A Brief Overview of
Private Set Intersection

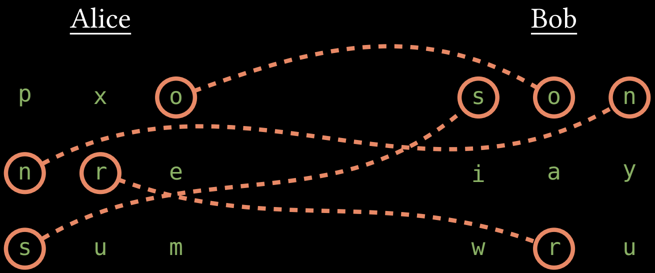
Mike Rosulek, Oregon State University

NIST STPPA, April 19, 2021

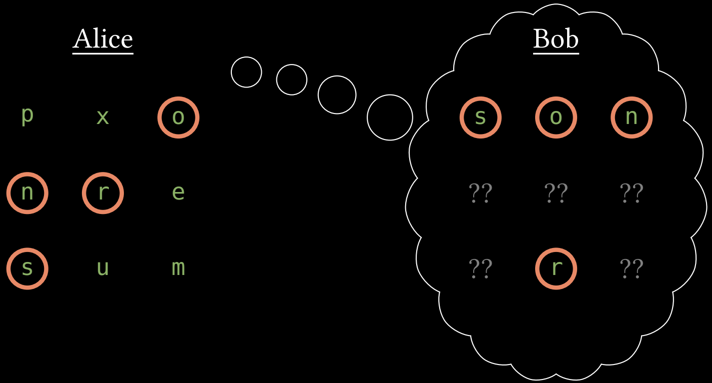
what is private set intersection (PSI)?

Alice				Bob		
p	x	o		s	o	n
n	r	e		i	a	y
s	u	m		w	r	u

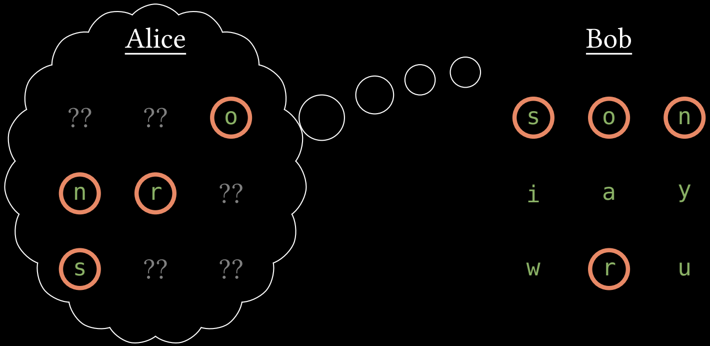
what is private set intersection (PSI)?



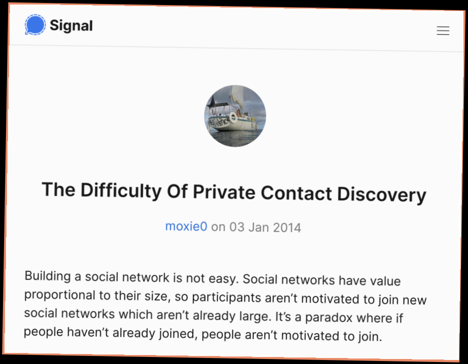
what is private set intersection (PSI)?



what is private set intersection (PSI)?



why use PSI?



Signal

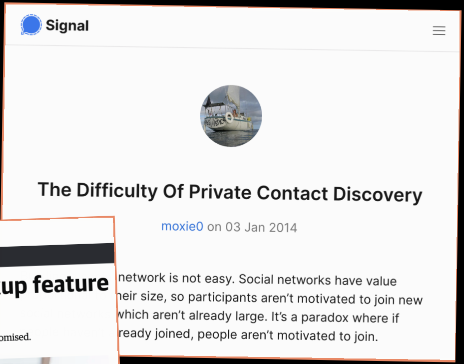
The Difficulty Of Private Contact Discovery

moxie0 on 03 Jan 2014

Building a social network is not easy. Social networks have value proportional to their size, so participants aren't motivated to join new social networks which aren't already large. It's a paradox where if people haven't already joined, people aren't motivated to join.

{my phone contacts} ∩ {users of your service}

why use PSI?

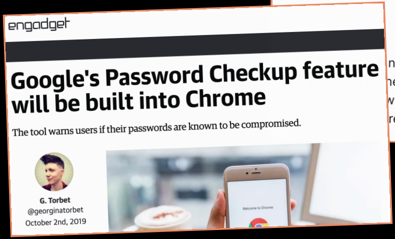


Signal

The Difficulty Of Private Contact Discovery

moxie0 on 03 Jan 2014

network is not easy. Social networks have value
heir size, so participants aren't motivated to join new
which aren't already large. It's a paradox where if
ready joined, people aren't motivated to join.



engadget

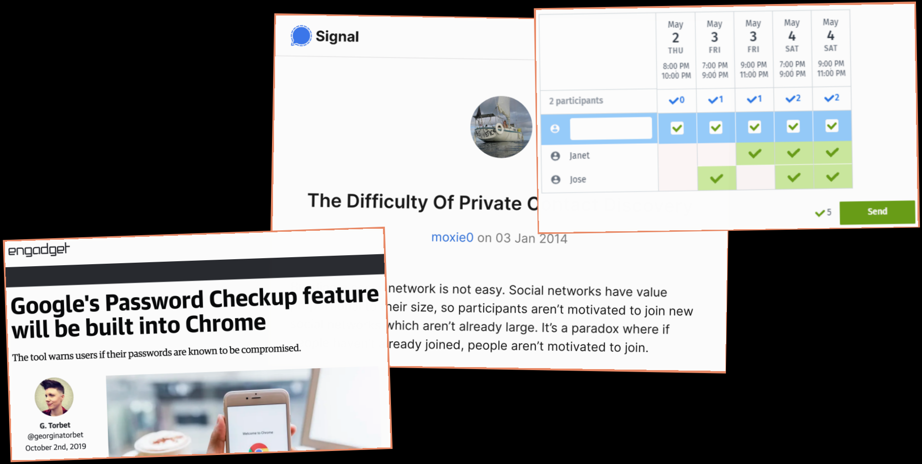
Google's Password Checkup feature will be built into Chrome

The tool warns users if their passwords are known to be compromised.

G. Torbet
@georginatorbet
October 2nd, 2019

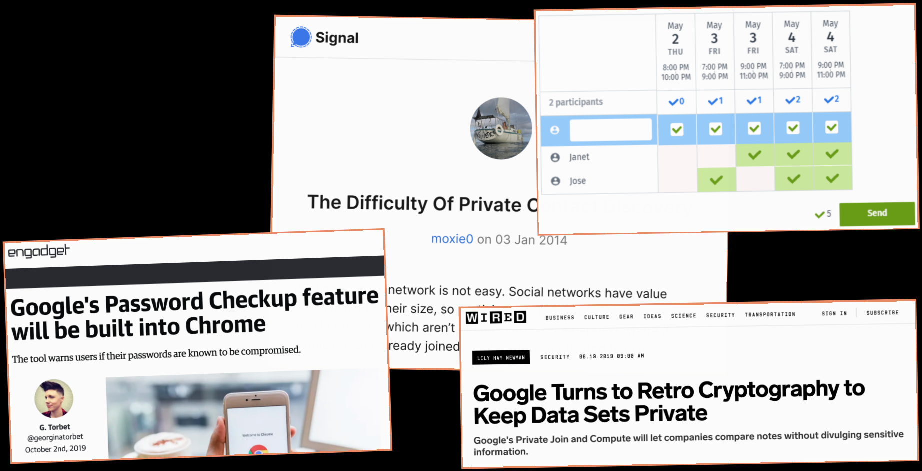
{my passwords} ∩ {passwords found in breaches}

why use PSI?



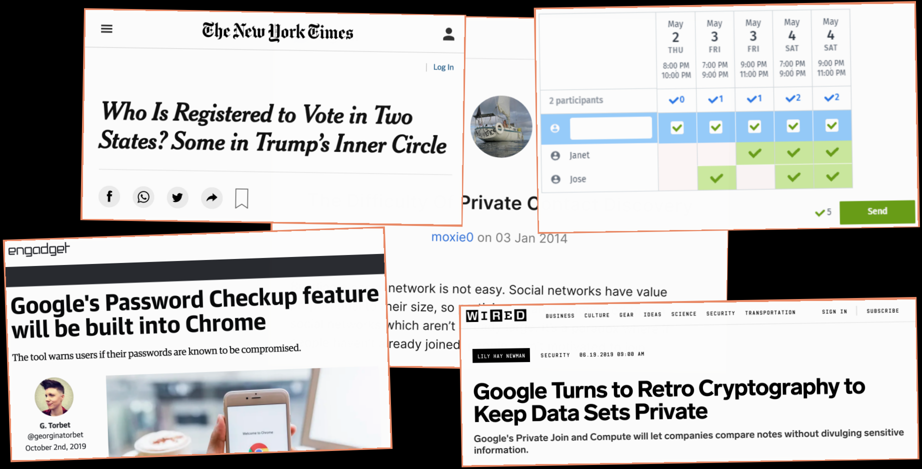
{my availability} ∩ {your availability}

why use PSI?



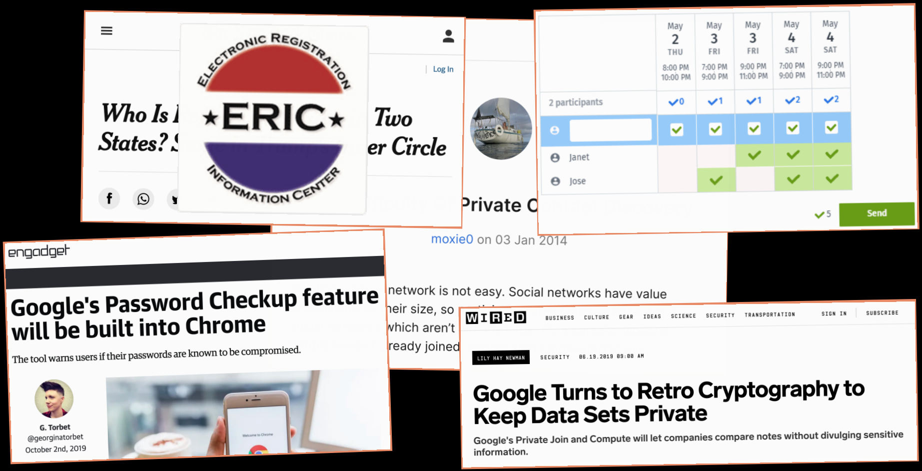
{people who saw ad} ∩ {customers who made purchases}

why use PSI?

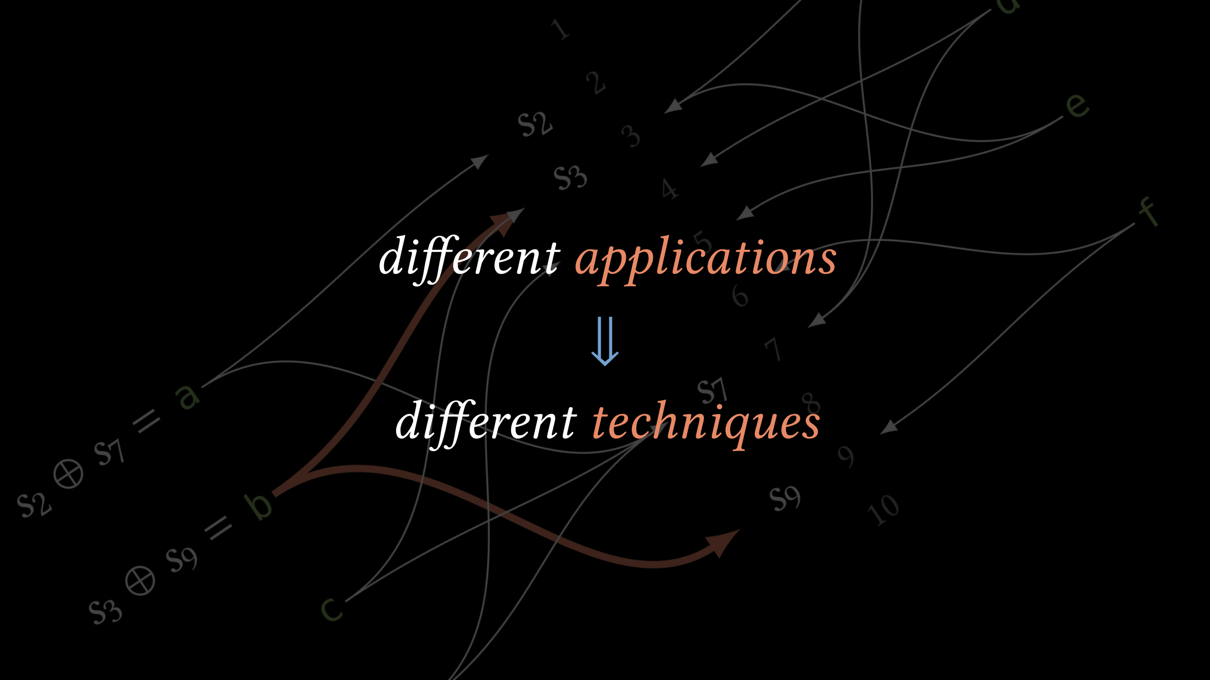


$\{\text{voters registered in OR}\} \cap \{\text{voters registered in NY}\}$

why use PSI?



{voters registered in OR} ∩ {voters registered in NY}



different applications

$\Downarrow$

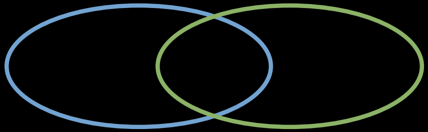
different techniques

PSI on **small sets** (hundreds)

- private availability poll
- key agreement techniques

PSI on **small sets** (hundreds)

- private availability poll
- key agreement techniques

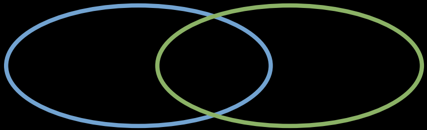


PSI on **large sets** (millions)

- double-registered voters
- OT extension; combinatorial tricks

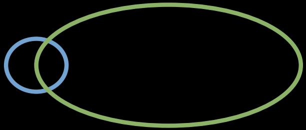
PSI on **small sets** (hundreds)

- private availability poll
- key agreement techniques



PSI on **large sets** (millions)

- double-registered voters
- OT extension; combinatorial tricks

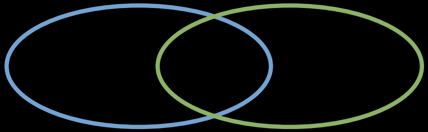


PSI on **asymmetric sets** (100 : billion)

- contact discovery; password checkup
- offline phase; leakage

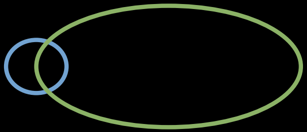
PSI on **small sets** (hundreds)

- private availability poll
- key agreement techniques



PSI on **large sets** (millions)

- double-registered voters
- OT extension; combinatorial tricks

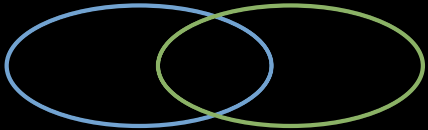


PSI on **asymmetric sets** (100 : billion)

- contact discovery; password checkup
- offline phase; leakage

computing on the intersection

- sales statistics about intersection
- generic MPC



PSI on **small sets** (hundreds)

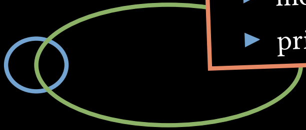
- private availability poll
- key agreement techni

PSI on **large sets** (millions)

- double-registered voters
- n; combinatorial tricks

Not to mention:

- approximate/fuzzy matching
- more than 2 parties/sets
- private set *union*



PSI on **asymmetric sets** (100 : billion)

- contact discovery; password checkup
- offline phase; leakage

computing on the intersection

- sales statistics about intersection
- generic MPC

PSI on ***small sets***

key agreement techniques

*a **bad** mental model for PSI*

Alice

$x_1, x_2, \ldots$

H = good cryptographic hash function

Bob

$y_1, y_2, \ldots$

$$H(y_1), H(y_2), \ldots \longleftarrow$$

*a **bad** mental model for PSI*

Alice

$x_1, x_2, \ldots$

H = good cryptographic hash function

Bob

$y_1, y_2, \ldots$

$H(y_1), H(y_2), \ldots$

compare

$H(x_1), H(x_2), \ldots$

a **bad** mental model for PSI

Alice

$x_1, x_2, \ldots$

H = good cryptographic hash function

Bob

$y_1, y_2, \ldots$

$H(y_1), H(y_2), \ldots$

compare

$H(x_1), H(x_2), \ldots$

Dictionary attack:

- compute $H(u)$ for every u

*a **bad** mental model for PSI*

Alice

H = good cryptographic hash f

Bob

$x_1, x_2, \ldots$

$y_1, y_2, \ldots$


FEDERAL TRADE COMMISSION
PROTECTING AMERICA'S CONSUMERS

ABOUT THE FTC | NEWS & EVENTS | ENFORCEMENT | POLI

Home » News & Events » Blogs » Tech@FTC » Does Hashing Make Data "Anonymous"?

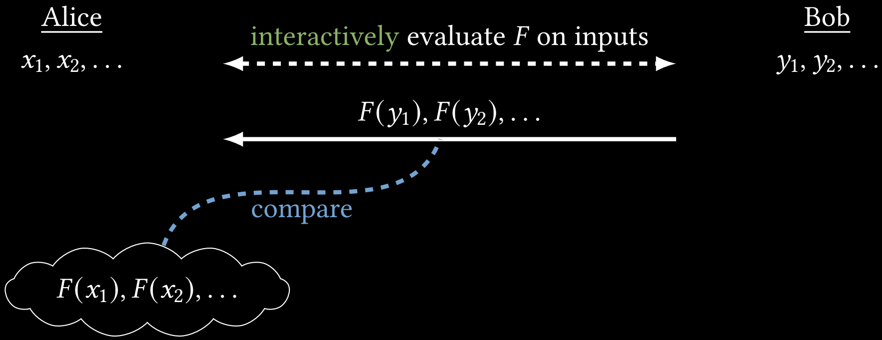
Does Hashing Make Data "Anonymous"?

By: Ed Felten, Chief Technologist | Apr 22, 2012 7:05AM

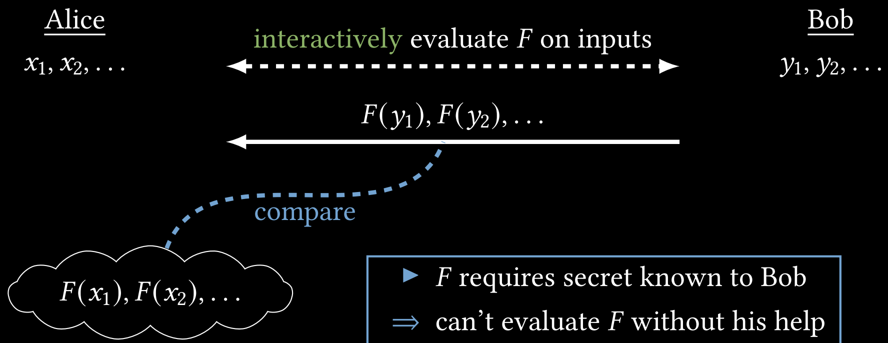
$H(x_1), H$

every u

*a **better** mental model for PSI*



*a **better** mental model for PSI*



- F requires secret known to Bob

$\Rightarrow$ can’t evaluate F without his help

- Alice “committed” to few x_i’s

Alice

x

Bob

y

Does $x = y$?

[Shamir80, Meadows86, Jablon96]

H = random oracle

Alice		Bob
x		y
	$\xrightarrow{\quad H(x)^a \quad}$	
	$\xleftarrow{\quad H(y)^b \quad}$	

[Shamir80, Meadows86, Jablon96]

Alice x

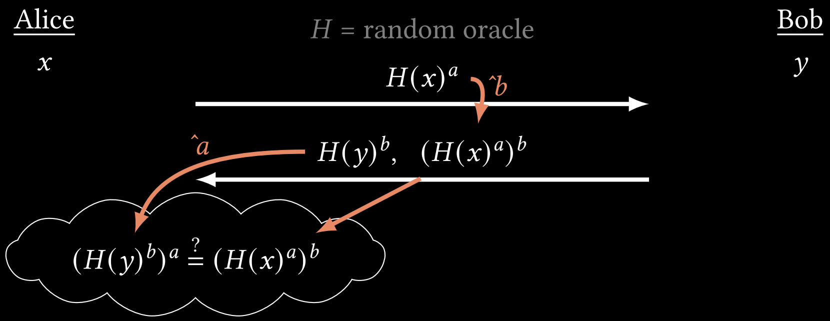
H = random oracle

Bob y

$$\xrightarrow{\quad H(x)^a \quad \hat{}b \quad}$$

$$\xleftarrow{\quad H(y)^b, \quad (H(x)^a)^b \quad}$$

[Shamir80, Meadows86, Jablon96]



[Shamir80, Meadows86, Jablon96]

Alice $\qquad\qquad$ H = random oracle $\qquad\qquad$ **Bob**

x $\qquad\qquad\qquad\qquad$ y

$$H(x)^a \xrightarrow{\hat{}b}$$

$$\hat{}a \qquad H(y)^b, \quad (H(x)^a)^b$$

$$(H(y)^b)^a \stackrel{?}{=} (H(x)^a)^b$$

interactively evaluate

$$F(x) = H(x)^{ab}$$

[Shamir80, Meadows86, Jablon96]

Alice x

H = random oracle

Bob y

$$H(x)^a \xrightarrow{\hat{}\,b}$$

$$\xleftarrow{\hat{}\,a} \quad H(y)^b, \quad (H(x)^a)^b$$

$$(H(y)^b)^a \stackrel{?}{=} (H(x)^a)^b$$

$$\boxed{\text{interactively evaluate} \quad F(x) = H(x)^{ab}}$$

$$x \neq y \overset{\text{RO}}{\Longrightarrow} H(y) \text{ independent of everything else} \overset{\text{DDH}}{\Longrightarrow} H(y)^b \approx \$$$

[Shamir80, Meadows86, Jablon96]

Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

What is $X \cap Y$*?*

[HubermanFranklinHogg99]

$$\begin{array}{ccc} \underline{\text{Alice}} & & \underline{\text{Bob}} \\ x_1, x_2, \ldots & \xrightarrow{H(x_1)^a,\, H(x_2)^a,\, \ldots} & y_1, y_2, \ldots \\ & \xleftarrow{H(y_1)^b,\, H(y_2)^b,\, \ldots,\, (H(x_1)^a)^b,\, (H(x_2)^a)^b,\, \ldots} & \end{array}$$

[HubermanFranklinHogg99]

Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$$\xrightarrow{H(x_1)^a,\, H(x_2)^a,\, \ldots}$$

$$\xleftarrow{H(y_1)^b,\, H(y_2)^b,\, \ldots,\, (H(x_1)^a)^b,\, (H(x_2)^a)^b,\, \ldots}$$

[HubermanFranklinHogg99]

- Malicious security via ZK [DeCristofaroKimTsudik10,JareckiLiu09]
- Authenticated items [DeCristofaroKimTsudik10]
- From generic key agreement [RosulekTrieu21]

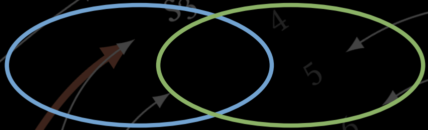
overview: PSI on small sets

for 256 items:

0.1 seconds; 10 KB

with malicious security!

[RosulekTrieu21]



PSI on ***large sets***

OT & hashing techniques; scaling to 1M items

scaling to 1 million items?

$$\xrightarrow{H(x_1)^a, H(x_2)^a, \ldots, H(x_{1000000})^a}$$

scaling to 1 million items?

$$\underbrace{H(x_1)^a, H(x_2)^a, \ldots, H(x_{1000000})^a}_{> \text{4 minutes!}}$$

batch oblivious PRF (OPRF)

Alice | Bob

1
2
3
4
5
6
7
8
9
⋮

batch oblivious PRF (OPRF)

Alice | Bob

x_1 1
x_2 2
x_3 3
x_4 4
x_5 5
x_6 6
x_7 7
x_8 8
x_9 9
$\vdots$

batch oblivious PRF (OPRF)

Alice		Bob
$F_1(x_1)$	1	$F_1(\cdot)$
$F_2(x_2)$	2	$F_2(\cdot)$
$F_3(x_3)$	3	$F_3(\cdot)$
$F_4(x_4)$	4	$F_4(\cdot)$
$F_5(x_5)$	5	$F_5(\cdot)$
$F_6(x_6)$	6	$F_6(\cdot)$
$F_7(x_7)$	7	$F_7(\cdot)$
$F_8(x_8)$	8	$F_8(\cdot)$
$F_9(x_9)$	9	$F_9(\cdot)$
	$\vdots$	

batch oblivious PRF (OPRF)

Alice				Bob
	$F_1(x_1)$	1	$F_1(\cdot)$	
	$F_2(x_2)$	2	$F_2(\cdot)$	
	$F_3(x_3)$	3	$F_3(\cdot)$	
	$F_4(x_4)$	4	$F_4(\cdot)$	
	$F_5(x_5)$	5	$F_5(\cdot)$	learns nothing about x_i's
	$F_6(x_6)$	6	$F_6(\cdot)$	
	$F_7(x_7)$	7	$F_7(\cdot)$	
	$F_8(x_8)$	8	$F_8(\cdot)$	
	$F_9(x_9)$	9	$F_9(\cdot)$	
		$\vdots$		

batch oblivious PRF (OPRF)

Alice				Bob
	$F_1(x_1)$	1	$F_1(\cdot)$	
	$F_2(x_2)$	2	$F_2(\cdot)$	
	$F_3(x_3)$	3	$F_3(\cdot)$	
	$F_4(x_4)$	4	$F_4(\cdot)$	
all other $F_i(x^*)$ look random	$F_5(x_5)$	5	$F_5(\cdot)$	learns nothing about x_i's
	$F_6(x_6)$	6	$F_6(\cdot)$	
	$F_7(x_7)$	7	$F_7(\cdot)$	
	$F_8(x_8)$	8	$F_8(\cdot)$	
	$F_9(x_9)$	9	$F_9(\cdot)$	
		$\vdots$		

batch oblivious PRF (OPRF)

Alice			Bob

Alice				Bob
	$F_1(x_1)$	1	$F_1(\cdot)$	
	$F_2(x_2)$	2	$F_2(\cdot)$	
	$F_3(x_3)$	3	$F_3(\cdot)$	
	$F_4(x_4)$	4	$F_4(\cdot)$	
all other $F_i(x^*)$ look random	$F_5(x_5)$	5	$F_5(\cdot)$	learns nothing about x_i's
	$F_6(x_6)$	6	$F_6(\cdot)$	
	$F_7(x_7)$	7	$F_7(\cdot)$	
	$F_8(x_8)$	8	$F_8(\cdot)$	
	$F_9(x_9)$	9	$F_9(\cdot)$	
		$\vdots$		

achieved very efficiently from OT extension

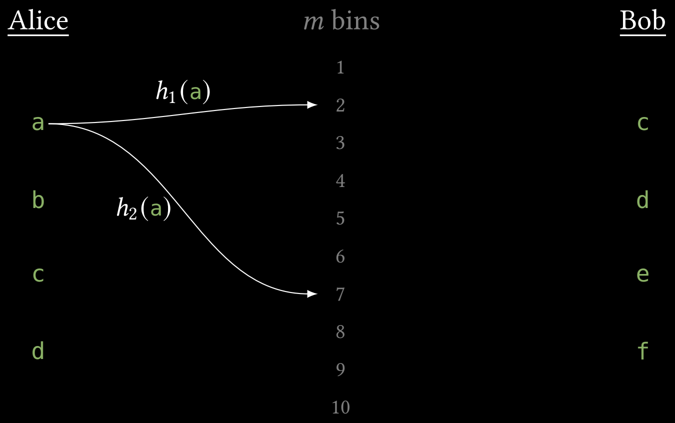
Alice	Bob
a	c
b	d
c	e
d	f

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]

Alice	m bins	Bob
	1	
	2	
a	3	c
b	4	d
	5	
c	6	e
	7	
d	8	f
	9	
	10	

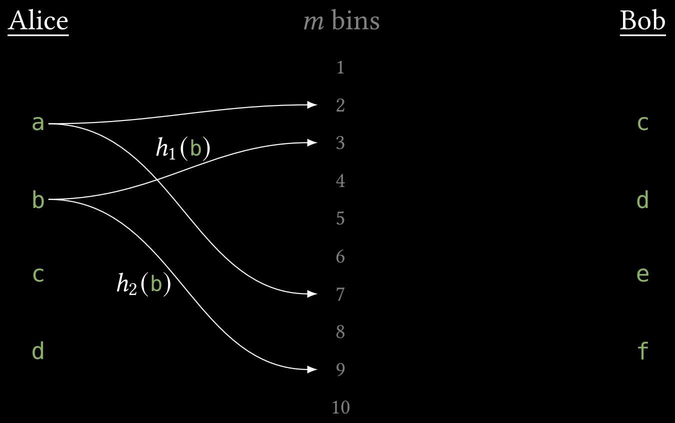
1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



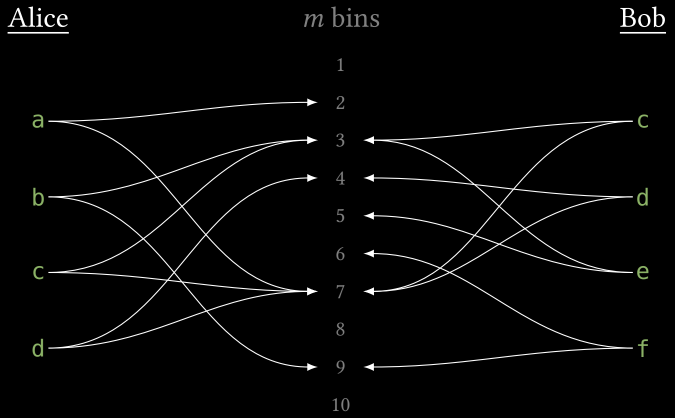
1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



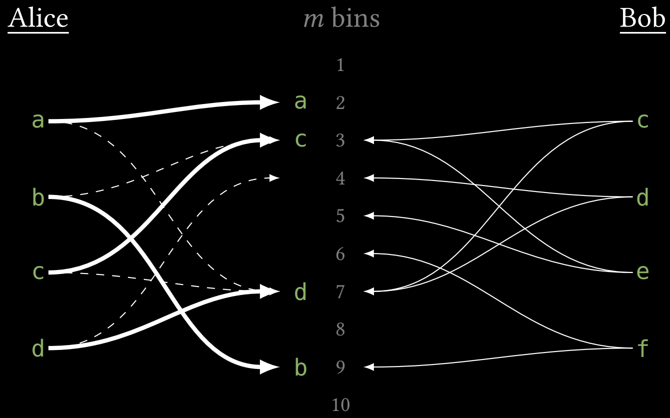
1. Agree on random
   $h_1, h_2 : \{0,1\}^* \to [m]$

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$

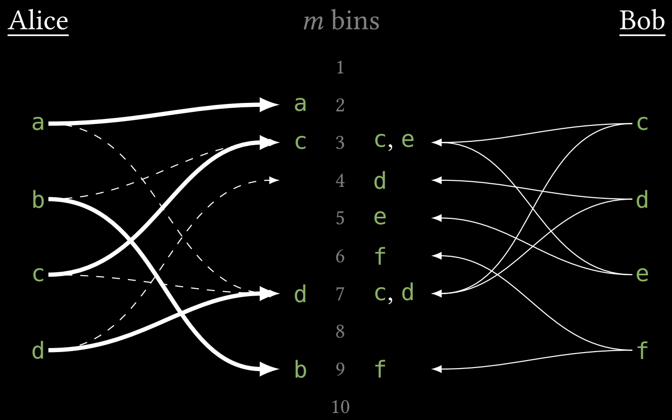
[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



1. Agree on random
$h_1, h_2 : \{0,1\}^* \to [m]$

2. Alice places each x into bin $h_1(x)$ or $h_2(x)$

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]

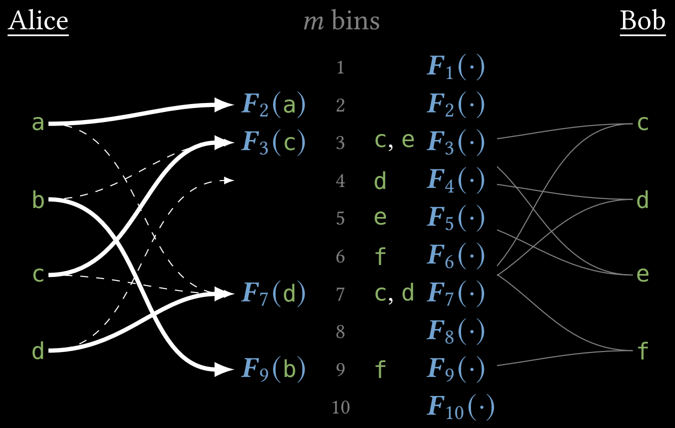


1. Agree on random $h_1, h_2 : \{0,1\}^* \longrightarrow [m]$

2. Alice places each x into bin $h_1(x)$ or $h_2(x)$

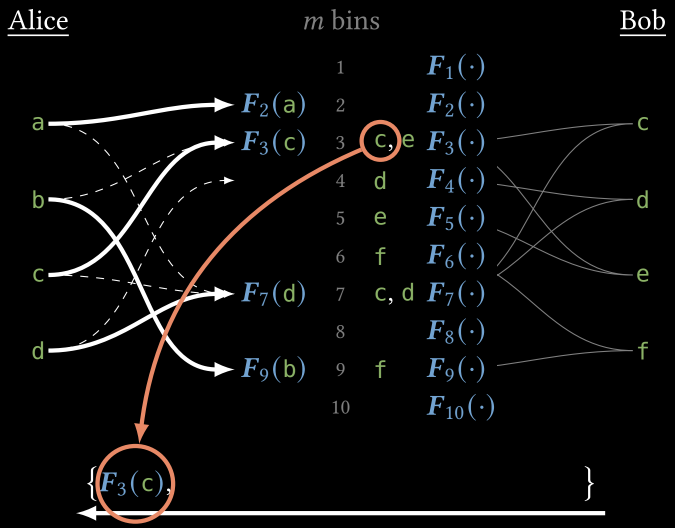
3. Bob places each x into bins $h_1(x)$ and $h_2(x)$

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



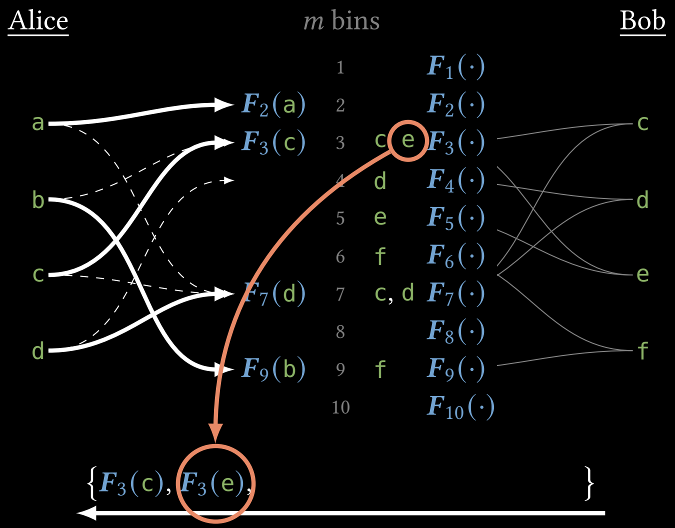
1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$
2. Alice places each x into bin $h_1(x)$ or $h_2(x)$
3. Bob places each x into bins $h_1(x)$ and $h_2(x)$
4. OPRF in each bin: Alice learns one $F_i(x)$; Bob learns entire $F_i(\cdot)$

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



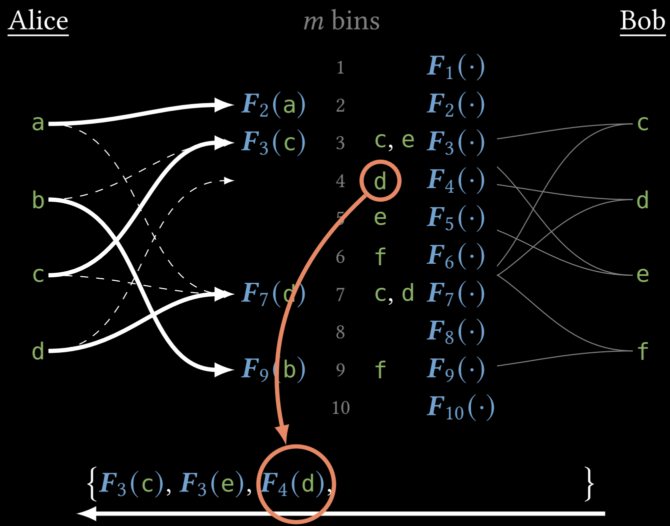
1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$
2. Alice places each x into bin $h_1(x)$ or $h_2(x)$
3. Bob places each x into bins $h_1(x)$ and $h_2(x)$
4. OPRF in each bin: Alice learns one $F_i(x)$; Bob learns entire $F_i(\cdot)$
5. Bob sends all $F_i(x)$ values

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



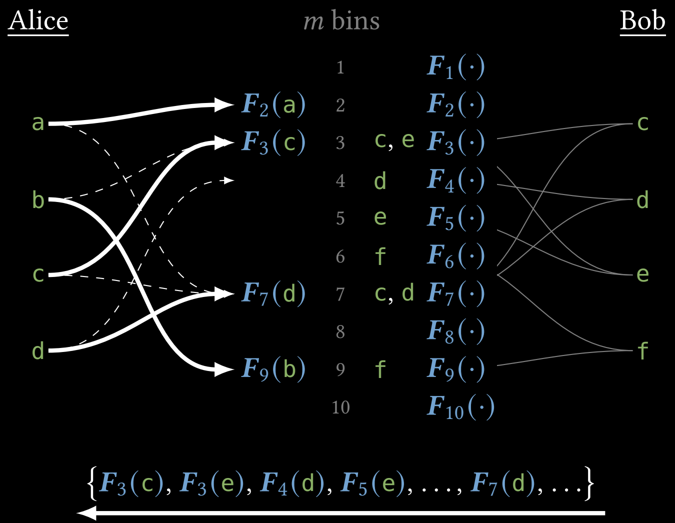
1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$
2. Alice places each x into bin $h_1(x)$ or $h_2(x)$
3. Bob places each x into bins $h_1(x)$ and $h_2(x)$
4. OPRF in each bin: Alice learns one $F_i(x)$; Bob learns entire $F_i(\cdot)$
5. Bob sends all $F_i(x)$ values

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



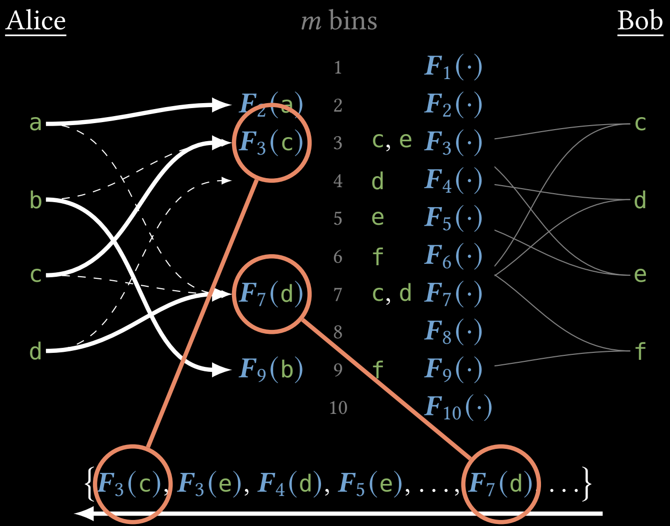
1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$
2. Alice places each x into bin $h_1(x)$ or $h_2(x)$
3. Bob places each x into bins $h_1(x)$ and $h_2(x)$
4. OPRF in each bin: Alice learns one $F_i(x)$; Bob learns entire $F_i(\cdot)$
5. Bob sends all $F_i(x)$ values

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$
2. Alice places each x into bin $h_1(x)$ or $h_2(x)$
3. Bob places each x into bins $h_1(x)$ and $h_2(x)$
4. OPRF in each bin: Alice learns one $F_i(x)$; Bob learns entire $F_i(\cdot)$
5. Bob sends all $F_i(x)$ values

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]



1. Agree on random $h_1, h_2 : \{0,1\}^* \to [m]$
2. Alice places each x into bin $h_1(x)$ or $h_2(x)$
3. Bob places each x into bins $h_1(x)$ and $h_2(x)$
4. OPRF in each bin: Alice learns one $F_i(x)$; Bob learns entire $F_i(\cdot)$
5. Bob sends all $F_i(x)$ values

[PinkasSchneiderZohner14, KolesnikovKumaresanRosulekTrieu16]

why isn’t it secure against malicious parties?

why isn't it secure against malicious parties?

Alice

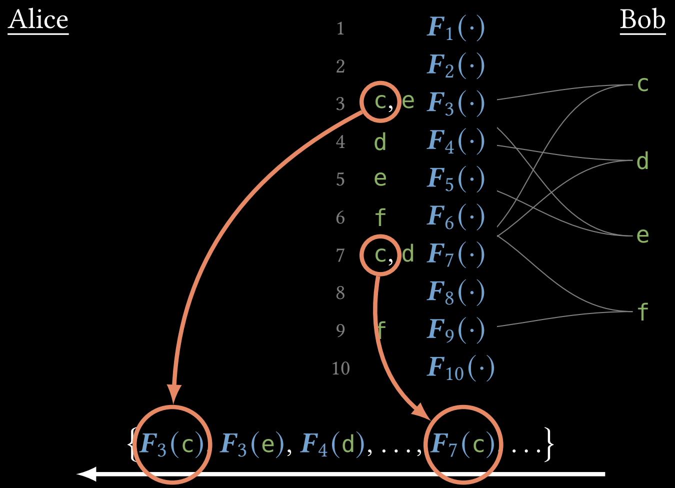
Bob

1		$F_1(\cdot)$	
2		$F_2(\cdot)$	c
3	c, e	$F_3(\cdot)$	
4	d	$F_4(\cdot)$	d
5	e	$F_5(\cdot)$	
6	f	$F_6(\cdot)$	e
7	c, d	$F_7(\cdot)$	
8		$F_8(\cdot)$	f
9	f	$F_9(\cdot)$	
10		$F_{10}(\cdot)$	

$\{F_3(\mathsf{c}), F_3(\mathsf{e}), F_4(\mathsf{d}), \ldots, F_7(\mathsf{c}), \ldots\}$

$\longleftarrow$

why isn't it secure against malicious parties?

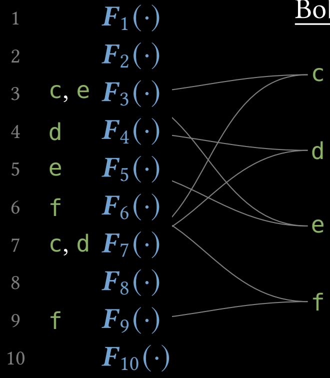


Bob should send two F-values per item

why isn't it secure against malicious parties?

Alice

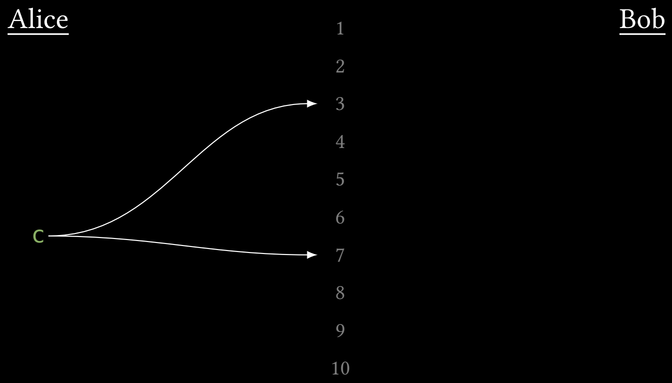
Bob



Bob should send two F-values per item , what if he sends only one?

$\{F_3(\mathsf{c}), F_3(\mathsf{e}), F_4(\mathsf{d}), \ldots, \cancel{F_7(\mathsf{c})}, \ldots\}$

why isn't it secure against malicious parties?



$$\{F_3(\mathtt{c}), F_3(\mathtt{e}), F_4(\mathtt{d}), \ldots, \cancel{F_7(\mathtt{c})}, \ldots\}$$

Bob should send two F-values per item , what if he sends only one?

Alice has c; does she include it in output?

why isn't it secure against malicious parties?

<u>Alice</u>

<u>Bob</u>

1

2

$F_3(c)$ 3

4

5

6

c

7

8

9

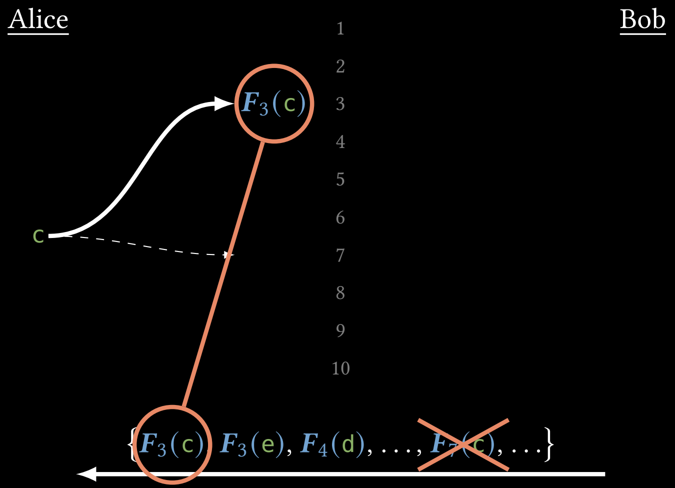
10

$\{F_3(c), F_3(e), F_4(d), \ldots, \cancel{F_7(c)}, \ldots\}$

Bob should send two F-values per item , what if he sends only one?

Alice has c; does she include it in output?

why isn't it secure against malicious parties?



Bob should send two F-values per item , what if he sends only one?

Alice has c; does she include it in output?

why isn't it secure against malicious parties?

Alice

Bob

1
2
3
4
5
6
7
8
9
10

$\mathtt{c}$

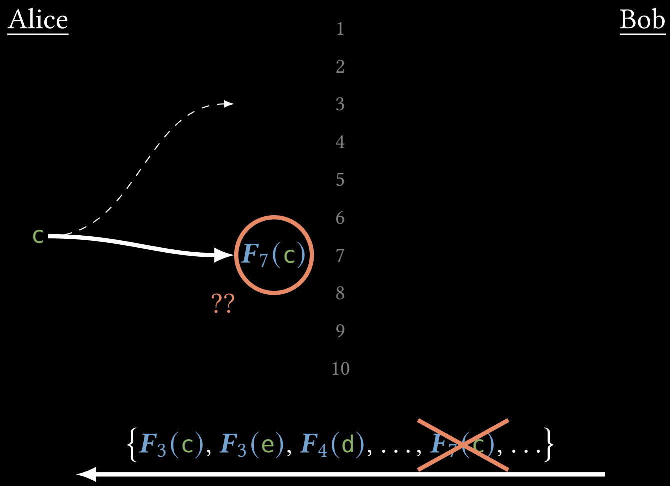
$\boldsymbol{F}_7(\mathtt{c})$

$\{\boldsymbol{F}_3(\mathtt{c}), \boldsymbol{F}_3(\mathtt{e}), \boldsymbol{F}_4(\mathtt{d}), \ldots, \cancel{\boldsymbol{F}_7(\mathtt{c})}, \ldots\}$

Bob should send two $\boldsymbol{F}$-values per item , what if he sends only one?

Alice has $\mathtt{c}$; does she include it in output?

why isn't it secure against malicious parties?

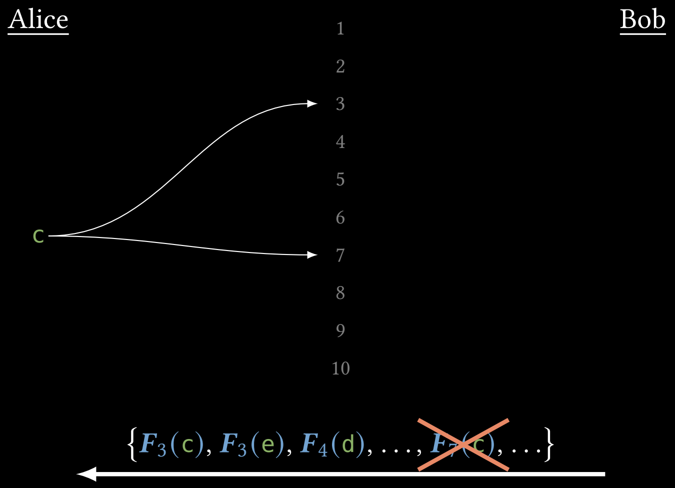


Bob should send two F-values per item , what if he sends only one?

Alice has c; does she include it in output?

Only if c placed in bin 3!

why isn't it secure against malicious parties?



Bob should send two F-values per item , what if he sends only one?

Alice has c; does she include it in output?

Only if c placed in bin 3!

- Depends on Alice's **entire input**!

$\Rightarrow$ can't simulate!

how do we overcome this problem?

[PinkasRosulekTrieuYanai20]

batch OPRF for malicious PSI

Alice		Bob
$F_1(x_1)$	1	$F_1(\cdot)$
$F_2(x_2)$	2	$F_2(\cdot)$
$F_3(x_3)$	3	$F_3(\cdot)$
$F_4(x_4)$	4	$F_4(\cdot)$
$F_5(x_5)$	5	$F_5(\cdot)$
$F_6(x_6)$	6	$F_6(\cdot)$
$F_7(x_7)$	7	$F_7(\cdot)$
$F_8(x_8)$	8	$F_8(\cdot)$
$F_9(x_9)$	9	$F_9(\cdot)$
	$\vdots$	

batch OPRF for malicious PSI

Alice		Bob
$F_1(x_1)$	1	$F_1(\cdot)$
$F_2(x_2)$	2	$F_2(\cdot)$
$F_3(x_3)$	3	$F_3(\cdot)$
$F_4(x_4)$	4	$F_4(\cdot)$
$F_5(x_5)$	5	$F_5(\cdot)$
$F_6(x_6)$	6	$F_6(\cdot)$
$F_7(x_7)$	7	$F_7(\cdot)$
$F_8(x_8)$	8	$F_8(\cdot)$
$F_9(x_9)$	9	$F_9(\cdot)$
	$\vdots$	

State of the art malicious batch OPRF [OrrùOrsiniScholl17]

- essentially same cost as semi-honest

batch OPRF for malicious PSI

Alice		Bob
$F_1(x_1)$	1	$F_1(\cdot)$
$F_2(x_2)$	2	$F_2(\cdot)$
$F_3(x_3)$	3	$F_3(\cdot)$
$F_4(x_4)$	4	$F_4(\cdot)$
$F_5(x_5)$	5	$F_5(\cdot)$
$F_6(x_6)$	6	$F_6(\cdot)$
$F_7(x_7)$	7	$F_7(\cdot)$
$F_8(x_8)$	8	$F_8(\cdot)$
$F_9(x_9)$	9	$F_9(\cdot)$
	$\vdots$	

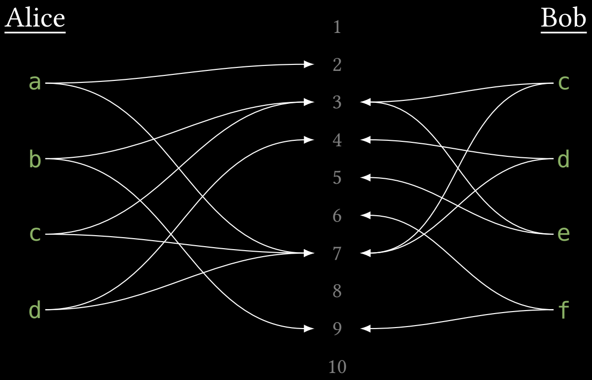
State of the art malicious batch OPRF [OrrùOrsiniScholl17]

- essentially same cost as semi-honest
- consistency check relies on an additive homomorphism:

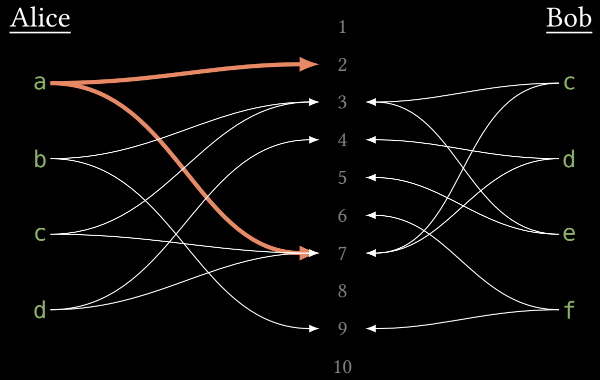
$$F_i(x) \oplus F_j(y) = F_{ij}(x \oplus y)$$

*: *a gross oversimplification*

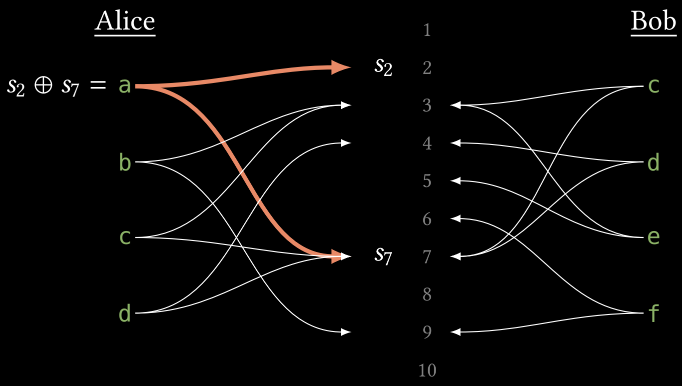
[PinkasRosulekTrieuYanai20] *protocol main idea:*



[PinkasRosulekTrieuYanai20] *protocol main idea:*

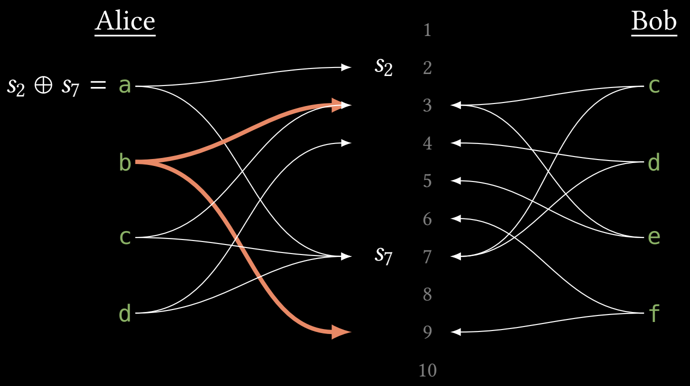


[PinkasRosulekTrieuYanai20] *protocol main idea:*



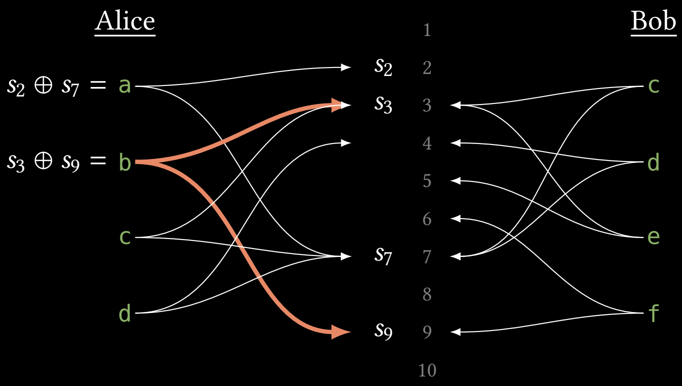
Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

[PinkasRosulekTrieuYanai20] *protocol main idea:*



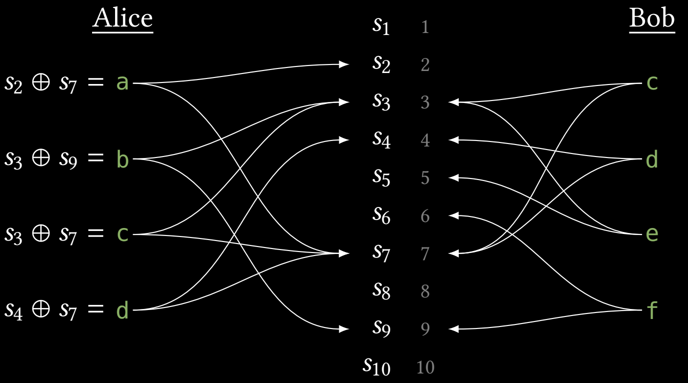
Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

[PinkasRosulekTrieuYanai20] *protocol main idea:*



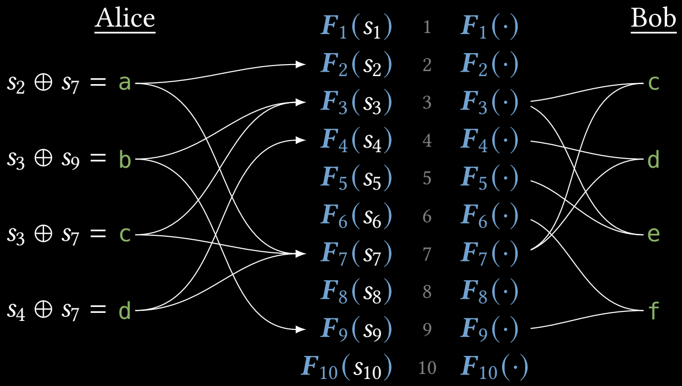
Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

[PinkasRosulekTrieuYanai20] *protocol main idea:*



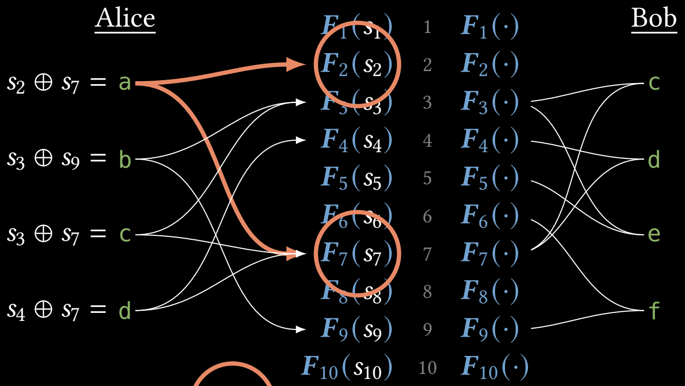
Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

[PinkasRosulekTrieuYanai20] *protocol main idea:*



Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

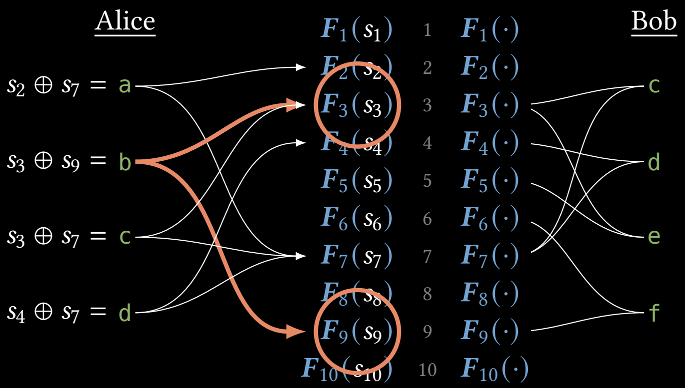
[PinkasRosulekTrieuYanai20] *protocol main idea:*



Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

$$F_2(s_2) \oplus F_7(s_7) = F_{27}(\mathsf{a})$$

[PinkasRosulekTrieuYanai20] *protocol main idea:*



Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

$$F_2(s_2) \oplus F_7(s_7) = F_{27}(\mathsf{a})$$
$$F_3(s_3) \oplus F_9(s_9) = F_{39}(\mathsf{b})$$

[PinkasRosulekTrieuYanai20] *protocol main idea:*

Alice

$s_2 \oplus s_7 = \mathsf{a}$

$s_3 \oplus s_9 = \mathsf{b}$

$s_3 \oplus s_7 = \mathsf{c}$

$s_4 \oplus s_7 = \mathsf{d}$

$F_1(s_1)$	1	$F_1(\cdot)$
$F_2(s_2)$	2	$F_2(\cdot)$
$F_3(s_3)$	3	$F_3(\cdot)$
$F_4(s_4)$	4	$F_4(\cdot)$
$F_5(s_5)$	5	$F_5(\cdot)$
$F_6(s_6)$	6	$F_6(\cdot)$
$F_7(s_7)$	7	$F_7(\cdot)$
$F_8(s_8)$	8	$F_8(\cdot)$
$F_9(s_9)$	9	$F_9(\cdot)$
$F_{10}(s_{10})$	10	$F_{10}(\cdot)$

Bob

c

d

e

f

Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

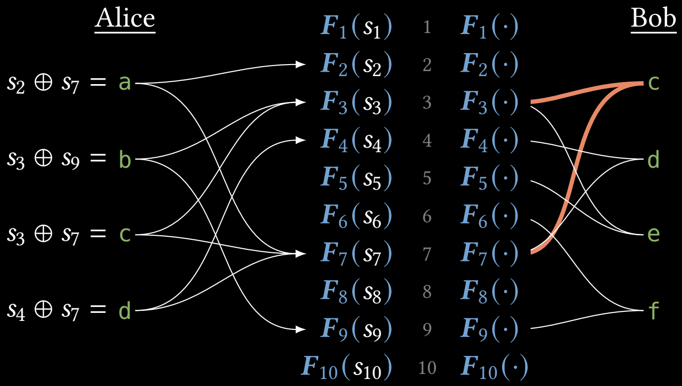
$F_2(s_2) \oplus F_7(s_7) = F_{27}(\mathsf{a})$

$F_3(s_3) \oplus F_9(s_9) = F_{39}(\mathsf{b})$

$F_3(s_3) \oplus F_7(s_7) = F_{37}(\mathsf{c})$

$F_4(s_4) \oplus F_7(s_7) = F_{47}(\mathsf{d})$

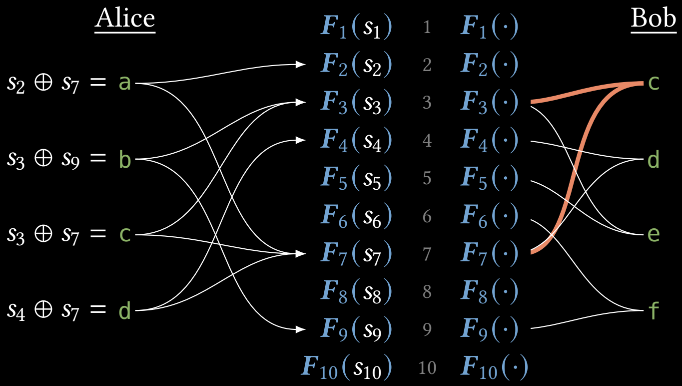
[PinkasRosulekTrieuYanai20] *protocol main idea:*



Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

$$F_2(s_2) \oplus F_7(s_7) = F_{27}(\mathsf{a})$$
$$F_3(s_3) \oplus F_9(s_9) = F_{39}(\mathsf{b})$$
$$F_3(s_3) \oplus F_7(s_7) = F_{37}(\mathsf{c})$$
$$F_4(s_4) \oplus F_7(s_7) = F_{47}(\mathsf{d})$$

[PinkasRosulekTrieuYanai20] *protocol main idea:*



Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

$F_2(s_2) \oplus F_7(s_7) = F_{27}(\mathsf{a})$

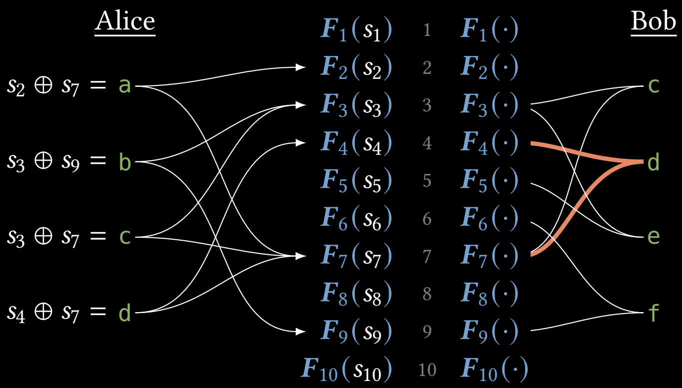
$F_3(s_3) \oplus F_9(s_9) = F_{39}(\mathsf{b})$

$F_3(s_3) \oplus F_7(s_7) = F_{37}(\mathsf{c})$

$F_4(s_4) \oplus F_7(s_7) = F_{47}(\mathsf{d})$

$\{F_{37}(\mathsf{c}), \qquad\qquad \}$ $\longleftarrow$

[PinkasRosulekTrieuYanai20] *protocol main idea:*

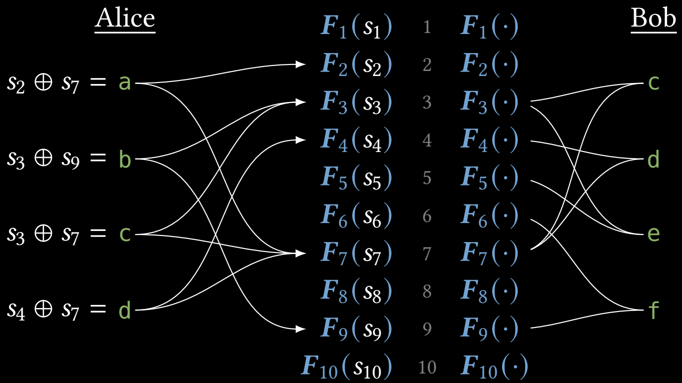


Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

$$F_2(s_2) \oplus F_7(s_7) = F_{27}(\mathsf{a})$$
$$F_3(s_3) \oplus F_9(s_9) = F_{39}(\mathsf{b})$$
$$F_3(s_3) \oplus F_7(s_7) = F_{37}(\mathsf{c})$$
$$F_4(s_4) \oplus F_7(s_7) = F_{47}(\mathsf{d})$$

$$\{F_{37}(\mathsf{c}), F_{47}(\mathsf{d}), \quad \}$$
$$\longleftarrow$$

[PinkasRosulekTrieuYanai20] *protocol main idea:*



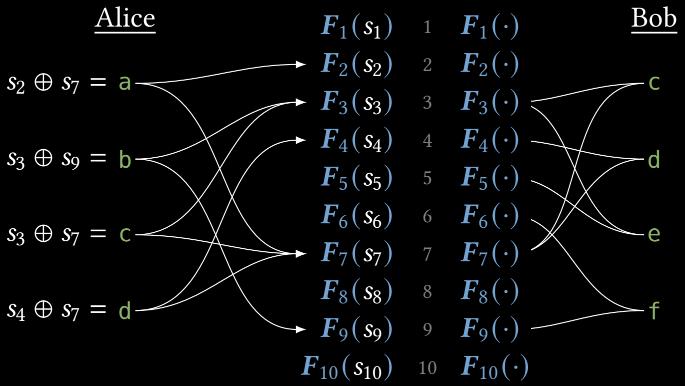
Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

Bob sends only one F-value per item

$$F_2(s_2) \oplus F_7(s_7) = F_{27}(\mathsf{a})$$
$$F_3(s_3) \oplus F_9(s_9) = F_{39}(\mathsf{b})$$
$$F_3(s_3) \oplus F_7(s_7) = F_{37}(\mathsf{c})$$
$$F_4(s_4) \oplus F_7(s_7) = F_{47}(\mathsf{d})$$

$$\{F_{37}(\mathsf{c}), F_{47}(\mathsf{d}), F_{35}(\mathsf{e}), F_{69}(\mathsf{f})\} \longleftarrow$$

[PinkasRosulekTrieuYanai20] *protocol main idea:*



Alice secret-shares x into bins $h_1(x)$ and $h_2(x)$

Bob sends only one F-value per item

$$F_2(s_2) \oplus F_7(s_7) = F_{27}(\mathsf{a})$$
$$F_3(s_3) \oplus F_9(s_9) = F_{39}(\mathsf{b})$$
$$F_3(s_3) \oplus F_7(s_7) = F_{37}(\mathsf{c})$$
$$F_4(s_4) \oplus F_7(s_7) = F_{47}(\mathsf{d})$$

$$\{F_{37}(\mathsf{c}), F_{47}(\mathsf{d}), F_{35}(\mathsf{e}), F_{69}(\mathsf{f})\} \longleftarrow$$

overview: PSI on large sets

for 1 million items:

4.5 – 5 seconds; 128 – 145 MB

[KolesnikovKumaresanRosulekTrieu16]

[GarimellaPinkasRosulekTrieuYanai21]

overview: PSI on large sets

for 1 million items:

4.5 – 5 seconds; 128 – 145 MB

semi-honest security

[KolesnikovKumaresanRosulekTrieu16]

[GarimellaPinkasRosulekTrieuYanai21]

overview: PSI on large sets

for 1 million items:

4.5 – 5 seconds; 128 – 145 MB

malicious security

[KolesnikovKumaresanRosulekTrieu16]

[GarimellaPinkasRosulekTrieuYanai21]

*PSI on **asymmetric sets***

offline preprocessing techniques and leakage; scaling to billions of items

how to scale to billions of items?

THE VERGE TECH REVIEWS SCIENCE CREATORS ENTERTAINMENT

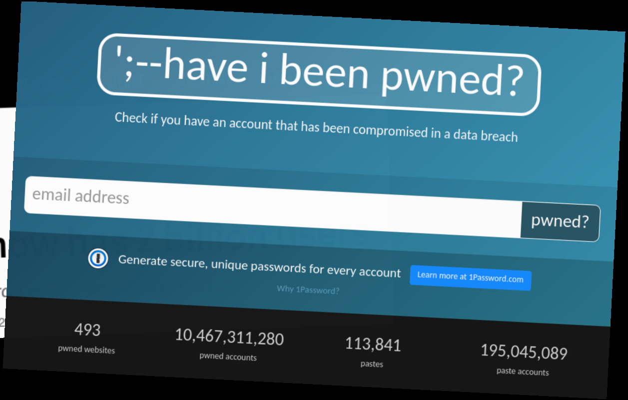
APPS TECH FACEBOOK

WhatsApp now has 2 billion users

And it has no plans to drop end-to-end encryption

By Jon Porter | @JonPorty | Feb 12, 2020, 10:50am EST

how to scale to billions of items?



idea #1: offline preprocessing

Alice		Bob
$x_1, x_2, \ldots$	$H(x_1)^a, H(x_2)^a, \ldots$ $\longrightarrow$	$y_1, y_2, \ldots$
(hundreds)		*(billions!)*
	$H(y_1)^b, H(y_2)^b, \ldots,\ (H(x_1)^a)^b, (H(x_2)^a)^b, \ldots$ $\longleftarrow$	

see [KalesRechbergerSchneiderSenkerWeinert19]

idea #1: offline preprocessing

Alice

$x_1, x_2, \ldots$

(hundreds)

Bob

$y_1, y_2, \ldots$

(billions!)

$$\xrightarrow{H(x_1)^a, H(x_2)^a, \ldots}$$

$$\xleftarrow{H(y_1)^b, H(y_2)^b, \ldots, \; (H(x_1)^a)^b, (H(x_2)^a)^b, \ldots}$$

see [KalesRechbergerSchneiderSenkerWeinert19]

idea #1: offline preprocessing

Alice

$x_1, x_2, \ldots$

(hundreds)

Bob

$y_1, y_2, \ldots$

(billions!)

$$H(x_1)^a, H(x_2)^a, \ldots \longrightarrow$$

$$H(y_1)^b, H(y_2)^b, \ldots, \quad (H(x_1)^a)^b, (H(x_2)^a)^b, \ldots \longleftarrow$$

only dependence on Bob's set

see [KalesRechbergerSchneiderSenkerWeinert19]

idea #1: offline preprocessing

$H(y_1)^b, H(y_2)^b, \ldots$ ←

offline phase

Alice

$x_1, x_2, \ldots$

(hundreds)

Bob

$y_1, y_2, \ldots$

(billions!)

$H(x_1)^a, H(x_2)^a, \ldots$ →

$(H(x_1)^a)^b, (H(x_2)^a)^b, \ldots$ ←

see [KalesRechbergerSchneiderSenkerWeinert19]

idea #1: offline preprocessing

$H(y_1)^b, H(y_2)^b, \ldots$ ←

offline phase, shared among all clients

Alice

$x_1, x_2, \ldots$

(hundreds)

Bob

$y_1, y_2, \ldots$

(billions!)

$H(x_1)^a, H(x_2)^a, \ldots$ →

$(H(x_1)^a)^b, (H(x_2)^a)^b, \ldots$ ←

- Safe to reuse b for many PSIs $\Longrightarrow$ reuse offline phase for all clients!

see [KalesRechbergerSchneiderSenkerWeinert19]

idea #1: offline preprocessing

$H(y_1)^b, H(y_2)^b, \ldots$

offline phase, shared among all clients

Alice

$x_1, x_2, \ldots$

(hundreds)

Bob

$y_1, y_2, \ldots$

(billions!)

$H(x_1)^a, H(x_2)^a, \ldots$

$(H(x_1)^a)^b, (H(x_2)^a)^b, \ldots$

- Safe to reuse b for many PSIs $\Longrightarrow$ reuse offline phase for all clients!
- Clever encodings for offline message: 4GB / 1B items

see [KalesRechbergerSchneiderSenkerWeinert19]

idea #2: allow some leakage

100 items

Alice:

Bob:

1 billion items

see [LiPalAliSullivanChatterjeeRistenpart19]

idea #2: allow some leakage

Alice:

100 items

1 million bins

Bob:

1 billion items

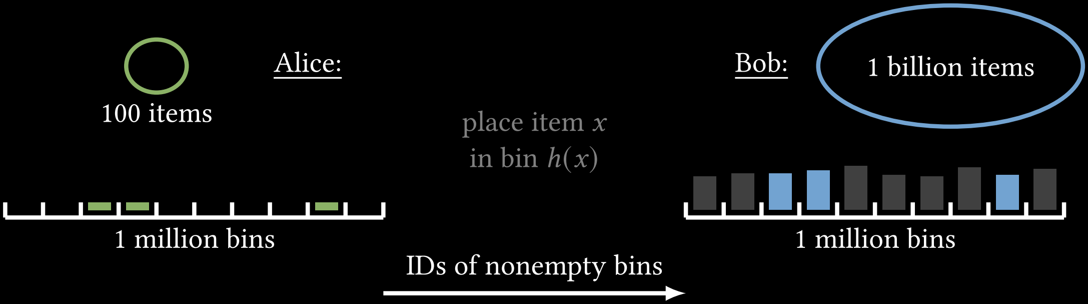
1 million bins

see [LiPalAliSullivanChatterjeeRistenpart19]

idea #2: allow some leakage

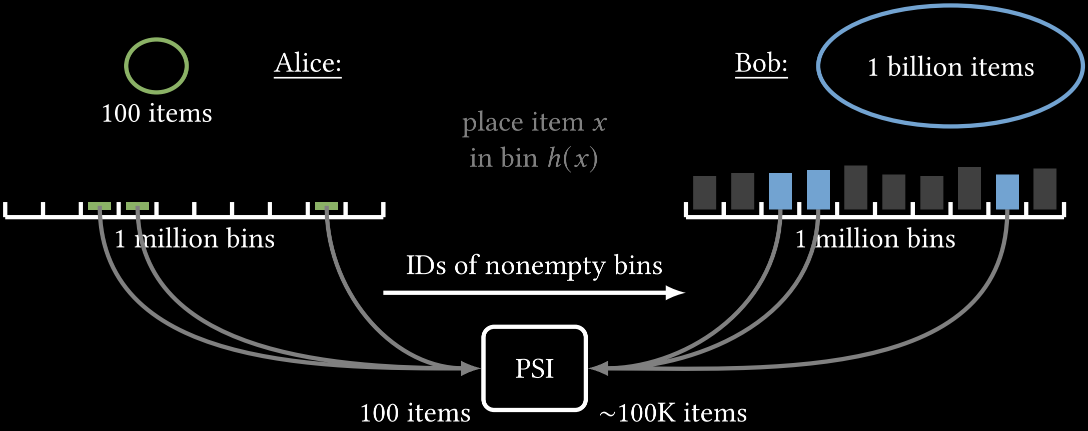
see [LiPalAliSullivanChatterjeeRistenpart19]

idea #2: allow some leakage



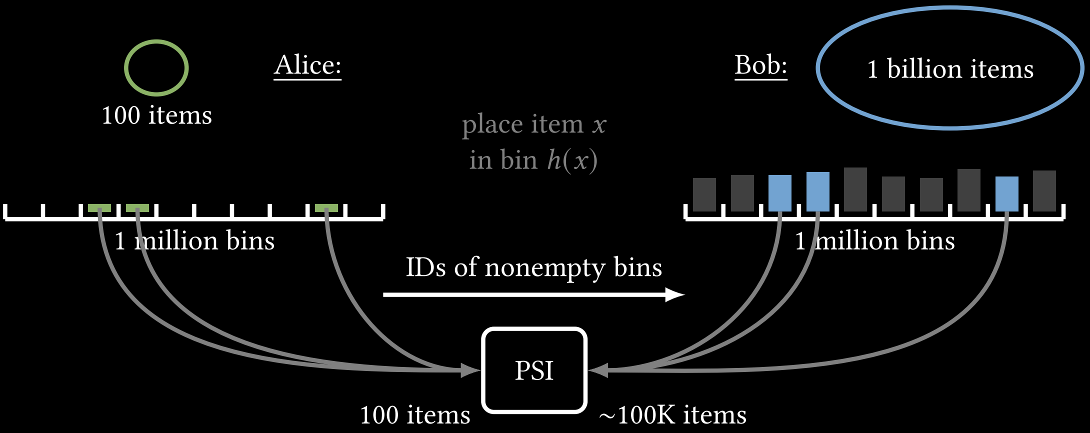
see [LiPalAliSullivanChatterjeeRistenpart19]

idea #2: allow some leakage



see [LiPalAliSullivanChatterjeeRistenpart19]

idea #2: allow some leakage



choice of h? see [LiPalAliSullivanChatterjeeRistenpart19]

overview: PSI on asymmetric sets

for 256 million vs 1000 items (no leakage):

offline setup: 33 seconds; 1 GB

discovery: 3 seconds; 6 MB

for 1 billion vs 100 items (under previous **leakage** scenario):

0.2 seconds; 1 MB

computing on the intersection

next frontier

Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$$H(x_1)^a, H(x_2)^a, \ldots \longrightarrow$$

$$H(y_1)^b, H(y_2)^b, \ldots, (H(x_1)^a)^b, (H(x_2)^a)^b, \ldots \longleftarrow$$

what is $X \cap Y$?

[HubermanFranklinHogg99]

Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$$H(x_1)^a, H(x_2)^a, \ldots$$

$$H(y_1)^b, H(y_2)^b, \ldots, \boxed{(H(x_1)^a)^b, (H(x_2)^a)^b, \ldots}$$

shuffle

what is $|X \cap Y|$*?*

[HubermanFranklinHogg99]

Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

$$\text{Alice} \xrightarrow{H(x_1)^a,\, H(x_2)^a,\, \ldots} \text{Bob}$$

$$\text{Alice} \xleftarrow{H(y_1)^b,\, H(y_2)^b,\, \ldots,\, \boxed{(H(x_1)^a)^b,\, (H(x_2)^a)^b,\, \ldots}} \text{Bob}$$

shuffle

what is $|X \cap Y|$?

what about computing other functions of the intersection? what about large sets?

[HubermanFranklinHogg99]

state of the art

Alice	Bob
$x_1, x_2, \ldots$	$y_1, y_2, \ldots$

- Using $O(n)$ communication, reduce PSI to $O(n)$ comparisons (vs n^2)

[PinkasSchneiderTkachenkoYanai19]

state of the art

Alice

$x_1, x_2, \ldots$

Bob

$y_1, y_2, \ldots$

interactive preprocessing

$r_1, r_2, \ldots$

$s_1, s_2, \ldots$

$$x_i \in Y \iff r_i = s_i$$

- Using $O(n)$ communication, reduce PSI to $O(n)$ comparisons (vs n^2)

[PinkasSchneiderTkachenkoYanai19]

state of the art

Alice	Bob
$x_1, x_2, \ldots$	$y_1, y_2, \ldots$

$\boxed{\text{interactive preprocessing}}$

Alice	Bob
$r_1, r_2, \ldots$	$s_1, s_2, \ldots$

$\boxed{\text{compare } r_i \stackrel{?}{=} s_i, (\forall i)}$

generic MPC

- Using $O(n)$ communication, reduce PSI to $O(n)$ comparisons (vs n^2)
- Perform the comparisons inside generic MPC $\rightsquigarrow$ compute on the result

[PinkasSchneiderTkachenkoYanai19]

overview: computing on the intersection

for 1 million items:

2 minutes ; 2.5 GB

[PinkasSchneiderTkachenkoYanai19]

overview: computing on the intersection

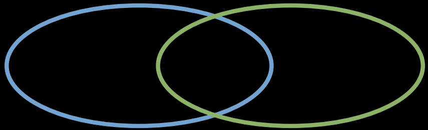
for 1 million items:

$$\underbrace{\text{2 minutes}}_{30\times\text{ plain PSI}} \; ; \; \underbrace{\text{2.5 GB}}_{20\times\text{ plain PSI}}$$

[PinkasSchneiderTkachenkoYanai19]

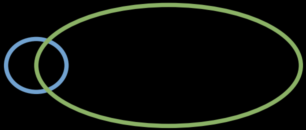
PSI on **small sets** (hundreds)

- efficient! 0.1sec / 256 items
- based on Diffie-Hellman KA



PSI on **large sets** (millions)

- fast! 4sec / 1M items
- OT extension & hashing techniques



PSI on **asymmetric sets**

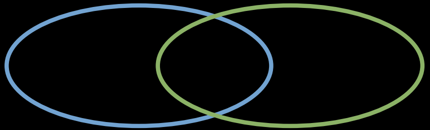
- huge challenges for practice
- allow leakage, preprocessing?

computing on the intersection

- many open problems
- 20-30× performance gap

PSI on **small sets** (hundreds)

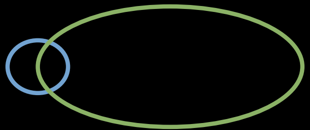
- efficient! 0.1sec / 256 items
- based on Diffie-Hellman KA



PSI on **large sets** (millions)

- fast! 4sec / 1M items
- OT extension & hashing techniques

thank you!



PSI on **asymmetric sets**

- huge challenges for practice
- allow leakage, preprocessing?

computing on the intersection

- many open problems
- 20-30× performance gap