Differential Privacy (Part I)
Computing on personal data

Individuals have lots of interesting data

and we would like to compute on it
Which kind of data?
Which computations?

- statistical correlations
  - genotype/phenotype associations
  - correlating medical outcomes with risk factors or events
- aggregate statistics
  - web analytics
- identification of events/outliers
  - intrusion detection
  - disease outbreaks
- data-mining/learning tasks
  - use customers’ data to update strategies
Ok, but we can compute on anonymised data, i.e., not including personally identifiable information… that should be fine, right?
First person identified from AOL Data:
Thelma Arnold

Michael Arrington

Wednesday, August 9th, 2006

On Sunday the news broke that AOL purposefully released 20 million partially anonymized search queries. On Monday AOL apologized, and later that evening the first web interface to the data went up.

Today the first person was positively identified from the data – Thelma Arnold, a 62-year-old widow who lives in Lilburn, Georgia.

Based on searches ranging from "numb fingers" to "60 single men" to "dog that urinates on everything," the New York Times was able to quickly determine and confirm her identity. Ms Arnold is AOL searcher no. 4417749.

Ms Arnold commented: "My goodness, it's my whole personal life...I had no idea somebody was looking over my shoulder."

AOL replied: "We apologize specifically to her...There is not a whole lot we can do."

Tags: AOL
Netflix data

- De-anonymize Netflix data [A. Narayanan and V. Shmatikov, S&P’08]
  - Netflix released its database as part of $1 million Netflix Prize, a challenge to the world’s researchers to improve the rental firm’s movie recommendation system
  - **Sanitization**: personal identities removed
  - Problem, **sparsity of data**: with large probability, no two profiles are similar up to $\epsilon$. In Netflix data, no two records are similar more than 50%
  - If the profile can be matched up to 50% to a profile in IMDB, then the adversary knows with good chance the true identity of the profile
  - In this work, efficient random algorithm to break privacy
Personally identifiable information

What Information is “Personally Identifiable”?  

Mr. X lives in ZIP code 02138 and was born July 31, 1945.  

These facts about him were included in an anonymized medical record released to the public. Sounds like Mr. X is pretty anonymous, right?  

Not if you’re Latanya Sweeney, a Carnegie Mellon University computer science professor who showed in 1997 that this information was enough to pin down Mr. X’s more familiar identity -- William Weld, the governor of Massachusetts throughout the 1990s.  

Gender, ZIP code, and birth date feel anonymous, but Prof. Sweeney was able to identify Governor Weld through them for two reasons. First, each of these facts about an individual (or other kinds of facts we might not usually think of as identifying) independently narrows down the population, so much so that the combination of (gender, ZIP code, birthdate) was unique for about 87% of the U.S. population. If you live in the United States, there’s an 87% chance that you don’t share all three of these attributes with any other U.S. resident. Second, there may be particular data sources available (Sweeney used a Massachusetts voter registration database) that let people do searches to bootstrap what they know about someone in order to learn more -- including traditional identifiers like name and address. In a very concrete sense, “anonymized” or “merely demographic” information about people may be neither. (And a website that asks “anonymous” users for seemingly trivial information about themselves may be able to use that information to make a unique profile for an individual, or even look up that individual in other databases.)

Many contemporary privacy rules and debates center on the notion of “personally identifiable information” (PII). The PII concept is used by several legal regimes and many organizations’ privacy policies; generally, information that identifies a particular person is considered much more sensitive than information that does not. For instance,

- Federal telecommunications privacy laws use “individually identifiable information” (about a subscriber) as a basis for the category of protected information called Customer Proprietary Network Information (CPNI);
- Federal health privacy regulations use “individually identifiable health information” (about a patient) as a basis for the category called Protected Health Information (PHI);
- Federal financial privacy laws, the EU Data Protection Directive, and state privacy laws all employ similar terms and concepts;

and, in each case, facts deemed “personally identifiable” or “individually identifiable” may receive dramatically higher protections under these laws and regulations.

But research by Prof. Sweeney and other experts has demonstrated that surprisingly many facts, including those that seem quite innocuous, neutral, or “common”, could potentially identify an individual. Privacy law, mainly clinging to a traditional intuitive notion of identifiability, has largely not kept up with the technical reality.

https://www.eff.org/deeplinks/2009/09/what-information-personally-identifiable
From the Facebook privacy policy...

While you are allowing us to use the information we receive about you, you always own all of your information. Your trust is important to us, which is why we don't share information we receive about you with others unless we have:

- received your permission;
- given you notice, such as by telling you about it in this policy; or
- removed your name or any other personally identifying information from it.
Ok, but I do not want to release an entire dataset! I just want to compute some innocent statistics… that should be fine, right?
Learning Your Identity and Disease from Research Papers: Information Leaks in Genome Wide Association Study

Rui Wang, Yong Fuga Li, Xiaofeng Wang, Haixu Tang, Xiaoyong Zhou
Indiana University Bloomington
Bloomington, IN

Abstract
Genome-wide association studies (GWAS) aim at discovering the association between genetic variations, particularly single-nucleotide polymorphism (SNP), and common diseases, which is well recognized to be one of the most important and active areas in biomedical research. Also renowned is the privacy implication of such studies, which has been brought into the limelight by the recent attack proposed by Homer et al. Homer’s attack demonstrates that it is possible to identify a GWAS participant from the allele frequencies of a large number of SNPs. Such a threat, unfortunately, was found in our research to be significantly understated. In this paper, we show that individuals can actually be identified from even a relatively small set of statistics, as those routinely published in GWAS papers. We present two attacks. The first one extends Homer’s attack with a much more powerful test statistic, based on the correlations among different SNPs described by coefficient of determination ($r^2$). This attack can determine the presence of an individual from the statistics related to a couple of hundred SNPs. The second attack can lead to complete disclosure of hundreds of participants’ SNPs, through analyzing the information derived from published statistics. We also found that these attacks can succeed even when the precisions of the statistics are low and part of data is missing. We evaluated our attacks on the real human genomes and concluded that such threats are completely realistic.

of genome-wide association study (GWAS) [7], a study that aims at discovering the association between human genes and common diseases. To this end, GWAS investigators determined the genotypes of two groups of participants, people with a disease (cases) and similar people without (controls) in an attempt to use statistical testing to identify genetic markers, typically single-nucleotide polymorphisms (SNP), that are associated to disease susceptibility genes [46]. If the variation of a SNP is found to be significantly higher in the case group than that in the control group, it is reported as a potential marker of the disease. Of great importance to such a study is privacy of the participants, whose sensitive information, personally identifiable genetic markers in particular, should not be leaked out without explicit consent. So far, this has been enforced through an informed consent from participants [9] and an agreement from investigators to ensure proper use of data according to the consent. Unfortunately, while this process prevents explicit misuse of participants’ DNA data, it turns out to be insufficient for deterring information leaks in a more implicit way. Particularly, this paper reports a surprising finding of our research: even the test statistics computed over a small set of SNPs, like those routinely published in GWAS papers, could reveal a substantial amount of genetic information about participants, and even lead to disclosure of their identities.

The inadequacy of privacy protection in current genome research has also been pointed out by other researchers. For example, Ma-
Database privacy

- Ad hoc solutions do not really work
- We need to formally reason about the problem...
What does it mean for a query to be privacy-preserving and how can we achieve that?
Blending into a crowd

- Intuition: “I am safe in a group of $k$ or more”
  - $k$ varies (3...6...100...10,000?)
- Why?
  - Privacy is “protection of being brought to the attention of others” [Gavison]
  - Rare property helps re-identify someone
Clustering-based definitions

- **k-anonymity**: attributes are suppressed or generalized until each row is identical to at least k-1 other rows.
  - At this point the database is said to be k-anonymous.
- Methods for achieving k-anonymity
  - **Suppression** - can replace individual attributes with a *
  - **Generalization** - replace individual attributes with a broader category (e.g., age 26 \(\Rightarrow\) age \([26-30]\))

- Purely syntactic definition of privacy
- What adversary does it apply to?
  - Does not consider adversaries with side information
  - Does not consider adversarial algorithm for making decisions (inference)
- Almost abandoned in the literature...
Notations

$Q(D_i) = R$

$Q$ is the privatized query run on the data set and $R$ is the result released to the public.
What do we want?

- I would feel safe participating in the dataset if
  - I knew that my answer had no impact on the released results
  - I knew that any attacker looking at the published results R couldn’t learn (with any high probability) any new information about me personally [Dalenius 1977]
  - Analogous to semantic security for ciphertexts
  - \( Q(D_{(l-me)}) = Q(D_l) \)
  - \( \text{Prob}(\text{secret}(me) \mid R) = \text{Prob}(\text{secret}(me)) \)
Why can’t we have it?

- If individuals had no impact on the released results...then the results would have no utility!

- If R shows there is a strong trend in the dataset (everyone who smokes has a high risk of cancer), with high probability, that trend is true for any individual. Even if she does not participate in the dataset, it is just enough to know that she smokes!

Achieving either privacy or utility is easy, getting a meaningful trade-off is the real challenge!

\[ Q(D_I) = Q(D_\emptyset) \]

- \[ \text{Prob}(\text{secret}(\text{me}) | R) > \text{Prob}(\text{secret}(\text{me})) \]
Even worse, if an attacker knows a function about me that’s dependent on general facts about the population:

- I am twice the average age
- I am in the minority gender

Then releasing just those general facts gives the attacker specific information about me. (Even if I don’t submit a survey!)

\[
\begin{align*}
\text{age}(\text{me}) &= 2 \times \text{mean}_\text{age} \\
\text{gender}(\text{me}) &\neq \text{top}_\text{gender} \\
\text{mean}_\text{age} &= 14 \\
\text{top}_\text{gender} &= F \\
\text{age}(\text{me}) &= 28 \\
\text{gender}(\text{me}) &= M
\end{align*}
\]
Impossibility result (informally)

- **Tentative definition:**
  
  For some definition of “privacy breach”,
  \( \forall \) distributions on databases,
  \( \forall \) adversaries \( A \),
  \( \exists A' \) such that
  
  \[
  \Pr(A(\text{San}(DB)) = \text{breach}) - \Pr(A'(\cdot) = \text{breach}) \leq \epsilon
  \]

- **Result:** for reasonable “breach”, if San(DB) contains information about DB, we can find an adversary that breaks this definition
Proof sketch (informally)

- Suppose DB is drawn uniformly random
- “Breach” is predicting a predicate $g(DB)$
- Adversary knows $H(DB)$, $H(H(DB) ; San(DB)) \oplus g(DB)$
  - $H$ is a suitable hash function
- By itself, the attacker’s knowledge does not leak anything about DB
- Together with $San(DB)$, it reveals $g(DB)$
Disappointing fact

- We can’t promise my data won’t affect the results
- We can’t promise that the attacker won’t be able to learn new information about me, given proper background information

What can we do?
One more try...

The chance that the sanitised released result will be $R$, is nearly the same whether or not I submitted my personal information.
Differential privacy

- Proposed by Cynthia Dwork in 2006
- **Intuition**: perturb the result (e.g., by adding noise) such that the chance that the perturbed result will be $C$ is nearly the same, whether or not you submit your info
- **Challenge**: achieve privacy while minimising the utility loss
A query mechanism $M$ is $\epsilon$-differentially private if, for any two adjacent databases $D$ and $D'$ (differing in just one entry) and $C \subseteq \text{range}(M)$

$$\Pr(M(D) \in C) \leq e^\epsilon \cdot \Pr(M(D') \in C)$$
Sequential composition theorem

Let $M_i$ each provide $\epsilon_i$-differential privacy. The sequence of $M_i(X)$ provides $(\sum_i \epsilon_i)$-differential privacy.

- **Privacy losses sum up**
- **Privacy budget** = maximum tolerated privacy loss
- If the privacy budget is exhausted, then the server administrator acts according to the policy
  - answers the query and reports a warning
  - does not answer further queries
Sequential composition theorem

Let $M_i$ each provide $\epsilon_i$-differential privacy. The sequence of $M_i(X)$ provides $(\sum_i \epsilon_i)$-differential privacy.

- Result holds against active attacker (i.e., each query depends on the previous ones’ result)
- Result proved for a generalized definition of differential privacy [McSharry, Sigmod’09]
- ⊕ denotes symmetric difference

A query mechanism $M$ is differentially private if, for any two databases $D$ and $D'$ and $C \subseteq range(M)$

$$\Pr(M(D) \in C) \leq e^{\epsilon \cdot |D \oplus D'|} \cdot \Pr(M(D') \in C)$$
Parallel composition theorem

Let $M_i$ each provide $\epsilon$-differential privacy. Let $D_i$ be arbitrary disjoint subsets of the input domain $\mathcal{D}$. The sequence of $M_i(X \cap D_i)$ provides $\epsilon$-differential privacy.

- When queries are applied to disjoint subsets of the data, we can improve the bound
- The ultimate privacy guarantee depends only on the worst of the guarantees of each analysis, not on the sum
What about group privacy?

• Differential privacy protects one entry of the database
• What if we want to protect several entries?
• We consider databases differing in $c$ entries
• By inductive reasoning, we can see that the probability dilatation is bounded by $e^{c\epsilon}$ instead of $e^\epsilon$, i.e.,

$$\Pr(M(D) \in C) \leq e^{c\cdot\epsilon} \cdot \Pr(M(D') \in C)$$

• To get $\epsilon$-differential privacy for $c$ items, one has to protect each of them with $\epsilon/c$-differential privacy
• **Exercise**: prove it
Achieving differential privacy

• So far we focused on the definition itself
• The question now is, how can we make a certain query differentially private?
• We will consider first a generally applicable sanitization mechanism, the Laplace mechanism
Sensitivity of a function

The sensitivity of a function $f : \mathcal{D} \rightarrow \mathbb{R}$ is defined as:

$$\Delta f = \max_{D, D'} |f(D) - f(D')|$$

for all adjacent $D, D' \in \mathcal{D}$

• Sensitivity measures how much the function amplifies the distance of the inputs

• Exercises: what is the sensitivity of
  • counting queries (e.g., “how many patients in the database have diabetes”) ?
  • “How old is the oldest patient in the database?”
Laplace distribution

• Denoted by Lap(b)
• Increasing b flattens the curve

\[ pr(z) = \frac{e^{-|z|}}{2b} \]

Variance = \(2b^2\)

Standard deviation \(\sigma = \sqrt{2b}\)
Laplace mechanism [Dwork et al., TCC’06]

Let $f : \mathcal{D} \rightarrow \mathbb{R}$ be a function with sensitivity $\Delta f$. Then $g = f(X) + \text{Lap}(\frac{\Delta f}{\varepsilon})$ is $\varepsilon$-differentially private.

- General sanitization mechanism
  - we have just to compute the sensitivity of the function
- Noise depends on $f$ and $\varepsilon$, not on the database!
- Remember how the Laplace distribution looks like: smaller sensitivity (and/or less privacy) means less distortion
- Exercise: how much noise do we have to add to sanitize the following question?
  - “How many people in the database are female?”
Differential Privacy (Part II)
Differential privacy (recap)

A query mechanism $M$ is $\epsilon$-differentially private if, for any two adjacent databases $D$ and $D'$ (differing in just one entry) and $C \subseteq \text{range}(M)$

$$\Pr(M(D) \in C) \leq e^\epsilon \cdot \Pr(M(D') \in C)$$
Sequential composition theorem (proof)

Let $M_i$ each provide $\epsilon_i$-differential privacy. The sequence of $M_i(X)$ provides $(\sum_i \epsilon_i)$-differential privacy.

**Proof.** For any sequence $r$ of outcomes $r_i \in \text{Range}(M_i)$ we write $M_i^r$ for mechanism $M_i$ supplied with $r_1, \ldots, r_{i-1}$. The probability of output $r$ from the sequence of $M_i^r(A)$ is

$$\text{Pr}[M(A) = r] = \prod_i \text{Pr}[M_i^r(A) = r_i].$$

Applying the definition of differential privacy for each $M_i^r$,

$$\prod_i \text{Pr}[M_i^r(A) = r_i] \leq \prod_i \text{Pr}[M_i^r(B) = r_i] \times \prod_i \exp(\epsilon_i \times |A \oplus B|).$$

Reconstituting the first product into $\text{Pr}[M(B) = r]$ gives the definition of $(\sum_i \epsilon_i)$-differential privacy. $\Box$
Parallel composition theorem (proof)

Let $M_i$ each provide $\epsilon$-differential privacy. Let $D_i$ be arbitrary disjoint subsets of the input domain $\mathcal{D}$. The sequence of $M_i(X \cap D_i)$ provides $\epsilon$-differential privacy.

**Proof.** For $A$ and $B$, let $A_i = A \cap D_i$ and $B_i = B \cap D_i$, and write $M_i^r$ for mechanism $M_i$ supplied with $r_1, \ldots, r_{i-1}$. The probability of output $r$ from the sequence of $M_i^r(A)$ is

$$\Pr[M(A) = r] = \prod_i \Pr[M_i^r(A_i) = r_i].$$

Applying the definition of differential privacy for each $M_i^r$,

$$\prod_i \Pr[M_i^r(A_i) = r_i] \leq \prod_i \Pr[M_i^r(B_i) = r_i] \times \prod_i \exp(\epsilon \times |A_i \oplus B_i|)$$

$$\leq \prod_i \Pr[M_i^r(B_i) = r_i] \times \exp(\epsilon \times |A \oplus B|).$$

Reassembly gives the definition of $\epsilon$-differential privacy. $\square$
Sensitivity of a function (recap)

The sensitivity of a function $f : \mathcal{D} \to \mathbb{R}$ is defined as:

$$\Delta f = \max_{D, D'} |f(D) - f(D')|$$

for all adjacent $D, D' \in \mathcal{D}$

• Sensitivity measures how much the function amplifies the distance of the inputs

• Exercises: what is the sensitivity of
  - counting queries (e.g., “how many patients in the database have diabetes”)?
  - “How old is the oldest patient in the database?”
Laplace distribution (recap)

- Denoted by \( \text{Lap}(b) \)
- Increasing \( b \) flattens the curve

\[
pr(z) = \frac{e^{-\frac{|z|}{2b}}}{2b}
\]

- Variance = \( 2b^2 \)
- Standard deviation \( \sigma = \sqrt{2b} \)
Laplace mechanism (recap)

Let \( f : \mathcal{D} \rightarrow \mathbb{R} \) be a function with sensitivity \( \Delta f \). Then 
\[
g = f(X) + \text{Lap}(\frac{\Delta f}{\epsilon})
\]
is \( \epsilon \)-differentially private.

- General sanitization mechanism
  - we have just to compute the sensitivity of the function
- Noise depends on \( f \) and \( \epsilon \), not on the database!
- Remember how the Laplace distribution looks like: smaller sensitivity (and/or less privacy) means less distortion
- Exercise: how much noise do we have to add to sanitize the following question?
  - “How many people in the database are female?”
Proof of Laplace mechanism

\[
\frac{Pr(f(D) + \text{Lap}(\Delta f/\epsilon) = y)}{Pr(f(D') + \text{Lap}(\Delta f/\epsilon) = y)} = e^{-\frac{|y - f(D)|\epsilon}{\Delta f}}
\]

\[
e^{-\frac{|y - f(D')|\epsilon}{\Delta f}} = e^{\frac{\epsilon}{\Delta f} \cdot (|y - f(D')| - |y - f(D)|)} \leq e^{\frac{\epsilon}{\Delta f} \cdot (f(D) - f(D'))}
\]

\[
\leq e^\epsilon
\]

• That’s it :)
Vector-valued queries

The sensitivity of a function $f : \mathcal{D} \rightarrow \mathbb{R}^d$ is defined as:

$$\Delta f = \max_{D,D'} \| f(D) - f(D') \|_1$$

for all adjacent $D, D' \in \mathcal{D}$

where $\| (x_1, \ldots, x_n) \|_1 = \sum_i |x_i|$ 

Let $f : \mathcal{D} \rightarrow \mathbb{R}^d$ be a function with sensitivity $\Delta f$. Then $g = f(X) + (Y_1, \ldots, Y_d)$, where the $Y_i$ are drawn i.i.d. from $\text{Lap}(\frac{\Delta f}{\epsilon})$, is $\epsilon$-differentially private.
Tools to play with

Applications of the Laplace sanitization mechanism:

- A Privacy-Integrated Query Language (PINQ)
  

- Fuzz: a typed functional language for differentially private computations
  
  http://privacy.cis.upenn.edu/software.html
Other sanitization mechanisms

• Ok, we know how to handle numeric queries
  • “How many people in this room have blue eyes?”
  • Perturb the result by an amount of noise proportional to the sensitivity of the query
• But what about non-numeric queries?
  • “What is the most common eye color in this room?”
• What if the perturbed answer isn’t almost as good as the exact answer?
  • “Which price would bring the most money from a set of buyers?”
Example: items for sale

- Could set the price of apples at $1.00 for profit $4.00
- Could set the price of apples at $4.01 for profit $4.01

- Best price: $4.01
- Second best price: $1.00
- Profit if you set the price at $4.02: $0
- Profit if you set the price at $1.01: $1.01
Exponential mechanism

[McSherry and Talwar, FOCS’07]

• A mechanism $M : \mathbb{N}^{|X|} \rightarrow R$ for some abstract range $R$
  • $R = \{\text{Red, Blue, Green, Brown, Purple}\}$
  • $R = \{$$1.00, \$1.01, \$1.02, \$1.03\}$
• Here the database is represented as a histogram
  • e.g., (Red,Green,Red,Brown,Blue,Green,Green) represented as (2,1,3,1,0)
• Paired with a quality score:
  
  $q : \mathbb{N}^{|X|} \times R \rightarrow \mathbb{R}$

  $q(D,r)$ represents how good output $r$ is for database $D$
  • the higher the score, the more appealing the result
The first idea is to define and compute the sensitivity of the scoring function (called **global sensitivity**): 

\[
GS(q) = \max_{r \in R, D, D'} : \| D - D' \|_1 \leq 1 \quad |q(D, r) - q(D', r)|
\]

The global sensitivity tells us the maximum change in the scoring function for two adjacent databases, for all possible results.
Exponential mechanism (cont’d)

Exponential($D, R, q, \epsilon$):

1. Let $\Delta = GS(q)$.

2. Output $r \sim R$ with probability proportional to

$$\exp\left(\epsilon q(D, r) \frac{2}{2\Delta}\right)$$

• Idea: make high quality outputs exponentially more likely at a rate that depends on the sensitivity of the quality score (and the privacy parameter)

$$\Pr(\text{Exponential}(D, R, q, \epsilon) = r) = \frac{\exp\left(\frac{\epsilon q(D, r)}{2\Delta}\right)}{\sum_{r' \in R} \exp\left(\frac{\epsilon q(D, r')}{2\Delta}\right)}$$
Exponential mechanism: privacy theorem

\[
\Pr[\text{Exponential}(D, R, q, \epsilon) = r] = \frac{\prod \exp\left(\frac{\epsilon q(D, r)}{2\Delta}\right)}{\prod \exp\left(\frac{\epsilon q(D, r)}{2\Delta}\right)}
\]

Exponential\((D, R, q, \epsilon)\) is \(\epsilon\)-differentially private
Exponential mechanism: privacy theorem

Exponential\((D, R, q, \epsilon)\) is \(\epsilon\)-differentially private

\[
= \left( \frac{\exp\left(\frac{\epsilon q(D, r)}{2\Delta}\right)}{\exp\left(\frac{\epsilon q(D', r)}{2\Delta}\right)} \right) \leq \\
\exp\left(\frac{\epsilon (q(D, r) - q(D', r))}{2\Delta}\right) = \exp\left(\frac{\epsilon \Delta}{2\Delta}\right) = \exp\left(\frac{\epsilon}{2}\right)
\]
Exponential mechanism: privacy theorem

\[
\text{Exponential}(D, R, q, \epsilon) \text{ is } \epsilon\text{-differentially private}
\]
Exponential mechanism: privacy theorem

Exponential($D, R, q, \epsilon$) is $\epsilon$-differentially private

$$\frac{\Pr[\text{Exponential}(D, R, q, \epsilon) = r]}{\Pr[\text{Exponential}(D', R, q, \epsilon) = r]} \leq \exp\left(\frac{\epsilon}{2}\right) \exp\left(\frac{\epsilon}{2}\right)$$

$$= \exp(\epsilon)$$
Exponential mechanism: accuracy theorem

• What about the accuracy of the answer?
• It depends...it is good if there is a sufficient mass of values of values $r$ with value $q$ close to optimum

Define:

- $OPT_q(D) = \max_{r \in R} q(D, r)$
- $R_{OPT} = \{ r \in R : q(D, r) = OPT_q(D) \}$
- $r^* = \text{Exponential}(D, R, q, \epsilon)$

$$\Pr(q(D, r^*) \leq OPT_q(D) - \frac{2\Delta}{\epsilon} \left( \log\left( \frac{|R|}{|R_{OPT}|} + t \right) \right) \leq e^{-t}$$

❖ For the proofs of these results, see www.cis.upenn.edu/~aaroth/courses/slides/Lecture3.pdf
Exponential mechanism: example

• “What is the most common eye color in this room?”
  • \( R = \{ \text{Red, Blue, Green, Brown, Purple} \} \)
• Here \( q \) returns the number of people with that eye colour
• We can answer with a color that is shared by
  \[
  OPT - \frac{2}{\epsilon} (\log(5) + 3) < OPT - \frac{7.4}{\epsilon} \quad \text{people}
  \]
• Except with probability \( \leq e^{-3} < .05 \)
• Independent of the number of people
• Very small error if \( n \) is large
Laplace mechanism vs exponential mechanism

• The Laplace mechanism is just an instance of the exponential mechanism
• For simplicity, let’s focus on functions of sensitivity 1
• We have just to define \( q(D, r) = -2|f(D) - r| \)
  • less noise gives better quality
• In general, it can be shown that the exponential mechanism captures any differentially private by choosing an appropriate scoring function [McSherry and Talwar, FOCS’07]
Approximate (or $(\varepsilon,\delta)$)-differential privacy

- Generalized definition of differential privacy allowing for a (supposedly small) additive factor
- Used in a variety of applications

A query mechanism $M$ is $(\varepsilon,\delta)$-differentially private if, for any two adjacent databases $D$ and $D'$ (differing in just one entry) and $C \subseteq \text{range}(M)$,

$$\Pr(M(D) \in C) \leq e^\varepsilon \cdot \Pr(M(D') \in C) + \delta$$
Differential Privacy (Part III)
Approximate (or $(\varepsilon, \delta)$)-differential privacy

- Generalized definition of differential privacy allowing for a (supposedly small) additive factor
- Used in a variety of applications

A query mechanism $M$ is $(\varepsilon, \delta)$-differentially private if, for any two adjacent databases $D$ and $D'$ (differing in just one entry) and $C \subseteq \text{range}(M)$

$$\Pr(M(D) \in C) \leq e^\varepsilon \cdot \Pr(M(D') \in C) + \delta$$
The Gaussian mechanism

The \( \ell_2 \)-sensitivity of \( f: \mathbb{N}^{|X|} \rightarrow \mathbb{R}^k \) is defined as
\[
\Delta_2(f) = \max \| f(x) - f(y) \|_2
\]
for all \( x, y \in \mathbb{N}^{|X|}, \| x - y \|_1 = 1 \)

For \( c^2 > 2 \ln(1.25/\delta) \), the Gaussian mechanism with parameter \( \sigma \geq c \Delta_2(f)/\varepsilon \) is \((\varepsilon, \delta)\)-differentially private.
Sparse Vector Technique

- [Hardt-Rothblum, FOCS’10] study the problem of $k$, adaptively chosen, low sensitivity queries where
  - only a very small number of these queries (say $c$) take values above a certain threshold $T$
  - the data analyst is only interested in such queries
  - useful to learn correlations, e.g., whether there is a dependency between smoke and cancer
- The data analyst could ask only the significant queries, but she does not know them in advance!
Histories and linear queries

- A histogram \( x \in \mathbb{R}^N \) represents a database (or a distribution) over a universe \( U \) of size \( |U|=N \)
  - Databases have support of size \( n \), whereas distributions do not necessarily have a small support
- We assume \( x \) is normalized so that \( \sum_{i \in U} x_i = 1 \)
- Here we focus on linear queries \( f : \mathbb{R}^N \rightarrow [0, 1] \)
  - can be seen as the inner-product \( \langle x, f \rangle \) for \( f \in [0, 1]^N \)
  - counting queries (i.e., how many elements in the database fulfill a certain predicate) are a special case
- Example: \( U = \{1,2,3\} \quad D = [1,2,2,3,1] \)
  - \( x = (2,2,1) \), after normalization \( (2/5,2/5,1/5) \)
  - “how many entries \( \leq 2 \)” \( \Rightarrow f = (1,1,0) \)
- By normalization, linear queries have sensitivity \( 1/n \)
**Intuition:** answer only those queries whose *sanitized* result is above the *sanitized* threshold.
SVT: accuracy

We say Sparse is \((\alpha, \beta)\)-accurate for a sequence of \(k\) queries \(Q_1, \ldots, Q_k\), if except with probability at most \(\beta\), the algorithm does not abort before \(Q_k\), and for all \(a_i \in \mathbb{R}\):

\[
|a_i - Q_i(D)| \leq \alpha
\]

and for all \(a_i = \bot\):

\[
Q_i(D) \leq T + \alpha
\]

- \(\alpha\) captures the distance between the sanitized result and the real result
- \(\beta\) captures the error probability
**SVT: accuracy theorem**

For any sequence of \( k \) queries \( Q_1, \ldots, Q_k \) such that \( L(T) = |\{i : Q_i(D) \geq T - \alpha\}| \leq c \), \( \text{Sparse}(D, \{Q_i\}, T, c) \) is \((\alpha, \beta)\)-accurate for:

\[
\alpha = 2\sigma (\log k + \log \frac{2}{\beta}) = \frac{4c (\log k + \log \frac{2}{\beta})}{\epsilon n}
\]

- The larger \( \beta \), the smaller \( \alpha \)
- The accuracy loss is logarithmic in the number of queries
The Sparse vector algorithm is $\epsilon$-differentially private.

- So, what did we prove in the end?
- You can estimate the actual answers and report only those in this range:

- We can fish out insignificant queries almost “for free”, paying only logarithmically for them in terms of accuracy.
SVT: approximate differential privacy

- Setting $\sigma = \frac{\sqrt{32c \ln 1/\delta}}{\epsilon n}$, we get the following theorems:

The Sparse vector algorithm is $(\epsilon, \delta)$-differentially private

For any sequence of $k$ queries $Q_1, \ldots, Q_k$ such that $L(T) = |\{i : Q_i(D) \geq T - \alpha\}| \leq c$, $\text{Sparse}(D, \{Q_i\}, T, c)$ is $(\alpha, \beta)$-accurate for:

$$\alpha = 2\sigma(\log k + \log \frac{2}{\beta}) = \frac{128c \ln \frac{1}{\delta}(\log k + \log \frac{2}{\beta})}{\epsilon n}$$
Limitations

✧ Differential privacy is a general purpose privacy definition, originally thought for databases and later applied to a variety of different settings

✧ At the moment, it is considered the state-of-the-art

✧ Still, it is not the holy grail and it is not immune from concerns, criticisms, and limitations

✧ Typically accompanied by some over-claims
No free lunch in data privacy

✧ Privacy and utility cannot be provided without making assumptions about how data are generated (no free lunch theorem)
✧ Privacy means hiding the evidence of participation of an individual in the data generating process
✧ If database rows are not independent, this is different from removing one row
  • Bob’s participation in a social network may cause new edges between pairs of his friends
✧ If there is group structure, differential privacy may not work very well...
No free lunch in data privacy (cont’d)

- This work disputes three popular over-claims
- “DP requires no assumptions on the data”
  - database rows must actually be independent, otherwise removing one row does not suffice to remove the individual’s participation
- If rows are not independent, deciding how many entries should be removed and which ones is far from being easy...
No free lunch in data privacy (cont’d)

- The attacker knows all entries of the database except for one, so “the more an attacker knows, the greater the privacy risks”

- Thus we should protect against the strongest attacker

- Careful! In DP, the more the attacker knows, the less noise we actually add
  - intuitively, this is due to the fact that we have less to hide

Figure 1: Probability (y-axis) of outputting Bob’s record as \( \epsilon \) (x-axis) varies for bit-differential privacy (top line), attribute-differential privacy (middle line), bounded-differential privacy (bottom line)
“DP is robust to arbitrary background knowledge”

Actually, DP is robust when certain subsets of the tuples are known to the attacker

Other types of background knowledge may instead be harmful

- e.g., previous exact query answers

DP composes well with itself, but not necessarily with other privacy definitions or release mechanisms

One can get a new, more generic, DP privacy guarantee if, after releasing exact query answers, a set of tuples (not just one), called neighbours, is altered in a way that is still consistent with previously answered queries (plausible deniability)
Geo-indistinguishability

Goal: protect user’s exact location, while allowing approximate information (typically needed to obtain a certain desired service) to be released

Idea: protect the user’s location within a radius $r$ with a level of privacy that depends on $r$

- corresponds to a generalized version of the well-known concept of differential privacy.
• Achieve $\ell$-privacy within $r$
  • the provider cannot easily infer the user’s location within, say, the 7th arrondissement of Paris
  • the provider can infer with high probability that the user is located in Paris instead of, say, London
More formally...

**Definition 3.1 (geo-indistinguishability).** A mechanism $K$ satisfies $\epsilon$-geo-indistinguishability iff for all $x, x'$:

$$d_P(K(x), K(x')) \leq \epsilon d(x, x')$$

Equivalently, the definition can be formulated as $K(x)(Z) \leq e^{\epsilon d(x, x')} K(x')(Z)$ for all $x, x' \in \mathcal{X}, Z \subseteq \mathcal{Z}$.

- Here $K(x)$ denotes the distribution (of locations) generated by the mechanism $K$ applied to location $x$.
- Achieved through a variant of the Laplace mechanism.
Browser extension

Location Guard 1.2.0.1-signed
by Marco Stronati, Kostas Chatzikokolakis

Hide your geographic location from websites. Report a fake location with the addition of random noise in order to protect your privacy. Per site settings with three configurable levels of accuracy or even the possibility to use a fixed location.

Download Now
Works with Firefox 31.0 - 40.0 • View other versions
Malicious aggregators

• So far we focused on malicious analysts...
• ...but aggregators can be malicious (or at least curious) too!
Existing approaches

- Secure hardware (or trusted server)-based mechanisms
- Fully distributed mechanisms with individual noise
Distributed Differential Privacy

“What’s the average age of your self-help group?”

How to compute differentially private queries in a **distributed** setting (attacker model, cryptographic protocols…)?
Smart-metering

- Remote reads
  - Reads every 15-30 min
- Manual reads
  - One reads every 3 months to 1 year

- Fine-grained smart-metering has multiple uses:
  - time-of-use billing, providing energy advice, settlement, forecasting, demand response, and fraud detection

  - American Recovery and Reinvestment Act (2009, $4.5bn)

- EU: Directive 2009/72/EC

- UK: deployment of 47 million smart meters by 2020
Smart-metering: privacy issues

- Meter readings are sensitive
  - Were you in last night?
  - You do like watching TV, don’t you?
  - Another ready meal in the microwave?
  - Has your boyfriend moved in?
Smart-meters: privacy issues (cont’d)

US Energy Department in smart grid privacy warning

Its biggest question is control over third-party access to consumer energy usage data

By Jaikumi 01/14/2010

Privacy concerns scotch Smart Meters plan in Holland

Back at the beginning of December I wrote

Multimedia Content Identification Through Smart Meter Power Usage Profiles

Ulrich Greveler1, Peter Glöskötter2, Benjamin Justus1, Dennis Loehr1

1 Department of Communications & Environment, Rhein-Waal University of Applied Sciences, D-47457 Krefeld, Germany
ulrich.greveler@hochschule-rhein-waal.de

2 Computer Security Lab, Münster University of Applied Sciences, D-48565 Steinfurt, Germany
(benjamin.justus, loehr)@fh-muenster.de

Detailed information about the behavior and activities of a particular household, controls needs to be implemented for ensuring the data is collected, used and shared in line with privacy expectations.

A smart grid basically uses digital technology to transmit, distribute and deliver power to consumers in a more reliable and efficient manner than traditional electricity systems.

“Home meters” allow two-way wireless communication with utilities - to forecast demand and charge more at peak times and even switch off individual appliances remotely.
Privacy-friendly smart metering

Goals:

- precise billing of consumption while revealing no consumption information to third parties
- privacy-friendly real-time aggregation
Protocol overview

- $r_i$ answer from client $i$
- $k_{ij}$ key shared between client $i$ and aggregator $j$
- $t$ label classifying the kind of reading
- $w_i$ weight given to i’s answers

Fig. 4. Aggregation of readings using weights $\tilde{w}$ in an infrastructure with 5 meters and 3 aggregators. The setup phase ensures that meter $i$ and aggregator $j$ share a key $k_{i,j}$. 
Protocol overview

- Geometric distribution, Geom(\(\alpha\)), with \(\alpha > 1\), is the discrete distribution with support \(\mathbb{Z}\) and probability mass function
  \[
  \frac{\alpha - 1}{\alpha + 1} \alpha^{-|k|}
  \]

- Discrete counterpart of Laplace distribution

---

Let \(f : D \to \mathbb{Z}\) be a function with sensitivity \(\Delta f\). Then \(g = f(X) + \text{Geom}(\frac{\varepsilon}{\Delta f})\) is \(\varepsilon\)-differentially private.
On the other hand this seems to be necessary to protect from malicious aggregators... we will see a more elegant and precise solution based on SMPC.

The noise increases with the number of aggregators (each adds noise that suffices to get \(\varepsilon\)-differential privacy).

\[
P \sum_{k \in \mathbb{Z}} \frac{\alpha + 1}{\alpha + 1 + |k|^2} = \frac{2P\alpha}{(\alpha - 1)^2}
\]

- \(P\) is the number of aggregators
- The protocol guarantees \(\varepsilon\)-differential privacy even if all except for one aggregators are dishonest.
Limitations of Existing Approaches

• Privacy vs utility tradeoff

• Lack of generality (and scalability)

• Inefficiency:
  significant computational effort on user’s side

• Answer pollution:
  single entity can pollute result by excessive noise
PrivaDA: Idea and Design

- Inputs are shared among computation parties
- Computation parties jointly compute differentially private statistics
- Required noise is generated in a distributed fashion
- No party learns the individual inputs

Differentially Private Data Aggregation with Optimal Utility
Fabienne Eigner¹, Aniket Kate², Matteo Maffei¹, Francesca Pampaloni³, and Ivan Pryvalov²
Our Contributions (PrivaDA)

• We leverage recent advances on SMPC for arithmetic operations
  • uses SMPC to compose user data
  • uses SMPC to jointly compute the sanitization mechanism

• We support three sanitization mechanisms
  • Lap, DLap, exponential mechanism, more are possible

• We employ $\beta$ computation parties

• We employ zero-knowledge proofs

• First publicly available library for efficient arithmetic SMPC operations in malicious setting
PrivaDA 101: Differentially Private Year of Birth

≈

1978

\[1505 + 474 = 1979\]
SMPC for Distributed Sanitization Mechanisms

• We employ recent SMPC for arithmetic operations
  • fixed-point numbers [Catrina & Saxena, FC’10]
  • floating point numbers [Aliasgari et al., NDSS’13]
  • integers [From & Jakobsen, 2006]

• Key SMPC primitives
  • RandInt(k)
  • IntAdd, FPAdd, FLAdd, FLMul, FLDiv
  • FL2Int, Int2FL, FL2FP, FP2FL
  • FLExp, FLLog, FLLT, FLRound
Algorithms for Sanitization Mechanisms

- We provide algorithms for Laplace, Discrete Laplace, and Exponential
- Trick: reduce the problem to random number generation
  - Lap(λ) = Exp(1/λ)-Exp(1/λ) with Exp(λ)=−ln U(0,1)/λ
  - DLap(λ) = Geo(1-λ)-Geo(1-λ) with Geo(λ)= ⌈Exp(- In (1-λ)) ⌉
  - Exp(ε/2) = draw r∈U(0,1] and check \[ r \cdot \sum_{j=1}^{m} e^{eq(D,a_j)} \in (\sum_{k=1}^{j-1} e^{eq(D,a_k)}, \sum_{k=1}^{j} e^{eq(D,a_k)}) \]

| In: | d₁, ..., dₙ; λ = \frac{Δf}{ε} |
| Out: | (\sum_{i=1}^{n} d_i) + Lap(λ) |
| 1: | d = \sum_{i=1}^{n} d_i |
| 2: | r_x \leftarrow U(0,1]; r_y \leftarrow U(0,1] |
| 3: | r_z = λ(ln r_x - ln r_y) |
| 4: | w = d + r_z |
| 5: return w |

In: d₁, ..., dₙ; a₁, ..., aₘ; λ = \frac{ε}{2} 
Out: winning a_k
1: I₀ = 0
2: for j = 1 to m do
3: z_j = \sum_{i=1}^{n} d_i(j) 
4: \delta_j = e^{λz_j} 
5: I_j = \delta_j + I_{j-1} 
6: r \leftarrow U(0,1]; r' = rI_m 
7: k = binary_search(r', ≤, I₀, ..., I_m) 
8: return a_k
Protocol for Distributed Laplace Noise

- For $\beta$ computation parties:

In: Shared fixed point form ($\gamma$, $f$) inputs $[d_1]_\beta, \ldots, [d_n]_\beta$; $\lambda = \frac{\Delta f}{\epsilon}$

Out: $w = \left( \sum_{i=1}^{n} d_i \right) + \text{Lap}(\lambda)$ in the fixed point form

1: $[d]_\beta = [d_1]_\beta$
2: for $i = 2$ to $n$ do
3: $[d]_\beta = \text{FPAdd}([d]_\beta, [d_i]_\beta)$
4: $[r_x] = \text{RandInt}(\gamma+1)$; $[r_y] = \text{RandInt}(\gamma+1)$
5: $\langle [v_x], [p_x], 0, 0 \rangle = \text{FP2FL}([r_x], \gamma, f = \gamma, \ell, k)$
6: $\langle [v_y], [p_y], 0, 0 \rangle = \text{FP2FL}([r_y], \gamma, f = \gamma, \ell, k)$
7: $\langle [v_{x/y}], [p_{x/y}], 0, 0 \rangle = \text{FLDiv}(\langle [v_x], [p_x], 0, 0 \rangle, \langle [v_y], [p_y], 0, 0 \rangle)$
8: $\langle [v_{ln}], [p_{ln}], [z_{ln}], [s_{ln}] \rangle = \text{FLLog2}(\langle [v_{x/y}], [p_{x/y}], 0, 0 \rangle)$
9: $\langle [v_z], [z], [v], [s] \rangle = \text{FLMul}(-[v_{ln}] \cdot e^2, \langle [v_{ln}], [p_{ln}], [z_{ln}], [s_{ln}] \rangle)$
10: $[z] = \text{FL2FP}(\langle [v_{z1}], [p_{z1}], [z_{z1}], [s_{z1}] \rangle, \ell, k, \gamma)$
11: $[w]_\beta = \text{FPAdd}([d]_\beta, [z]_\beta)$
12: return $w = \text{Rec}([w]_\beta)$
Protocol for Distributed Discrete Laplace Noise

• For β computation parties:

\[
\text{In: Shared integer number (\(\gamma\)) inputs } [d_1]_\beta, \ldots, [d_n]_\beta; \lambda = e^{-\frac{\epsilon}{\Delta f}}; \alpha = \frac{1}{\ln \lambda \log_2 e}
\]

\[
\text{Out: integer } w = (\sum_{i=1}^{n} d_i) + \text{DLap}(\lambda)
\]

1: \([d]_\beta = [d_1]_\beta
2: \text{for } i = 2 \text{ to } n \text{ do}
3: \quad [d]_\beta = \text{IntAdd}([d]_\beta, [d_i]_\beta)
4: \quad [r_x] = \text{RandInt}(\gamma + 1); [r_y] = \text{RandInt}(\gamma + 1)
5: \quad ([v_x], [p_x], 0, 0) = \text{FP2FL}([r_x], \gamma, f = \gamma, \ell, k)
6: \quad ([v_y], [p_y], 0, 0) = \text{FP2FL}([r_y], \gamma, f = \gamma, \ell, k)
7: \quad ([v_{lnx}], [p_{lnx}], [z_{lnx}], [s_{lnx}]) = \text{FLLog2}([p_x], [v_x], 0)
8: \quad ([v_{nlny}], [p_{nlny}], [z_{nlny}], [s_{nlny}]) = \text{FLLog2}([p_y], [v_y], 0)
9: \quad ([v_{alnx}], [p_{alnx}], [z_{alnx}], [s_{alnx}]) =
\quad \text{FLMul}(\alpha, [v_{nlny}], [p_{nlny}], [z_{nlny}], [s_{nlny}])
10: \quad ([v_{alny}], [p_{alny}], [z_{alny}], [s_{alny}]) =
\quad \text{FLMul}(\alpha, [v_{alnx}], [p_{alnx}], [z_{alnx}], [s_{alnx}])
11: \quad ([z_{z1}], [p_{z1}], [z_{z1}], [s_{z1}]) =
\quad \text{FLRound}(([v_{alnx}], [p_{alnx}], [z_{alnx}], [s_{alnx}]), 0)
12: \quad ([v_{z2}], [p_{z2}], [z_{z2}], [s_{z2}]) =
\quad \text{FLRound}(([v_{alnx}], [p_{alnx}], [z_{alnx}], [s_{alnx}]), 0)
13: \quad [z_1] = \text{FL2Int}(([v_{z1}], [p_{z1}], [z_{z1}], [s_{z1}]), \ell, k, \gamma)
14: \quad [z_2] = \text{FL2Int}(([v_{z2}], [p_{z2}], [z_{z2}], [s_{z2}]), \ell, k, \gamma)
15: \quad [w]_\beta = \text{IntAdd}([d]_\beta, \text{IntAdd}([z_1], [z_2]))
16: \text{return } w = \text{Rec}([w]_\beta)
Protocol for Distributed Exponential Mechanism

- For β computation parties:

In: $[d_1], \ldots, [d_n]$; the number $m$ of candidates; $\lambda = \frac{\epsilon}{2}$
Out: $m$-bit $w$, s.t. smallest $i$ for which $w(i) = 1$ denotes winning candidate $a_i$

1: $I_0 = \langle 0, 0, 1, 0 \rangle$
2: for $j = 1$ to $m$
3: $[z_j]_\beta = 0$
4: for $i = 1$ to $n$
5: $[z_j]_\beta = \text{IntAdd}([z_j]_\beta, [d_i(j)]_\beta)$
6: $\langle [v_z], [p_z], [z_z], [s_z] \rangle = \text{Int2FL}([z_j]_\beta, \gamma, \ell)$
7: $\langle [v_z'], [p_z'], [z_z'], [s_z'] \rangle = \text{FLMul}(\lambda \cdot \log_2 e, \langle [v_z], [p_z], [z_z], [s_z] \rangle)$
8: $\langle [v_{\delta}], [p_{\delta}], [z_{\delta}], [s_{\delta}] \rangle = \text{FLExp2}(\langle [v_z'], [p_z'], [z_z'], [s_z'] \rangle)$
9: $\langle [v_{I_j}], [p_{I_j}], [z_{I_j}], [s_{I_j}] \rangle = \text{Add}([v_{I_{j-1}}], [p_{I_{j-1}}], [z_{I_{j-1}}], [s_{I_{j-1}}], [v_{I_{j-1}}], [p_{I_{j-1}}], [z_{I_{j-1}}], [s_{I_{j-1}}])$
10: $[r] = \text{Rand}(\gamma, \ell)$
11: $\langle [v_r], [p_r], 0, 0 \rangle = \text{FLMul}(\langle [v_{I}], [p_{I}], [z_{I}], [s_{I}] \rangle)$
12: $\langle [v_r'], [p_r'], [z_r'], [s_r'] \rangle = \text{FLMul}(\langle [v_r], [p_r], [z_r], [s_r] \rangle)$
13: $j_{\text{min}} = 1$, $j_{\text{max}} = m$
14: while $j_{\text{min}} < j_{\text{max}}$
15: $j_M = \lfloor \frac{j_{\text{min}} + j_{\text{max}}}{2} \rfloor$
16: if $\text{FLLT}(\langle [v_{I_{j_M}}], [p_{I_{j_M}}], [z_{I_{j_M}}], [s_{I_{j_M}}] \rangle, \langle [v_r'], [p_r'], [z_r'], [s_r'] \rangle)$ then
17: $j_{\text{min}} = j_M + 1$
18: else $j_{\text{max}} = j_M$
19: return $w_{j_{\text{min}}}$
Attacker Model and Privacy Guarantees

- We consider two settings:
  - honest-but-curious (HbC) computation parties:
    - we assume that less than \( t < \beta/2 \) of \( \beta \) parties collude
  - malicious computation parties:
    - we assume that less than \( t < \beta/2 \) of \( \beta \) parties collude
    - we modify our SMPC such that correctness of each computation step is proved by zero-knowledge proofs

Main results:

- The SMPC protocols for LM, DLM, and EM are **differentially private** in the honest-but-curious setting.
- The SMPC protocols for LM, DLM, and EM are **differentially private** in the malicious setting under the strong RSA and decisional Diffie-Hellman assumptions.
Performance of SMPC Operations (in sec)

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>HbC</th>
<th>Malicious</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>$\beta = 3,$</td>
<td>$\beta = 5,$</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$t = 1$</td>
<td>$t = 2$</td>
</tr>
<tr>
<td>Float</td>
<td>FLAdd</td>
<td>0.48</td>
<td>0.76</td>
</tr>
<tr>
<td></td>
<td>FLMul</td>
<td>0.22</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td>FLScMul</td>
<td>0.20</td>
<td>0.28</td>
</tr>
<tr>
<td></td>
<td>FLDiv</td>
<td>0.54</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>FLLT</td>
<td>0.16</td>
<td>0.23</td>
</tr>
<tr>
<td></td>
<td>FLRound</td>
<td>0.64</td>
<td>0.85</td>
</tr>
<tr>
<td>Convert</td>
<td>FP2FL</td>
<td>0.83</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>Int2FL</td>
<td>0.85</td>
<td>1.22</td>
</tr>
<tr>
<td></td>
<td>FL2Int</td>
<td>1.35</td>
<td>1.91</td>
</tr>
<tr>
<td></td>
<td>FL2FP</td>
<td>1.40</td>
<td>1.96</td>
</tr>
<tr>
<td>Log</td>
<td>FLLLog2</td>
<td>12.0</td>
<td>17.0</td>
</tr>
<tr>
<td>Exp</td>
<td>FLExp2</td>
<td>7.12</td>
<td>9.66</td>
</tr>
</tbody>
</table>

Libraries: GMP, Relic, Boost, and OpenSSL

Setup: 3.20 GHz (Intel i5) Linux machine with 16 GB RAM, using a 1 Gbps LAN
Performance of LM, DLM and EM

• For $\beta = 3$ and $t = 1$ and number of users $n = 100,000$

• The HbC setting
  • Distributed LM protocol: 15.5 sec
  • Distributed DLM protocol: 31.3 sec
  • Distributed EM protocol: 42.3 sec
  (for number of candidates $m = 5$)

• The malicious setting
  • Distributed LM protocol: 344 sec
Caveats with number representations

• Careful with finite representation of real numbers!
• E.g., porosity of FL representation breaks Laplace
• In the above papers, solutions based on suitable rounding and truncation mechanisms
• Can be easily integrated in our framework
Implementation and Performance

- Operations performed by computation parties
- No critical timing restrictions on DDP computations in most real-life scenarios
- Users simply forward their shared values to the computation parties (< 1 sec)

Demonstrates practicality of PrivaDA (even on computationally limited devices, such as smartphones)
Differential Privacy
(Part IV)
Cryptographic protocols essential in distributed systems...

- e-banking
- e-commerce
- e-mail
- e-voting
- e-passports
- online auctions
- file sharing
- social networks

Tons of attacks... never ending list!

- Needham-Schroeder (1996)
- Microsoft Passport (2001)
- Public-key Kerberos (2006)
- DAA (2007, 2008)
- French Electronic Passport (2010)
- 802.11i WEP (2001)
- ISAKMP (2005)

Flaws hard to spot, proofs hard to get right
Conceptual flaws in protocol design

- Needham-Schroeder (1996)
- Microsoft Passport (2001)
- Public-key Kerberos (2006)
- DAA (2007, 2008)
- French Electronic Passport (2010)

Cryptographic breaches

- 802.11i WEP (2001)
- LogJam (2015)

Implementation mistakes

- SSL (2001, 2009)
- ISAKMP (2005)
- Heartbleed (2014)
- Freak (2015)
How do we prove a protocol correct?

Call for automated verification techniques, proving end-to-end security guarantees across all three layers (protocol logic, cryptography, implementation)

Type systems particularly well suited

Modular, well-established theory, amenable to automation
### A few recent results in this line of research

<table>
<thead>
<tr>
<th>F7</th>
<th>Security Properties</th>
<th>PL Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>authorization policies</td>
<td>refinement types</td>
</tr>
<tr>
<td></td>
<td>secrecy</td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>authorization policies</td>
<td>refinement types</td>
</tr>
<tr>
<td></td>
<td>secrecy</td>
<td>union and intersection types</td>
</tr>
<tr>
<td></td>
<td></td>
<td>type disjointness</td>
</tr>
<tr>
<td>AF7</td>
<td>resource-aware policies</td>
<td>affine refinement types</td>
</tr>
<tr>
<td>F*</td>
<td>authorization policies</td>
<td>refinement types</td>
</tr>
<tr>
<td></td>
<td>proof assistant</td>
<td>monadic reasoning</td>
</tr>
<tr>
<td>DF7</td>
<td>differential privacy</td>
<td>affine types</td>
</tr>
</tbody>
</table>

If you are curious, have a look at
- “Dependent Types and Multi-Monadic Effects in F*”, Swamy et al., POPL’16
Non-tracking web analytics

- Threat model: non-colluding, HbC aggregators & malicious publishers

- Manual proof of differential privacy
- Further auditing protocol to discover publishers dropping clients’ answers
The snapshot attack

New attack:
ai is counted 4 times
≈ execute query 4 times

We are going to see how to formally and automatically prove differential privacy properties in distributed systems by a type system
DDP: definition

- Consider protocol as a query
  - PD denotes protocol P on database D
- Attacker is given access to query function (protocol)
  - O(PD) denotes attacker interacting with PD
- Given two similar databases D,D’:
  - if no attacker can distinguish between query on D and query on D’, then the query is differentially private

\[
\text{A protocol } P \text{ is } \varepsilon\text{-differentially private iff for all databases } D,D' \text{ such that } D \sim D' \text{ and every opponent } O \\
Pr[O(PD)=1] \leq e^\varepsilon \cdot Pr[O(PD')=1]
\]
Sneak Peek into our Type System

• As shown in [Reed & Pierce ICFP’10, Gaboardi et al. POPL’12]:
  • if each value is used at most k times, then query k-sensitive
  • linear type systems track how often a sensitive value is used
• Enforcing linearity in distributed setting hard (because attacker can replay/duplicate values!)
• Our type system:
  • checks whether each value of database is used at most k times
  • before releasing query results to the attacker, they must be sanitized with the special primitive add_noise(s) for noise addition (e.g, Lap(s))

if $s = k/\varepsilon$ and protocol has type $\vdash_{k\tau} !_{\infty} R$
then protocol $\varepsilon$-differentially private
Syntax of the language

\[ a \]

\[ h ::= \text{inl} \mid \text{inr} \mid \text{fold} \]

\[ M, N, D ::= \]
\[ x \]
\[ c \]
\[ f \]
\[ (M, N) \]
\[ h M \]
\[ \lambda x.A \]
\[ \text{read}_{\alpha: \tau} \]
\[ \text{write}_{\alpha: \tau} \]

\[ A, B, P, Q ::= \]
\[ M \]
\[ M N \]
\[ \text{let } x = A \text{ in } B \]
\[ \text{let } (x, y) = M \text{ in } A \]
\[ \text{case } M \text{ of } x \text{ in } A \text{ else } B \]
\[ \text{unfold } M \text{ as } x \text{ in } A \text{ else } B \]
\[ M = N \]
\[ \text{ref}_{\tau} \]
\[ \text{add\_noise}_{s} M \]

label
constructor
value
variable
constant from \( \Sigma \)
function from \( \Sigma \)
multiplicative pair
construction
function (\( x \) bound in \( A \))
reference read
reference write
expression
value
application
let (\( x \) bound in \( B \))
split (\( x, y \) bound in \( A \))

\text{case } (x \text{ bound in } A \text{ and } B)
\text{unfold } (x \text{ bound in } A)
syntactic equality
reference creation
Laplace noise addition
Semantics of the language

\[
[S, (\lambda x. A) N] \overset{\text{det}}{\longrightarrow_1} [S, A\{N/x]\]
\]

\[
[S, \text{let } (x, y) = (M, N) \text{ in } ]A \overset{\text{det}}{\longrightarrow_1} [S, A\{M/x\}\{N/y\}]
\]

\[
[S, \text{case inl } M \text{ of } x \text{ in } A \text{ else } B ] \overset{\text{det}}{\longrightarrow_1} [S, A\{M/x\}]
\]

\[
[S, \text{case inr } M \text{ of } x \text{ in } A \text{ else } B ] \overset{\text{det}}{\longrightarrow_1} [S, B\{M/x\}]
\]

\[
[S, \text{unfold fold } M \text{ as } x \text{ in } A \text{ else } B ] \overset{\text{det}}{\longrightarrow_1} [S, A\{M/x\}]
\]

\[
[S, \text{unfold } M \text{ as } x \text{ in } A \text{ else } B ] \overset{\text{det}}{\longrightarrow_1} [S, B] \quad \text{if } \forall N. M \neq \text{fold } N
\]

\[
[S, \text{let } x = M \text{ in } A] \overset{\text{det}}{\longrightarrow_1} [S, A\{M/x\}]
\]

\[
[S, \text{let } x = A \text{ in } B] \overset{\ell}{\longrightarrow_p} [S', \text{let } x = A' \text{ in } B] \quad \text{if } [S, A] \overset{\ell}{\longrightarrow_p} [S', A']
\]

\[
[S, \text{ref}_\tau] \overset{\text{det}}{\longrightarrow_1} [S \cup \{a : \tau \mapsto \text{none}\}, (\text{read}_{a:\tau}, \text{write}_{a:\tau})]
\]

\[
[S \cup \{a : \tau \mapsto M\}, \text{read}_{a:\tau}(\cdot)] \overset{\text{det}}{\longrightarrow_1} [S \cup \{a : \tau \mapsto \text{none}\}, M]
\]

\[
[S \cup \{a : \tau \mapsto M\}, \text{write}_{a:\tau} N] \overset{\text{det}}{\longrightarrow_1} [S \cup \{a : \tau \mapsto \text{some } N\}, ()]
\]

\[
[S, \text{add}_s \text{noise}_{s_1} r_1] \overset{\text{\textsc{noise} (r_1, r_2, s)}}{\longrightarrow_{\text{\textsc{lap}_s}(r_2)}} [S, r]
\quad \text{where } r_1, r_2 \in \mathbb{R} \text{ and } r = r_1 +_{\mathbb{R}} r_2
\]

\[
[S, M = M] \overset{\text{det}}{\longrightarrow_1} [S, \text{true}]
\]

\[
[S, M = N] \overset{\text{det}}{\longrightarrow_1} [S, \text{false}] \quad \text{where } M \neq N
\]

\[
[S, f \ F] \overset{\text{det}}{\longrightarrow_1} [S, C] \quad \text{if } f(F') = \Sigma C
\]
Types

\[ \tau, \rho ::= !_k \phi \]

\[ \phi, \psi ::= \]

- \( !_k \text{int} \): integer that can be used at most \( k \) times
- public data are given types of the form \( !_\infty \Phi \), since the attacker can manipulate them at will
- \( !_k \Phi \rightarrow \tau \): function from \( \Phi \) to \( \tau \) that uses the argument at most \( k \) times

Indexed types

- \( \tau, \rho ::= !_k \phi \)
- \( \phi, \psi ::= \)
- \( b \): base type
- \( \alpha \): type variable
- \( \mu \alpha.\tau \): iso-recursive type (\( \alpha \) bound in \( \tau \))
- \( \tau + \tau \): sum type
- \( \tau \otimes \tau \): multiplicative pair type
- \( \tau \rightarrow \tau \): function type
Linear Types and Sensitivity

- Define a distance $\delta_\tau$ on types
- Functions of type $!_k \Phi \rightarrow \tau$ are $k$ sensitive in $\Phi \rightarrow \tau$

\[
\delta_{!_k \phi}(x, y) = k \cdot \delta_\phi(x, y)
\]
\[
\delta_{\tau \otimes \rho}((x_1, x_2), (y_1, y_2)) = \delta_\tau(x_1, x_2) + \delta_\rho(y_1, y_2)
\]
\[
\delta_{\tau \circ \rho}(f, g) = \max_{x \in \tau}(\delta_\rho(f(x), g(x)))
\]

\[
\text{A function } f \text{ is } k\text{-sensitive in } \tau_1 \rightarrow \tau_2 \text{ iff}
\]
\[
\delta_{\tau_2}(f(x), f(y)) \leq k \cdot \delta_{\tau_1}(x, y) \text{ for all } x, y \in \tau_1
\]
Sealing-based cryptography

• We model cryptography through regular language constructs
• In particular, we make usage of sealing and unsealing functions
  • share a (secret) reference to a list, containing msg*ciph pairs

\[
\begin{align*}
\text{Seal}(\alpha) &= \alpha \rightarrow !_{\infty}\mathbb{R} \\
\text{Unseal}(\alpha) &= !_{\infty}\mathbb{R} \rightarrow \alpha
\end{align*}
\]

encrypt:  \[ M \]  \[ c \]  \[ !_{\infty}\mathbb{R} \]

decrypt:  \[ !_{\infty}\mathbb{R} \]  \[ (\alpha \rightarrow !_{\infty}\mathbb{R}) \]  \[ \alpha \]

store
the plaintext in the
list along with a fresh
ciphertext

store
the plaintext in the
list along with a fresh
ciphertext

check
if ciphertext is in
the list and return
plaintext

Plaintext
Ciphertext
(fresh value)
Opponent typability

Definition  (Opponent) A closed expression is an opponent if all typing annotations occurring therein are $!_{\infty} \mathbb{R}$.

Lemma  (Opponent typability) For every opponent $O$, we have that $\emptyset \vdash O : !_{\infty} \mathbb{R}$. 
Kinding and subtyping

If a value comes from the attacker, we can decide to treat it privately.

Only types with infinite replication index can be sent to the attacker.
Typing values

Typing rules are defined for replication index 1: $!_k I$ below makes them work for arbitrary replication indexes.

- Trivial typing for pre-defined constants and functions.
- Typing environment splitting (split replication indexes).
- Only needed to type-check the attacker.
- Introduces replication indexes if the environment contains enough resources.

- Notation:
  - $\text{inl}$: $(\tau, \tau + \tau')$
  - $\text{inr}$: $(\tau, \tau' + \tau)$
  - $\text{fold}$: $(\tau \{ \mu \alpha. \tau / \alpha \}, \mu \alpha. \tau)$
  - $\text{Read}(\tau)$: $\Delta!_\infty \text{Unit} - \omega!_1 \text{Option}(\tau)$
  - $\text{Write}(\tau)$: $\omega!_\infty \text{Unit}$
Typing expressions

Standar subtyping rule

Splitting pairs preserves replication index

Only rule introducing an infinite replication index

Can only branch on public values
Soundness results

**Definition** (Generalized $\epsilon$-DDP ($\epsilon, \tau$-DDP)). $P$ is $\epsilon, \tau$-differentially private iff for all constant terms $D, D'$ of type $\tau$ and all opponents $O$,

$$\Pr[O(PD) \rightarrow^* 1] \leq e^{\epsilon \cdot \delta_\tau(D, D')} \cdot \Pr[O(PD') \rightarrow^* 1]$$

**Theorem** ($((\epsilon/k, \tau)$-Differential Privacy). For all $k \in \mathbb{R}^{>0}$, all types $\tau$, and all closed expressions $P$ such that the following conditions hold:

- the parameter of all noise addition primitives is set to $k/\epsilon$ (i.e., they are of the form `add_noise_{k/\epsilon} M`)
- $\emptyset \vdash P : \tau \rightarrow \rho$ for some $\rho \in \mathcal{OPP}

$P$ is $\epsilon/k, \tau$-differentially private.
Typing the Example: HbC Aggregator

Problem: ciphertexts could be decrypted more than once!
Solution: prevent double processing of the same payload
Protocol fix: discard duplicate ciphertexts and change $q_{id}$ in each session in order to keep the queue size small
Typing the Example: Malicious Publisher

Result for Publisher j:
**Algorithmic Variant**

Let \( \text{Alg} \)

\[
\Gamma \vdash \hat{\phi} \quad x : ?_k \phi \in \Gamma \\
k \geq 1 \\
\Gamma' = \Gamma\{x : ?_{k-1} \phi / x : ?_k \phi\} \\
\Gamma \vdash_{\text{alg}} x : ?_1 \phi ; \Gamma'
\]

\[
\text{LET} \ A : \tau; \Delta \\
\Delta, x : \tau \vdash_{\text{alg}} B : \tau'; \Gamma' \\
\Gamma \vdash_{\text{alg}} \text{let } x = A \text{ in } B : \tau' ; \Gamma' / x
\]

Use all resources in the first hypothesis and those left in the second one.

The algorithmic variant is sound and complete.

Effect system to track unused resources.