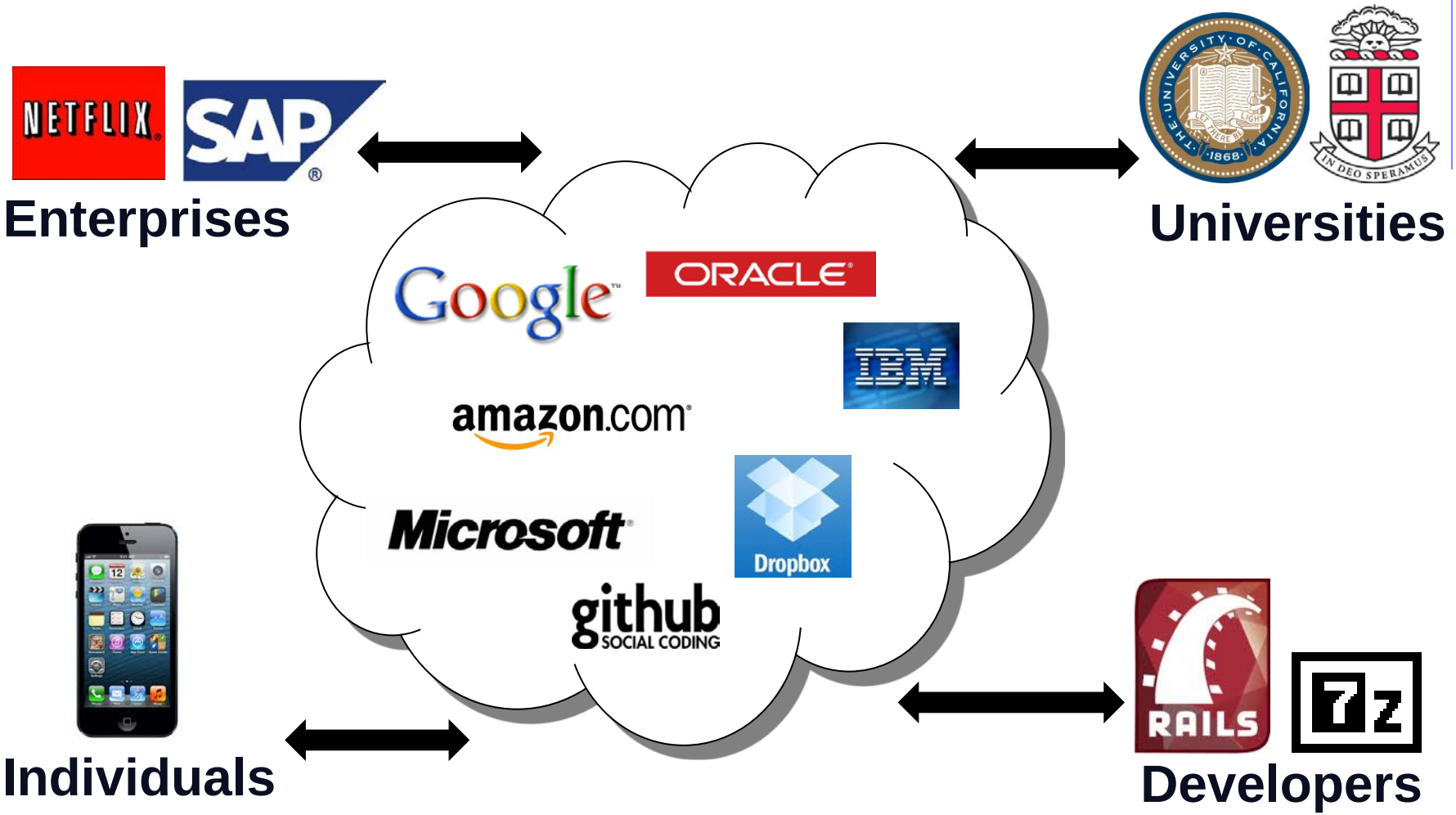
Trustworthy and Private Computing in the Cloud

Charalampos (Babis) Papamanthou
University of Maryland
cpap@umd.edu

Cloud computing today

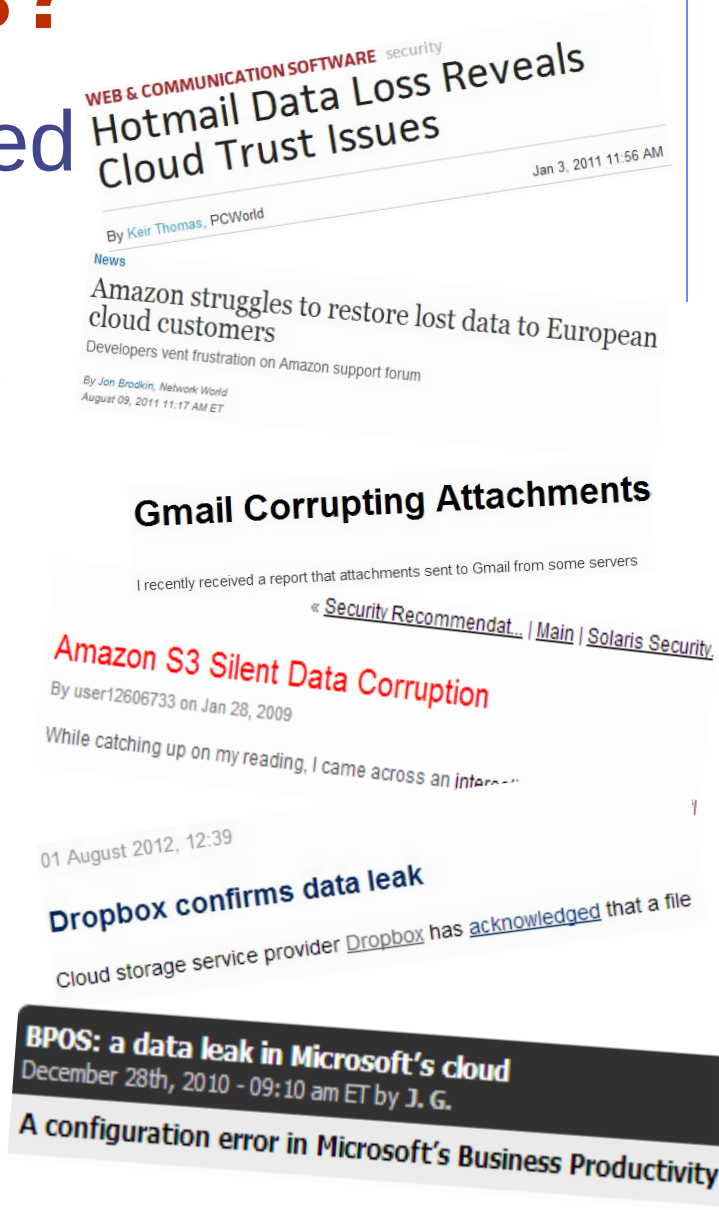


Are there any threats?

- Cloud providers are untrusted
  - Can **lose** data
  - Can return **corrupted** results
  - Can **leak** information

…we will have no liability to you for any unauthorized access or use, corruption, deletion, destruction or loss of any of your content or applications…

Amazon web service customer agreement
`http://aws.amazon.com/agreement/`



Do people care?

- Customers are paying for the services
  - They want **reliable** storage
  - They want **correctness** guarantees
  - They want to keep their **privacy**

*…**58%** of the public and **86%** of business leaders are excited about the possibilities of cloud computing. But more than **90%** of them are worried about security, availability, and privacy of their data as it rests in the cloud…*

Microsoft survey in 2010
`http://news.cnet.com/8301-1009_3-10437844-83.html`

What do I do?

- **Enable people to use the cloud safely**
- **Verifiability in the cloud**
  - **Verify** that the cloud did the work correctly
- **Privacy in the cloud**
  - Use the cloud in a **privacy-preserving** manner

efficient both in theory and in practice

provably secure

no assumptions at the server

My work

practicality

fast crypto

Dropbox

amazon web services™

my work

popular cloud applications

slow crypto

key/value store

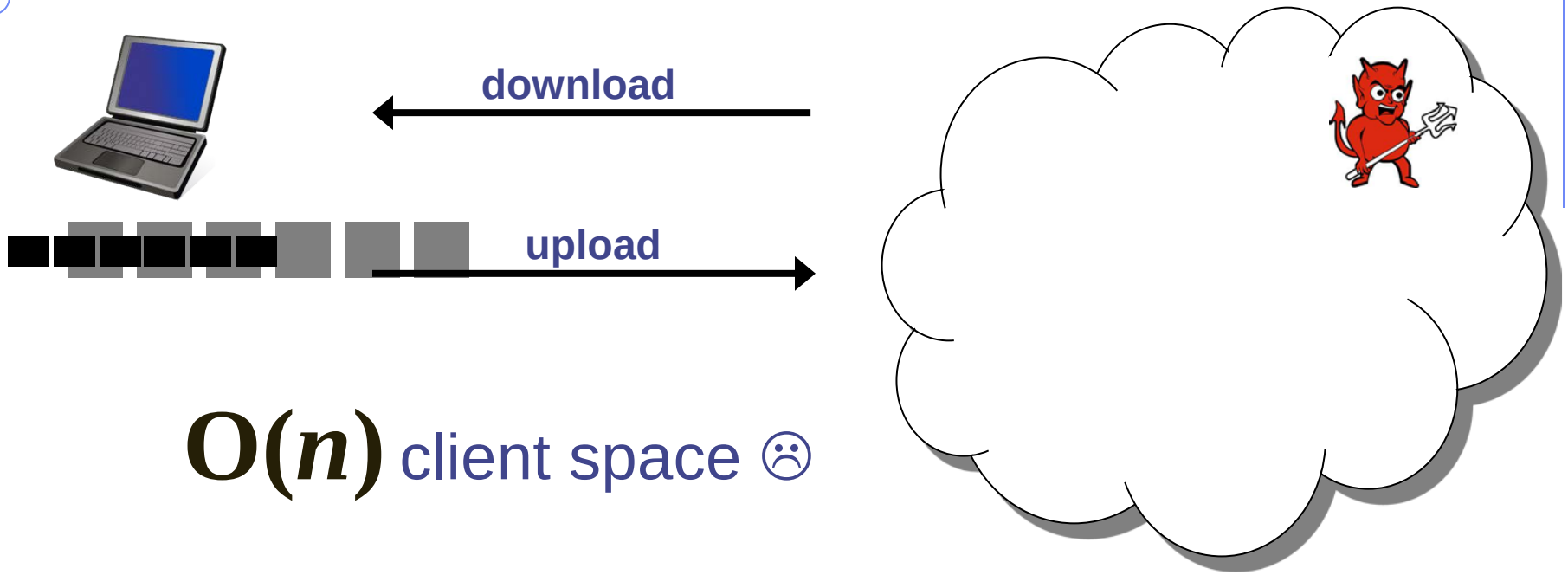
any application

expressiveness

Roadmap

- **Motivation**
- **Verifiability in the cloud**
  - Storage applications [STORAGESS08, CCS09, CCS13]
  - Information retrieval applications [CRYPTO11, VLDB12]
- **Privacy in the cloud**
  - Searching encrypted data [CCS12, FC13]
- **Research agenda and vision**

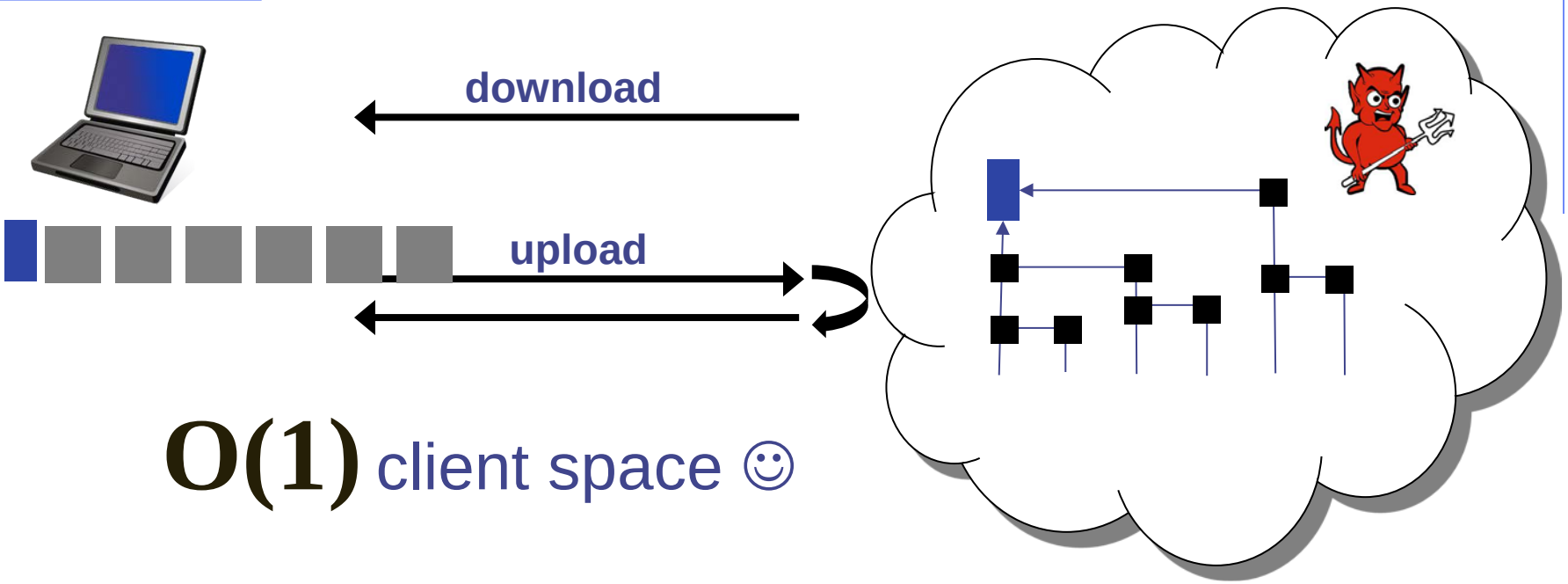
Storing your files in the cloud



$O(n)$ client space ☹

- How to verify that a file has **not** been corrupted?
  - Keep a hash (i.e., checksum) **locally** for each file
  - Download: **recompute** and **check**
  - Upload: **compute** and **store** new hash

A more efficient protocol

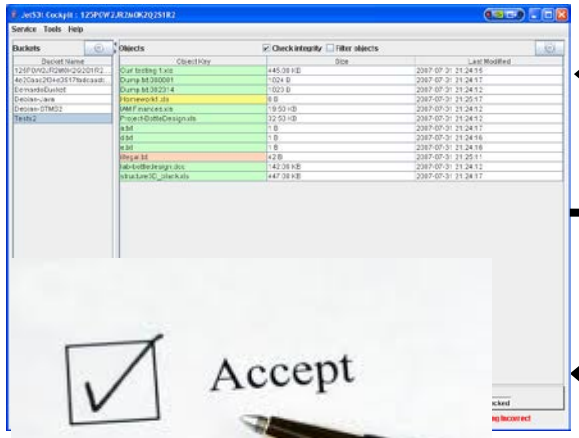


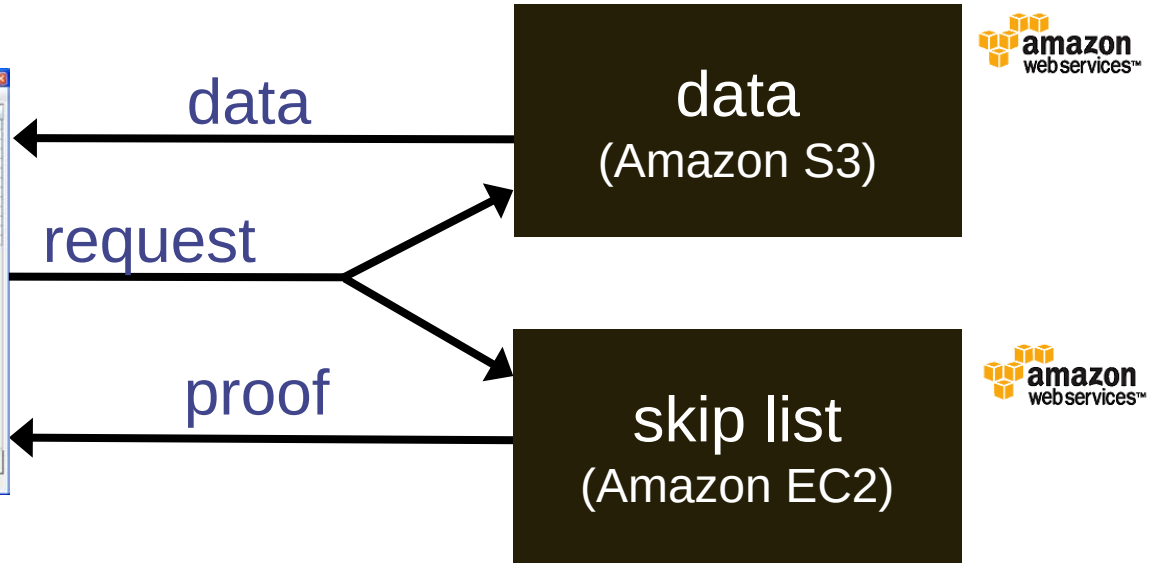
O(1) client space ☺

- Hashing over a skip list and store only the roothash
  - Download: Fetch $\mathbf{O(\log n)}$ hashes
  - Upload: An **interactive** protocol
- Technical contributions
  - **Parallel** $\mathbf{O(\log n)}$ update complexity

Implementation with Amazon S3

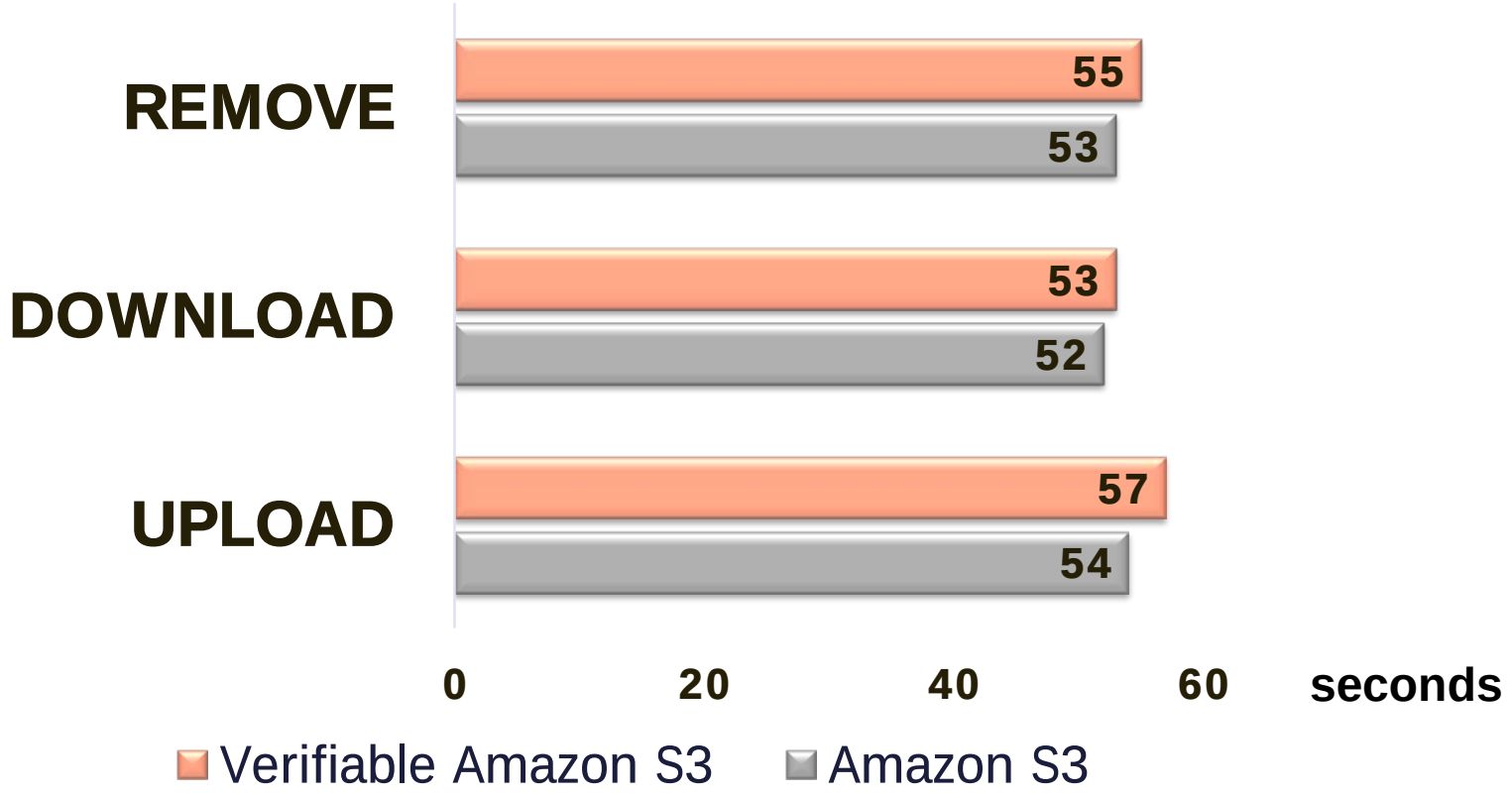
- Extended an S3 client app (`JetS3t Cockpit`)
- Used Amazon EC2 and Amazon S3





Performance

- Stored 80,000 1KB files on Amazon S3
- Used 1,000 1KB files as workload
- **Negligible** overhead



Other considerations on storage

How can you verify **all the files more efficiently**?

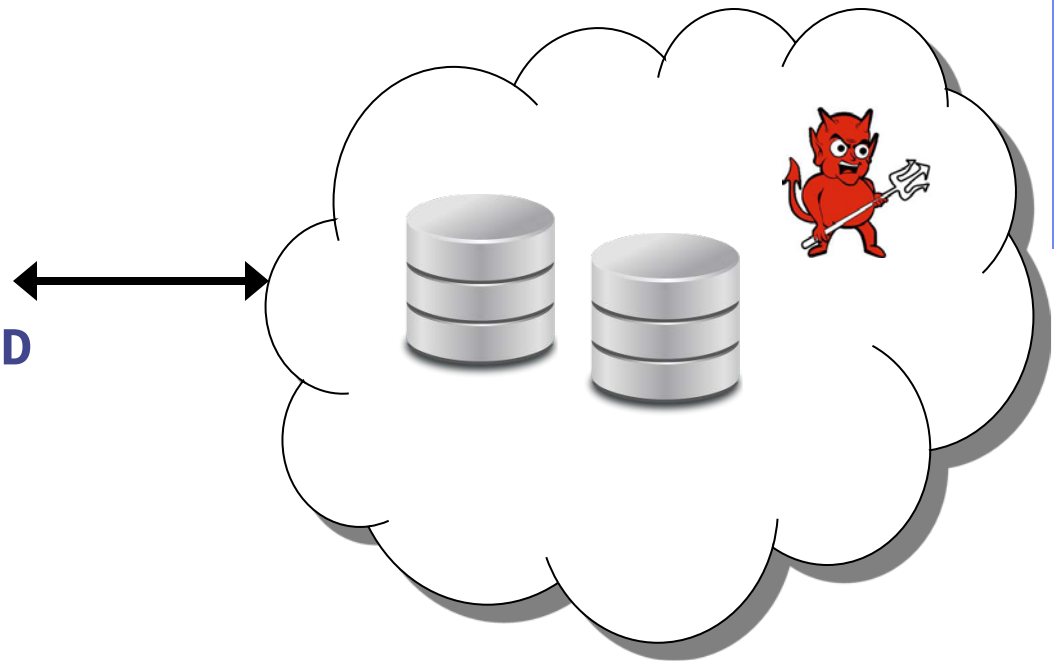
- Dynamic proofs of retrievability [CCS09, CCS13]

What happens **after a verification fails**?

- Provable data recovery [in progress]

Querying a database in the cloud

```
SELECT *
FROM EMP, DEPT
WHERE EMP.eID = DEPT.dID
```

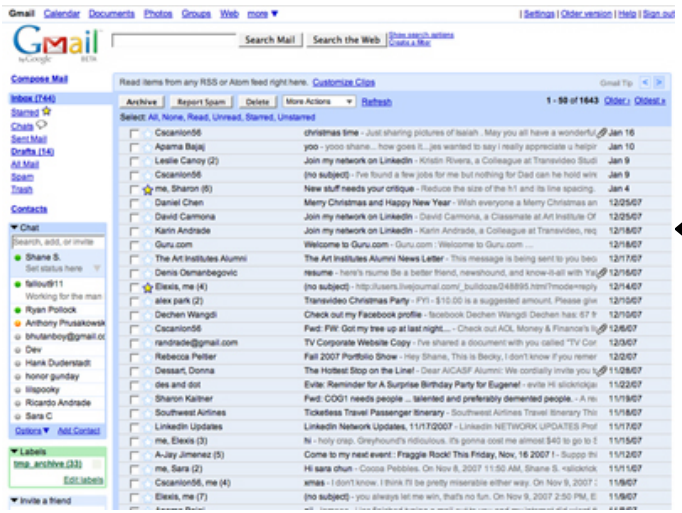


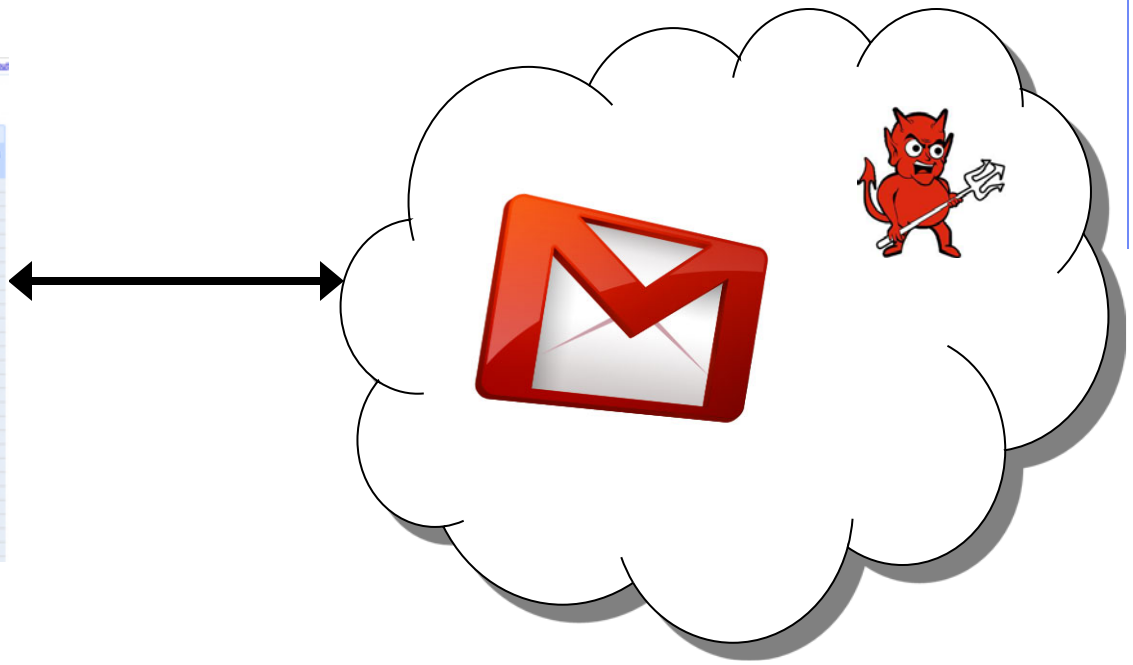
- equi-join queries
- not-in queries

eID	name
1	Alex
22	James
34	George

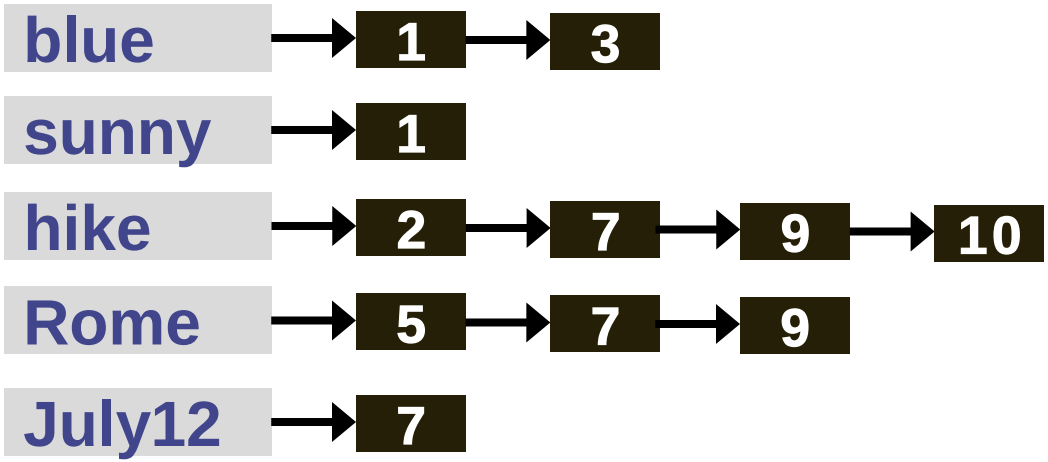
dID	division
1	Sales
21	Advertising
22	IT
76	Sales
88	Sales

Searching my email inbox

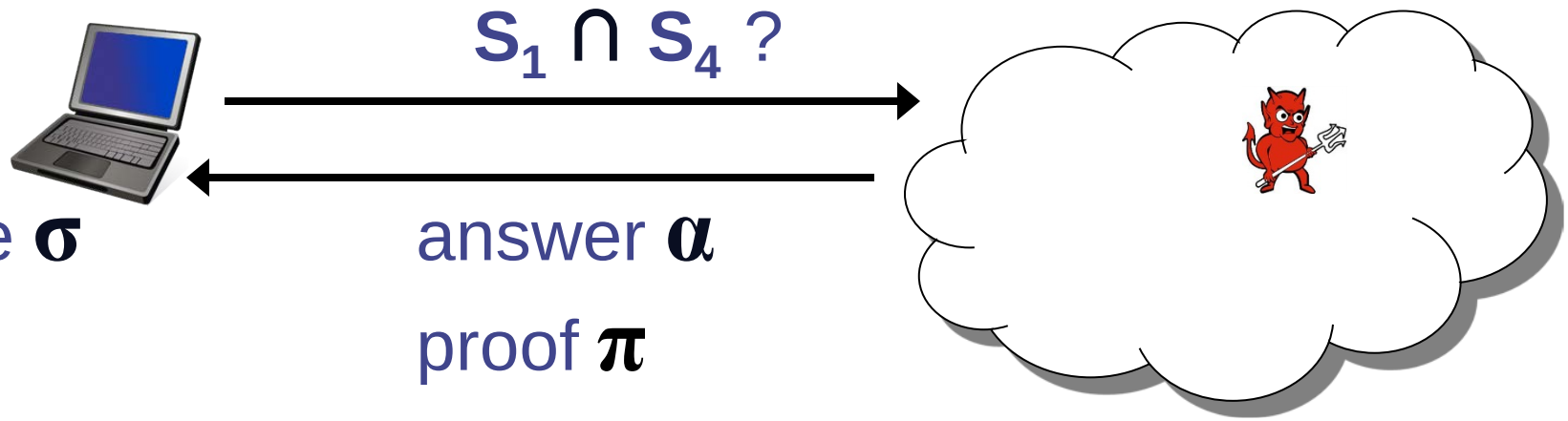


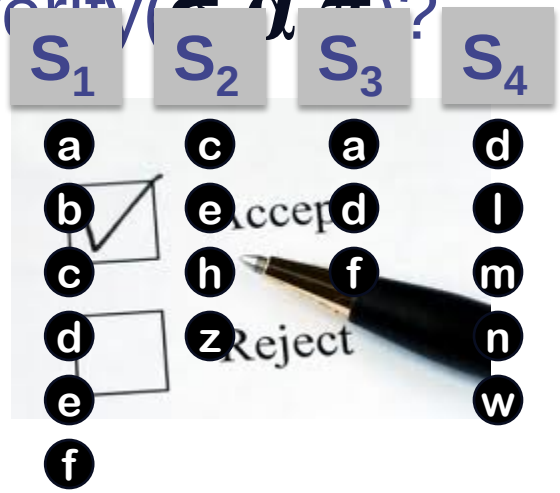


- keyword search
- temporal queries



Verifiable set operations





My contribution

in theory				
	server comp.	proof	client verif.	update
CRYPTO11	$n \log^2 n$	δ	δ	1
generic circuits*	$> N$	1	δ	N/A

n: intersected sets size δ: answer size N: collection size

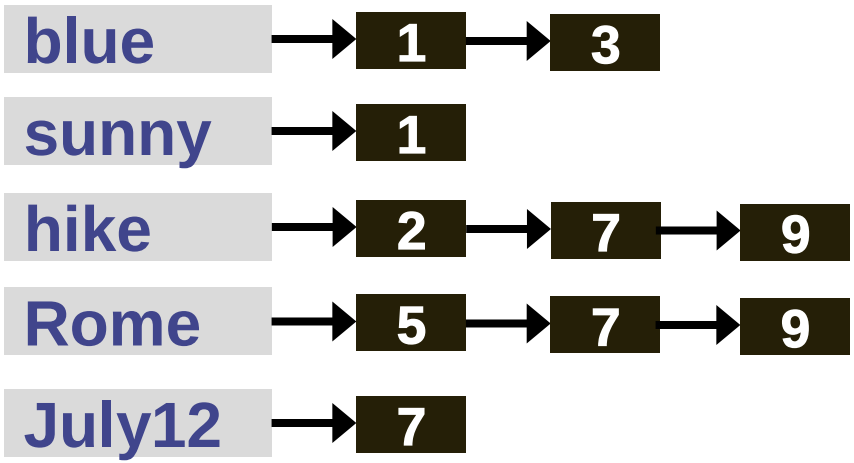
in practice: n = 10,000, δ = 100, $N = n$				
	server comp.	proof	client verif.	update
VLDB12	1.7 s	5 KB	20 ms	0.2 ms
generic circuits**	37 hours	< 40 KB	< 1 ms	N/A

*[Gennaro et al., EUROCRYPT13]

**[Setty et al., EUROSYS13]

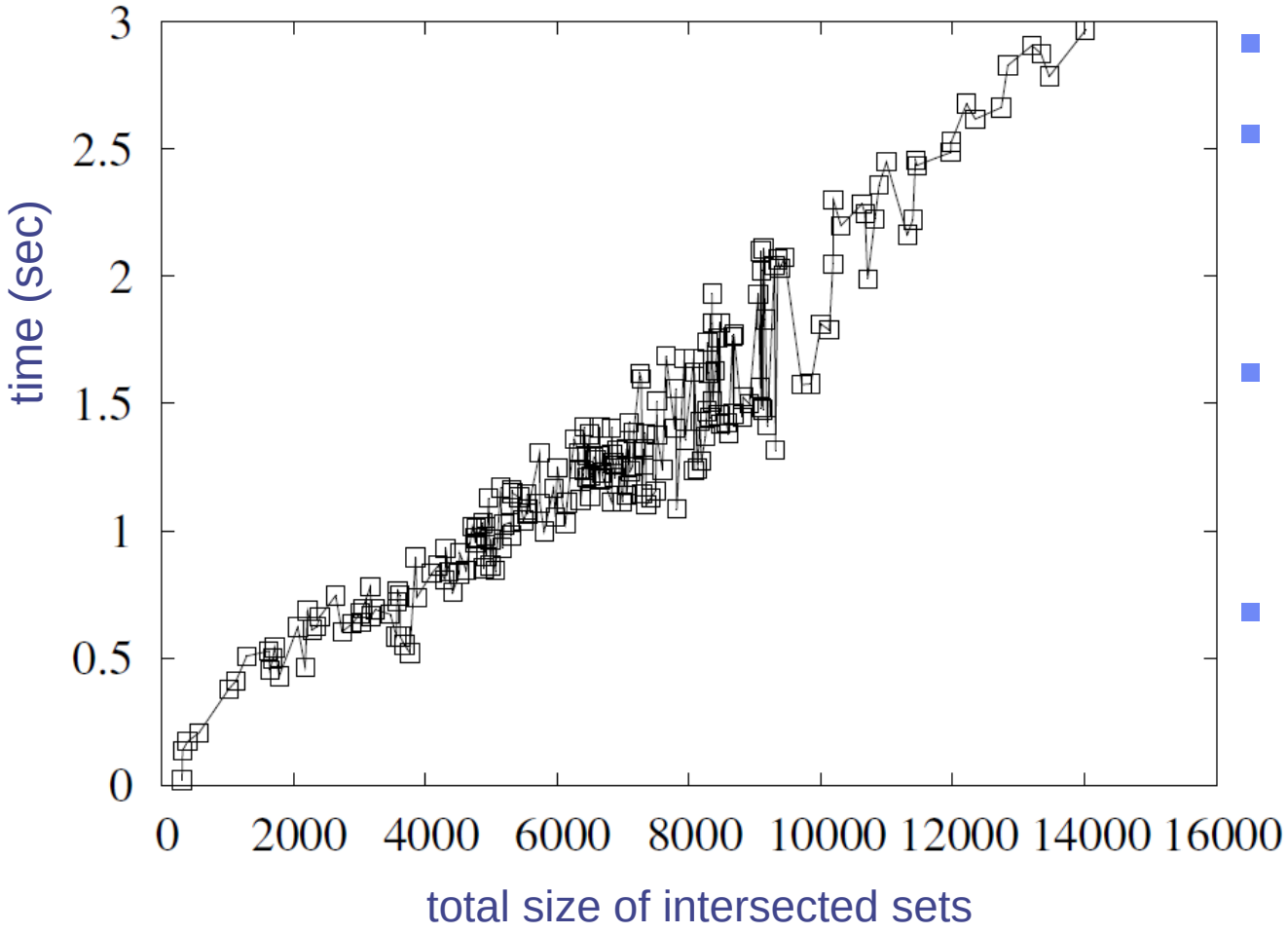
Implementation & experiments

- C++
- NTL [1] and LiDIA [2] libraries for arithmetic ops
- DCLXVI [3] library for group operations
- Synthetic data
- Wall street journal corpus
  - Articles between 1987-1992
  - 173K articles &135K terms



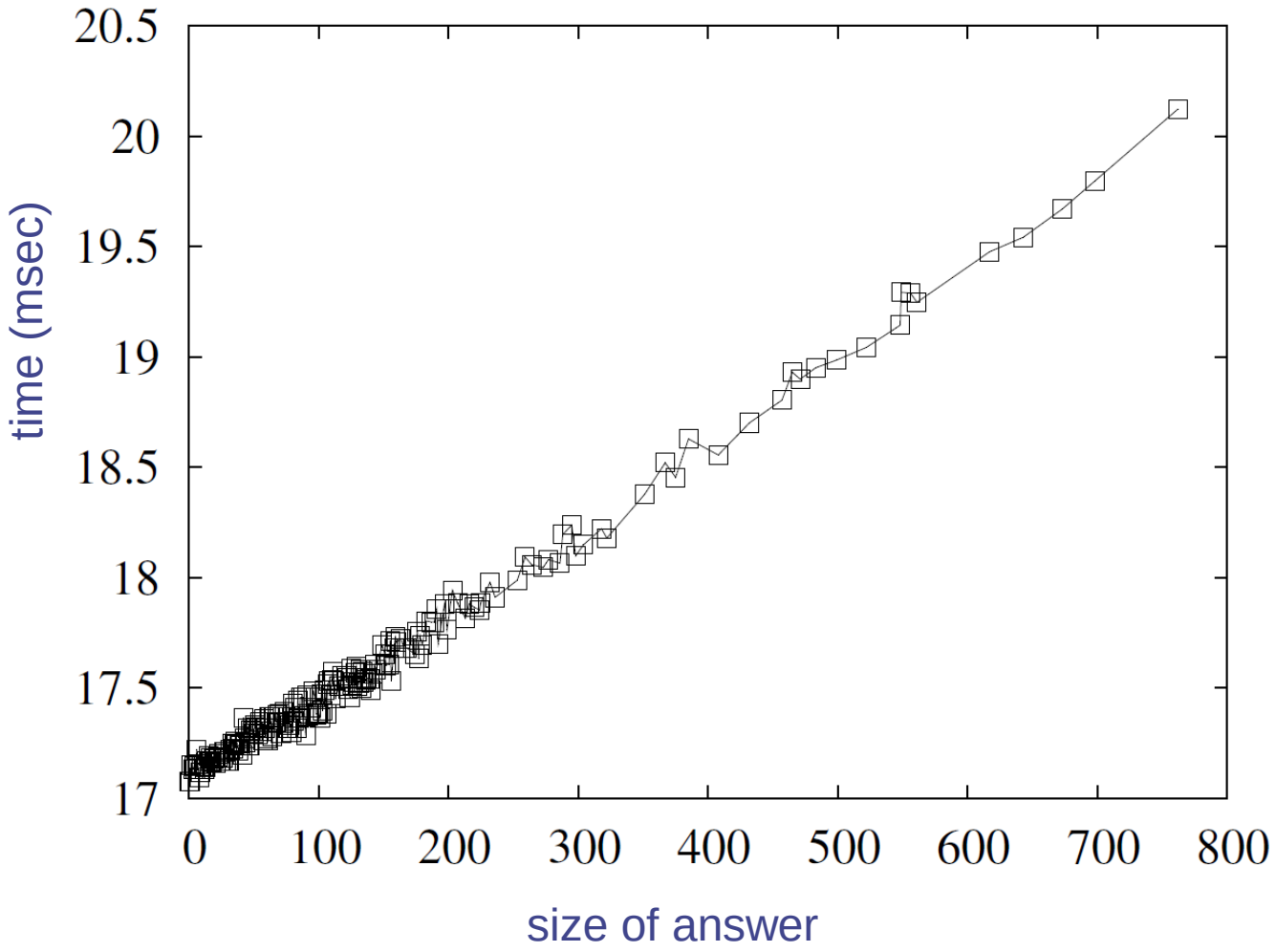
[1] shoup.net/ntl
[2] cdc.informatik.tu-darmstadt.de/TI/LiDIA/
[3] cryptojedi.org/crypto/

Time to compute proof (server)



- Two terms
- Answer mostly between 0-200 articles
- Scales even for popular keywords
- Parallelized when possible

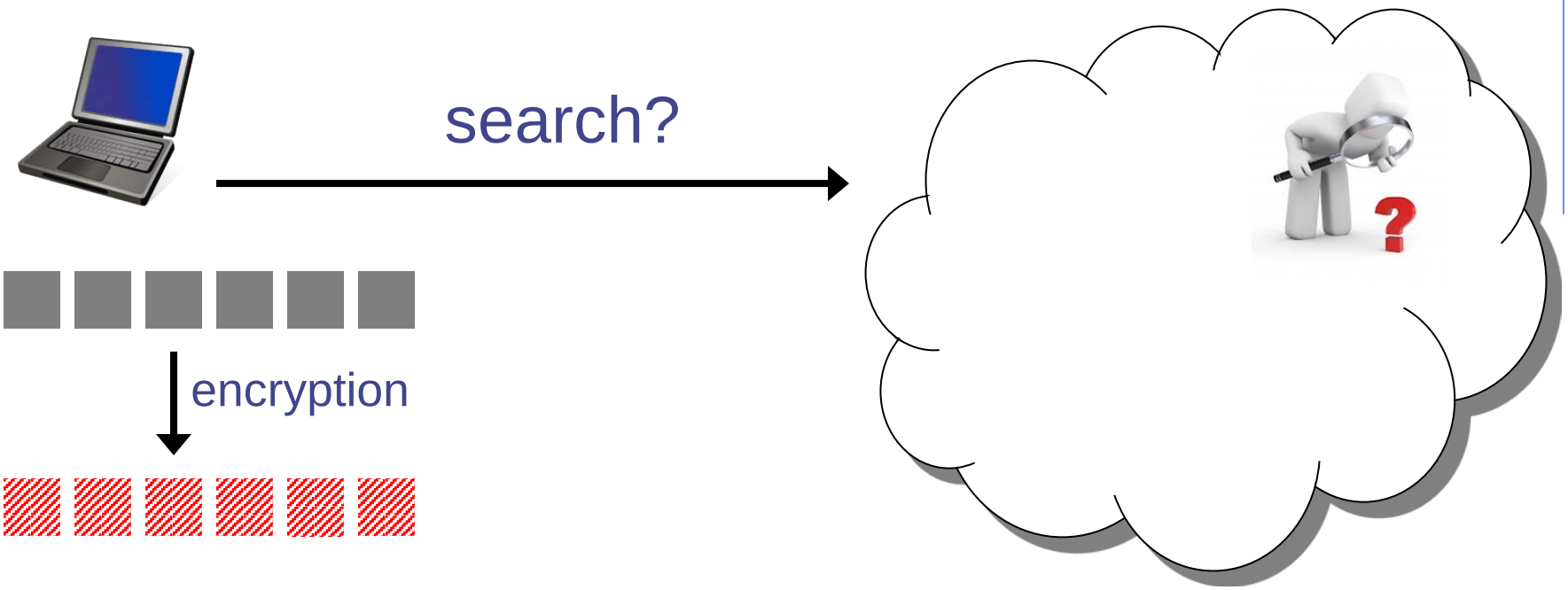
Time to verify proof (client)



- Proportional to the size of the answer
- Order of milliseconds

How about privacy?

Is encryption enough?



- How can you search your encrypted files?
  - Not feasible in principle
  - Encrypt with fully-homomorphic encryption (FHE)?
    - Not very practical

Searchable encryption

search token

- Rich body of work since 2000
  - Scalable encryption
  - Sequential algorithms
- **My work**: First dynamic efficient scheme, [CCS12]
  - Privately indexes keywords, not only files
  - Efficient system implementation
- **My work**: First parallel scheme, [FC13]
  - Uses a tree-based approach

Research agenda

Practical protocols

- Construct customized schemes, e.g., geometric/text processing queries
- Consider highly dynamic/streaming data

Secure cloud computing in the age of big data

- Take advantage of multicore architectures, e.g., build parallel algorithms
- Implement on hardware

Reconsider the model

- Relax assumptions, e.g., more efficiency with trusted hardware
- Consider multi-client settings

Theoretical directions

- Cryptographic verification via certification algorithms
- Explore other crypto primitives (e.g., lattice-based cryptography)

Thank you!

cpap@umd.edu