Differential Privacy In Times Of Adversity

Ari Biswas + Graham Cormode

Motivation

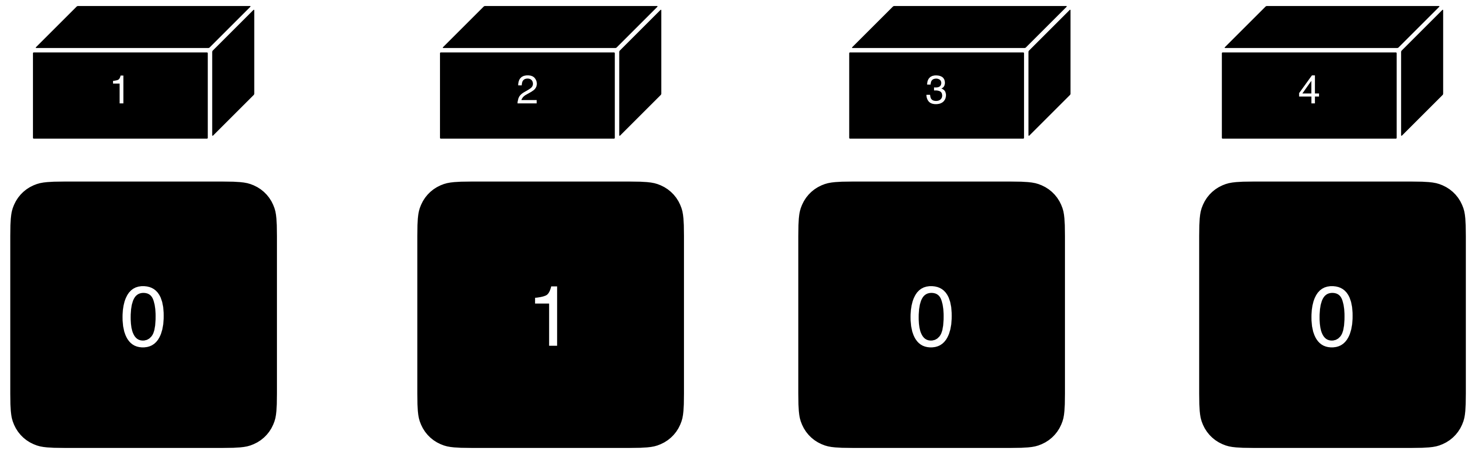
- A survey on sensitive topics where an estimate of the right answer is good enough.
- Each participant selects one out of M choices
- Let x_i represent the i'th persons data
- **Want to know the average number of YES values in each category**

Binary Choice : $x_i \in \{0,1\}$

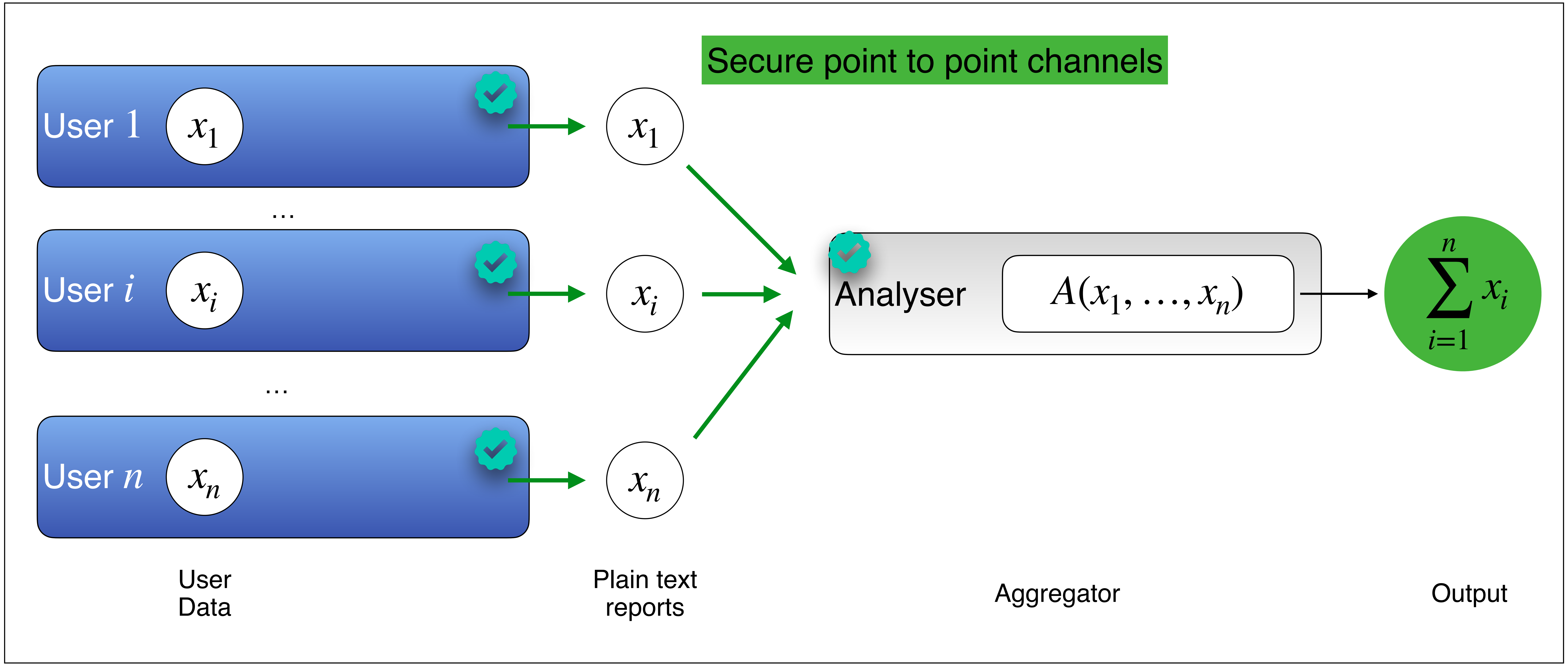
Question: have you ever cheated on an exam ?

M choices: $x_i \in \{1,2,\ldots,M\}$

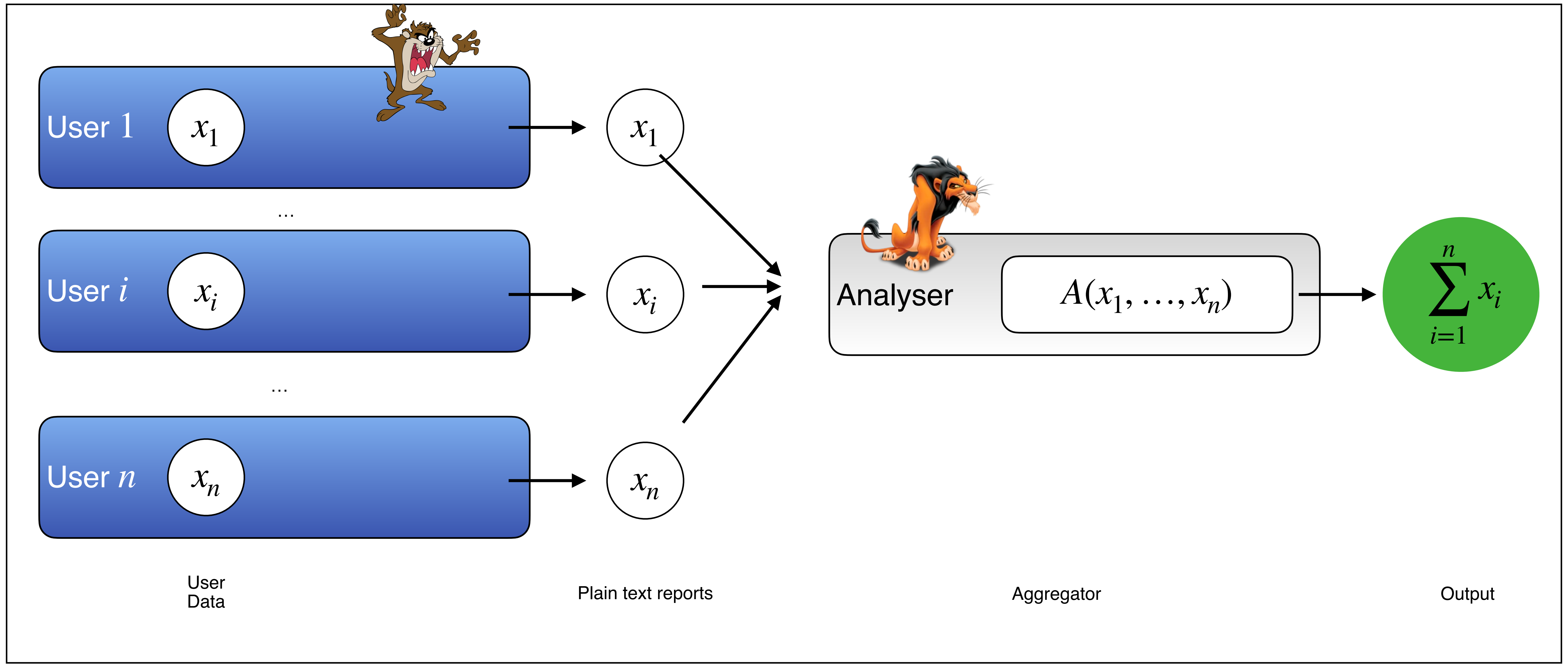
Question: what is the highest grade of felony you have been convicted of?



Ideal solution



Notations and Assumptions

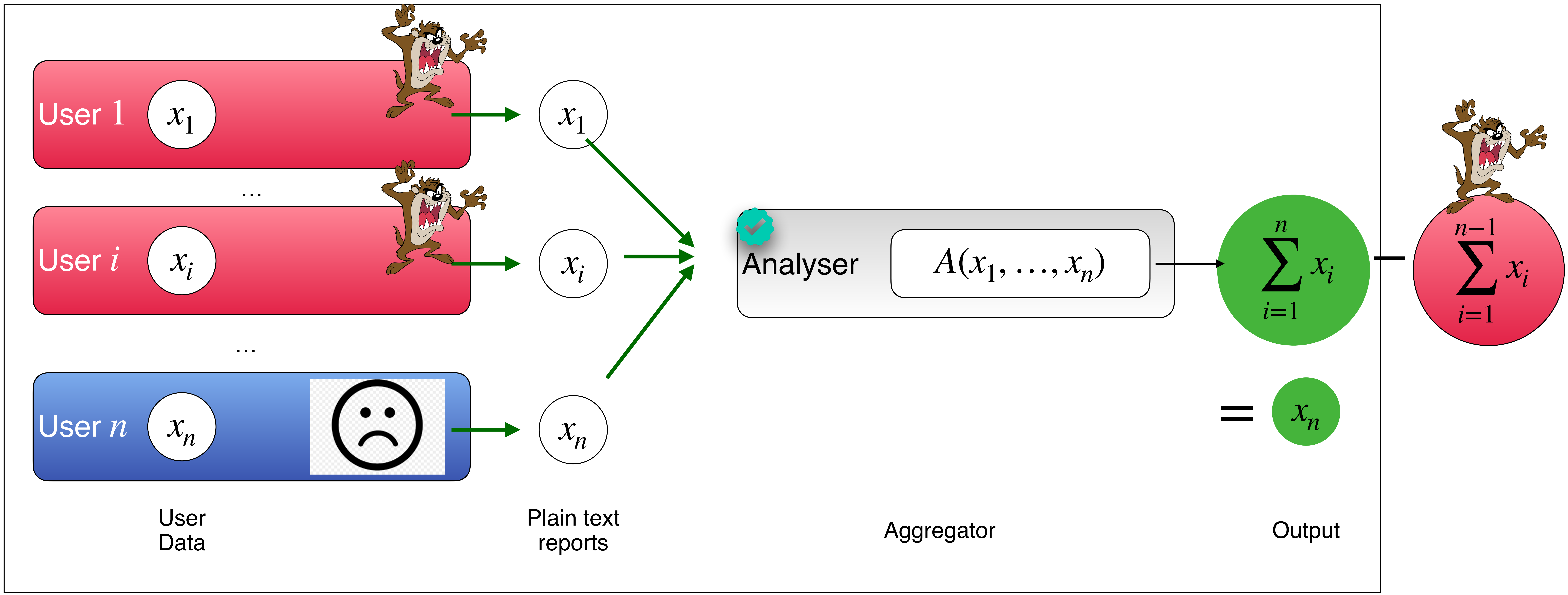


Deviates from prescribed protocol arbitrarily: **Active Adversary**

Follow all prescribed protocols but try and learn as much additional information from protocol transcript: **Passive Adversary/Honest but curious/ Semi honest**

Will always assume secure point to point and broadcast channels.

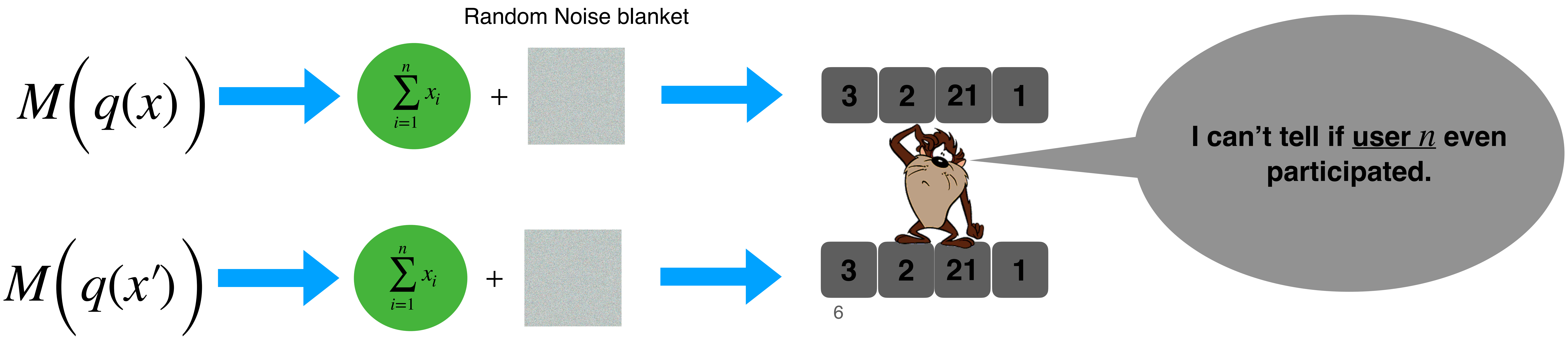
Adversary controls $n-1$ users



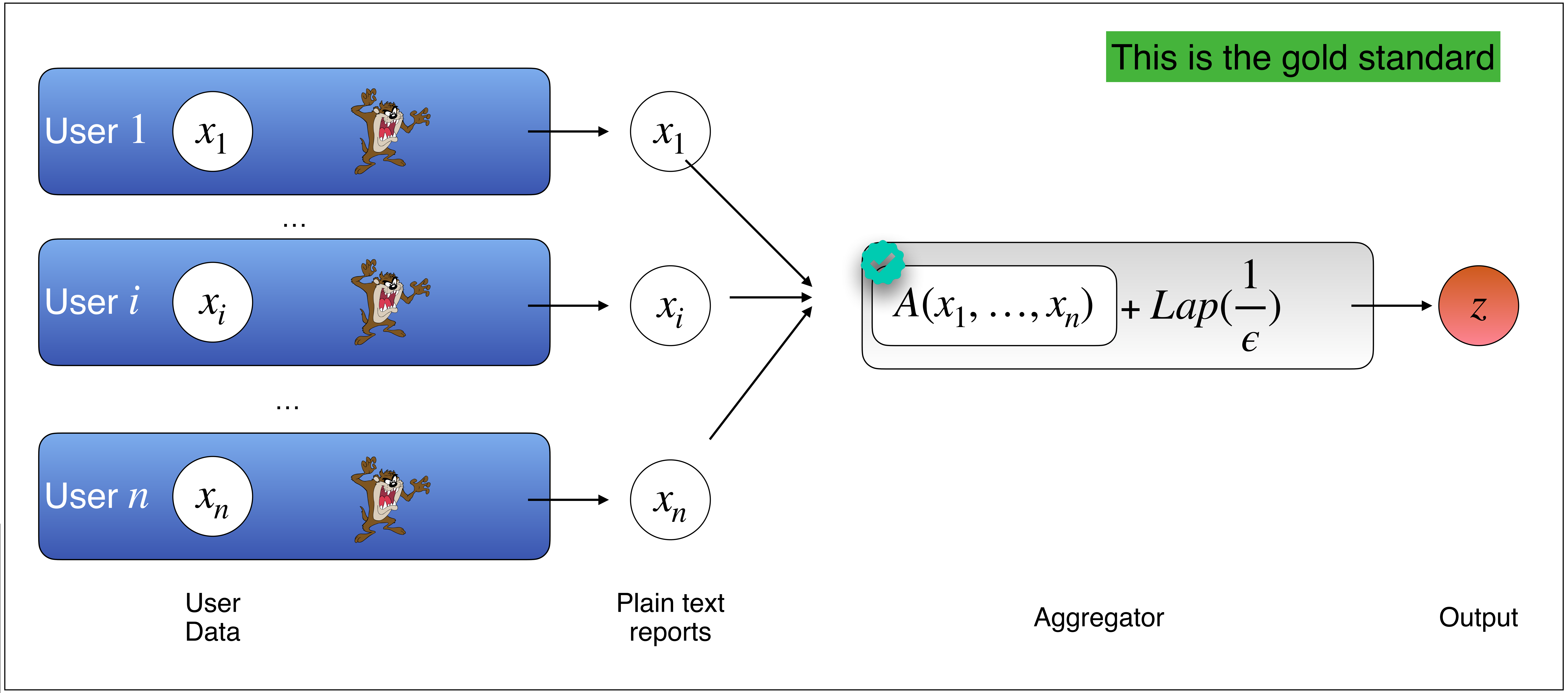
Differential Privacy

An algorithm $M : (X^n \times Q) \to Z$ satisfies (ϵ, δ) differentially private if for every two neighbouring datasets $x \sim x' \in X^n$ and for every query $q \in Q$ we have

$$\forall T \subseteq Z,\ \mathbb{P}[M(x,q) \in T] \leq e^{\epsilon}\, \mathbb{P}[M(x',q) \in T] + \delta$$



Laplace Mechanism

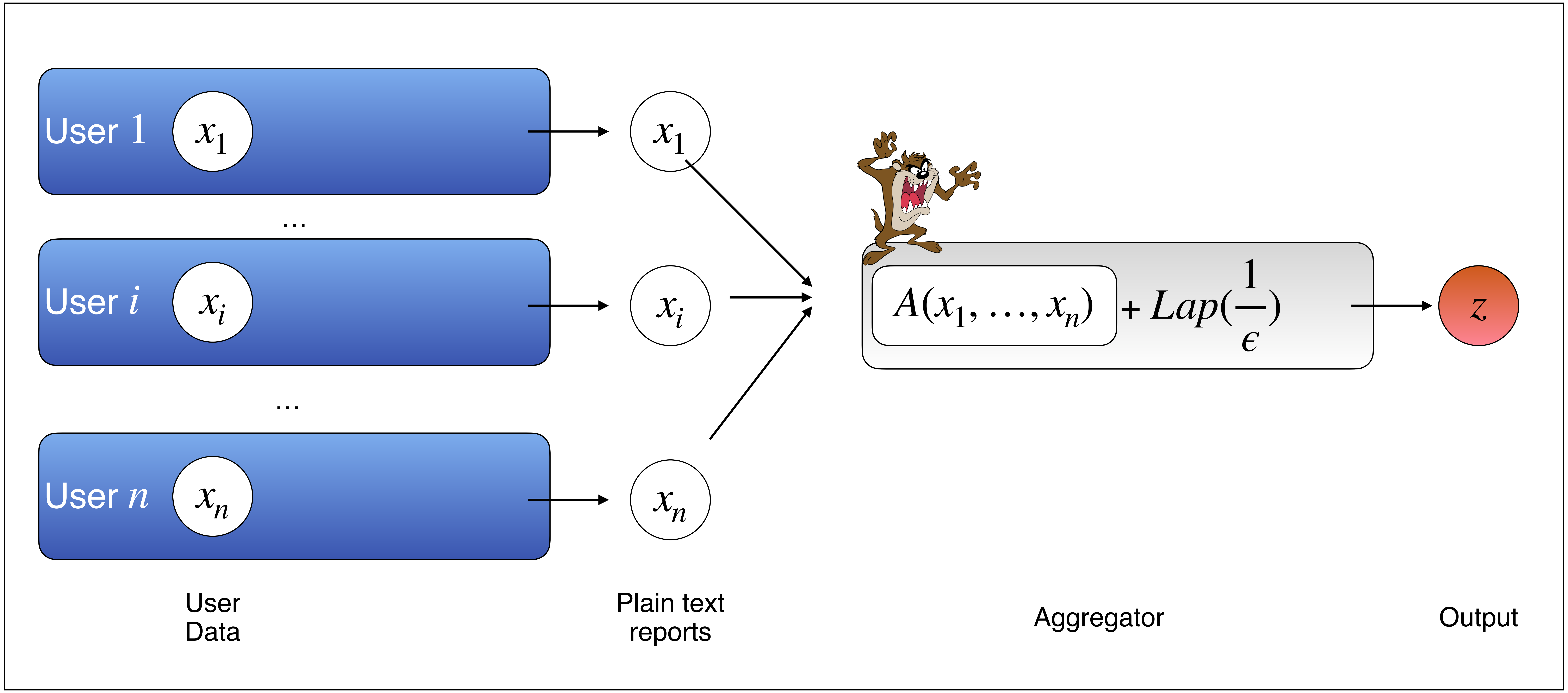


$$f = \sum_{i=1} x_i$$

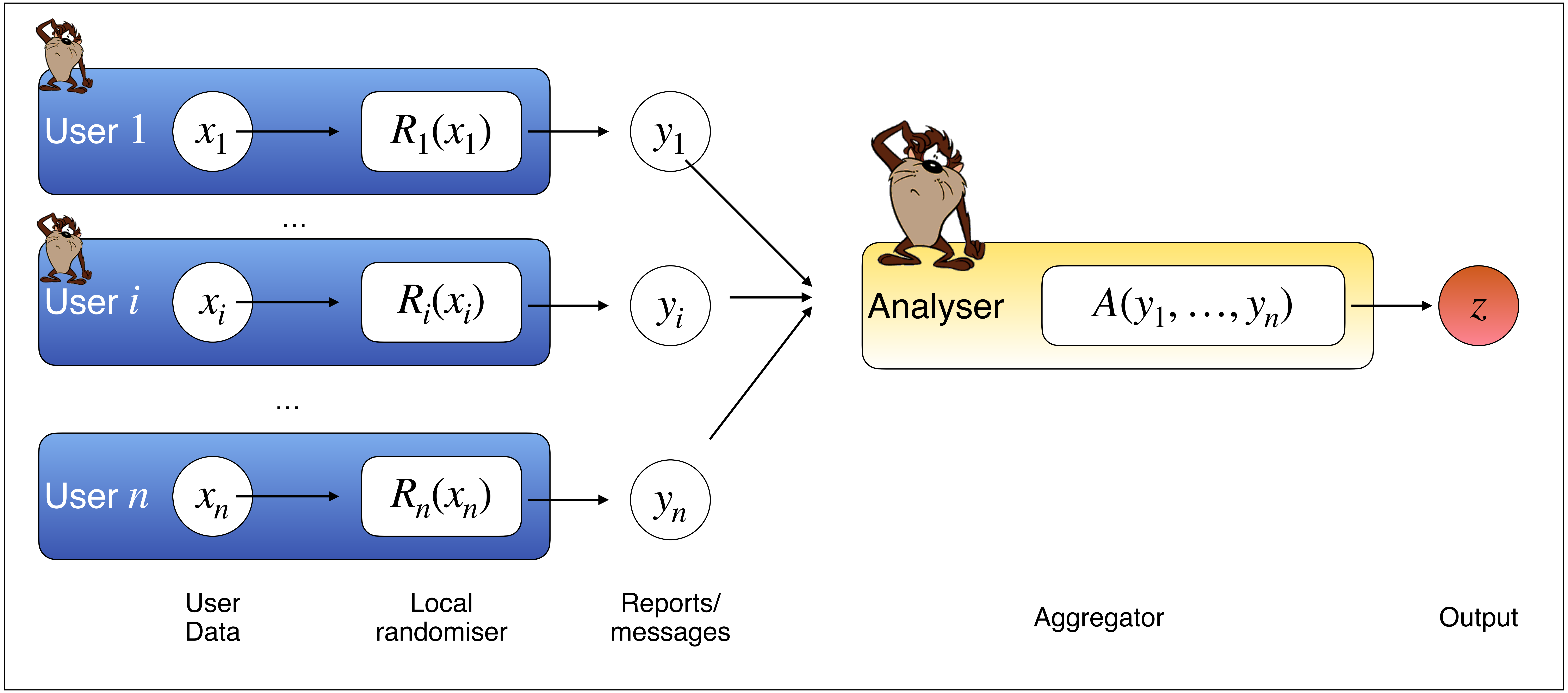
$$\hat{f} = \sum_{i=1}^{n} x_i + Lap(\frac{1}{\epsilon})$$

Central error $|\hat{f} - f| \le O(\frac{1}{\epsilon})$

What If I cannot trust the curator ?



Local Differential Privacy: Noisify at the other end

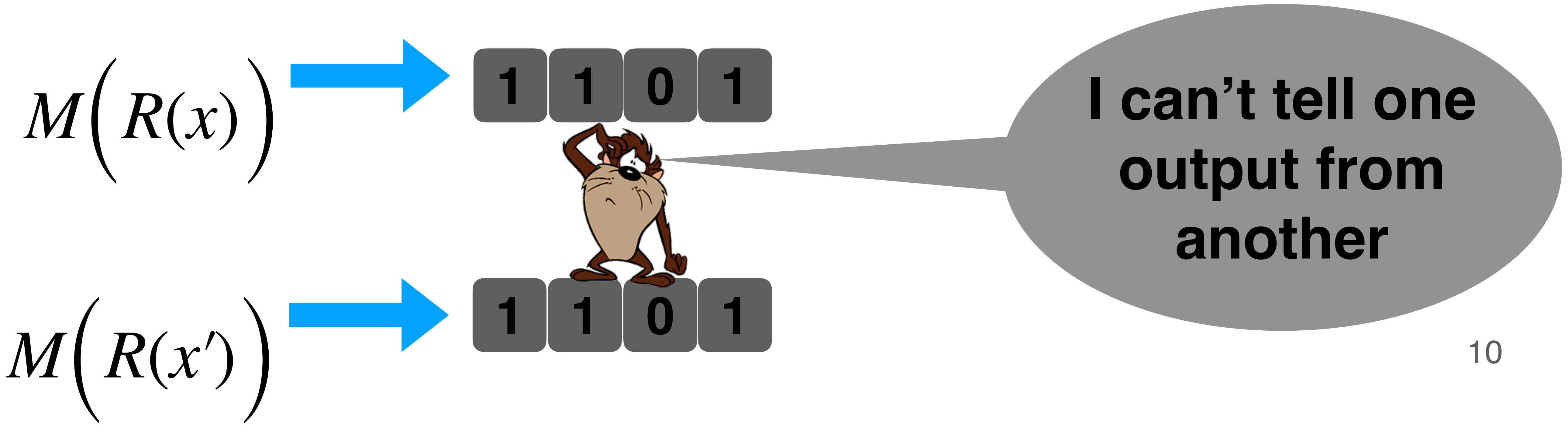


When not mentioned specifically we assume $|x_i| = |y_i|$

Local Differential Privacy

An algorithm $R : X \to Y$ satisfies (ϵ, δ) local differential privacy if for every two users $x, x' \in X$

$$\forall T \subseteq Y,\ \mathbb{P}[R(x) \in T] \leq e^{\epsilon}\, \mathbb{P}[R(x') \in T] + \delta$$



Randomised Response: A way to get local DP

Let $p \in (0,1/2)$

$$y_i = \begin{cases} x_i & \text{with pr. } \frac{1}{2} + p \\ 1 - x_i & \text{with pr. } \frac{1}{2} - p \end{cases}$$

$f = \sum_{i=1} x_i$ What we want to estimate

$\hat{f} = \sum_{i=1}^{n} \left[\frac{1}{2p}(y_i - 1/2 + p)\right]$ What we estimate

$\mathbb{E}[\hat{f}] = f$

(In expectation they are the same)

LDP error $|\hat{f} - f| \leq O(\frac{\sqrt{n}}{\epsilon})$ ⟵ Unavoidable

Central error $|\hat{f} - f| \leq O(\frac{1}{\epsilon})$

Randomised Response is optimal in LDP

Duchi, Jordan, and Wainwright, 'Local Privacy, Data Processing Inequalities, and Statistical Minimax Rates'.

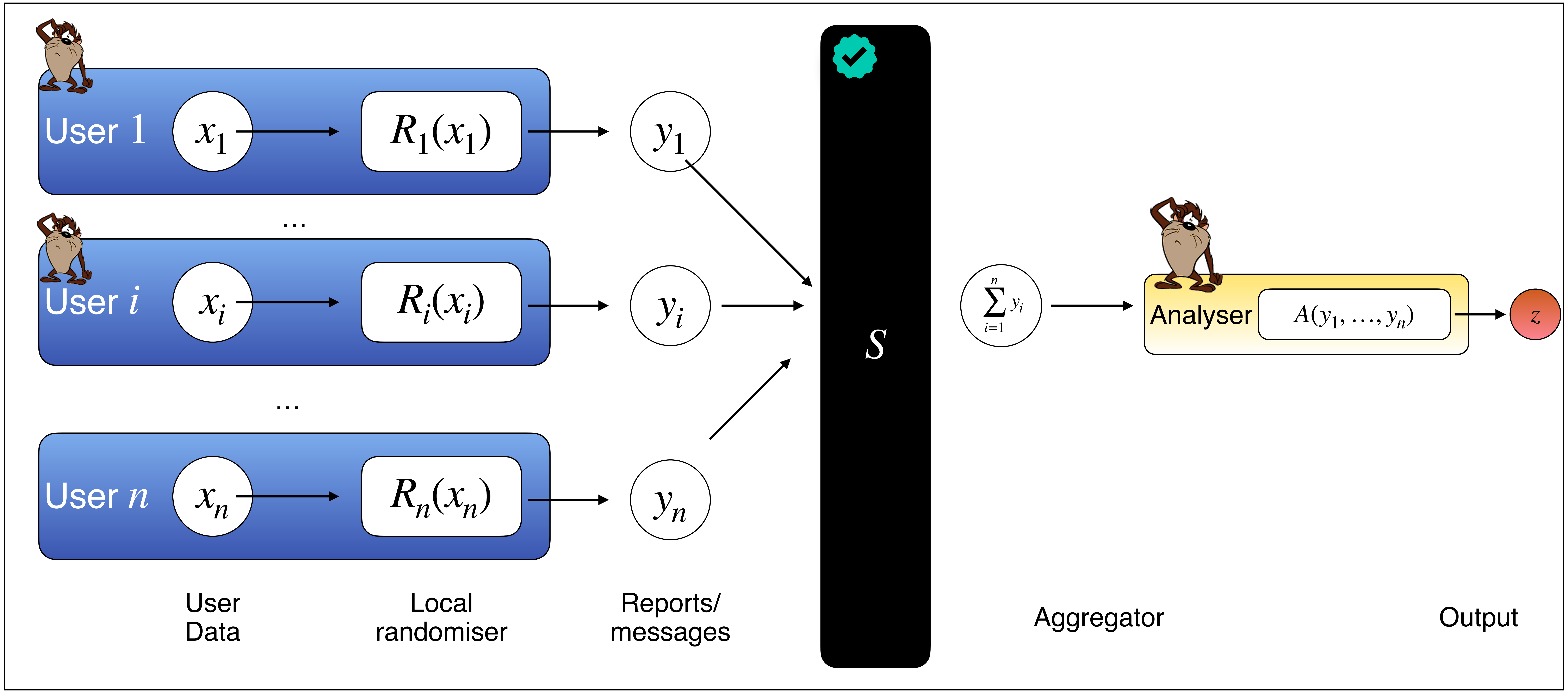
If you can attack Randomised Response, you can attack all LDP algorithms

Cheu, Smith, and Ullman, 'Manipulation Attacks in Local Differential Privacy'.

Comparison is unfair — LDP imposes stricter privacy constraints

Each user has to generate enough to noise to hide himself as opposed to each user has to generate enough noise to hide amongst the crowd.

Shuffle Privacy



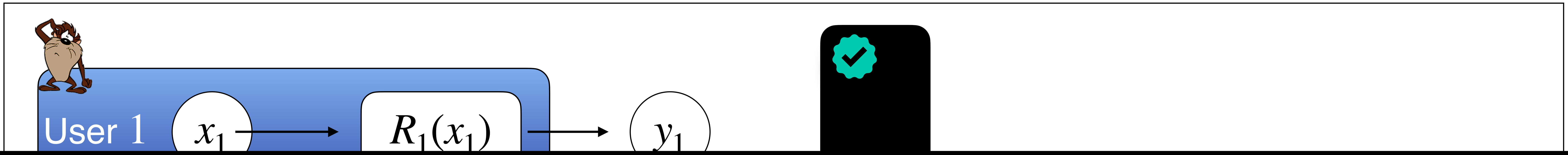
Randomised Response gives near central error

Balle et al., 'The Privacy Blanket of the Shuffle

If we do not restrict $|x| = |y|$ we can do just as well as central
Balcer and Cheu, 'Separating Local & Shuffled Differential Privacy via Histograms'.

If we do not restrict $|x| = |y|$ we can do just as well as central
Ghazi et al., 'Differentially Private Aggregation in the Shuffle Model'.

Shuffle Privacy



Rich history of research and improvements but all this work has been done under the semi honest model.

Randomised Response gives near central error

Balle et al., 'The Privacy Blanket of the Shuffle

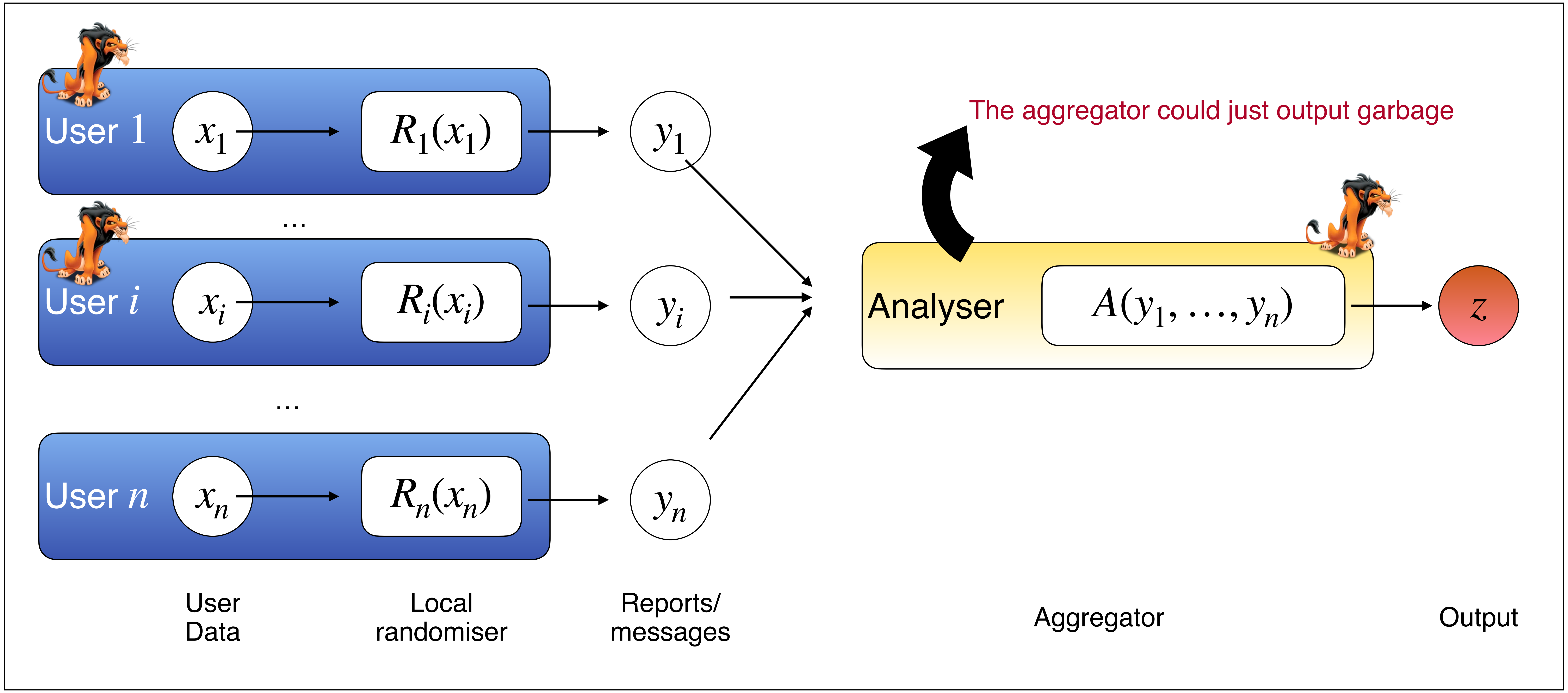
If we do not restrict $|x| = |y|$ we can do just as well as central

Balcer and Cheu, 'Separating Local & Shuffled Differential Privacy via Histograms'.

If we do not restrict $|x| = |y|$ we can do just as well as central

Ghazi et al., 'Differentially Private Aggregation in the Shuffle Model'.

What happens when the curious become dishonest?

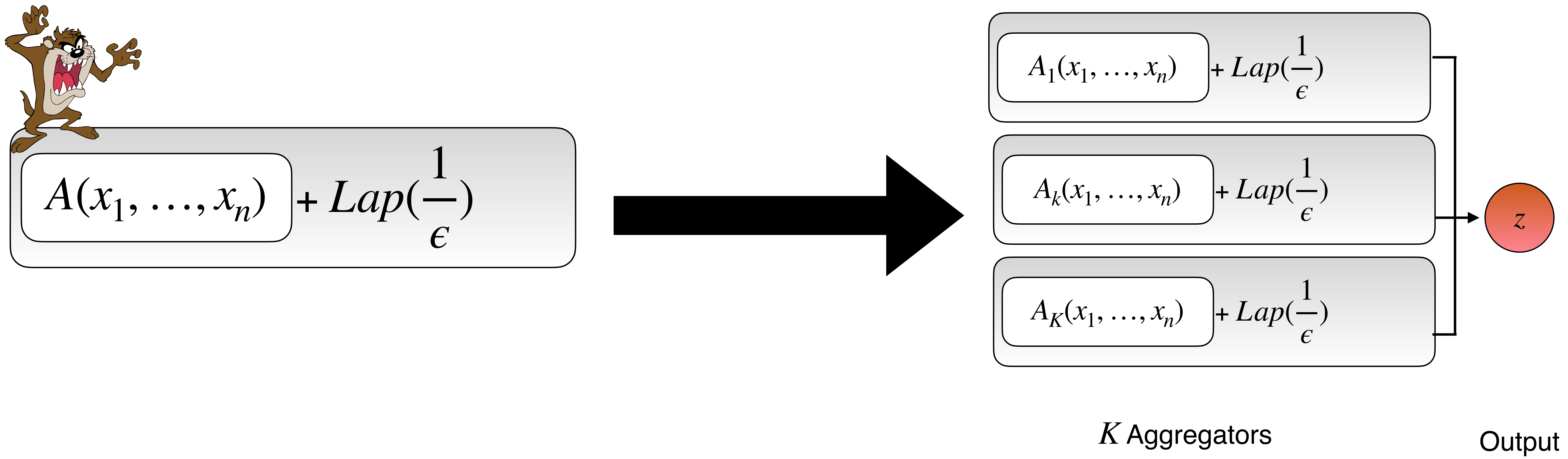


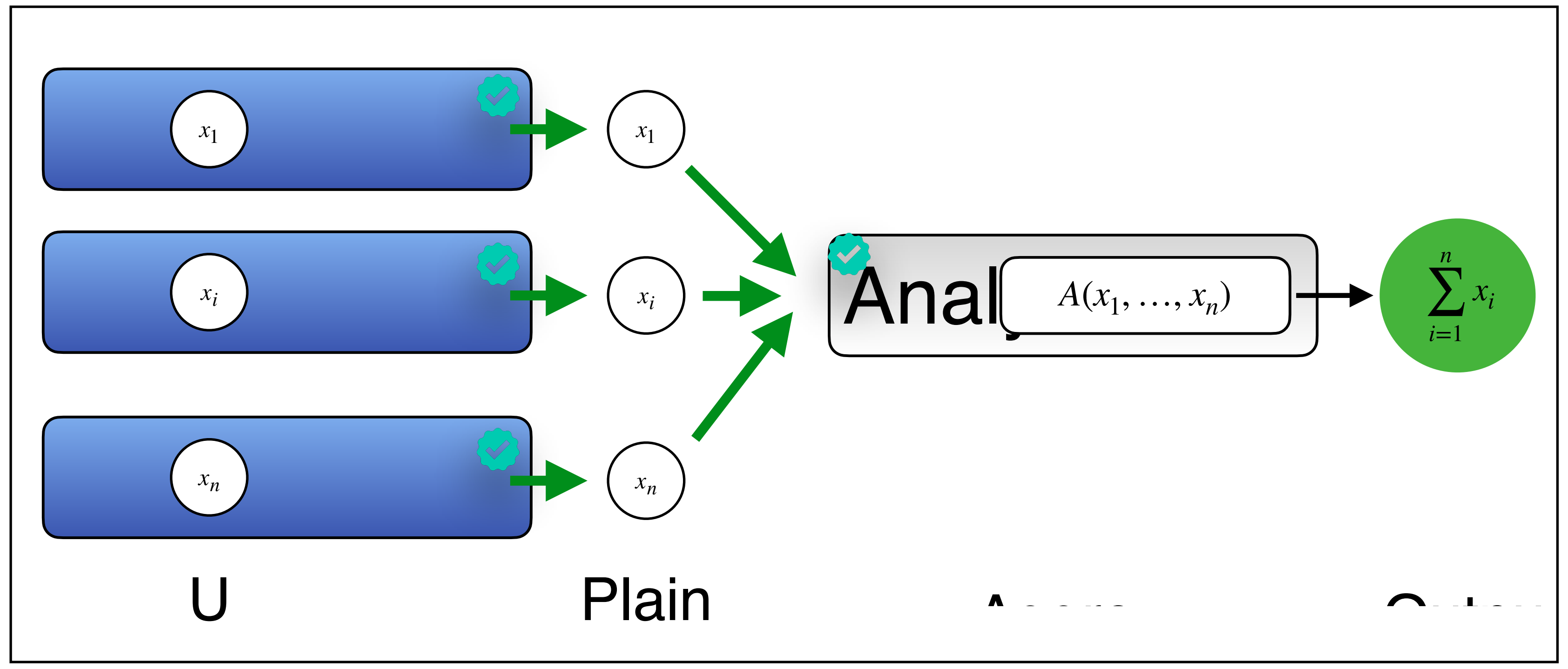
Derive bounds for how many users have to be corrupted for randomised response utility to be indistinguishable from $Bin(n, \frac{1}{2})$

Cheu, Smith, and Ullman, 'Manipulation Attacks in Local Differential Privacy'.

Relax the ideal world to allow for more curators

What does it mean to be fine ?





Don't have enough money to corrupt all K curators. Let's hope that as long as at least **1** server is semi honest we will be fine

What we consider to be fine*

* Fine is determined by assumptions that K curators and only guaranteed that 1 server is semi honest

- Perfect Privacy of inputs + Differential privacy + Output is meaningful

Not possible
Ben Or, Goldwasser and Widgerson, ‘Completeness theorems for non cryptographic fault tolerant distributed computation’

- Perfect Privacy of inputs + Differential privacy + Output is not guaranteed to be meaningful

Poplar: The focus is on lightweight protocols and the emphasis is on privacy
Boneh et al., ‘Lightweight Techniques for Private Heavy Hitters’.

- Computational privacy of inputs + differential privacy + output is “guaranteed”to be meaningful
    - If an adversary violates any of this, the honest server detects this and tells everyone that they cheated and voids the protocol.

Our work: Lightweight-ish but focus on realiability

Informal defintion

Linear Secret Sharing

- Two algorithms share and reconstruct s.t $s \in \mathbb{Z}_q$
  - Share(s) = $[s]_1, \ldots, [s]_K$ such that $[s]_i \xleftarrow{R} \mathbb{Z}_q$ for $i \in [K-1]$ and $$[s]_K = s - \sum_{i=1}^{K-1} [s]_i$$
  - Reconstruct($[s]_1, \ldots, [s]_K$) = $$\sum_{i=1}^{K} [s]_i = s$$

Example in $\mathbb{Z}_{11}$ and $K = 3$

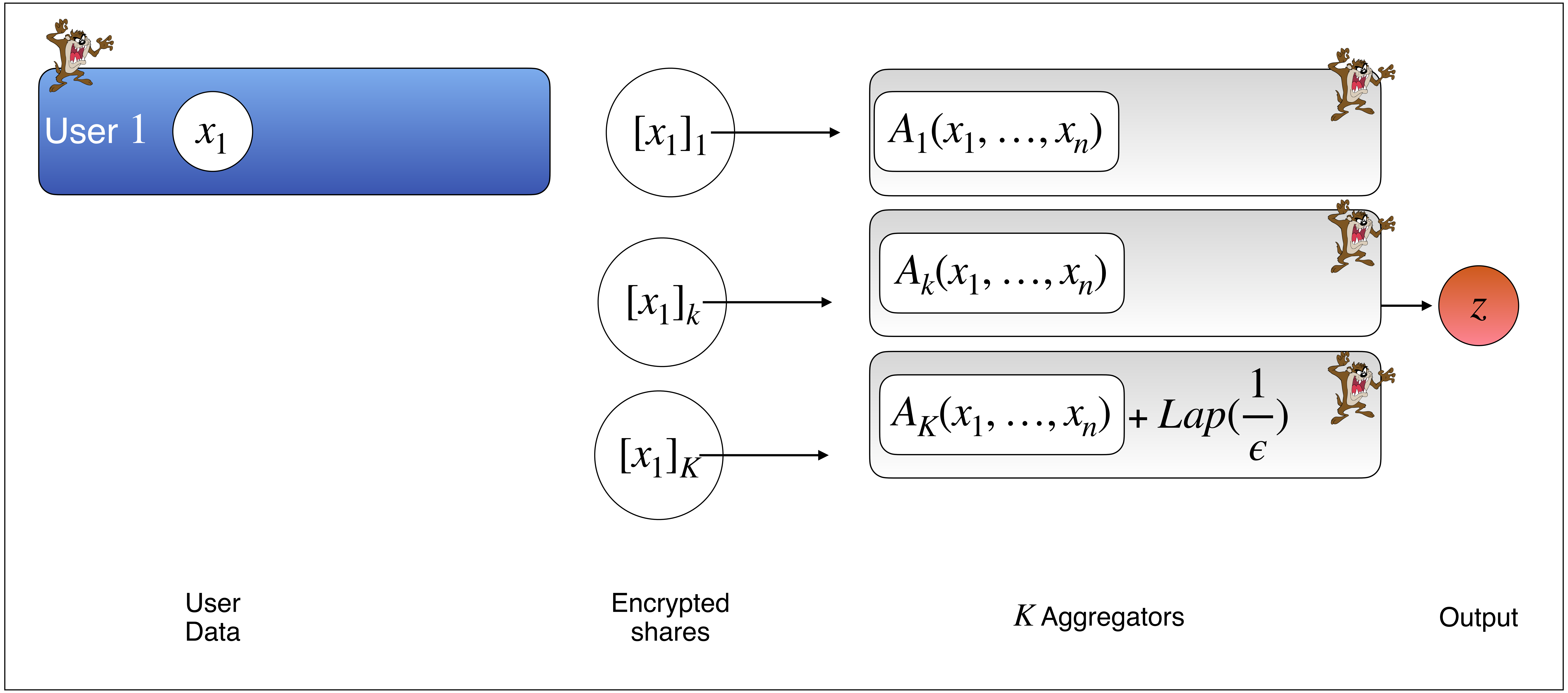
Secret $s = 7$

$[s]_1 = 4$
$[s]_2 = 5$
$[s]_3 = 9$

$$\sum_{i=1}^{3} [s]_i = (4 + 5 + 9) \mod 11 = 7$$

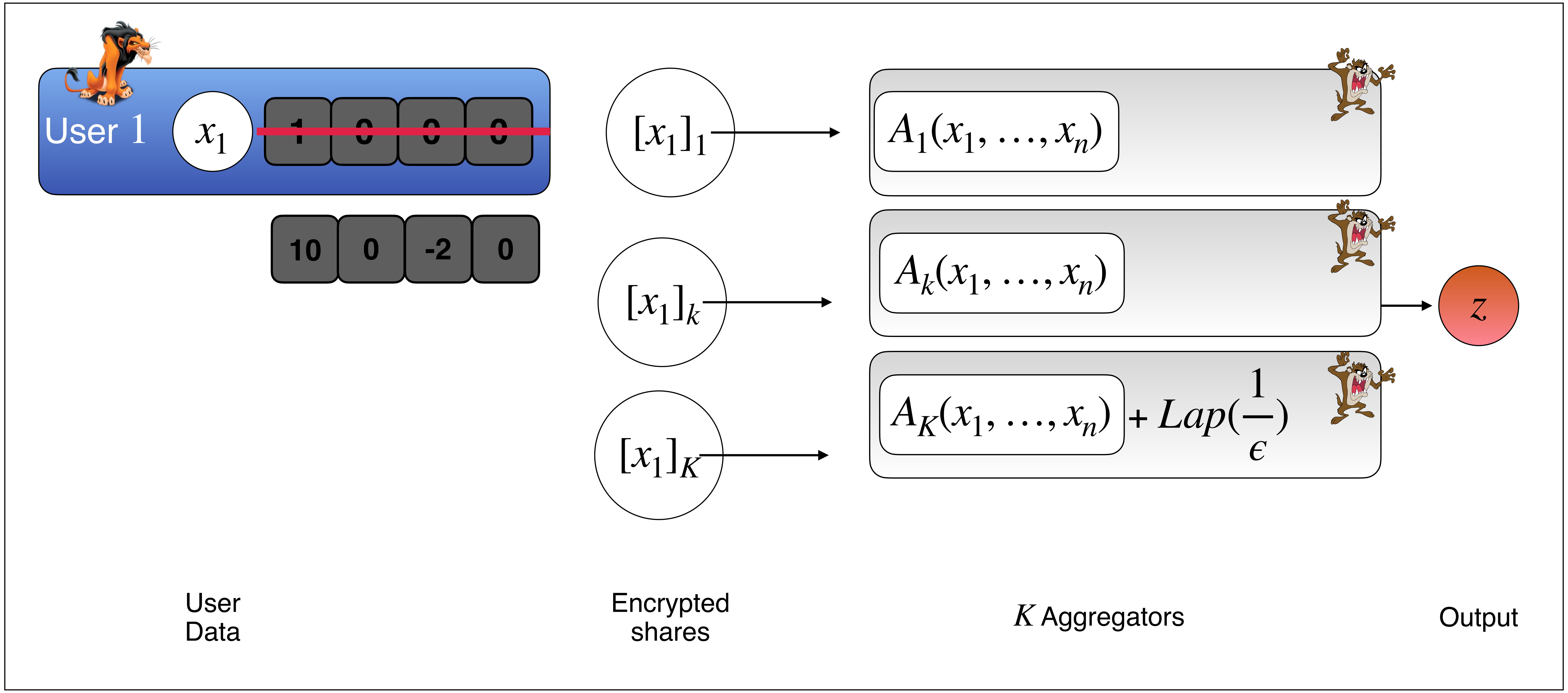
Adversary in possession of $K-1$ shares learns no Shannon information about s

PRIO



Correctness: $\sum_{i=1}^{n} \sum_{k=1}^{K} [x_i]_k = \sum_{i=1}^{n} x_i$

Ballot stuffing



Sketching protocol

$[x]_1 \in \mathbb{Z}_q^M$

$[x]_3 \in \mathbb{Z}_q^M$

$[x]_2 \in \mathbb{Z}_q^M$

Sketching protocol from work in 2016 on function secret sharing

Boyle, Gilboa, and Ishai, ‘Function Secret Sharing’.

Sketching protocol

$[x]_1 \in \mathbb{Z}_q^M$

$[x]_3 \in \mathbb{Z}_q^M$

1. Server 1 samples $r_1, \ldots, r_M$ where $r_i \xleftarrow{R} \mathbb{Z}_q$ independently and broadcasts it to other servers

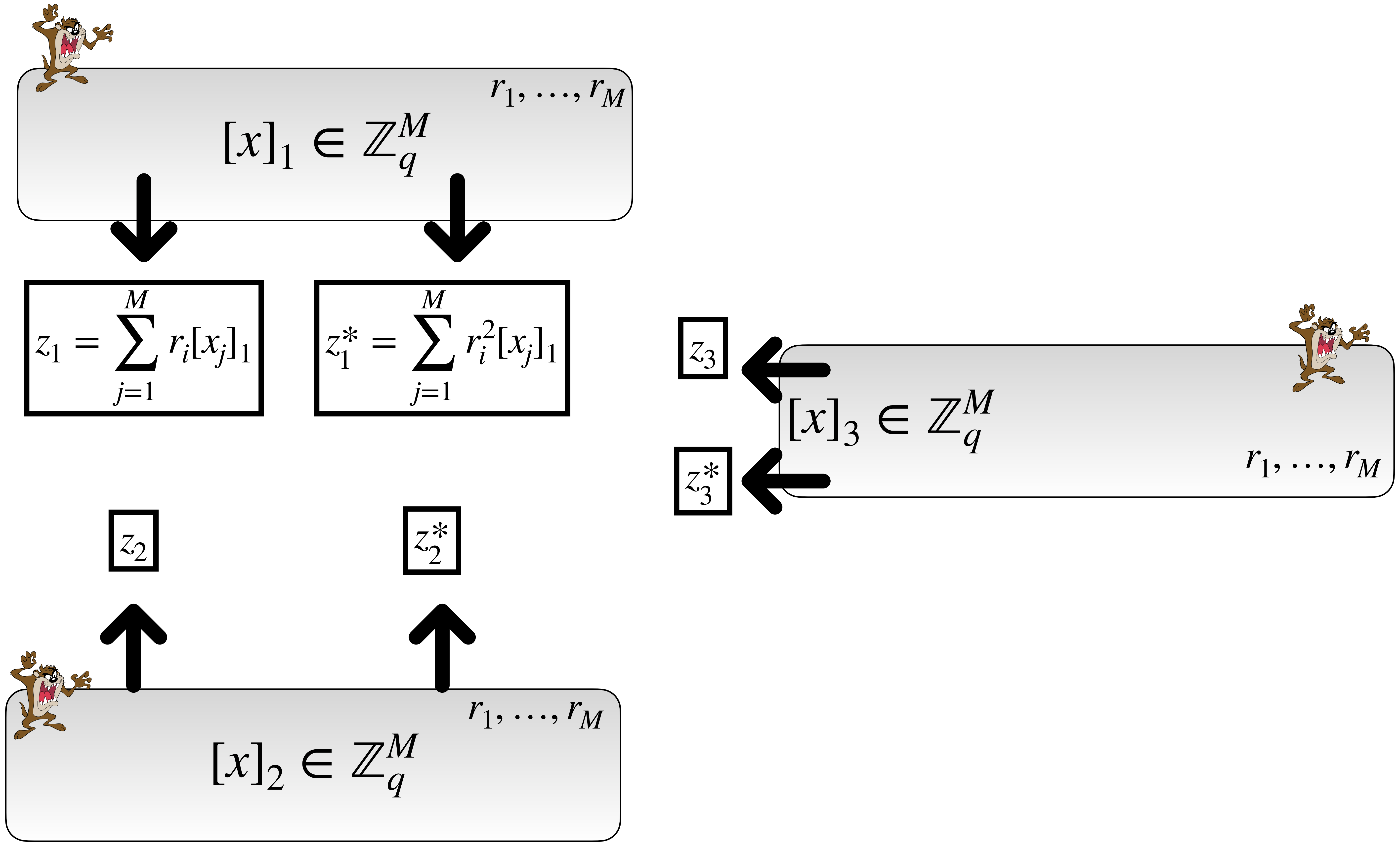
$[x]_2 \in \mathbb{Z}_q^M$

INTUITION:

We will create a random $p(\vec{r})$ degree two polynomial. If sampled $\vec{r}$ is not a root, then the only way to 0 out this polynomial is to have a single non zero entry equal to 1

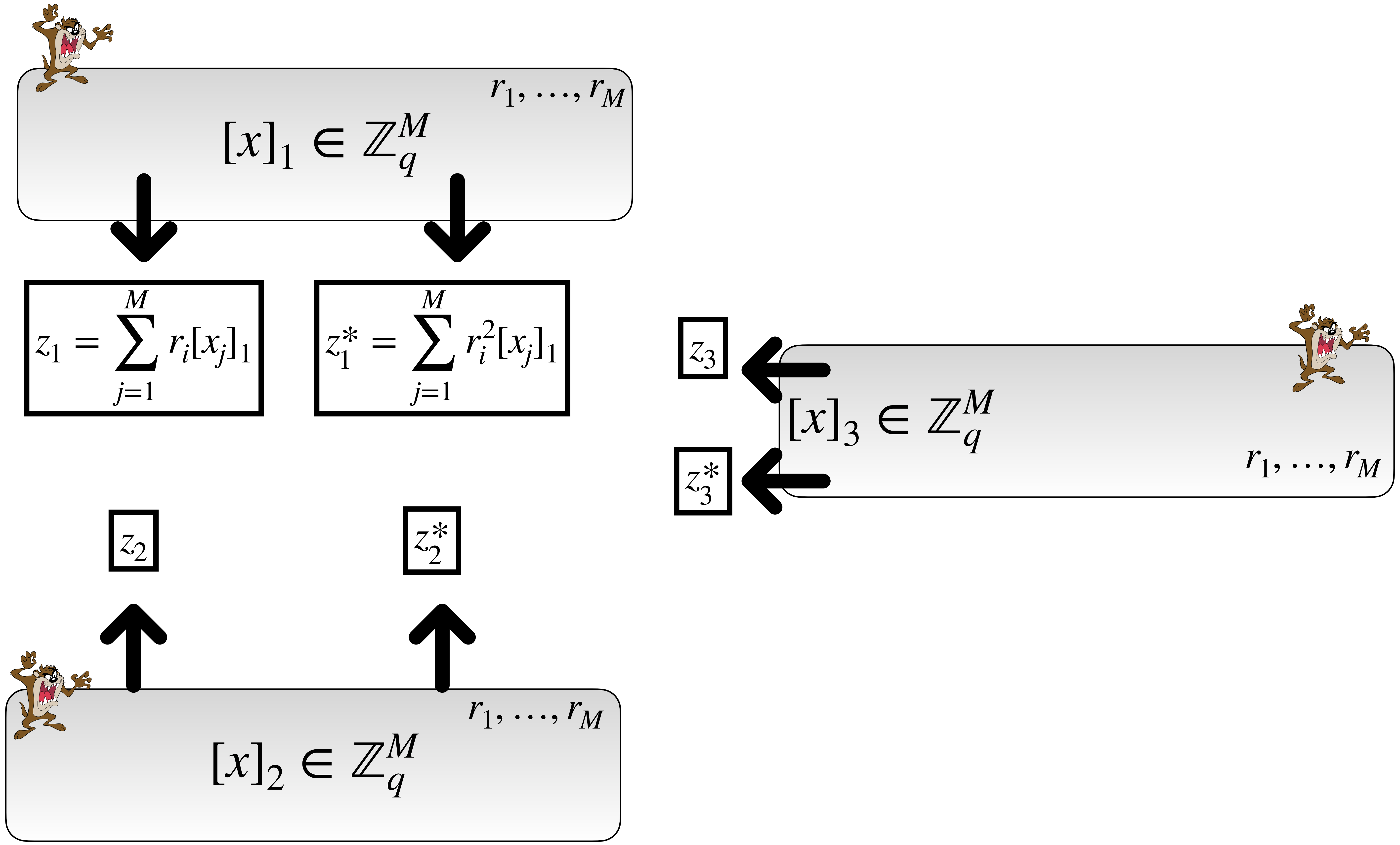
The cheating client does not know the values of $\vec{r}$ thus with probability $\frac{2}{q}$ fails to pass to the test

Sketching protocol



1. Server 1 samples $r_1, \ldots, r_M$ where $r_i \xleftarrow{R} \mathbb{Z}_q$ independently and broadcasts it to other servers
2. Server k broadcasts $z_k = \sum_{j=1}^{M} r_i [x_j]_k$ and $z_k^* = \sum_{j=1}^{M} r_i^2 [x_j]_k$

Sketching protocol

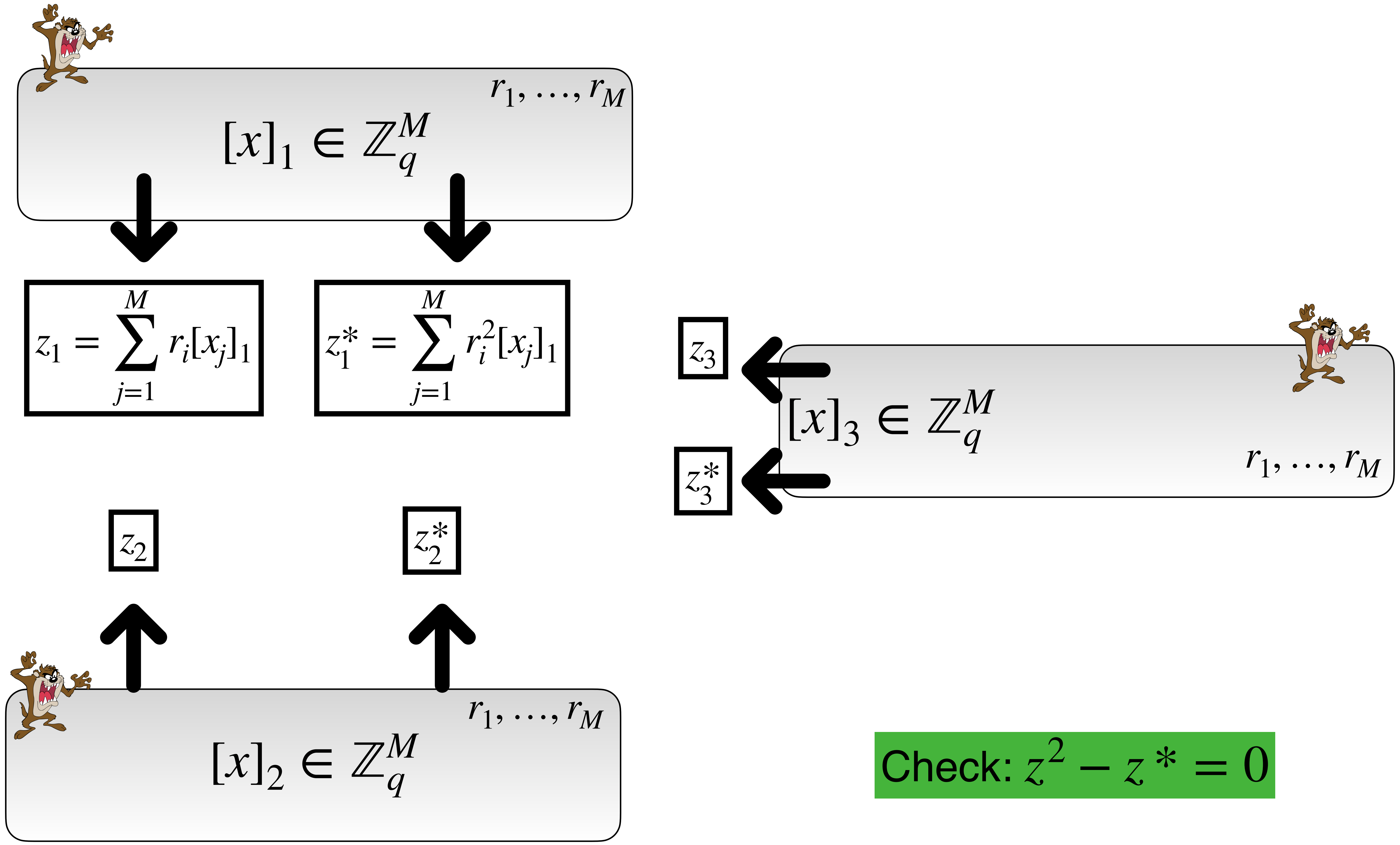


1. Server 1 samples $r_1, \ldots, r_M$ where $r_i \xleftarrow{R} \mathbb{Z}_q$ independently and broadcasts it to other servers

3. Server k broadcasts $z_k = \sum_{j=1}^{M} r_i[x_j]_k$ and $z_k^* = \sum_{j=1}^{M} r_i^2[x_j]_k$

4. Each server computes $z = \sum_{i=1}^{3} z_i$ and $z^* = \sum_{i=1}^{3} z_i^*$

Sketching protocol



Check: $z^2 - z^* = 0$

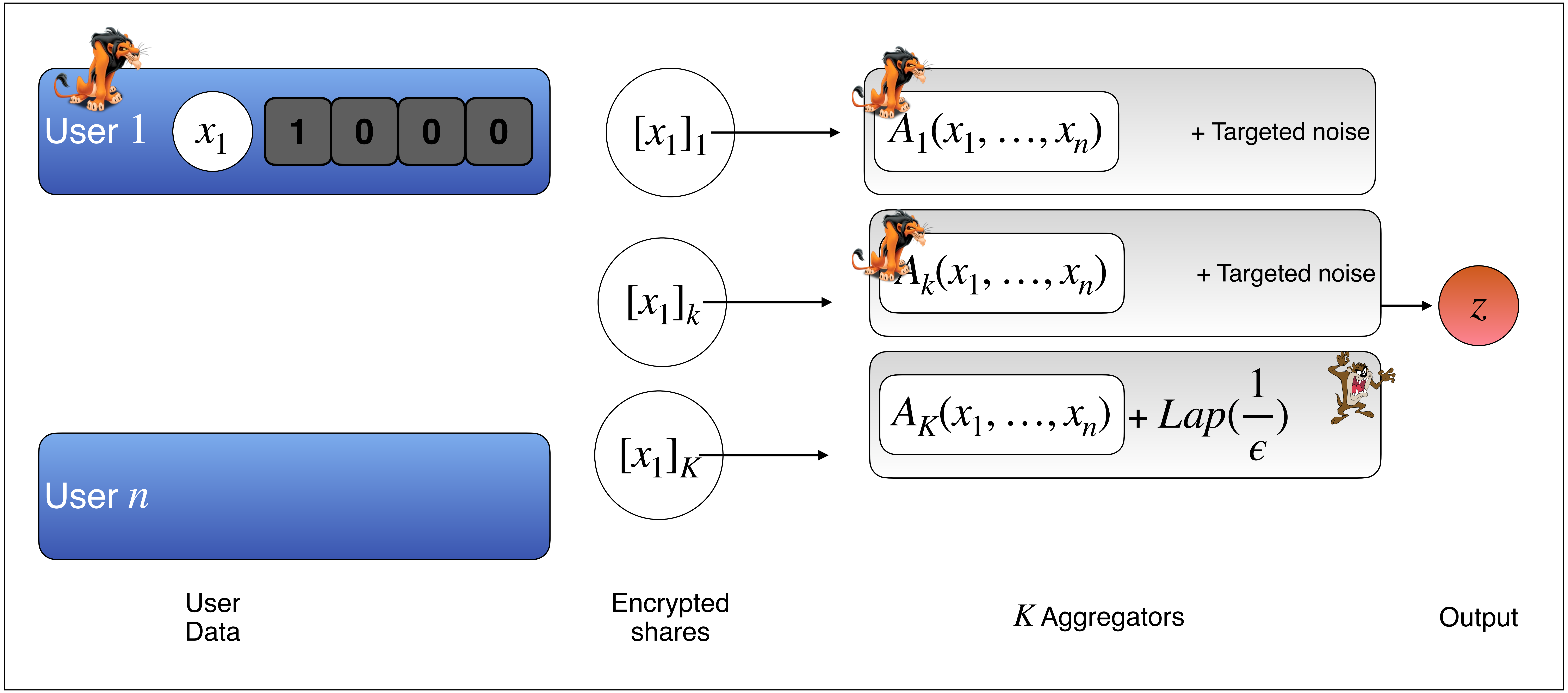
1. Server 1 samples $r_1, \ldots, r_M$ where $r_i \xleftarrow{R} \mathbb{Z}_q$ independently and broadcasts it to other servers

3. Server k broadcasts $z_k = \sum_{j=1}^{M} r_i [x_j]_k$ and $z_k^* = \sum_{j=1}^{M} r_i^2 [x_j]_k$

4. Each server computes $z = \sum_{i=1}^{3} z_i$ and $z^* = \sum_{i=1}^{3} z_i^*$

$$
\begin{aligned}
z^2 - z^* &= \Big(\sum_{i \in [M]} r_i v_i\Big)^2 - \sum_{i \in [M]} r_i^2 v_i \\
&= \sum_{i \in [M]} r_i^2 v_i (v_i - 1) + 2 \sum_{i,j: i \neq j} r_i r_j v_i v_j
\end{aligned}
$$

K-1 corrupt servers



Is sketching still secure ?

Sliding Attack on honest client

Adds +1 at some index and subtracts -1

$r_1, \dots, r_M$

$[x]_1 \in \mathbb{Z}_q^M$

$z_1 = \sum_{j=1}^{M} r_i [x_j]_1$

$z_1^* = \sum_{j=1}^{M} r_i^2 [x_j]_1$

$x =$ 0 1 0 0

$\boxed{z_3}$

$[x]_3 \in \mathbb{Z}_q^M$

$r_1, \dots, r_M$

Leaks 1 bit of information.

$\boxed{z_2}$ $\boxed{z_2^*}$ $\boxed{z_3^*}$

$r_1, \dots, r_M$

$[x]_2 \in \mathbb{Z}_q^M$

Check: $z^2 - z^* = 0$

Malicious Sketching

$\kappa[x]_1, [x]_1 \in \mathbb{Z}_q^M$

<u>**Only the honest client knows κ**</u>

Servers now also broadcast

$$z_k^{**} = \sum_{i=1}^{M} r_i(\kappa[x_i]_k)$$

$\kappa[x]_3, [x]_3 \in \mathbb{Z}_q^M$

Check: $(z^2 - z^*) + (\kappa z - z^{**}) = 0$

Show that the protocol is zero knowledge and a dishonest server does not learn any new information

Boneh et al., ‘Lightweight Techniques for Private Heavy Hitters’.

$\kappa[x]_2, [x]_2 \in \mathbb{Z}_q^M$

The secrecy of κ prevents a sliding attack.
We are abstracting details of implementation: In reality the client also has to supply beaver triples or Shares of κ

Collusions break Sketching protocols

$\kappa[x]_1, [x]_1 \in \mathbb{Z}_q^M$

<u>Only the honest client knows κ</u>

Servers now also broadcast

$$z_k^{**} = \sum_{i=1}^{M} r_i(\kappa[x_i]_k)$$

$\kappa[x]_3, [x]_3 \in \mathbb{Z}_q^M$

Check: $(z^2 - z^*) + (\kappa z - z^{**}) = 0$

Corrupt client has an illegal input, wants to be included in the protocol.

Just tells the corrupt servers what they sent to honest server 1

$\kappa[x]_2, [x]_2 \in \mathbb{Z}_q^M$

Collusions break Sketching protocols

$\kappa[x]_1, [x]_1 \in \mathbb{Z}_q^M$

$z_1 = \sum_{j=1}^{M} r_i[x_j]_1$

$z_1^* = \sum_{j=1}^{M} r_i^2[x_j]_1$

z_2

z_2^*

$\kappa[x]_2, [x]_2 \in \mathbb{Z}_q^M$

z_3

z_3^*

$\kappa[x]_3, [x]_3 \in \mathbb{Z}_q^M$

<u>Only the honest client knows κ</u>

Servers now also broadcast

$$z_k^{**} = \sum_{i=1}^{M} r_i(\kappa[x_i]_k)$$

Corrupt client has an illegal input, wants to be included in the protocol.

Just tells the corrupt servers what they sent to honest server 1

Set

$z_1 + z_2 = -z_1$

$z_1^* + z_2^* = -z_1^*$

$z_1^{**} + z_2^{**} = -z_1^{**}$

Check: $(z^2 - z^*) + (\kappa z - z^{**}) = 0$

Our contribution

- We want the same trust model as PRIO/ POPLAR
- We want central privacy error guarantees
- **If any party deviates from the protocol, the honest party can detect it as such and prove it to a court of law that this party deviated from the protocol.**
- Thus the output of the our protocol is either ABORT or valid
- This comes at the cost of 1-bit leak of information and some lightweight public key cryptography

Formalised as Covert Security

Aumann and Lindell, 'Security Against Covert Adversaries'.

Contributions

Theorem 6.1. *A collusion between a dishonest client and $K - 1$ servers cannot include an illegal input with a success probability any greater than their advantage in the discrete log game.*

Theorem 6.2. *If a corrupt client can force honest servers to abort the protocol, then they have a non negligible advantage in the DLOG game.*

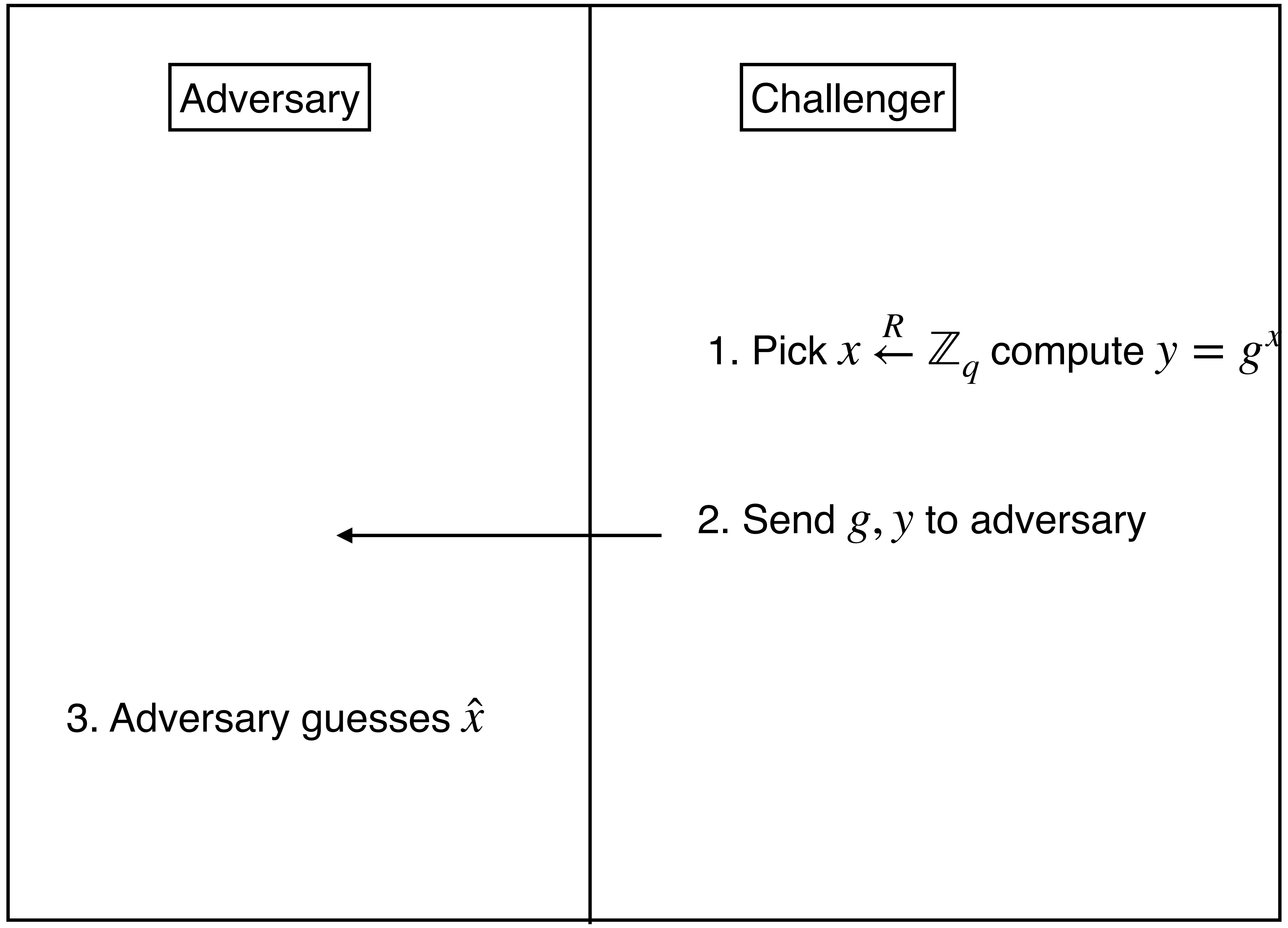
Theorem 6.3. *If a corrupt server tampers with input shares of an honest client, then the honest server always detects such tampering and aborts the protocol with non negligible probability.*

Theorem 6.4. *(Informal) If the output of the protocol is not $ABORT$, then the output is guaranteed to be differentially private and preserve near central accuracy guarantees of semi honest protocols.*

Pederson Commitments

- Let $\mathbb{G}_q$ be a sub group of $\mathbb{Z}_p^*$ with order q where p and q are large primes such that $q \mid p - 1$
- We have a secret $s \in \mathbb{Z}_q$ and we want to commit to it. Then a Peterson commitment to s is given by $c = Com(s, t) = g^s h^t$ where $t \xleftarrow{R} \mathbb{Z}_q$ and g, h are randomly selected generators for $\mathbb{G}_q$
- Given c, a computationally unbounded adversary $\mathscr{A}$ cannot infer any information about s **(Perfectly Hiding)**
- Given c, if adversary $\mathscr{A}$ that can find $(s', t') \neq (s, t)$ such that $Com(s, t) = Com(s', t')$, then $\mathscr{A}$ can solve the DLOG attack game **(Computationally Binding)**

Discrete Log Attack Game



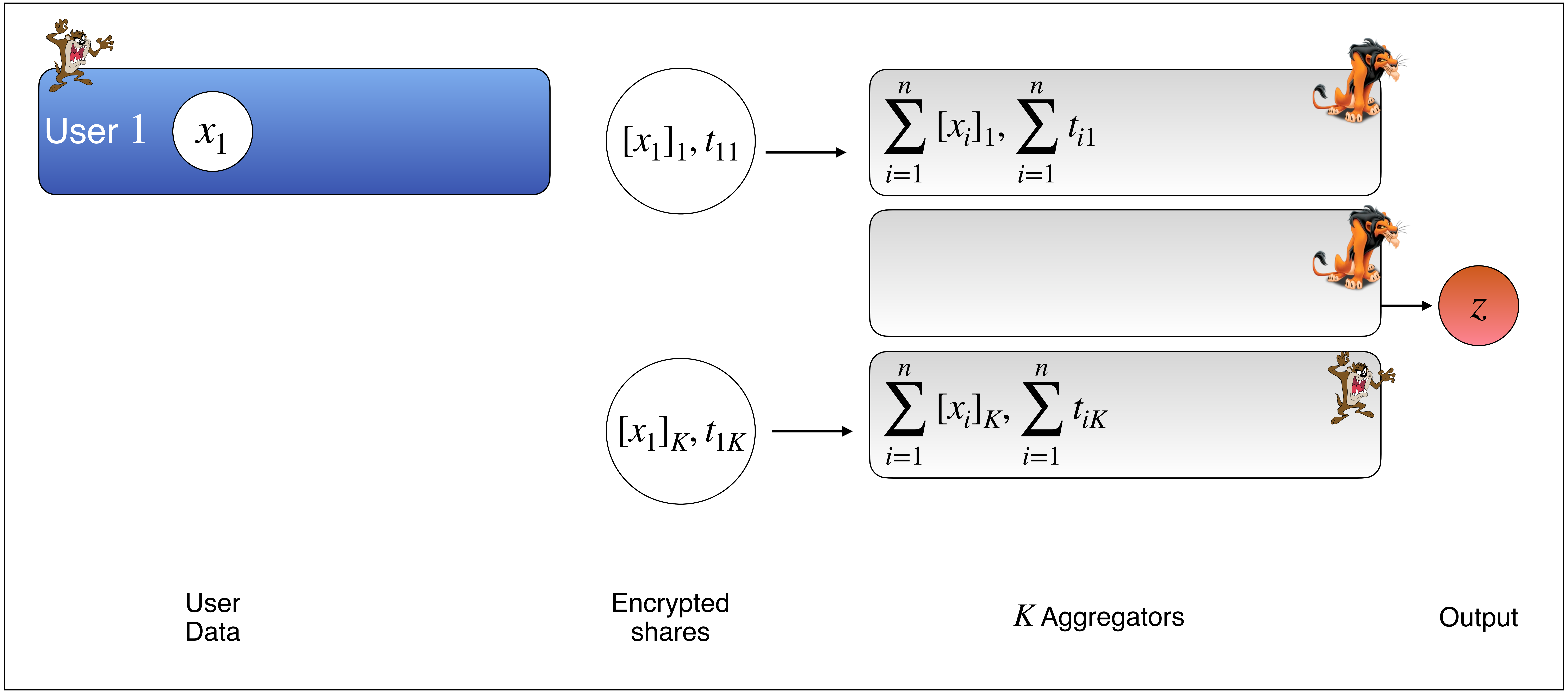
Advantage$(\mathscr{A}, \mathbb{G}_q) := Pr[\hat{x} = x]$

We do not know any PPT algorithm that has non negligible advantage in guessing x for large enough q.

Linearity trick — very useful

- Given $c_1 = Com(s_1, t_1)$ and $c_2 = Com(s_2, t_2)$, then
- $c_1c_2 = Com(s_1 + s_2, t_1 + t_2)$
- Addition in secret space is multiplication in commitment space

Committed Input sharing



Public board for everyone to see

$Com([x_i]_k, t_{ik})$ for all $i \in [n], k \in [k]$

Distributed and verifiable noise generation

- Key idea if the algorithm is linear — then we will use commitments to our advantage and not allow the corrupt parties to deviate
- Binomial noise can be generated using a distributed and linear protocol.

See Dwork et al., 'Our Data, Ourselves'.

Binomial Mechanism

- Given a bit $b_i \in \{0,1\}$from any arbitrary distribution
- And given $c_i \sim Bernoulli(1/2)$
- $z_i = c_i \oplus b_i$ is guaranteed to be $z_i \sim Bernoulli(1/2)$
- Use this along with commitments to get what what we want

Future work

- Can we lose that 1 bit leakage?
- Non linear protocols — could we leverage non-malleable codes for secret sharing ?
- Can we add bias to the noise sampler ?