Practical Confidentiality Preserving Big Data Analysis

Julian Stephen, Savvas Savvides, Russell Seidel & Patrick Eugster

Purdue University
Introduction

• Potential of cloud
  – Computation infrastructure available on demand
  – Ability to adapt to changing requirements
  – Institutions opt for “in-house” clouds for data storage and analysis

• Challenges

June 12, 2014

Redwood Regional Medical Group
Santa Rosa, California

MED  PHYS

A thumb drive containing 33,702 patient records was stolen from the Redwood Regional Medical Group in Santa Rosa California. An employee placed the thumb drive in a "zipped container in an unlocked locker", where the drive was stolen.

June 11, 2014

PF Chang's
Scottsdale, Arizona

BSO  HACK

P.F Chang's is investigating a potential data breach, when credit cards showed up on an underground website that criminals use. Brian Krebs broke the story, when the banks he contacted confirmed that the cards had been used at P.F Chang's restaurants.

http://www.privacyrights.org/
Confidentiality in the Cloud

• Fully homomorphic encryption (FHE)
  – Prohibitive overhead
  – Limited expressiveness

• Partially homomorphic encryption (PHE)
  – Allows certain operations to be performed over encrypted text
  – E.g.,
    • Paillier [Paillier; EuroCrypt’99] ► AHE
    • ElGamal [ElGamal; IEEE ToIT’86] ► MHE

• Conjecture
  – Many big data analysis jobs can be performed using a combination of partially homomorphic cryptosystems
Background

• Partially homomorphic encryption
  – AHE: $D( E( x_1 ) \psi E( x_2 ) ) = x_1 + x_2$
  – MHE: $D( E( x_1 ) \psi E( x_2 ) ) = x_1 \times x_2$

• Deterministic encryption
  – DET: $x_1 = x_2 \Rightarrow E( x_1 ) = E( x_2 )$

• Order preserving encryption
  – OPE: $x_1 < x_2 \Rightarrow E( x_1 ) < E( x_2 )$

• MapReduce [Dean&Ghemawat; OSDI'04]

• Pig and PigLatin [Gates et.al; VLDB’09]
  – Pig Latin: high level data flow language for expressing data analysis programs
  – Pig: runtime environment, generates MapReduce programs
Crypsis

• Goal: Confidentiality preserving big data analysis
  – Leverage advantages of PHE whenever possible
    • Multiple column encryption, expression rewriting
  – Client side completion
  – Transform Pig Latin scripts to accept encrypted input

• E.g., (source PigLatin program)

A = LOAD ‘input1’ AS (a0, a1);
B = LOAD ‘input2’ AS (x0);
C = FILTER A BY a0 > 10;
D = GROUP C BY a1;
E = FOREACH D GENERATE group AS b0, SUM(C.a0) AS b1;
F = JOIN E BY b0, B BY x0;
STORE F INTO ‘out’;
Crypsis: Transformation Example [1]

- Target script

```
A = LOAD 'enc_input1' AS (a0_ope, a0_ah, a1_det);
B = LOAD 'enc_input2' AS (x0_det);
C = FILTER A BY OPE_GR(a0_ope, 0xD0004D3D841327F2CCE71..);
D = GROUP C BY a1_det;
E = FOREACH D GENERATE group AS b0, ENCSUM(C.a0_ah) AS b1;
F = JOIN E BY b0, B BY x0_det;
STORE F INTO 'out';
```
Crypsis: Transformation Example [2]

• Line 1

\[
A = \text{LOAD} \ 'input1' \ AS \ (a0, a1);
\]

\[
A = \text{LOAD} \ 'enc\_input1' \ AS \ (a0\_ope, a0\_ah, a1\_det);
\]

• Line 3

\[
C = \text{FILTER} \ A \ BY \ a0 > 10;
\]

\[
C = \text{FILTER} \ A \ BY \ \text{OPE\_GR}(a0\_ope, \ 0x\text{D0004D3D841327F2CCE71...});
\]
Crypsis: Architecture

Database

Pig Script

Encryption Service

Script Transformation

Coordinator

Encrypted Database

Secure UDFs

Unmodified Pig Service

Data

Encrypted Data

Transformed Script

Trusted tier (Client)

Untrusted tier (Server)
Transformation Steps

• **Script analysis**
  – Generate Data Flow Graph (DFG)
  – Relations as nodes and data flows as edges

• **Identify encryption scheme required for fields**
  – Extract input fields and expressions in DFG
  – Track lineage of fields using DFG edges

• **Identify encryption schemes available**
  – Uses MET and Encryption Service

• **Transformation**
  – Generates new DFG using available encryption schemes
  – Replace operations by their cryptographic equivalents
A = LOAD ‘input1’ AS (a0, a1);
B = LOAD ‘input2’ AS (x0);
C = FILTER A BY a0 > 10;
D = GROUP C BY a1;
E = FOREACH D GENERATE group AS b0, SUM(C.a0) AS b1;
F = JOIN E BY b0, B BY x0;
STORE F INTO ‘out’;

*det = assert_deterministic
Transformation Steps

• Script analysis
  – Generate Data Flow Graph (DFG)
  – Relations as nodes and data flows as edges

• Identify encryption scheme required for fields
  – Extract input fields and expressions in DFG
  – Track lineage of fields using DFG edges

• Identify encryption schemes available
  – Uses MET and Encryption Service

• Transformation
  – Generates new DFG using available encryption schemes
  – Replace operations by their cryptographic equivalents
Re-encryption

- PHE not ‘always’ sufficient
  
  \[
  A = \text{LOAD} \text{ ‘input1’ AS } (a_0, a_1, a_2);
  \]
  
  \[
  B = \text{FILTER} A \text{ BY } (a_0 + a_1) < 10;
  \]
  
  \[
  \text{STORE} B \text{ INTO} \text{ ‘out’};
  \]

- Re-encryption

  \[
  B = \text{FILTER} A \text{ BY } \text{OPE\_GR( REENC(a_0 + a_1, ‘OPE’) , OPE(10))};
  \]
Evaluation

• PigMix2
  – 11 ec2 c3.large instances (2 vCPUs, 3.75GB ram)
  – Over 3 million rows (5GB of data)
Current Limitations

• Iterations
  – Pig Latin is a query based language
  – No support for iterations or recursion
  – Iterations over encrypted data can be very expensive

• User defined functions (UDFs)
  – Many big data analysis languages propose UDFs vs. built-in/pre-defined operators and functions
  – Black-boxes from perspective of program analysis (cannot support transformation)
  – Can analyze byte-code on Java UDFs
Related Work

• CryptDB [Popa et al.; CACM’12]
  – Encrypted database for SQL (subset)
  – No re-encryption; client-side query completion
  – No parallelism

• MrCrypt [Lesani et al.; OOPSLA’13]
  – Program analysis for individual MapReduce tasks
  – “Language”-independent, but limited

• DepSky [Bessani et al.; Eurosys’11]
  – Storage
  – Quorum-based replication with secret sharing for privacy and integrity
  – Only AHE
Conclusion & Future work

• Confidentiality is a concern

• PHE has great potential in this direction

• Crypsis
  – Proof of concept for PigLatin based big data analysis over encrypted data
  – Identifies appropriate encryption schemes and transforms source program to work on encrypted data
  – PigMix queries over encrypted data with 3x overhead

• Future work
  – Sampling based heuristics to identify better execution path
  – Expression rewriting to further improve expressivity
  – Ability to specify field level confidentiality setting