You Cannot Hide for Long: De-Anonymization of Real-World Dynamic Behaviour

George Danezis
Microsoft Research
Cambridge, UK
gdane@microsoft.com

Carmela Troncoso
Gradiant
Vigo, Spain
ctroncoso@gradiant.org

ABSTRACT

Disclosure attacks against anonymization systems have traditionally assumed that users exhibit stable patterns of communications in the long term. We use datasets of real traffic to show that this assumption does not hold: usage patterns email, mailing lists, and location-based services are dynamic in nature. We introduce the sequential statistical disclosure technique, which explicitly takes into account the evolution of user behavior over time and outperforms traditional profiling techniques, both at detection and quantification of rates of actions. Our results demonstrate that despite the changing patterns of use: low sending rates to specific receivers are still detectable, surprisingly short periods of observation are sufficient to make inferences about users' behaviour, and the characteristics of real behaviour allows for inferences even in secure system configurations.

Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*; C.2.0 [**Computer-communications networks**]: General—*Security and protection*

Keywords

Privacy; traffic analysis; anonymization; de-anonymization

1. INTRODUCTION

Anonymization is a key building block in providing confidentiality in privacy-preserving technologies. Encrypted confidential information, such as medical status or political interests, may be inferred from the identities of communication partners (e.g., communicating with a specialized doctor, or posting to a mailing list), or the location from which the communication takes place (e.g., specialized clinics, political headquarters). Through anonymization, user actions are separated from their identity to provide some degree of privacy. In a system relying on anonymization actions can only be ascribed to a potential set of users, often called the anonymity set.

Anonymization can be used directly to provide private communications, such as email [7] or web-browsing[10]. Furthermore, several privacy-preserving technologies assume that an anonymization layer is available to ensure that data are not linkable through the network addresses of their users. For instance, selective disclosure credentials, like U-prove [14] and Idemix [11], also assume that credentials are not linkable through user network addresses. Anonymization is also required in electronic election systems, to de-link the individual casting a vote from the resulting plaintext ballot. As a last example, specific location privacy-preserving mechanisms need to ensure that the identity of users and locations cannot be linked, e.g., as a means to computing privacy friendly aggregate traffic density maps [5].

A perfect anonymization mechanism would guarantee that all actions could always be ascribed to any user in the system with equal probability. However, the creation of anonymity sets is often constrained by implementation considerations, such as latency or network overheads. As a result, the protection that practical anonymization mechanisms offer is never perfect and some amount of information about the link between identities and actions is leaked. This work proposes models to make use of these leakages to reconstruct the behavioral patterns and actions of users over time.

There is a long line of work demonstrating that observations from anonymization systems, that provide both anonymity and unlinkability between actions, can be used to reconstruct user behavioural profiles [1] as long as these are *stable over time*. In this work, we extend this class of attacks to show that they are applicable in much wider settings: we do away with the stability assumption, we show they are effective against real-world anonymized traces of actions, they can be made robust to false-positives, and that they are effective given much less time and traffic than previously thought.

Specifically the contributions of this paper include:

- An analysis framework that allows an adversary to infer the rate at which users perform specific actions when these actions are mediated over an anonymous channel, and when rates may change over time. We call this new family of techniques *sequential statistical disclosure*. To our knowledge, this is the first model to tackle dynamic action profiles of users that change over time – a key feature when deploying attacks in the real world.

WPES'13, November 4, 2013, Berlin, Germany.

ACM 978-1-4503-2485-4/13/11 ...$15.00.
http://dx.doi.org/10.1145/2517840.2517846.

- A Bayesian model to extract probability distributions over a user profile from anonymized observations. Our method is based on sequential Monte-Carlo techniques, also known as particle filters. We adapt standard methods to the proposed traffic analysis model to ensure that they effectively track user profiles over time; they deal with users' a-priori low sending rates without producing unacceptable rates of false positives or negatives; and they run efficiently. This is the first application of principled Bayesian modelling combined with Sequential Monte-Carlo to the problem of generic de-anonymization.

- An evaluation of the sequential statistical disclosure techniques and comparison with adapted state of the art statistical disclosure attacks [18]. For the first time, we evaluate the statistical disclosure attack's performance in presence of the dynamic profiles that are extracted from real-world datasets. As opposed to previous studies, that made extensive use of simplified synthetic traces, we study three real-world applications: traces of anonymized real email communications, traces of anonymized traffic to a mailing list server, and traces of anonymized locations from a real-world service. We show that our techniques can de-anonymize a significant fractions of communications, given surprisingly little information.

Our findings provide an insight into the effectiveness of statistical attacks against real usage patterns. We conclude that such attacks are more effective than anticipated: i) they can be effective for rather low action rates, ii) they are effective over a much shorter period of time than previously thought, and iii) they can be effective for secure configurations of the anonymity system.

This paper starts with a review of the relevant literature on traffic analysis, in Sect. 2, that a familiar reader can safely skip. Sect. 3 describes the probabilistic model underlying the sequential statistical disclosure attack, and Sect. 4 describes the training and sequential Monte-Carlo techniques devised to infer its hidden parameters. Sect. 5 describes the datasets used, and presents a thorough evaluation of the proposed and previous schemes, especially in terms of false-positives. The final section offers some conclusions.

2. BACKGROUND & RELATED WORK

The risks of de-anonymizing rich data sets has been highlighted in the context of census record microdata [24] (also critically re-examined by Golle [13]), social networks [20, 27] and movie preference graphs [19]. These works attack anonymized releases of full user profiles that contain features (also known as quasi-identifiers) that can be used to re-identify the profiles. Our work is concerned with anonymization at the level of individual actions, for example a single message a sender sends to a specific email address, or to a mailing list, anonymously; or a single location revealed by a user anonymously. The anonymity mechanism used hides the relation between the actor and the action, but also obscures the relationship between actions over time, making the techniques for re-identifying large profile unusable.

Anonymization of single actions over time is mostly performed and studied in the context of anonymous communications. In fact we often use the term "sender" to be synonymous to a generic actor, and "receiver" to simply indicate the label associated with an observable action. Anonymous communications channels were first introduced by Chaum [4], and their goal is to hide communication partners in a network.

The study of generic long-term attacks against anonymity systems began with the exposition of long-term intersection attacks by Berthold et al. [2]. They describe simple attacks against senders in anonymity systems that *persistently* send to a single receiver. The original intersection attack was largely conceptual, but Kesdogan introduced the Disclosure attack [1] that extended long-term attacks to senders with multiple *persistent* contacts over time. In this work we show that these attacks can be performed even against shorter term (and changing) patterns of communications, without the need for them to be persistent in the long term.

The Hitting Set attack [16] extends the reach of the Disclosure attack and has been the subject of considerable study [15, 22]. The model of the Hitting Set attack is quite sensitive to its assumptions: the anonymous channel has to be a threshold mix, and the target sender Alice needs to have a bounded and known number of friends, to whom she sends at a known rate. The latter assumption makes the attack precise and powerful, but also limits its generality. The family of Statistical Disclosure [6] attacks, on the other hand, aggressively simplifies and linearises the operation of the anonymity system, and the behaviour of Alice. As a result, it is quicker to perform and more adaptive, at the cost of providing less accurate results. In this work we use a variant of the Statistical Disclosure Attack [18] that focuses on accurately estimating the background traffic to receivers, and adapt it to Timed Mixes. To evaluate the performance of all attacks we use the established least squares error metric [21].

Our approach takes into account that each observable interaction of an actor can only be linked with a single observable action (one message in the anonymity system results in one message out of the system) in the tradition of the Perfect Matching Disclosure [26]. However, since we only consider a single actor-action target at a time, the computations become simpler. Furthermore, instead of following an optimization approach, we try to characterize full posterior distributions of the knowledge of the adversary about the target link given the evidence. This Bayesian approach follows the same tradition as Vida [8], which first described long-term attacks in the language of Bayesian inference. However, instead of using Gibbs sampling we implement an inference algorithm using on-line tracking methods namely sequential Monte-Carlo sampling. These techniques, also known as particle filters, have been developed in the context of non-linear tracking, for example for radar. An excellent tutorial on particle filters by Arulampalam et al. [17] provides all the background needed to follow the details of the proposed inference algorithm. Those keen on an in depth background on tracking techniques can refer to Ristic et al. [23].

We have taken a great amount of care to avoid making classification mistakes, which have been overlooked by previous attacks, but are frequent when analysing real datasets. Diaz et al. [9], discuss the dangers of applying a strict likelihood based model with a prior or side information that is incorrect, since no amount of evidence contrary to the prior improves the results. Finally, *the base rate fallacy* has been identified as a key challenge when it comes to deploying traffic analysis techniques in the wild, most notably and anonymously by The 23$^{\text{rd}}$ Raccoon [25]. One of the main advantages of the proposed model is that it directly incorporates information about the low prior associated with any actor-action relationship in order to minimize false positive rates.

3. A MODEL FOR TRACKING DYNAMIC USER PROFILES

The problem of de-anonymizing messages sent through an anonymity system can be cast as a problem of statistical inference. Given some knowledge of the functioning of the anonymity system, and some general assumptions about the behaviour of users, the task of the adversary is to infer who is talking to whom, from anonymized observations.

Our aim is to link senders (actors) with actions (such as specific receivers) where patterns of behaviour change over time. We use a model-based Bayesian inference approach to solve this concrete traffic analysis inference problem: this means we define a generative model of the system, namely all the mechanisms and probability distributions involved in generating the public traffic traces from the secret user profiles. Then we "invert" this model (using Bayes' rule) to get the probability of the secret user profiles, given the observations. Since doing so analytically is infeasible, we implement an approximation based on sequential Monte-Carlo sampling (also known as a particle filter). We note that, as any model, the one we propose necessarily abstracts details of any particular system and makes assumptions about its operation and use. The quality of such a model cannot be judged in terms of its faithfulness to every real-world detail, but rather must be judged in terms of its effectiveness in achieving the task at hand, namely de-anonymization.

In this work, we assume an adversary is provided with records of users sending messages through the anonymity system, and records of received messages / actions out of the anonymity system, in "batches" over time. [1] We assume that the traffic within distinct "batches" of the anonymity system is opaque, and that the anonymity sets resulting are perfect: any sender could have been sending messages to any receiver within a batch. Real-world anonymity systems need to keep "batches" small in order to minimize the latency suffered by messages[2].

This paper casts the profile inference problem in the context of three example anonymity-based applications. However, we note that the problem we solve is isomorphic to many other data privacy mechanisms. In fact, any privacy system that involves hiding the relation between a visible set of actors and a set of actions in consecutive periods can be analysed using straight-forward variants of our techniques.

3.1 A Basic Model

We devote this section to describing the generative model underlying the sequential statistical disclosure attack. Fig. 1 illustrates the model in the standard plate notation for graphic models. Circles represent variables, while rectangles represent operations such as sampling or addition. Directed edges denote conditional dependence amongst variables. Plates represent independent executions of a process. Observations visible to the adversary are under the dotted line.

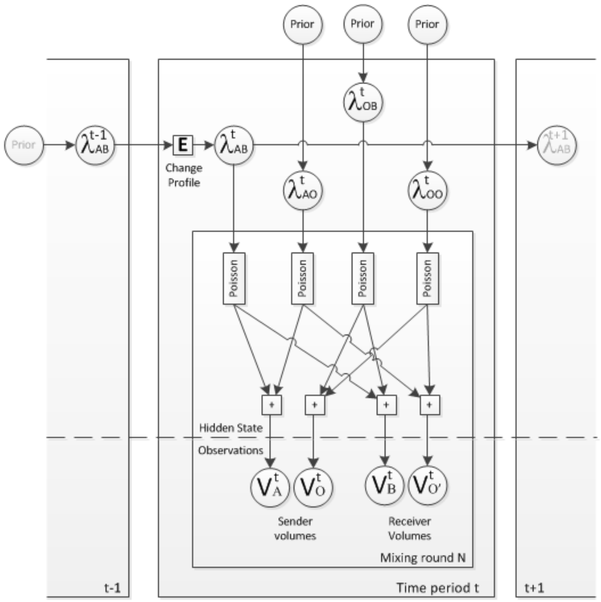

Figure 1: The generative model underlying the Sequential Statistical Disclosure in standard plate notation.

Consider an actor Alice (A), who may send messages to user Bob (B) over a number of consecutive epochs (denoted t, for time). We assume that in each epoch Alice sends a number of messages through the anonymity system to Bob, at some Poisson rate λ^t_{AB} ($\lambda^t_{AB} \to 0$ if Alice does not send any messages to Bob). Of course, other users (O) may also send messages to Bob with an aggregate Poisson rate λ^t_{OB}, as well as to other receivers with an aggregate Poisson rate λ^t_{OO}. Alice, may also send messages to others with a Poisson rate λ^t_{AO}.

We assume that all messages generated within an epoch are anonymized using an arbitrary set of (perfect) anonymity systems. Since epochs can be quite long, and the volume of traffic large, in a realistic setting not all messages are mixed with each other. To model this situation we consider that N anonymizers are used consecutively in each epoch (e.g., a timed mix with an equal time threshold, or a k-anonymity based location privacy mechanism used over time).

We consider that all messages sent to an anonymizer are perfectly mixed together. This means that an adversary can only observe the aggregate volume of messages Alice sends $V^{(t,n)}_A$ to this anonymizer, and the aggregate volume that Bob receives $V^{(t,n)}_B$ from the anonymizer; along with the volume sent and received by other users ($V^{(t,n)}_O$ and $V^{(t,n)}_{O'}$ respectively). However, there is no way to tell whether a message from Alice was sent to Bob or to someone else within each mixing batch (t, n). The total number of messages sent or received in an epoch t can be computed as $V^t_x = \sum_n V^{(t,n)}_x$

Alice draws a number of messages to be sent during a whole epoch using a Poisson distribution: $V^t_A \xleftarrow{\$} \text{Poisson}(\lambda^t_{AB} + \lambda^t_{AO})$. The anonymity system then chooses an arbitrary partition of the messages that Alice, into disparate mixing rounds, such that the total number of messages across rounds $V^{(t,n)}_A$ is equal to V^t_A. Since both $V^{(t,n)}_A$ and V^t_A are

[1] While this model is stronger than the more recent ones applied to Onion Routing systems [12, 10], it allows us to get an insight into the fundamental limits of strong anonymity systems.

[2] As an example in our evaluation we assume messages (or other actions) are batched daily.

observed directly, we are not concerned about the distribution over partitions.

Furthermore, within a batch we model the number of message sent from Alice to Bob as a binomial distribution, with probability $p^t_{AB} = \lambda^t_{AB}/(\lambda^t_{AB}+\lambda^t_{AO})$. Similarly, others send to Bob with a probability $p^t_{OB} = \lambda^t_{OB}/(\lambda^t_{OB}+\lambda^t_{OO})$ per message. Thus, we have:

$$\text{Hidden} \quad \begin{cases} V^{(t,n)}_{AB} \xleftarrow{\$} \text{Binom}(p^t_{AB}, V^{(t,n)}_{A}) \\ V^{(t,n)}_{OB} \xleftarrow{\$} \text{Binom}(p^t_{OB}, V^{(t,n)}_{O}) \end{cases} \tag{1}$$

$$\text{Visible} \quad \begin{cases} V^{(t,n)}_{B} = V^{(t,n)}_{AB} + V^{(t,n)}_{OB} \\ V^{(t,n)}_{O'} = V^{(t,n)}_{A} + V^{(t,n)}_{O} - V^{(t,n)}_{B} \end{cases} \tag{2}$$

Observe that the exact volumes exchanged by each pair of participants are hidden, while the total volumes sent and received by each party, at each mixing round and epoch, are revealed. We note that extending this model to allow for dummy traffic, or other noise, is simple: one could simply add some noise from the distribution of dummy traffic according to the process by which is it generated, or some approximation of it, to the observed aggregates.

We assume that within an epoch the sending rates for Alice and others are stable, but allow them to differ between epochs. This way we can perform traffic analysis against sender-receiver pairs that have changing patterns of communications. Our model assumes we can approximately model the way in which sender rates-receiver change over time. In particular, we assume that the adversary knows the distribution for the sending rate from Alice to Bob at time t (λ^t_{AB}) given the rate at time $t-1$ (λ^{t-1}_{AB}). We call this the *profile evolution probability* $E(\lambda^t_{AB}|\lambda^{t-1}_{AB})$, and model the evolving rate of Alice-Bob as sampled from this distribution as:

$$\lambda^t_{AB} \xleftarrow{\$} E(\lambda^t_{AB}|\lambda^{t-1}_{AB})\,. \tag{3}$$

The profile evolution probability $E(\lambda^t_{AB}|\lambda^{t-1}_{AB})$ expresses both the existence of communications between any two parties, as well as its intensity. Of course the initial λ^0_{AB} has no previous epoch to inform it, and thus we assume it is generated from a very broad prior ($\lambda^0_{AB} \xleftarrow{\$}$ *prior*).

While we want to accurately model changes in the rate of traffic between Alice and Bob, we do not want our inference to be influenced by other receivers' actions. For this reason we assume that the rate at which others receive traffic is Poisson and determined in each batch independently from other batches, with a very broad prior ($\lambda^t_{AO} \xleftarrow{\$}$ *prior*, $\lambda^t_{OO} \xleftarrow{\$}$ *prior*). For the broad prior for all rates we use a Gamma($\alpha = 1, \beta = 1$) distribution (which is the conjugate prior of the Poisson distribution).

The model fully characterizes the distribution over all aspects of the system that lead to the adversary's observations. The main remaining challenge is to build a reverse inference algorithm that, given the observations, provides estimates of hidden variables. The key hidden variable of interest to the adversary is the rate of sending between Alice and Bob during each epoch, λ^t_{AB}. All other quantities are nuisance parameters in the eyes of the adversary, which need to be co-estimated to ensure good accuracy.

3.2 The Likelihood Function

At the heart of our inference technique, in Sect. 4, lies the likelihood function that is fully characterized by the generative model. The likelihood function:

$$L(V^{(t,n)}_{\{A,B,O,O'\}}|\lambda^t_{\{\star\}}),$$

where $\lambda^t_{\{\star\}}$ is a shorthand for $\lambda^t_{\{AB,AO,OB,OO\}}$, represents how likely the observation of the adversary is within an epoch, given a set of sending rates from Alice and others to all users. In the reminder of this section we use the shorthand L when referring to the likelihood function.

At epoch t the adversary observes the total volumes of traffic sent and received by each party: $V^{(t,n)}_A$, $V^{(t,n)}_B$, $V^{(t,n)}_O$, and $V^{(t,n)}_{O'}$ for every batch $n \in N$. Assuming a hidden state of the system (i.e., the set of rates λ^t_{AB}, λ^t_{AO}, λ^t_{OB}, λ^t_{OO}), the likelihood L that this state has generated the observation can be computed as follows:

$$L = \Pr[V^t_A; \lambda^t_{AB} + \lambda^t_{AO}] \cdot \Pr[V^t_O; \lambda^t_{OB} + \lambda^t_{OO}] \prod_{n=1}^{N} L^n \tag{4}$$

$$L^n = \sum_{k=0}^{\min(V^{(t,n)}_A, V^{(t,n)}_B)} \Pr_b[k; V^{(t,n)}_A, p^t_{ab}] \cdot \Pr_b[V^{(t,n)}_B - k; V_O, p^t_{ob}]$$

where L^n is the likelihood that these set of rates generated each individual batch $0 \le k \le \min(V^{(t,n)}_A, V^{(t,n)}_B)$ models the possible number of messages sent from A to B in the current batch n. $\Pr_b[n; N, p]$ is a shorthand for the Binomial distribution, $p^t_{ab} = \lambda^t_{AB}/(\lambda^t_{AB} + \lambda^t_{AO})$ and $p^t_{ob} = \lambda^t_{OB}/(\lambda^t_{OB} + \lambda^t_{OO})$. $\Pr[k; \lambda]$ is a shorthand for the Poisson distribution, $\Pr[k; \lambda] = \text{Poisson}(k; \lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$.

This expression can be computed incrementally and efficiently. Given the Binomial distribution's probability mass function $\Pr_b[k; V, p] = \binom{V}{k} p^k (1-p)^{V-k}$:

$$\Pr[k+1; V, p] = \Pr[k; V, p] \cdot \frac{V-k}{k+1} \cdot \frac{p}{1-p}$$

$$\Pr[k-1; V, p] = \Pr[k; V, p] \cdot \frac{k}{V-k+1} \cdot \frac{1-p}{p}$$

3.3 The Profile Evolution Probability

The distribution $E(\lambda^t_{AB}|\lambda^{t-1}_{AB})$ represents the probability that the sending rate λ^{t-1}_{AB} evolves to the rate λ^t_{AB} in epoch t. It is in effect a model of the relation of actor-action rates across epochs.

We observe from our real-world datasets that rates λ^t_{AB} have structure. Either a sender knows a particular receiver and they send messages with some positive rate, or they do not know them which leads to a rate of zero. Whether a rate is positive or zero is rather stable over time and we need to model it as such.

As a result, we model the evolution of a rate λ^{t-1}_{AB} into a rate λ^t_{AB} as a two stage process. We first define a set of probabilities that determine whether a link with a positive or zero rate at time $t-1$ retains a positive or zero rate at time t. With probability p_{PZ} a rate that is positive becomes zero, and with probability p_{ZZ} a rate that is zero remains zero. These are sufficient to determine the probabilities of transiting from zero to a positive rate ($p_{ZP} = 1 - p_{ZZ}$) and retaining a positive rate ($p_{PP} = 1 - p_{PZ}$).

If the rate at time either t or $t-1$ is positive we define a distribution on their difference. We have observed very heavy tails over the distribution of differences of rates between epochs. Consequently, we model it as a mixture of two exponential distributions: with some probability p the rate

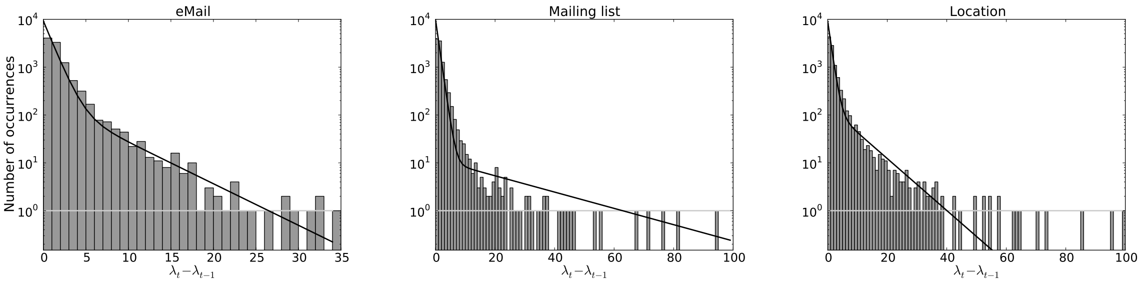

Figure 2: The goodness of fit between the mixture of exponentials (Δ, Eq. 5) component (line) of the model for $E(\lambda_{AB}^t|\lambda_{AB}^{t-1})$ and the observed differences of positive rates in the three datasets (histogram).

difference is drawn from an exponential distribution with a parameter λ_s, and with probability $(1-p)$ it is drawn from an exponential with a parameter λ_b. By naming convention, we ensure that $\lambda_s \leq \lambda_b$, which can be interpreted as the second exponential modelling the long tail of the distribution of differences of rates.

Overall the likelihood $E(\lambda_{AB}^t|\lambda_{AB}^{t-1})$ can be computed as:

$$\begin{aligned} \delta &= |\lambda_{AB}^t - \lambda_{AB}^{t-1}| \\ \Delta &= p \cdot \mathrm{Exp}(\delta; \lambda_s) + (1-p) \cdot \mathrm{Exp}(\delta; \lambda_b) \end{aligned} \tag{5}$$

$$E(\lambda_{AB}^t|\lambda_{AB}^{t-1}) = \begin{cases} p_{ZZ} & \text{if } \lambda^t = 0 \wedge \lambda^{t-1} = 0 \\ p_{ZP} \cdot \Delta & \text{if } \lambda^t > 0 \wedge \lambda^{t-1} = 0 \\ p_{PZ} \cdot \Delta & \text{if } \lambda^t = 0 \wedge \lambda^{t-1} > 0 \\ p_{PP} \cdot \Delta & \text{if } \lambda^t > 0 \wedge \lambda^{t-1} > 0 \end{cases} \tag{6}$$

The parameters p_{PZ}, p_{ZZ}, p, λ_s and λ_b are sufficient to determine the model for the transition probability $E(\lambda_{AB}^t|\lambda_{AB}^{t-1})$. In our evaluation we infer those five numbers directly from non-anonymized traces for each dataset separately, using an expectation maximization (EM) algorithm. The parameters are estimated on a separate partition of the data from the one used for the evaluation, to prevent over-fitting.

Table 1 summarises these values for the datasets used in the evaluation, and Fig. 2 demonstrates the goodness of fit of the mixture of exponentials with the observed distribution of differences in the real datasets. We note that in all datasets rates can vary considerably across epochs as witnessed by the non-negligible probability of switching from zero to positive and back, as well as the heavy tails exhibited by the distribution over differences of positive rates. These indicate that the assumption of stability of sending patterns over time, at the heart of previous long-term attacks, does not hold for traces of real-world actions.

4. THE PARTICLE FILTER

Sequential Monte-Carlo algorithms, also known as particle filters, are a family of techniques for inferring the hidden parameters of sequential models. Our models are sequential in nature, since λ_{AB}^t is evolving forward in time, and information needed to model λ_{AB}^t at time t is contained in the value of λ_{AB}^{t-1} at time $t-1$.

Particle filters act on a collection of sample hidden states of the system under observation, each represented by a particle. The distribution of particles follows the posterior probability distribution of the hidden states given the evidence processed by the filter. Here, each particle represents a sample (λ_{AB}^t, λ_{OB}^t) at time t, and the collection of particles is a non-parametric representation of the posterior distribution over the hidden sending rates. Hence, the mean, variance and other statistics of this collection can be used to estimate hidden parameters of interest.

```
function SSDFilter($V_{A,O,B,O'}^{(t,n)}$)
    for all particles $i$ do
        $(\lambda_{ABi}^0, \lambda_{OBi}^0) \sim$ priors;
    end for
    for all epochs $t$ do
        for all particles $i$ do
            $\lambda_{Ai}, w_{Ai} \xleftarrow{\$} \mathrm{Gamma}(V_A^t + 1, 1)$;
            $\lambda_{Oi}, w_{Oi} \xleftarrow{\$} \mathrm{Gamma}(V_O^t + 1, 1)$;
            $\lambda_{Bi}, w_{Bi} \xleftarrow{\$} \mathrm{Gamma}(V_B^t + 1, 1)$;
            $\lambda'_{ABi}, w_\theta \xleftarrow{\$}$ Mixture $\mathcal{M}$
            if $\lambda'_{ABi} > \lambda_{Ai}$ or $\lambda'_{ABi} > \lambda_{Bi}$ then
                reject & continue;
            end if
            $\lambda'_{OBi} \leftarrow \lambda_{Bi} - \lambda'_{ABi}$;
            $w_{ABi} \leftarrow L(V_{\{A,B,O,O'\}}^{(t,n)}|\lambda_{\{\star\}}^t) \cdot E(\lambda'_{ABi}|\lambda_{ABi}^{t-1})$;
            $w_i \leftarrow w_{ABi}/w_{Ai} \cdot w_{Oi} \cdot w_{Bi} \cdot w_\theta$;
        end for
        for all particles $i$ do
            $(\lambda_{ABi}^t, \lambda_{OBi}^t) \leftarrow$ Re-sample $(\lambda'_{ABi}, \lambda'_{OBi}) \sim w_i$;
        end for
    end for
    return $(\lambda_{ABi}^{\max t}, \lambda_{OBi}^{\max t})$;
end function
```

Figure 3: The SSD particle filter algorithm

Particle filters make use of a crucial property of the posterior distribution resulting from the application of Bayes rule. The probability of a hidden state given the observations at time t, is proportional to the likelihood of the observation given the hidden state, as well as to the probability of the hidden state given previous hidden states at time $t-1$. Concretely, for the sequential disclosure attack model:

$$\begin{aligned} \Pr[(\lambda_{AB}^t, \lambda_{OB}^t)|V_{A,B,O,O'}^{(t,n)}] \sim & L(V_{A,B,O,O'}^{(t,n)}|\lambda_{\{\star\}}^t) \cdot \\ & E(\lambda_{AB}^t|\lambda_{AB}^{t-1}) \cdot \Pr[(\lambda_{AB}^{t-1}, \lambda_{OB}^{t-1})], \end{aligned}$$

where $L()$ is the likelihood of either the basic or robust model (see section 3.2) and $E()$ is the profile evolution probability (see section 3.3). Conceptually, particles from previous time periods represent the prior distribution $\Pr[(\lambda_{AB}^{t-1}, \lambda_{OB}^{t-1})]$. They are each associated with a new particle, sampled from a mixture distribution with heavy tails, which is re-weighted using the likelihood $L(V_{\{A,B,O,O'\}}^{(t,n)}|\lambda_{\{\star\}}^t)$ and $E(\lambda_{AB}^t|\lambda_{AB}^{t-1})$.

Finally, the new particle set is re-sampled according to the weight to represent the posterior distribution at time t.

The pseudo-code for the particle filter algorithm is presented in Fig. 3. Particles are initialized with rates following the prior distribution as zero or positive, as well as their value if they are positive (using $E(\lambda^0_{AB}|0)$). Then for each epoch and particle we sample plausible candidate rates for Alice, Bob, and other senders using a Gamma distribution with a mean equal to the observed volumes of traffic. Each particle, representing a possible prior state at $t-1$, is associated with a sampled candidate particle from a heavy tailed distribution $\mathcal{M}$, for time t. We tailor this distribution to propose some particles representing no communication, some with typical communication volumes, and some with outlandishly high rates (to ensure robustness). If the candidate rate for λ'_{AB} is larger than the rate with which Alice sends or Bob receives, we simply reject it (without re-sampling). The other sending rates and the likelihood of that particle are then computed. We re-weight the candidate particles multiplying the likelihood and the profile evolution probability, and dividing the probability which witch rates are proposed. Once all particles have been processed, they are re-sampled according to their weights. The new collection of particles represents the posterior distribution at time t, and the analysis of the next epoch can commence.

We experimented with a varying number of particles, and found a number between 250–550 performs well, both to estimate the probability that Alice is sending to Bob at all, as well as their communication rate. Lower number of particles can be used for performance, but would only allow coarser inferences to be drawn, that are also subject to more statistical noise. There is an advantage in keeping the number of particles low, as each additional particle requires computing the likelihood function once more per epoch.

5. EVALUATION

5.1 Experimental Setup

To evaluate the sequential statistical disclosure attack we use traffic extracted from three real datasets of different nature:

eMail: The Enron dataset of email logs[3] released on August 2009. The dataset contains around 0.5M emails messages from 150 Enron employees. We discard 11 users that have sent fewer than 20 messages during the collection period. We note that traffic during the weekend is systematically very sparse, but chose to not remove it (or delay its delivery) to keep our analysis faithful to the constraint of this dataset.

Mailing list: A dataset we collected through processing the public archives of mailing list posts from the Independent Media Centre[4]. For each publically archived message we collected a pseudonym for the sender email address, a pseudonym for the target mailing list, and the date and time recorded. The dataset contains 293414 messages from 28237 unique senders, to 693 unique mailing list, over 105 months, in the period of Jun. 2004 - Feb. 2013.

	p_Z	p_{PZ}	p_{ZZ}	p	λ_s	λ_e
eMail	0.958	0.04	0.995	0.88	1.0	4.0
Mailing list	0.982	0.02	0.998	0.97	1.0	22.0
Location	0.993	0.06	0.999	0.87	1.0	7.0

Table 1: Estimated parameters of prior and $E(\lambda^t_{AB}|\lambda^{t-1}_{AB})$.

Location: we use the Gowalla dataset[5], collected from a location-based social networking website where users share their locations by checking-in. It contains of 6 442 890 check-ins from 196 591 users collected over the period of Feb. 2009 - Oct. 2010.

The choice of datasets was based on both availability but also relevance. In particular the email communications and mailing list communications are collected from organizations that may well have considered using privacy technologies, such as anonymization, to protect their communications.

The traffic from each of the datasets is used as input to an anonymity system that collects messages for one day before outputting them in a batch. We choose this time threshold to guarantee a reasonable balance between delay and anonymity. In Fig. 4 we show the batch size for different message delays. For the eMail and Mailing list cases one day at least is required to ensure a mean batch size in the order of a hundred, i.e., to ensure that the mean size of the anonymity set of messages is ~ 100. Shorter delay results in small batches, some of them having only one message, and hence provide small anonymity. Longer delays improve anonymity, but we consider waiting more than one day to be intolerable for users. The Location dataset, on the other hand, would allow for smaller delays but we choose to maintain one day for ease of comparison. (Note that the average batch size in the Location case is larger than 15 000 messages).

We consider stable epochs to last *only for one week* (hence the anonymity system outputs 7 batches per epoch). This choice is validated by the experiments, in which we see significant fluctuations in the users' sending rates even for such short periods. For this epoch duration we estimate the prior probability of communication, and the mixture parameters necessary to compute the probability of rate evolution according to $E(\lambda^t_{AB}|\lambda^{t-1}_{AB})$ for the three datasets using separate training datasets and the algorithm described in Sect. 3.3. The estimated parameters are shown in Table 1. The additional prior p_Z is computed directly and denotes the probability of a sender-receiver pair having a rate of zero within an epoch.

In our experiments we refer to two types of traces: the *conversation traces set* and the *silent traces set*. The former is a set of traces in which a target sender (A) communicates with a target receiver (B) for at least 16 weeks in a row (i.e., $\lambda^t_{AB} > 0$), with at most three successive silent epochs in between. The latter contains traces in which both A and B appear in all epochs but the target sender A does not communicate with recipient B (i.e., $\lambda^t_{AB} = 0$). The performance of any analysis against the second set is very important as the majority of traces encountered in the wild (96% in the eMail dataset, 98% in the Mailing list dataset, and 99% in the Location dataset) are silent, thus any analysis must yield credible results against them.

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

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

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

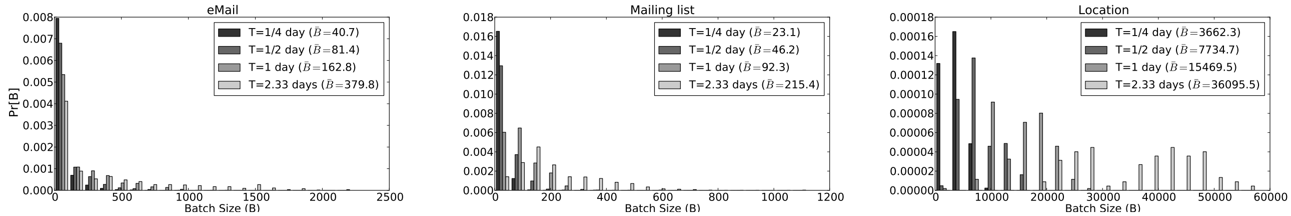

Figure 4: Batch sizes (B) for the three datasets depending on the timed threshold (T).

The anonymized traces for each of the datasets processed through a Timed Mix with period 1 day, and epoch length of 7 days have the following characteristics. The Location batches have a mean size of 14258 ($\sigma = 4338$), and the mean sending rate of target relations in the communicating traces is 5 per epoch ($\sigma = 12$), and the rate that others send to the same target has a mean of 1095 per epoch ($\sigma = 2360$). The eMail batches have a mean size of 231 ($\sigma = 147$). The eMail communication set features an average rate of sending of about 4 per epoch ($\sigma = 5$), and others send to the target with an average rate of 1 per epoch ($\sigma = 2$). Finally, the Mailing list set exhibits average batch sizes of 127 ($\sigma = 147$). Users in the communication traces send with a mean rate of 2 per epoch ($\sigma = 5$), and others send to the same target with a mean rate of 12 per epoch ($\sigma = 12$). The batch size of the Mailing List and Location datasets never falls to zero (the minimum batch size for the mailing list is > 10 and for the Location dataset > 4000), while this is the case occasionally for the eMail list set (on weekend days). All silent traces exhibit similar characteristics to the communication trace sets, except that the rate of sending of the target pair is zero. (All numbers rounded, since their exact value varies.)

5.2 Baseline Analysis of Timed Mixes

Traditional statistical disclosure attacks (SDA), have been formulated against simple threshold models of anonymity systems as well as pool mixes, and assume that action profiles are stable over time. Yet, the anonymity system we consider operates similarly to timed mixes, that gather messages for a fixed period of time, mix them, and send them out in a batch. To provide a fair comparison, and tease out the benefits of taking into account the evolving nature of profiles, we adapt the statistical disclosure variant by Dingledine and Mathewson [18] to the timed mix model. At any time the adversary keeps track of the background volume other senders direct to Bob using the volumes send by others and received by Bob. When Alice participates in a batch, this background estimate is used to approximate her contribution of messages to Bob. As expected, the analysis is extremely fast. For completeness, we reproduce the algorithm in the Appendix.

We compare the new techniques to the timed-mix SDA above, since it represents as a state of the art statistical disclosure attack, adapted to the mix types under consideration.[6] We consider two different variants of the SDA. The traditional *long* variant that all previous observed periods, and effectively assumes that the behavior of Alice, as well as the other senders, is overall stable. Since users are assumed to change their sending behavior over the observation period, we also consider a *short* variant that considers only traffic within the batches of an epoch to estimate the rate of a relationship.

5.3 Sample Trace Profile Inference

The SSD particle filter takes as input a sequence of counts per round of mixing per epoch for Alice, Bob and others ($V^{(t,n)}_{A,B,O,O'}$) and returns a set of particles per round t, namely ($\lambda^t_{ABi}, \lambda^t_{OBi}$). Each set of particles represent the posterior distribution of our belief about λ^t_{AB} given all the evidence in epochs zero to t. For every epoch we use these particles to compute the estimated mean value of $\lambda^t_{AB_{SSD}} = 1/|i| \cdot \sum_i \lambda^t_{ABi}$. We also use each set of particles to compute the 95% confidence intervals around our estimate of λ^t_{AB}, by discarding particles with the top and bottom 2.5% extreme values.

Figure 5 provides a graphical representation of the results of the filter applied to a sample trace. The analysis is performed over 16 weeks of traffic from a single user from the eMail dataset transmitting to a single receiver at a varying rate between 1 and 5 messages per week. The user messages have been mixed in batches of average size 244. The continuous line represents the real sending rate. The discontinuous line with squares ■ represents the mean estimate of the SSD model about the sending rate ($\hat{\lambda}^t_{SSD}$). The light grey region illustrates the 95% confidence intervals around the estimated mean. The value of each particle is also plotted for each epoch t as a semi-transparent gray circle, to illustrate their density.

For illustration purposes the result of the short term SDA ($\hat{\lambda}^{\text{short}}_{SDA}$) and long term SDA ($\hat{\lambda}^{\text{long}}_{SDA}$) are plotted with stars ★ and circles • respectively. We see the long term SDA is overly sensitive to previous behavior, and the short term SDA provides poorer accuracy. Neither provide any confidence intervals or other indication of their accuracy. This example illustrates that even at rather low sending rates (< 5) the SSD model provides good results close to the actual sending rate, and confidence intervals that can distinguish sending behavior from silent behaviour.

We further estimate the probability the rate is very low, defined as $\lambda^t_{AB} < 0.1$, by computing the fraction of particles under that threshold. The lower graph plots the fraction of particles indicating no sending is taking place.

5.4 Evaluation against conversation and silent trace sets

We measure the quality of the inference of the hidden λ^t_{AB} using a square error metric. For all attacks, we compute

[6] We use techniques from [18] as a baseline since in independent work we show that more recent techniques such as [21] do not offer significant advantages. For detailed comparisons see technical report `http://webs.uvigo.es/gpscuvigo/sites/default/files/publications/main-globalsip.pdf`

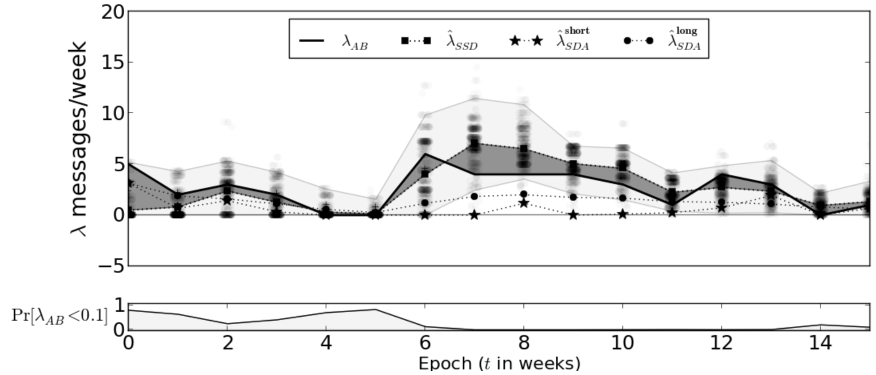

Figure 5: A sample trace with traffic analysis results over time.

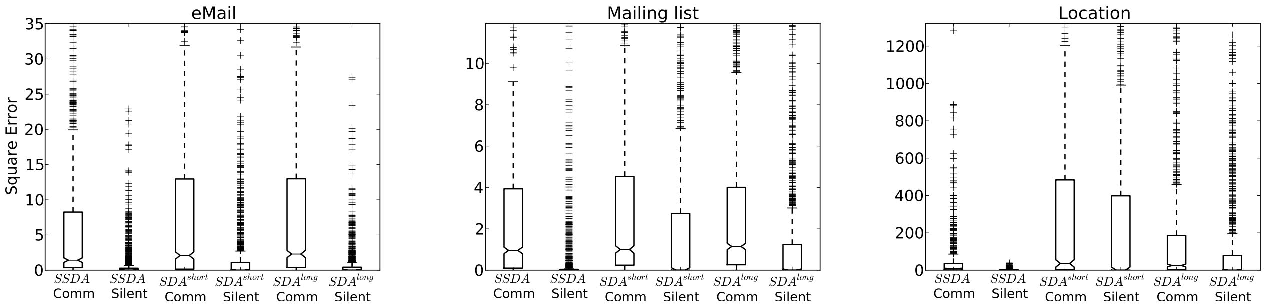

Figure 6: Square error between the rate prediction and actual rate of senders for individual epochs of the Sequential Statistical Disclosure compared with the traditional statistical disclosure over the short and long term. Communication and Silent traces.

$(\lambda^t_{AB} - \max(\hat{\lambda}^t_{AB}, 0))^2$ (when SDA estimates are negative we consider them to be equal to zero). Lower errors denote higher accuracy when profiling. Figure 6 summarizes the performance of the SSD model, as well as short and long term SDA against 100 user traces over 16 weeks. A standard box plot is provided summarising the distribution of square errors in the communication trace set (comm) and silent trace set (silent), for all attacks and for all data sets.

Fig. 6 illustrates that the typical quality of the SSD estimate of hidden communication rates is higher than for the traditional SDA models. We observe this when deploying the SSD model against traces containing conversations (Comm) in all datasets. The mean square error is for the SSD 11.7, 5.18, and 84.7 in eMail, Mailing List and Location respectively compared with 12.7, 9.1, and 3783.8 for the short SDA and 19.6, 6.8, 83069.5 for the long SDA (the difference between the means of the SSD and the long SDA for the Mailing list dataset is only significant with CI 95%, while other differences are significant with CI 99%). Importantly the SSD model performs very well against traces where no conversation is taking place (Silent). For silent traces the square error of the SSD is 0.7, 0.6 and 1.2 for the eMail, Mailing List and Location data sets respectively and 2.8, 7.0 and 2364.7 for the short SDA and 0.8, 4.3 and 358.3 for the long SDA (the difference between the SSD and long SDA for the eMmail dataset if not significant, while others are significant with a 99% CI). Long term SDA assumes that the rates of Alice to Bob, as well as others to Bob are stable across the weeks of analysis. It performs better against the Mailing List and Location datasets, but worse against the eMail dataset, where this assumptions does not hold. The short term SDA, performs as well as the long term SDA in the eMail dataset. Yet, it still under performs compared to the SSD method for two key reasons: it cannot use any past information, and it makes use of a much more naive likelihood model within each epoch.

It is worth noting that the values of the square error in Fig. 6 have a direct interpretation. They represent the variance of the error of the adversary when performing the traffic analysis attack. The SSD error variance for the eMail dataset is on average less than $\sigma^2_{\text{err}} < 12$. Modelling the error as a Normal distribution this means that the 95% C.I. intervals would be within about 2 standard deviations of the mean, i.e. $\mu_{\text{err}} \pm \sigma_{\text{err}} = \mu_{\text{err}} \pm 6.9$ (or 3 standard deviation for a 99% C.I.). This empirical error indicates that for sending rates below about 7 messages per week the adversary will not reliably detect a communication, and that more frequent communications are susceptible to being detected. For the Location data set this heuristic threshold is around 18. We stress this is a very approximate rule of thumb – the confidence intervals resulting from the particle filter should provide a better estimate of the actual error of any specific trace under analysis, and Sect. 5.5 studies detection in more detail.

While Fig. 6 illustrates the performance of different models epoch by epoch, Fig. 7 illustrates the sum of square errors for a whole 16 epoch trace. The analysis is performed on the conversation trace set (Comm) and the silent trace

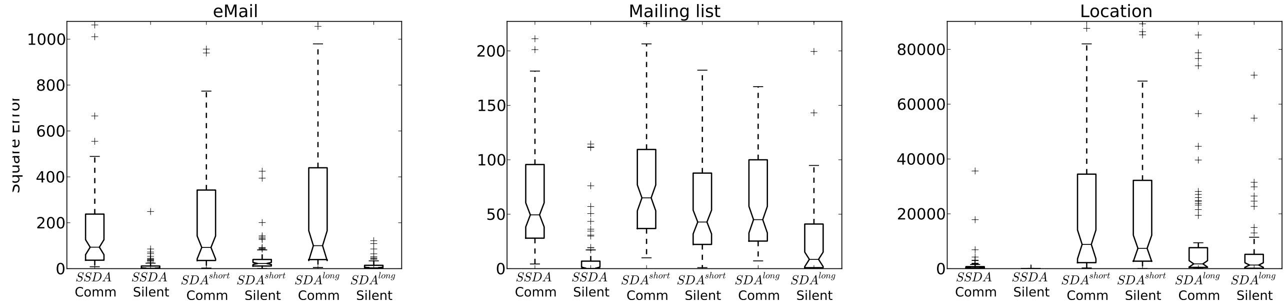

Figure 7: Square error between the rate prediction and actual rate of senders for whole traces of 16 weeks of the Sequential Statistical Disclosure compared with the traditional statistical disclosure over the short and long term. Communication and Silent traces.

set (Silent). We can qualitatively[7] say that lower per epoch error leads to lower per trace error for the SSD compared with the other models in the eMail and Location datasets. For the mailing list set the SSD model and long term SDA are equivalent for communication traces, but in all datasets the SSD models outperform the SDA for silent traces. Interestingly, the noise of the short term SDA results in high error for both the conversation and silent trace set. This is troubling, as it would lead an analyst to mistakenly believe some conversation took place within the period. The long term SDA does suffers less from false positives against silent traces.

5.5 Communication detection

Sect. 5.4 illustrates the square error of the SSD model and the SDA models in infering the rate of selected communication and silent trace sets. Yet the most important factor to determine the practicality of an attack is its susceptibility to false positives, namely epochs when no messages are send from Alice to Bob, that we nevertheless classify as containing communications. As all the datasets suggest the prior probability that any communication takes place is extremely low, and even a low false positive rate is likely to produce a volume of mistaken patterns larger than the real patterns.

We study the true positive and false positive performance of the SSD and SDA models using Receiver Operating Characteristics (ROC) curves. An ROC curve plots the performance of a binary classifier in terms of the trade-off that it can achieve between true positive rate and false positive rate. The true positive rate denotes the fraction of positives that are classified as positives, and the false positive rate denotes the fraction of negatives that have falsely been categorized as positive. We turn both the SSD and SDA models into binary classifiers to detect in each epoch whether the rate between specific a sender and receiver was above a threshold. For the eMail and Mailing list data set we chose this threshold to be $\lambda_{AB} \geq 5$ and for the location dataset $\lambda_{AB} \geq 10$, slightly lower than the heuristic based on the square error observed (see previous section). The particle filter implementing the SSD inference profiles a distribution over the inferred rate, that allows us to calculate a probability $\Pr[\lambda_{AB} \geq 5] = \gamma$. We can classify an epoch as positive if γ exceeds some threshold, and negative otherwise. Modulating this threshold provides a different trade off between true positive rates and false positive rates. The SDA models do not provide a measure of certainty, so we use the actual estimated rate as the feature, and categorize as positive or negative traces according to a threshold on the estimated rate.

Fig. 8 summarises the trade-off between true positive and false positive rates for all attacks against all datasets. The results were obtained through the analysis of 200 traces of 16 weeks each for each dataset. Note that the x-axis denoting false negatives has been cropped at 0.25 to better illustrate the performance for low false positives.

We observe that the SSD model outperforms the SDA models for in the extremely low false positive region. In fact in the eMail and Location datasets it achieves a significant rate of detection of true positives (about 30% and 20% respectively) at less than 1% false positive rate. The high performance in the eMail dataset may not come as a surprise, due to the relatively small batch sizes; the SDA models also offer good performance, for slightly larger rates of false positives. What is extremely surprising is the good performance against the Location dataset, where batch sizes number thousands of messages. In this set the relatively high performance could be explained by observing that a number of locations are only reported by a single sender. This hypothesis is supported by the relatively poor performance of all models against the Mailing List dataset, where a specific mailing list is expected to receive considerable volume of traffic by multiple senders.

We conclude that in terms of certainty the SSD model outperforms the traditional SDA models, and its estimates of error are useful indicators of the quality of the inference and the expected false positives. A higher level conclusion relates to the quality of protection that can be expected from an anonymity system: the literature has so far concentrated on measures such as batch size, but in fact we demonstrate that the nature of the high level traffic patterns can have a profound effect on the ability to de-anonymize a large fraction of relationships. The Mailing list and eMail datasets have comparable batch sizes, yet the quality of protection they lead to is quite different; while the Location dataset leads to enormous batch sizes, the quality of protection for a sizeable number of higher volume senders is pretty poor.

6. CONCLUSION

The Disclosure Attack [1] first illustrated that despite the use of an anonymity system an adversary can infer users'

[7] The relatively low sample of full traces we analyzed does not allow us yet to state these with greater certainty.

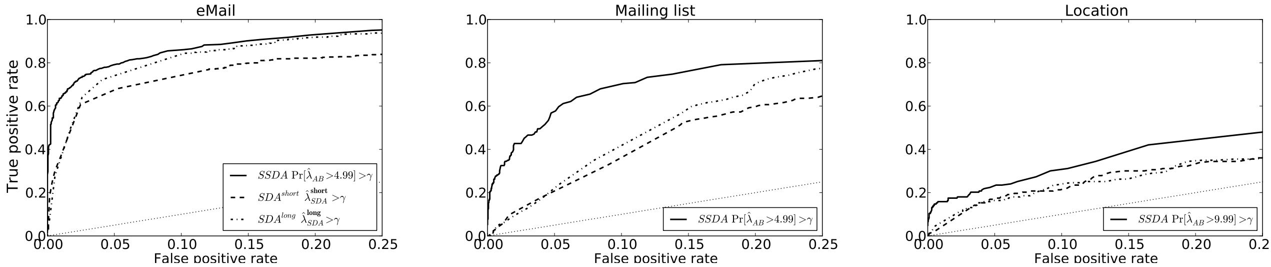

Figure 8: ROC curves for the detection performance for the three attacks (eMail and Mailing list: $\lambda_{AB} > 5$, and Location: $\lambda_{AB} > 10$).

communication profiles. The literature since predicated the success of these attacks on observations over long periods of profiles that are stable. Our results show that de-anonymization risks on real-world patterns of behaviour materialize even when using anonymity systems for a relatively short time, and despite profiles constantly evolving.

The sequential statistical disclosure (SSD) allows a traffic analyst to systematically combine information from successive epochs to track behaviour changes despite of the anonymity system. It is rooted in well-understood Bayesian inference techniques, and yields profile estimates alongside reliable information about their validity. These can be used to assess the quality of the inference, and in particular distinguish the existence of a communication patterns with lower false positives than previously expected.

The structured approach of defining a model, and then a particle filter based inference engine to estimate its hidden parameters offers a lot of flexibility: tracking additional hidden values simply involves augmenting the state of particles, and making use of additional side information involves modifying the priors or likelihood function. A better model of how profiles evolve over time can also be used straightforwardly.

We have evaluated the SSD against three real-world data sets of different nature, comparing it to state-of-the-art disclosure attacks. Our method outperforms previous proposals both at detecting the existence of communication and at quantifying its intensity. The key to this success is to correctly model real traffic transitions between communication and silent periods, as well as carefully take into account the prior rate of communications to tune the detector. We show that in the presence of enough evidence (volumes larger than 5 messages per week for the eMail and Mailing list traffic, and 10 for the Location dataset) the sequential statistical inference identifies a significant fraction of conversing users with high accuracy.

Thus the modelling assumptions that disclosure attacks introduced inadvertently mislead the community to believe that any attacks would be less effective on real-traffic than what our experiment demonstrate: they can be effective for rather low action rates; they are effective over a much shorter period of time than previously thought; and they can be effective against system configurations previously considered secure (e.g., in the Location dataset batches' sizes vary between 6400 and 26600 messages). These observation should motivate the study of anonymizing systems under real-world patterns of use, and a re-examination of key figures of merit such as the size of batches that have been used in the past as proxies for the security offered.

7. REFERENCES

[1] Dakshi Agrawal and Dogan Kesdogan. Measuring anonymity: The disclosure attack. *IEEE Security & Privacy*, 1(6):27–34, 2003.

[2] Oliver Berthold, Andreas Pfitzmann, and Ronny Standtke. The disadvantages of free mix routes and how to overcome them. In Hannes Federrath, editor, *Workshop on Design Issues in Anonymity and Unobservability*, volume 2009 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2000.

[3] Nikita Borisov and Ian Goldberg, editors. *Privacy Enhancing Technologies, 8th International Symposium, PETS 2008, Leuven, Belgium, July 23-25, 2008, Proceedings*, volume 5134 of *Lecture Notes in Computer Science*. Springer, 2008.

[4] David Chaum. Untraceable electronic mail, return addresses, and digital pseudonyms. *Commun. ACM*, 24(2):84–88, 1981.

[5] Peter Chien and Dan Boneh. Privately calculating location statistics. On-line at `http://crypto.stanford.edu/locpriv/privstats/index.php`, 2011.

[6] George Danezis. Statistical disclosure attacks. In Dimitris Gritzalis, Sabrina De Capitani di Vimercati, Pierangela Samarati, and Sokratis K. Katsikas, editors, *SEC*, volume 250 of *IFIP Conference Proceedings*, pages 421–426. Kluwer, 2003.

[7] George Danezis, Roger Dingledine, and Nick Mathewson. Mixminion: Design of a type iii anonymous remailer protocol. In *IEEE Symposium on Security and Privacy*, pages 2–15. IEEE Computer Society, 2003.

[8] George Danezis and Carmela Troncoso. Vida: How to use bayesian inference to de-anonymize persistent communications. In Ian Goldberg and Mikhail J. Atallah, editors, *Privacy Enhancing Technologies*, volume 5672 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2009.

[9] Claudia Díaz, Carmela Troncoso, and Andrei Serjantov. On the impact of social network profiling on anonymity. In Borisov and Goldberg [3], pages 44–62.

[10] Roger Dingledine, Nick Mathewson, and Paul F. Syverson. Tor: The second-generation onion router. In *USENIX Security Symposium*, pages 303–320. USENIX, 2004.

[11] Morris Dworkin. Cryptographic protocols of the identity mixer library, v. 2.3.0. IBM research report RZ3730, IBM Research, 2010.

http://domino.research.ibm.com/library/cyberdig.nsf/index.html.

[12] David M. Goldschlag, Michael G. Reed, and Paul F. Syverson. Hiding routing information. In Ross J. Anderson, editor, *Information Hiding*, volume 1174 of *Lecture Notes in Computer Science*, pages 137–150. Springer, 1996.

[13] Philippe Golle. Revisiting the uniqueness of simple demographics in the us population. In *Proceedings of the 5th ACM workshop on Privacy in electronic society*, pages 77–80. ACM, 2006.

[14] Louis C. Guillou and Jean-Jacques Quisquater, editors. *Advances in Cryptology - EUROCRYPT '95, International Conference on the Theory and Application of Cryptographic Techniques, Saint-Malo, France, May 21-25, 1995, Proceeding*, volume 921 of *Lecture Notes in Computer Science*. Springer, 1995.

[15] Dogan Kesdogan, Dakshi Agrawal, Dang Vinh Pham, and Dieter Rautenbach. Fundamental limits on the anonymity provided by the mix technique. In *IEEE Symposium on Security and Privacy*, pages 86–99. IEEE Computer Society, 2006.

[16] Dogan Kesdogan and Lexi Pimenidis. The hitting set attack on anonymity protocols. In Jessica J. Fridrich, editor, *Information Hiding*, volume 3200 of *Lecture Notes in Computer Science*, pages 326–339. Springer, 2004.

[17] Simon Maskell and Neil Gordon. A tutorial on particle filters for on-line nonlinear/non-gaussian bayesian tracking. In *Target Tracking: Algorithms and Applications (Ref. No. 2001/174), IEE*, pages 2–1. IET, 2001.

[18] Nick Mathewson and Roger Dingledine. Practical traffic analysis: Extending and resisting statistical disclosure. In David Martin and Andrei Serjantov, editors, *Privacy Enhancing Technologies*, volume 3424 of *Lecture Notes in Computer Science*, pages 17–34. Springer, 2004.

[19] Arvind Narayanan and Vitaly Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Symposium on Security and Privacy*, pages 111–125. IEEE Computer Society, 2008.

[20] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. In *IEEE Symposium on Security and Privacy*, pages 173–187. IEEE Computer Society, 2009.

[21] Fernando Pérez-González and Carmela Troncoso. Understanding statistical disclosure: A least squares approach. In Simone Fischer-Hübner and Matthew Wright, editors, *Privacy Enhancing Technologies*, volume 7384 of *Lecture Notes in Computer Science*, pages 38–57. Springer, 2012.

[22] Dang Vinh Pham, Joss Wright, and Dogan Kesdogan. A practical complexity-theoretic analysis of mix systems. In Vijay Atluri and Claudia Díaz, editors, *ESORICS*, volume 6879 of *Lecture Notes in Computer Science*, pages 508–527. Springer, 2011.

[23] Branko Ristic, Sanjeev Arulampalam, and Neil Gordon. *Beyond the Kalman filter: Particle filters for tracking applications*. Artech House Publishers, 2004.

[24] Latanya Sweeney. Simple demographics often identify people uniquely. *Health (San Francisco)*, pages 1–34, 2000.

[25] The 23rd Raccoon. How i learned to stop ph34ring nsa and love the base rate fallacy. The Tor project email archive. URL: http://archives.seul.org/or/dev/Sep-2008/msg00016.html, September 28 2008.

[26] Carmela Troncoso, Benedikt Gierlichs, Bart Preneel, and Ingrid Verbauwhede. Perfect matching disclosure attacks. In Borisov and Goldberg [3], pages 2–23.

[27] Gilbert Wondracek, Thorsten Holz, Engin Kirda, and Christopher Kruegel. A practical attack to de-anonymize social network users. In *IEEE Symposium on Security and Privacy*, pages 223–238. IEEE Computer Society, 2010.

APPENDIX

A. TIMED MIX SDA ALGORITHM

```
function TimedMixSDA($V_{A,O,B,O'}^{(t,n)}$)
    $\hat{u}_O, \hat{u}_{OB}, \hat{l}_{AB} \leftarrow 0, 0, 0$
    for all $V_{A,O,B,O'}^{(t,n)}$ do
        $T \leftarrow V_A^{(t,n)} + V_O^{(t,n)}$                 ▷ $T \equiv$ Batch size.
        $\hat{u}_O \leftarrow \hat{u}_O + V_O^{(t,n)}$
        $\hat{u}_{OB} \leftarrow \hat{u}_{OB} + V_O^{(t,n)} \cdot V_B^{(t,n)} / T$
        if $V_A^{(t,n)} > 0$ then
            $\hat{p}_{OB} \leftarrow \hat{u}_{OB} / \hat{u}_O$
            $\hat{l}_{AB} \leftarrow \hat{l}_{AB} + V_B^{(t,n)} - \hat{p}_{OB} \cdot V_O^{(t,n)}$
        end if
    end for
    return $\hat{l}_{AB}$        ▷ Estimated rate from Alice to Bob.
end function
```