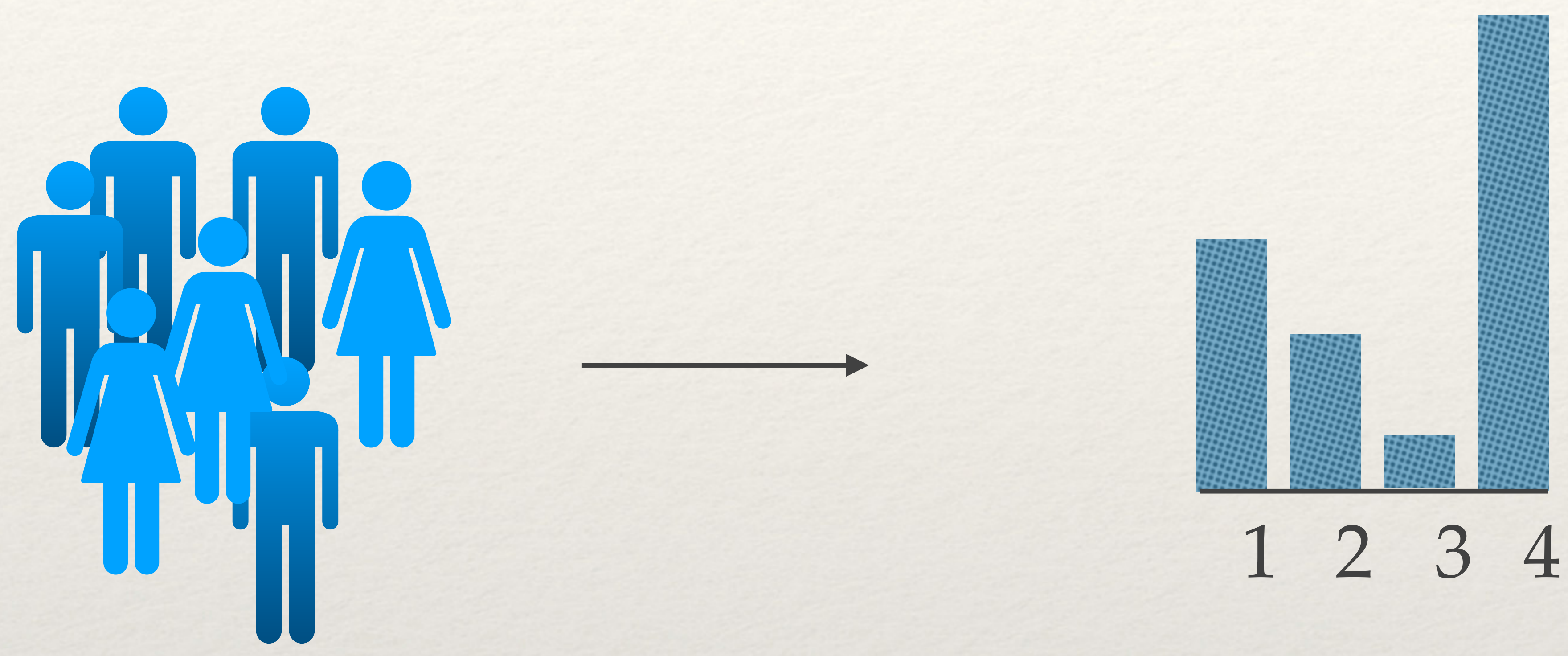
To Appear In CCS 2023

Interactive Proofs For Differential Privacy

Ari Biswas
Graham Cormode

Motivating Problem: Counting

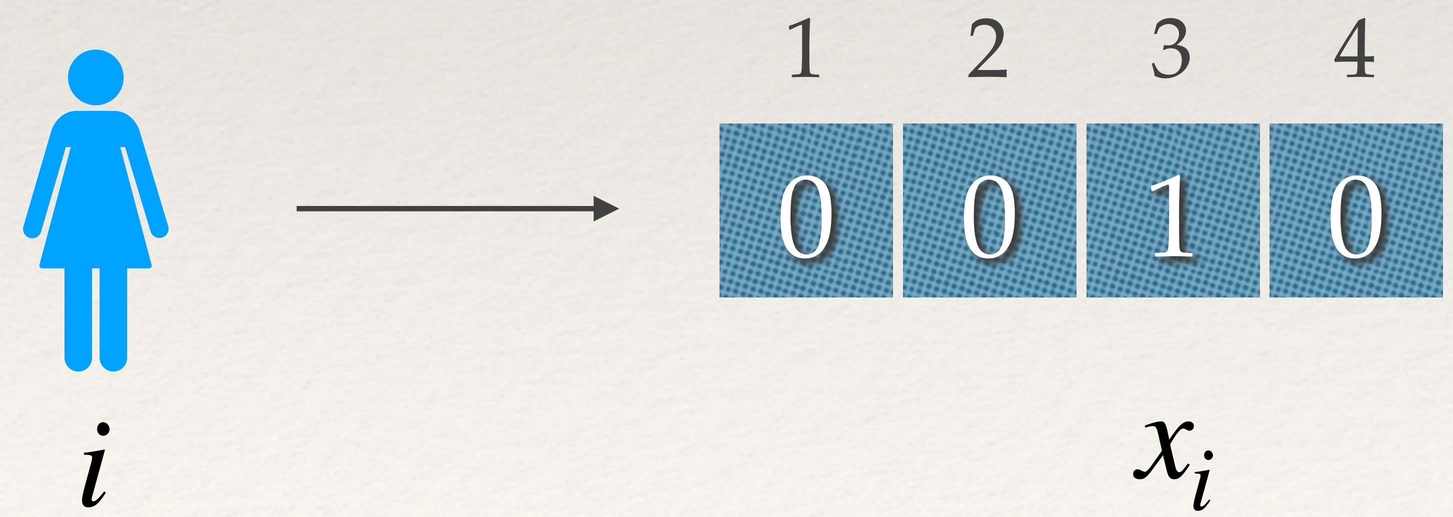


The local government of Wolvercote, a small village in Oxfordshire want to know if they should change public healthcare policy.

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

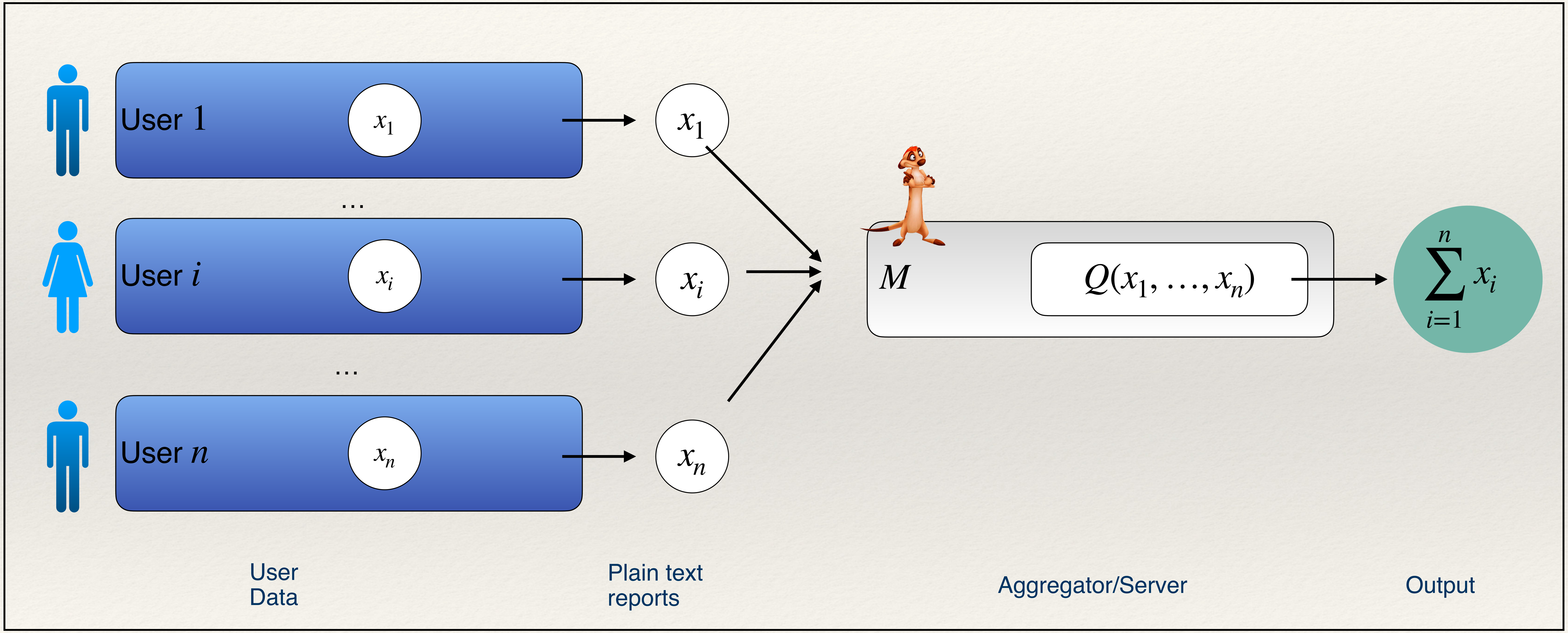
Survey Question

Each resident is asked to vote for a single policy only

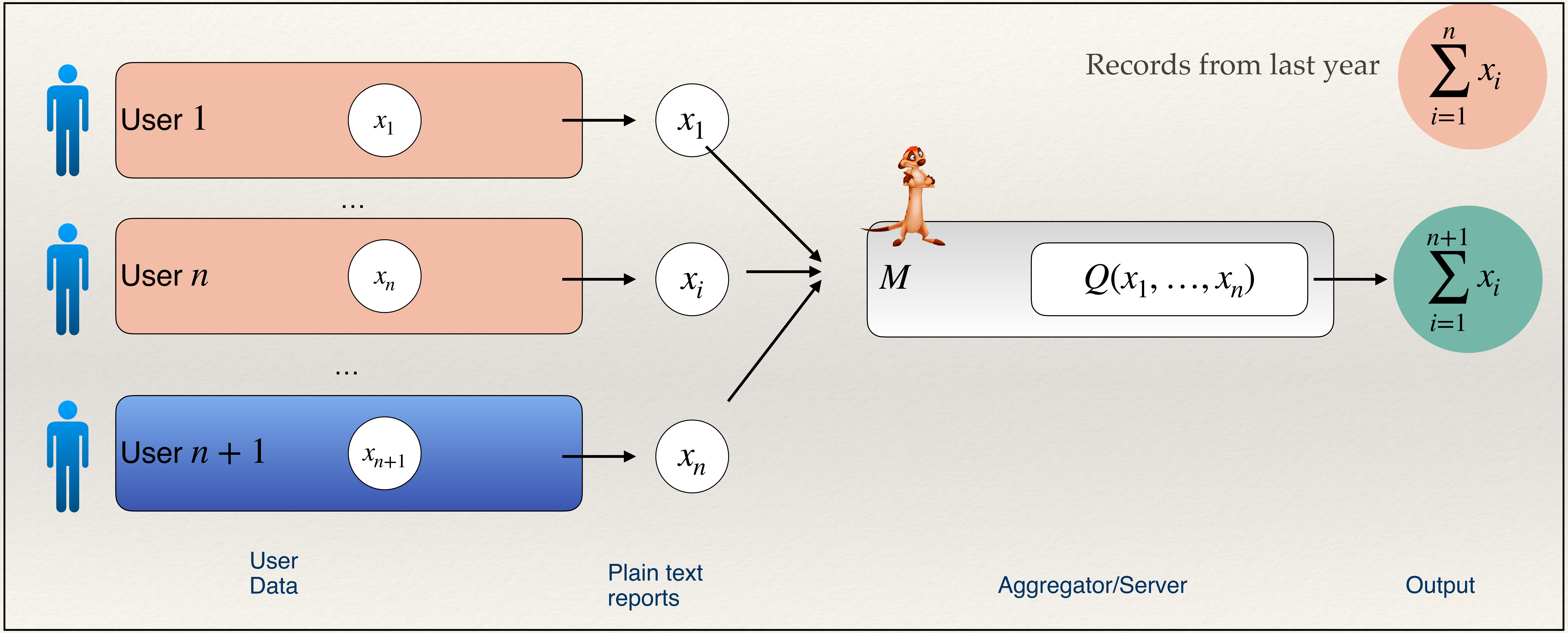


1: Mandatory Vaccination
2: Increase Pay Towards Healthcare workers
3: Decrease Taxes Towards Healthcare
4: Increase Taxes Towards Healthcare

An Ideal Solution



A New Person Moves in

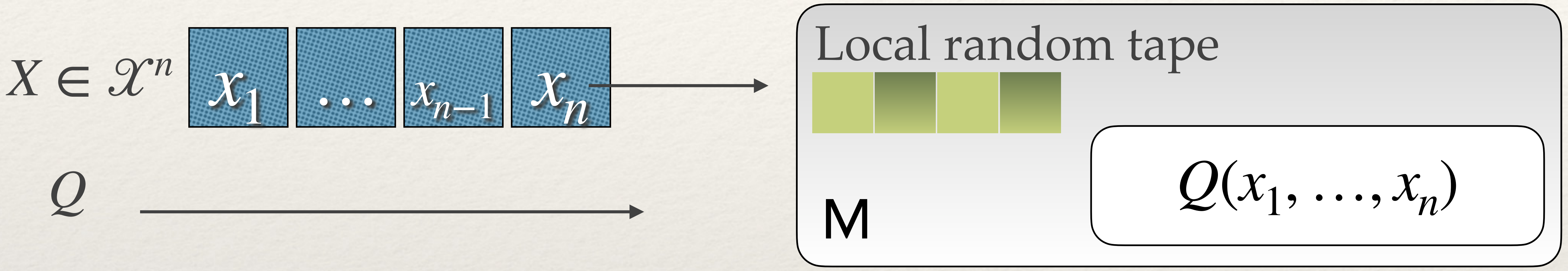


Randomness To The Rescue

- 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 $Q(X)$

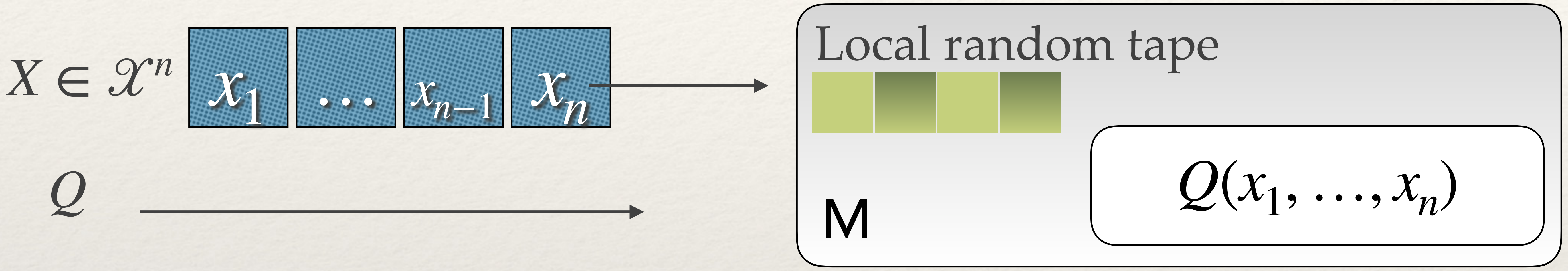


The output of M is a random value sampled according to $\mathsf{M}(X, Q)$, where the randomness comes from the M's private local randomness.

Thus $\mathsf{M}(X, Q)$ defines a probability distribution over $\mathcal{Y}$

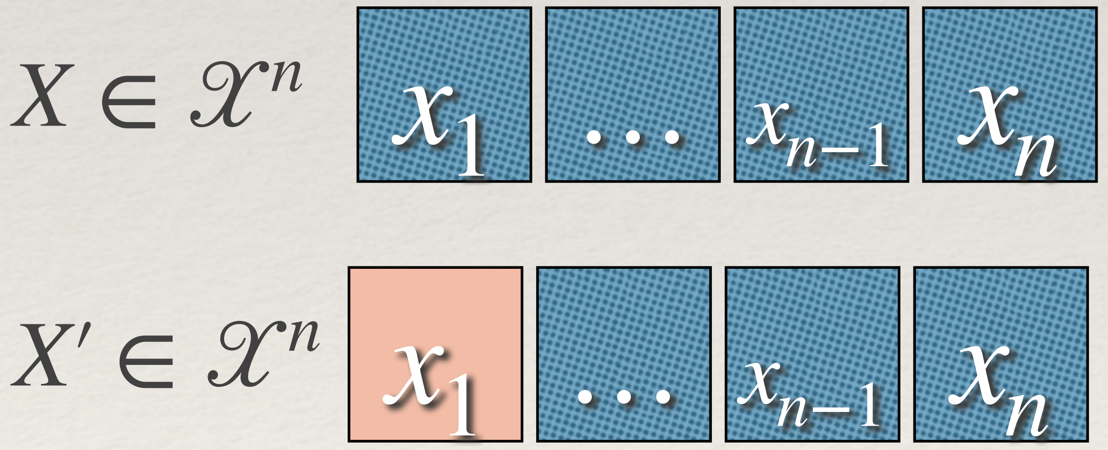
(ϵ, δ)-Differential Privacy (DP)

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



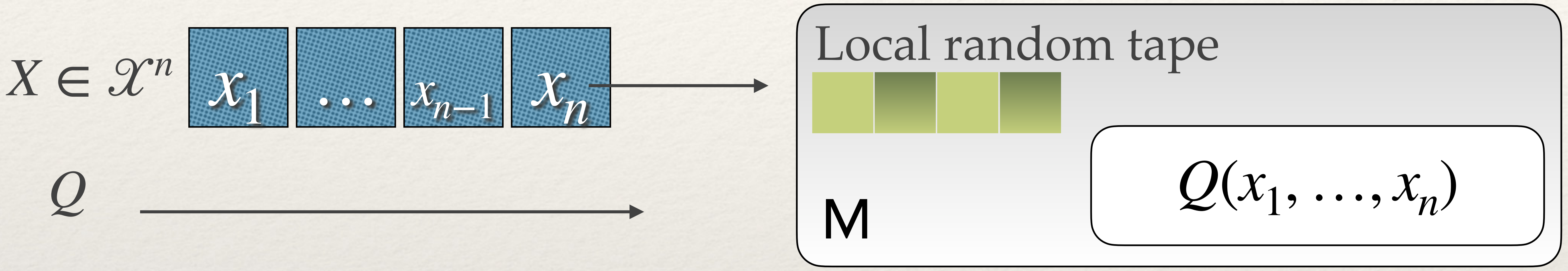
M is said to be $(\epsilon, \delta)-$Differentially Private if for any subset $T \subseteq \mathcal{Y}$

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



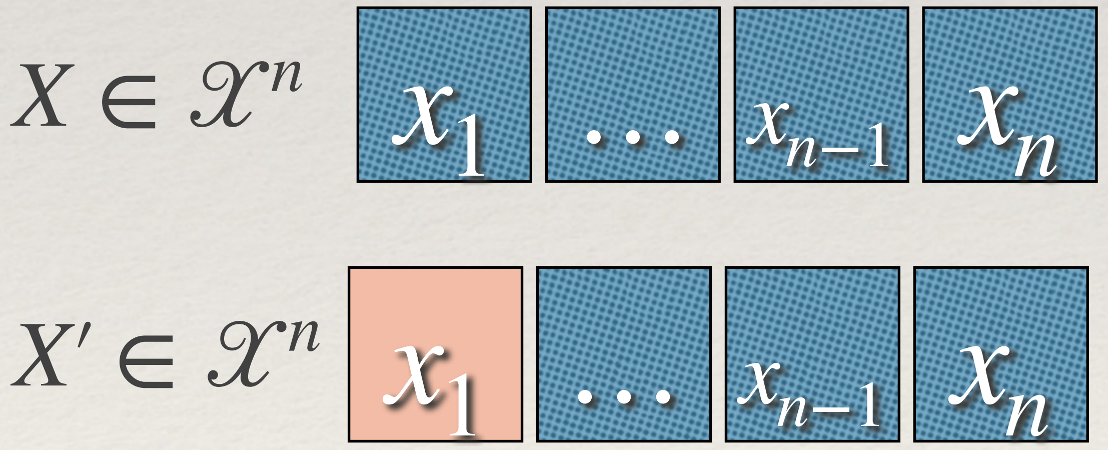
(ϵ, δ)-Differential Privacy (DP)

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



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

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

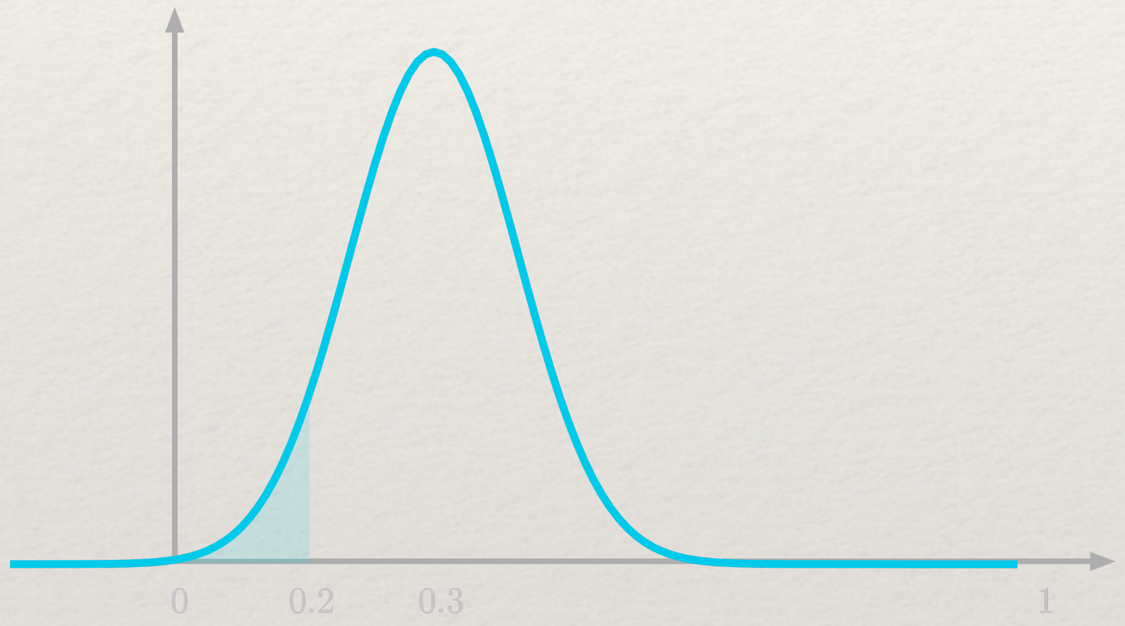


$$\Pr_{y \xleftarrow{\$} \boxed{M(X,Q)}}[y \in T] \le e^{\epsilon} \Pr_{y \xleftarrow{\$} \boxed{M(X',Q)}}[y \in T] + \delta$$

Understanding The Definition: Bayesian Perspective

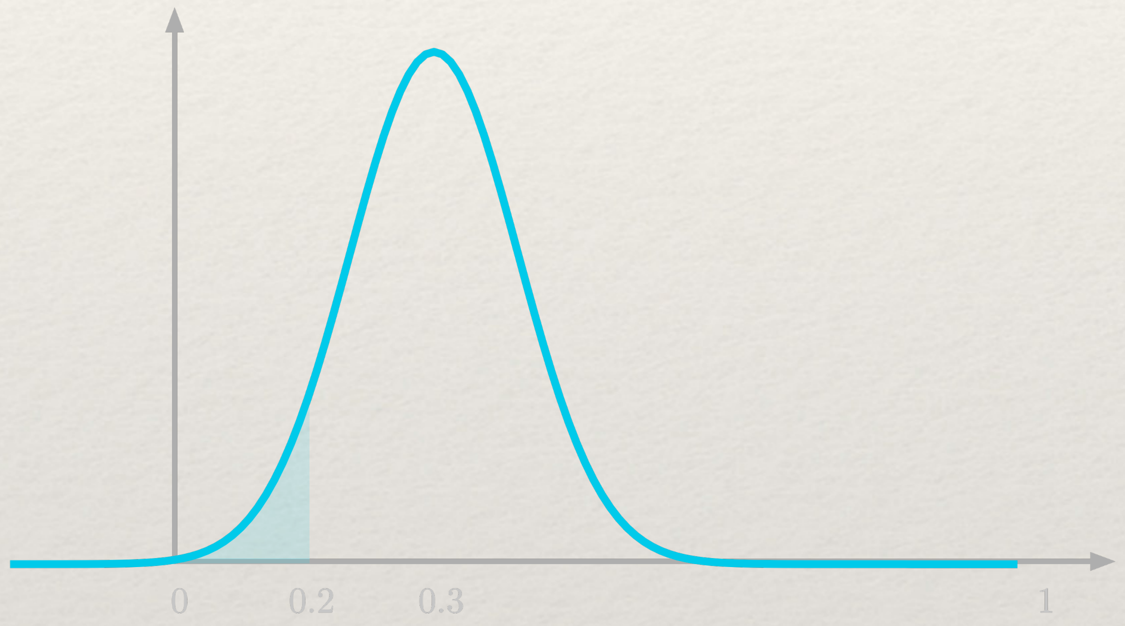
$X' \in \mathcal{X}^n$ $\boxed{x_1}$ $\boxed{\dots}$ $\boxed{x_{n-1}}$ $\boxed{x_n}$

Computationally unbounded algorithm, that knows all values in X except x_1



$\Pr[x_1 = x]$

Adversaries prior belief about x_1



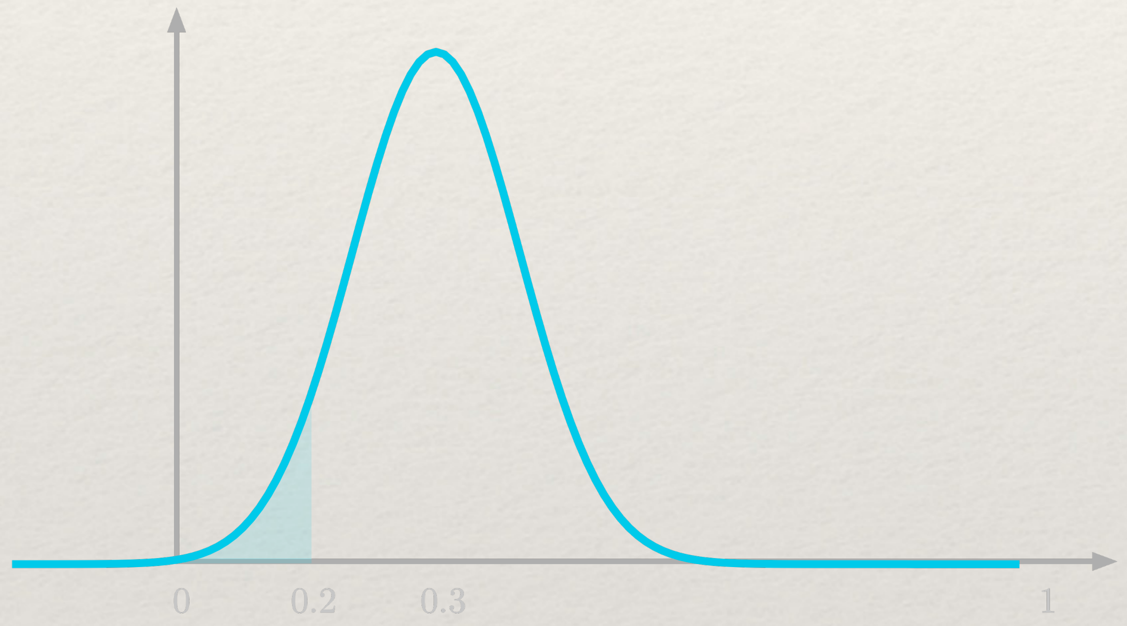
$\Pr[x_1 = x \;\; \mathsf{M}(X, Q) = y]$

Adversaries updated posterior about x_1 now that it has seen a sample y from $\mathsf{M}(X, Q)$

Understanding The Definition: Bayesian Perspective

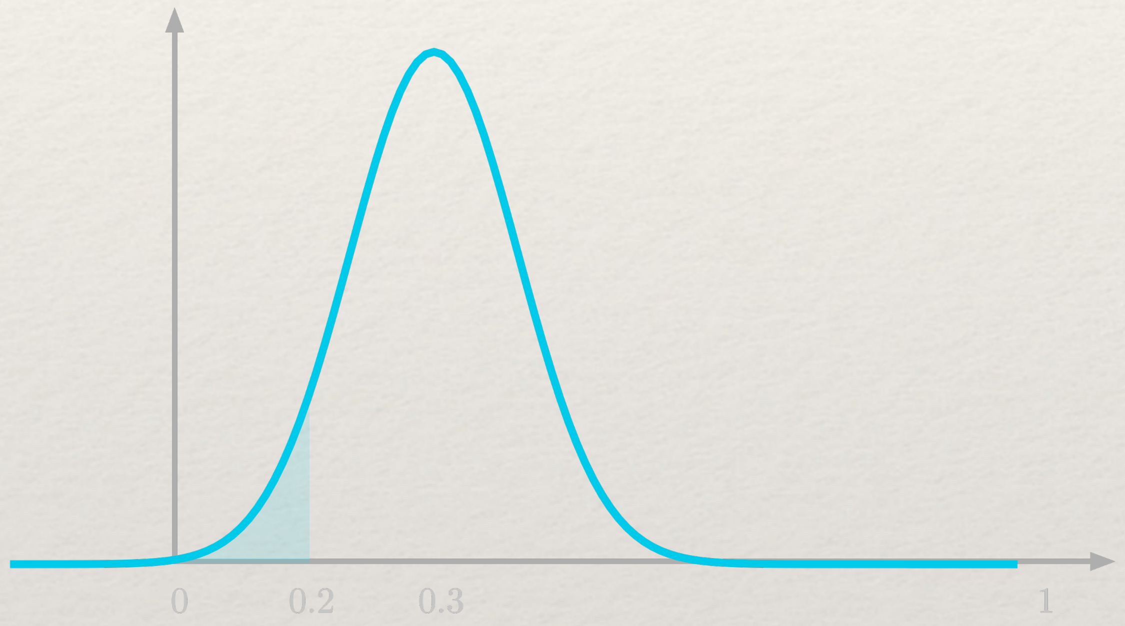
$X' \in \mathscr{X}^n$ | x_1 | $\dots$ | x_{n-1} | x_n

Computationally unbounded algorithm, that knows all values in X except x_1



$\Pr[x_1 = x]$

Adversaries prior belief abou



$\Pr[x_1 = x, \mathrm{M}(X, Q) = y]$

If M is (ϵ, δ)−DP, then with probability alteast $1 - \delta$
$\mathrm{TV}(D_1, D_2) \leq \epsilon$

Utility Of A DP Algorithm

An algorithm $\mathsf{M} : \mathscr{X}^n \times Q \to \mathscr{Y}$ for releasing a DP version of $y = Q(X)$ where $(\mathscr{Y}, d)$ 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

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

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

$d(x, y) = \| x - y \|_\infty$

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

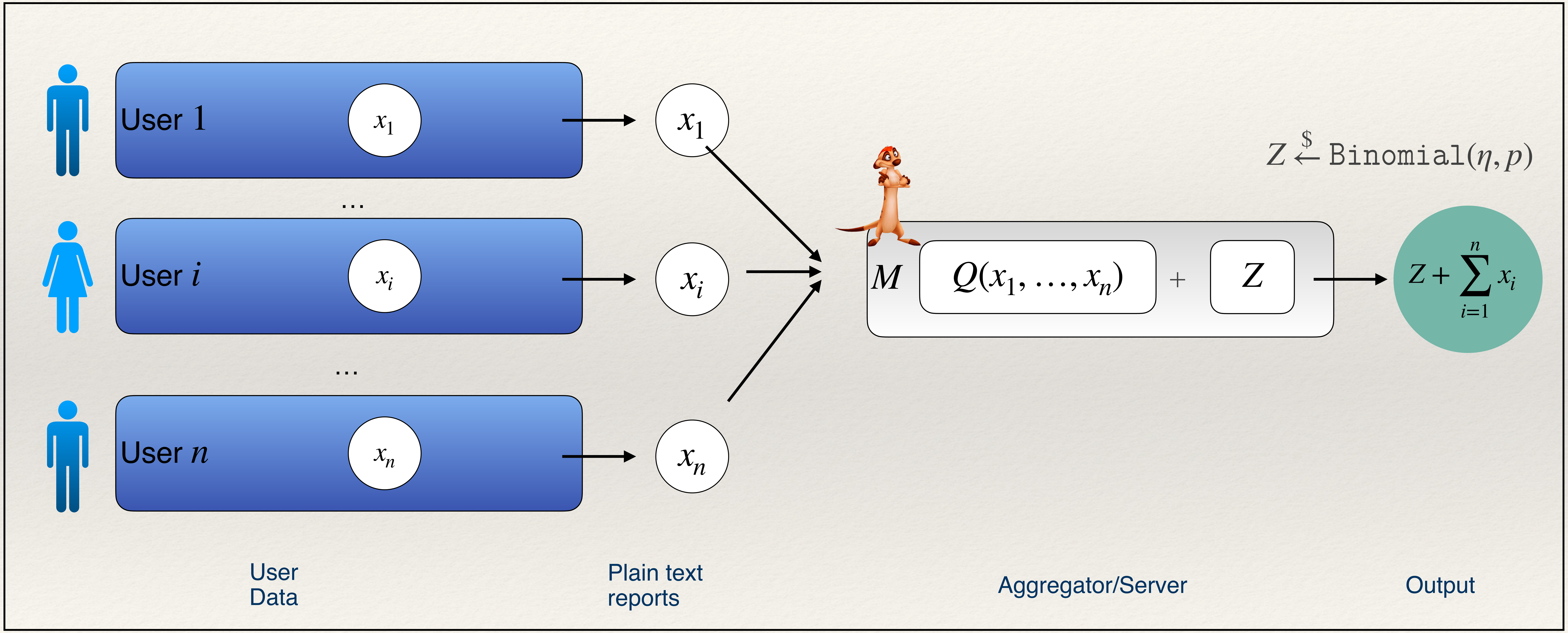
$\boxed{Q(x_1, \ldots, x_n)} + \boxed{\texttt{Laplace}(b)} \longrightarrow \boxed{M(X; Q)}$ $\quad \epsilon = \frac{1}{b}, \delta = 0$

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

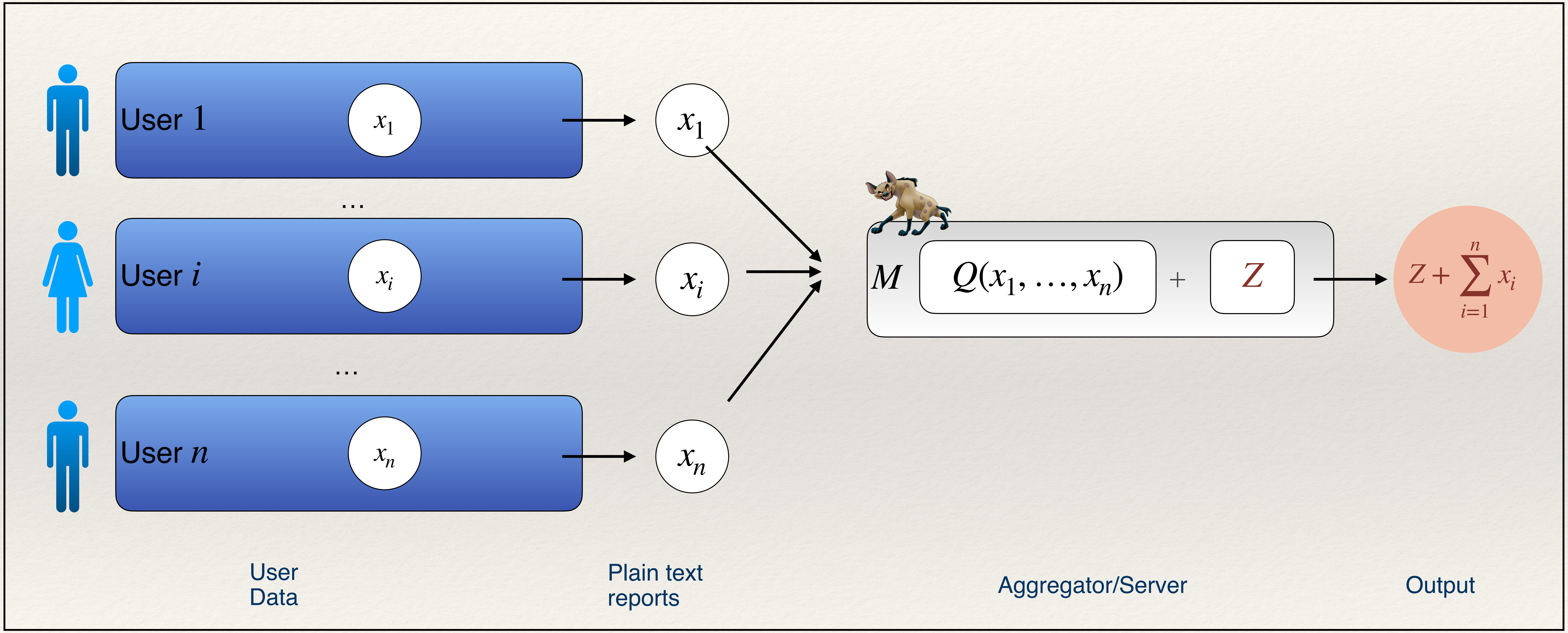
$\ldots \qquad \ldots$

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

Back To Our Ideal World



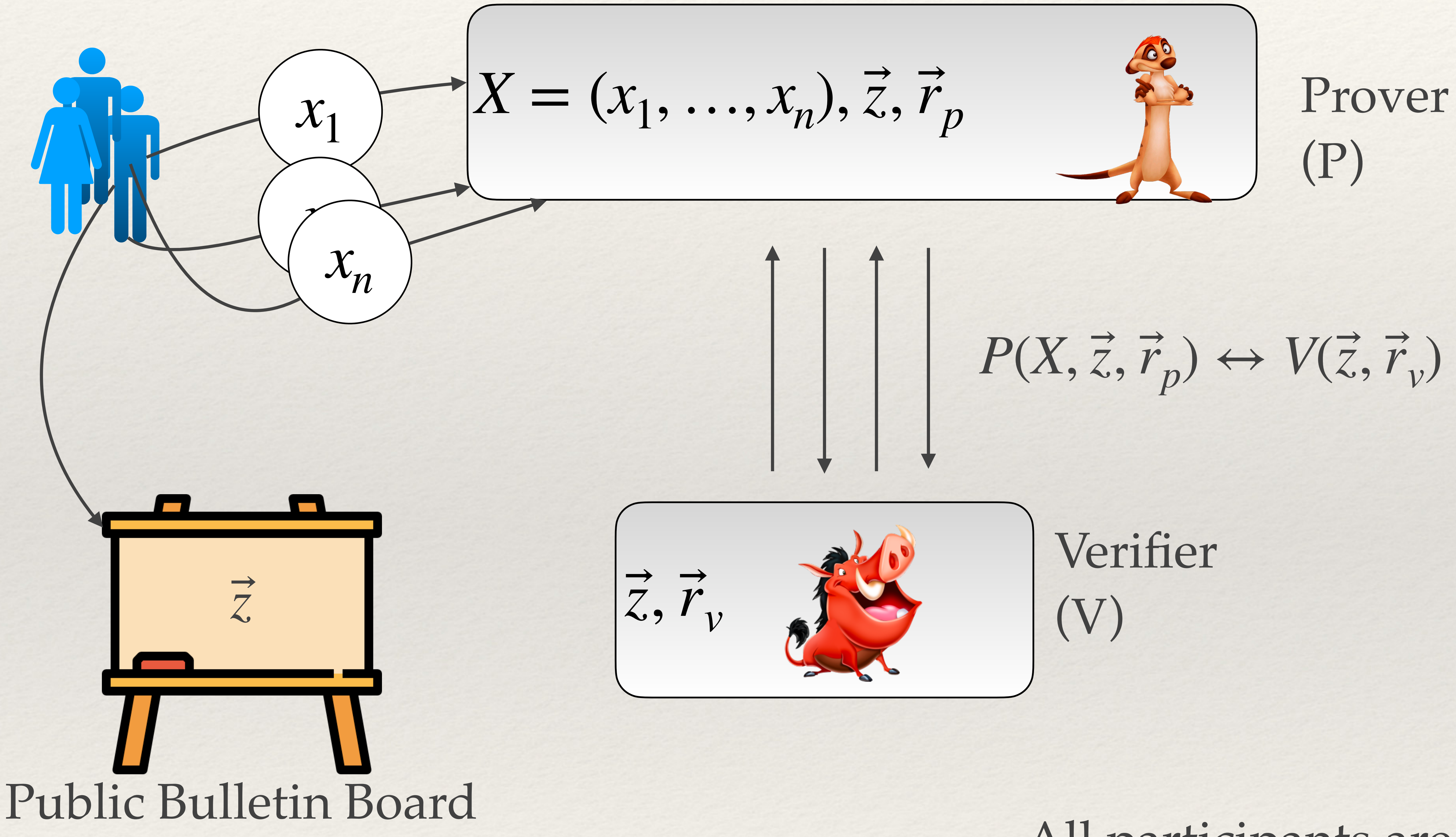
What If We Cannot Trust The Server ?



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.

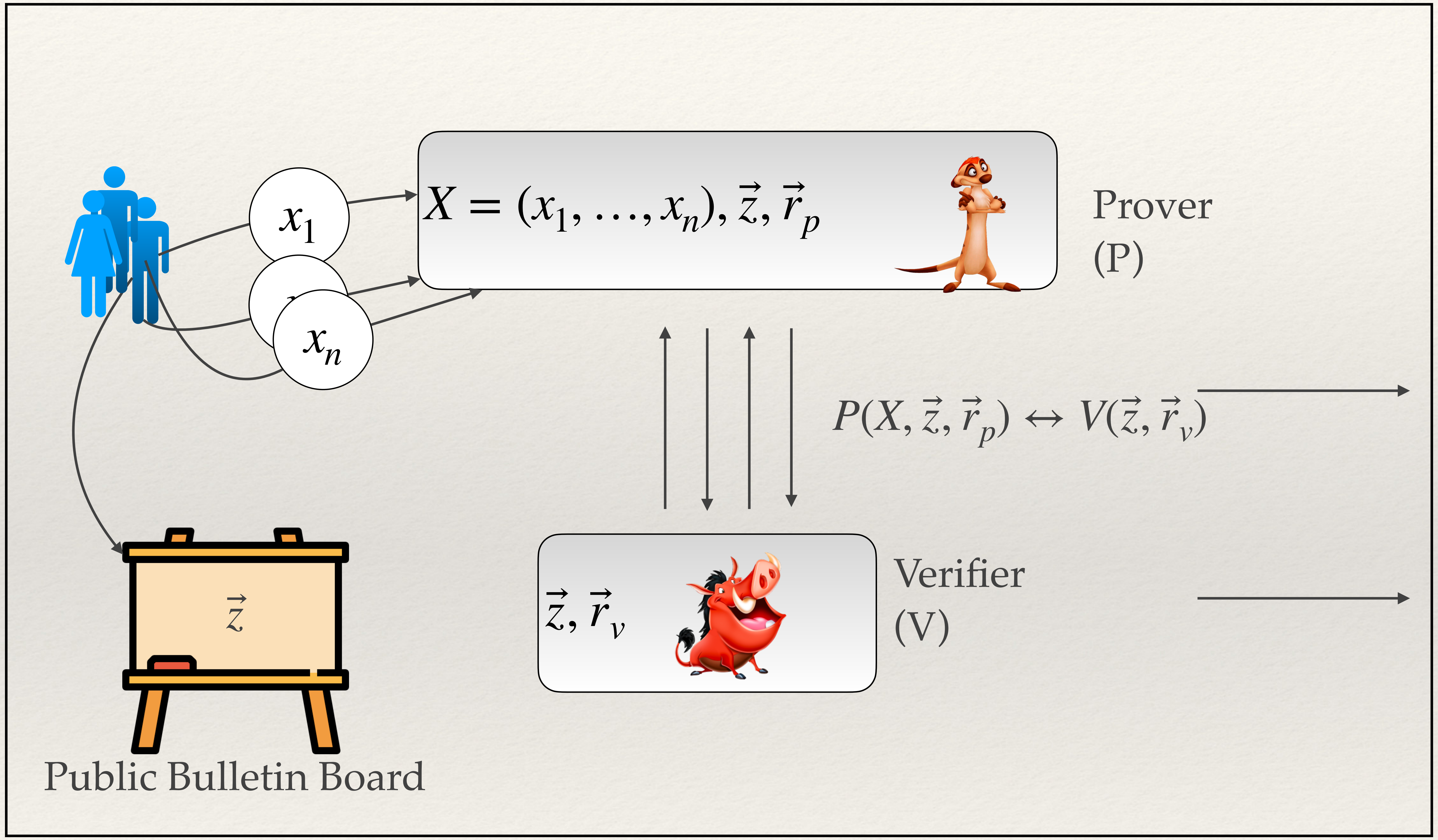
Client Server Verifiable DP



Clients send inputs to the server as usual, but additionally they publish information to a public bulletin board

All participants are PPT Turing Machines.

Verifiable DP



Security Parameter

$\kappa \in \mathbb{N}$

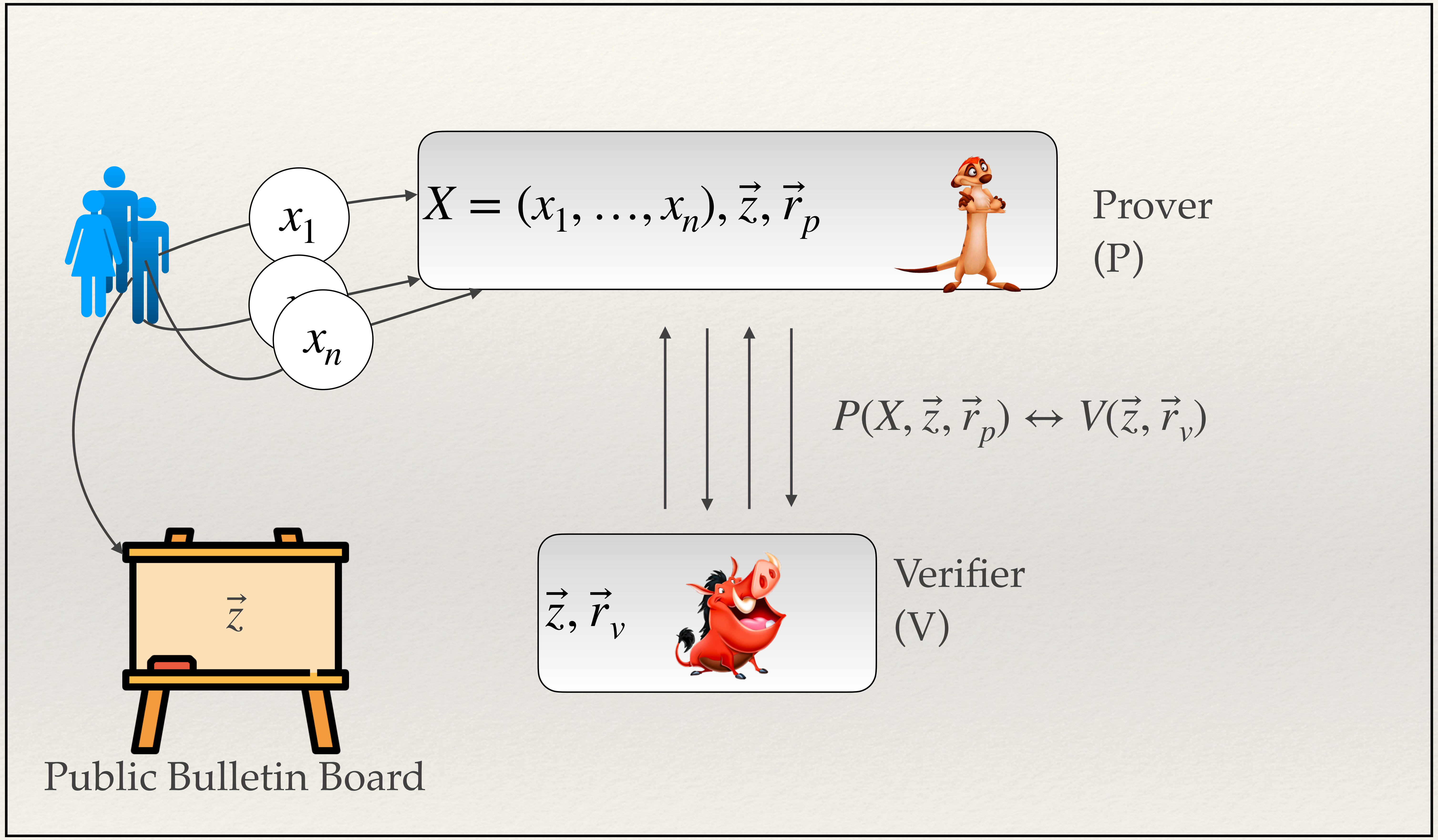
Typically, the size of the input in bits.

Prover interacts with the verifier over multiple rounds and **finally outputs** y.

The verifier looks at the board and the messages either **accepts or rejects the claim that** $y \xleftarrow{\$} M(X, Q)$

$\texttt{Verify}(P \leftrightarrow V) \in \{0,1\}$

Verifiable DP

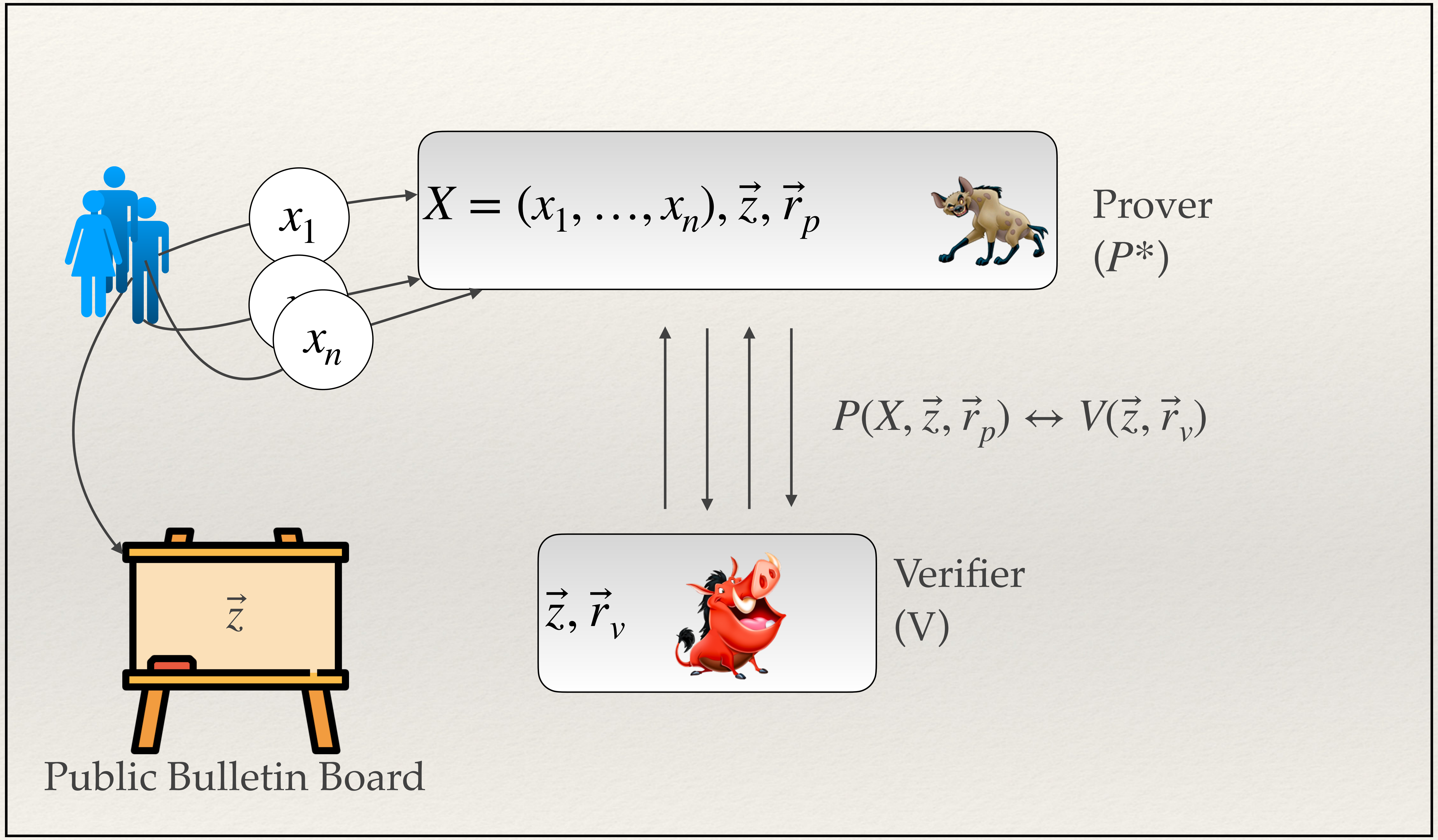


Completeness:

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

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

Verifiable DP

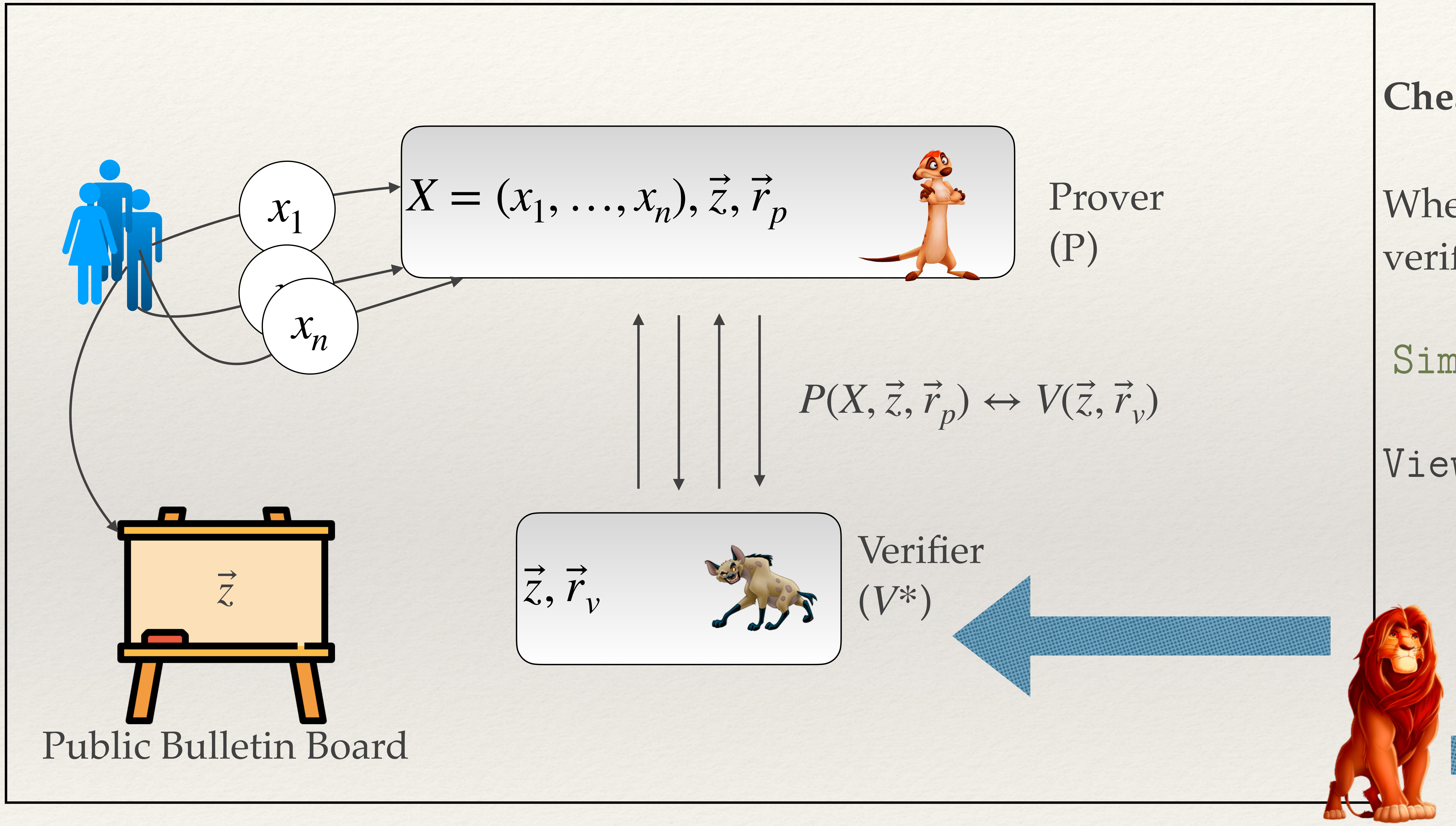


Soundness

For any cheating prover P^* that samples y from a distribution $\mathscr{D}$ such that $\mathsf{TV}(\mathsf{M}(X, Q), \mathscr{D}) > 0$

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

Verifiable DP

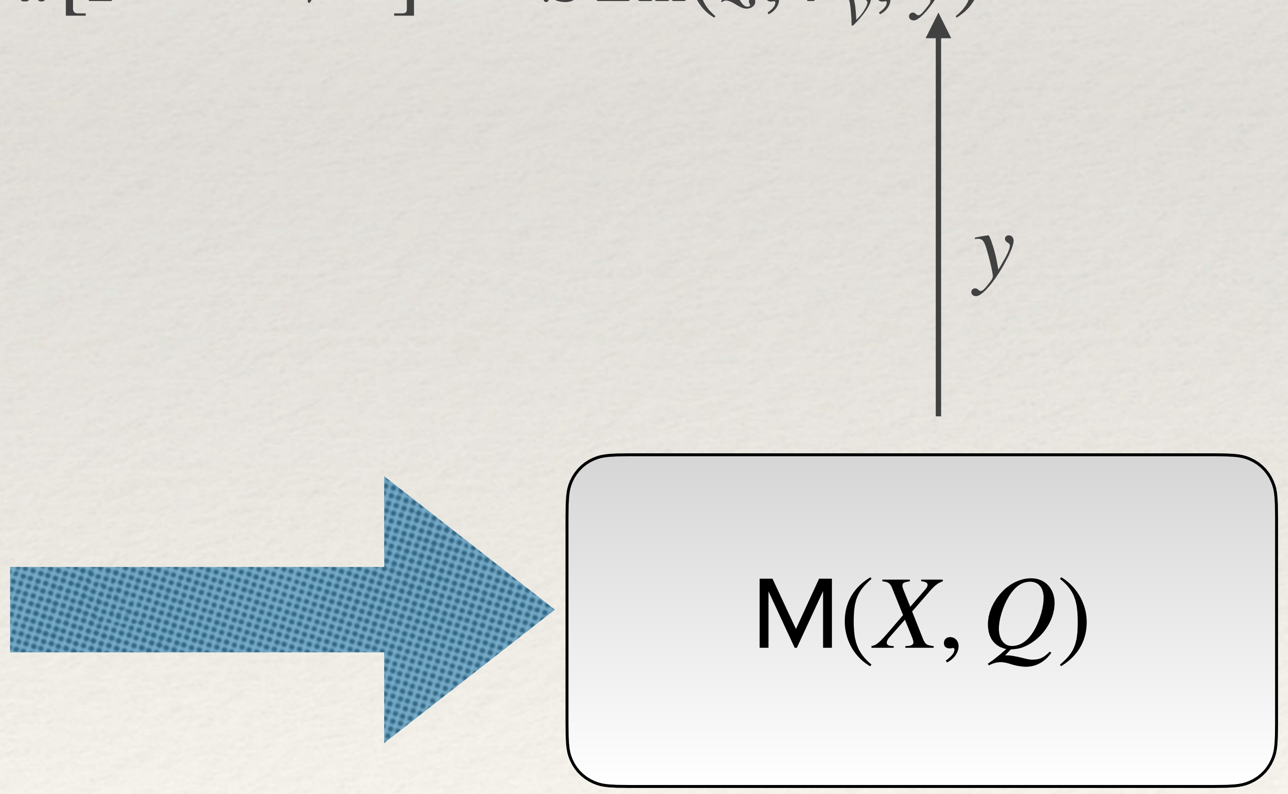


Cheating Verifier Zero Knowledge

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

$\mathtt{Sim}$  that has oracle access to V^*

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



The Soundness/ZK conflict

Prover (P)   Verifier(V)

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

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

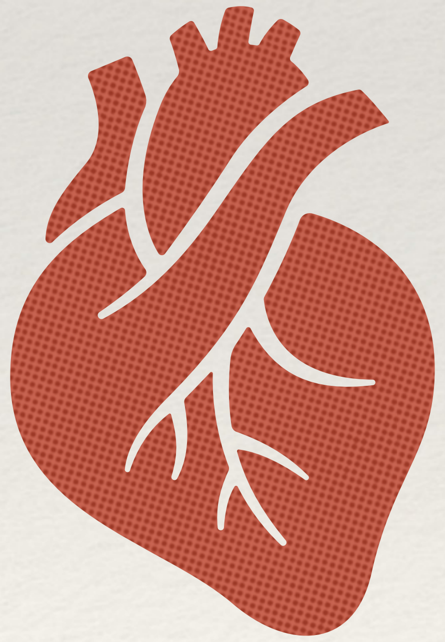
$$y = \boxed{Q(x_1, \ldots, x_n)} + \boxed{Z} \longrightarrow \boxed{M(X; Q)}$$

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

Some Crypto Prelims

Commitments

Two stage interactive protocol between a Committer and a Receiver

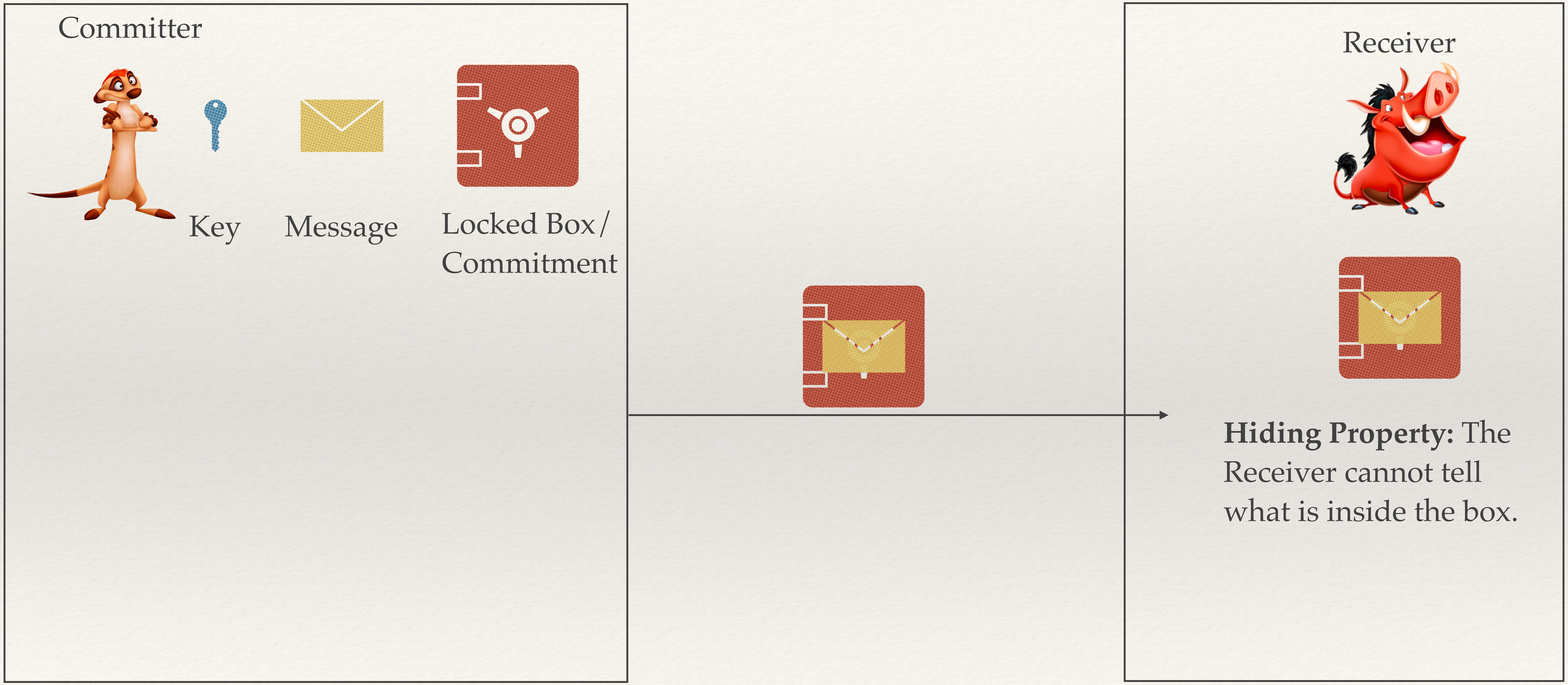
Commit Phase

Committer

Receiver

Reveal Phase

Commit Phase



Reveal Phase

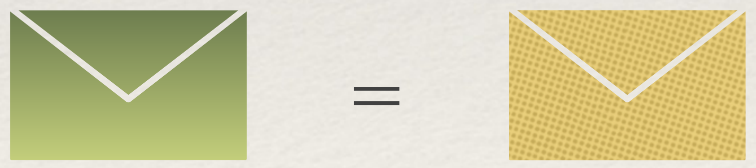
Committer

Key   Message   Locked Box / Commitment

Receiver

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

Use key to open box and reveal message and check



Example: Pedersen Commitments

Key

Message

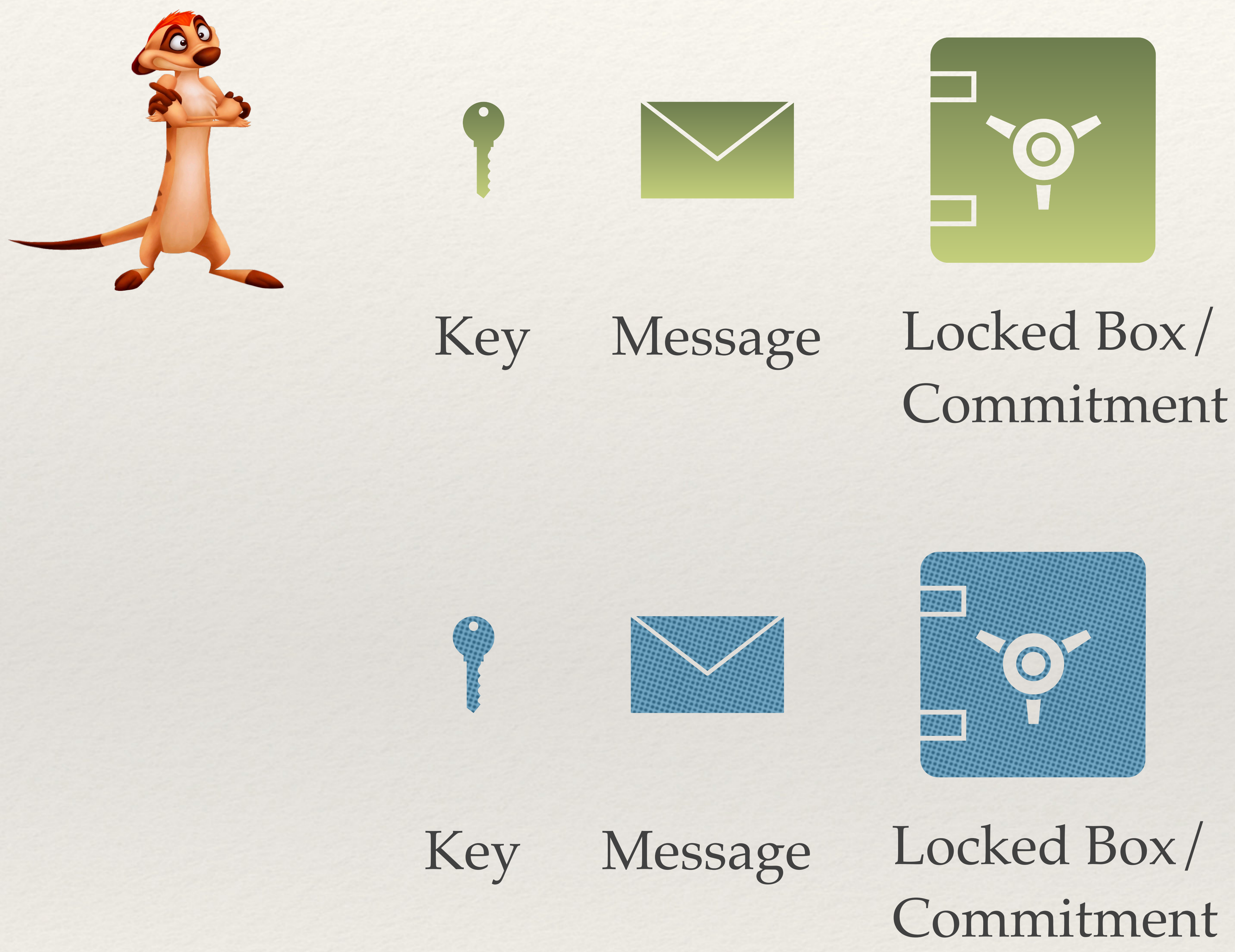
Locked Box/
Commitment

$r \xleftarrow{\$} \mathbb{Z}_q$ $\quad x \in \mathbb{Z}_q$ $\quad c = g^x h^r$

$\boxed{c \text{ is statistically hiding and computationally binding}}$

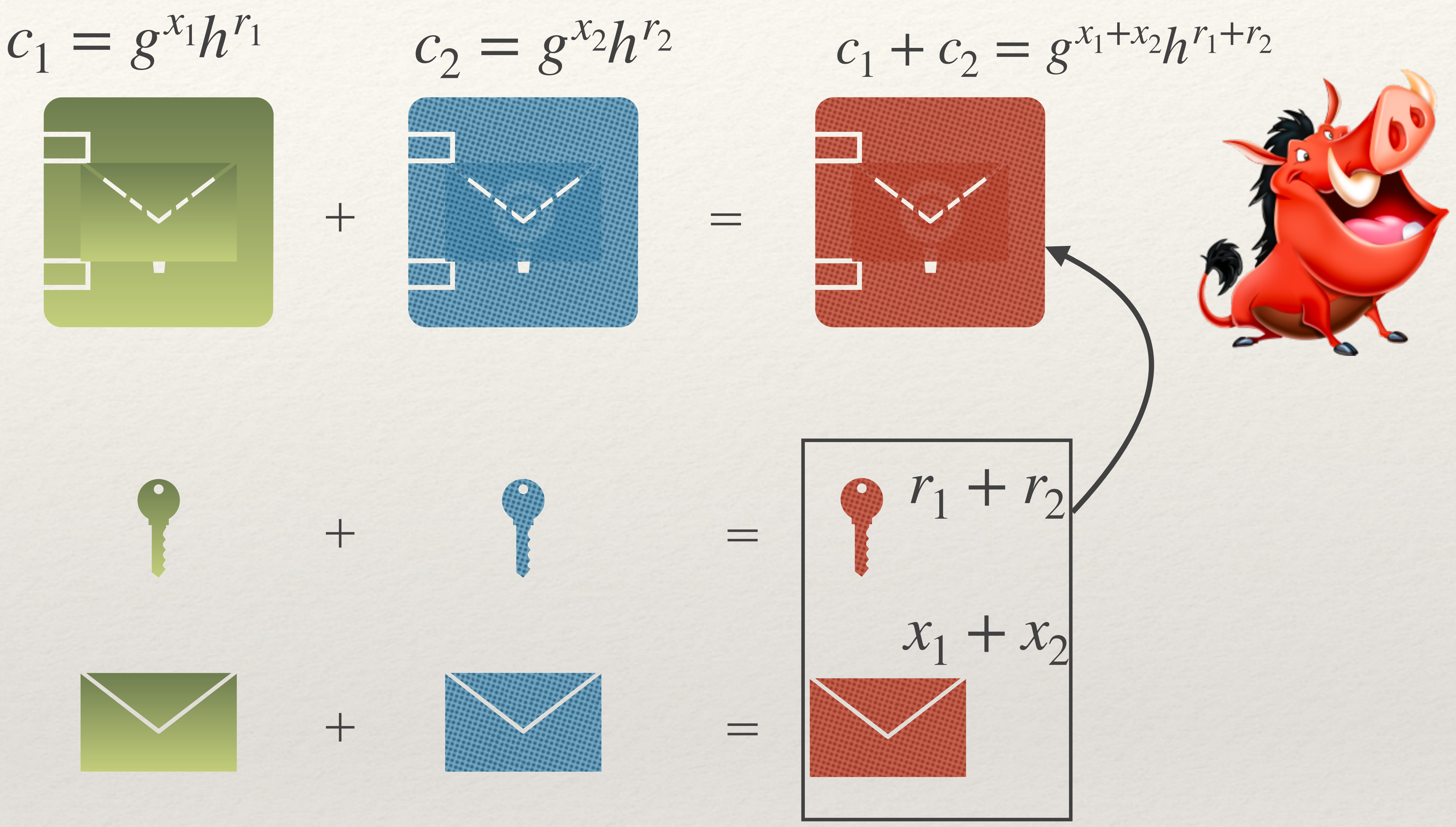
Let $\mathbb{G}_q$ be a prime order cyclic group with operation $+$ and $g \in \mathbb{G}_q$ and $h \in \mathbb{G}_q$ be generators.

Homomorphic Commitments



The combined keys open the combined boxes

Pedersen Commitments Are Homomorphic

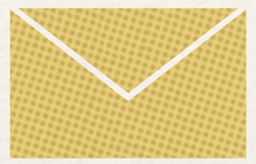


The combined keys open
the combined boxes

Disjunctive OR Arguments

Committer

Key



Message

Locked Box/ Commitment

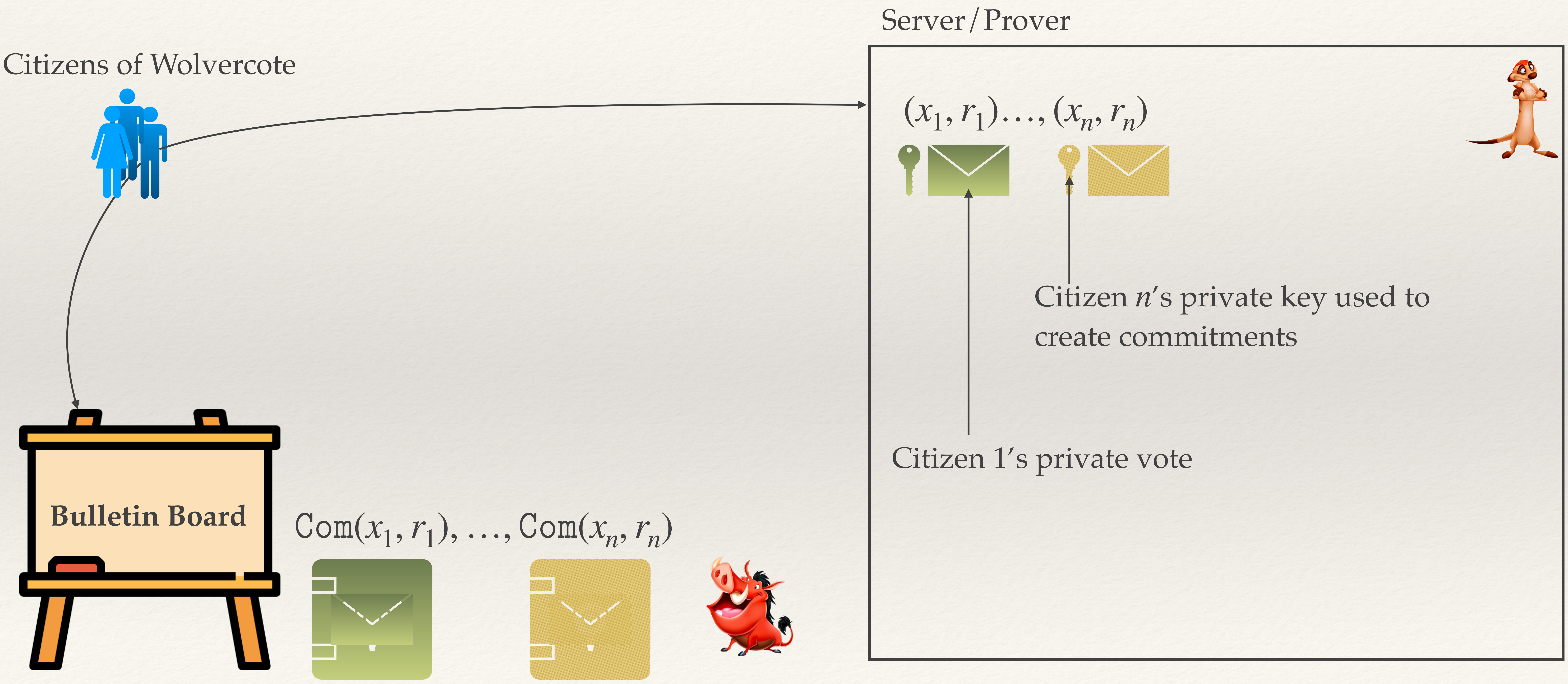
Receiver

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.

Main Protocol

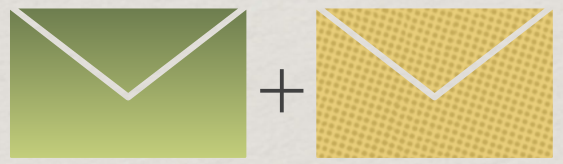


Non Private Counting Is Was

Server / Prover

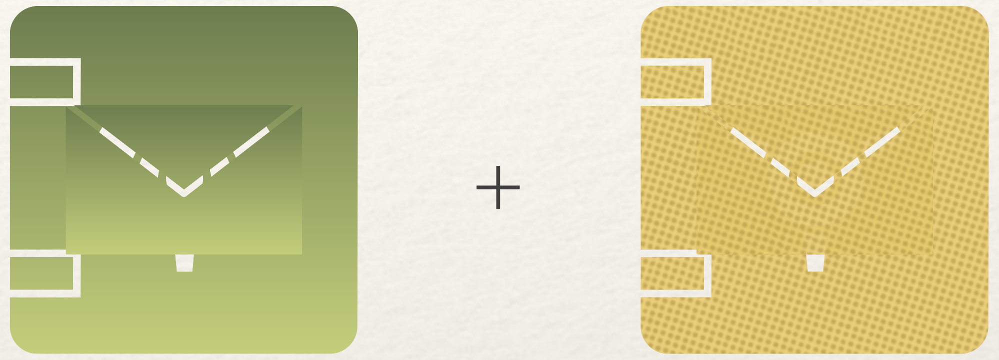
$(x_1, r_1) \ldots, (x_n, r_n)$

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 Is Was

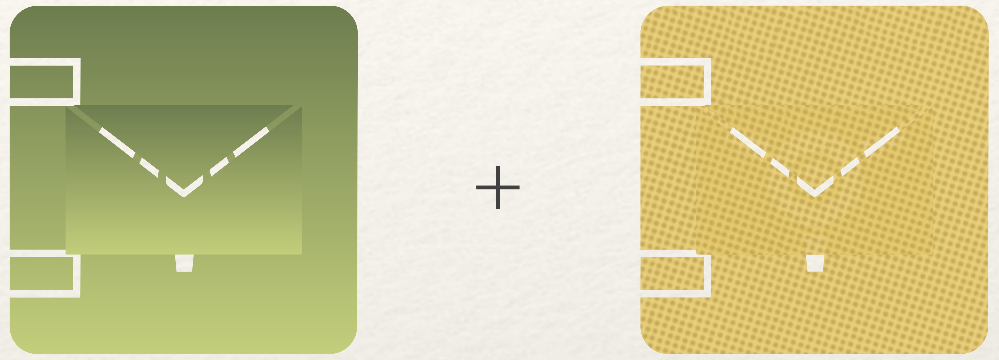
Server / Prover

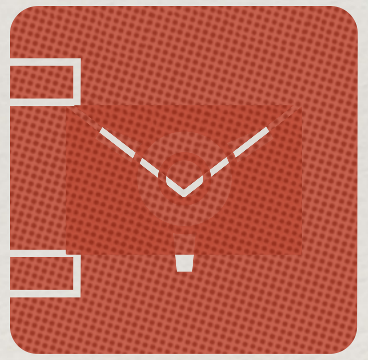
$(x_1, r_1) \ldots, (x_n, r_n)$

Citizen n's private key used to create commitments

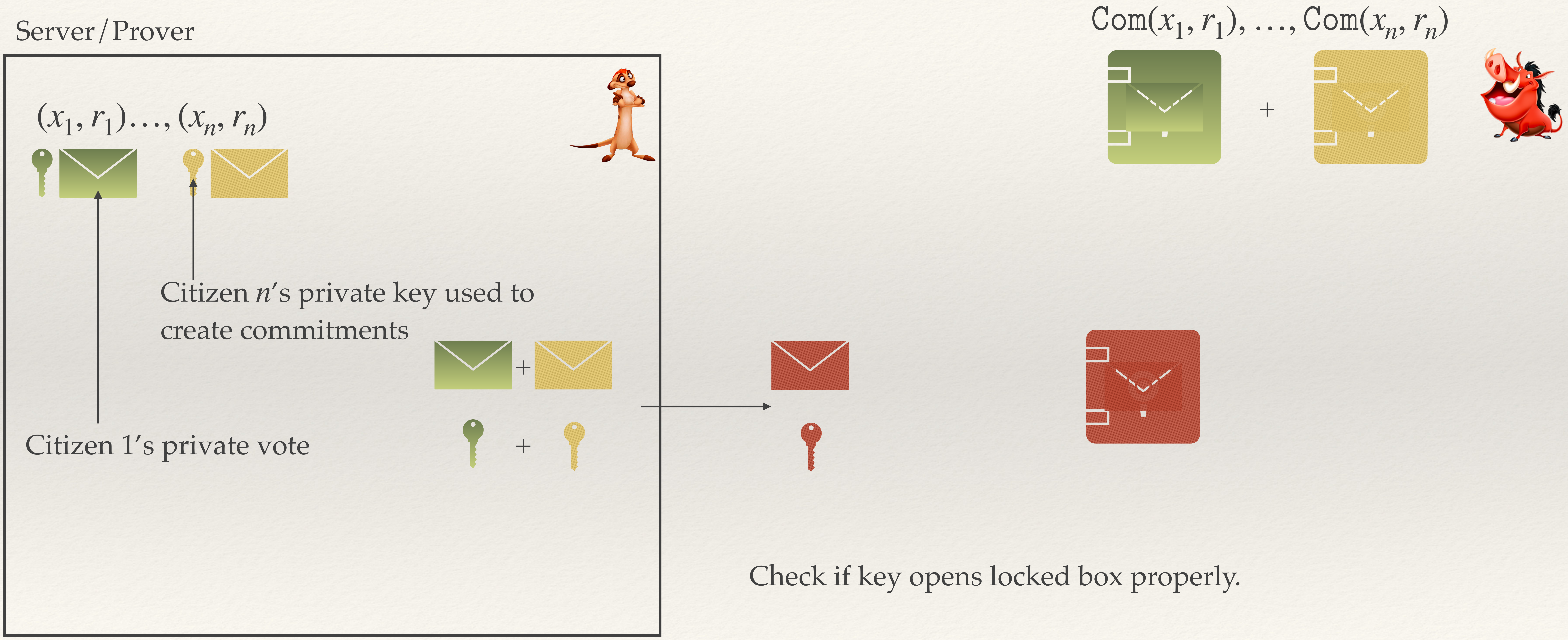
Citizen 1's private vote

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

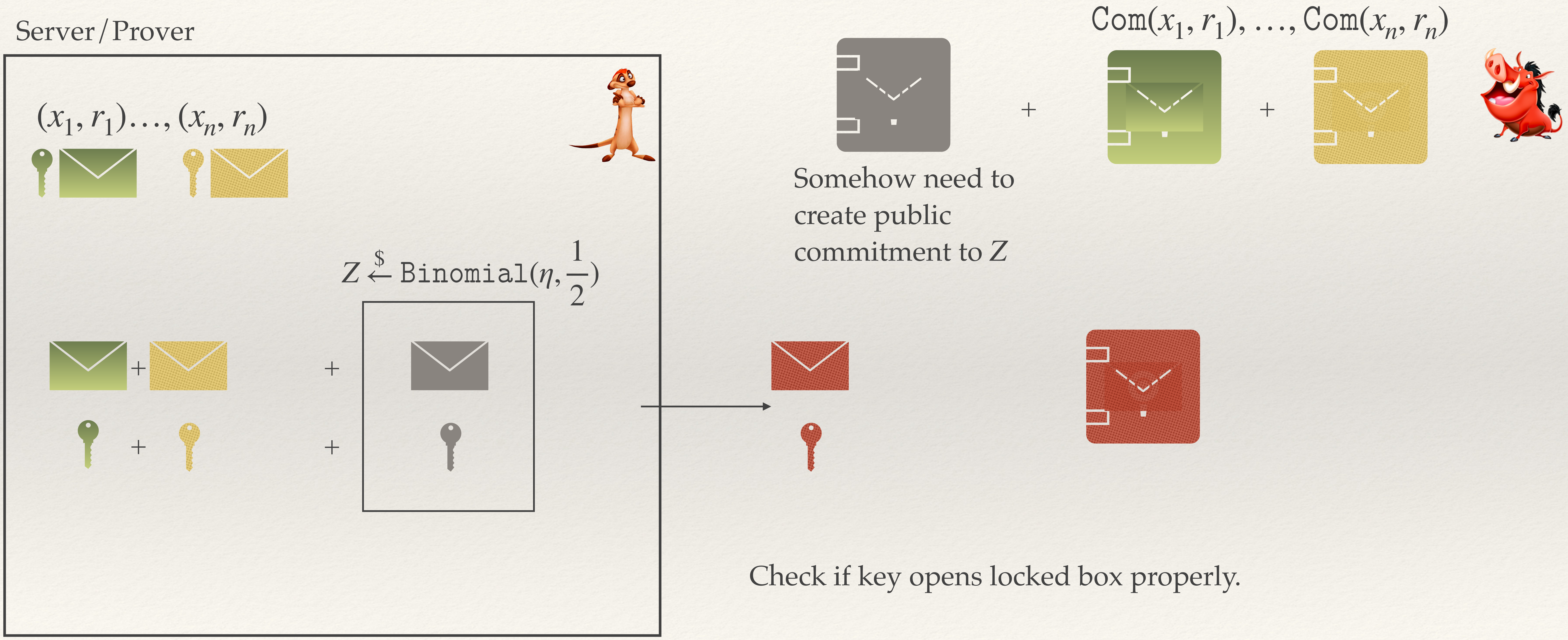




Non Private Counting Is Was

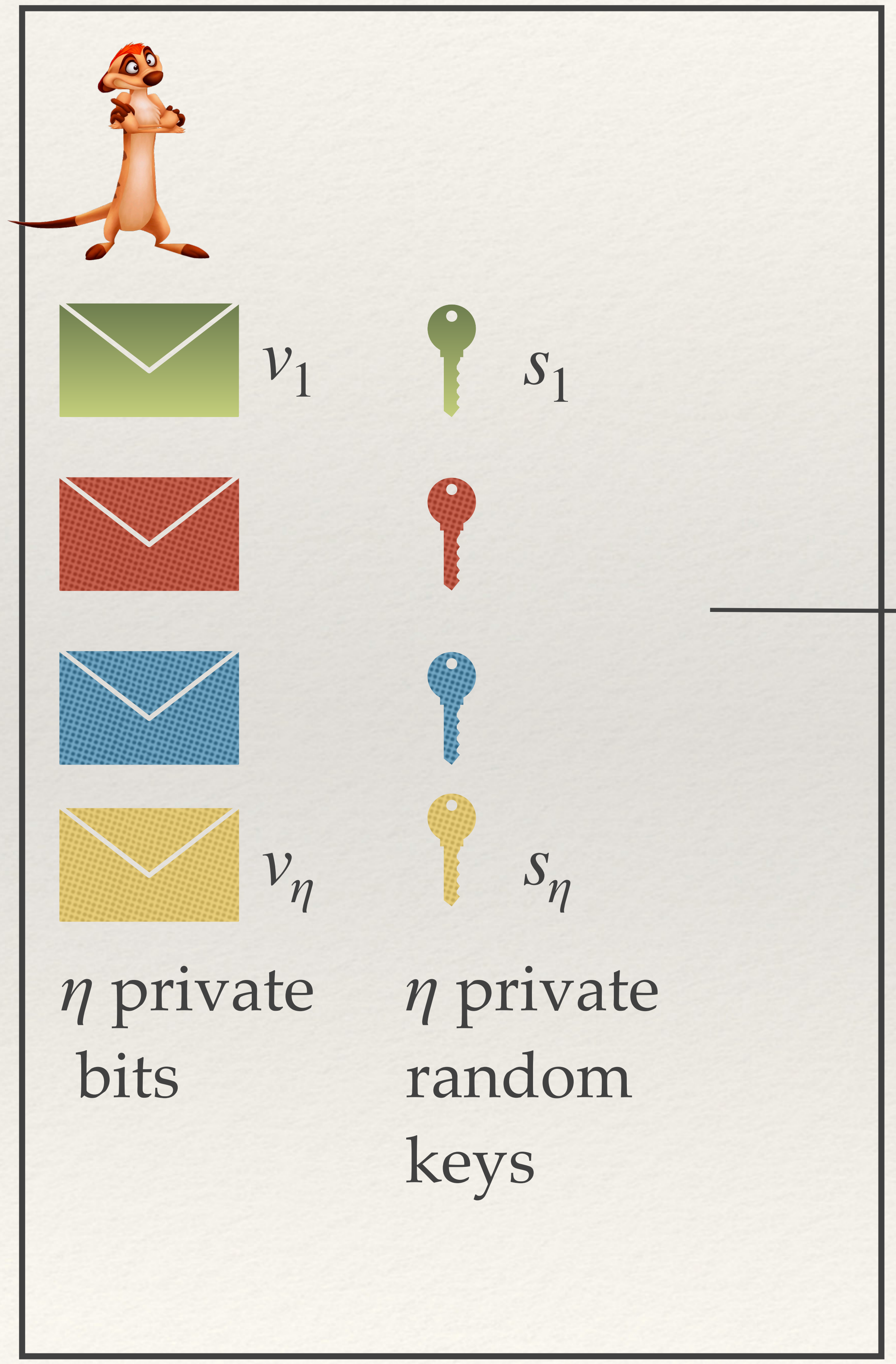


Verifiable DP counting - Essence



A Simple Trick

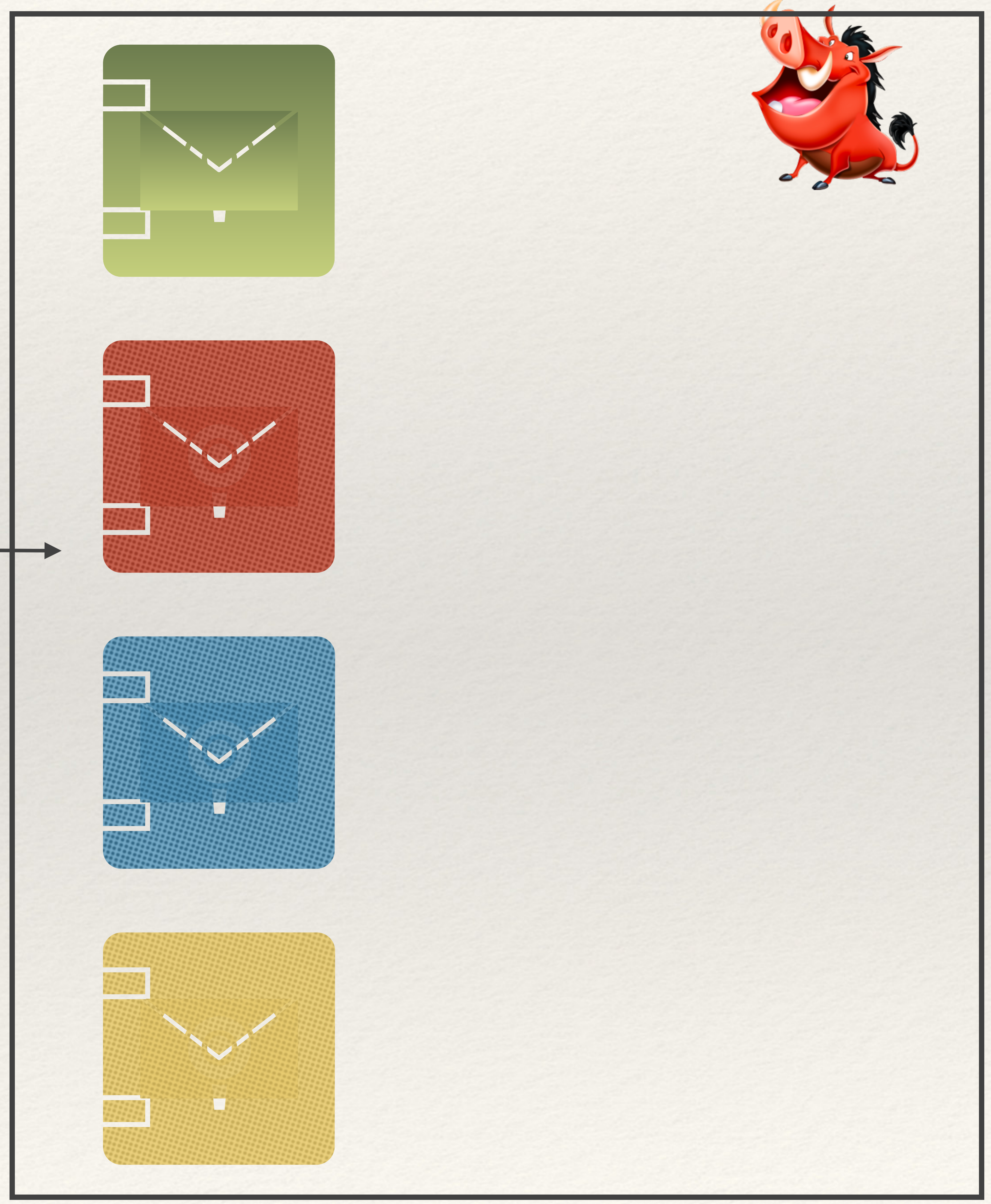
Server / Prover



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

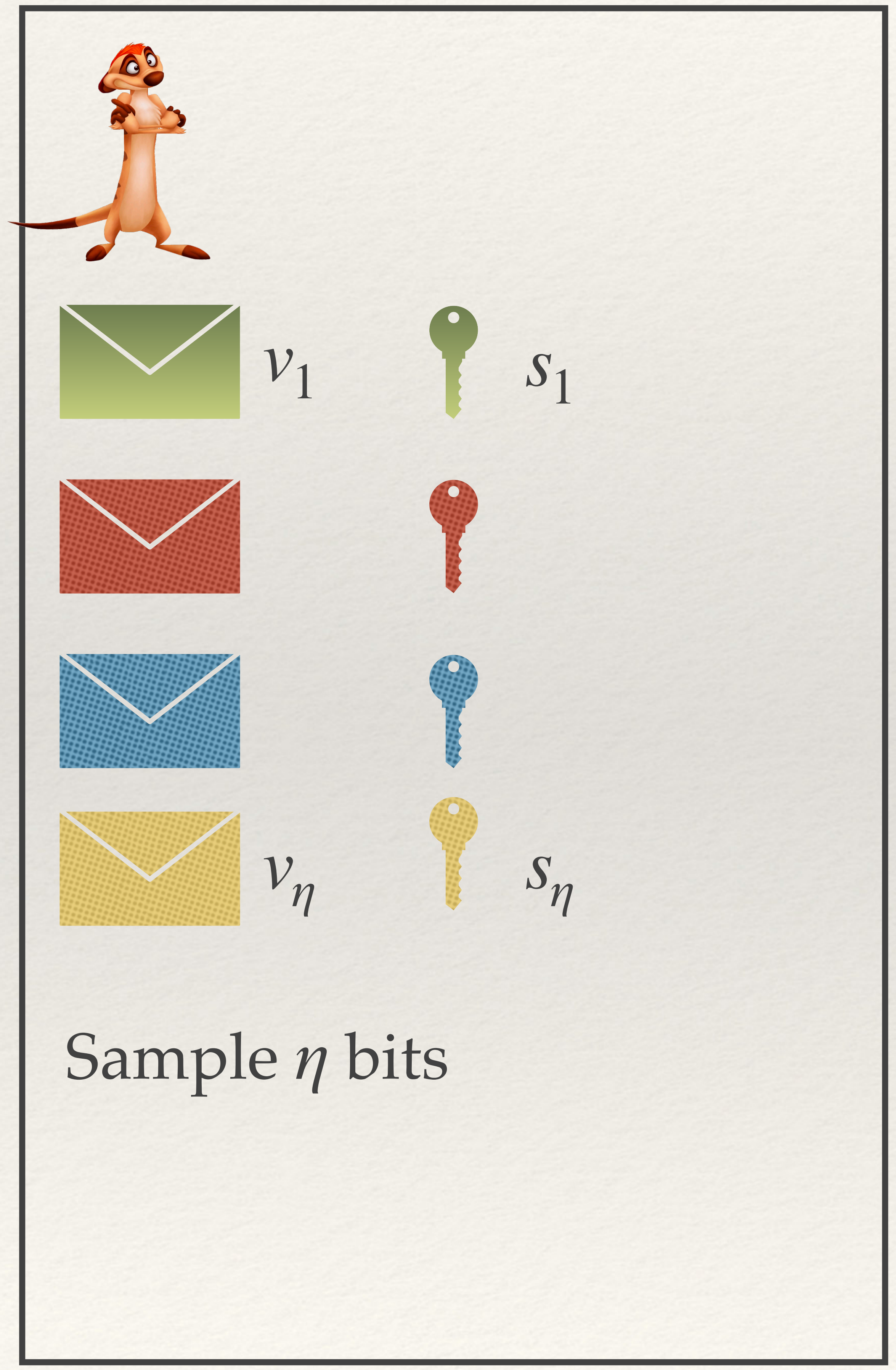
All the verifier knows is that these boxes are a commitment to a bit.

Verifier

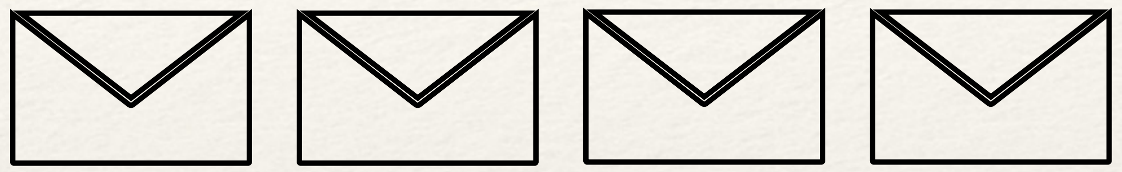


A Simple Trick

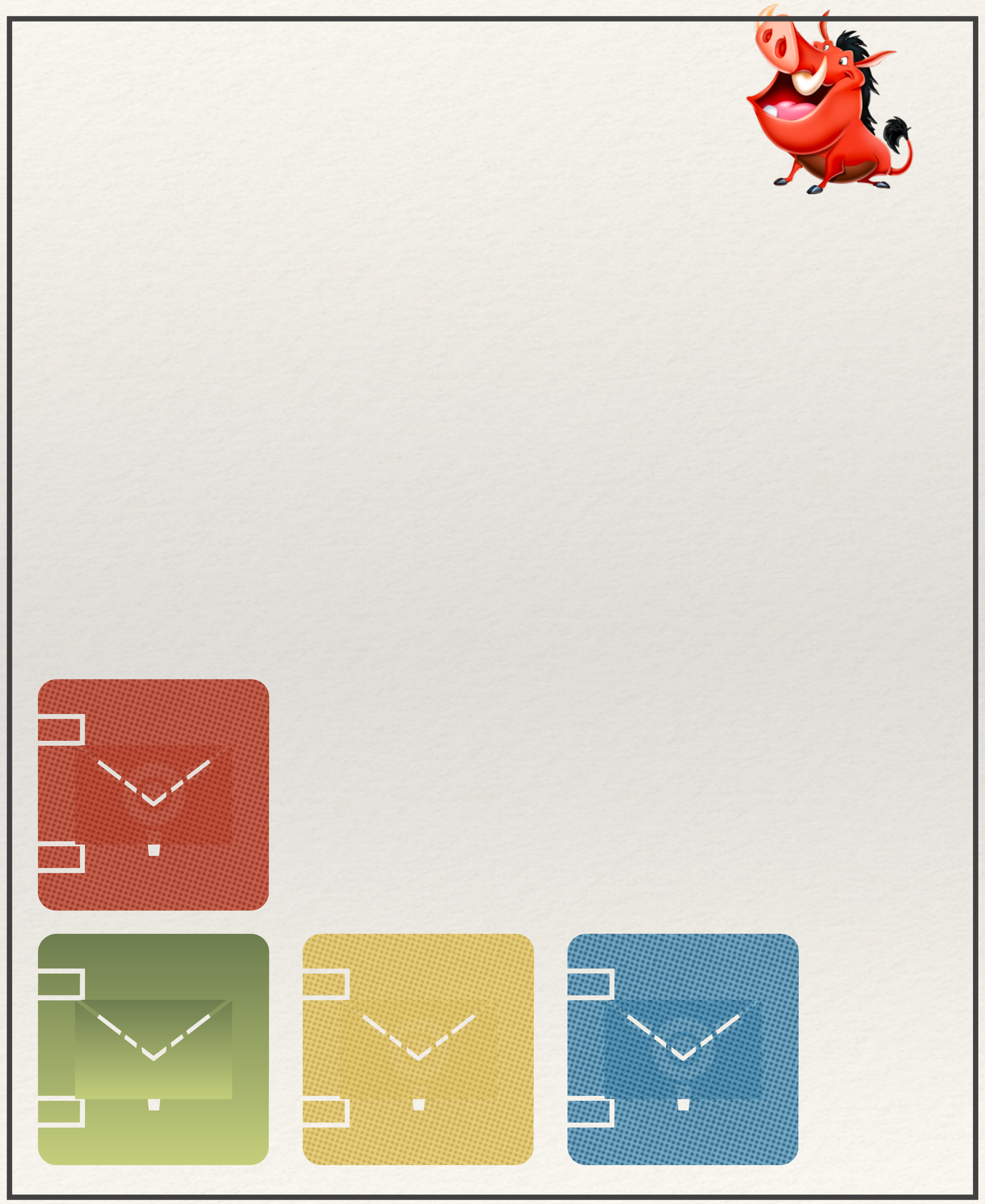
Server / Prover



Verifier generates η public unbiased coins

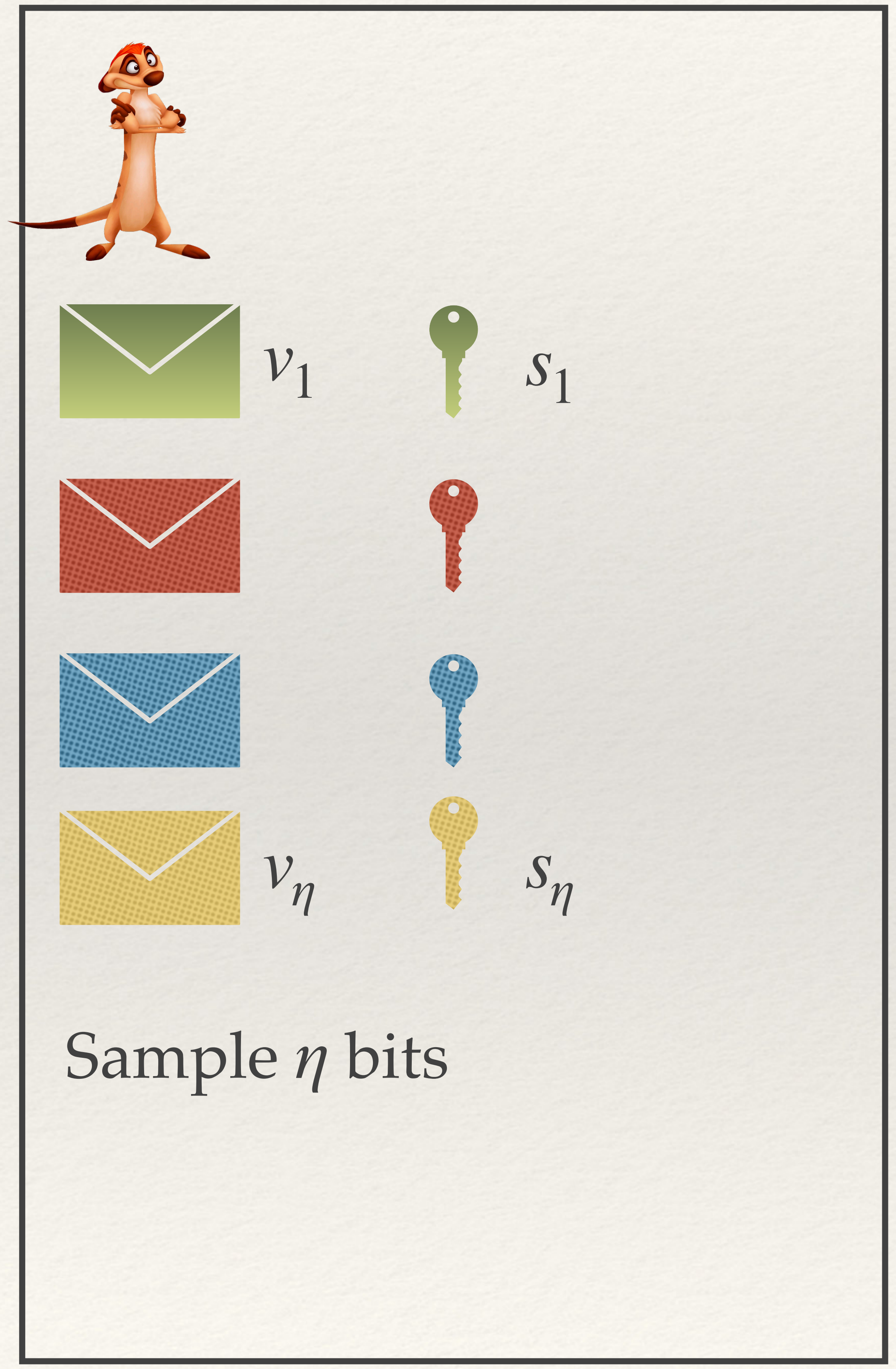


Verifier

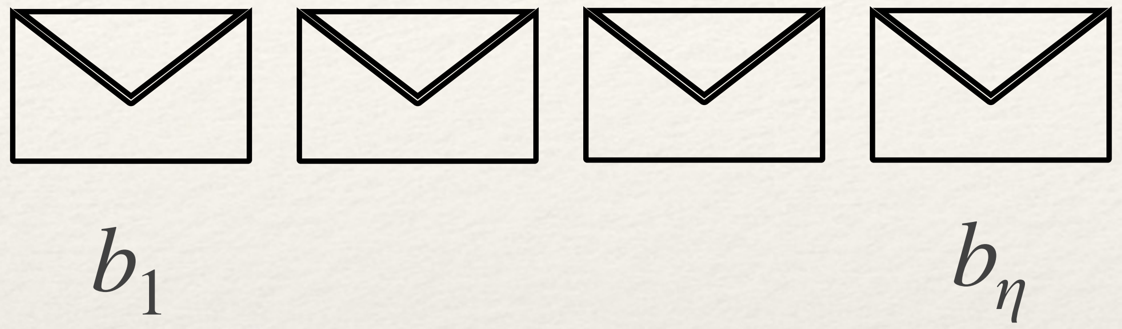


The Final Trick

Server/Prover



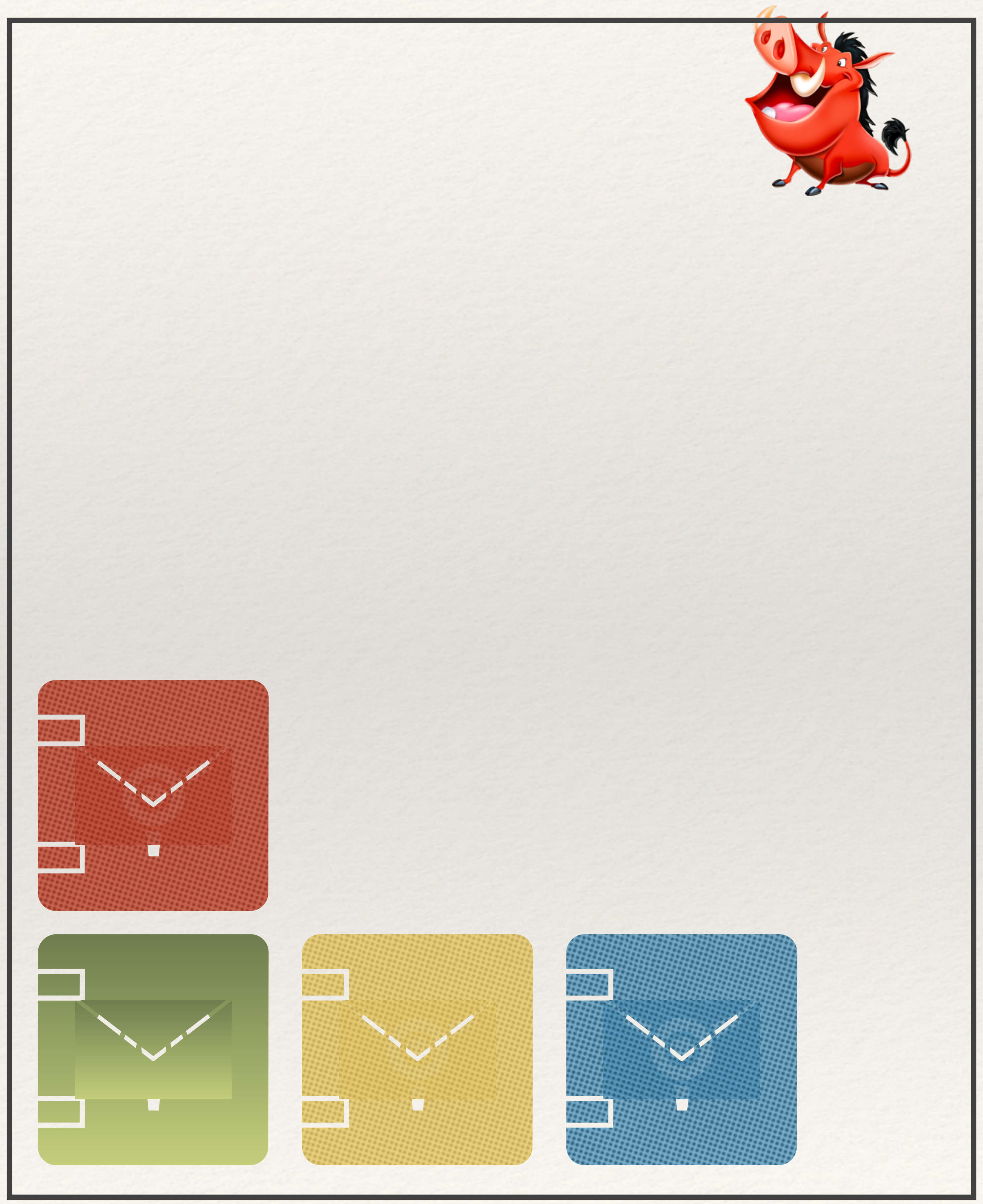
Verifier generates η public unbiased coins



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

Otherwise, leave v_i unchanged.

Verifier

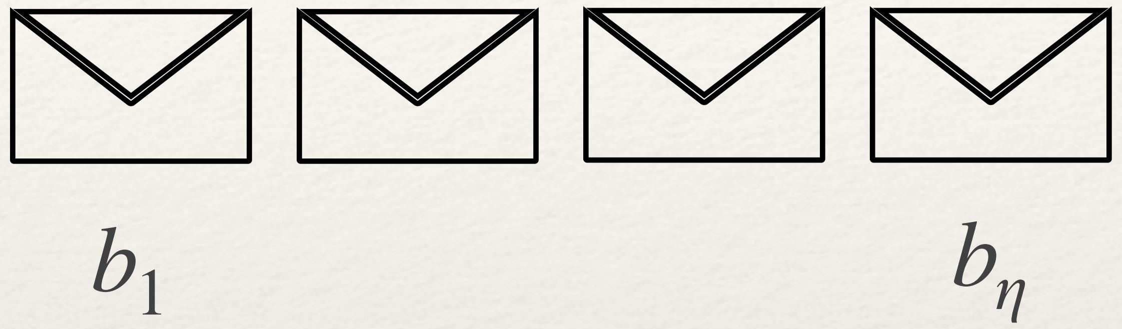


The Final Trick

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

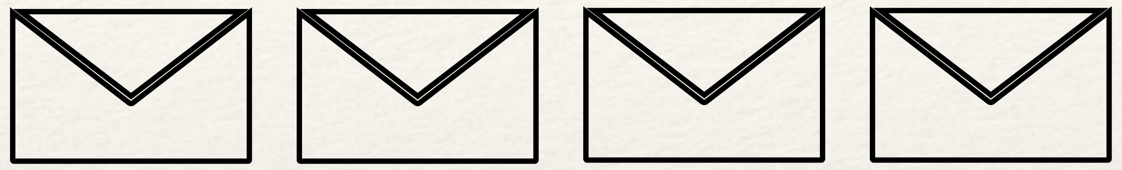
Server/Prover

v_1 s_1

v_η s_η

Sample η bits

Verifier generates η public unbiased coins



b_1 b_η

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

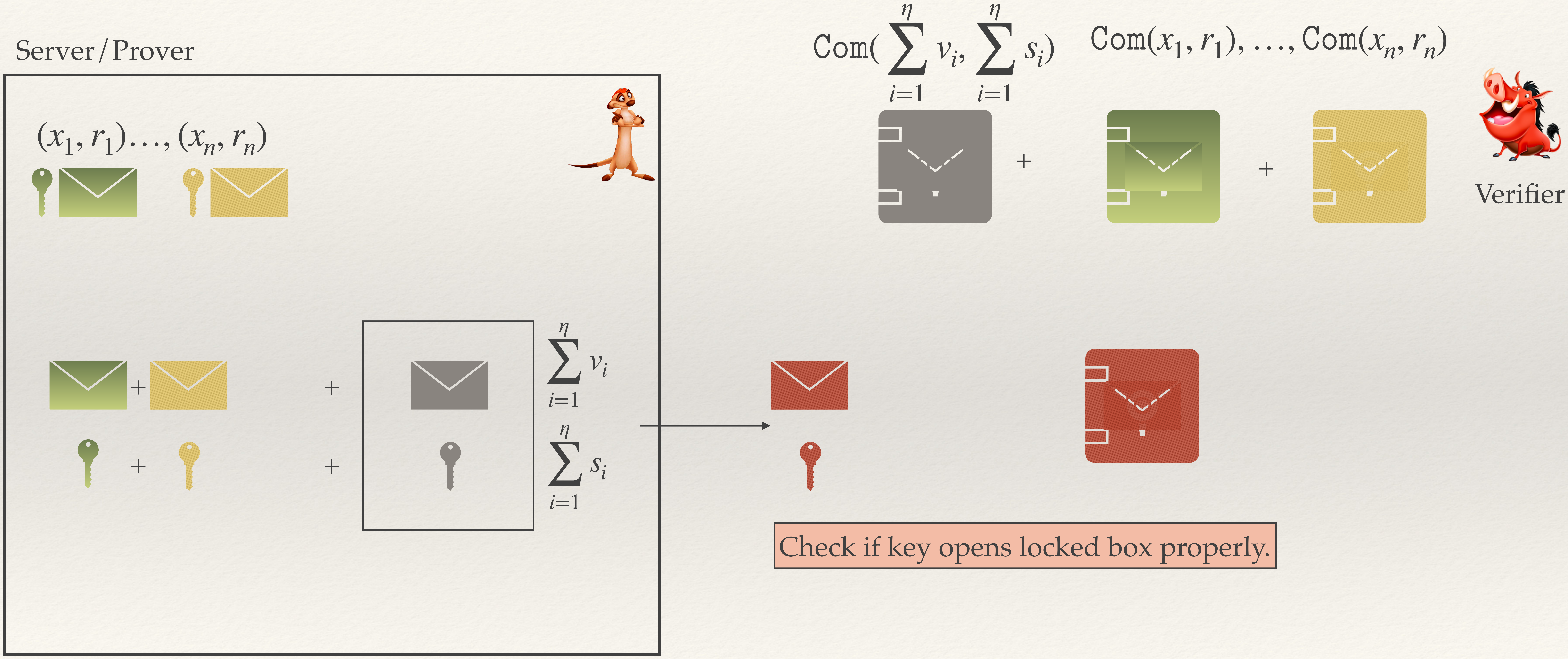
Otherwise, leave v_i unchanged.

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

(ϵ, δ)-Computational DP

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

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

$X \in \mathcal{X}^n$ $\boxed{x_1}\boxed{\dots}\boxed{x_{n-1}}\boxed{x_n}$ $\longrightarrow$

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

M $\boxed{Q(x_1, \dots, x_n)}$ $\longrightarrow$ $\boxed{M(X, Q)}$ $\boxed{M(X', Q)}$

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

$$\Pr[D(\boxed{M(X,Q)} \in T) = 1] \le e^{\epsilon} \Pr[D(\boxed{M(X',Q)} \in T) = 1] + \delta$$

~~OPEN~~ PROBLEM
Is there a significant advantage if we relaxed this to be computational indistinguishability instead of statistical ?

$\boxed{M(X', Q)}$ $\boxed{M(X, Q)}$

????

Was X released or was it X'

Polynomially bounded algorithm D

Prior Separation Attempts

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

If $\mathcal{Y} \subseteq \mathbb{R}^d$ and utility is measured in terms of the L_p norm then there is NO advantage to relaxing privacy.

$$\boxed{u(X, M(X,Q) = \| M(X,Q) - Q(X) \|_p}$$

GKY11- TCC
Limits of computational differential privacy in the client server setting

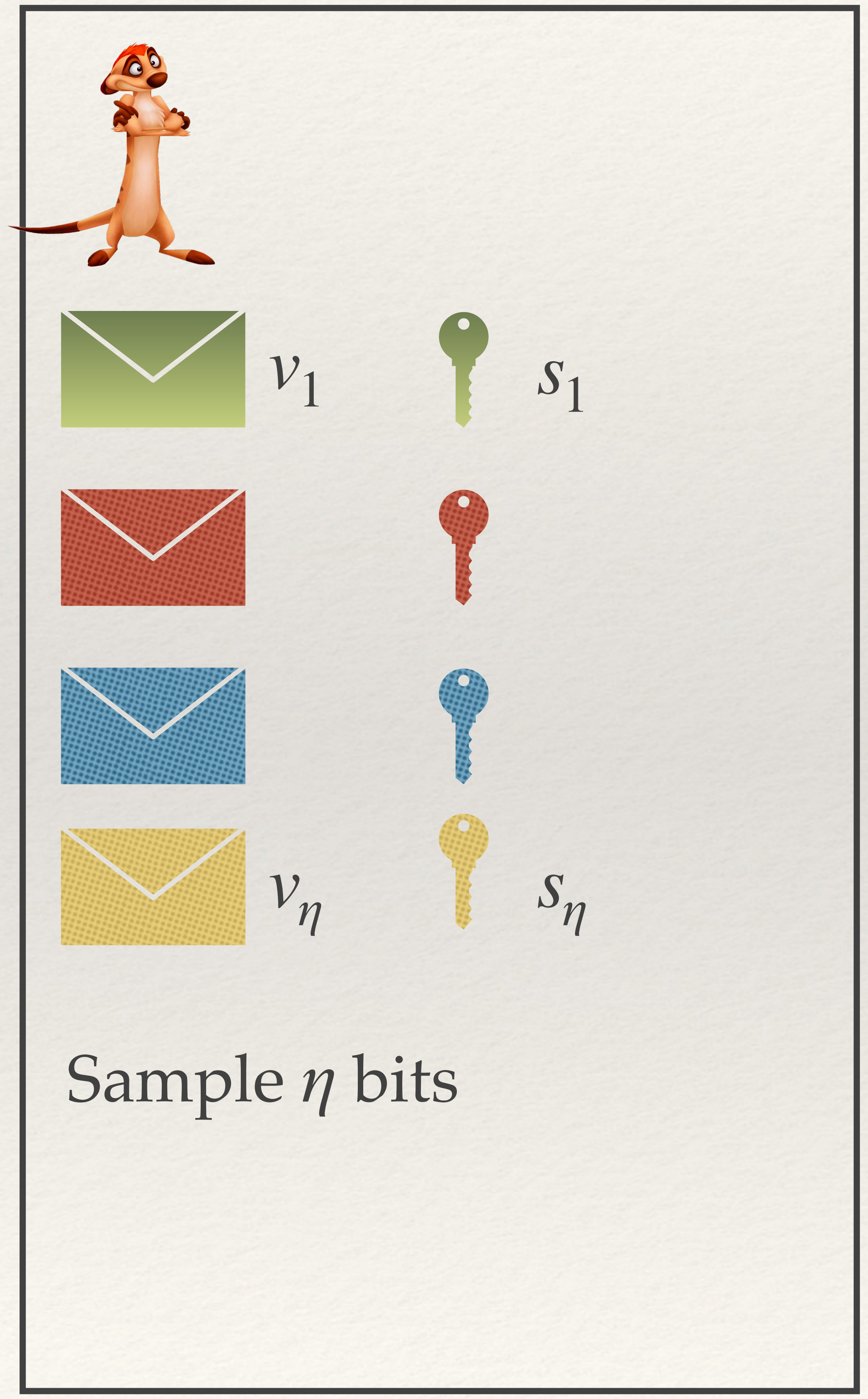
Thus $\mathcal{Y}$ needs to be a more complex structure like a circuit, a graph or a proof.

GIKKM23-Arxiv
Separating Computational and Statistical Differential Privacy (Under Plausible Assumptions)

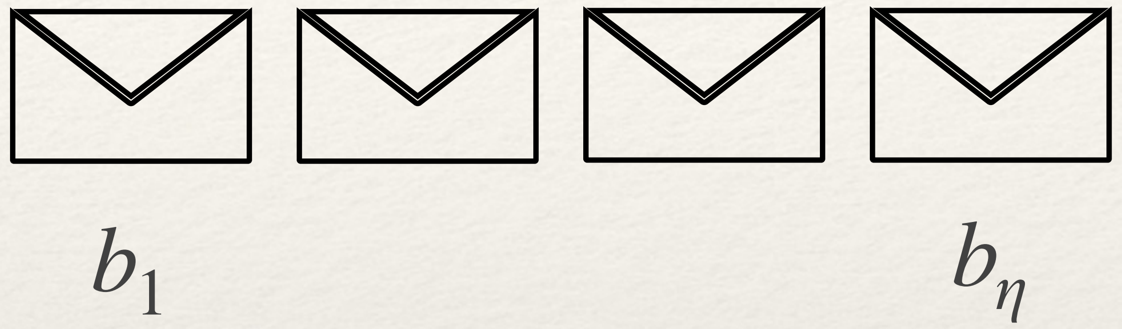
BV16 - TCC
Separating Computational and Statistical Differential Privacy in the Client-Server Model

Where's the Separation ?

Server / Prover

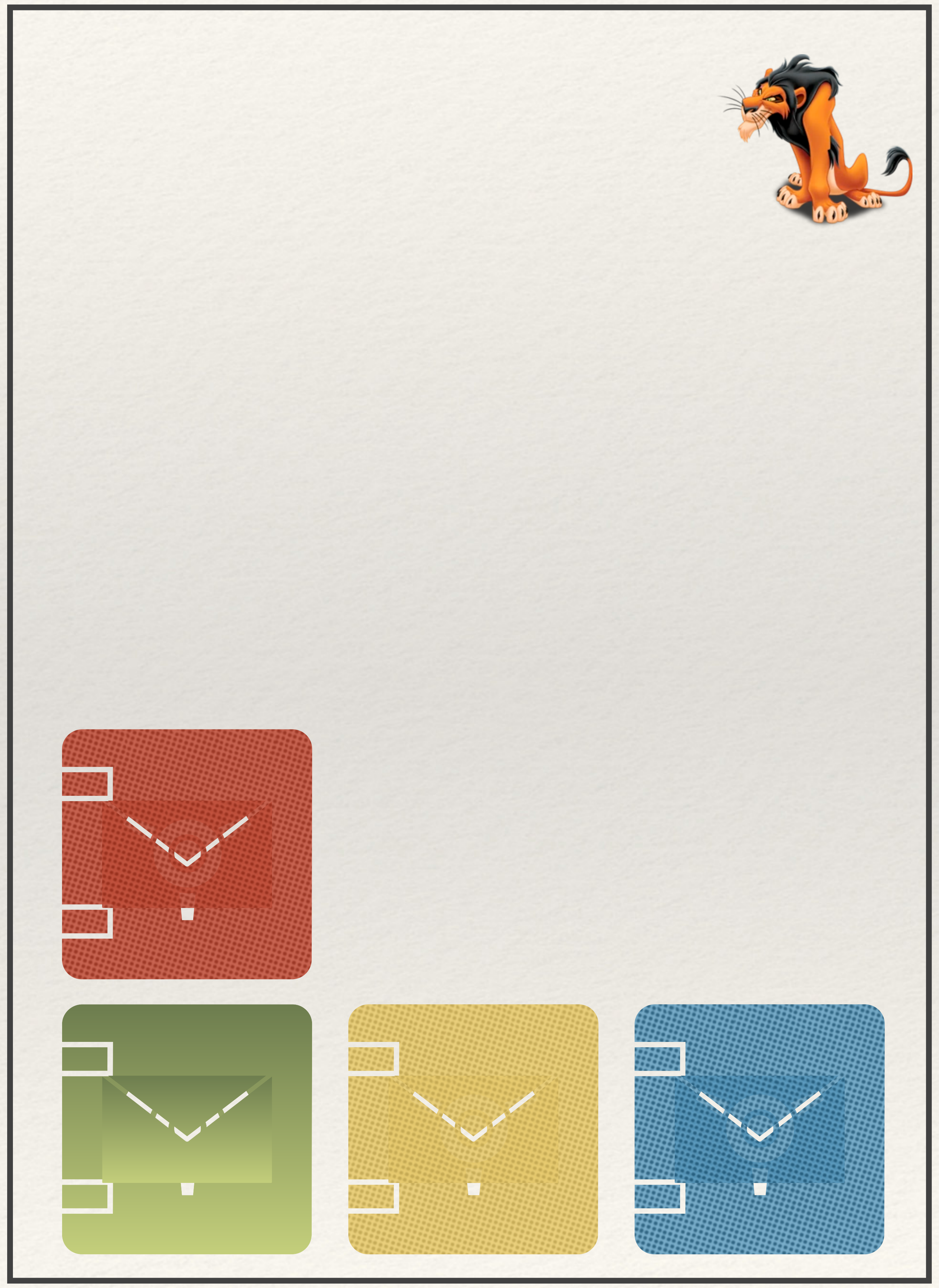


Generate η public unbiased coins by playing Morra



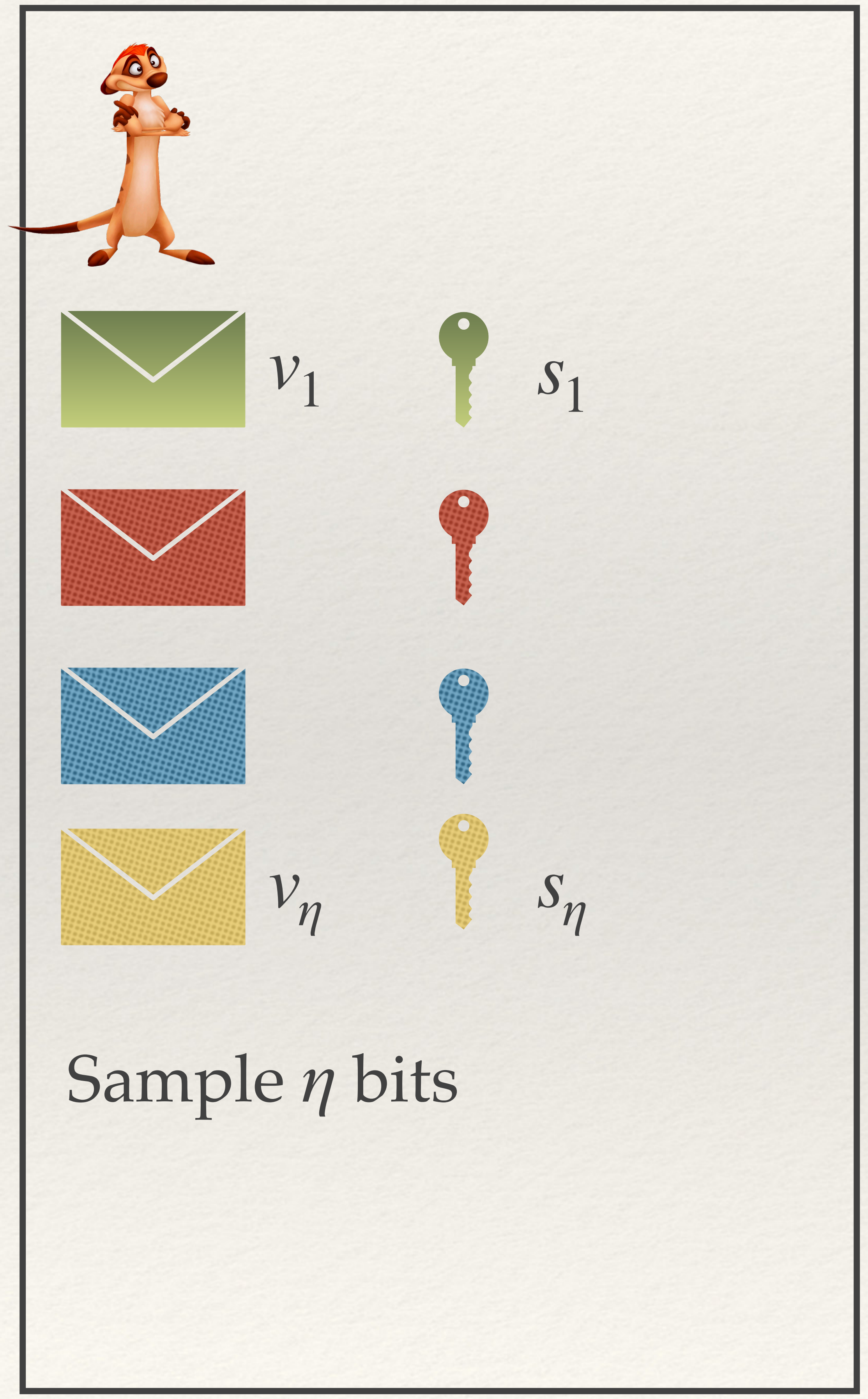
A key component in verifying the servers DP noise was to generate unbiased public randomness.

Verifier

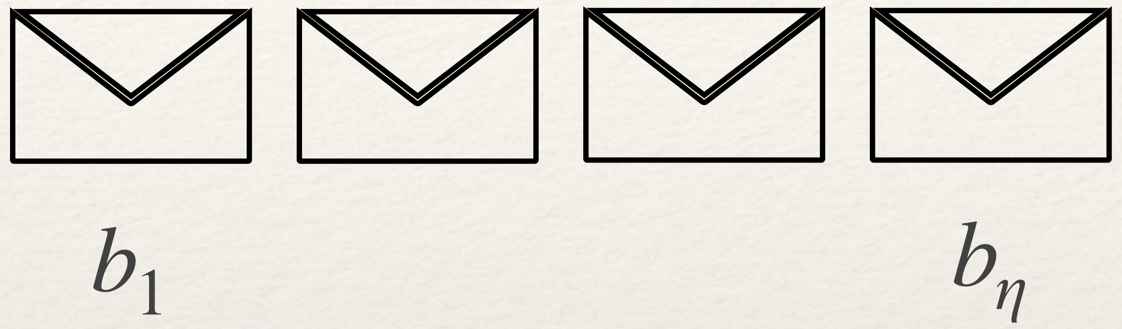


Where's the Separation ?

Server / Prover



Generate η public unbiased coins by playing Morra



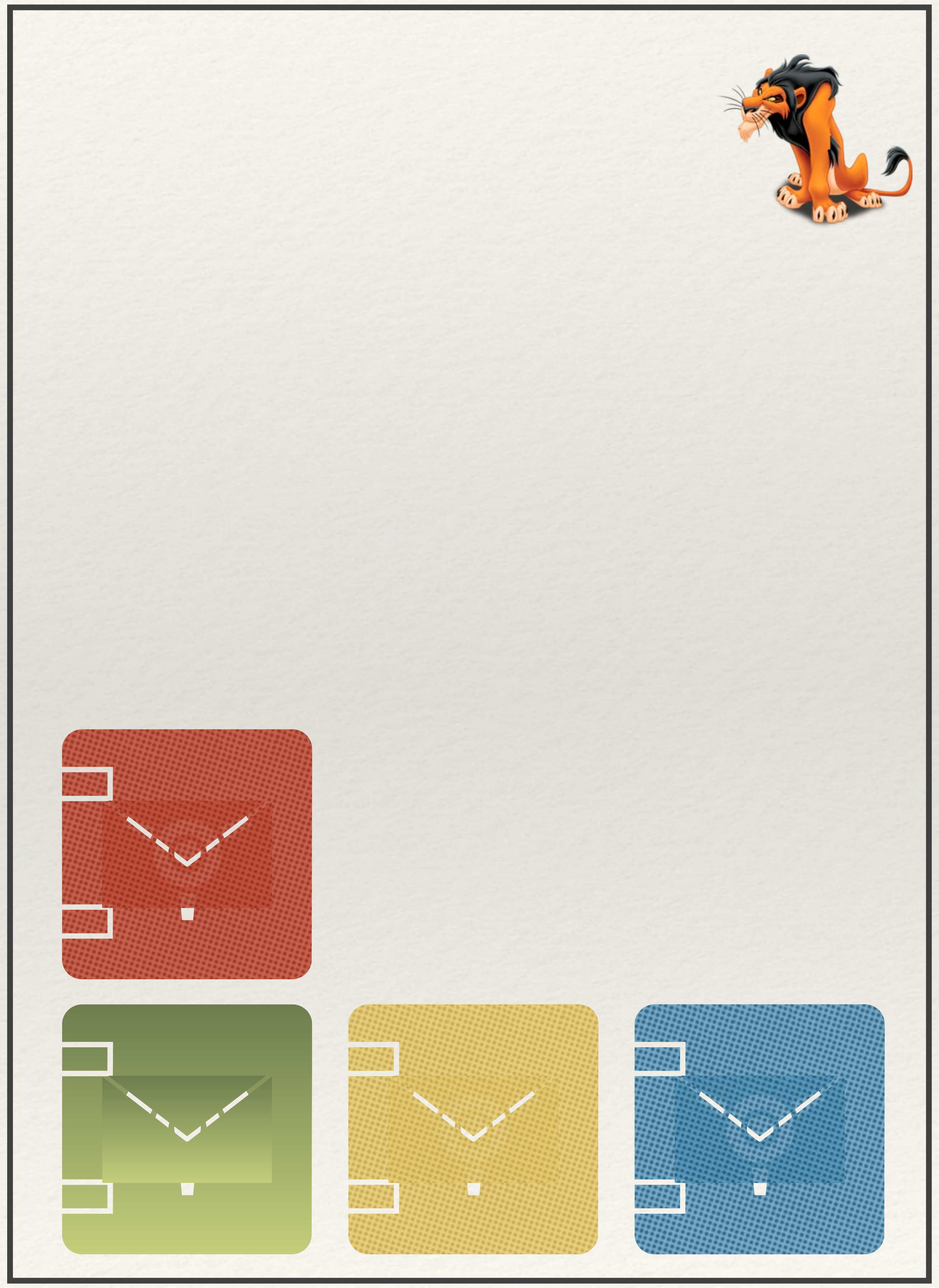
A key component in verifying the servers DP noise was to generate unbiased public randomness.

Coin-flipping $\Longrightarrow$ One way Functions $\Longrightarrow$ Commitments

Coin Flipping with Constant Bias Implies One way Functions - HO14

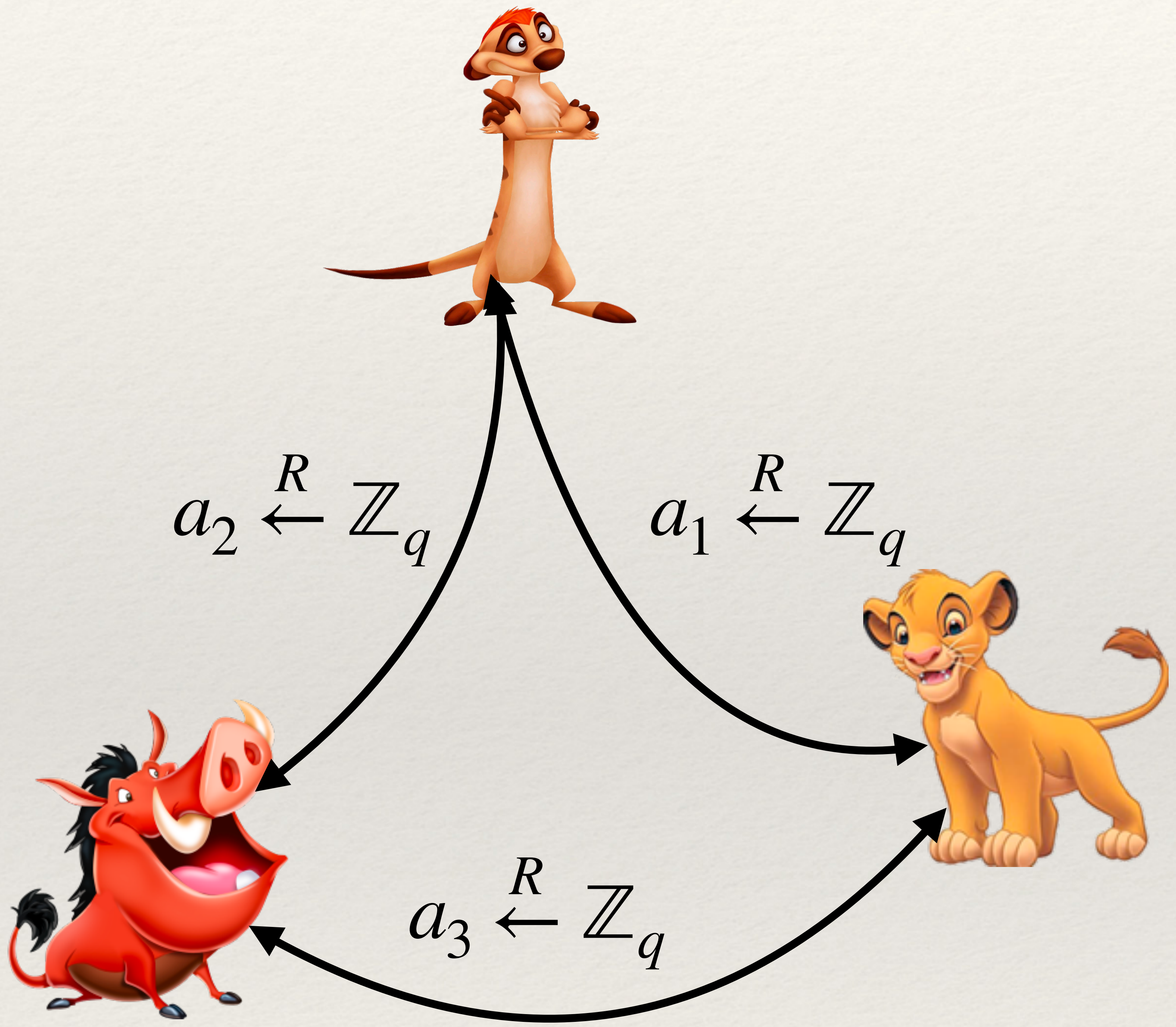
Coin Flipping with *any* Constant Bias Implies One way Functions - BHT21

Verifier



Questions

Public Coin Flipping (Morra)



1. Each party samples a random value from $\mathbb{Z}_q$
2. Each party broadcasts a commitment to the value.
3. Each party opens their commitments in the reverse order in which they broadcasted commitments.
4. Everyone checks all the opens are good.
5. $\tilde{b} = \sum_{i=1}^{K} a_i \mod q$
6. If $\tilde{b} \leq \frac{q}{2}$, we set $b = 1$
7. Else $b = 0$

As long as a single party is honest, this protocol generates an unbiased bit b