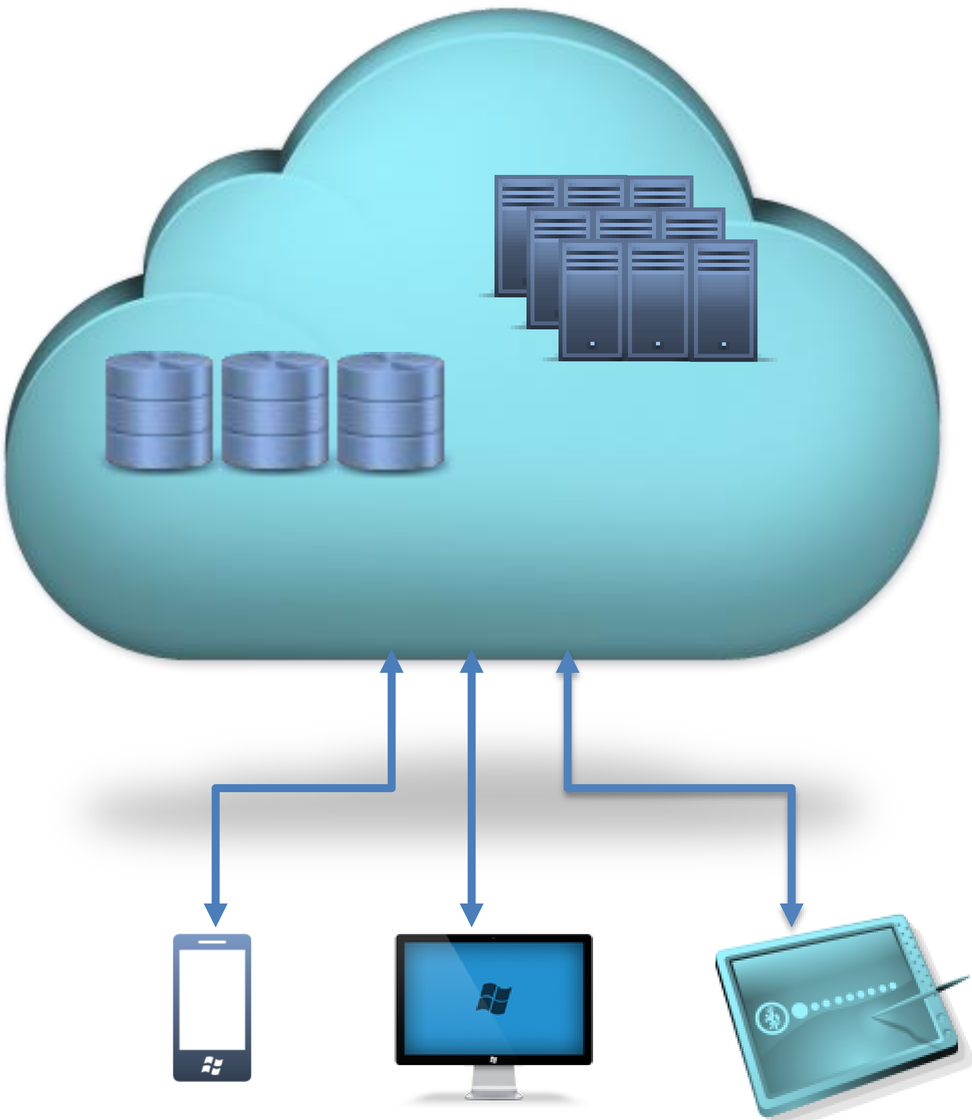
Querying Encrypted Data

Arvind Arasu, Ken Eguro,

Ravi Ramamurthy, Raghav Kaushik

Microsoft Research

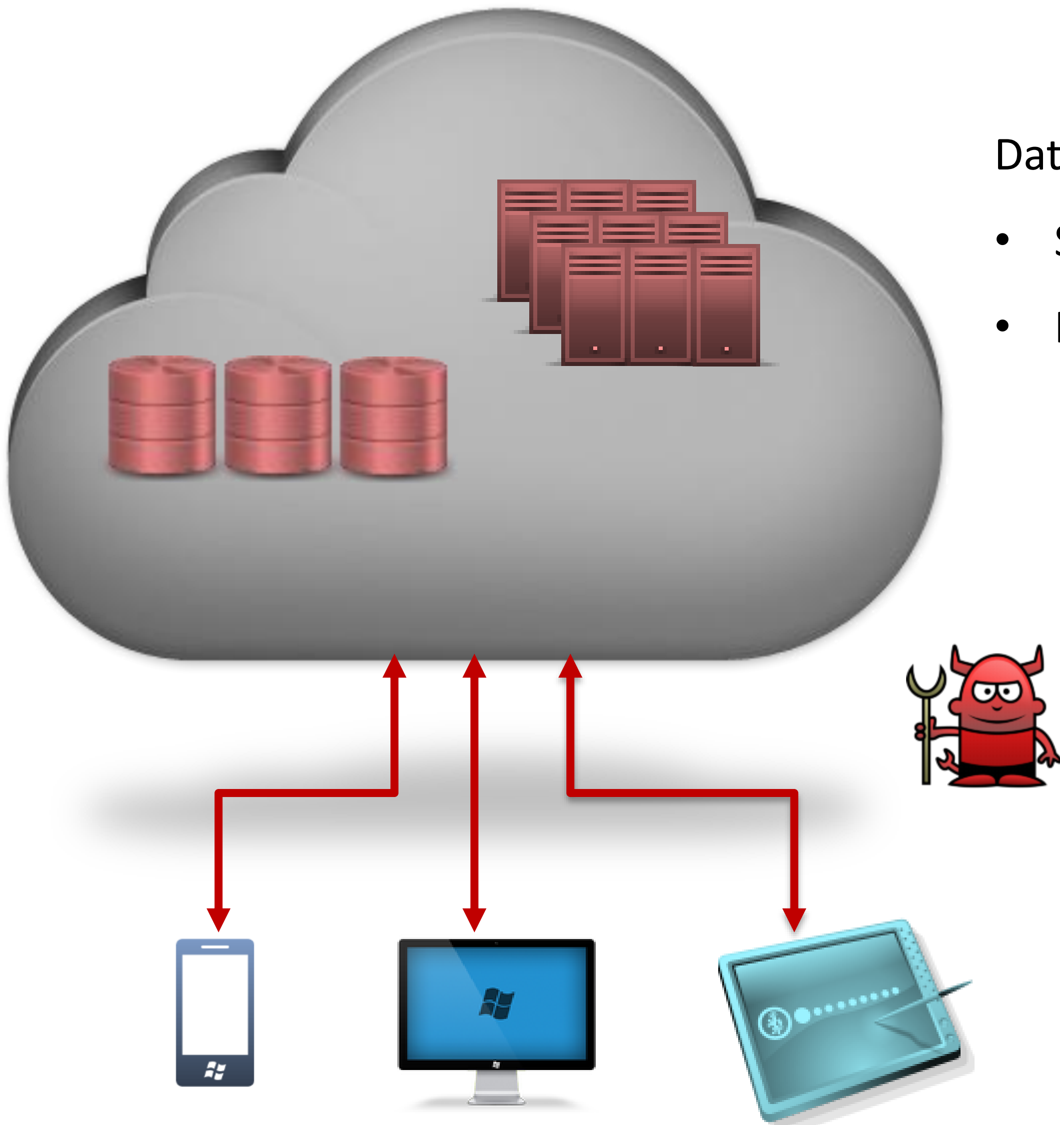
Cloud Computing



- Well-documented benefits
- Trend to move computation and data to cloud
- Database functionality
  - Amazon RDS
  - Microsoft SQL Azure
  - Heroku PostegreSQL
  - Xeround

[AF+09, NIST09]

Security Concerns

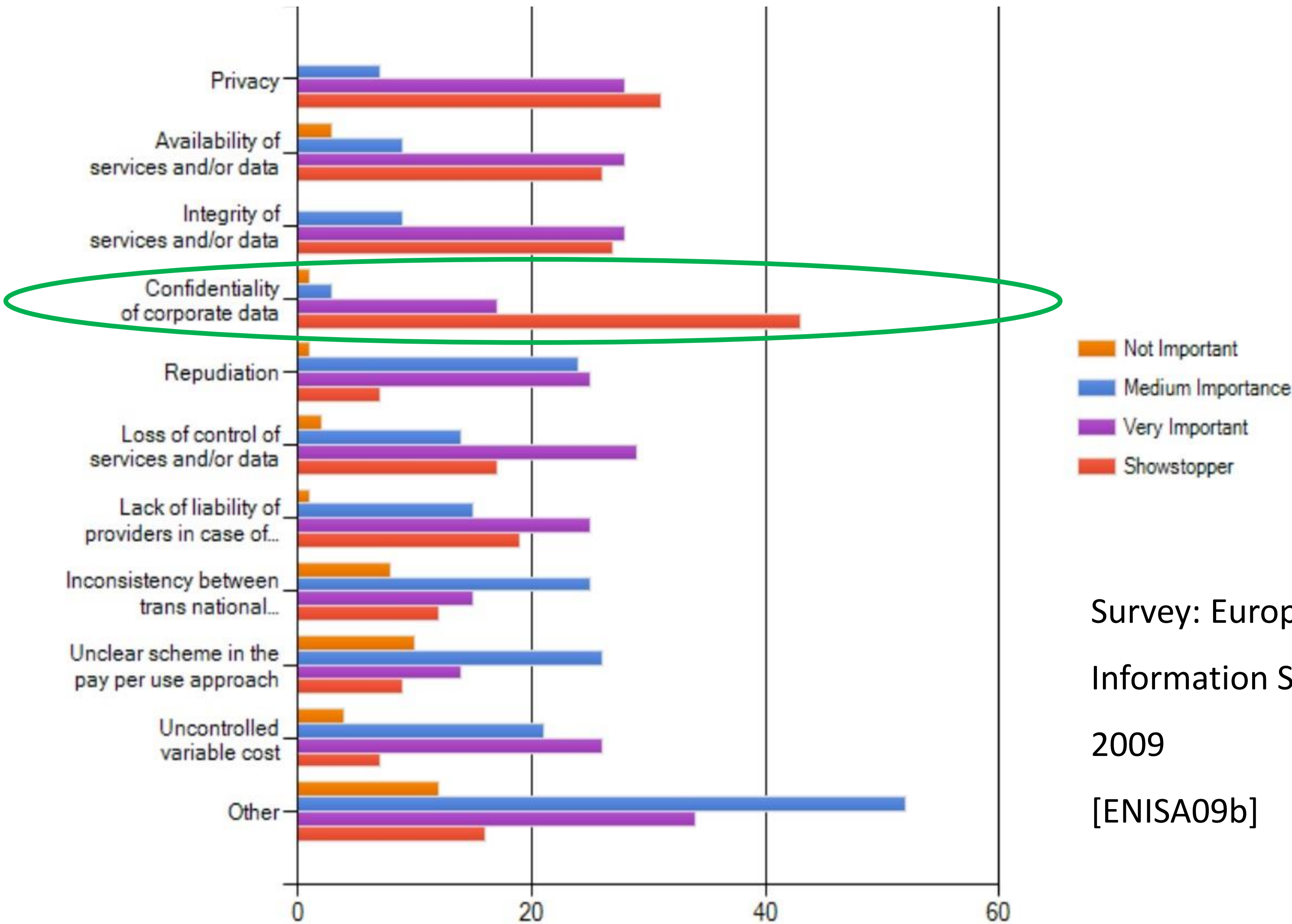


Data in the cloud vulnerable to:

- Snooping administrators
- Hackers with illegal access
  - Compromised servers

[CPK10, ENISA09a]

What are your main cloud computing concerns?



Survey: European Network and Information Security Agency, Nov 2009

[ENISA09b]

Sensitive Data in the Cloud: Examples

Software as a Service Applications

Billing
- Aria Systems
- eVapt
- nDEBIT
- Redi2
- Zuora

CRM
- 37 Signals
- Capsule
- Dynamics
- Intouchcrm
- LiveOps
- Oracle CRM
- Parature
- Responsys
- RO|Enablement
- Salesforce.com
- Save My Table
- Solve 360

ERP
- Acumatica ERP
- Blue Link Elite
- Epicor Express
- NetSuite
- OrderHarmony
- Plex Online

Health
- CECity
- SNO

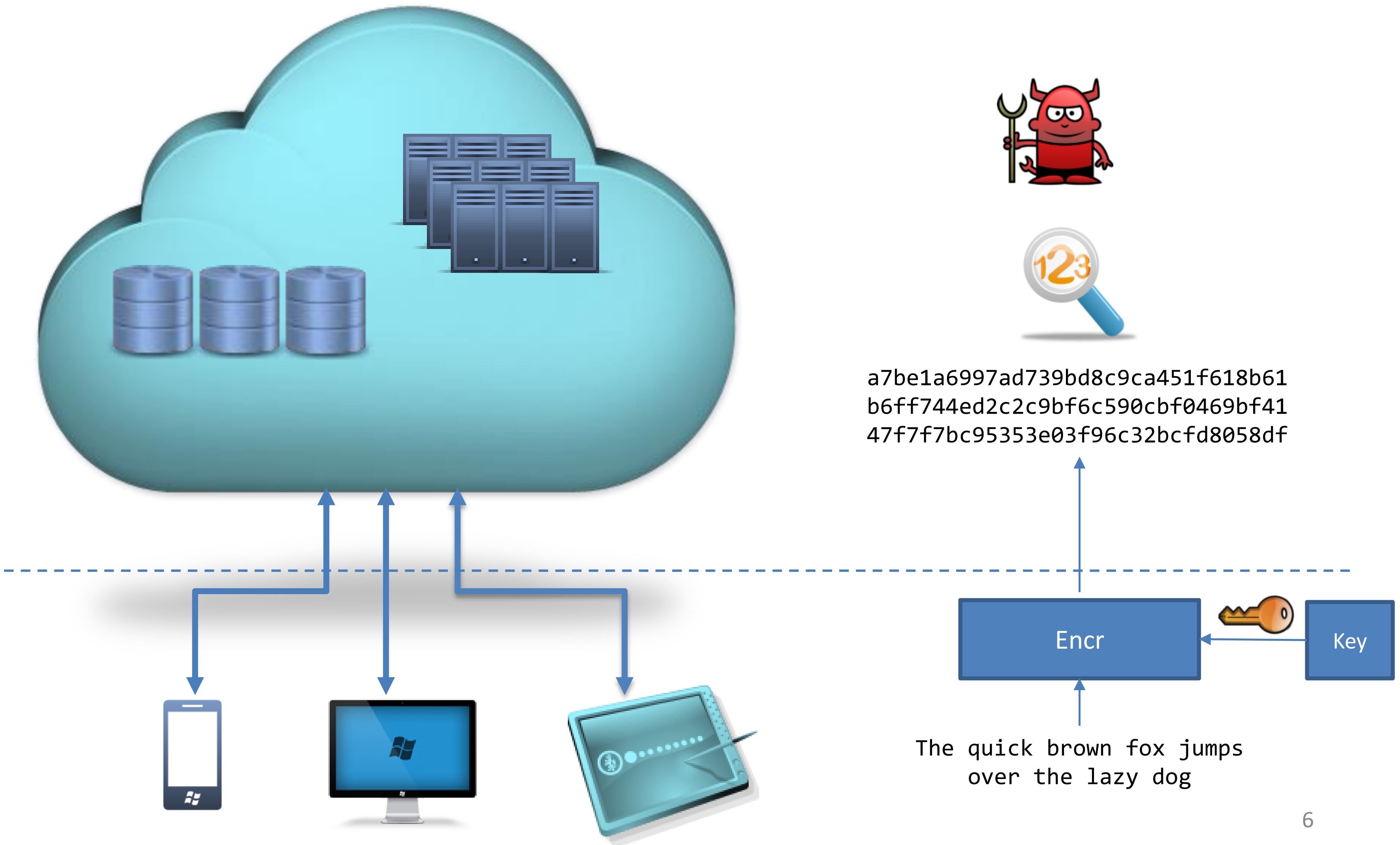
Personal Data
- Google Docs
- Microsoft Office
- Mint.com

Personal data

Corporate data

Source: http://cloudtaxonomy.opencrowd.com/taxonomy/

Data Encryption

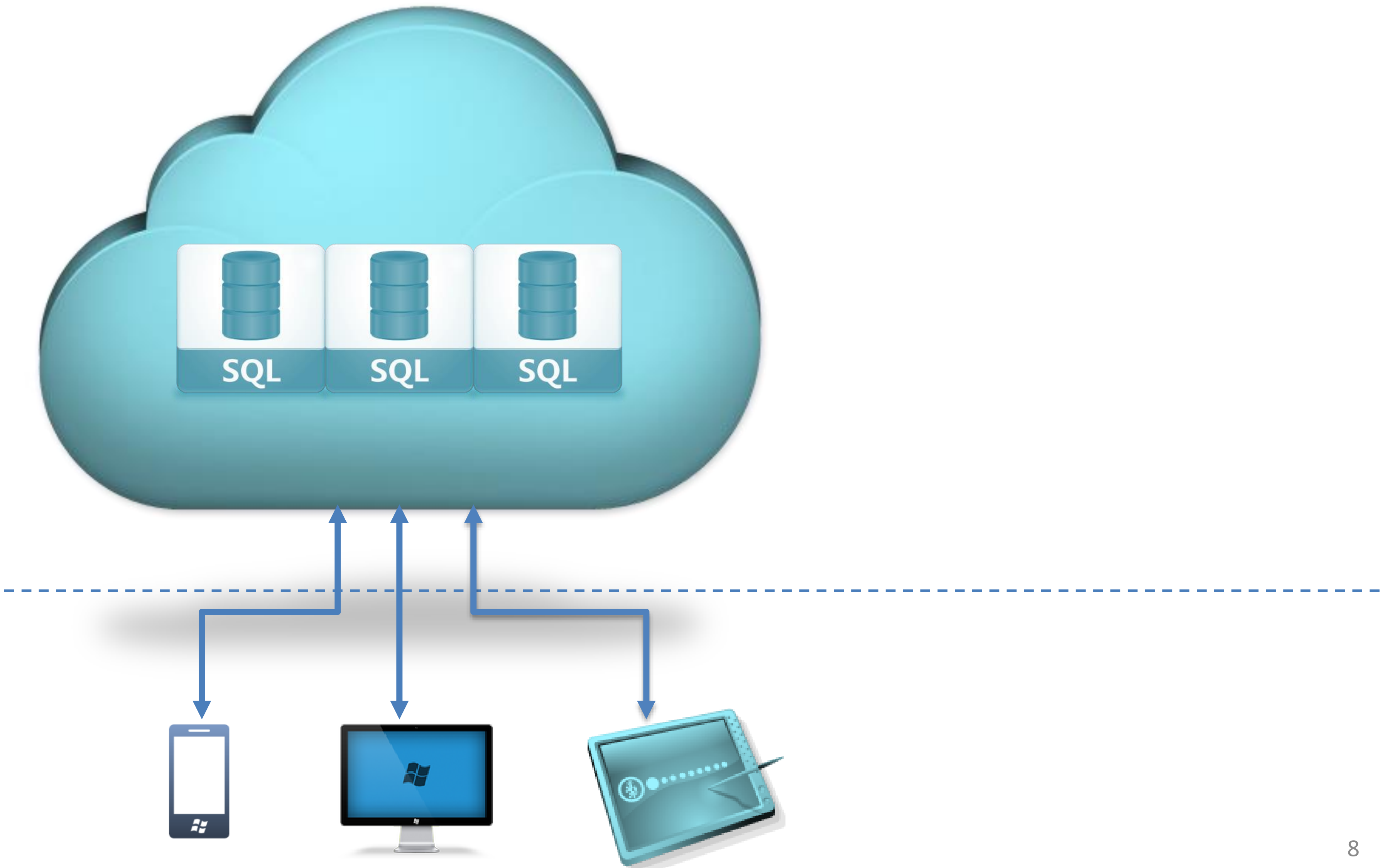


AWS Security Advice

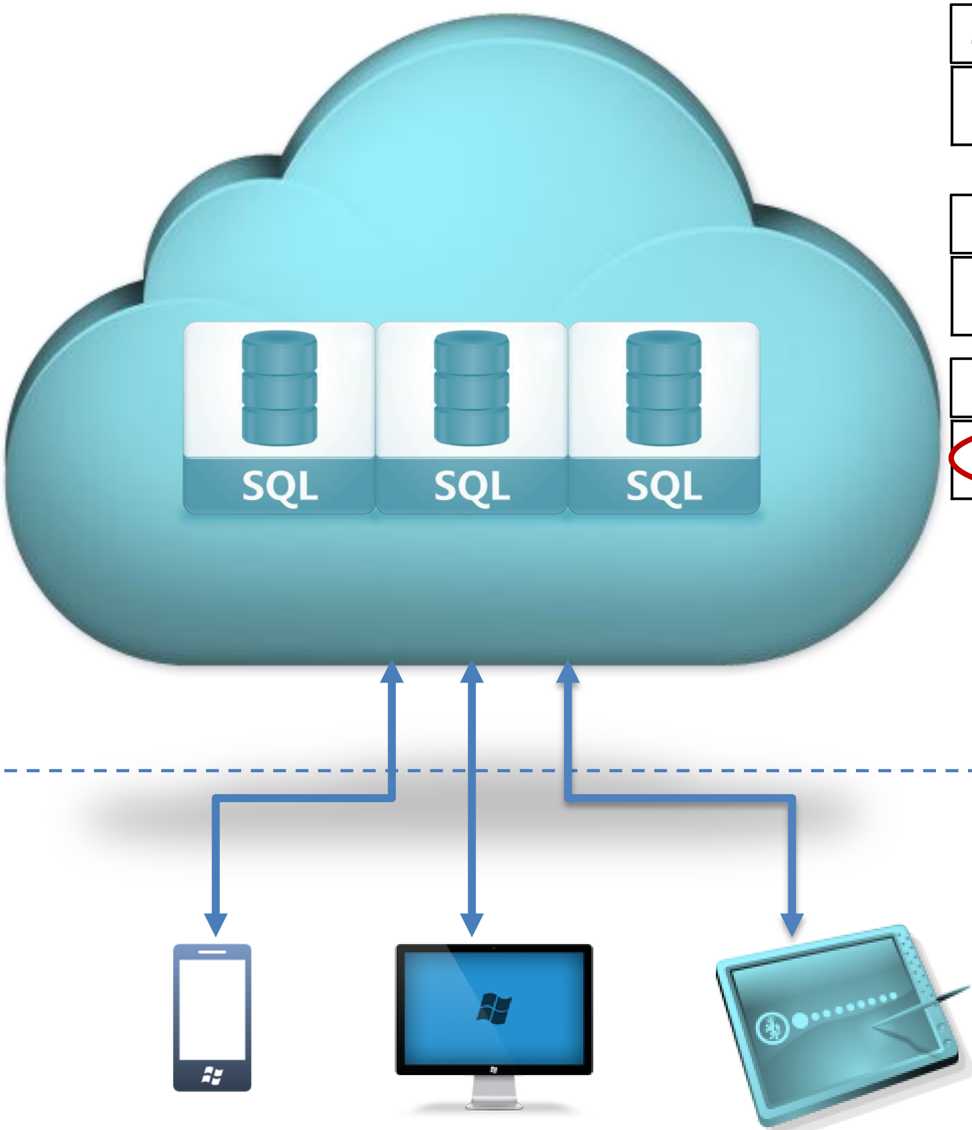
7.2. Security. We strive to keep Your Content secure, but cannot guarantee that we will be successful at doing so, given the nature of the Internet. Accordingly, without limitation to Section 4.3 above and Section 11.5 below, you acknowledge that you bear sole responsibility for adequate security, protection and backup of Your Content. We strongly encourage you, where available and appropriate, to use encryption technology to protect Your Content from unauthorized access and to routinely archive Your Content. We will have no liability to you for any unauthorized access or use, corruption, deletion, destruction or loss of any of Your Content.

Source: http://aws-portal.amazon.com/gp/aws/developer/terms-and-conditions.html

Encryption and DbaaS: Functionality



Example: Online Course Database



Student

StudentId	Name	Addr	GPA	CreditCard	...

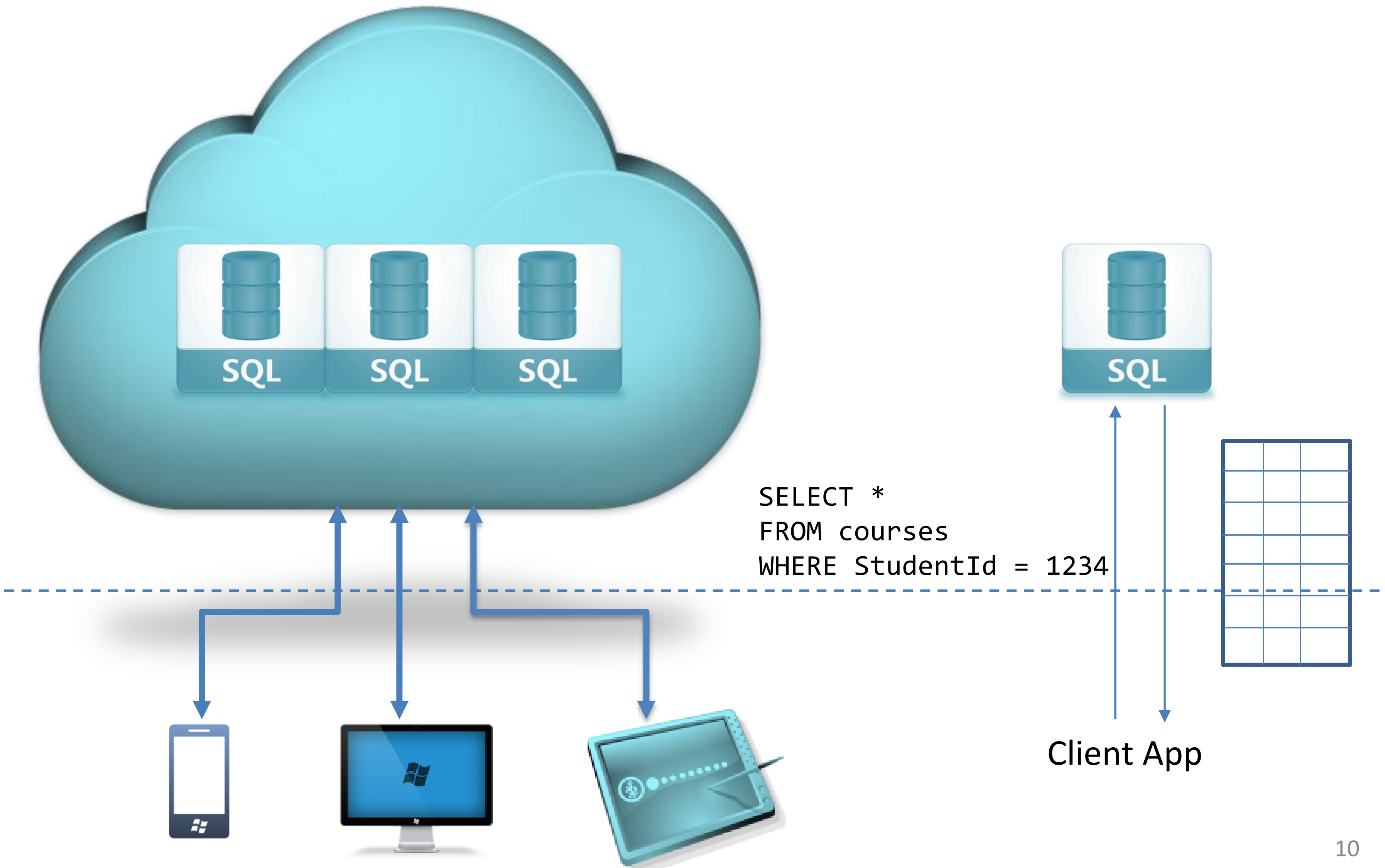
Course

CourseId	Name	InstrId	...

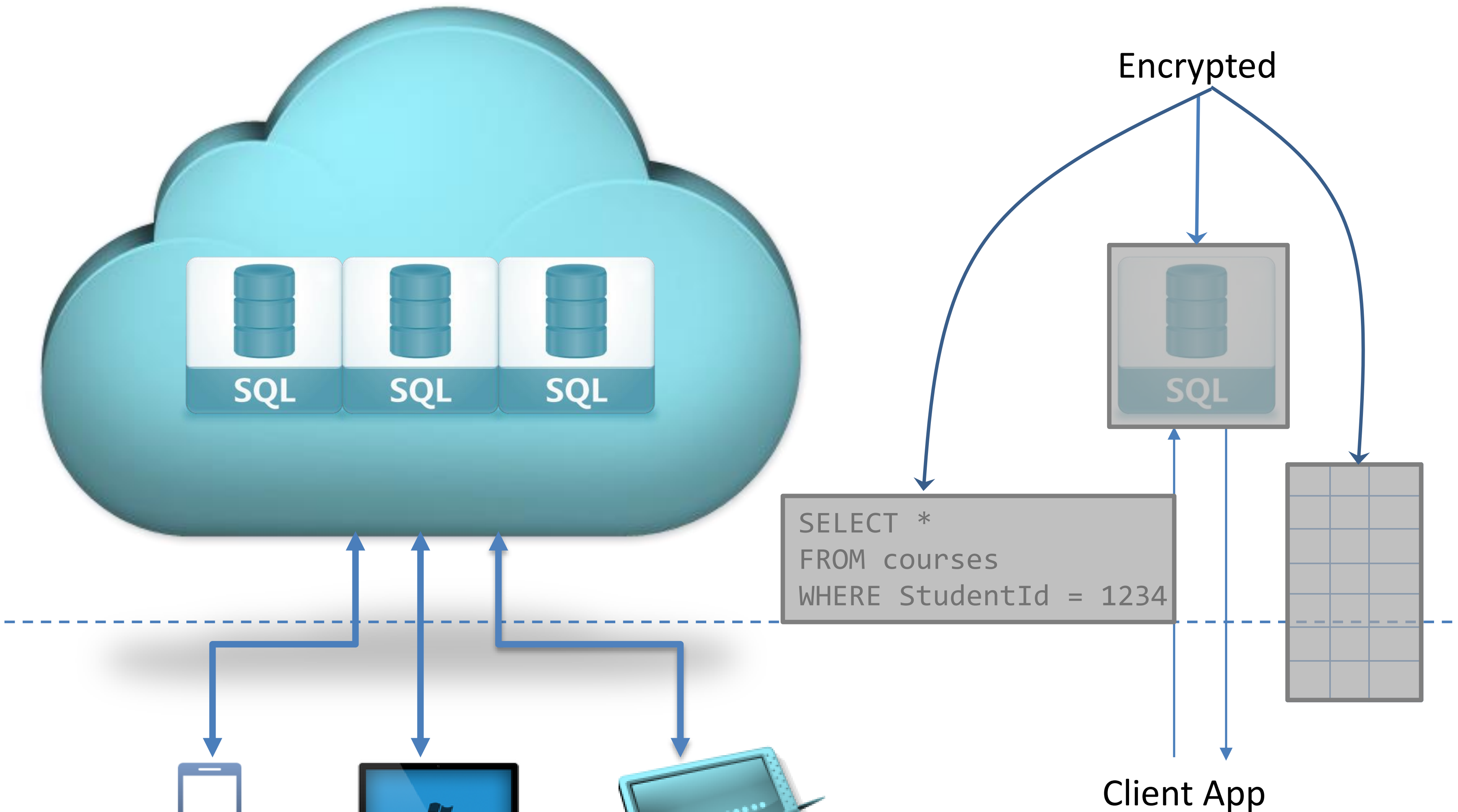
StudentCourse

CourseId	StudentId	Grade	...

Encryption and DbaaS: Functionality



Encryption and DbaaS: Functionality



[HIL+02]
SIGMOD Test of Time Award

Tutorial Overview

- Survey of existing work
  - Building blocks
  - End-to-end systems
    - Security-Performance-Generality tradeoff
    - Taxonomy, organization
- Open problems & Challenges
- Random pontifications

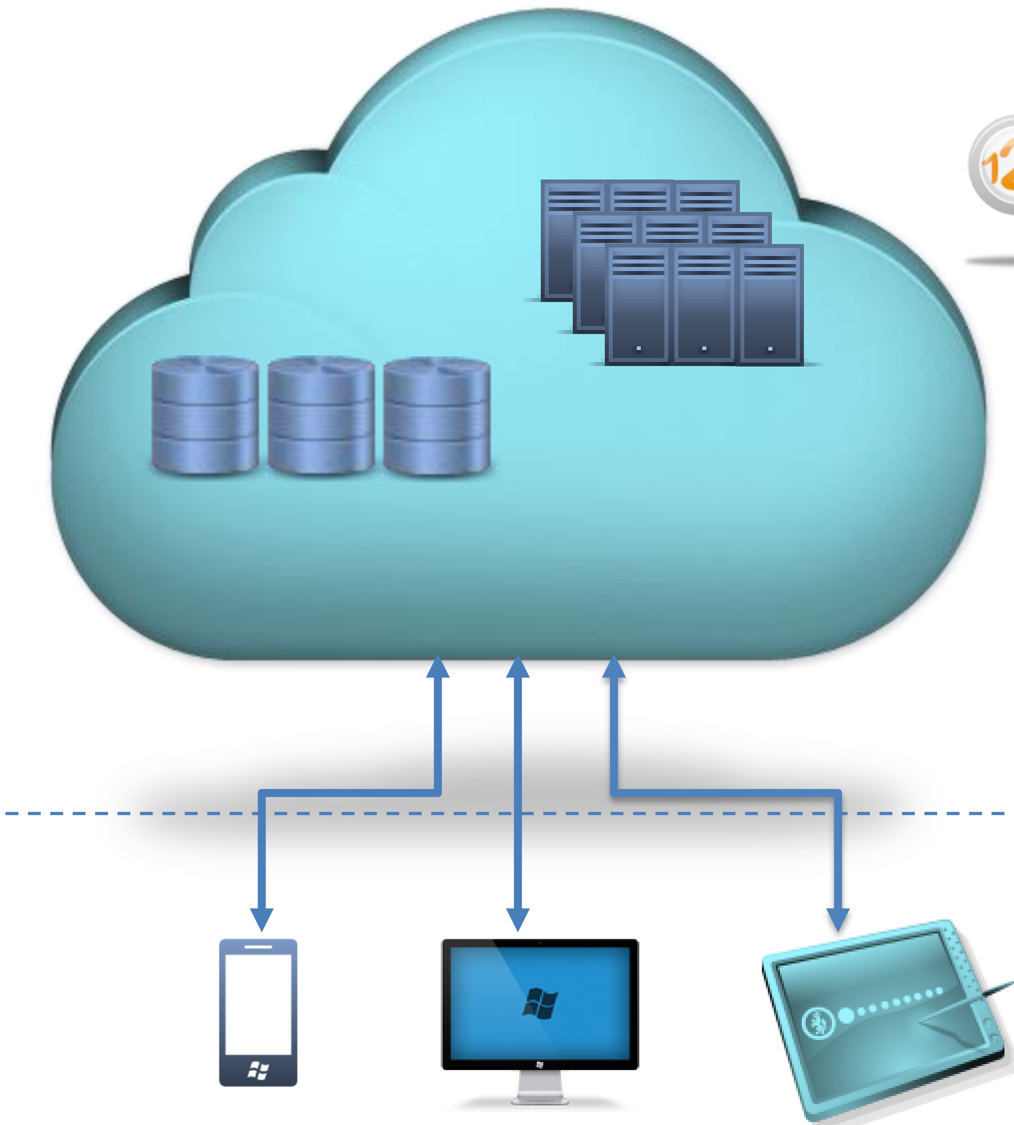
Tutorial Goals & Non-Goals

- Takeaway goals:
  - Interesting & Important area
  - Lots of open (systems) problems
  - Multi-disciplinary
- Non-goals:
  - Latest advances Elliptic Curve Cryptography

Roadmap

- Introduction
- Overview
- Basics of Encryption
- Trusted Client based Systems
- Secure In-Cloud Processing
- Security
- Conclusion

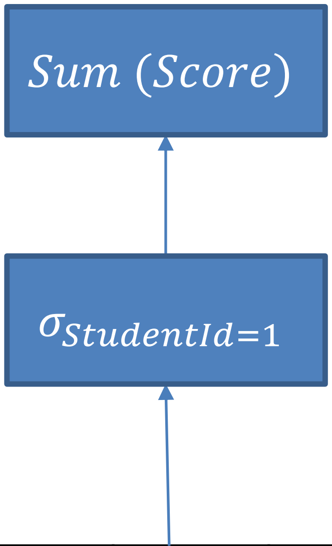
Passive Adversary



- Passive
- Honest but curious
- Does not alter:
  - Database
  - Results

Design systems for active adversary

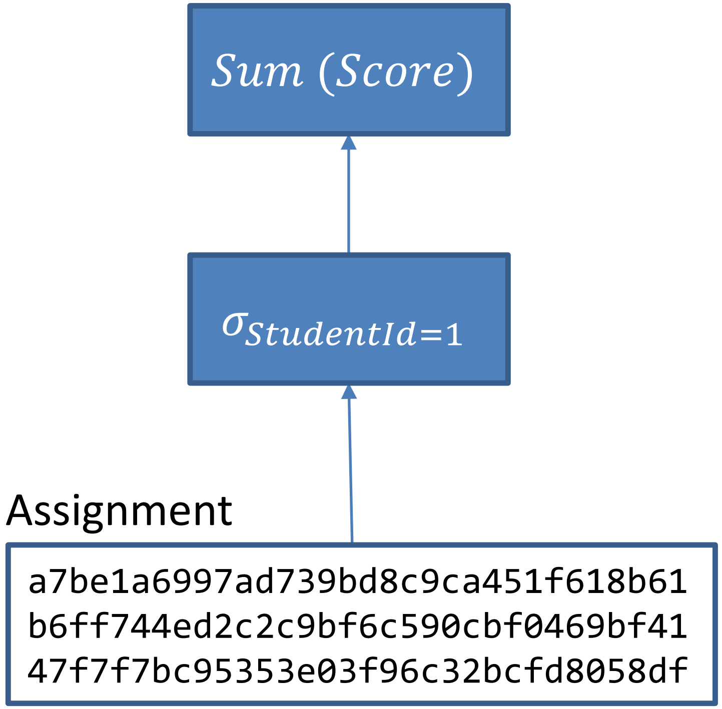
Encryption: Fundamental Challenge



StudentId	AssignId	Score
1	1	68
1	2	71
3	4	99
…	…	…

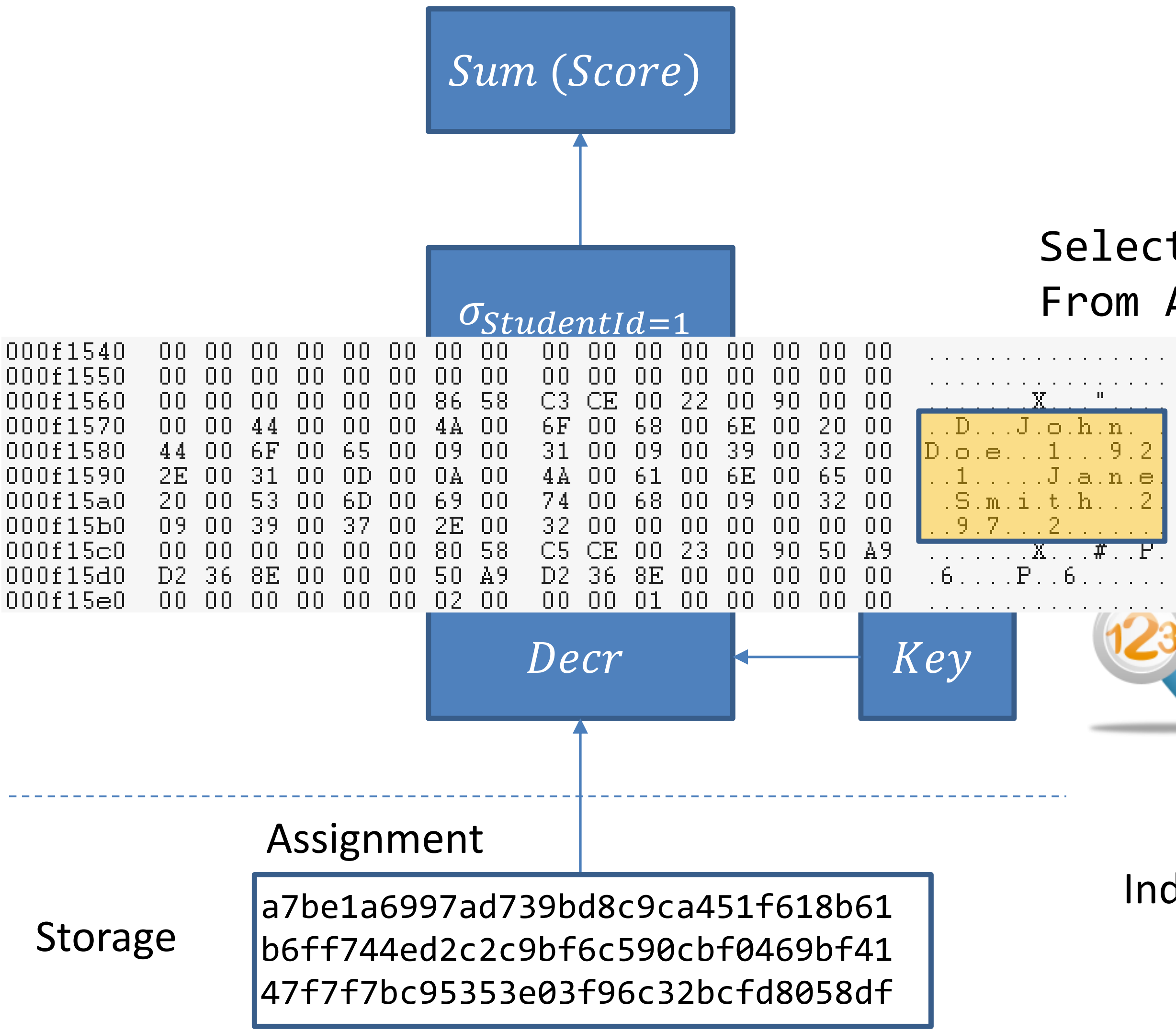
```
Select Sum (Score)
From Assignment
Where StudentId = 1
```

Encryption: Fundamental Challenge




```
Select Sum (Score)
From Assignment
Where StudentId = 1
```

Encryption: Fundamental Challenge



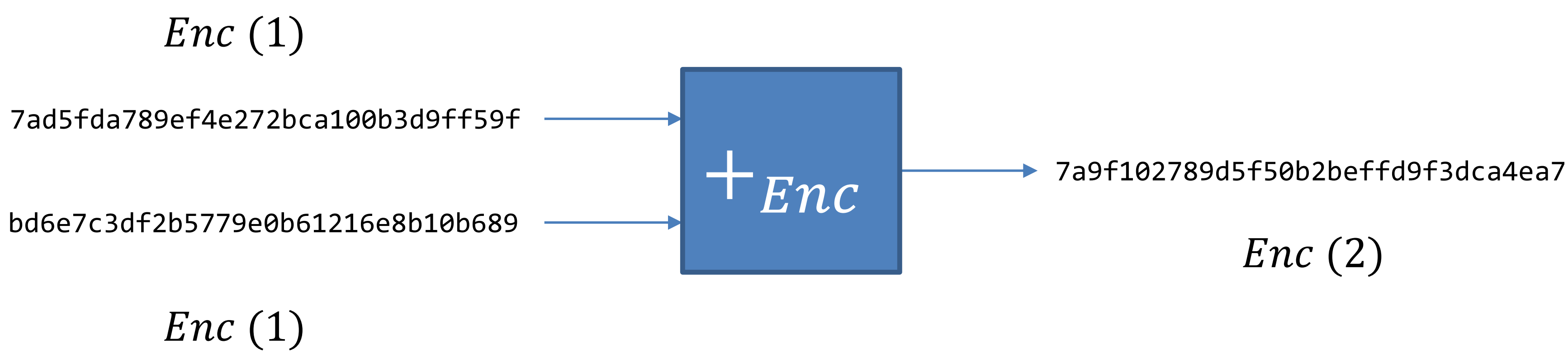

```
Select Sum (Score)
From Assignment
      StudentId = 1
```

Industry state-of-the-art:
[OTDE, STDE]

Solution Landscape

- Two fundamental techniques
  - Directly compute over encrypted data
    - Special *homomorphic* encryption schemes

Homomorphic Encryption

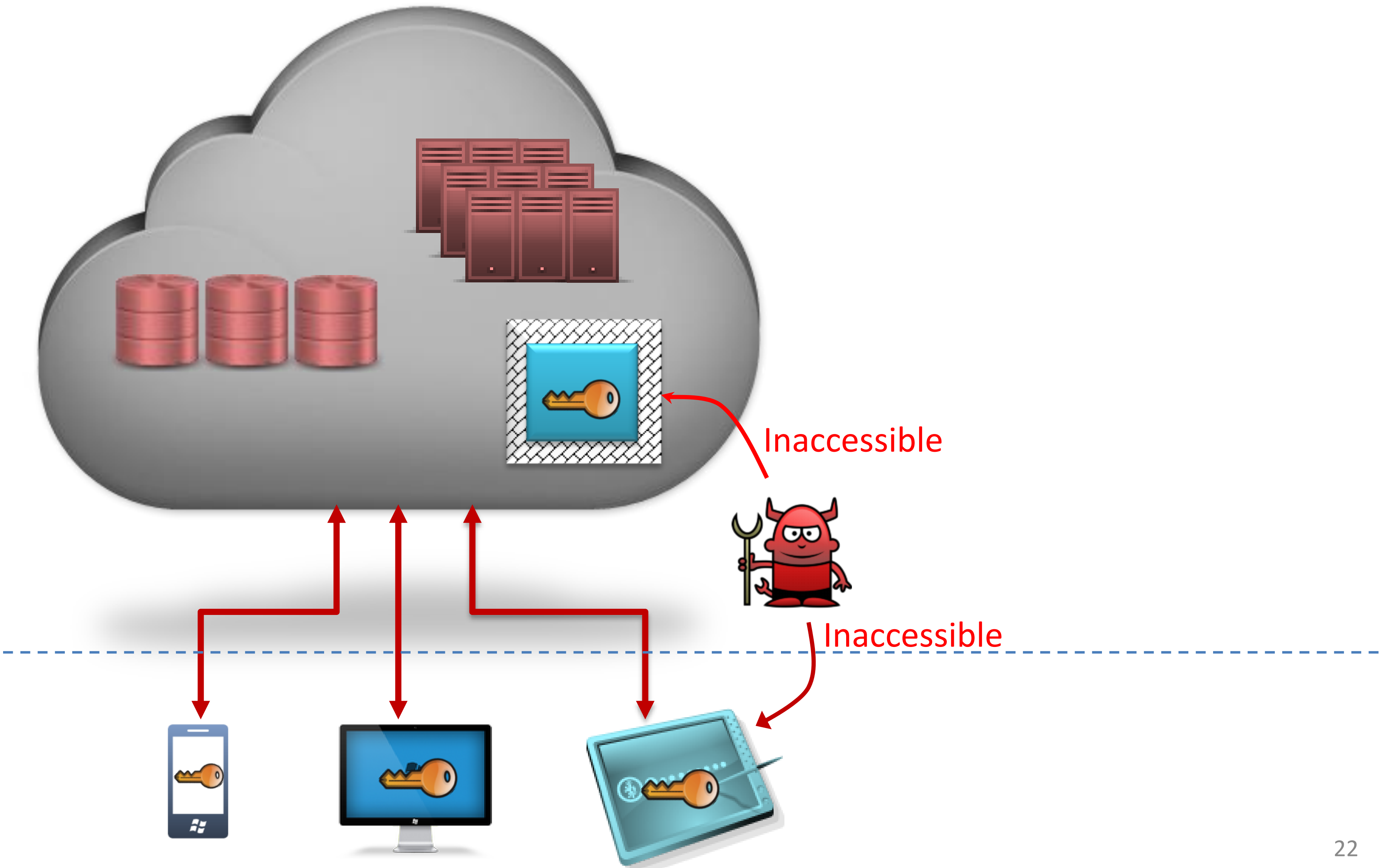


Encryption key is not an input

Solution Landscape

- Two fundamental techniques
  - Directly compute over encrypted data
    - Special *homomorphic* encryption schemes
    - Challenge: limited class of computations
    - Challenge: Not composable
  - Use a “secure” location
    - Computations on plaintext

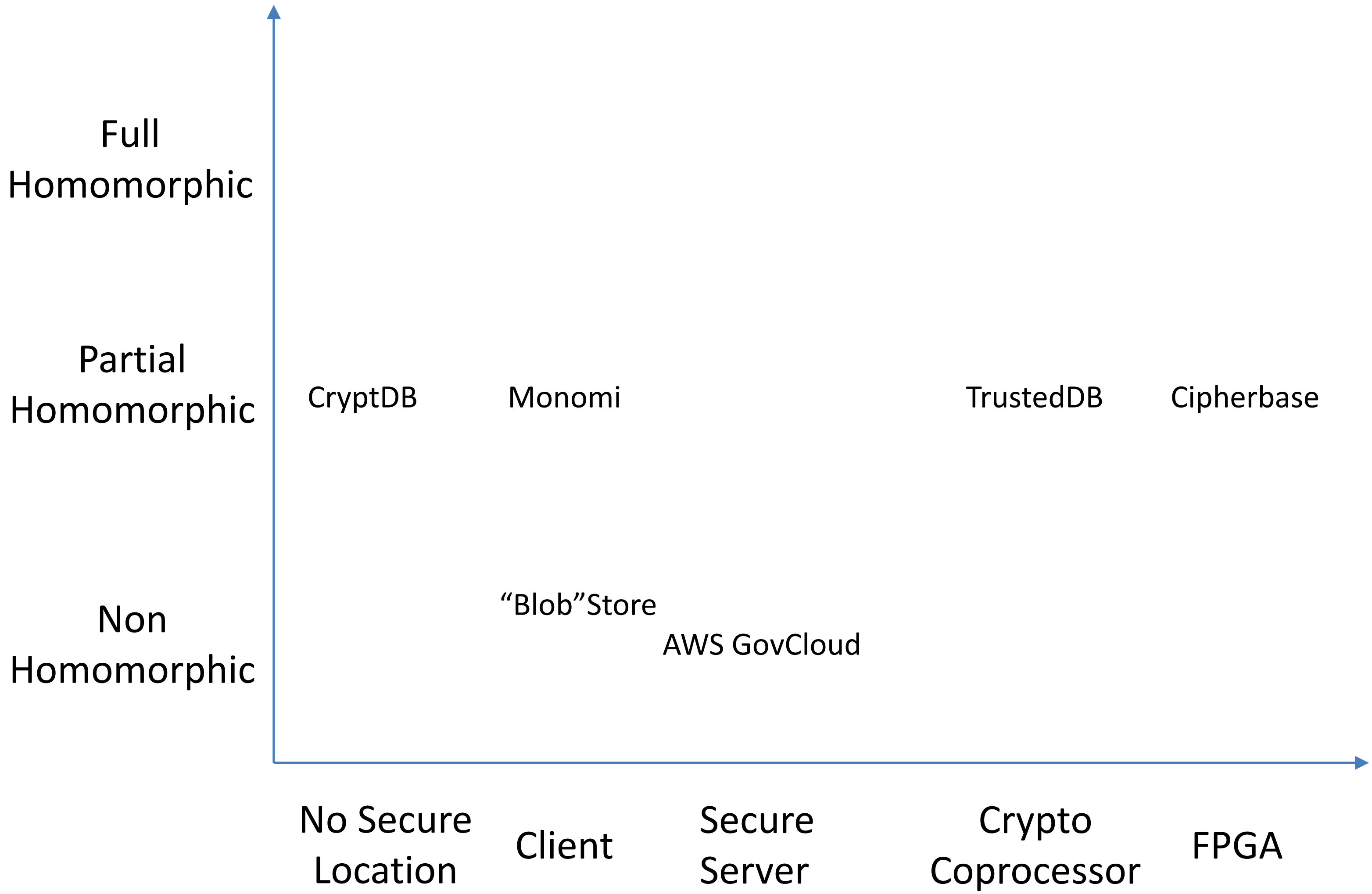
Secure Location



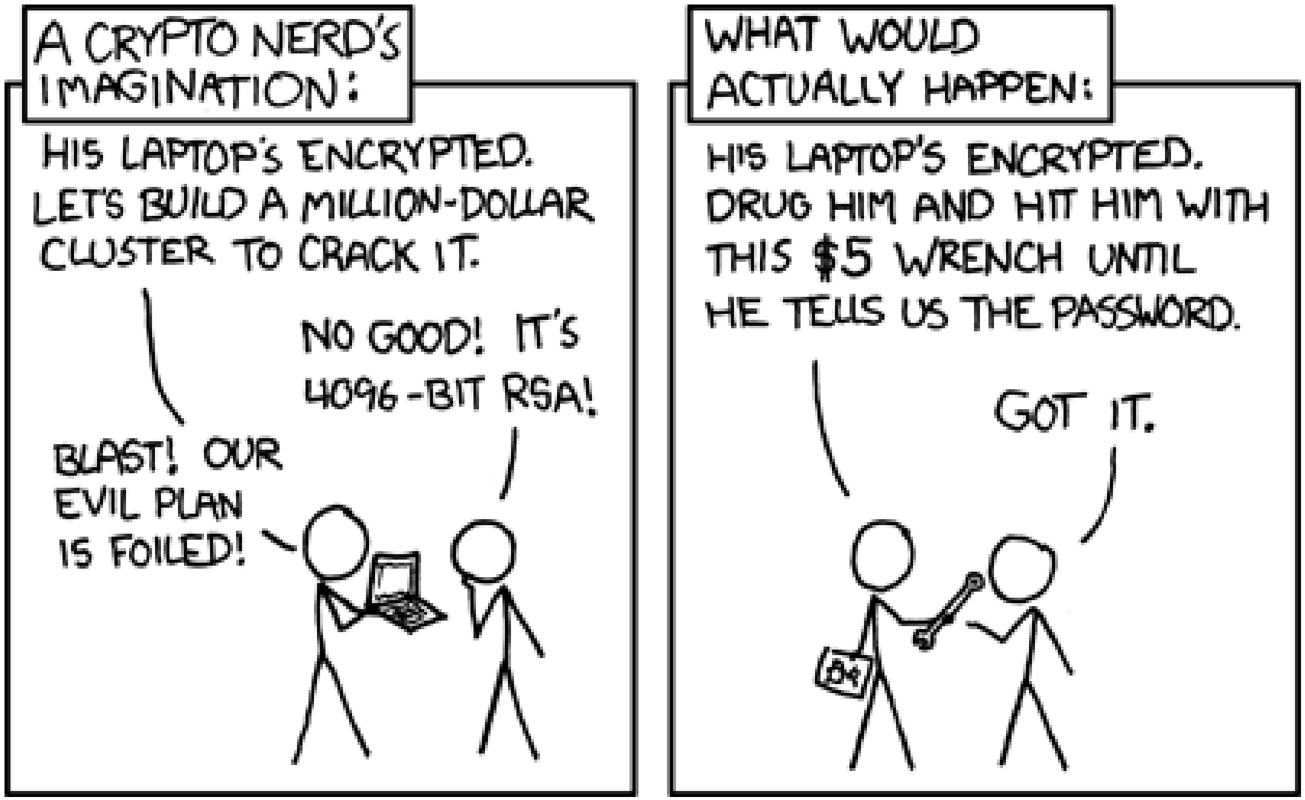
Solution Landscape

- Two fundamental techniques
  - Directly compute over encrypted data
    - Special *homomorphic* encryption schemes
    - Challenge: limited class of computations
  - Use a “secure” location
    - Computations on plaintext
    - Challenge: Expensive
      - Build a plane from black box material

Systems Landscape



Encryption == Security?

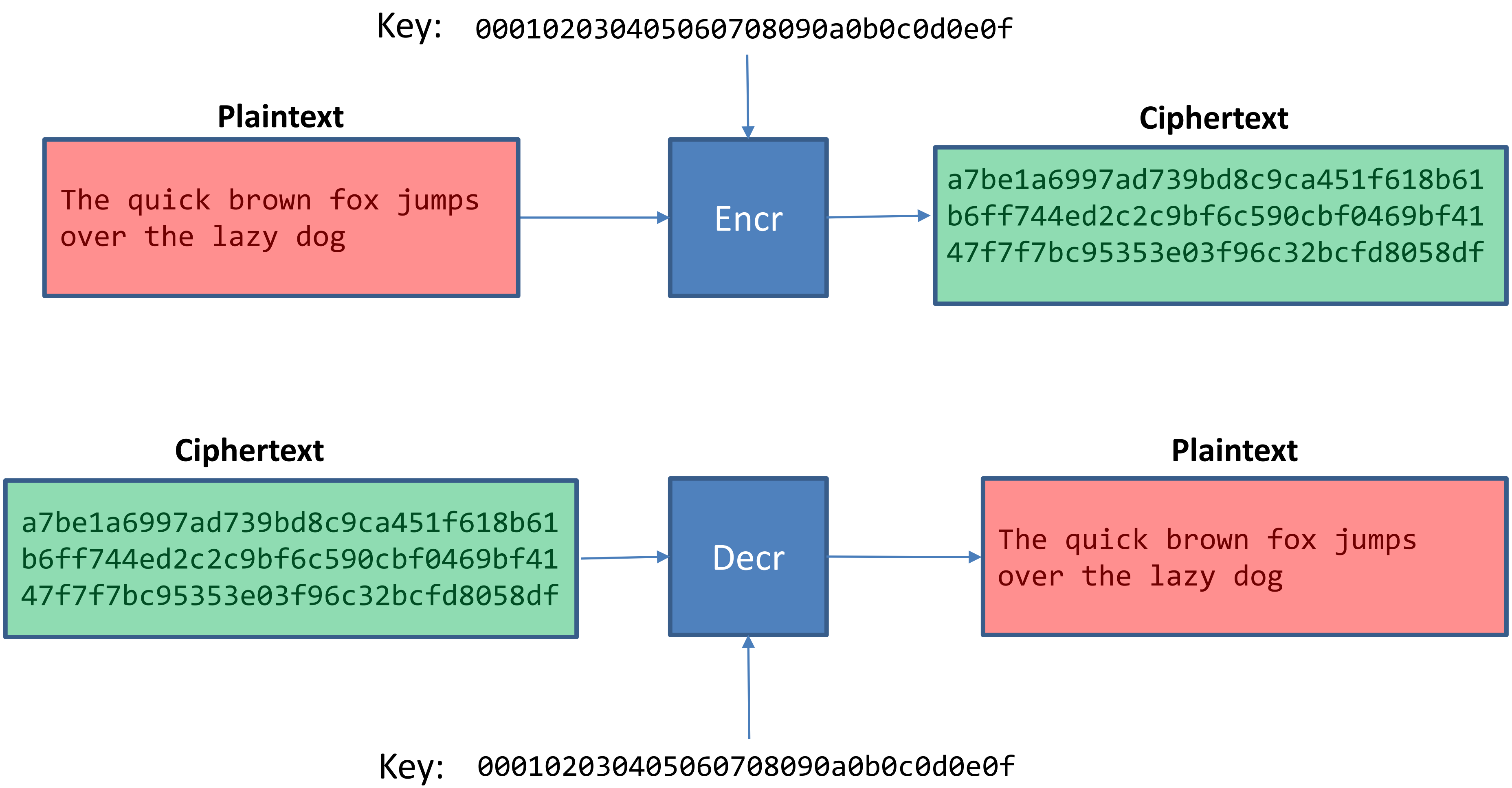


Source: http://xkcd.com/538/

Roadmap

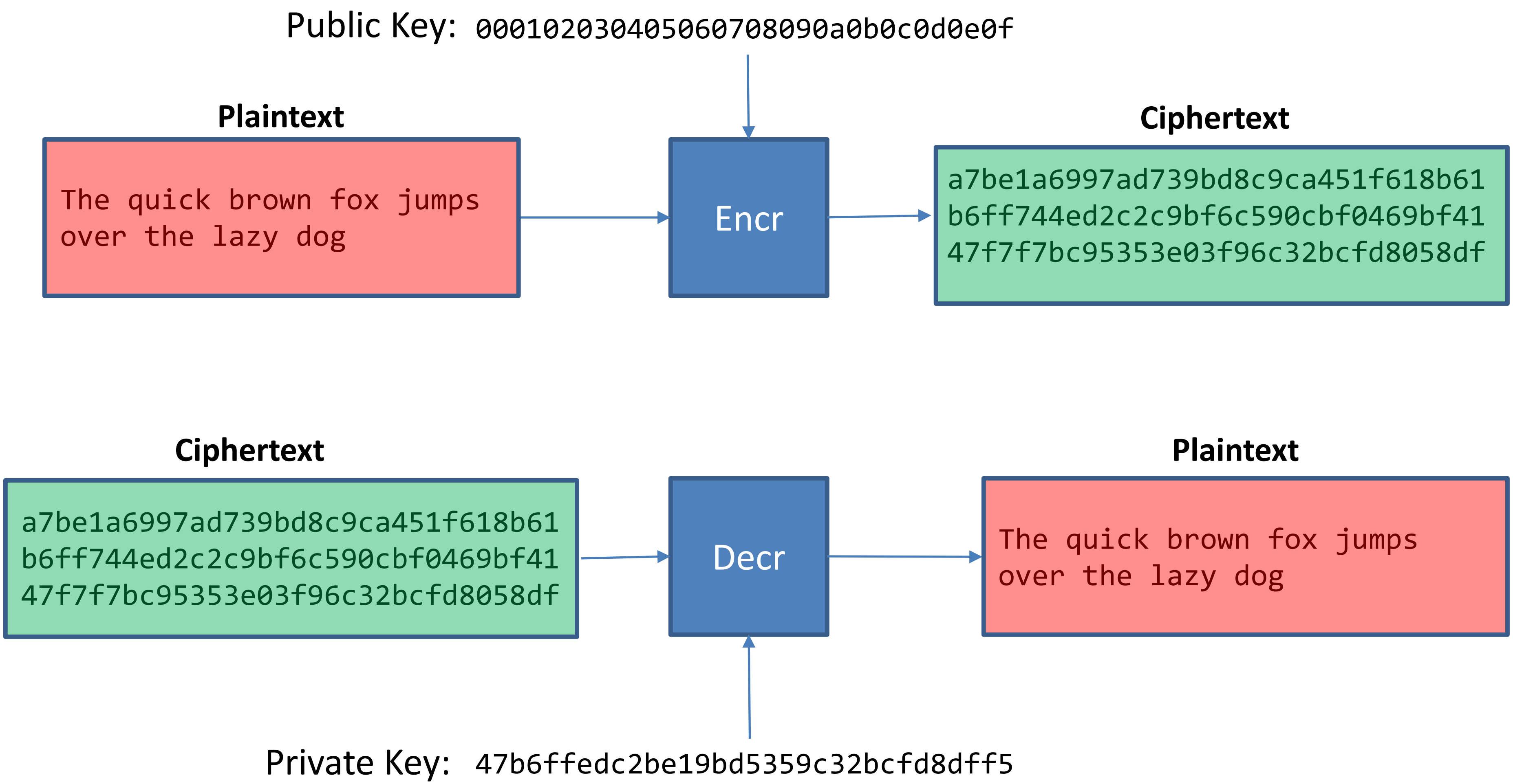
- Introduction
- Overview
- Basics of Encryption
- Trusted Client based Systems
- Secure In-Cloud Processing
- Security
- Conclusion

Encryption Scheme



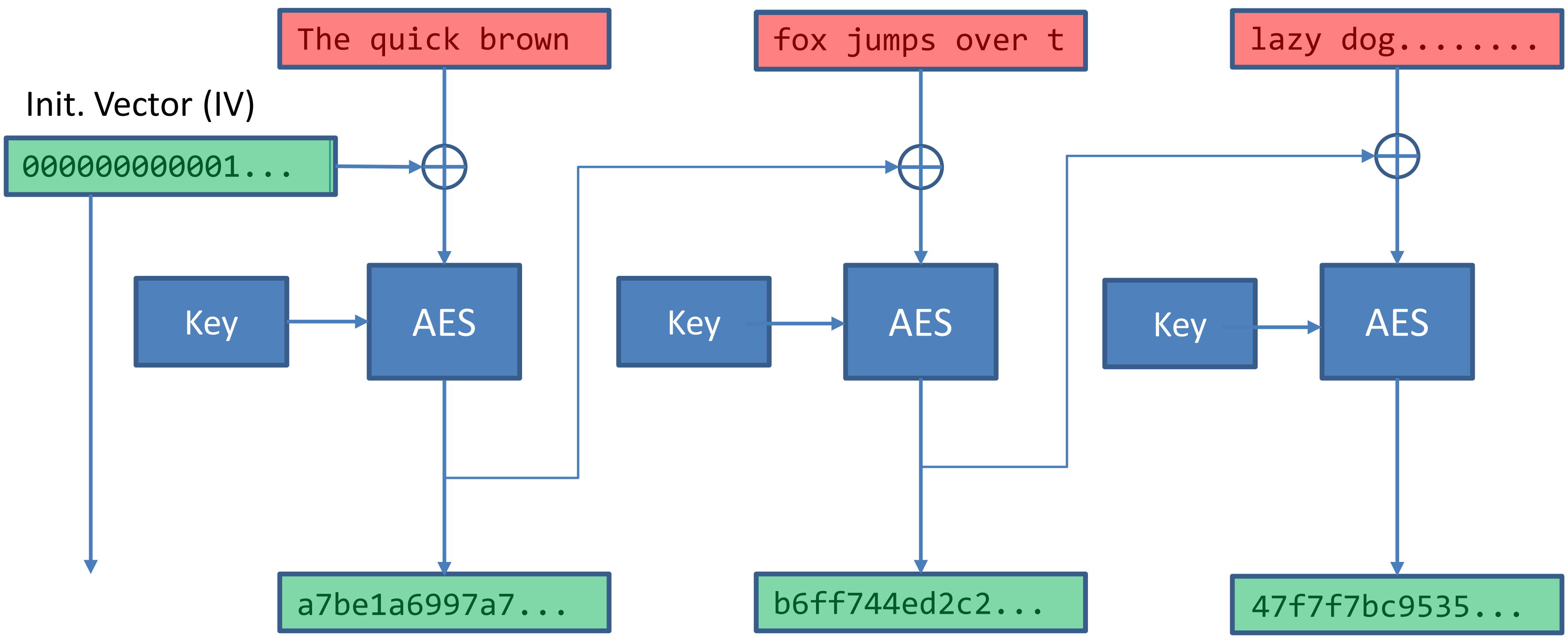
Crypto Textbook: [KL 07]

Encryption Scheme



Crypto Textbook: [KL 07]

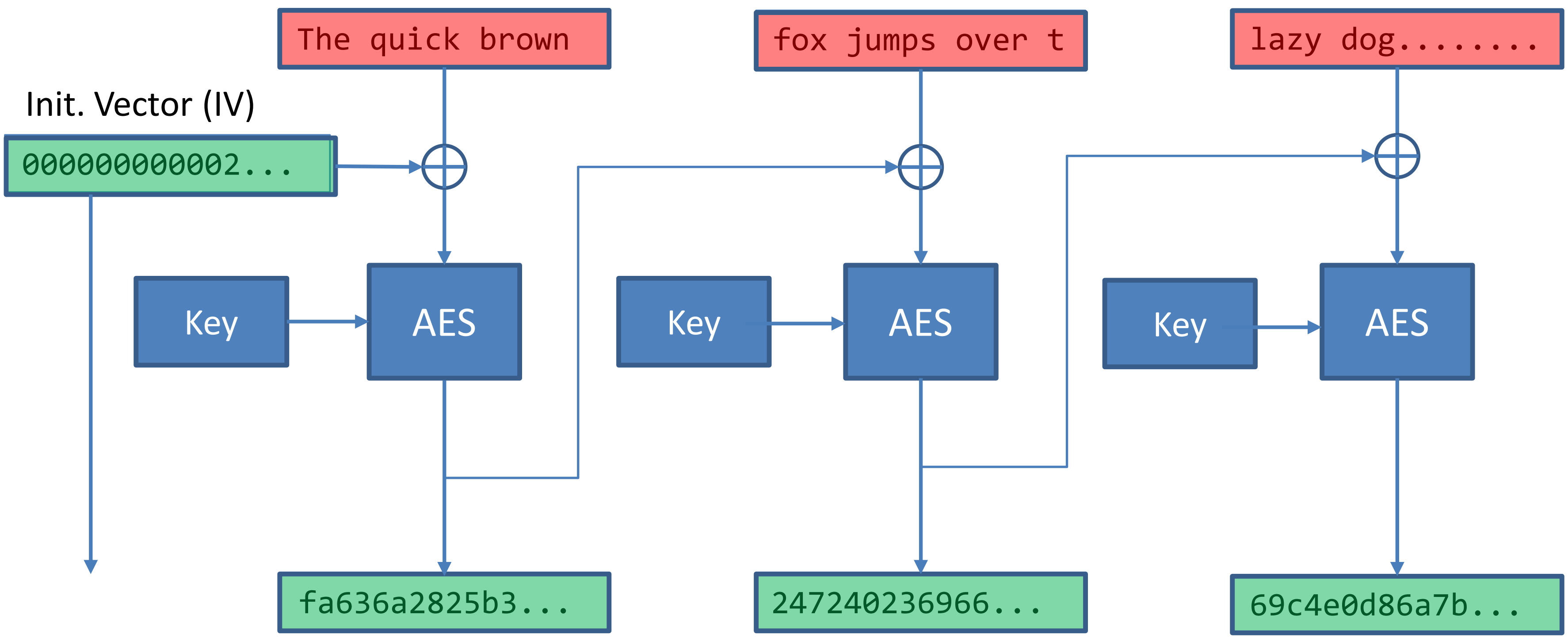
AES + CBC Mode



Variable IV => Non-deterministic

[AES, KL 07]

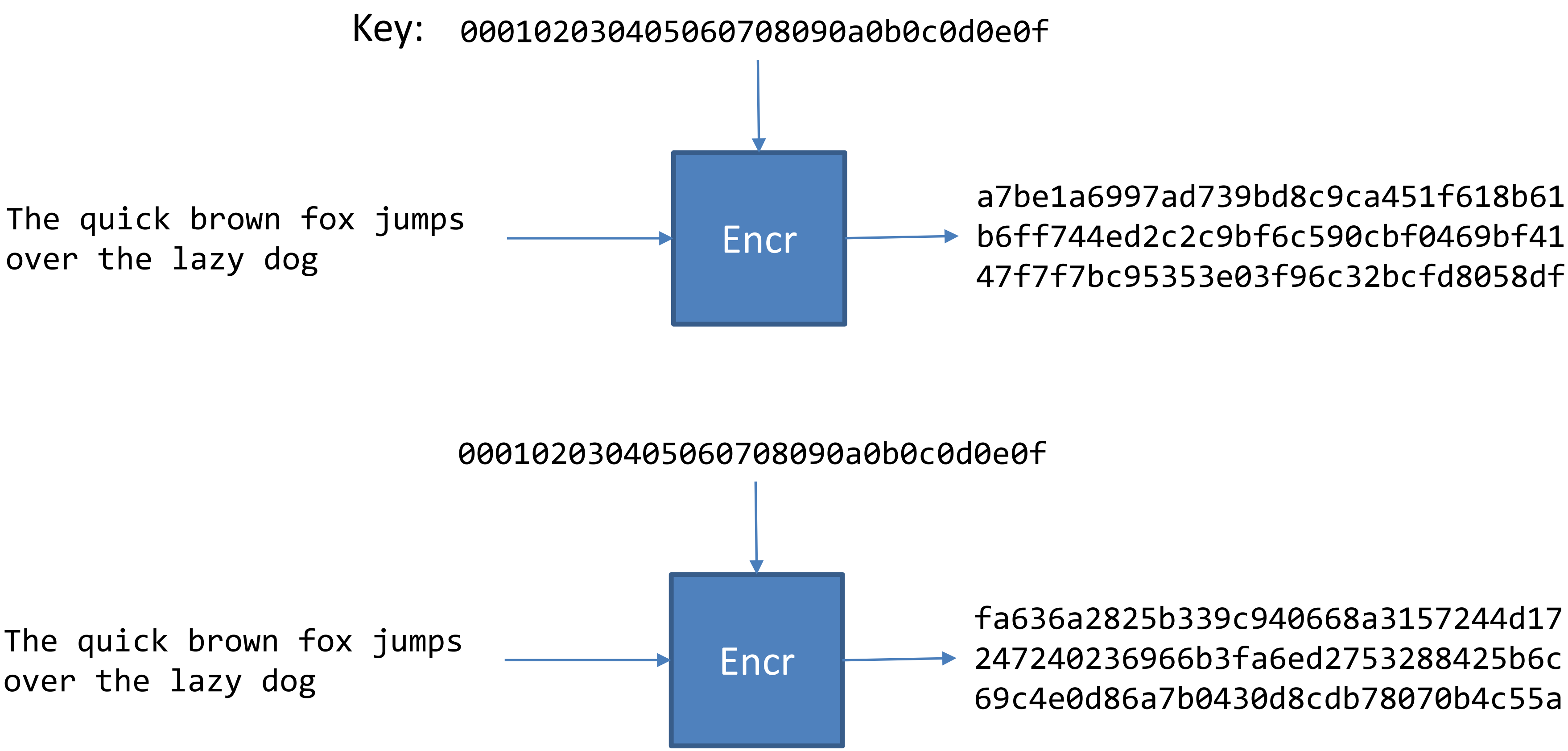
AES + CBC Mode



Variable IV => Non-deterministic

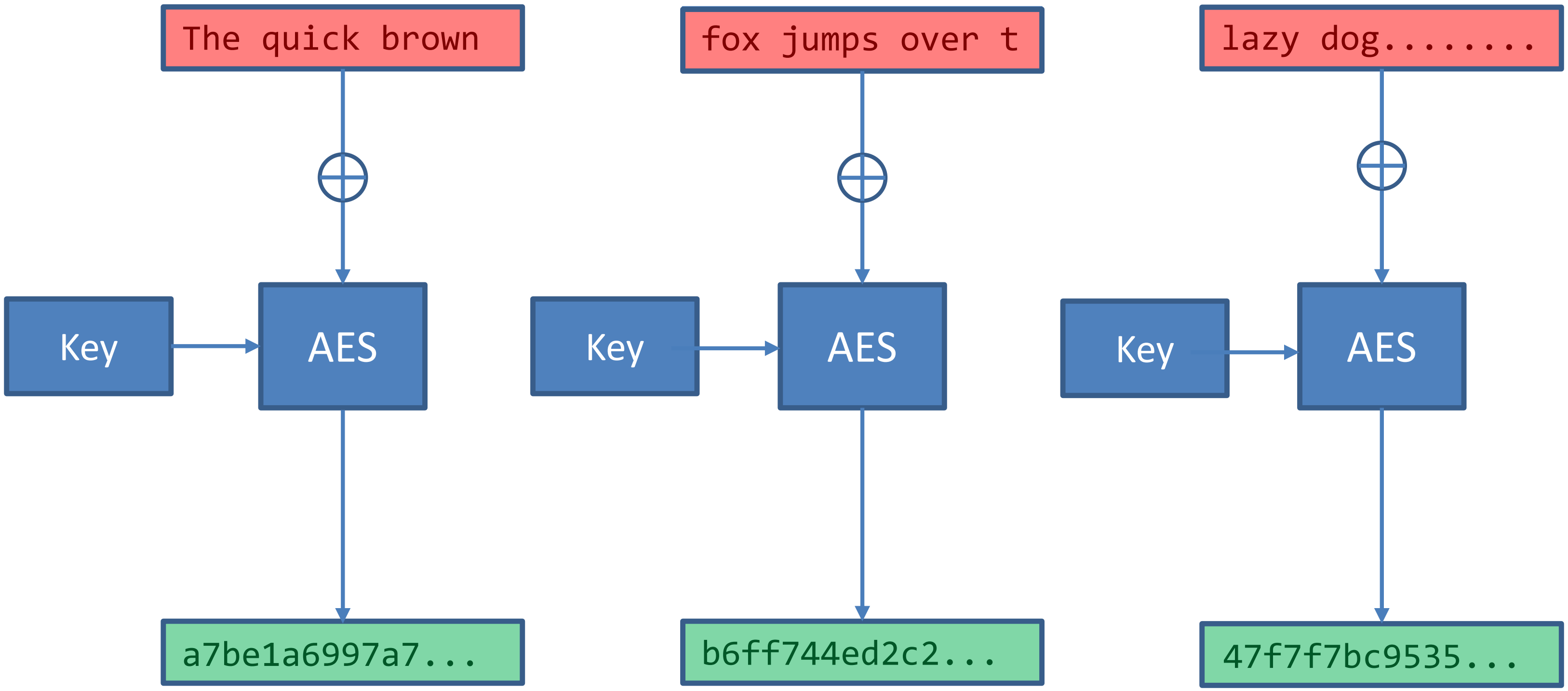
[AES, KL 07]

Nondeterministic Encryption Scheme



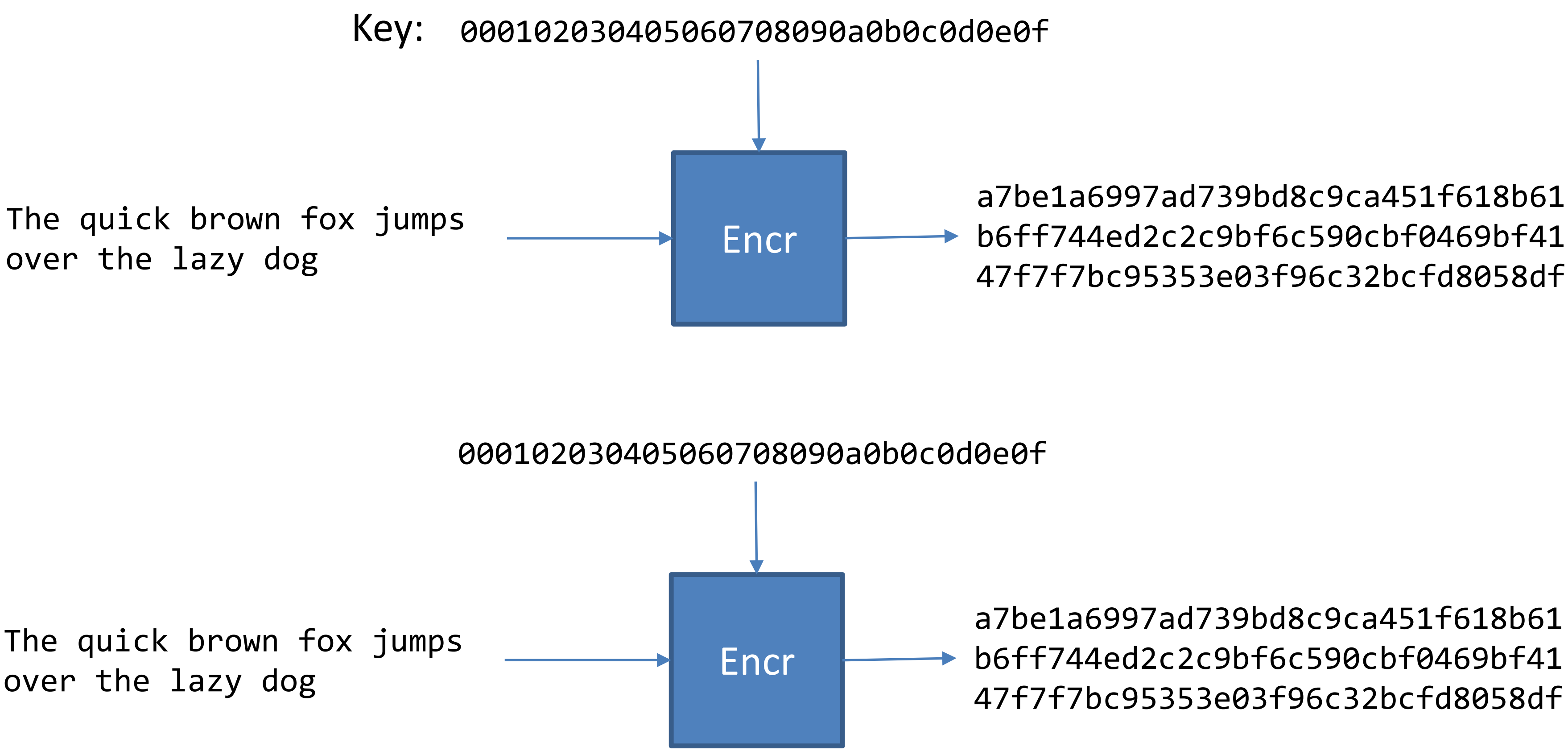
Example: AES + CBC + variable IV

AES + ECB Mode



[AES, KL 07]

Deterministic Encryption Scheme

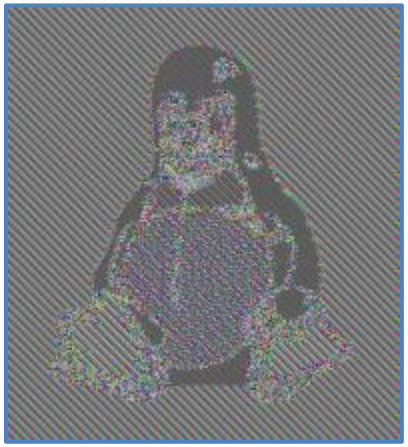


Example: AES + ECB

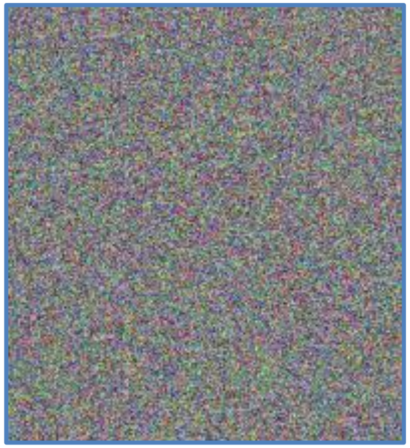
More secure deterministic encryption: [PRZ+11]

Strong Security => Non-Deterministic

Original



Deterministic

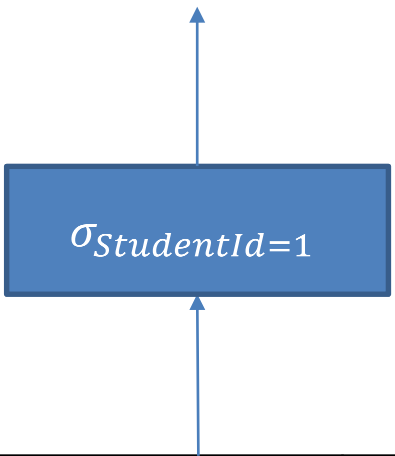


Non-Deterministic

Source: http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

Deterministic Encryption

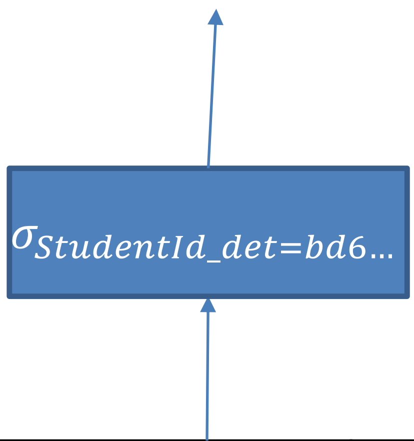
```
select *
from assignment
where studentid = 1
```



StudentId	AssignId	Score
1	1	68
1	2	71
3	4	99
…	…	…

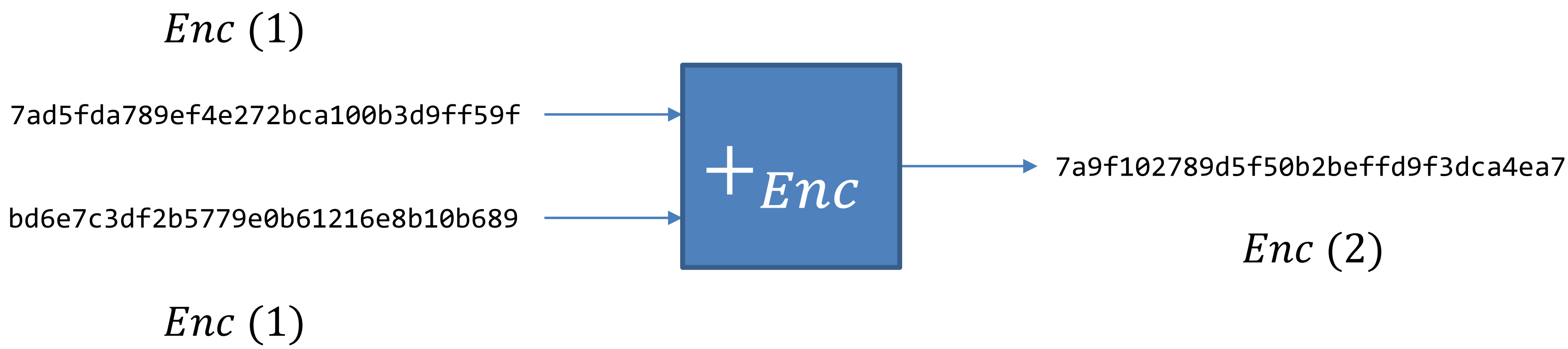
Deterministic Encryption

```sql
select *
from assignment
where studentid_det = bd6e7c3df2b5779e0b61216e8b10b689
```



StudentId_DET	AssignId	Score
bd6e7c3df2b5779e0b61216e8b10b689	1	68
bd6e7c3df2b5779e0b61216e8b10b689	2	71
7ad5fda789ef4e272bca100b3d9ff59f	4	99
…	…	…

Homomorphic Encryption



Encryption key is not an input

Order Preserving Encryption

Value	Enc (Value)
1	0x0001102789d5f50b2beffd9f3dca4ea7
2	0x0065fda789ef4e272bcf102787a93903
3	0x009b5708e13665a7de14d3d824ca9f15
4	0x04e062ff507458f9be50497656ed654c
5	0x08db34fb1f807678d3f833c2194a759e

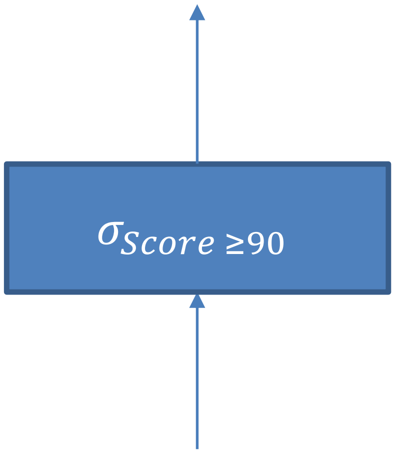
$$x < y \rightarrow Enc\ (x) < Enc\ (y)$$

[AKSX04, BCN11, PLZ13]

SIGMOD Test of Time Award 2014

Order-Preserving Encryption

```sql
select *
from assignment
where score >= 90
```



StudentId	AssignId	Score
1	1	68
1	2	71
3	4	99
…	…	…

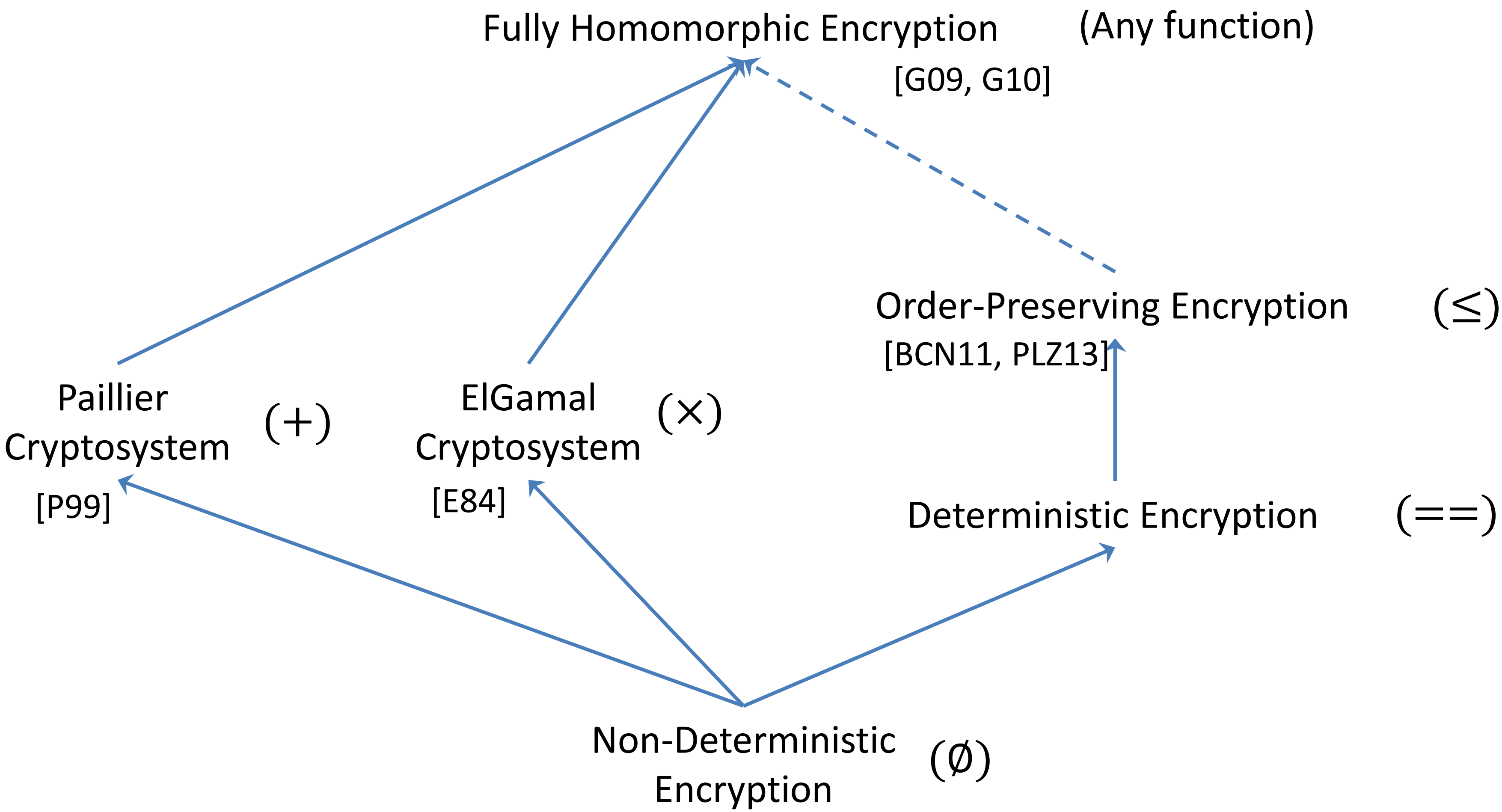
Order-Preserving Encryption

```sql
select *
from assignment
where score_OPE >= 0x04e062ff507458f9be50497656ed654c
```

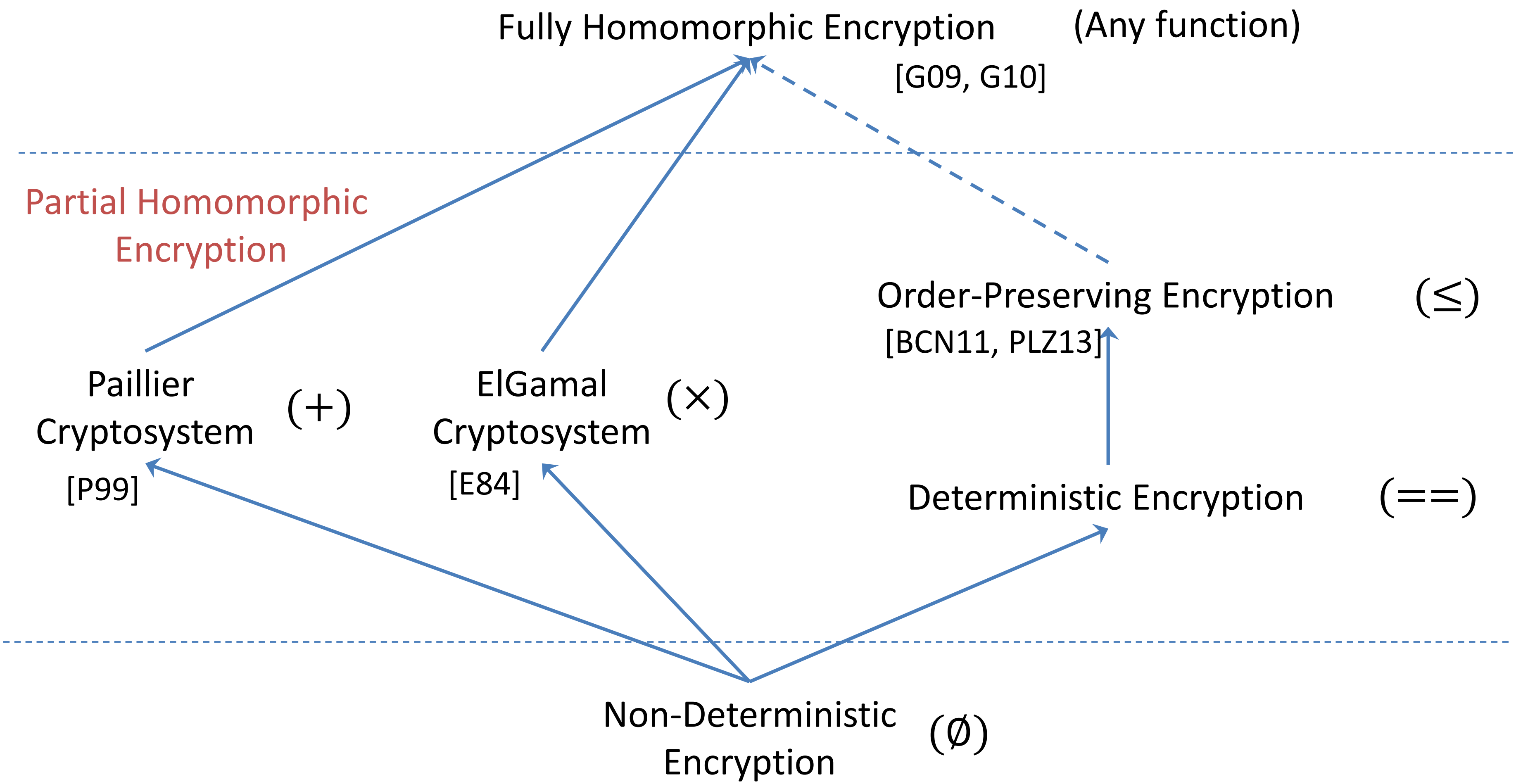
$$\sigma_{Score\_ope \geq 04e0\ldots}$$

StudentId	AssignId	Score_OPE
1	1	0x0065fda789ef4e272bcf102787a93903
1	2	0x009b5708e13665a7de14d3d824ca9f15
3	4	0x08db34fb1f807678d3f833c2194a759e
…	…	…

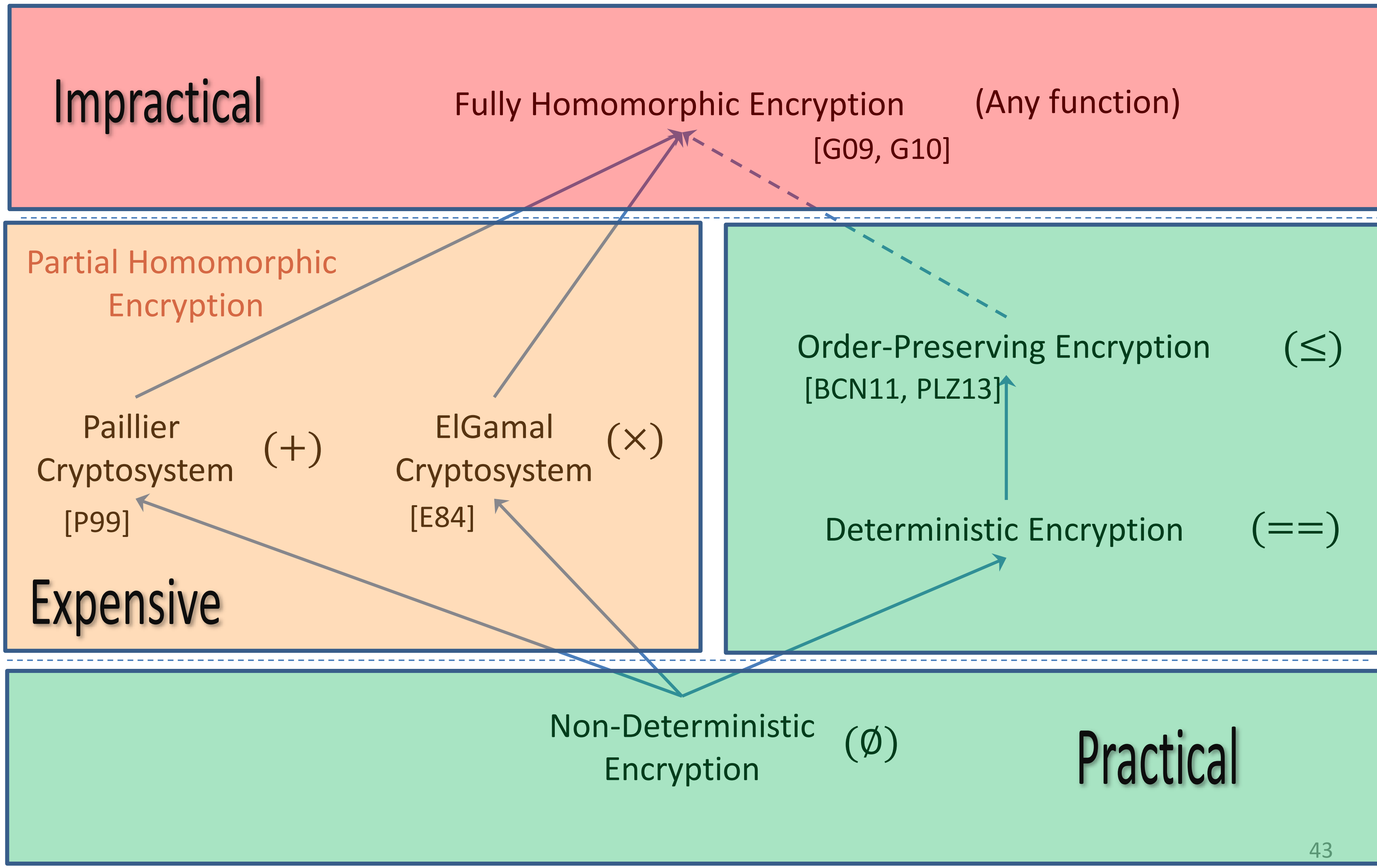
Homomorphic Encryption Schemes



Homomorphic Encryption Schemes



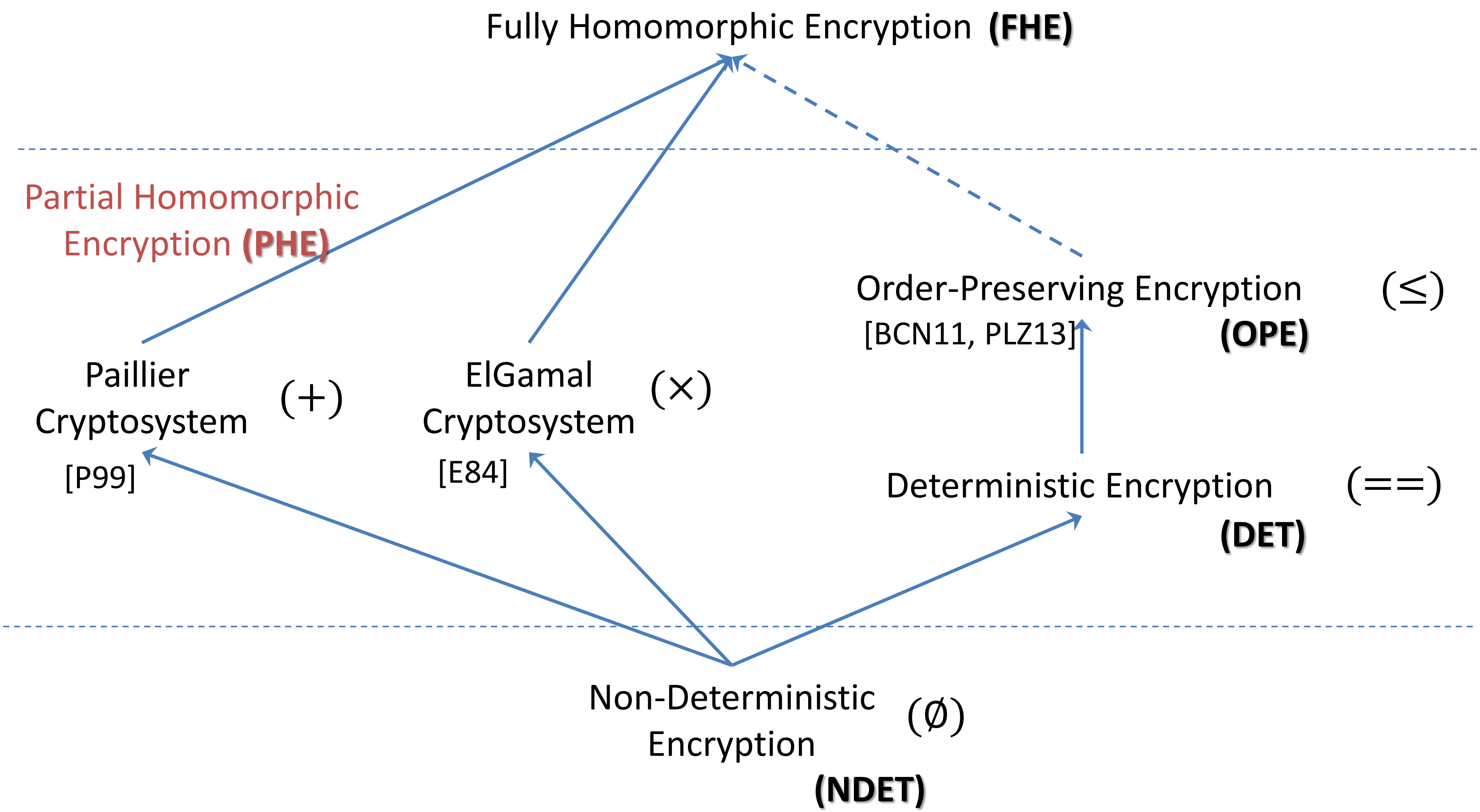
Homomorphic Encryption Schemes



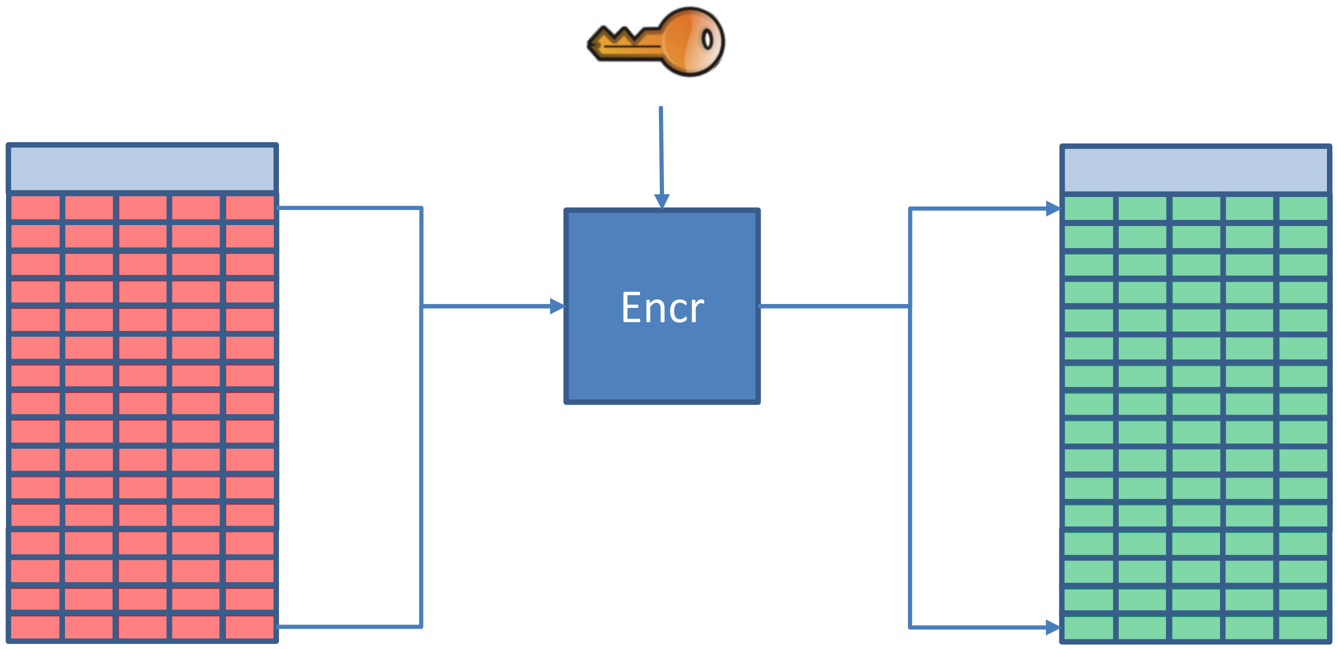
Homomorphic Encryption Schemes: Performance

Scheme	Space for 1 integer (bits)	Time for 1 operation
Fully Homomorphic Encryption	2^{14}	Cosmic time scales
Paillier ElGamal	2048	≈ ms
Deterministic Order-preserving	128	≈ µs

Homomorphic Encryption Schemes: Notation



How do I Encrypt a Database?

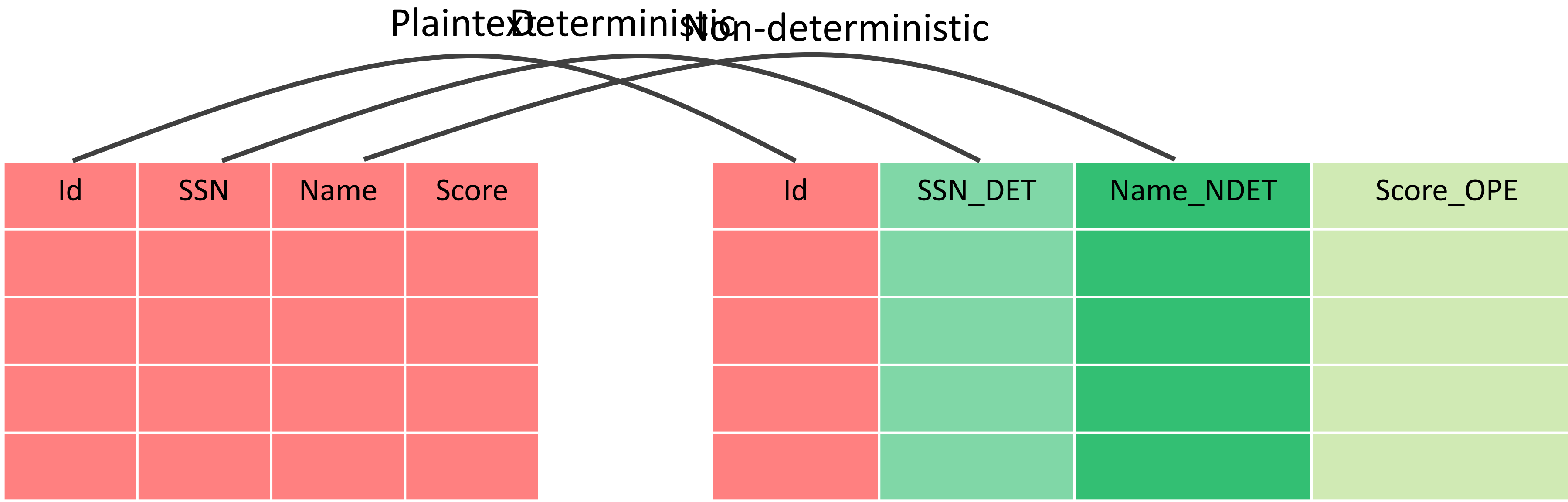


Cell granularity

Advantage:

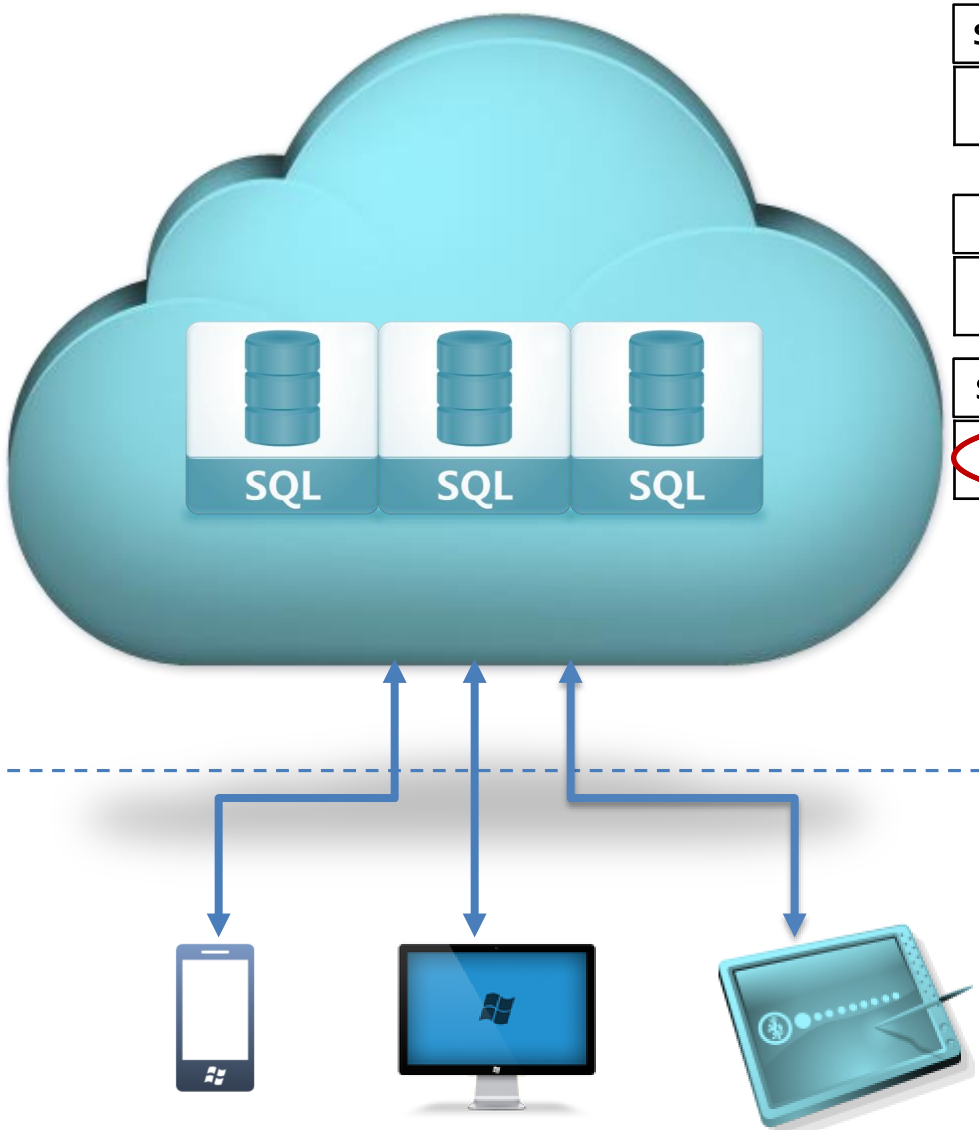
- Random access to a cell contents
- Mix n Match encryption

Mix n Match Encryption



Not covered: Deriving multiple keys. See [PRZ+11] for an example.

Example: Online Course Database



Student

StudentId	Name	Addr	GPA	CreditCard	...

Course

CourseId	Name	InstrId	...

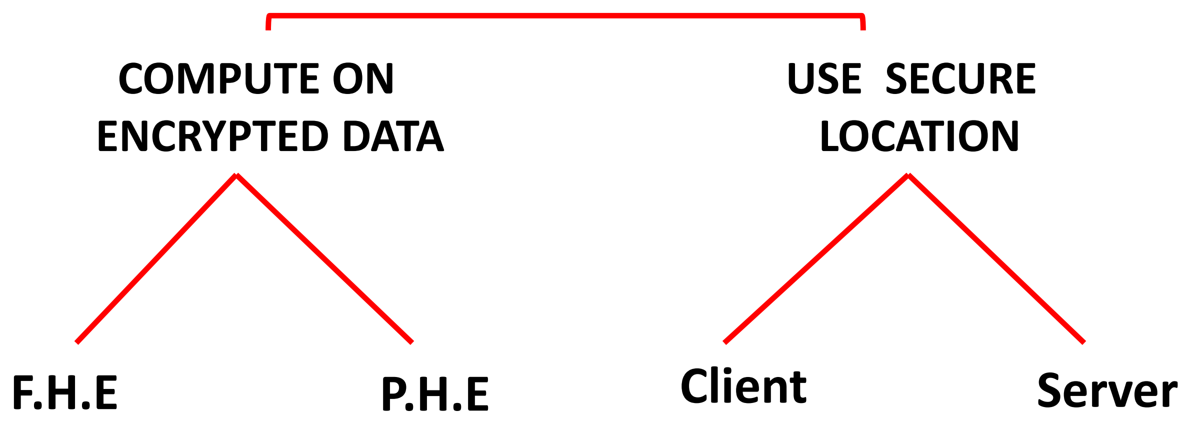
StudentCourse

CourseId	StudentId	Grade	...

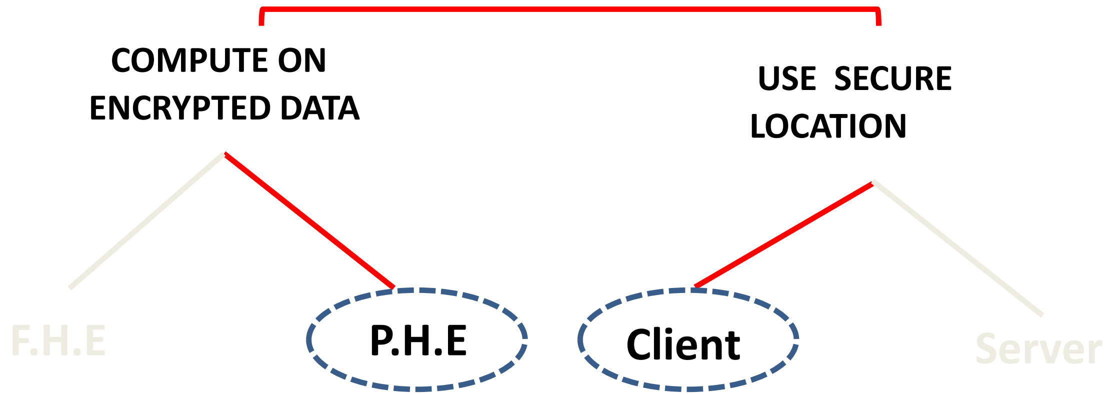
Roadmap

- Introduction
- Overview
- Basics of Encryption
- Trusted Client based Systems
- Secure In-Cloud Processing
- Security
- Conclusion

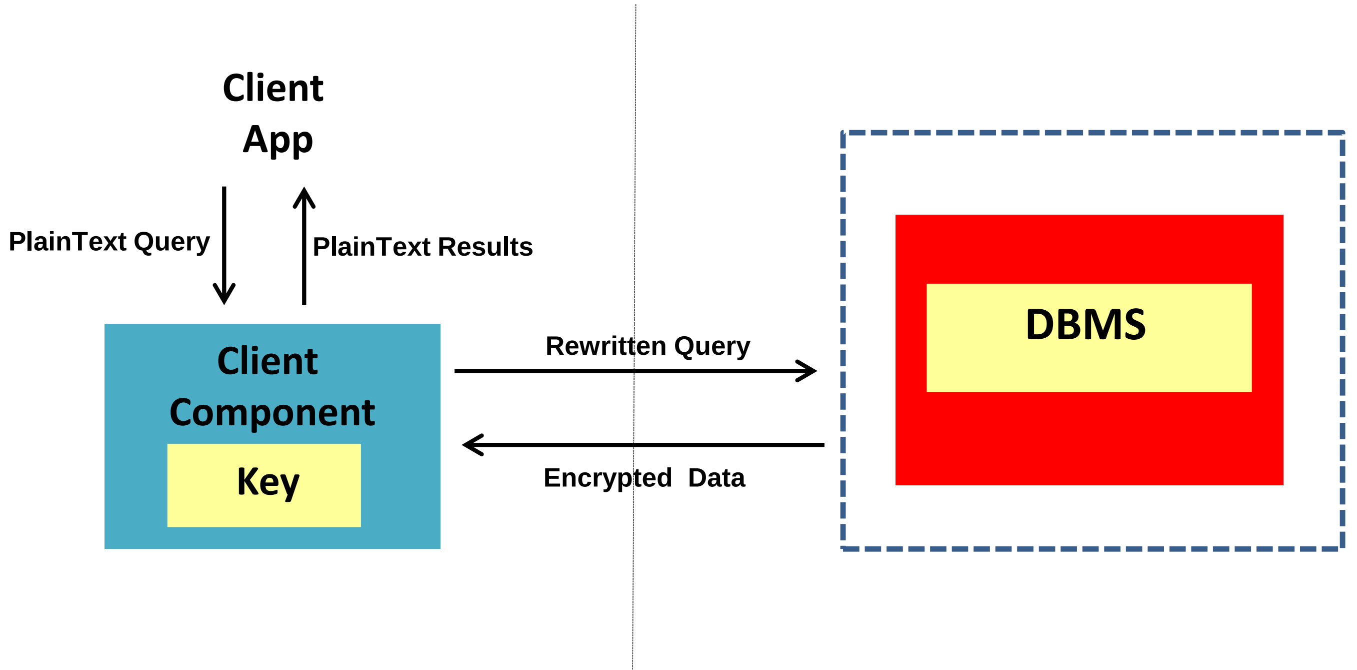
Design Choices



Trusted Client based Systems



Trusted Client Architecture

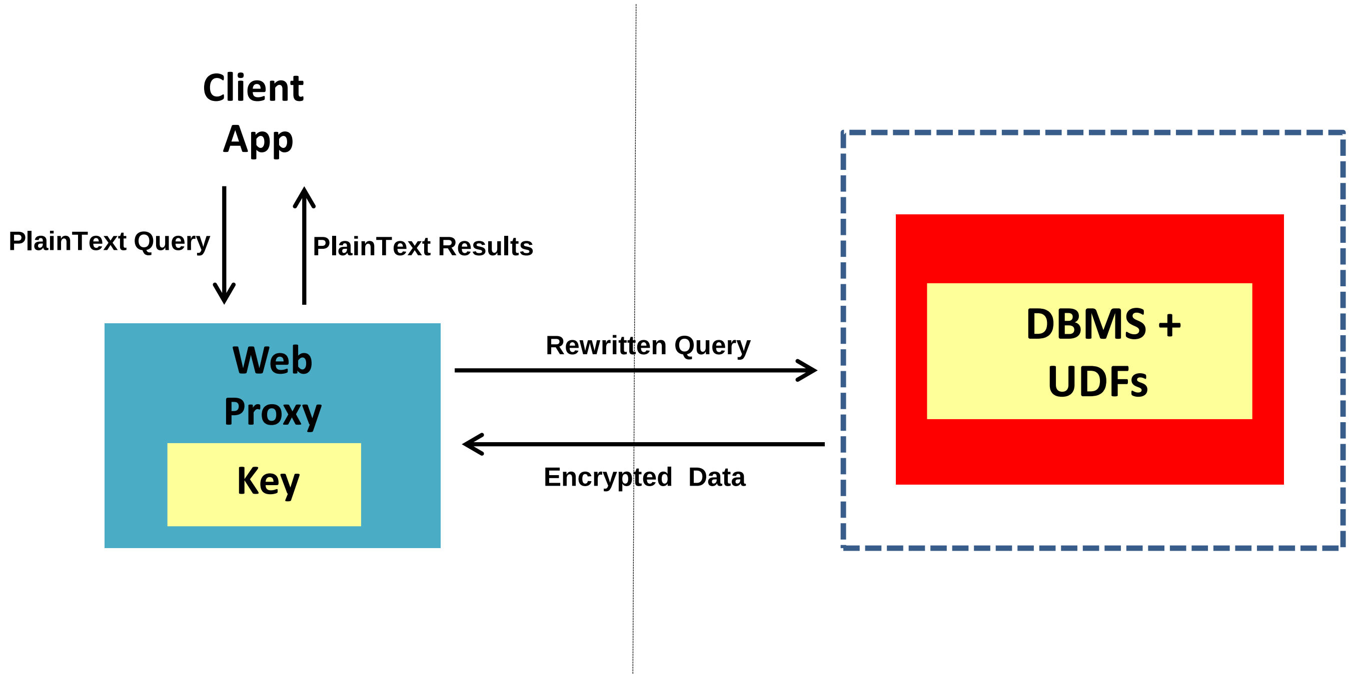


- Data not decrypted in DBMS
  - Only ciphertext seen in the DBMS
- No changes to DBMS/Client App

Systems

- Minimal Client Computation
  - Use P.H.E (Cryptdb)
- Residual Query Processing in Client
  - Blob Store
  - Use in conjunction with P.H.E (Monomi)

CryptDB Architecture



- Web proxy rewrites queries, decrypts result
- Leverage P.H.E techniques

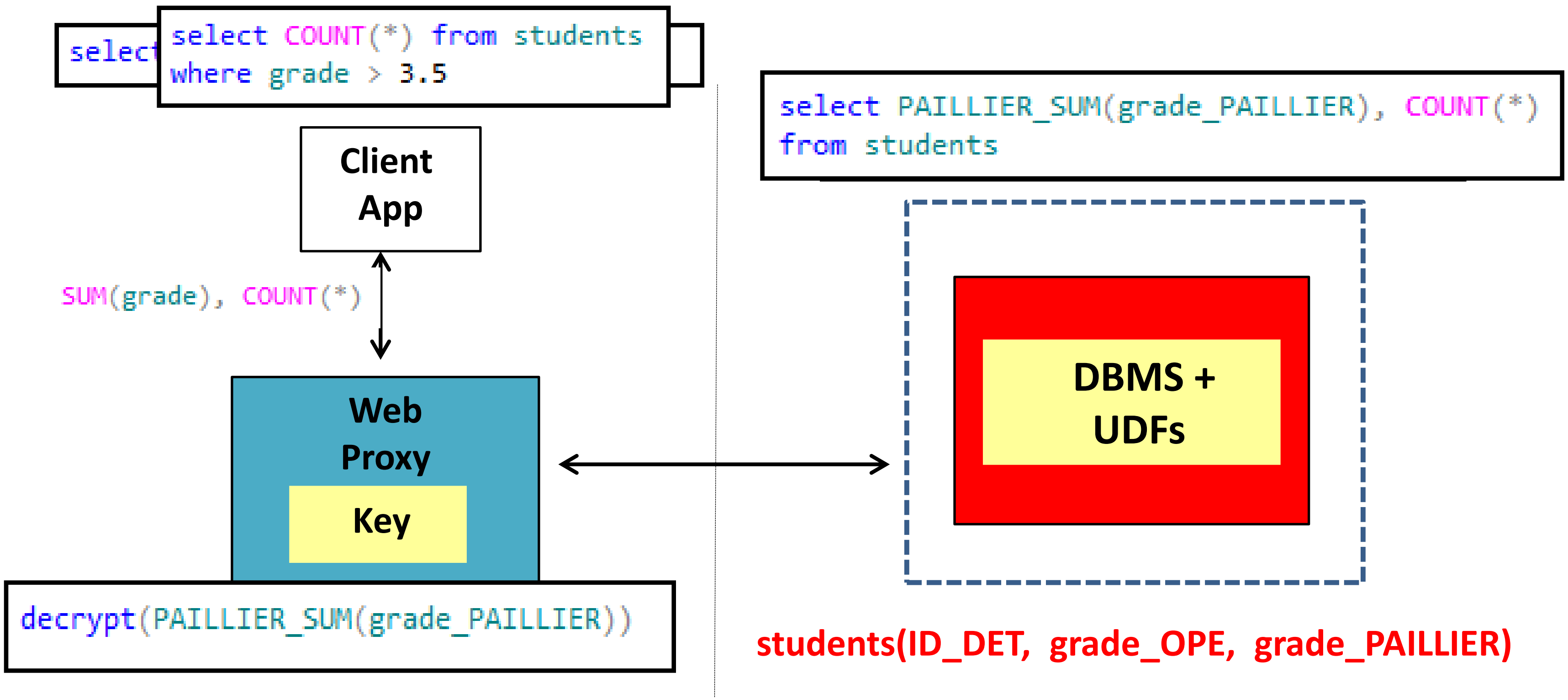
[PRZ+11]

Database Design

- students(ID, grade)
  - Point Lookups on ID column
  - SELECT and AGGREGATION queries on grade
- students(ID_DET, grade_OPE)
- students(ID_DET, grade_OPE, grade_PAILLIER)
  - Need to store columns encrypted in multiple ways
  - Static/Dynamic design based on workload

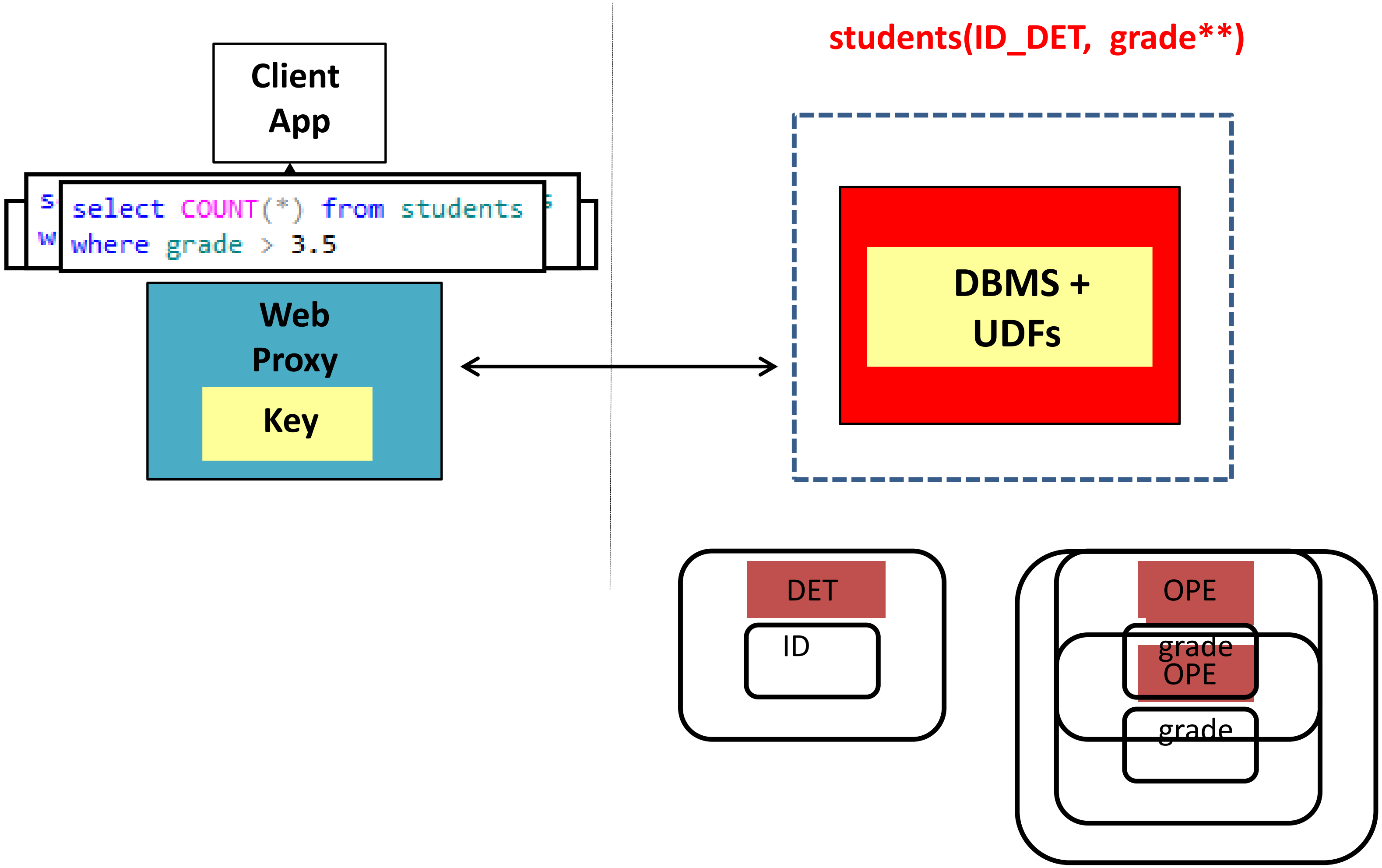
[PRZ+11]

Query Processing



[PRZ+11]

Dynamic Database Design

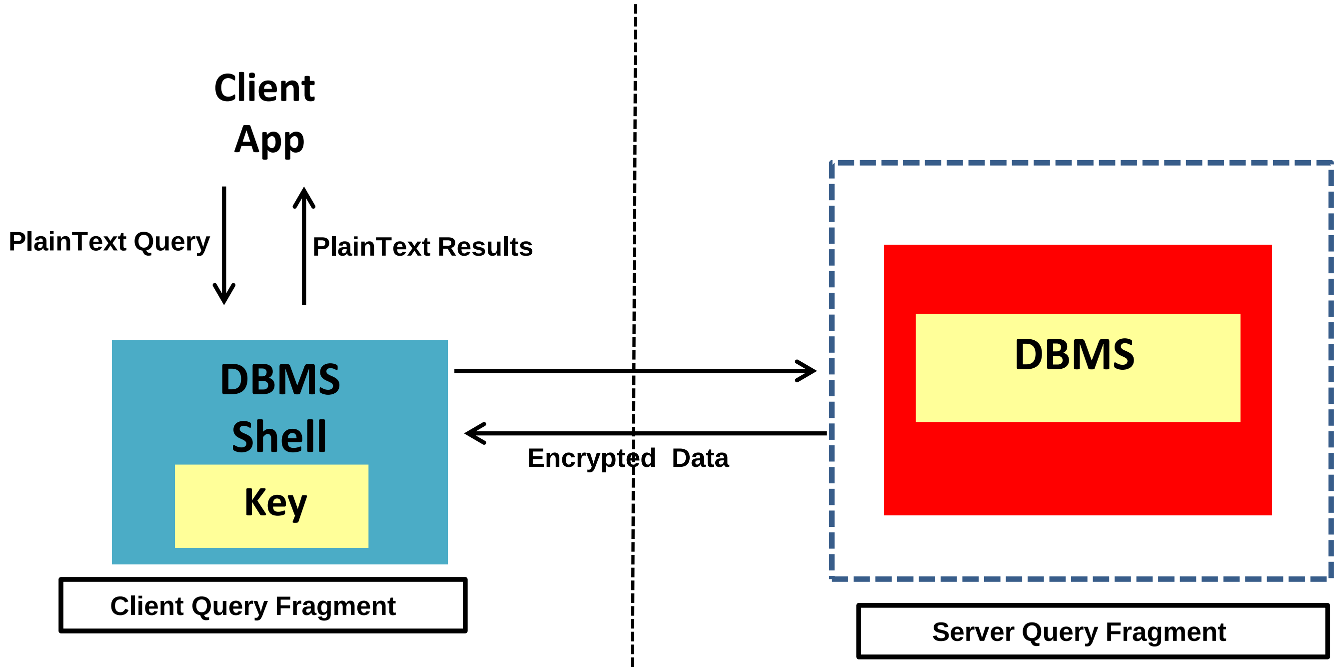


[PRZ+11] [FK13a] [FK13b]

Systems

- No Client Computation
  - Leverage P.H.E
  - e.g., Cryptdb
- Residual Query Processing on Client
  - e.g., Blob store
  - Use in conjunction with P.H.E (e.g., Monomi)

Computation in Trusted Client



- Distributed query processing between DBMS shell and untrusted DBMS

[HMI02] [HIL+02] [HMH08] [TFM13]

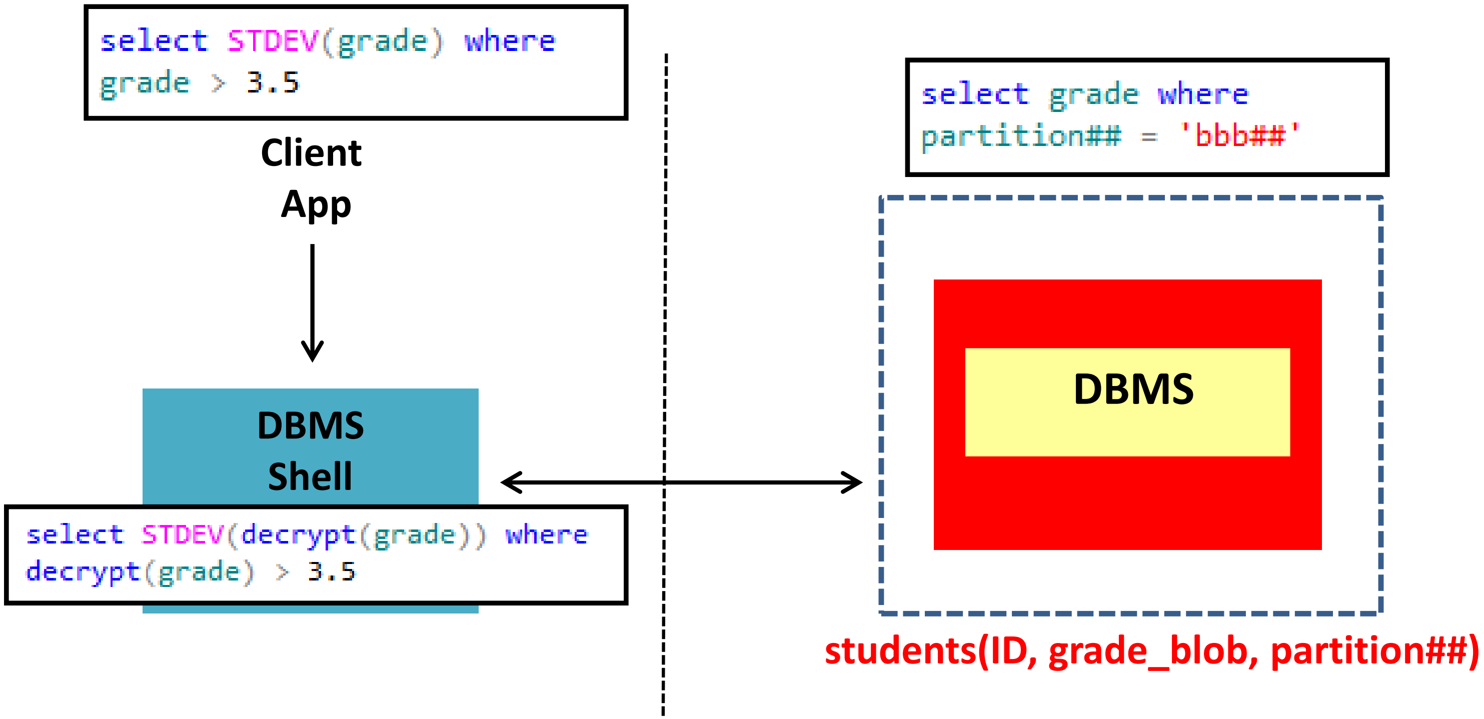
Blob Store: Database Design

- Encrypted data stored as 'blobs' (No computation)
  - students(ID, grade_blob)
- Use additional "fake" partitions to index blobs
  - students(ID, grade_blob, partition##)

grade	partition##
0 - 1.0	ccc##
1.0 – 2.0	aaa##
2.0 – 3.0	ddd##
3.0 – 4.0	bbb##

[HIL+02]

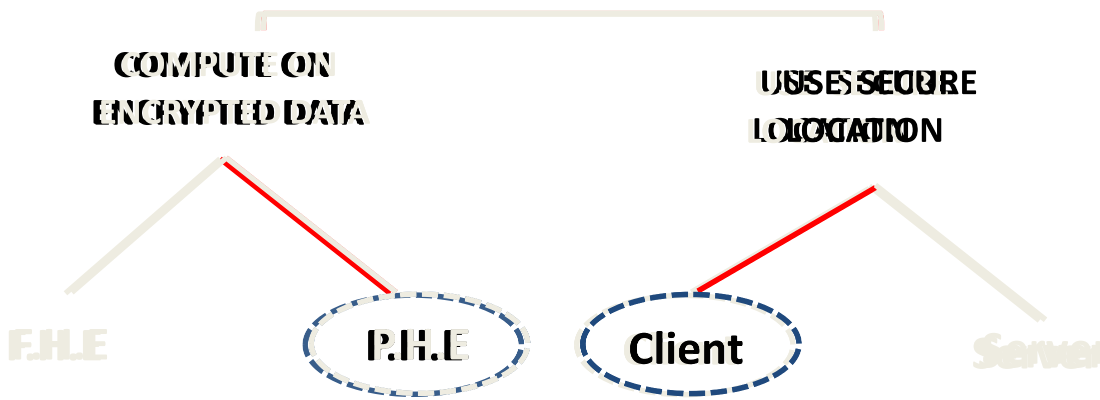
Blob Store: Query Processing



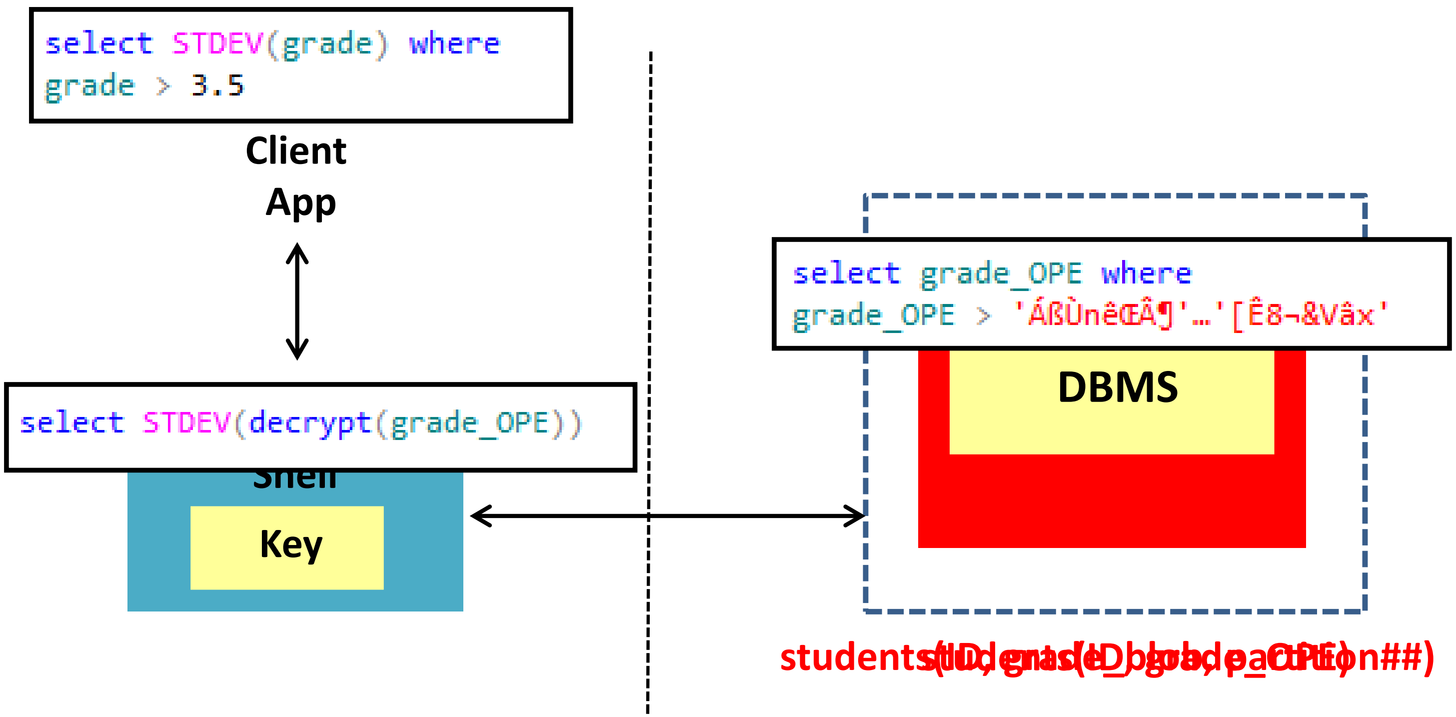
- Distributed query processing
  - Choosing appropriate partitioning
  - “Optimal” Query Splitting

[HIL+02] [HIM05] [HMT04]

Trusted Client based Systems



Augmenting Blob Store



- Use P.H.E to push more computation to DBMS
  - Monomi

[TFM13]

Pre-computation for complex queries

- Find student submissions that have been handed in a day late

```
select id, count(*) from students
where submissiondate = deadline + 1
group by id
```

- Students(ID_DET, submissiondate_DET, deadline_DET, deadline_PAILLIER)
  deadlineplusone_DET)
  - Cannot "Mix and Match" different encryptions

```
select id, count(*) from students
where submissiondate = deadline_plusone
group by id
```

[TFM13]

Trusted Client: Summary

- No Server changes required to DBMS
- Works well for workloads where amount of data shipped is small
  - Physical Design is important for distributed queries
  - Pre-computation is not free
- Generality of approach is unproven
  - Integrity constraints, Triggers etc.
  - Automated tools to migrate database applications

Limitation: P.H.E

- P.H.E is not “free” – space overheads
  - For Paillier, to store one integer (32 bits), the ciphertext need to use 2048 bits!
  - Compact representation for paillier/OPE that is updatable – open problem.

- P.H.E is inherently limited – cannot address all of SQL

```
select STDEV(grade) from students
```

[GZ07]

Limitation: Robustness

- Stored procedure to find student submissions that have been handed in late (with delay as a parameter)

```
create procedure ComputeLateSubmissions (@delay)
as
  select id, count(*) from students
  where submissiondate = deadline + @delay
  group by id
```

- Cannot pre-compute all possible input values
  - Why store the table in the cloud!

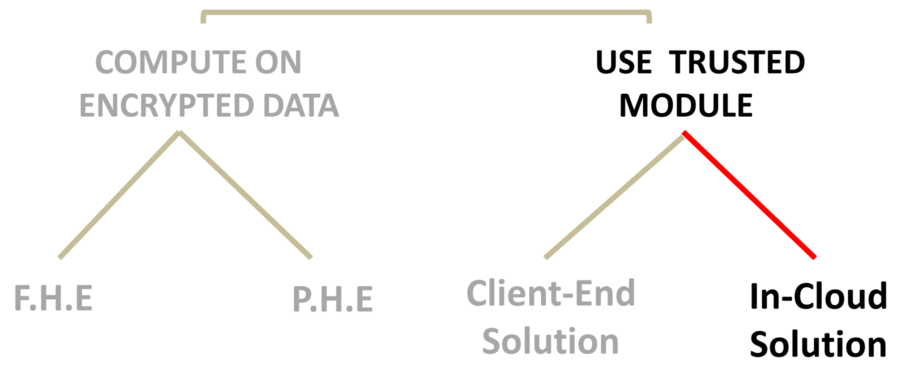
Still to come …

Is it possible to design an encrypted DBMS where only the results are shipped to the client irrespective of query complexity ?

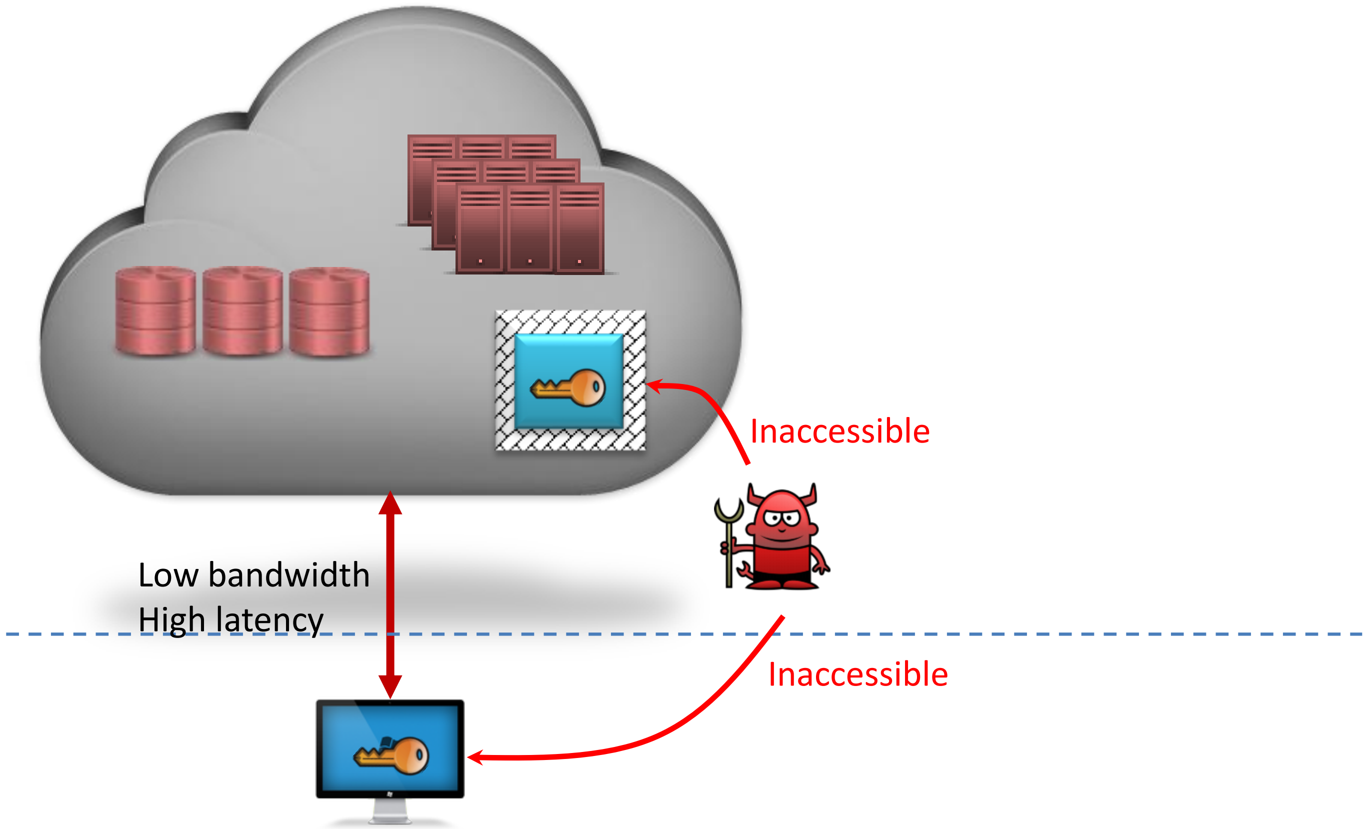
Roadmap

- Introduction
- Overview
- Basics of Encryption
- Trusted Client based Systems
- Secure In-Cloud Processing
- Security
- Conclusion

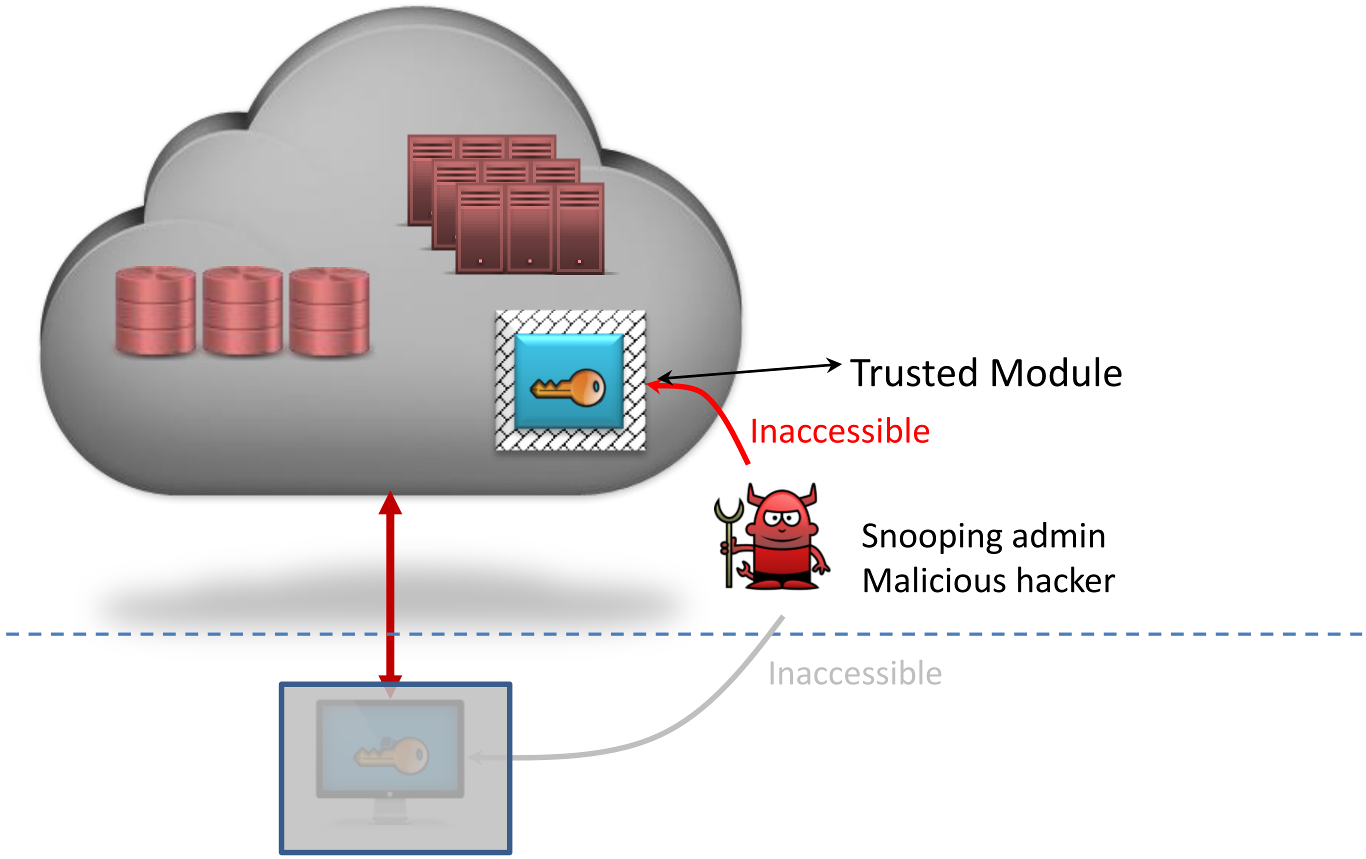
Secure In-Cloud Processing



Secure In-Cloud Processing



Secure In-Cloud Processing



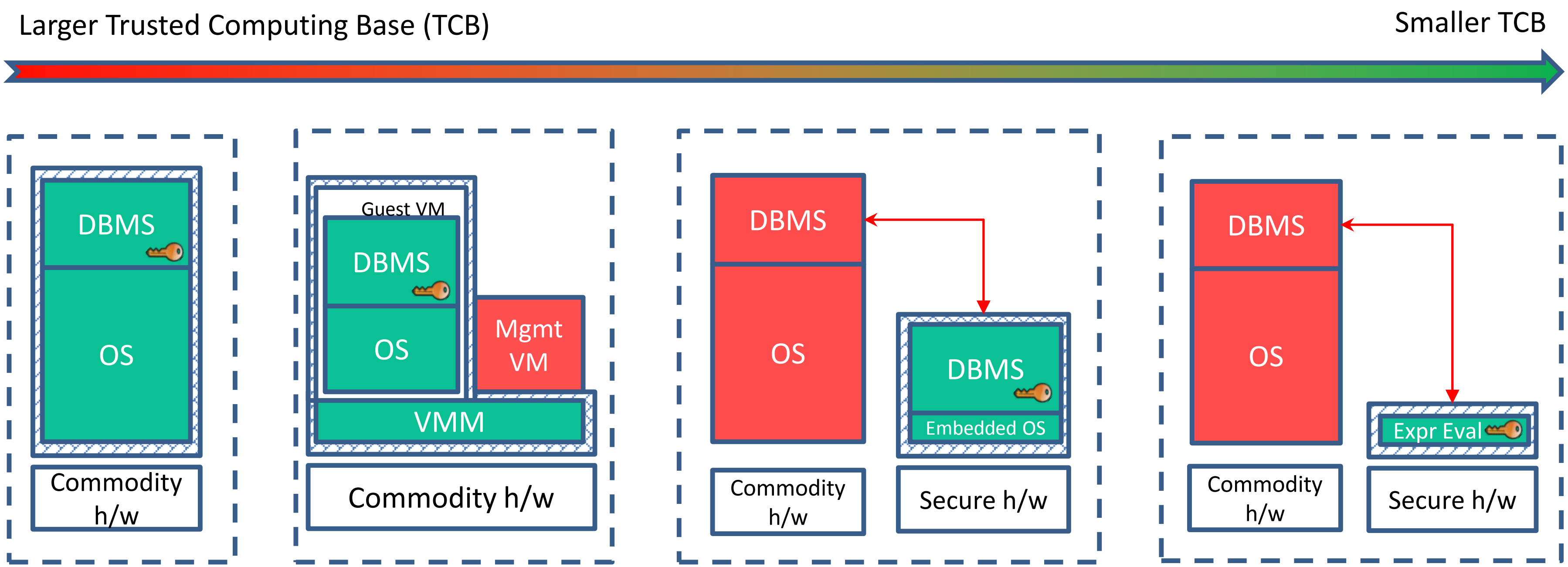
Trusted Module Design Space

Trusted Functionality (Trusted Computing Base)

Physical protection & h/w provided isolation

	OS+	VMM+	DBMS*	<<DBMS
Physical Server Protection				
Secure Co-Processor				
FPGA				
Intel SGX				

Trusted Functionality (Simplified)



OS+

[AWSGC]

VMM+

CloudVisor [ZCC+11]

Drawbridge [PBH+ 11]

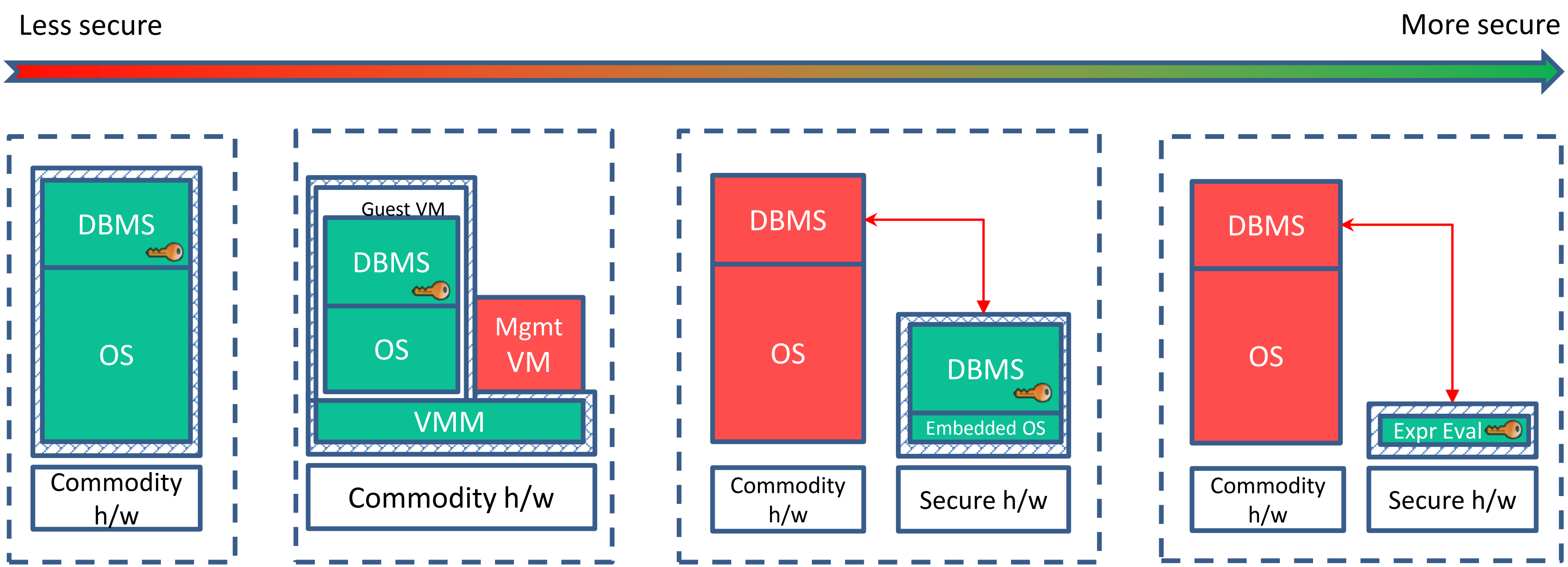
DBMS

TrustedDB [BS11]

<< DBMS

Cipherbase [ABE+12]

Trusted Functionality (Simplified)



OS+

[AWSGC]

VMM+

CloudVisor [ZCC+11]

Drawbridge [PBH+ 11]

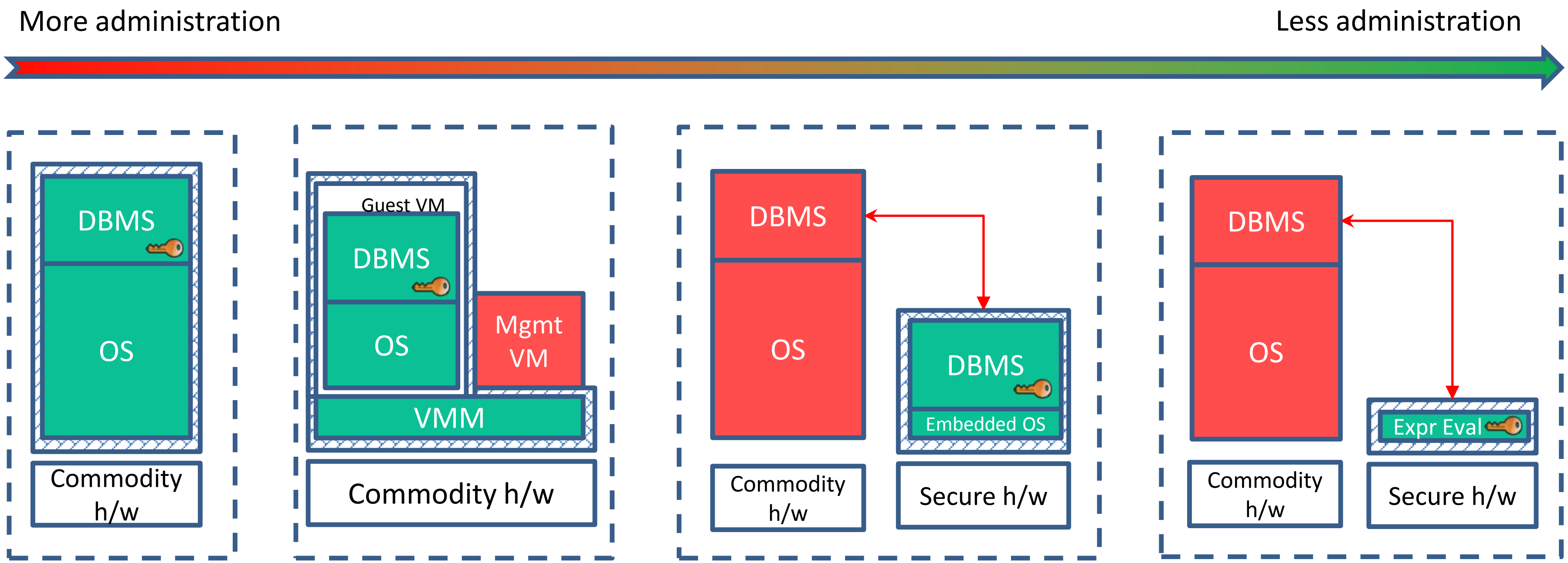
DBMS

TrustedDB [BS11]

<< DBMS

Cipherbase [ABE+12]

Trusted Functionality (Simplified)



OS+

[AWSGC]

VMM+

CloudVisor [ZCC+11]
Drawbridge [PBH+ 11]

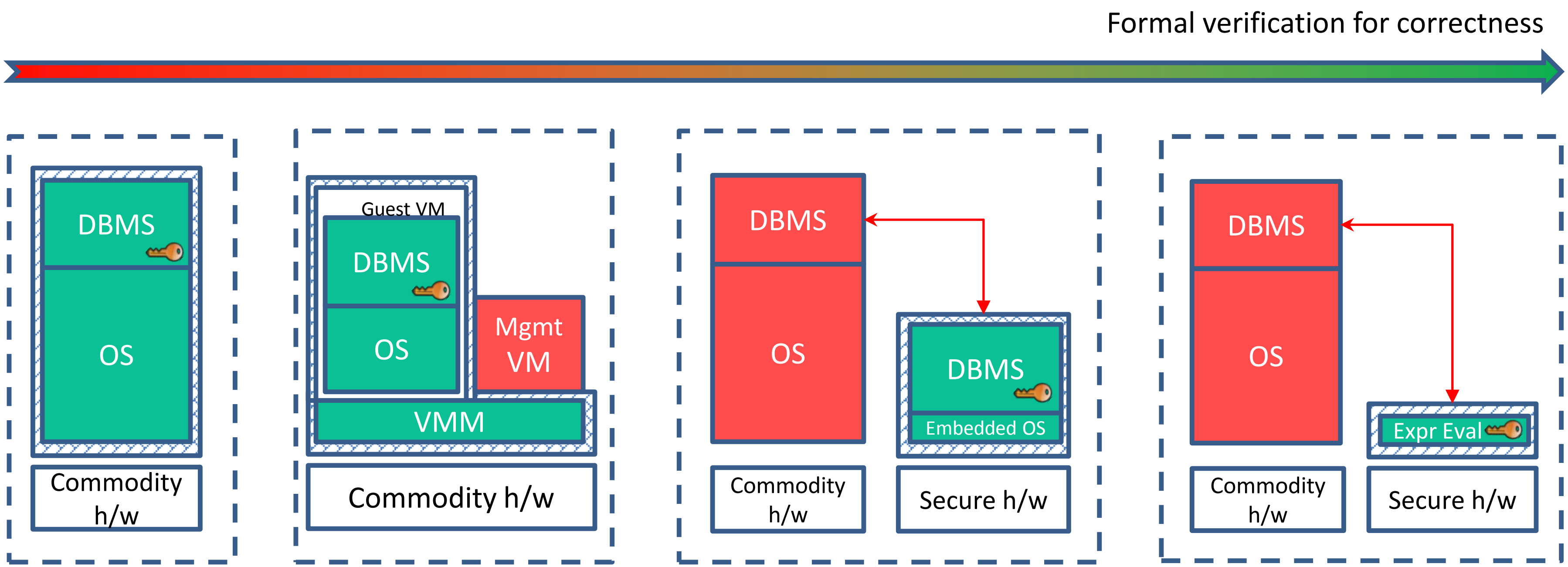
DBMS

TrustedDB [BS11]

<< DBMS

Cipherbase [ABE+12]

Trusted Functionality (Simplified)

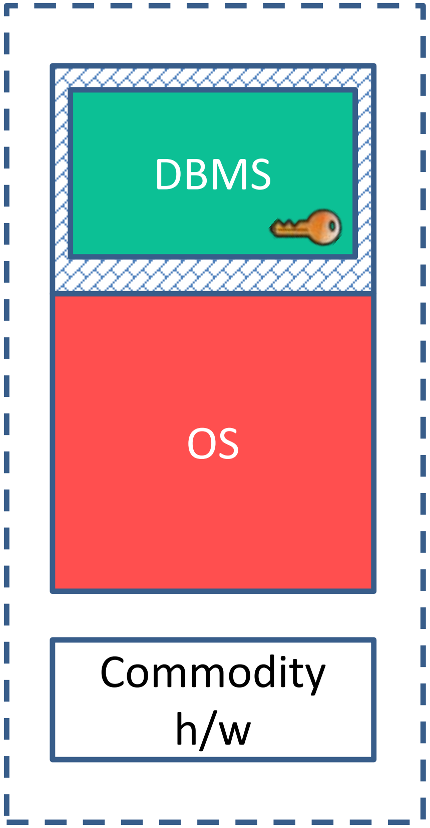


OS+	VMM+	DBMS	<< DBMS
[AWSGC]	CloudVisor [ZCC+11] Drawbridge [PBH+ 11]	TrustedDB [BS11]	Cipherbase [ABE+12]

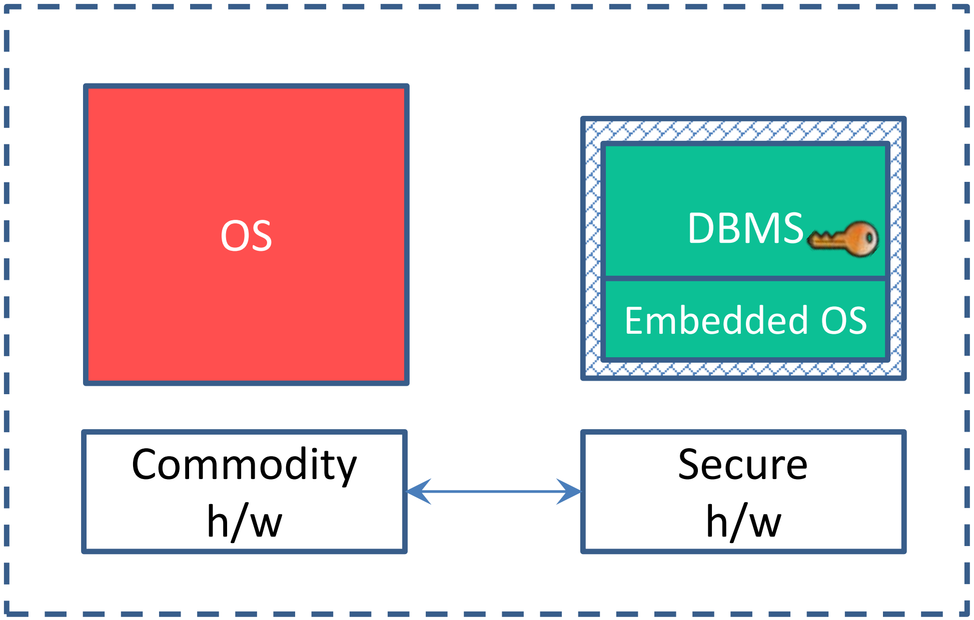
Formal Software Verification

- seL4 microkernel [KEH+ 09]:
  - 8700 lines of C code
  - 22 person year effort
- Verification tools:
  - Boogie [BCD+ 05]

Isolation



Isolation



Trusted Module Design Space

Trusted Functionality (Trusted Computing Base)

Physical protection & h/w provided isolation

	OS+	VMM+	DBMS*	<<DBMS
Physical Server Protection				
Secure Co-Processor				
FPGA				
Intel SGX				

Previous Use of Secure Hardware

- Secure Co-Processor
  - ATMs, smart cards
- Hardware Security Modules
  - Tamper-proof crypto acceleration [CloudHSM]
- FPGAs
  - Military use
- TPM chips
  - Laptops, etc.

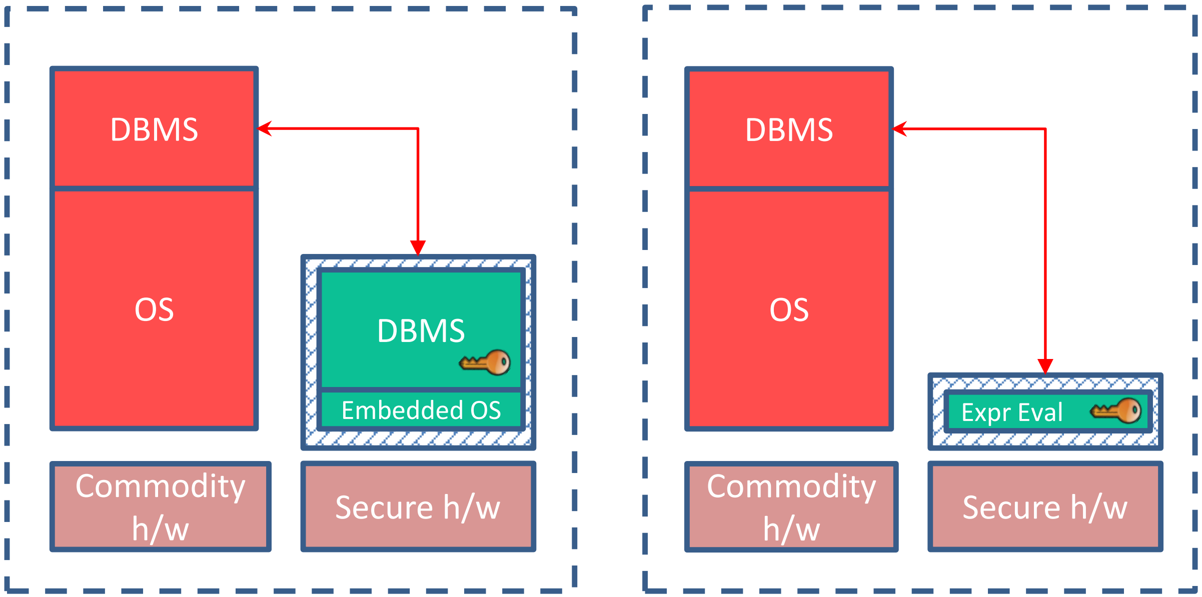
[TCGNotes]

IBM 4764 PCI-X Cryptographic Coprocessor

Secure FPGA

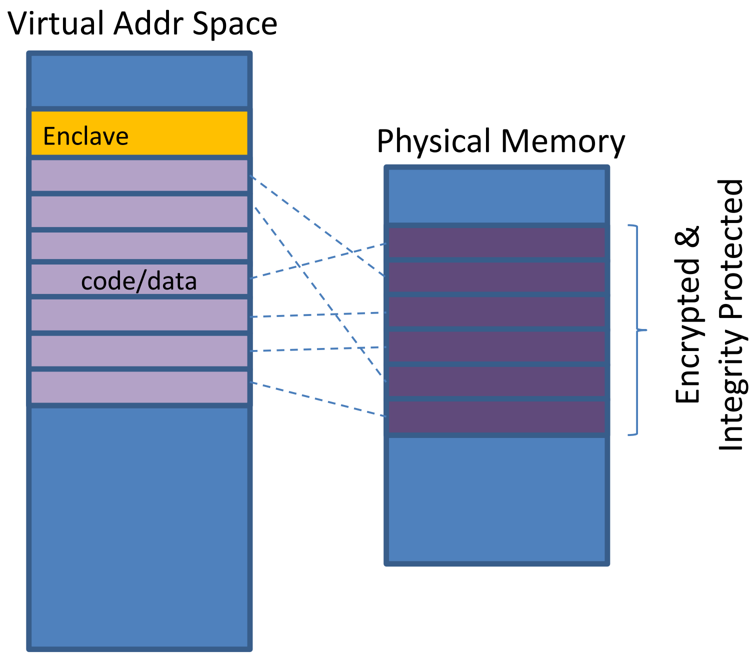
Trusted Hardware => Architectural Isolation

Separate memory space



Intel Software Guard Extensions

- Extensions to Intel Architecture
- Isolation to code + data within a designated region called *enclave*
  - Confidentiality
  - Integrity



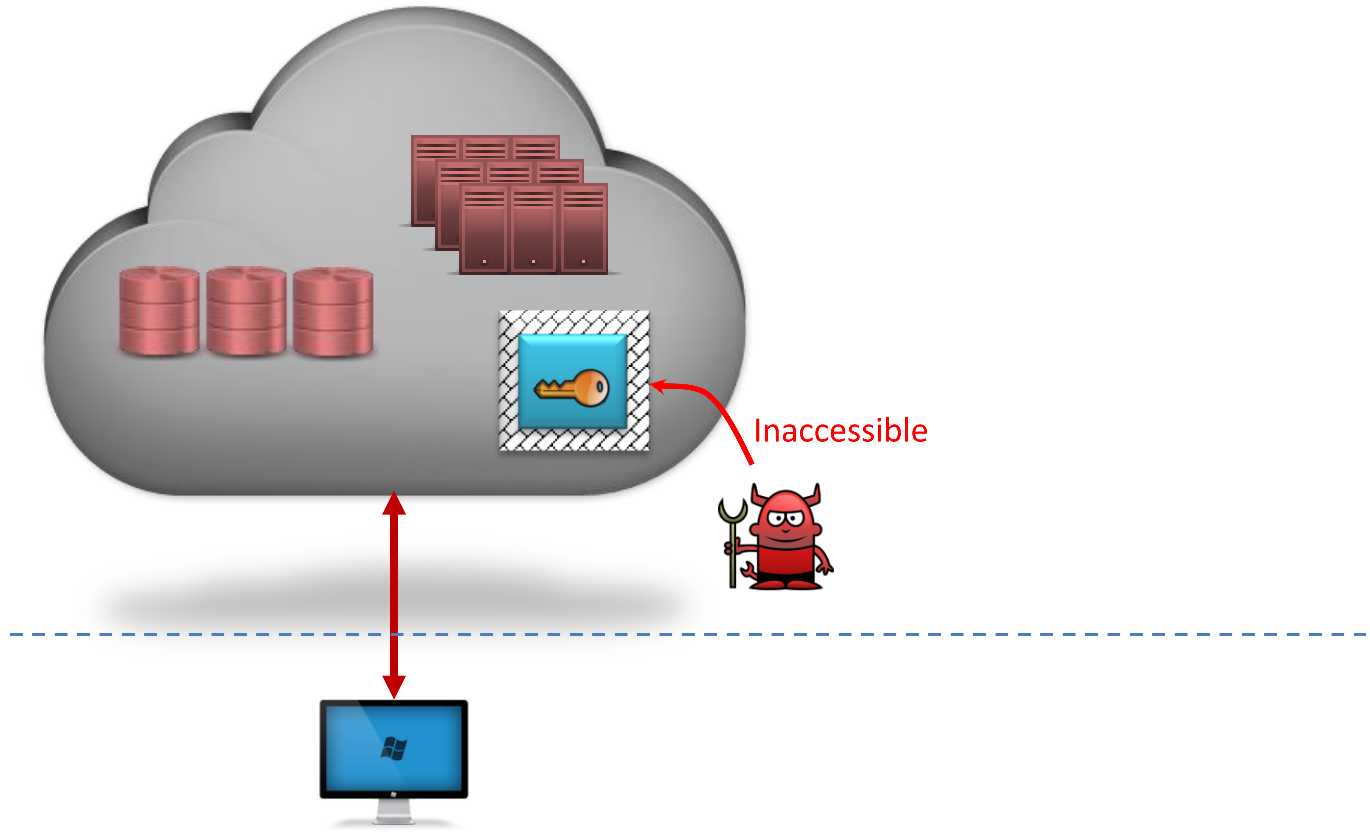
Ack: Andrew Baumann

[MAB+ 13, AGJ+ 13, HLP+ 13]

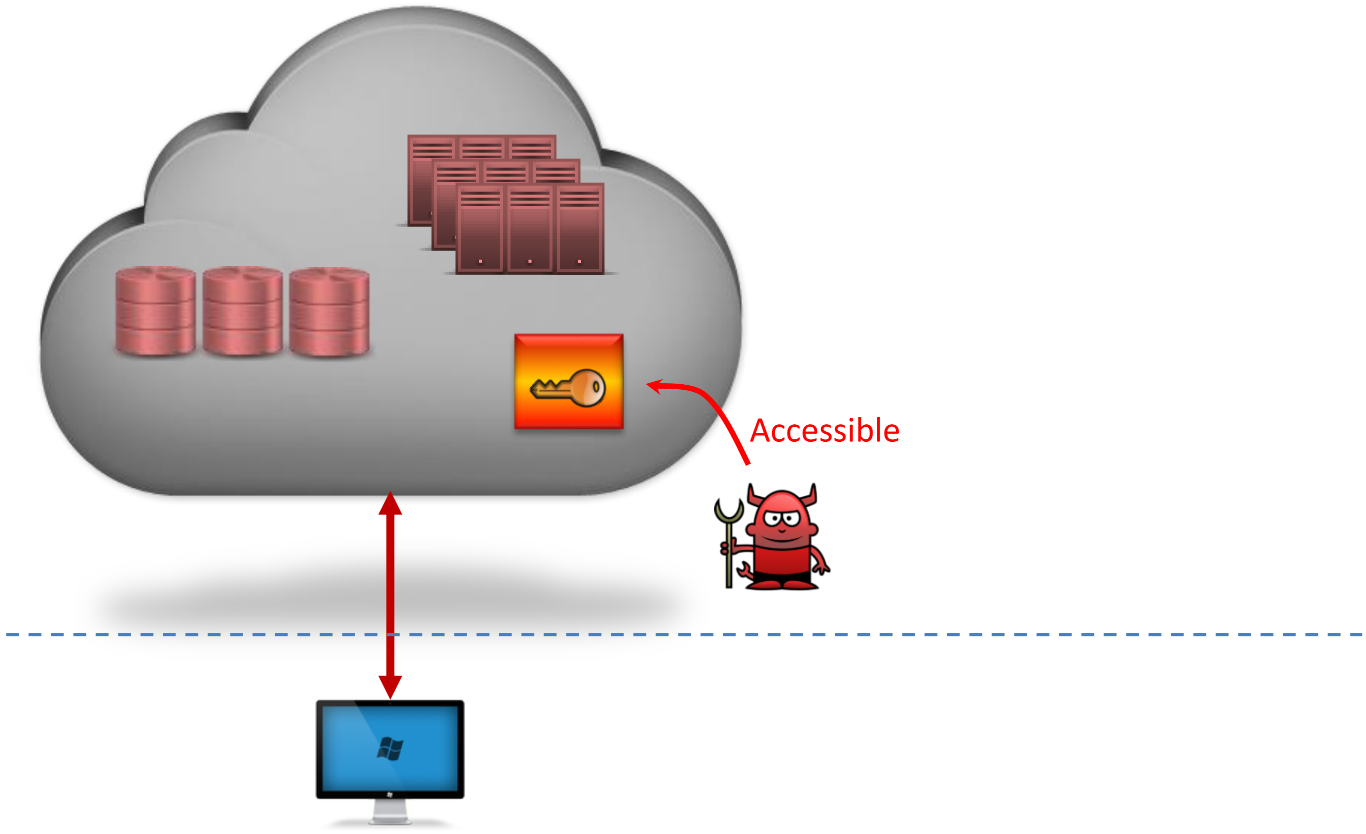
Performance Characteristics of Secure Hardware

Secure Hardware	Representative Unit	Performance Characteristics
Secure Co-Processor	IBM 4765	2 x 400 MHz CPU, 128 MB DRAM, 64 MB Flash
FPGA	Xilinx Virtex 6	≈ 150 MHz
Intel SGX	??	??

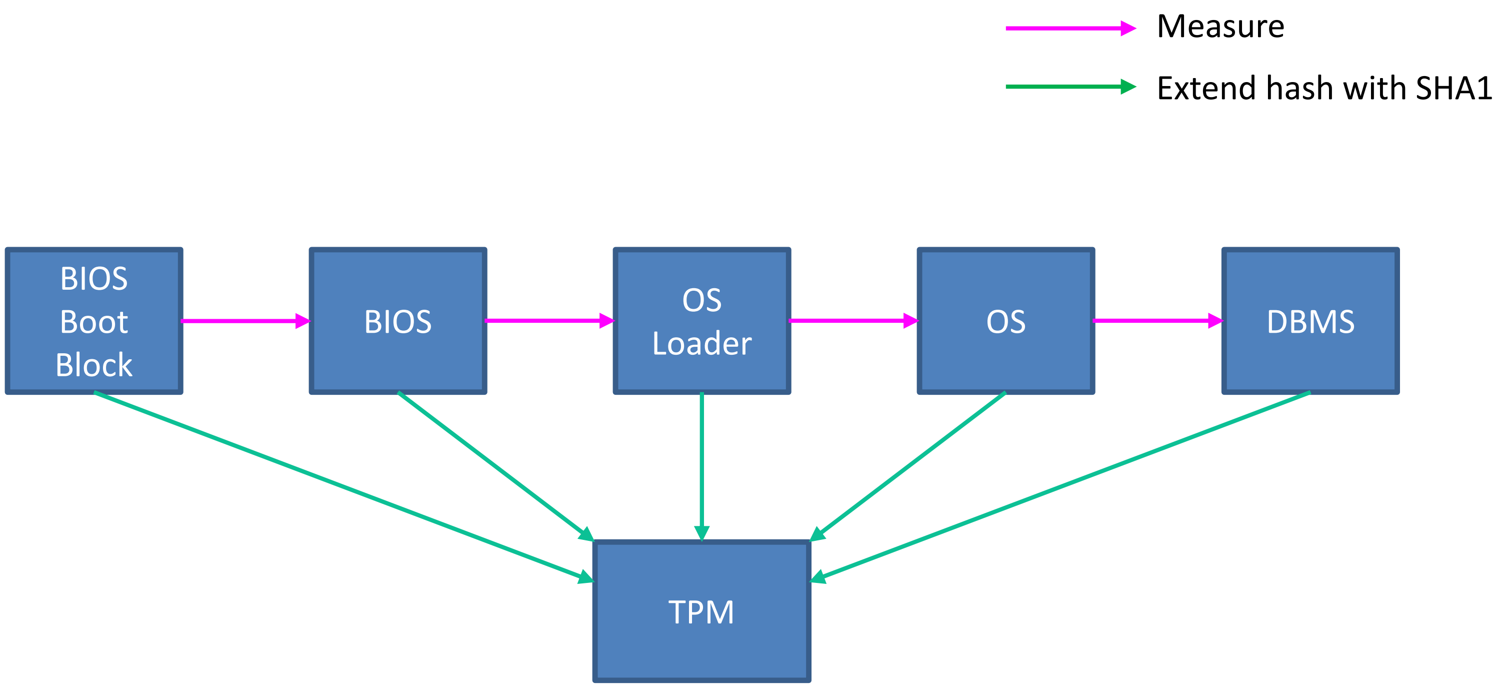
Verifying Identity in the Cloud



Verifying Identity in the Cloud



Verifying Identity of Remote Code



Source: http://crypto.stanford.edu/cs155old/cs155-spring11/lectures/08-TCG.pdf

Trusted Module Design Space

Trusted Functionality (Trusted Computing Base)

Physical protection & h/w provided isolation

	OS+	VMM+	DBMS*	<<DBMS
Physical Server Protection	Traditional DBMS	Traditional DBMS		
Secure Co-Processor				
FPGA				
Intel SGX				

Trusted Module Design Space

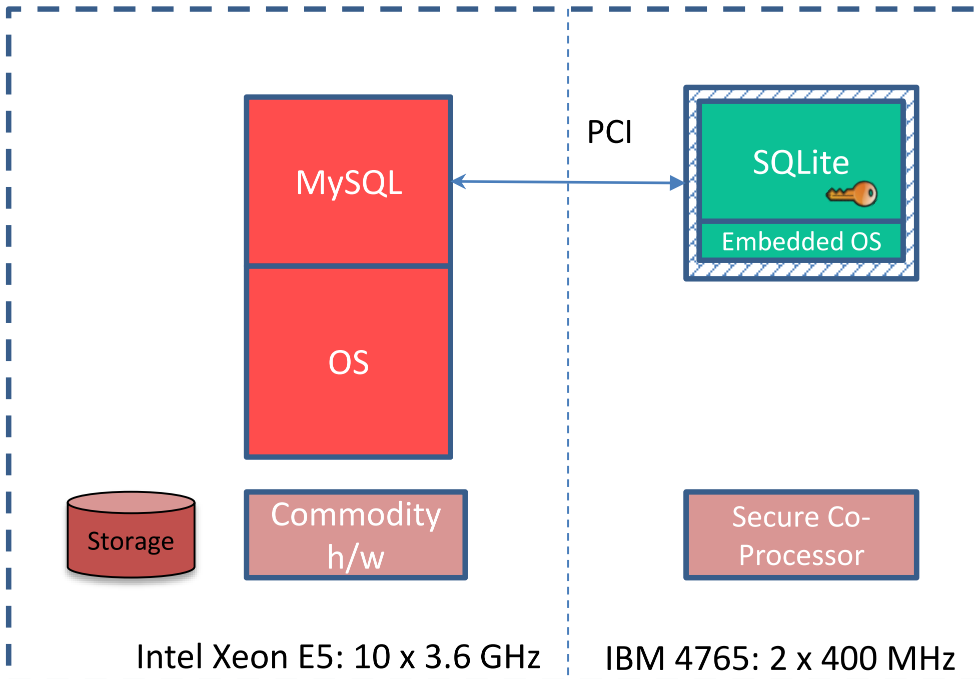
Trusted Functionality (Trusted Computing Base)

Physical protection & h/w provided isolation

	OS+	VMM+	DBMS*	<<DBMS
Physical Server Protection	Traditional DBMS	Traditional DBMS		
Secure Co-Processor			TrustedDB	
FPGA				Cipherbase
Intel SGX			?	?

TrustedDB [BS11]

- Distributed query processing
  - Untrusted: MySQL
  - Trusted: SQLite
- Persistent storage: untrusted system



60 GB RAM, etc

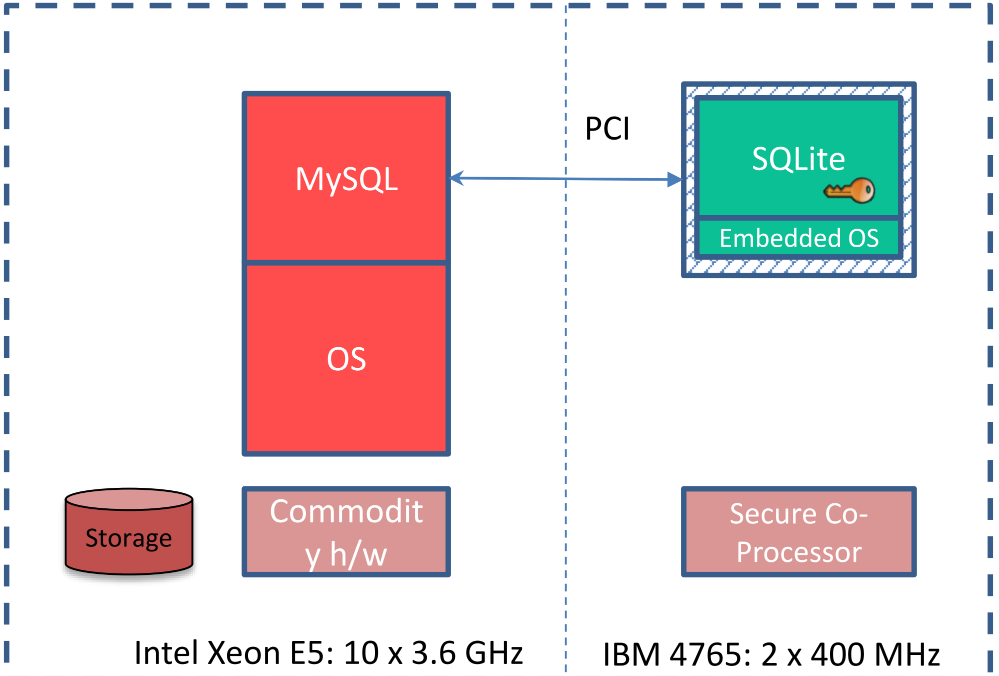
128 MB DRAM, 64 MB Flash

Source: http://meseec.ce.rit.edu/551-projects/fall2013/4-2.pdf

TrustedDB [BS11]

Sensitive columns

```
SELECT SUM(l_extendedprice * l_discount) as revenue
FROM  lineitem
WHERE l_shipdate >= '1993-01-01'       AND
      l_shipdate < '1994-01-01'        AND
      l_discount between 0.05 and 0.07 AND
      l_quantity < 24
```



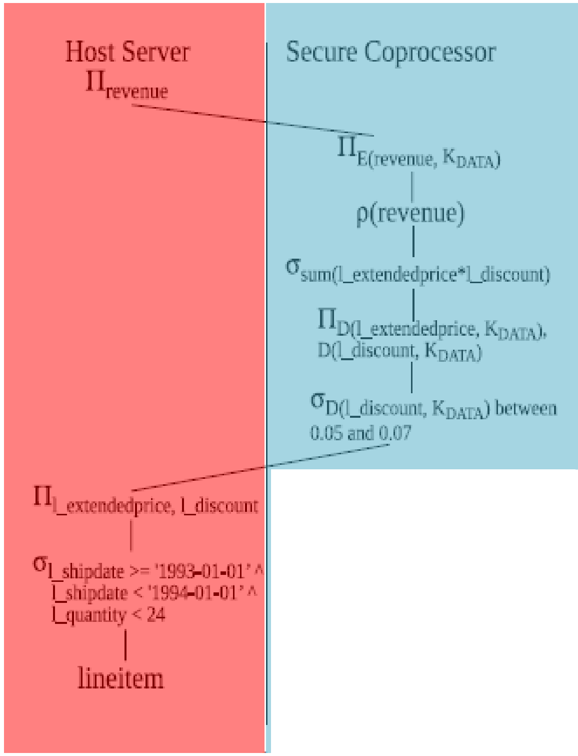
60 GB RAM, etc

128 MB DRAM, 64 MB Flash

TrustedDB [BS11]

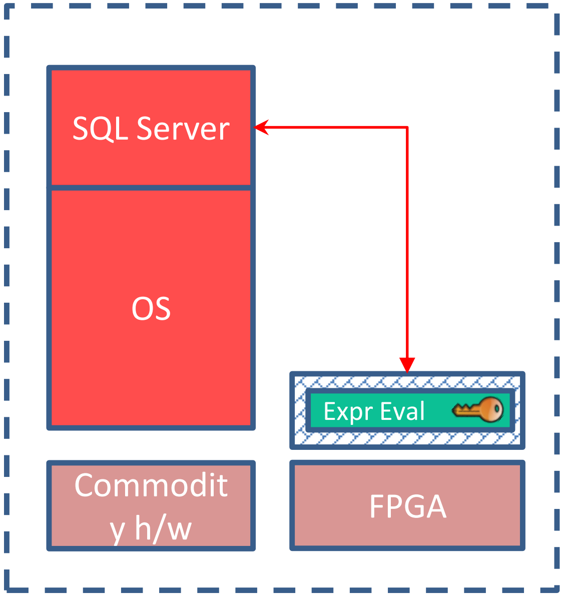
Sensitive columns

```
SELECT SUM(l_extendedprice * l_discount) as revenue
FROM  lineitem
WHERE l_shipdate >= ‘1993-01-01’       AND
      l_shipdate < ‘1994-01-01’        AND
      l_discount between 0.05 and 0.07 AND
      l_quantity < 24
```



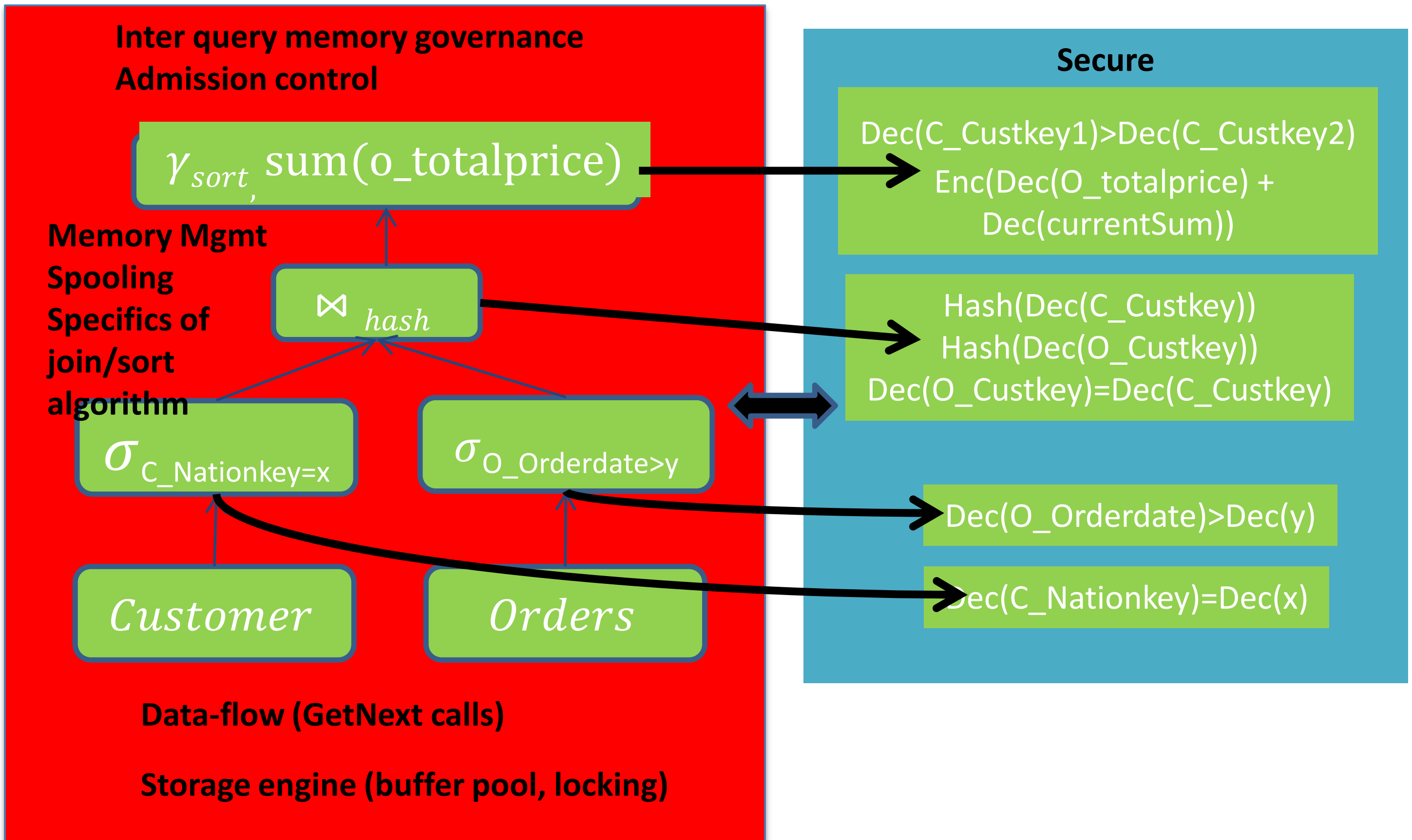
Cipherbase [ABE+ 13]

- Expression evaluation runs on secure hardware
  - Stack machine
  - Called as a "sub-routine" during query processing
- Everything else: commodity hardware
  - Modified SQL server
- Secure hardware:
  - Currently FPGA
- Design mostly agnostic to choice of secure hardware



Disclaimer: work done by tutorial presenters

Dedicated Expression Evaluation



[ABE+12, ABE+13]

Loosely-Coupled (TrustedDB) vs. Tightly-Coupled (Cipherbase)

Feature	LC	TC	Comments
Performance	👎	👍	TC: smaller footprint on secure hardware
Security	=	=	Comparable. Both leak access pattern but keep data encrypted.
Small TCB	👎	👍	By design
Functionality	👎	👍	LC: cannot run heavy weight DBMS on secure hardware
Software Engg	👍	👎	TC: fine-grained changes to DBMS
Choice of secure h/w	👎	👍	TC: smaller footprint can work on different secure hardware

LC: Loosely coupled (TrustedDB)   TC: Tightly coupled (Cipherbase)

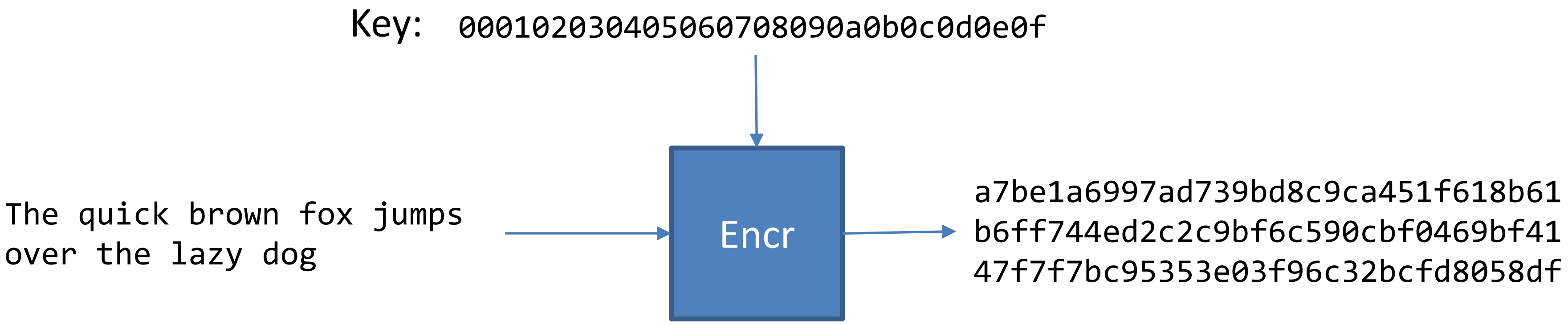
Summary

- Secure in-cloud trusted compute resources
- Relatively unstudied
- Secure Hardware landscape changing
- Lots of open issues
  - Architecture
  - Security
  - Details
    - Query Optimization

Roadmap

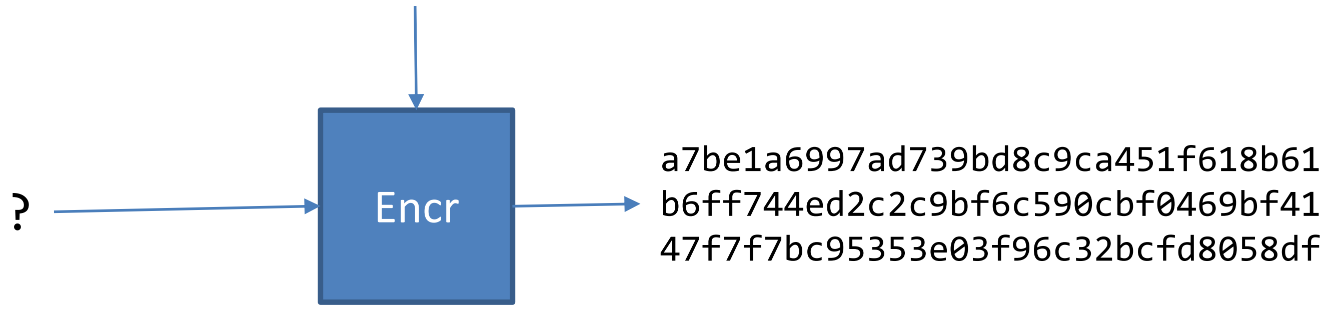
- Introduction
- Overview
- Basics of Encryption
- Trusted Client based Systems
- Secure In-Cloud Processing
- Security
- Conclusion

Encryption and Security

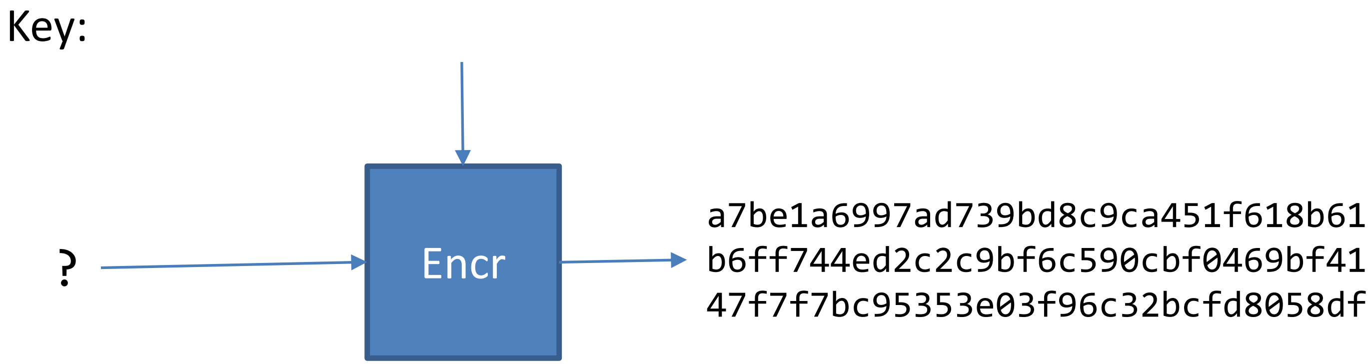


Encryption and Security

Key:



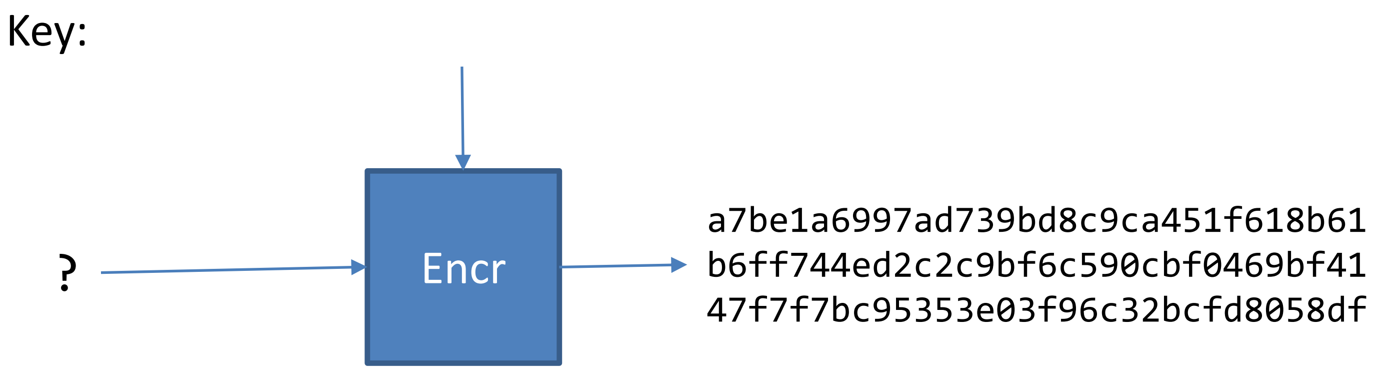
Encryption and Security



- **Semantic security**:
  - No information leakage except input length
- Winner of last year's Turing Award

[KL07]

Encryption and Security



- Encryption schemes such as AES in CBC mode (non-deterministic) are believed to be semantically secure

Security of Database Encryption

Disease
Flu
Diabetes
Flu
Cold

→

Disease_NDET
!x8J
)zFr#x
^@tG
BxU3

- Apply AES-CBC to every cell
- Leaks cell lengths

Deterministic Encryption

Disease
Flu
Diabetes
Flu
Cold

Disease_DET
!x8J
)zFr#x
!x8J
BxU3

- No. of distinct values + frequency distribution

  [BFO+08]

Order-Preserving Encryption

[AKS+04, BCN11, PLZ13, SBMKV13]

Age
12
51
24
36

→

Age_OPE
0x000a
0x0f12
0x00a1
0x00b2

- Ideal:
  - Security: Leak only order of cell values
  - Immutable: existing ciphertext does not change when new plaintext is inserted

Actual Proposals

Scheme	Guarantees	Leakage Besides Order
Agrawal et al. [AKS+04]	None	Yes
Boldyreva et al. [BCL09]	Yes	Half of plaintext bits
Popa et al. [PLZ13]	Yes	No, but a *mutable* scheme
Cipherbase [ABE+13]	Yes	No, and *immutable*

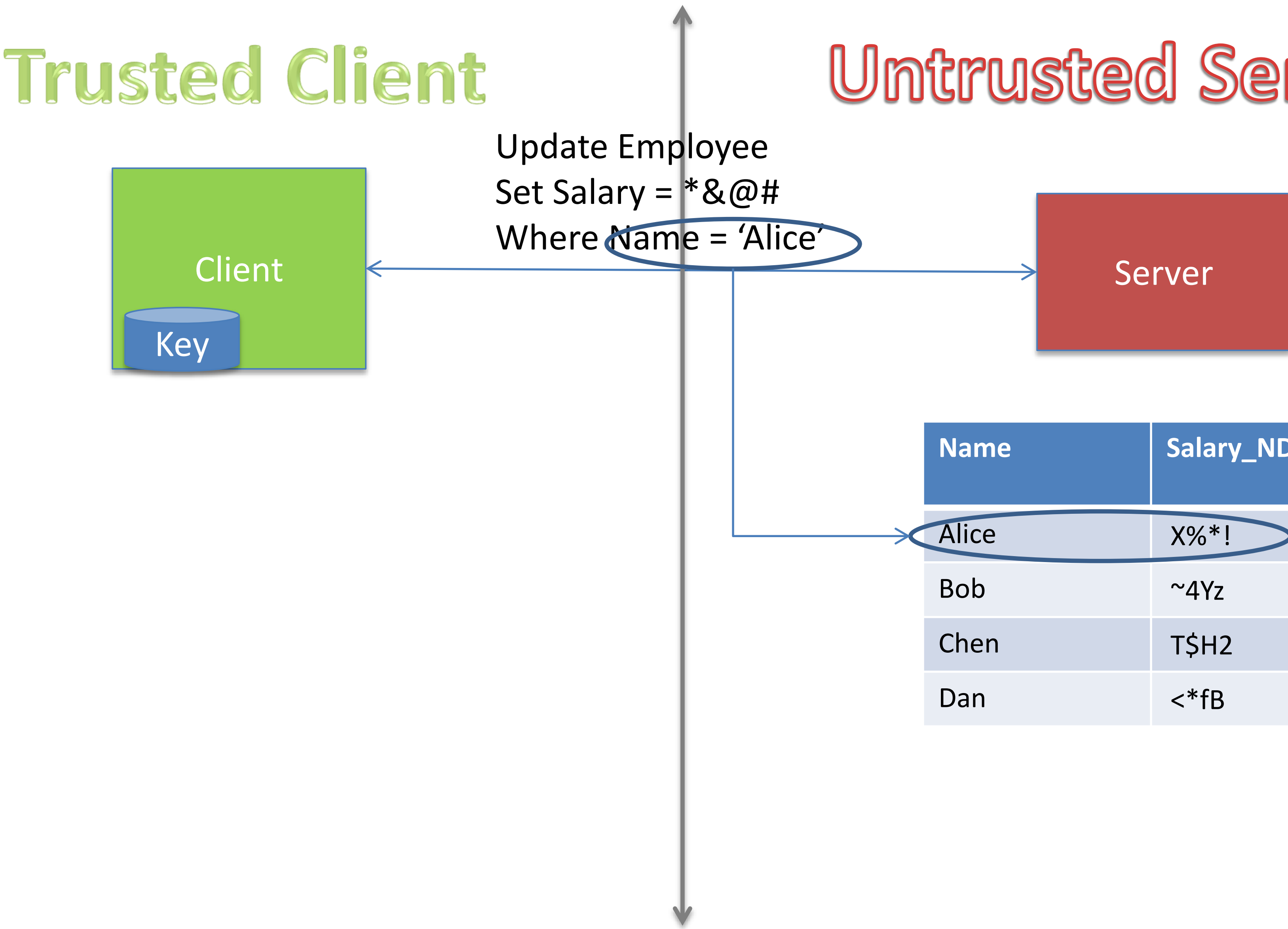
Overall Security of Data Encryption

Name	Age	Disease
Alice	12	Flu
Bob	51	Diabetes
Chen	24	Flu
Dan	36	Cold

Name_NDET	Age	Disease_DET
X%*!	12	!x8J
~4Yz	51	)zFr#x
T$H2	24	!x8J
<*fB	36	BxU3

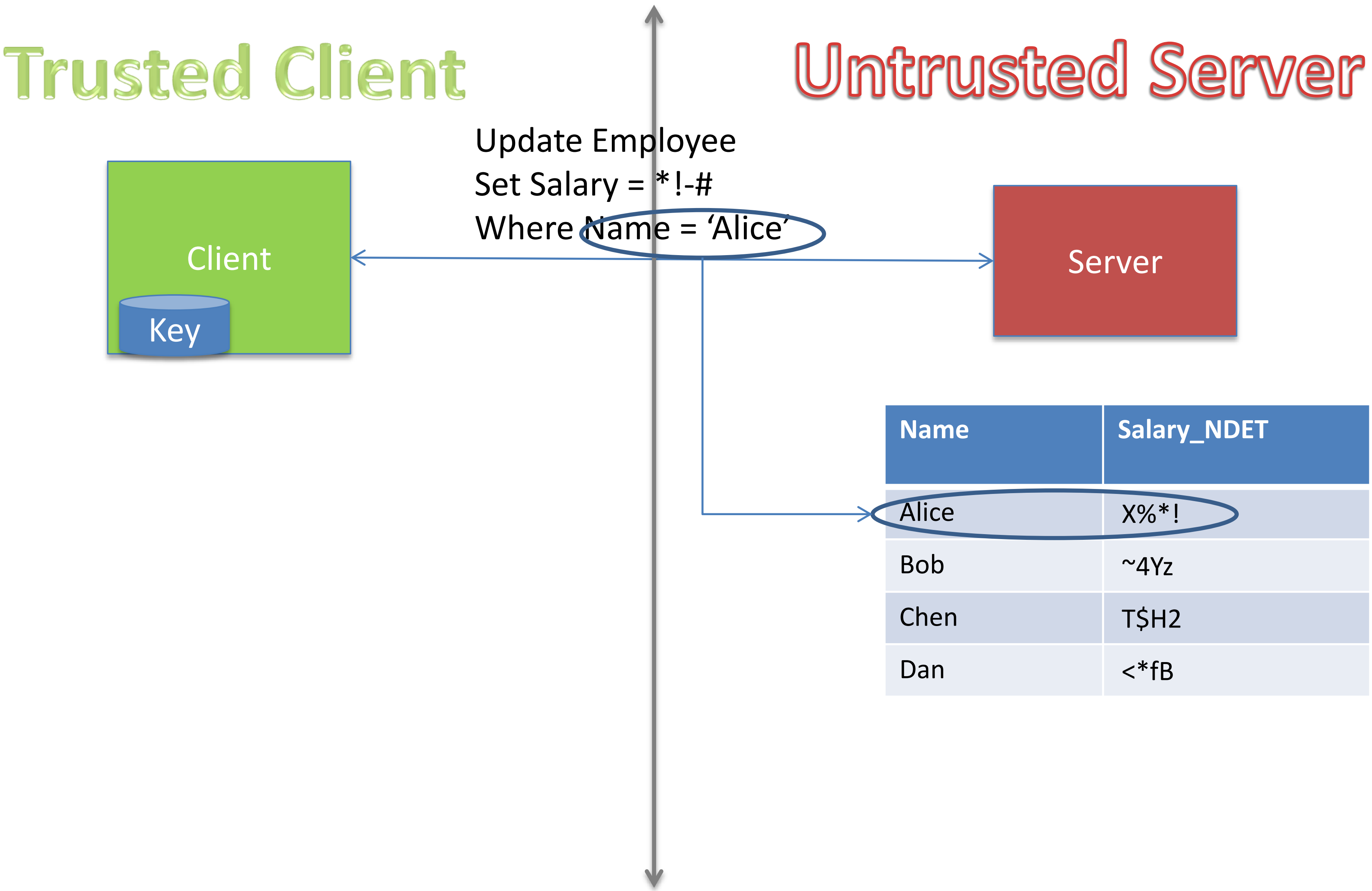
- Name: AES-CBC non-deterministic
- Age: Clear-text
- Disease: AES deterministic
- Total information leaked = "sum" of column-level leakage

Impact of Querying & Updating



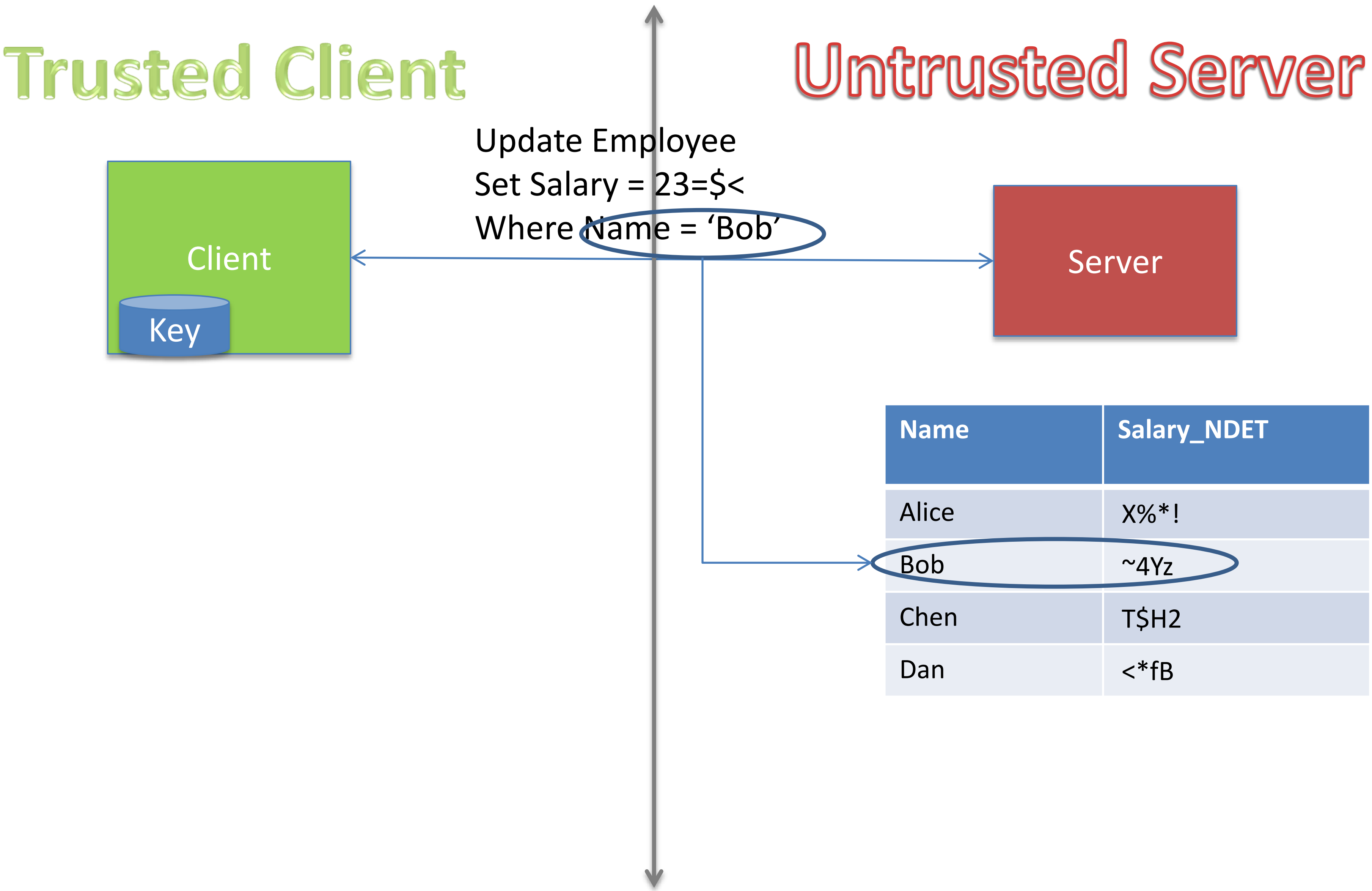
Name	Salary_NDET
Alice	X%*!
Bob	~4Yz
Chen	T$H2
Dan	<*fB

Impact of Querying & Updating



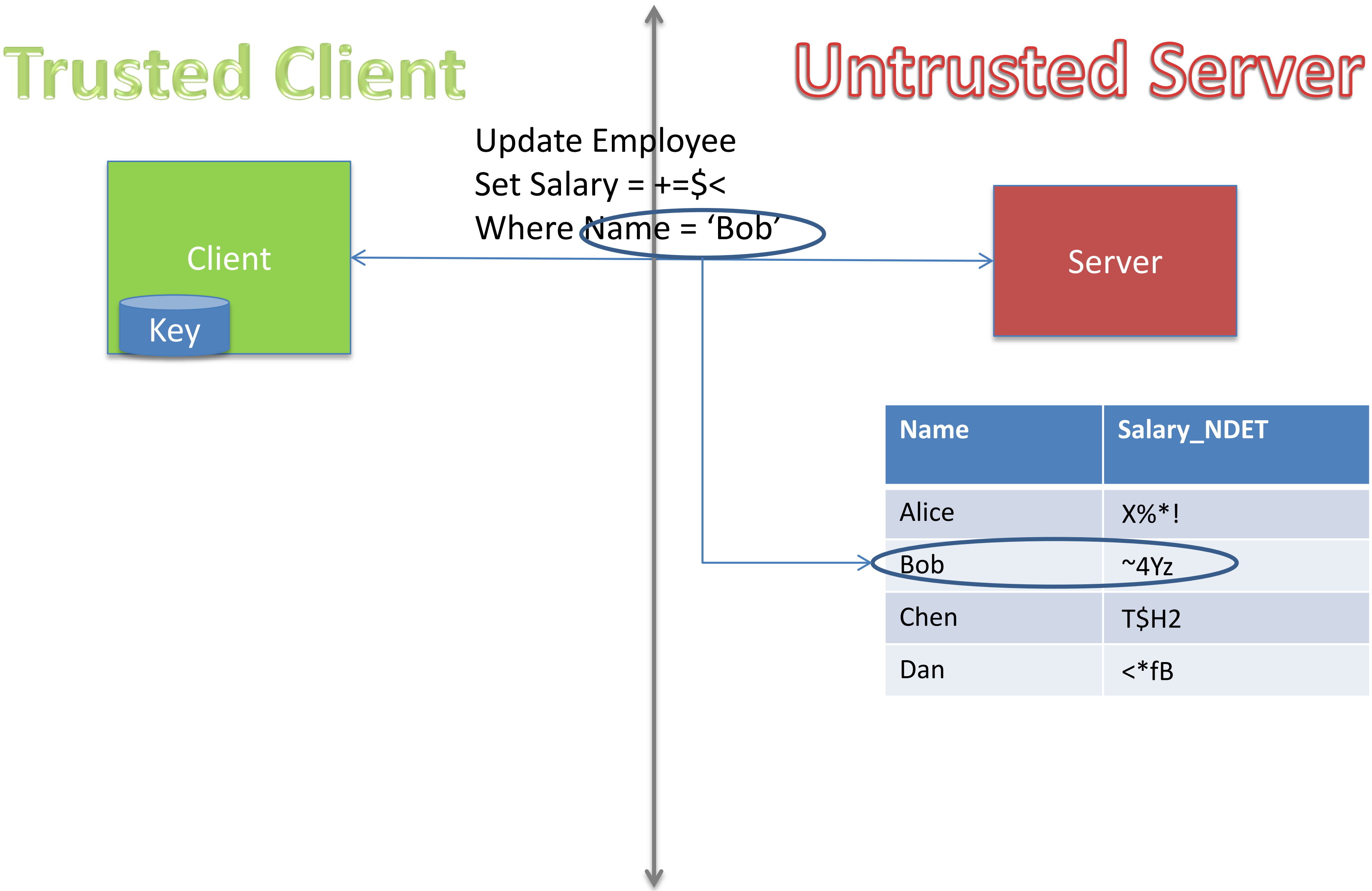
Name	Salary_NDET
Alice	X%*!
Bob	~4Yz
Chen	T$H2
Dan	<*fB

Impact of Querying & Updating



Name	Salary_NDET
Alice	X%*!
Bob	~4Yz
Chen	T$H2
Dan	<*fB

Impact of Querying & Updating

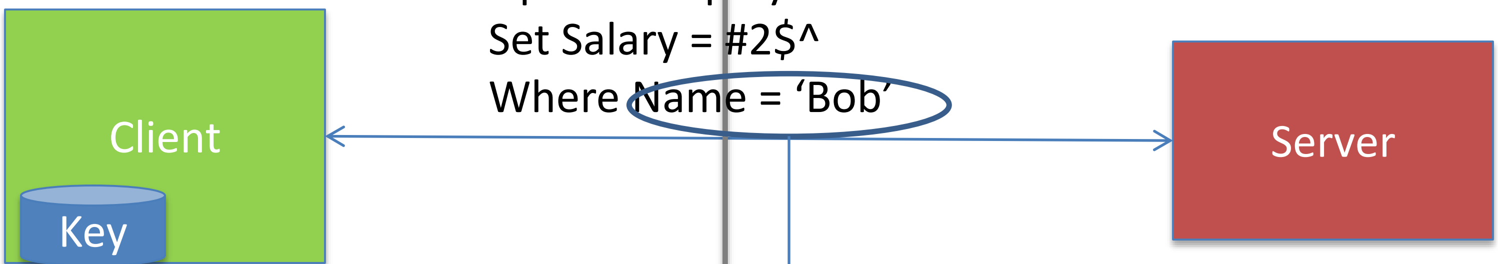


Name	Salary_NDET
Alice	X%*!
Bob	~4Yz
Chen	T$H2
Dan	<*fB

Impact of Querying & Updating

Trusted Client | Untrusted Server

Update Employee
Set Salary = #2$^
Where Name = ‘Bob’

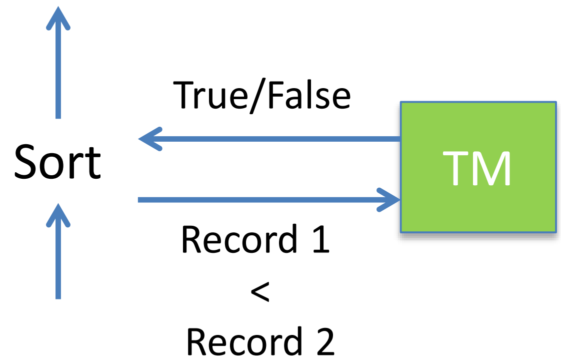


Name	Salary_NDET
Alice	X%*!
Bob	~4Yz
Chen	T$H2
Dan	<*fB

- Background knowledge
  - Full-time employees earn more
  - S… ly-wage … ated more
- Lea… em…
  - … Bob’s

Query access patterns reveal information!

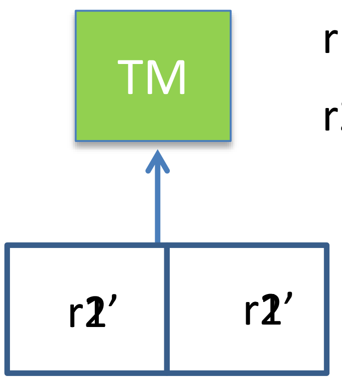
Impact of Querying & Updating



- Sort leaks ordering

What if input and outputs are all encrypted?

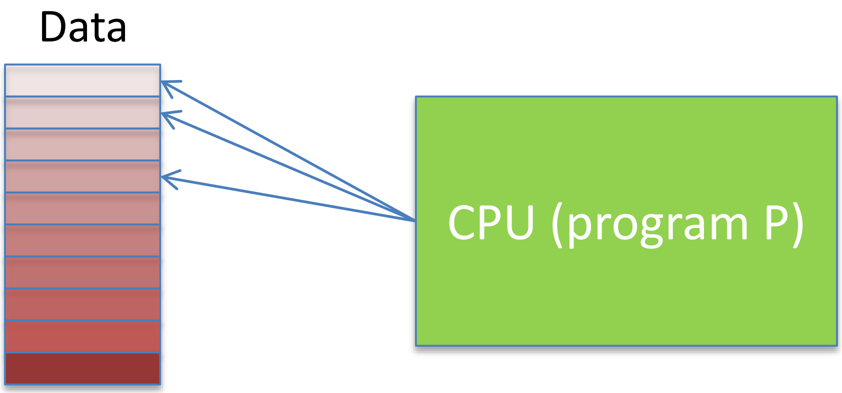
- Still leaks ordering



r1 <= r2

r2 < r1

Access Patterns Leak Information



Encryption across the stack (disk + in-memory) does NOT imply no information leakage

The overall query workflow reveals information

***Dynamic* security** (different from security of data at rest)

Design Space

Cipherbase, TrustedDB, CryptDB, Monomi, BlobStore

Stop with encryption

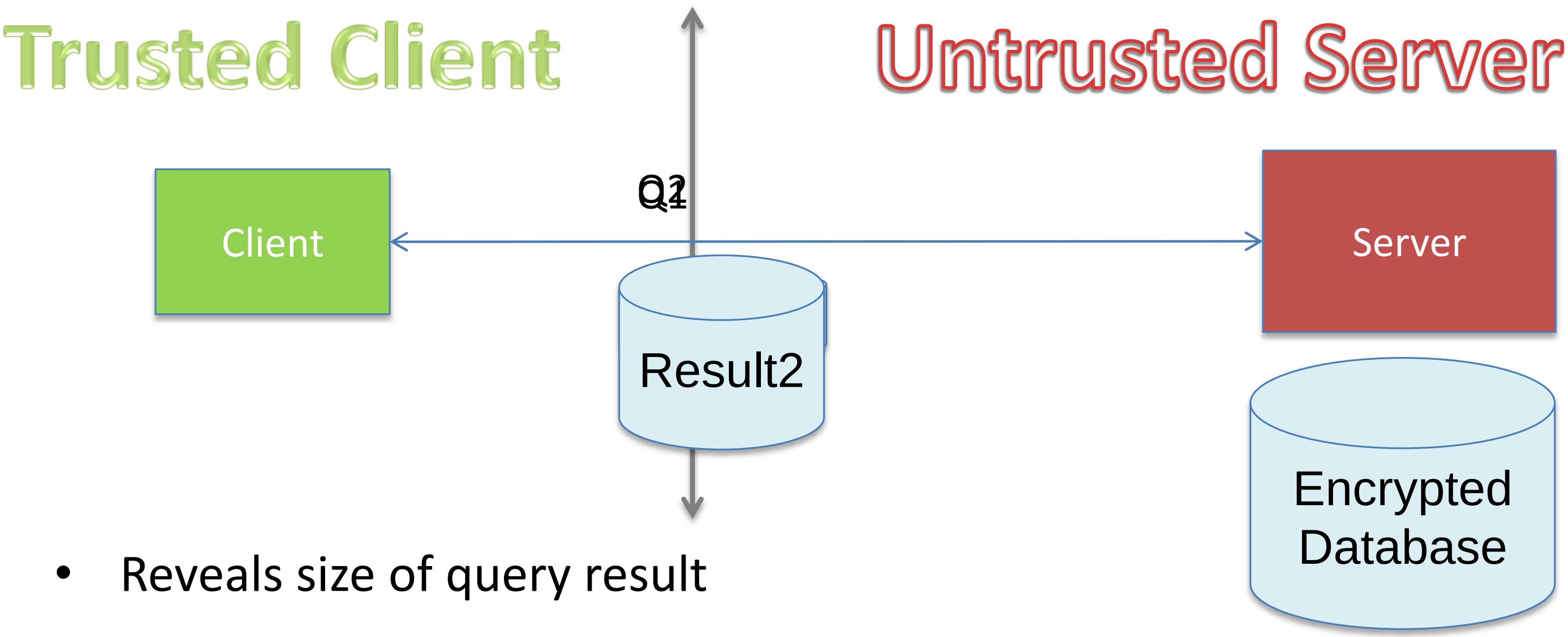
Can we bridge this gap?

Full Leakage

Operations on column	Leakage
Equality (including joins)	Frequency distribution
Indexing/Sorting /range predicates	Order

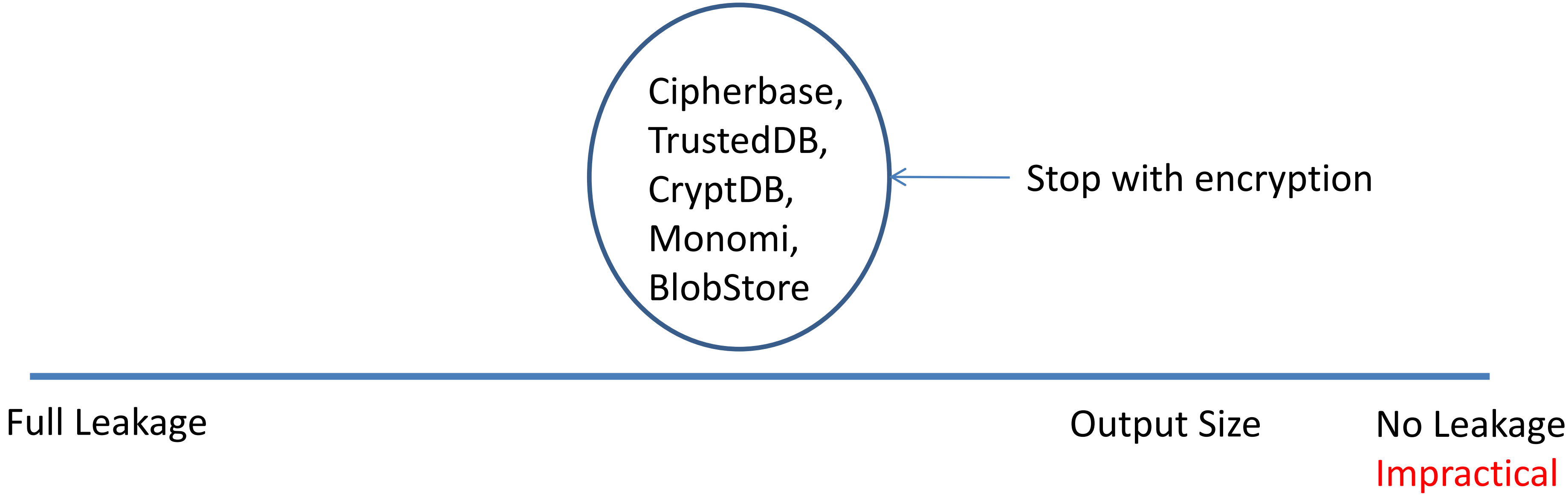
No Leakage

No Leakage



- Reveals size of query result
- Hide query result size by making all query result sizes equal to maximum size
  - Joins reduce to cross products
  - Impractical

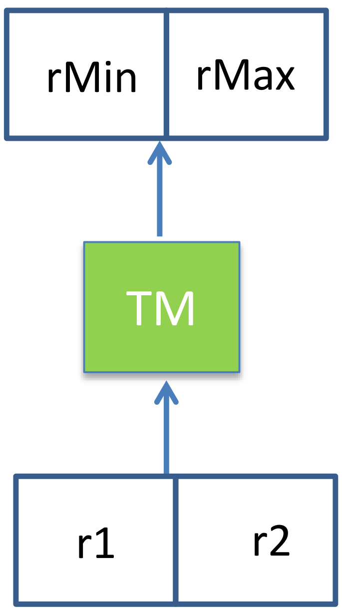
Design Space



Secure Query Processing [AK14]

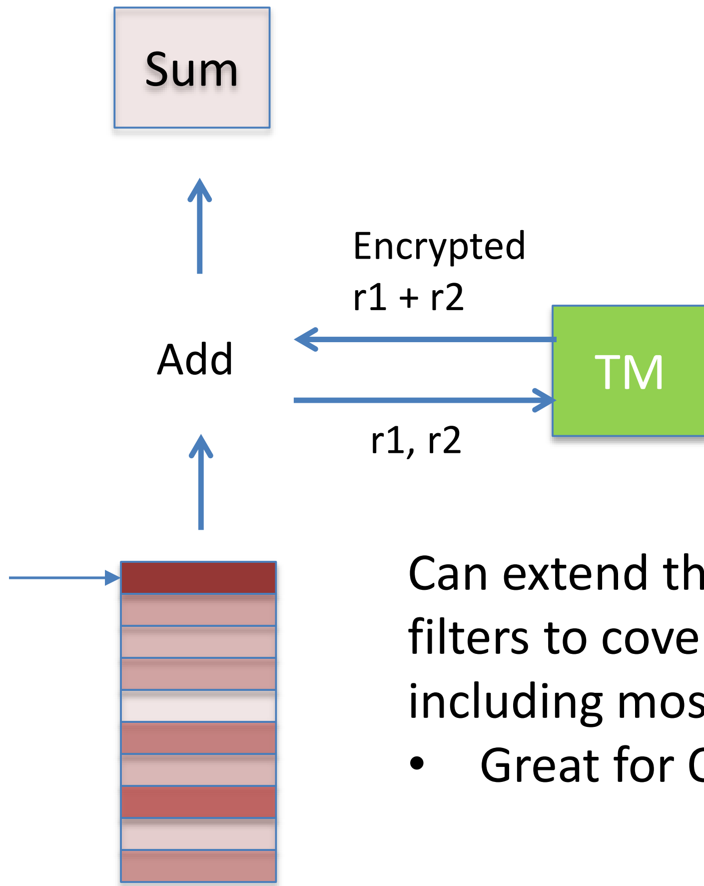
- Goal: Query Processing algorithms that reveal only output size
- Algorithms for non-trivial subset of SQL
- Approach: design *oblivious* algorithms that have same access pattern independent of input

Oblivious Sort



- Bubble sort can be made oblivious
  - Same set of comparisons independent of input
- O(n log n) oblivious sort algorithms known [Go11]

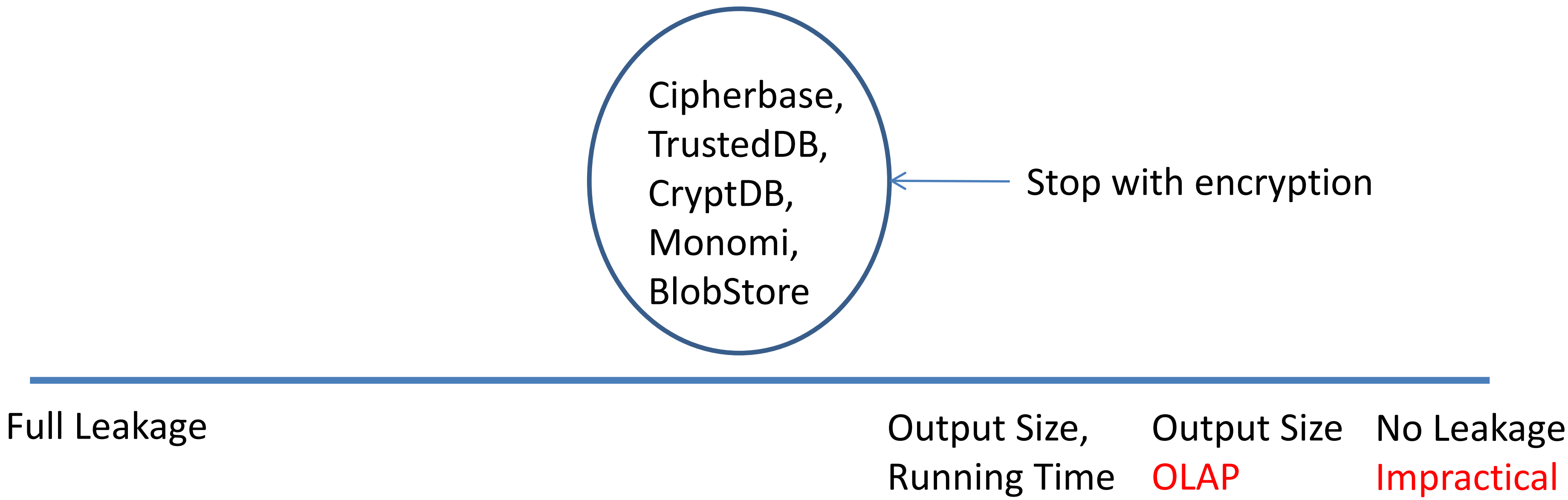
Aggregation



Can extend these ideas to group-by, join, filters to cover a large class of SQL including most TPC-H queries [AK14]

- Great for OLAP scenarios

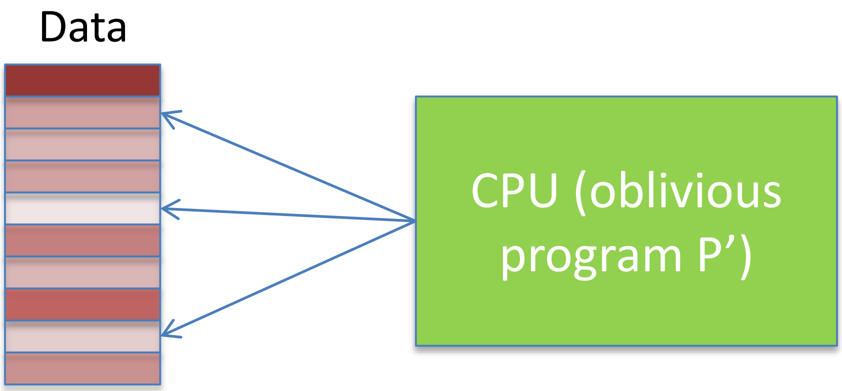
Design Space



What about full SQL?

- Updates
- Indexing
- Transactions (concurrency)
- Stored procedures
- Constraints
- Triggers

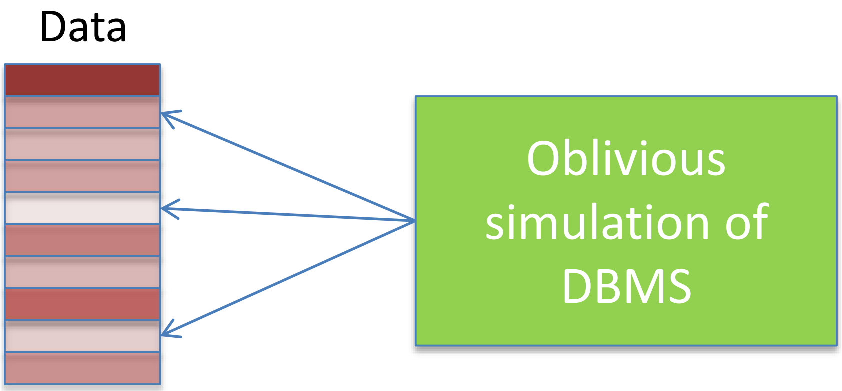
Oblivious Simulation



- Simulation: P' equivalent to P
- Theoretically Efficient: Running time of P' within polylog factor of running time of P
- Oblivious: Access patterns of P' look random
- Information leakage: input size, output size, running time

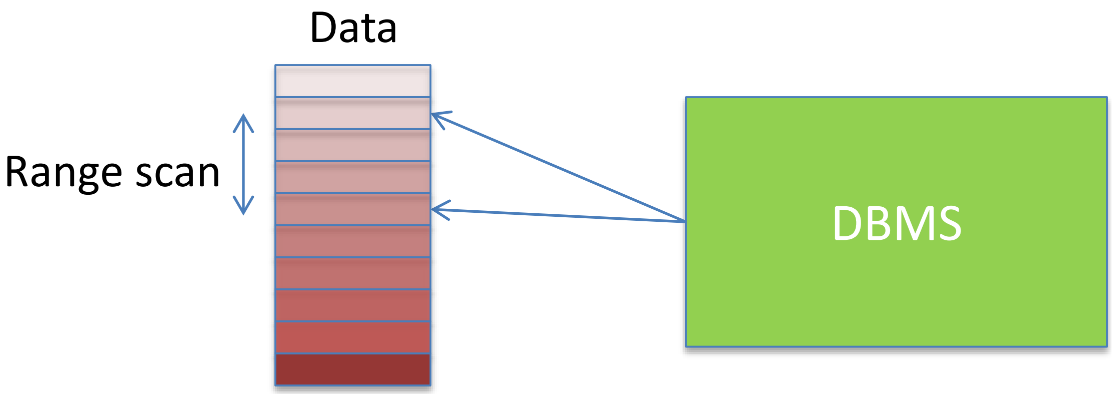
[GO96, W12, SS13]

Application to DBMS



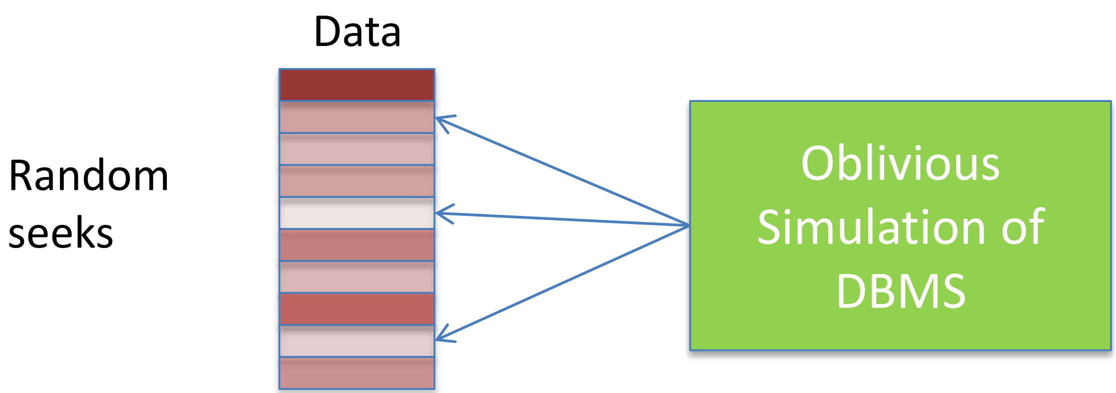
But…

- Destroys spatial and temporal locality of reference



But…

- Destroys spatial and temporal locality of reference



- Range scan of 100M records on hard disk → 100M seeks
  - 10^5 seconds (~1 day)

Design Space

Cipherbase,
TrustedDB,
CryptDB,
Monomi,
BlobStore

Stop with encryption

Full Leakage

Output Size,
Running Time
Impractical

Output Size
OLAP

No Leakage
Impractical

Is there a stronger and practically achievable security model for full SQL?

Summary

Trusted Client

DBMS

Key

Data

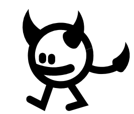
Name	Age	Disease
Alice	12	Flu
Bob	51	Diabetes
Chen	24	Flu
Dan	36	Cold

Untrusted Server

DBMS

Homo-morphic Encryption

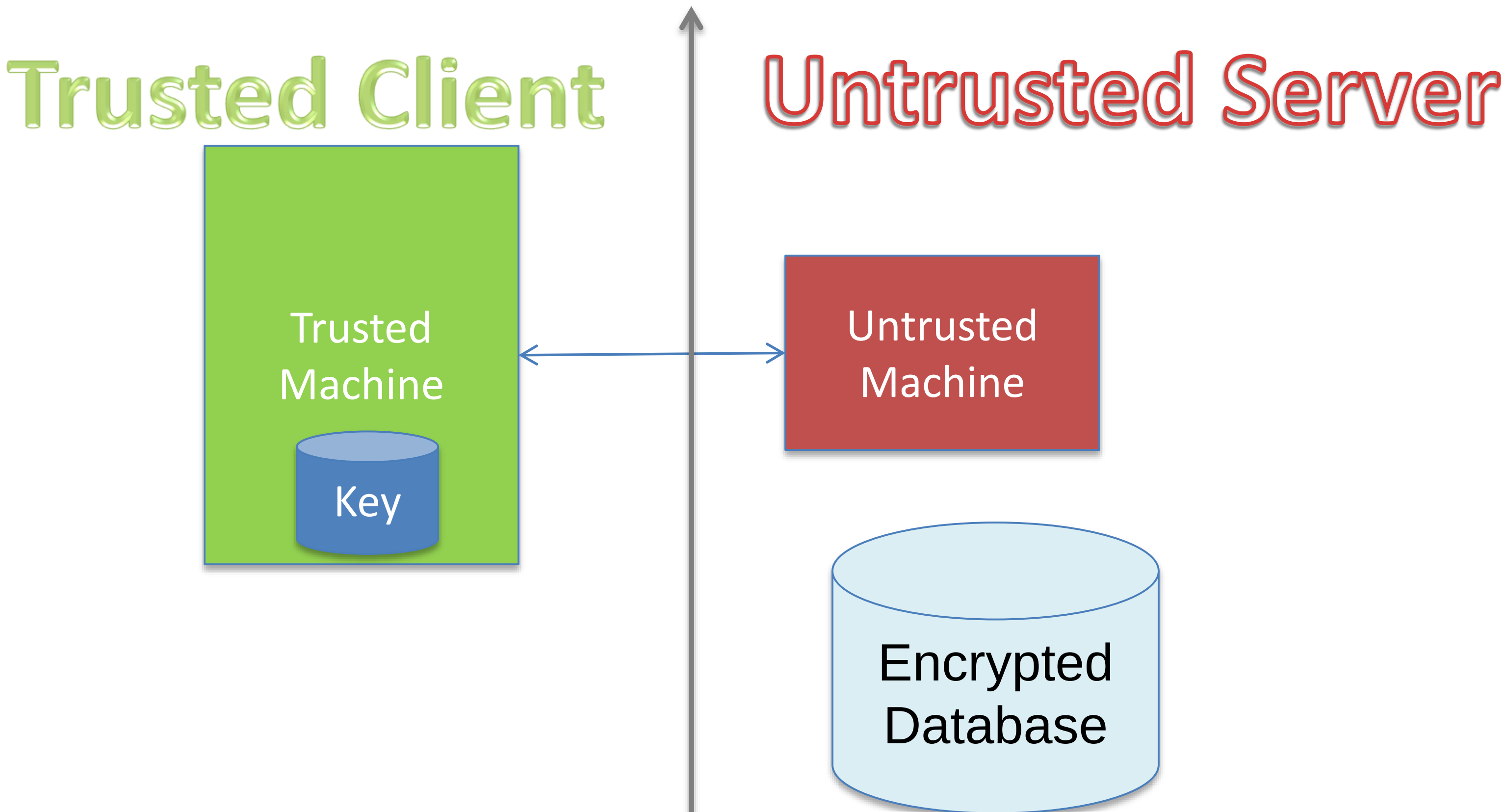
Name	Age	Disease
X%*!	)C	!x8J
~4Yz	##	)zFr#x
T$H2	!*	^@tG
<*fB	@$	BxU3



Cloud Admin

- Super-user with console access

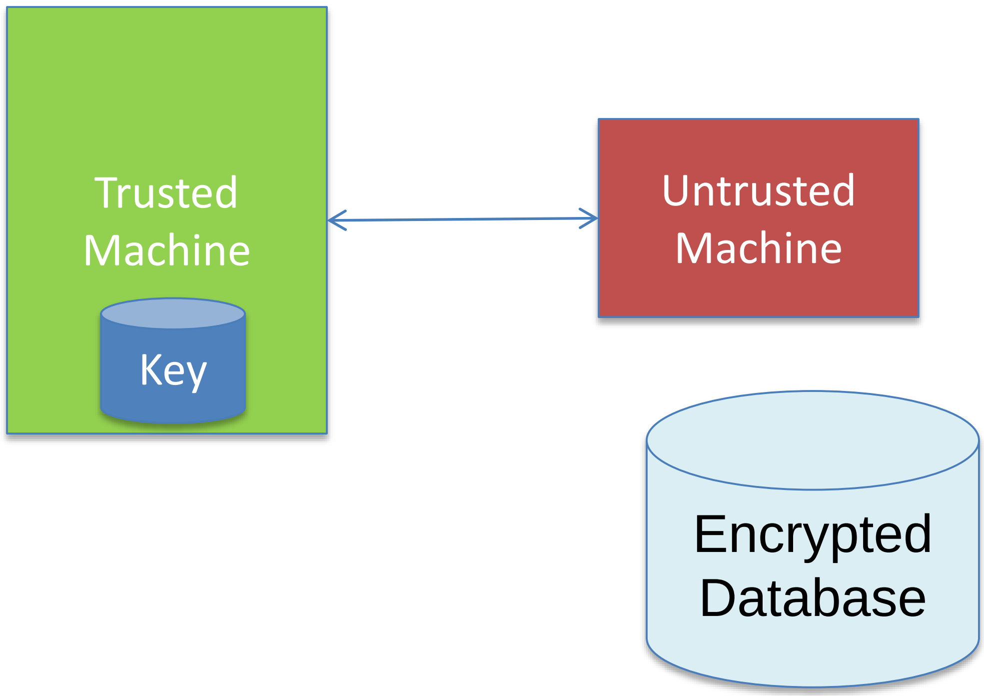
Summary



Name	Age	Disease
X%*!	)C	!x8J
~4Yz	##	)zFr#x
T$H2	!*	^@tG
<*fB	@$	BxU3

Summary

Untrusted Server



Name	Age	Disease
X%*!	)C	!x8J
~4Yz	##	)zFr#x
T$H2	!*	^@tG
<*fB	@$	BxU3

Other Challenges

- Application Security
  - DBMS is only a part of the overall system stack
- Usability
  - Clients need tools and interpretable security models to navigate security-performance tradeoff
- Connections to other areas of security
  - Data privacy, access control, auditing

Bibliography

- [ABE+12] Arvind Arasu, Spyros Blanas, Ken Eguro, Manas Joglekar, Raghav Kaushik, Donald Kossmann, Ravishankar Ramamurthy, Prasang Upadhyaya, Ramarathnam Venkatesan: Engineering Security and Performance with Cipherbase. IEEE Data Eng. Bull. 35(4): 65-72 (2012).
- [ABE+13] Orthogonal Security With Cipherbase. Arvind Arasu, Spyros Blanas, Ken Eguro, Raghav Kaushik, Donald Kossmann, Ravi Ramamurthy, and Ramaratnam Venkatesan. CIDR 2013.
- [AGJ+ 13] Ittai Anati, Shay Gueron, Simon Johnson and Vincent Scarlata. Innovative Technology for CPU Based Attestation and Sealing. Workshop on Hardware and Architectural Support for Security and Privacy. 2013.
- [AKSX04] R. Agrawal, J. Kiernan, R. Srikant, Y. Xu. Order Preserving Encryption for Numeric Data. SIGMOD 2004.
- [AES] AES Standard. FIPS 197. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf
- [AF+ 09] Above the Clouds: A Berkeley View of Cloud Computing. by Michael Armbrust, Armando Fox, and others. Tech Report EECS-2009-28, Univ. of Calif., Berkeley.
- [AKS+04] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Order-preserving encryption for numeric data. In SIGMOD 2004.

Bibliography

- [AWSGC] Amazon GovCloud. http://aws.amazon.com/govcloud-us/.
- [BS13] S.Bajaj, R. Sion. CorrectDB: SQL Engine with Practical Query Authentication. PVLDB 6(7). 2013.
- [B68] K.E. Batcher, *Sorting networks and their applications*, Proceedings of the AFIPS Spring Joint Computer Conference 32, 307–314 (1968).
- [BCD+ 05] Michael Barnett, Bor-Yuh Evan Chang, Robert DeLine, Bart Jacobs, K. Rustan M. Leino: Boogie: A Modular Reusable Verifier for Object-Oriented Programs. FMCO 2005: 364-387
- [BCL09] Order-Preserving Symmetric Encryption. Alexandra Boldyreva, Nathan Chenette, Younho Lee, Adam O'Neill. EUROCRYPT 2009.
- [BCN11] Order-Preserving Encryption Revisited: Improved Security Analysis and Alternative Solutions. Alexandra Boldyreva, Nathan Chenette, Adam O'Neill. CRYPTO 2011.
- [BFO+08] M. Bellare, M. Fischlin, A. O'Neill, T. Ristenpart: Deterministic Encryption: Definitional Equivalences and Constructions without Random Oracles. CRYPTO 2008.

Bibliography

- [BG11] Luc Bouganim, Yanli Guo: Database Encryption. Encyclopedia of Cryptography and Security (2nd Ed.) 2011.
- [BM76] R. Bayer, J. K. Metzger. On the encipherment of search trees and random access files. ACM TODS 1(1) 1976.
- [BOA] Buffer Overflow Attack. Lecture Notes. http://www.cse.scu.edu/~tschwarz/coen152_05/Lectures/BufferOverflow.html
- [BP02] Luc Bouganim, Philippe Pucheral: Chip-Secured Data Access: Confidential Data on Untrusted Servers. VLDB 2002
- [BS11] Sumeet Bajaj, Radu Sion: TrustedDB: a trusted hardware based database with privacy and data confidentiality. SIGMOD Conference 2011.
- [CloudHSM] Amazon Cloud HSM. http://aws.amazon.com/cloudhsm/
- [CPK 10] What's New About Cloud Computing Security?. Yanpei Chen, Vern Paxson and Randy H. Katz. Tech Report EECS-2010-5. Univ. of Calif., Berkeley.

Bibliography

- [E84] A public key cryptosystem and a signature scheme based on discrete logarithms. Taher El Gamal. CRYPTO 1984.
- [ENISA 09a] Cloud Computing Risk Assessment. European Network and Information Security Agency. 2009.
- [ENISA 09b] An SME perspective on cloud computing (survey). European Network and Information Security Agency, 2009.
- [G09] Fully homomorphic encryption using ideal lattices. Craig Gentry. STOC 2009.
- [G10] Computing arbitrary functions of encrypted data. Craig Gentry. CACM 2010.
- [G11] Michael T. Goodrich. Data-oblivious external-memory algorithms for the compaction, selection, and sorting of outsourced data. In SPAA, pages 379–388, 2011.
- [GO96] O. Goldreich, R. Ostrovsky: Software Protection and Simulation on Oblivious RAMs. J. ACM 43(3): 431-473 (1996)
- [GZ07] Tingjian Ge, Stanley B. Zdonik. Answering Aggregation Queries in a Secure System Model. VLDB 2007.

Bibliography

- [GZ07b] Tingjian Ge, Stanley B. Zdonik: Fast, Secure Encryption for Indexing in a Column-Oriented DBMS. ICDE 2007.
- [HIL+02] Executing SQL over Encrypted Data in the Database-Service-Provider Model. Hakan Hacigumus, Balakrishna R. Iyer, Chen Li, Sharad Mehrotra, SIGMOD 2002.
- [HIM04] Hakan Hacigümüs, Balakrishna R. Iyer, Sharad Mehrotra: Efficient Execution of Aggregation Queries over Encrypted Relational Databases. DASFAA 2004.
- [HIM05] Hakan Hacigümüs, Balakrishna R. Iyer, Sharad Mehrotra: Query Optimization in Encrypted Database Systems. DASFAA 2005.
- [HIM05b] Efficient Execution of Aggregation Queries over Encrypted Relational Databases. Hakan Hacigümüs, Balakrishna R. Iyer, Sharad Mehrotra. DASFAA 2005.
- [HLP+ 13] Matthew Hoekstra, Reshma Lal, Pradeep Pappachan and others. Using Innovative Instructions to Create Trustworthy Software Solutions. Workshop on Hardware and Architectural Support for Security and Privacy. 2013.

Bibliography

- [HMH08] Bijit Hore, Sharad Mehrotra, Hakan Hacigümüs: Managing and Querying Encrypted Data. Handbook of Database Security 2008
- [HMI02] Providing Database as a Service. Hakan Hacigumus, Sharad Mehrotra, Balakrishna R. Iyer. ICDE 2002.
- [HMT04] Bijit Hore, Sharad Mehrotra, Gene Tsudik: A Privacy-Preserving Index for Range Queries. VLDB 2004.
- [K09] G. Klein et al, "seL4: formal verification of an OS kernel" SOSP 2009.
- [KEH+ 09] Gerwin Klein, Kevin Elphinstone, Gernot Heiser, June Andronick, David Cock, Philip Derrin, Dhammika Elkaduwe, Kai Engelhardt, Rafal Kolanski, Michael Norrish, Thomas Sewell, Harvey Tuch, Simon Winwood: seL4: formal verification of an OS kernel. SOSP 2009
- [KL07] Introduction to Modern Cryptography. Jonathan Katz and Yehuda Lindell. Chapman & Hall/CRC Press. 2007.

Bibliography

- [MAB+ 13] Frank Mckeen, Ilya Alexandrovich, Alex Berenzon and others. Innovative Instructions and Software Model for Isolated Execution. Workshop on Hardware and Architectural Support for Security and Privacy. 2013.
- [MT05] E. Mykletun, G. Tsudik. Incorporating a Secure Coprocessor in the Database-as-a-Service Model. IWIA Workshop 2005.
- [NIST 09] P. Mell and T. Grance. NIST definition of cloud computing. National Institute of Standards and Technology. October 7, 2009.
- [OTDE] Oracle Transparent Data Encryption. http://www.oracle.com/technetwork/database/options/advanced-security/index-099011.html
- [P99] Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. Pascal Paillier. EUROCRYPT 1999.
- [PBH+ 11] Donald E. Porter, Silas Boyd-Wickizer, Jon Howell, Reuben Olinsky, Galen C. Hunt: Rethinking the library OS from the top down. ASPLOS 2011

Bibliography

- [PLZ13] An Ideal-Security Protocol for Order-Preserving Encoding. Raluca Ada Popa, Frank H Li, Nickolai Zeldovich. Symp on Security and Privacy, 2013.
- [PRZ+11] CryptDB: protecting confidentiality with encrypted query processing. Raluca A. Popa, Catherine M. S. Redfield, Nickolai Zeldovich, Hari Balakrishnan. SOSP 2011.
- [RAD78] R. Rivest, L. Adleman, M. Dertouzos. On Data Banks and Privacy Homomorphisms. In Foundations of Secure Computation, pages 169-178, 1978
- [S96] Applied Cryptography. Bruce Schneier. John Wiley & Sons, 1996.
- [SS05] Trusted Computing Platforms: Design and Applications. Sean W Smith. Springer. 2005.
- [SS13] E. Stefanov, E. Shi. ObliviStore: High Performance Oblivious Cloud Storage. IEEE S&P. 2013.
- [STDE] Sql Server Transparent Data Encryption. http://technet.microsoft.com/en-us/library/bb934049.aspx

Bibliography

- [TCGNotes] Trusted Computing Architecture and its applications. CS255 Lecture Notes. Stanford University. http://crypto.stanford.edu/cs155old/cs155-spring11/lectures/08-TCG.pdf
- [TFM13] Stephen Tu, M. Frans Kaashoek, Samuel Madden et al. Processing Analytical Queries over Encrypted Data. VLDB 2013.
- [TPMSpec] TPM Main Specification. http://www.trustedcomputinggroup.org/resources/tpm_main_specification
- [VYK12] Vaibhav Khadilkar, Kerim Yasin Oktay, Murat Kantarcioglu, Sharad Mehrotra: Secure Data Processing over Hybrid Clouds. IEEE Data Eng. Bull. 35(4): 46-54 (2012).
- [W12] P. Williams. Oblivious Remote Data Access Made Parallel. PhD Thesis. 2012.
- [ZCC+] Fengzhe Zhang, Jin Chen, Haibo Chen, Binyu Zang: CloudVisor: retrofitting protection of virtual machines in multi-tenant cloud with nested virtualization. SOSP 2011