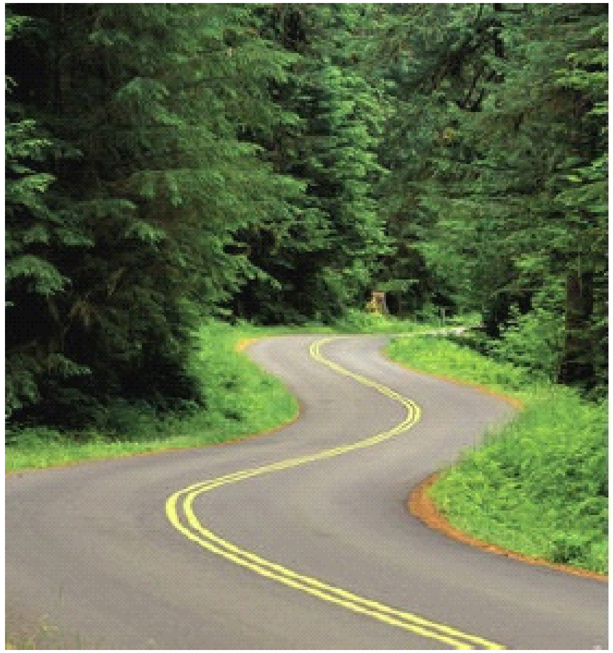
Cooperative Private Searching in Clouds

Jie Wu

Department of Computer and Information Sciences
Temple University

Road Map

- Cloud Computing Basics
  - Cloud Computing Security
  - Privacy vs. Performance
- Proposed Scheme
  - Trust Third Party (TTP)
  - Protocol Design
- Evaluation
  - Analytical study
  - Simulation study
- Extensions
- Some Challenges



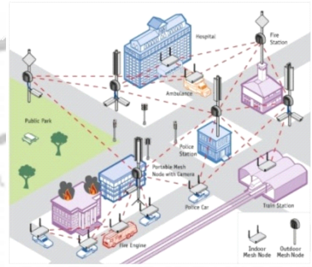
Center for Networked Computing

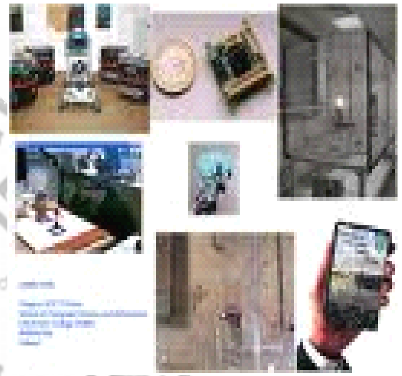
Jie Wu, Center Director
Computer and Information Sciences, Temple University

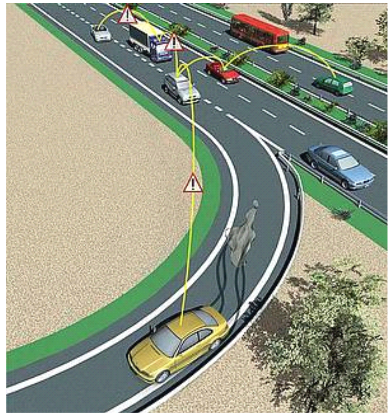
Networks: wireless and mobile

Internet (NSF GENI)

- Mesh networks (NSF MRI)
- Sensor networks (NSF NeTS)
- Delay-tolerant networks (NSF TC)
- Underwater networks (Navy Yard)
- Vehicular networks (SEPTA Regional Rail)





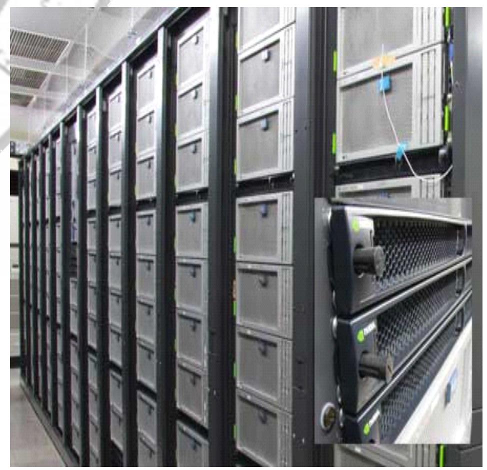


Network security and privacy

- Wireless networks (ARO)
- RFID and medical applications

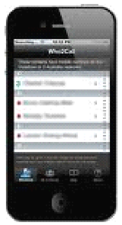
High performance computing

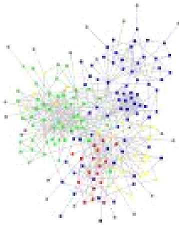
- NSF GPU/CPU supercomputer (MRI)



Social networks and cloud computing

- Online social networks (Amazon & NSF)

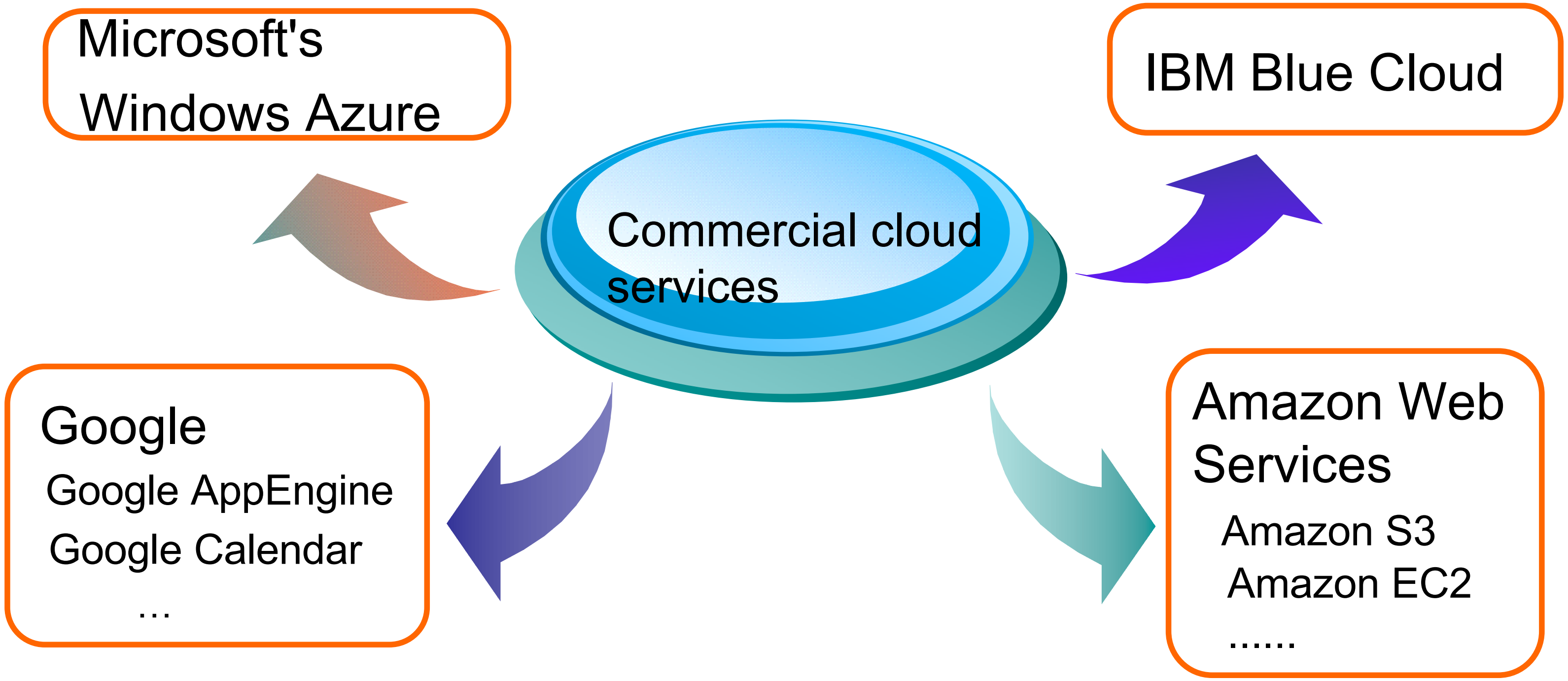




PSU

1. Cloud Computing Basics

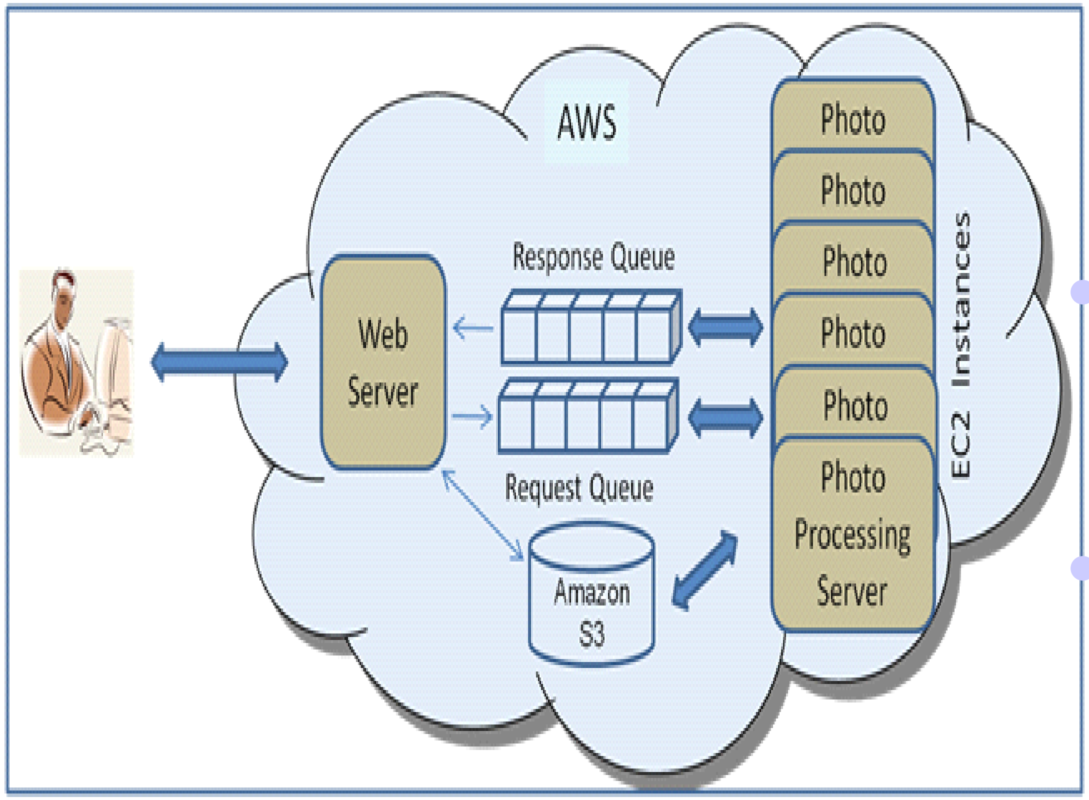
Cloud Computing Providers



Examples of Cloud Computing (I)

- Google Docs: Users can edit documents on-line.
- Google Calendar: Maintain a collaborated schedule with friends.
- Google Picasa: Share and process pictures on-line.
- Google AppEngine : Users can run web apps on Google's infrastructure

Examples of Cloud Computing (II)



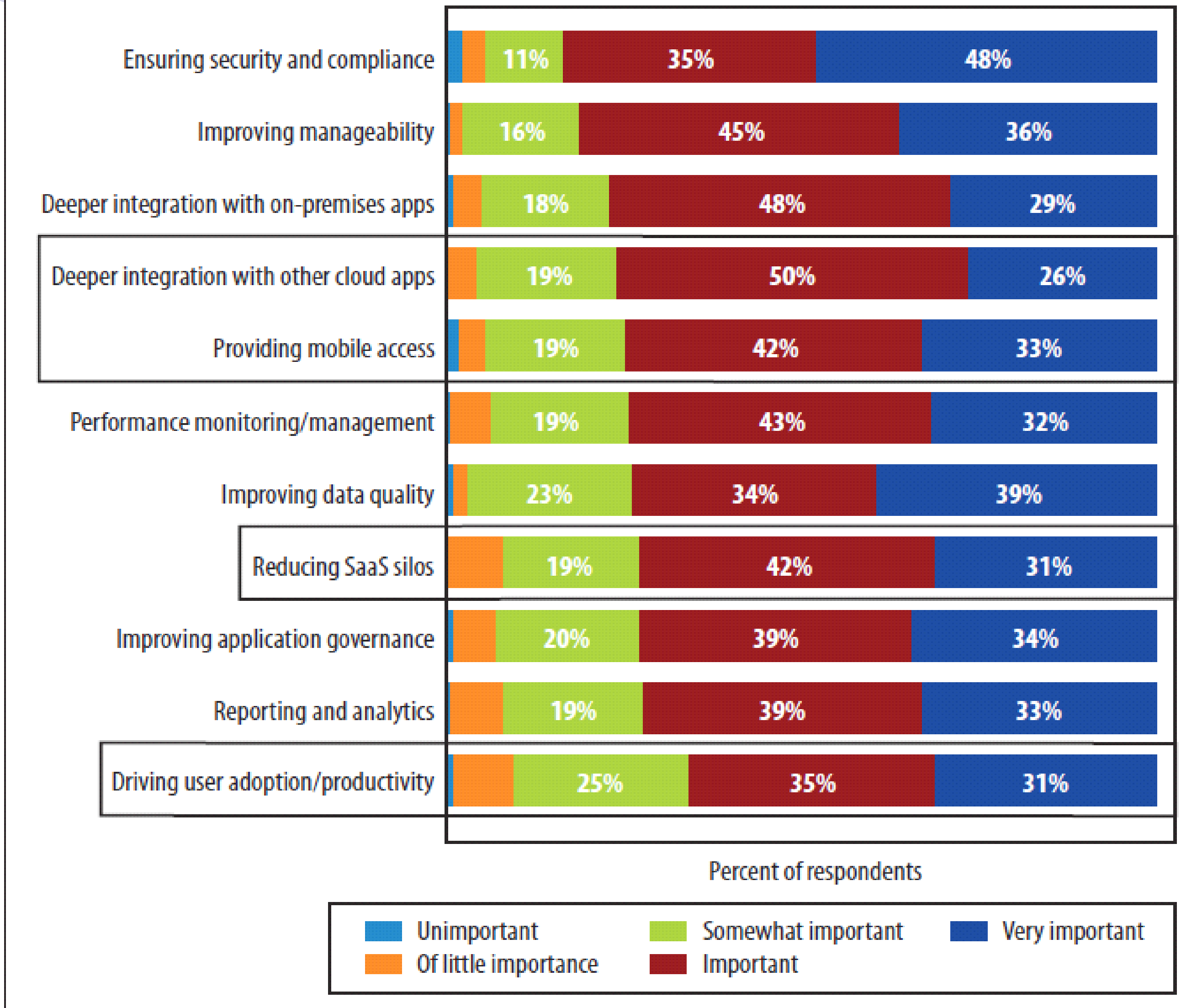
- Amazon Elastic Compute Cloud (EC2): Run applications or OS on Amazon's infrastructure.
- Amazon Simple Storage Service (S3): Store files on Amazon's storage servers

What is Cloud Computing?

- Key characteristics (NIST)
  - On-demand service.
  - Broad network access.
  - Resource pooling.
  - Rapid elasticity.
  - Metered service.
  - Encompasses many different technologies.

Cloud computing		
Thin clients	Utility computing	Grid computing

Cloud adopters' priorities



IEEE Computer 2011 Vol 44 Issue3

Cloud Computing Security

- Different types of security
  - Insider attacks, vulnerabilities for virtualization, data loss, data leakage, vendor lock-in, and privacy.
- Conventional wisdom is to design cryptographic solutions to achieve security.
  - But there are other considerations.
- A secure cloud solution with very high overhead will not be practical !

Analogy to Streaming Multimedia

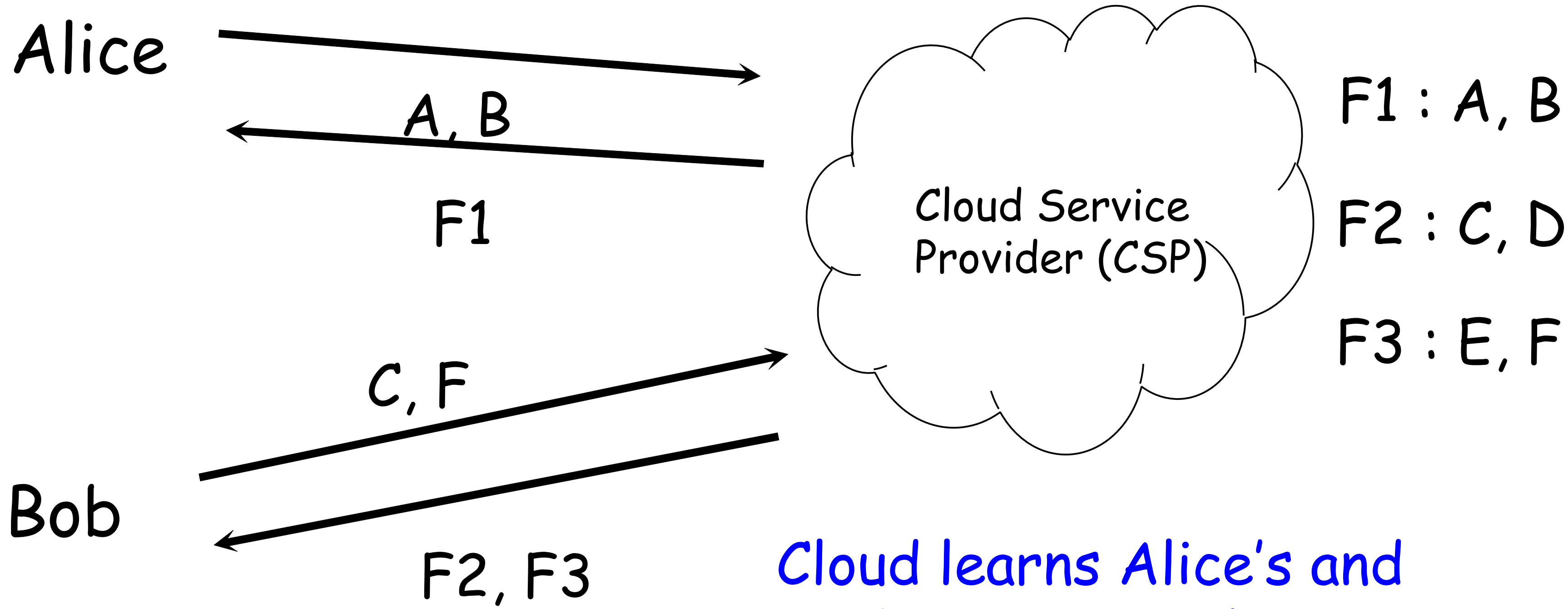
- How to improve the user's experience of watching multimedia content ?
- Improving Internet bandwidth will improve streaming performance.
- But that is not the only approach.
  - Better buffering algorithms.
  - Improving data compression and coding.
  - Deploying web caches.
  - ……

Privacy and Performance

- How to protect user privacy ....
  - defending against insider attacks, data leakage, etc.
- ..... while ensuring good systems performance.
  - includes conserving bandwidth, reducing energy through minimizing computation and communication (green clouds).

2. Proposed Scheme

Example: Users Querying the Cloud



Cloud learns Alice's and Bob's queries and responses.

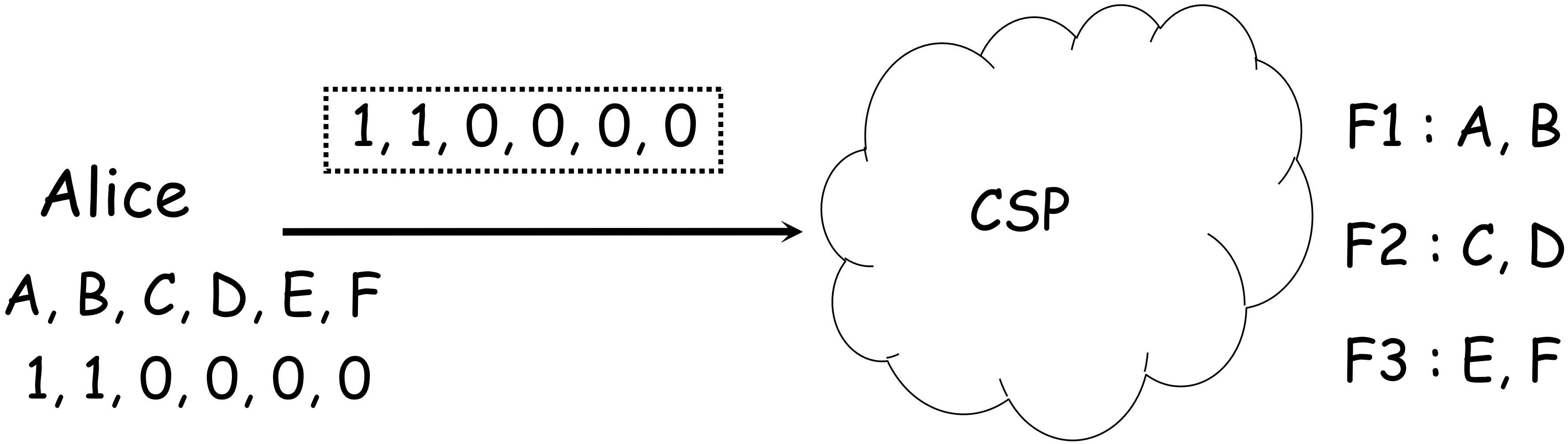
Search and Access Privacy

Cloud neither learns what the user is searching For, nor which files are returned to a user.

Cloud: honest but curious.

It will obey general rules, but still wants to know some additional information.

Prior Solution (CRYPTO 2005)

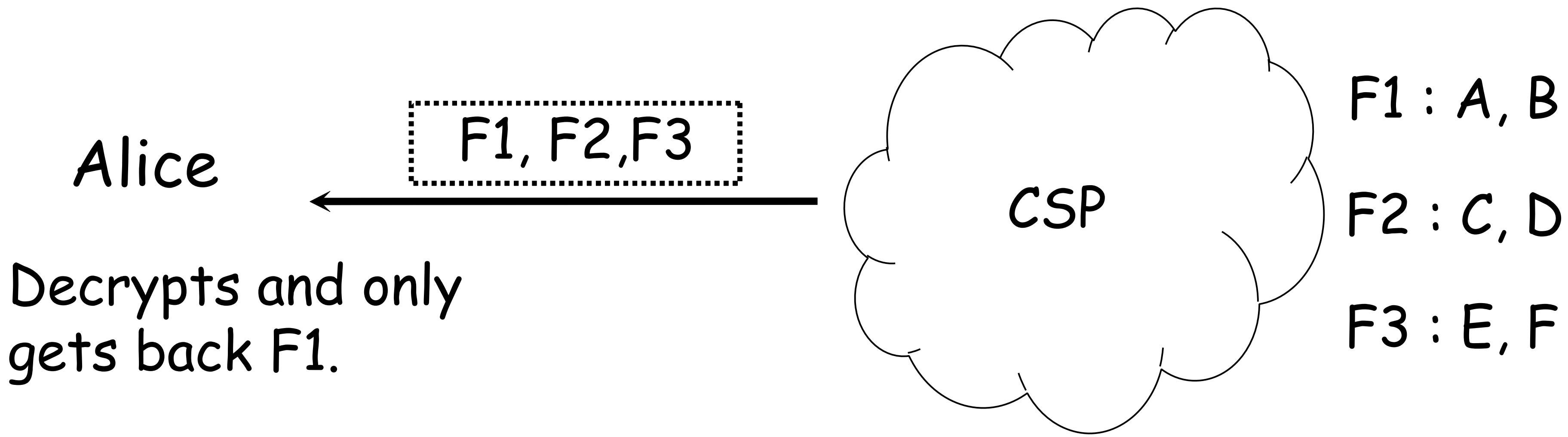


Alice issues a special query 1, 1, 0, 0, 0, 0

with a dictionary of d slots encrypted using homomorphic encryption (processing encrypted files without decryption).

Cloud compares all files and returns all.

Prior Solution (CRYPTO 2005)



Cloud computes and returns to Alice F1, F2,F3

F1, F2,F3 is a compressed version of F1,F2,F3.

Cloud does not know what files are returned.

Brief Background

*Homomorphic encryption allows us to perform some operations on encrypted data **without** decryption.*

Let E() be encryption.

- $E(x)*E(y) = E(x+y)$ where $E(x) = f(g^x)$ (Paillier system)
- $E(X)^Y = E(X * Y)$

key trick: map unwanted file F to 0

- $E(0)^{|F|}=E(0*|F|) = E(0)$
- Users encrypt interests in E(0) or E(1)

Returned files can be easily compressed without conflict, as all unwanted files are now E(0)

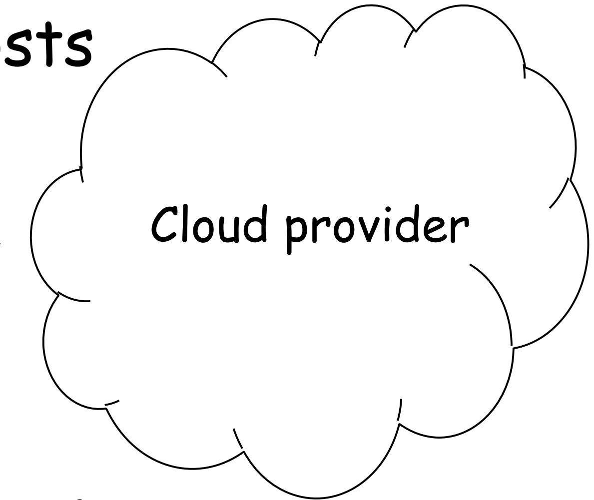
Key Points

- Prior research solves the privacy problem.
- But the overhead is high
  - High bandwidth and computation costs

1, 1, 1, 0, 0, 1

Alice 1, 1, 0, 0, 0, 0

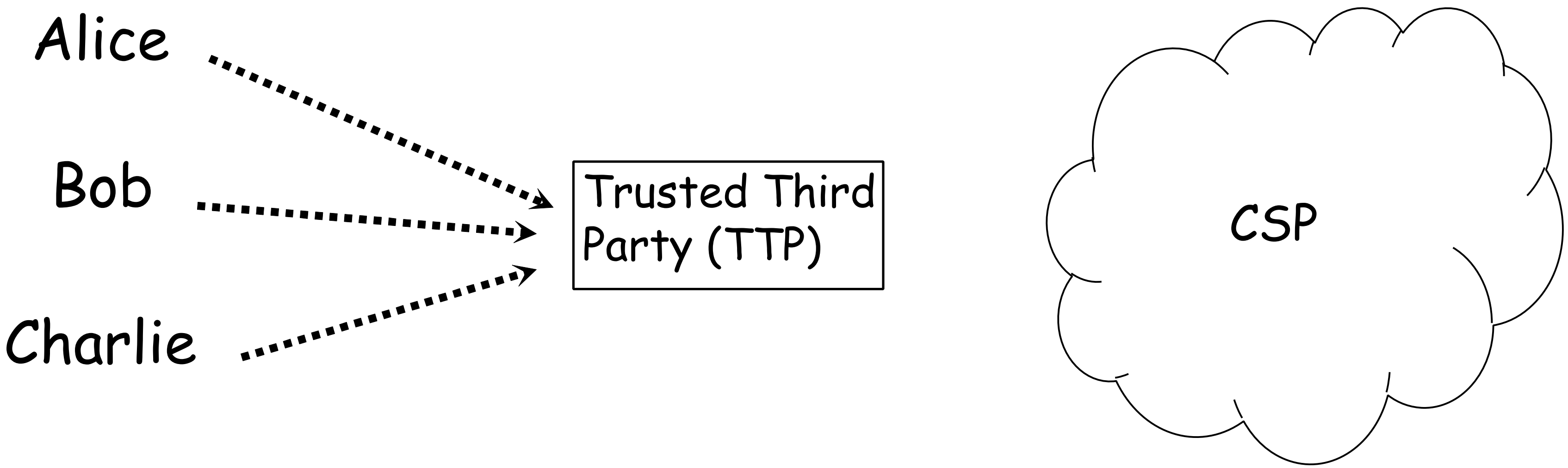
Bob 0, 0, 1, 0, 0, 1



We can combine queries

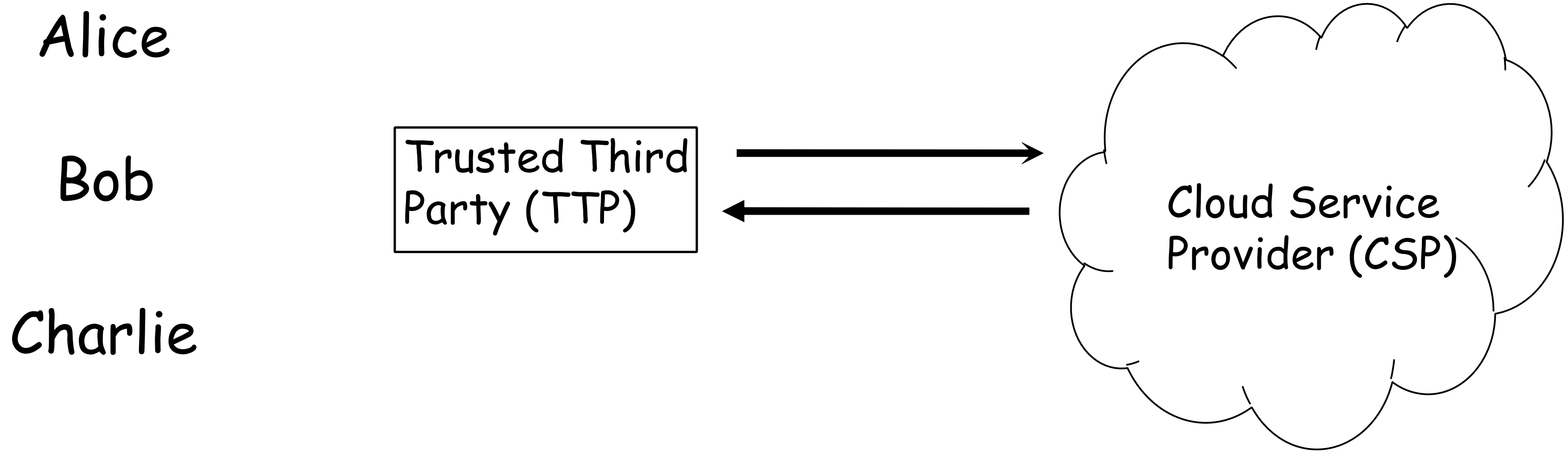
- computation: 1 computation for n users
- bandwidth: d (dictionary size) and 2f log (f/p) (from cloud), where f (# of returned files) depends on # of keywords and their distribution and 2 log (f/p) is the redundancy level given a tolerable failure rate p

Trusted Third Party (TTP)



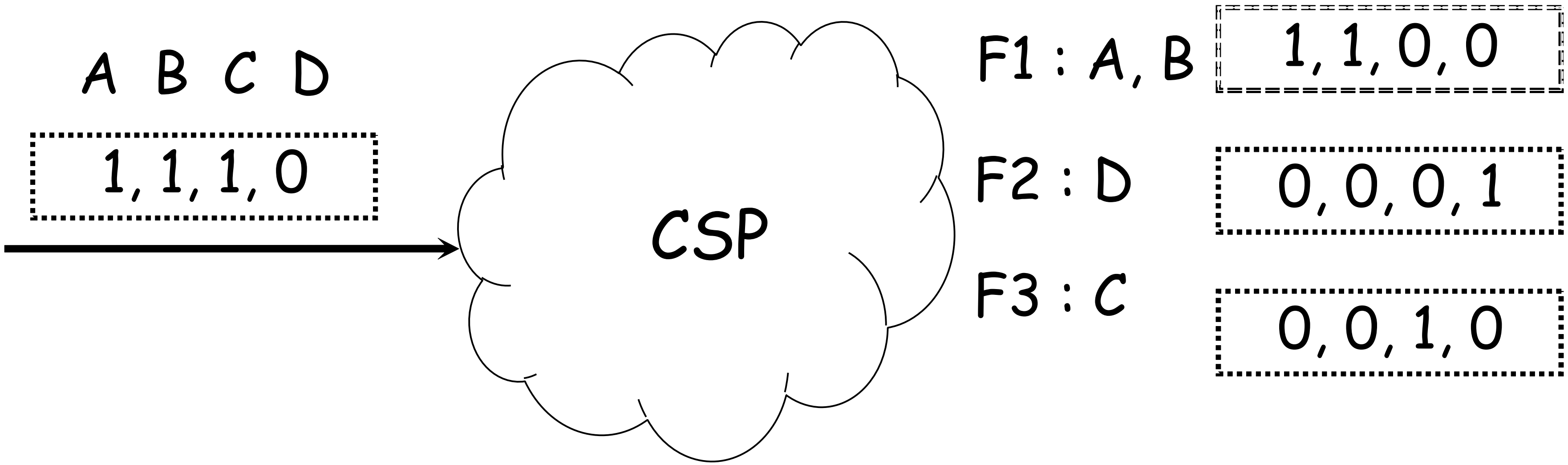
- Deploy a Trusted Third Party (TTP), but we want to limit the amount of trust.
- Users forward their queries to TTP.
  - How to combine queries?
  - How to prevent TTP from learning everybody's queries?

Conceptually



- TTP aggregates queries and sends them to the cloud. The cloud returns answers to TTP.
- TTP looks like a "single" user.
  - How does the cloud compute aggregated query?

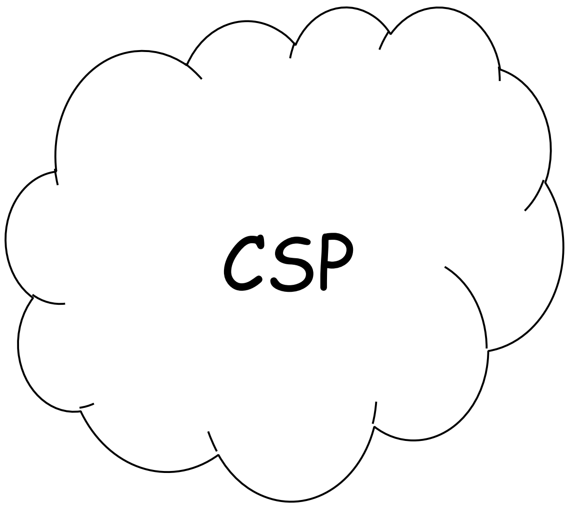
Cloud Processing Query



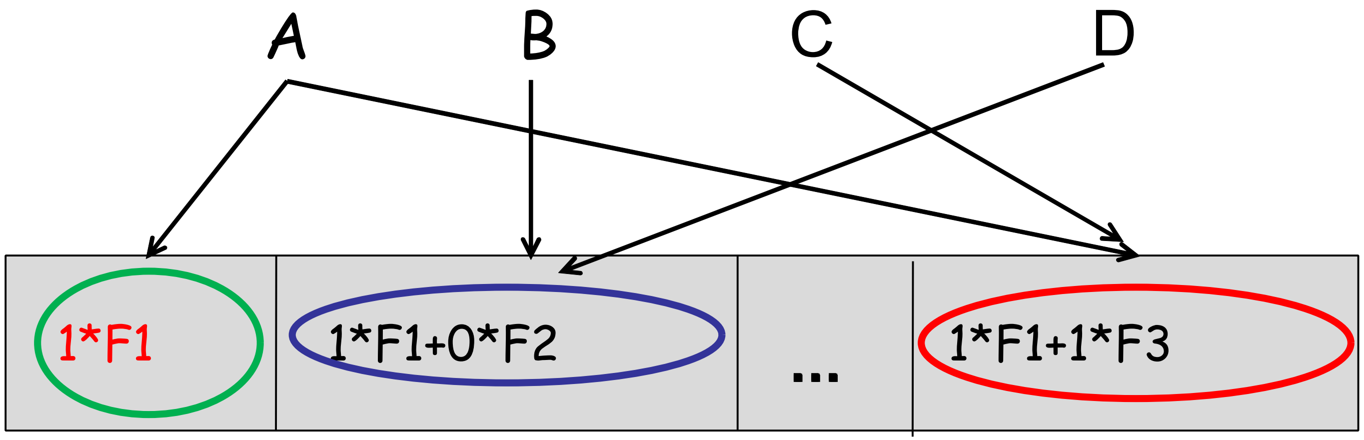
For each file, the scalar product E(c) is computed, and the file content, |F|, is then powered by E(C), i.e., $E(c)^{|F|} = E(c*|F|)$.

F1 : (2, 2*|F1|), F2 : (0, 0*|F2|), F3 : (1, 1*|F3|)

Cloud Preparing Response



Cloud returns 2 buffers to the TTP.
A file name (e.g. F1) buffer



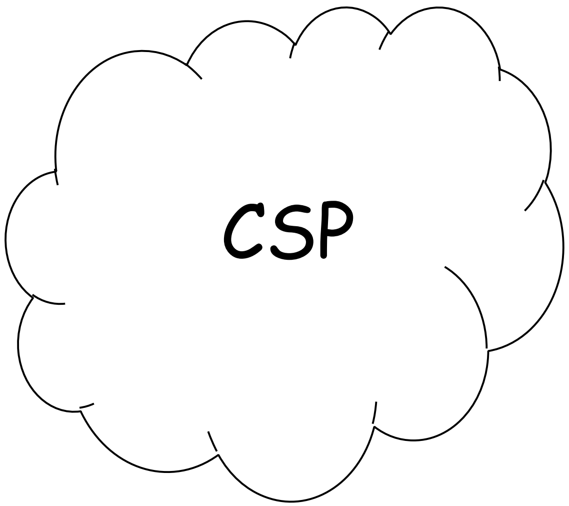
File name buffer

No collision

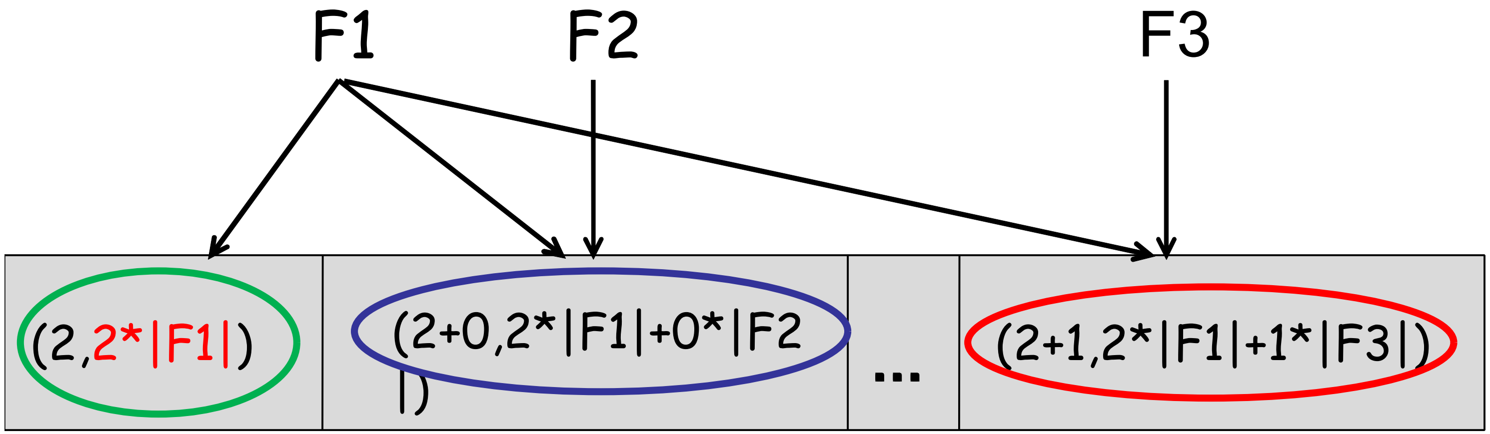
Collision with 0
Still can recover
e.g. 1*F1+0*F2|= 1*F1

Collision
Cannot recover

Cloud preparing response



A file content buffer (e.g. |F1|).



File content buffer

No collision

Collision with 0
Still can recover
e.g. 2*|F1|+0*|F2| = 2*|F1|

Collision
Cannot recover

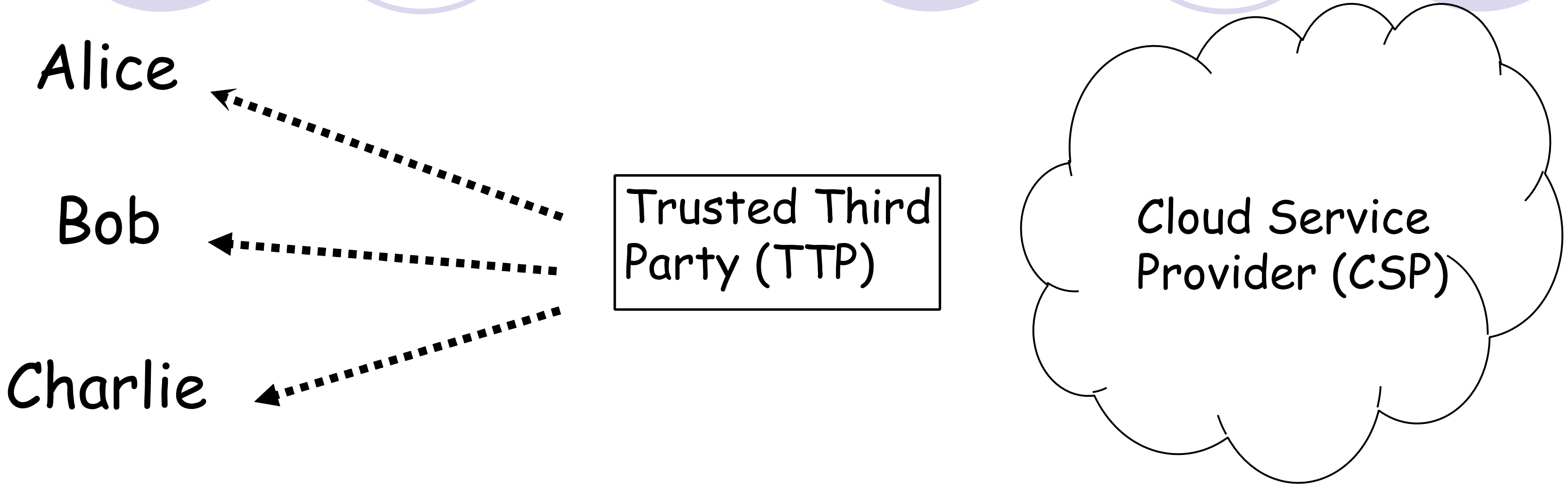
Size of Return Buffers

- If the user wants to retrieve f files with failure probability p, then each file should be thrown r times into a buffer of size $2rf$, s.t.

  $r = \log(f/p)$.

- To retrieve K keywords and f files with failure probability p, we should construct two buffers with size $2K \cdot \log(K/p)$ and $2f\log(f/p)$, respectively.

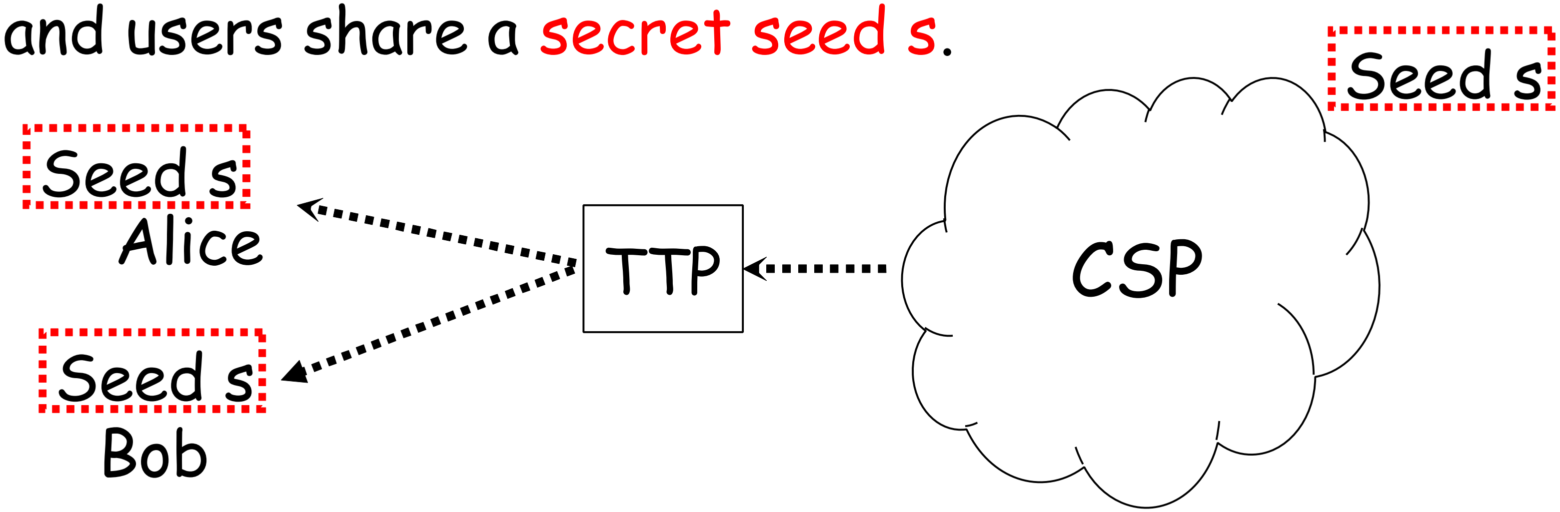
Conceptually



- TTP processes the cloud response and returns answers to users.
  - How to efficiently return answers to users?
  - How to prevent TTP from learning new information?

TTP Returns Results

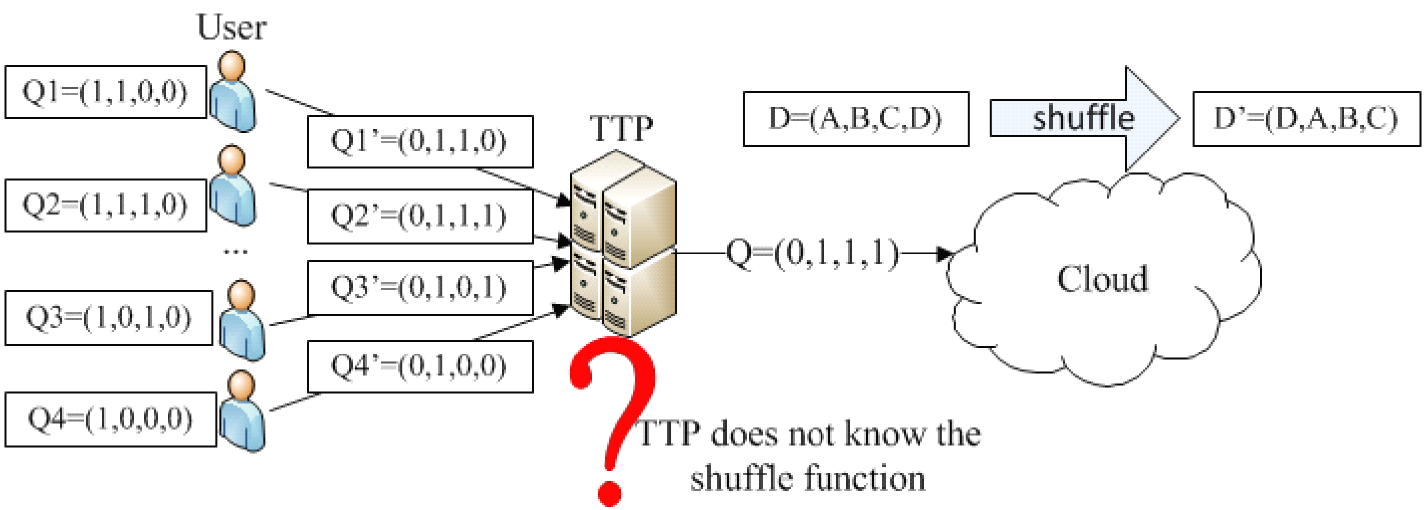
- TTP has to decrypt the file content buffer to return file contents to users.
- To prevent TTP from learning file content, cloud and users share a secret seed s.



- Cloud generates random number (R) for each file (F)
- Cloud will XOR each file content: |F| XOR R.
- Users can recover |F| by computing R with shared s.

Shuffle for Limited Trusted TTP

- Protocol lets each user shuffle their query with a shuffle function to prevent the cloud from learning a users' query.
- The shuffle function is known to users and the cloud, but not the TTP.



3. Evaluation: Analytical Study

d: number of query terms, t: total number of files in cloud, and f: number of returned files.

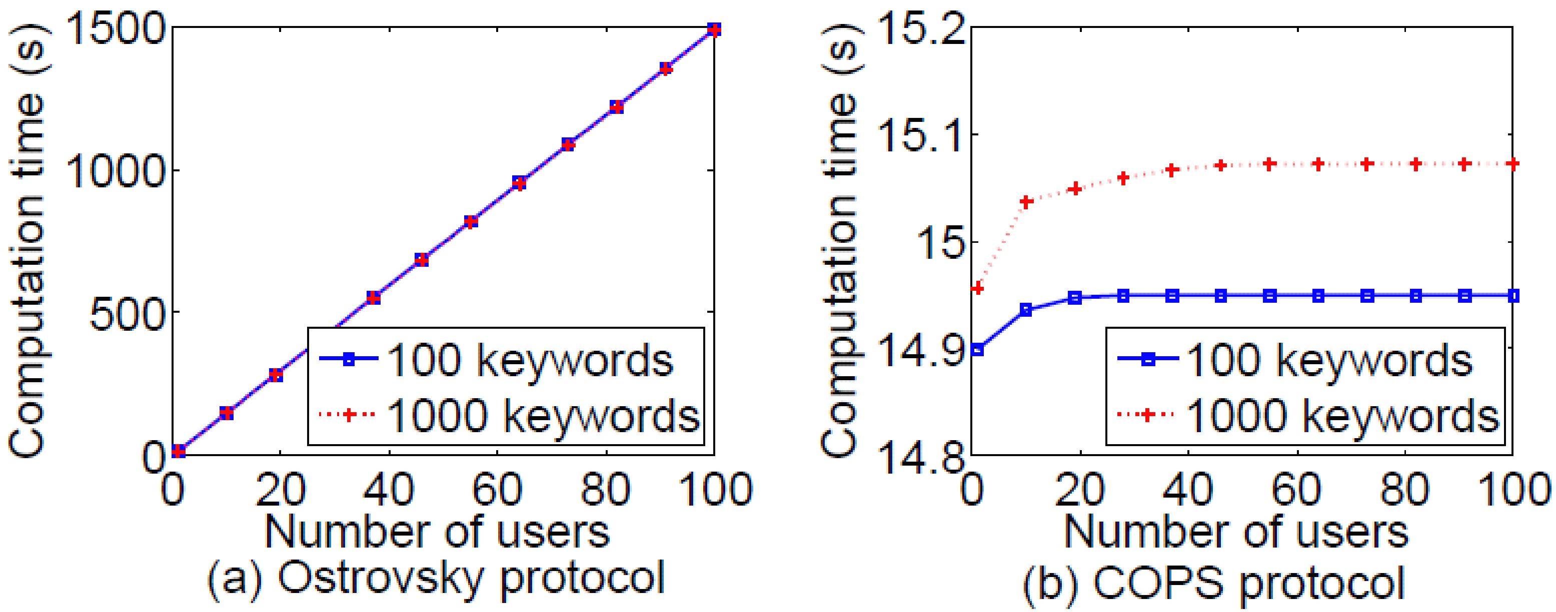
- Ignore TTP bandwidth and TTP computation costs.
- $f = t\,(1 - (1-q)^K)$, where q is the probability a keyword appears in a file.

	1 user	n users (previous)	n users (our sol.)
Transmission cost (from users to cloud)	O(d)	O(n *d)	O(d)
Transmission cost (from cloud to users)	O(f log (f/p))	O(n* f log (f/p))	O(c*flog (f/p)) $1 < c \leq n$
Computation cost (within the cloud)	O(t)	O(n*t)	O(t)

Simulation Parameters

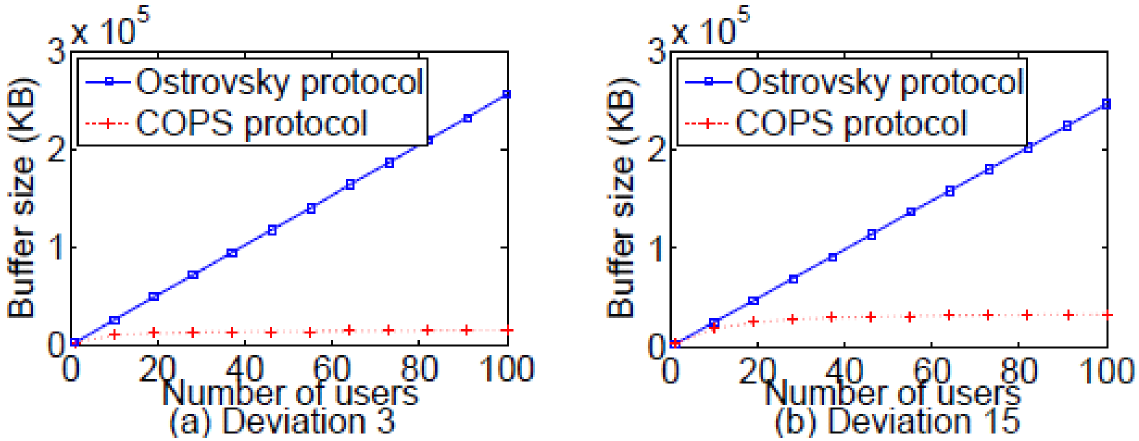
Notation	Description	Value
$\|F\|$	File content	$1KB$
$\|w\|$	Keyword content	$1KB$
n	Number of users	1-100
d	Number of keywords in the dictionary	100-1,000
k	Number of keywords in each query	1-5
l	Number of keywords in each file	1-5
t	Number of files stored in the cloud	10^3
p	Failure probability	0.1

Evaluation Results (I)

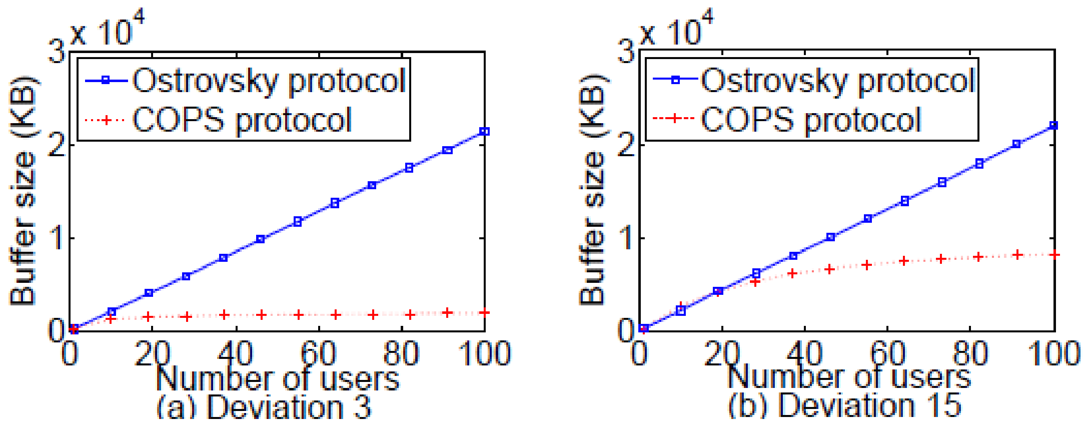


Dictionary contains 100-1,000 keywords.
Each file contains 1-5 keywords
With 1024-bit keys, multiplications and exponentiations take 6.3us and 14.7 ms

Evaluation Results (II)

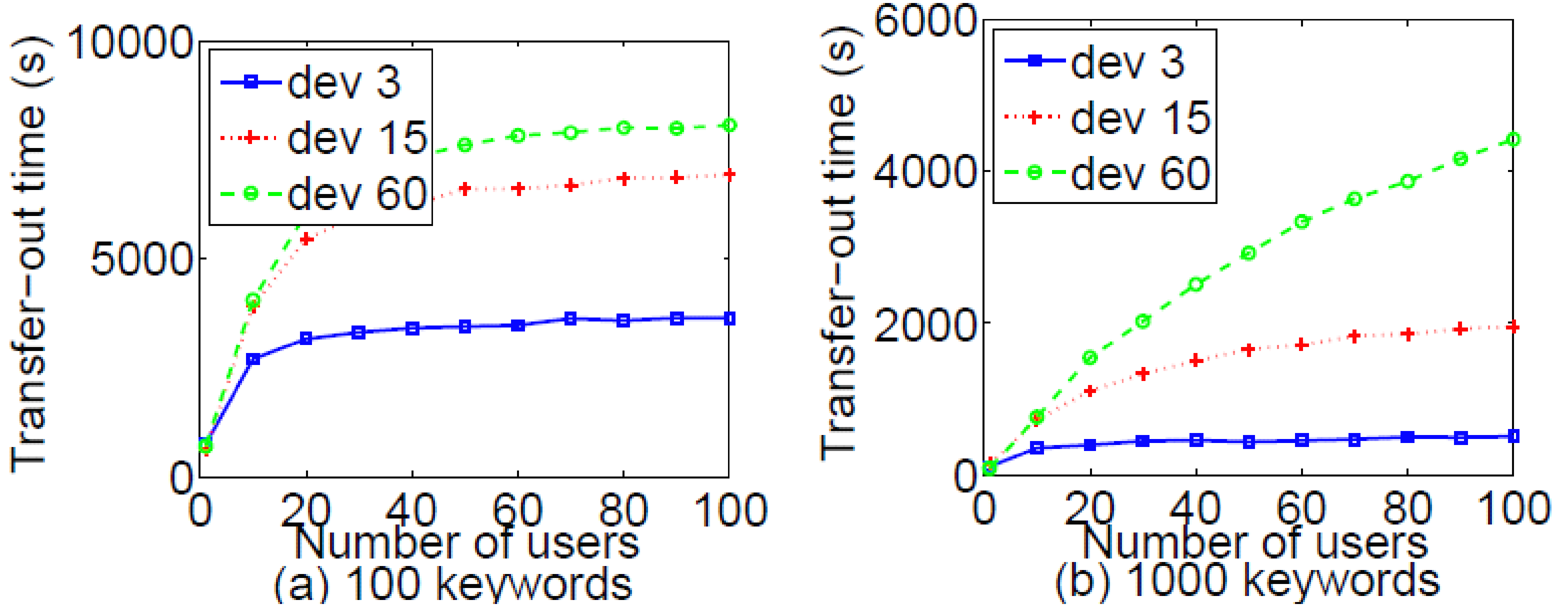


Dictionary contains *100* keywords. Keyword distribution follows normal distribution, with different deviations (dev). The higher the dev, the less common the keywords are among users.



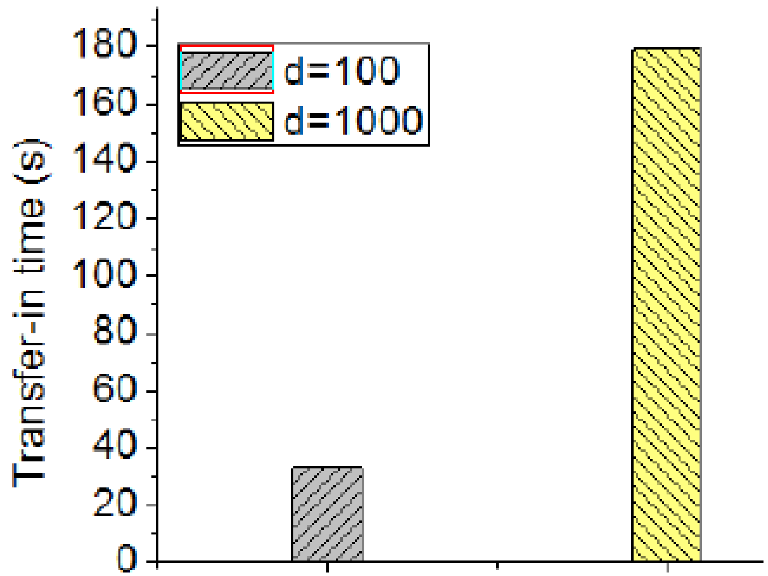
Dictionary contains *1,000* keywords.

Evaluation results (III)



Transfer-out a buffer from the cloud to the TTP

Evaluation results (IV)



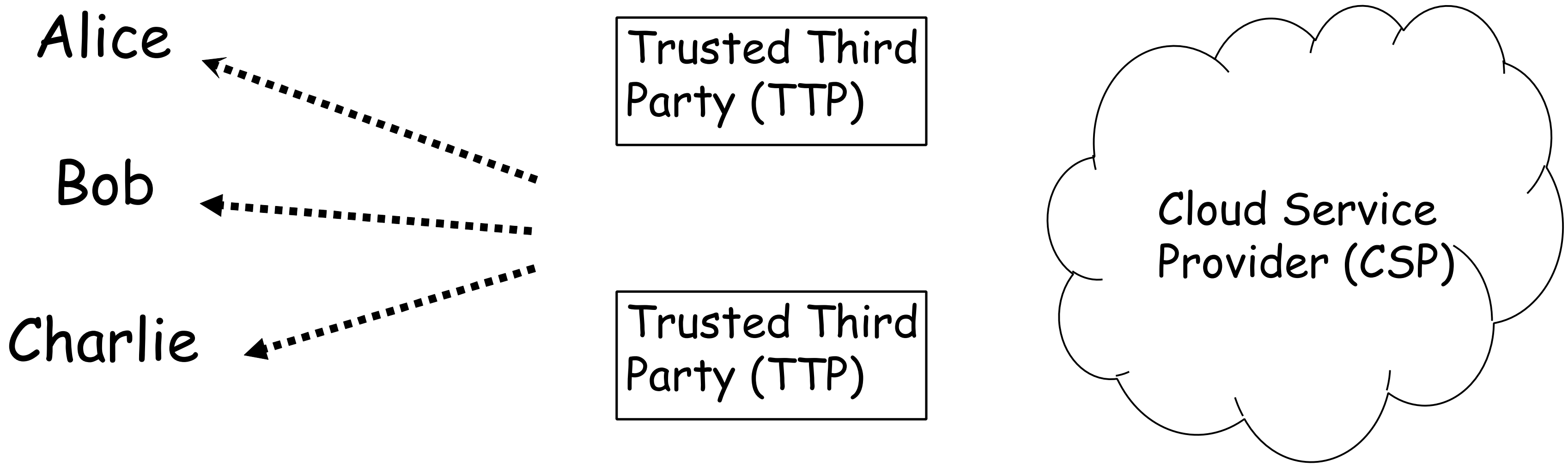
Transfer-in a dictionary from the TTP to the cloud

4. Extensions

- Ranked Queries: differential service
  - Cloud returns a certain percentage of matched files for a given rank.
  - Rank privacy: cloud provides differential query service without knowing which level of service is chosen by the user.
  - Mask matrix: allows the cloud to filter out a certain percentage of matched files.

Extensions (Cont'd)

- Multiple TTPs: resolve bottleneck at TTP



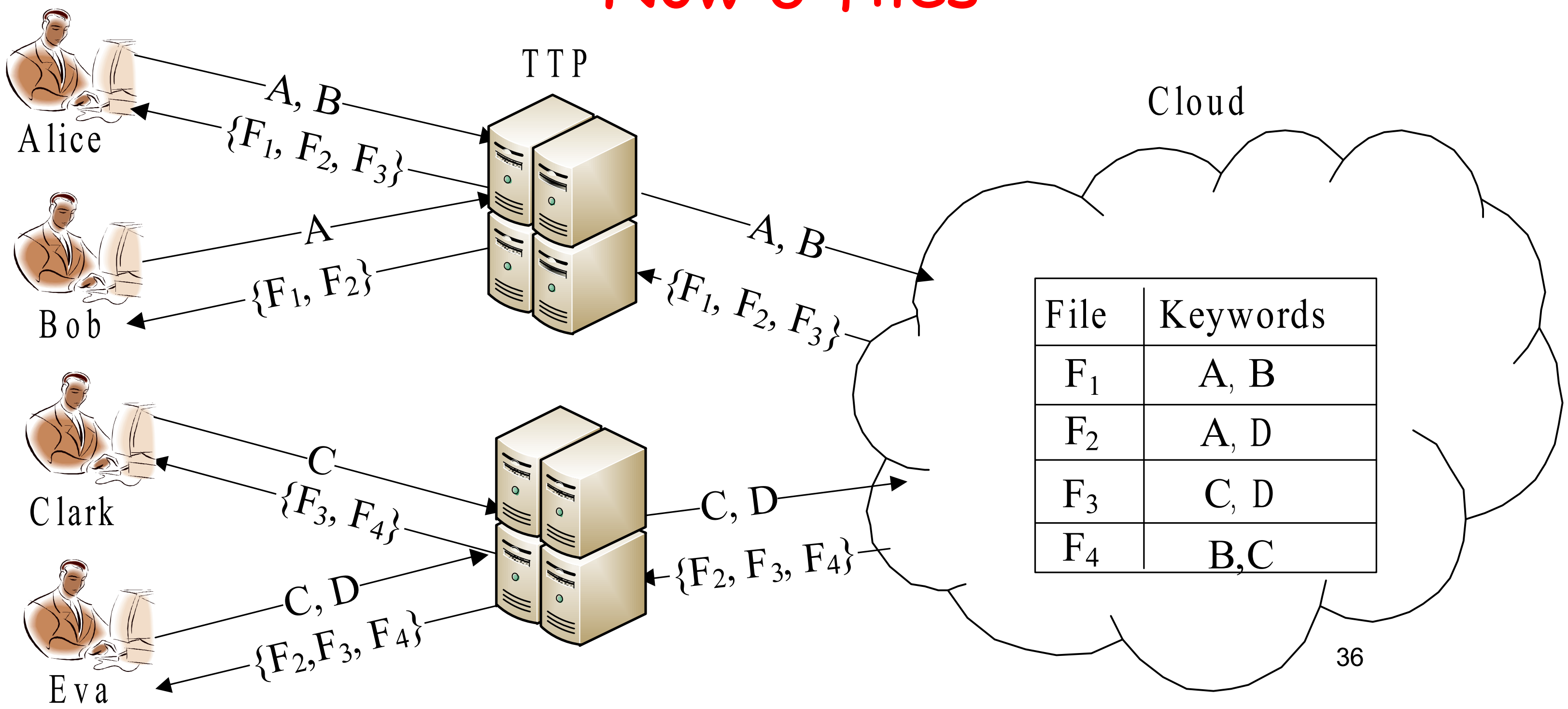
Extensions (Cont'd)

- Cost Efficiency
  - For a given group #, group users with overlapping keywords to minimize # of 1s in each group have the same size.
- Load Balancing
  - For a given U, create a minimum # of groups such that 1s in each group are bounded by U.
- Robustness
  - For a given K, use one of the grouping criteria in such a way that each query appears in at least K different groups.

4. Multiple TTPs: Grouping

- If Alice and Clark are in a group, and Bob and Eva are in another group, the cloud needs to return 8 files.

Now 6 files



Problem formulation

- Classifying n users into k groups, so that the number of keywords in k combined queries, i.e., the total number of 1s, is minimized.

- Basic idea: K-Mean-based Dynamic Grouping
- Choose k queries as the seeds, and classify the queries that are closest to the seed into a group.

K-Mean-based Dynamic Grouping

$Q_1 = \langle 11100000 \rangle$	$Q_5 = \langle 00000111 \rangle$
$Q_2 = \langle 11000000 \rangle$	$Q_6 = \langle 00000011 \rangle$
$Q_3 = \langle 11000000 \rangle$	$Q_7 = \langle 00000011 \rangle$
$Q_4 = \langle 00010000 \rangle$	$Q_8 = \langle 00001000 \rangle$

P1: Random

P4: Random Robust

KMDG: P2

- Balance group size

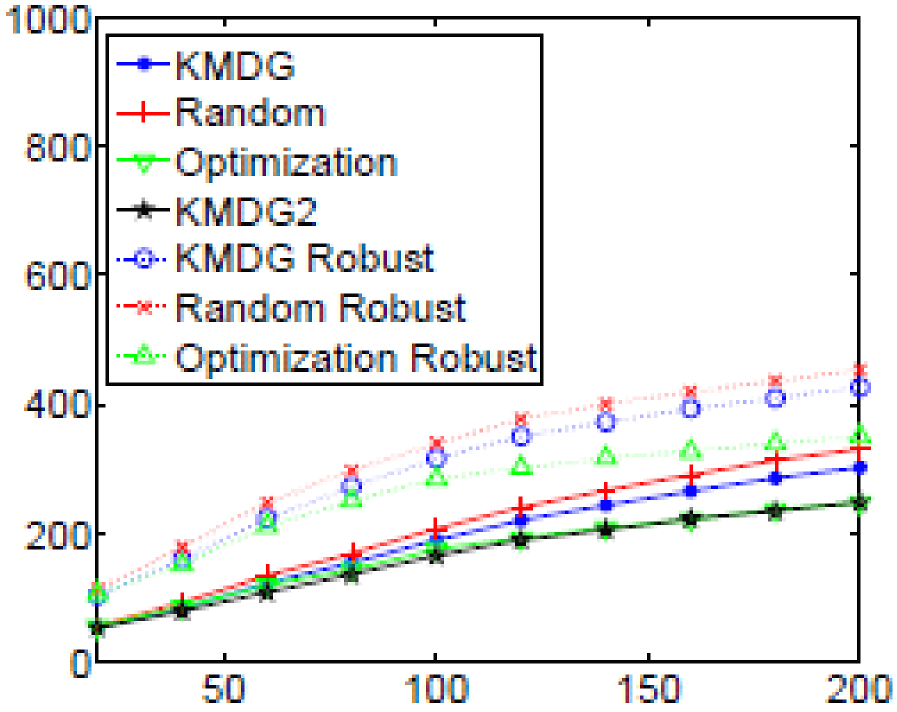
KMDG2: P3

- Balance # of 1s

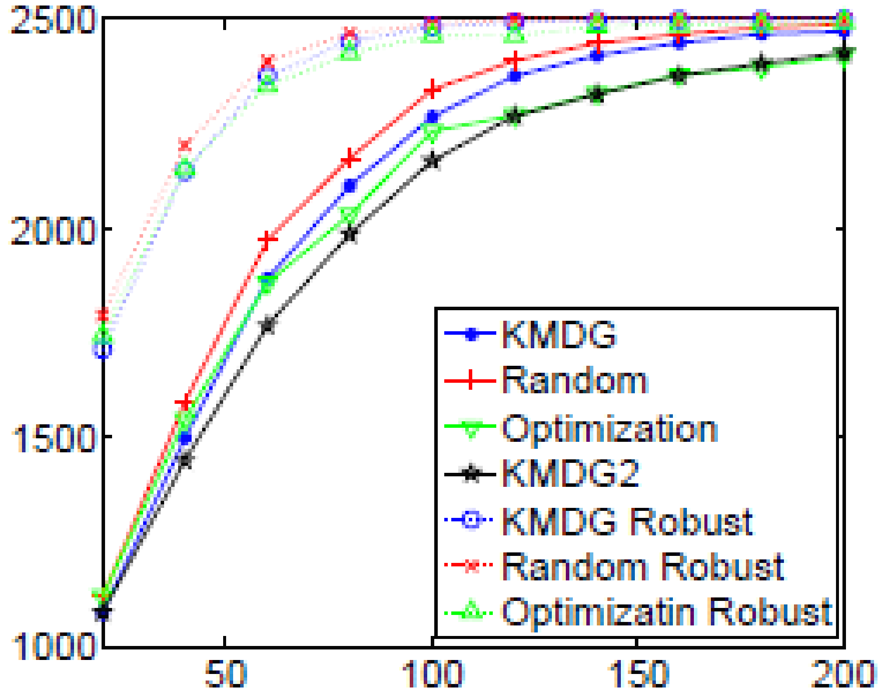
KMDG Robust: P5

	g_1	g_2	g_3	g_4
P1	Q_1 11100000 Q_5 00000111	Q_2 11000000 Q_6 00000011	Q_3 11000000 Q_7 00000011	Q_4 00010000 Q_8 00001000
P2	Q_1 11100000 Q_4 00010000	Q_2 11000000 Q_3 11000000	Q_6 00000011 Q_7 00000011	Q_5 00000111 Q_8 00001000
P3	Q_1 11100000	Q_2 11000000 Q_3 11000000 Q_4 00010000	Q_5 00000111	Q_6 00000011 Q_7 00000011 Q_8 00001000
P4	Q_1 11100000 Q_5 00000111 Q_2 11000000 Q_6 00000011	Q_3 11000000 Q_7 00000011 Q_4 00010000 Q_8 00001000	Q_1 11100000 Q_5 00000111 Q_2 11000000 Q_6 00000011	Q_3 11000000 Q_7 00000011 Q_4 00010000 Q_8 00001000
P5	Q_1 11100000 Q_4 00010000 Q_2 11000000 Q_3 11000000	Q_6 00000011 Q_7 00000011 Q_5 00000111 Q_8 00001000	Q_1 11100000 Q_4 00010000 Q_2 11000000 Q_3 11000000	Q_6 00000011 Q_7 00000011 Q_5 00000111 Q_8 00001000

Experiment results



The total number of 1



Bandwidth (MB)

X axis: number of users

K=5 and dictionary size 100

5. Future Trends and Challenges

Increasing popularity of multicloud environments.

- How do we select which cloud should perform a particular task?

Convergence of mobile and cloud computing.

- Increase functionality of smartphones by outsourcing data and computation to the cloud.

Viable cost/price model.

- Workable model between CSP and users.

Thank you

My Research Team

Qin Liu

Dr. Chiu C. Tan