Differential Privacy and its Application in Aggregation

Part 2 — Privacy-preserving Aggregation

presenter: Le Chen

Nanyang Technological University

lechen0213@gmail.com

October 12, 2013

Outline

Reference

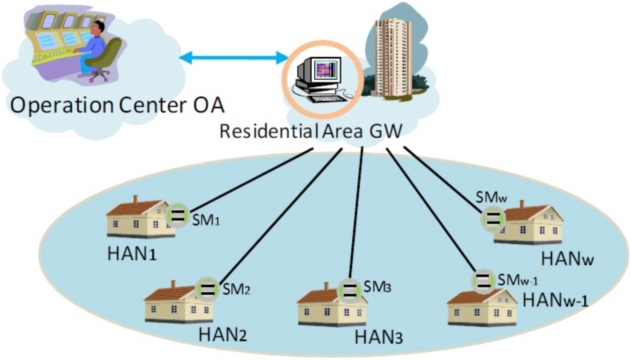
- Cynthia Dwork.
  Differential Privacy.
  Invited talk at ICALP, Venice, Italy, July 10-14, 2006.
  Automata, Languages and Programming, Lecture Notes in Computer Science Volume 4052, 2006, pp 1-12.
- Wikipedia.
  <http://en.wikipedia.org/wiki/Differential_privacy>
- Elaine Shi, T-H. Hubert Chan, Eleanor Rieffel, Richard Chow and Dawn Song.
  Privacy-Preserving Aggregation of Time-Series Data.
  In Network and Distributed System Security Symposium (NDSS), 2011.

Reference

T-H. Hubert Chan, Elaine Shi, and Dawn Song
Privacy-Preserving Stream Aggregation with Fault Tolerance.
16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012. Financial Cryptography and Data Security, Lecture Notes in Computer Science Volume 7397, 2012, pp 200-214.

Motivation of Aggregation

- Statistics: In many practical applications, a data aggregator wishes to mine data coming from multiple organizations or individuals, to study patterns or statistics over a population.



Motivation of Aggregation

- Aggregator:
  - No aggregator.
  - Structure based aggregator.
  - Third-party aggregator.
- Advantage: Communication and computation overhead can be significantly decreased.
- Protect individual privacy:
  - Masking value.
  - Distributed differential privacy.

Two Techniques

- Basic construction - masking value.
- Distributed differential privacy.

Basic Construction - Masking Value

- A trusted dealer chooses a random generator $g \in \mathbb{G}$, and $n+1$ random secrets $s_0, s_1, \cdots, s_n \in \mathbb{Z}_p$ such that $s_0 + s_1 + s_2 + \cdots + s_n = 0$.
- The aggregator obtains the capability $sk_0 := s_0$, and participant i obtains the secret key $sk_i := s_i$.
- NoisyEnc: $c \leftarrow g^{\hat{x}} H(t)^{sk_i}$, where $\hat{x} = x + r \mod p$.
- AggrDec: $V = H(t)^{s_0} \prod_{i=1}^{n} c_i = g^{\sum_{i=1}^{n} \hat{x}_i}$.

Basic Construction - Masking Value

- A trusted dealer chooses a random generator $g \in \mathbb{G}$, and $n+1$ random secrets $s_0, s_1, \cdots, s_n \in \mathbb{Z}_p$ such that $s_0 + s_1 + s_2 + \cdots + s_n = 0$.
- The aggregator obtains the capability $sk_0 := s_0$, and participant i obtains the secret key $sk_i := s_i$.
- NoisyEnc: $c \leftarrow g^{\hat{x}} H(t)^{sk_i}$, where $\hat{x} = x + r \mod p$.
- AggrDec: $V = H(t)^{s_0} \prod_{i=1}^{n} c_i = g^{\sum_{i=1}^{n} \hat{x}_i}$.

Basic Construction - Masking Value

- A trusted dealer chooses a random generator $g \in \mathbb{G}$, and $n+1$ random secrets $s_0, s_1, \cdots, s_n \in \mathbb{Z}_p$ such that $s_0 + s_1 + s_2 + \cdots + s_n = 0$.
- The aggregator obtains the capability $sk_0 := s_0$, and participant i obtains the secret key $sk_i := s_i$.
- NoisyEnc: $c \leftarrow g^{\hat{x}} H(t)^{sk_i}$, where $\hat{x} = x + r \mod p$.
- AggrDec: $V = H(t)^{s_0} \prod_{i=1}^{n} c_i = g^{\sum_{i=1}^{n} \hat{x}_i}$.

Basic Construction - Masking Value

- A trusted dealer chooses a random generator $g \in \mathbb{G}$, and $n+1$ random secrets $s_0, s_1, \cdots, s_n \in \mathbb{Z}_p$ such that $s_0 + s_1 + s_2 + \cdots + s_n = 0$.
- The aggregator obtains the capability $sk_0 := s_0$, and participant i obtains the secret key $sk_i := s_i$.
- NoisyEnc: $c \leftarrow g^{\hat{x}} H(t)^{sk_i}$, where $\hat{x} = x + r \mod p$.
- AggrDec: $V = H(t)^{s_0} \prod_{i=1}^{n} c_i = g^{\sum_{i=1}^{n} \hat{x}_i}$.

Basic Construction - Masking Value

- To decrypt the sum $\sum_{i=1}^{n} \hat{x}_i$, it suffices to compute the discrete log of V base g.
- When the plaintext space is small, decryption can be achieved through a brute-force search.
- A better approach is to use Pollard's lambda method which requires decryption time roughly square root in the plaintext space.

Basic Construction - Masking Value

- To decrypt the sum $\sum_{i=1}^{n} \hat{x}_i$, it suffices to compute the discrete log of V base g.
- When the plaintext space is small, decryption can be achieved through a brute-force search.
- A better approach is to use Pollard's lambda method which requires decryption time roughly square root in the plaintext space.

Basic Construction - Masking Value

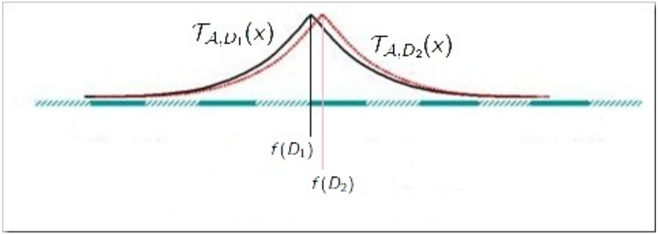
- To decrypt the sum $\sum_{i=1}^{n} \hat{x}_i$, it suffices to compute the discrete log of V base g.
- When the plaintext space is small, decryption can be achieved through a brute-force search.
- A better approach is to use Pollard's lambda method which requires decryption time roughly square root in the plaintext space.

Differential Privacy — Review

- ϵ-differential privacy ($\epsilon = \frac{\Delta(f)}{\lambda}$):

$$\Pr[\mathcal{A}(D_1) \in S] \leq e^{\epsilon} \times \Pr[\mathcal{A}(D_2) \in S],$$

- Laplace noise:



Motivation

- In previous differential privacy literature, a trusted aggregator is responsible for adding an appropriate magnitude of noise before releasing the statistics.
- Our approach is to let the participants add noise to their data before encrypting them (distributed).
- One naive solution is to rely on a single participant to add an appropriate magnitude of noise r to her data before submission.

Motivation

- In previous differential privacy literature, a trusted aggregator is responsible for adding an appropriate magnitude of noise before releasing the statistics.
- Our approach is to let the participants add noise to their data before encrypting them (distributed).
- One naive solution is to rely on a single participant to add an appropriate magnitude of noise r to her data before submission.

Motivation

- In previous differential privacy literature, a trusted aggregator is responsible for adding an appropriate magnitude of noise before releasing the statistics.
- Our approach is to let the participants add noise to their data before encrypting them (distributed).
- One naive solution is to rely on a single participant to add an appropriate magnitude of noise r to her data before submission.

Compromised Participants

- In particular, a subset of the participants may be compromised and collude with the data aggregator.
- In the worst case, if every participant believes that the other $n-1$ participants may be compromised and collude with the aggregator, each participant would need to add sufficient noise to ensure the privacy of her own data.

Compromised Participants

- In particular, a subset of the participants may be compromised and collude with the data aggregator.
- In the worst case, if every participant believes that the other $n-1$ participants may be compromised and collude with the aggregator, each participant would need to add sufficient noise to ensure the privacy of her own data.

Compromised Participants

- If at least γ fraction of the participants are honest and not compromised, then we can distribute the noise generation task amongst these participants. Each participant may add less noise, and as long as the noise in the final statistic is large enough, individual privacy is protected.

- Our scheme assumes that the participants have an a priori estimate on the lower bound for γ.

Compromised Participants

- If at least γ fraction of the participants are honest and not compromised, then we can distribute the noise generation task amongst these participants. Each participant may add less noise, and as long as the noise in the final statistic is large enough, individual privacy is protected.
- Our scheme assumes that the participants have an a priori estimate on the lower bound for γ.

Algebraic Constraints

- Most encryption schemes require that the plaintext be picked from a group comprised of discrete elements.
- We choose to use a symmetric geometric distribution instead of the more commonly used Laplace distribution.

Algebraic Constraints

- Most encryption schemes require that the plaintext be picked from a group comprised of discrete elements.
- We choose to use a symmetric geometric distribution instead of the more commonly used Laplace distribution.

Symmetric Geometric Distribution

Definition:

- Let $\alpha > 1$. We denote by $\mathsf{Geom}(x, \alpha)$ the symmetric geometric distribution that takes integer values x such that the probability mass function at x is $\frac{\alpha-1}{\alpha+1} \cdot \alpha^{-|x|}$.
- The symmetric geometric distribution $\mathsf{Geom}(x, \alpha)$ can be viewed as a discrete version of the Laplace distribution Lap(x, λ) (where $\alpha \approx \exp(\frac{1}{\lambda})$), whose probability density function is $x \mapsto \frac{1}{2\lambda} \exp(-\frac{|x|}{\lambda})$.

Symmetric Geometric Distribution

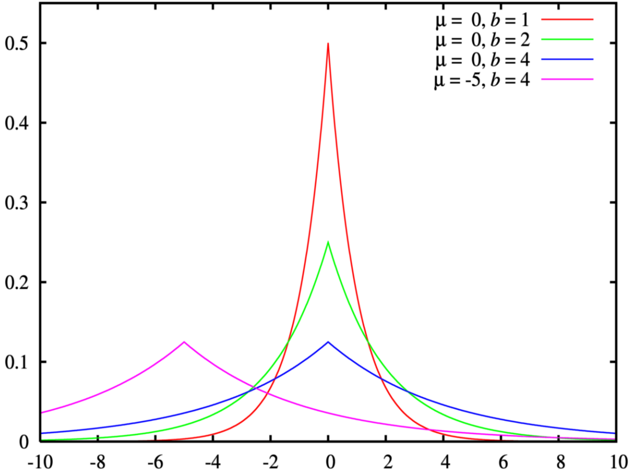
Definition:

- Let $\alpha > 1$. We denote by $\text{Geom}(x, \alpha)$ the symmetric geometric distribution that takes integer values x such that the probability mass function at x is $\frac{\alpha-1}{\alpha+1} \cdot \alpha^{-|x|}$.
- The symmetric geometric distribution $\text{Geom}(x, \alpha)$ can be viewed as a discrete version of the Laplace distribution Lap(x, λ) (where $\alpha \approx \exp(\frac{1}{\lambda})$), whose probability density function is $x \mapsto \frac{1}{2\lambda} \exp(-\frac{|x|}{\lambda})$.

Differential Privacy

- Let $\epsilon > 0$. Suppose u and v are two integers such that $|u - v| \leq \Delta$. Let r be a random variable having distribution $\text{Geom}(\alpha)$, where $\alpha \approx \exp(\frac{1}{\lambda}) = \exp(\frac{\epsilon}{\Delta})$. Then, for any integer k, $\Pr[u + r = k] \leq \exp(\epsilon) \cdot \Pr[v + r = k]$.

Laplace Distribution



Laplace Distribution Probability Density Function

Error

- Our mechanism ensures small error of roughly $O(\frac{\Delta}{\epsilon}\sqrt{\frac{1}{\gamma}})$ magnitude.

- Consider the extreme case when $\gamma = O(\frac{1}{n})$, i.e., each participant believes that all other participants may be compromised. Then, our accumulated noise would be $O(\frac{\Delta}{\epsilon}\sqrt{\frac{1}{\gamma}}) = O(\frac{\Delta}{\epsilon}\sqrt{n})$.

- According to the central limit theorem, the sum of n independent symmetric noises of magnitude $O(\frac{\Delta}{\epsilon})$ results in a final noise of magnitude $O(\frac{\Delta}{\epsilon}\sqrt{n})$ with high probability.

Error

- Our mechanism ensures small error of roughly $O(\frac{\Delta}{\epsilon}\sqrt{\frac{1}{\gamma}})$ magnitude.
- Consider the extreme case when $\gamma = O(\frac{1}{n})$, i.e., each participant believes that all other participants may be compromised. Then, our accumulated noise would be $O(\frac{\Delta}{\epsilon}\sqrt{\frac{1}{\gamma}}) = O(\frac{\Delta}{\epsilon}\sqrt{n})$.
- According to the central limit theorem, the sum of n independent symmetric noises of magnitude $O(\frac{\Delta}{\epsilon})$ results in a final noise of magnitude $O(\frac{\Delta}{\epsilon}\sqrt{n})$ with high probability.

Error

- Our mechanism ensures small error of roughly $O(\frac{\Delta}{\epsilon}\sqrt{\frac{1}{\gamma}})$ magnitude.
- Consider the extreme case when $\gamma = O(\frac{1}{n})$, i.e., each participant believes that all other participants may be compromised. Then, our accumulated noise would be $O(\frac{\Delta}{\epsilon}\sqrt{\frac{1}{\gamma}}) = O(\frac{\Delta}{\epsilon}\sqrt{n})$.
- According to the central limit theorem, the sum of n independent symmetric noises of magnitude $O(\frac{\Delta}{\epsilon})$ results in a final noise of magnitude $O(\frac{\Delta}{\epsilon}\sqrt{n})$ with high probability.

Distributed Differential Privacy

Definition ((ϵ, δ)-DD-Privacy)

- Suppose $\epsilon > 0$, $0 \leq \delta < 1$ and $0 < \gamma \leq 1$. We say the function f achieves (ϵ, δ)-distributed differential privacy (DD-privacy) under γ fraction of uncompromised participants if the following condition holds.

$$\Pr[f(\hat{x}) \in S] \leq \exp(\epsilon) \cdot \Pr[f(\hat{y}) \in S] + \delta.$$

Distributed Differential Privacy

Algorithm 1: DD-Private Data Randomization Procedure.

Let $\alpha := \exp(\frac{\epsilon}{\Delta})$ and $\beta := \frac{1}{\gamma n} \log \frac{1}{\delta}$.
Let $\mathbf{x} = (x_1, \ldots x_n)$ denote all participants' data in a certain time period.
foreach participant $i \in [n]$ **do**

Sample noise r_i according to the following distribution.

$$r_i \leftarrow \begin{cases} \mathsf{Geom}(\alpha) & \text{with probability } \beta \\ 0 & \text{with probability } 1 - \beta \end{cases}$$

Randomize data by computing $\widehat{x}_i \leftarrow x_i + r_i \mod p$.

- Lemma. Let $\epsilon > 0$ and $0 < \delta < 1$. Suppose at least γ fraction of participants are uncompromised. Then, the above randomization procedure achieves (ϵ, δ)-DD-privacy with respect to **sum**, for $\beta = \min\{\frac{1}{\gamma n} \log \frac{1}{\delta}, 1\}$

Parameters

- $\epsilon > 0$.
- $0 < \gamma \leq 1$.
- $0 < \delta < 1$.
- $\beta = \frac{1}{\gamma n} \log \frac{1}{\delta}$.
- Δ.
- $\alpha \approx \exp(\frac{1}{\lambda}) = \exp(\frac{\epsilon}{\Delta})$.

The Parameter ϵ

- $\epsilon > 0$.
- The privacy parameter.

The Parameter γ

- $0 < \gamma \leq 1$.
- The proportion of trusted participants.

The Parameter δ

- $0 < \delta < 1$.
- Another privacy parameter.

The Parameter β

- $\beta = \frac{1}{\gamma n} \log \frac{1}{\delta}$.
- The probability that a trusted participant generates a noise.

Relationship of γ, δ, and β

- $\beta = \frac{1}{\gamma n} \log \frac{1}{\delta}$.
- Given γ.
- Given δ.

The Parameter Δ

- $\Delta = \max\{|f(\hat{x}) - f(\hat{y})|\}$.
- The probability that a trusted participant generates a noise.

The Parameter α

- $\alpha \approx \exp(\frac{1}{\lambda}) = \exp(\frac{\epsilon}{\Delta})$.
- The magnitude of noise, the larger α is the smaller the noise is.

Relationship of α, and Δ

- $\alpha \approx \exp(\frac{1}{\lambda}) = \exp(\frac{\epsilon}{\Delta})$.
- The larger Δ is, the larger the noise is needed.

Confliction

- Can the basic construction extend to support distributed differential privacy?
- The basic construction needs all users to participate, or the masking value cannot be canceled.
- So it is impossible for the basic construction to support distributed differential privacy.

Confliction

- Can the basic construction extend to support distributed differential privacy?
- The basic construction needs all users to participate, or the masking value cannot be canceled.
- So it is impossible for the basic construction to support distributed differential privacy.

Confliction

- Can the basic construction extend to support distributed differential privacy?
- The basic construction needs all users to participate, or the masking value cannot be canceled.
- So it is impossible for the basic construction to support distributed differential privacy.

Conclusion & Discussion

- We introduced the basic construction that uses masking value and the distributed differential privacy.
- Achieving distributed differential privacy with small error is not easy.
- It also depends on the query situation.
- Discussion?

Conclusion & Discussion

- We introduced the basic construction that uses masking value and the distributed differential privacy.
- Achieving distributed differential privacy with small error is not easy.
- It also depends on the query situation.
- Discussion?

Conclusion & Discussion

- We introduced the basic construction that uses masking value and the distributed differential privacy.
- Achieving distributed differential privacy with small error is not easy.
- It also depends on the query situation.
- Discussion?

Conclusion & Discussion

- We introduced the basic construction that uses masking value and the distributed differential privacy.
- Achieving distributed differential privacy with small error is not easy.
- It also depends on the query situation.
- Discussion?