Structure-Aware Private Set Intersection from Function Secret Sharing

Speaker - Gayathri Garimella

Brown University

Joint work with Benjamin Goff, Peihan Miao, Mike Rosulek and Jaspal Singh

NIST Workshop on Privacy Enhancing Cryptography
24th Sep 2024

Private Set Intersection (PSI)

Alice

input S_A

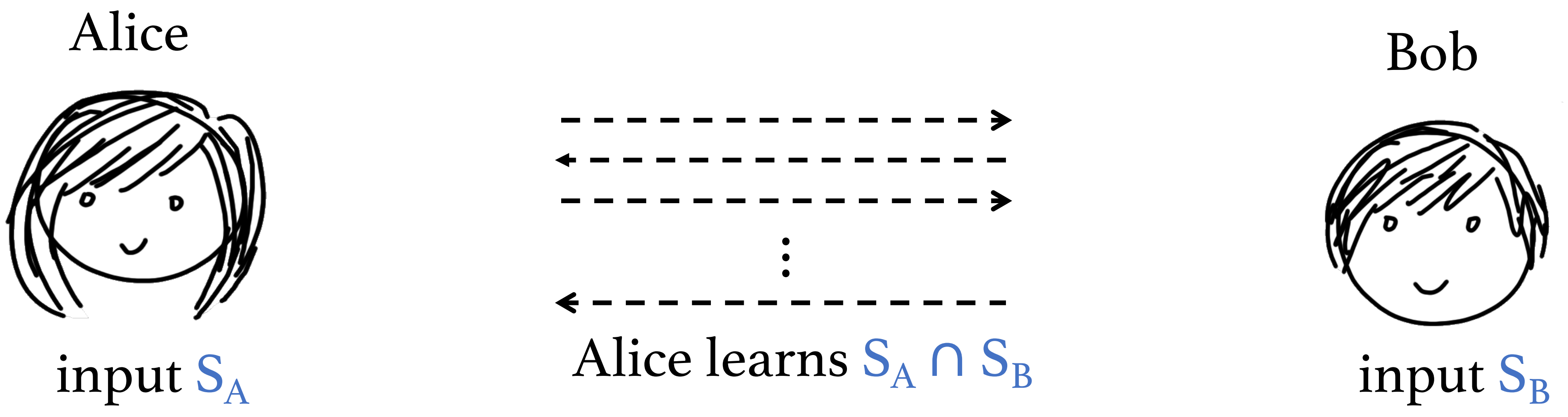
{p, r, i, v, a, t, e}

Bob

input S_B

{s, e, c, u, r, i, t, y}

Private Set Intersection (PSI)



Alice

input S_A

Alice learns $S_A \cap S_B$

Bob

input S_B

$\{p, \underline{r}, \underline{i}, v, a, \underline{t}, \underline{e}\}$

identifies only common elements

$\{?, e, ?, ?, r, i, t, ?\}$

learns nothing about Alice's input

PSI Research

Approaches

- Diffie-Hellman [Mea86, HFH99, JL10, DKT10, IKN+20, RT21…]
- Oblivious Polynomial Evaluation [FNP04, KS05, dMRY11…]
- RSA [DT10, ADT11]
- Bloom Filters [DCW13, RR17a]
- FHE [CLR17, CHLR18, CMDG+21]
- Circuit-based [HEK12, PSSZ15, PSWW18, PSTY19, GarimellaMR+21]
- OT [PSZ14, PSSZ15, KKRT16, RR17, PRTY19, CM20, PRTY20, RS21, GarimellaPR+21]
- Vector OLE [RS21, GarimellaPR+21, CRR21, RR22, BPSY23…]

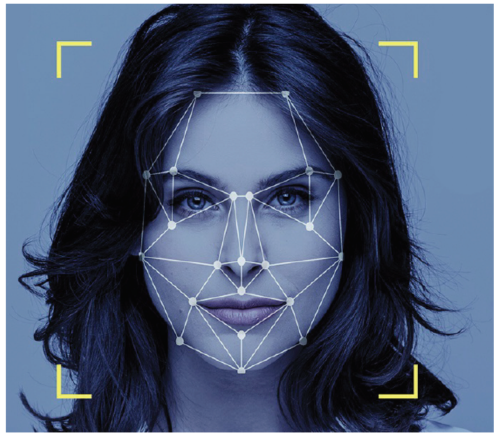
Settings

- Semi-honest/Malicious [RR17, OOS17, CHLR18, PRTY20, RS21, GarimellaPR$^+$21, BPSY23…]
- Plain/Cardinality/Associated-sum: [PSTY19, KK20, MPR$^+$20, IKN$^+$20, GarimellaMR$^+$21, RS21, CGS22...]
- PS Union: [DC17, KRTY19, GarimellaPR+21, JSZ+22, LG23, BPSY23, GNT24…]
- Balanced/Unbalanced/Laconic: [ABD$^+$21, ALOS22, DKL$^+$23, GHMM24..]
- Two-party/Multi-party: [HV17, NTY21, BMRR21, CDG$^+$21, GarimellaPR$^+$21, ENOP22, BHV$^+$23, GTY24..]
- Updatable: [KLS+17, ATD20, BMX22..]
- Fuzzy PSI: [CFR$^+$21,UCK$^+$21 ..]

Fuzzy matching

PSI enables exact matches between elements, but what if..

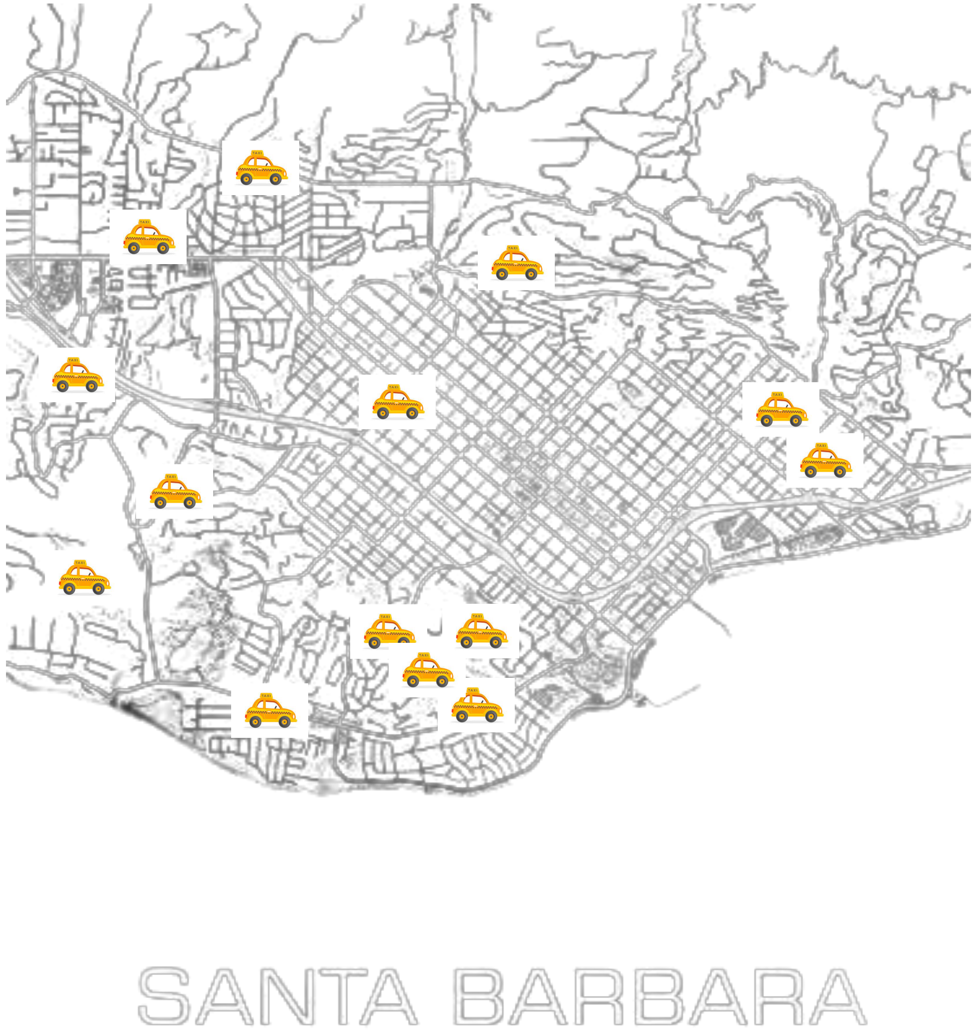
- Password breach with typos
- Biometrics fingerprint, facial recognition matching…
- Fuzzy Personal Identifiable Information (PII) matching – (name, physical address, contact, ..)
- Privacy-preserving ride-sharing application using fuzzy GPS matching (next slide)



Fuzzy matching

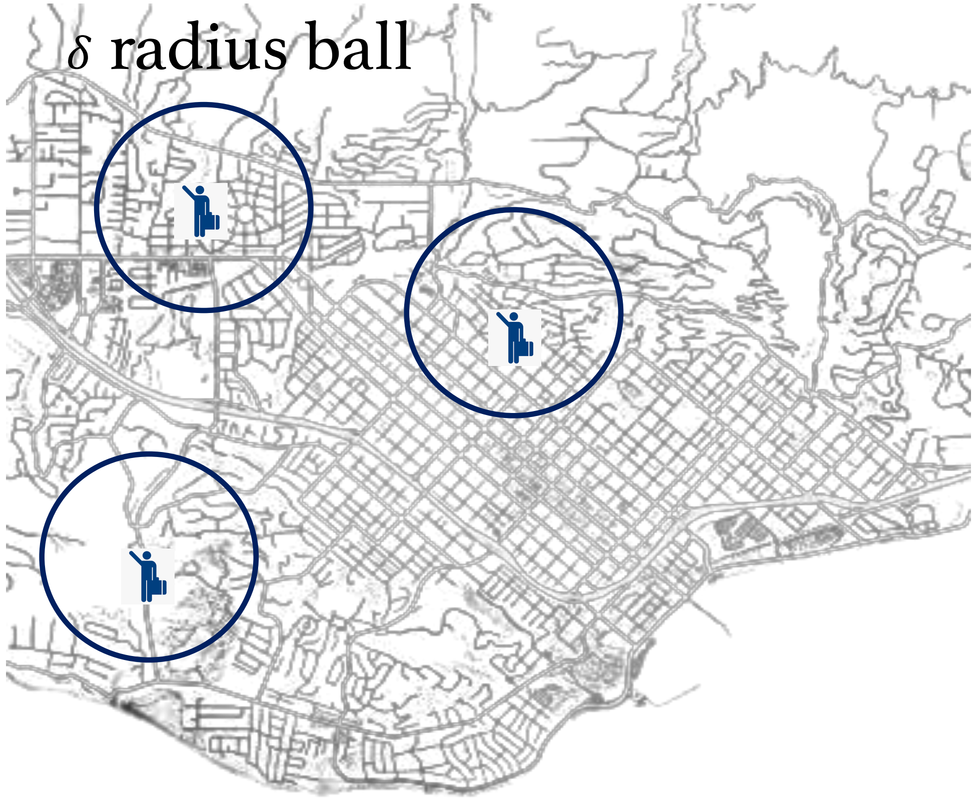
privacy-preserving ride hailing service

SANTA BARBARA

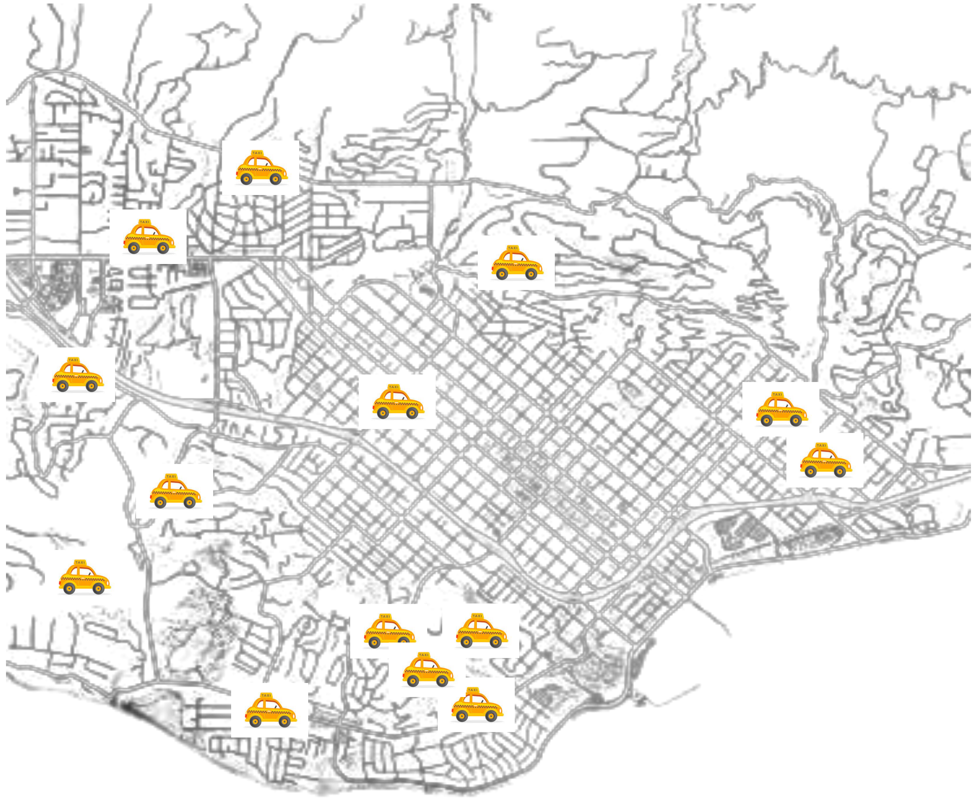


Fuzzy matching

privacy-preserving ride hailing service



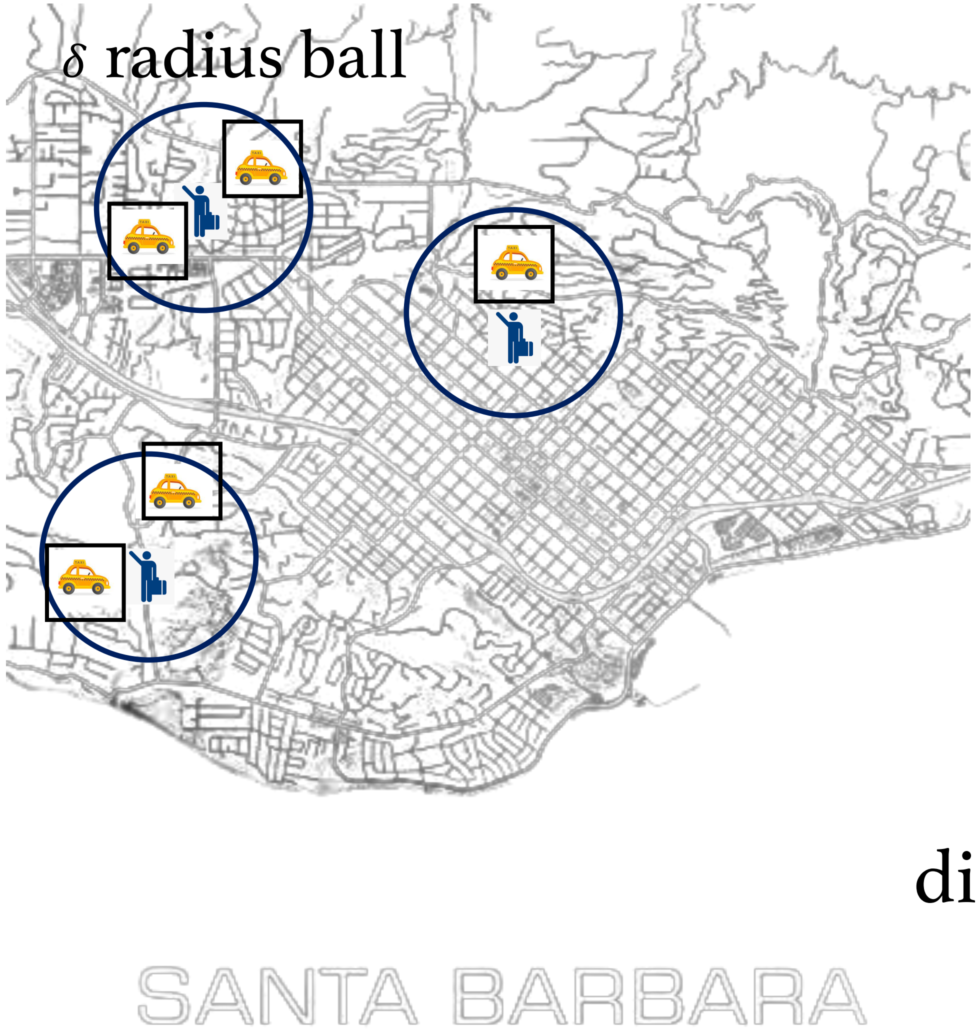
δ radius ball

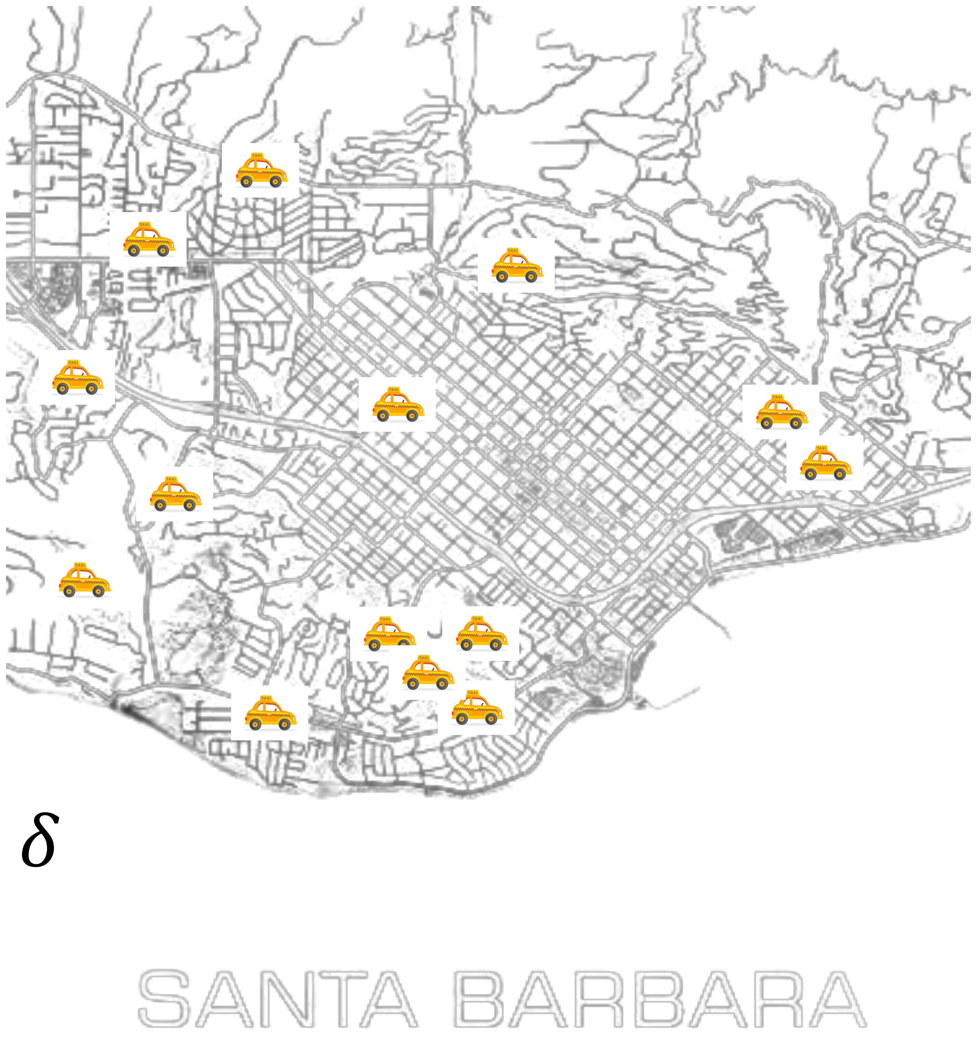


dist(🙋, 🚕) $\leq \delta$

Fuzzy matching

privacy-preserving ride hailing service





dist(🙋, 🚕) $\leq \delta$

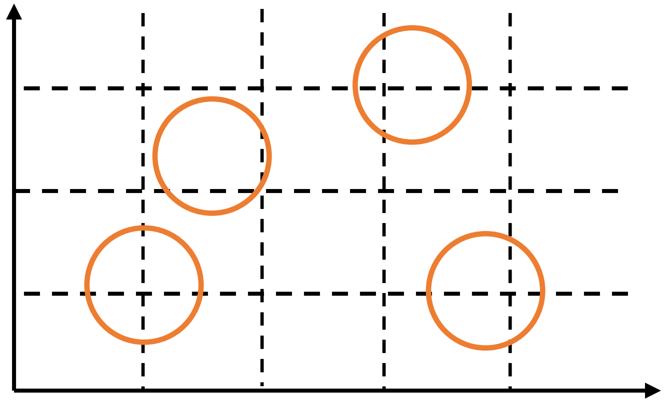
Naïve solution

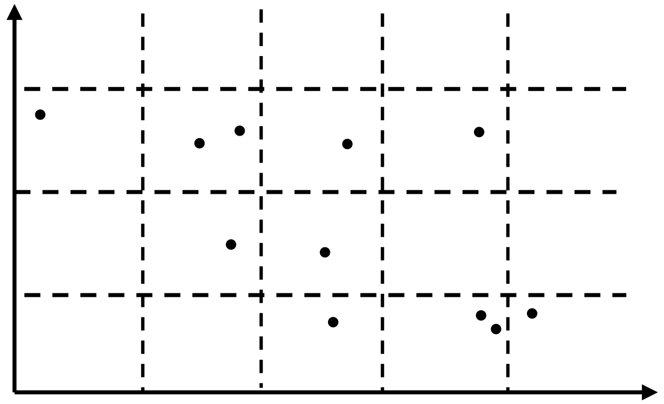
Alice's enumerates her structured input S_A

reduces to standard PSI

input S_A = {all the points inside structure}

input S_B





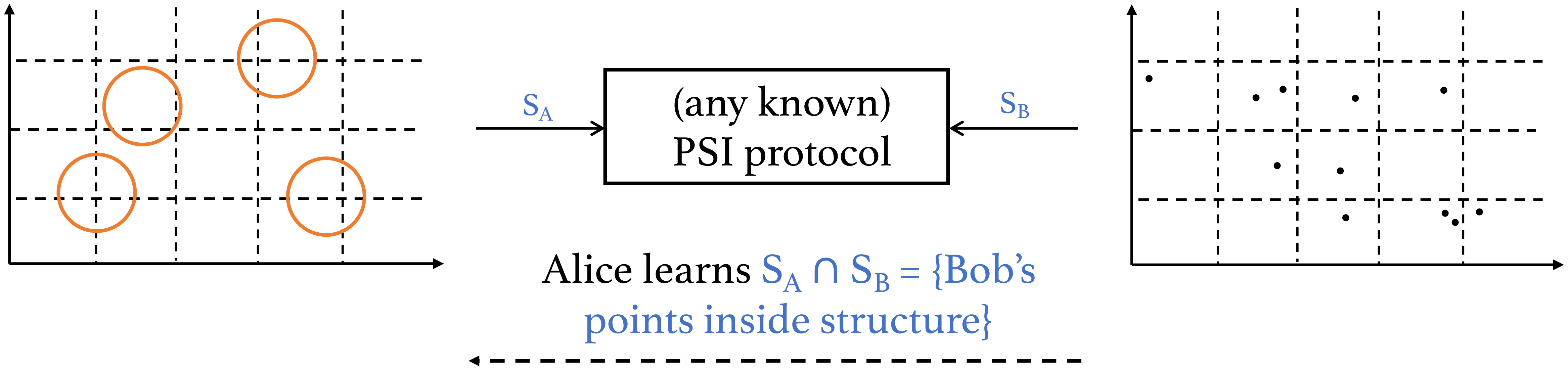
Naïve solution

Alice's enumerates her structured input S_A

reduces to standard PSI

input S_A = {all the points inside structure}

input S_B



S_A

(any known)
PSI protocol

S_B

Alice learns $S_A \cap S_B$ = {Bob's points inside structure}

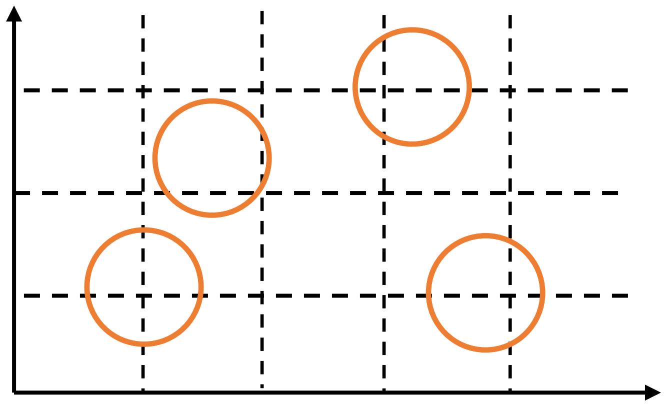
Naïve solution

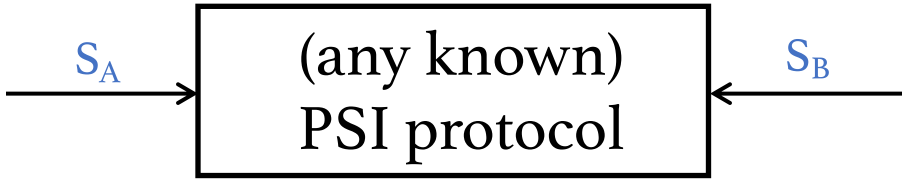
Comm and / or Comp cost $O((|S_A| + |S_B|).\kappa)$

~ total volume $|S_A|$ of balls in Alice's input

input S_A = {all the points inside structure}

input S_B

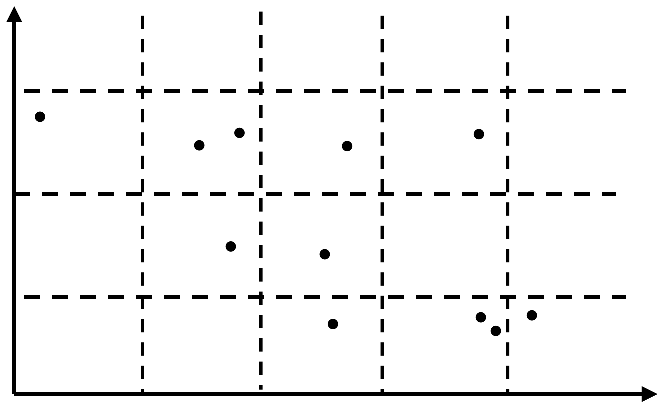




S_A

(any known) PSI protocol

S_B



Alice learns $S_A \cap S_B$ = {Bob's points inside structure}

Naïve solution

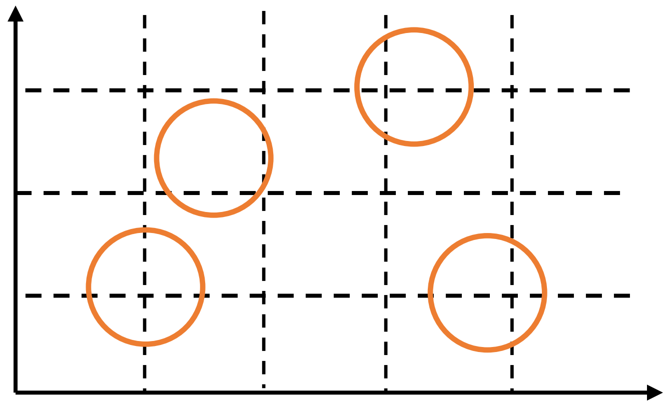
Can the protocol cost scale with description size (# of balls) in Alice's input?

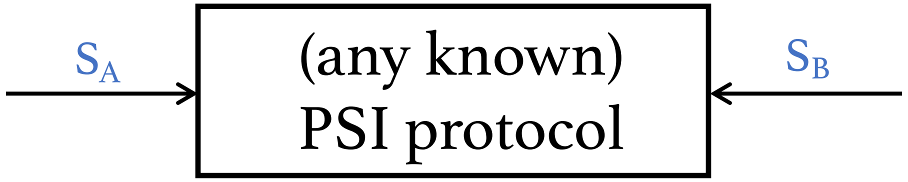
Comm and / or Comp cost $O((|S_A| + |S_B|). \kappa)$

~ total volume $|S_A|$ of balls in Alice's input

input S_A = {all the points inside structure}

input S_B

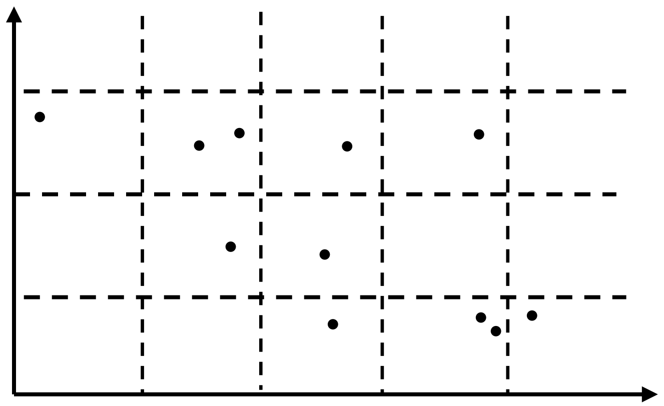




S_A

(any known)
PSI protocol

S_B



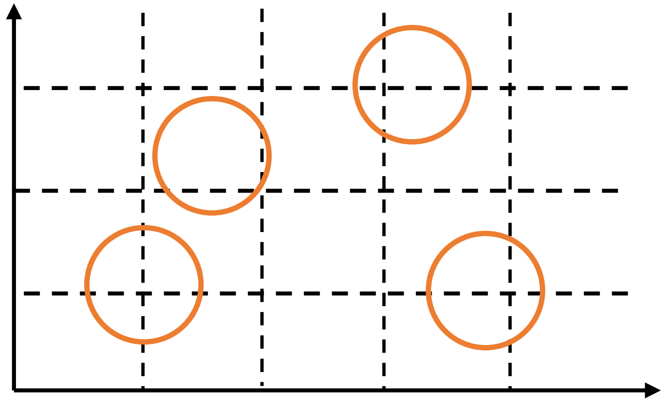
Alice learns $S_A \cap S_B$ = {Bob's points inside structure}

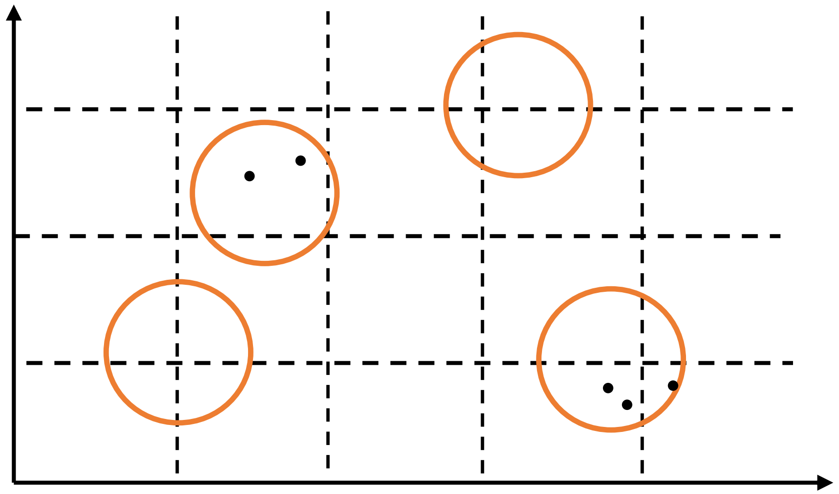
Structure-Aware Private Set Intersection (sa-PSI)

[GarimellaRosulekSingh22] a variant of PSI where Alice's input has a publicly known structure

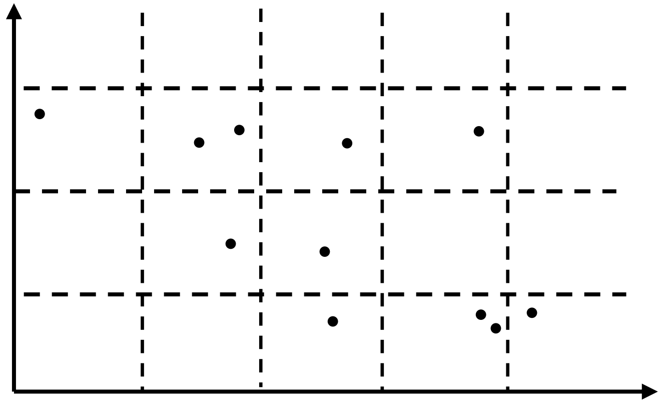
Examples - interval, ball or union of balls in some well-defined metric space, …

input S_A





input S_B



Alice learns $S_A \cap S_B$ = {all points inside balls}

Structure-Aware PSI, with applications to Fuzzy Matching. CRYPTO 2022. *Gayathri Garimella*, Mike Rosulek and Jaspal Singh

Research question:

Can we design “more efficient” PSI protocols, when one party has a structured input from a publicly known set family?

Research question:

Can we design "more efficient" PSI protocols, when one party has a structured input from a publicly known set family?

Can the communication and / or computation cost scales with a succinct description of the structured input, instead of the set cardinality?

might be prohibitively large for many realistic applications

Summary of results

Structure–Aware Private Set Intersection

Structure-Aware PSI, with applications to Fuzzy Matching. CRYPTO 2022.
Gayathri Garimella, Mike Rosulek and Jaspal Singh

- ✓ Semi-honest adversaries
- ✓ Comm ~ description size
- ✓ Tool: Boolean Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

Summary of results

Structure–Aware Private Set Intersection

Structure-Aware PSI, with applications to Fuzzy Matching.
CRYPTO 2022.
Gayathri Garimella, Mike Rosulek and Jaspal Singh

- ✓ Semi-honest adversaries
- ✓ Comm ~ description size
- ✓ Tool: Boolean Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

Malicious-secure, Structure-Aware PSI.
CRYPTO 2023.
Gayathri Garimella, Mike Rosulek and Jaspal Singh

- ✓ Malicious adversaries
- ✓ Comm ~ description size
- ✓ Tool: derandomizable Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

Summary of results

Structure–Aware Private Set Intersection

Structure-Aware PSI, with applications to Fuzzy Matching.
CRYPTO 2022.
Gayathri Garimella, Mike Rosulek and Jaspal Singh

- ✓ Semi-honest adversaries
- ✓ Comm ~ description size
- ✓ Tool: Boolean Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

Malicious-secure, Structure-Aware PSI.
CRYPTO 2023.
Gayathri Garimella, Mike Rosulek and Jaspal Singh

- ✓ Malicious adversaries
- ✓ Comm ~ description size
- ✓ Tool: derandomizable Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

Computation Efficient Structure Aware PSI.
CRYPTO 2024.
Gayathri Garimella, Benjamin Goff and Peihan Miao

- ✓ Semi-honest adversaries
- ✓ Comm and Comp ~ description size
- ✓ Tool: incremental Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

Summary of results

Structure–Aware Private Set Intersection

✓ Structure-Aware PSI, with applications to Fuzzy Matching. CRYPTO 2022. Gayathri Garimella, Mike Rosulek and Jaspal Singh

- ✓ Semi-honest adversaries
- ✓ Comm ~ description size
- ✓ Tool: Boolean Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

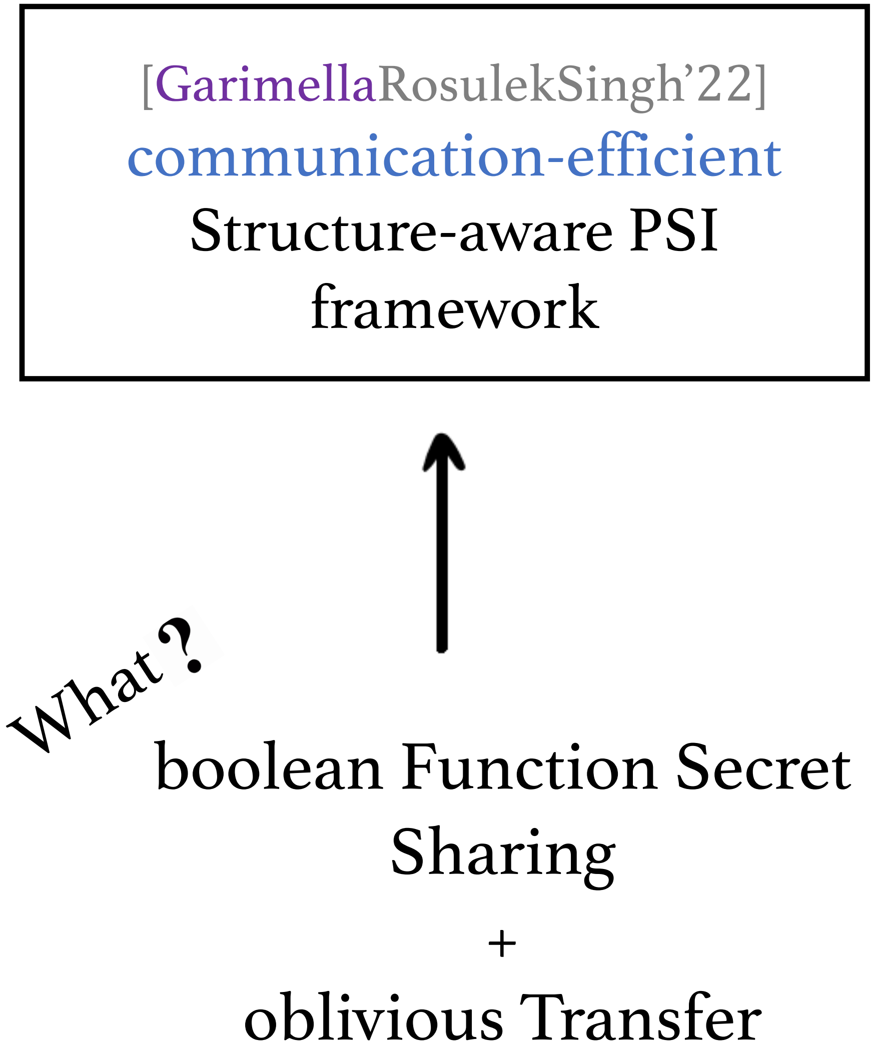
Malicious-secure, Structure-Aware PSI. CRYPTO 2023. Gayathri Garimella, Mike Rosulek and Jaspal Singh

- ✓ Malicious adversaries
- ✓ Comm ~ description size
- ✓ Tool: derandomizable Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

Computation Efficient Structure Aware PSI. CRYPTO 2024. Gayathri Garimella, Benjamin Goff and Peihan Miao

- ✓ Semi-honest adversaries
- ✓ Comm and Comp ~ description size
- ✓ Tool: incremental Function Secret sharing
- ✓ PSI construction for {union of L_∞ balls} set family

Result 1:



[GarimellaRosulekSingh'22]
communication-efficient
Structure-aware PSI
framework

What?

boolean Function Secret Sharing
+
oblivious Transfer

Building block 1:

Boolean Function Secret Sharing

given input $S_A \in S$ from a class of structured sets

boolean Function Secret Sharing (bFSS)

[BoyleGilboaIshai15] – style FSS for set membership in S_A function

Building block 1:

Boolean Function Secret Sharing

given input $S_A \in \mathcal{S}$ from a class of structured sets

<u>boolean Function Secret Sharing</u> (bFSS)

[BoyleGilboaIshai15] – style FSS for set membership in S_A function

share(S_A) ⟶ ░, ▤ where ░, ▤ $\approx \$\$$

Building block 1:

Boolean Function Secret Sharing

given input $S_A \in \mathcal{S}$ from a class of structured sets

boolean Function Secret Sharing (bFSS)

[BoyleGilboaIshai15] – style FSS for set membership in S_A function

share(S_A) $\longrightarrow$ ░ , ▤ where ░ , ▤ $\approx \$\$$

$\forall x \in S_A \Longrightarrow$ **ev**(░ , x) $\oplus$ **ev**(▤ , x) $=0$

$\forall x \notin S_A \Longrightarrow$ **ev**(░ , x) $\oplus$ **ev**(▤ , x) $=1$

Building block 1:

Boolean Function Secret Sharing

given input $S_A \in \mathcal{S}$ from a class of structured sets

boolean Function Secret Sharing (bFSS)

[BoyleGilboaIshai15] – style FSS for set membership in S_A function

share(S_A) ⟶ ░ , ▤ where ░ , ▤ ≈ $\$\$$

$\forall x \in S_A \Longrightarrow$ **ev**(░ , x) $\oplus$ **ev**(▤ , x) $= 0$

$\forall x \notin S_A \Longrightarrow$ **ev**(░ , x) $\oplus$ **ev**(▤ , x) $= 1$

succinctness: | ▤ | , | ░ | $= \sigma <<$ | S_A |

Building block 1:

Boolean Function Secret Sharing

given input $S_A \in \mathcal{S}$ from a class of structured sets

<u>boolean Function Secret Sharing</u> (bFSS)

[BoyleGilboaIshai15] – style FSS for set membership in S_A function

share(S_A) ⟶ ⬚, ▤ where ⬚, ▤ ≈ $$

$\forall x \in S_A \Longrightarrow$ **ev**(⬚ , x) $\oplus$ **ev**(▤ , x) = 0

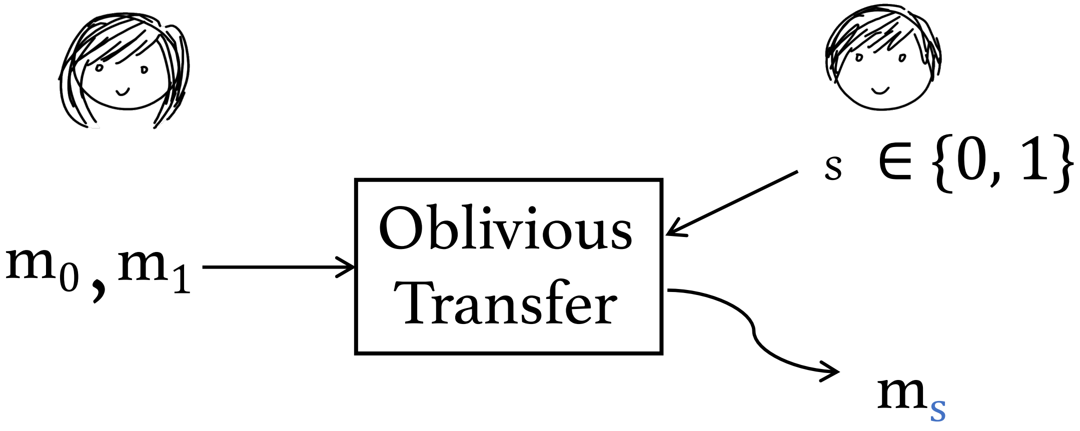
$\forall x \notin S_A \Longrightarrow$ **ev**(⬚ , x) $\oplus$ **ev**(▤ , x) = 1

succinctness: |▤|, |⬚| $= \sigma <<$ | S_A |

[BGI15, BGI16, BCG+21, BGIK22] - PRG based constructions for set families like {singleton, 1-d interval, d-dimensional interval..}

Building block 2:

Oblivious Transfer [Rabin'81]

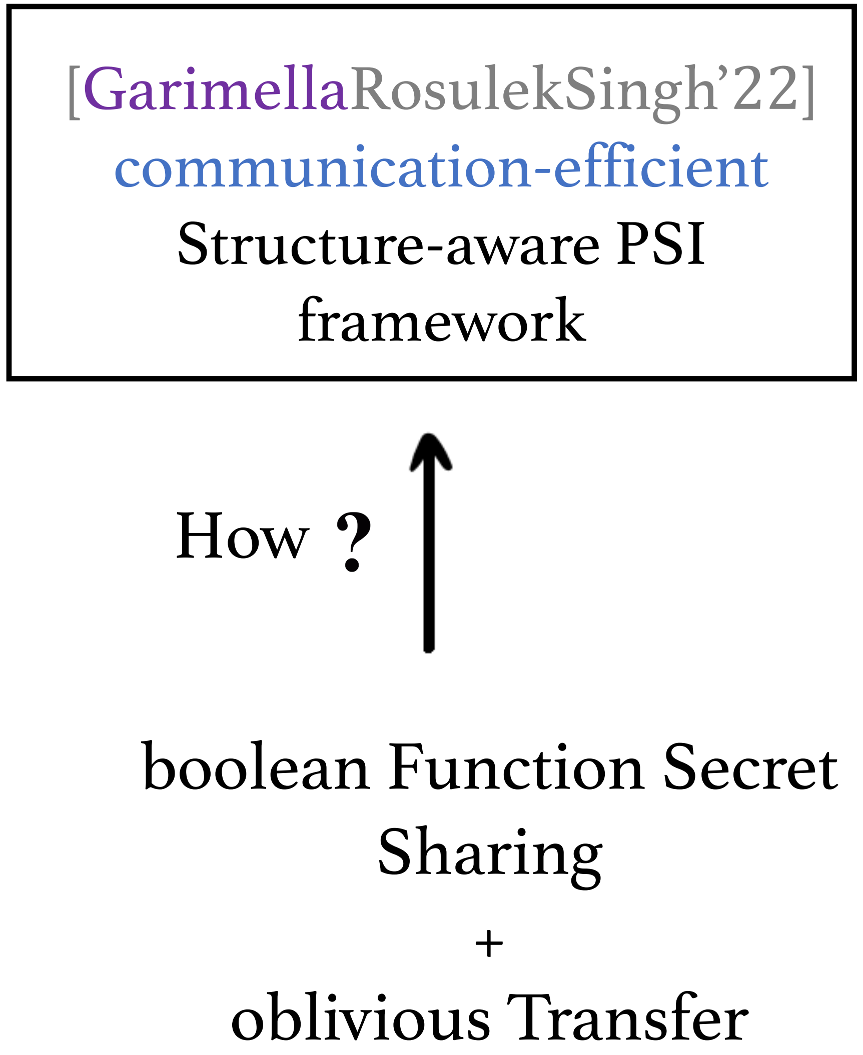


many OTs can be instantiated efficiently (largely using symmetric key operations) from OT extension [IKNP03]

How to exchange secrets with oblivious transfer. Cryptology ePrint Archive, 2005. Michael O Rabin.

Extending Oblivious Transfers Efficiently. CRYPTO 2003. Yuval Ishai, Joe Kilian, Kobbi Nissim and Erez Petrank.

Now, let's see how to realize sa-PSI



[GarimellaRosulekSingh'22]
communication-efficient
Structure-aware PSI
framework

How ?

boolean Function Secret
Sharing
+
oblivious Transfer

How does the sa-PSI protocol work?

assumptions: OT-hybrid, hamming correlation robust hash

input S_A

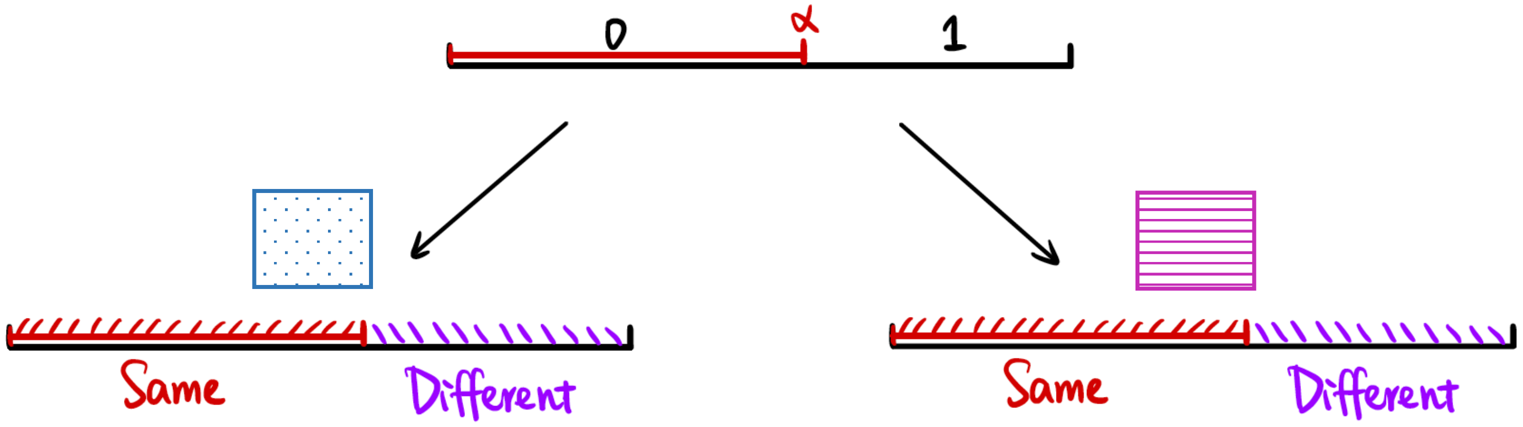
input S_B

1. Alice generates bFSS shares of her input S_A

Example: Alice's input is One-sided Interval

input S_A

1. Alice generates bFSS shares of her input S_A = {one-sided interval}



Recall definition:

given input A ∈ $\mathcal{S}$ from a class of structured sets

<u>boolean Function Secret Sharing</u> (bFSS)

[BoyleGilboaIshai15] – style FSS for set membership in A function

share(A) $\longrightarrow$ ▦, ▤ where ▦, ▤ $\approx \$\$$

$\forall x \in A \Longrightarrow$ ev(▦ , x) $\oplus$ ev(▤ , x) = 0

$\forall x \notin A \Longrightarrow$ ev(▦ , x) $\oplus$ ev(▤ , x) = 1

succinctness: |▤|, |▦| $= \sigma <<$ | S_A |

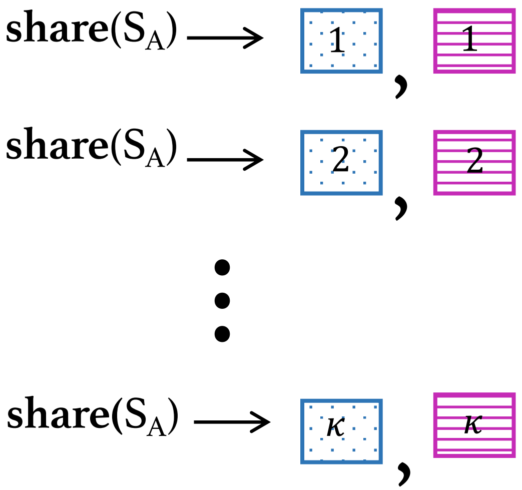
How does the sa-PSI protocol work?

input S_A

input S_B

1. generates κ instances of bFSS shares

2. picks κ choice bits to learn ░ or ▤



share(S_A) → ░1, ▤1

share(S_A) → ░2, ▤2

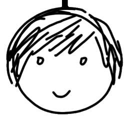
⋮

share(S_A) → ░κ, ▤κ

How does the sa-PSI protocol work?

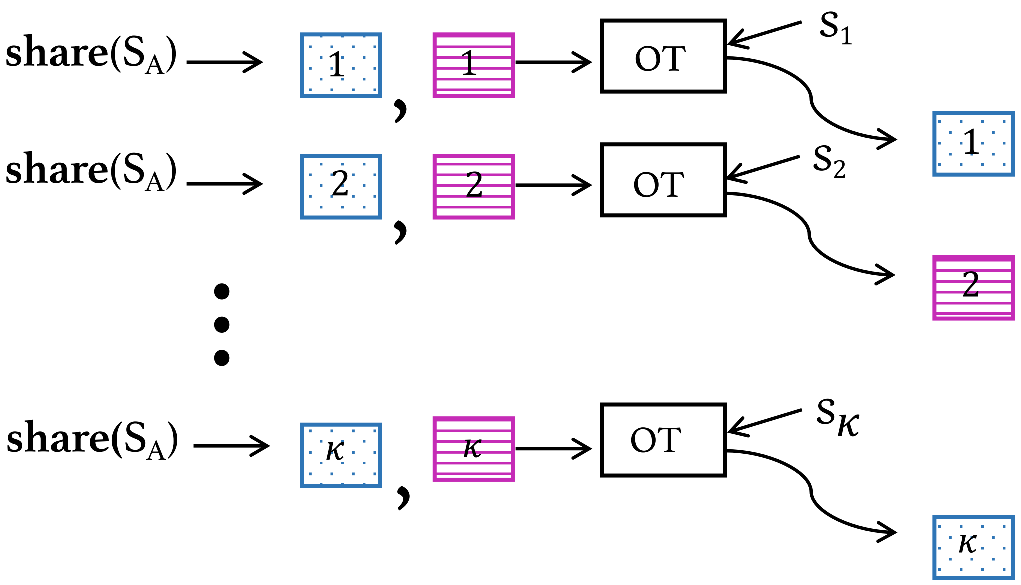
input S_A

input S_B



1. generates κ instances of bFSS shares

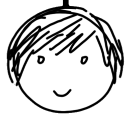
2. picks κ choice bits to learn ⬚ or ▤



How does the sa-PSI protocol work?

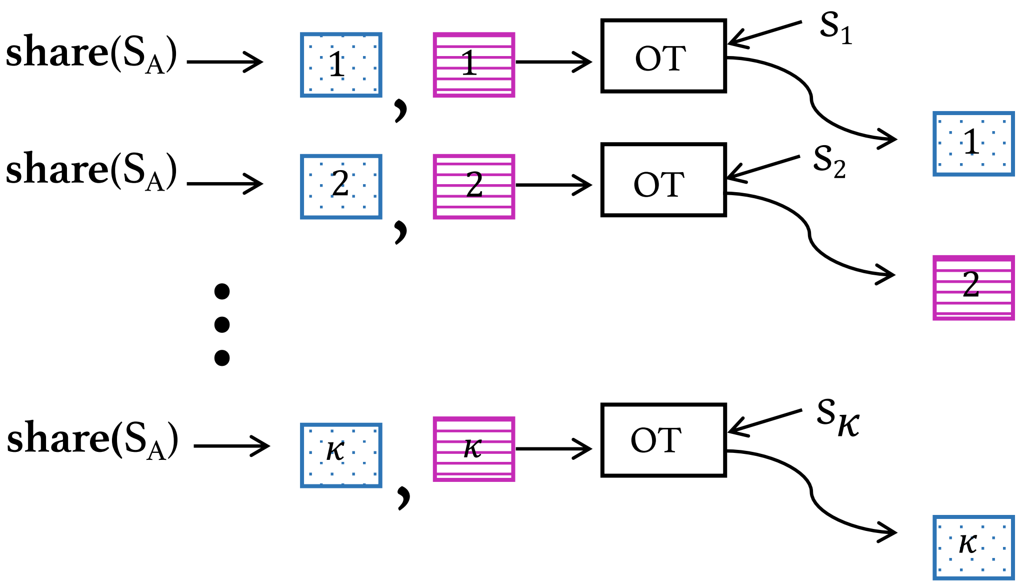
input S_A

input S_B



1. generates κ instances of bFSS shares

2. picks κ choice bits to learn ⬚ or ▤

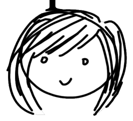


Bob computes F(x) on all his inputs

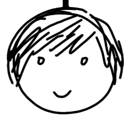
$$\boxed{\mathrm{F}(x) = \mathbf{H}(\mathbf{ev}(\boxed{1}, x) \| \mathbf{ev}(\boxed{2}, x) \| \cdots \mathbf{ev}(\boxed{\kappa}, x))}$$

How does the sa-PSI protocol work?

input S_A

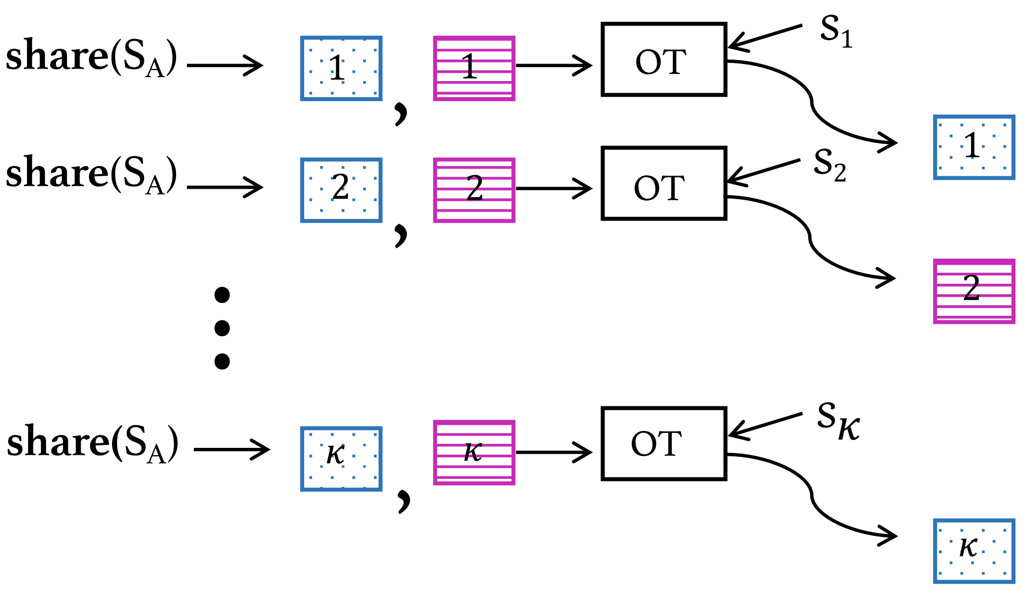


input S_B



1. generates κ instances of bFSS shares

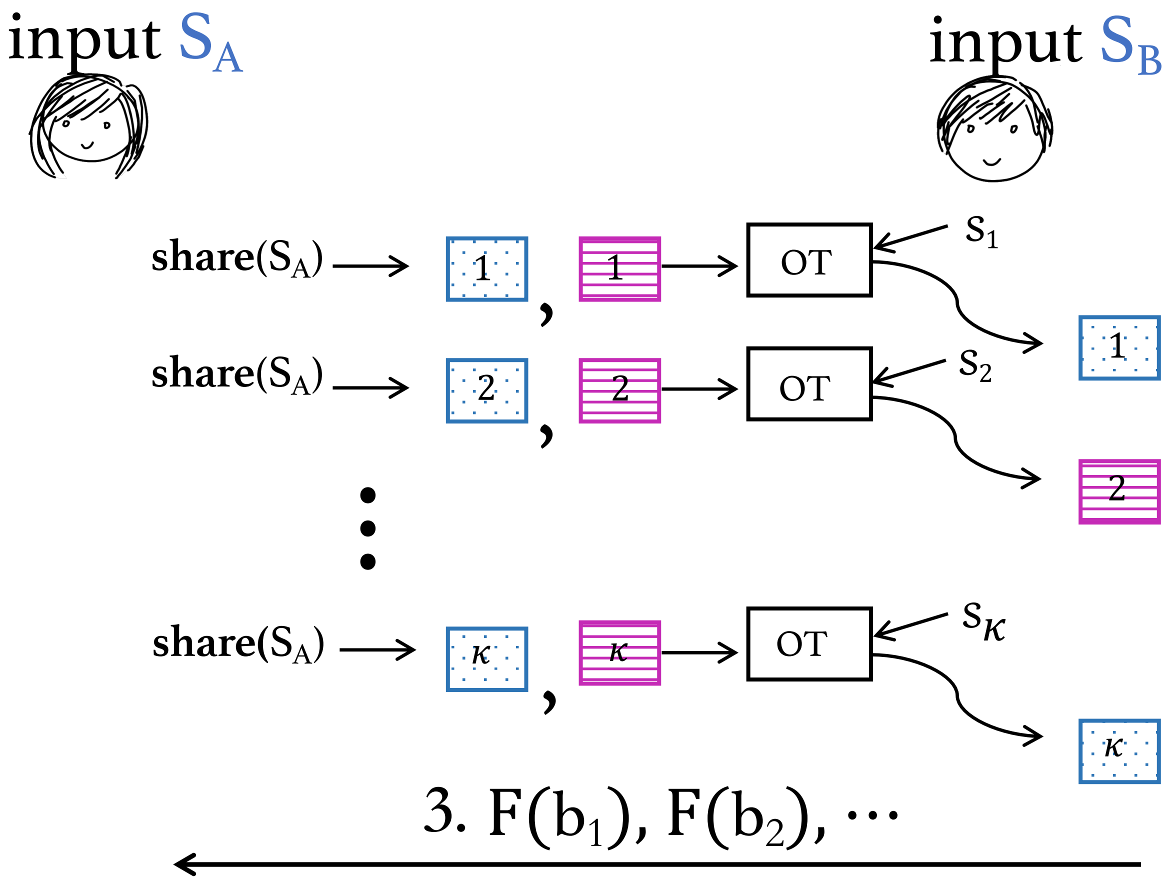
2. picks κ choice bits to learn ⬚ or ▤



3. $F(b_1), F(b_2), \cdots$

$$F(x) = \mathbf{H}(\mathbf{ev}(\boxed{1}, x) \| \mathbf{ev}(\boxed{2}, x) \| \cdots \mathbf{ev}(\boxed{\kappa}, x))$$

How does the sa-PSI protocol work?



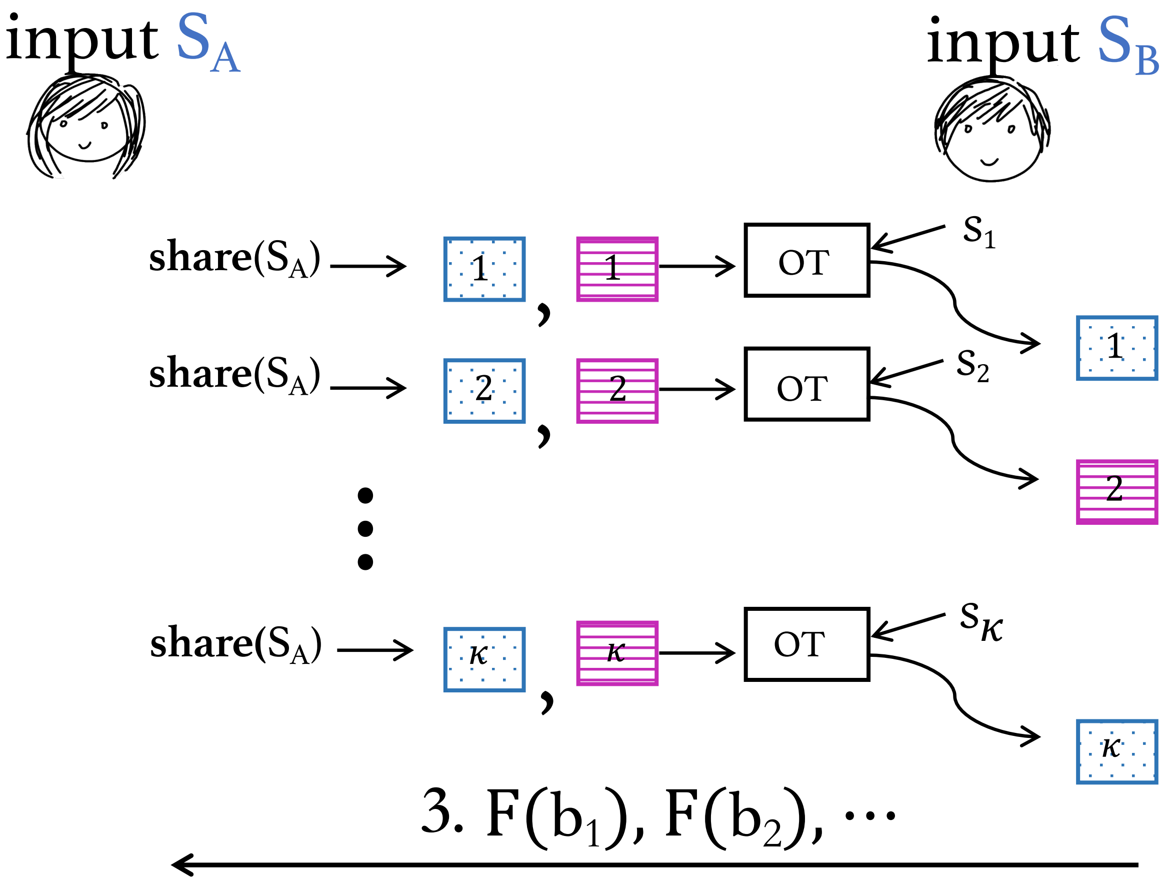
if $x \in S_A \Longrightarrow$ **ev**(▤ , x) = **ev**(⬚ , x)

if $x \in S_A \quad \Longrightarrow$ Alice can compute $F(x)$

if $x \notin S_A \quad \Longrightarrow F(x) \approx$ \$\$ looks random

$$F(x) = \mathbf{H}(\mathbf{ev}(\boxed{1}, x) \| \mathbf{ev}(\boxed{2}, x) \| \cdots \mathbf{ev}(\boxed{\kappa}, x))$$

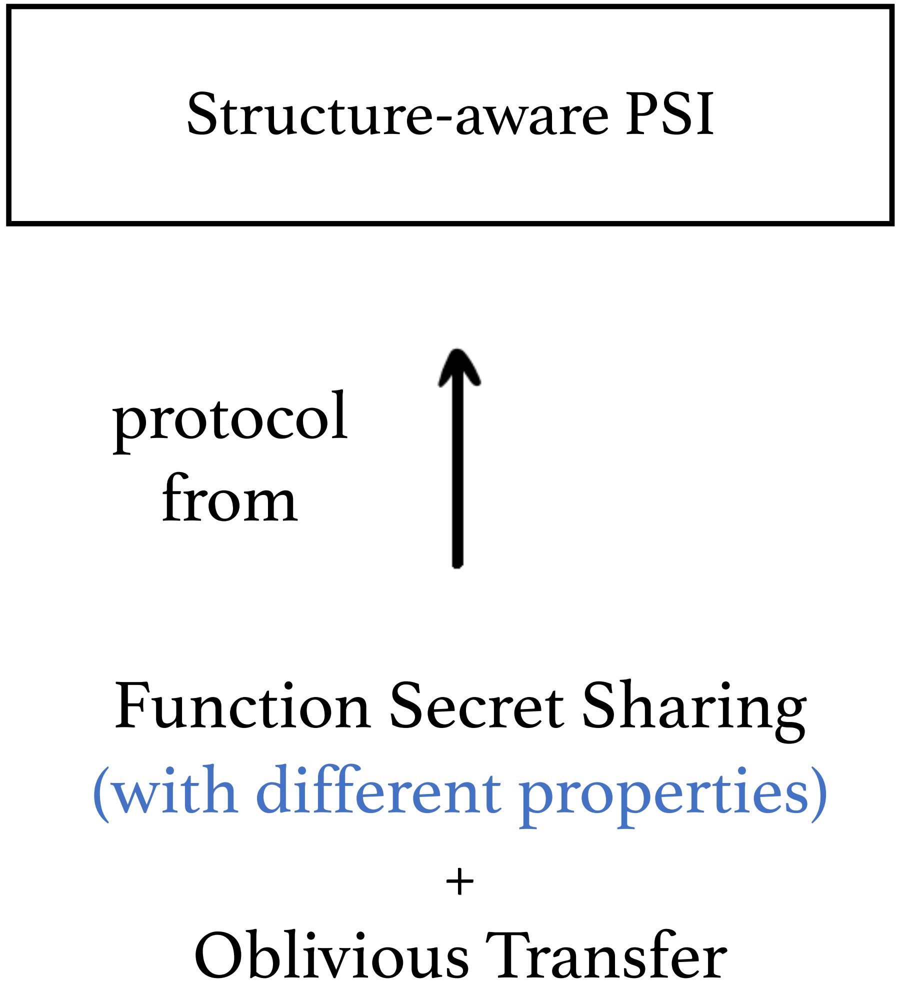
Full protocol



3. $\forall a \in S_A$, compute $F(a)$
4. locally compare to learn intersection

$$F(x) = \mathbf{H}(\mathbf{ev}(\boxed{1}, x) \| \mathbf{ev}(\boxed{2}, x) \| \cdots \mathbf{ev}(\boxed{\kappa}, x))$$

Summary



Structure-aware PSI

protocol
from

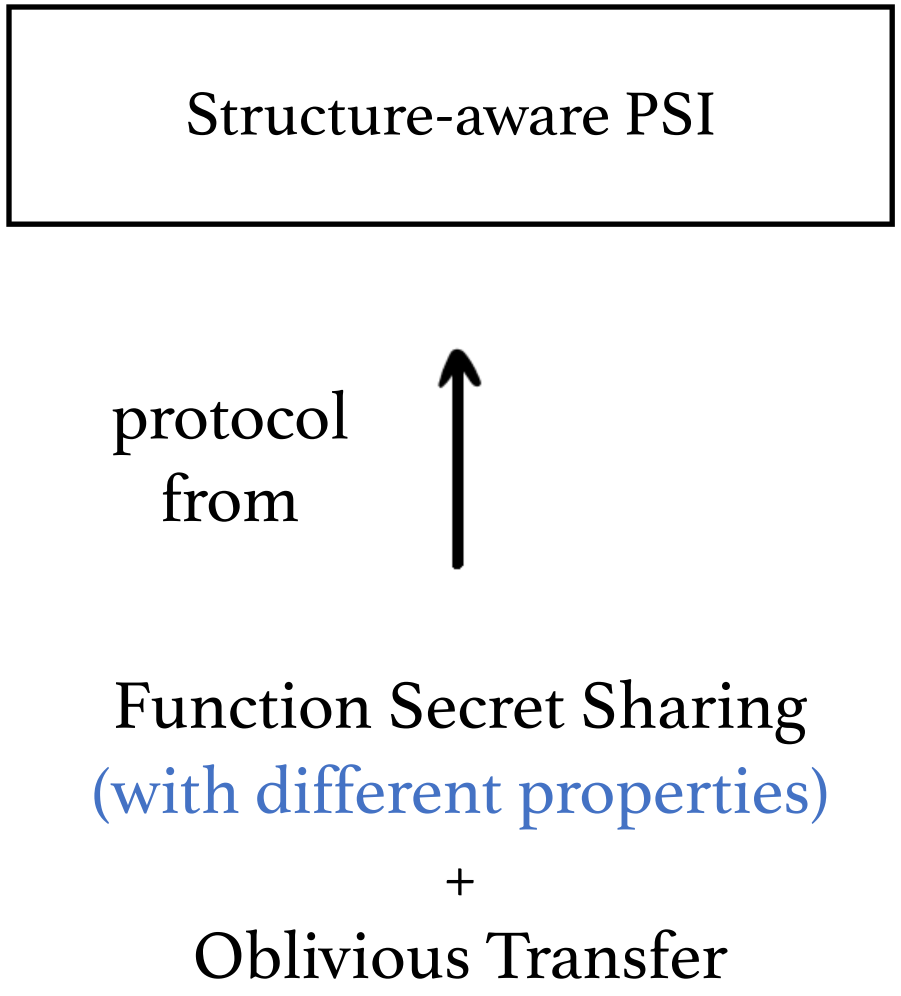
Function Secret Sharing
(with different properties)
+
Oblivious Transfer

- Formalized Structure Aware PSI
- Construct a general PSI framework from (variants of) Function Secret Sharing
  - ✓ semi-honest
  - ✓ malicious adversaries
  - ✓ comm + comp scale with description of structured set
- Formalize the properties of required FSS
- Present FSS constructions for {union of L_∞ balls metric space} set family

Future Directions

- Can we extend our techniques to other distance metrics like L2 norm, Hamming distance metrics?
- Can we construct FSS for other structures (motivated by other applications beyond fuzzy matching)?
- Can we improve the malicious framework to get comp and comm ~ description of set family?

Takeaway



Structure-aware PSI

protocol from

Function Secret Sharing

(with different properties)

+

Oblivious Transfer