Multi-User Encrypted Data Processing

Amani ABOU RIDA[1], Maxime BROS[2], Olivier GIMENEZ[3], Héber HWANG ARCOLEZI[4], Quentin YANG[5]

CIRM, Marseille, France

25-29 October 2021

Sébastien Canard **Jérémy Chotard**

[1]University of Strasbourg, CNRS, ICube, [2]University of Limoges, XLIM, [3]IRISA Laboratory, Rennes and Orange Labs, [4]University of Bourgogne-Franche-Comté, CNRS, [5]University of Lorraine, Inria Nancy-Grand Est, LORIA.

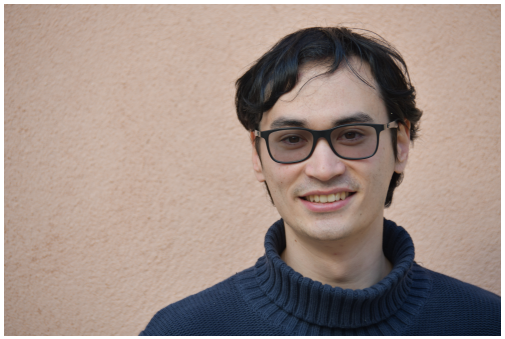
Our Team

Amani **ABOU RIDA**

Maxime **BROS**

Olivier **GIMENEZ**

Héber **HWANG ARCOLEZI**



Quentin **YANG**

Multi-User Encrypted Data

Users

Pollster

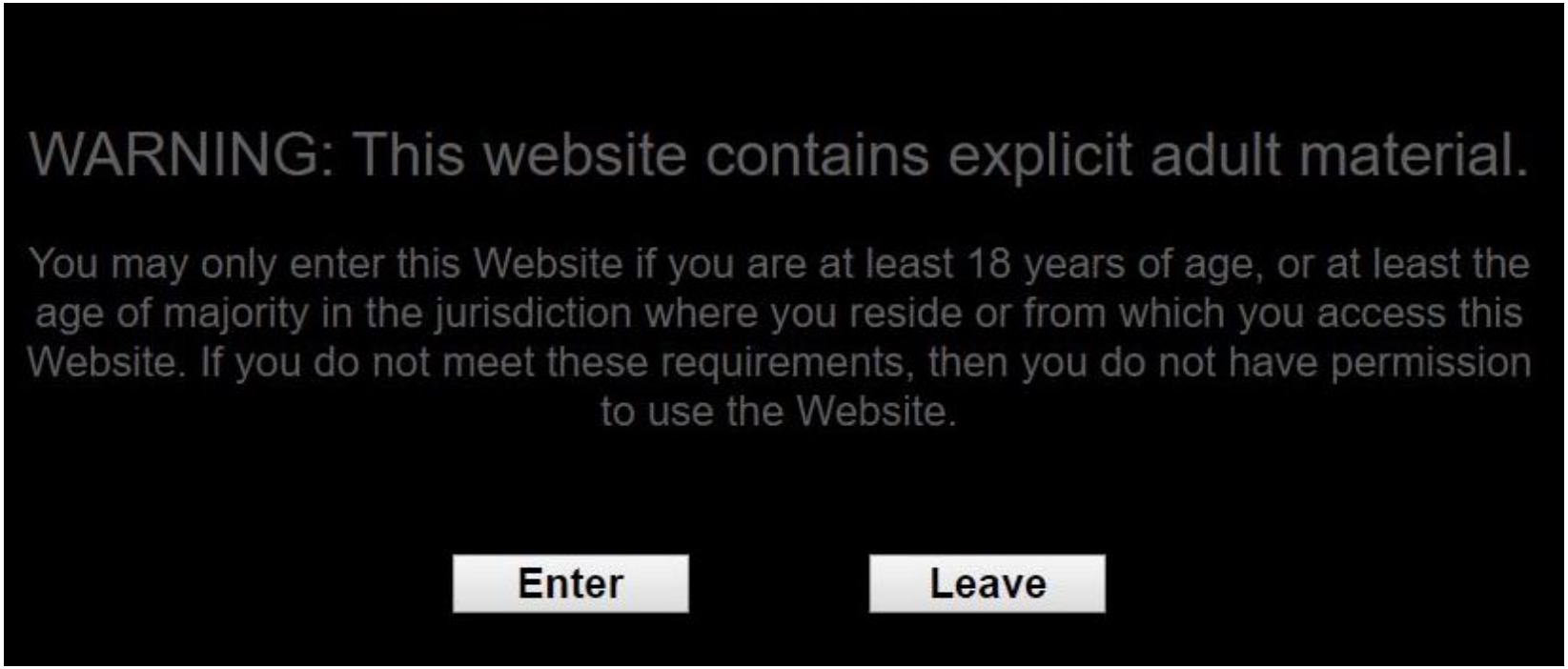
Orange

Data $\{x_i\}_i$ $\longrightarrow$ Heavy computations $\longrightarrow$ Get statistics

Goals

- Efficiency
- Privacy
- User-friendly
- Few actions from pollster

Data can be sensitive...



Warning message of www.youporn.com

Attacker Models

- Full Trust
- Semi-Honest
- Mallicious with corruption of a reasonable amount of users
- A Coalition of mallicious with corruption

Considered Privacy

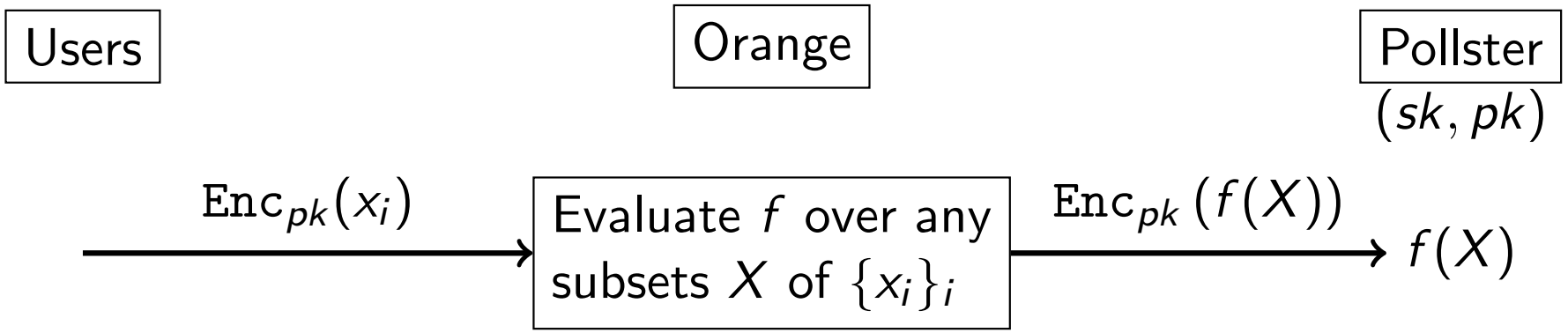
- **Partial Privacy:** Orange can compute f over a subset of users
- **Ideal Privacy:** Orange can only compute f over all users

Functions

- Compute the Mean
- Compute the Weighted Mean
- Compute the Linear Regression
- Compute Any Given Function

Imprecision of Partial Privacy

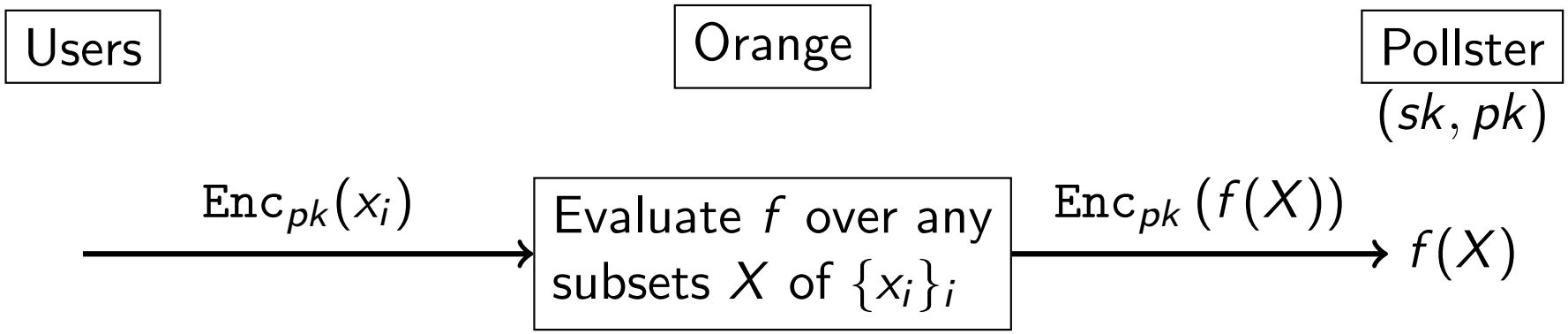
Let f be the mean function:



Partial Privacy $\Longrightarrow$ Ideal Privacy

Imprecision of Partial Privacy

Let f be the mean function:



Problem

$$\forall i, \quad X = \{x_i\}, \quad f(X) = x_i.$$

Partial Privacy $\Longrightarrow$ Ideal Privacy

Outline

Homomorphic Encryption

Definition (Homomorphic property)

$$\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x) \cdot \mathtt{Enc}_{pk}(y)) = x * y.$$

Example: Exponential ElGamal (additive)

$pk = (g, h = g^s)$ $\quad$ $sk = s$

$\mathtt{Enc}_{pk}(x) = (g^r, g^x h^r)$ $\quad$ $\mathtt{Dec}_{sk}((u, v)) = vu^{-s}$

$$\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x)\mathtt{Enc}_{pk}(y)) = g^{x+y}$$

Zero Knowledge Proof (ZKP)

Used to prove correct computation.

- Proof of knowledge
- Zero Knowledge
- Non-Interactive

Generalization: a NP language $\mathcal{L}$ with relation $\mathcal{R} \subset \mathcal{W} \times \mathcal{X}$.

Outline

Dummy Scheme

$\boxed{\text{Users}}$ $\boxed{\text{Orange}}$ $\boxed{\text{Pollster}}$

sk

$$\xleftarrow{\quad pk \quad}$$

$$c_i = \mathtt{Enc}_{pk}(x_i) \xrightarrow{(c_i)_i}$$

$$M = \prod_i c_i^{w_i} \xrightarrow{M}$$

$$\mu = \mathtt{Dec}_{sk}(M)$$

$$\text{result} = \mu$$

Dummy Scheme

$\boxed{\text{Users}}$ $\boxed{\text{Orange}}$ $\boxed{\text{Pollster}}$

pk sk

$$\xleftarrow{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}$$

$$c_i = \mathtt{Enc}_{pk}(x_i) \xrightarrow{(c_i)_i}$$

$$M = \prod_i c_i^{w_i} \xrightarrow{M}$$

$$\mu = \mathtt{Dec}_{sk}(M)$$

$$\text{result} = \mu$$

Full Trust, Ideal Privacy, No Attribute, (Weighted) Mean

Outline

Secret Sharing (SS)

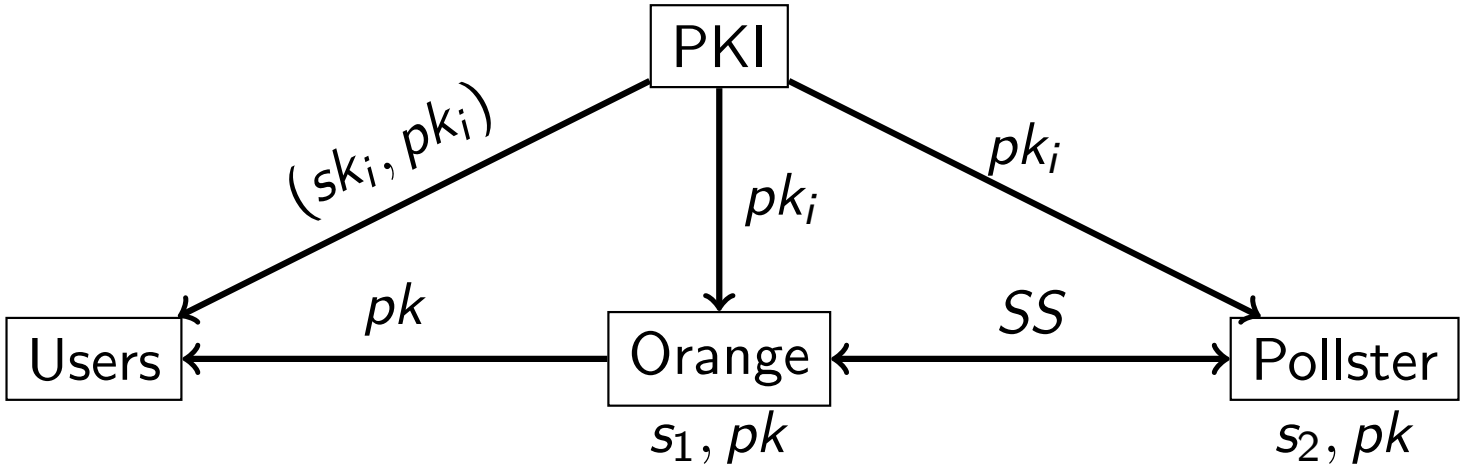
Orange		Pollster
$s_1 \xleftarrow{\$} \mathbb{Z}_p$		$s_2 \xleftarrow{\$} \mathbb{Z}_p$
$S_1 = g^{s_1}$		$S_2 = g^{s_2}$
	$\xrightarrow{H(S_1)}$	
	$\xleftarrow{H(S_2)}$	
	$\xrightarrow{S_1}$	
	$\xleftarrow{S_2}$	
$pk = S_1 S_2$	Shared Public Key $pk = g^{s_1+s_2}$	$pk = S_1 S_2$

Shared Secret: $s_1 + s_2$

Decrypting $\mathtt{Enc}_{pk}(m)$ requires 2 **Partial Decryptions** ($\mathtt{PDec}$):

$$\mathtt{PDec}_{s_2}(\mathtt{PDec}_{s_1}(\mathtt{Enc}_{pk}(m))) \quad = \quad \mathtt{PDec}_{s_1}(\mathtt{PDec}_{s_2}(\mathtt{Enc}_{pk}(m))) \quad = \quad m$$

Key Distribution



No Trust Scheme (Step 1)

Users	Orange	Pollster
pk_i, sk_i	pk, s_1	pk, s_2

$\sigma_i = Sign_{sk_i}(c_i)$ $\xrightarrow{(c_i, \sigma_i)}$

$M = \prod c_i^{w_i}$

$\widetilde{m} = \mathtt{PDec}_{s_1}(M)$ $\xrightarrow{(c_i, \sigma_i)_i, M, \widetilde{m}}$

$M \stackrel{?}{=} \prod c_i^{w_i}$

$Verify_{pk_i}(\sigma_i, c_i)$

$\mu = \mathtt{PDec}_{s_2}(\widetilde{m})$

Attack

Users	Orange	Pollster
pk_i, sk_i	pk, s_1	pk, s_2

$$\xrightarrow{(c_i, \sigma_i)}$$

$$M = \prod c_i^{w_i}$$
$$\widetilde{m} = \mathtt{PDec}_{s_1}(c_1) \xrightarrow{(c_i, \sigma_i)_i, M, \widetilde{m}}$$

$$M = \prod c_i^{w_i}$$
$$Verify_{pk_i}(\sigma_i, c_i)$$
$$x_1 = \mathtt{PDec}_{s_2}(\widetilde{m})$$

No Trust Scheme (Step 2)

Users	Orange	Pollster
pk_i, sk_i	pk, s_1	pk, s_2

$$M = \prod c_i^{w_i}$$
$$\widetilde{m} = \mathtt{PDec}_{s_1}(M)$$
$$\pi = ZKP(\widetilde{m} = \mathtt{PDec}_{s_1}(M)) \xrightarrow{(c_i, \sigma_i)_i, M, \widetilde{m}, \pi}$$

$$M \stackrel{?}{=} \prod c_i^{w_i}$$
$$Verify_{pk_i}(\sigma_i, c_i)$$
$$\widetilde{m} \stackrel{?}{=} \mathtt{PDec}_{s_1}(M) \text{ with } \pi$$
$$\mu = \mathtt{PDec}_{s_2}(\widetilde{m})$$

No Trust Scheme (Step 2)

Users	Orange	Pollster
pk_i, sk_i	pk, s_1	pk, s_2

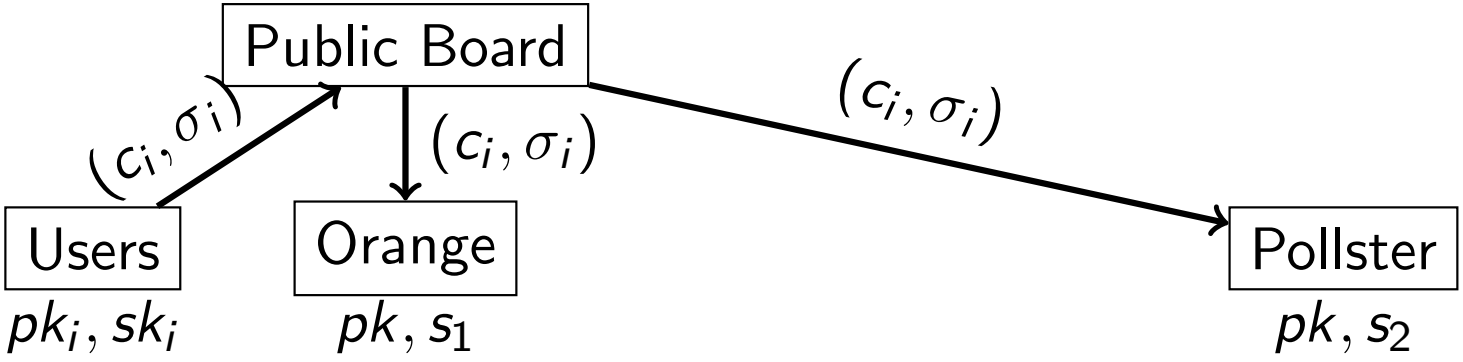
$$M = \prod c_i^{w_i}$$
$$\widetilde{m} = \mathtt{PDec}_{s_1}(M)$$
$$\pi = ZKP(\widetilde{m} = \mathtt{PDec}_{s_1}(M)) \xrightarrow{(c_i, \sigma_i)_i, M, \widetilde{m}, \pi}$$

$$M \stackrel{?}{=} \prod c_i^{w_i}$$
$$Verify_{pk_i}(\sigma_i, c_i)$$
$$\widetilde{m} \stackrel{?}{=} \mathtt{PDec}_{s_1}(M) \text{ with } \pi$$
$$\mu = \mathtt{PDec}_{s_2}(\widetilde{m})$$

Semi-Honest, Ideal Privacy, No Attribute, (Weighted) Mean

Outline

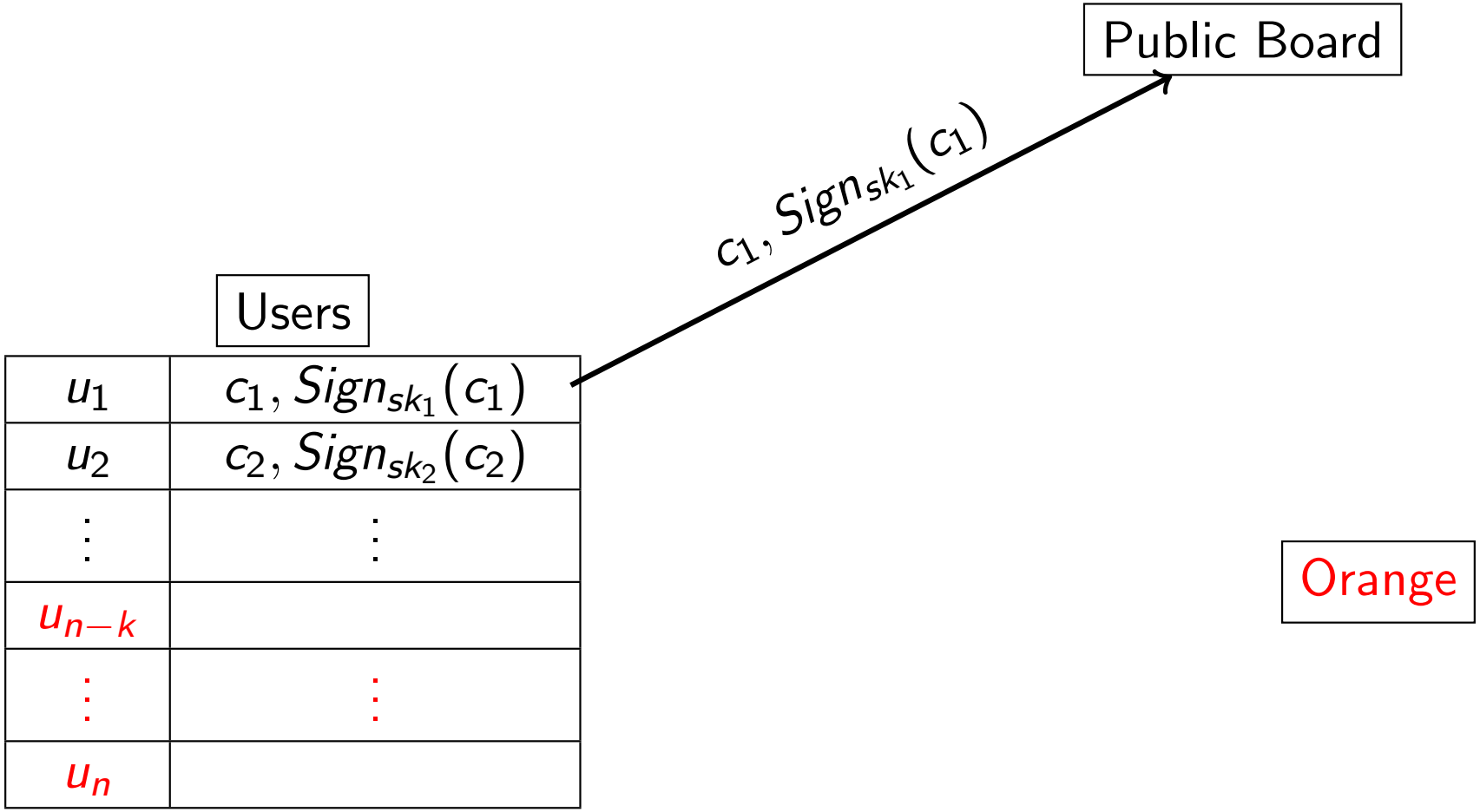
No Drop Scheme



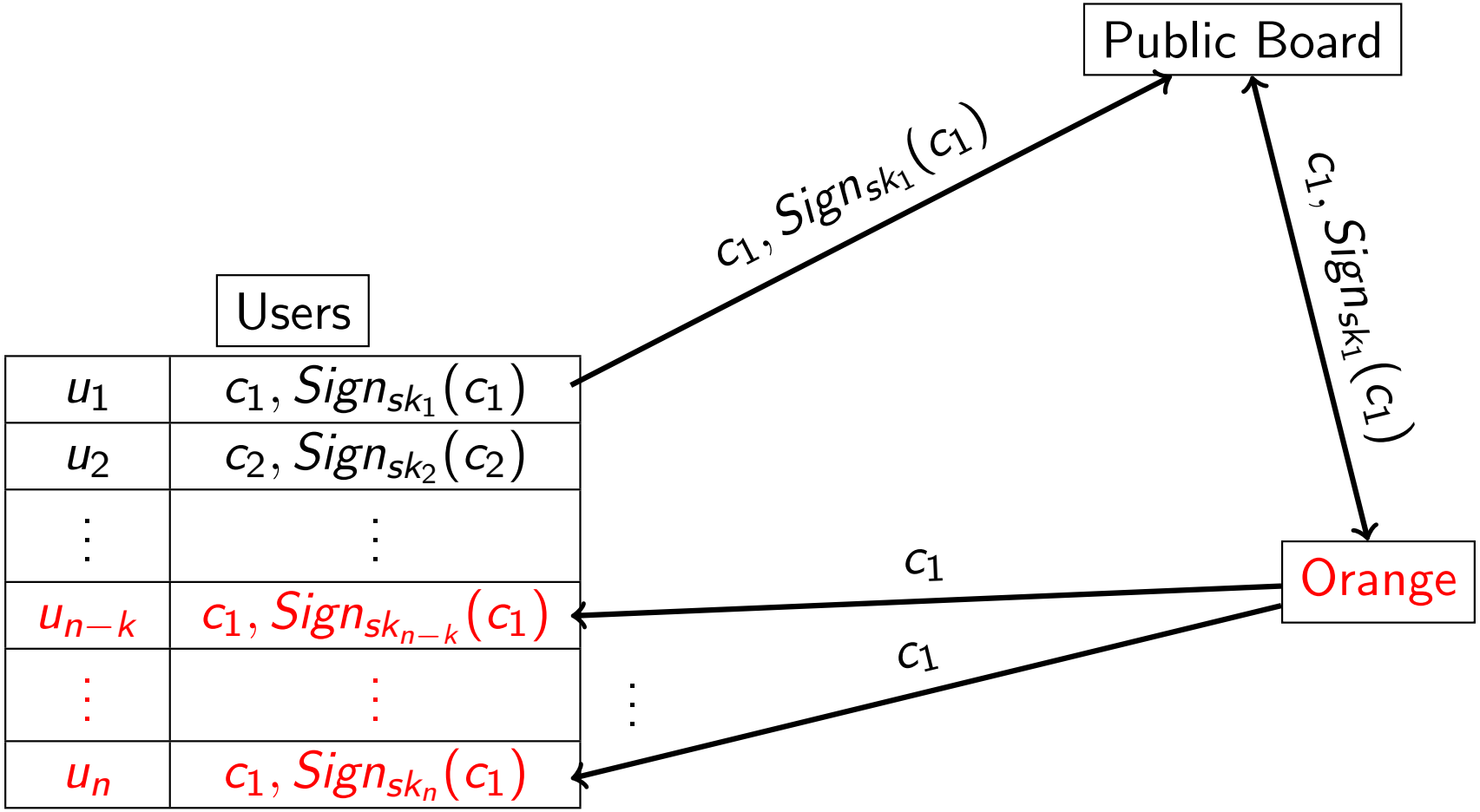
$$M = \prod c_i^{w_i}$$
$$\widetilde{m} = \mathtt{PDec}_{s_1}(M)$$
$$\pi = ZKP(\widetilde{m} = \mathtt{PDec}_{s_1}(M)) \xrightarrow{(c_i, \sigma_i)_i, M, \widetilde{m}, \pi}$$

$$M \stackrel{?}{=} \prod c_i^{w_i}$$
$$Verify_{pk_i}(\sigma_i, c_i)$$
$$\widetilde{m} \stackrel{?}{=} \mathtt{PDec}_{s_1}(M) \text{ with } \pi$$
$$\mu = \mathtt{PDec}_{s_2}(\widetilde{m})$$

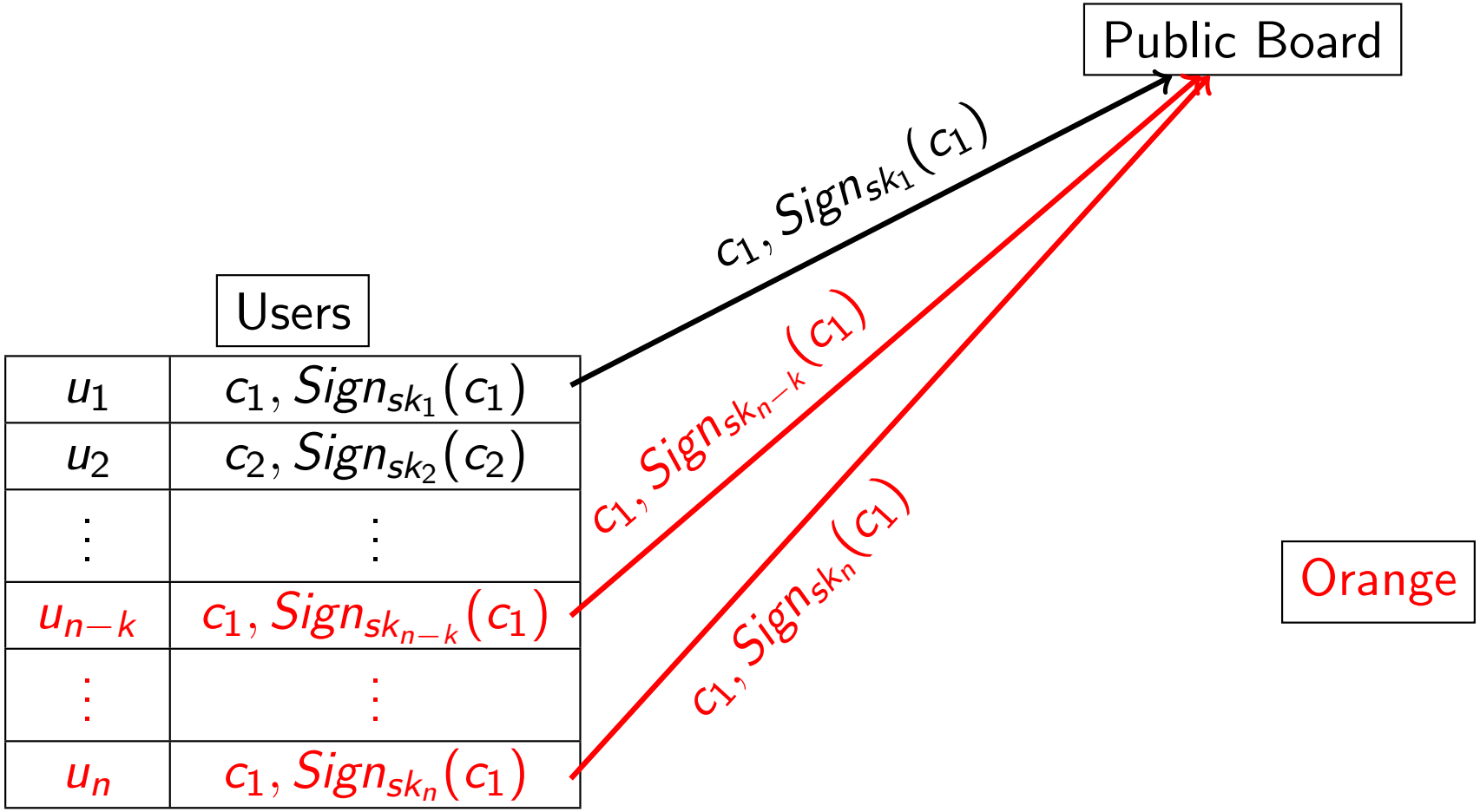
Attack


Public Board
$c_1, Sign_{sk_1}(c_1)$
Users
u_1 | $c_1, Sign_{sk_1}(c_1)$
u_2 | $c_2, Sign_{sk_2}(c_2)$
u_{n-k}
u_n
Orange

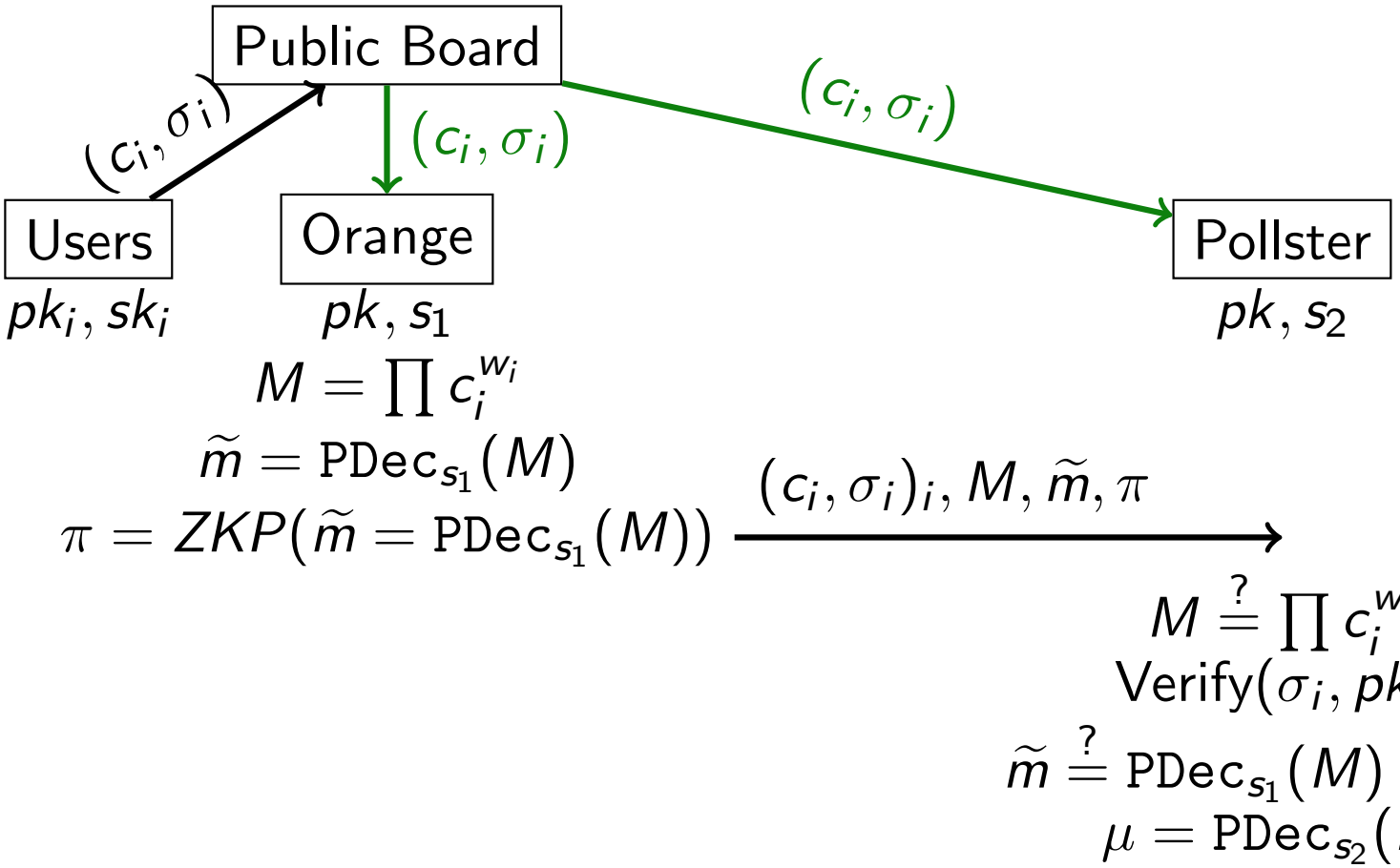
Attack


Public Board
$c_1, Sign_{sk_1}(c_1)$
$c_1, Sign_{sk_1}(c_1)$
Users
u_1 $c_1, Sign_{sk_1}(c_1)$
u_2 $c_2, Sign_{sk_2}(c_2)$
u_{n-k} $c_1, Sign_{sk_{n-k}}(c_1)$
u_n $c_1, Sign_{sk_n}(c_1)$
c_1
c_1
Orange

Attack


Public Board
Users
u_1 | $c_1, Sign_{sk_1}(c_1)$
u_2 | $c_2, Sign_{sk_2}(c_2)$
$\vdots$ | $\vdots$
u_{n-k} | $c_1, Sign_{sk_{n-k}}(c_1)$
$\vdots$ | $\vdots$
u_n | $c_1, Sign_{sk_n}(c_1)$
$c_1, Sign_{sk_1}(c_1)$
$c_1, Sign_{sk_{n-k}}(c_1)$
$c_1, Sign_{sk_n}(c_1)$
Orange

No Copy Scheme



The public board waits for a **deadline** before publishing the results.

Malicious and Corruption, Ideal Privacy, No Attribute, (Weighted) Mean

Outline

Functional Encryption

Definition (Multi-Input Functional Encryption)

$$\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x_1), \cdots, \mathtt{Enc}_{pk}(x_n)) = f(x_1, \cdots, x_n).$$

Example: AGT's scheme

Agrawal, Goyal, Tomida, **Multi-input Quadratic Functional Encryption from Pairings.** CRYPTO 2021.

Fully Homomorphic Encryption

Definition (Fully Homomorphic property)

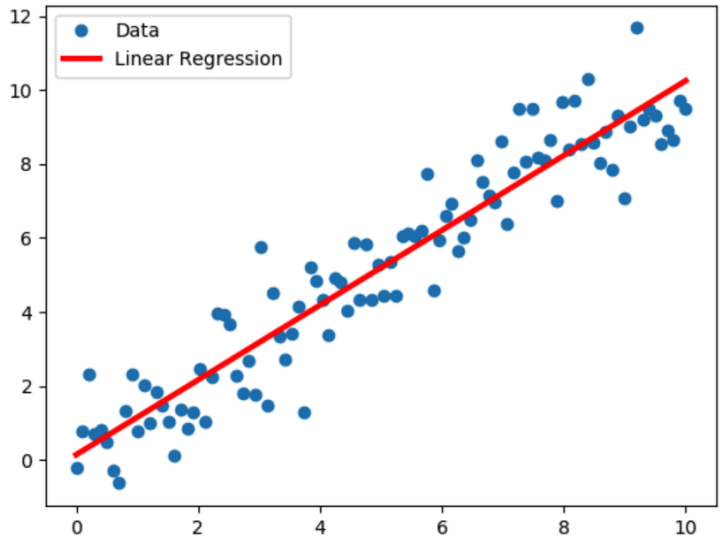
$$\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x) + \mathtt{Enc}_{pk}(y)) = x + y,$$
$$\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x)\mathtt{Enc}_{pk}(y)) = xy.$$

Example:

Boneh, Gennaro, Goldfeder, Jain, Kim, Rasmussen, Sahai, **Threshold Cryptosystems from Threshold Fully Homomorphic Encryption,** CRYPTO 2018.

Outline

What about Linear Regression?



Data $(x_i, y_i)_i$ from users, $\overline{x} = \mathtt{mean}(\{x_i\}_i)$, $\overline{y} = \mathtt{mean}(\{y_i\}_i)$.

Find coefficients a, b s.t. $y \approx ax + b$.

$$a = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}, \qquad b = \overline{y} - a\overline{x}.$$

What about Linear Regression?

Idea 1: Use Functional Encryption

Wish List for Christmas: An Encryption scheme s.t.

$$\begin{aligned}&\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x_1), \mathtt{Enc}_{pk}(y_1), \cdots, \mathtt{Enc}_{pk}(x_n), \mathtt{Enc}_{pk}(y_n))\\ &= \left(\frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}, \overline{y} - a\overline{x}\right)\end{aligned}$$

What about Linear Regression?

Idea 1: Use Functional Encryption

Wish List for Christmas: An Encryption scheme s.t.

$$\begin{aligned}&\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x_1), \mathtt{Enc}_{pk}(y_1), \cdots, \mathtt{Enc}_{pk}(x_n), \mathtt{Enc}_{pk}(y_n))\\ &= \left(\frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}, \overline{y} - a\overline{x}\right)\end{aligned}$$

Problem:

We could not find Santa.

What about Linear Regression?

Recall: Slope of the Linear Regression $= \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}$

Idea 2: Use Homomorphic Functional Encryption for inner product

Wish List for Christmas: A **homomorphic** Encryption scheme such that

$$\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x_1), \mathtt{Enc}_{pk}(y_1), \cdots, \mathtt{Enc}_{pk}(x_n), \mathtt{Enc}_{pk}(y_n)) = \sum_i x_i y_i$$

What about Linear Regression?

Recall: Slope of the Linear Regression $= \dfrac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}$

Idea 2: Use Homomorphic Functional Encryption for inner product

Wish List for Christmas: A **homomorphic** Encryption scheme such that

$$\mathtt{Dec}_{sk}(\mathtt{Enc}_{pk}(x_1), \mathtt{Enc}_{pk}(y_1), \cdots, \mathtt{Enc}_{pk}(x_n), \mathtt{Enc}_{pk}(y_n)) = \sum_i x_i y_i$$

- $\forall\ i$,
  $\mathtt{Enc}_{pk}(x_i)\mathtt{Enc}_{pk}(\overline{x})^{-1} = \mathtt{Enc}_{pk}(x_i - \overline{x})$,
  $\mathtt{Enc}_{pk}(y_i)\mathtt{Enc}_{pk}(\overline{y})^{-1} = \mathtt{Enc}_{pk}(y_i - \overline{y})$,
- Compute $N = \sum_i (x_i - \overline{x})(y_i - \overline{y})$ using $\mathtt{Dec}_{sk}$,
- Compute $D = \sum_i (x_i - \overline{x})^2$ using $\mathtt{Dec}_{sk}$.

What about Linear Regression?

- $\forall\ i$,
  $\mathtt{Enc}_{pk}(x_i)\mathtt{Enc}_{pk}(\overline{x})^{-1} = \mathtt{Enc}_{pk}(x_i - \overline{x})$,
  $\mathtt{Enc}_{pk}(y_i)\mathtt{Enc}_{pk}(\overline{y})^{-1} = \mathtt{Enc}_{pk}(y_i - \overline{y})$,
- Compute $N = \sum_i (x_i - \overline{x})(y_i - \overline{y})$ using $\mathtt{Dec}_{sk}$,
- Compute $D = \sum_i (x_i - \overline{x})^2$ using $\mathtt{Dec}_{sk}$.

Problem:

It leaks extra data.

What about Linear Regression?

- $\forall\ i$,
  $\mathtt{Enc}_{pk}(x_i)\mathtt{Enc}_{pk}(\overline{x})^{-1} = \mathtt{Enc}_{pk}(x_i - \overline{x})$,
  $\mathtt{Enc}_{pk}(y_i)\mathtt{Enc}_{pk}(\overline{y})^{-1} = \mathtt{Enc}_{pk}(y_i - \overline{y})$,
- Compute $N = \sum_i (x_i - \overline{x})(y_i - \overline{y})$ using $\mathtt{Dec}_{sk}$,
- Compute $D = \sum_i (x_i - \overline{x})^2$ using $\mathtt{Dec}_{sk}$.

Problem:

It leaks extra data.

A solution? Use random masking.

Reveal αN and αD for some random α.

What about Linear Regression?

Recall: $\frac{N}{D} = \frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2}$ from $\begin{array}{l} X_i = \mathtt{Enc}_{pk}(x_i - \overline{x}), \\ Y_i = \mathtt{Enc}_{pk}(y_i - \overline{y}). \end{array}$

- Choose random α.
- Compute $X_i' = X_i^{\alpha} = \mathtt{Enc}_{pk}(\alpha(x_i - \overline{x}))$,
- Evaluate $\mathtt{Dec}_{sk}((X_i', Y_i)_{1 \leq i \leq n}) = \alpha N$.
- Evaluate $\mathtt{Dec}_{sk}((X_i', X_i)_{1 \leq i \leq n}) = \alpha D$.

What about Linear Regression?

Summary of Idea 2 (Homomorphic Functional Encryption for inner product)

- $\mathtt{Enc}_{pk}(x)\mathtt{Enc}_{pk}(y) = \mathtt{Enc}_{pk}(x+y)$,
- $\mathtt{Dec}_{sk}((\mathtt{Enc}_{pk}(x_i), \mathtt{Enc}_{pk}(y_i)) = \sum_i x_i y_i$,
- Zero Knowledge Range Proof for α,
- Zero Knowledge Proof for the X_i'.

Problems

- α cannot be too large.
- Verifying zero knowledge proofs (for α and X_i') requires heavy computations.
- So far, no scheme matches our specific needs.

Let's do some basic maths!

The sums are done over n terms:

$$\begin{aligned}\frac{\sum_i (x_i - \overline{x})(y_i - \overline{y})}{\sum_i (x_i - \overline{x})^2} &= \frac{n^2 \sum_i (x_i - \overline{x})(y_i - \overline{y})}{n^2 \sum_i (x_i - \overline{x})^2} \\ &= \frac{\sum_i (nx_i - n\overline{x})(ny_i - n\overline{y})}{\sum_i (nx_i - n\overline{x})^2} \\ &= \frac{\sum_i (nx_i - \sum_i x_i)(ny_i - \sum_i y_i)}{\sum_i (nx_i - \sum_i x_i)^2}\end{aligned}$$

Our trick

We got rid of the integer divisions in the numerator and in the denominator!

What about Linear Regression?

Idea 3: Use Fully Homomorphic Encryption

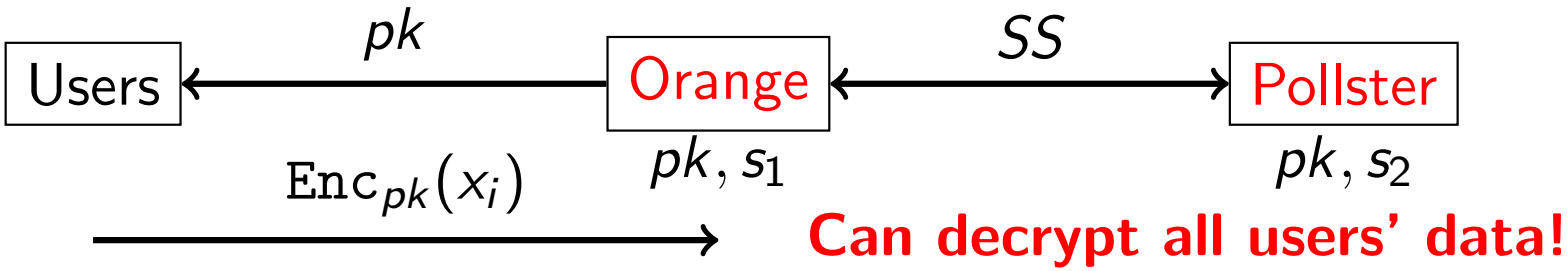
- $\mathtt{Dec}_{sk}((\mathtt{Enc}_{pk}(x)\mathtt{Enc}_{pk}(y)) = xy$,
- $\mathtt{Dec}_{sk}((\mathtt{Enc}_{pk}(x) + \mathtt{Enc}_{pk}(y)) = x + y$,
- Zero Knowledge Range Proof for α,
- Zero Knowledge Proof for the X_i'

Problem

- Usually FHE not efficient at all... **but not with our trick!**
- The proofs are hard to verify.

Outline

Impact of a coalition



Protecting users against a coalition

Any idea?

1. Encrypt with own secret key.
2. Users participate in the Distributed Key Generation.

Protecting users against a coalition

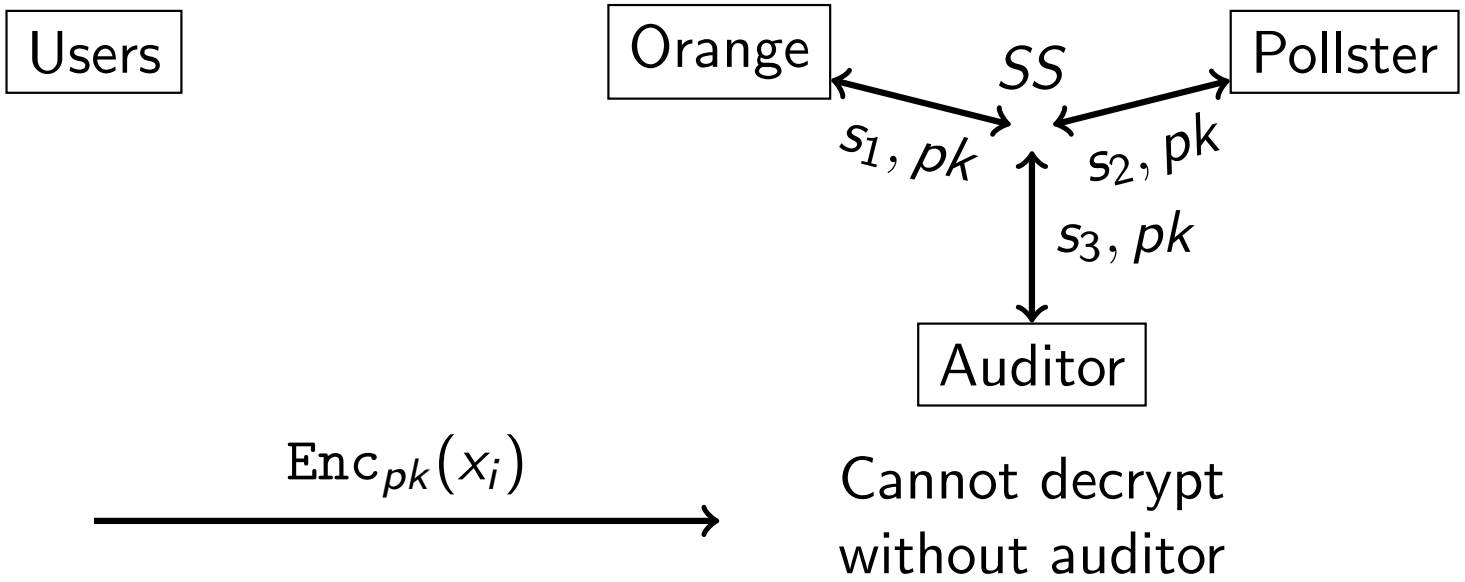
Any idea?

1. Encrypt with own secret key. Cannot aggregate data.
2. Users participate in the Distributed Key Generation. Not realistic.

Protecting users against a coalition

Any idea?

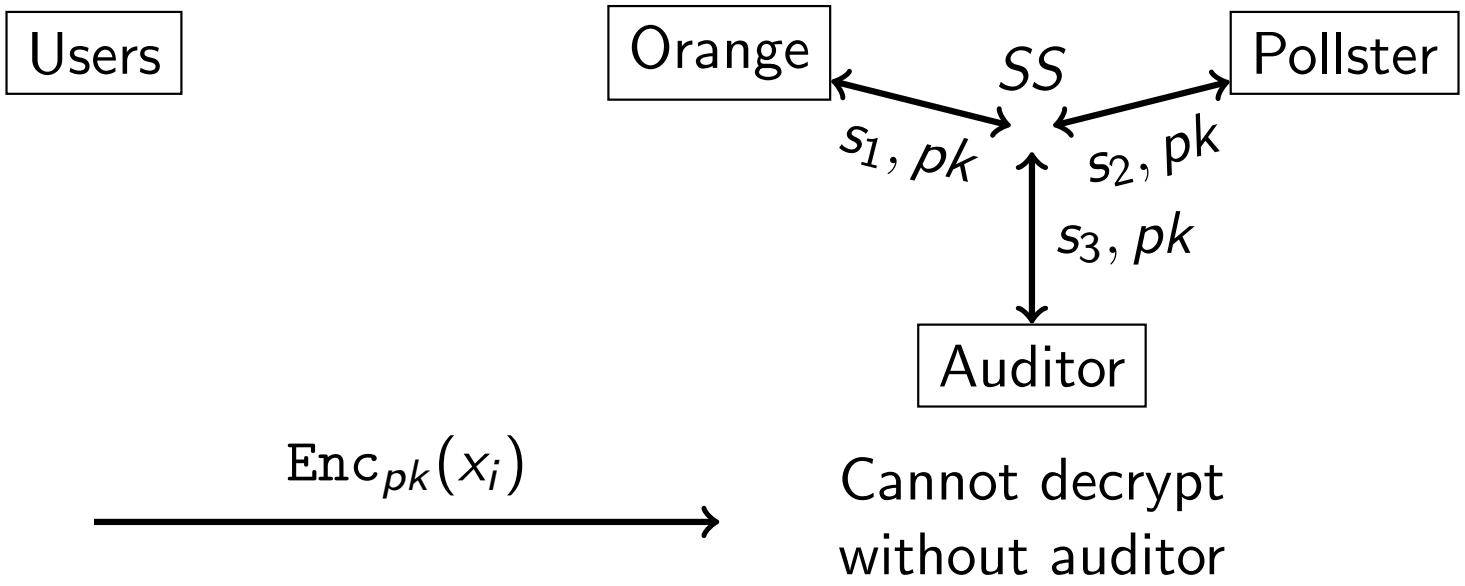
1. Encrypt with own secret key. Cannot aggregate data.
2. Users participate in the Distributed Key Generation. Not realistic.
3. Trust an Auditor.



Protecting users against a coalition

Any idea?

1. Encrypt with own secret key. Cannot aggregate data.
2. Users participate in the Distributed Key Generation. Not realistic.
3. Trust an Auditor. Proof are hard to verify.



Outline

Some Flaws of Privacy-Preserving Publication of Statistics

"Anonymity Threshold" $k = 10$

Zip	Age	Gender	Nationality	Fav. App	Hours in Social Media	
*	[21; 31[	*	*	*	17	10 individuals — Mean: ≥ 15
*	[21; 31[	*	*	*	15	
⋮	⋮	⋮	⋮	⋮	$\geq$15	
*	[21; 31[	*	*	*	18	
*	[31; 41[	*	*	Porn	*	10 individuals — Mode: Porn
*	[31; 41[	*	*	Porn	*	
⋮	⋮	⋮	⋮	Porn	⋮	
*	[31; 41[	*	*	Porn	*	

Obvious (*among many*) attack by "Homogeneity":

- Known 29-year-old app-friend $\rightsquigarrow$ "high" spending time in social media.
- Known 35-year-old app-friend $\rightsquigarrow$ porn (published frequency and/or mode).

Remark

- Attacker has *background knowledge* $\rightsquigarrow$ privacy breaks down.
- Find a criterion for a database to be anonymized $\rightsquigarrow$ always flawed.
- Privacy should apply to the anonymization mechanism, not to its output.

Privacy-Preserving Publication of Statistics

Anonymization mechanism $\mathcal{M} \rightsquigarrow$ not leak private information.

- Dalenius (1977): For all[1] $X_1, X_2 \in \mathcal{X}$, $\mathcal{M}(X_1) = \mathcal{M}(X_2)$.
- Dwork (2006): Dalenius' goal is fundamentally **impossible** to achieve.

Differential Privacy (DP) $\rightsquigarrow \mathcal{M}(X_1) \approx \mathcal{M}(X_2)$.

- You can quantify the privacy loss (parameter ε);
- You no longer need attack modeling.

[1] Neighboring databases (differ in only one record): $||X_1 - X_2||_1 = 1$

First DP Scheme: "Centralized"

Assuming no coalition of Orange and Pollster

- Assumption: Users trust Orange only.
- Orange computes f over ciphertexts of users C;
- Orange computes the ciphered DP noise $C_N = \mathtt{Enc}_{pk}(N)$;
- Orange reports $\mathcal{M}(C, f(.), C_N) = f(C) + C_N$;

Remark

- Centralized DP is still subject to security breaches.
- Does not solve the coalition between Orange and Pollster.

Second DP scheme: "Decentralized"

Mitigating the coalition of Orange and Pollster

- Assumption: Users **do not** trust anyone else in the system;
- E.g., DP **mean** estimation.
- Each user $i \in \mathcal{I}$ with input value x_i **do**:
  Compute $\hat{x}_i = x_i + N$
  Report: $\hat{c}_i = \mathtt{Enc}_{pk}(\hat{x}_i)$
- *Orange* computes f over all $\hat{c}_i$.

Remark

- Local DP algorithms are (usually) designed for a specific task!
- DP noise in each x_i may compromise data utility.

Conclusion

Goal: privacy-preserving scheme for a pollster service by Orange.

What we did:

- Several schemes for different levels of attacks
- Solutions with FE and FHE

Problem encountered:

- Currentlty no FE compatible with our needs
- Only leads for coalition attack

Further Work:

- Security proofs of our schemes
- Further check over Differential Privacy solutions
- Suitable FE scheme

Thanks!