty$, the strategy followed to maximize $\text{Tr}\left\{\mathbf{M}\mathbf{C}_e\mathbf{M}\right\}$ and $\text{Tr}\left\{\mathbf{M}\mathbf{R}_{xx}^{-1}\mathbf{M}\right\}$ is the same, and therefore the delay characteristics that maximize each of these functions are similar. In that case, (18) can be formulated as

Quasi-optimal Pool Mix Design Problem:

$$\begin{aligned} \mathbf{d}'_{opt} = \underset{\mathbf{d}}{\text{argmax}} \quad & \text{Tr}\left\{\mathbf{M}\mathbf{R}_{xx}^{-1}\mathbf{M}\right\} \\ \text{subject to} \quad & \sum_{k=0}^{\rho-1} d_k = 1, \quad d_k \geq 0, \ \forall k \\ & \sum_{k=1}^{\rho-1} k \cdot d_k \leq \bar{\delta} \end{aligned} \tag{19}$$

Analyzing this problem is much easier than (18), as it depends on less parameters: note that we only need to consider up to second order moments of the input, and that the dependence on v_i, γ_i and ϵ_i is gone. These user parameters still affect the MSE, but they do so via terms that become independent of the delay characteristic when $N \to \infty$ is comparable to ρ. Interestingly, the solutions of (18) and (19) are very close in our real datasets, as we empirically show in Section V, which indicates that we are in the case of N being comparable to ρ as $\rho \to \infty$ in all the scenarios for which we have data. We remark that for other scenarios where $N \ll \rho$, the system designer will have to rely on (18) to choose the delay characteristic of the mix.

In order to provide more insight into the shape of the optimal delay characteristic when N and ρ are not comparable, we now study the solution of (19) under different assumptions, when $\rho \to \infty$ and $N \ll \rho$. In order to do that, we first consider that $\mathbf{R}_{xx} \approx \mathbf{\Sigma}_{xx}$ (c.f. [21]), where $\mathbf{\Sigma}_{xx}$ is the covariance matrix of the input processes $\{\hat{X}^r_{d,i}\}$, i.e., if $\mathbf{X}_c \doteq \mathbf{X} - \mathbf{1}_\rho \boldsymbol{\mu}^T$, then $\mathbf{\Sigma}_{xx} \doteq \text{E}\left\{\mathbf{X}_c^T \mathbf{D}^T \mathbf{D} \mathbf{X}_c\right\}/\rho$. It will be helpful to define additional notation: the variance of the input processes is



(a) *Email*, threshold mix, $t = 100$.

(b) *Location*, threshold mix, $t = 100$.

(c) *MailingList*, threshold mix, $t = 100$.

(d) *Email*, timed mix, $\tau = 12h$.

(e) *Location*, timed mix, $\tau = 1h$.

(f) *MailingList*, timed mix, $\tau = 24h$.

Fig. 4: Overall MSE of LSDA in different realistic scenarios, as a function of the firing probability of the binomial pool mix (α), compared with the theoretical MSE predicted by our formula and the existing ones [21].

denoted by $\mu_2(i) \doteq \text{Var}\{X_i^r\}$. With the variance of all users, we build $\mathbf{M}_2 \doteq \text{diag}\{[\mu_2(1), \cdots, \mu_2(N)]\}$. We define the autocorrelation of the delay characteristic of the mix at lag l as $R_{dd}[l] \doteq \sum_{r=l}^{\rho-1} d_r d_{r-l}$ for $l \geq 0$, and $R_{dd}[l] = R_{dd}[-l]$ otherwise. Note that matrix $\mathbf{D}^T\mathbf{D}$ is $\rho \times \rho$ Toeplitz whose r, s-th entry is $R_{dd}[r-s]$. Based on this, we can decompose $\mathbf{\Sigma}_{xx}$ as

$$\mathbf{\Sigma}_{xx} \doteq \frac{1}{\rho}\text{E}\left\{\mathbf{X}_c^T\mathbf{D}^T\mathbf{D}\mathbf{X}_c\right\} = \sum_{l=-\rho+1}^{\rho-1} \mathbf{C}_2[l] \cdot R_{dd}[l] \,, \quad (20)$$

where $\mathbf{C}_2[l]$ is an $N \times N$ matrix containing the covariances between all the input processes with lag l, i.e., the m, n-th entry of $\mathbf{C}_2[l]$ is $\text{Cov}\left\{X_m^r, X_n^{r+l}\right\}$.

We start by assuming that the input processes are independent white processes. We then analyze how auto-correlations and cross-correlations in the input process affect the design of the optimal delay characteristic, and provide some insights into what shape this function takes when we cannot make any assumptions on the input processes.

1) White input processes: We start by analyzing the simple scenario where the input processes $\{X_i^r\}$ are uncorrelated and white. In that case, we have $\mathbf{C}_2[l] = \mathbf{0}_{N\times N}$ for $l \neq 0$ and $\mathbf{C}_2[0] = \mathbf{M}_2$. By using the expansion in (20), we get that $\mathbf{\Sigma}_{xx} = \mathbf{M}_2 \cdot R_{dd}[0]$, and therefore the optimization problem (19) becomes that of looking for the $\mathbf{d}$ that minimizes $R_{dd}[0]$ subject to the constraints.

This problem can be solved using the method of Lagrange multipliers. Assume that L is the largest index such that $d_k = 0$ when $k > L$. We use the fact that $d_k \geq d_{k+1}$ (otherwise, there would be another vector $\mathbf{d}$ that obtains the same value of $R_{dd}[0]$ for less average delay), and that $d_k \geq 0$ to find that $d_k = \lambda_1 - \lambda_2 \cdot k$ for $k \in \{0, \cdots, L\}$, with $\lambda_1, \lambda_2 > 0$ and $d_k = 0$ for $k > L$. This means that the values of our solution $\mathbf{d}'_{opt}$ are points of a straight line with negative slope. We then use these equations together with the constraints to find that the solution to this problem is the following:

a) Given an average delay in rounds $\bar{\delta}$, pick $L = \lceil 3\bar{\delta} \rceil$.
b) Then, set

$$d_k = \frac{2}{L+1}\left(\frac{L+1+(L-3\bar{\delta})-k}{L+2}\right) \quad (21)$$

for $k = 0, \cdots, L$. All the other d_k for $k > L$ are set to 0.

We refer to the pool mix implementing this delay characteristic as the *ramp* pool mix, due to the shape of the delay characteristic obtained, which we denote by $\mathbf{d}_{rmp}$. It is interesting to note that, when the inputs are white, the optimal delay function in the sense of maximizing the global MSE is user-independent as it does not depend on the input moments or the sending behavior of the users. Therefore, this design is very useful when there is no a priori information about the users.

2) Linear model for auto-correlations: We now assume that we can write the matrix $\mathbf{X}$ we observe as $\mathbf{X} = \mathbf{G}\tilde{\mathbf{X}}$, where $\tilde{\mathbf{X}}$ is a matrix containing uncorrelated white processes (as in the previous case) and $\mathbf{G}$ is a convolution matrix with the same structure as $\mathbf{D}$, containing in its first column the taps of the FIR filter $\mathbf{g} \doteq [g_0, g_1 \cdots, g_{\rho-1}]^T$. This filter introduces auto-correlations in the inputs processes of the users. It is straightforward to show that, in that case,

$$\mathbf{\Sigma}_{xx} = \frac{1}{\rho}\text{E}\left\{\tilde{\mathbf{X}}_c^T\mathbf{G}^T\mathbf{D}^T\mathbf{D}\mathbf{G}\tilde{\mathbf{X}}_c\right\} = \mathbf{M}_2 \cdot (R_{dd}[l] * R_{gg}[l])|_{l=0} \,, \quad (22)$$

where $*$ denotes the convolution operation. Therefore, in this case, the optimal delay characteristic is the one that, given the constraints, minimizes $(R_{dd}[l] * R_{gg}[l])|_{l=0}$. We can compare this with the previous scenario by looking at the frequency domain. Let $\Lambda_{dd}[k]$ and $\Lambda_{gg}[k]$ be the coefficients of the ρ-point DFT of d_k and g_k, respectively. Assuming that $\mathbf{D}$ and $\mathbf{G}$ are circulant (the border effects can be disregarded as ρ grows), the optimal delay function $\mathbf{d}$ is the one that minimizes

$$(R_{dd}[l] * R_{gg}[l])|_{l=0} \approx \frac{1}{\rho}\sum_{k=0}^{\rho-1} |\Lambda_{dd}[k]|^2 \cdot |\Lambda_{gg}[k]|^2 \,. \qquad (23)$$

We could have solved the previous case (white inputs) following this frequency analysis, obtaining that the optimal delay characteristic in that case is the one that minimizes $\sum_{k=0}^{\rho-1} |\Lambda_{dd}[k]|^2$ given some delay and normalization constraints. Now, we have a specific $\Lambda_{gg}[k]$ that depends on the filter taps g_k that “colors” the input processes. The spectrum of the optimal delay characteristic $|\Lambda_{dd}[k]|^2$ will take smaller values in those frequency bins where $|\Lambda_{gg}[k]|^2$ is larger, and larger values in those bins where $|\Lambda_{gg}[k]|^2$ is smaller. In that sense, we can see the effect of $\mathbf{g}$ as an additional constraint in the problem, that causes $\mathbf{d}$ to somehow “whiten” the input processes, while satisfying the constraints of the problem.

In this example, we have assumed that the autocorrelation of all the input processes is the same, given by the filter $\mathbf{g}$. If we have different autocorrelations per user (i.e., individual filters $\mathbf{g}(i)$ for $i = 1, \cdots, N$), formulating the problem in the same way we can see that the optimal solution consists on designing a particular delay characteristic for each user, based on the same idea above.

3) Linear model for cross-correlations: Similar to the previous scenario, we now assume that there is an $N \times N$ matrix $\mathbf{S}$ that generates our observation $\mathbf{X}$ by making linear combinations of N uncorrelated white processes in $\tilde{\mathbf{X}}$, i.e., $\mathbf{X} = \tilde{\mathbf{X}}\mathbf{S}$. The processes in $\mathbf{X}$ are now white and correlated processes. We assume that matrix $\mathbf{S}$ is non-singular, otherwise LSDA could not be applied directly and we would have to work in a subspace where the solution is possible. In that case,

$$\mathbf{\Sigma}_{xx} = \frac{1}{\rho}\S^T \cdot \mathrm{E}\left\{\tilde{\mathbf{X}}_c^T\mathbf{D}^T\mathbf{D}\tilde{\mathbf{X}}_c\right\} \cdot \S = \S^T\mathbf{M}_2\S \cdot R_{dd}[0]\,, \quad (24)$$

and therefore the solution is again the ramp pool mix obtained in (21).

4) Generic input processes: When the observed matrix $\mathbf{X}$ cannot be written as a combination of the examples above, i.e., $\mathbf{X} = \mathbf{G}\tilde{\mathbf{X}}\mathbf{S}$, then, besides $R_{dd}[0]$, other autocorrelation terms $R_{dd}[k]$ can take part in the optimization problem. A toy example for this is the case where we have $N = 2$ white users, and user $i = 2$ always sends the same number of messages user $i = 1$ has sent in the previous round, i.e., $X_2^r = X_1^{r-1}$. This can represent, for example, a user that always replies to each message she receives in the next round, or a repeater. In this case, $\mathbf{\Sigma}_{xx} = \mathbf{I}_{2\times2} \cdot R_{dd}[0] \cdot \mu_2(1) + (\mathbf{1}_{2\times2} - \mathbf{I}_{2\times2}) \cdot R_{dd}[1] \cdot \mu_2(1)$, and we obtain that the optimal delay function is the one that minimizes $R_{dd}[0] - R_{dd}[1]$ subject to the constraints. This results in a bell-shaped delay characteristic, which is far from the straight line we obtain for the cases 1 and 3 studied before.

For a generic input process, we cannot find a closed-form solution for the delay characteristic. We can only expect to find a delay function more similar to a straight line when the input correlations are small, and a bell-shaped function when the correlations between the processes are large, or even when they are small but the number of users is large.

V. Evaluation

In this section, we evaluate the performance of the different delay characteristics proposed in the previous sections. We build the following pool mixes, that differ on their delay characteristic:

1) The optimal pool mix, whose delay characteristic is given by the solution to (18), i.e., $\mathbf{d}_{opt}$.
2) The quasi-optimal pool mix, whose delay characteristic is given by the solution to (19) when no assumptions on the input processes are made, i.e., $\mathbf{d}'_{opt}$.
3) The ramp pool mix, whose delay characteristic, given by (21) and denoted by $\mathbf{d}_{rmp}$, is the solution to (19) under the assumptions that the input processes are white and uncorrelated.
4) The binomial pool mix, which has been widely used in the literature and claimed as the optimal pool mix in terms of anonymity in previous works [7], [13]. The delay characteristic of this pool is denoted by $\mathbf{d}_{bin}$ and is given by $d_k = \alpha(1-\alpha)^k$, where α is a parameter between 0 and 1 controlling the delay of the messages inside the pool.

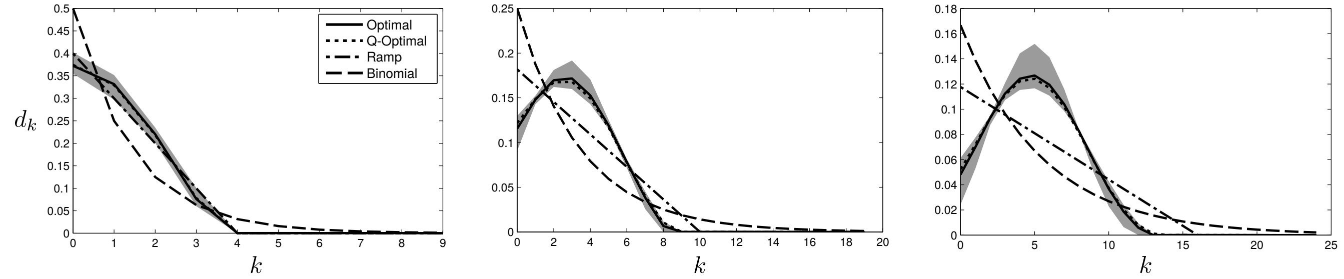
Each of these designs is assigned a flushing condition and evaluated with real data, as explained in Section II-C. All the simulations are performed using Matlab software, including the optimization tools to solve (18) and (19).

A. Shape of the delay characteristic

We first compare the shape of the delay characteristics of the four pool mix designs, for different values of the average delay in rounds $\bar{\delta}$. This is shown in Fig. 5. Since $\mathbf{d}_{opt}$ and $\mathbf{d}'_{opt}$ are different for each input dataset, we plot the *average* result in the figure. The gray area represents the maximum and minimum values obtained for each $d_k \in \mathbf{d}_{opt}$ in the datasets.

The figure confirms that the average shape of the delay characteristic of the optimal and quasi-optimal designs is very similar for all the values of average delay $\bar{\delta}$ we test, which confirms our intuitions in Section IV-B. It is also worth noticing that these delay functions are *non-decreasing* and bell-shaped:this happens because the number of users N in the real datasets we have used for evaluation is comparable to the number of rounds observed ρ, as explained in [22].

We show in Table III the variance of the delay of each design (we show the average variance over all datasets for the optimal and quasi-optimal pool mixes). Again, the optimal and quasi-optimal designs have very similar variance, as their shape is almost the same. These pool mixes do not only maximize the error but also have the smallest variance, which means that, when using them, users can expect a delay in rounds close to the average value $\bar{\delta}$ for each of their messages, while for the other types of designs the delay is less predictable. It is also worth noticing that the variance of the ramp pool mix is half the variance of the binomial one, which makes the ramp pool mix a more appealing option when no information about the users is available to the system designer.



(a) $\bar{\delta} = 1$ round.

(b) $\bar{\delta} = 3$ rounds.

(c) $\bar{\delta} = 5$ rounds.

Fig. 5: Comparison between the delay characteristic of different pool mix designs.

TABLE III: Expected variance of the delay (in number of rounds) introduced by each type of pool for different values of average delay.

$\bar{\delta}$	1	2	3	4	5
$Var\{\mathbf{d}_{bin}\}$	2.00	6.00	12.00	20.00	30.00
$Var\{\mathbf{d}_{rmp}\}$	1.00	3.00	6.00	10.00	15.00
$Var\{\mathbf{d}'_{opt}\}$	0.91	2.31	4.08	6.05	8.20
$Var\{\mathbf{d}_{opt}\}$	0.90	2.23	3.92	5.79	7.84

B. Performance of the pool mix designs

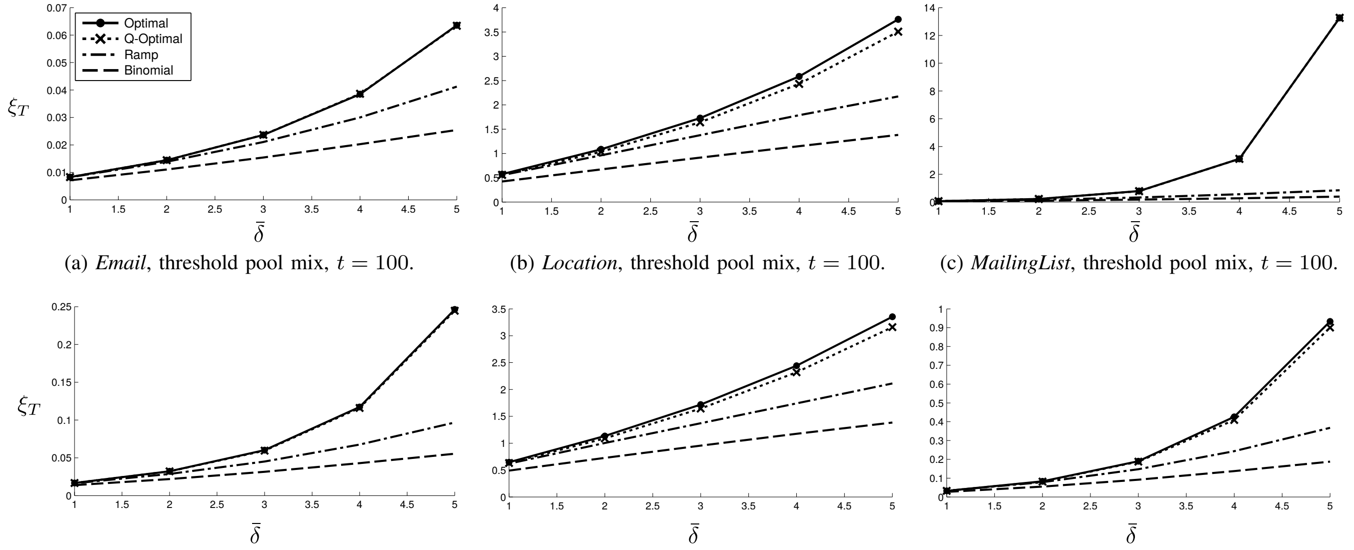
We evaluate the protection that the different pool mix designs offer against the LSDA adversary. Figure 6 shows the global MSE (ξ_T) obtained by using the different pool mix designs for different values of average delay (we have omitted the value at $\bar{\delta} = 0$, as all the pools are equivalent in that case, i.e., $d_0 = 1$). We can see that the ramp pool mix considerably improves the protection of the users in the system when compared with the traditional binomial pool mix, but the optimal and quasi-optimal designs achieve a substantially better result. The difference between these latter is small, although the optimal pool mix performs slightly better in every case. For an average delay of $\bar{\delta} = 5$ rounds, the ratios between the MSE achieved by the optimal pool mix and the MSE achieved by the binomial pool mix for each dataset in Fig. 6 are, in order, 2.5, 4.4, 2.7, 2.4, 34.3 and 5.0. Since the dependence of the MSE on the number of rounds observed is $1/\rho$, we can also interpret these numbers as ratios on the number of rounds. For example, in *MailingList* dataset using a timer with $\tau = 24h$ as flushing condition and allowing a maximum average delay of $\bar{\delta} = 5$ rounds (Fig. 6f, ratio of 5.0), users exchanging messages during a month using a binomial pool mix would get the same degree of protection against a profiling adversary than users communicating for 5 months with our optimal pool mix. If we use a threshold of $t = 100$ as flushing condition instead, the optimal design allows users to exchange messages for almost three years while having more protection than users exchanging messages for a month with a binomial pool mix. These results highlight the importance of the delay strategy in the privacy of the system: choosing a well-designed delay characteristic can make a huge difference in the performance.

VI. Comparison with Related Work

In this section, we compare our work with other attempts at finding the optimal delay characteristic for a pool mix. There are two works that have performed this analysis. On the one hand, Danezis analyzes in [13] the delay characteristic of a continuous pool mix [23], i.e., a pool mix that does not operate in batches or rounds, but applies to each input process $X_i(t)$ a random delay which can be modeled by a continuous probability density function $d(t)$. However, the experiments of this paper perform a time discretization, where the mix works in so-called "simulation tics". These simulation tics are equivalent to our communication rounds, so we can consider both scenarios equivalent and apply our analysis here. On the other hand, Rebollo-Monedero et al. [7] study threshold pool mixes that work by storing messages and forwarding k of them to their recipients when the pool contains $n \geq k$ of them. We have not considered this flushing condition in our cases of study, as we are considering that the flushing condition is independent of the current number of messages in the pool, but our framework can easily accommodate it.

The approach to measure anonymity used by both Danezis [13] and Rebollo-Monedero [7] is radically different from ours. They use information-theoretic metrics, mainly Shannon's entropy, to measure the anonymity of single messages; while we use an estimation-theoretic approach to measure the error of the adversary when profiling a user. The information-theoretic approach works as follows: for a target output message, it builds a probability distribution describing the likelihood that any input message corresponds with the target output. Anonymity is then measured as the entropy of this probability distribution: maximal entropy implies maximum anonymity since it represents the case where the output message is equally likely to have come from each input; and minimal entropy (zero) indicates minimum anonymity, i.e., that the output message can unequivocally be related to an input.

Under this anonymity definition, and using Shannon's result that states that the distribution that maximizes the entropy when there is a constraint on the average delay is the geometric distribution (exponential for the continuous case), both Danezis [13] and Rebollo-Monedero et al. [7] obtain that the binomial pool mix (called exponential pool mix in the continuous case [13]) is the optimal design, i.e., the one that maximizes anonymity.



(a) *Email*, threshold pool mix, $t = 100$.

(b) *Location*, threshold pool mix, $t = 100$.

(c) *MailingList*, threshold pool mix, $t = 100$.

(d) *Email*, timed pool mix, $\tau = 12h$.

(e) *Location*, timed pool mix, $\tau = 1h$.

(f) *MailingList*, timed pool mix, $\tau = 24h$.

Fig. 6: Performance of pool mixes in different realistic scenarios and using different flushing strategies (timed and threshold mixes), as a function of the average delay ($\bar{\delta}$). Each line represents the overall MSE of LSDA (ξ_T) using a different delay characteristic.

In order to arrive to this conclusion both Danezis and Rebollo-Monedero et al. make unrealistic assumptions. Danezis assumes that the arrival of messages follows a Poisson distribution, but it is known that in real scenarios this assumption is not fulfilled (e.g., see [21]). Rebollo-Monedero considers that the inter-arrival times have a common expectation and variance and they are uncorrelated. In this paper we have shown that not only these assumptions are not met by real traffic, but also that the user auto- and cross-correlations have great impact on the adversary's error. In fact, the optimal delay function under the Shannon's entropy criterion depends on the user behavior statistics, and it is in general different for each user and/or population.

In order to show that under real traffic conditions the optimality of the binomial mix claimed in [7], [13] does not hold, we compare its performance to the ramp pool conducting the following experiment described in [13]. We consider a scenario in which there is only one sender that sends messages to one of only two possible receivers. These receivers also get messages from other users, from whom the adversary is not able to see the inputs but knows the distribution of their messages.

The attack proposed by Danezis is based on a hypothesis test: either the observed input goes to the first receiver (H_0) or to the second (H_1). In order to decide for one of the two, Danezis computes a log-likelihood ratio $\log L_{H_0/H_1}$. Given a threshold η, the adversary decides H_0 when $\log L_{H_0/H_1} > \eta$. The choice of the threshold η depends on the number of simulation tics observed by the adversary and the desired performance: a low η would increase the probability of deciding H_0 when H_0 is true (i.e., increase the true positive rate, TPR) but it would also increase the probability of incorrectly deciding H_0 when H_1 holds (i.e., increase the false positive rate, FPR).

We have implemented this attack and simulated the experiment in Matlab.[4] For each value of threshold η, we perform 10 000 repetitions of the experiment with 1 000 simulation tics and compute the TPR and FPR for both the binomial pool mix and ramp pool mix (21) configured for the same average delay $\bar{\delta} = 30$ rounds. We plot in Figure 7 the receiver operating characteristic (ROC) curve, i.e., the TPR versus the FPR obtained, for both designs. We see that the ramp pool mix outperforms the binomial pool mix since, for any given TPR, the ramp pool mix always achieves a larger FPR, i.e., the adversary will wrongly choose H_0 when H_1 holds more often when the ramp pool is used.

The result of our experiments shows that, even though the binomial pool mix maximized the information-theoretic measure of sender anonymity introduced in [24], it is not optimal against the message tracing attack proposed in [13]. The reason is that information-theoretic metrics only consider the probability distribution of inputs for a given output message, disregarding the distribution of all the other messages. Hence, they do not reflect adequately how a given input blends with other incoming traffic, which is key against attacks aiming at tracing messages.

VII. Conclusions

In this work, we study the design of pool mixes, the basic building blocks of high-latency anonymous communication systems. We carry out such study from an estimation-theoretic point of view, deriving a theoretical model for user behavior,

[4] For a detailed description of this experiment and the parameters used, please see Section 3.2 in [13]. In order to compute the FPR, we have also simulated the H_1 scenario.

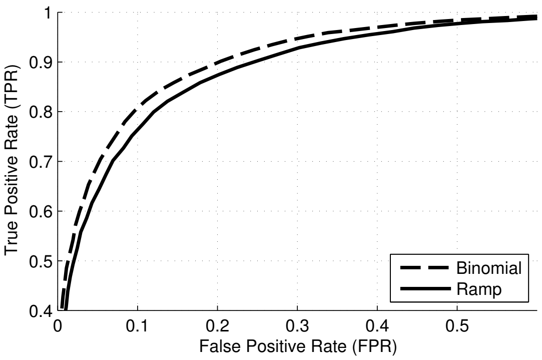


Fig. 7: Receiver operating characteristic for Danezis's classifier in [13], given 1000 simulation tics, for the binomial pool mix and our ramp pool mix.

which we validate with real data, and obtaining a mathematical expression for the estimation error of the best profiling adversary against pool mixes. We use this estimation error as a metric of privacy, and obtain the delay characteristic of the pool mix that maximizes this metric. Since computing this optimal design requires a lot of information, we also propose a quasi-optimal solution which is much easier to compute and to understand, although its application is more limited. Our work shows that the optimal pool mix design depends on the users' behavior, and therefore it is impossible to compute it when no information about the users is available. In order to solve this, we also propose the ramp pool mix, a sub-optimal but user-independent design that is useful when the number of rounds observed is much larger than the number of users in the system.

We compare the performance of our proposals and the state-of-the-art binomial pool mix against a profiling adversary, and show that our constructions substantially increase the protection provided to users. We further show that, contrary to prior belief [7], [13], the binomial pool mix is neither optimal against message-tracing attacks.

Appendix A
Derivation of the second-order moments of the output, given the inputs

Our goal is to derive expressions for the expected value and the second-order moments of the outputs $Y^r_{j,i}$ given the inputs $\mathbf{X}$. To make the derivations easier, in this section we use the random variable $\tilde{Y}^u_{j,i}$, that models the number of messages sent by sender i in round u, that are addressed to receiver j (but can reach them in another round, since they may be delayed inside the pool). Those messages enter the pool, and leave in that round or in the subsequent ones. We define $Y^{r,u}_{j,i}$ as the number of those messages leaving in round r. Note that $Y^r_{j,i} = \sum_{u=1}^{r} Y^{r,u}_{j,i}$. When there is no pool, we also have $Y^r_{j,i} = \tilde{Y}^r_{j,i}$. We also use $v_{j,i} \doteq p_{j,i}(1-p_{j,i})$ and note that $v_i = \sum_{j=1}^{M} v_{j,i}$.

We start by building the relations between $\tilde{Y}^u_{j,i}$ and the inputs. These can be easily established by looking at Fig. 2.

$$\begin{aligned}
\mathrm{E}\left\{\tilde{Y}^r_{j,i}|\mathbf{X}\right\} &= \mathrm{E}\left\{\mathrm{E}\left\{\tilde{Y}^r_{j,i}|X^r_{i,SP}, X^r_{i,DE}\right\}|X^r_i\right\} \\
&= \mathrm{E}\left\{(X^r_{i,SP} + X^r_{i,DE})\cdot p_{j,i}|X^r_i\right\} \\
&= \mathrm{E}\left\{X^r_i \cdot p_{j,i}|X^r_i\right\} = X^r_i \cdot p_{j,i}\,.
\end{aligned} \tag{25}$$

Since $\mathrm{E}\left\{\tilde{Y}^r_{j,i}|X^r_{i,SP}, X^r_{i,DE}\right\} = X^r_i \cdot p_{j,i}$, then the variance of this expected value conditioned on X^r_i is zero. Therefore,

$$\begin{aligned}
\mathrm{Var}\left\{\tilde{Y}^r_{j,i}|\mathbf{X}\right\} &= \mathrm{E}\left\{\mathrm{Var}\left\{\tilde{Y}^r_{j,i}|X^r_{i,SP}, X^r_{i,DE}\right\}|X^r_i\right\} \\
&= \mathrm{E}\left\{\left(X^r_{i,SP} + (X^r_{i,DE})^2\right)\cdot v_{j,i}|X^r_i\right\} \\
&= \left(X^r_i(1-\gamma_i) + (X^r_i\gamma_i)^2 + X^r_i\gamma_i(1-\gamma_i)\right) v_{j,i} \\
&= \left(X^r_i + X^r_i(X^r_i - 1)\gamma_i^2\right) v_{j,i}\,.
\end{aligned} \tag{26}$$

Similarly, it can be shown that

$$\begin{aligned}
\mathrm{Cov}\left\{\tilde{Y}^r_{j,i}, \tilde{Y}^{r+l}_{j,i}|\mathbf{X}\right\} &= \mathrm{E}\left\{X^r_{i,DE}X^{r+l}_{i,DE}\epsilon_i^{|l|}v_{j,i}|X^r_i, X^{r+l}_i\right\} \\
&= X^r_i X^{r+l}_i \gamma_i^2 \epsilon_i^{|l|} v_{j,i}\,.
\end{aligned} \tag{27}$$

Now, we show the relations between $Y^r_{j,i}$ and $\tilde{\mathbf{Y}}$ in the following equations, where we use that $Y^{r,u}_{j,i}|\tilde{\mathbf{Y}}$ and $Y^{r+l,t}_{j,i}|\tilde{\mathbf{Y}}$ are uncorrelated for any l when $u \neq t$:

$$\mathrm{E}\left\{Y^r_{j,i}|\tilde{\mathbf{Y}}\right\} = \sum_{u=1}^{r}\mathrm{E}\left\{Y^{r,u}_{j,i}|\tilde{Y}^u_{j,i}\right\} = \sum_{u=1}^{r}\tilde{Y}^u_{j,i}\cdot d_{r-u}\,. \tag{28}$$

$$\begin{aligned}
\mathrm{Var}\left\{Y^r_{j,i}|\tilde{\mathbf{Y}}\right\} &= \sum_{u=1}^{r}\sum_{t=1}^{r}\mathrm{Cov}\left\{Y^{r,u}_{j,i}, Y^{r,t}_{j,i}|\tilde{\mathbf{Y}}\right\} \\
&= \sum_{u=1}^{r}\mathrm{Var}\left\{Y^{r,u}_{j,i}|\tilde{\mathbf{Y}}\right\} \\
&= \sum_{u=1}^{r}\tilde{Y}^u_{j,i}\cdot d_{r-u}(1-d_{r-u})\,.
\end{aligned} \tag{29}$$

$$\begin{aligned}
\mathrm{Cov}\left\{Y^r_{j,i}, Y^s_{j,i}|\tilde{\mathbf{Y}}\right\} &= \sum_{u=1}^{r}\sum_{t=1}^{s}\mathrm{Cov}\left\{Y^{r,u}_{j,i}, Y^{s,t}_{j,i}|\tilde{\mathbf{Y}}\right\} \\
&= \sum_{u=1}^{\min(r,s)}\mathrm{Cov}\left\{Y^{r,u}_{j,i}, Y^{s,u}_{j,i}|\tilde{\mathbf{Y}}\right\} \\
&= -\sum_{u=1}^{\min(r,s)}\tilde{Y}^u_{j,i}\cdot d_{r-u}d_{s-u}\,.
\end{aligned} \tag{30}$$

We can now get the results we were looking for. Combining equations (28) and (25), we get $\mathrm{E}\left\{Y^r_{j,i}|\mathbf{X}\right\} = \sum_{u=1}^{r} X^u_i d_{r-u} p_{j,i}$ or, in matricial form,

$$\mathrm{E}\left\{\mathbf{Y}|\mathbf{X}\right\} = \mathbf{D}\cdot\mathbf{X}\cdot\mathbf{P} = \hat{\mathbf{Z}}\cdot\mathbf{P}\,. \tag{31}$$

Likewise, using the law of total variance together with the equations above we can get closed-form expressions for $\mathrm{Var}\left\{Y^r_{j,i}|\mathbf{X}\right\}$ and $\mathrm{Cov}\left\{Y^r_{j,i}, Y^s_{j,i}|\mathbf{X}\right\}$. These expressions are too long and we do not need them for the purpose of this document, so we just note that, added along j, they can be written in matricial form as

$$\begin{aligned}
\mathbf{\Sigma}_{\mathbf{Y}|\mathbf{X}} &= \mathrm{diag}\left\{\mathbf{DX}\cdot\mathbf{1}_N\right\} - \mathbf{D}\cdot\mathrm{diag}\left\{\mathbf{X}\cdot\mathbf{r}_1\right\}\cdot\mathbf{D}^T \\
&\quad + \mathbf{D}\cdot\left[\sum_{i=1}^{N}\left(\mathbf{X}_i\mathbf{X}_i^T \circ \boldsymbol{E}_i\right)\cdot r_2(i)\right]\cdot\mathbf{D}^T\,.
\end{aligned} \tag{32}$$

The definition of $\mathbf{r}_1$ and $r_2(i)$ can be found after (17) in Section III-B.

References

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

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

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

[4] G. Danezis, "Mix-networks with restricted routes," in *Privacy Enhancing Technologies*, 2003, pp. 1–17.

[5] C. Diaz and B. Preneel, "Taxonomy of mixes and dummy traffic," in *Information Security Management, Education and Privacy*, 2004, pp. 217–232.

[6] C. Diaz and A. Serjantov, "Generalising mixes," in *Privacy Enhancing Technologies*, 2003, pp. 18–31.

[7] D. Rebollo-Monedero, J. Parra-Arnau, J. Forné, and C. Diaz, "Optimizing the design parameters of threshold pool mixes for anonymity and delay," *Computer Networks*, vol. 67, no. 0, pp. 180–200, July 2014.

[8] G. Danezis and C. Troncoso, "Vida: How to use Bayesian inference to de-anonymize persistent communications," in *9th Privacy Enhancing Technologies Symposium*, 2009, pp. 56–72.

[9] C. Troncoso, B. Gierlichs, B. Preneel, and I. Verbauwhede, "Perfect matching disclosure attacks," in *8th Symposium on Privacy Enhancing Technologies*, 2008, pp. 2–23.

[10] G. Danezis, "Statistical disclosure attacks: Traffic confirmation in open environments," in *Proceedings of Security and Privacy in the Age of Uncertainty*, Athens, 2003, pp. 421–426.

[11] N. Mathewson and R. Dingledine, "Practical traffic analysis: Extending and resisting statistical disclosure," in *4th Workshop on Privacy Enhancing Technologies*, 2004, pp. 17–34.

[12] 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.

[13] G. Danezis, "The traffic analysis of continuous-time mixes," in *Privacy Enhancing Technologies*, 2005, pp. 35–50.

[14] D. Rebollo-Monedero, J. Parra-Arnau, C. Diaz, and J. Forné, "On the measurement of privacy as an attackers estimation error," *International Journal of Information Security*, vol. 12, no. 2, pp. 129–149, 2013.

[15] S. Oya, C. Troncoso, and F. Perez-Gonzalez, "Meet the family of statistical disclosure attacks," in *IEEE Global Conference on Signal and Information Processing*, 2013, pp. 233–236.

[16] F. Pérez-González, C. Troncoso, and S. Oya, "A least squares approach to the static traffic analysis of high-latency anonymous communication systems," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 9, pp. 1341–1355, Sept 2014.

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

[18] D. Kesdogan and L. Pimenidis, "The hitting set attack on anonymity protocols," in *6th Workshop on Information Hiding*, 2004, pp. 326–339.

[19] G. Danezis, C. Diaz, and C. Troncoso, "Two-sided statistical disclosure attack," in *7th Symposium on Privacy Enhancing Technologies*, 2007, pp. 30–44.

[20] F. Pérez-González and C. Troncoso, "Understanding statistical disclosure: A least squares approach," in *12th Symposium on Privacy Enhancing Technologies*, 2012, pp. 38–57.

[21] S. Oya, C. Troncoso, and F. Pérez-González, "Understanding the effects of real-world behavior in statistical disclosure attacks," in *IEEE International Workshop on Information Forensics and Security*, 2014, pp. 72–77.

[22] S. Oya, F. Pérez-González, and C. Troncoso, "Technical report for id tnet-2015-00294 "optimal delay characteristic when the number of users is comparable to the number of rounds"," http://gpsc.uvigo.es/sites/default/files/publications/TechRepToN2016.pdf.

[23] D. Kesdogan, J. Egner, and R. Bschkes, "Stop-and-Go-MIXes providing probabilistic anonymity in an open system," in *Information Hiding*, 1998, pp. 83–98.

[24] A. Serjantov and G. Danezis, "Towards an information theoretic metric for anonymity," in *2nd Workshop on Privacy Enhancing Technologies*, 2002, pp. 41–53.

Simon Oya received the Telecommunication Engineer degree from the University of Vigo, Spain, in 2012. He is currently pursuing the Ph.D. degree in Telecommunication Engineering in the same university.

His research interest is the study of privacy-preserving technologies from a signal processing point of view, focusing on anonymous communication channels.

Fernando Pérez-González (M'90-SM'09-F'16) received the Telecommunication Engineer degree from the University of Santiago, Santiago, Spain in 1990 and the Ph.D. degree in telecommunications engineering from the University of Vigo, Vigo, Spain, in 1993.

He joined the faculty of the School of Telecommunication Engineering, University of Vigo, as an assistant professor in 1990, where he is currently a Professor. From 2009 to 2011 he was the Prince of Asturias Endowed Chair of Information Science and Technology with the University of New Mexico, Albuquerque, NM, USA, where he is currently a Research Professor. His research interests lie in the areas of digital communications, adaptive algorithms, privacy enhancing technologies, and information forensics and security. He has coauthored several international patents related to watermarking for video surveillance, integrity protection of printed documents, fingerprinting of audio signals, and digital terrestrial broadcasting systems.

Prof. Pérez-González has co-authored over 50 papers in leading international journals and more than 160 peer-reviewed conference papers. He has been the principal investigator of the University of Vigo group which participated in several European projects, including CERTIMARK, ECRYPT, REWIND, NIFTY and WITDOM. From 2007 to 2010 he was Program Manager of the Spanish National R&D Plan on Electronic and Communication Technologies, Ministry of Science and Innovation. From 2007 to 2014 he was Executive Director of the Galician Research and Development Center in Advanced Telecommunications (GRADIANT). He served as an Associate Editor of IEEE SIGNAL PROCESSING LETTERS (2005-2009) and the IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY (2006-2010). Currently, he is an Associate Editor of the LNCS TRANSACTIONS ON DATA HIDING AND MULTIMEDIA SECURITY, and the EURASIP INTERNATIONAL JOURNAL ON INFORMATION FORENSICS AND SECURITY.

Carmela Troncoso is a senior researcher at IMDEA Software Institute in Spain. She holds a Master in Telecommunications Engineering from the University of Vigo (2006) and a Ph.D. in Engineering from the Katholieke Universiteit Leuven (2011).

Her research interest include the design and analysis of privacy-preserving technologies, with particular focus on anonymous communications and location privacy.

Dr. Troncoso has co-authored over 25 papers in leading international journals and prestigious conference proceedings in the fields of security and privacy. She has been participated in several security oriented projects both and national and international level.