Privacy Preserving Group Linkage

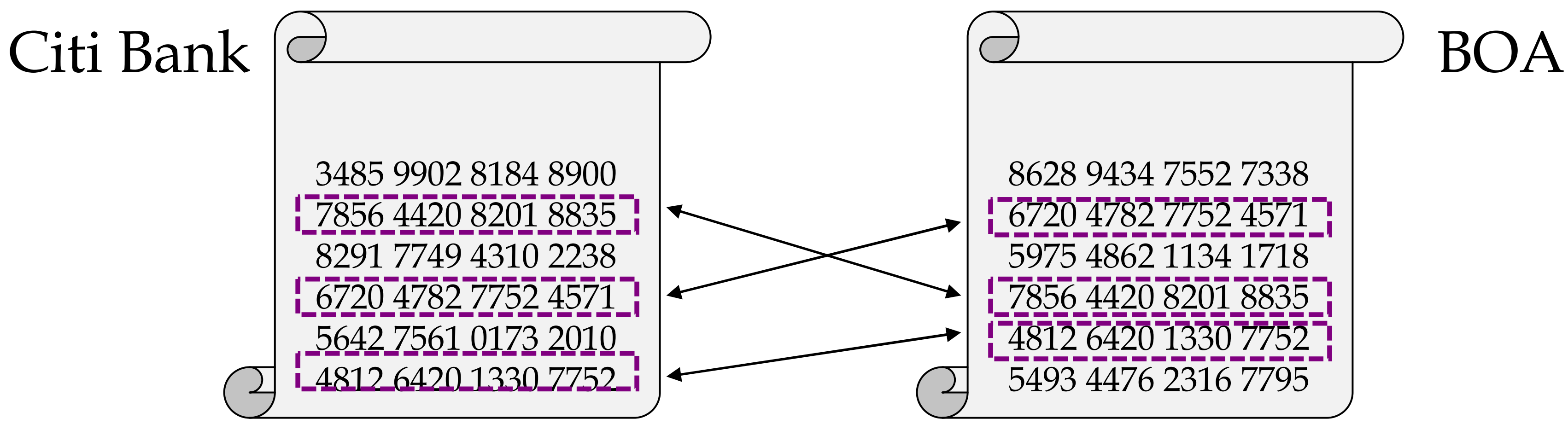
Fengjun Li[1], Yuxin Chen[1], Bo Luo[1], Dongwon Lee[2], and Peng Liu[2]

[1]EECS Department, University of Kansas,

[2]College of IST, Penn State University

Record Linkage

- Record linkage is to identify related ***records*** associated with the same entity from multiple databases



Privacy-Preserving Record Linkage

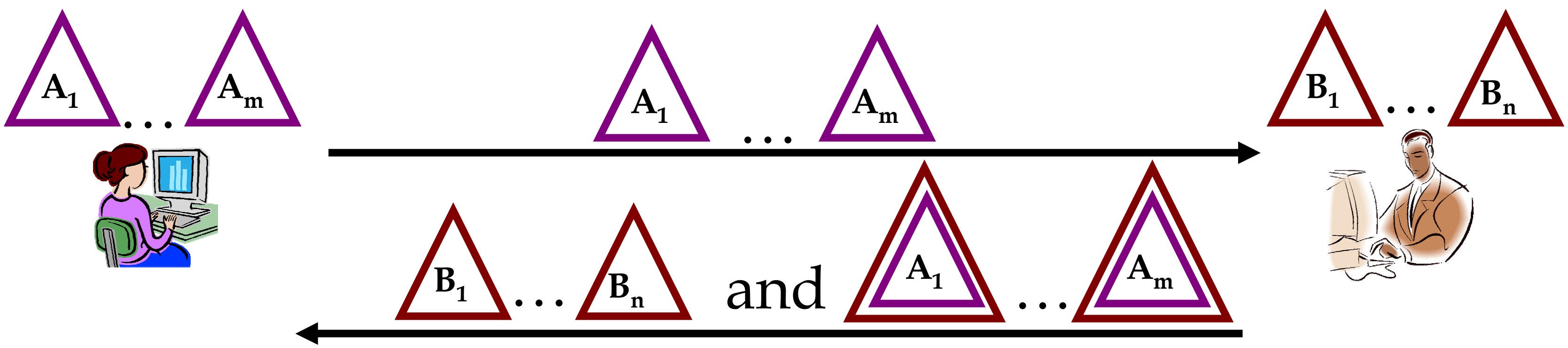
- Privacy becomes an issue when data is sensitive.
  - I will only share with you on the “linked records”
  - I will not give you the plain text of my primary keys.

- Secure multi-party set intersection problem
  - Solutions based on commutative encryption
  - Solutions based on homomorphic encryption

AES Protocol – Commutative Encryption Based

- **Commutative Encryption:** *using the same set of commutative keys, the encrypted content can be recovered in* ***any arbitrary order****.*

- **AES Protocol** [Agrawa et. al., SIGMOD 2003]:



Alice compares

to find intersection.

FNP Protocol – Homomorphic Encryption Based

- **Homomorphic encryption:** *allows certain algebraic operations in the plaintext to be performed on the ciphertext* ***without decryption***.

- **FNP Protocol** [Freedman et. al., EUROCRYP 2004]:

{r1,r2,…,rm}

1. Constructs polynomial $R(x) = \prod (x - r_i)$
2. Computes coefficients in $R(x) = \sum_{u=0}^{u=m} \alpha_u x^u$

Encrypt coefficients with homomorphic key: $E(\alpha_0), E(\alpha_1), \ldots, E(\alpha_m)$

{s1,s2,…,sn}

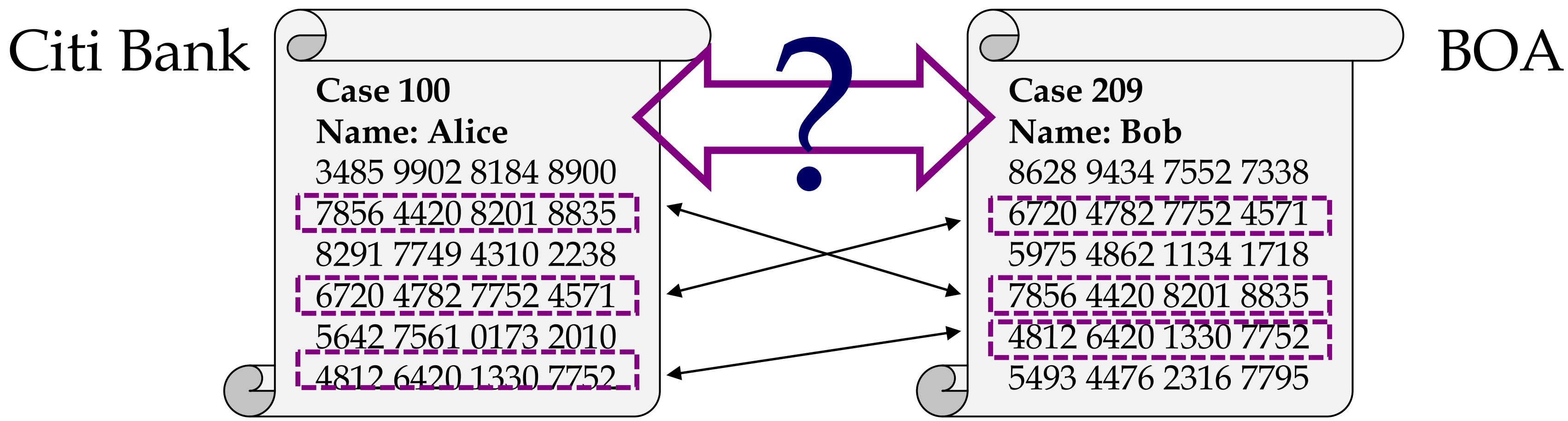
3. Re-construct encrypted polynomial: $E(R(x)) = \sum_{u=0}^{u=m} E(\alpha_u) x^u$
4. Evaluate **E**(R(sj)) for each element sj
5. Choose random γand v, and compute **E**(γ×R(sj)+v). For each sj□**R∩S**, **E**(R(sj))=0, and **E**(γ×R(sj)+v)=E(v).

E(γ×R(sj)+v)

6. Decrypt **E**(γ×R(sj)+v), and the number of v = |**R∩S**|.

Group Linkage

- Extended from record linkage [On et. al., ICDE 2007]
  - Records -> groups of records

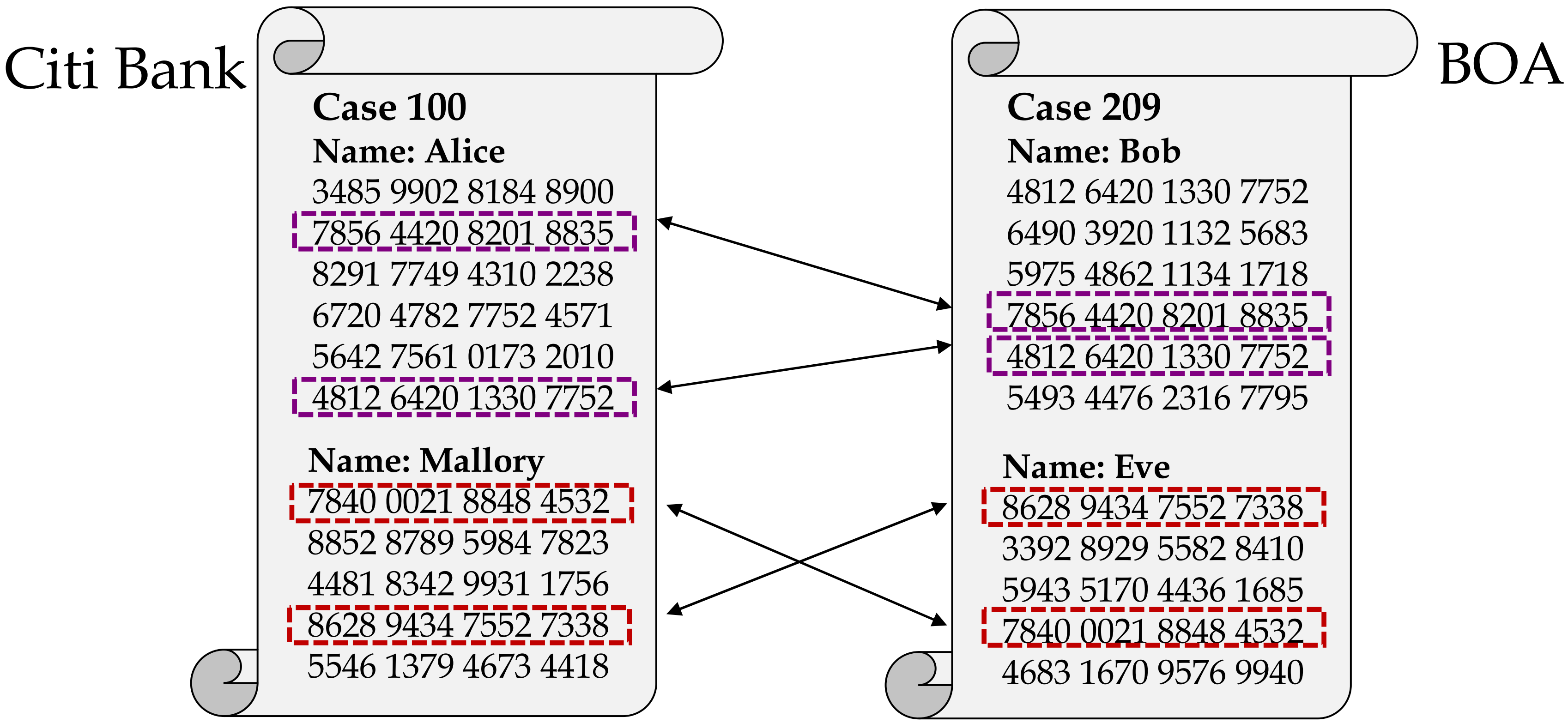


- Group linkage is to identify related ***groups of records*** associated with the same entity from multiple databases

Group Linkage

- For two sets of groups of records $\mathbf{R}=\{R_1, \ldots, R_u\}$ and $\mathbf{S}=\{S_1, \ldots, S_v\}$, GL calculates ***group-level similarity*** **SIM**(R,S), and determines if **R** and **S** are associated with the same entity
  - For $R=\{r_1,\ldots r_m\}$ and $S=\{s_1,\ldots s_n\}$, calculate *record-level similarity* **sim**(r,s)
  - **SIM**(**R**,**S**) is a function of **sim**(r,s)

Group Linkage: Exact Matching Example



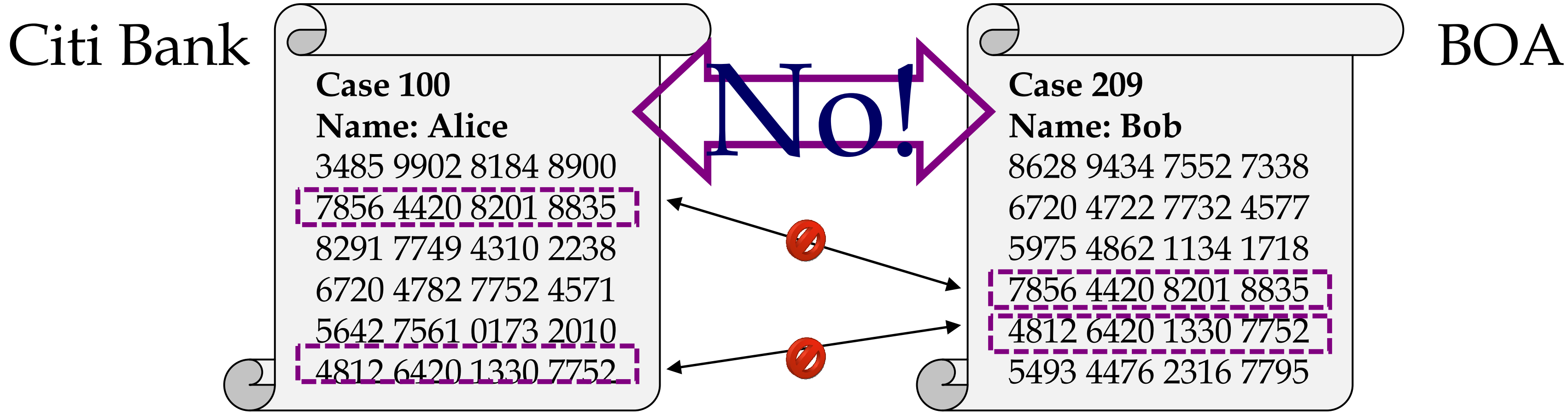
Group Linkage: Approximate Matching Example

- modeling and representation of data, metadata, ontologies, and processes
- querying of scientific data

- modeling and representation of data and knowledge for scientific domains
- querying and analysis of scientific data.

Group Membership Inference Problem

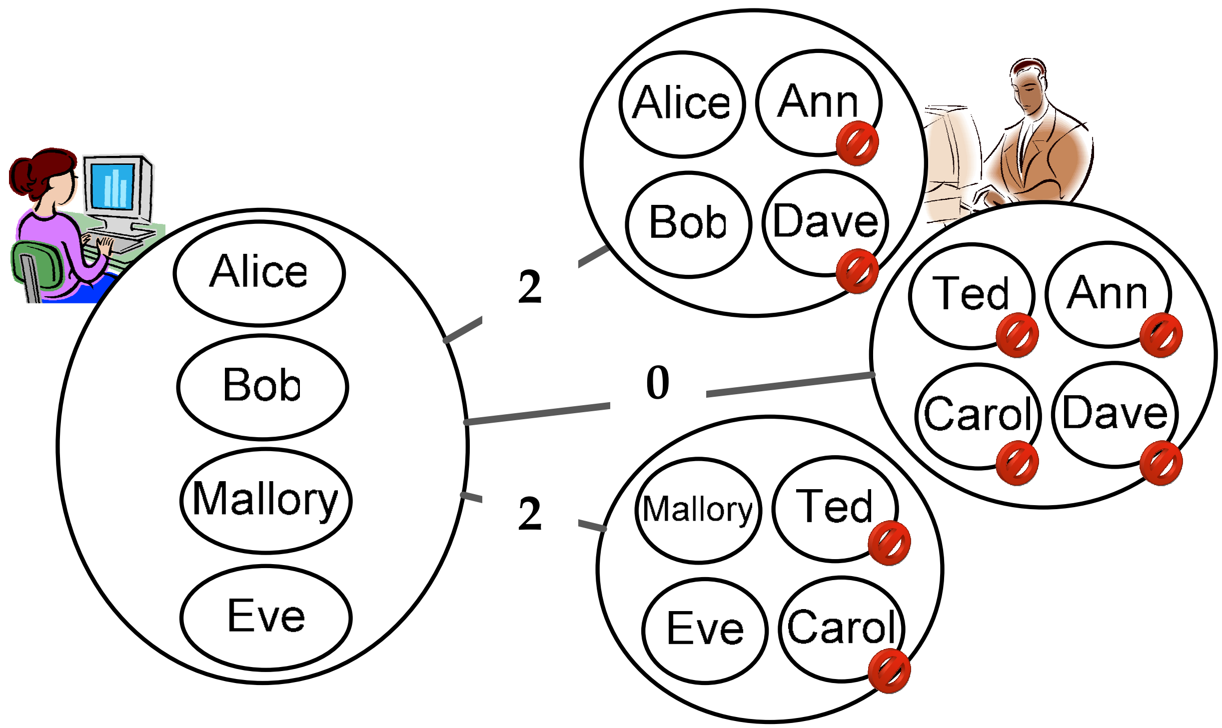
- Two parties share two groups after they confirm both groups are associated with the same entity.
- Privacy?
  - Cannot share "intersect" records when two groups are not linked.



Privacy-Preserving Group Linkage (PPGL)

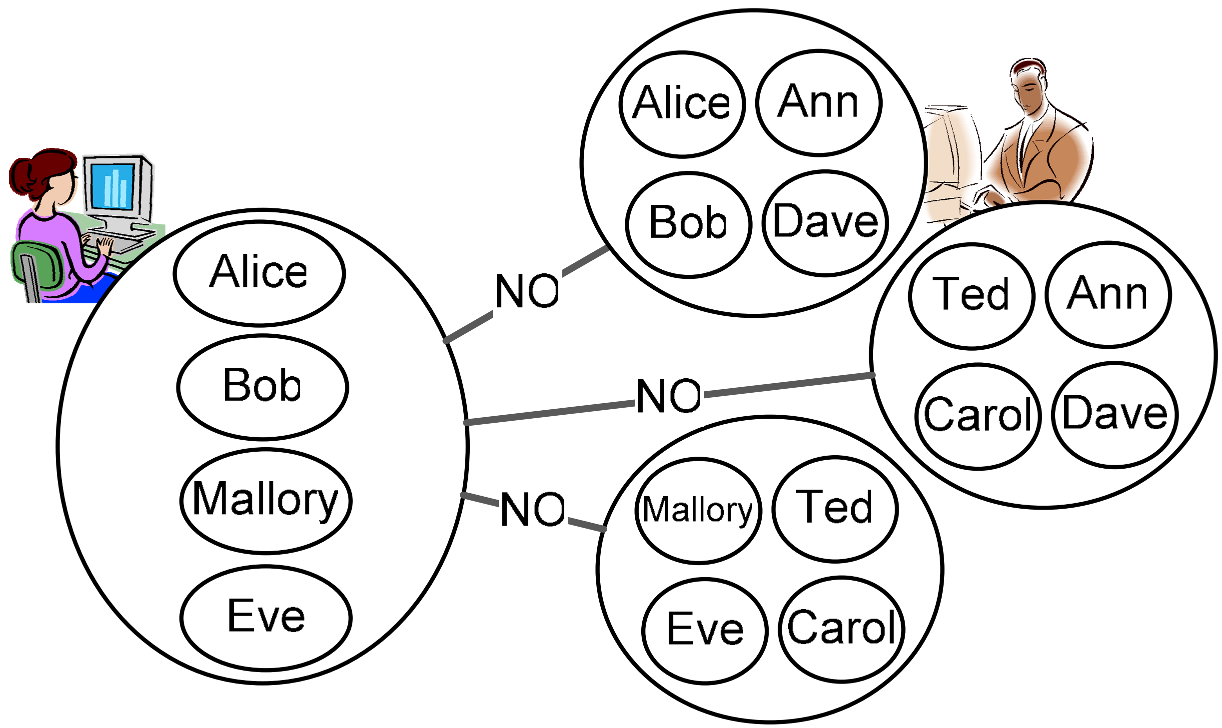
- PPRL protocols can be applied in PPGL
  - Secure set intersection size
  - The intersection size can be used to calculate group-level similarity
- However, directly applying PPRL protocol suffers from ***group membership inference*** problem

Group Membership Inference Problem



– Identities of overlapped group members can be inferred
– An attacker can manipulate the group members to infer more

Ideal PPGL Protocol



- Alice and Bob negotiate a similarity threshold
- For each group-wise comparison, Bob answers only “YES” or “NO”, instead of calculated similarity value

Threshold Privacy-Preserving Group Linkage

- **TPPGL Problem:** Alice and Bob preset a threshold θ, and follow the protocol to match two groups **R** and **S**. In the end, they learn only |**R**|, |**S**|, and a Boolean result B, where B = true iff **SIM**(**R**, **S**) $\geq \theta$.

- We propose three TPPGL protocols for both exact matching and approximate matching
  - K-combination approach for TPPGL-E
  - Homomorphic encryption approach for TPPGL-E
  - TPPGL-A protocol with record-level cut-off

K-Combination Approach for TPPGL-E

- Alice has a set of groups **R**=$\{r_1,\ldots,r_m\}$, and Bob has a set of groups **S**=$\{s_1,\ldots,s_n\}$. They negotiate a similarity threshold θ.
- Calculate the ***minimum number of identical records*** in **R** and **S** for them to be linked

$$\mathbf{SIM}(\mathbf{R}, \mathbf{S}) = k/(|\mathbf{R}|+|\mathbf{S}|-k) \geq \theta, \quad \text{so} \quad k = \left\lceil \frac{(m+n)\theta}{1+\theta} \right\rceil$$

- We enumerate all *k-combinations* of Alice's and Bob's group elements. **R** and **S** are linked iff there is at least one identical k-combination.

Input: Alice's group **R**=$\{r_1,\ldots,r_m\}$, Bob's group **S**=$\{s_1,\ldots,s_n\}$, and a pre-negotiate similarity threshold θ

Protocol:

⟵ Bob's group size n

Alice calculates k with m and n ⟶

- Alice creates $p = C_k^m$ k-cobinations and sort them: {A1,…, Ap};

  Bob creates $q = C_k^n$ k-combinations and sort them: {B1,…, Bq};

- Alice applies hash function to obtain: {h(A1),…, h(Ap)};

  Bob applies hash function to obtain: {h(B1),…, h(Bq)};

- Alice encrypts hashed k-combinations: {E**r**(h(A1)),…, E**r**(h(Ap))};

  Bob encrypts hashed k-combinations: {E**s**(h(B1)),…, E**s**(h(Bq))}

{E**r**(h(A1)),…, E**r**(h(Ap))} ⟶

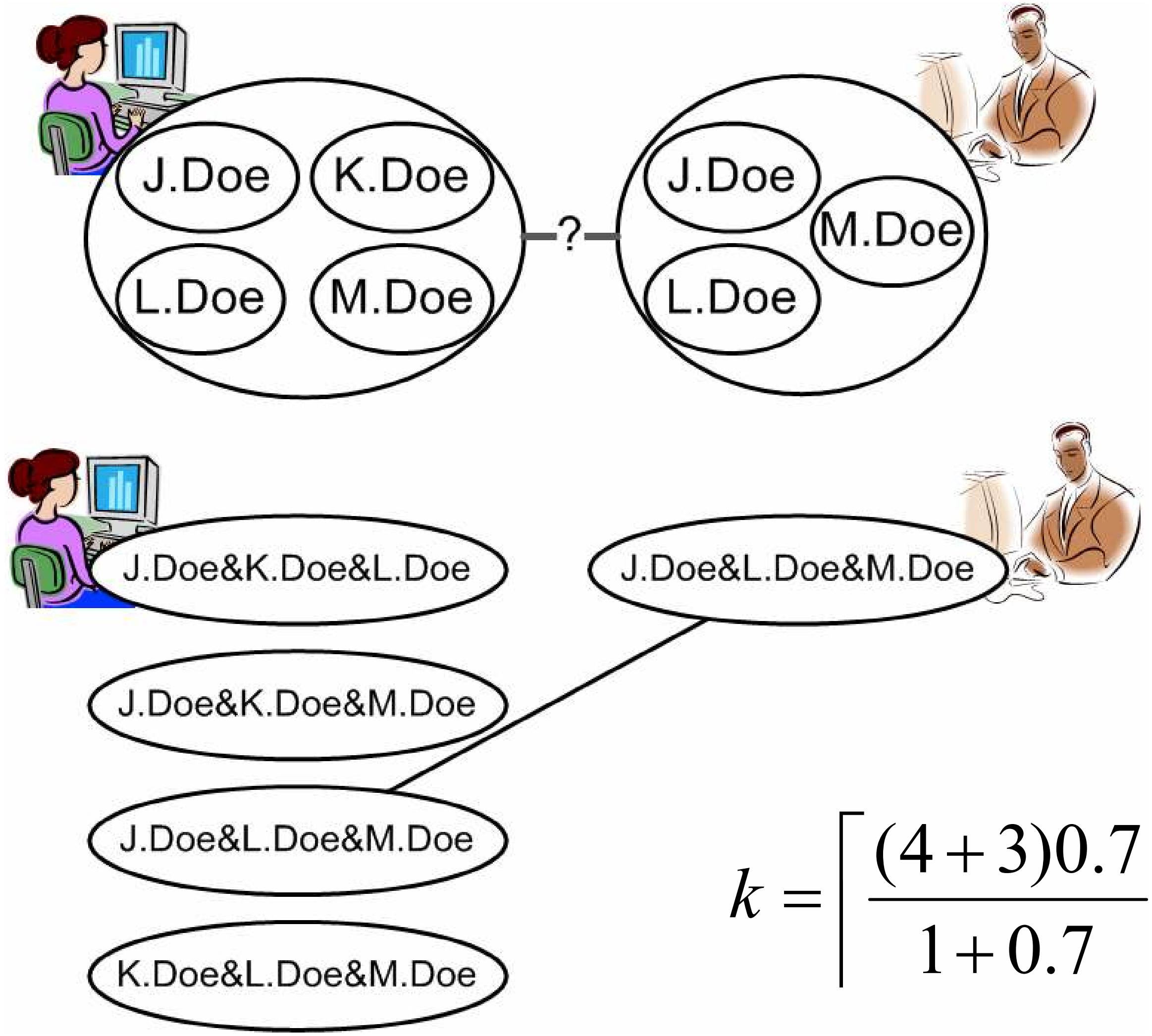
⟵ {E**s**(h(B1)),…, E**s**(h(Bq))}, E**s**({E**r**(h(A1)),…, E**r**(h(Ap))})

- Alice encrypts {E**s**(h(B1)),…, E**s**(h(Bq))} with E**r**, and compares E**r**(E**s**(h(B))) and E**s**(E**r**(h(A)))

- If the intersection size is greater than 1, group similarity is greater than θ

Result: Alice and Bob learn |**R**|, |**S**|, and if group similarity > θ

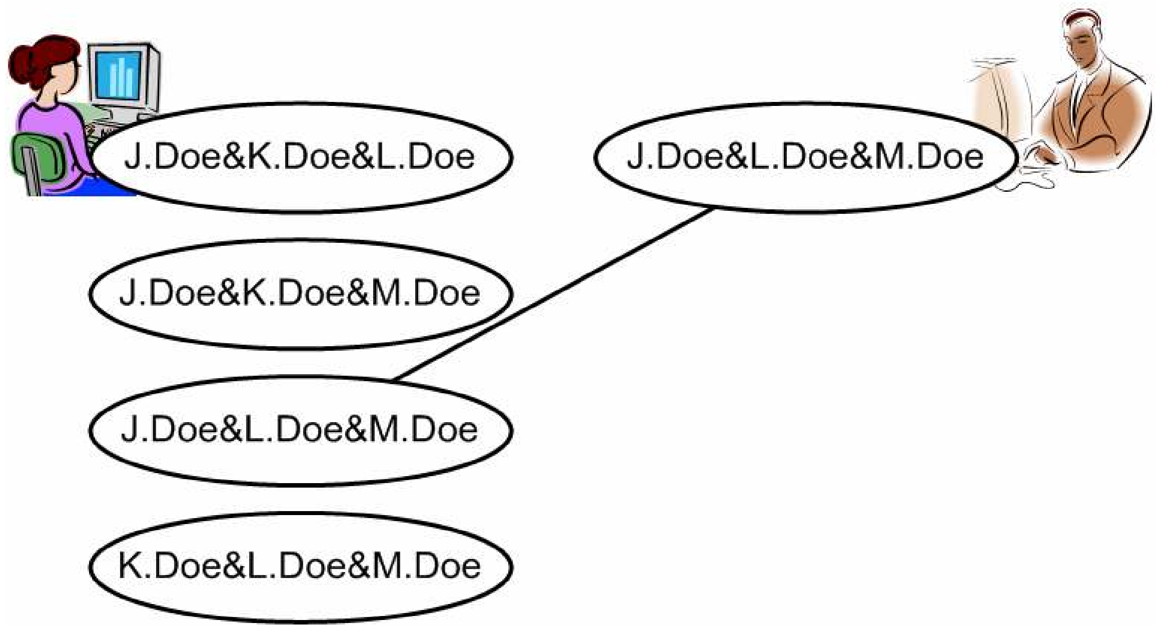
K-Combination Approach Example



$$k = \left\lceil \frac{(4+3)0.7}{1+0.7} \right\rceil = 3$$

K-Combination Approach for TPPGL-E

- Problem?
  - Computation!



Homomorphic Encryption Approach for TPPGL-E

Input: Alice's group **R**=$\{r_1,\ldots,r_m\}$, Bob's group **S**=$\{s_1,\ldots,s_n\}$, and a pre-negotiate similarity threshold θ

Protocol:

Bob group size n

k; keys for homomorphic encryption

- Alice constructs $R(x)=\prod(x-r_i)$ and computes coefficients α_u that $R(x)=\sum_{u=0}^{u=m}\alpha_u x^u$

Alice encrypts the coefficients $\{E(\alpha_0),..,E(\alpha_m)\}$ and send to Bob

- For each sj, Bob evaluates the polynomial to get **Enc**(R(sj)), without decryption
- Bob chooses a random value γ, and a pre-set special value v. For each **Enc**(R(sj)), Bob computes **Enc**(γ× R(sj)+v).
- Bob chooses a random number kb, and injects kb number of **Enc**(v) into the set. Meanwhile, Bob also injects random number of random values into this set.

Bob permutes the polluted set of **Enc**(γ× R(sj)+v)

- Alice decrypts all items, and counts the number of v values: kb+|**R**∩**S**|

Enc(kb+|**R** **S**|)

- Bob calculates **Enc**(kb+|**R**∩**S**|)-**Enc**(kb+k)=**Enc**(|**R**∩**S**|-k), and then creates random number γ'<< N, and v'<γ'

Enc(Γ ' ×(**R**∩**S** -k)+v')

- Alice decrypts m=γ'×(|**R**∩**S**|-k)+v', and output "YES" if m<N/2, or "NO" if m>N/2

Result: Alice and Bob learn |**R**|, |**S**|, and if group similarity > θ

Group Linkage with Approximate Matching

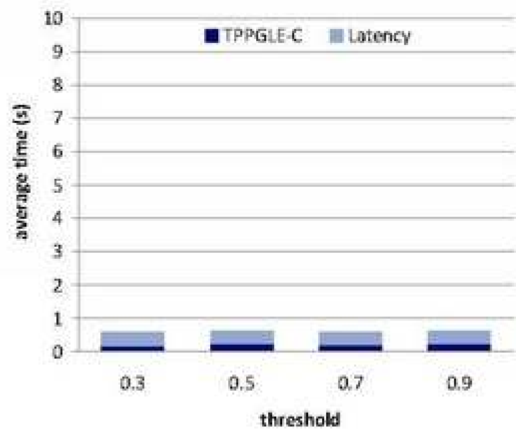
- Alice holds a group of records
- Bob holds a group of records
- Record level similarity: inner product with cut-off
- Group level similarity:

$$\mathbf{SIM}(R, S) = BM_{sim,\rho}(R, S)$$
$$= \min(m', n') / (|R| + |S| + \min(m', n'))$$

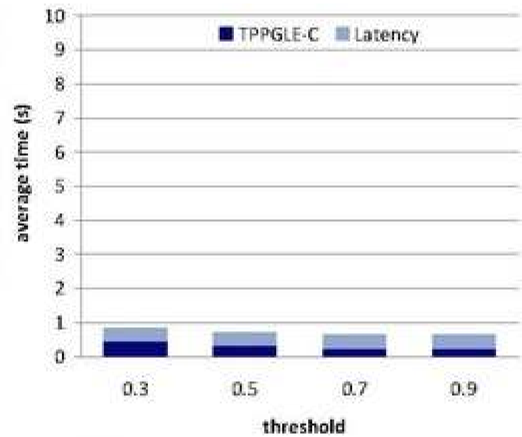
Experiment Results

- Three real data sets [Tang et. al., KDD 2009]
  - AN: a co-author network with 640,134 authors and 1,554,643 co-author relationships
  - CN: a paper citation network of 2,329,760 papers and 12,710,347 citations
  - MN: a movie network with 142,426 relationships
  - Generate synthetic groups
- Evaluate *end-to-end execution time* with varying *group-size* (with 5, 10, 15 records per group) and *threshold* θ(□{0.3, 0.5, 0.7, 0.9})

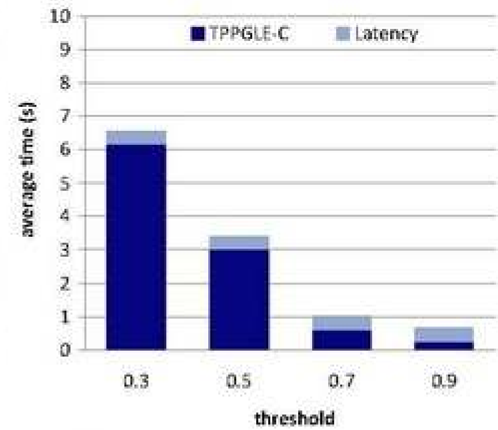
Average End-to-End Execution Time



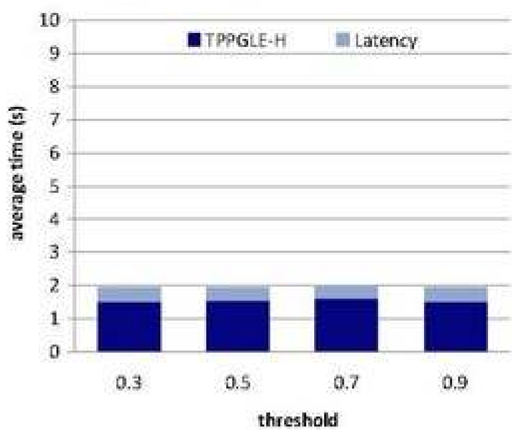
(a) Group size = 5.



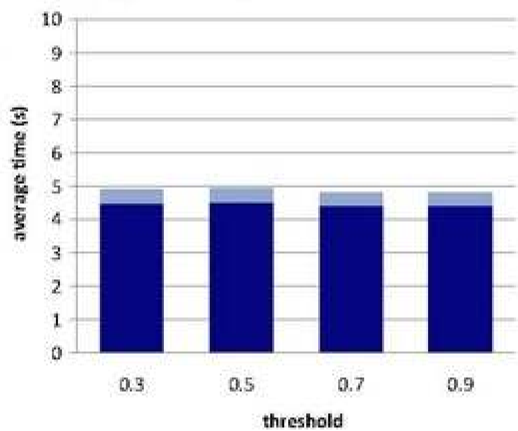
(b) Group size = 10.



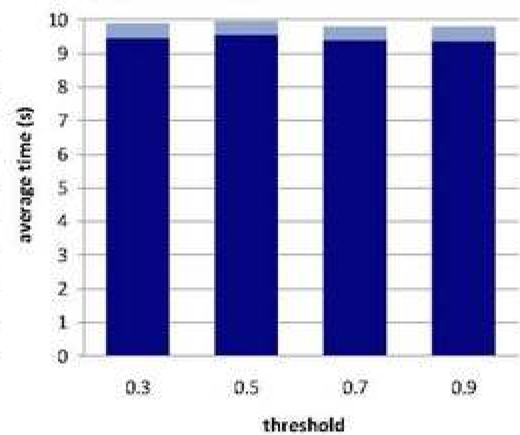
(c) Group size = 15.



(d) Group size = 5.



(e) Group size = 10.



(f) Group size = 15.

Thank You!

Questions?