Secure Grouping and Aggregation with MapReduce

Radu Ciucanu Matthieu Giraud
Pascal Lafourcade Lihua Ye

28 July 2018
SECRYPT, Porto

LABORATOIRE D'INFORMATIQUE,
DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES

Example of Grouping and Aggregation

Name	Department	Salary
Alice	Computer Science	1900
Bob	Mathematics	1750
Mallory	Computer Science	1800
Oscar	Physics	2000
Carol	Mathematics	1600

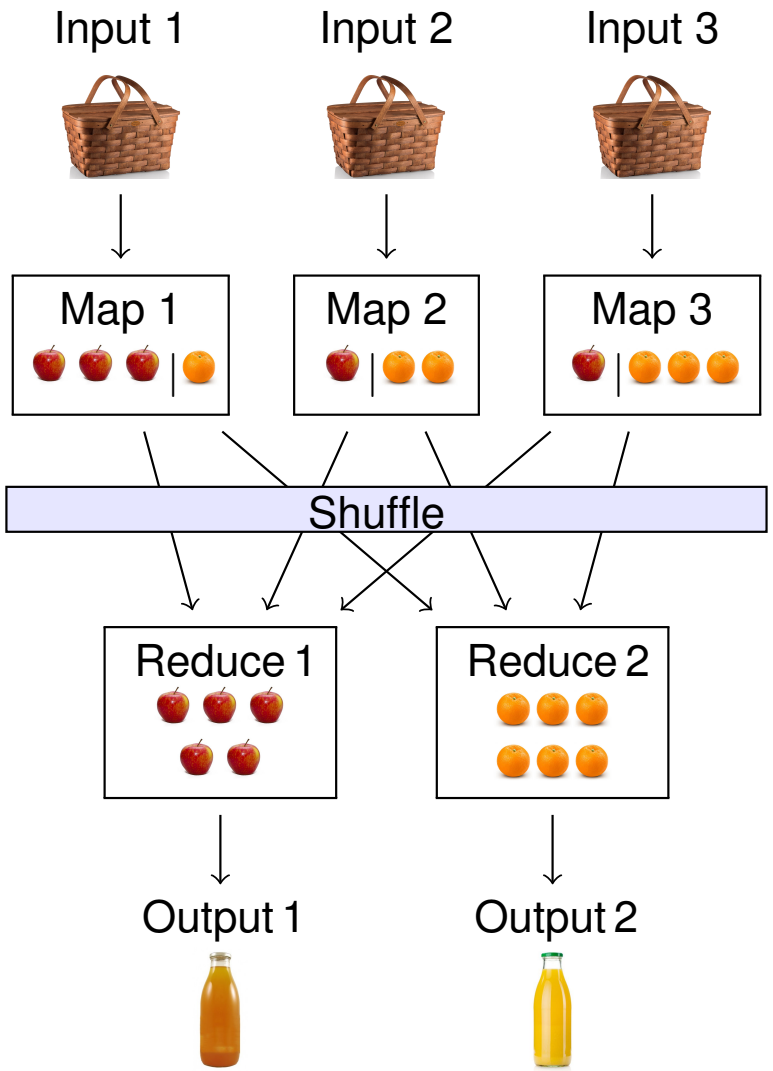
Example of Grouping and Aggregation

Name	Department	Salary
Alice	Computer Science	1900
Bob	Mathematics	1750
Mallory	Computer Science	1800
Oscar	Physics	2000
Carol	Mathematics	1600

Department	COUNT	SUM	AVG	MAX	MIN
Computer Science	2	3700	1850	1900	1800
Mathematics	2	3350	1675	1750	1600
Physics	1	2000	2000	2000	2000

MapReduce

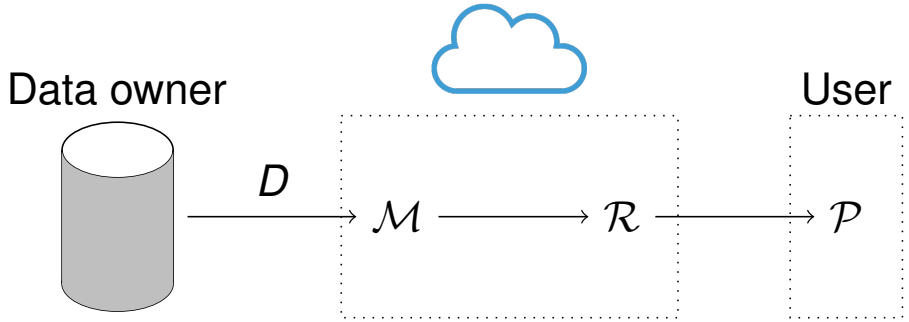
- Partitioning input data
- Scheduling program execution on machines
- Performing the shuffle
- Handling machine failures



Programmer gives:

- Input files
- **Map** and **Reduce**

Grouping and Sum with MapReduce



Name	Department	Salary
Alice	Computer Science	1900
Bob	Mathematics	1750
Mallory	Computer Science	1800
Oscar	Physics	2000
Carol	Mathematics	1600

$\gamma_{Dept,\text{SUM(Salary)}}(D)$

Map:

$\mathcal{M} \to \mathcal{R}$: $\{(\pi_{Dept}(t), \pi_{Salary}(t))\}_{t \in D}$

Department	Salary
Computer Science	1900
Mathematics	1750
Computer Science	1800
Physics	2000
Mathematics	1600

Reduce:

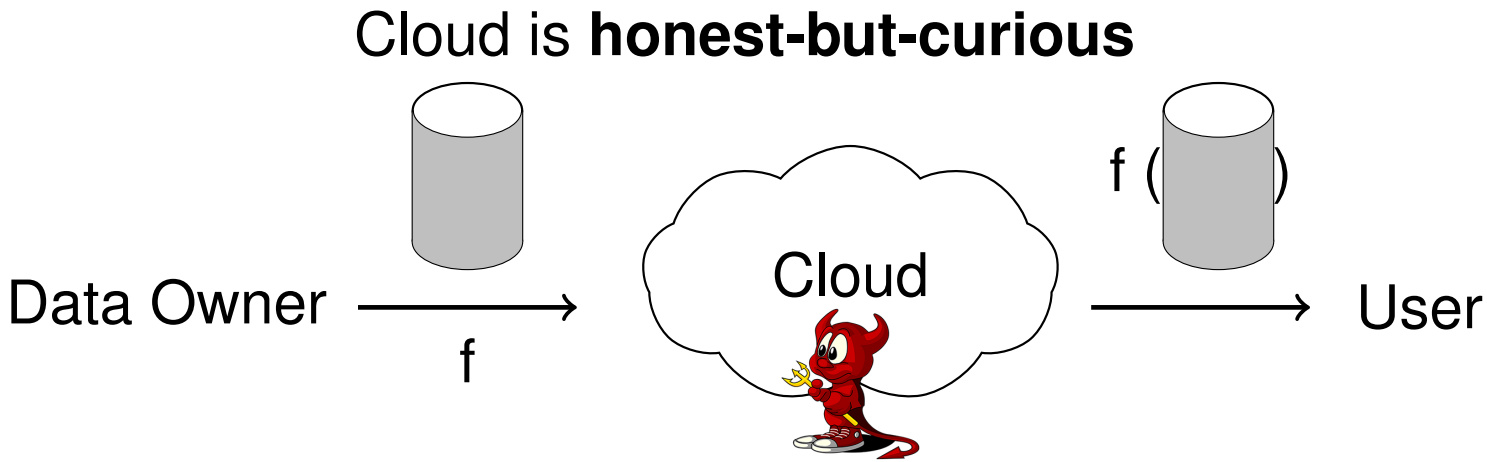
Input: (*key*, *values*)

sum $= \sum_{\pi_{Dept}(t) \in values} \pi_{Salary}(t)$

$\mathcal{R} \to \mathcal{P}$: $(\pi_{Dept}(t), \text{sum})$.

Department	SUM
Computer Science	3700
Mathematics	3350
Physics	2000

Security Model



Security properties

- Secrecy of  and f()
- User queries f() but cannot learn 

Contributions

Secure MapReduce Algorithms:

- COUNT
- SUM
- AVG
- MAX
- MIN

Secure Private Approach

- Cloud nodes do not learn 
- Cloud nodes do not learn f()
- User does not learn 

Outline

Pseudo-Random Permutation

Idea:

$$f : \{0,1\}^n \times \{0,1\}^{n_0} \rightarrow \{0,1\}^{n_1}$$

- Deterministic
- Result indistinguishable from a random
- Not inversible

Notation: Data owner picks a key k and uses $f_k(m)$

Fully Homomorphic Encryption (Gentry 2009)

Idea:

Perform ANY computations on encrypted data

$$\forall f, \forall x_i, f(\mathcal{E}_k(x_1), \ldots, \mathcal{E}_k(x_n)) = \mathcal{E}_k(f(x_1, \ldots, x_n))$$

- Not yet efficient enough

Partial Homomorphic Encryption

Paillier's Cryptosystem (1999)

- Public Key encryption
- Probabilistic encryption
- $\mathcal{E}_{pk}(x + y) = \mathcal{E}_{pk}(x) \cdot \mathcal{E}_{pk}(y)$

$$\mathcal{E}_{pk}(x \cdot y) = (\mathcal{E}_{pk}(x))^y$$

Oder Preserving Encryption

Agrawal et al. Cryptosystem (2004)

Let $c_1 = \mathcal{E}_k(m_1)$ and $c_2 = \mathcal{E}_k(m_2)$

$$\text{if } m_1 < m_2 \text{ then } c_1 < c_2$$

- Symmetric encryption

Outline

COUNT, SUM and AVG

Preprocessing on data

- All data are encrypted with Paillier with pk_U
- All data d have $f_k(d)$

Name	Dept	Salary
A	CS	1900
B	Math	1750
M	CS	1800
O	Phy	2000
C	Math	1600

$\Rightarrow$

Name	Dept	Salary
$f_k(A), \mathcal{E}_{pk_U}(A)$	$f_k(CS), \mathcal{E}_{pk_U}(CS)$	$f_k(1900), \mathcal{E}_{pk_U}(1900)$
$f_k(B), \mathcal{E}_{pk_U}(B)$	$f_k(Math), \mathcal{E}_{pk_U}(Math)$	$f_k(1750), \mathcal{E}_{pk_U}(1750)$
$f_k(M), \mathcal{E}_{pk_U}(M)$	$f_k(CS), \mathcal{E}_{pk_U}(CS)$	$f_k(1800), \mathcal{E}_{pk_U}(1800)$
$f_k(O), \mathcal{E}_{pk_U}(O)$	$f_k(Phy), \mathcal{E}_{pk_U}(Phy)$	$f_k(2000), \mathcal{E}_{pk_U}(2000)$
$f_k(C), \mathcal{E}_{pk_U}(C)$	$f_k(Math), \mathcal{E}_{pk_U}(Math)$	$f_k(1600), \mathcal{E}_{pk_U}(1600)$

Secure Private COUNT

Name	Dept	Salary
$f_k(A)$,$\mathcal{E}_{pk_U}(A)$	$f_k(CS)$, $\mathcal{E}_{pk_U}(CS)$	$f_k(1900)$, $\mathcal{E}_{pk_U}(1900)$
$f_k(B)$,$\mathcal{E}_{pk_U}(B)$	$f_k(Math)$, $\mathcal{E}_{pk_U}(Math)$	$f_k(1750)$, $\mathcal{E}_{pk_U}(1750)$
$f_k(M)$,$\mathcal{E}_{pk_U}(M)$	$f_k(CS)$, $\mathcal{E}_{pk_U}(CS)$	$f_k(1800)$, $\mathcal{E}_{pk_U}(1800)$
$f_k(O)$,$\mathcal{E}_{pk_U}(O)$	$f_k(Phy)$, $\mathcal{E}_{pk_U}(Phy)$	$f_k(2000)$, $\mathcal{E}_{pk_U}(2000)$
$f_k(C)$,$\mathcal{E}_{pk_U}(C)$	$f_k(Math)$, $\mathcal{E}_{pk_U}(Math)$	$f_k(1600)$, $\mathcal{E}_{pk_U}(1600)$

$(f_k(CS), (\mathcal{E}_{pk_U}(CS), \mathcal{E}_{pk_U}(1)))$ $(\mathcal{E}_{pk_U}(CS), \mathcal{E}_{pk_U}(2))$
$(f_k(CS), (\mathcal{E}_{pk_U}(CS)\ \mathcal{E}_{pk_U}(1)))$
$(f_k(Math), (\mathcal{E}_{pk_U}(Math), \mathcal{E}_{pk_U}(1)))$ $(\mathcal{E}_{pk_U}(Math), \mathcal{E}_{pk_U}(2))$
$(f_k(Math), (\mathcal{E}_{pk_U}(Math), \mathcal{E}_{pk_U}(1)))$
$(f_k(Phy), (\mathcal{E}_{pk_U}(Phy), \mathcal{E}_{pk_U}(1)))$ $(\mathcal{E}_{pk_U}(Phy), \mathcal{E}_{pk_U}(1))$

$$\gamma_{A,\text{COUNT}(*)}(D)$$

Map:

$$\mathcal{M} \rightarrow \mathcal{R}\text{: } \left\{(\pi_A f_k(t), (\pi_A \mathcal{E}_{pk_U}(t), \mathcal{E}_{pk_U}(1)))\right\}_{t \in D}$$

Reduce:

$$\text{count} = \mathcal{E}_{pk_U}(\textstyle\sum_{\pi_A f_k(t) \in values} 1) = \prod_{\pi_A f_k(t) \in values} \mathcal{E}_{pk_U}(1)$$

$$\mathcal{R} \rightarrow \mathcal{P} : (\pi_A \mathcal{E}_{pk_U}(t), \text{count}).$$

Secure Private SUM

Name	Dept	Salary
$f_k(A)$,$\mathcal{E}_{pk_U}(A)$	$f_k(CS)$, $\mathcal{E}_{pk_U}(CS)$	$f_k(1900)$, $\mathcal{E}_{pk_U}(1900)$
$f_k(B)$,$\mathcal{E}_{pk_U}(B)$	$f_k(Math)$, $\mathcal{E}_{pk_U}(Math)$	$f_k(1750)$, $\mathcal{E}_{pk_U}(1750)$
$f_k(M)$,$\mathcal{E}_{pk_U}(M)$	$f_k(CS)$, $\mathcal{E}_{pk_U}(CS)$	$f_k(1800)$, $\mathcal{E}_{pk_U}(1800)$
$f_k(O)$,$\mathcal{E}_{pk_U}(O)$	$f_k(Phy)$, $\mathcal{E}_{pk_U}(Phy)$	$f_k(2000)$, $\mathcal{E}_{pk_U}(2000)$
$f_k(C)$,$\mathcal{E}_{pk_U}(C)$	$f_k(Math)$, $\mathcal{E}_{pk_U}(Math)$	$f_k(1600)$, $\mathcal{E}_{pk_U}(1600)$

$(f_k(CS), (\mathcal{E}_{pk_U}(CS), \mathcal{E}_{pk_U}(1900)))$
$(f_k(CS), (\mathcal{E}_{pk_U}(CS)\ \mathcal{E}_{pk_U}(1800)))$
$(f_k(Math), (\mathcal{E}_{pk_U}(Math), \mathcal{E}_{pk_U}(1750)))$
$(f_k(Math), (\mathcal{E}_{pk_U}(Math), \mathcal{E}_{pk_U}(1600)))$
$(f_k(Phy), (\mathcal{E}_{pk_U}(Phy), \mathcal{E}_{pk_U}(2000)))$

$(\mathcal{E}_{pk_U}(CS), \mathcal{E}_{pk_U}(3700))$

$(\mathcal{E}_{pk_U}(Math), \mathcal{E}_{pk_U}(3350))$

$(\mathcal{E}_{pk_U}(Phy), \mathcal{E}_{pk_U}(2000))$

$$\gamma_{A,\mathrm{SUM}(B)}(D)$$

Map:

$$\mathcal{M} \rightarrow \mathcal{R}\text{: } \left\{(\pi_A f_k(t), (\pi_A \mathcal{E}_{pk_U}(t), \pi_B \mathcal{E}_{pk_U}(t)))\right\}_{t \in D}$$

Reduce:

$$\mathrm{sum} = \mathcal{E}_{pk_U}(\textstyle\sum_{\pi_A f_k(t) \in values} \pi_B(t)) = \prod_{\pi_A f_k(t) \in values} \pi_B \mathcal{E}_{pk_U}(t)$$

$$\mathcal{R} \rightarrow \mathcal{P} : (\pi_A \mathcal{E}_{pk_U}(t), \mathrm{sum}).$$

Secure Private AVG

$$\gamma_{A,\mathrm{AVG}(B)}(D)$$

Map:

$\mathcal{M} \rightarrow \mathcal{R}$: $\left\{(\pi_A f_k(t), (\pi_A \mathcal{E}_{pk_U}(t), \pi_B \mathcal{E}_{pk_U}(t), \mathcal{E}_{pk_U}(1)))\right\}_{t \in D}$

Reduce:

$\mathrm{count} = \mathcal{E}_{pk_U}(\sum_{\pi_A f_k(t) \in values} 1) = \prod_{\pi_A f_k(t) \in values} \mathcal{E}_{pk_U}(1)$

$\mathrm{sum} = \mathcal{E}_{pk_U}(\sum_{\pi_A f_k(t) \in values} \pi_B(t)) = \prod_{\pi_A f_k(t) \in values} \pi_B \mathcal{E}_{pk_U}(t)$

$\mathcal{R} \rightarrow \mathcal{P} : (\pi_A \mathcal{E}_{pk_U}(t), (\mathrm{sum}, \mathrm{count}))$.

Outline

MIN & MAX

Preprocessing on data

- All data are encrypted with OPE with a shared key K_{DU}
- And encrypted with the public key of the node pk_C
- All data d have $f_k(d)$

Secure Private MIN

Name	Dept	Salary
$f_k(A)$,$E_{k_{DU}}(A)$	$f_k(CS)$, $E_{k_{DU}}(CS)$	$f_k(1900)$, $E_{k_{DU}}(1900)$
$f_k(B)$,$E_{k_{DU}}(B)$	$f_k(Math)$, $E_{k_{DU}}(Math)$	$f_k(1750)$, $E_{k_{DU}}(1750)$
$f_k(M)$,$E_{k_{DU}}(M)$	$f_k(CS)$, $E_{k_{DU}}(CS)$	$f_k(1800)$, $E_{k_{DU}}(1800)$
$f_k(O)$,$E_{k_{DU}}(O)$	$f_k(Phy)$, $E_{k_{DU}}(Phy)$	$f_k(2000)$, $E_{k_{DU}}(2000)$
$f_k(C)$,$E_{k_{DU}}(C)$	$f_k(Math)$, $E_{k_{DU}}(Math)$	$f_k(1600)$, $E_{k_{DU}}(1600)$

$(f_k(CS), (E_{k_{DU}}(CS), E_{k_{DU}}(1900)))$ $(E_{k_{DU}}(CS), E_{k_{DU}}(1800))$
$(f_k(CS), (E_{k_{DU}}(CS)\ E_{k_{DU}}(1800)))$
$(f_k(Math), (E_{k_{DU}}(Math), E_{k_{DU}}(1750)))$ $(E_{k_{DU}}(Math), E_{K_{DU}}(1600))$
$(f_k(Math), (E_{k_{DU}}(Math), E_{k_{DU}}(1600)))$
$(f_k(Phy), (E_{k_{DU}}(Phy), E_{k_{DU}}(2000)))$ $(E_{k_{DU}}(Phy), E_{k_{DU}}(2000))$

$$\gamma_{A,\mathrm{MIN}(B)}(D)$$

Map:

$\mathcal{M} \rightarrow \mathcal{R}$: $\left\{(\pi_A f_k(t), (\pi_A \mathcal{E}_{pk_U}(t), \pi_B(t)))\right\}_{t \in D}$

Reduce:

$\mathrm{M} = min_{\pi_A f_k(t) \in values} \mathcal{D}_{sk_D} \pi_B(t)$

$\mathcal{R} \rightarrow \mathcal{P} : (\pi_A \mathcal{E}_{pk_U}(t), \mathrm{M})$.

Outline

Security

Theorem

The SP-SUM, SP-COUNT, SP-AVG, SP-MIN, and SP-MAX protocols securely compute the grouping and aggregation in the ROM in the presence of honest-but-curious adversary even if cloud nodes collude.

Combiners and Improvements

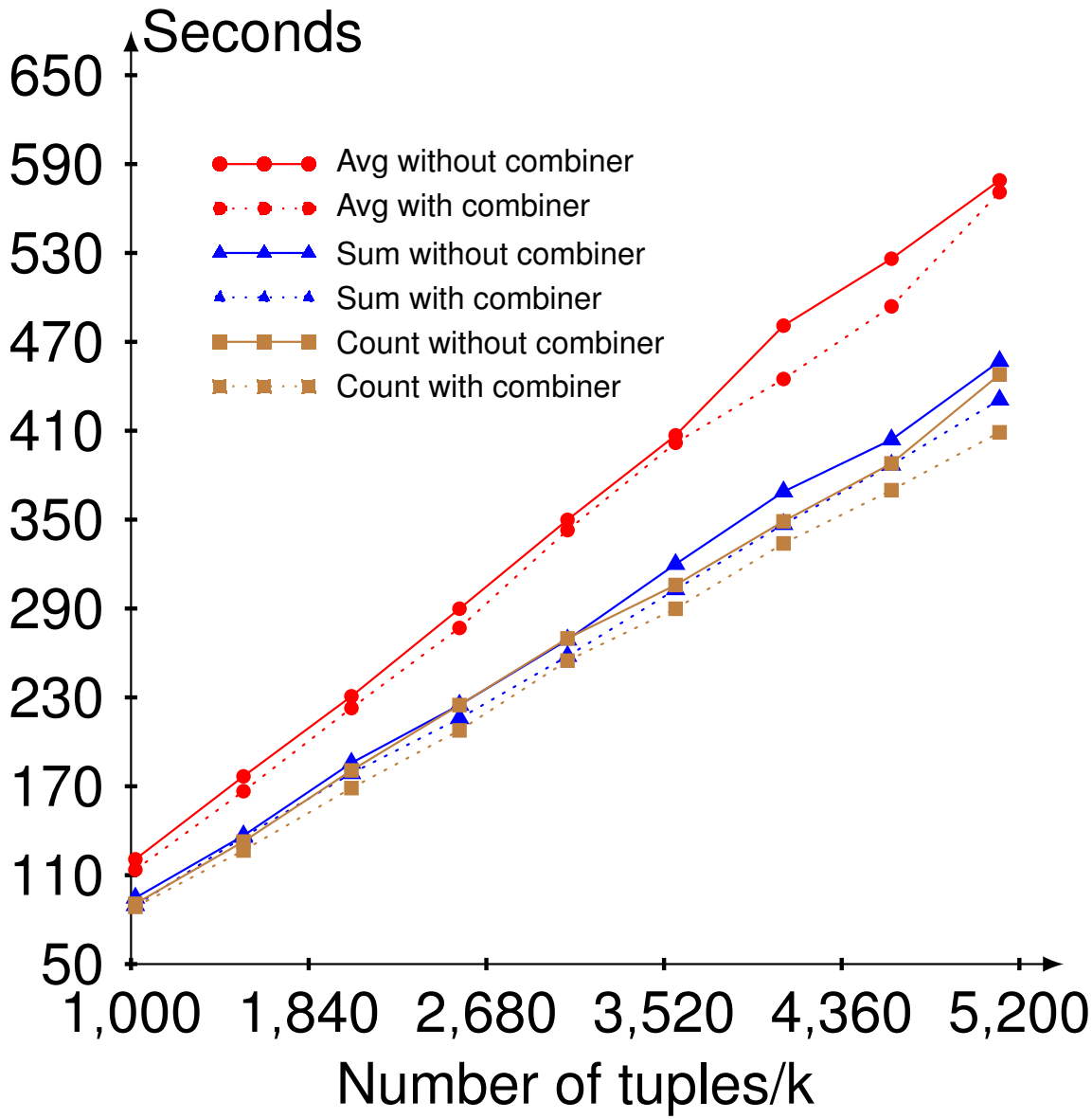
COUNT SUM AVG

- Map can perform some aggregations

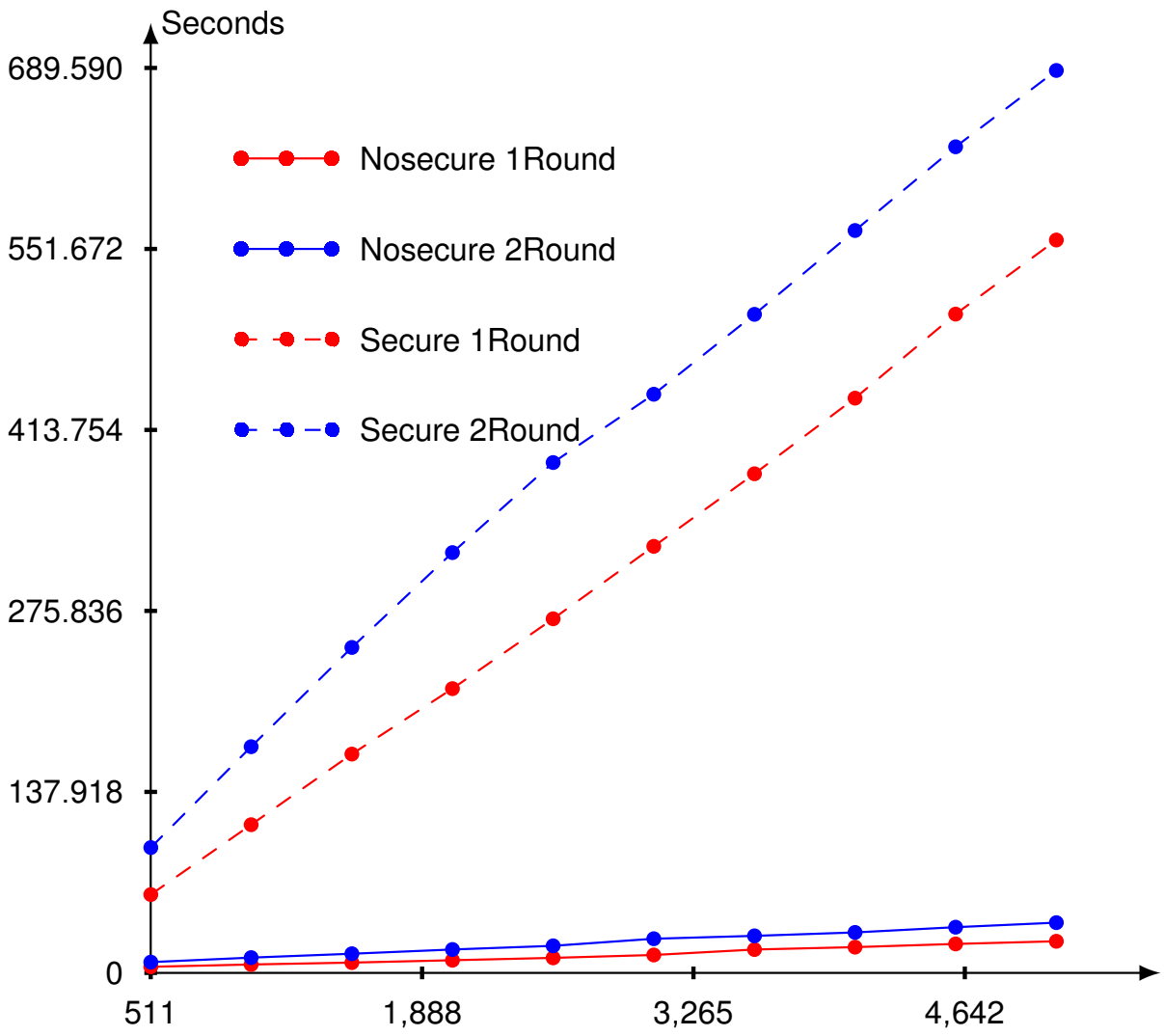
MAX & MIN

We can split it into 2 rounds to counter possible frequency attacks against OPE

Performances of COUNT, SUM & AVG



Performances of MIN



Outline

Conclusion

- Secure-Private MapReduce:
  COUNT, SUM, AVG, & MAX MIN
- Using Paillier and OPE
- Honest-but-curious adversay

Next step

- Combinaisons of COUNT, SUM, AVG, MAX & MIN

Questions?

pascal.lafourcade@uca.fr

LIMOS
LABORATOIRE D'INFORMATIQUE,
DE MODÉLISATION ET D'OPTIMISATION DES SYSTÈMES