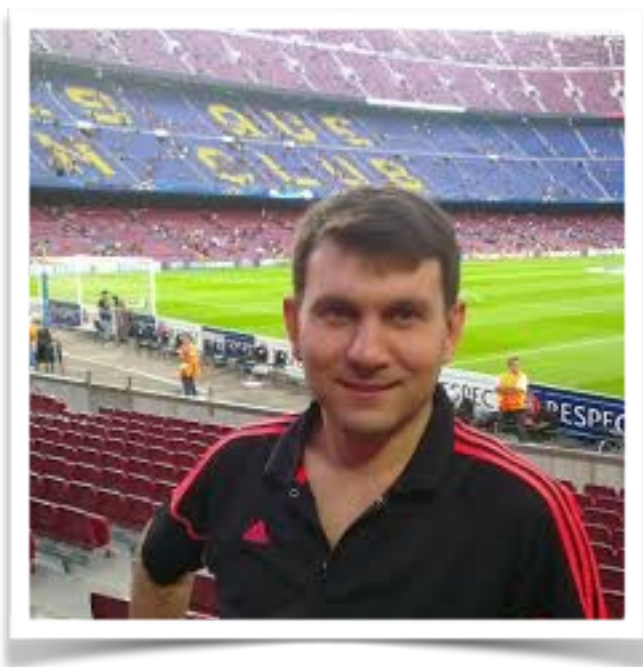


Enabling the Aviation CO₂ Allowance Trading Through Secure Market Mechanisms

Massimiliano Zanin

mz@innaxis.org





Problem:

Computing without trust

Airlines benchmarking

Airport slot trading

CO₂ allowance trading

Data mining on safety data

Safety benchmarking

...

Problem:

SWIM, the solution?

SWIM, information transfer enabler

Based on a public-key infrastructure

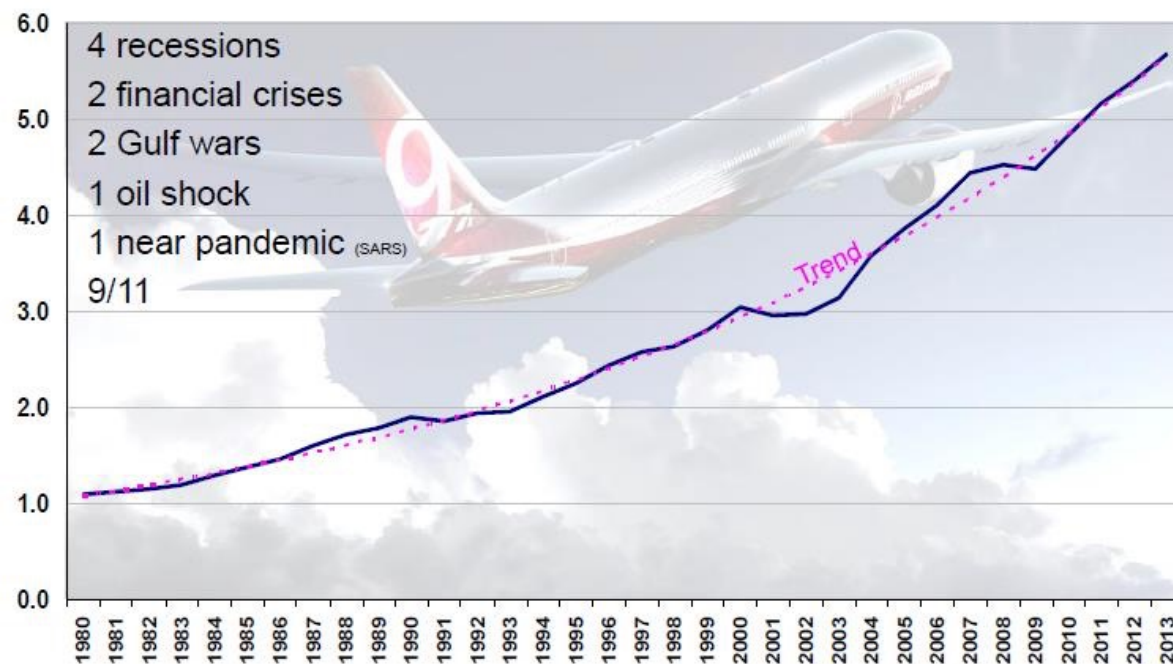
privacy as good as the privacy of the worst procedure
implemented by the entities



Back to the starting point!

Problem:

CO₂ allowance trading



Increasing air traffic
=
increasing CO₂ emissions

Emission trading or *cap and trade*

Problem:

Emission trading or *cap and trade*

Upper limit to the amount of pollutants that can be emitted

Rights to emit, to be traded in a specific market.

More emissions: buy additional rights

Less emissions: rights can be sold in the market

Efficient emissions reduction through a market mechanism, as green companies are receiving indirect incentives.

Problem:

Emission trading or *cap and trade*

CO₂ emissions $\propto$ fuel consumption $\propto$ take-off weight

Fairness of the trading system

Secure Multi-party Computation (SMC)

Subfield of cryptography.

Methods for parties to jointly compute a function over their inputs, while keeping these inputs private.

Andrew C. Yao, 1982:
the millionaire problem



Secure Data Cloud

Objectives:

Introduce SMC into Air Transport
Software Reference Framework
Simulation and analysis of two Case Studies

CO₂ allowance trading

Primary market

Secondary market

Seller:

Industry

An airline

Buyers:

One or more airlines

Seller's input data:

Minimum price

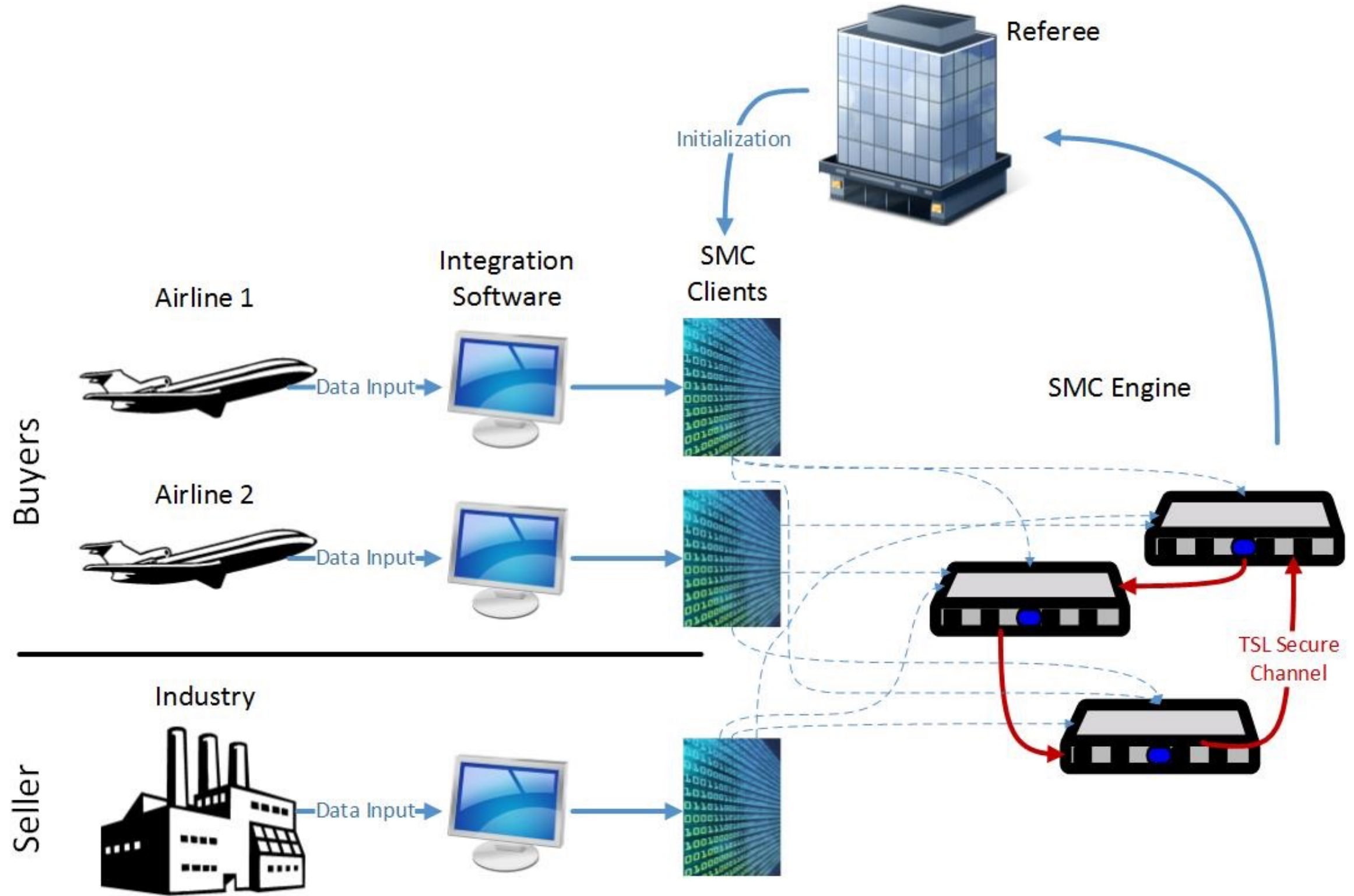
Buyers' input data:

Bid

Result:

Max bid, *iif* bid > minimum price

Management of ties



Secure auction



Secure ranking



Secure evaluation of $a > b$

Secure evaluation of $a > b$

000000047859283759201

000000057483928374627

1. Find the first non-equal digit
2. Compare them for $a > b$

Alice

Secret number:

$[0 \ 0 \ 1]$

Bob

Secret number:

$[0 \ 1 \ 0]$

P_1

Secret number:

a

P_2

Secret number:

b

Alice

Secret number:

[0 0 1]

Shares:

[0] [0] [1]

Bob

Secret number:

[0 1 0]

Shares:

[0] [1] [0]

P_1

Secret number:

a

Shares:

$\{[a_3], [a_2], [a_1]\}$

P_2

Secret number:

b

Shares:

$\{[b_3], [b_2], [b_1]\}$

Alice

Secret number:

[0 0 1]

Shares:

[0] [0] [1]

Sharing the shares:

[0] **[0]** [1]

[0] **[1]** [0]

Bob

Secret number:

[0 1 0]

Shares:

[0] [1] [0]

[0] [1] [0]

[0] [0] [1]

P_1

Secret number:

a

Shares:

$\{[a_3], [a_2], [a_1]\}$

Sharing the shares:

P_2

Secret number:

b

Shares:

$\{[b_3], [b_2], [b_1]\}$

Alice

[0 0 1]

[0] [0] [1]
[0] [1] [0]

Bob

[0 1 0]

[0] [1] [0]
[0] [0] [1]

XOR of every bit:

[1] [1] [0]

P_1

a

$\{[a_3], [a_2], [a_1]\}$

P_2

b

$\{[b_3], [b_2], [b_1]\}$

XOR of every bit:

$c_i = [a_i \oplus b_i]$

Alice

[0 0 1]

[0] [0] [1]
[0] [1] [0]

Bob

[0 1 0]

[0] [1] [0]
[0] [0] [1]

XOR of every bit:

[1] [1] [0]

Prefix-OR:

[1] [1] [0]

P_1

a

P_2

b

$\{[a_3], [a_2], [a_1]\} \quad \{[b_3], [b_2], [b_1]\}$

XOR of every bit:

$c_i = [a_i \oplus b_i]$

Prefix-OR:

$d_i = \bigvee_{(j \text{ from } 3 \text{ to } i)} c_j$

Alice

[0 0 1]

[0] [0] [1]
[0] [1] [0]

Bob

[0 1 0]

[0] [1] [0]
[0] [0] [1]

XOR of every bit:

[1] [1] [0]

Prefix-OR:

[1] [1] [0]

Evolution of d :

[1] [0] [0]

P_1

a

P_2

b

$\{[a_3], [a_2], [a_1]\}$ $\{[b_3], [b_2], [b_1]\}$

XOR of every bit:

$$c_i = [a_i \oplus b_i]$$

Prefix-OR:

$$d_i = \bigvee_{(j \text{ from } 3 \text{ to } i)} c_j$$

Evolution of d :

$$[e_i] = [d_i - d_{i+1}]$$

Alice

[0 0 1]

[0] [0] [1]
[0] [1] [0]

Bob

[0 1 0]

[0] [1] [0]
[0] [0] [1]

P_1

a

P_2

b

$\{[a_3], [a_2], [a_1]\}$ $\{[b_3], [b_2], [b_1]\}$

[1] [1]

[0]

[1] [1]

[0]

[1] [0]

[0]

$$c_i = [a_i \oplus b_i]$$

$$d_i = \bigvee_{(j \text{ from } 3 \text{ to } i)} c_j$$

$$[e_i] = [d_i - d_{i+1}]$$

$a < b$:

$$[0 \ 1 \ 0] = 1$$

$a < b$:

$$\text{Sum} ([e_i] \times [b_i])$$

Alice

[0 0 1]

[0] [0] [1]
[0] [1] [0]

[1] [1]

[1] [1]

[1] [0]

Bob

[0 1 0]

[0] [1] [0]
[0] [0] [1]

[0] [0]

[0] [0]

[0]

P_1

a

P_2

b

$\{[a_3], [a_2], [a_1]\}$ $\{[b_3], [b_2], [b_1]\}$

Detect bits that
are different in
both shares

$$c_i = [a_i \oplus b_i]$$

$$d_i = \bigvee_{(j \text{ from } 3 \text{ to } i)} c_j$$

$$[e_i] = [d_i - d_{i+1}]$$

$a < b$:

$$[0 \ 1 \ 0] = 1$$

$a < b$:

$$\text{Sum} ([e_i] \times [b_i])$$

Alice

[0 0 1]

[0] [0] [1]
[0] [1] [0]

[1] [1]

[1] [1]

[1] [0]

Bob

[0 1 0]

[0] [1] [0]
[0] [0] [1]

[0]

Detect the first bit
that is different in
both shares

$a < b$:

[0 1 0] = 1

P_1

a

$\{[a_3], [a_2], [a_1]\}$

P_2

b

$\{[b_3], [b_2], [b_1]\}$

$$c_i = [a_i \oplus b_i]$$

$$d_i = \bigvee_{(j \text{ from } 3 \text{ to } i)} c_j$$

$$[e_i] = [d_i - d_{i+1}]$$

$a < b$:

Sum ($[e_i] \times [b_i]$)

Alice

[0 0 1]

[0] [0] [1]
[0] [1] [0]

Bob

[0 1 0]

[0] [1] [0]
[0] [0] [1]

P_1

a

P_2

b

$\{[a_3], [a_2], [a_1]\} \quad \{[b_3], [b_2], [b_1]\}$

[1] [1]

[0]

[1] [1]

[0]

[1] [0]

[0]

$$c_i = [a_i \oplus b_i]$$

$$d_i = \bigvee_{(j \text{ from } 3 \text{ to } i)} c_j$$

$$[e_i] = [d_i - d_{i+1}]$$

$a < b$:

[0 1 0]

Check if b has a
1 in that position

$a < b$:

Sum ($[e_i] \times [b_i]$)

No drawbacks?

Of course they are...

High computational cost

Lots of shares should be exchanged between the parties

Cost of encrypting and decrypting information

Visit our demonstrator!

Analysing delay reports

An example of Secure Multi-party Computation in ATM

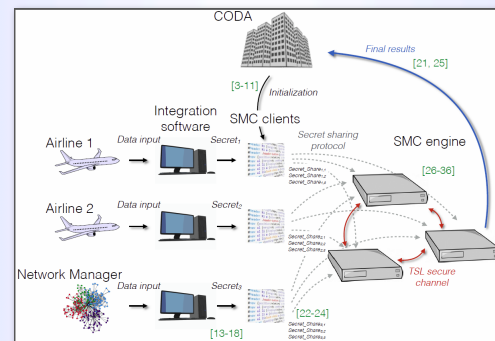
By the SecureDataCloud team: www.innaxis.org/securedatacloud

The scenario:

Creation of delay reports using *cleared* information coming from different stakeholders, securely merged in order to achieve additional knowledge about causes of delays and their evolution through time. Several stakeholders collaborate by introducing delay information inside the system: pilot, airline representatives, ATC officers, EUROCONTROL's Network Manager Unit, airport representatives, and handling organisations. Different types of analyses will be performed, including average delays estimation, averages by causes, and airlines ranking.



The secure computation:



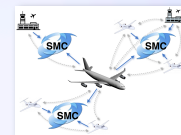
Secure Slot Trading

An example of Secure Multi-party Computation in ATM

By the SecureDataCloud team: www.innaxis.org/securedatacloud

The scenario:

We consider an airline planning to operate a new route between two airports. Therefore, at a strategic level, the airline firstly tries to buy slots from both airports, *i.e.* in the primary market. Afterwards, if this first step was not successful, it may try to buy a suitable slot in the secondary market, *i.e.* from other airlines. These two different operations are depicted in the picture: in the upper left part, the airline enters an auction in the primary market; conversely, in the lower left part, it tries to buy the asset from other airlines.



The secure computation:

