PURDUE UNIVERSITY

Department of Computer Science

Practical Confidentiality Preserving Big Data Analysis in Untrusted Clouds

28 Jan 2015

Julian Stephen, Savvas Savvides, Russell Seidel
and Patrick Eugster

Problem Statement

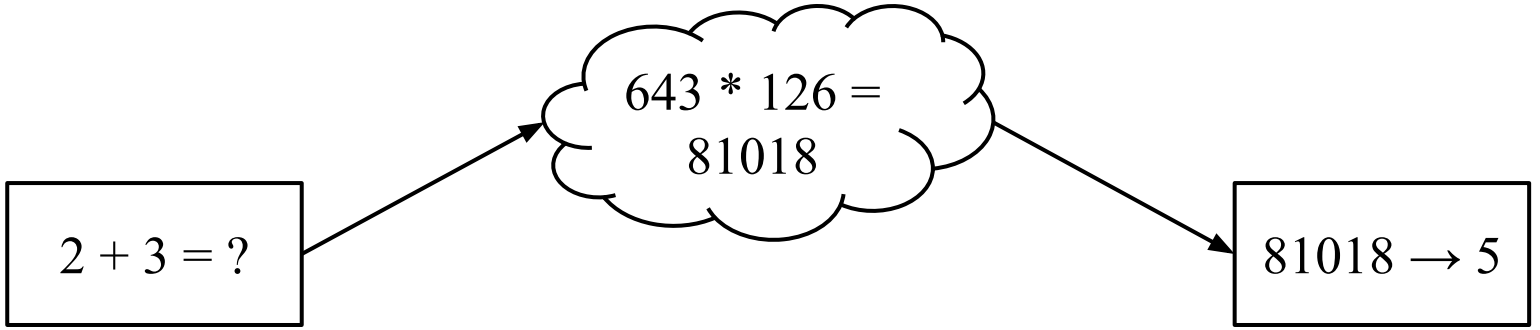
PURDUE UNIVERSITY

"Today, running your business on private servers is on the same level of odd behavior as carrying scuba tanks to provide a private air supply"
RIP Server, Peter Coffee, Mar 29, 2014

- Data breach: "The Cloud Multiplier Effect" (Ponemon Institute)
  - 36 percent of business-critical applications are housed in the cloud
  - 30 percent of business information is stored in the cloud
  - Increased use of cloud can increase the probability of a $20 million breach by 3x
- Challenges
  - How safe is it to trust a third party cloud provider?
  - How can banking, finance and insurance sectors leverage this potential?

Preserving Confidentiality

- *Fully* homomorphic encryption (FHE)
  - Prohibitive overhead, getting more practical
  - Limited expressiveness
- *Partially* homomorphic encryption (PHE)
  - Allows for certain operations to be performed in encrypted form
  - Paillier [Paillier;EuroCrypt'99] ► AHE: $D(E(x1) \oplus E(x2)) = x1 + x2$
  - ElGamal [ElGamal; ToIT'86] ► MHE: $D(E(x1) \oplus E(x2)) = x1 * x2$
  - DET (=), OPE (<)



Crypsis Intuition

PURDUE UNIVERSITY

- Avoid using FHE, use more practical PHE cryptosystems instead
- Partition programs according to attributes and use a different PHE cryptosystem for each
- We can use multiple PHE cryptosystems of the same column in parallel
- Use re-encryption when PHE alone would fail
  - Use trusted base to decrypt and encrypt under desired cryptosystem
  - May be faster than FHE

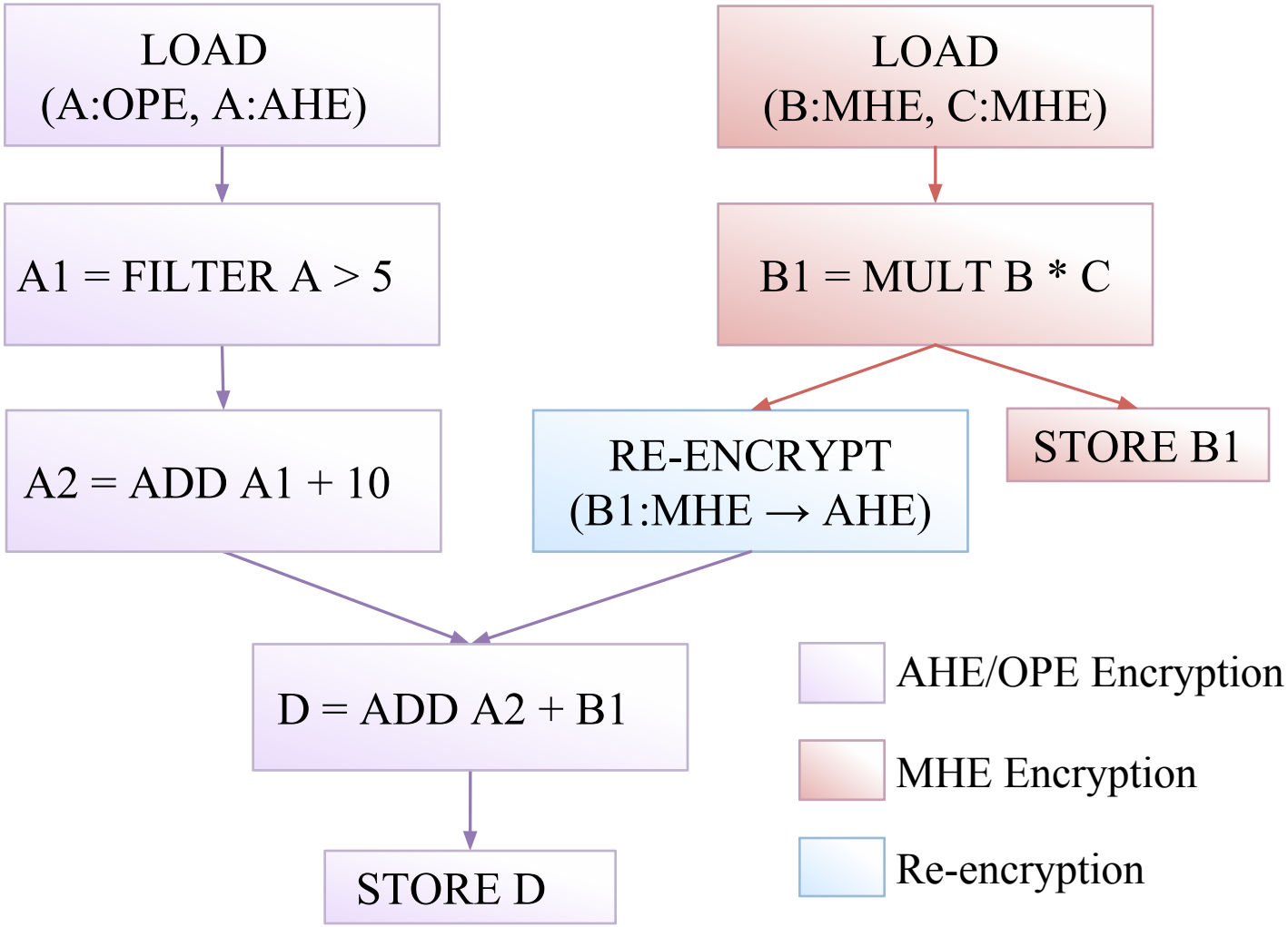
Background

PURDUE UNIVERSITY

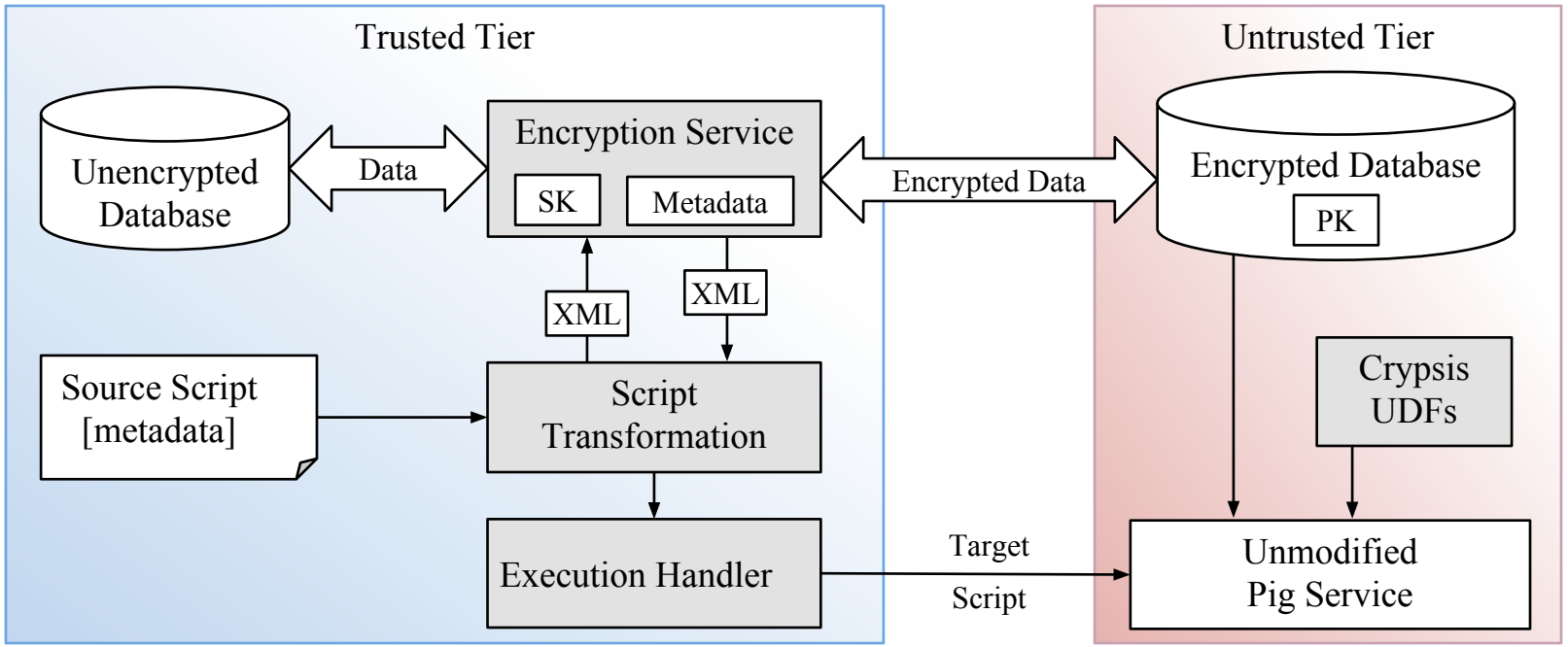
- Mapreduce [Dean&Ghemawat; OSDI'04]
  - Parallel execution (map and reduce functions)
  - Hadoop version 1.2.1
- Pig and Pig Latin [Gates et.al; VLDB'09]
  - Pig Latin - Data flow language for expressing data analysis programs
  - Pig - runtime environment, generates Mapreduce programs
- Example Pig Latin script

```
A = LOAD "infile" AS (a0, a1);
B = FILTER A BY a0 > 10;
C = GROUP B BY a1;
D = FOREACH C GENERATE SUM(C.a0) AS b1;
STORE D INTO "outfile";
```

Example


LOAD
(A:OPE, A:AHE)
LOAD
(B:MHE, C:MHE)
A1 = FILTER A > 5
B1 = MULT B * C
A2 = ADD A1 + 10
RE-ENCRYPT
(B1:MHE → AHE)
STORE B1
D = ADD A2 + B1
STORE D
AHE/OPE Encryption
MHE Encryption
Re-encryption

Architecture Overview



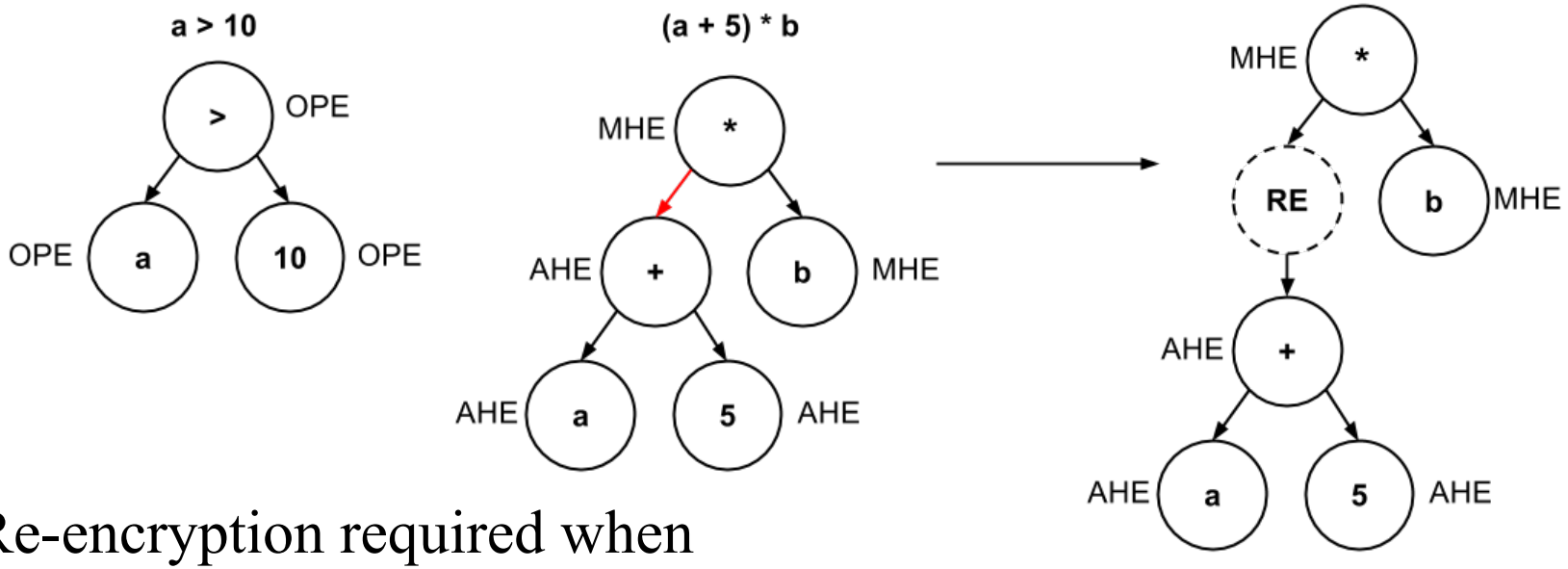
Practical Confidentiality Preserving Big Data Analysis
[J. Stephen et al; USENIX HotCloud14]

Script Transformation

PURDUE UNIVERSITY

- Script analysis
  - Generate Data Flow Graph (DFG)
  - Nodes are relational operations (LOAD, FOREACH, etc…)
  - Edges are data flow between relational operations
- Generate Map of Expression Trees (MET)
  - Contains all expressions of the script
  - Keys are used to assign expressions to DFG
- Generate Set of Annotated Fields (SAF)
  - One entry for each <relation, field> of the script
  - <relation, field>, parent, available encryptions, required encryptions
  - Get available encryptions from lineage of field (parent)
  - Get required encryptions using MET

Identifying re-encryptions



- Re-encryption required when
  - Required encryption not available
  - Incompatible operations e.g. addition followed by a multiplication
- 17 PigMix2 benchmarks (PigMix1 + 5)
  - Only script 8 requires re-encryption (averaging)
  - 1 additional script requires same attribute available in 2 encryptions

Transformation Example

PURDUE UNIVERSITY

Source Script

```
EMP = LOAD "employees" AS (
   salary,
   department);
HIGH_PD = FILTER EMP BY
   salary > 80000;
HP_DEP = GROUP HIGH_PD BY
   department;
TOTAL = FOREACH HP_DEP GENERATE
   group AS department,
   SUM(HIGH_PD.salary) AS total;
STORE TOTAL INTO "salary_per_dep";
```

Target Script

```
EMP = LOAD "employees_enc" AS (
   salary_ope, salary_ahe,
   department_det);
HIGH_PD = FILTER EMP BY
   ENCGT(salary_ope, 98...24);
HP_DEP = GROUP HIGH_PD BY
   department_det;
TOTAL = FOREACH HP_DEP GENERATE
   group AS department_det,
   ENCSUM(HIGH_PD.salary_ahe) AS t;
STORE F INTO "salary_per_dep_enc";
```

Program Analysis for Secure Big Data Processing
[J. Stephen et al; IEEE/ACM ASE2014]

Name Obfuscation

PURDUE UNIVERSITY

Target Script

```
EMP = LOAD "employees_enc" AS (
   salary_ope, salary_ahe,
   department_det);
HIGH_PD = FILTER EMP BY
   ENCGT(salary_ope, 98...24);
HP_DEP = GROUP HIGH_PD BY
   department_det;
TOTAL = FOREACH HP_DEP GENERATE
   group AS department_det,
   ENCSUM(HIGH_PD.salary_ahe) AS total;
STORE F INTO "salary_per_dep_enc";
```

Obfuscated Target Script

```
A = LOAD "input" AS (
   a1, a2,
   a3);
B = FILTER A BY
   f1(a1, 98...24);
C = GROUP B BY
   a3;
D = FOREACH C GENERATE
   group AS d1,
   f2(B.a2) AS d2;
STORE F INTO "output";
```

Crypsis Implementation

PURDUE UNIVERSITY

- Compiler
  - Performs script transformation
  - Identifies and applies optimizations
- Crypsis UDFs
  - Replace operations and aggregation functions with their encrypted version
  - Allows for an unmodified Pig service
- Expressiveness
  - +, -, ~, *, ^, ==, <, ≤, >, ≥
  - Aggregation functions: SUM, MAX, MIN, DISTINCT, ORDERBY, AVG, MEDIAN, ABS
  - Negative numbers
  - Floating point numbers (limited)

Optimizations

PURDUE UNIVERSITY

- Minimize number of re-encryptions
  - Expression rewriting
  - Operation reordering
  - Selective encryption
- Avoid redundant computations
  - Repeated sub-expressions
- Reduce amount of data computed on
  - Filter reordering
  - Packing

Expression Rewriting / Operation Reordering

PURDUE UNIVERSITY

Before

```
A = LOAD "infile" AS (col);
B = FOREACH A GENERATE
    col * 2 AS x,
    col * 3 AS y;
C = FOREACH B GENERATE
    (x + y) * 2 AS result;
STORE C INTO "outfile";
```

After

```
A = LOAD "infile" AS (col);
B = FOREACH A GENERATE
    col * 2 AS x,
    col * 3 AS y;
C = FOREACH B GENERATE
    x * 2 + y * 2 AS result;
STORE C INTO "outfile";
```

Before

```
A = LOAD "infile" AS (col);
B = FOREACH A GENERATE
    col + 10 AS x;
C = ORDER B BY x;
STORE C INTO "outfile";
```

After

```
A = LOAD "infile" AS (col);
B = ORDER A BY col;
C = FOREACH B GENERATE
    col + 10 AS x;
STORE C INTO "outfile";
```

Selective Encryption

- Often only parts of the input data hold sensitive information
  - Selectively encrypt
  - Reduce overall size of data
  - Reduce required re-encryptions e.g. (a + b) * c
- Secondary homomorphic property
  - AHE: D(E(x1) ⊙ x2) = x1 * x2
  - MHE: D(E(x1) ⊙ x2) = x1 ^ x2

```
X = ENC_ADD(a_ahe, b_ahe)
Y = REENCRYPT(X, ahe->mhe)
Z = ENC_MULT(Y, c_mhe)
```

```
X = ENC_ADD(a_ahe, b_ahe)
Y = ENC_PMULT(X, c_plain)
```

Repeated Sub-expressions

PURDUE UNIVERSITY

Before

```
ITEMS = LOAD "infile" AS (price, discount, tax);
PRICES = FOREACH ITEMS GENERATE
   price * (1 - discount) AS disc_price,
   price * (1 - discount) * (1 + tax) AS charge;
STORE C INTO "outfile";
```

After

```
ITEMS = LOAD "infile" AS (price, discount, tax);
DISCOUNT = FOREACH ITEMS GENERATE
   price * (1 - discount) AS disc_price, tax;
PRICES = FOREACH DISCOUNT GENERATE
   disc_price,
   disc_price * (1 + tax) AS charge;
STORE C INTO "outfile";
```

Filter Reordering

PURDUE UNIVERSITY

Before

```
A = LOAD "infile" AS (
    salary,
    rating);
B = FOREACH A GENERATE
    salary + 100 AS bonus,
    rating;
C = FILTER B BY
    rating > 8;
STORE C INTO "outfile";
```

After

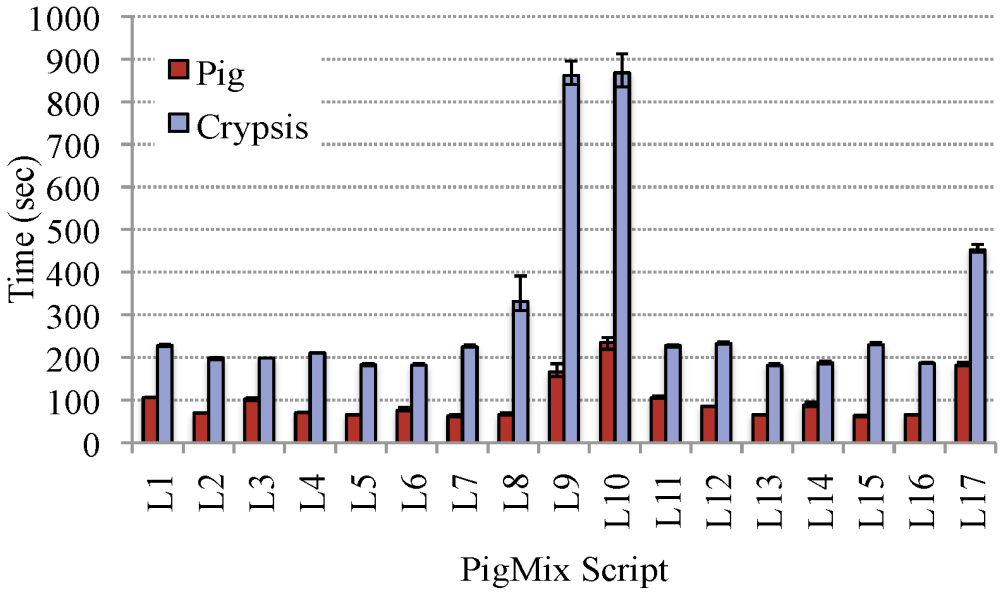
```
A = LOAD "infile" AS (
    salary,
    rating);
B = FILTER A BY
    rating > 8;
C = FOREACH B GENERATE
    salary + 100 AS bonus,
    rating;
STORE C INTO "outfile";
```

Packing

- Ciphertext space is much larger than plaintext space
- Pack multiple values in a single plaintext before encrypting
- Must handle overflows
- Can be applied on AHE and MHE (limited)

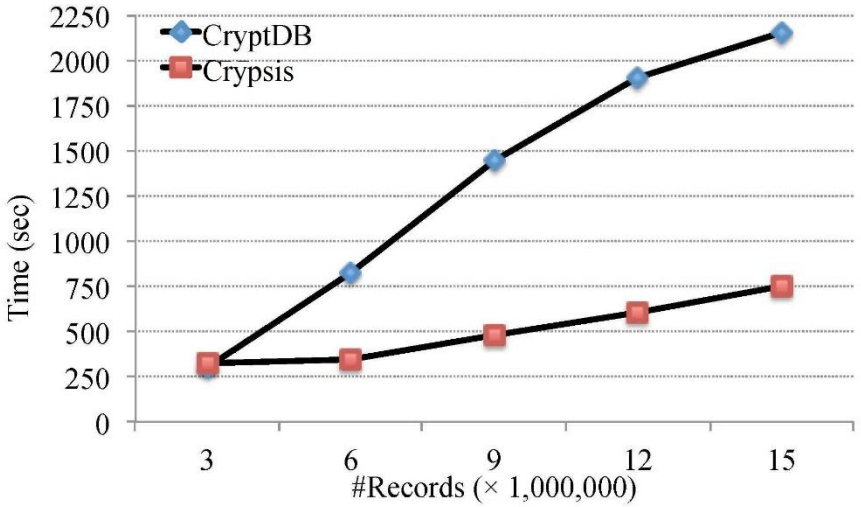
Padding n	x_n	...	Padding 1	x_1
		+		
Padding n	y_n	...	Padding 1	y_1
		=		
$x_n + y_n$			$x_1 + y_1$	

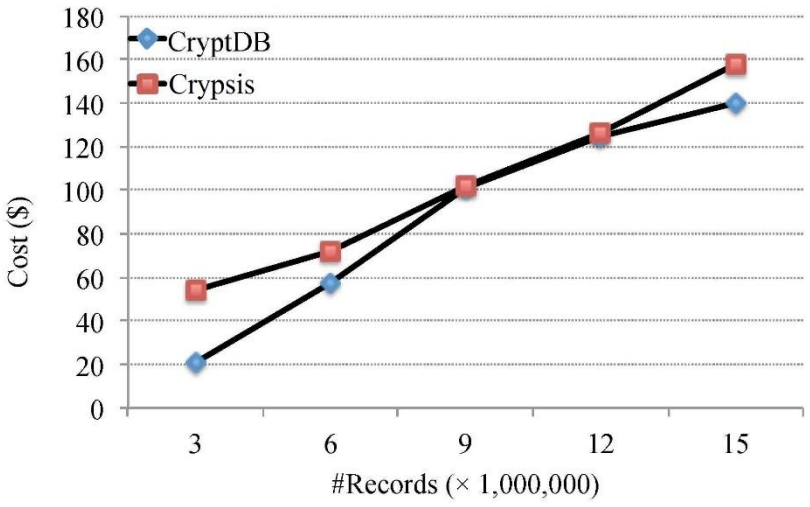
Evaluation (PigMix)



- 11 ec2 c3.large instances (2 vCPUs, 3.75GB RAM)
- 5GB of data (over 3 million rows)
- An average of 3x overhead in terms of latency
- FHE can exhibit several 100 times overhead

Evaluation (CryptDB Comparison)





- 3 m3.medium instances (1 vCPUs, 3.75GB RAM)
- ~ 3x faster for 15 million records (7GB)
- Similar overall cost

Related Work

PURDUE UNIVERSITY

- CryptDB [Popa et al.;SOSP'11]
  - Encrypted database for MySQL (subset)
  - No Parallelism
  - No re-encryption; client-side query completion
- Monomi [Tu et al; VLDB'13]
  - Uses techniques to improve performance of complex queries on encrypted data
  - Built on top of Postgres, Centralized Design
- MrCrypt [Lesani et al.;OOPSLA'13]
  - Program analysis for individual MapReduce tasks
  - No re-encryption

Conclusions and Future Work

PURDUE UNIVERSITY

- Cloud computing
  - On demand computation infrastructure has great potential
  - Inherent confidentiality concerns
- Crypsis
  - Addresses confidentiality concerns.
  - Efficient big data analysis over encrypted data
- Future work
  - More fine-grained encryption system (annotations)
  - Identify more opportunities to reduce re-encryptions

Thank you!

PURDUE UNIVERSITY

? && /**/