CCS 2023, Copenhagen, Denmark

Interactive Proofs For Differentially Private Counting

Ari Biswas [1]
Graham Cormode [1, 2]

[1] University Of Warwick
[2] Meta AI

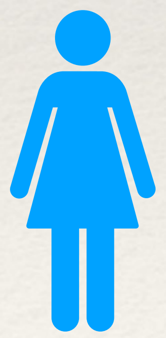
Motivating Problem: Counting

The local government of Wolvercote, a small village in Oxfordshire want to know if they should change public healthcare policy.

In order to gauge public opinion they conduct a survey over the population of the village.

Survey Question

Each resident is asked to vote for a single policy only



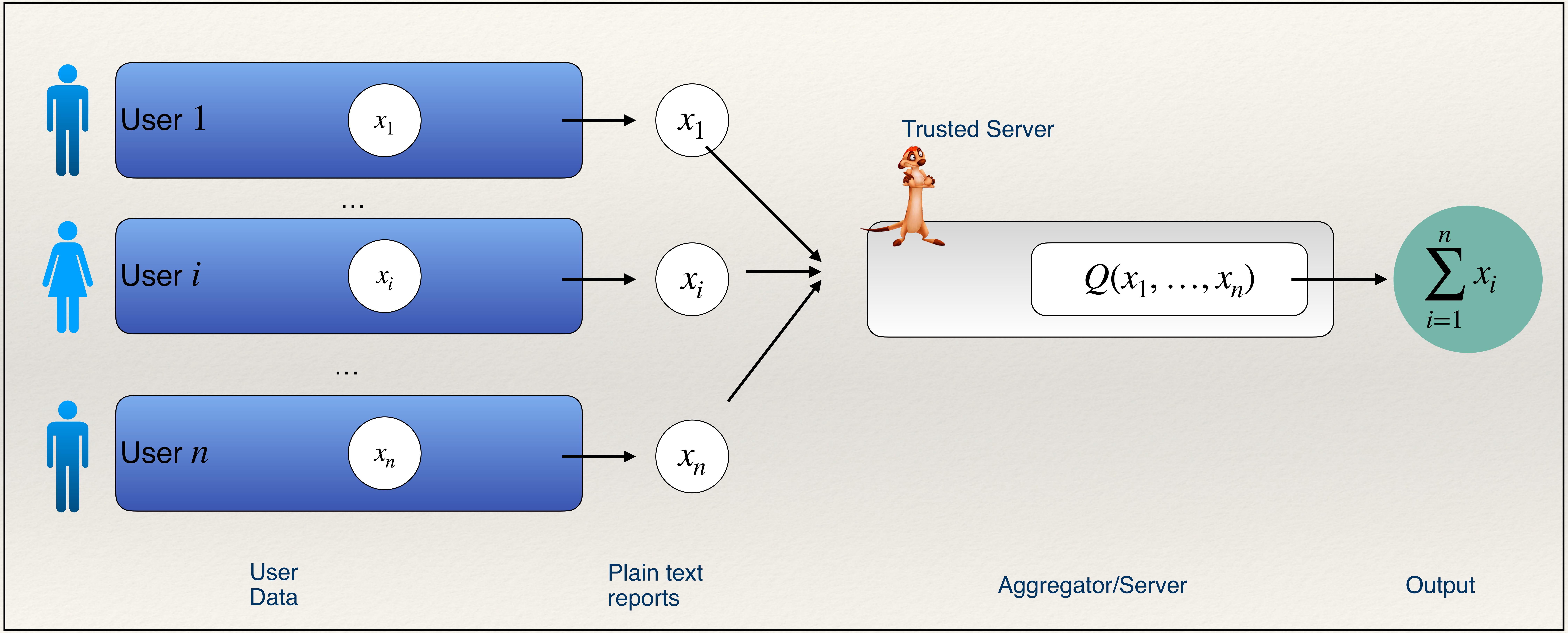
i

1	2	3	4
0	0	1	0

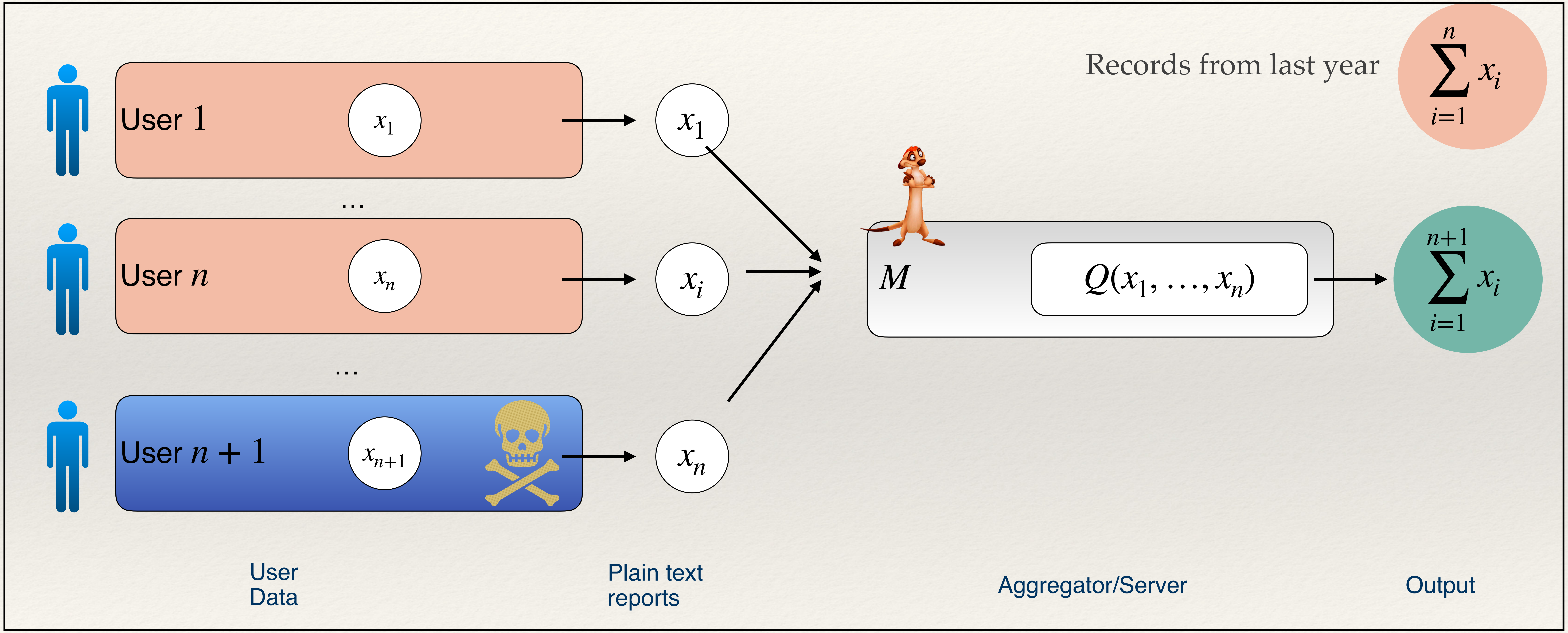
x_i

1: Mandatory Vaccination
2: Increase Pay Towards Healthcare workers
3: Decrease Taxes Towards Healthcare
4: Increase Taxes Towards Healthcare

An Ideal Solution



A New Person Moves in

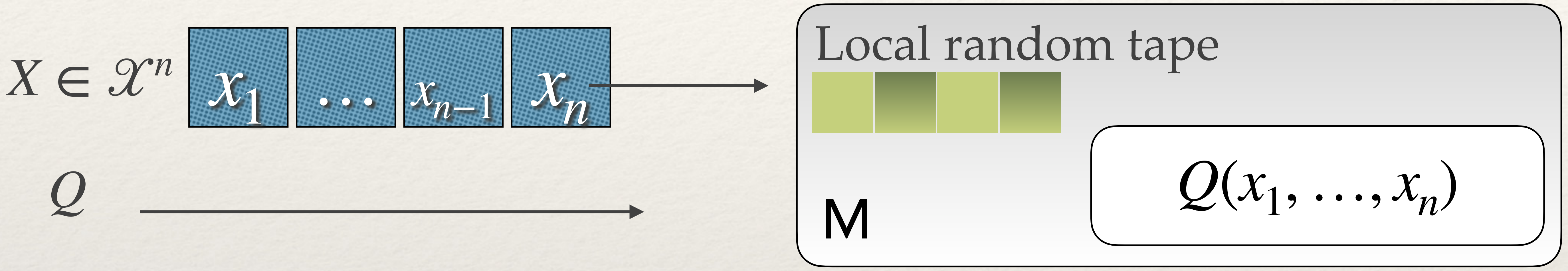


Randomness To The Rescue

- In this scenario, there is no deterministic algorithm that can help prevent information leakage about the n'th users value.
- Thus we **MUST** randomness to obfuscate information about the new user.

(ϵ, δ)-Differential Privacy (DP)

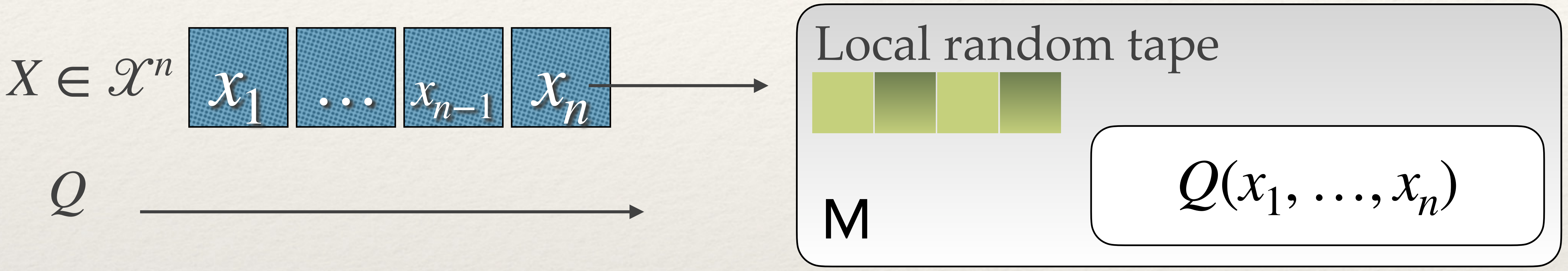
An algorithm $\mathsf{M} : \mathcal{X}^n \times \mathcal{Q} \to \mathcal{Y}$ for releasing $Q(X)$



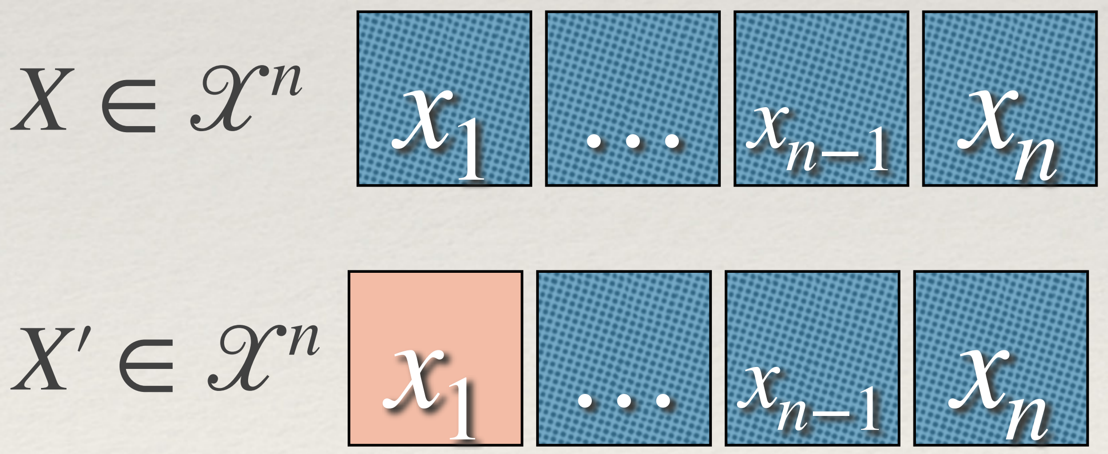
$\mathsf{M}(X, Q)$ Is a distribution

(ϵ, δ)-Differential Privacy (DP)

An algorithm $\mathsf{M} : \mathcal{X}^n \times \mathcal{Q} \to \mathcal{Y}$ for releasing $Q(X)$

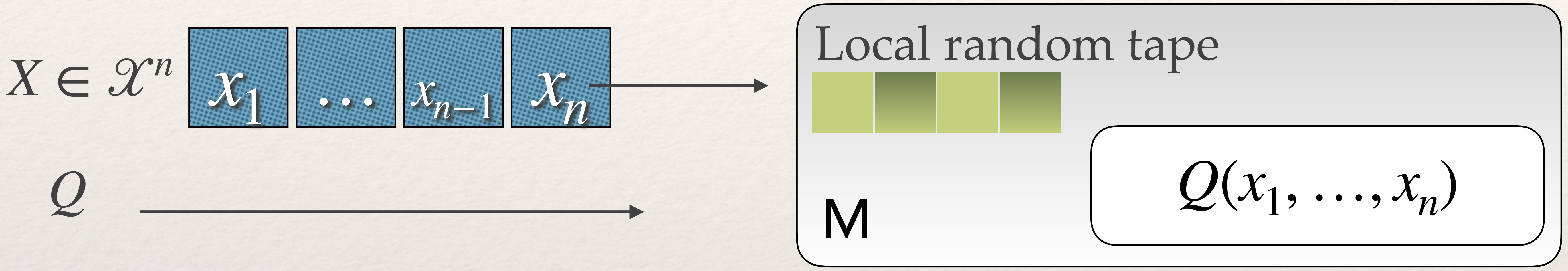


For any neighbouring datasets $X \sim X'$ i.e datasets that differ by just one element



(ϵ, δ)-Differential Privacy (DP)

An algorithm $\mathsf{M} : \mathcal{X}^n \times \mathcal{Q} \rightarrow \mathcal{Y}$ for releasing $Q(X)$



M is said to be $(\epsilon, \delta)-$Differentially Private if for any subset $T \subseteq \mathcal{Y}$

For any neighbouring datasets $X \sim X'$ i.e datasets that differ by just one element

$X \in \mathcal{X}^n$: x_1 | $\dots$ | x_{n-1} | x_n

$X' \in \mathcal{X}^n$: x_1 | $\dots$ | x_{n-1} | x_n

$$\Pr_{y \xleftarrow{\$} M(X,Q)} [y \in T] \leq e^{\epsilon} \Pr_{y \xleftarrow{\$} M(X',Q)} [y \in T] + \delta$$

Utility Of A DP Algorithm

An algorithm $\mathsf{M} : \mathcal{X}^n \times Q \to \mathcal{Y}$ for releasing a DP version of $y = Q(X)$ where $(\mathcal{Y}, d)$ is a metric space we define utility

$$\text{Error} = \mathbb{E}_{\hat{y} \xleftarrow{\$} \mathsf{M}(X,Q)} \left[d(\hat{y}, y) \right]$$

Candidate metrics

$\mathcal{Y} = \mathbb{R}^d \qquad d(x, y) = \quad x - y \quad {}_1$

$\mathcal{Y} = \mathbb{Z}_q^d \qquad d(x, y) = \quad x - y \quad {}_2^2$

$d(x, y) = \quad x - y \quad {}_\infty$

If we draw a sample from $\mathsf{M}(X, Q)$, then on average how far is that sample from the true untampered answer.

DP Counting

$$\boxed{Q(x_1, \ldots, x_n)} = \sum_{i=1}^{n} x_i$$

M

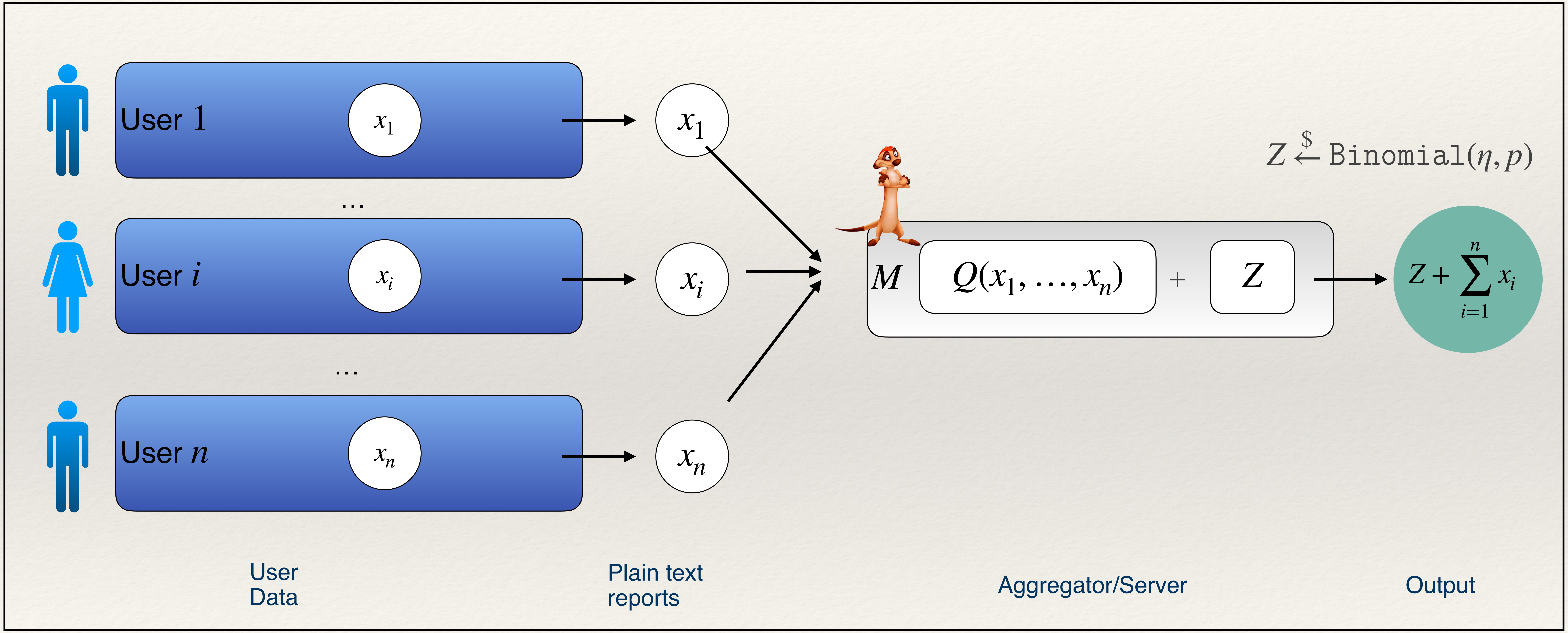
$\boxed{Q(x_1, \ldots, x_n)} + \boxed{\texttt{Laplace}(b)} \longrightarrow \boxed{M(X; Q)}$ $\quad \epsilon = \frac{1}{b}, \delta = 0$

$\boxed{Q(x_1, \ldots, x_n)} + \boxed{\texttt{Binomial}(\eta, p)} \longrightarrow \boxed{M(X; Q)}$ $\quad \epsilon = \sqrt{\frac{50}{\eta p} \ln \frac{2}{\delta}}$

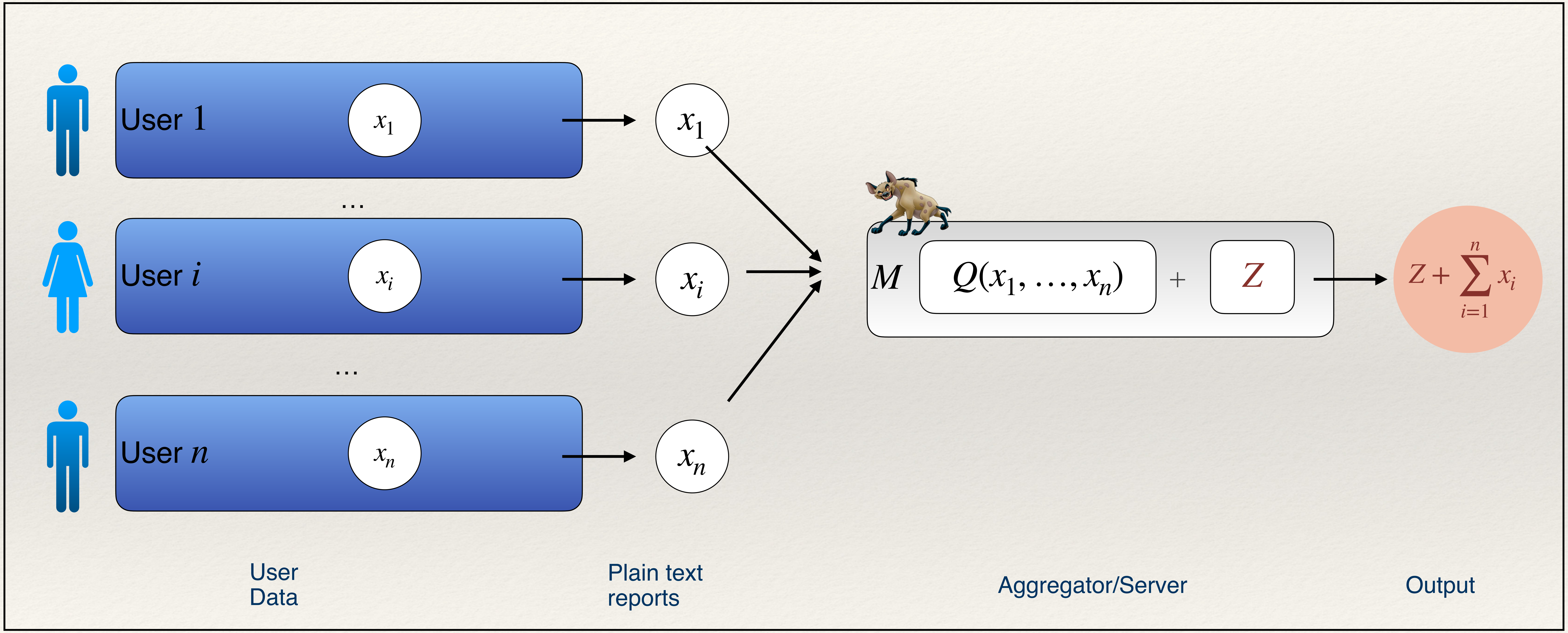
$\ldots \qquad \ldots$

$\boxed{Q(x_1, \ldots, x_n)} + \boxed{\texttt{Gaussian}(0, \sigma)} \longrightarrow \boxed{M(X; Q)}$ $\quad \epsilon = \sqrt{2 \ln \frac{1.25}{\delta \sigma^2}}$

Back To Our Ideal World



What If We Cannot Trust The Server ?



What Do We Want

- We want outputs to be differentially private
- However, we also want the output to be <u>reliable</u> i.e, by that we mean any error in the output must come as a result of DP noise and that only.

Need Some Crypto

Commitments

Two stage interactive protocol between a Committer and a Receiver

Commit Phase

Committer

Receiver

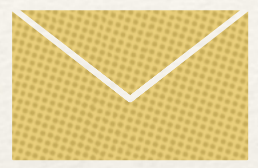
Reveal Phase

Commit Phase

Committer



Key


Message


Locked Box/
Commitment

Receiver

Hiding Property: The Receiver cannot tell what is inside the box.

Reveal Phase

Committer

Key

Message

Locked Box / Commitment

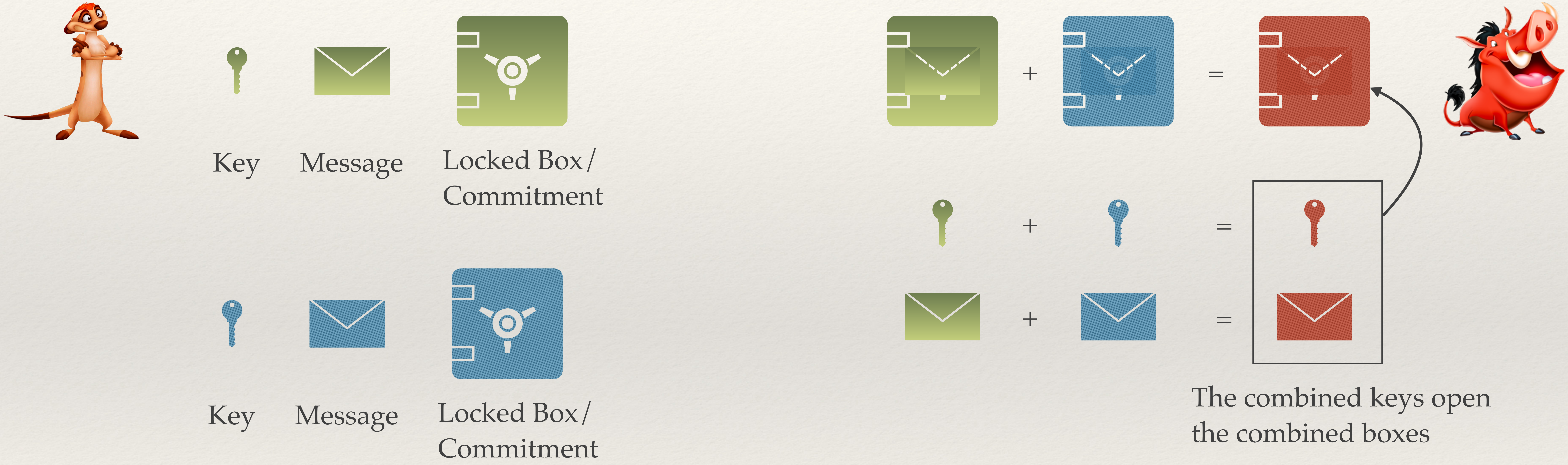
Binding Property: The sender cannot find a different *key, message* pair to open the box correctly

Receiver

Use key to open box and reveal message and check

=

Homomorphic Commitments



Disjunctive OR Arguments

Committer

Key Message Locked Box / Commitment

Receiver

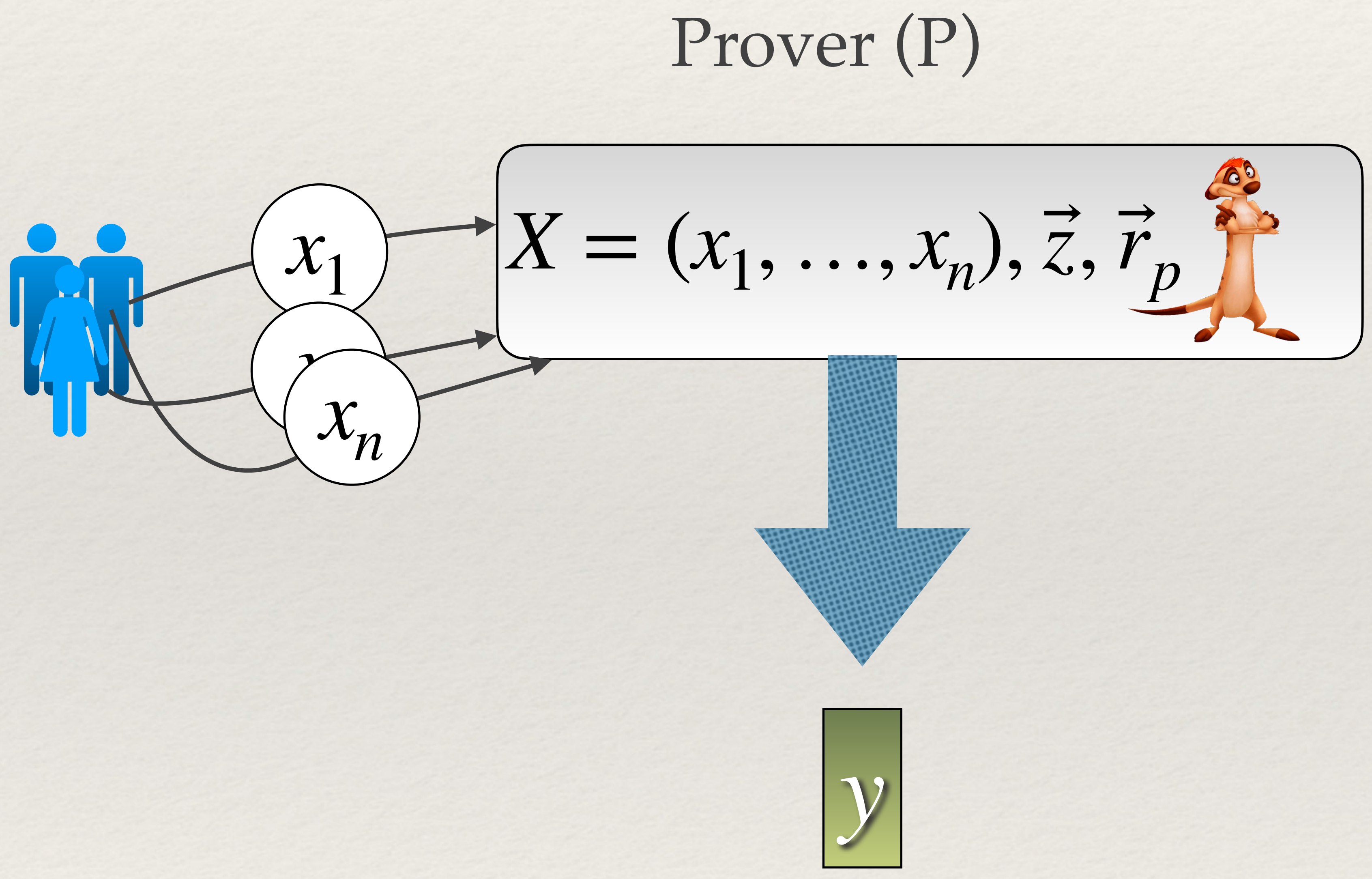
The prover can convince the receiver that the message is either 0 or 1 without revealing which one it is

Quick Recap

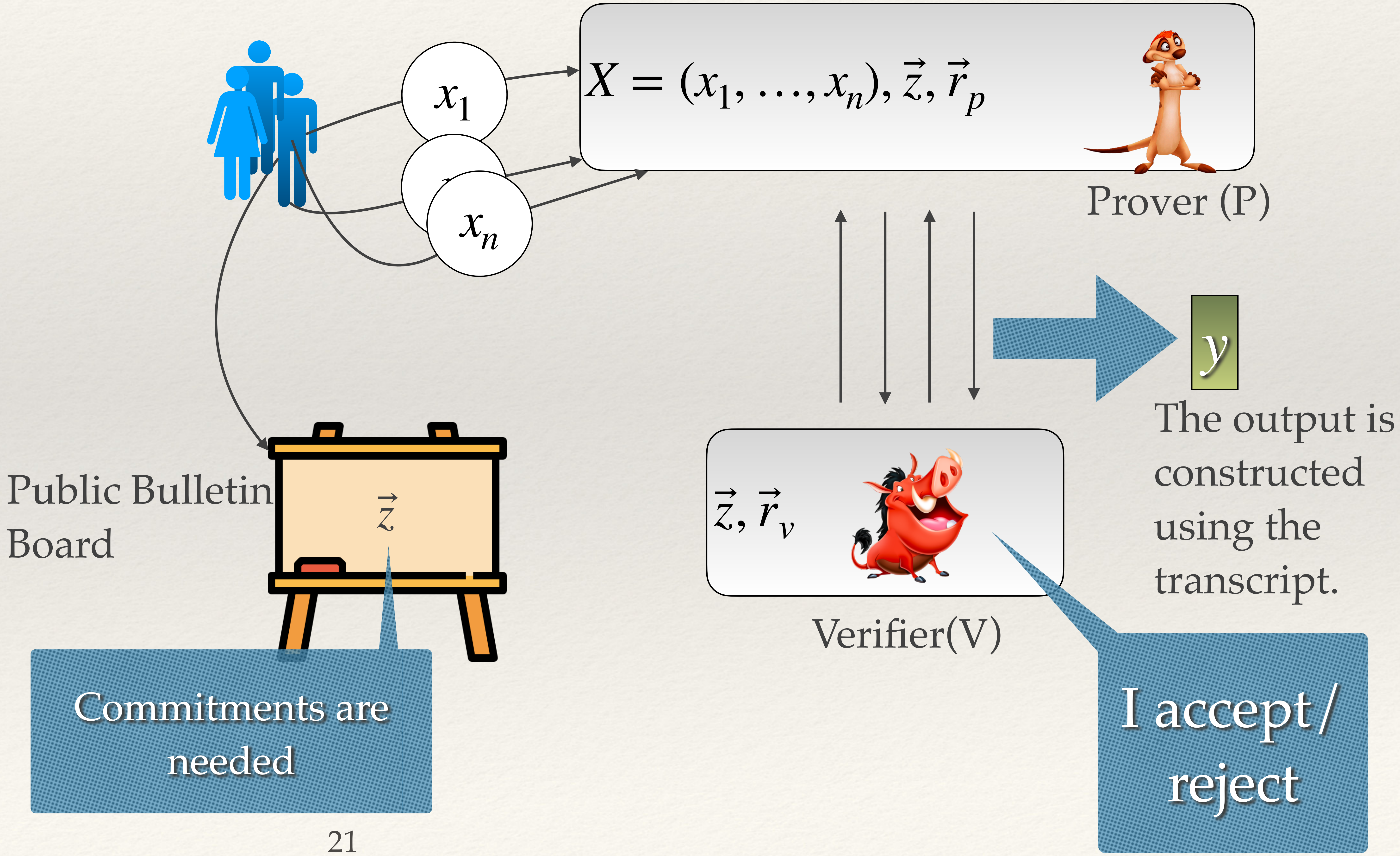
- We have commitments that are homomorphic and support OR arguments.

Verifiable - The Setting

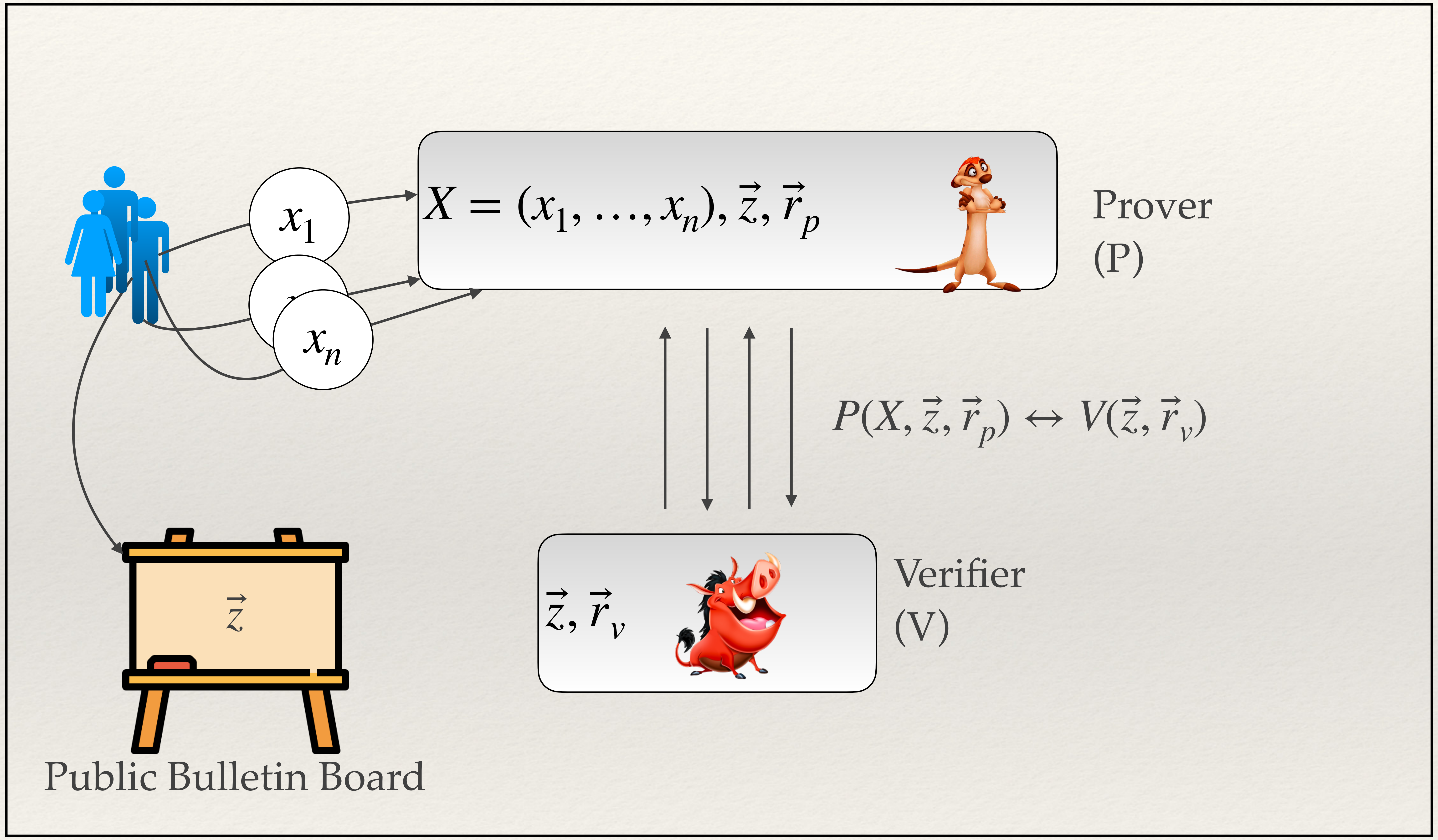
Un-verifiable DP



Verifiable DP



Verifiable DP

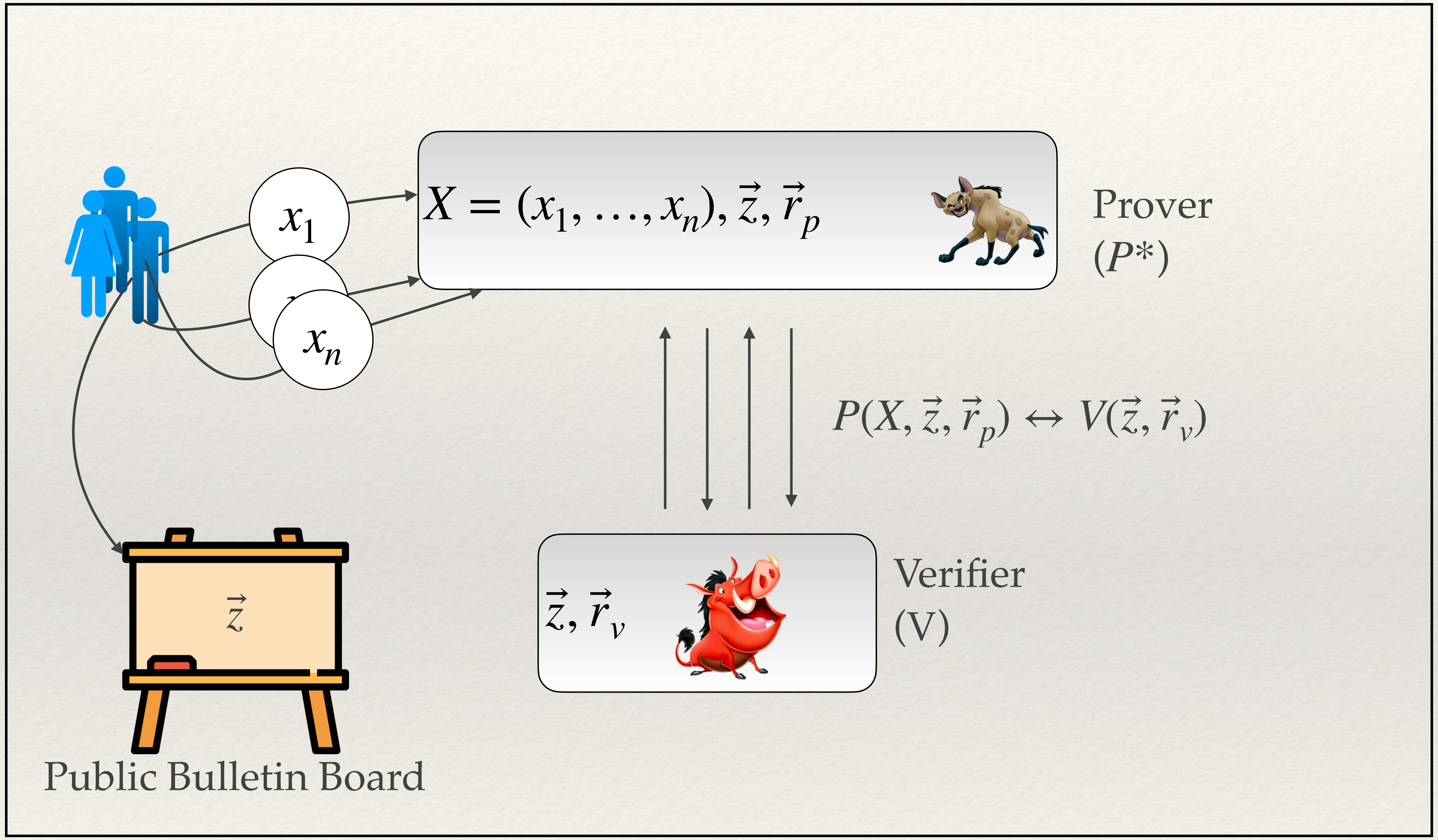


Completeness:

If both the prover and the verifier are honest, then $y \xleftarrow{\$} M(X, Q)$ and

$\Pr[\mathtt{Verify}(P \leftrightarrow V) = 1] = 1$

Verifiable DP

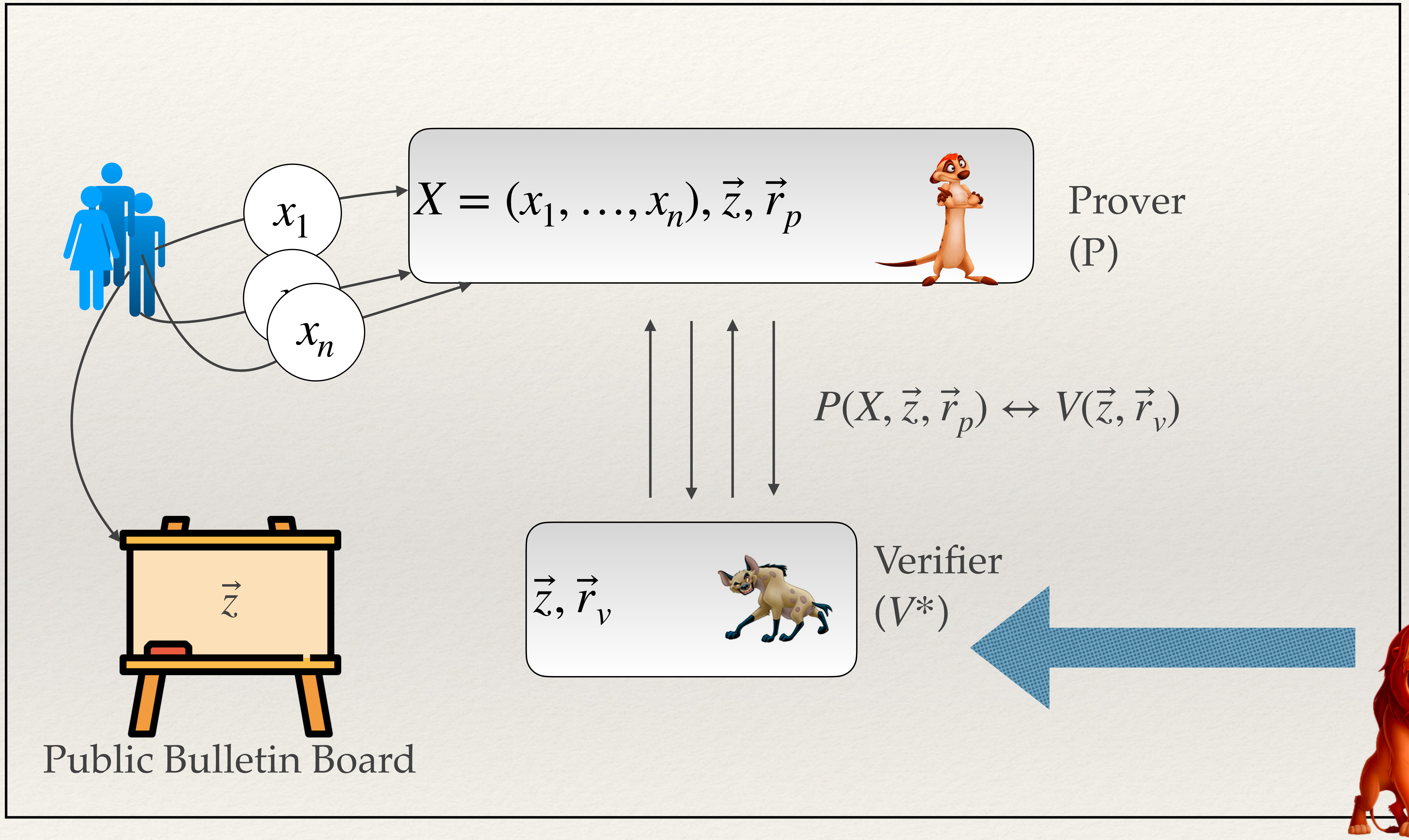


Soundness

For any cheating prover P^* that samples y from a distribution $\mathcal{D}$ such that $\mathsf{TV}(\mathsf{M}(X, Q), \mathcal{D}) > \mu(\kappa)$

$\Pr[\mathtt{Verify}(P^* \leftrightarrow V) = 1] \leq 1/3$

Verifiable DP

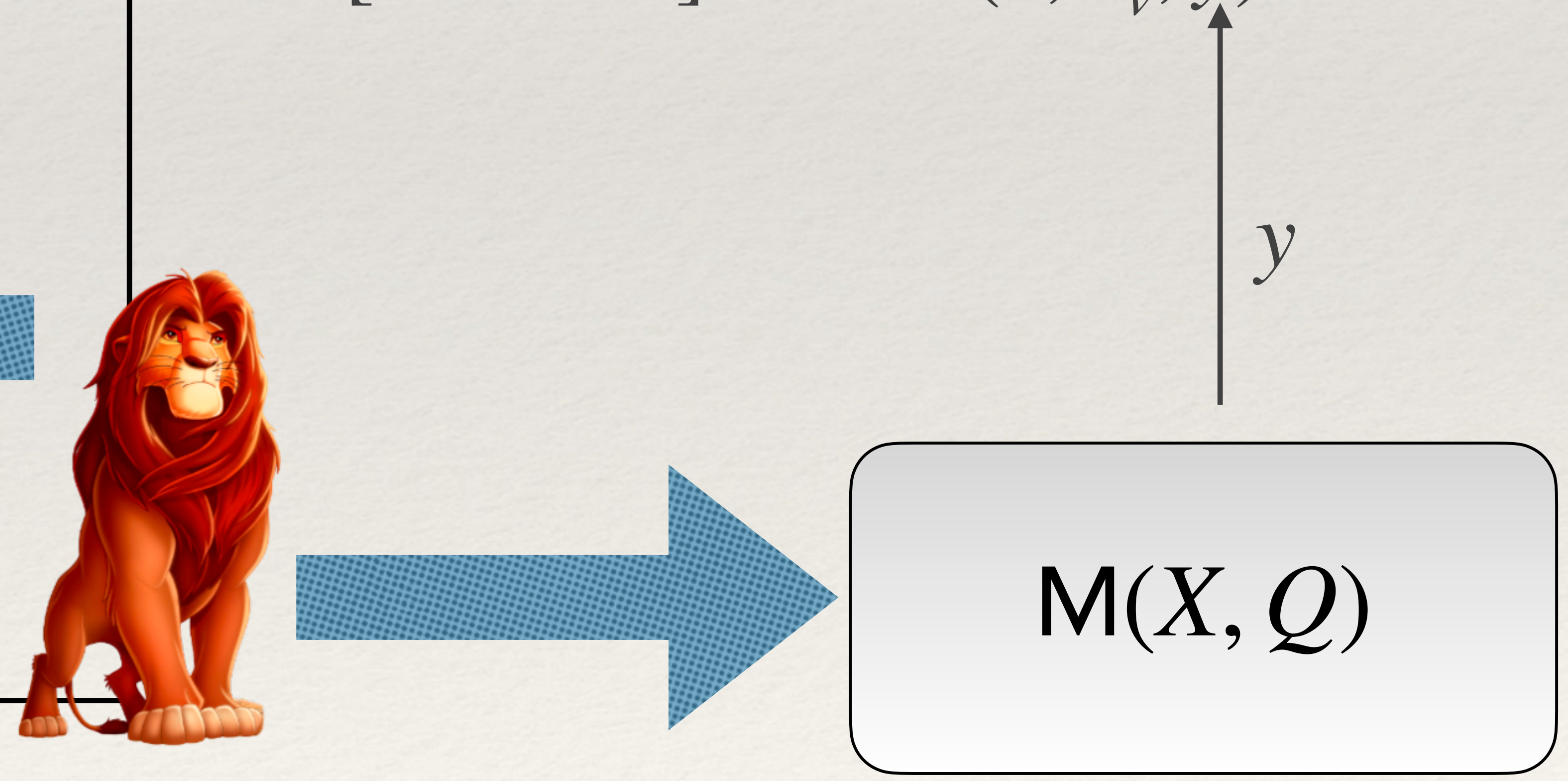


Cheating Verifier Zero Knowledge

When $y \xleftarrow{\$} M(X, Q)$, for any cheating verifier V^* there exists a PPT algorithm

`Sim`  that has oracle access to V^*

$\texttt{View}[P \leftrightarrow V^*] \equiv \texttt{Sim}(\vec{z}, \vec{r}_v, y)$



The Soundness/ZK conflict

Prover (P) Verifier(V)

$X, \vec{z}, \vec{r}_p$ $\rightleftarrows$ $\vec{z}, \vec{r}_v$

$$Z \xleftarrow{\$} \texttt{Binomial}(\eta, \frac{1}{2})$$

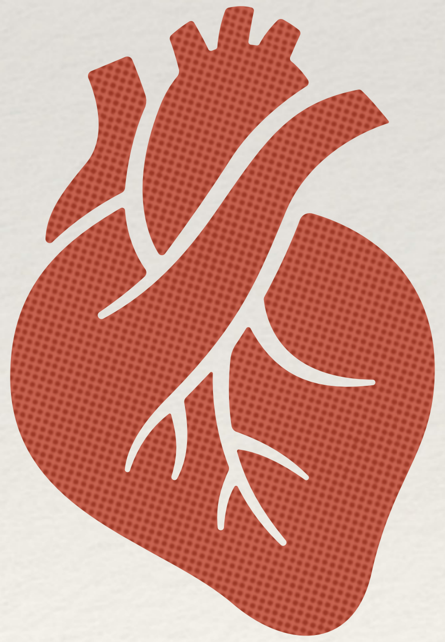
$$y = \boxed{Q(x_1, \ldots, x_n)} + \boxed{Z} \longrightarrow \boxed{M(X; Q)}$$

The output is a function of the provers local randomness. However the prover cannot ever reveal this randomness to the verifier as it would compromise DP.

The prover must find a way to prove that Z was sampled from the right distribution without ever revealing any information about Z.

However, we also need some shared information (like say public randomness) for the verifier to be able to confident that Z is sampled correctly.

THE HEART OF THE PROBLEM



*Not to be confused with Proof Of Knowledge

** The noise used is not **pseudorandom** noise either

Non Private Counting

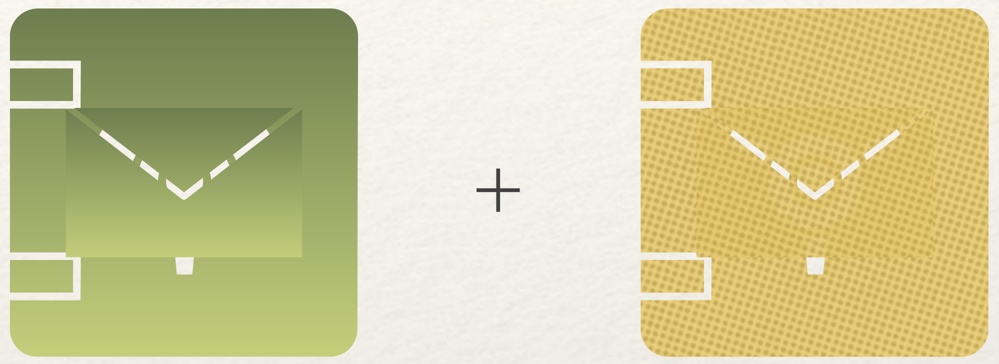
Server/Prover

$(x_1, r_1) \ldots, (x_n, r_n)$

Citizen n's private key used to create commitments

Citizen 1's private vote

$\mathtt{Com}(x_1, r_1), \ldots, \mathtt{Com}(x_n, r_n)$



Non Private Counting

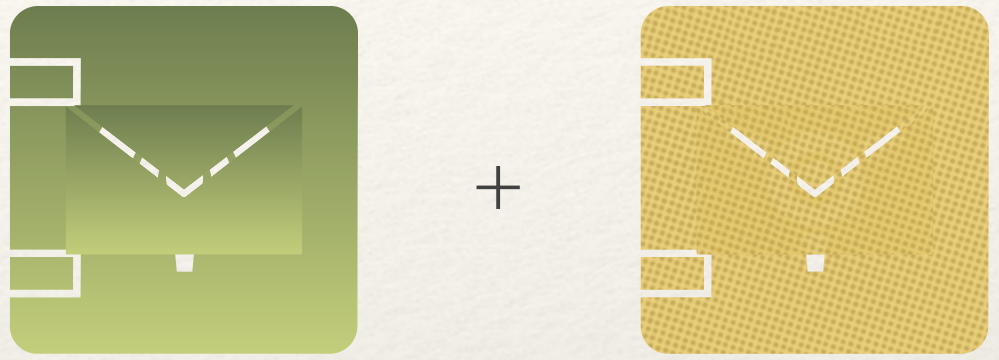
Server / Prover

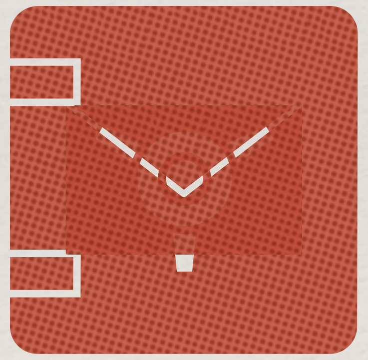
$(x_1, r_1) \ldots, (x_n, r_n)$

Citizen n's private key used to create commitments

Citizen 1's private vote

$\texttt{Com}(x_1, r_1), \ldots, \texttt{Com}(x_n, r_n)$





Non Private Counting

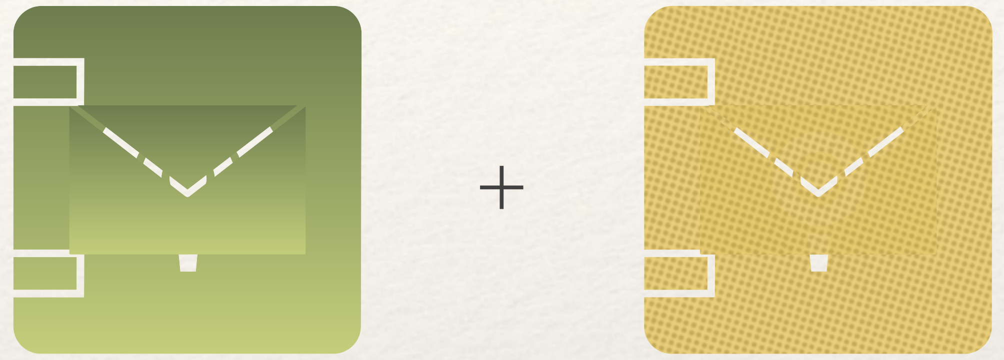
Server/Prover

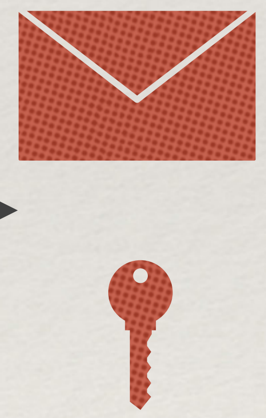
$(x_1, r_1)\ldots, (x_n, r_n)$

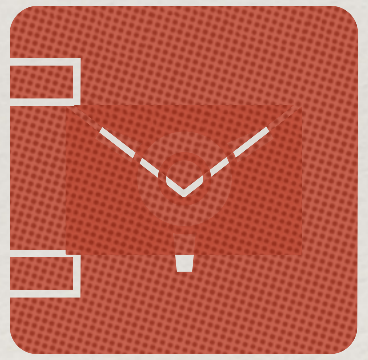
Citizen n's private key used to create commitments

Citizen 1's private vote

$\texttt{Com}(x_1, r_1), \ldots, \texttt{Com}(x_n, r_n)$

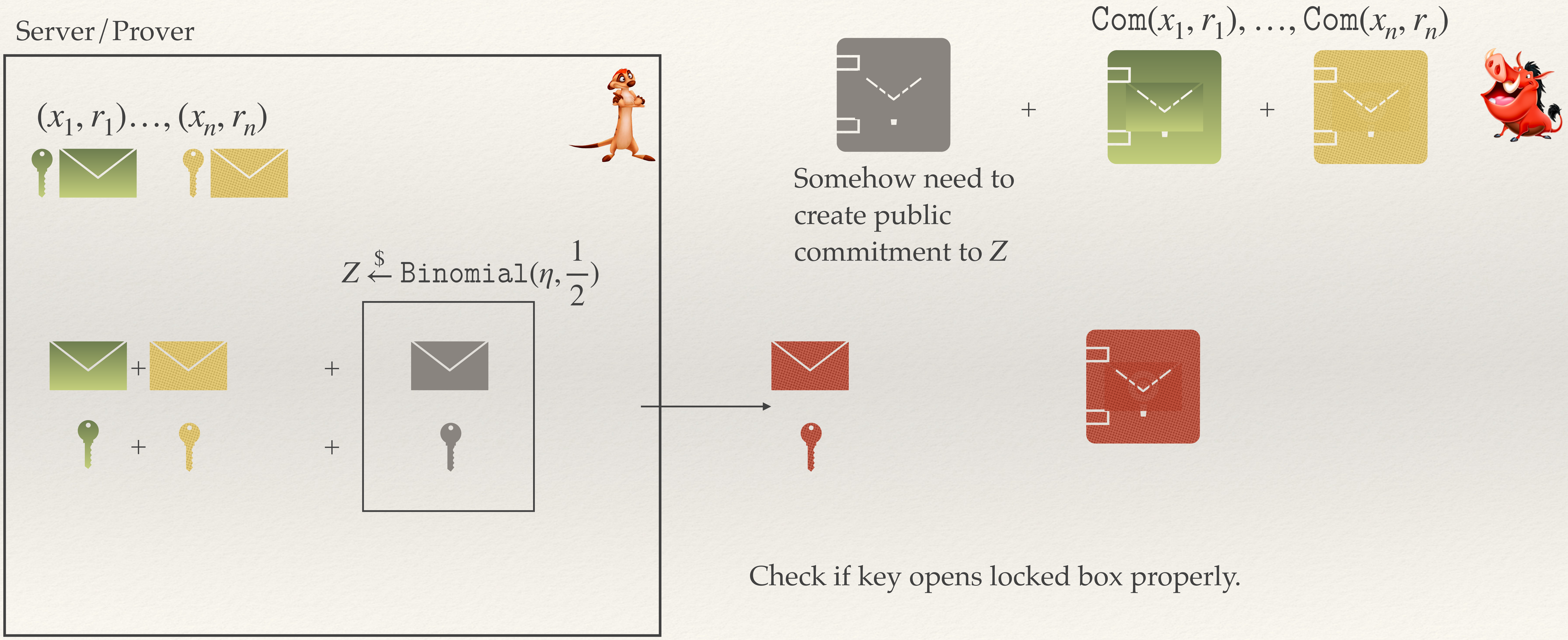






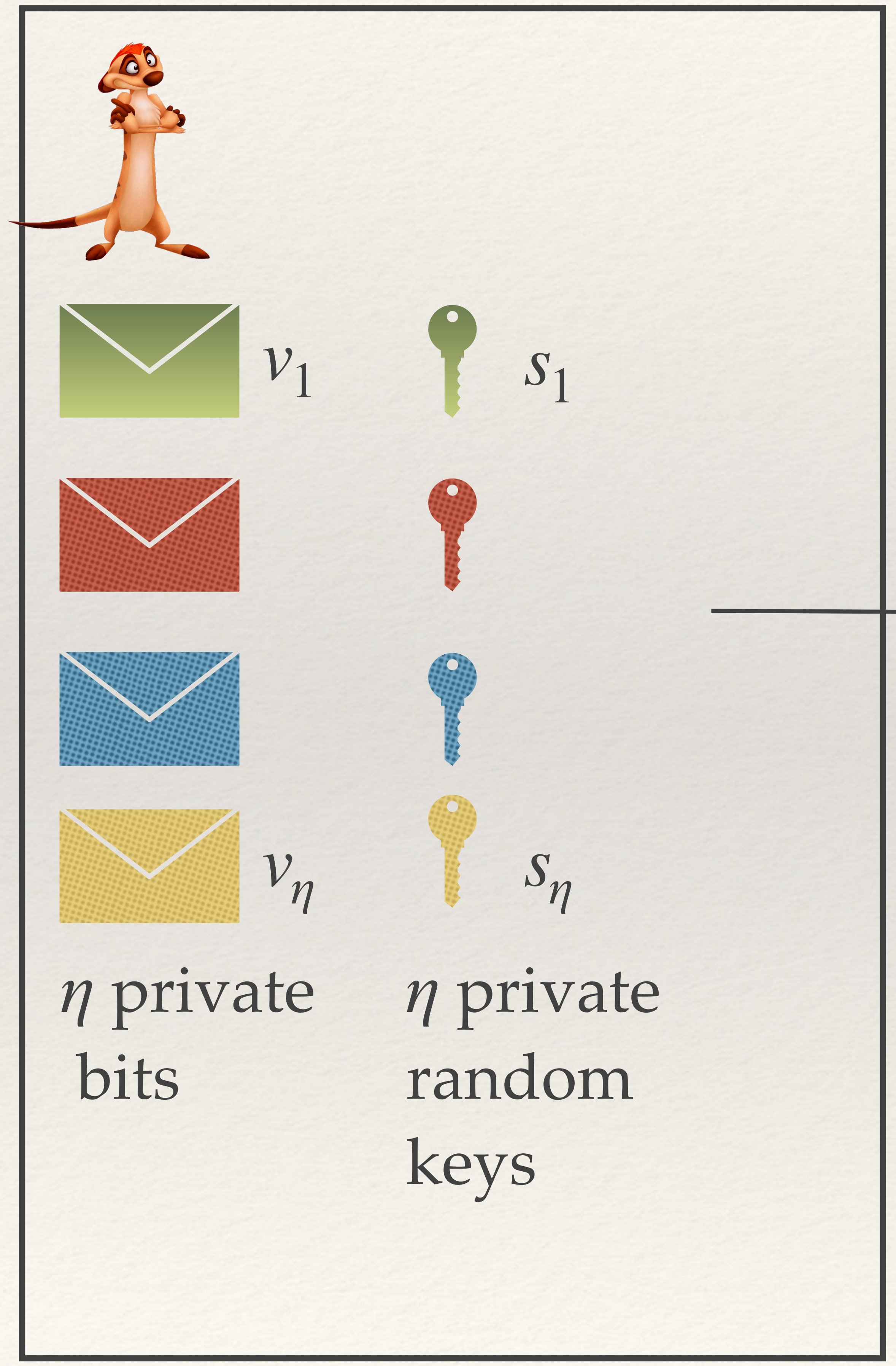
Check if key opens locked box properly.

Verifiable DP counting - Essence



A Simple Trick

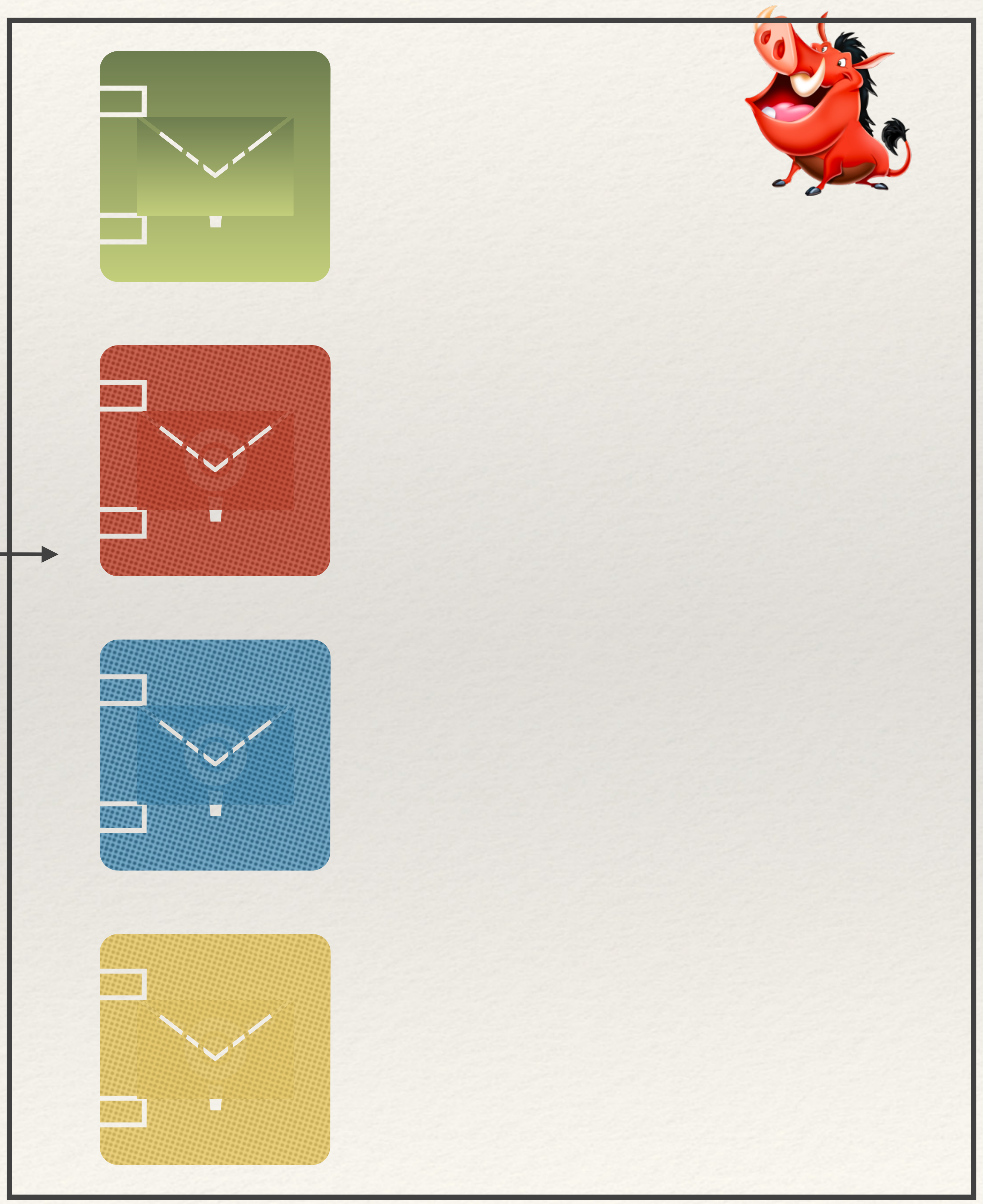
Server / Prover



Note we cannot say anything about the distribution from which these bits are being sampled.

All the verifier knows is that these boxes are a commitment to a bit.

Verifier

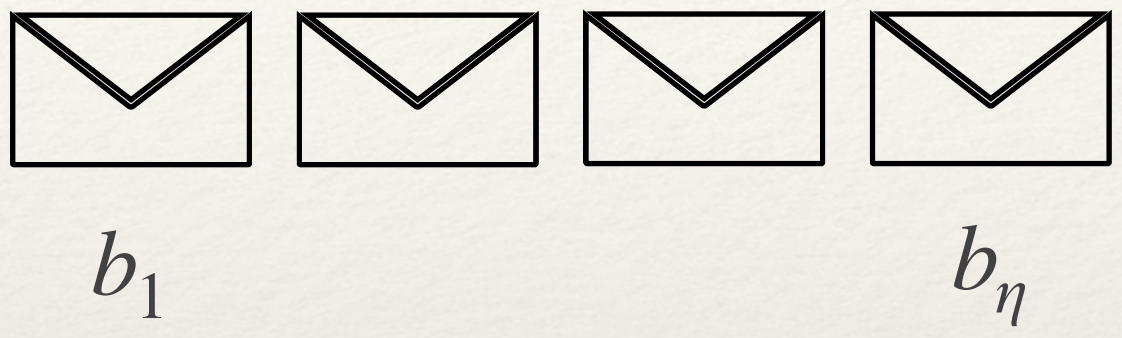


A Simple Trick

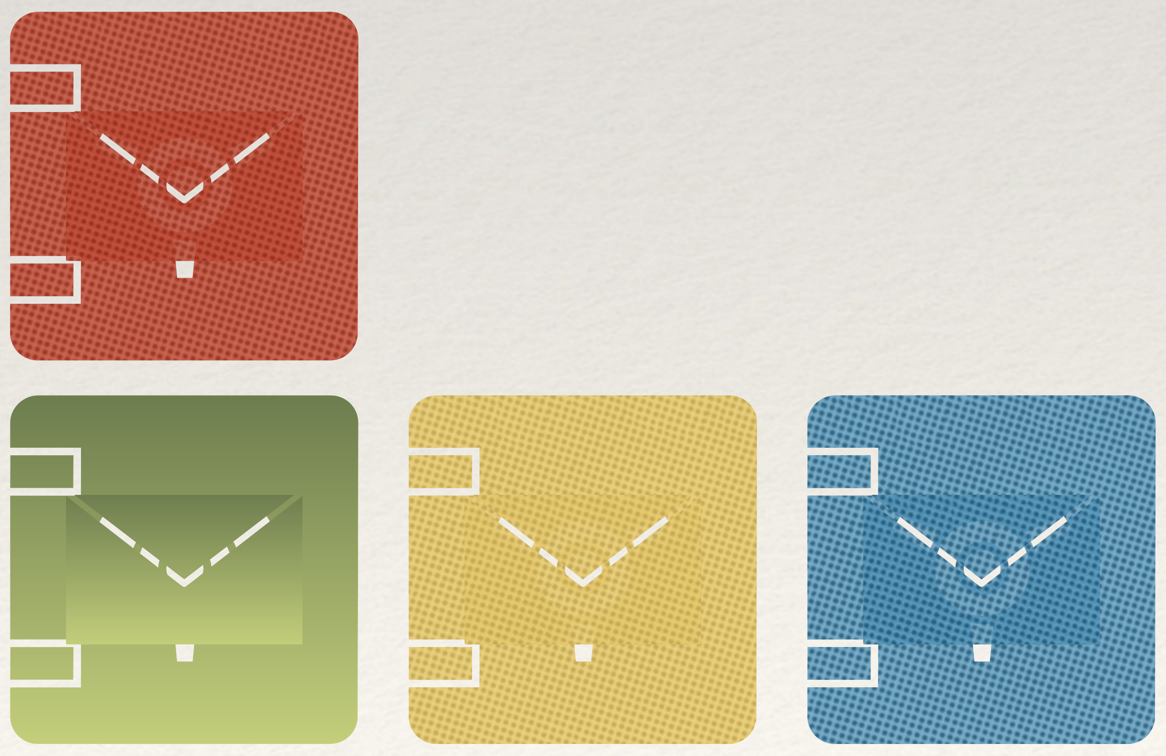
Server / Prover

Sample η bits

Verifier generates η public unbiased coins



Verifier

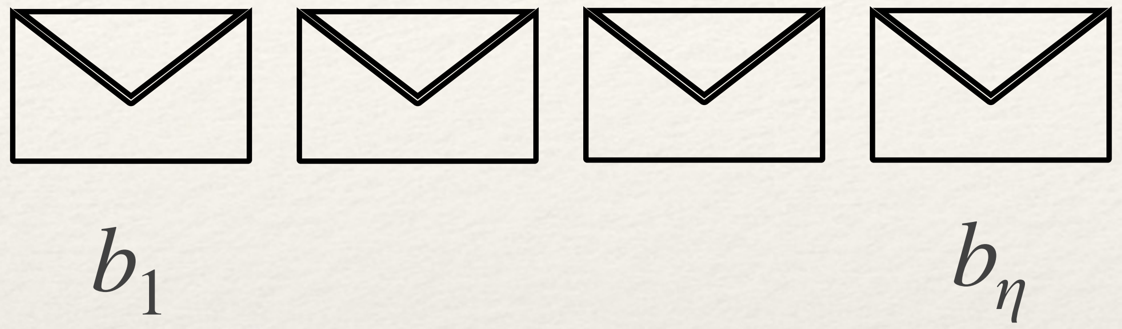


The Final Trick

Server / Prover

Sample η bits

Verifier generates η public unbiased coins



If $b_i = 1$ then set $v_i = 1 - v_i$

Otherwise, leave v_i unchanged.

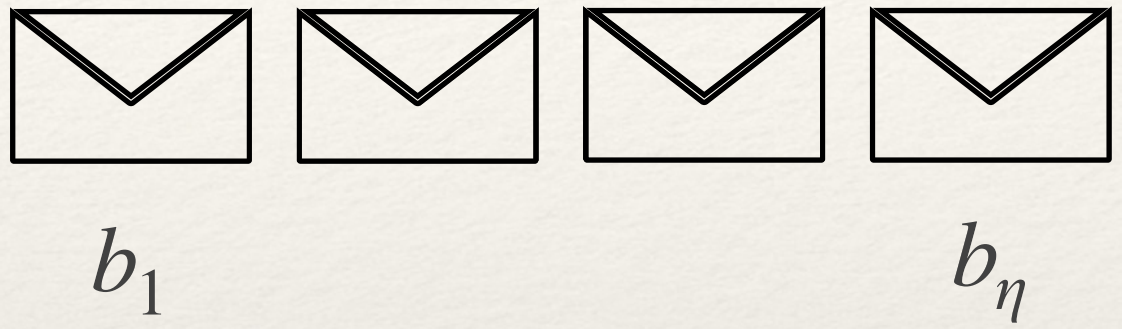
Verifier

The Final Trick

Server/Prover

Sample η bits

Verifier generates η public unbiased coins



If $b_i = 1$ then set $v_i = 1 - v_i$ and $s_i = 1 - s_i$

Otherwise, leave v_i and s_i unchanged.

Observation 1:
The updates are LINEAR conditioned on b_i

Without ever seeing v_i the verifier can update
$\texttt{Com}(v_i, s_i) = \texttt{Com}(1,1) - \texttt{Com}(v_i, s_i)$

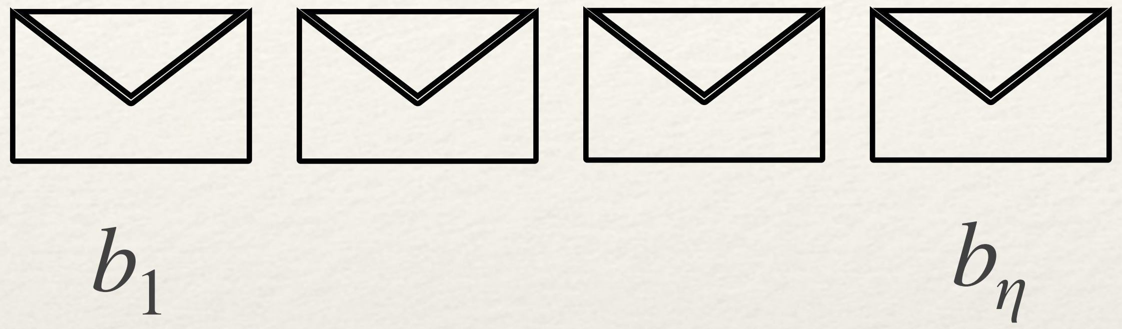
Verifier

The Final Trick

Server / Prover

Sample η bits

Verifier generates η public unbiased coins



If $b_i = 1$ then set $v_i = 1 - v_i$

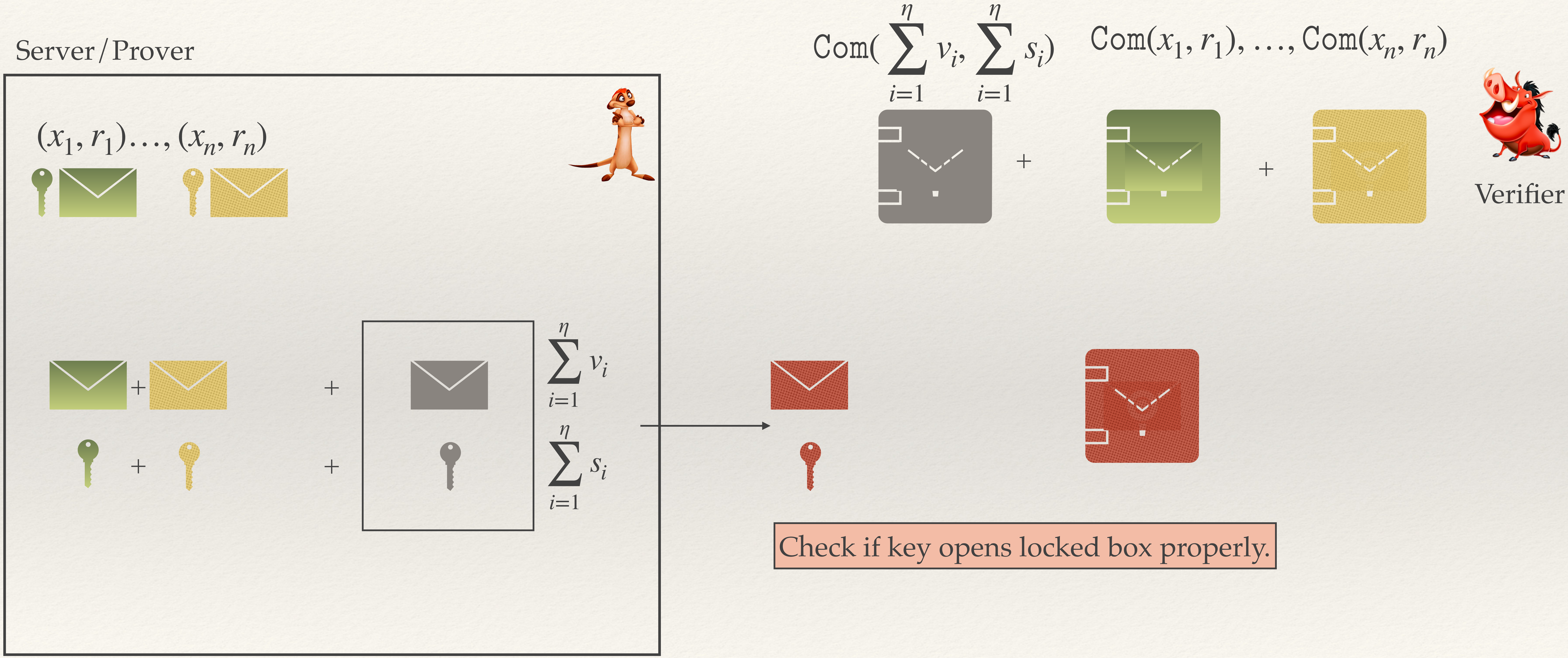
Otherwise, leave v_i unchanged.

Observation 2:
The above conditional statement is equivalent to
$v_i = v_i \oplus b_i$

This forces the provers bit to have the correct distribution.

Verifier

Final Check



Server / Prover
$\mathtt{Com}(\sum_{i=1}^{\eta} v_i, \sum_{i=1}^{\eta} s_i)$
$\mathtt{Com}(x_1, r_1), \ldots, \mathtt{Com}(x_n, r_n)$
$(x_1, r_1) \ldots, (x_n, r_n)$
+
+
+
+
+
+
$\sum_{i=1}^{\eta} v_i$
$\sum_{i=1}^{\eta} s_i$
Verifier
Check if key opens locked box properly.