



# Privacy Preserving Set Intersection

Using Acceptably Inaccurate Probabilistic Data Structures

Dinusha Vatsalan

04 July 2017

Work done in collaboration with Prof. Peter Chrsiten from the Australian National University, Prof. Vassilios S. Verykios and Dr. Dimitrios Karapiperis from Hellenic Open University, Greece

[www.data61.csiro.au](http://www.data61.csiro.au)



# Outline



- **Privacy preserving set intersection (PPSI)**
- **Two categories of techniques**
  - Cryptographic methods
  - Probabilistic methods
- **Probabilistic data structures**
  - Bloom filters, counting Bloom filters, count-min sketches, and more
- **PPSI using Bloom filters**
- **PPSI using counting Bloom filters**
- **PPSI using count-min sketches**
- **Experimental evaluation**
- **Outlook to research directions**

# Privacy Preserving Set Intersection (PPSI)



- Computing set intersection in multi-sets of an arbitrary large number of distinct elements privately and efficiently for privacy preserving data mining
- Example applications:
  - **Health surveillance system** – monitoring drug consumption at pharmacies and hospitals located at different places to alert when drug usage exceeds a threshold
  - **Crime detection or national security application** – monitoring the number of times certain online services are accessed
  - **Transport services** – gathering statistics about movements and commuting paths to improve services and predict future trends
- In all these applications large sets held by different parties need to be intersected to identify common elements in the sets along with their counts of occurrences; however privacy issues preclude sharing individual data for set intersection

# The Threat Model



- **Privacy preserving context**
  - Parties should not be able to learn other parties' data
  - The consumer of the PPSI protocol (for example, a researcher or organization) should not learn individual parties' data as well as non-frequent/non-common elements
  - Eavesdropper should not be able to learn any parties' data
- **Honest-but-curious adversary model**
  - Parties follow the protocol, but are curious to learn about other parties' data
- **Collusion is possible**
  - Two or more parties collude with the aim to learn other parties' data

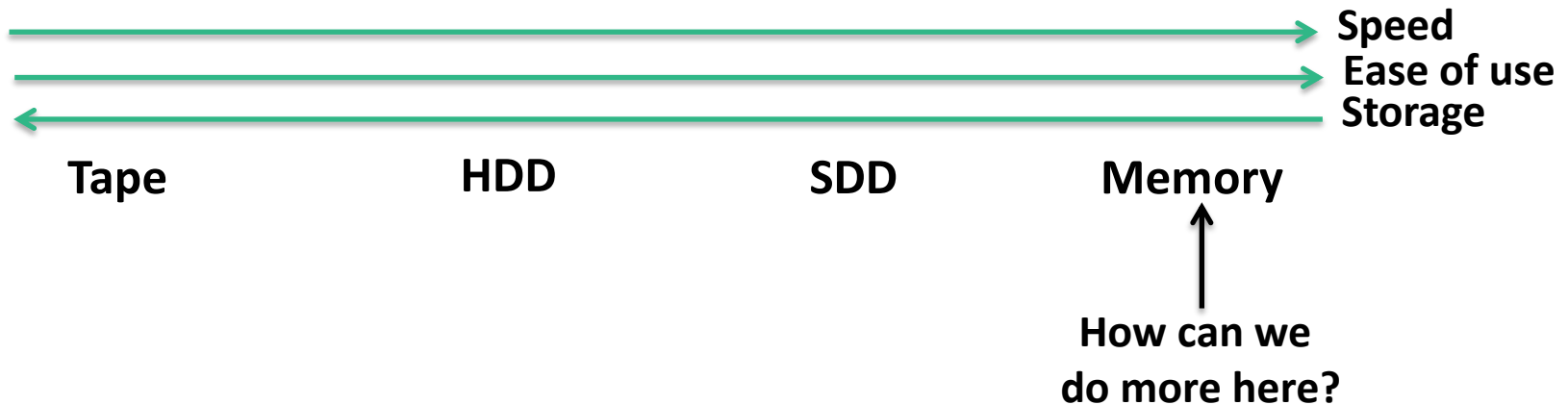
# Two Categories of Techniques



- **Cryptographic methods**
  - Example: Secure scalar product, asymmetric cryptography
  - Highly accurate
  - Provably secure
  - But, computationally expensive
- **Probabilistic methods:**
  - Example: Bloom filters and variations, sketches, and cuckoo filters are probabilistic data structures and noise addition, differential privacy, and k-anonymity are perturbation techniques
  - Highly efficient for processing, storing, and computation
  - Acceptable inaccurate – allows false positives
  - Controllable privacy – trade-off between privacy and accuracy

# Probabilistic Data Structures

- **Motivation**



- **Used for set membership**

- Predictable level of inaccuracy
- Privacy preserving due to false positives

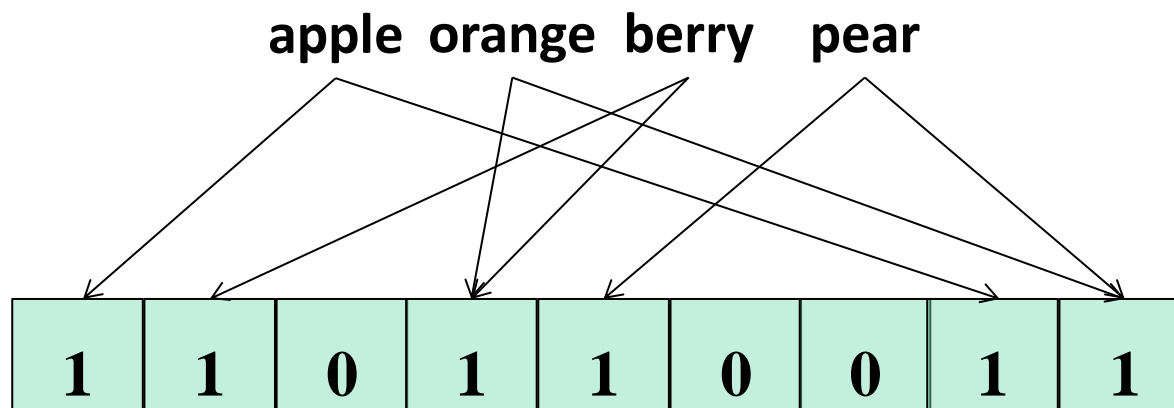
- **Data structures:**

- Bloom filters and variations (such as counting Bloom filters), Count-min sketches, HyperLogLog, and Cuckoo filters

**Probabilistic data structures**

# Bloom Filters

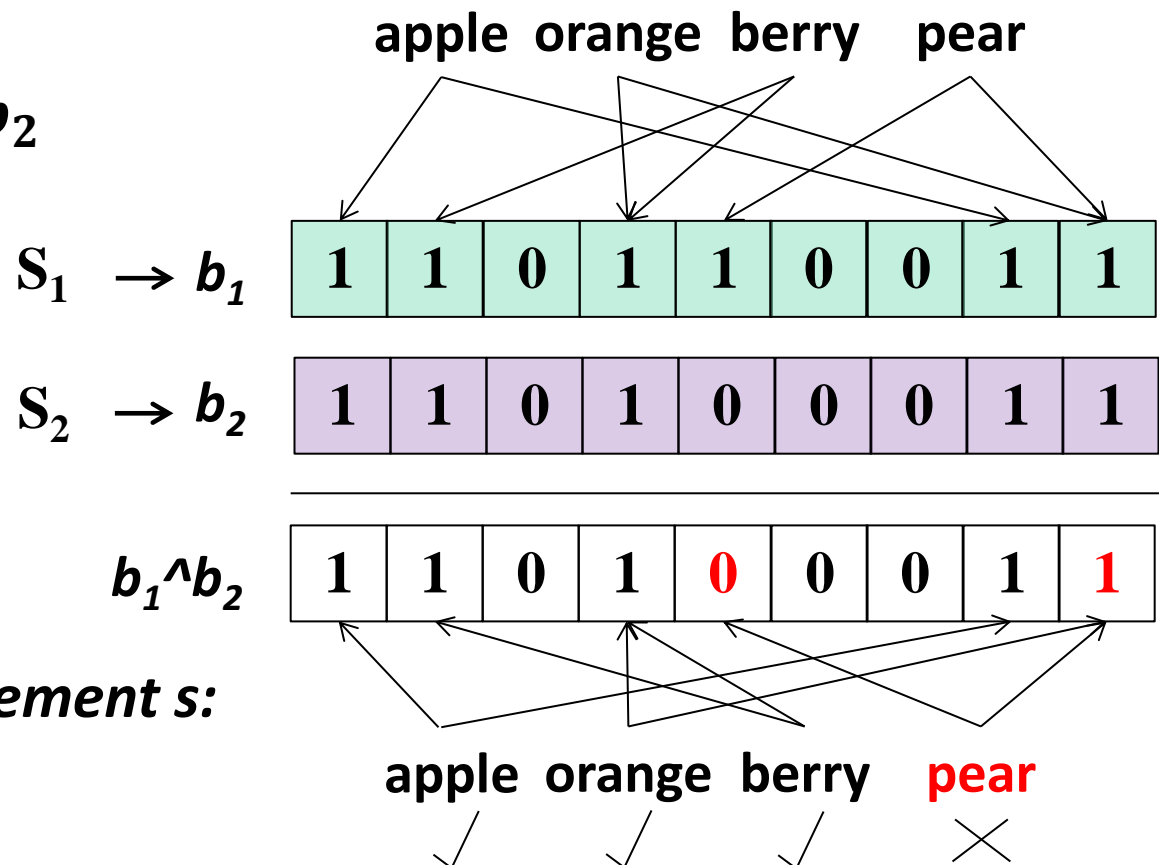
- Bloom filter is a bit vector  $b$  initially set to 0-bits
- $k$  independent hash functions  $h(.)$  are used to hash-map each element in a set  $S$  into a Bloom filter (BF) of length  $l$  bits by setting the corresponding bits to 1
  - $\forall_S \forall_{j=1}^k b[h_j(s)] = 1$
- E.g. hash-mapping a set  $S = ['apple', 'orange', 'berry', 'pear']$  into a BF of  $l=9$  bits using  $k=2$  hash functions:



# PPSI using Bloom Filters

- Assume two sets  $S_1 = [\text{'apple'}, \text{'orange'}, \text{'berry'}, \text{'pear'}]$  and  $S_2 = [\text{'apple'}, \text{'orange'}, \text{'berry'}]$  encoded into two BFs  $b_1$  and  $b_2$

- Intersection list  $b_1 \cap b_2$*



- Test membership of element  $s$ :*

- $\bigvee_{j=1}^k h_j(s) == 1$



# PPSI using Bloom Filters (contd..)

- Assume  $p$  multiple (more than two) sets from  $p$  parties
- The set intersection can be distributed among  $p$  parties
  - Lower computational cost at each party ( $O(n.l/p)$ )
  - Lower information gain
  - Same communication cost ( $O(l.n.p)$ )

	P <sub>1</sub>			P <sub>2</sub>			P <sub>3</sub>		
$b_1$	1	1	0	1	1	0	0	1	1
$b_2$	1	1	0	1	1	0	0	0	1
$b_3$	1	0	0	1	1	1	0	0	1
$b_1 \wedge b_2 \wedge b_3$	1	0	0	1	1	0	0	0	1

# PPSI using Bloom Filters (contd..)



- Bloom filters are simple and efficient
- False positive probability for  $n$  elements:
  - $f = (1 - e^{\{-\frac{kn}{l}\}})^k$
  - Controllable by tuning the Bloom filter parameters  $k$  and  $l$
  - The larger the  $f$  the better the privacy gain
- Weaknesses of Bloom filters:
  - Do not store counts of occurrence
  - Static – no deletion or modification is allowed
- Variations of Bloom filters:
  - Counting Bloom filters
  - Spectral Bloom filters
  - Deletable Bloom filters

# PPSI using counting Bloom Filters

- A counting Bloom filter is an integer array of length  $l$  containing counts of values in each bit position  $\beta$ ,  $1 \leq \beta \leq l$  over  $p$  sets of elements

- $\forall_s \forall_{j=1}^k h_j(s) += 1$

- $c = \sum_{i=1}^p b_i$

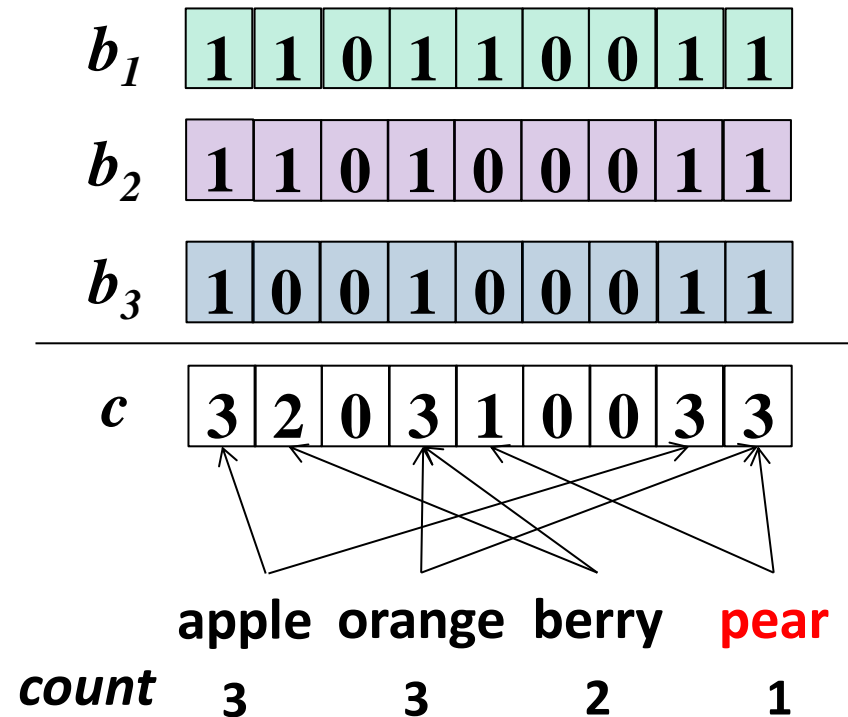
- PPSI of multi-sets:

- Given  $c$  of multi-sets

- Set membership of an element  $s$ :

- $- iff(\forall_{j=1}^k h_j(s) > 0)$

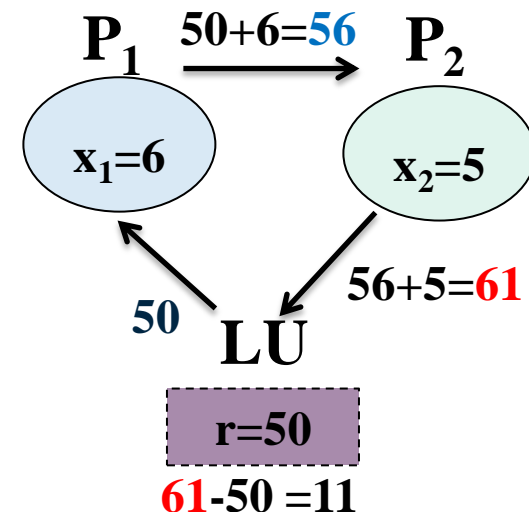
- $count(s) = \min_{j=1}^k (h_j(s))$



# PPSI using counting Bloom Filters (contd..)

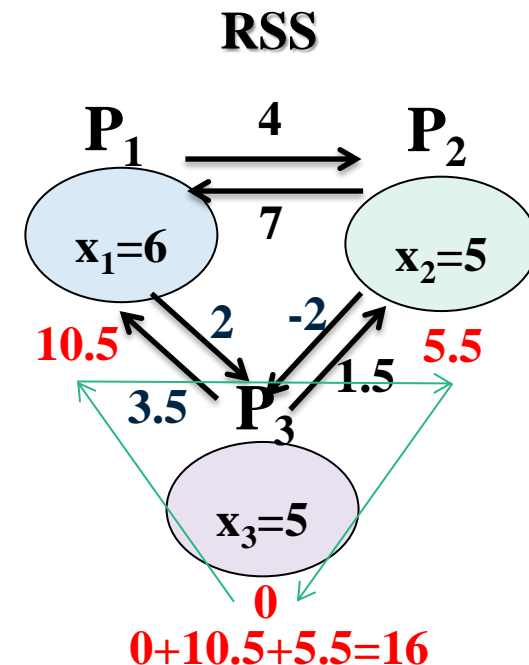
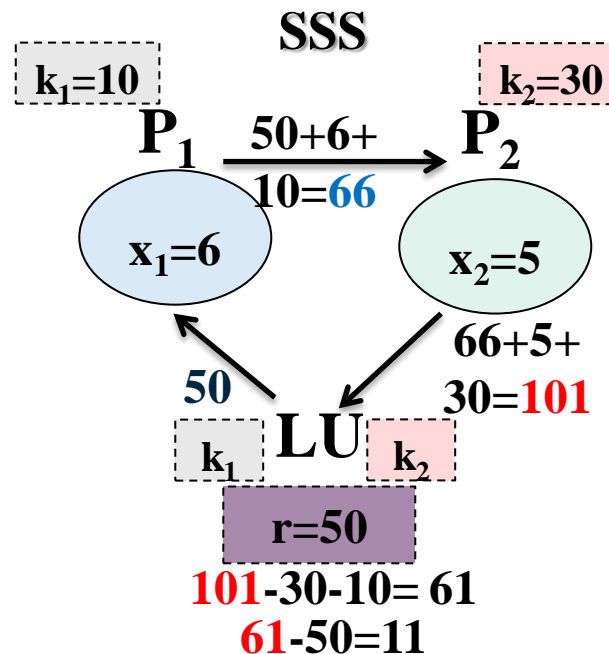
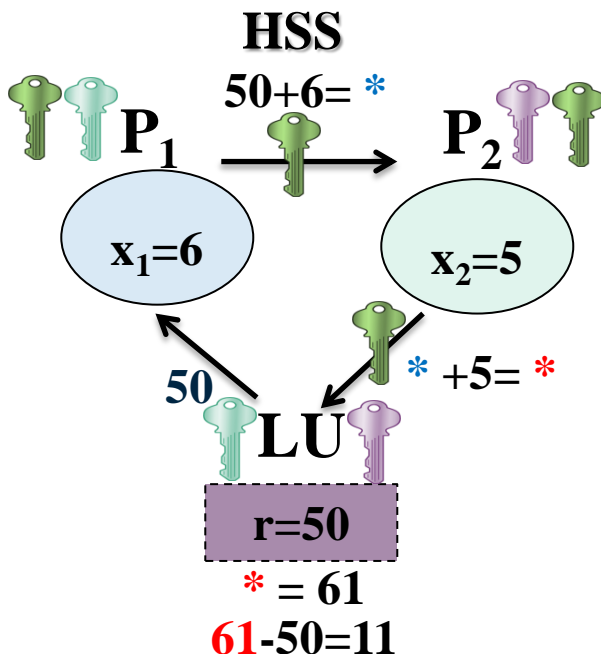


- Need to choose counters large enough to avoid overflow
- Poisson approximation suggests 4 bits/counter
- Storage becomes expensive with larger frequency
  - Assume average frequency count is  $d$
  - Every position in the counting Bloom filter requires  $2^d$  bits
  - The total memory consumption is  $l \times \lceil \log_2(d) \rceil$
- Secure summation
  - Privacy issues due to collusion
    - Two or more parties collude to infer the counts of a non-colluding party
  - Collusion resistant protocols



# Collusion Resistant Secure Summation

- Basic secure summation protocol is susceptible to collusion risk
- Extended secure summation protocols:
  - Homomorphic-based secure summation (HSS)
  - Salting-based secure summation (SSS)
  - Random sharing-based secure summation (RSS)



# Count-min Sketches

- An array of  $D$  rows and  $W$  cells in each row, initialized to 0
- $D$  independent hash functions  $h(.)$  are used to hash-map each element in a set  $S$  into a Count-min sketch (CS) by incrementing the corresponding bit in each of the  $D$  rows by 1
  - $\forall_s \forall_{j=1}^D c[j, h_j(s)] = 1$

$s_1$

0	0	1	0
1	0	0	0
1	0	0	0

$s_2$

1	0	1	0
2	0	0	0
1	0	1	0

$s_1$

1	0	2	0
3	0	0	0
2	0	1	0

- $count(s) = \min_{j=1}^D (h_j(s))$

Example:  $count(s_1) = \min(2, 3, 2) = 2$

$s_1$

1	0	2	0
3	0	0	0
2	0	1	0

# Count-min Sketches (contd..)



- Each element is hashed by randomly chosen pairwise independent hash functions
  - $h_j(s) = [(a_j s + b_j) \bmod P] \bmod W,$
  - Where  $j = 1, \dots, D$ , and  $P$  is a large prime number
- For any  $s_1, s_2 \in S$ , the probability of collision of the result of the hash function  $h_j$  is
  - $Pr(h_1(s_1) == h_2(s_2)) \leq 1/w$
- Let  $C(s_i)$  be the count estimate of  $s_i$  and  $C'(s_i)$  the real estimate
  - $\|S\|_1$  is the L1 norm of  $\sum_{i=1}^n C'(s_i)$
  - In order to get an estimate that satisfies  $C(s_i) \leq C'(s_i) + \epsilon \|S\|_1$  ( $\epsilon > 1$  is acceptable error) with probability  $1 - \delta$ 
    - $D$  should be  $\lceil \ln(1/\delta) \rceil$  and  $W$  should be  $\lceil \ln(e/\epsilon) \rceil$

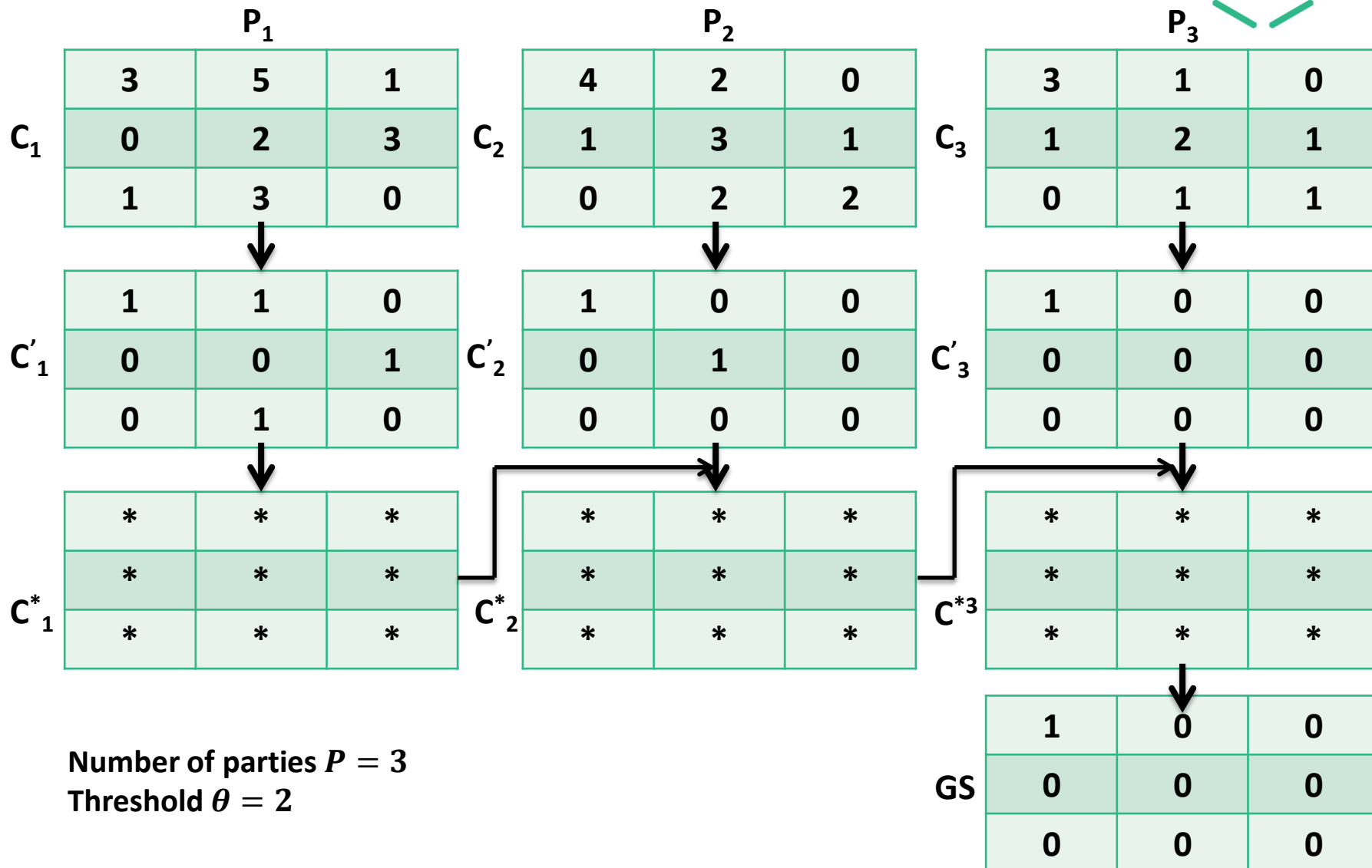
# PPSI using Count-min Sketches



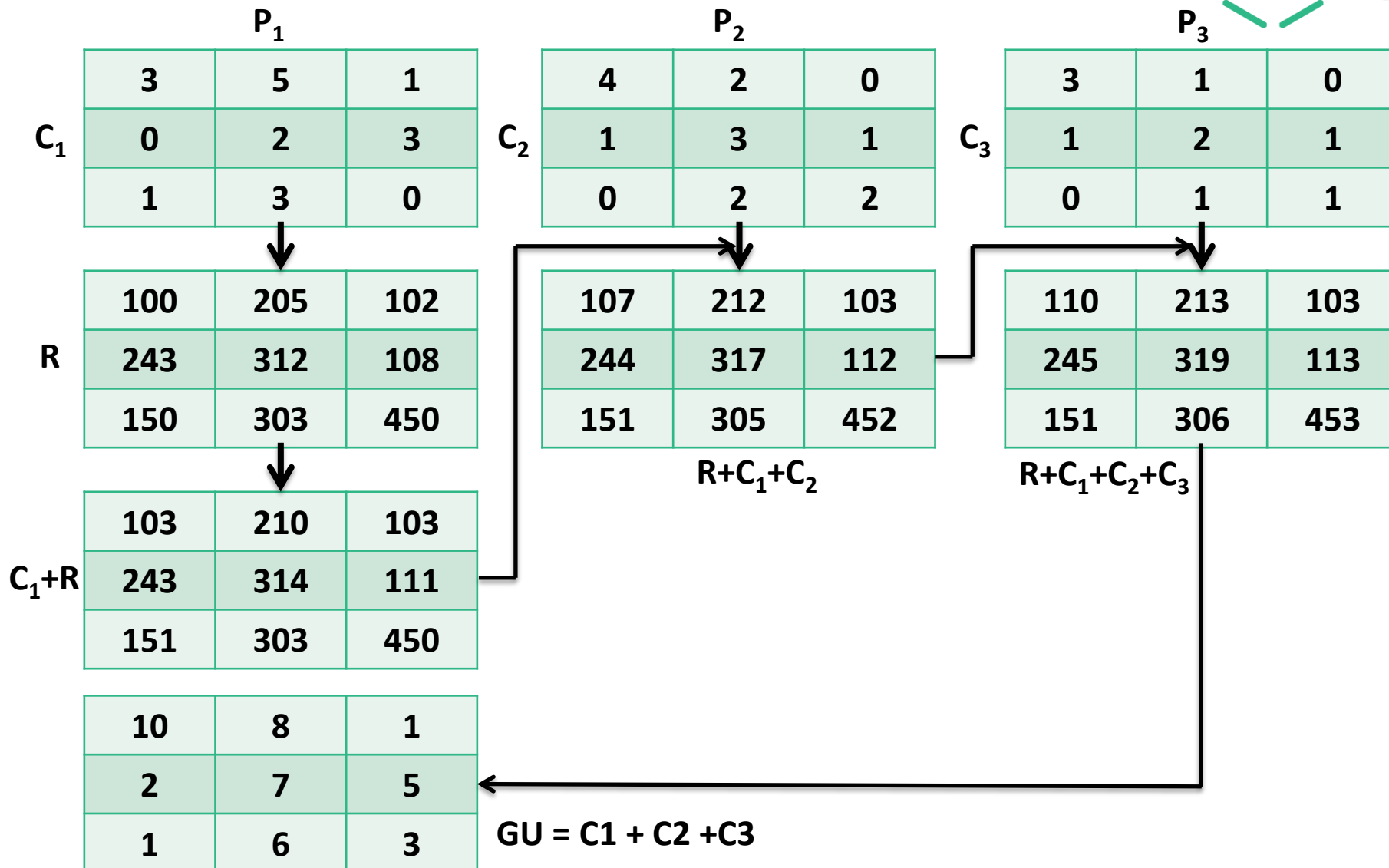
- **Creating local synopsis by each party using a count-min sketch**
  - Count-min sketches utilize space – sublinear with the number of elements of a set represented by it
- **Calculate a global synopsis that contains intersection of multi-sets**
  - Also the counts of occurrences
  - Linearity of sketches: sketch produced by adding cell-wise two or multiple sketches is the union of these sketches
- **Proposed two PPSI protocols using count-min sketches:**
  - Homomorphic-based
  - Perturbation-based



# Homomorphic-based



# Homomorphic-based (contd..)



# Homomorphic-based (contd..)

1	0	0
0	0	0
0	0	0

 $\times$ 

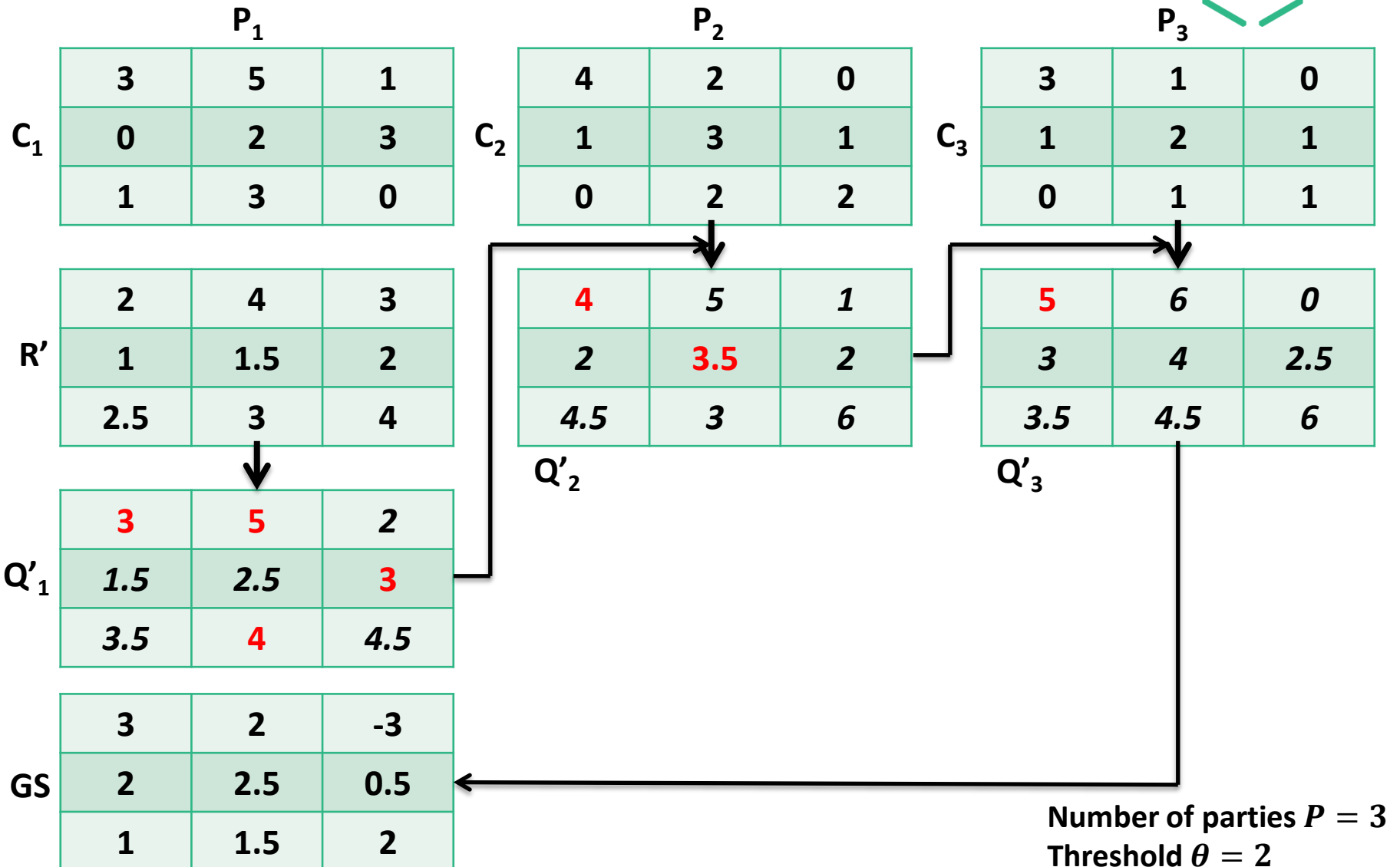
10	8	1
2	7	5
1	6	3

 $=$ 

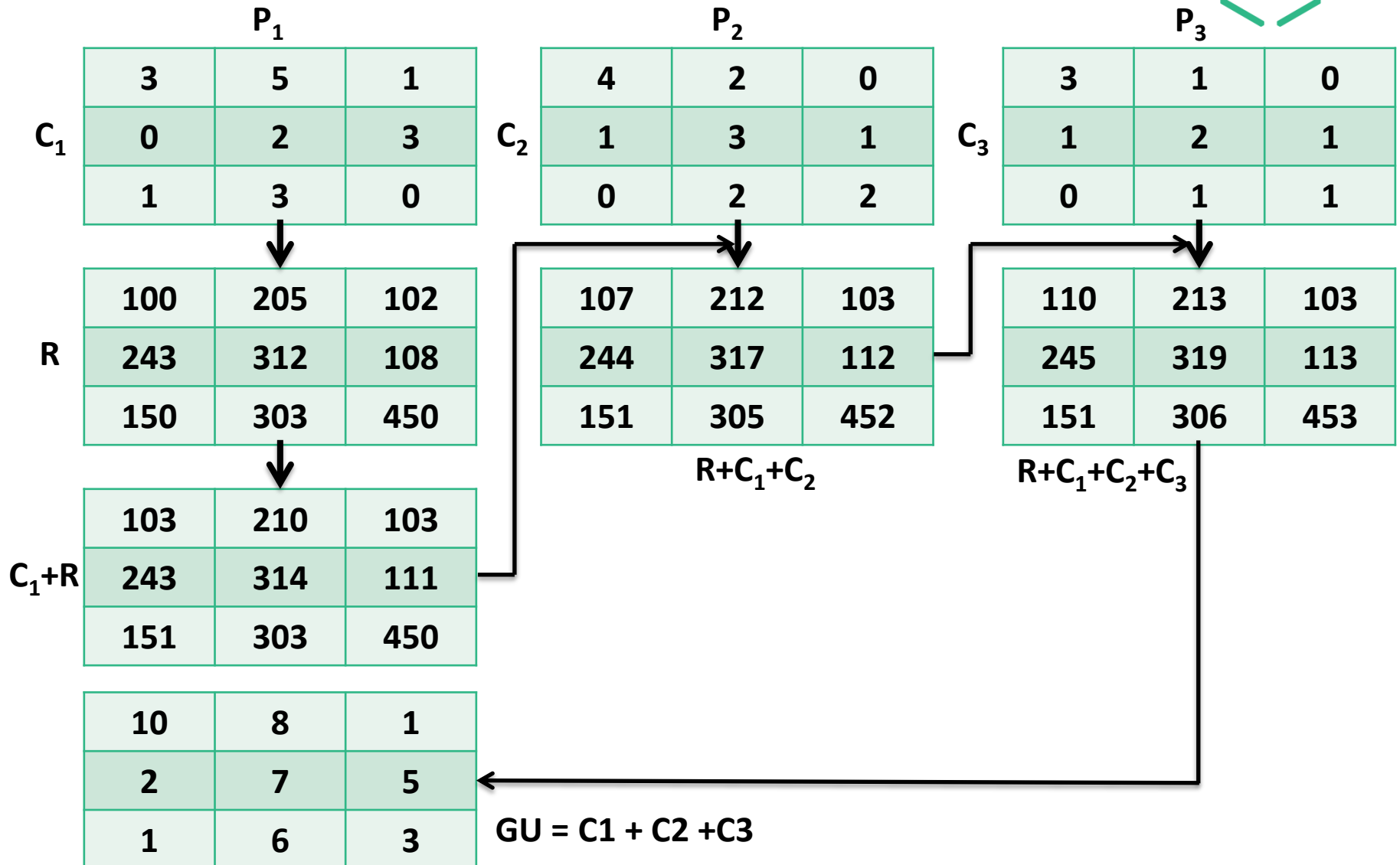
10	0	0
0	0	0
0	0	0

- The number of homomorphic operations required at each party is
  - $O(D \times W)$
- The main computational overhead is the encryption of sketches and secure multiplication of sketches to generate GS
- High communication cost due to the size of encrypted sketches
  - Proportional to the size of sketches multiplied by a constant factor

# Perturbation-based



# Perturbation-based (contd..)



# Perturbation-based (contd..)

GS

<b>3</b>	2	-3
2	2.5	0.5
1	1.5	2

$$\bigvee_i c[i] = 1 \text{ iff } c[i] = p \\ \text{else } c[i] = 0$$

<b>1</b>	0	0
0	0	0
0	0	0

GS'

<b>1</b>	0	0
0	0	0
0	0	0

×

GU

10	8	1
2	7	5
1	6	3

=

10	0	0
0	0	0
0	0	0

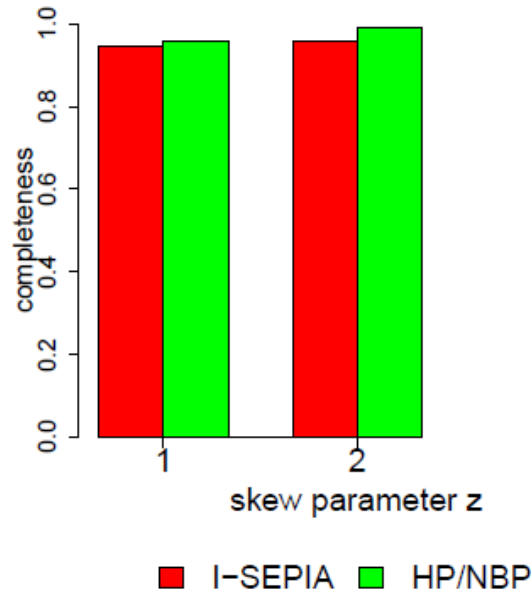
- Symmetric noise is added (a random value drawn from a Laplace distribution, location and scale parameters are set to 0 and 1, respectively) to sanitize the number of parties with infrequent ( $< \theta$ ) values

# Evaluation

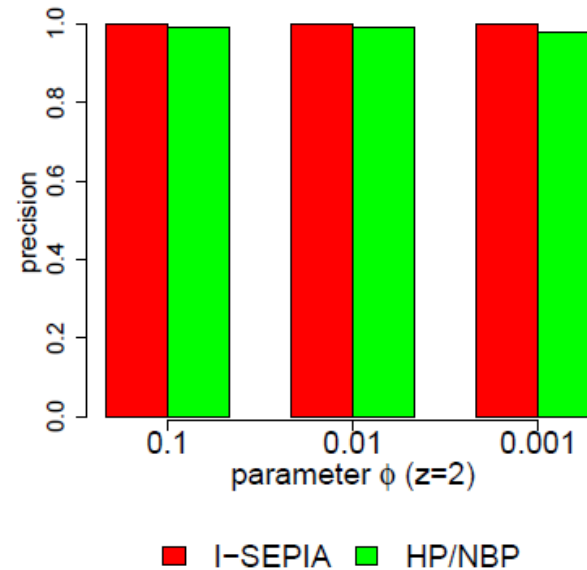


- **Application domain:**
  - Monitoring and the identification of common Web resources appearing at five local parties
- **Datasets:**
  - Synthetic dataset –  $10^9$  occurrences of  $10^6$  distinct elements following zipf distribution (skew parameters  $z=1$  and  $z=2$ )
  - Real dataset – anonymized list of top 1000-ranked Web sites from a Greek IT company
- **Measures:**
  - Efficiency – execution time and space required
  - Accuracy – precision, recall, and completeness measure
    - Completeness =  $1 - \frac{\sum_{s \in S} |C'(s) - C(s)|}{\sum_{s \in S} C(s)}$ , where  $C(.)$  and  $C'(.)$  are the actual and estimated counts, respectively
- **Baseline:**
  - PPSI in Sepia library - ISepia

# Evaluation using Synthetic Dataset (contd..)



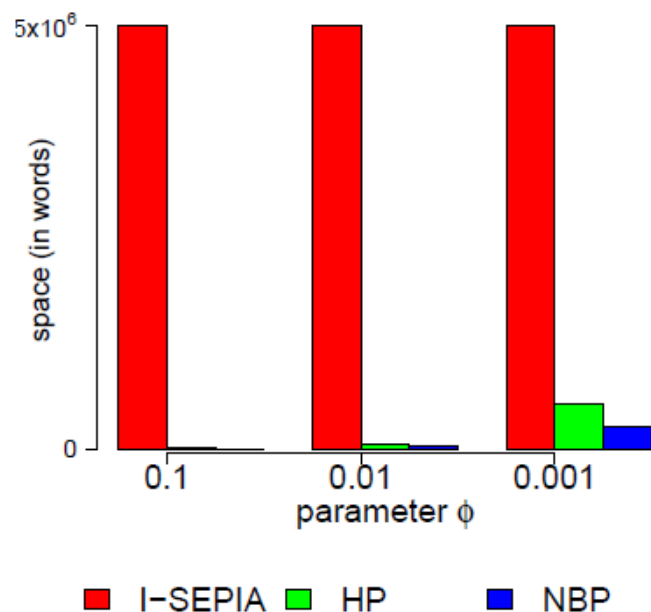
(a) By using skewed data, HP/NBP exhibit high completeness rates ( $\phi = 0.01$ ).



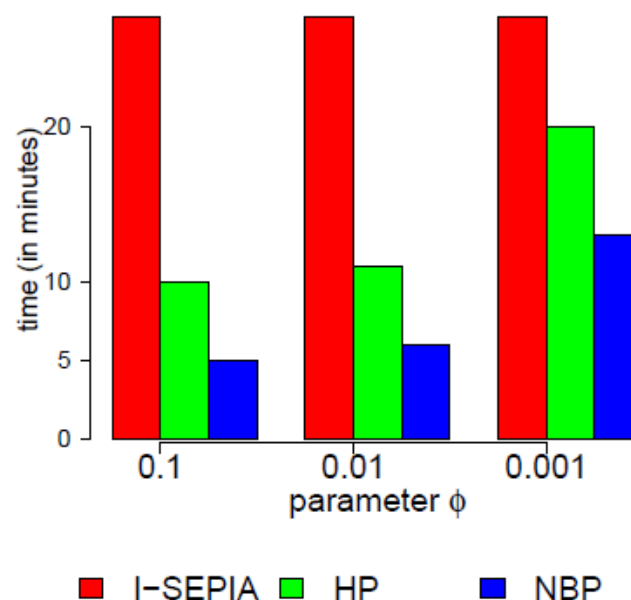
(d) The precision rates of HP/NBP are almost 1.0, by using highly skewed data.



# Evaluation using Synthetic Dataset (contd..)

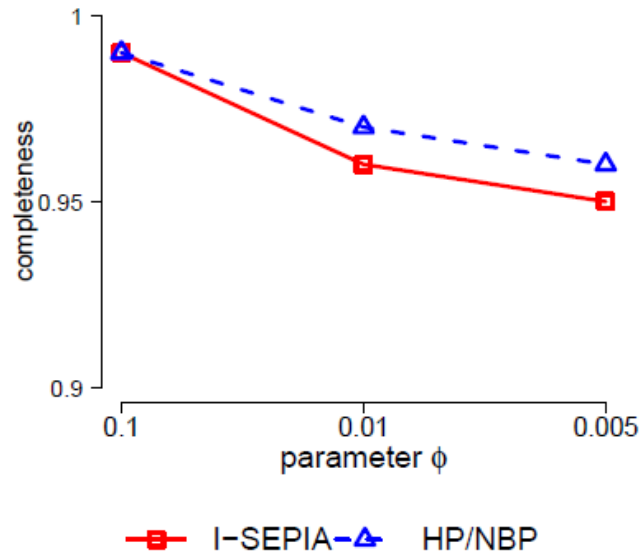


(e) Space requirements in words.

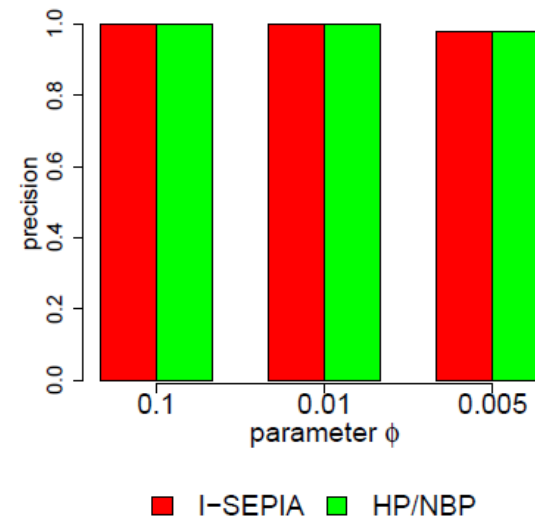


(f) Time performance in minutes.

# Evaluation using Real Dataset



(b) The completeness rates of our protocols are constantly above 0.95.

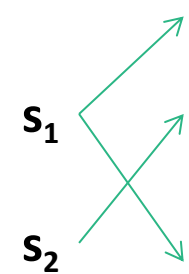


(c) The precision rates for both our protocols and I-SEPIA are almost the same, very close to 1.0.

# Other Probabilistic Data Structures

- Cuckoo filters

- Highly space-efficient
  - Efficient than Bloom filters when FPP < 3%
- Two candidate blocks for an item  $s$ 
  - $h_1(s) = \text{hash}(s)$
  - If  $h_1(s)$  empty: insert  $f = \text{fingerprint}(s)$
  - else: insert  $f$  into  $h_2(s) = h_1(s) \oplus \text{hash}(f)$



00		
01		
02	101	
03		
04	100	
05		
06		
07	101	101
08		
09		

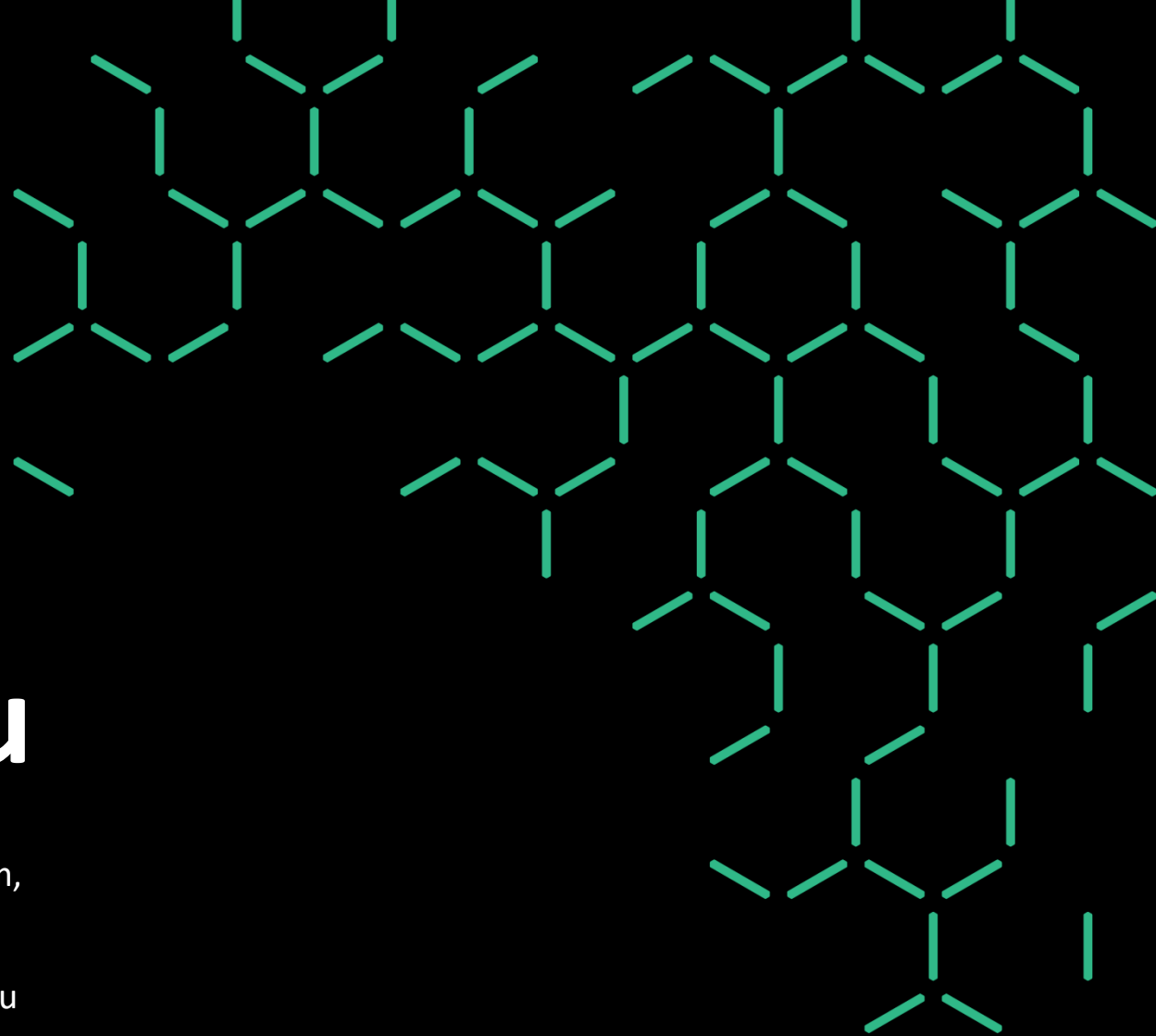
- HyperLogLog

- Count-distinct problem
  - How many unique elements in a multi-set?
  - Approximate way for efficient calculation of huge lists
  - Use the highest number of consecutive zeroes in the binary representation of the hash for each element to predict the cardinality of the entire set

# Conclusion and Research Directions



- **Presented protocols for PPSI using probabilistic data structures and perturbation-based privacy techniques**
  - Experimental study shows the accuracy and efficiency of these protocols
- **Research directions:**
  - Privacy preserving aggregated mobility data
    - Using probabilistic data structures for practical applications
      - Transport planning and management
      - Privacy preserving recommendation systems
      - Business applications – targeted marketing
  - A framework of probabilistic data structures for privacy preserving techniques
    - Study space/time/accuracy/privacy trade-off in different techniques and their applicability for different applications



# Thank you

Dinusha Vatsalan  
Research Scientist, Data Privacy Team,  
Networks Research Group

t +61 2 9490 5734

e [dinusha.vatsalan@data61.csiro.au](mailto:dinusha.vatsalan@data61.csiro.au)

w <https://research.csiro.au/ng>

[www.data61.csiro.au](http://www.data61.csiro.au)



# References



- *Mitzenmacher and Eli, Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis, Cambridge University Press, 2017*
- *Lai et al., An efficient Bloom filter based solution for multi-party private matching, Security and Management 2006*
- *Karapiperis et al., Large-scale multi-party counting set intersection using a space efficient global synopsis, DASFAA 2015*
- *Many et al., Fast private set operations with sepia, Technical report no.345, ETH Zurich, 2012*
- *Burkhardt and Dimitropoulos, Privacy-preserving distributed network troubleshooting – bridging the gap between theory and practice, ACM Transactions on Information Systems Security, 14(4), 2011*
- *Vatsalan and Christen, Scalable Privacy-Preserving Record Linkage for Multiple Databases, CIKM 2014*
- *Vatsalan et al., Scalable privacy-preserving linking of multiple databases using counting Bloom filters, ICDMW 2016*