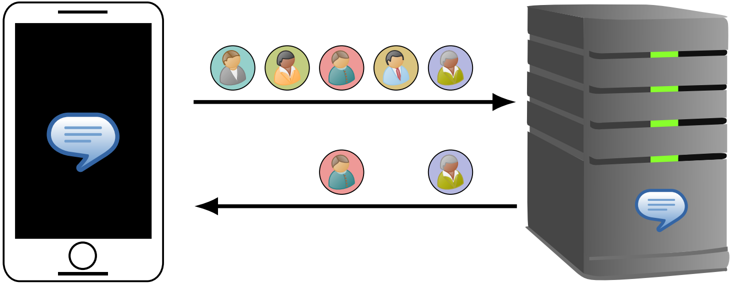


Mobile Private Contact Discovery

https://contact-discovery.github.io/

Daniel Kales

Secure Messaging Summit, September 3rd, 2020

Outline

Contact Discovery

Existing Approaches

Private Set Intersection

- using Oblivious Pseudorandom Functions
- using Private Information Retrieval
- using Fully Homomorphic Encryption

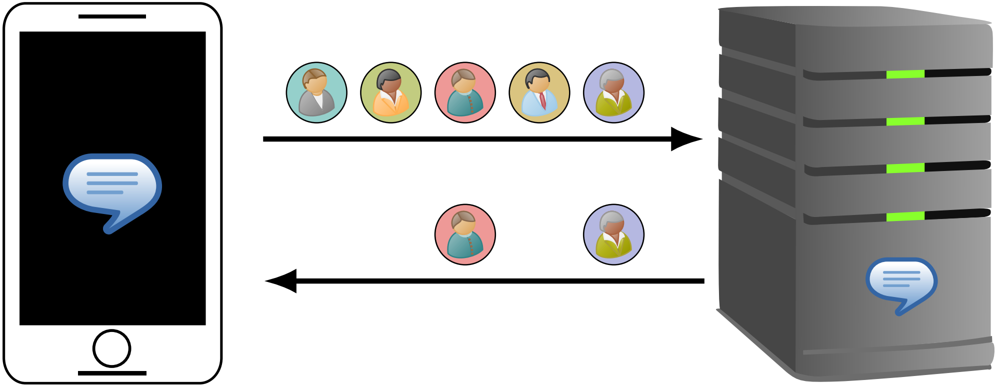
Conclusion & Outlook

Contact Discovery

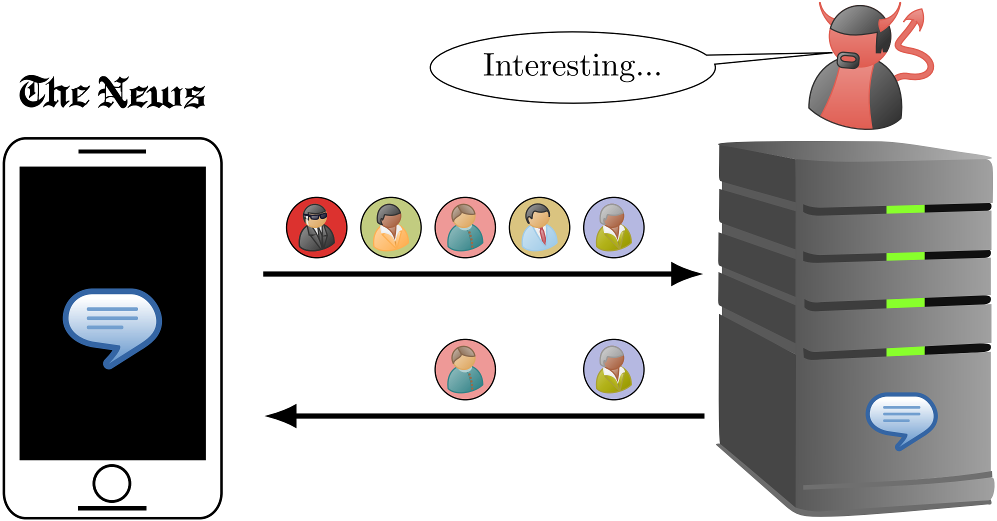
Finding your friends

Mobile Contact Discovery

Procedure executed when new user signs up to messaging service.



Privacy Concerns!

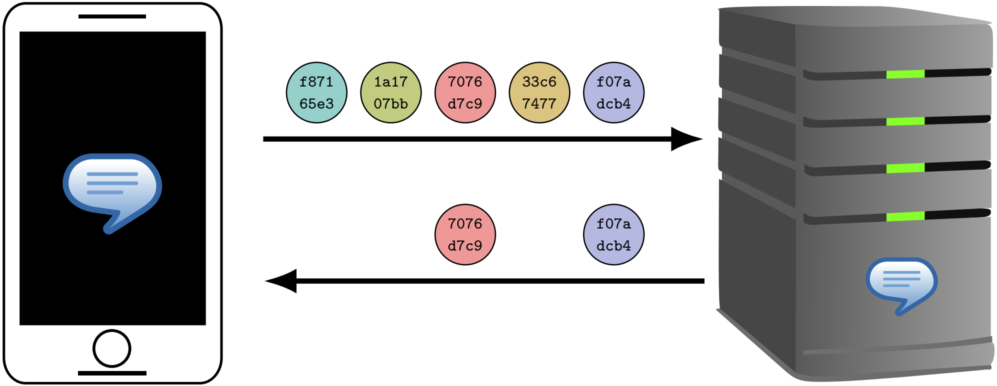


Existing Approaches

What is done today?

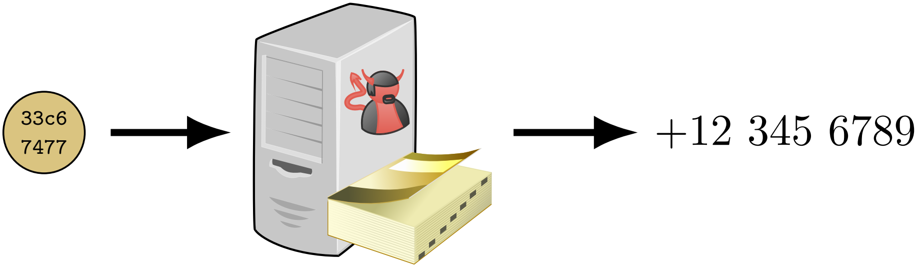
A naive Solution - Hashing

Basic Idea: Send hashes of phone numbers instead



A naive Solution - Hashing (cont.)

Problem: Phone Numbers do not have a lot of entropy!



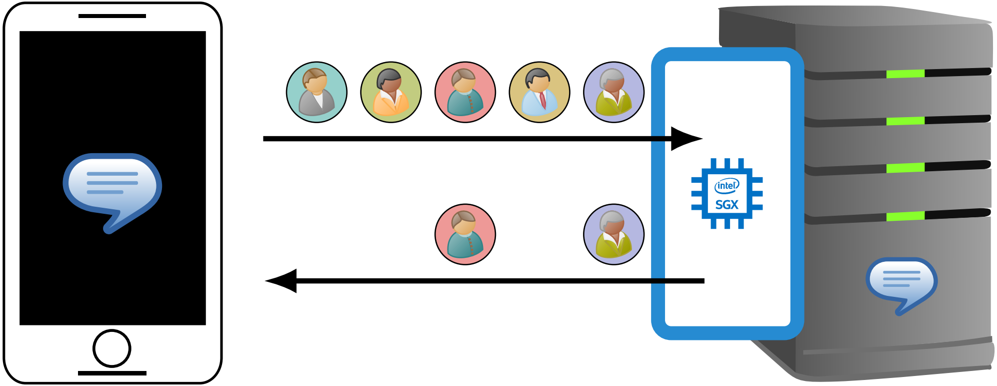
33c6 7477

$+12\ 345\ 6789$

- Easy for powerful server to brute-force hashes
  - Hash cracking tools, rainbow tables,…
  - Even salts do not help (much) against targeted attacks

Trusting Hardware

Perform contact discovery in trusted execution environment.



Existing Situation in the Mobile Messaging World

We performed a survey in our 2019 paper "Mobile Private Contact Discovery at Scale".

Messenger	Naïve Hashing	Analysis Technique
Confide*	✓	Privacy Policy
Dust*	✗	Network Traffic
Eleet*	✗	Privacy Policy
G DATA Secure Chat	✓	Network Traffic
Signal (legacy / non-SGX)	✓	Source Code
SIMSme	✓	Network Traffic
Telegram	✗	Privacy Policy
Threema	✓	Privacy Policy
Viber	✗	Privacy Policy
WhatsApp	✗	Privacy Policy
Wickr Me	✓	Privacy Policy
Wire	✓	Privacy Policy

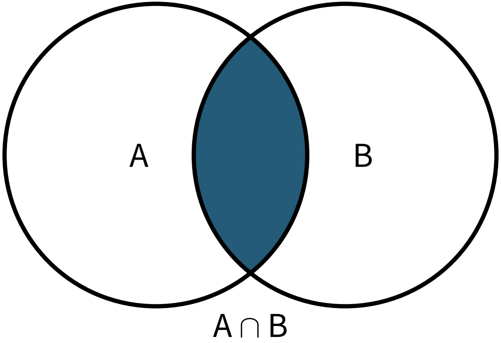
*contact discovery is optional

Private Set Intersection

$A \cap B$ (but with privacy)

Background - Private Set Intersection

- Compute intersection of two sets
- Privacy-preserving (other party learns nothing about items outside intersection)



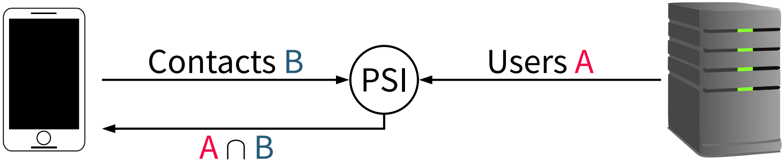
Background - Parameters in PSI

Many different scenarios for PSI

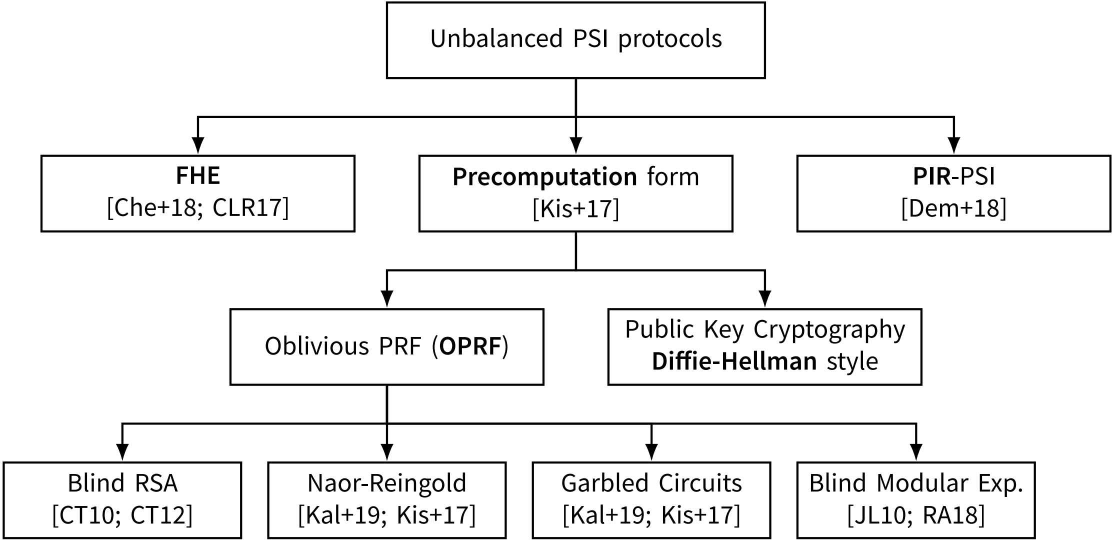
- Balanced vs. unbalanced set sizes
- Security against semi-honest vs. malicious parties
- Leakage of parties' set sizes allowed?
- Different cryptographic building blocks
  - Generic multiparty computation
  - Public-key cryptography
  - Oblivious transfer

PSI for Mobile Private Contact Discovery

- Popular messengers have millions, if not billions of users.
  - typical phone address books have 100-1000 contacts.
  - → unbalanced PSI
- “The poster child of use-cases for unbalanced PSI”



Unbalanced PSI Protocols



Unbalanced PSI protocols

FHE
[Che+18; CLR17]

Precomputation form
[Kis+17]

PIR-PSI
[Dem+18]

Oblivious PRF (**OPRF**)

Public Key Cryptography
Diffie-Hellman style

Blind RSA
[CT10; CT12]

Naor-Reingold
[Kal+19; Kis+17]

Garbled Circuits
[Kal+19; Kis+17]

Blind Modular Exp.
[JL10; RA18]

Oblivious Pseudorandom Functions

Problem with hash-based solution:

- No secret information, server can brute-force hash

Idea: What if we “encrypt” items instead?

- We cannot give both parties key (essentially equal to hashing with salt)

Oblivious Pseudorandom Functions

Problem with hash-based solution:

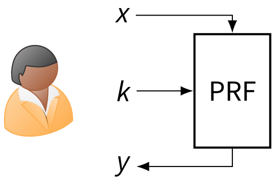
- No secret information, server can brute-force hash

Idea: What if we "encrypt" items instead?

- We cannot give both parties key (essentially equal to hashing with salt)

Pseudorandom Function

$$y = \mathrm{PRF}_k(x)$$



Oblivious Pseudorandom Functions

Problem with hash-based solution:

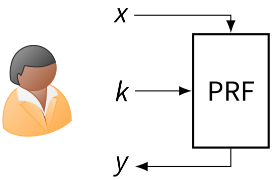
- No secret information, server can brute-force hash

Idea: What if we “encrypt” items instead?

- We cannot give both parties key (essentially equal to hashing with salt)

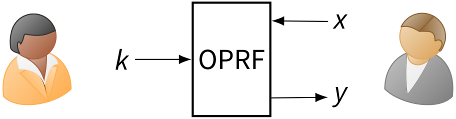
Pseudorandom Function

$$y = \mathrm{PRF}_k(x)$$



Oblivious Pseudorandom Function

$$y = \mathrm{PRF}_k(x)$$



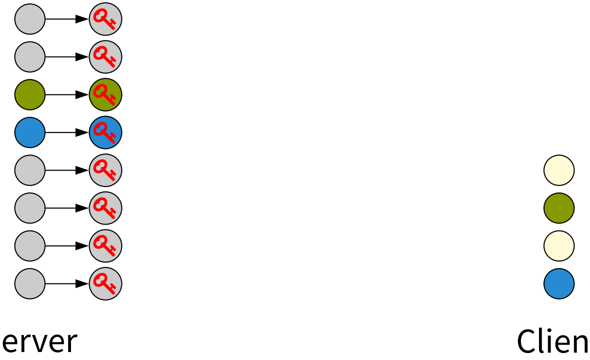
PSI using OPRF Evaluation

Basic protocol idea:

Server
Client

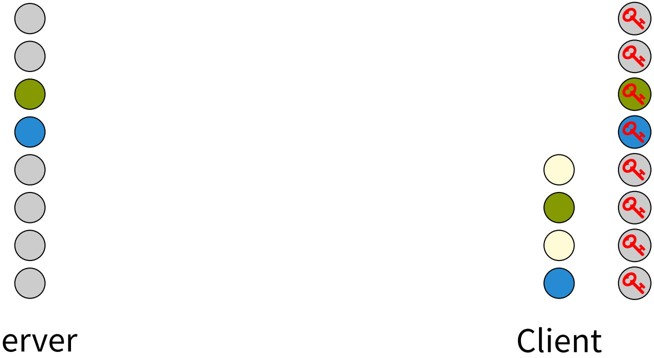
PSI using OPRF Evaluation

Basic protocol idea:



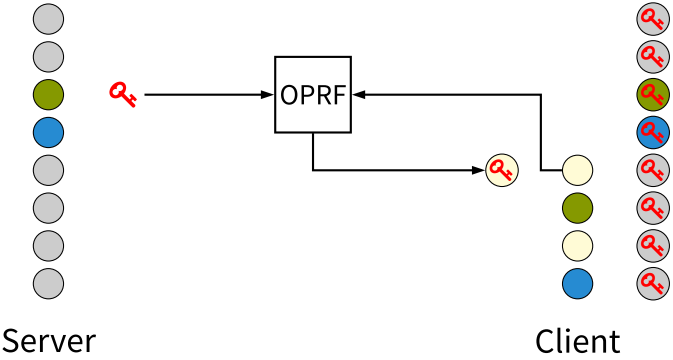
PSI using OPRF Evaluation

Basic protocol idea:



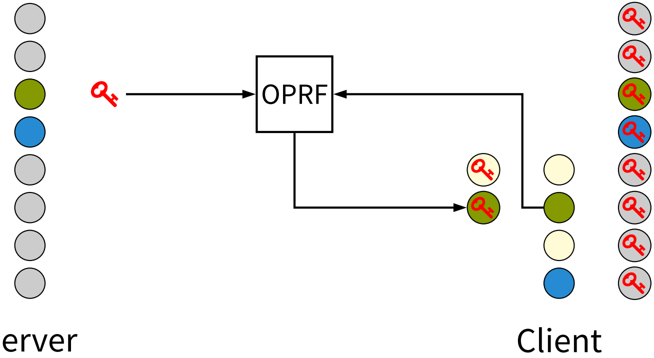
PSI using OPRF Evaluation

Basic protocol idea:



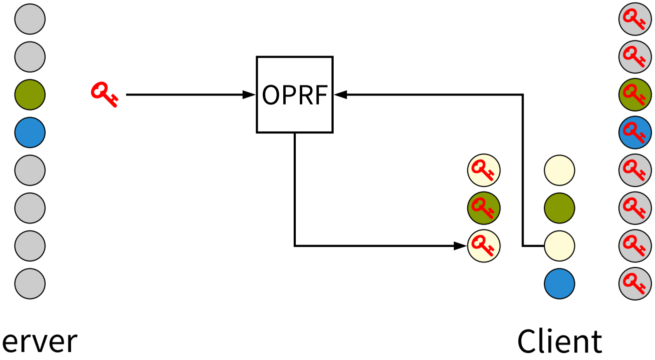
PSI using OPRF Evaluation

Basic protocol idea:



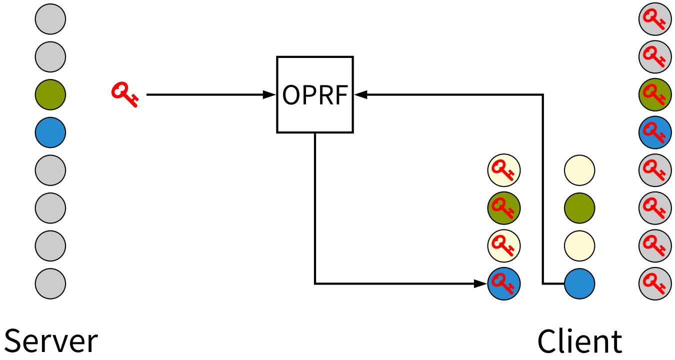
PSI using OPRF Evaluation

Basic protocol idea:



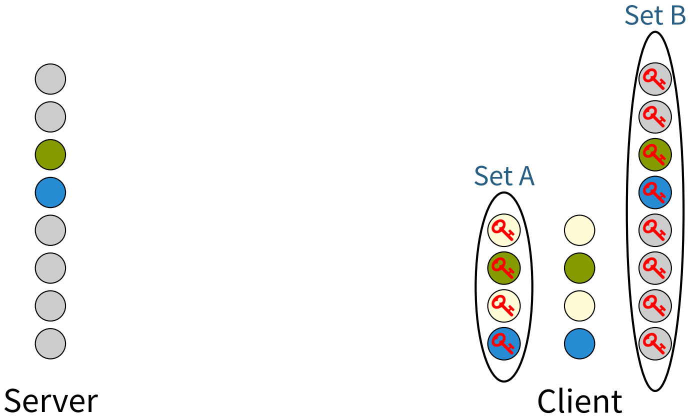
PSI using OPRF Evaluation

Basic protocol idea:



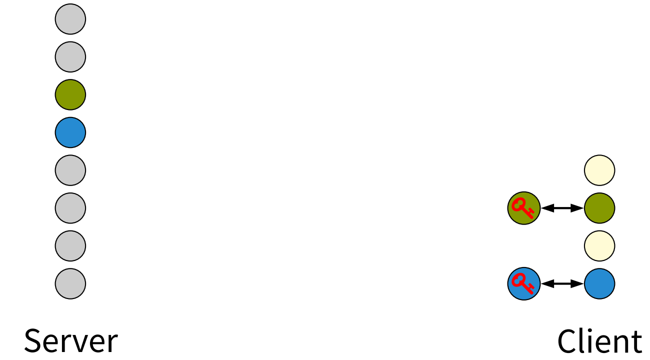
PSI using OPRF Evaluation

Basic protocol idea:



PSI using OPRF Evaluation

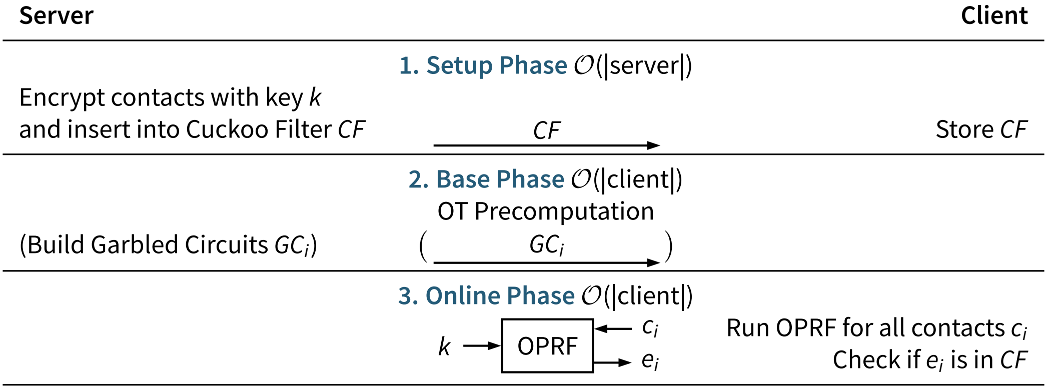
Basic protocol idea:



OPRF-based PSI for Unequal Set Sizes

Kiss et al. [Kis+17] explored unbalanced PSI for mobile use-cases.

- Split into Setup, Base, and Online phases



Server **Client**

1. Setup Phase $\mathcal{O}(|\text{server}|)$

Encrypt contacts with key k
and insert into Cuckoo Filter CF $\xrightarrow{CF}$ Store CF

2. Base Phase $\mathcal{O}(|\text{client}|)$
OT Precomputation

(Build Garbled Circuits GC_i) $\left(\xrightarrow{GC_i} \right)$

3. Online Phase $\mathcal{O}(|\text{client}|)$

$k \rightarrow \boxed{\text{OPRF}}$ $\leftarrow c_i$, $\rightarrow e_i$

Run OPRF for all contacts c_i
Check if e_i is in CF

Mobile Private Contact Discovery at Scale [Kal+19]

Our improvements over previous work

- Security against malicious receiver at negligible cost
- Lower communication
  - Use of LowMC instead of AES for garbled circuits
  - ECC version of Naor-Reingold PRF
- Better Cuckoo Filter parameters and novel compression
- High-performance native ARMv8-A implementation
  - Up to 1000x performance gain

Paper and Implementation at
contact-discovery.github.io

Mobile Private Contact Discovery at Scale (cont.)

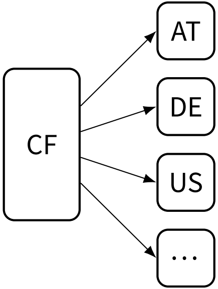
Parameters		PSI Protocol	Base + Online Time [s]		Communication [MiB]	
$\|\text{Server}\|$	$\|\text{Client}\|$		WiFi	LTE	$S \rightarrow C$	$S \leftarrow C$
2^{28}	1 024	LowMC-GC-PSI	3.54	8.59	22.01	2.02
		ECC-NR-PSI	**2.92**	**6.53**	**4.07**	**2.00**
	1	LowMC-GC-PSI	0.17	0.18	0.04	0.02
		ECC-NR-PSI	**0.13**	**0.13**	**0.01**	**0.01**

- Fast online phase ($\mathcal{O}(|\text{Client}|)$)
- Downside: large one-time setup transfer ($\mathcal{O}(|\text{Server}|)$)
  - Size of initial cuckoo filter for 2^{28} contacts is 1 GiB
  - Size of initial cuckoo filter for 2^{20} contacts is 4 MiB

Privacy Tradeoff: Database Sharding

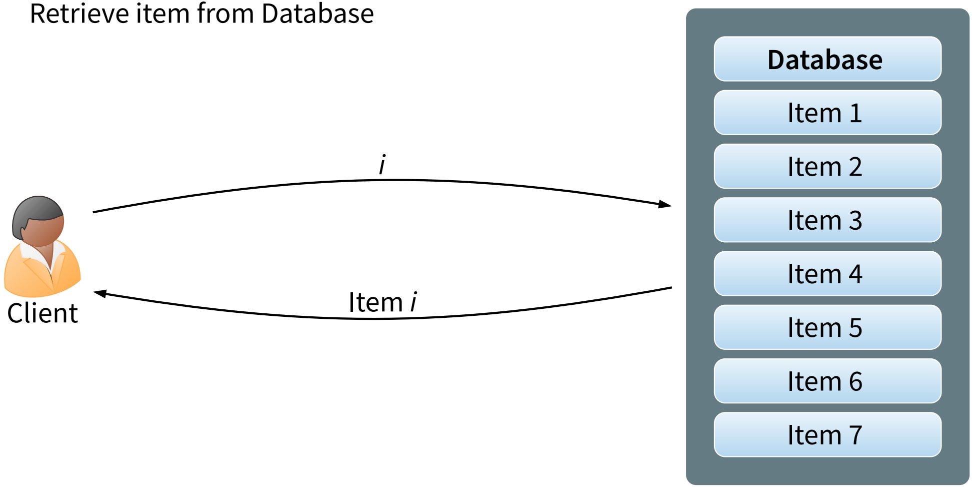
Solution to reduce data transfer for cuckoo filter

- Split into region-based shards
  - problem: leaks information
  - e.g., person has a contact in a different country
- Split into random shards
  - e.g., based on hash-prefix of phone number
  - Reduced leaks, but gets less efficient for many contacts



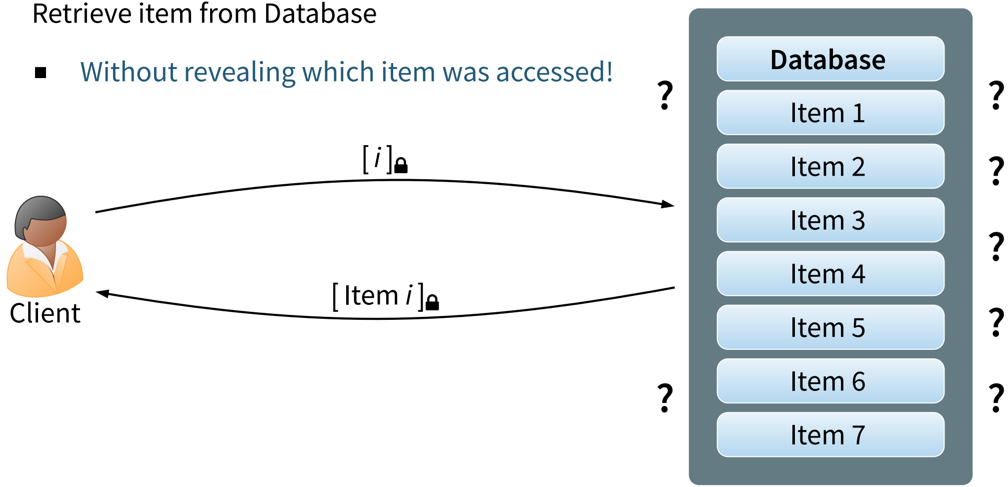
Private Information Retrieval

- Retrieve item from Database



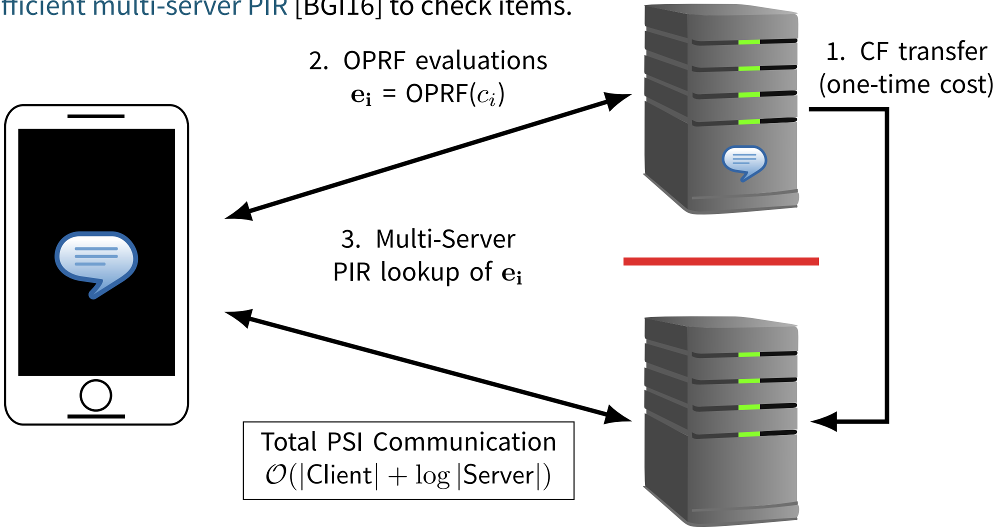
Private Information Retrieval

- Retrieve item from Database
  - Without revealing which item was accessed!



Combining OPRF-PSI with PIR

Use efficient multi-server PIR [BGI16] to check items.



1. CF transfer (one-time cost)

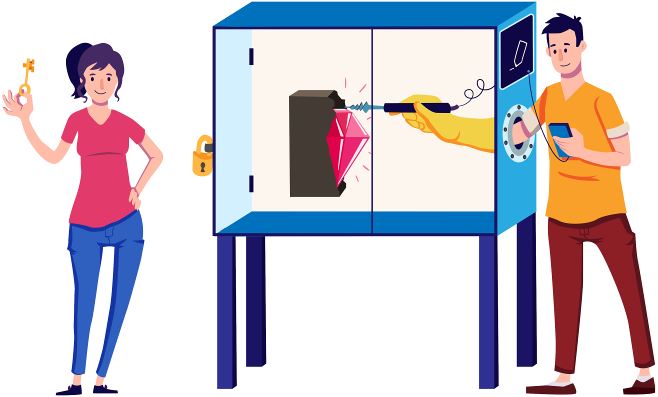
2. OPRF evaluations $\mathbf{e_i}$ = OPRF(c_i)

3. Multi-Server PIR lookup of $\mathbf{e_i}$

Total PSI Communication $\mathcal{O}(|\text{Client}| + \log|\text{Server}|)$

Fully Homomorphic Encryption (FHE)

FHE enables us to perform operations on encrypted data.

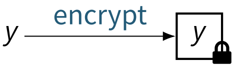


PSI using FHE (basic protocol)

Client	**Server**
y	x_1
	x_2
	x_3
	x_4

PSI using FHE (basic protocol)

Client



Server

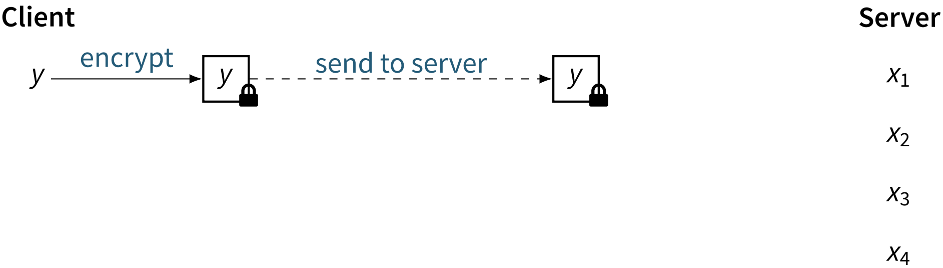
x_1

x_2

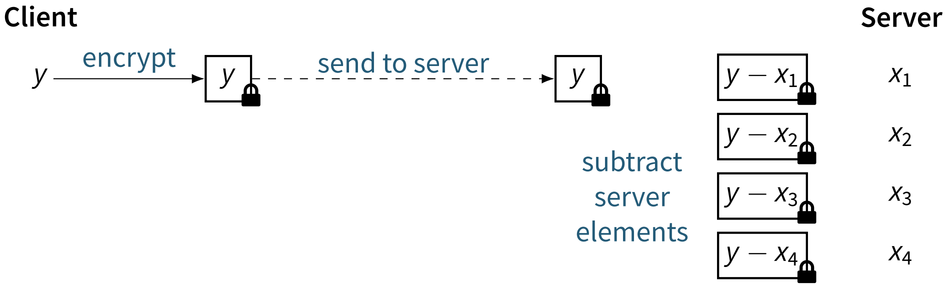
x_3

x_4

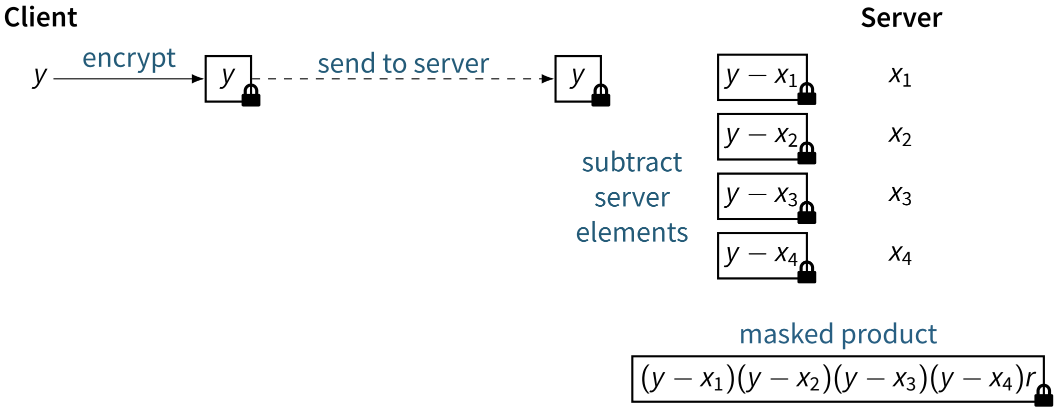
PSI using FHE (basic protocol)



PSI using FHE (basic protocol)



PSI using FHE (basic protocol)



Client

Server

y

encrypt

$\boxed{y}$

send to server

$\boxed{y}$

subtract server elements

$\boxed{y - x_1}$ x_1

$\boxed{y - x_2}$ x_2

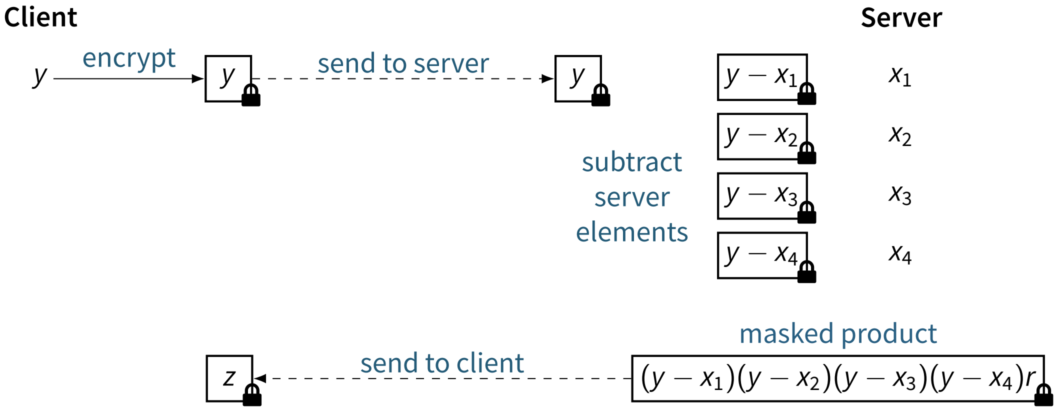
$\boxed{y - x_3}$ x_3

$\boxed{y - x_4}$ x_4

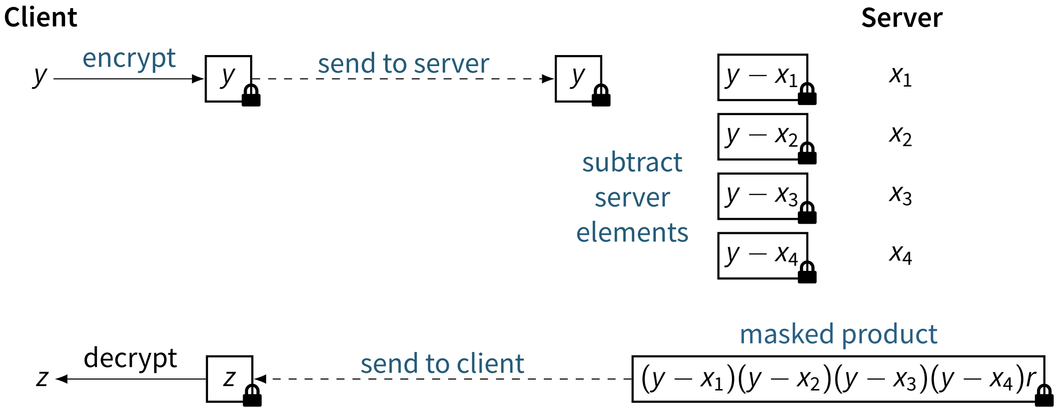
masked product

$\boxed{(y - x_1)(y - x_2)(y - x_3)(y - x_4)r}$

PSI using FHE (basic protocol)



PSI using FHE (basic protocol)



Client

Server

y encrypt $\boxed{y}$ send to server $\boxed{y}$

subtract server elements

$\boxed{y - x_1}$ x_1

$\boxed{y - x_2}$ x_2

$\boxed{y - x_3}$ x_3

$\boxed{y - x_4}$ x_4

masked product

$\boxed{(y - x_1)(y - x_2)(y - x_3)(y - x_4)r}$

send to client $\boxed{z}$ decrypt z

$$z = \begin{cases} 0 & \text{if} \quad y \in X, \\ \text{random} & \text{otherwise.} \end{cases}$$

Performance of FHE-based approaches

- Lots of additional optimizations ([Che+18; CLR17])
  - SIMD HE operations, Cuckoo Hashing, OPRF pre-processing, …
- Communication complexity: $\mathcal{O}(|\text{Client}|)$
  - No large offline transfer needed!
- Computational complexity: $\mathcal{O}(|\text{Server}|)$
  - Expensive FHE operations!

$\lvert\text{Server}\rvert$	$\lvert\text{Client}\rvert$	Offline [s]	Online [s]	Communication [MB]
2^{28}	1024	4 628 (32 threads)	12.1 (32 threads)	18.57

Conclusion & Outlook

The Quest for efficient unbalanced PSI protocols

PSI is a highly active research topic!

- New papers at top-tier conferences each year
  - Most focused on balanced set sizes
- OPRF-based solutions need more efficient offline phase
- FHE-based solutions need faster FHE schemes

Goals for practical deployment:

# registered users	$>$ 1 billion
# Entries per address book	10 000
Latency	$<$ 2s
Communication	$<$ 10 MiB

Limitations of PSI

Even perfectly secure and efficient PSI cannot protect against all attacks:

- Enumeration attacks
  - Try to find out which numbers are registered with a service
  - Countermeasure: Rate limiting
- Metadata leakage in Contact Discovery APIs
  - Some solutions send (a lot of) additional information
  - Attacks on existing Contact Discovery APIs
    - Brand-new paper at `https://contact-discovery.github.io`
    - Closer look at APIs of WhatsApp, Signal, Telegram

Questions

?

The End

Contact Discovery

Existing Approaches

Private Set Intersection

- using Oblivious Pseudorandom Functions
- using Private Information Retrieval
- using Fully Homomorphic Encryption

Conclusion & Outlook

References I

[BGI16] Elette Boyle, Niv Gilboa, and Yuval Ishai. **Function Secret Sharing: Improvements and Extensions.** ACM Conference on Computer and Communications Security. ACM, 2016, pp. 1292–1303.

[Che+18] Hao Chen, Zhicong Huang, Kim Laine, and Peter Rindal. **Labeled PSI from Fully Homomorphic Encryption with Malicious Security.** ACM Conference on Computer and Communications Security. ACM, 2018, pp. 1223–1237.

[CLR17] Hao Chen, Kim Laine, and Peter Rindal. **Fast Private Set Intersection from Homomorphic Encryption.** ACM Conference on Computer and Communications Security. ACM, 2017, pp. 1243–1255.

[CT10] Emiliano De Cristofaro and Gene Tsudik. **Practical Private Set Intersection Protocols with Linear Complexity.** Financial Cryptography. Vol. 6052. Lecture Notes in Computer Science. Springer, 2010, pp. 143–159.

[CT12] Emiliano De Cristofaro and Gene Tsudik. **Experimenting with Fast Private Set Intersection.** TRUST. Vol. 7344. Lecture Notes in Computer Science. Springer, 2012, pp. 55–73.

References II

[Dem+18] Daniel Demmler, Peter Rindal, Mike Rosulek, and Ni Trieu. **PIR-PSI: Scaling Private Contact Discovery**. PoPETs 2018.4 (2018), pp. 159–178.

[JL10] Stanislaw Jarecki and Xiaomin Liu. **Fast Secure Computation of Set Intersection**. SCN. Vol. 6280. Lecture Notes in Computer Science. Springer, 2010, pp. 418–435.

[Kal+19] Daniel Kales, Christian Rechberger, Thomas Schneider, Matthias Senker, and Christian Weinert. **Mobile Private Contact Discovery at Scale**. USENIX Security Symposium. USENIX Association, 2019, pp. 1447–1464.

[Kis+17] Ágnes Kiss, Jian Liu, Thomas Schneider, N. Asokan, and Benny Pinkas. **Private Set Intersection for Unequal Set Sizes with Mobile Applications**. PoPETs 2017.4 (2017), pp. 177–197.

[RA18] Amanda C. Davi Resende and Diego F. Aranha. **Faster Unbalanced Private Set Intersection**. Financial Cryptography. Vol. 10957. Lecture Notes in Computer Science. Springer, 2018, pp. 203–221.