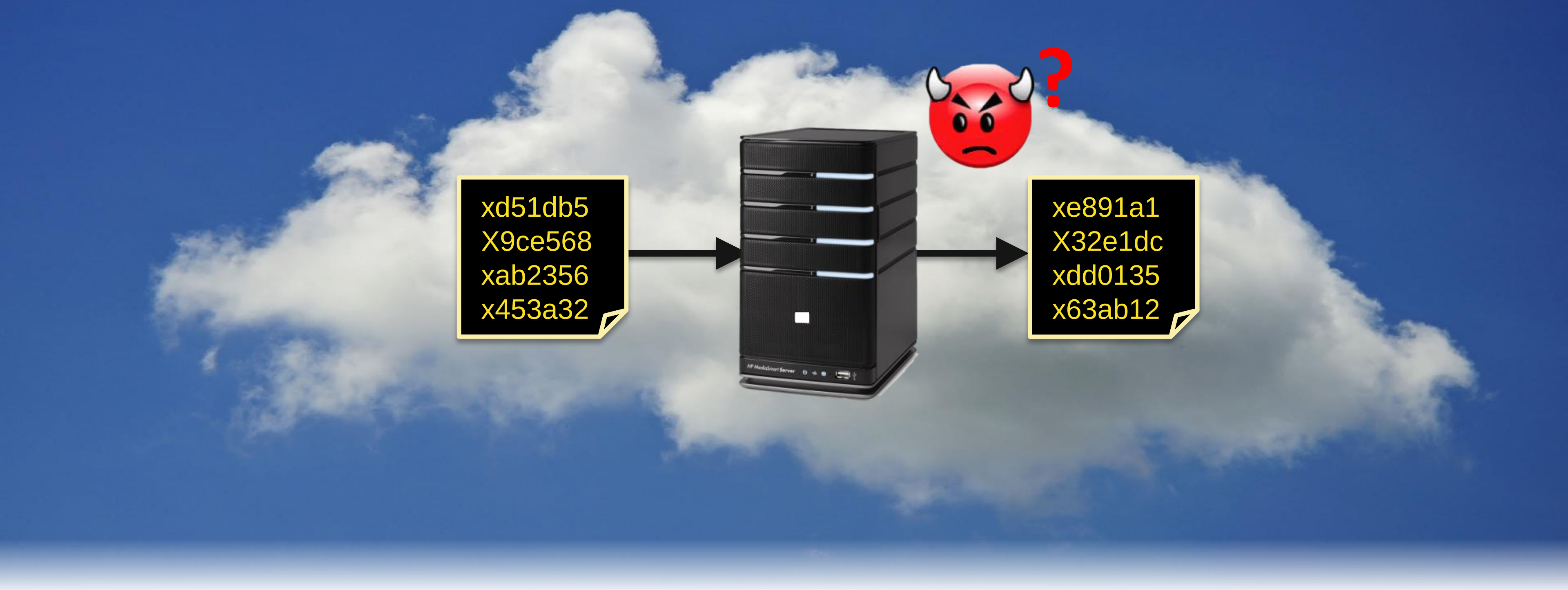


Building systems that compute on encrypted data

Raluca Ada Popa
MIT

Security experts warn of increasing data breaches and privacy risks

Suggested Content

e agents were

LivingSocial Hacked — More Than 50 Million Customer Names, Emails, Birthdates and Encrypted Passwords Accessed (Intern

APRIL 26, 2013 AT 1:15 PM PT

LivingSocial, the daily
owned in part by A

Some Victims of Online Hacking Edge Into the Lig

Compromise of confidential data is prevalent

WordPress firm Automattic suffers root-level hack

dn't Trust Facebook with
mployee's Revelations

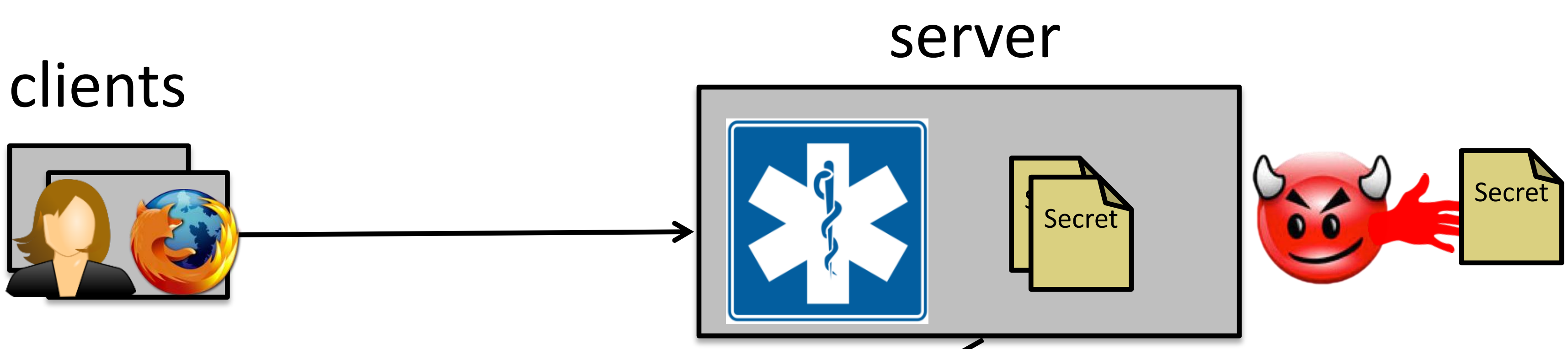
Privacy, security still top cloud concerns

Asia Cloud Forum editors | November 13, 2013

Asia Cloud Forum

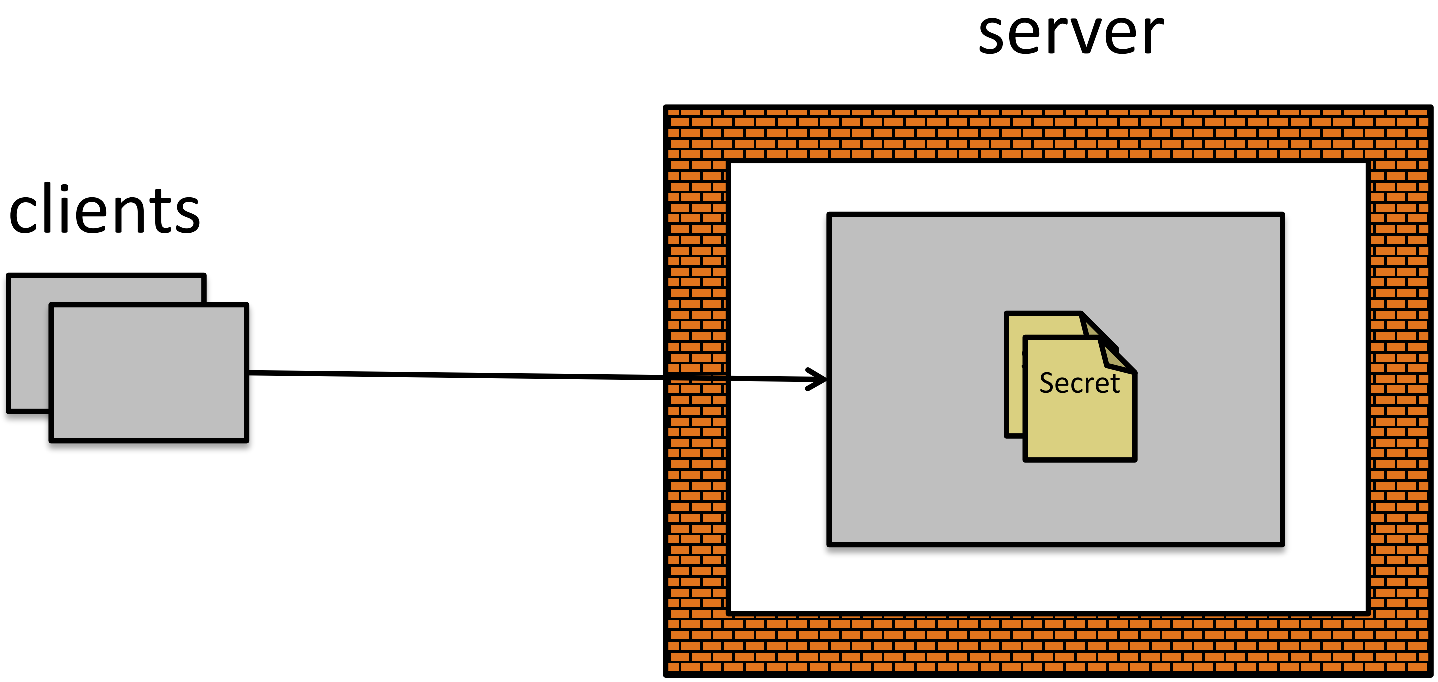
An online survey of Microsoft partners has revealed that traditional concerns about

Problem setup



no computation	computation
storage	databases, web applications, mobile applications, machine learning, etc.
✓ encryption	??

Current systems strategy



Prevent attackers from breaking into servers

Lots of existing work

- Checks at the operating-system level
- Language-based enforcement of a security policy
- Static or dynamic analysis of application code
- Checks at the network level
- Trusted hardware

…

Data still leaks even with these mechanisms

because

attackers eventually break in!

Attacker examples

Attacker:

hackers 

cloud employees 

increasingly many companies store data on external clouds

government 

accessed private data according to 

Reason they succeed:

software is complex

insiders: legitimate server access!

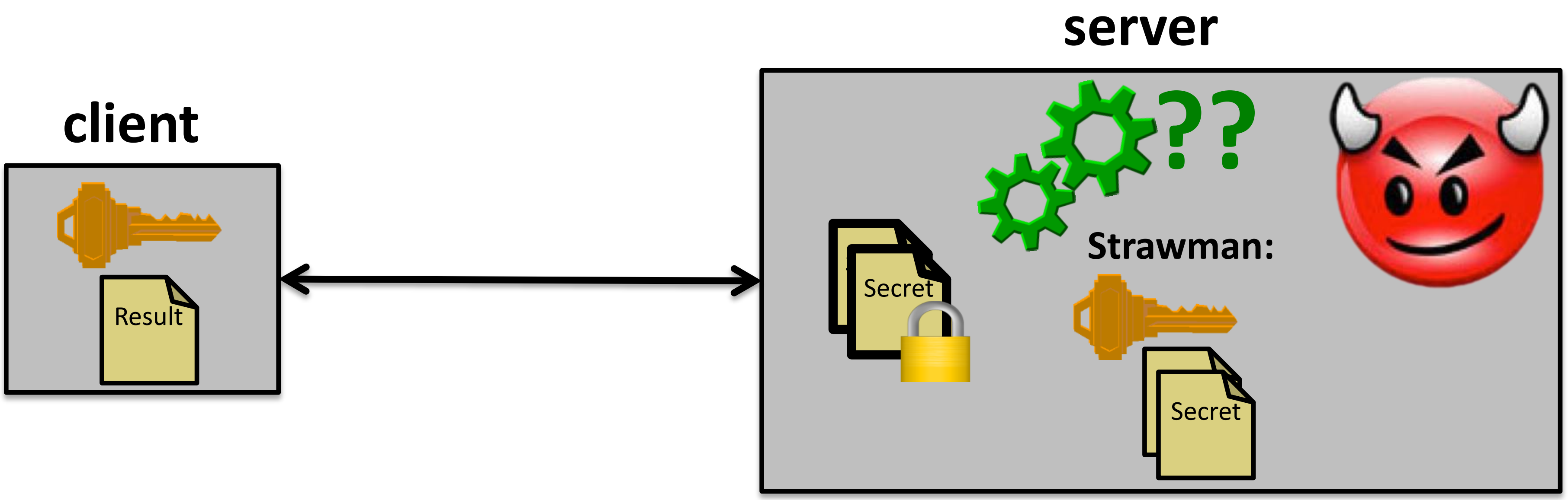
e.g., physical access

My work

Systems that protect confidentiality even against attackers with access to all server data

My approach

Servers store, process, and compute on encrypted data *in a practical way*

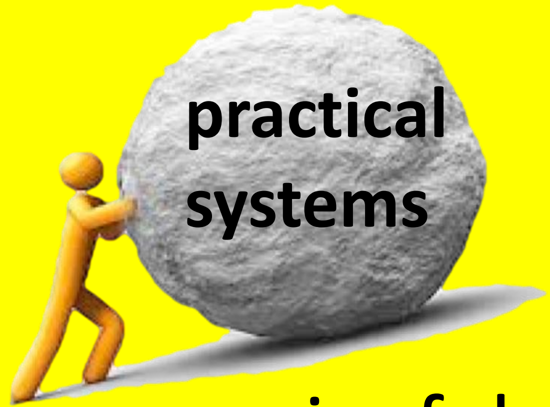


Computing on encrypted data in cryptography
[Rivest-Adleman-Dertouzos'78]

Fully homomorphic encryption (**FHE**) [Gentry'09]

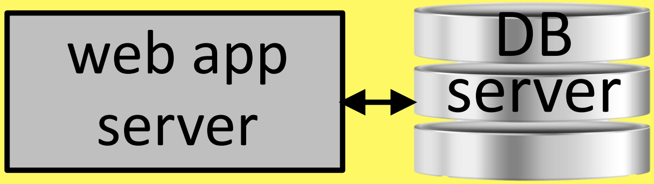
prohibitively slow, e.g., slowdown X 1,000,000,000

My work: **practical systems**



real-world performance + large class of real applications + meaningful security

My contributions

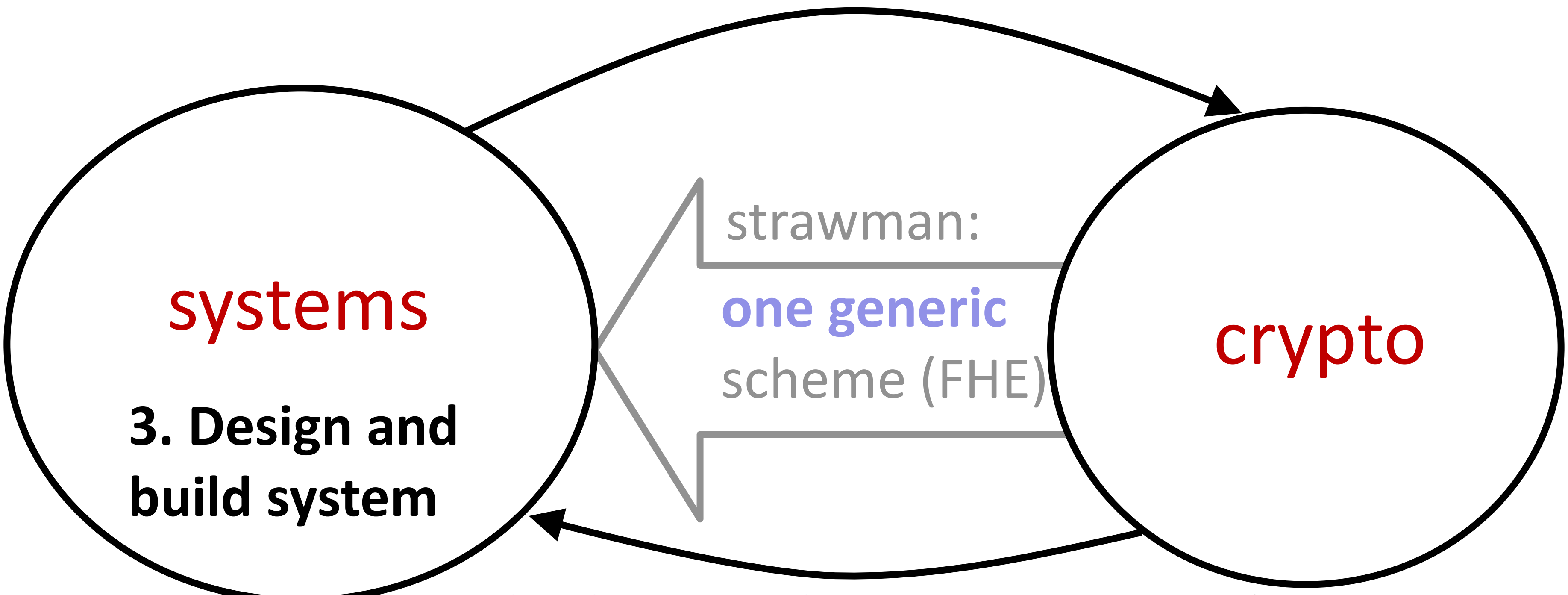
System:		**Server under attack:**
Databases:	**CryptDB** [SOSP'11][CACM'12] mOPE, adjJOIN [Oakland'13]	DB server
Web apps:	**Mylar** [NSDI'14] multi-key search	
Mobile apps: 	**PrivStats** [CCS'11] **VPriv** [Usenix Security'09]	mobile app server

Theory:

In general: **Functional encryption** [STOC'13] [CRYPTO'13]

Combine systems and cryptography

1. identify core operations needed



2. **multiple specialized** encryption schemes

New schemes:

- mOPE, adjJOIN for CryptDB
- multi-key search for Mylar

My contributions

System:		Server under attack:
Databases:	**CryptDB**	DB server
Web apps:	**Mylar**	web app server ↔ DB server
Mobile apps:	**PrivStats** **VPriv**	mobile app server

Theory:

In general: **Functional encryption**

CryptDB

[SOSP'11: **Popa**-Redfield-Zeldovich-Balakrishnan]

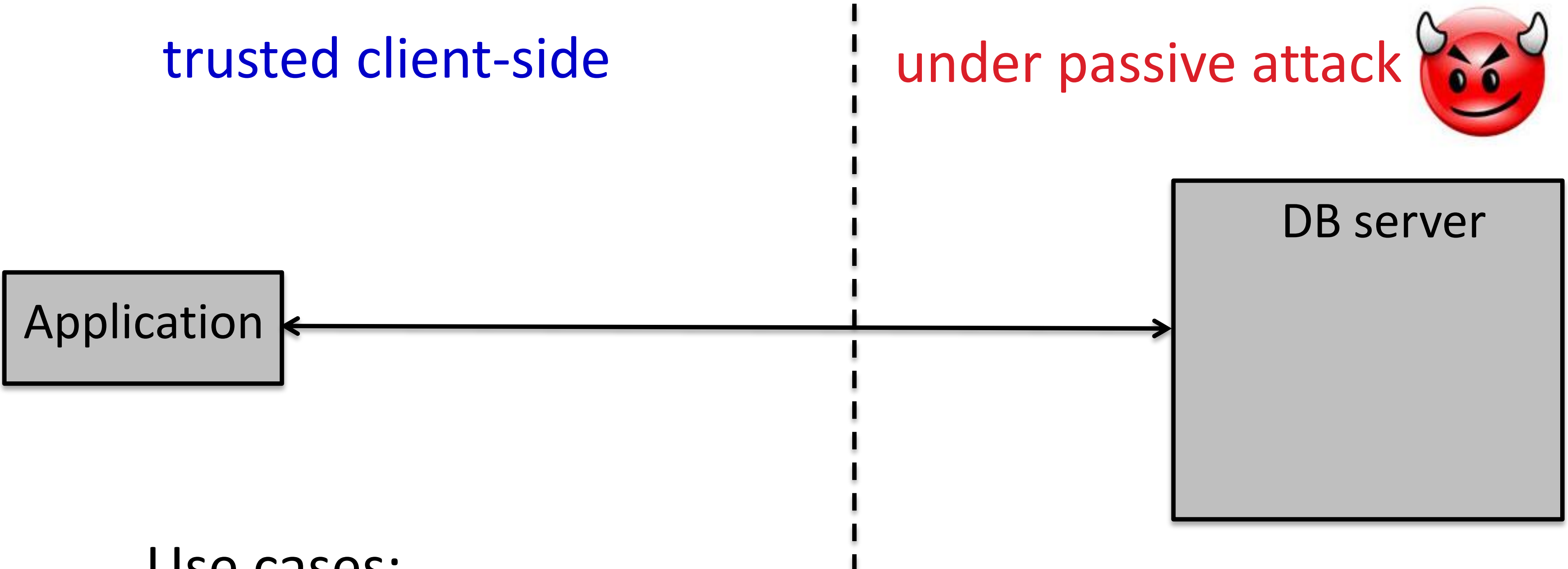
First practical database system (DBMS) to process most SQL queries on encrypted data

Related work

- Systems work: [Hacigumus et al.'02][Damiani et al.'03][Ciriani et al'09]
  - no formal confidentiality guarantees
  - restricted functionality
  - client-side filtering

- Theory work:
  - General computation: FHE [Gentry'09]
    - very strong security: forces slowdown - many queries must always scan and return the whole DB
    - prohibitively slow (10^9x)
  - Specialized schemes [Amanatidis et al.'07][Song et al.'00][Boldyreva et al.'09]

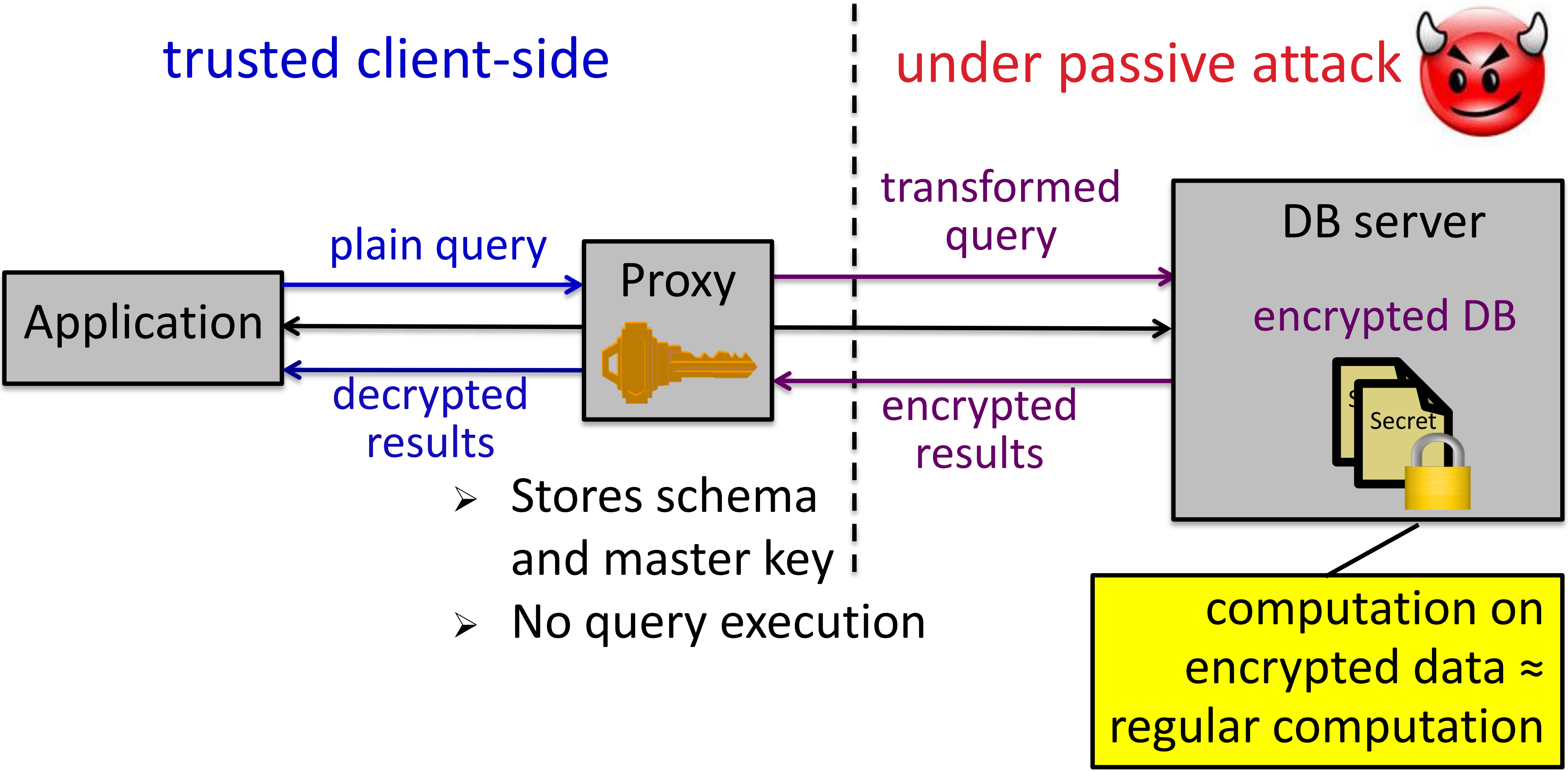
Setup



Use cases:

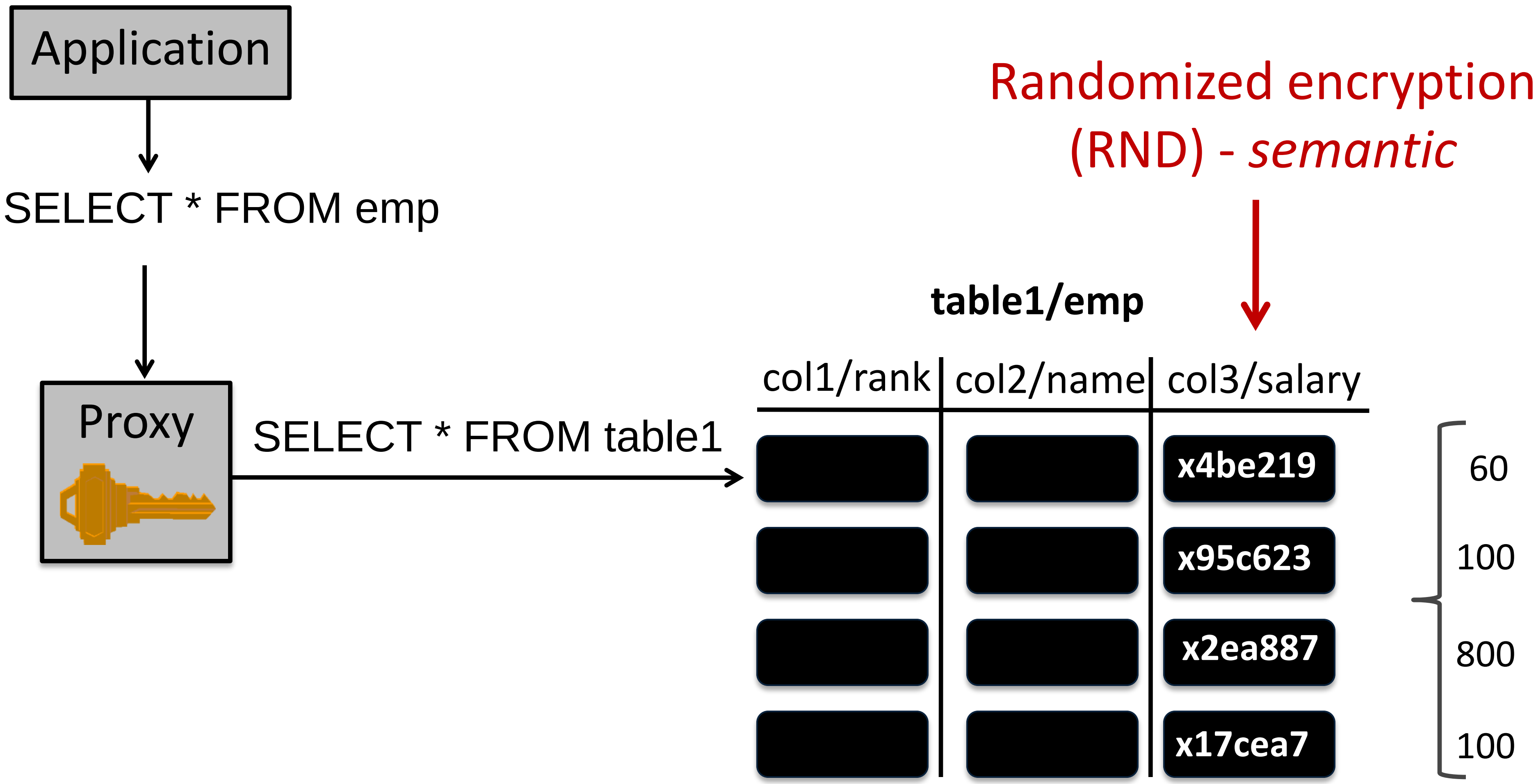
- Outsource DB to the cloud (DBaaS)
  - e.g. Encrypted BigQuery
- Local cluster: hide DB content from sys. admins.

Setup



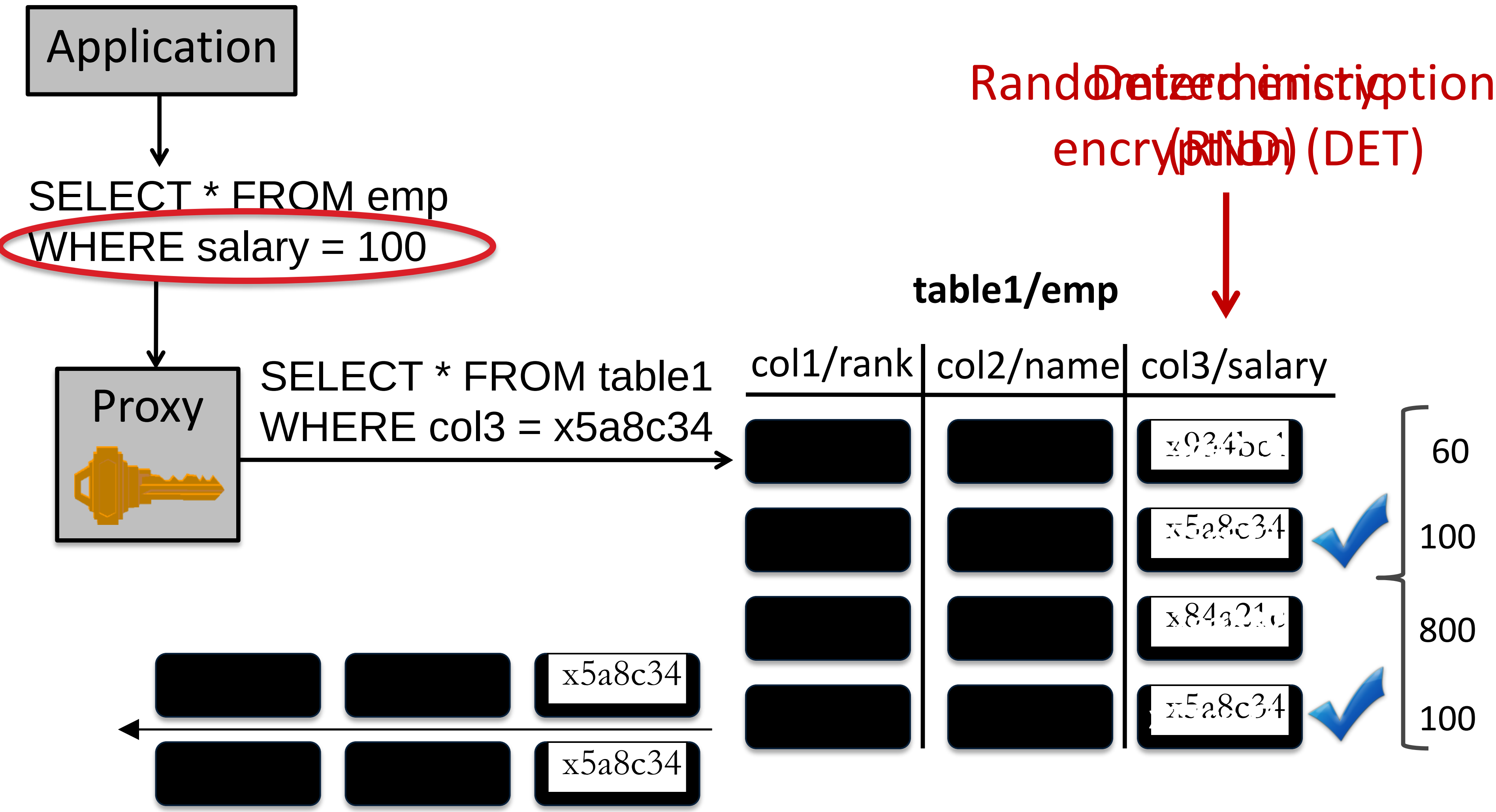
- Stores schema and master key
- No query execution

Example



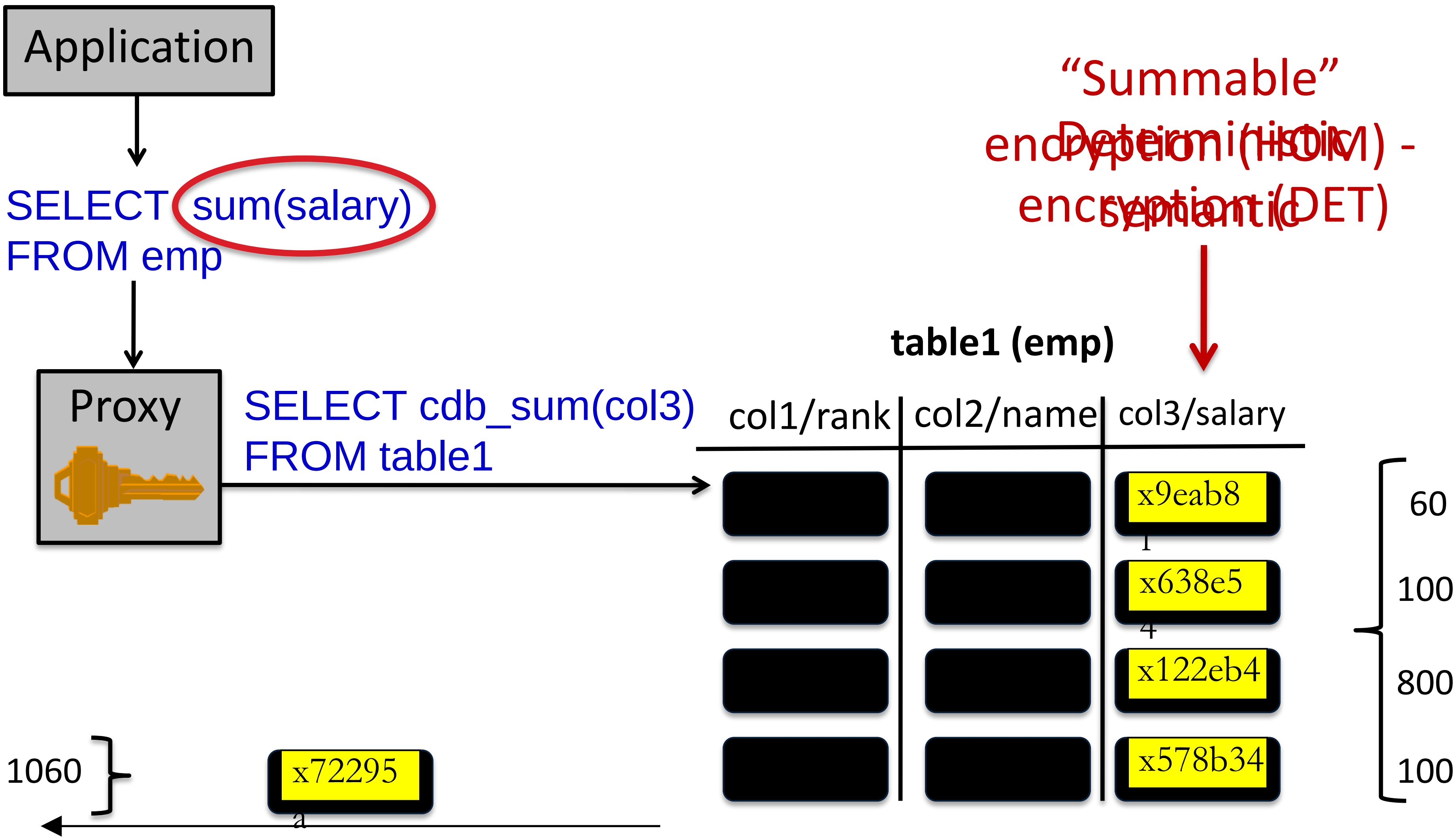
Application
SELECT * FROM emp
Proxy
SELECT * FROM table1
Randomized encryption (RND) - *semantic*
table1/emp
col1/rank
col2/name
col3/salary
x4be219
x95c623
x2ea887
x17cea7
60
100
800
100

Example



Application
SELECT * FROM emp
WHERE salary = 100
Proxy
SELECT * FROM table1
WHERE col3 = x5a8c34
Randomized encryption (RND)
Deterministic (DET)
table1/emp
col1/rank
col2/name
col3/salary
x5a8c34
x84a21c
x5a8c34
60
100
800
100
x5a8c34
x5a8c34

Example



Application
SELECT sum(salary)
FROM emp
Proxy
SELECT cdb_sum(col3)
FROM table1
“Summable”
encryption (HOM) - semantic
Deterministic encryption (DET)
table1 (emp)
col1/rank
col2/name
col3/salary
x9eab8
x638e5
x122eb4
x578b34
60
100
800
100
1060
x72295

Techniques

1. Use SQL-aware set of efficient encryption schemes (meta technique!)

Most SQL can be implemented with a few core operations

2. Adjust encryption of data based on queries

3. Query rewriting algorithm

1. SQL-aware encryption schemes

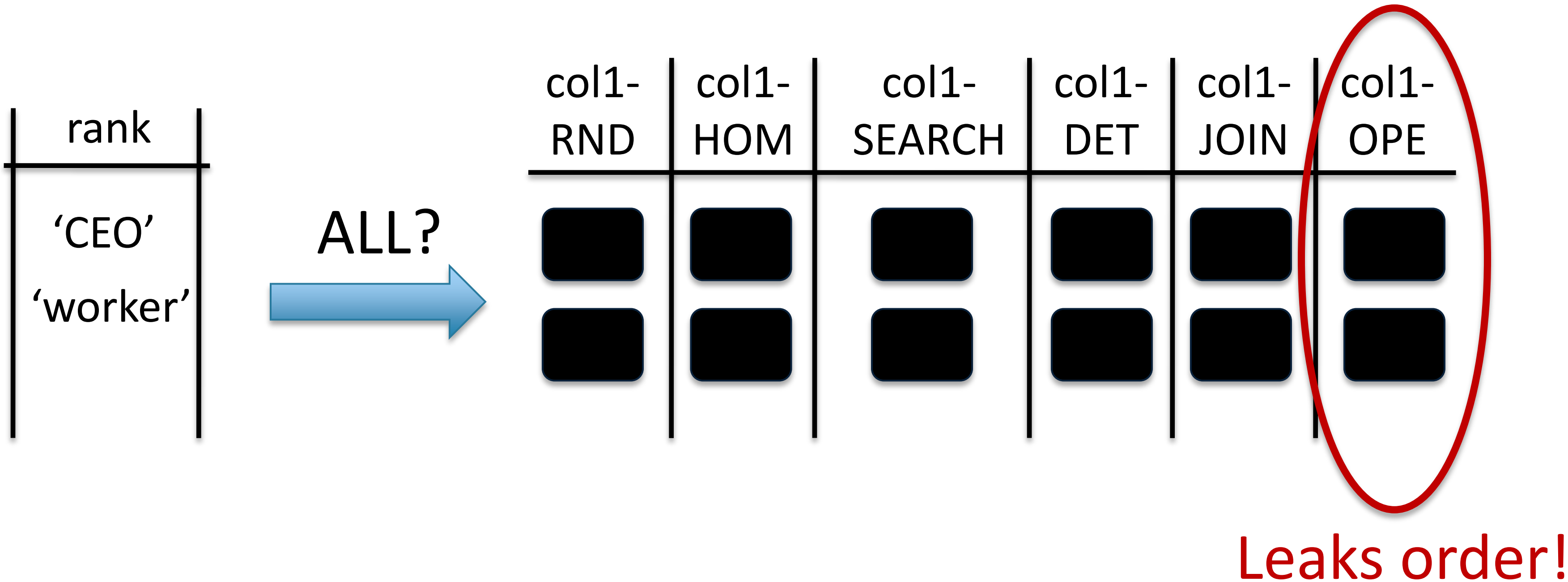
Security	Scheme	Construction	Function	SQL operations:
≈ semantic security	RND	AES in UFE	data moving	e.g., SELECT, UPDATE, DELETE, INSERT, COUNT
≈ semantic security	HOM	Paillier	addition	e.g., SUM, +
≈ semantic security	SEARCH	Song et al.,'00	word search	restricted ILIKE
reveals only repeat pattern	DET	AES in CMC	equality	e.g., =, !=, IN, GROUP BY, DISTINCT
reveals only repeat pattern	JOIN	our new scheme	join	
reveals only order	OPE	our new scheme [Oakland'13]	order	e.g., >, <, ORDER BY, ASC, DESC, MAX, MIN, GREATEST, LEAST

$x < y \Longleftrightarrow Enc(x) < Enc(y)$

How to encrypt each data item?

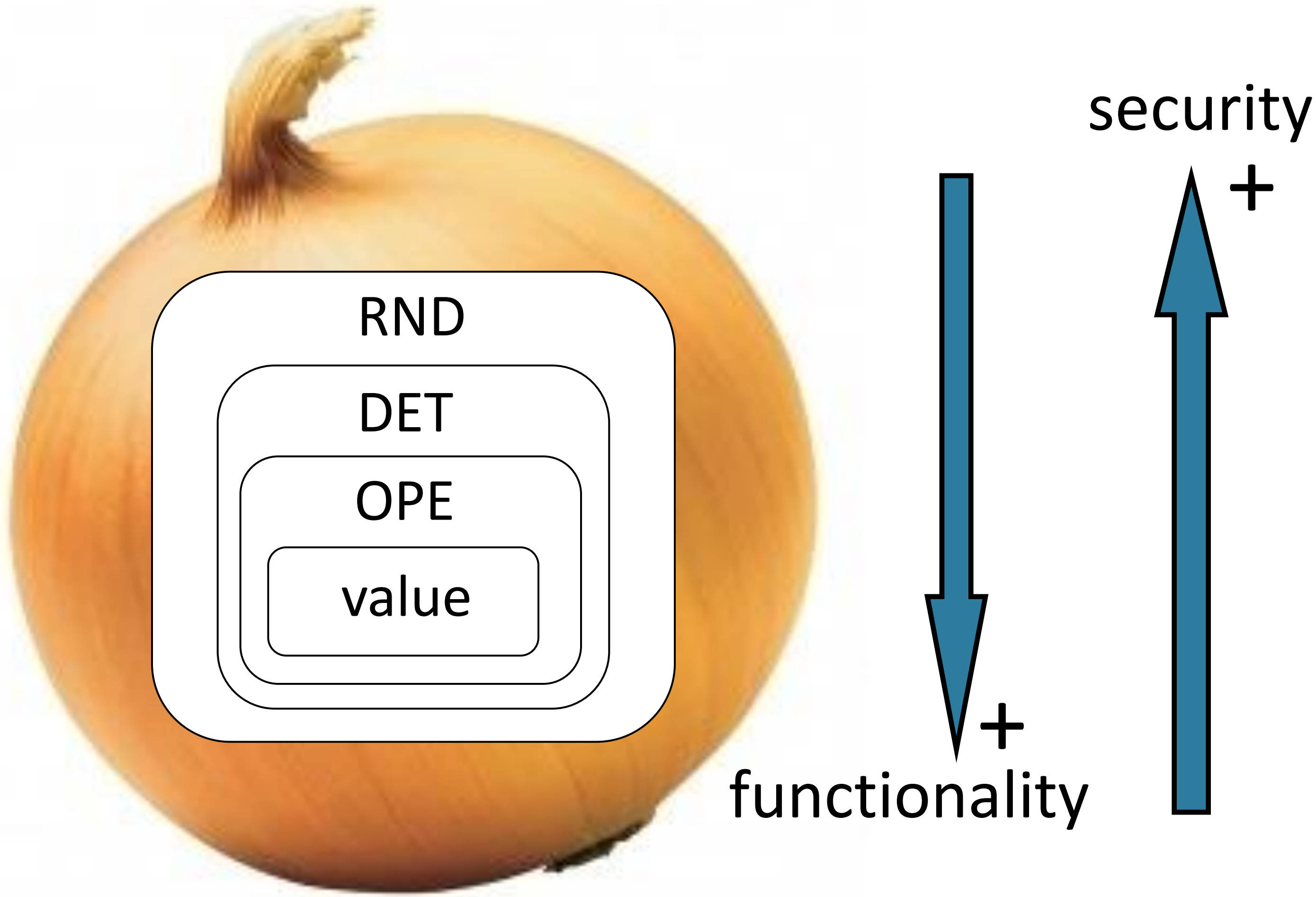
Goals:
1. Support queries
2. Use most secure encryption schemes

Challenge: may not know queries ahead of time



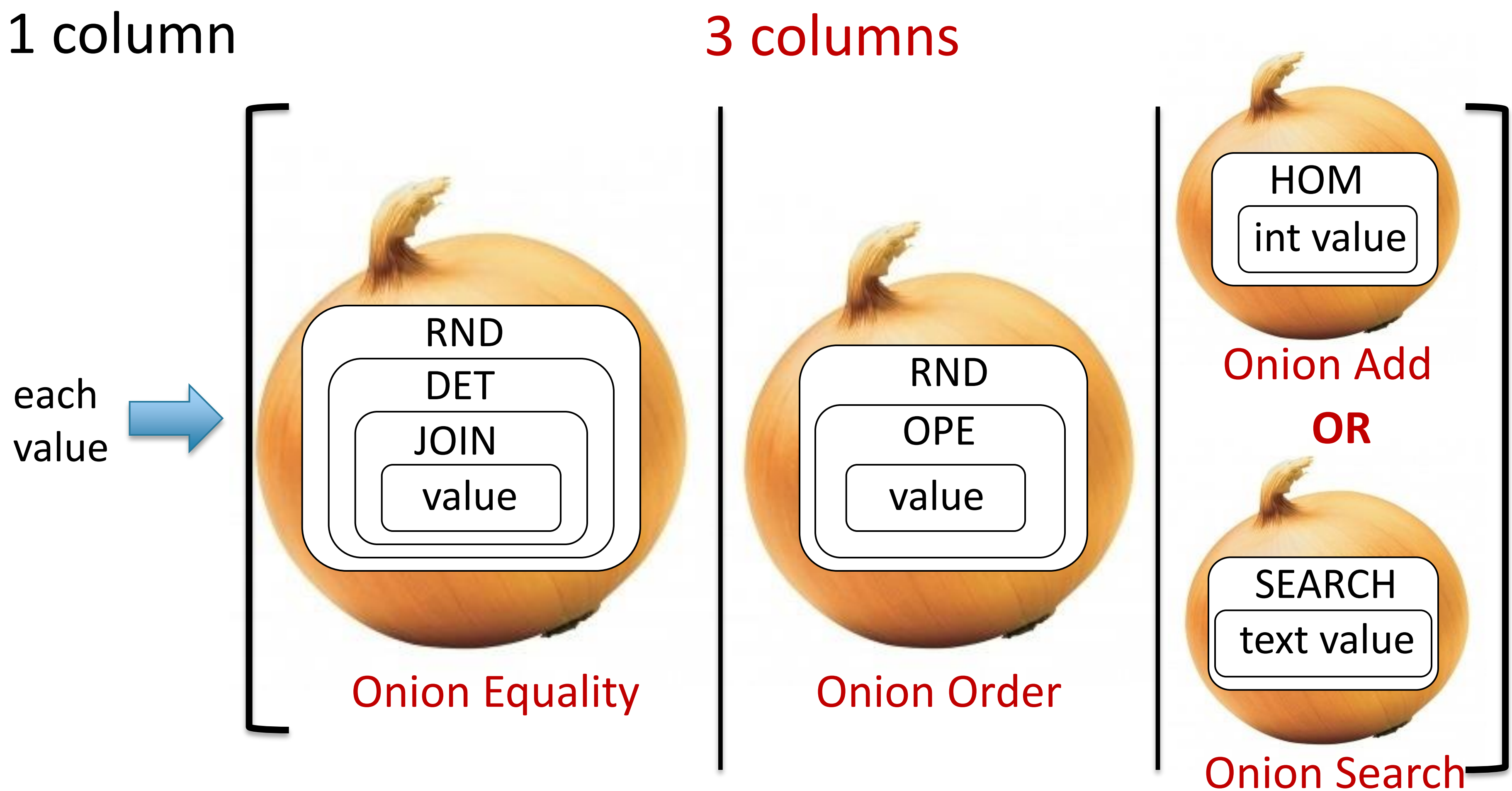
Onion

Onion of encryptions



Adjust encryption: strip off layer of the onion

Onions of encryptions



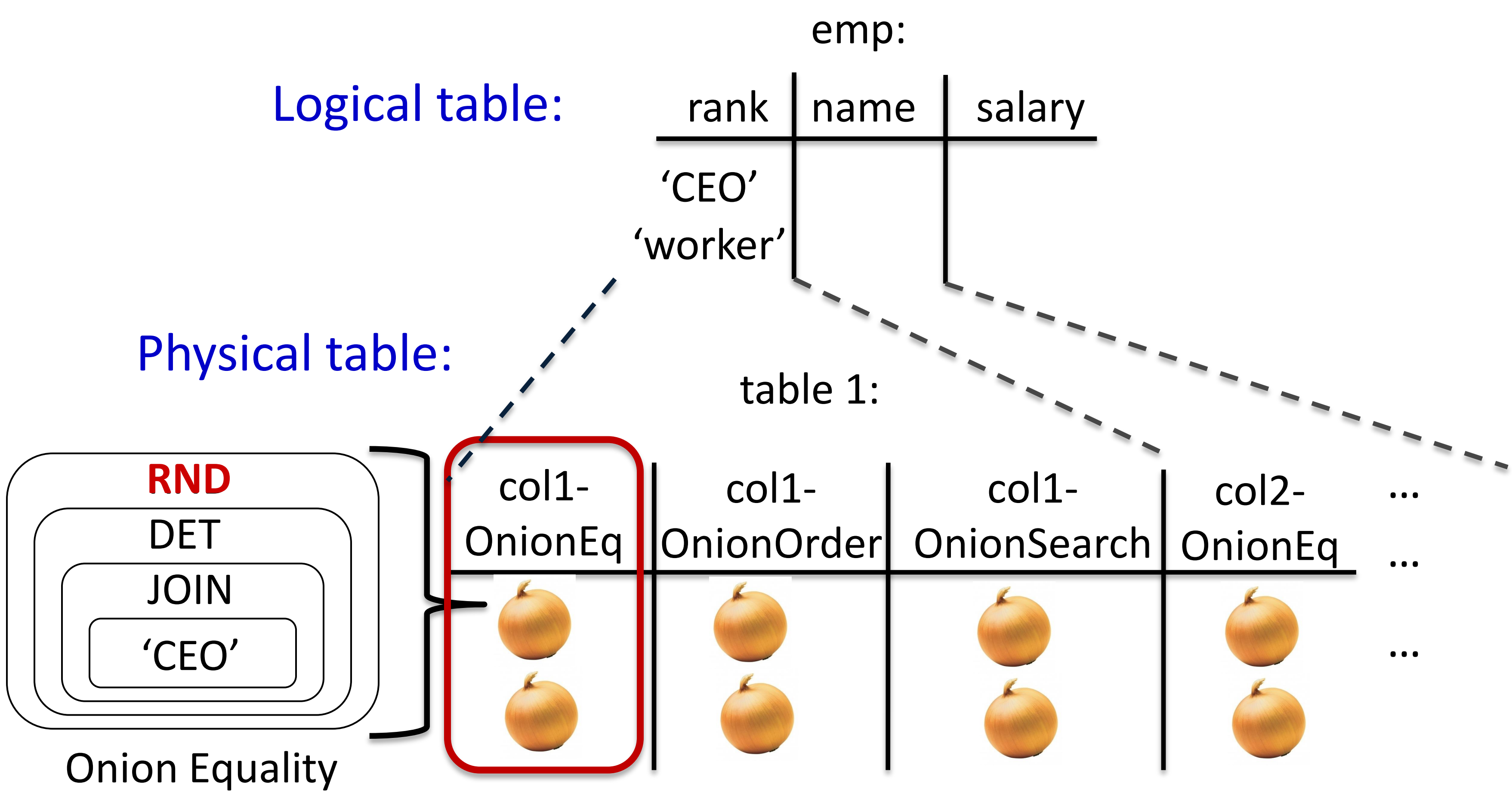
Same key for all items in a column for same onion layer

Onion evolution

- Start out the database with the most secure encryption scheme
- If needed, adjust onion level
  - Proxy gives decryption key to server
  - Proxy remembers onion layer for columns

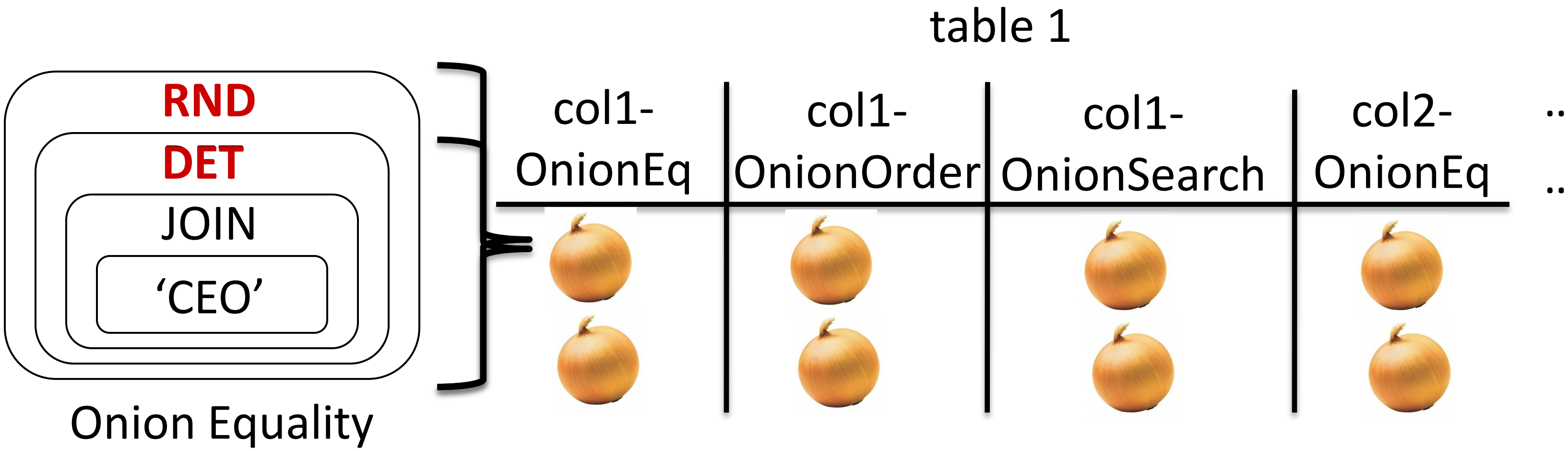
Lowest onion level is never removed

Example



SELECT * FROM emp WHERE rank = 'CEO'

Example (cont'd)



SELECT * FROM emp WHERE rank = 'CEO'

UPDATE table1 SET col1-OnionEq =
Decrypt_RND(key, col1-OnionEq)

SELECT * FROM table1 WHERE col1-OnionEq = xda5c0407

Security threshold

Data owner can specify minimum level of security

CREATE TABLE emp (…, credit_card SENSITIVE integer, …)

RND, HOM, DET for unique fields

≈ semantic security

Security guarantee

Columns annotated as sensitive have semantic security (or similar).

Encryption schemes exposed for each column are the most secure enabling queries.

equality → repeats

sum → semantic

no filter → semantic

common in practice

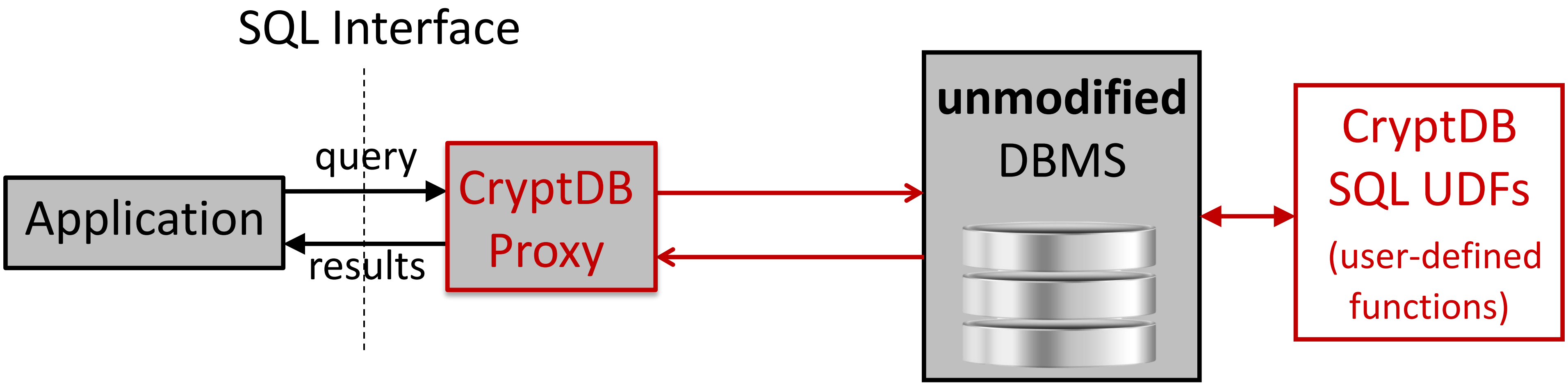
Limitations & Workarounds

Queries not supported:

- More complex operators, e.g., trigonometry
- Certain combinations of encryption schemes:
  - e.g., $\underbrace{\boxed{\text{salary + raise}}}_{\text{HOM}}$ > 100K

→ use query splitting, query rewriting

Implementation



No change to the DBMS!

Largely no change to apps!

Evaluation

1. Does it support real queries/applications?
2. What is the resulting confidentiality level?
3. What is the performance overhead?

Real queries/applications

Application	Encrypted columns	# cols with queries not supported
phpBB	23	0
HotCRP	22	0
grad-apply	103	0
TPC-C	92	0
sql.mit.edu	128,840	1,094

apps with sensitive columns

tens of thousands of apps

SELECT 1/log(series_no+1.2) …

… WHERE sin(latitude + PI()) …

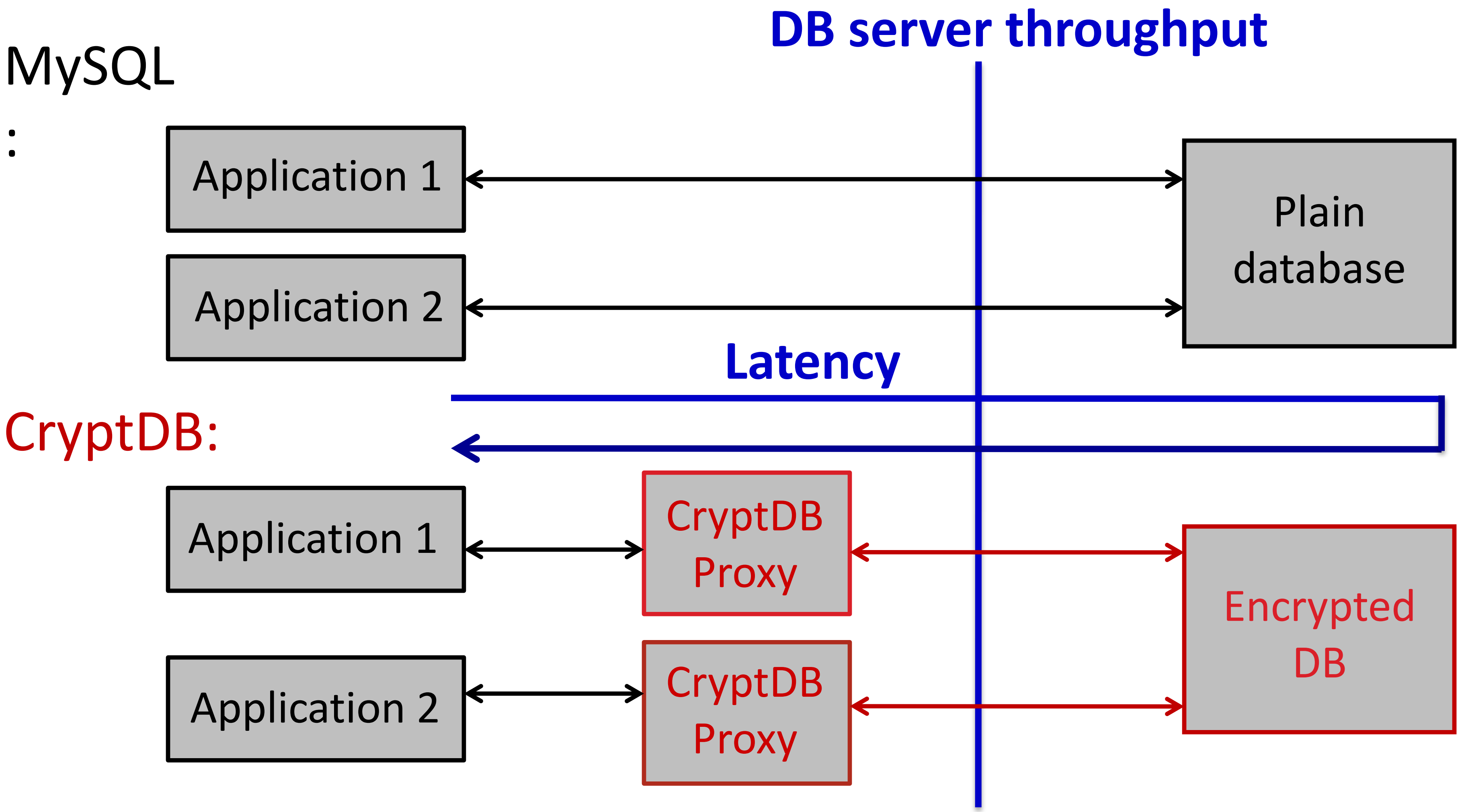
Confidentiality level

		Final onion state		
Application	Encrypted columns	Min level: ≈semantic	Min level: DET/JOIN	Min level: OPE
phpBB	23	21	1	1
HotCRP	22	18	1	2
grad-apply	103	95	6	2
TPC-C	92	65	19	8
sql.mit.edu	128,840	80,053	34,212	13,131

Most columns at semantic

Most columns at OPE were less sensitive

Performance

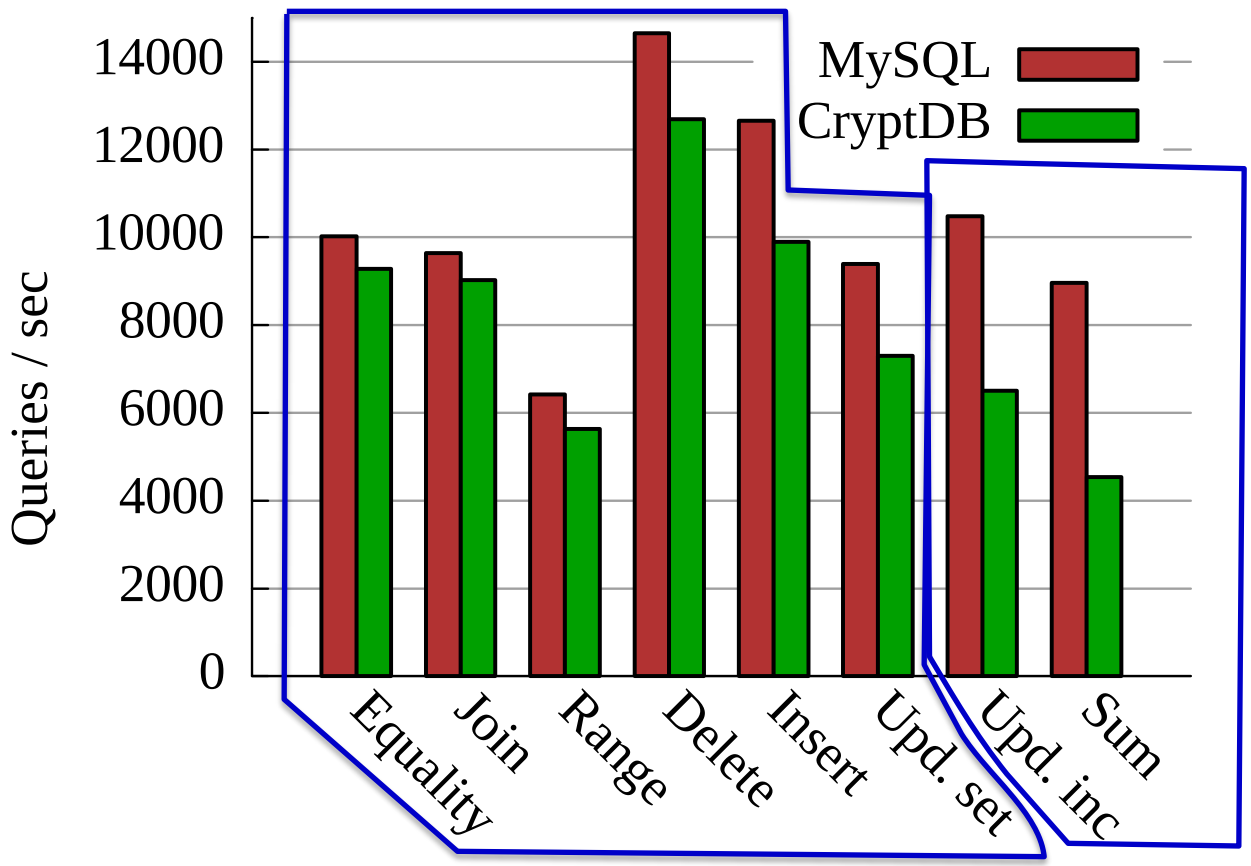


Hardware: 2.4 GHz Intel Xeon E5620 – 8 cores, 12 GB RAM

TPC-C performance

Latency (per query): 0.10ms MySQL vs. 0.72ms CryptDB

Throughput loss over MySQL: 26%



Homomorphic addition

No cryptography at the DB server in the steady state!

Adoption

http://css.csail.mit.edu/cryptdb/

Google Encrypted BigQuery [http://code.google.com/p/encrypted-bigquery-client/]

Úlfar Erlingsson, head of security research, Google

"CryptDB was really eye-opening in establishing the practicality of providing a SQL-like query interface to an encrypted database"

"CryptDB was [..] directly influential on the design and implementation of Encrypted BigQuery."

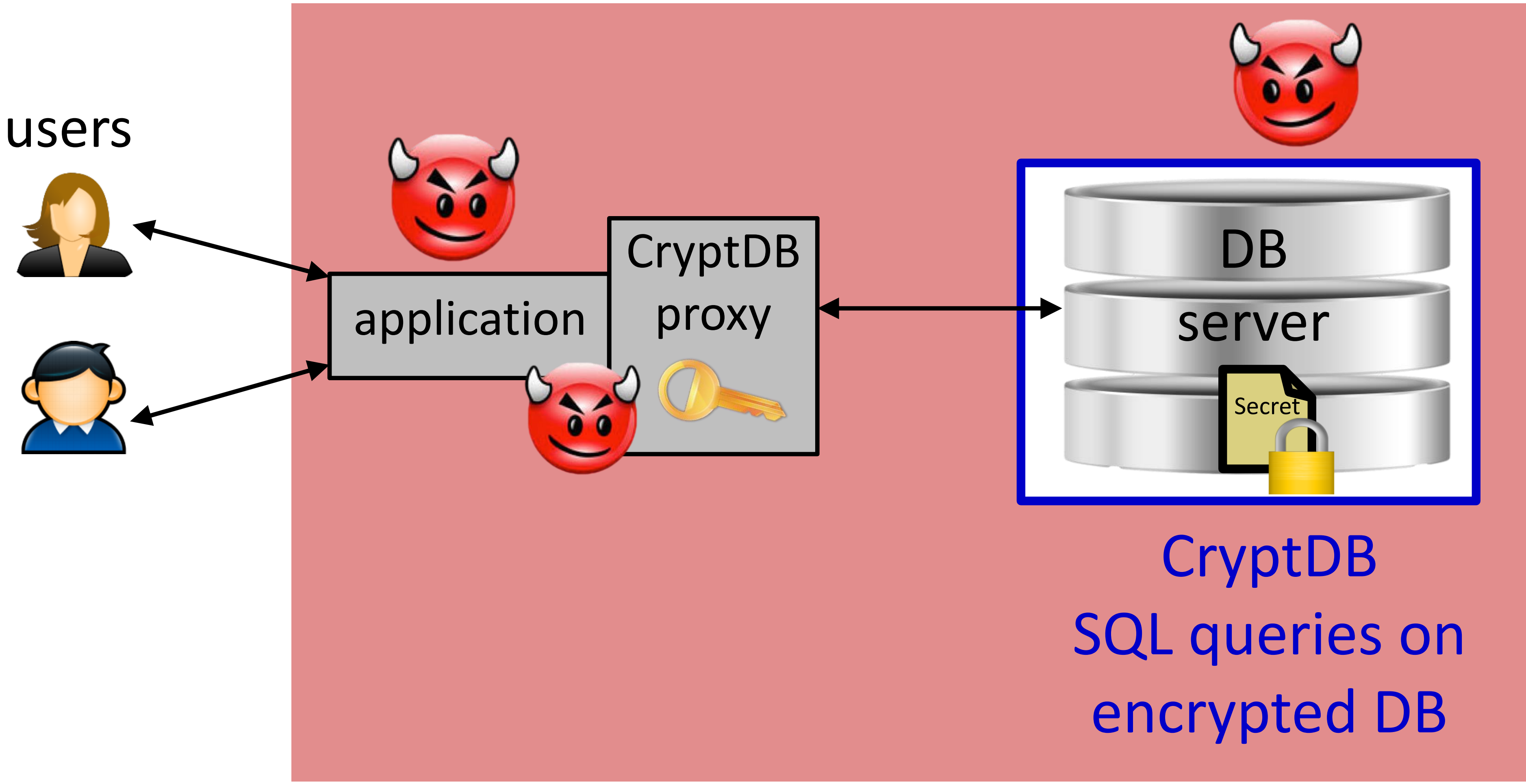
SAP: SEEED implemented on top of the SAP HANA DBMS

MIT Lincoln Laboratory: Encrypted version of the D4M Accumulo NoSQL engine

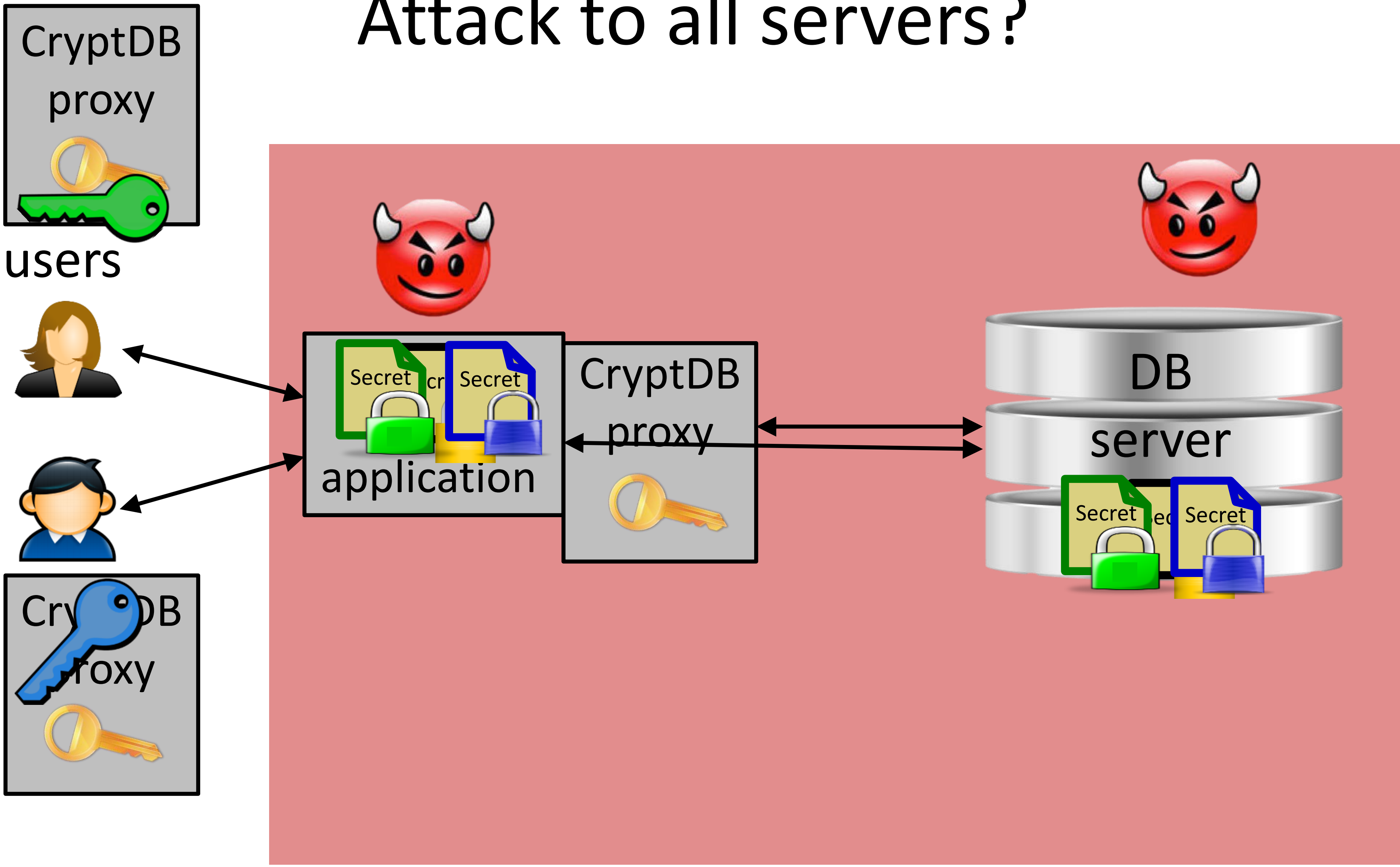
sql.mit.edu: Users opted-in to run Wordpress over our CryptDB source code

Demo

Attack to all servers?

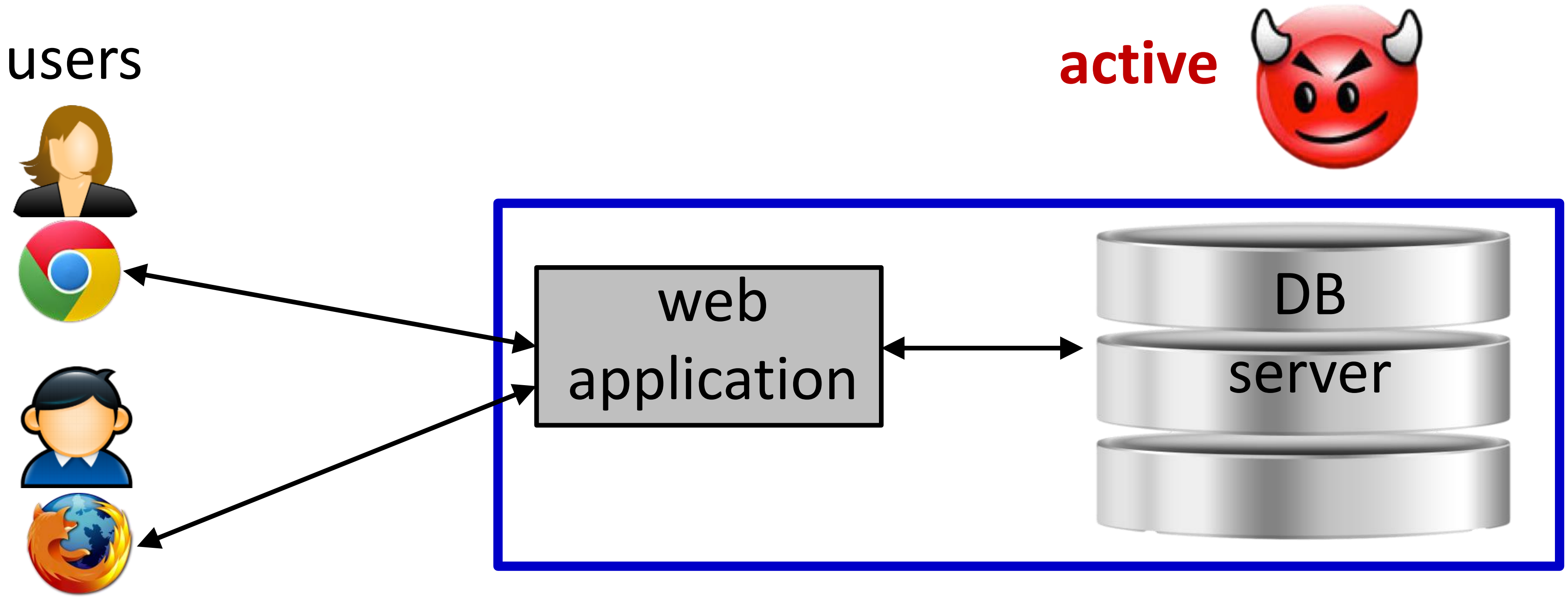


Attack to all servers?



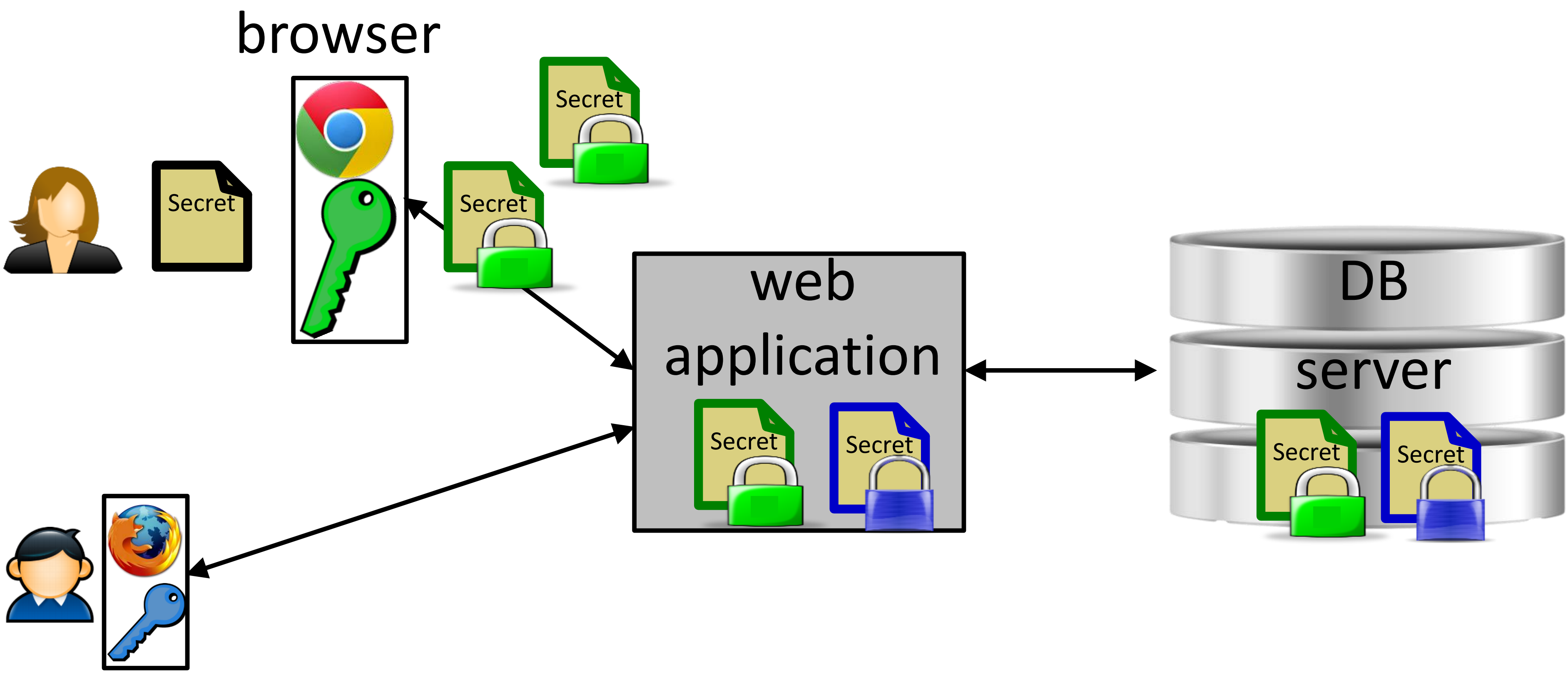
[NSDI'14: **Popa**-Stark-Valdez-Helfer-Zeldovich-Kaashoek-Balakrishnan]

Mylar



- Framework for building web applications
- Protects confidentiality against attacks to all servers

Overview



Plaintext data exists only in browsers

Computation in web applications

1. Mylar is a client-side application framework

2. Non client-side computation: meta technique!

- data sharing
- search

Challenges

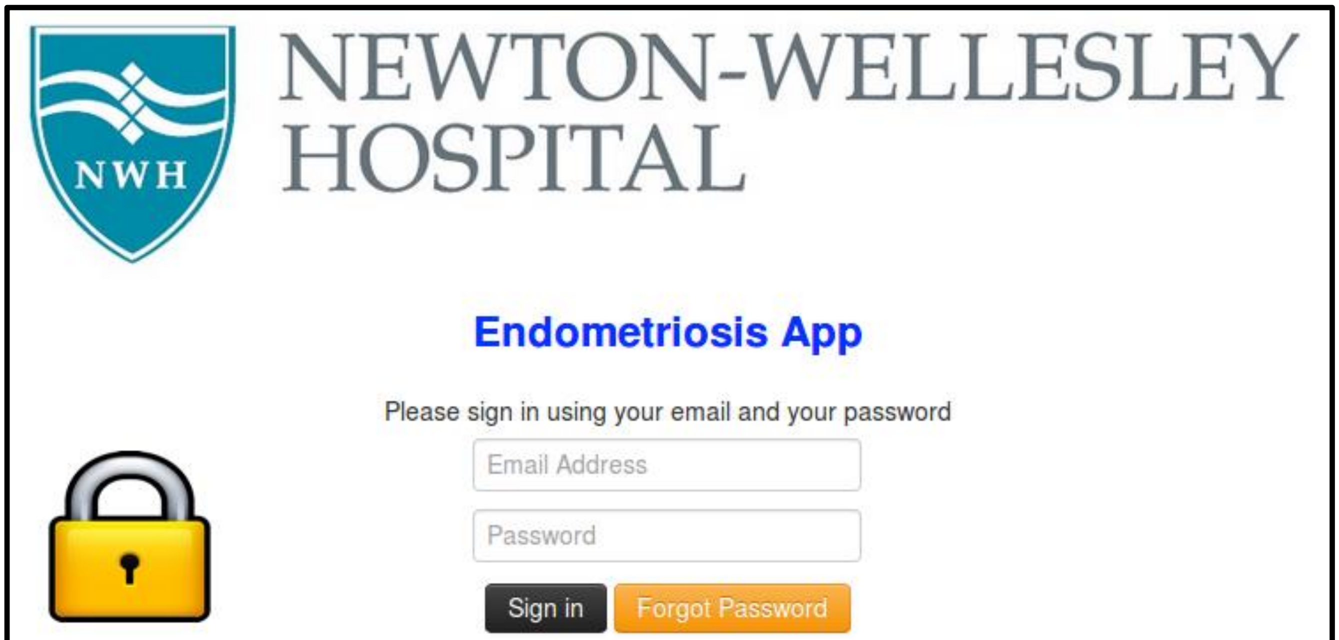
- Active attacker
  - key certification
- Multiple keys
  - multi-key search

Applications

http://css.csail.mit.edu/mylar/

chat medical class website forum calendar photo sharing

Few developer annotations to secure an application, modest overhead



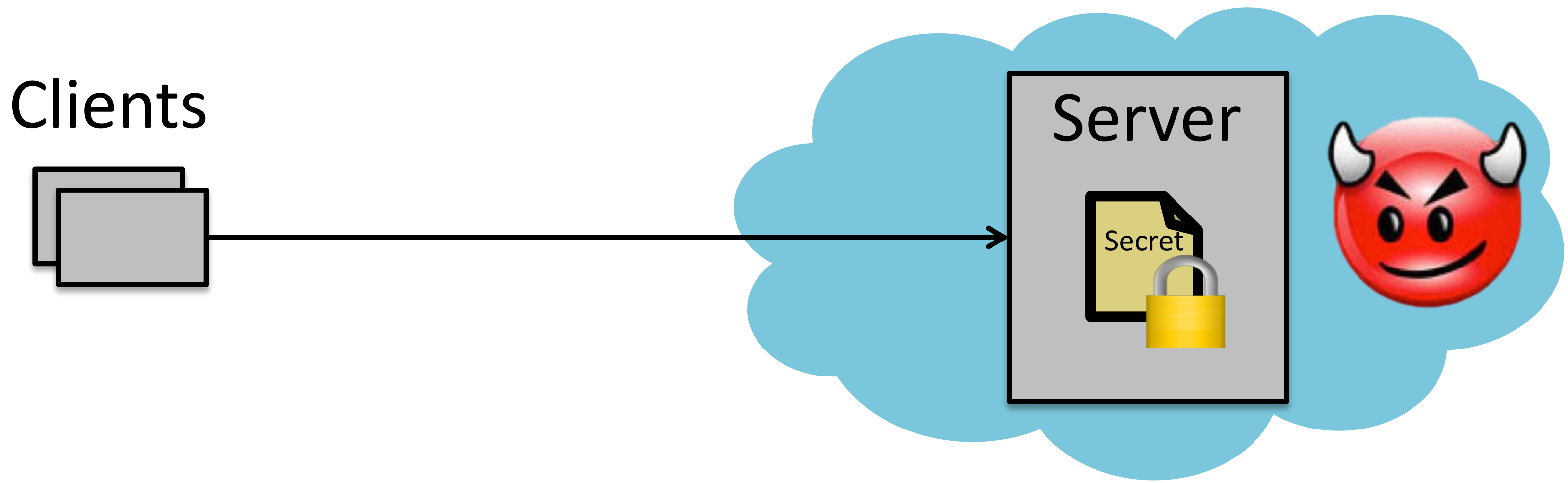
My contributions

System:		Server under attack:
Databases:	**CryptDB** [SOSP'11][CACM'12] mOPE, adjJOIN [Oakland'13]	DB server
Web apps:	**Mylar** [NSDI'14] multi-key search	web app server ↔ DB server
Mobile apps: 	**PrivStats** [CCS'11] **VPriv** [Usenix Security'09]	mobile app server

Theory: **Functional encryption** [STOC'13] [CRYPTO'13]

- Proof of concept for general functions
- Solved old open problem: **reusable garbled circuits**

System design principles



Assume all server data will leak!

Store, process, and compute on encrypted data.

Technique for practicality:

1. identify core operations
2. use an efficient encryption scheme for each

Future work

Other systems computing on encrypted data:

Genomics analytics and machine learning

Future work

Other systems computing on encrypted data:

Genomics analytics and machine learning

Big data & compression

big data

compressed big data

encrypted big data

encrypted

compressed big data

How to compute on it??

Future work

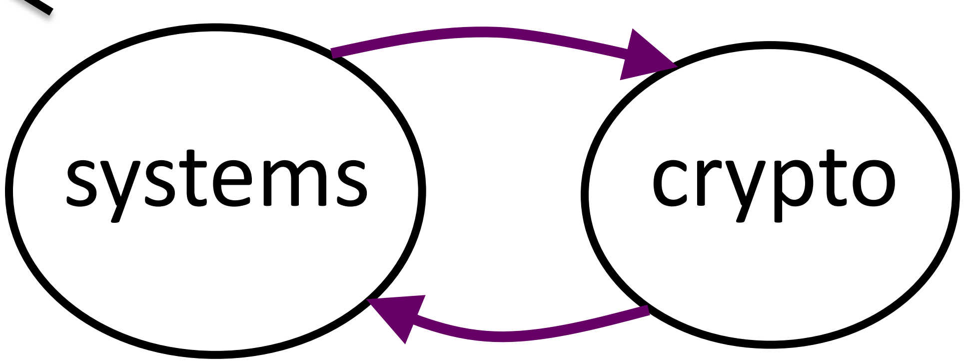
Other systems computing on encrypted data:

- Genomics analytics and machine learning
- Big data & compression

Security beyond confidentiality:

- Correctness of computation

Client-side security



Collaborators

CryptDB: Catherine Redfield, Nickolai Zeldovich, Hari Balakrishan, Aaron Burrows

Mylar: Steven Valdez, Jonas Helfer, Nickolai Zeldovich,
Frans M. Kaashoek, Hari Balakrishnan

PrivStats, VPriv: Andrew Blumberg, Hari Balakrishnan, Frank H. Li

Functional encryption: Shafi Goldwasser, Yael Kalai, Vinod Vaikuntanathan, Nickolai Zeldovich

and others for other projects.

Future work

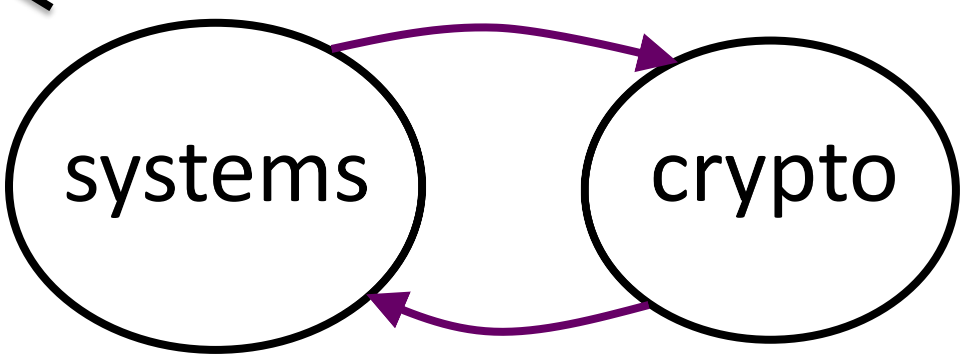
Other systems computing on encrypted data:

- Genomics analytics and machine learning
- Big data & compression

Security beyond confidentiality:

- Correctness of computation

Client-side security



THANK YOU!