How (Not) to Apply Differential Privacy to Anonymity Networks

Scott Coull
scott.coull@redjack.com
Differential Privacy

The gold standard in privacy definitions

• Some popular interpretations:
  – No assumptions on the data
  – Protection even when attacker knows all but one of the rows in the database
  – Robust to arbitrary background knowledge
The gold standard in privacy definitions

misinterpretations

• Some popular interpretations:
  – No assumptions on the data
  – Protection even when an attacker knows all but one of the rows in the database
  – Robust to arbitrary background knowledge
Many “objects” in network data

Many relationships among the “objects”

Relationships specified by underlying protocols and hidden processes

Implicit independence assumption
Objects in Computer Networks

Objects include users, workstations, web pages, emails, flows, packets ...

Protocols impose structure on network data and govern interaction of objects

One way to represent our understanding of the structure via an ontology
Objects in Computer Networks

- Group
  - Alice
  - Bob
  - Eve
  - Email
  - Web Page
  - P2P
    - HTML
    - Image
      - TCP Session
        - Packet
      - Video
        - TCP Session
          - Packet
        - TCP Session
          - Packet
Objects in Computer Networks

Many such ontologies may exist

May not even be a hierarchy

Objects carry information about both ancestors and descendants

Adversary may have a more complete (complex) “view” than we do
Independence Assumptions

- Differential privacy developed in the context of databases of individuals

- What happens when records imply each other’s presence?
  - Nothing good
Independence Assumptions

• No Free Lunch in Data Privacy\(^1\)
  – Knowledge of correlations leads to failure
  – Example: Remove edge from a social network graph; growth pattern changes significantly

Independence Assumptions

In contrast to the forest fire and copying models, this model is actually an ergodic Markov chain (whose state space consists of graphs) and has a steady state distribution. Thus the number of edges does not have a tendency to increase over time as with the other two models. The original MVS model is a continuous time process, which we discretize in a way similar to [24]. In this model, the number of nodes remains constant. At each timestep, with probability $\xi$, a link is added between two random nodes. With probability $\xi$, a randomly selected node selects one of its neighbors at random and asks for an introduction to one of its neighbors. Finally $k$ edges are deleted, where $k$ is a random variable drawn from a Poisson($\lambda$) distribution. Initially both communities have an equal number of nodes and community membership does not change. Initially each node has 2 within community links. We perform two sets of experiments. In the first set of experiments, the network has 20 nodes and we vary $\xi$, the friend-introduction probability. We compute the number of cross-community edges after 100 time steps (before the network has reached a steady state). This is shown in Figure 4. In the second set of experiments, we set the network size to be 20, we set $\xi = 1$, and we vary the number of iterations to let the network achieve its steady state. This is shown in Figure 5. Note that once the process achieves steady state, the initial conditions (or, more generally, network configurations in the distant past) are irrelevant and the evidence that other edges provide about the participation of Bob's initial link is automatically destroyed by the network model itself. However, there can still be evidence about edges formed in the not-so-distant past.

We see from Figure 4, that depending on network parameters, the expected difference in cross-community edges can be moderately large, but we see that this difference disappears as the network is allowed to reach its steady state (Figure 5). Thus differential privacy may be a reasonable choice for limiting inference about the participation of Bob's edge in such a scenario. However, applying differential privacy here still requires the assumption that the data were generated...
Independence Assumptions

- Same problems with network data...
  - Ontology describes exactly the correlations we can use to infer whether object is present
  - Example: Remove a TCP handshake packet; total traffic volume should change

- If we ignore these semantics the best we get is effectively packet privacy
Independence Assumptions
Independence Assumptions
The Pufferfish Framework

- Generalization of differential privacy that explicitly states assumptions:
  - Objects we are trying to protect – \( \mathcal{S} \)
  - Mutually exclusive secret pairs
    \[ \mathcal{S}_{pairs} \subset \mathcal{S} \times \mathcal{S} \]
  - Set of data generating distributions – \( \mathcal{D} \)

The Pufferfish Framework

- For all possible outputs $\omega \in \text{range}(M)$
- For all pairs of potential secrets $(s_i, s_j) \in S_{pairs}$
- For all distributions $\theta \in \mathbb{D}$

\[
P(M(Data) = \omega | s_i, \theta) \leq e^\epsilon P(M(Data) = \omega | s_j, \theta)
\]
\[
P(M(Data) = \omega | s_j, \theta) \leq e^\epsilon P(M(Data) = \omega | s_i, \theta)
\]
The Pufferfish Framework

- Also supports a nice semantic interpretation of the definition:

\[
e^{-\epsilon} \leq \frac{P(s_i | M(Data) = \omega, \theta)}{P(s_j | M(Data) = \omega, \theta)} \cdot \frac{P(s_i | \theta)}{P(s_j | \theta)} \leq e^{\epsilon}
\]

- Adversary’s belief in \( s_i \) changes to at most \( e^{\epsilon} \alpha \) and at least \( e^{-\epsilon} \alpha \)
Challenges

- How do we define the data generation distributions?
  - Network data is notoriously difficult to model
  - Is it possible to design the protocol to make this easier for us?
  - G. Danezis et al.’s MCMC sampling?
Challenges

• How do we define the private algorithm (M) for anonymity networks?
  – For general network data we are probably out of luck since there is just so much to measure
  – For encrypted traffic we have a more restrictive set of measurements – time, size
Final Thoughts

• Can we think about multiple attacker models at once, in a single metric?
  – Yes, and we probably have to if we want to move away from the break-fix cycle.
  – I think Pufferfish accommodates for this in the secrets and data generating distributions.
Final Thoughts

- Are Bayesian approaches used for mixes extensible to Tor-like systems?
  - Yes...probably...maybe?
  - A Bayesian view on the problem is likely to yield semantically-meaningful guarantees.
  - Pufferfish definitions have a well-defined Bayesian interpretation that actually tells us something useful.
Should differential privacy be an inspiration for this space?

- Most other commonly used definitions, like k-anonymity, are vulnerable to direct attack.
- Differential privacy at least offers the possibility of a strong guarantee.
- Applying it to anonymity networks is going to be tricky due to difficulty in modeling the data.
Final Thoughts

- What is the role of user modeling in this space?
  - Seems to be very important.
  - Some model is going to be necessary to make reasonable assumptions for definitions.
  - It is not just about user modeling, but modeling general relationships in the data.