Privacy-preserving Information Sharing: Crypto Tools and Applications

Emiliano De Cristofaro

University College London (UCL)

https://emilianodc.com

Privacy-preserving *what*?

Parties with limited mutual **trust** willing or required to share information

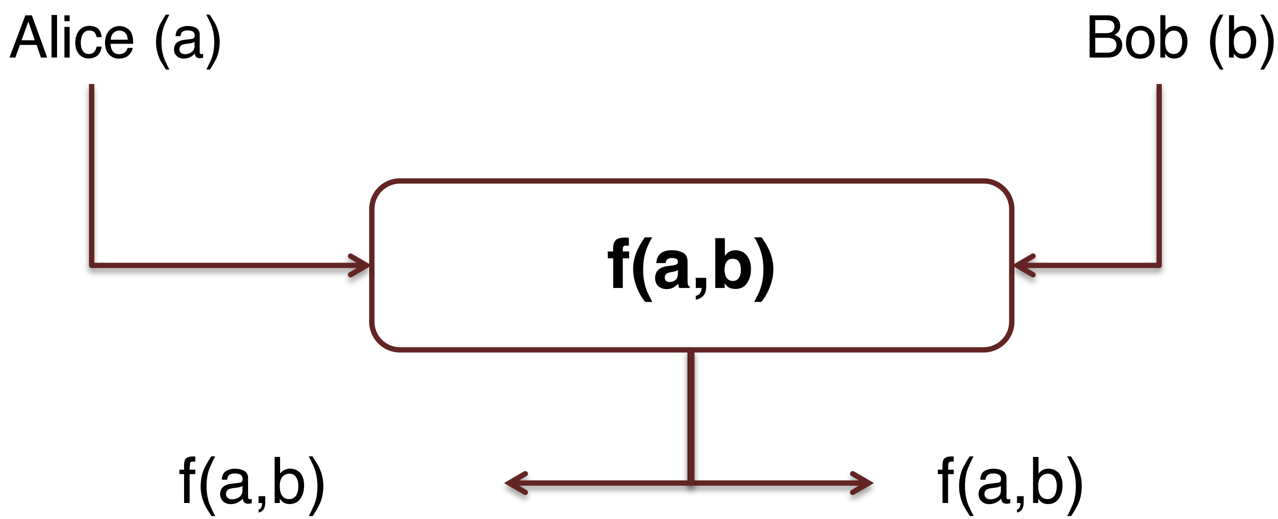
Only the required **minimum** amount of information should be disclosed in the process

Outline

1. Tools for two parties and a case study
2. Some applications
3. Multiple parties
4. Inference from shared information

Let's start with two parties…

Secure Computation (2PC)



Security?

Goldreich to the rescue!

Oded Goldreich. Foundations of Cryptography: Basic Applications, Ch. 7.2. Cambridge Univ Press, 2004.

Computational Indinguishability

Execution in “ideal world” with a trusted third party (TTP)

vs

Execution in “real world” (crypto protocol)

Who are the Adversaries?

Outside adversaries?

Not considered! Network security “takes care” of that

Honest but curious (HbC)

“Honest”: follows protocol specifications, do not alter inputs

“Curious”: attempt to infer other party’s input

Malicious

Arbitrary deviations from the protocol

Security a bit harder to formalize/prove (need to simulate the ideal world)

How to Implement 2PC?

1. Garbled Circuits

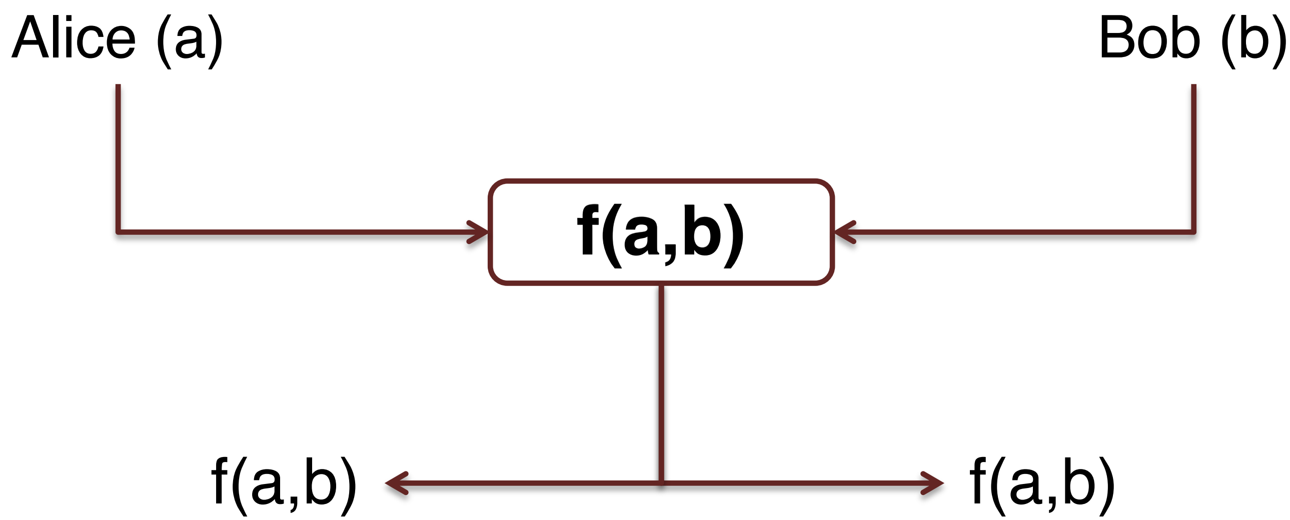
Sender prepares a *garbled* circuit and sends it to the receiver, who *obliviously* evaluates the circuit, learning the encodings corresponding to both her and the sender's output

2. Special-Purpose Protocols

Implement one specific function (and only that?)

Usually based on public-key crypto properties (e.g., homomorphic encryption)

Privacy-Preserving Information Sharing with 2PC?



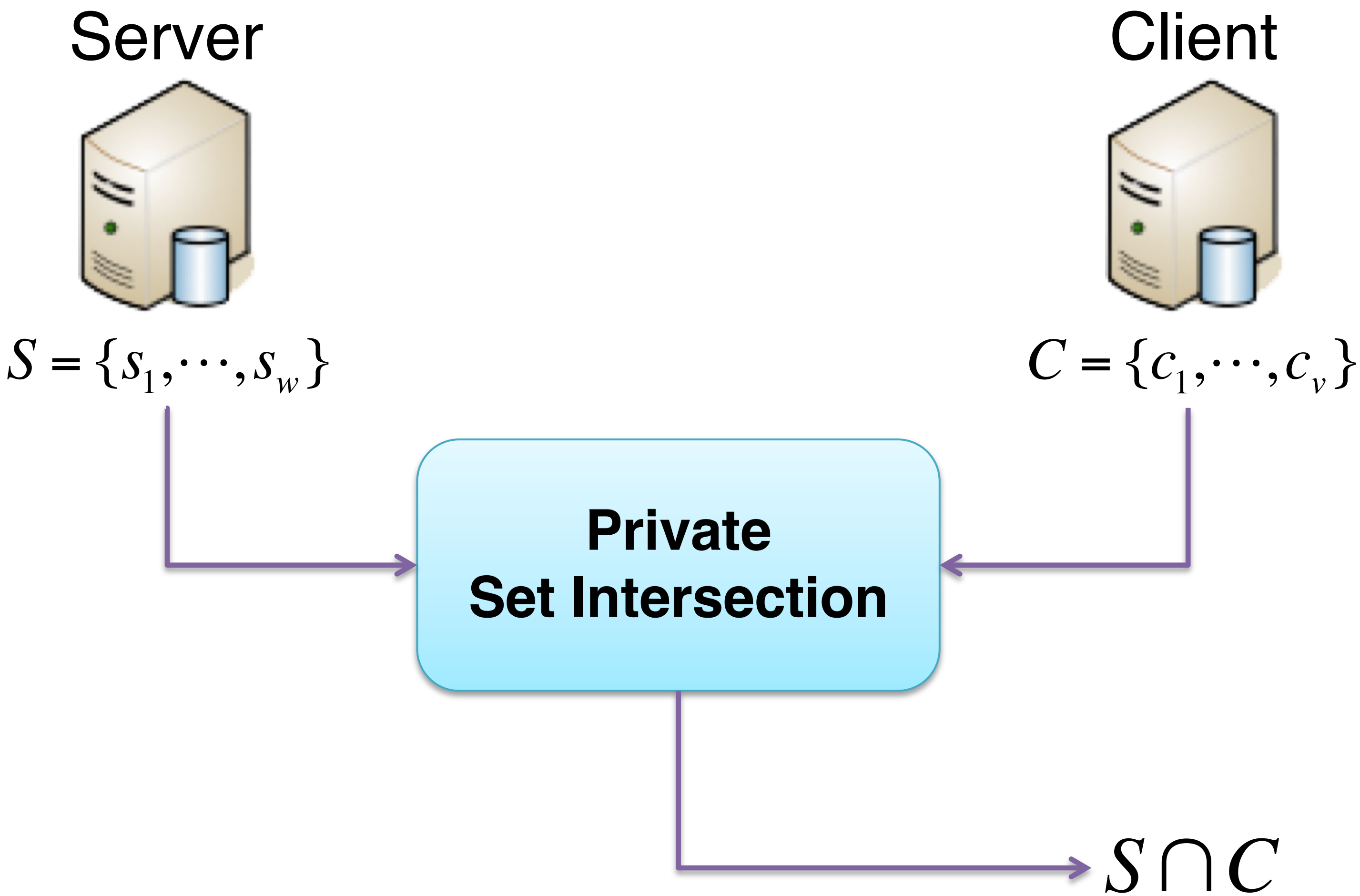
Map information sharing to f(·,·)?

Realize secure f(·,·) efficiently?

Quantify information disclosure from output of f(·,·)?

A Case Study:
Private Set Intersection

Private Set Intersection (PSI)



Private Set Intersection?

DHS (Terrorist Watch List) and **Airline** (Passenger List)

Find out whether any suspect is on a given flight

IRS (Tax Evaders) and **Swiss Bank** (Customers)

Discover if tax evaders have accounts at foreign banks

Etc.

Straightforward PSI

Server

$S = \{s_1, \cdots, s_w\}$

Client

$C = \{c_1, \cdots, c_v\}$

Straightforward PSI?

For each item s, the Server sends SHA-256(s)

For each item c, the Client computes SHA-256(c)

Learn the intersection by matching SHA-256's outputs

What's the problem with this?

Background: Pseudorandom Functions

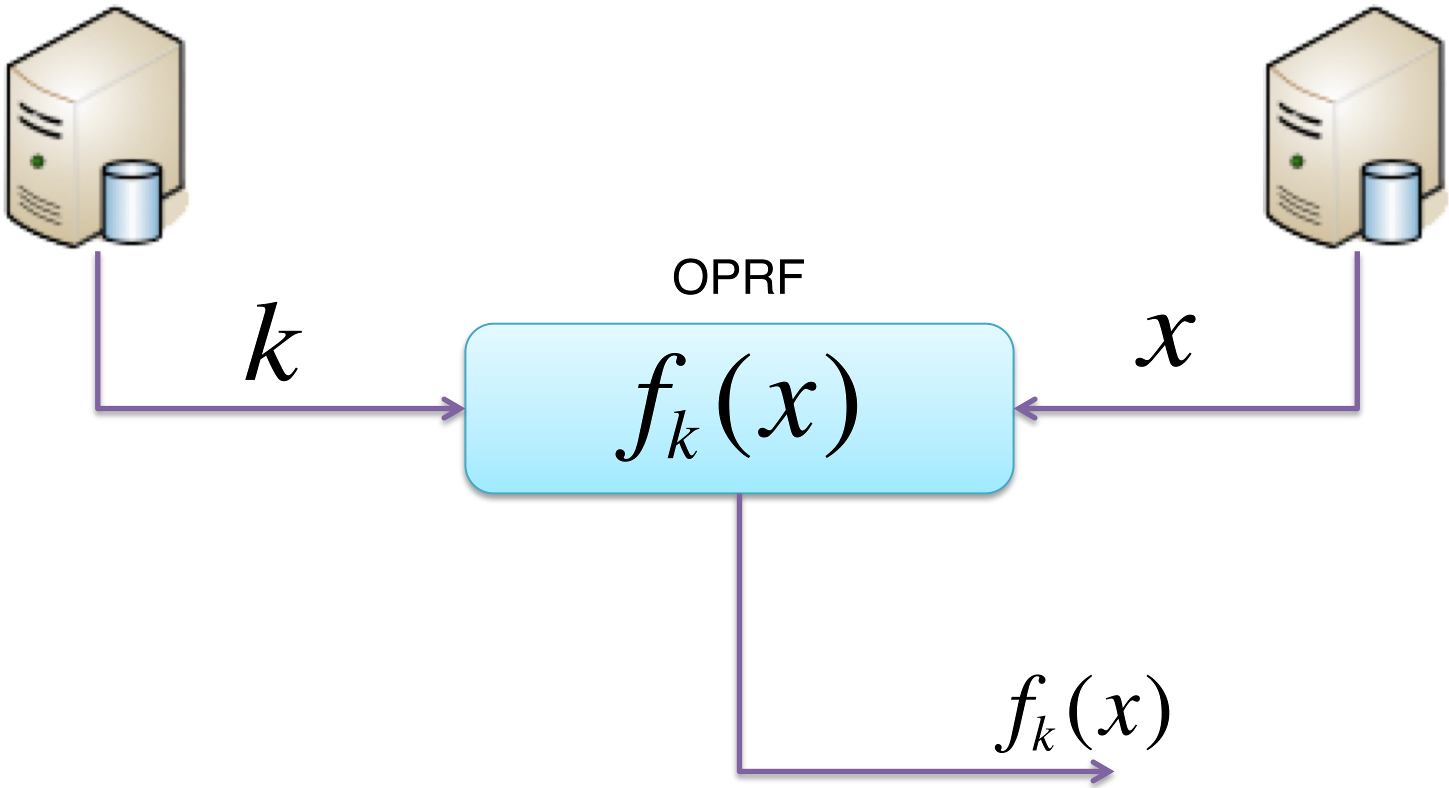
A **deterministic** function:

$$\begin{tikzcd} x \arrow[r] & \boxed{f} \arrow[r] & f_k(x) \\ & k \arrow[u] & \end{tikzcd}$$

Efficient to compute

Outputs of the function “look” **random**

Oblivious PRF



OPRF-based PSI

OPRF

$$f_k(x)$$

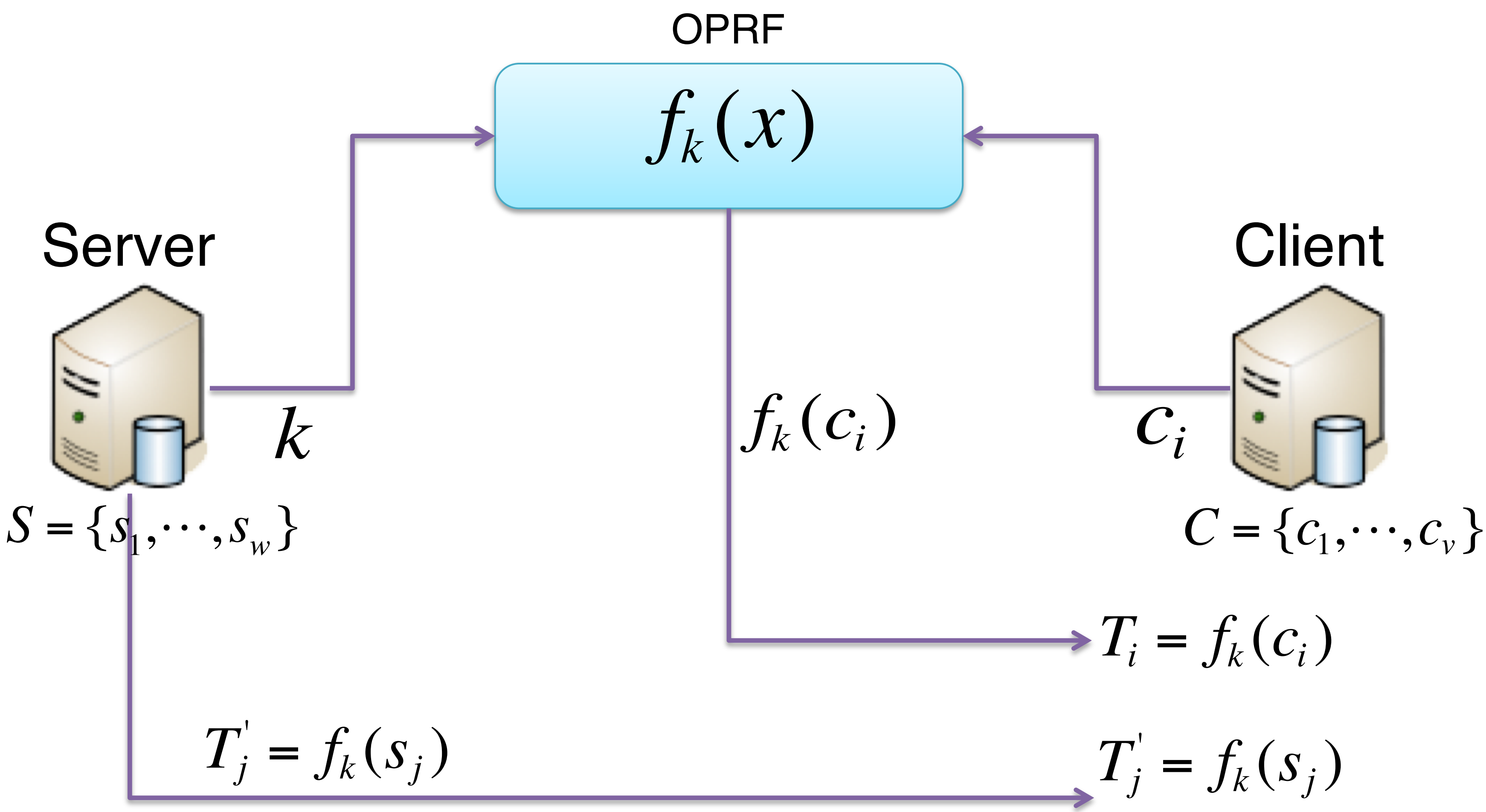
Server

$S = \{s_1, \cdots, s_w\}$

Client

c_i

OPRF-based PSI



Unless s_j is in the intersection T_j' looks random to the client

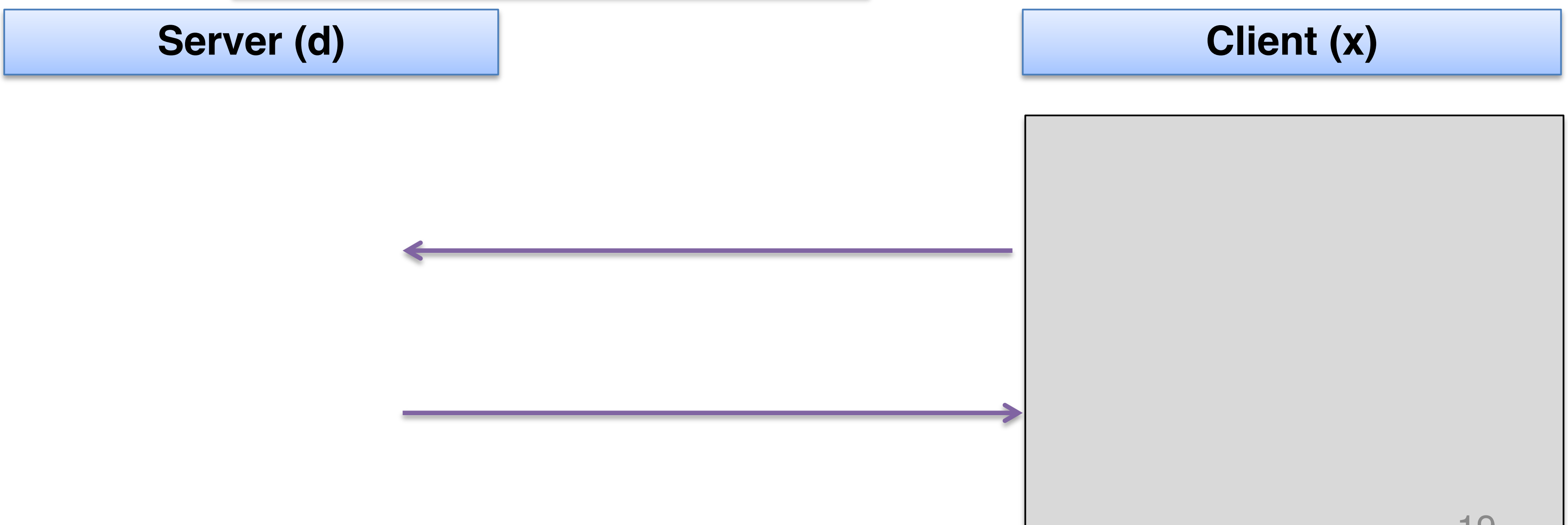
OPRF from Blind-RSA Signatures

RSA Signatures: $(N = p \cdot q, e), d \qquad e \cdot d \equiv 1 \bmod (p-1)(q-1)$

$$Sig_d(x) = H(x)^d \bmod N,$$

$$Ver(Sig(x), x) = 1 \Leftrightarrow Sig(x)^e = H(x) \bmod N$$

PRF: $\boxed{f_d(x) = H(sig_d(x))}$ (H one way function)



OPRF from Blind-RSA Signatures

RSA Signatures: $(N = p \cdot q, e), d \qquad e \cdot d \equiv 1 \bmod (p-1)(q-1)$

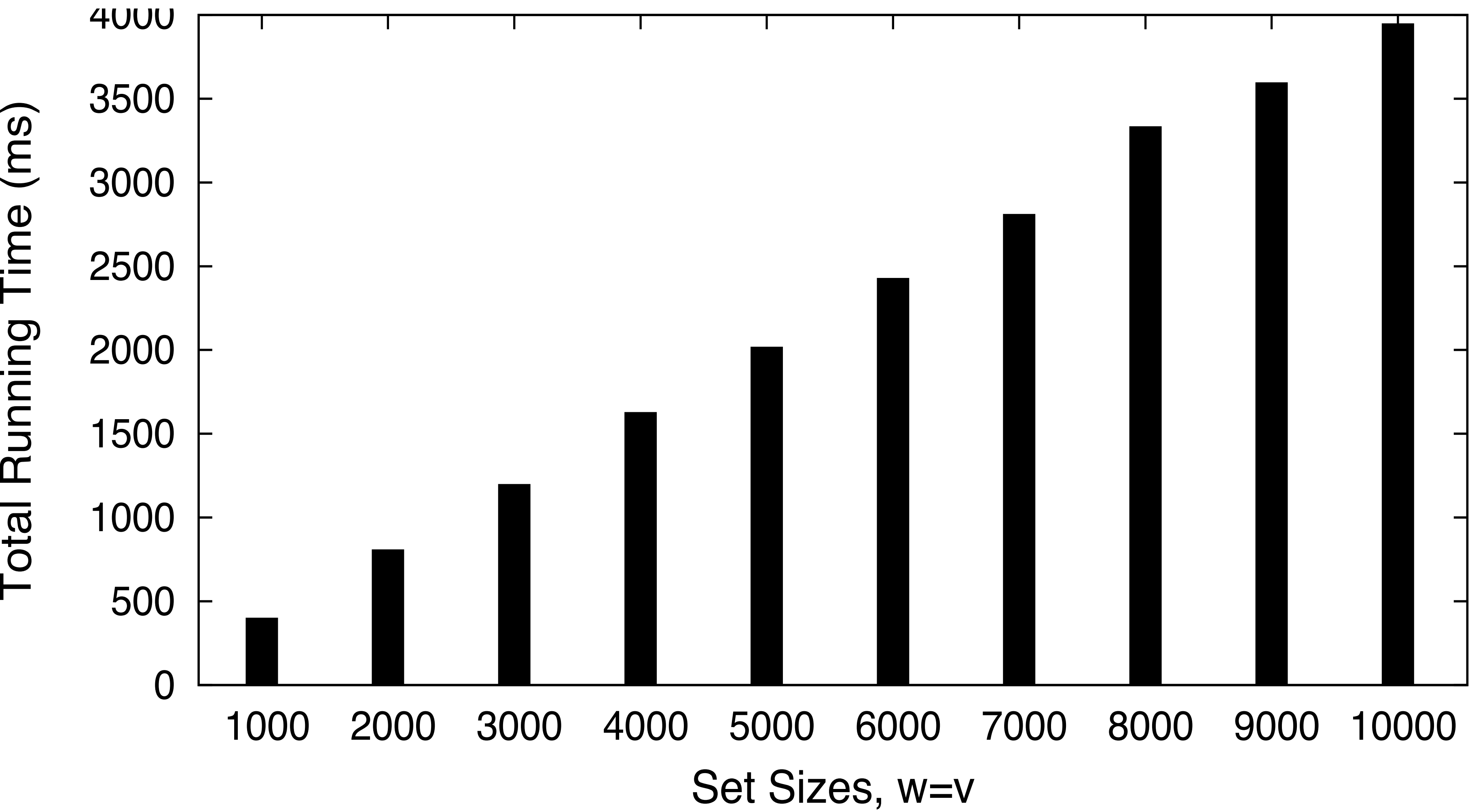
$$Sig_d(x) = H(x)^d \bmod N,$$

$$Ver(Sig(x), x) = 1 \Leftrightarrow Sig(x)^e = H(x) \bmod N$$

PRF: $\boxed{f_d(x) = H(sig_d(x))}$ (H one way function)

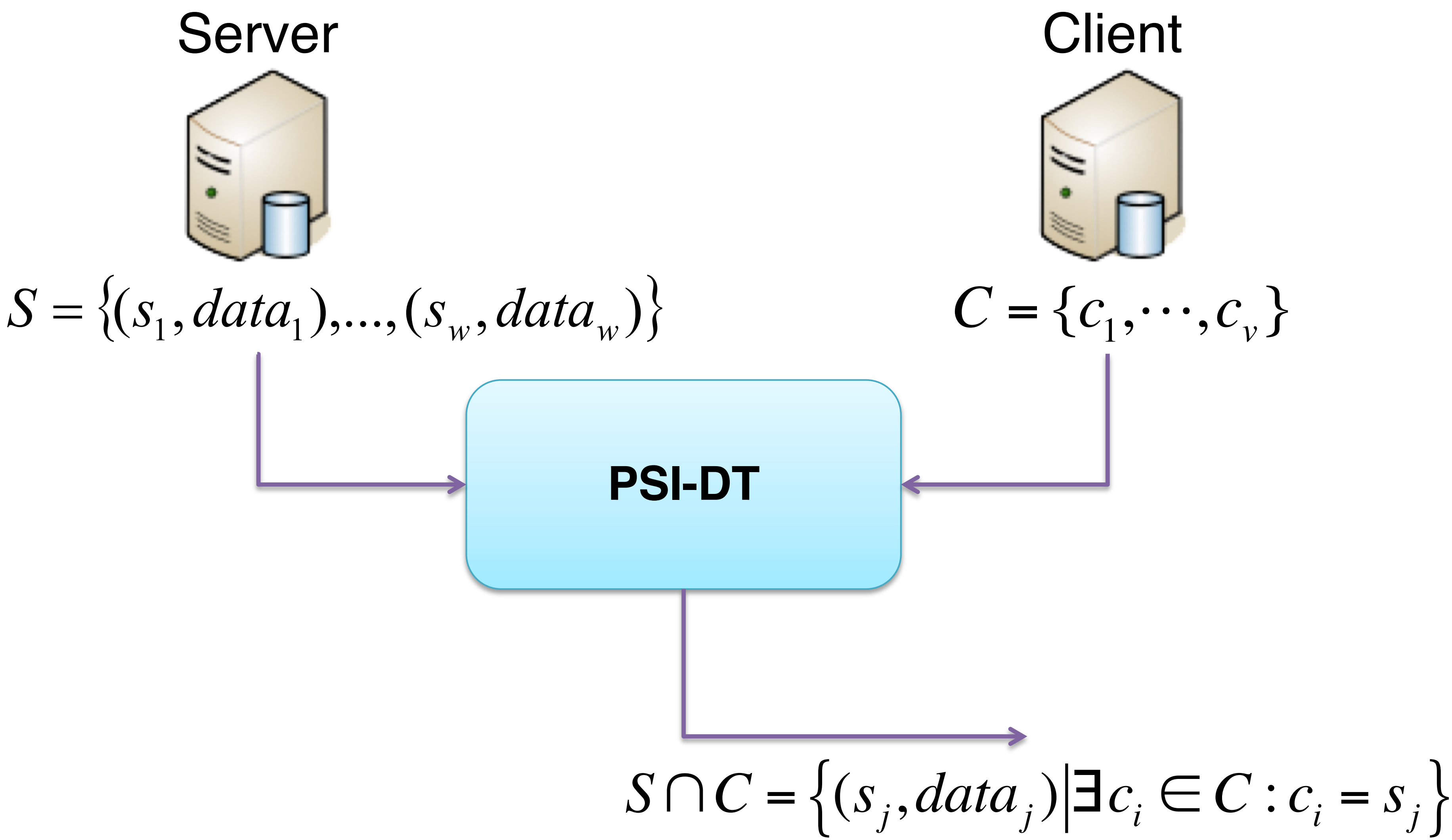
Server (d)		**Client (x)**
		$r \in Z_N$
	$\boxed{a = H(x) \cdot r^e}$ $\longleftarrow$	
	$\boxed{b = a^d}$ $\longrightarrow$	
	$(= H(x)^d r^{\cancel{ed}})$	$sig_d(x) = b / r$
		$f_d(x) = H(sig_d(x))$

Performance



See: De Cristofaro, Lu, Tsudik, Efficient Techniques for Privacy-preserving Sharing of Sensitive Information, TRUST 2011

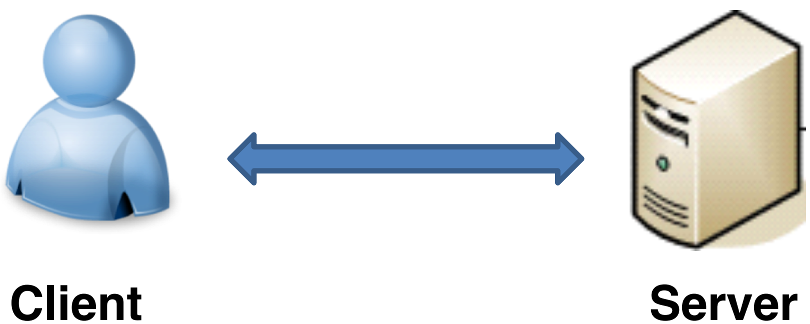
PSI w/ Data Transfer (PSI-DT)



How can we build PSI-DT?

PSI w/ Data Transfer

$$\text{SELECT * FROM DB WHERE } (attr_1^* = val_1^* \text{ OR } \cdots \text{ OR } attr_v^* = val_v^*)$$



$\mathbf{attr_1}$	$\mathbf{attr_2}$	$\cdots$	$\mathbf{attr_m}$	
val_{11}	val_{12}	$\cdots$	val_{1m}	R_1
val_{21}	val_{22}	$\cdots$	val_{2m}	R_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
val_{w1}	val_{w2}	$\cdots$	val_{wm}	R_w

DB

A closer look at PSI

Server

$S = \{s_1, \cdots, s_w\}$

Client

$C = \{c_1, \cdots, c_v\}$

Private
Set Intersection

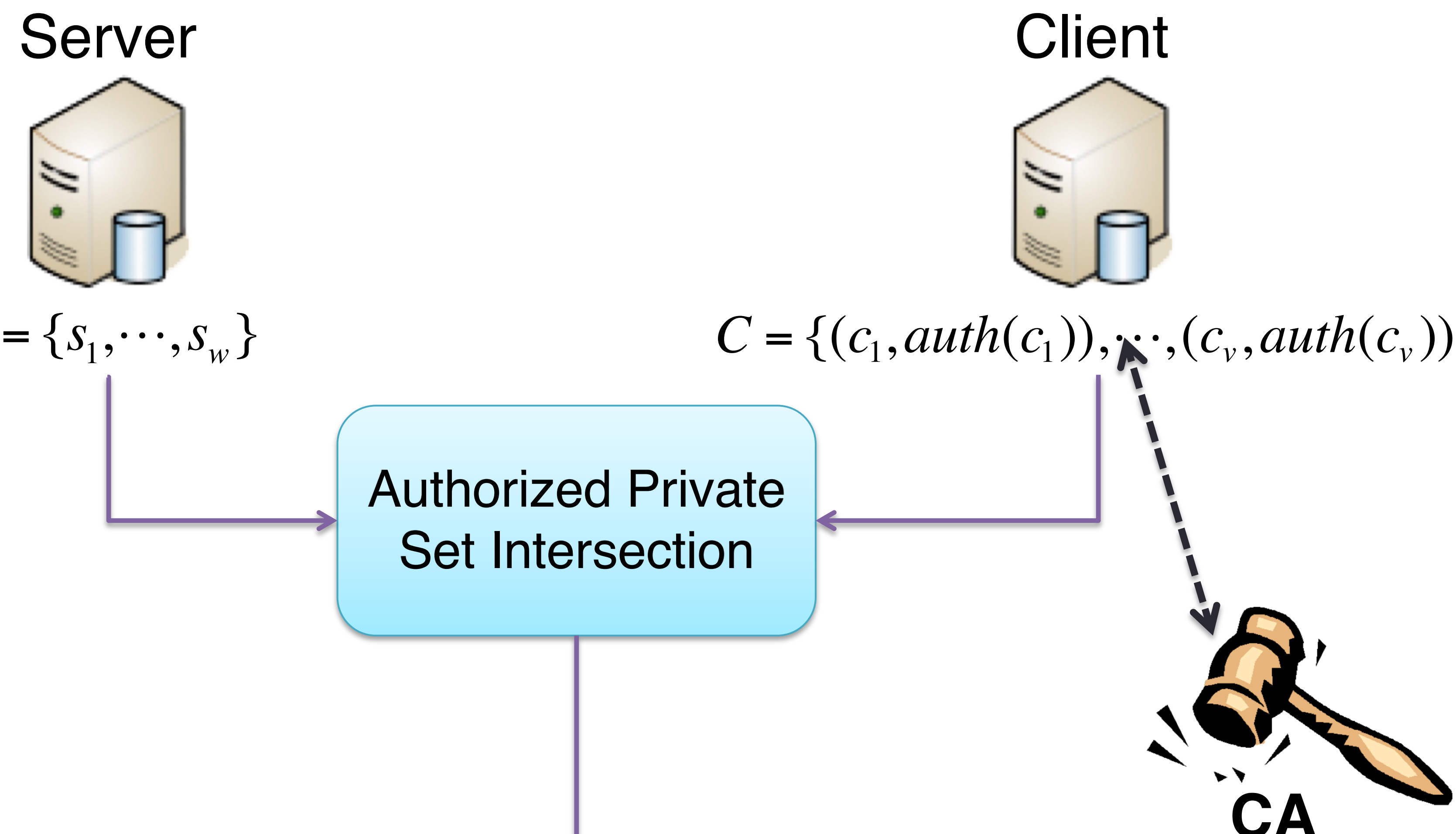
What if the client populates *C* with its best guesses for *S?*

Client needs to prove that inputs satisfy a policy or be authorized

Authorizations issued by appropriate authority

Authorizations need to be verified <u>implicitly</u>

Authorized Private Set Intersection (APSI)

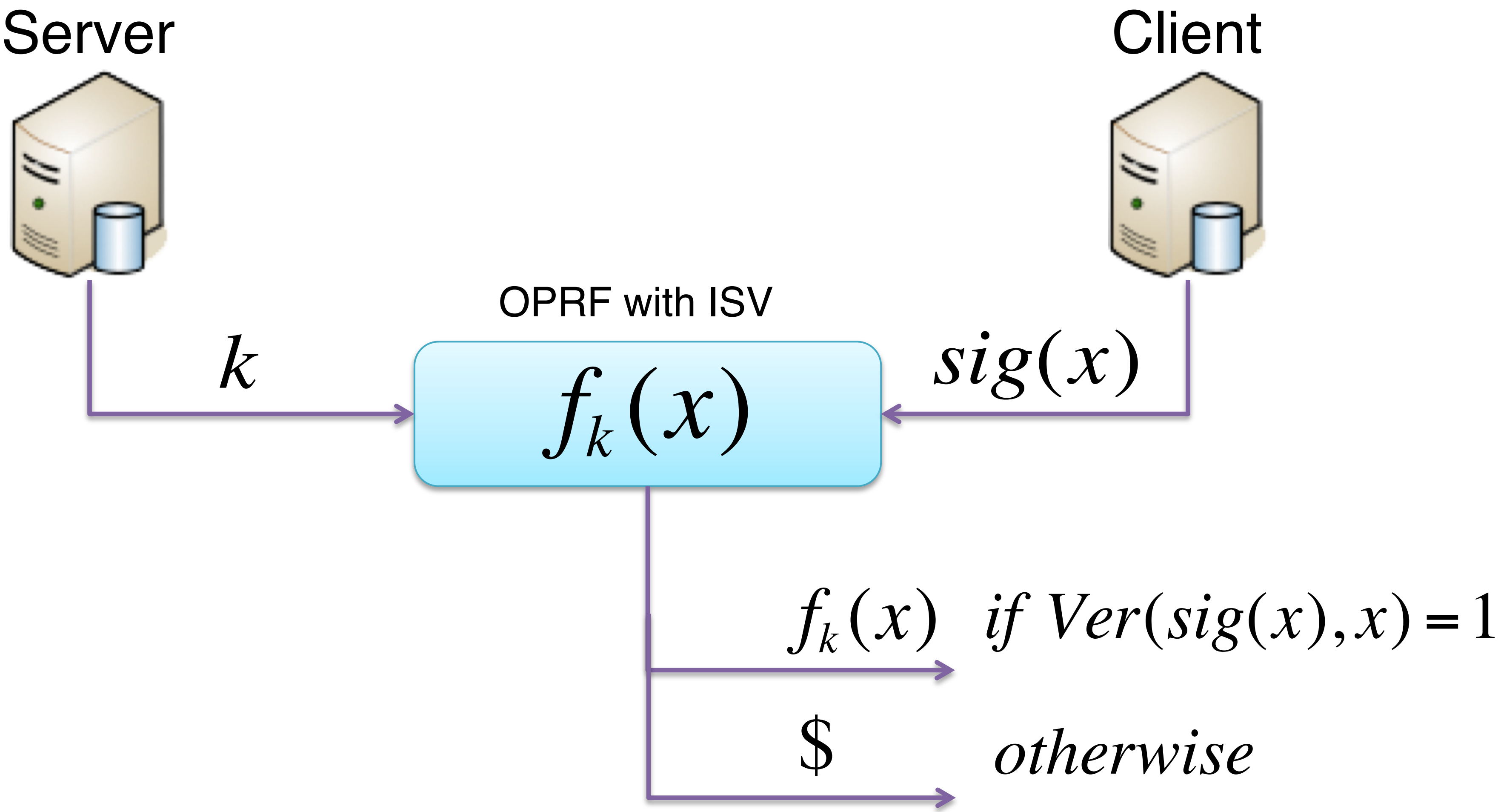


$S = \{s_1, \cdots, s_w\}$

$C = \{(c_1, auth(c_1)), \cdots, (c_v, auth(c_v))\}$

$$S \cap C \stackrel{def}{=} \left\{ s_j \in S \,\middle|\, \exists c_i \in C : c_i = s_j \wedge \boxed{auth(c_i) \; is \; valid} \right\}$$

OPRF w/ Implicit Signature Verification



A simple OPRF-like with ISV

Court issues authorizations: $Sig(x) = H(x)^d \bmod N$

OPRF: $\boxed{f_k(x) = F(H(x)^{2k} \bmod N)}$

Server (k)		**Client ($H(x)^d$)**
		$r \in Z_N$
	$\xleftarrow{\quad a = H(x)^d g^r \quad}$	
	$\xrightarrow{\quad b = a^{2ek}; g^k \quad}$	$H(x)^{2k} = b/g^{2erk}$
(Implicit Verification)	$(b = H(x)^{2e\cancel{d}k} g^{2rek})$	$f_k(x) = F(H(x)^{2k})$

OPRF with ISV – Malicious Security

OPRF: $\boxed{f_k(x) = F(H(x)^{2k})}$

Server (k)

Client (H(x)^d)

$r \in Z_N$

$a = H(x)^d g^r$ $\quad$ $\alpha = H(x)(g')^r$ $\longleftarrow$

$\pi = ZKPK\{r : a^{2e}/\alpha^2 = (g^e/g')^{2r}\}$

g^k $\quad$ $b = a^{2ek}$ $\quad$ $\pi' = ZKPK\{k : b = a^{2ek}\}$ $\longrightarrow$

$(b = H(x)^{2e\cancel{d}k} g^{2rek})$

$H(x)^{2k} = b/g^{2erk}$

$f_k(x) = F(H(x)^{2k})$

Proofs in Malicious Model

See:

De Cristofaro, Kim, Tsudik. Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model

Asiacrypt 2010

PSI with Garbled Circuits

Lots of progress recently!

Optimized Circuits

Oblivious Transfer Extensions

Better techniques to extend to malicious security

See:

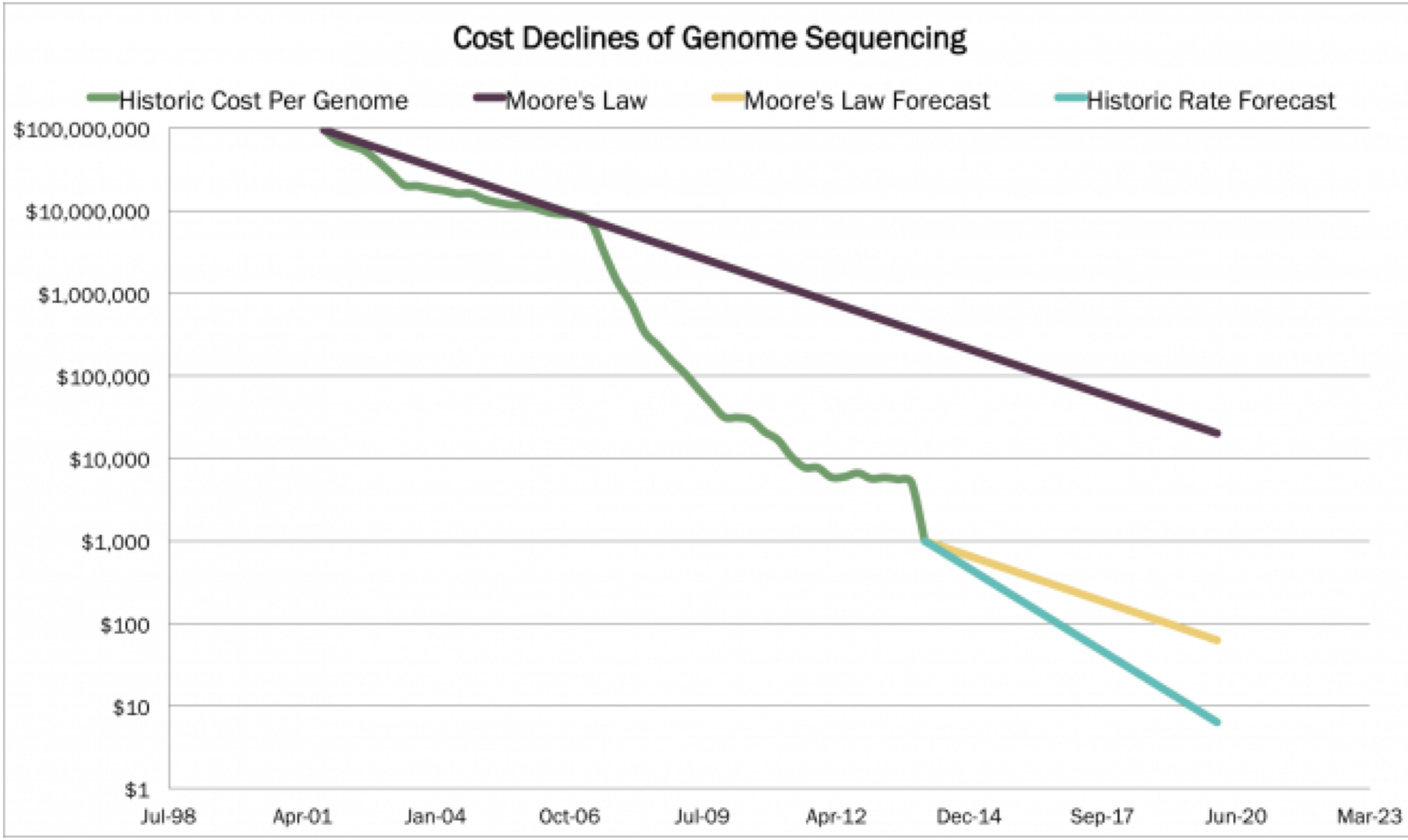
Pinkas et al., Scalable Private Set Intersection Based on OT Extension. ACM TOPS 2018

[More]

Quiz!

Go to kahoot.it

Applications to Genomics



From: James Bannon, ARK

1/05/2011 @ 4:57PM | 30,076 views

The First Child Saved By DNA Sequencing

+ Comment Now + Follow Comments

Comprehensive whole genome sequence analyses yields novel genetic and structural insights for Intellectual Disability

Farah R. Zahir, Jill C. Mwenifumbo, Hye-Jung E. Chun, Emilia L. Lim, Clara D. M. Van Karnebeek, Madeline Couse, Karen L. Mungall, Leora Lee, Nancy Makela, Linlea Armstrong, Cornelius F. Boerkoel, Sylvie L. Langlois, Barbara M. McGillivray, Steven J. M. Jones, Jan M. Friedman † and Marco A. Marra †

BMC Genomics 2017 **18**:403

https://doi.org/10.1186/s12864-017-3671-0 |

Received: 4 November 2016 | Accepted: 29 March 2017 | Published: 24 May 2017

Genomics promises a leap forward for rare disease diagnosis

Faster and cheaper DNA sequencing brings new hope to patients

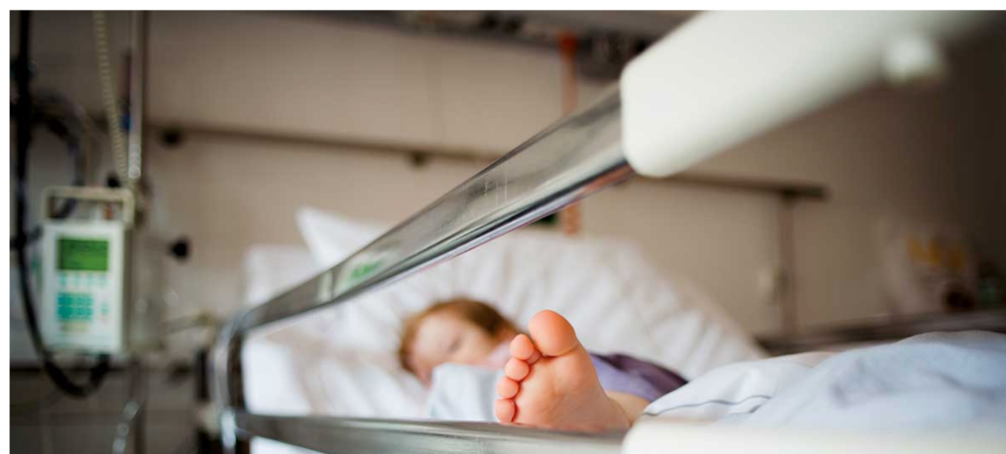
Jessica suffers from a rare condition that was diagnosed through DNA analysis

Clive Cookson FEBRUARY 28, 2017

Home | News | Health

THIS WEEK 26 March 2018

Three critically ill children helped by speedy genome sequencing



Genome Privacy

1. **Genome is treasure trove of sensitive information**
2. **Genome is the ultimate identifier**
3. **Genome data cannot be revoked**
4. **Access** to one's genome ≈ **access** to **relatives'** genomes
5. **Sensitivity does not degrade over time**

See: genomeprivacy.org

Genetic Paternity Test

A Strawman Approach for Paternity Test:

On average, ~99.5% of any two human genomes are identical

Parents and children have even more similar genomes

Compare candidate's genome with that of the alleged child:

Test positive if percentage of matching nucleotides is > 99.5 + τ

First-Attempt Privacy-Preserving Protocol:

Use secure computation for the comparison

PROs: High-accuracy and error resilience

CONs: Performance not promising (3 billion symbols in input)

In our experiments, computation takes a few days

Genetic Paternity Test

Wait a minute!

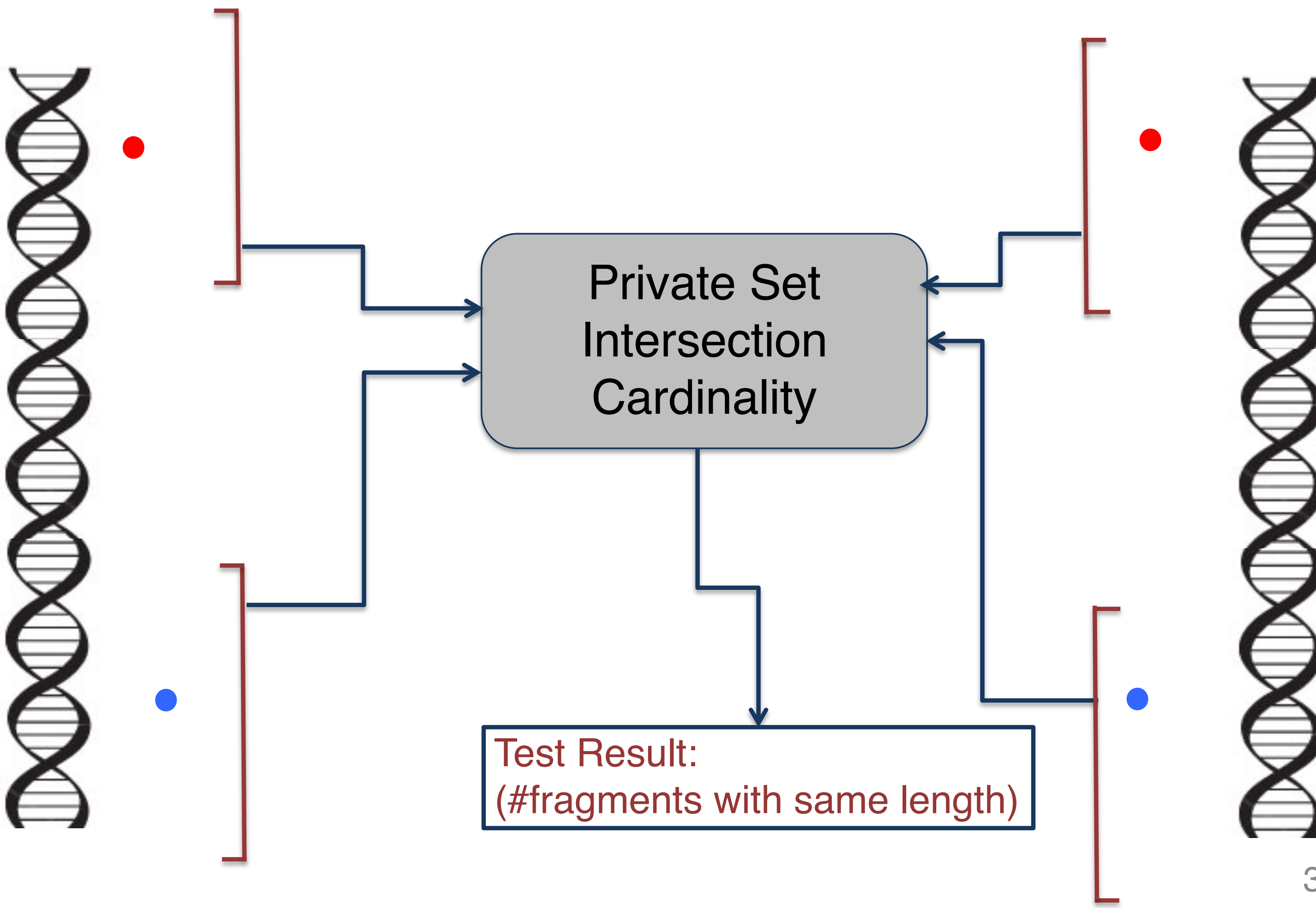
~99.5% of any two human genomes are identical

Why don’t we compare *only* the remaining 0.5%?

We can compare by counting how many

But… We don’t know (yet?) where *exactly* this 0.5% occur!

Private RFLP-based Paternity Test



Personalized Medicine (PM)

Drugs designed for patients' genetic features

- Associating drugs with a unique genetic fingerprint
- Max effectiveness for patients with matching genome
- **Test drug's "genetic fingerprint" against patient's genome**

Examples:

- *tmpt* gene – relevant to leukemia
  - (1) G->C mutation in pos. 238 of gene's c-DNA, or (2) G->A mutation in pos. 460 and one A->G is pos. 419 cause the *tpmt* disorder (relevant for leukemia patients)
- *hla-B* gene – relevant to HIV treatment
  - One G->T mutation (known as *hla-B*5701* allelic variant) is associated with extreme sensitivity to abacavir (HIV drug)

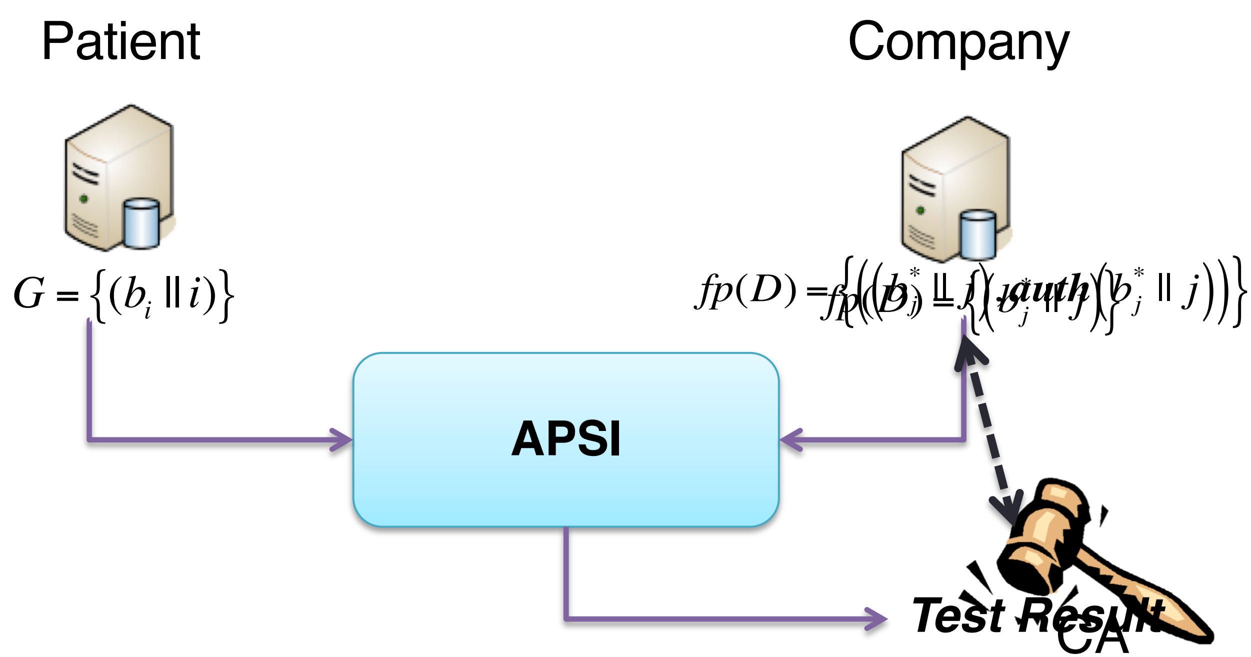
Reducing P³MT to APSI

Intuition:

FDA acts as *CA*, Pharmaceutical company as *Client*, Patient as *Server*

Patient's private input set: $G = \left\{ (b_i \,\|\, i) \,\middle|\, b_i \in \{A, C, G, T\} \right\}_{i=1}^{3 \cdot 10^9}$

Pharmaceutical company's input set: $fp(D) = \left\{ \left(b_j^* \,\|\, j \right) \right\}$



Multiple Parties?

Sharing Statistics?

Examples:

1. Smart metering
2. Recommender systems for online streaming services
3. Statistics about mass transport movements
4. Traffic statistics for the Tor Network

How about privacy?

Private Recommendations

The BBC keeps 500-1000 free programs on iPlayer

- No tracking, no ads (taxpayer funded)

Valuable to gather statistics, give recommendations

- “You might also like”
- E.g., “similar” users have watched both Dr Who and Sherlock Holmes, you have only watched Sherlock, why don’t you watch Dr Who?

Item-KNN Recommendation

Predict favorite items for users based on their own ratings and those of “similar” users

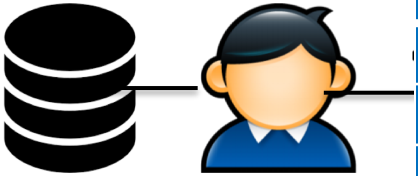
Consider ***N*** users, ***M*** TV programs and binary ratings (viewed/not viewed)

Build a co-views matrix ***C***, where C_{ab} is the number of views for the pair of programs (a,b)

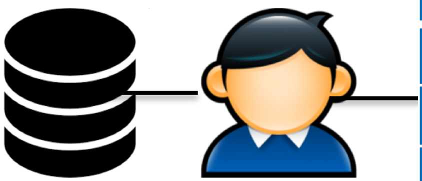
Compute the **Similarity Matrix**

$$\{Sim\}_{ab} = \frac{C_{ab}}{\sqrt{C_a \cdot C_b}}$$

Identify K-Neighbours (***KNN***) based on matrix

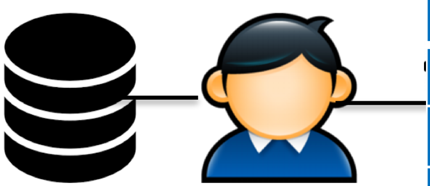


	Dr Who	Sherlock	Earth
Dr Who	1	-	-
Sherlock	1	1	-
Earth	0	0	0



	Dr Who	Sherlock	Earth
Dr Who	1	-	-
Sherlock	1	1	-
Earth	1	1	1

	Dr Who	Sherlock	Earth
Dr Who	3	-	-
Sherlock	2	2	-
Earth	1	1	1



	Dr Who	Sherlock	Earth
Dr Who	1	-	-
Sherlock	0	0	-
Earth	0	0	0

Privacy-Preserving Aggregation

Goal: aggregator collects matrix, s.t.

- Can only learn aggregate counts (e.g., 237 users have watched both a and b)
- Not who has watched what

Use additively homomorphic encryption?

- $Enc_{PK}(a) * Enc_{PK}(b) = Enc_{PK}(a+b)$
- How can I used it to collect statistics?

Keys summing up to zero

Users $U_1, U_2, \ldots, U_N$

Each has $k_1, k_2, \ldots, k_N$ s.t. $k_1+k_2+\ldots+k_N=0$

Now how can I use this?

User $\mathcal{U}_i$ $(i \in [1, N])$ | Tally

$x_i \in_r \mathbb{G}, y_i := g^{x_i} \bmod q \quad \xrightarrow{\quad y_i \quad}$

$k_{i_\ell} := \sum_{j \neq i} \mathrm{H}(y_j^{x_i} || \ell || s) \cdot (-1)^{i>j} \bmod 2^{32} \quad \xleftarrow{\quad \{y_j\}_{j \in [1,N]} \quad}$

$b_{i_\ell} := X_{i_\ell} + k_{i_\ell} \bmod 2^{32} \quad \xrightarrow{\quad \{b_{i_\ell}\}_{\ell=1}^{L} \quad}$ Fault recovery (if needed)

$\xleftarrow{\quad \mathcal{U}^{on} \quad}$

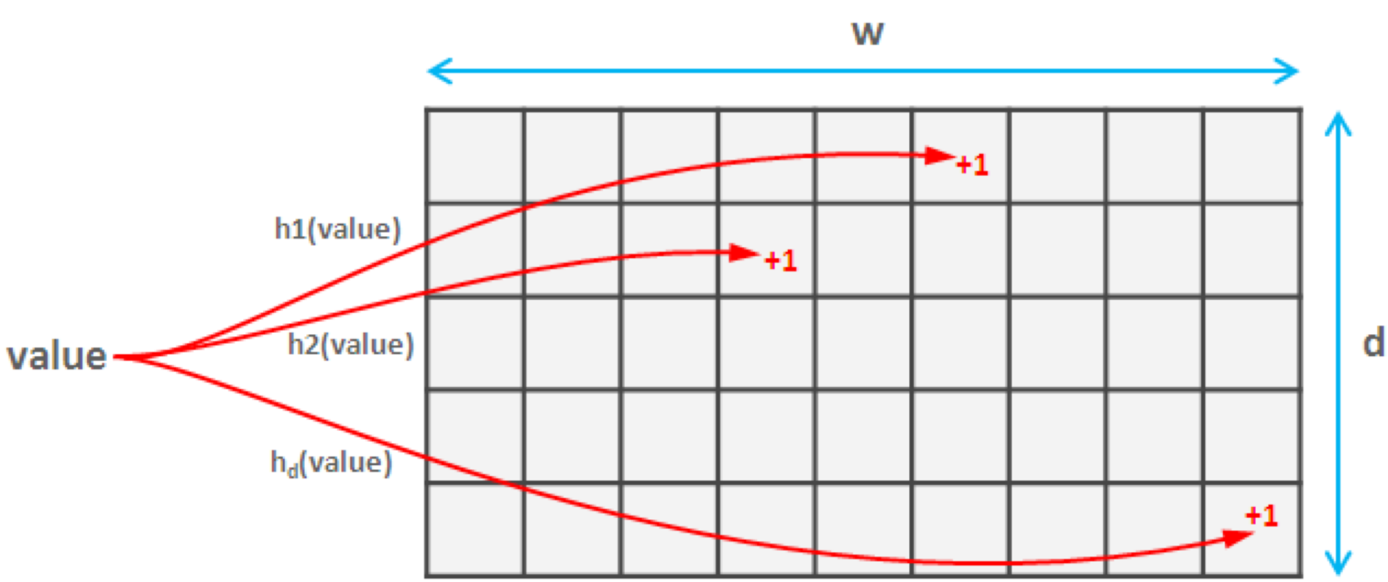
$k'_{i_\ell} := \sum_{\substack{j \neq i, \\ j \notin \mathcal{U}^{on}}} \mathrm{H}(y_j^{x_i} || \ell || s) \cdot (-1)^{i>j} \bmod 2^{32} \quad \xrightarrow{\quad \{k'_{i_\ell}\}_{\ell=1}^{L} \quad} \quad C'_\ell := \left(\sum_{i \in \mathcal{U}^{on}} b_{i_\ell} - \sum_{i \in \mathcal{U}^{on}} k'_{i_\ell} \right) \bmod 2^{32}$

Is this efficient?

Preliminaries: Count-Min Sketch

An estimate of an item's frequency in a stream

- Mapping a stream of values (of length T) into a matrix of size O(logT)
- The sum of two sketches results in the sketch of the union of the two data streams



Security & Implementation

Security

In the honest-but-curious model under the CDH assumption

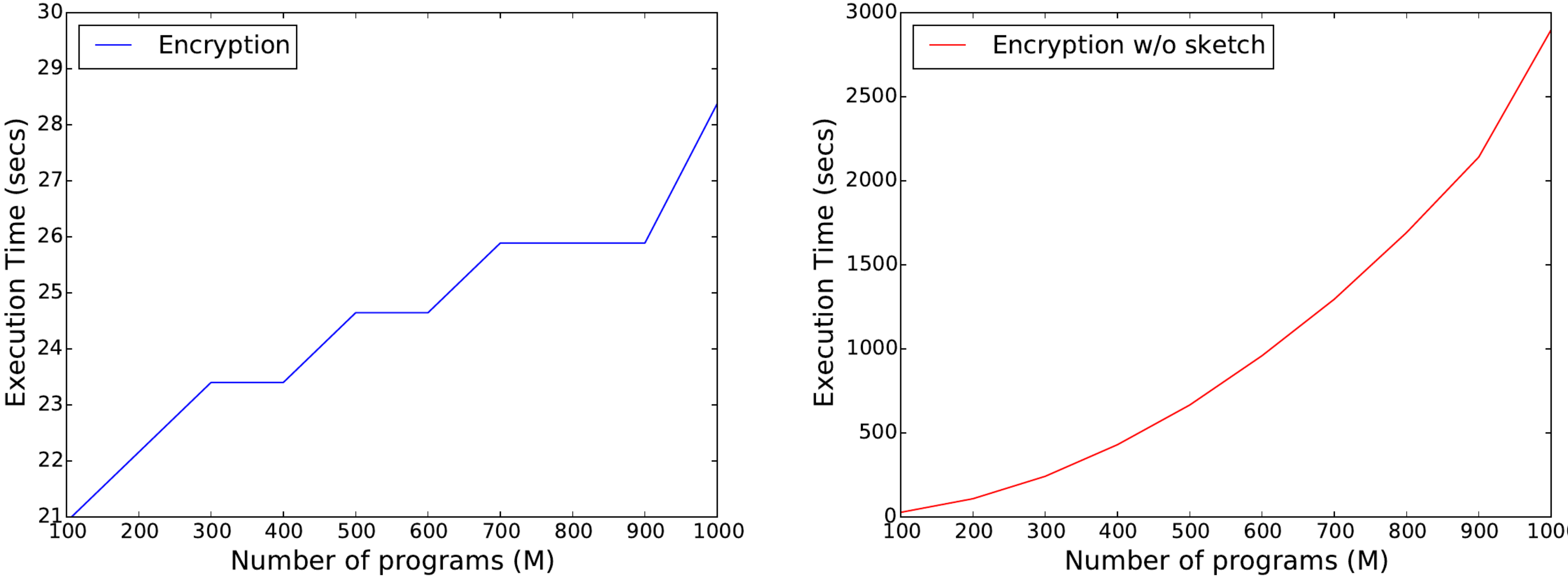
Prototype implementation:

Tally as a Node.js web server

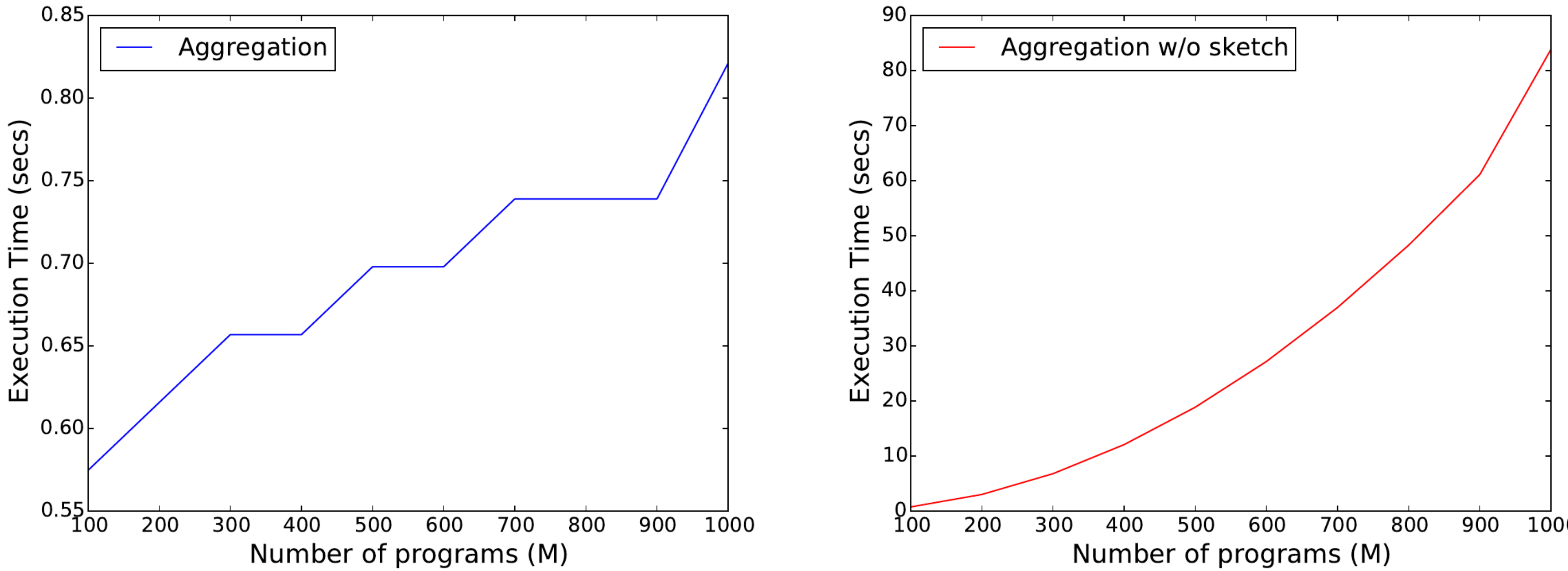
Users run in the browser or as a mobile cross-platform application (Apache Cordova)

Transparency, ease of use, ease of deployment

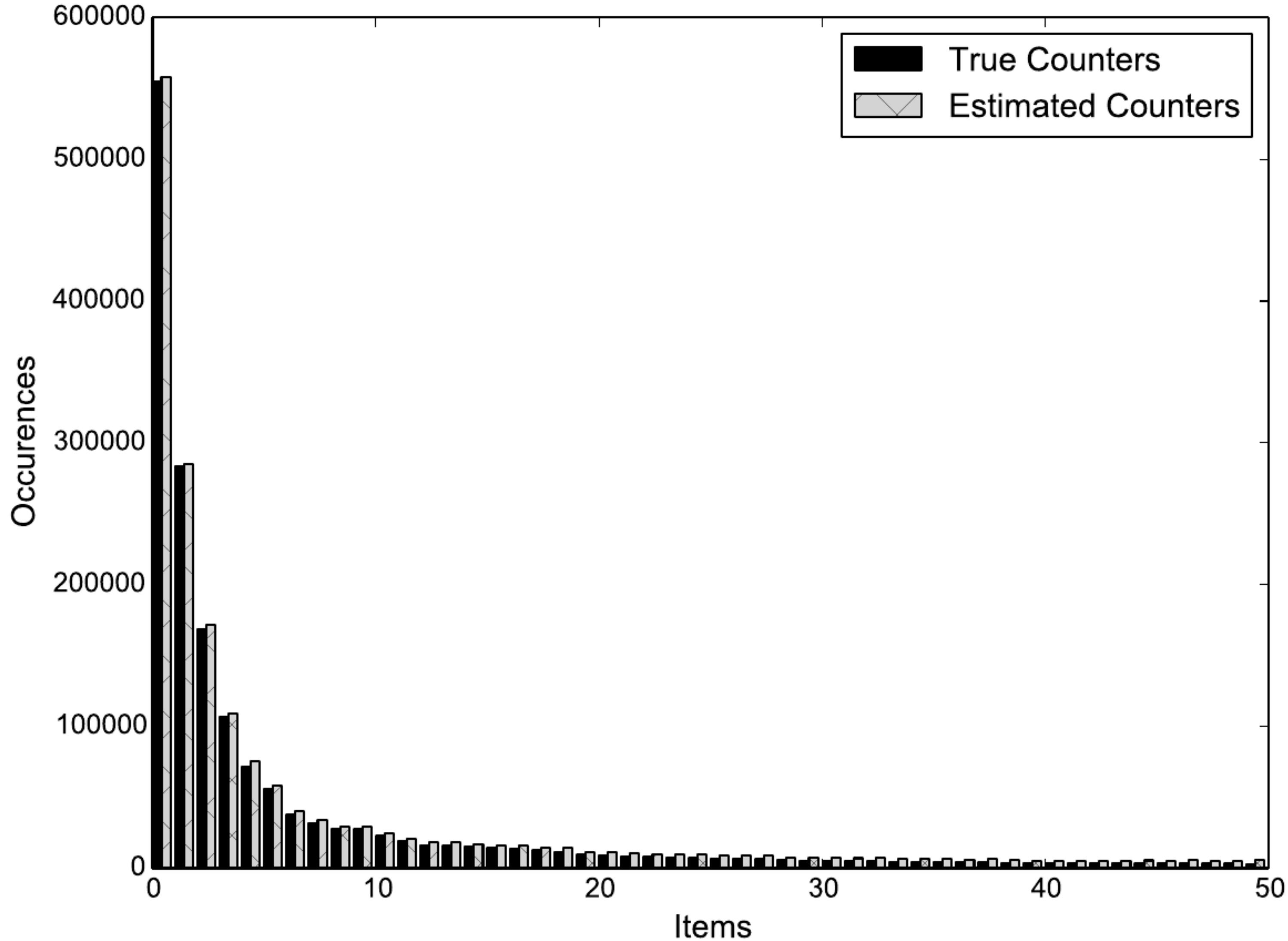
User side



Server side



Accuracy



True Counters
Estimated Counters
Occurences
Items
0
100000
200000
300000
400000
500000
600000
10
20
30
40
50

Tor Hidden Services

Aggregate statistics about the number of hidden service descriptors from multiple HSDirs

Median statistics to ensure robustness

See Melis, Danezis, De Cristofaro, Efficient Private Statistics with Succinct Sketches. NDSS'16

Mobility Analytics

Use location/movement data to improve urban and transportation planning

- Google Maps, Waze
- Telefonica's SmartSteps

Mmm… what about privacy?

- Infer life-style, political/religious inclinations
- Anonymization ineffective

How about using only aggregate statistics?

- How many people at location X at time t? (Not who)

Our work in this space

1. Mobility analytics using aggregate locations? [1]

Is it useful? What tasks can we perform?

2. How much privacy do aggregates leak? [2]

How can we quantify it?

3. Identify users contributing to aggregates [3]?

Membership inference attacks?

[1] Apostolos Pyrgelis, Gordon Ross, Emiliano De Cristofaro. Privacy-Friendly Mobility Analytics using Aggregate Location Data. In ACM SIGSPATIAL 2016

[2] Apostolos Pyrgelis, Carmela Troncoso, Emiliano De Cristofaro. What Does The Crowd Say About You? Evaluating Aggregation-based Location Privacy. In PETS 2017

[3] Apostolos Pyrgelis, Carmela Troncoso, Emiliano De Cristofaro. Knock Knock, Who's There? Membership Inference on Aggregate Location Data. NDSS 2018. **Distinguished Paper Award.**

Mobility & Privacy

Aggregation often considered as a privacy defense [NDSI'12, CCS'15, NDSS'16]

- **But do users lose privacy from the aggregates?**

Differential Privacy (DP) to the rescue?

- Add noise to the statistics to bound the privacy leakage
- (Input or output perturbation)

The problem with DP…

- Does it really tell us about the privacy loss?
- Epsilon gives a theoretical upper-bound (indistinguishability)
- How do we tune it? What does it mean in practice?

TFL Data

Logs of anonymized oyster card trips including Underground (LUL), National Rail (NR), Overground (LRC), Docklands Light Railway (DLR)

Monday, March 1 to Sunday, March 28, 2010

60 million trips as performed by 4 million unique users, over 582 stations

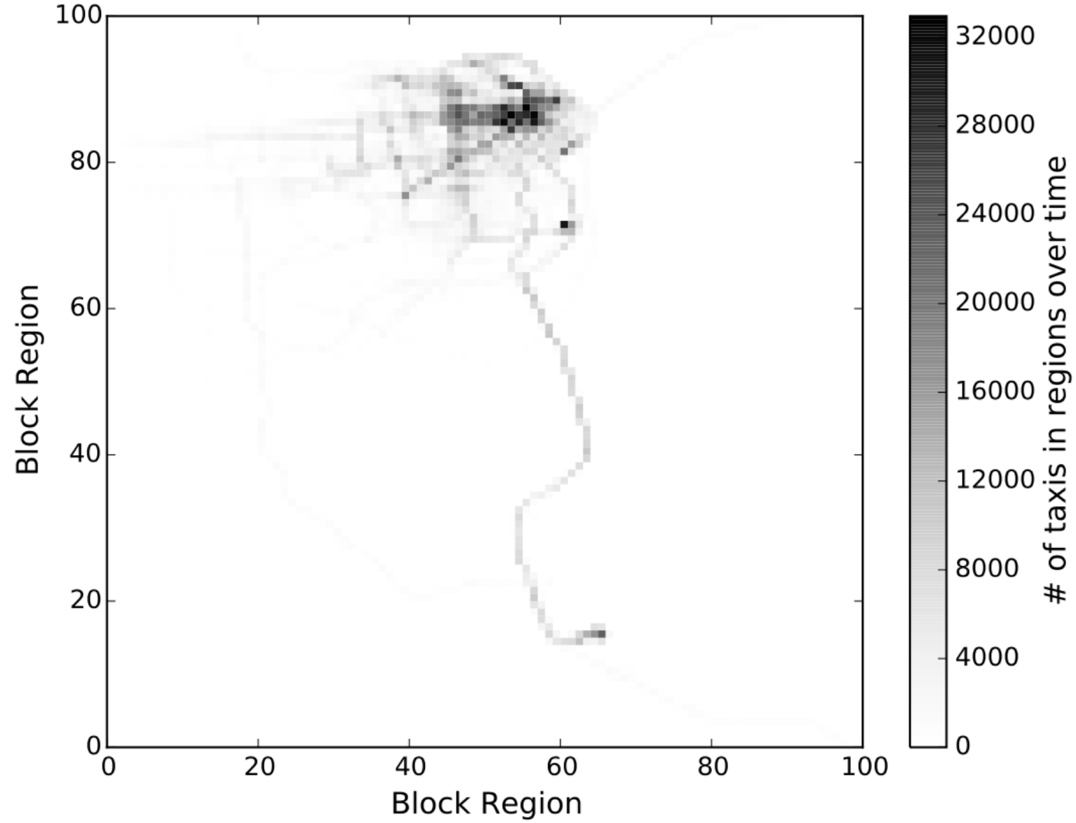
San Francisco Cabs (SFC)

Mobility traces of 536 cabs in SF (May 19 to June 8, 2008)

11 million GPS coords

San Francisco grid of 100 x 100 regions

0.19 × 0.14 sq mi



Membership Inference

Given a set of aggregates over some locations and some time slots…

Can you distinguish whether user u* was part of those aggregates?

Game Parameters: (U, m, T_I)

Adv($\mathcal{P}$) | **Ch**($\mathcal{L}$)

Pick $u^* \in U$ $\xrightarrow{u^*}$

$\Upsilon \subset_\$ U \setminus \{u^*\}$ of size $m-1$
$b \leftarrow_\$ \{0,1\}$

If $b == 0$:
$U_0 := \Upsilon \cup \{u^*\}$
If $b == 1$:
$u \leftarrow_\$ U \setminus \{u^*\} \setminus \Upsilon$
$U_1 := \Upsilon \cup \{u\}$

$\forall s \in S,\ \forall t \in T_I,$
$A_{U_b}[s,t] := \sum_{j \in U_b} L_j[s,t]$

$\xleftarrow{A_{U_b}}$

$b' \leftarrow d(u^*, A_{U_b}, m, T_I, \mathcal{P})$
Output $b' \in \{0,1\}$

Methodology

Model **adversarial prior knowledge**

1. Knows ground truth for a subset of locations for a while, i.e., which users were there
2. Knows ground truth for a subset of users, i.e., whether they were part of the aggregates

Model task as a **distinguishing function**

On input target u*, parameters of the game, and aggregates, decide yes/no

We use a supervised machine learning classifier trained on the prior

Metrics

Standard Area Under the Curve (AUC)

- Count TP, FP, TN, FN for the task, derive ROC curve, compute AUC

Privacy Loss (PL)

- Advantage over random guess (0.5)

Experiments TL;DR

(See paper for plots, detailed, experiments, etc.)

Membership inference works quite well overall

- Privacy loss is never negligible, even for large groups

Adversarial performance does not depend only on size of the groups, but also on prior and characteristics of the dataset

- TFL commuters lose more privacy than SFC cabs (regular vs unpredictable)

How about DP Aggregates?

Established framework to release statistics that are free from inference is differential privacy (DP)

- Don’t release raw aggregates but noisy ones
- Use Laplace, Gaussian, Fourier Perturbation, etc.

How much privacy do you gain?

1. Train on raw aggregates from prior knowledge
2. Add noise on prior knowledge, train on noisy aggregates

DP Experiments TL;DR

Overall, DP does work to reduce the extent of membership inference

However… we find out, among other things:

- Training on noisy aggregates much more effective
- Privacy gain decreases very fast with smaller ε values
- Poor utility overall for Laplace and Gaussian
- Fourier retains utility but only for large-ish ε

The Road Ahead…

This slide is intentionally left blank