FAST PRIVATE SET INTERSECTION FROM FULLY HOMOMORPHIC ENCRYPTION

CCS 2017

Microsoft Research

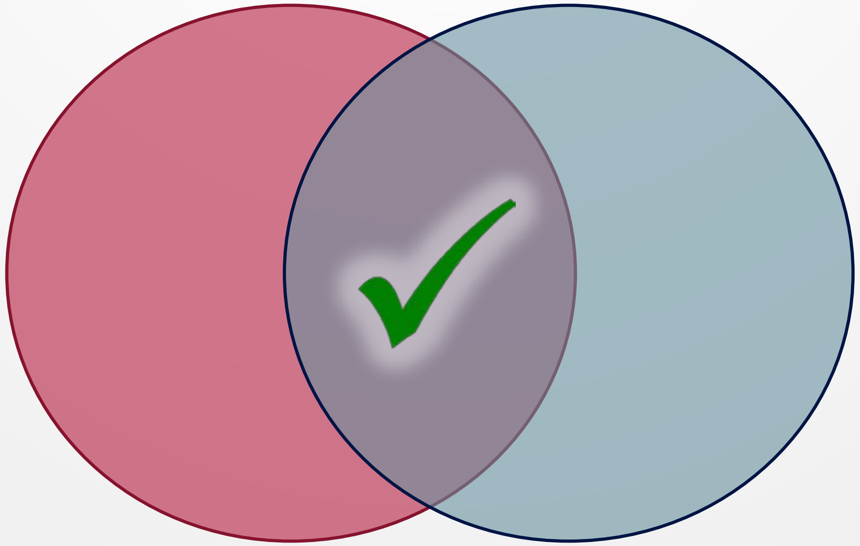
Hao Chen
Kim Laine

Peter Rindal

Oregon State University OSU

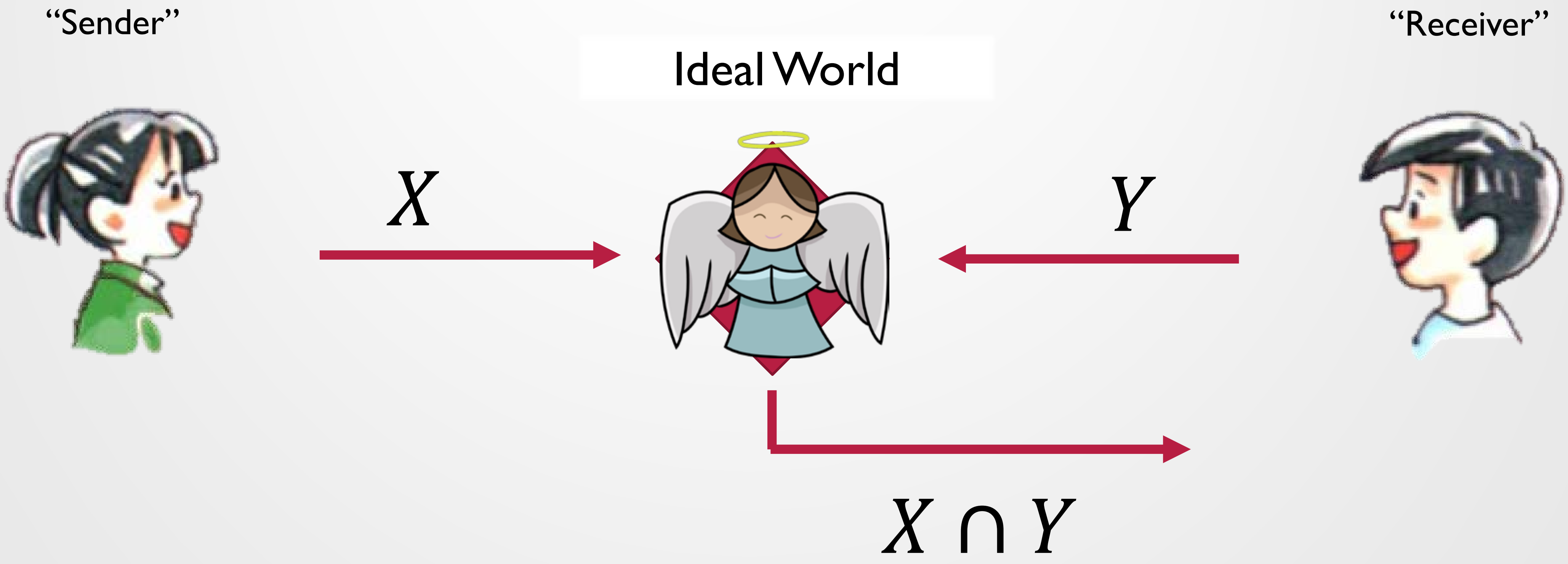
Private Set Intersection (PSI)

X Y



$X \cap Y$

Private Set Intersection (PSI)



A *Sampling* of PSI Over the Decades

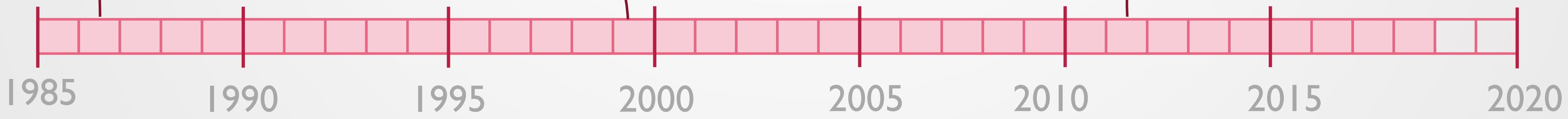
Diffie-Hellman

[Meadows86]
Private equality test

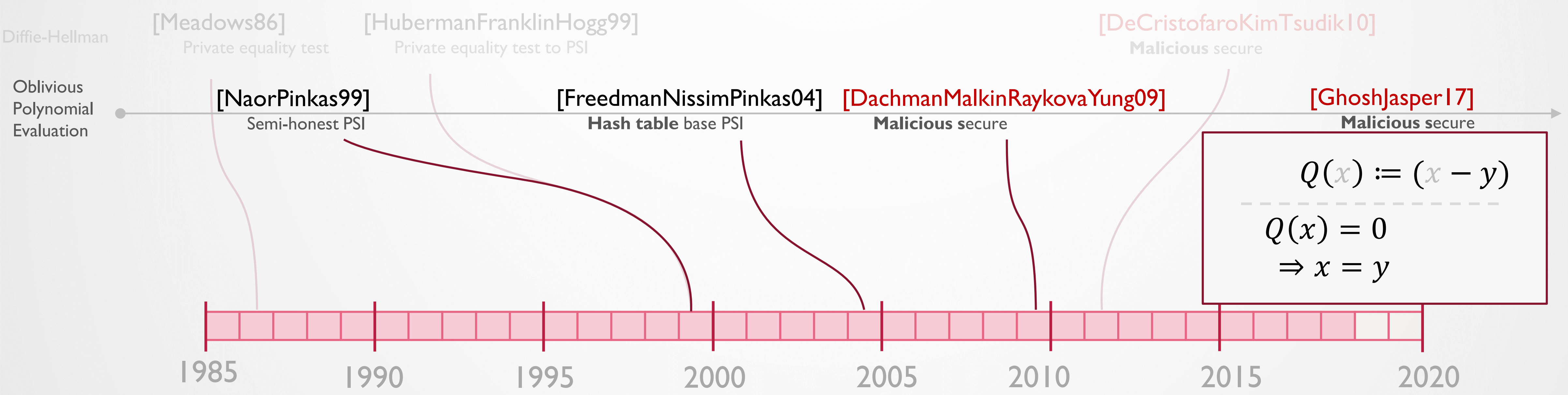
[HubermanFranklinHogg99]
Private equality test to PSI

[DeCristofaroKimTsudik10]
Malicious secure

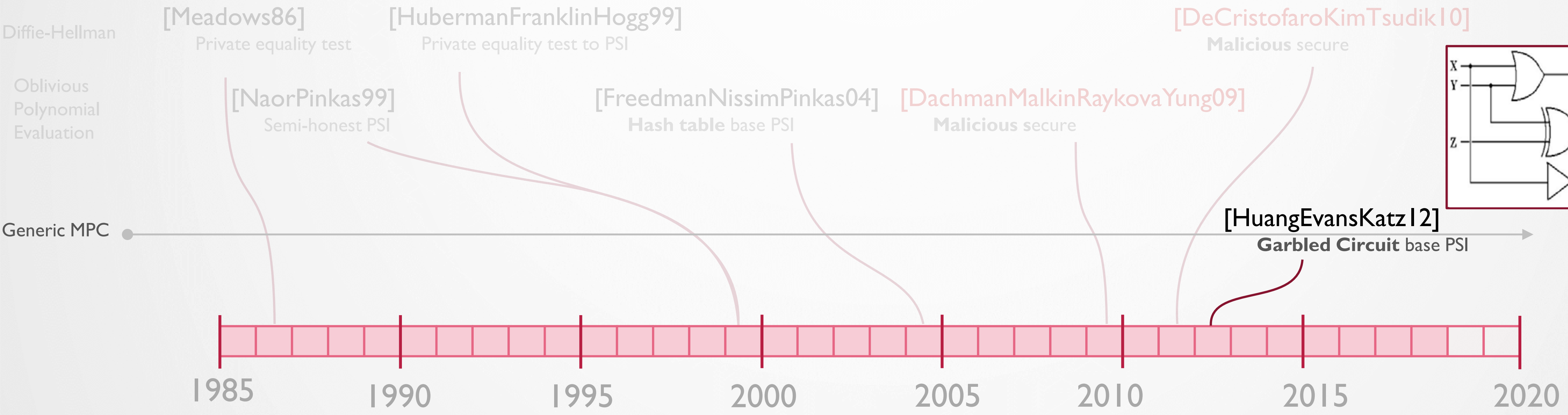
$$x^{\alpha\beta} = y^{\beta\alpha}$$
$$\Rightarrow x = y$$

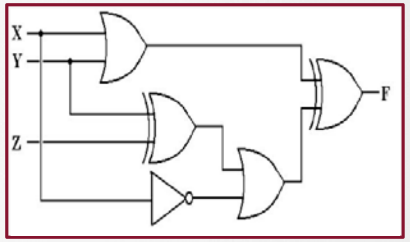


A *Sampling* of PSI Over the Decades



A *Sampling* of PSI Over the Decades





A *Sampling* of PSI Over the Decades

Diffie-Hellman

[Meadows86]
Private equality test

[HubermanFranklinHogg99]
Private equality test to PSI

[DeCristofaroKimTsudik10]
Malicious secure

Oblivious Polynomial Evaluation

[NaorPinkas99]
Semi-honest PSI

[FreedmanNissimPinkas04]
Hash table base PSI

[DachmanMalkinRaykovaYung09]
Malicious secure

Generic MPC

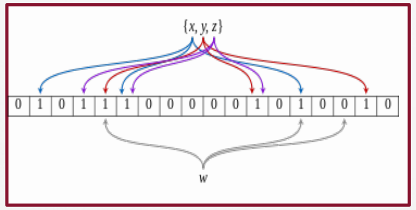
[HuangEvansKatz12]
Garbled Circuit base PSI

1985 1990 1995 2000 2005 2010 2015 2020

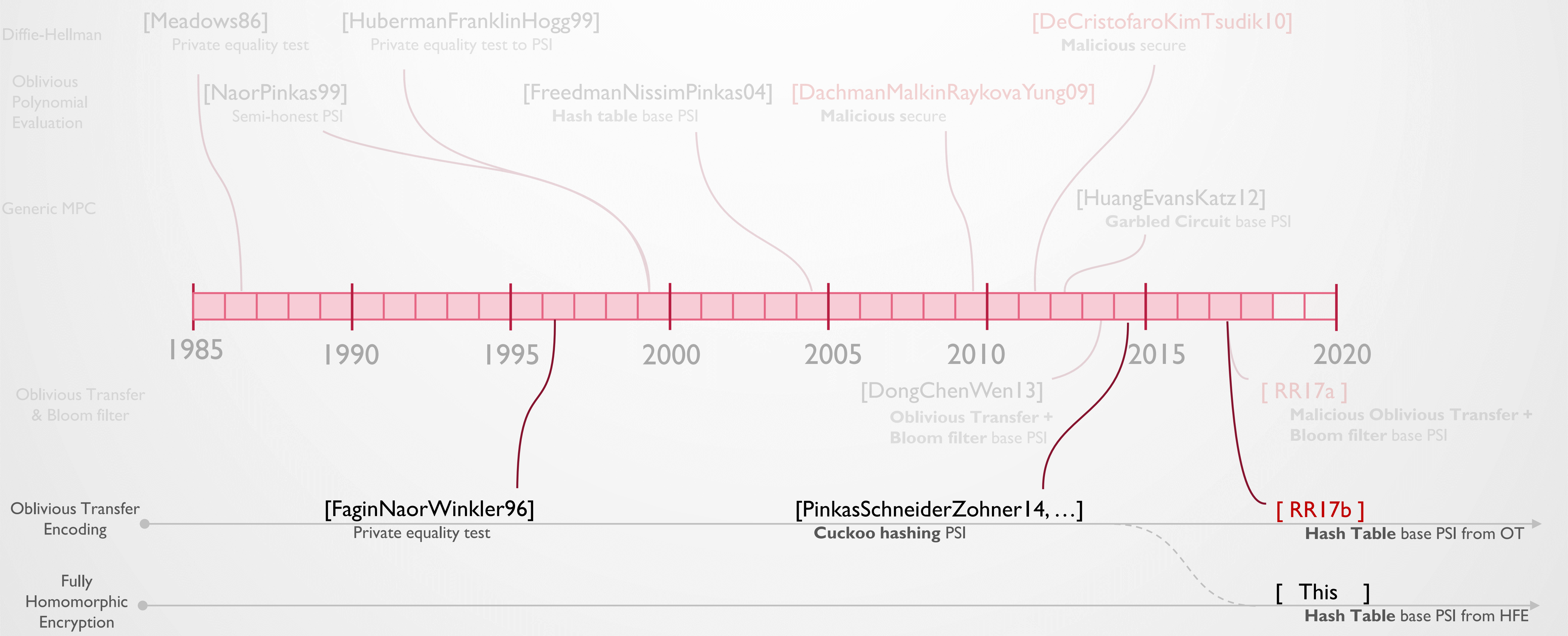
Oblivious Transfer & Bloom filter

[DongChenWen13]
Oblivious Transfer + Bloom filter base PSI

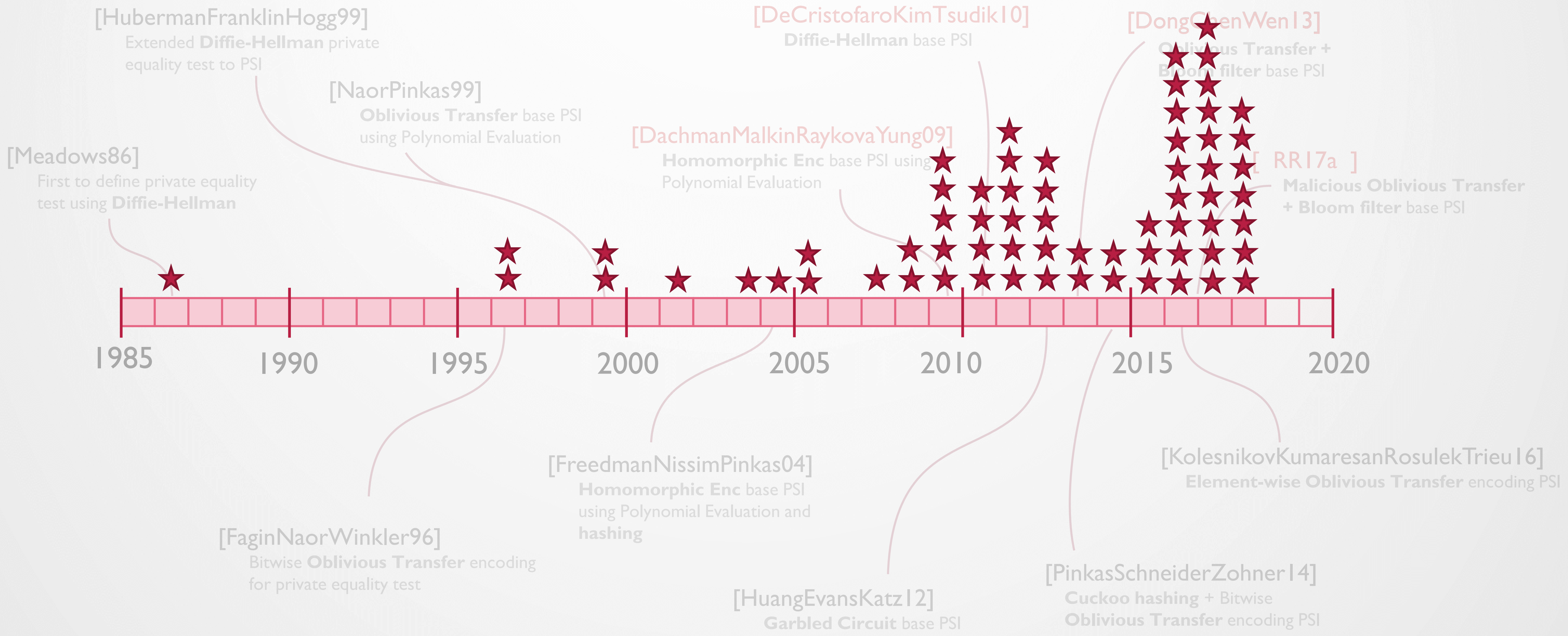
[RR17a]
Malicious Oblivious Transfer + Bloom filter base PSI



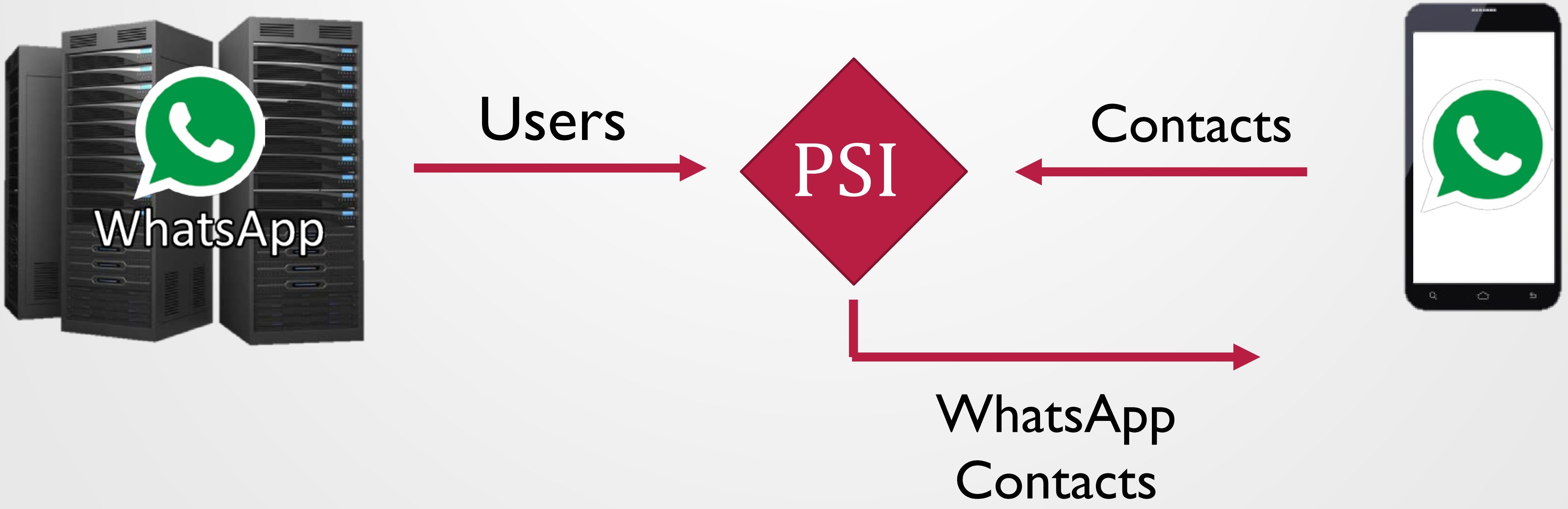
A *Sampling* of PSI Over the Decades



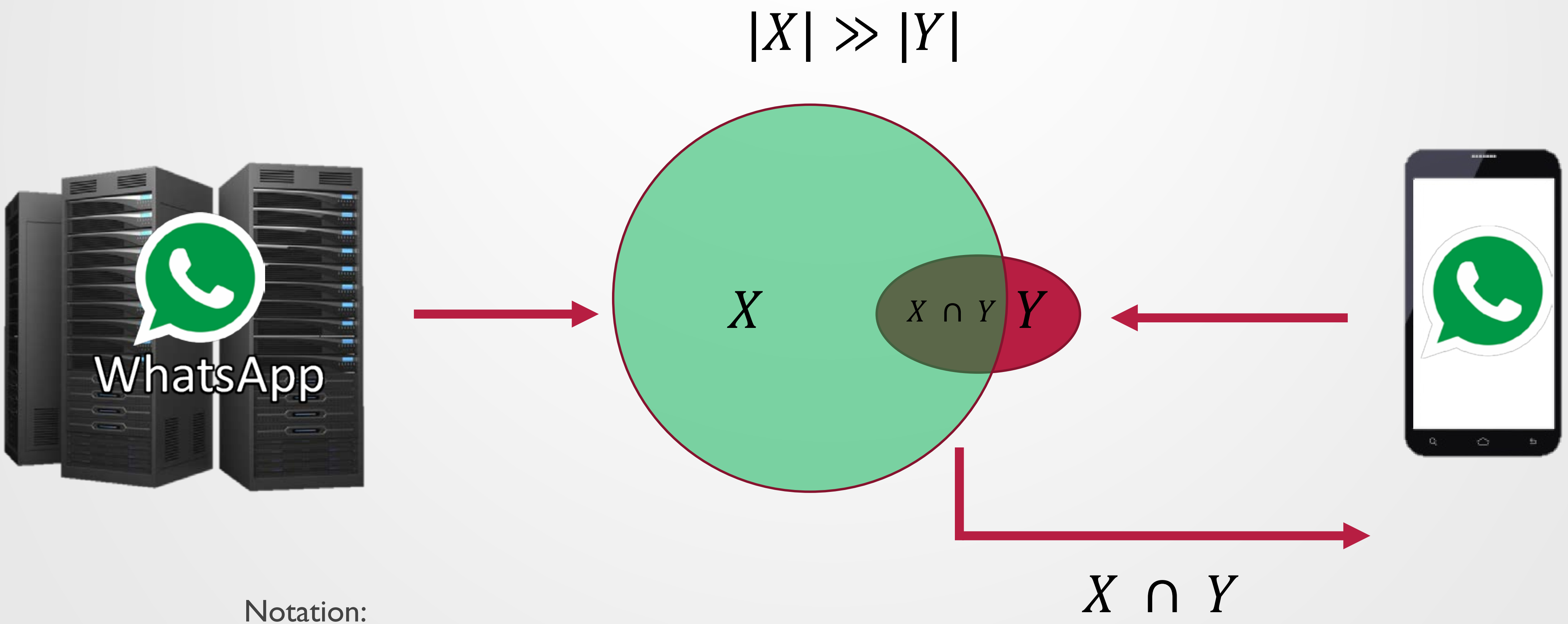
A *Sampling* of PSI Over the Decades



App: Contact discovery



App: Contact discovery

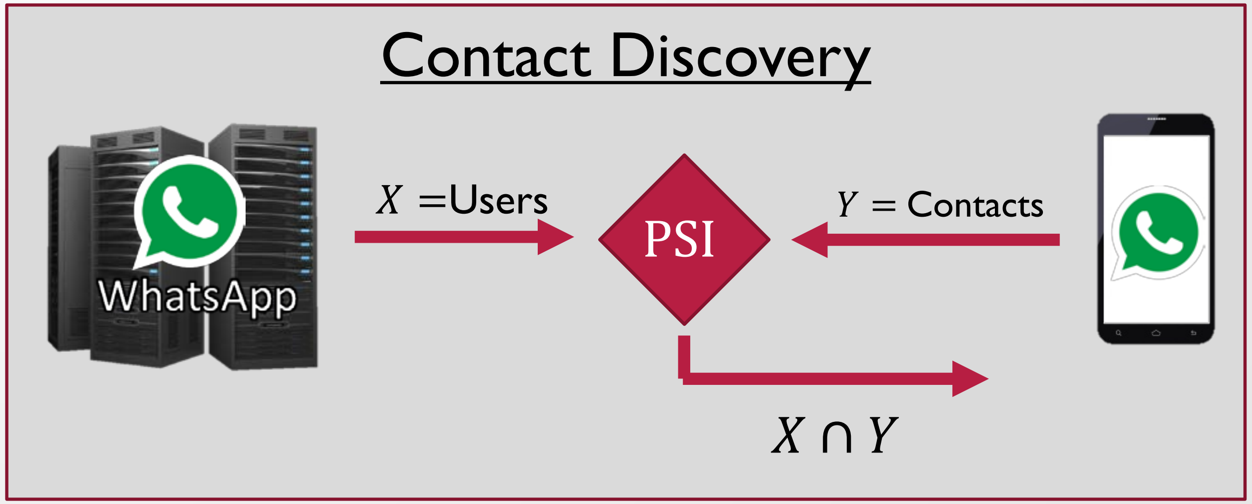


Notation:

- $N = |X|$
- $n = |Y|$

Shortcomings of Prior Work

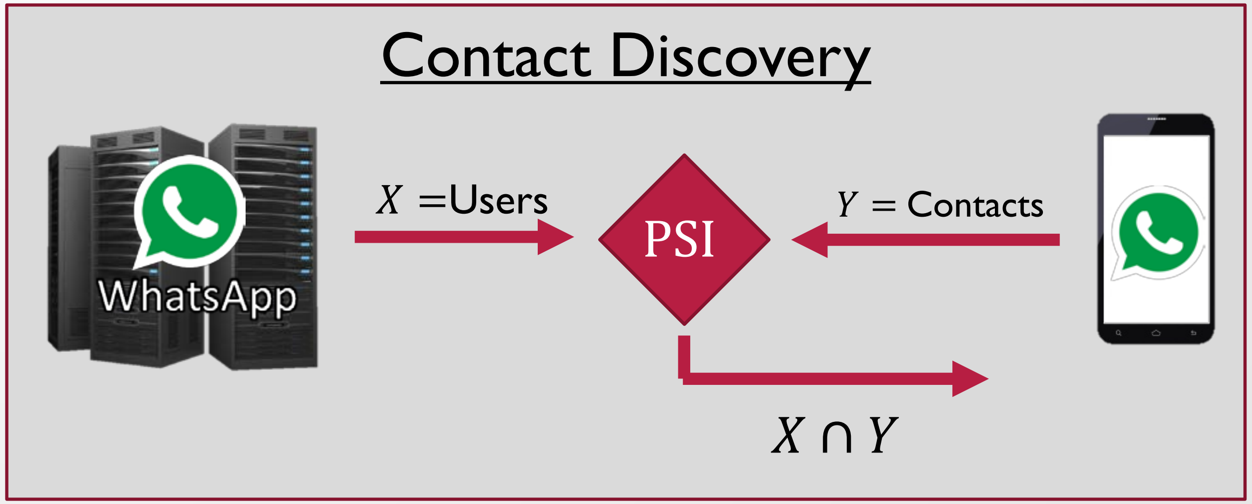
- Communication linear in both sets $O(|X| + |Y|)$
  - What about $|X| \gg |Y|$?
  - Insecure solution:
    - Send small set to other party
    - Comm. = $O(\min(|X|, |Y|))$
  - Can we match this?
    - Almost…



*Some prior works achieve sublinear communication for **related** problems.

Shortcomings of Prior Work

- Communication linear in both sets $O(|X| + |Y|)$
  - What about $|X| \gg |Y|$?
  - Insecure solution:
    - Send small set to other party
    - Comm. = $O(\min(|X|, |Y|))$
  - Can we match this?
    - Almost…
  - Computation = $O(|X|)$
  - Communication = $O(|Y| \log |X|)$

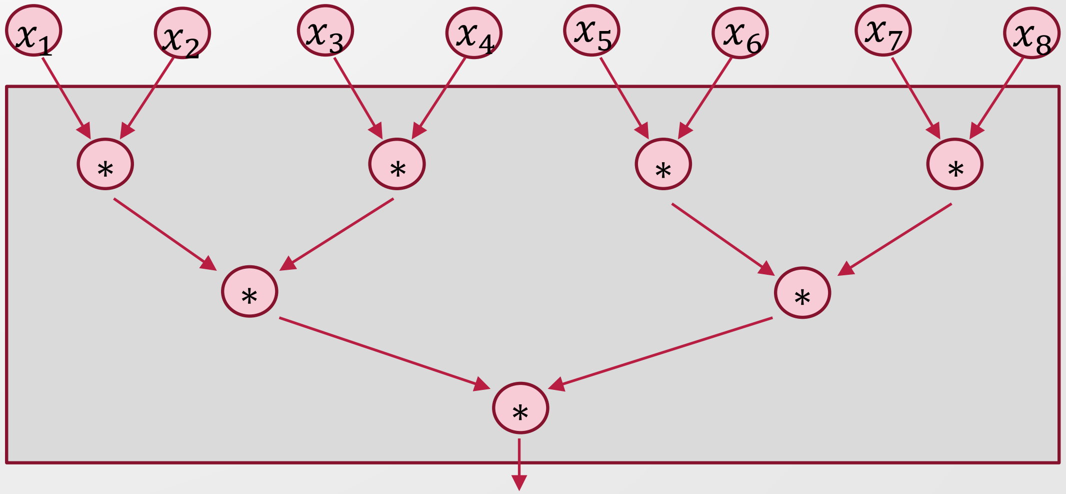


*Some prior works achieve sublinear communication for **related** problems.

Fully Homomorphic Encryption (FHE)

- Encryption technique that allows computation
  - $\mathrm{Enc}_k\big(f(x)\big) \equiv f\big(\mathrm{Enc}_k(x)\big)$
  - f can perform $+, -, *$
  - Addition and subtraction are very cheap.
  - Multiplication is very expensive.
    - Limited multiplication depth
    - E.g. $f(x) = \prod_{i=1}^{8} x_i$
    - Inefficient beyond depth ~6

$$
\begin{tikzcd}
x \arrow[r, "\mathrm{Enc}_k(x)"] \arrow[d, "f(x)"'] & \llbracket x \rrbracket \arrow[d, "f(\llbracket x \rrbracket)"] \\
y \arrow[r, "\mathrm{Enc}_k(y)"] & \llbracket y \rrbracket
\end{tikzcd}
$$



Equality Test from FHE

[ChenLaineRindal17]

- Want to test if $y = x$

$$[\![y]\!] \coloneqq \mathrm{Enc}_k(y)$$

$$[\![z]\!] \coloneqq [\![y]\!] - x \qquad \xrightarrow{[\![z]\!]}$$

$$z = 0 \Leftrightarrow y = x$$

- Issue: Receiver can recover $x = y - z$!
  - Need to randomize z

Equality Test from FHE

[ChenLaineRindal17]

- Want to test if $y = x$

$[\![y]\!] \coloneqq \mathrm{Enc}_k(y)$

Sample $r \leftarrow \mathbb{Z}_p^*$

$[\![z]\!]$

$[\![z]\!] \coloneqq ([\![y]\!] - x)r$

$z = 0 \Leftrightarrow y = x$

- Issue: Receiver can recover $x = y - z$!
  - Need to randomize z
  - Elements are in the prime field $\mathbb{Z}_p = \{0,1, \dots, p-1\}$
  - For a random $r \in \mathbb{Z}_p^* = \{1, \dots, p-1\}$
    - xr is a random elements in $\mathbb{Z}_p^*$, given non-zero x

Membership from FHE

[ChenLaineRindal17]

- Want to test if $y \in X$

$[\![y]\!] \coloneqq \mathrm{Enc}_k(y)$

Sample $r \leftarrow \mathbb{Z}_p^*$

$$[\![z]\!] \coloneqq r \prod_{x \in X} ([\![y]\!] - x)$$

$[\![z]\!]$

$z = 0 \Leftrightarrow y \in X$

- Issue: Depth of the computation is $\log N = \log |X|$
  - E.g. $N = 2^{28} \;\Rightarrow\; \text{depth} = 28 > 6$
- Observe the polynomial
  - Symmetric poly. ⇒ efficiently computable
- Need to compute y^N in low degree…

$$\begin{aligned}[\![z]\!] \coloneqq f(y) &= r \prod_{x \in X} (y - x) \\ &= a_N\, y^N + \cdots + a_2 y^2 + a_1 y + a_0\end{aligned}$$

Windowing: computing y^N in low depth

- Need to compute $[\![z]\!] := a_N\, y^N + \cdots + a_2 y^2 + a_1 y + a_0$
- Depth $\log N$ solution, send $[\![y]\!]$ and compute:
  - $[\![y^2]\!] = [\![y]\!][\![y]\!]$
  - $[\![y^4]\!] = [\![y^2]\!][\![y^2]\!]$
  - $\ldots$
- Depth 0 solution, send all $[\![y]\!], [\![y^2]\!], \ldots, [\![y^N]\!]$
  - $O(N)$ communication…
- Depth $\mathbf{\log\log N}$ soltuion, send $[\![y]\!], [\![y^2]\!], [\![y^4]\!], \ldots, \left[\!\left[y^{2^i}\right]\!\right], \ldots, \left[\!\left[y^{2^{\log N}}\right]\!\right]$
  - Compute all other powers in depth $\log\log N$
    - E.g. $[\![y^7]\!] = [\![y^4]\!][\![y^2]\!]\ [\![y]\!]$
    - E.g. $N = 2^{28} \Rightarrow \text{ depth} = 5$
  - $O(\log N)$ communication.

Membership from FHE

[ChenLaineRindal17]

- Want to test if $y \in X$:

$\{[\![y^i]\!] \mid i = 1,2,4,8,\dots,\log N\}$

Compute all $[\![y^i]\!]$

$[\![z]\!]$

$$[\![z]\!] \coloneqq r \prod_{x \in X}([\![y]\!] - x)$$
$$\coloneqq a_N\, y^N + \cdots + a_1 y + a_0$$

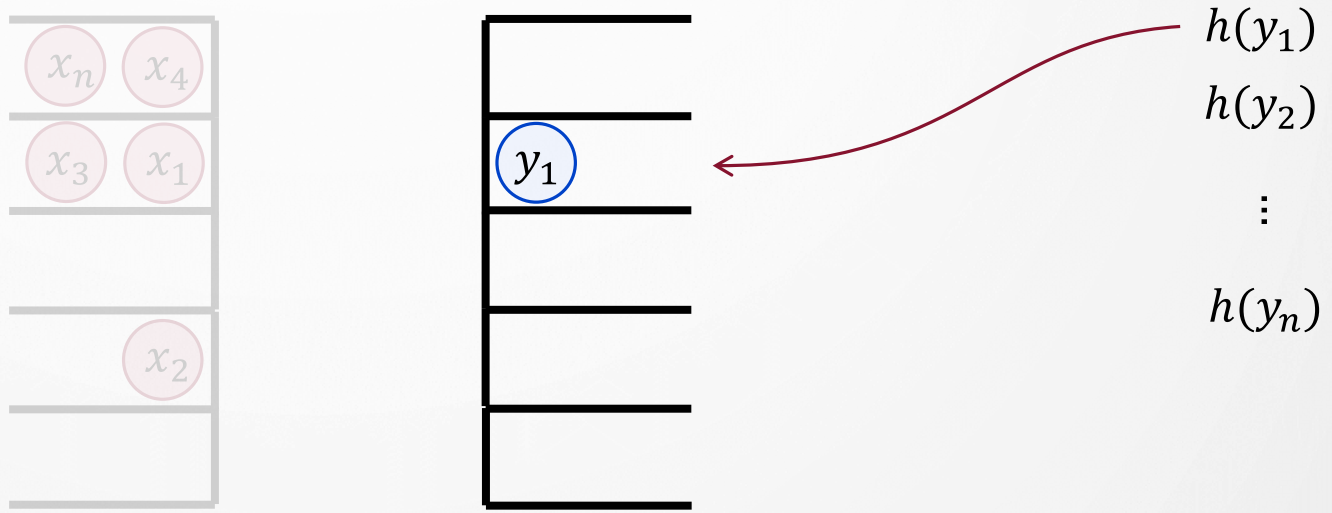
$z = 0 \Leftrightarrow y \in X$

- Performance,
  - Computation $= O(N)$
  - Depth $= O(\log\log N)$
  - Communication $= O(\log N)$
- Set intersection: For $y \in Y$, run set membership protocol
  - Require $O(nN)$ computation!!
  - Where $n = |Y|$,
  - e.g. $n = 1000$

Cuckoo Hashing

[PinkasScheiderZohner14, ChenLaineRindal17]

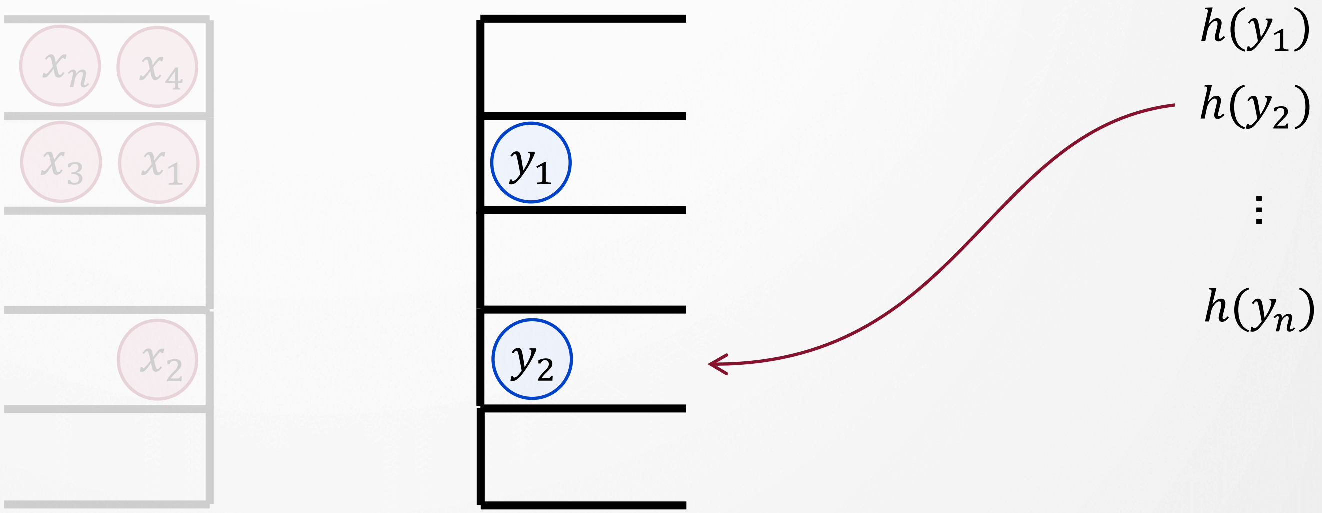
- Receiver performs Cuckoo hashing



Cuckoo Hashing

[PinkasScheiderZohner14, ChenLaineRindal17]

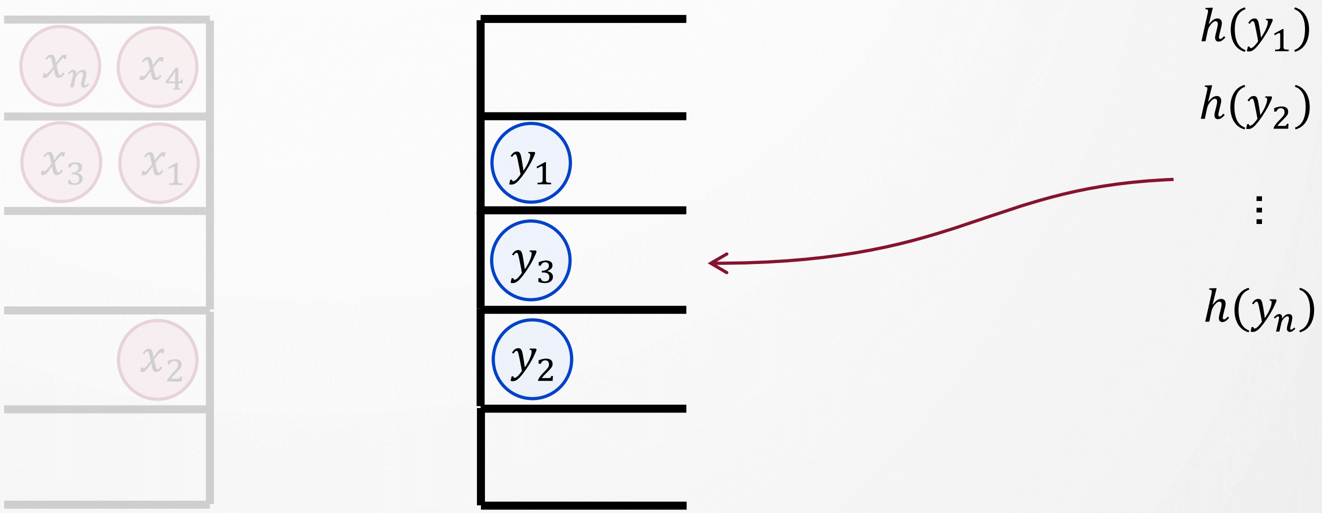
- Receiver performs Cuckoo hashing



Cuckoo Hashing

[PinkasScheiderZohner14, ChenLaineRindal17]

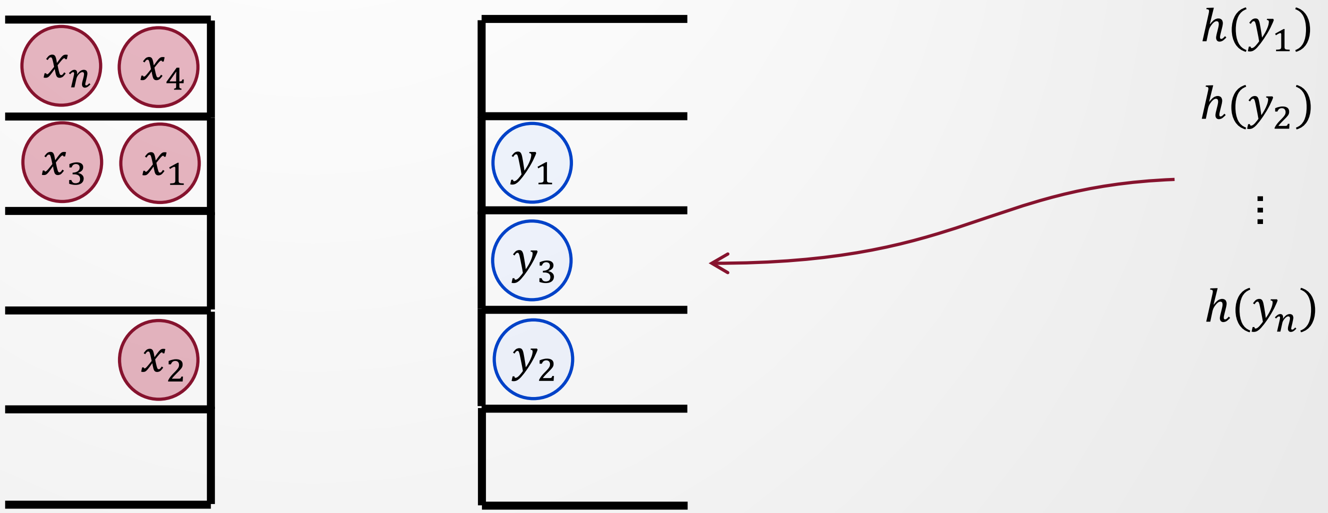
- Receiver performs Cuckoo hashing



Cuckoo Hashing

[PinkasScheiderZohner14,
ChenLaineRindal17]

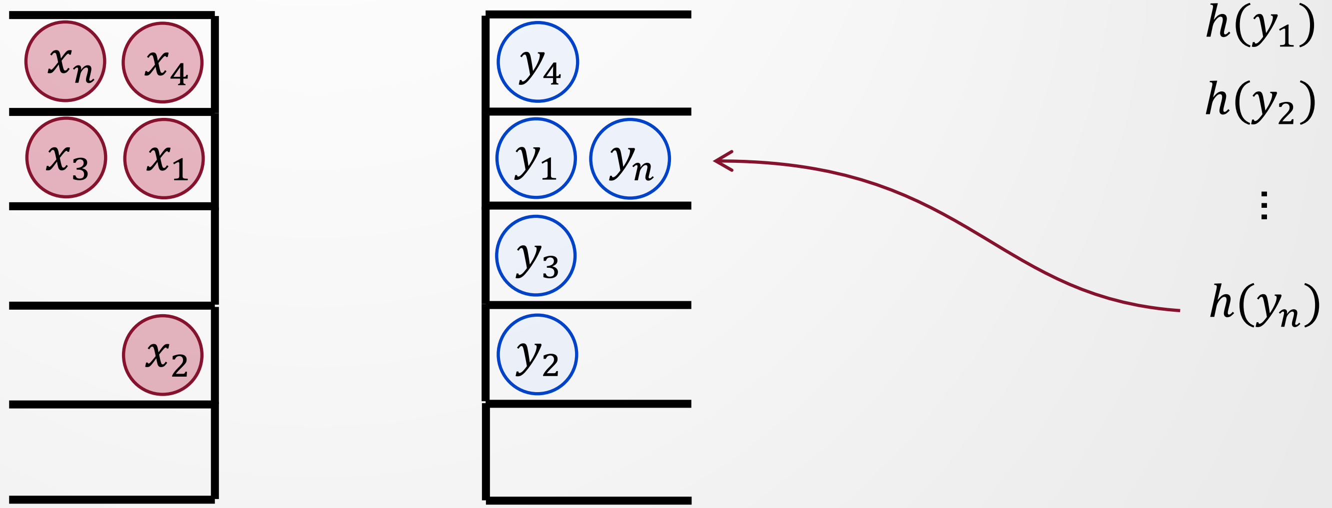
- Receiver performs Cuckoo hashing



Cuckoo Hashing

[PinkasScheiderZohner14, ChenLaineRindal17]

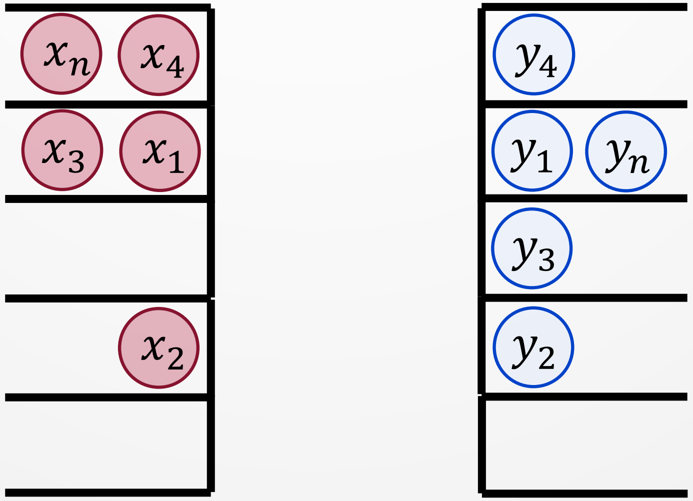
- Receiver performs Cuckoo hashing



Cuckoo Hashing

[PinkasScheiderZohner14, ChenLaineRindal17]

- Receiver performs Cuckoo hashing

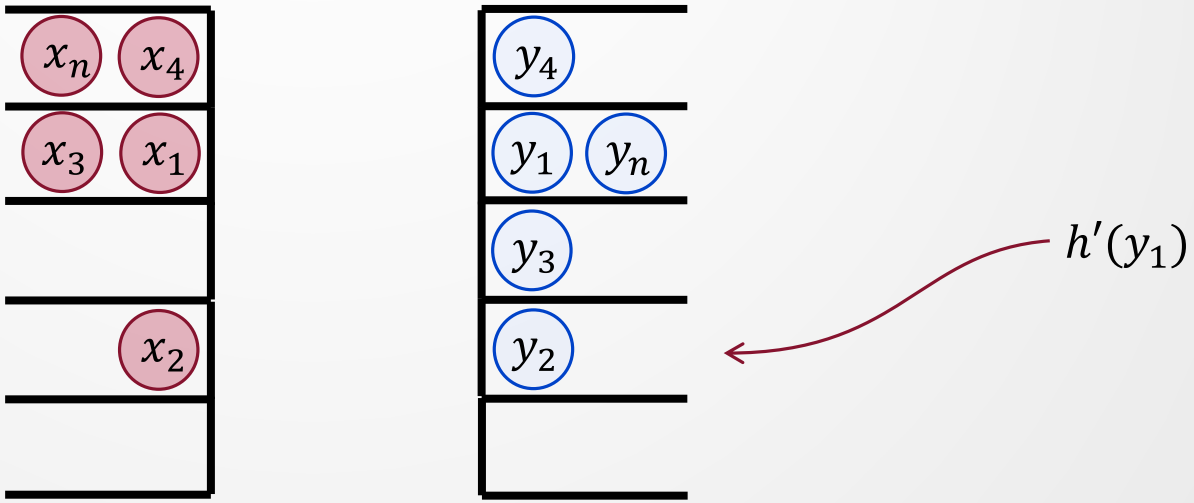


← Collision: $h(y_1) = h(y_n)$

Cuckoo Hashing

[PinkasScheiderZohner14, ChenLaineRindal17]

- Receiver performs Cuckoo hashing

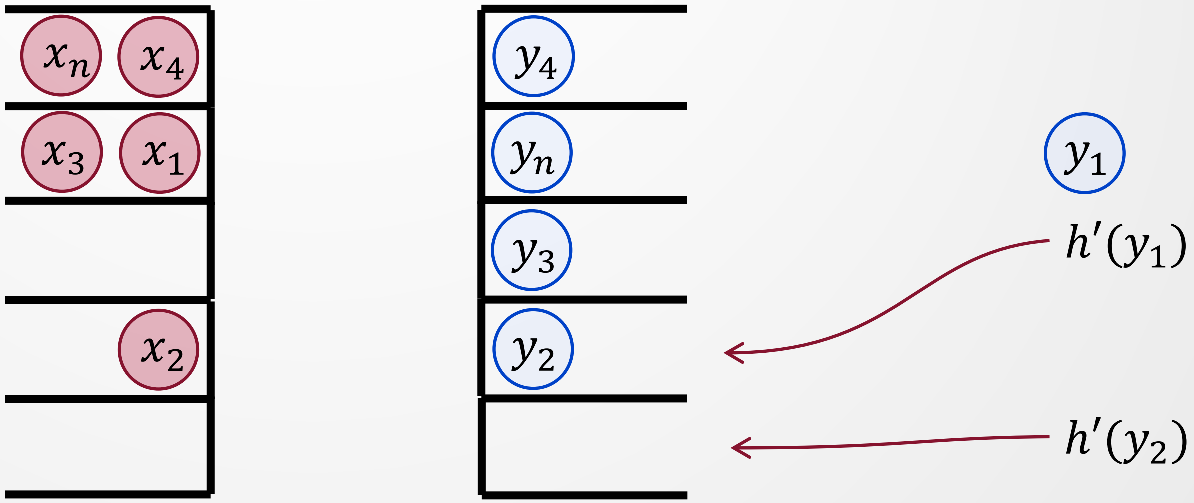


- Use two hash functions h, h'

Cuckoo Hashing

[PinkasScheiderZohner14,
ChenLaineRindal17]

- Receiver performs Cuckoo hashing

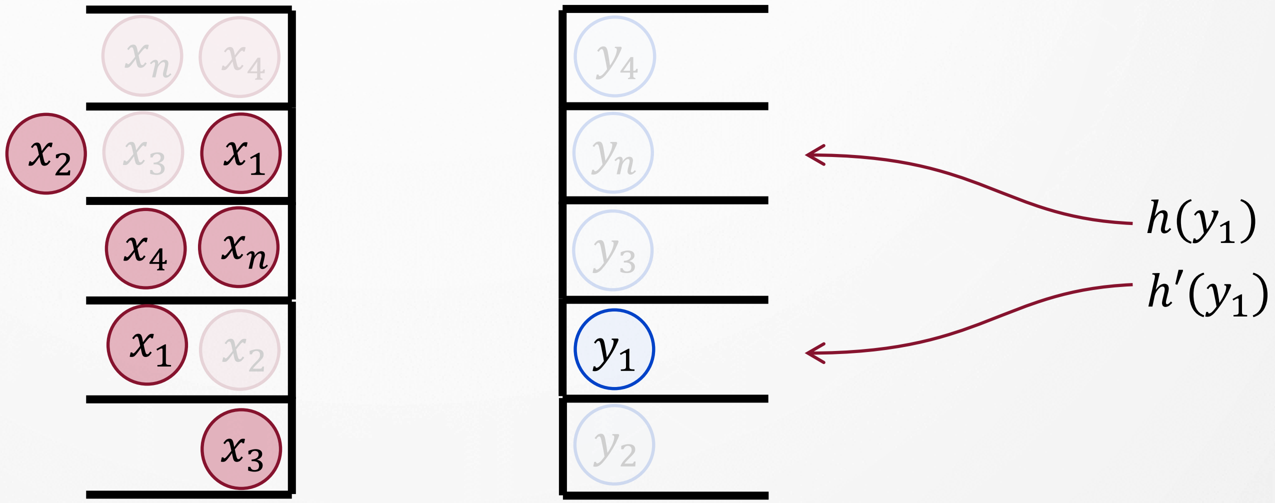


- Use two hash functions h, h'

Cuckoo Hashing

[PinkasScheiderZohner14, ChenLaineRindal17]

- Receiver performs Cuckoo hashing

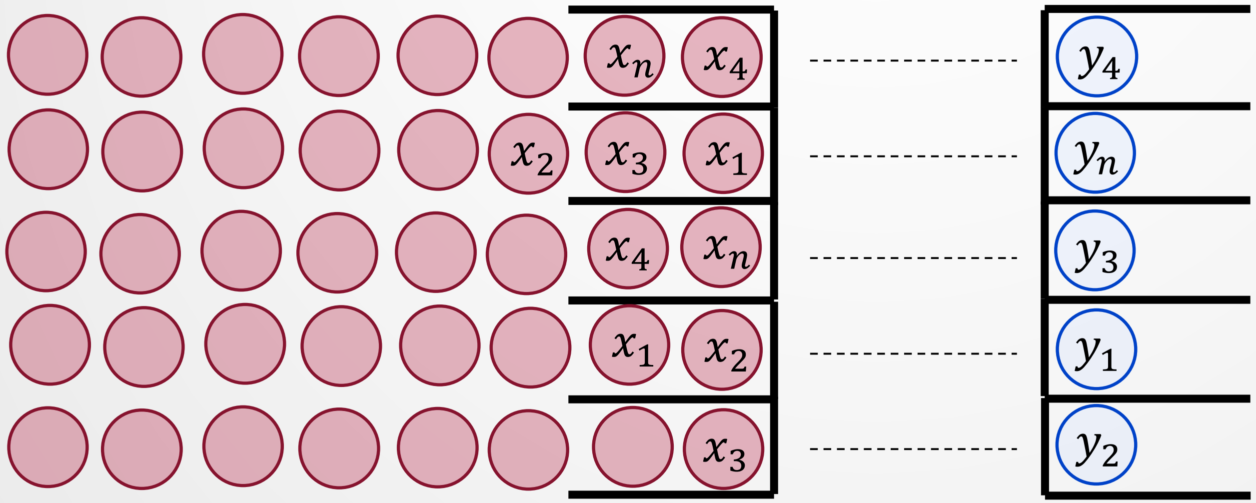


- Use two hash functions h, h'

Cuckoo Hashing

[PinkasScheiderZohner14, ChenLaineRindal17]

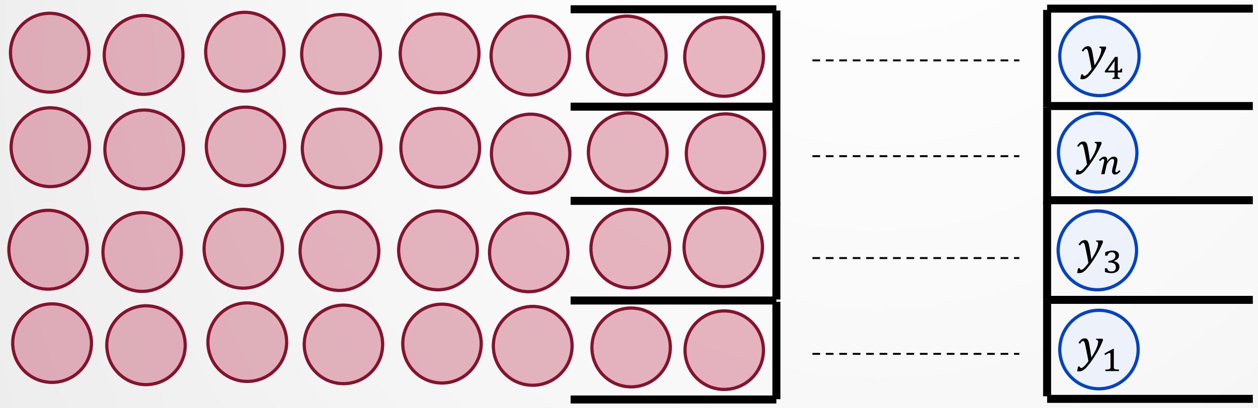
- Receiver performs Cuckoo hashing



- Use two hash functions h, h'
- For each bin, perform 1 membership test
  - When $N \gg n$, bin size $O(N/n)$
  - Overall complexity $O(N)$

Optimization: FHE Batching

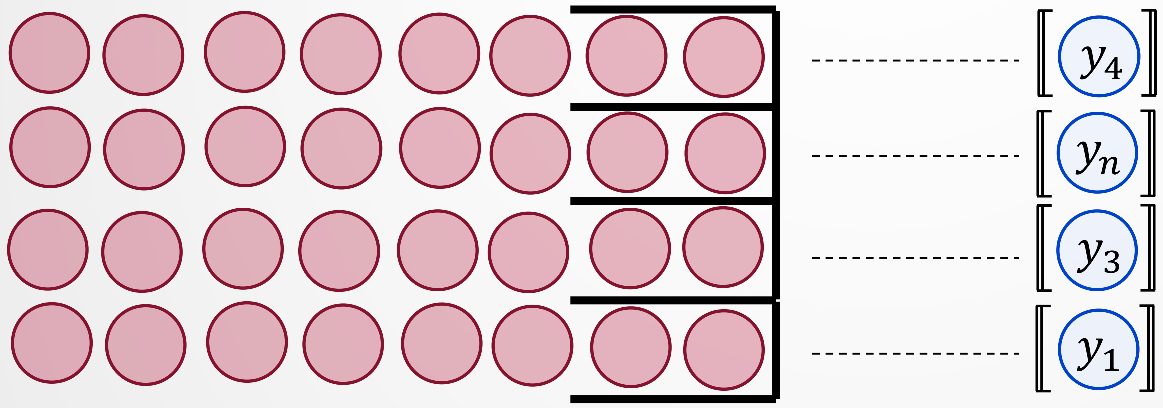
[ChenLaineRindal17]



- Fully homomorphic encryption naturally support "SIMD" type operations
  - A single FHE cipher-text/plaintext can be large…
  - Use Chinese Remainder Theorem (CRT) to pack several items into 1 cipher-text
    - E.g. 4096

Optimization: FHE Batching

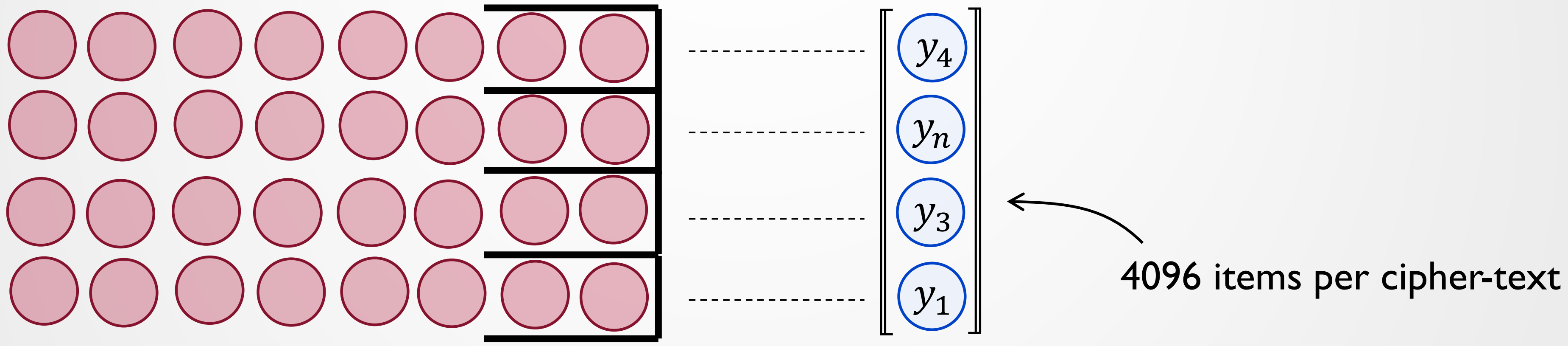
[ChenLaineRindal17]



- Fully homomorphic encryption naturally support "SIMD" type operations
  - A single FHE cipher-text/plaintext can be large…
  - Use Chinese Remainder Theorem (CRT) to pack several items into 1 cipher-text
    - E.g. 4096

Optimization: FHE Batching

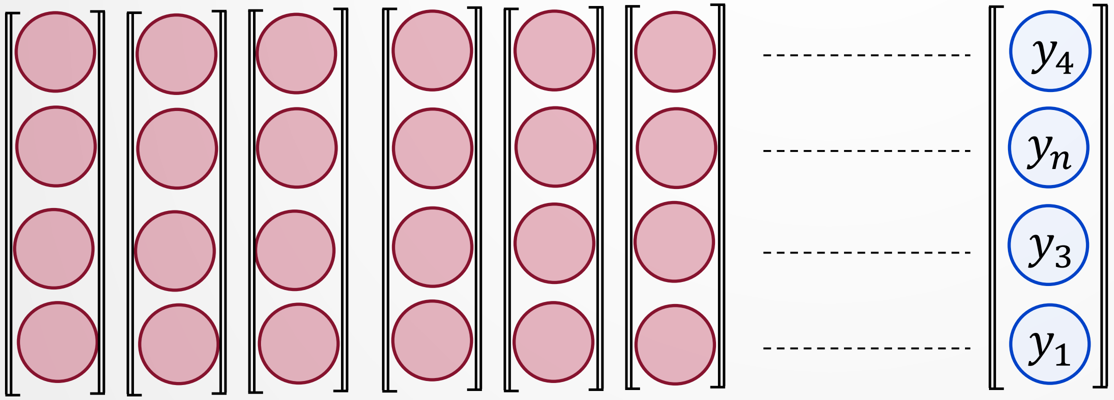
[ChenLaineRindal17]



- Fully homomorphic encryption naturally support "SIMD" type operations
  - A single FHE cipher-text/plaintext can be large…
  - Use Chinese Remainder Theorem (CRT) to pack several items into 1 cipher-text
    - E.g. 4096

Optimization: FHE Batching

[ChenLaineRindal17]



4096 items per cipher-text

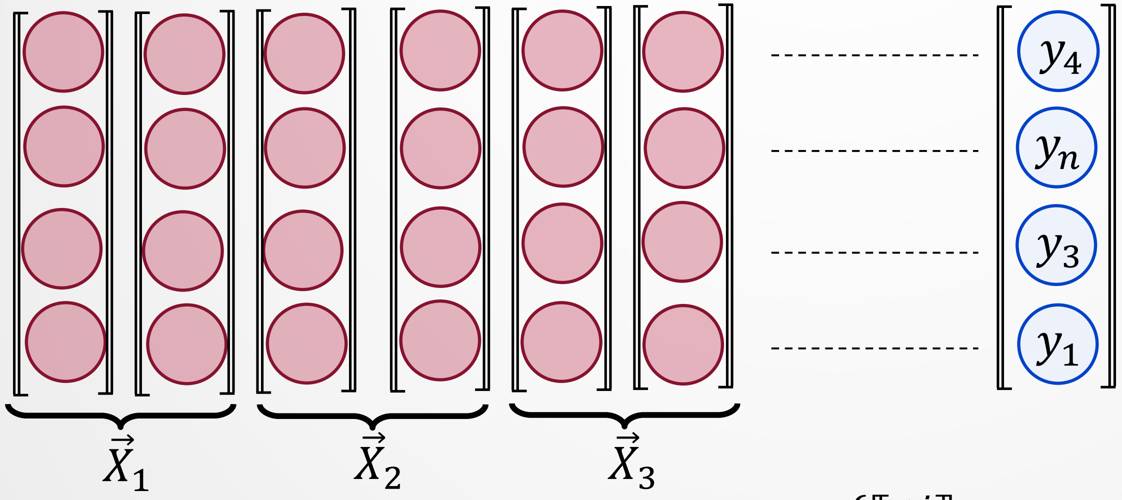
$\{[\![\vec{y}^{\,i}]\!] \mid i = 1,2,4,\ldots\}$

$[\![\vec{z}]\!] := \vec{r} \prod_{\vec{x} \in \vec{X}} ([\![\vec{y}]\!] - \vec{x})$ $\xrightarrow{[\![\vec{z}]\!]}$ $\vec{z}[i] = 0 \Leftrightarrow y[i] \in X$

- Fully homomorphic encryption naturally support "SIMD" type operations
  - A single FHE cipher-text/plaintext can be large…
  - Use Chinese Remainder Theorem (CRT) to pack several items into 1 cipher-text
    - E.g. 4096

Optimization: Splitting

[ChenLaineRindal17]



$\{[\![\vec{y}^i]\!] \mid i = 1,2,4,\ldots\}$

For $i = 1, \ldots, s$:

$[\![\vec{z}_i]\!] = \vec{r} \prod_{\vec{x}_i \in \vec{X}_i} ([\![\vec{y}]\!] - \vec{x}_i)$

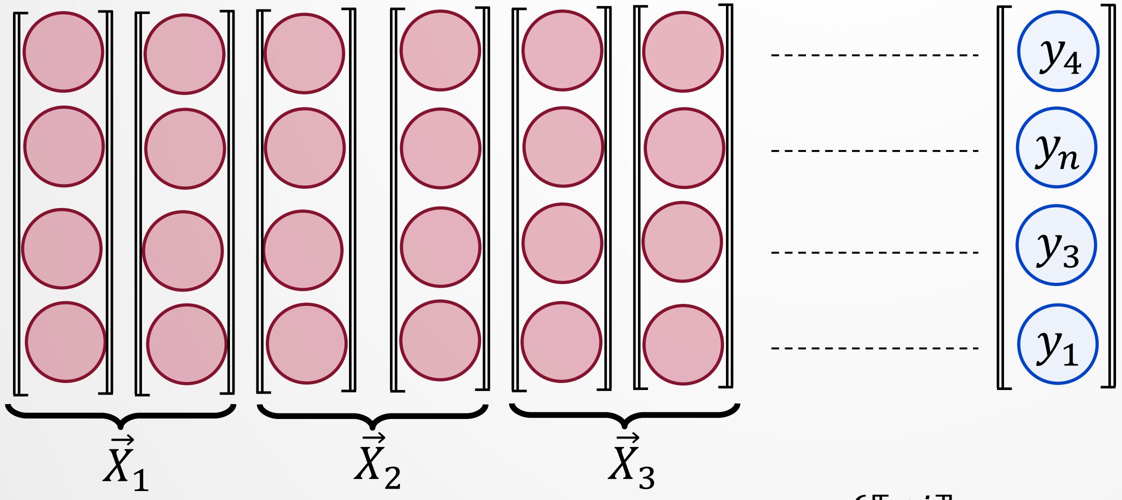
$[\![\vec{z}_1]\!], \ldots, [\![\vec{z}_s]\!]$

$\vec{z}_j[i] = 0 \Leftrightarrow \vec{y}[i] \in X$

- Observe that the communication is unbalanced.
- Partition $\vec{X}$ into s splits $\vec{X}_1, \ldots, \vec{X}_s$
  - Reduces depth to $\log\log\frac{N}{ns}$
  - Large impact in practice, e.g. depth =3 .

Final Protocol

[ChenLaineRindal17]



$\{[\![\vec{y}^{\,i}]\!] \mid i = 1,2,4,\dots\}$ $\longleftarrow$

For $i = 1, \dots, s$:

$$[\![\vec{z_i}]\!] \coloneqq \vec{r} \prod_{\vec{x}_i \in \vec{X_i}} ([\![\vec{y}]\!] - \vec{x_i})$$

$[\![\vec{z}_1]\!], \dots, [\![\vec{z}_s]\!]$ $\longrightarrow$

$$\vec{z_j}[i] = 0 \Leftrightarrow \vec{y}[i] \in X$$

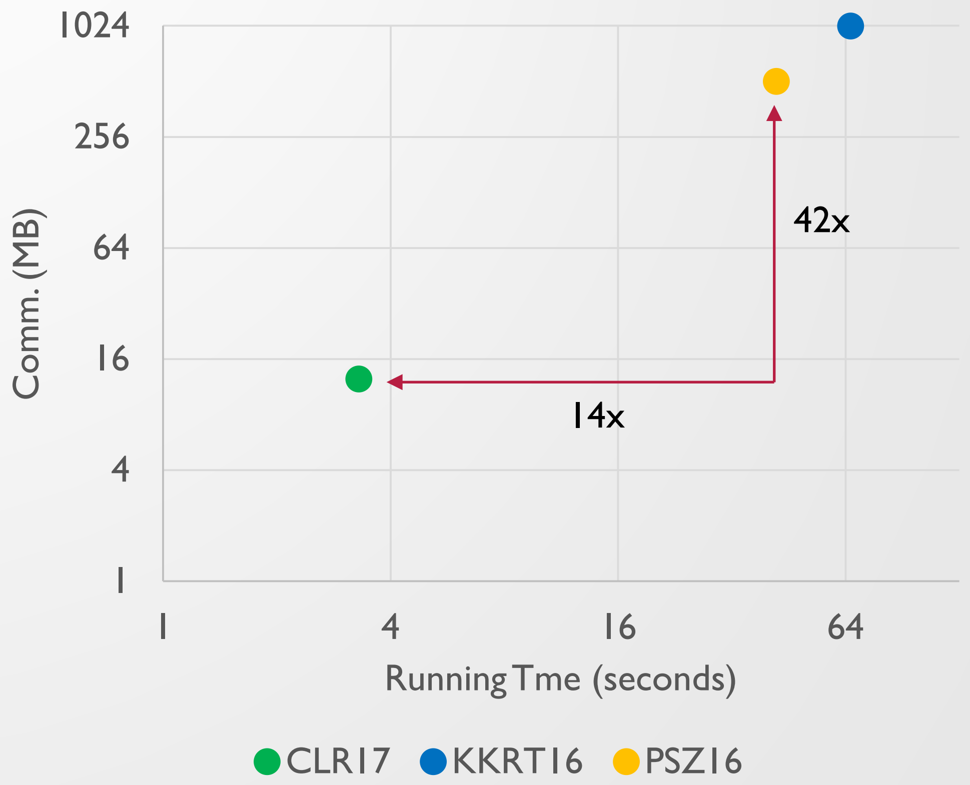
- Sender:
  - $O(N)$ Computation w/ quasi-constant depth
  - $O(n \ \log N)$ communication
  - Practical on server
- Receiver:
  - $O(n \log N)$ Encryptions/Decryptions
  - $O(n \ \log N)$ communication
  - Practical on cellphone

Performance

- CLR17 (with unpublished updates to code)
  - Optimized for unequal set sizes
  - Semi-honest hash table technique similar to [KKRT16]
  - Uses Fully Homomorphic Encryption
  - Very low communication when receiver's set is much smaller then sender's
  - Communication = $O(n \log N)$ bits
    - Previous approaches required $O(N + n)$ bits

(online time)

$n = 5{,}000$ vs $N = 16{,}777{,}216$



The End

Microsoft Research

Hao Chen
Kim Laine

Peter Rindal

Oregon State University OSU