

# Privacy Preserving Subgraph Matching on Large Graphs in Cloud

## Large Graphs in Cloud

Zhao Chang<sup>1</sup>, Lei Zou<sup>2</sup>, Feifei Li<sup>1</sup>

<sup>1</sup>University of Utah, <sup>2</sup>Peking University



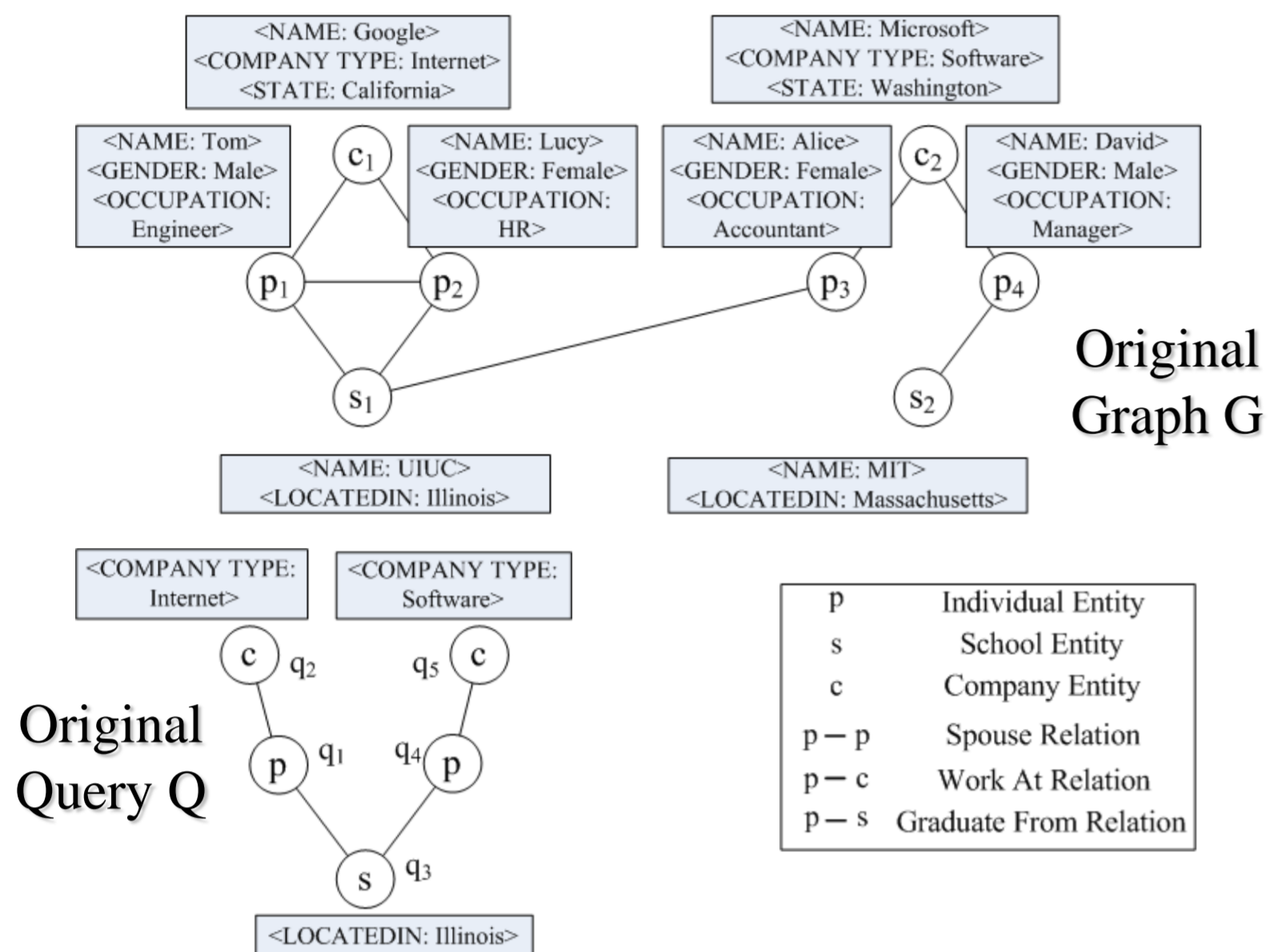
### INTRODUCTION

#### Subgraph Matching on Large Graphs

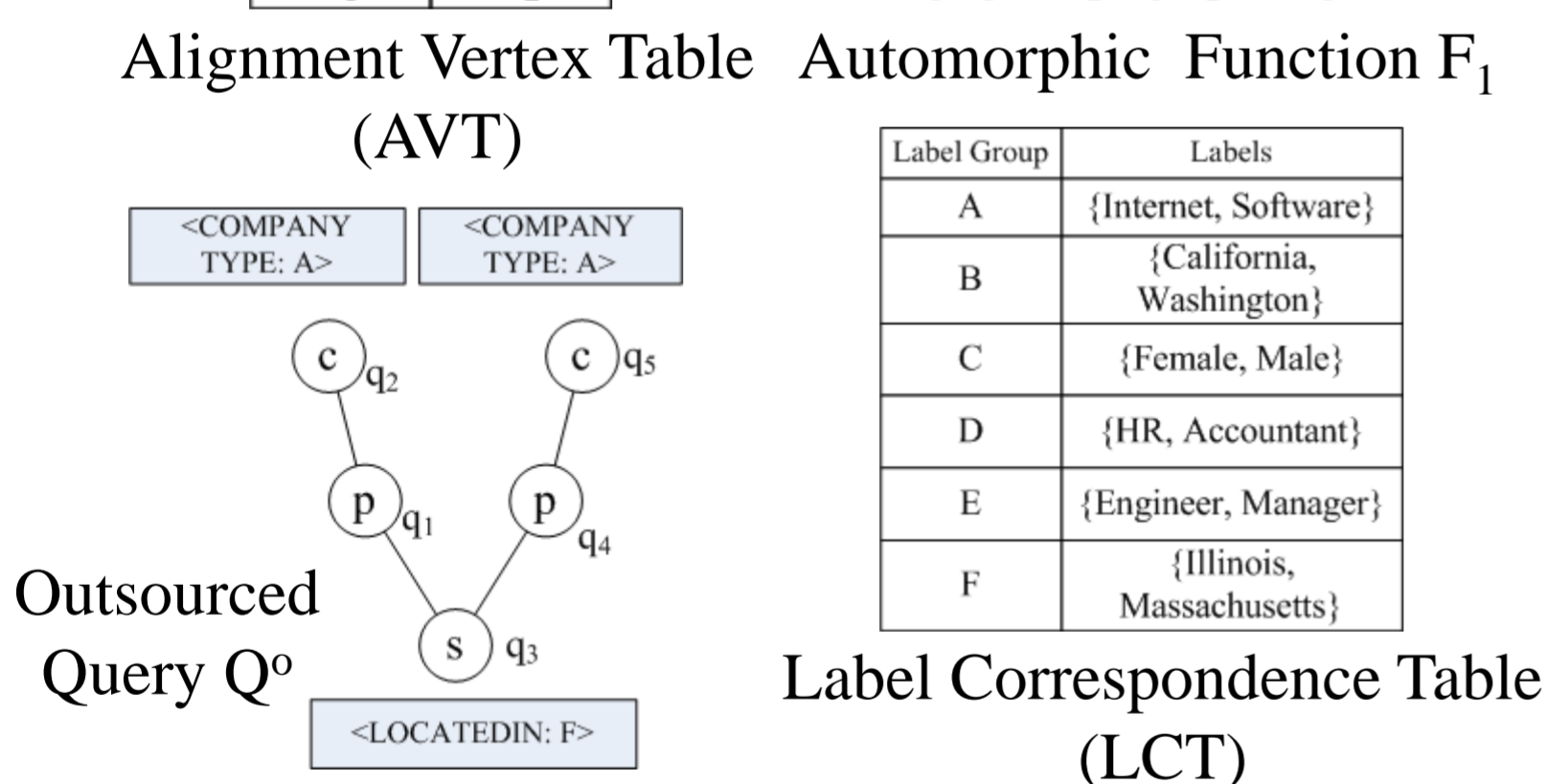
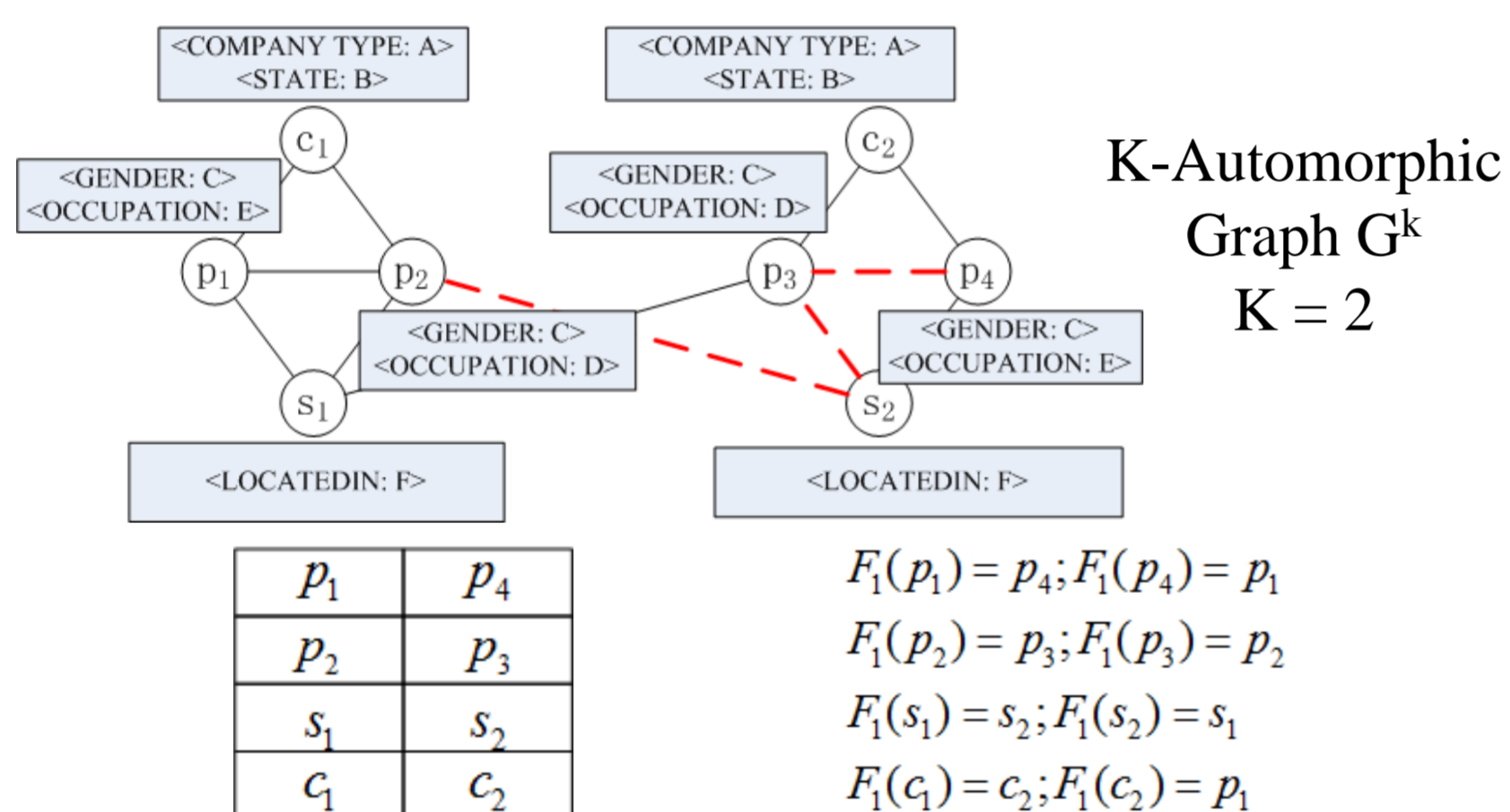
- Problem: high computational complexity
- Solution: cloud computing

#### Data Privacy

- Problem: untrusted cloud
- Solution: construct a k-automorphic graph and anonymize query graphs



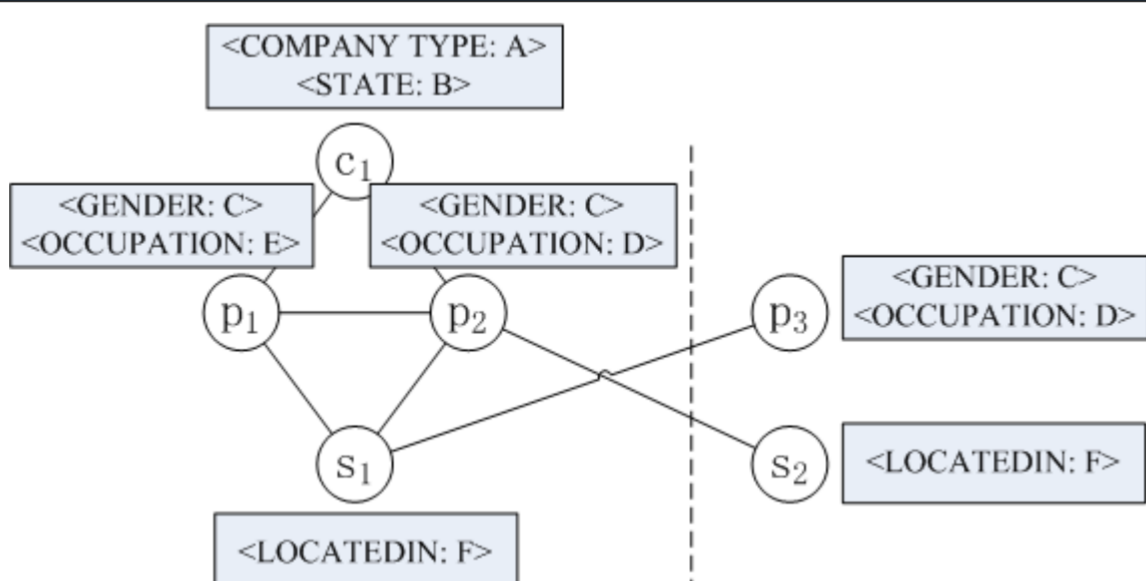
### K-AUTOMORPHISM



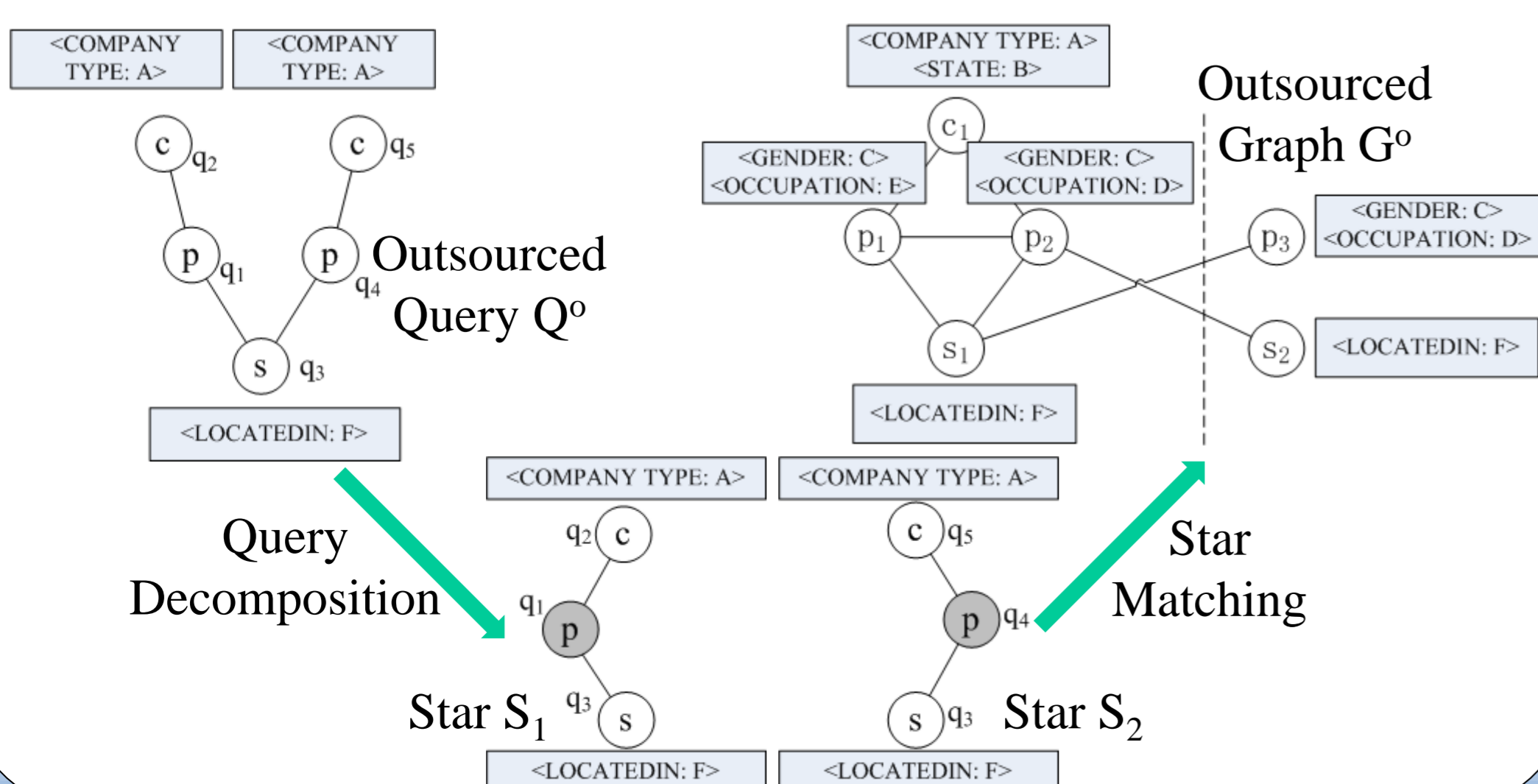
### OPTIMIZATION 1

#### Optimization 1

- Upload  $G^o$  rather than  $G^k$  to the cloud
- Save cloud storage



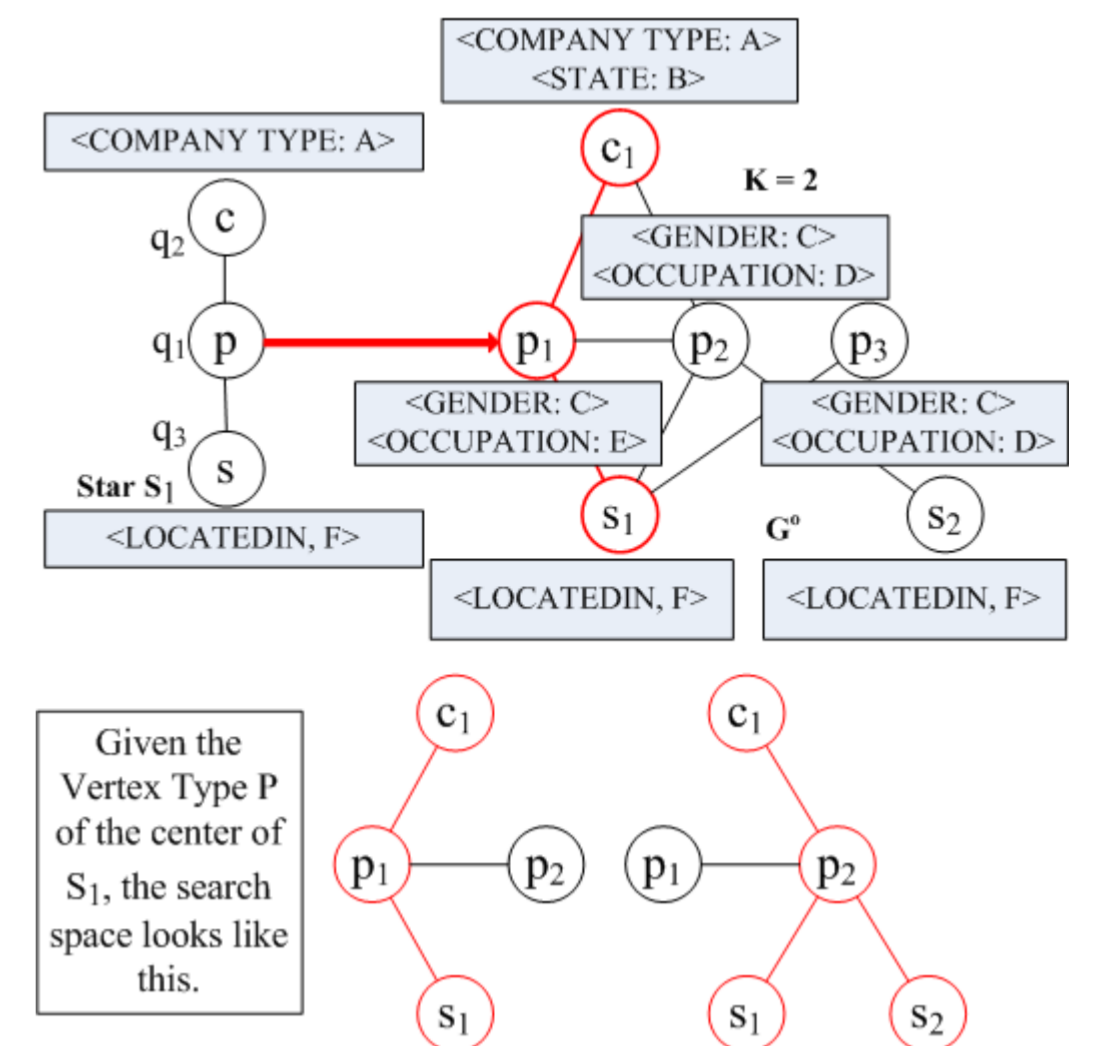
### STAR MATCHING



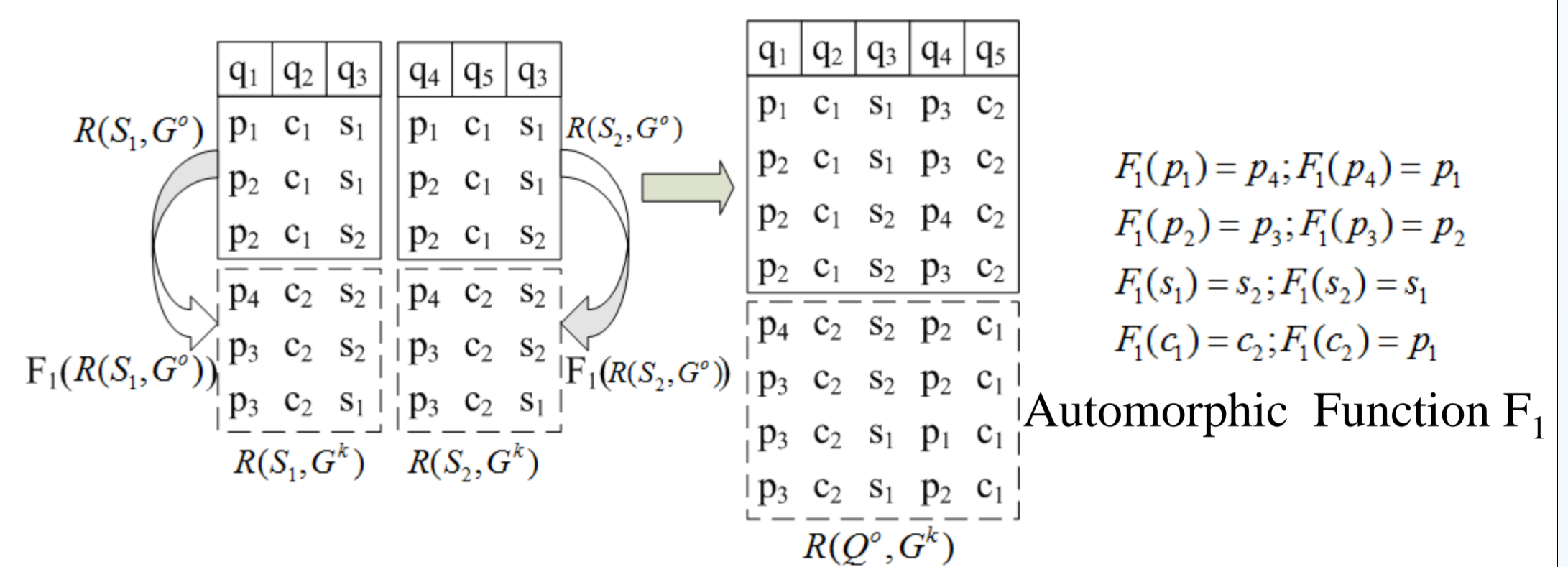
### OPTIMIZATION 2

#### Optimization 2

- Estimate the search space of matching a star query
- Better label generalization, better query decomposition
- Save query time cost



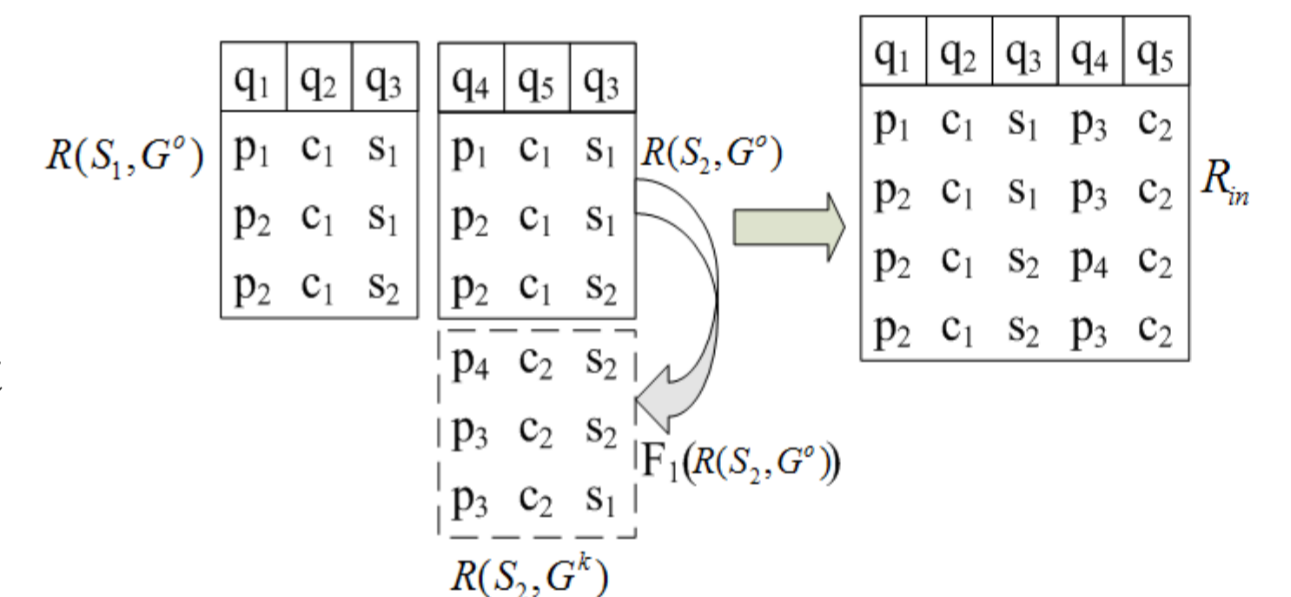
### RESULT JOIN



### OPTIMIZATION 3

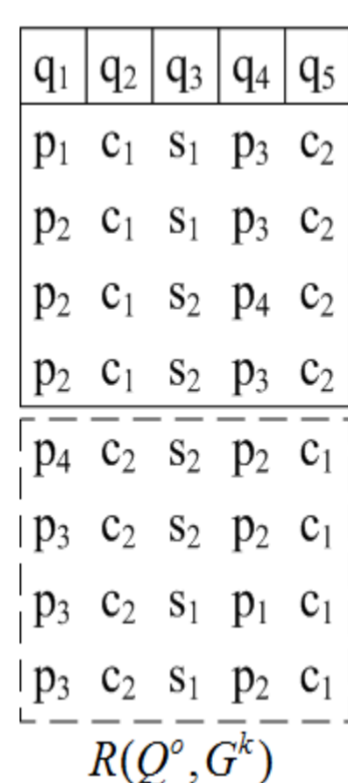
#### Optimization 3

- Generate  $R_{in}$  rather than all the results
- Save query time cost and communication cost

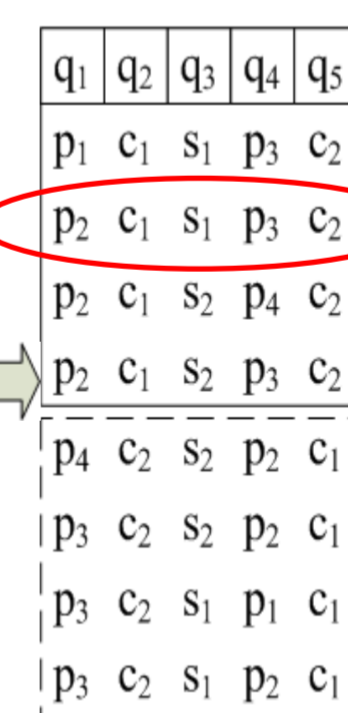


### CLIENT FILTERING

#### Generating Candidate Results

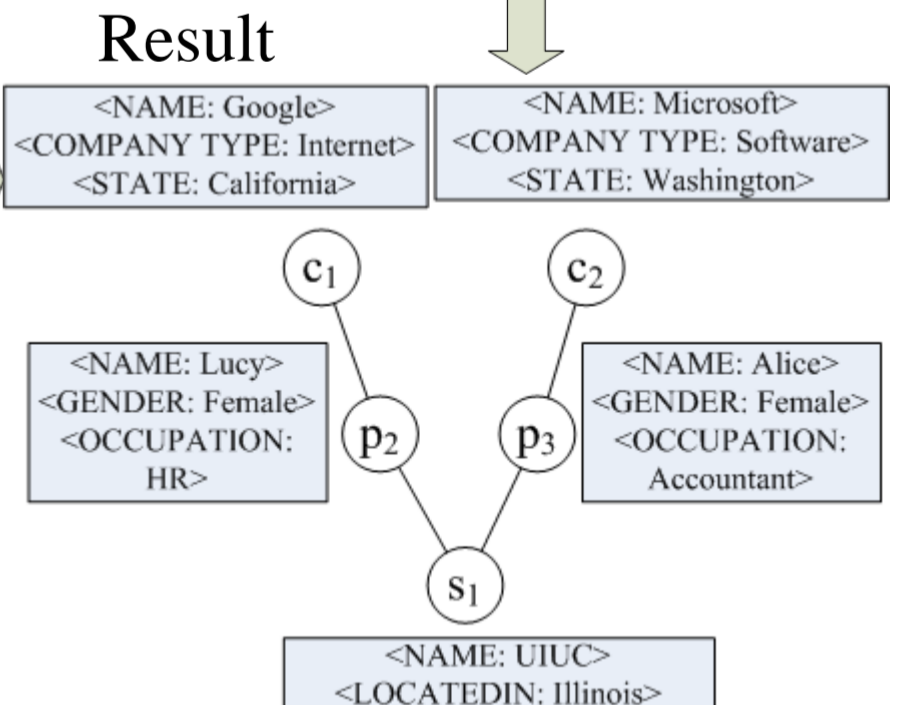


#### Filtering

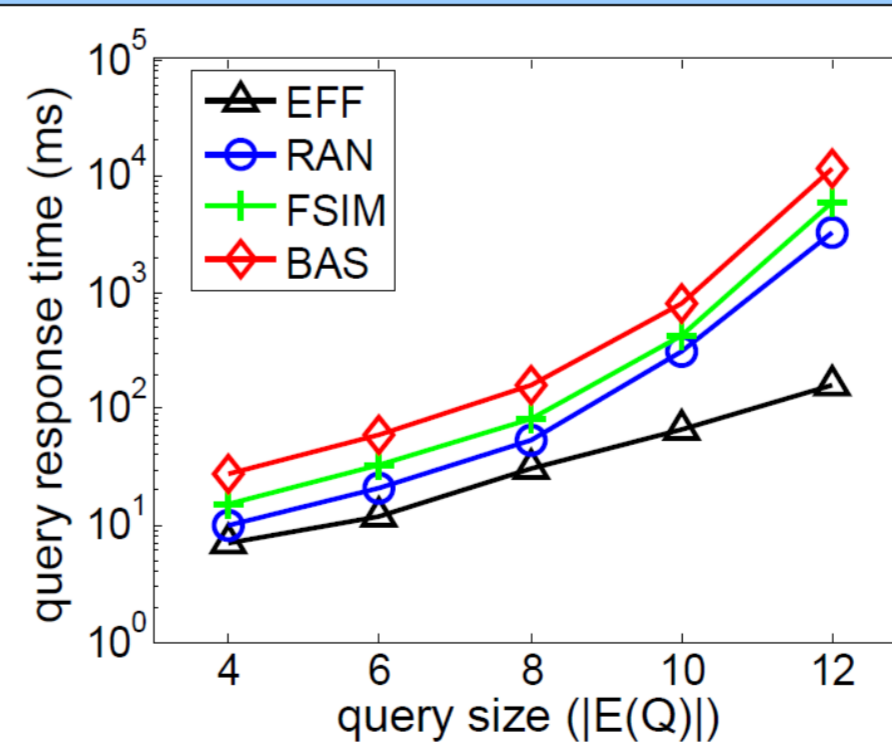


#### Original Query Q

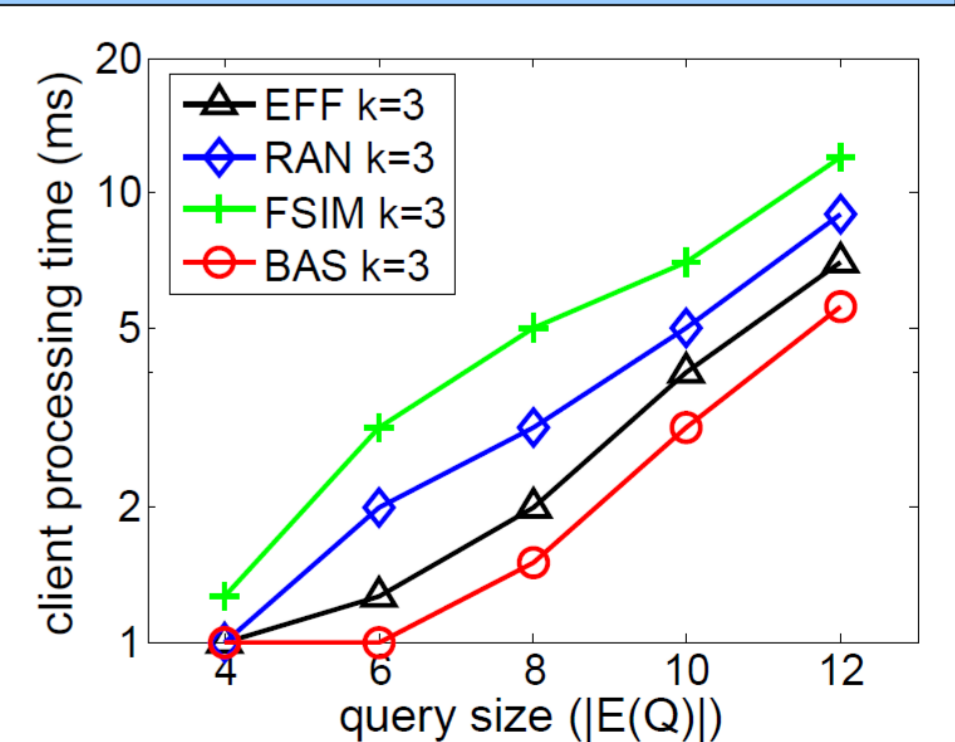
#### Matching Result



### EXPERIMENTAL RESULTS



Cloud processing time on DBpedia (K = 3)



Client processing time on DBpedia (K = 3)

### CONCLUSION

- We present an efficient framework.
- We protect both structural and label privacy.
- We explore a number of optimization techniques.
- We perform extensive experiments on large real graphs.