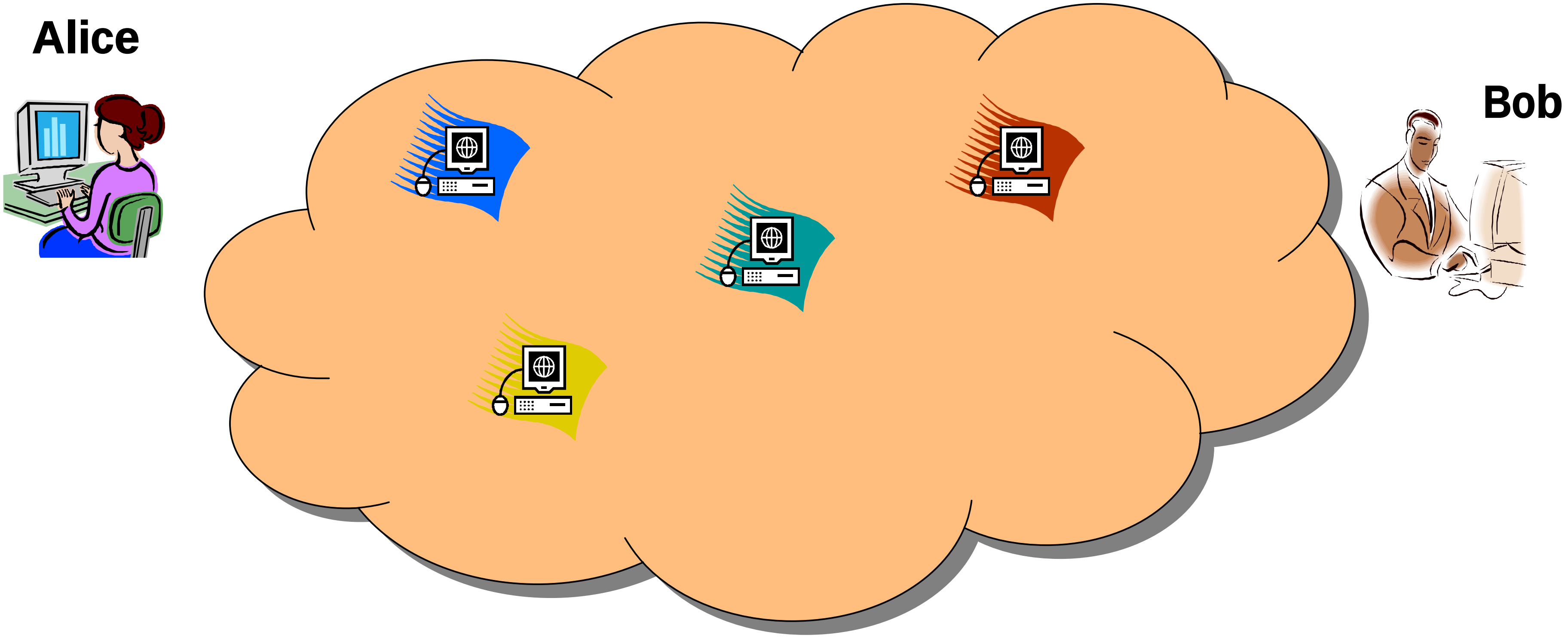
Slicing the Onion: Anonymity Without PKI

Sachin Katti

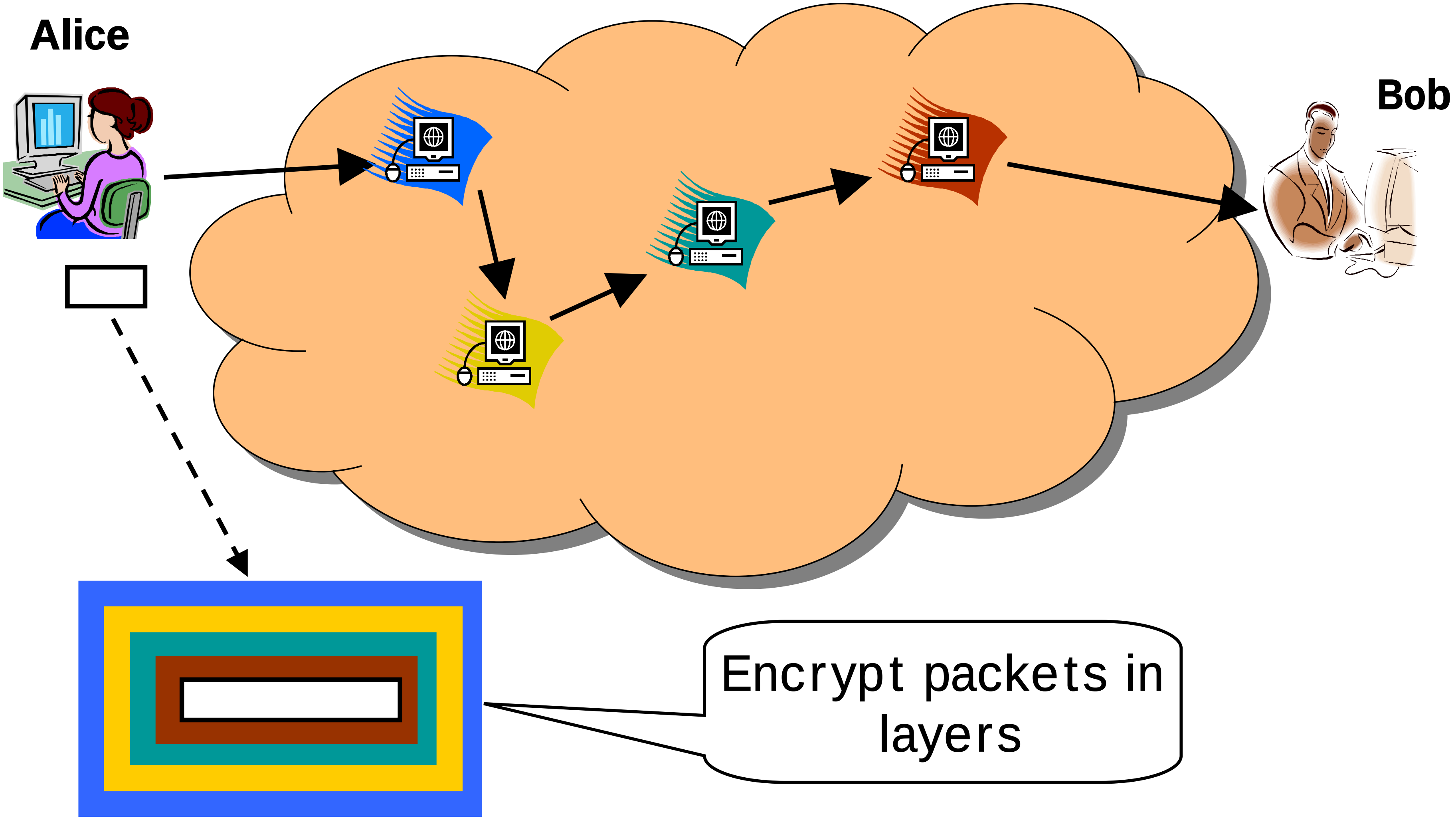
Dina Katabi & Katya Puchala

State of the art: Onion Routing over P2P

State of the art: Onion Routing over P2P

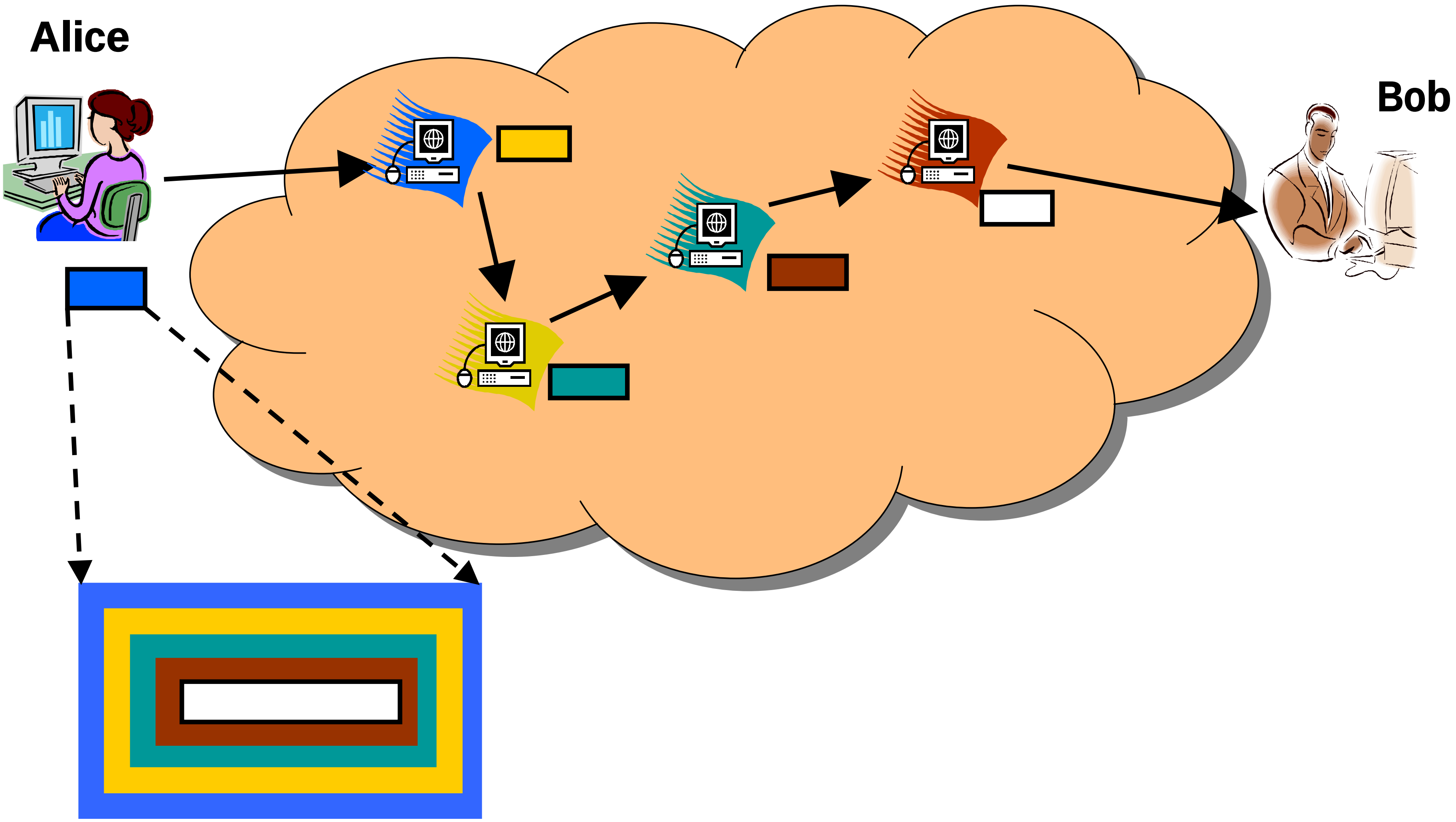


State of the art: Onion Routing over P2P

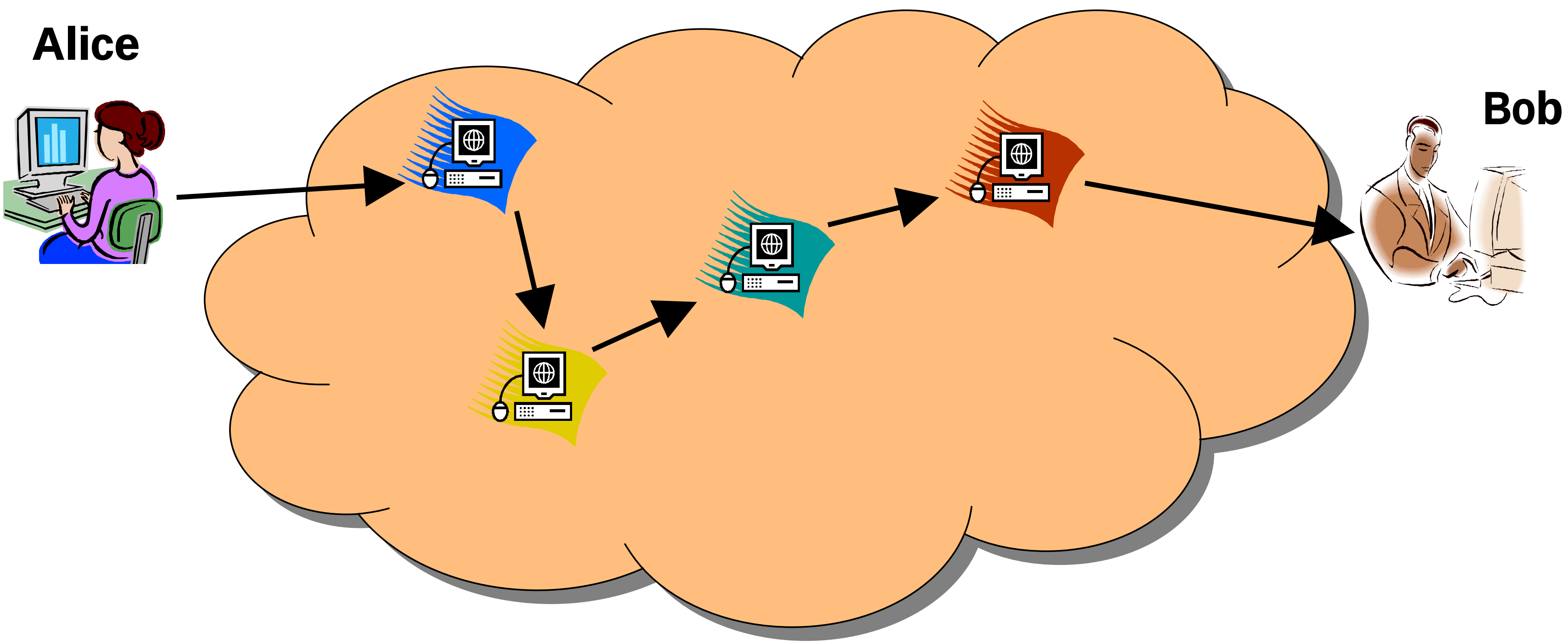


Encrypt packets in layers

State of the art: Onion Routing over P2P

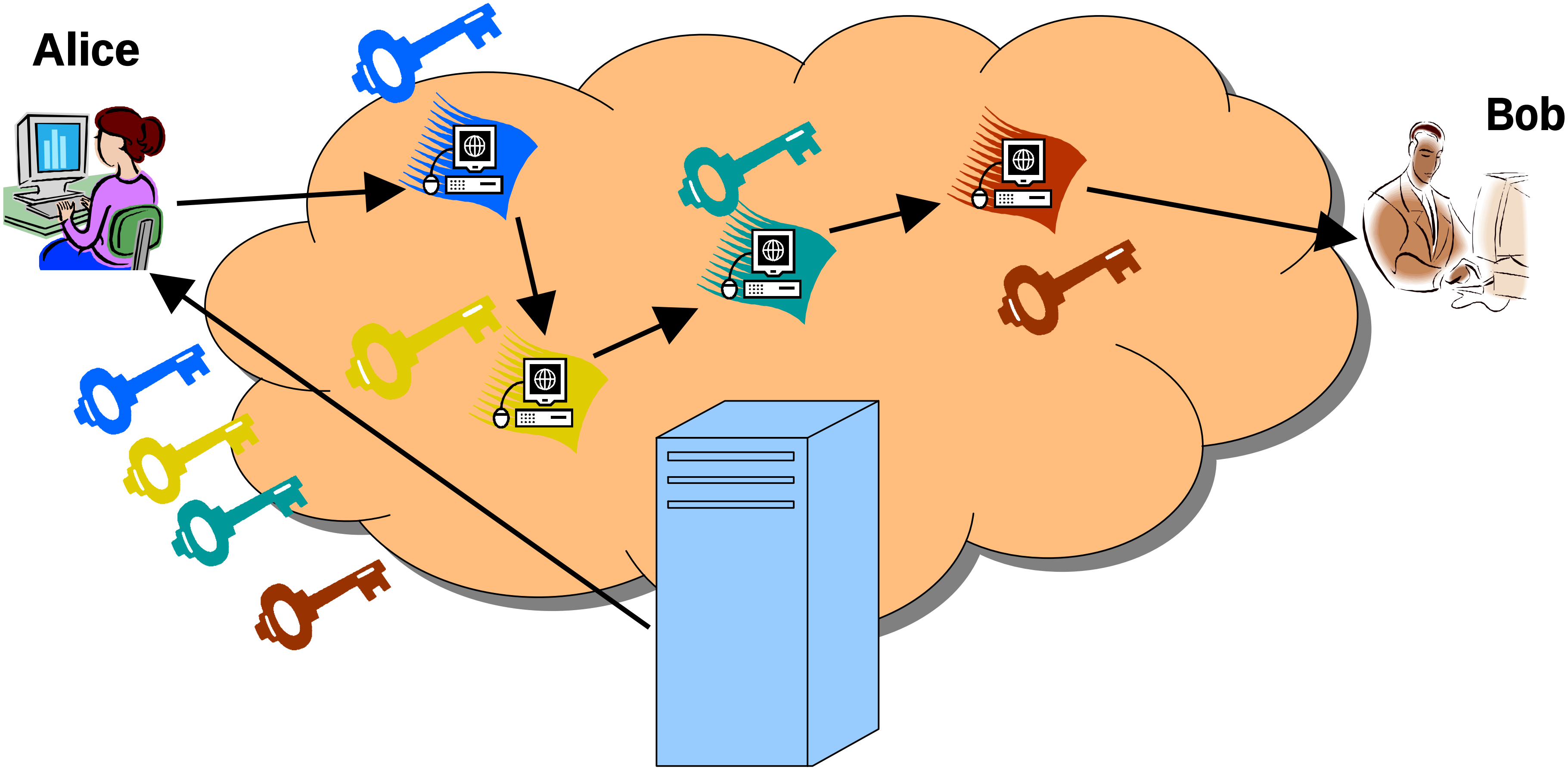


State of the art: Onion Routing over P2P



- Each node only knows its previous hop and next hop
- Bob does not know the identity of Alice either

What’s the catch?



Alice

Bob

Centralized trusted PKI

PKI Showstoppers!

- Key distribution
- Key updates
- Compulsion attacks
- Trust model

Can we have anonymity without PKI?

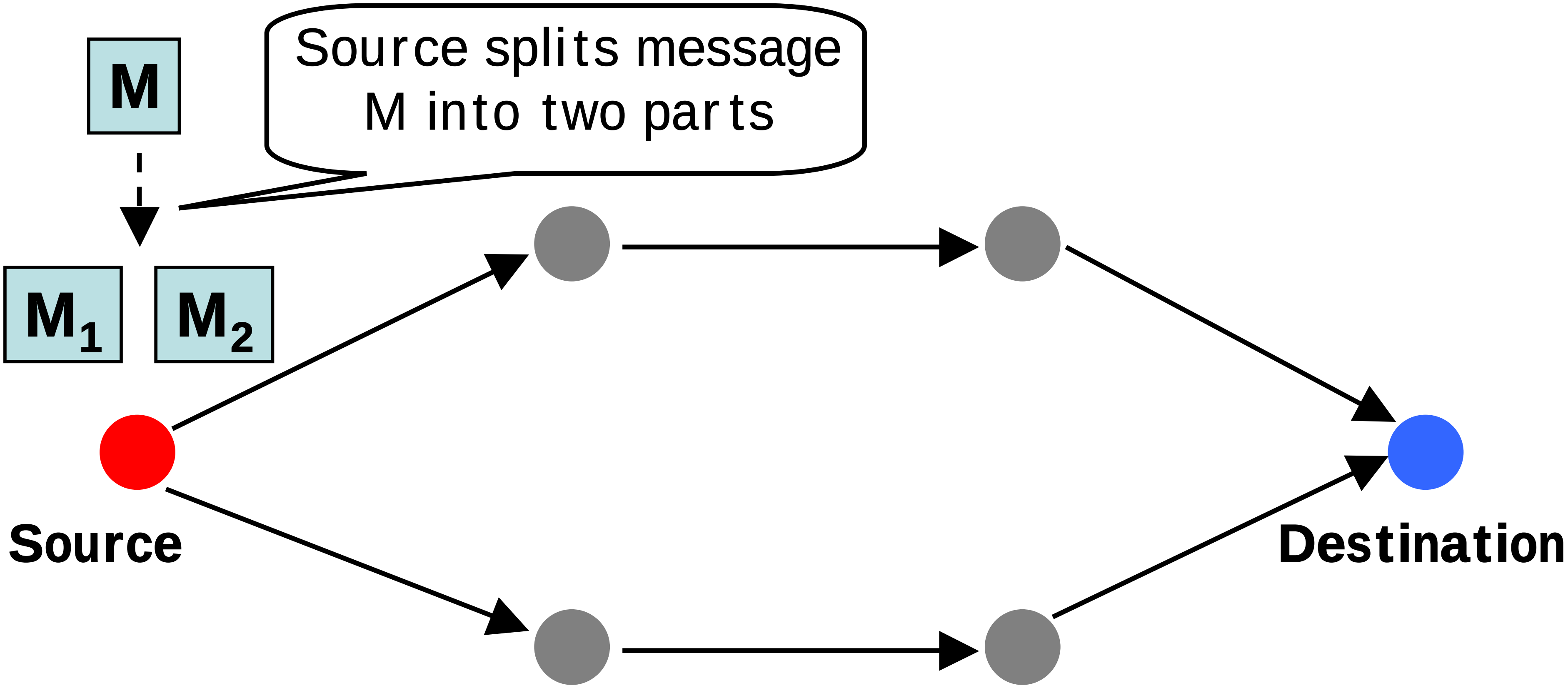
This talk...

How to do anonymous communication without PKI

What kind of anonymity?

- Message confidentiality
- Source anonymity
- Destination anonymity

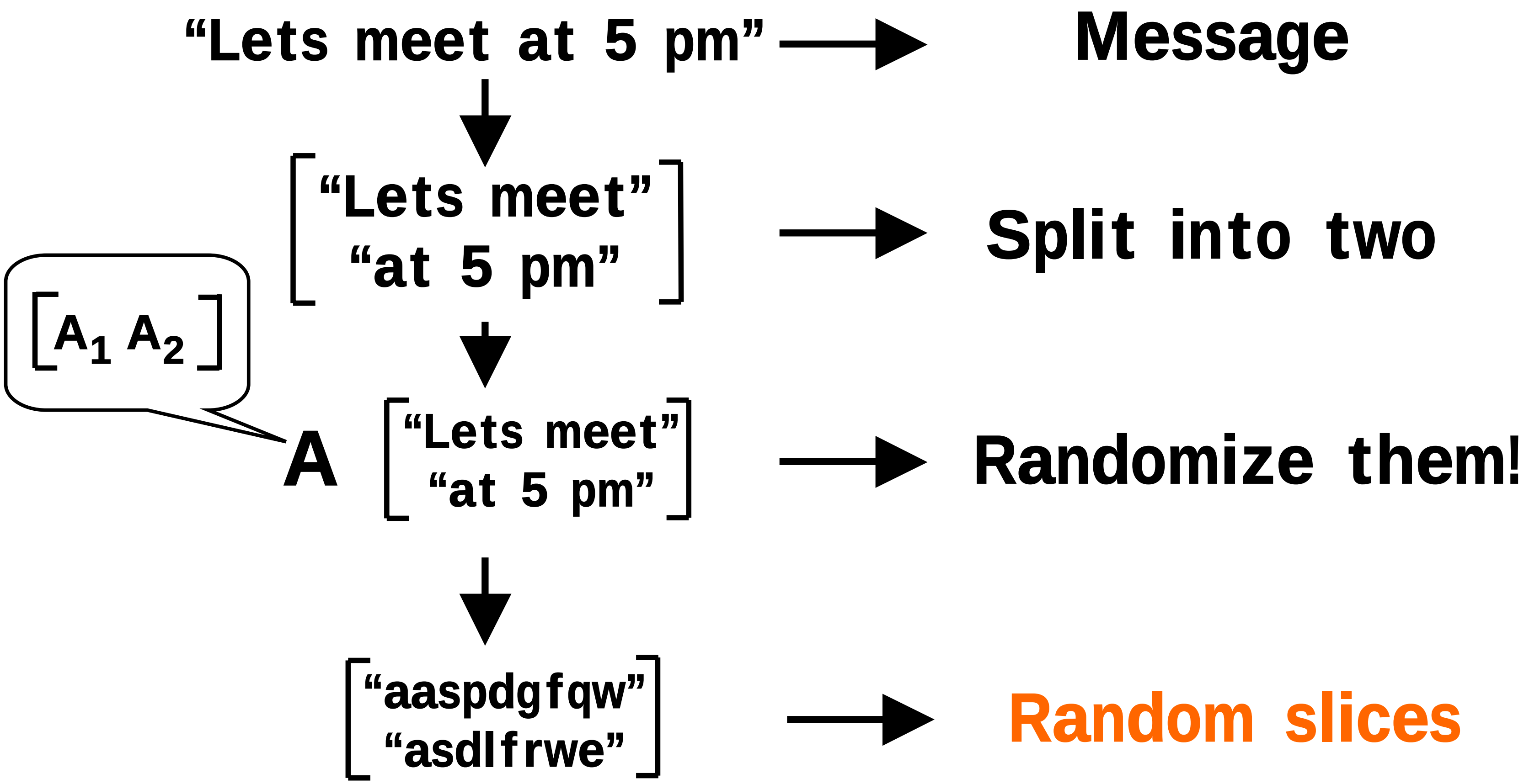
Confidentiality without PKI



Source splits message M into two parts

Source sends **M_1** and **M_2** along node disjoint paths

Confidentiality without PKI



“Lets meet at 5 pm” → Message

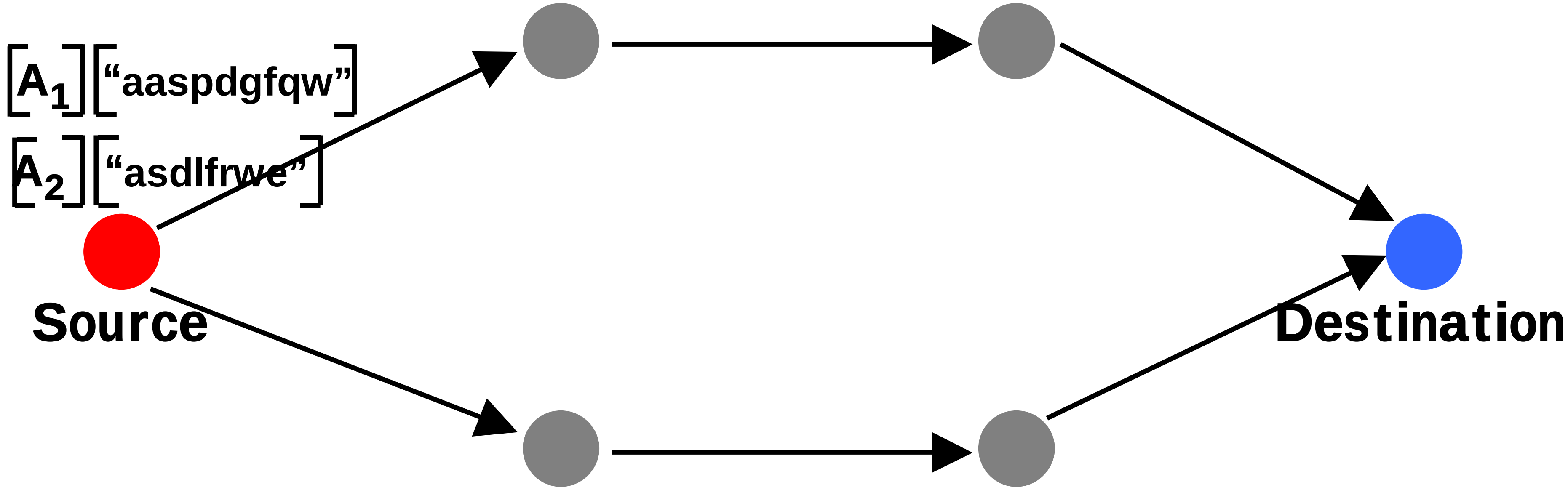
[“Lets meet” “at 5 pm”] → Split into two

$[A_1 \; A_2]$

A [“Lets meet” “at 5 pm”] → Randomize them!

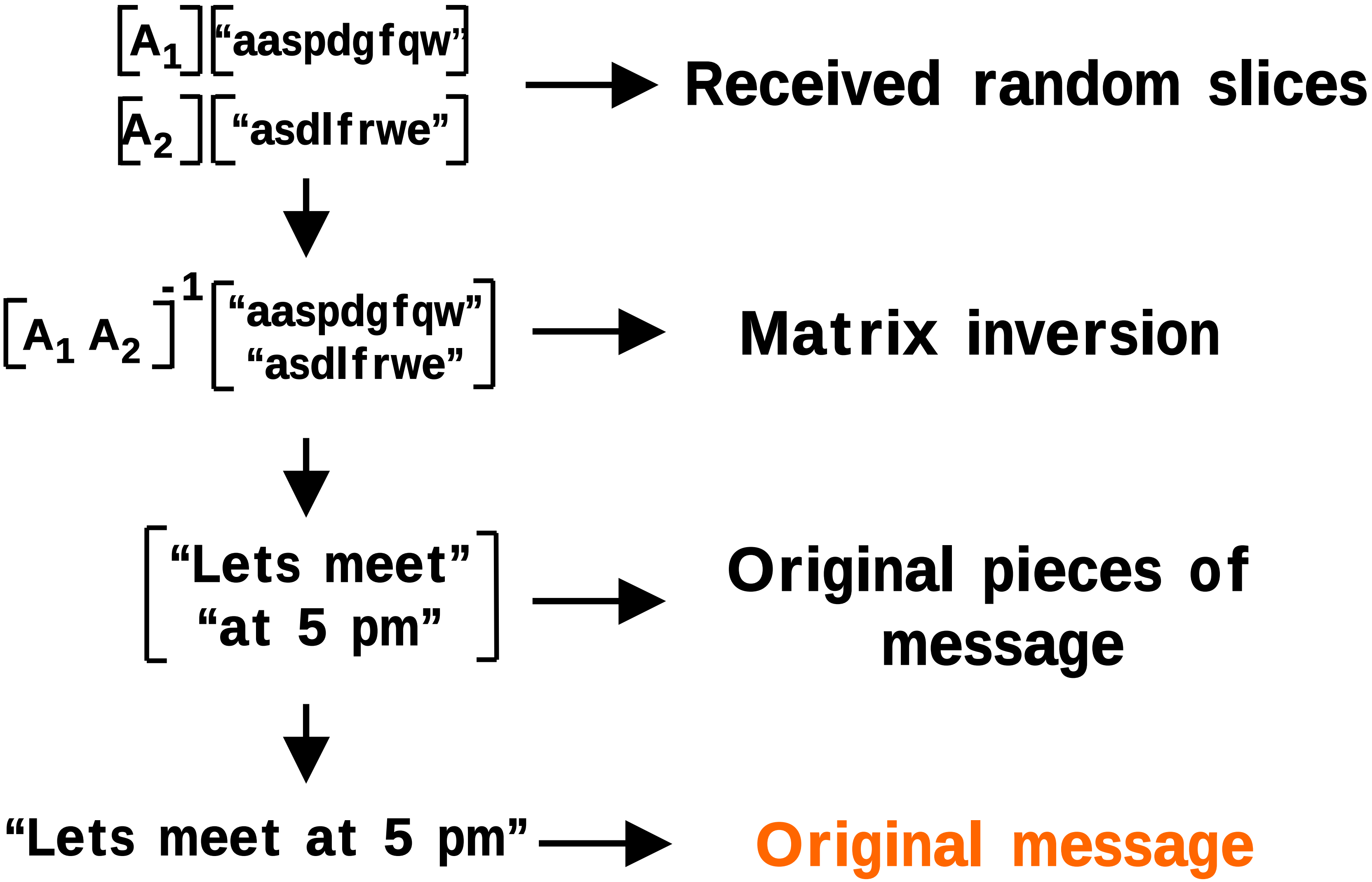
[“aaspdgfqw” “asdlfrwe”] → Random slices

Confidentiality without PKI



Reconstruct original information from the slices

Confidentiality without PKI



$$\begin{bmatrix}A_1\end{bmatrix}\begin{bmatrix}\text{“aaspdgfqw”}\end{bmatrix}$$

$$\begin{bmatrix}A_2\end{bmatrix}\begin{bmatrix}\text{“asdlfrwe”}\end{bmatrix}$$

Received random slices

$$\begin{bmatrix}A_1 & A_2\end{bmatrix}^{-1}\begin{bmatrix}\text{“aaspdgfqw”}\\ \text{“asdlfrwe”}\end{bmatrix}$$

Matrix inversion

$$\begin{bmatrix}\text{“Lets meet”}\\ \text{“at 5 pm”}\end{bmatrix}$$

Original pieces of message

“Lets meet at 5 pm”

Original message

What about anonymity?

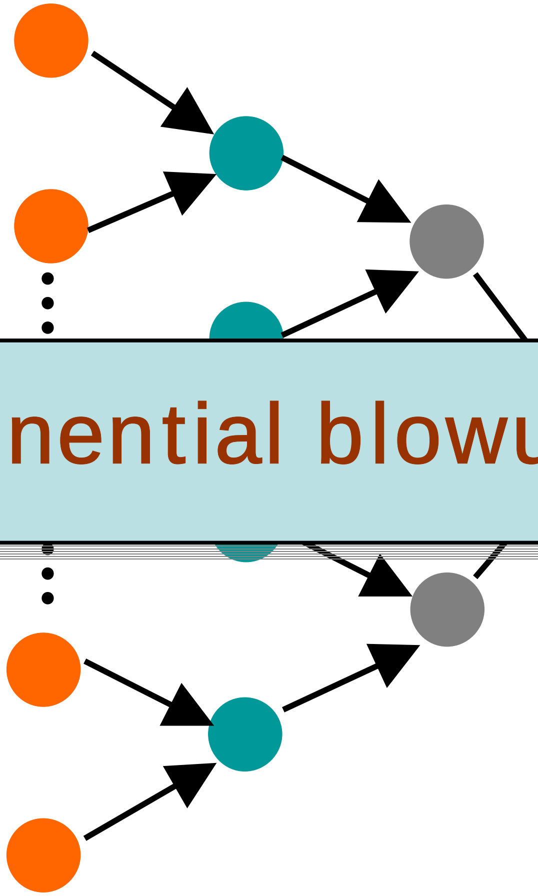
Idea : Build anonymity from confidentiality

What about anonymity?

Idea : Build anonymity from confidentiality

Source tells each relay the ID of its next hop in a confidential message

Challenge



Exponential blowup!

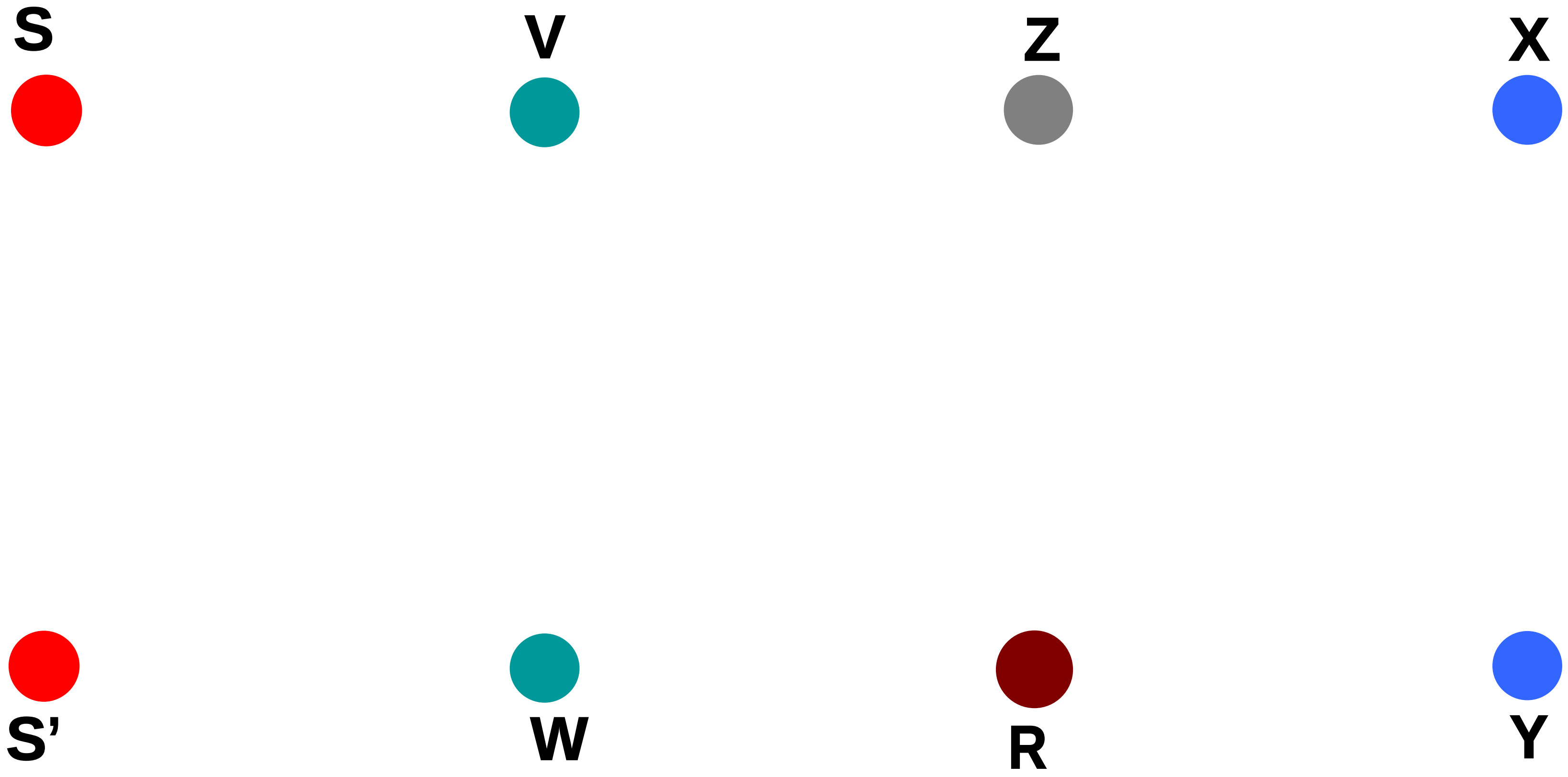
Challenge : Exponential Blowup

Solution : Node Reuse

Illustrative Example

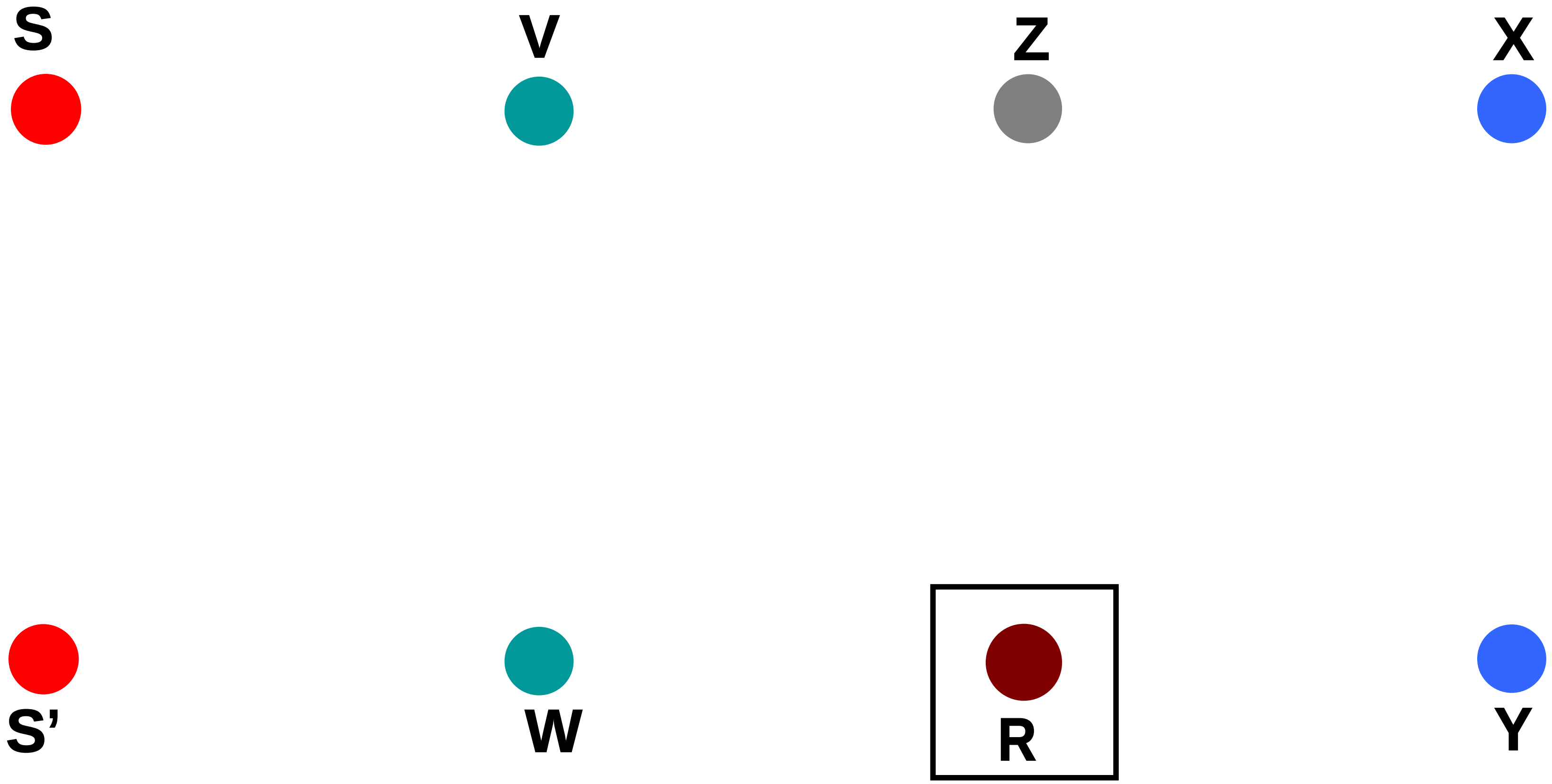
Source has multiple IP addresses

Illustrative Example



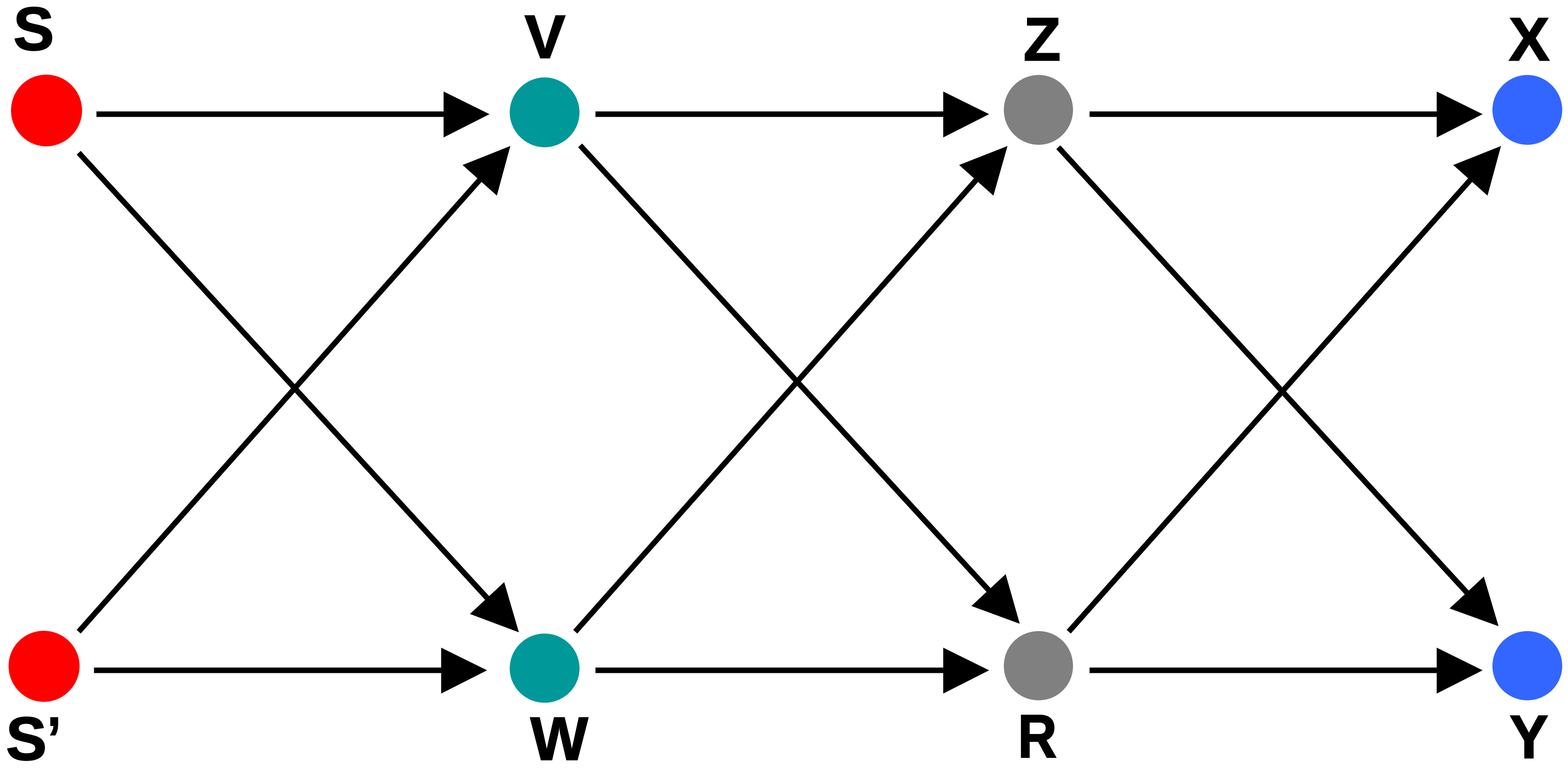
Source picks relays and organizes them in stages

Illustrative Example

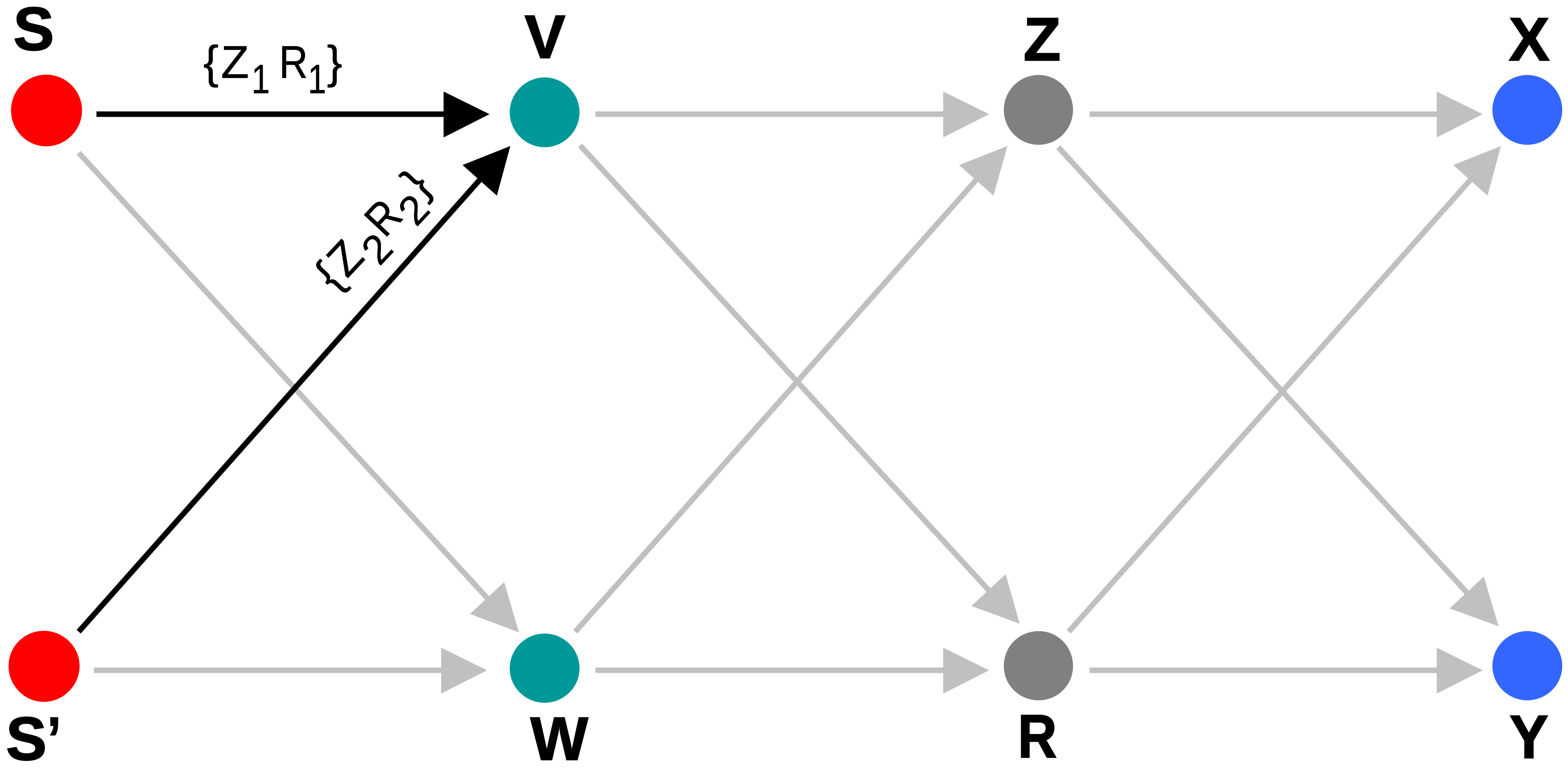


Destination is placed randomly

Illustrative Example

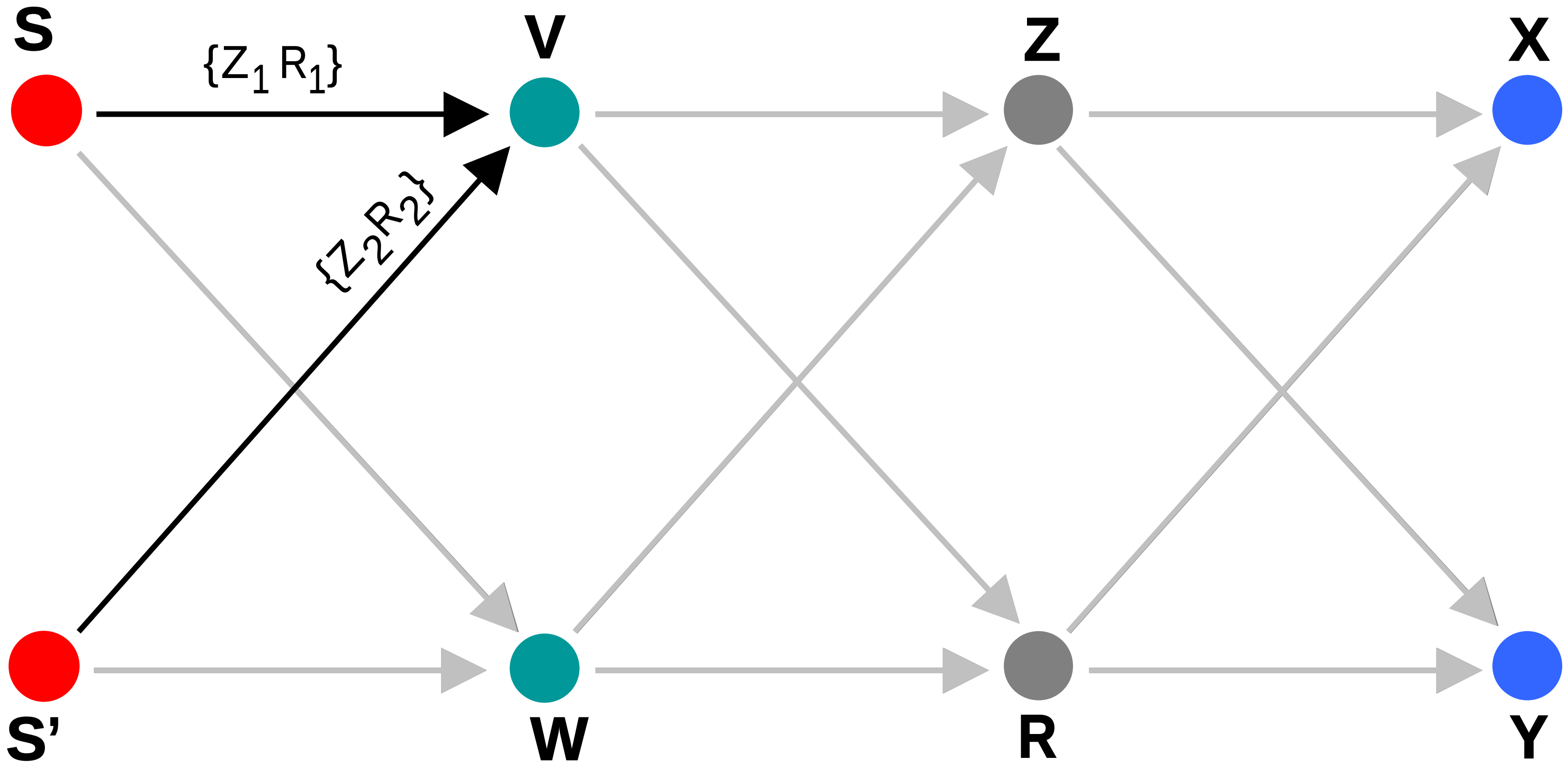


Illustrative Example



V needs to know Z and R

Illustrative Example



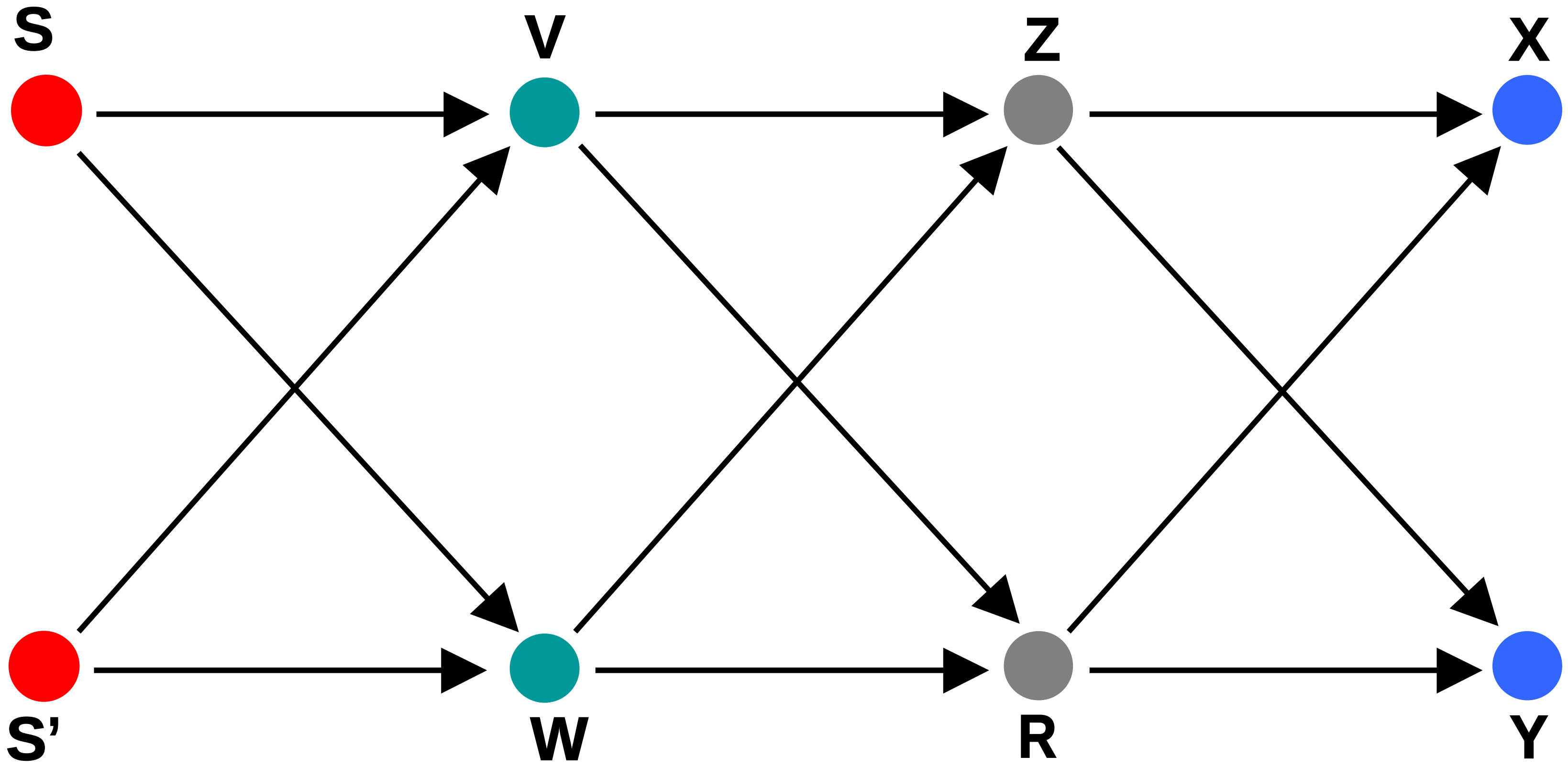
$\{Z_1 \;\; R_1\}$
$\{Z_2 \;\; R_2\}$

$\downarrow$

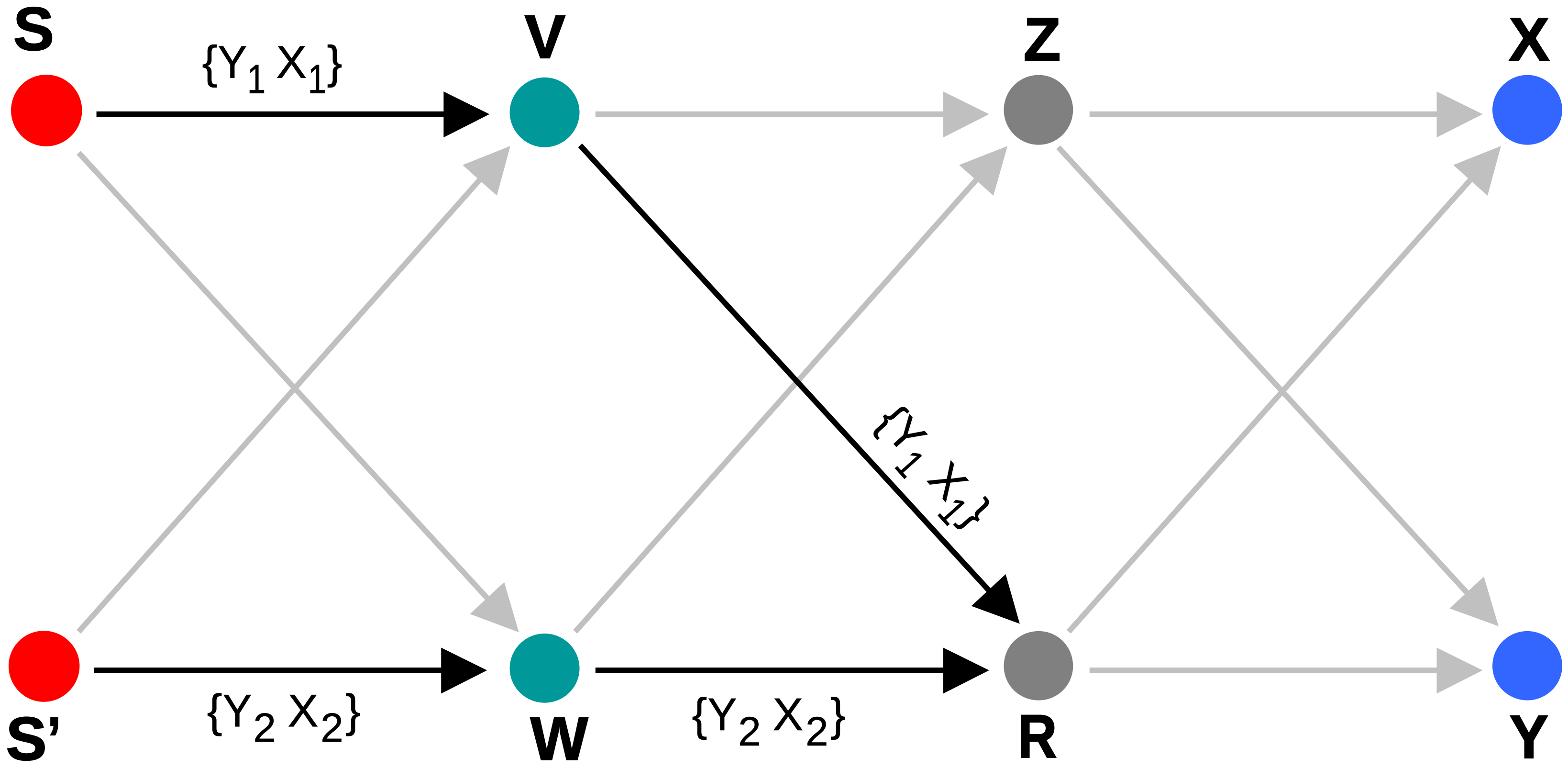
$\{Z \;\; R\}$

V combines the two slices to get its next hops Z and R

Illustrative Example



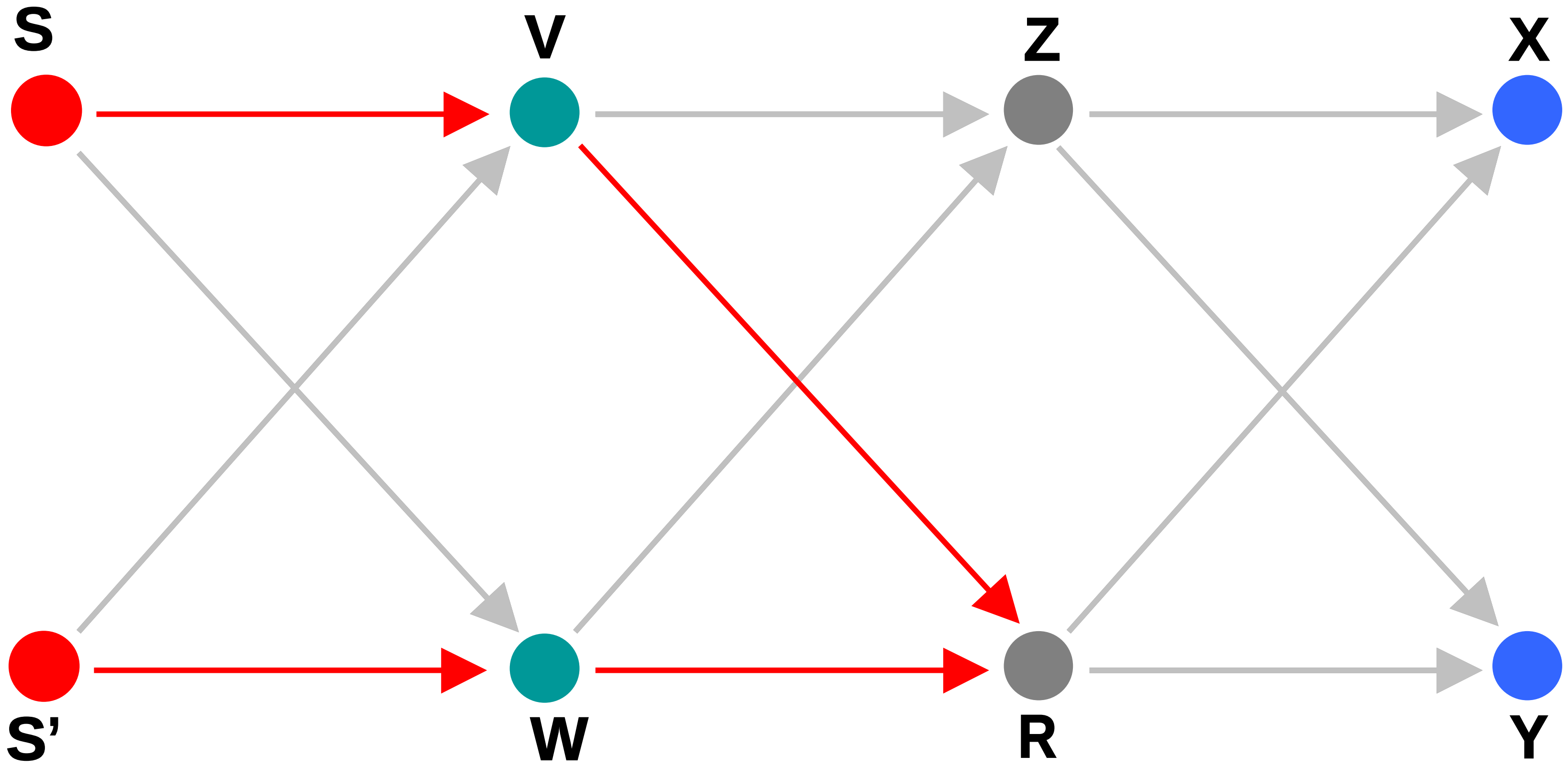
Illustrative Example



R can combine incoming slices to get X and Y

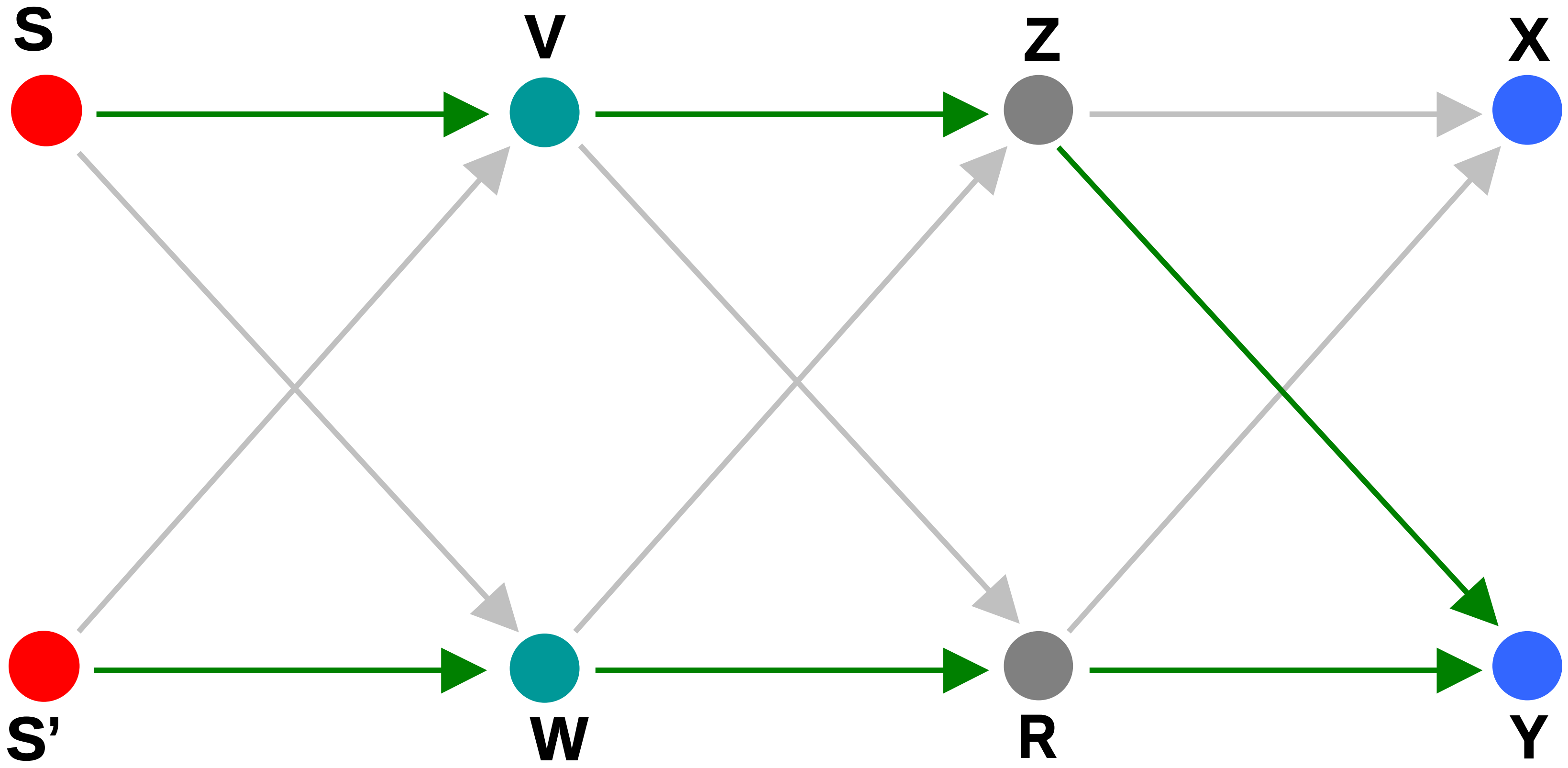
R needs to know X and Y

Illustrative Example



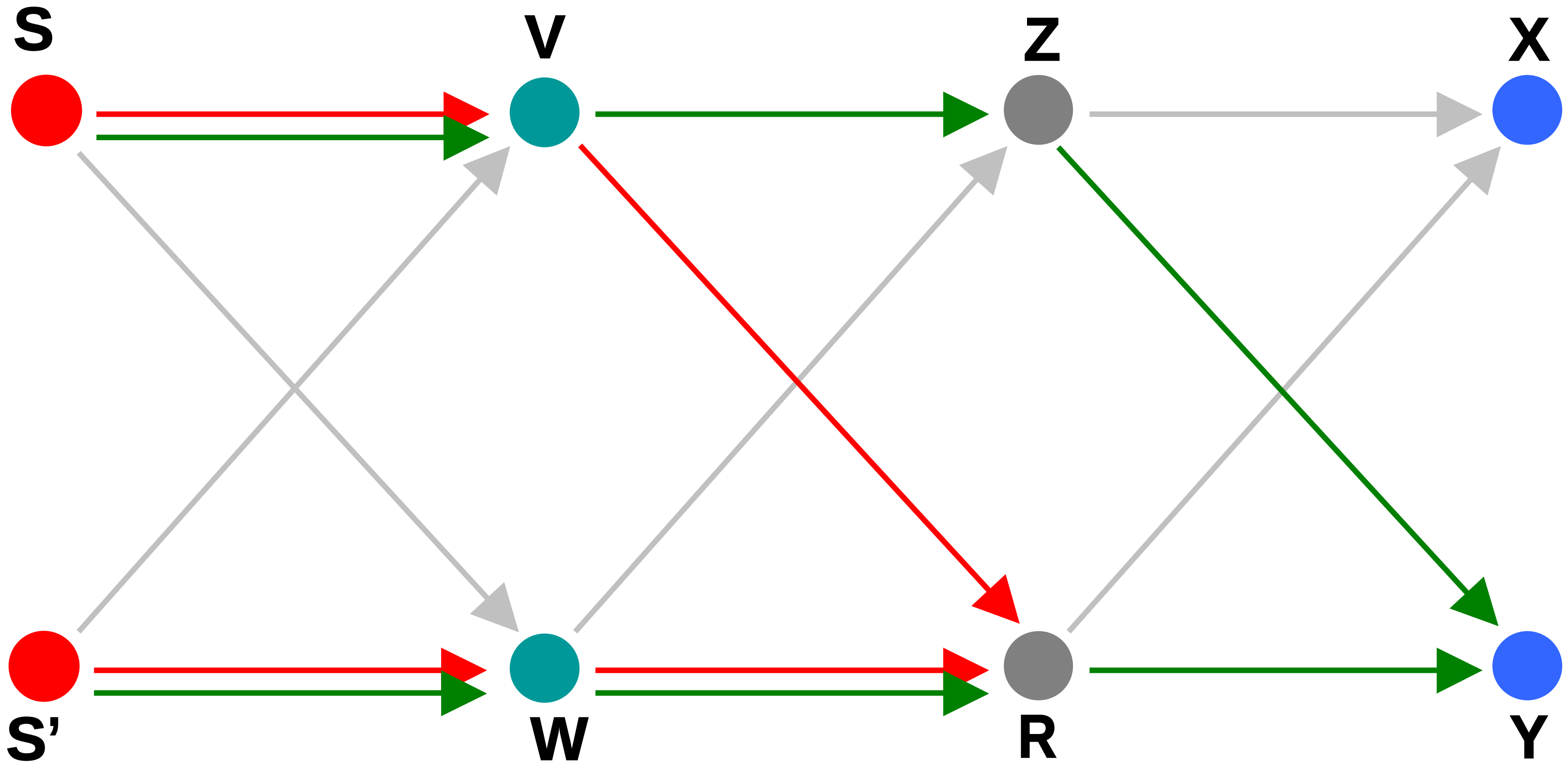
Node disjoint paths to R

Illustrative Example



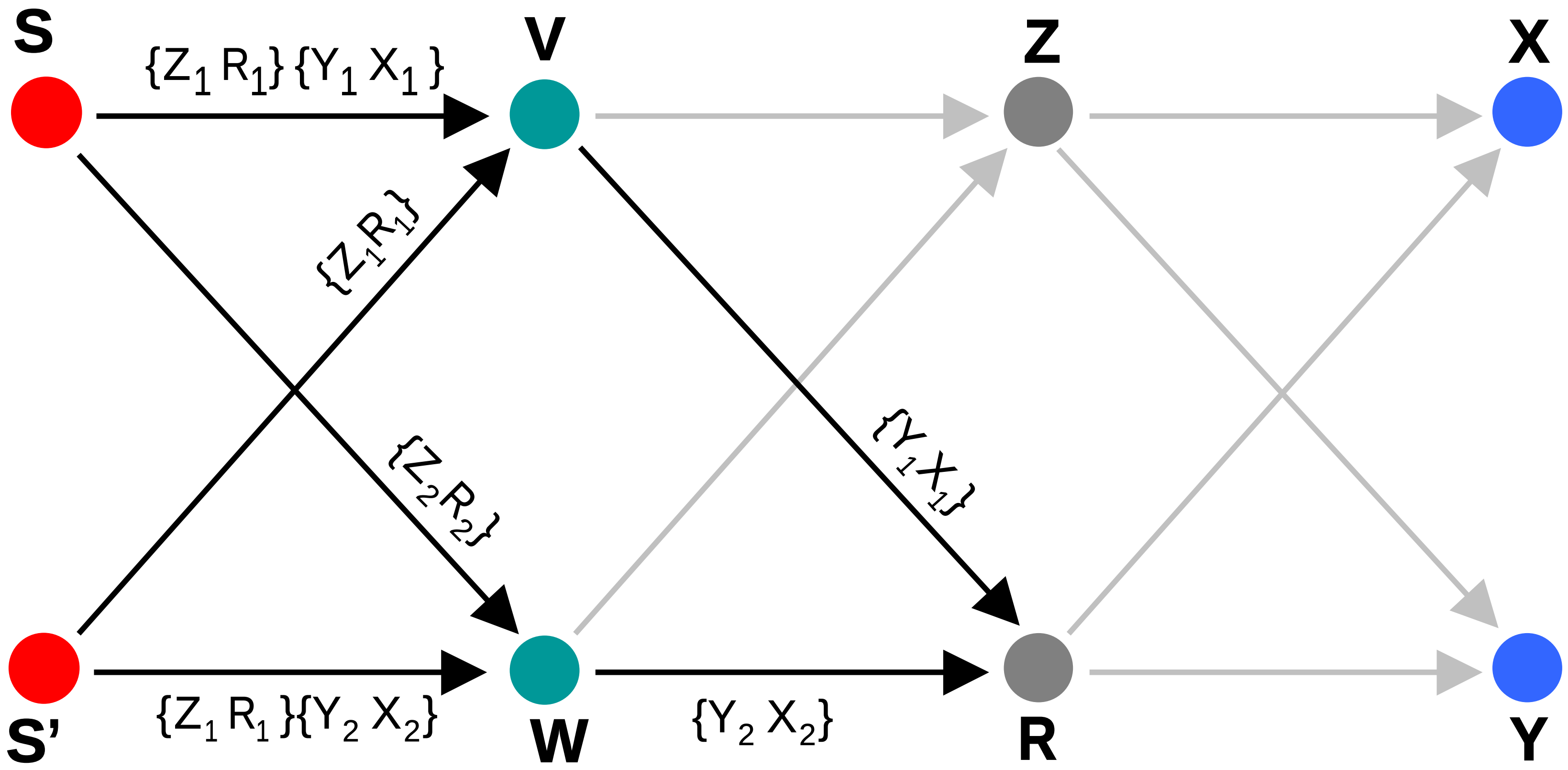
Node disjoint paths to Y

Illustrative Example



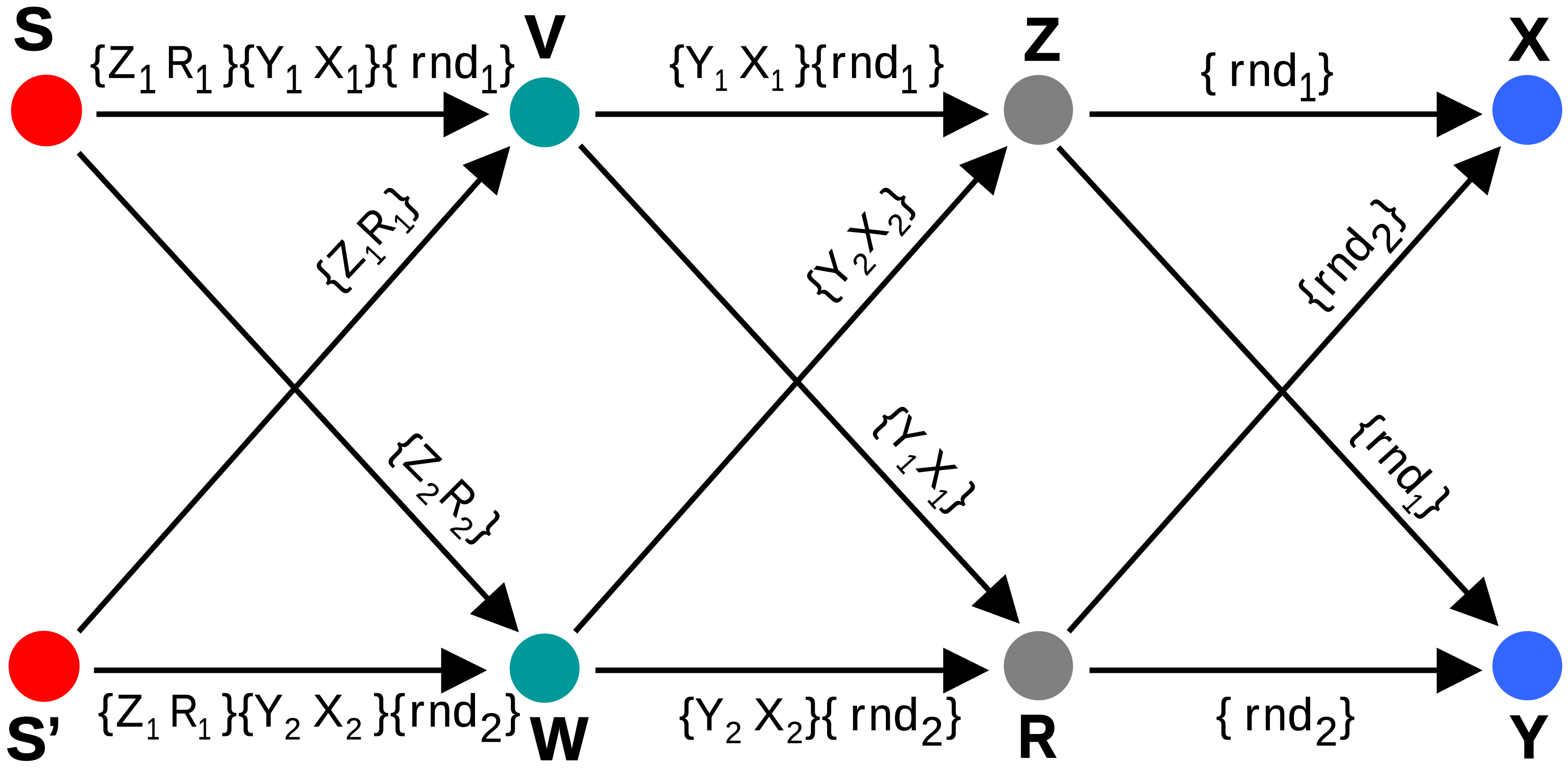
Node V is reused to construct disjoint paths to R and Y

Illustrative Example



Send slices in the same packet

Illustrative Example



Small number of nodes

Slicing Protocol

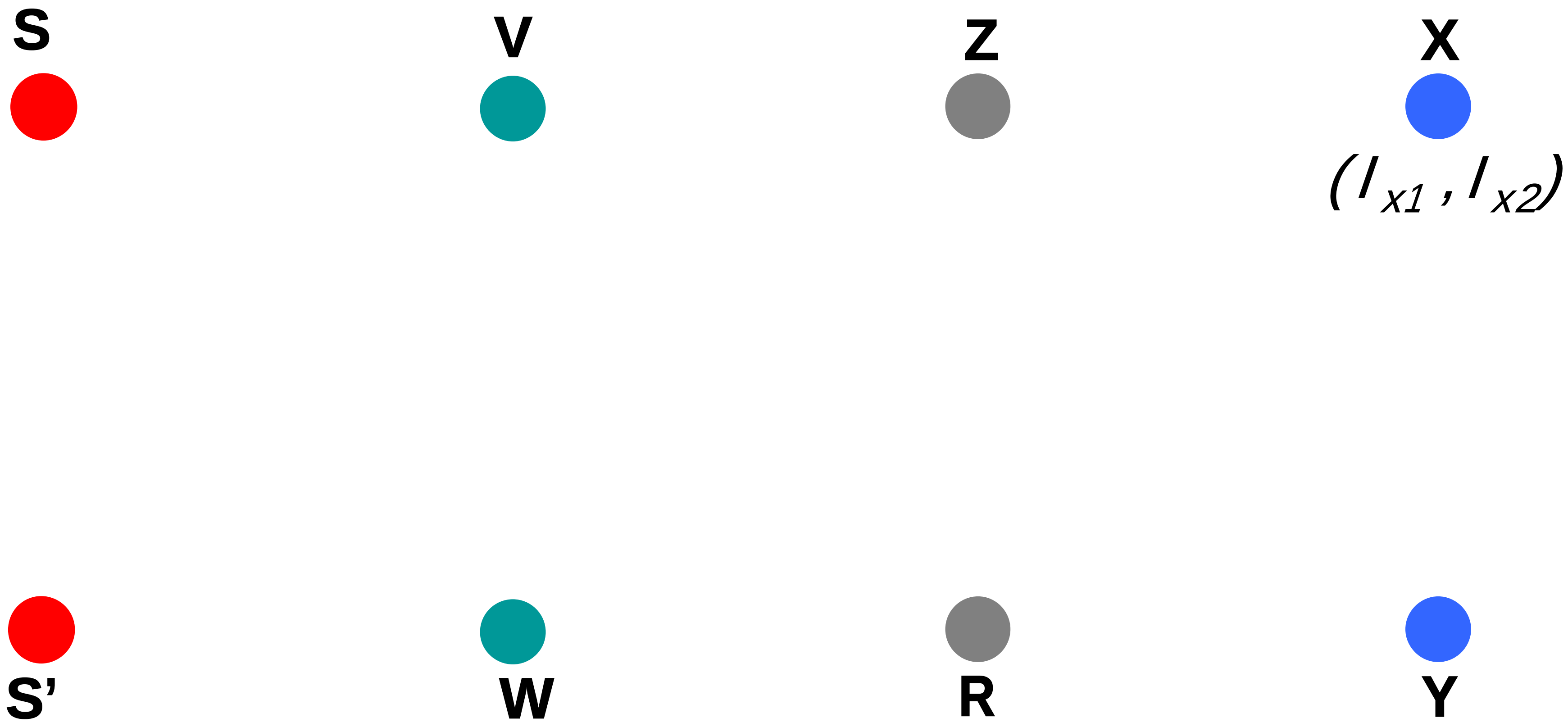
- Parameters
  - No. of stages → L
  - Splitting factor → d
- Information for each relay I
  - Next hop IP addresses
  - Receiver flag
  - Symmetric session key **(no PKI problems)**

Slicing Protocol

- Source picks $L*d$ relays including the receiver
- Relays are organized into L stages of d nodes each
- For each relay source computes I
- Source divides each I into d random slices $(I_1, \ldots\ldots, I_d)$

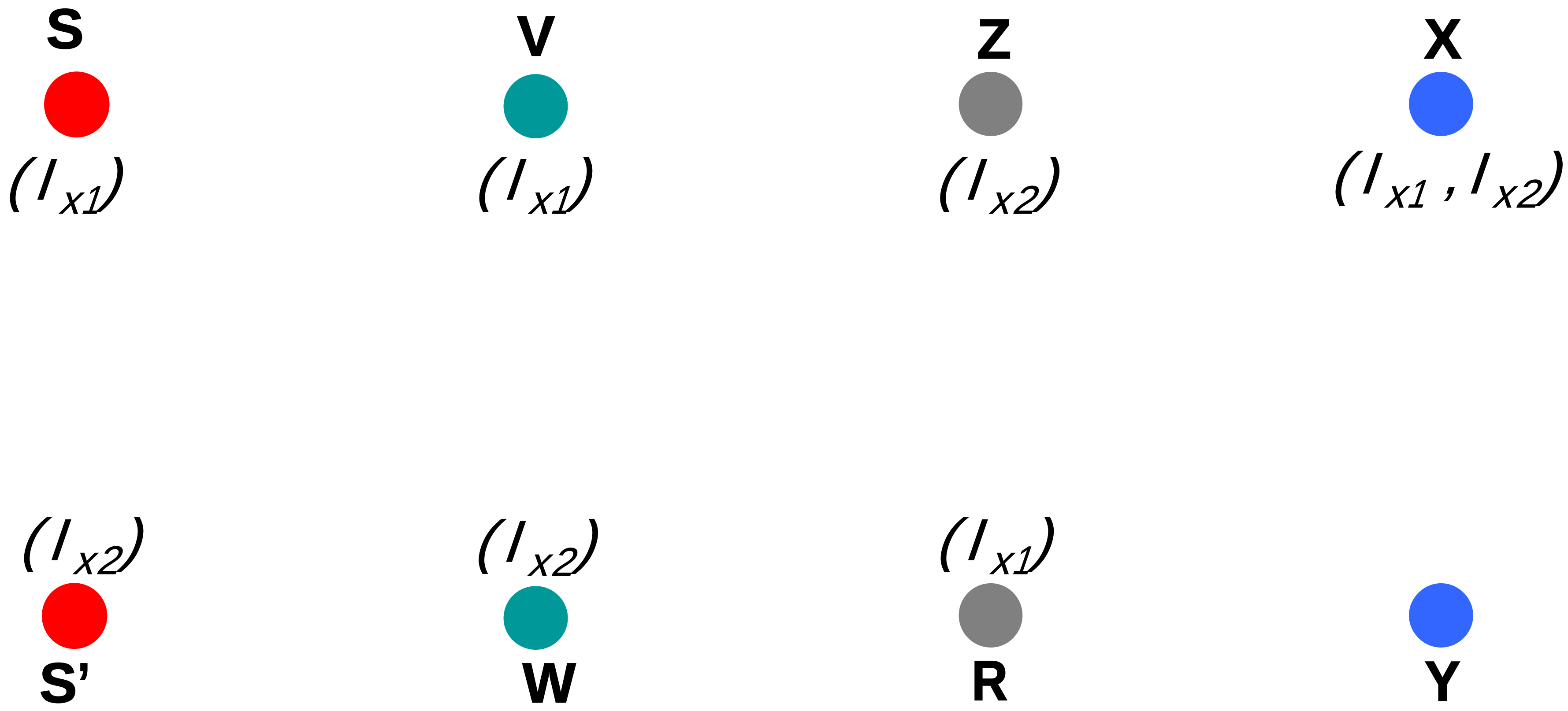
Slicing Protocol

- Relay X has to get the *d* slices $(I_{x1} ,......, I_{xd})$



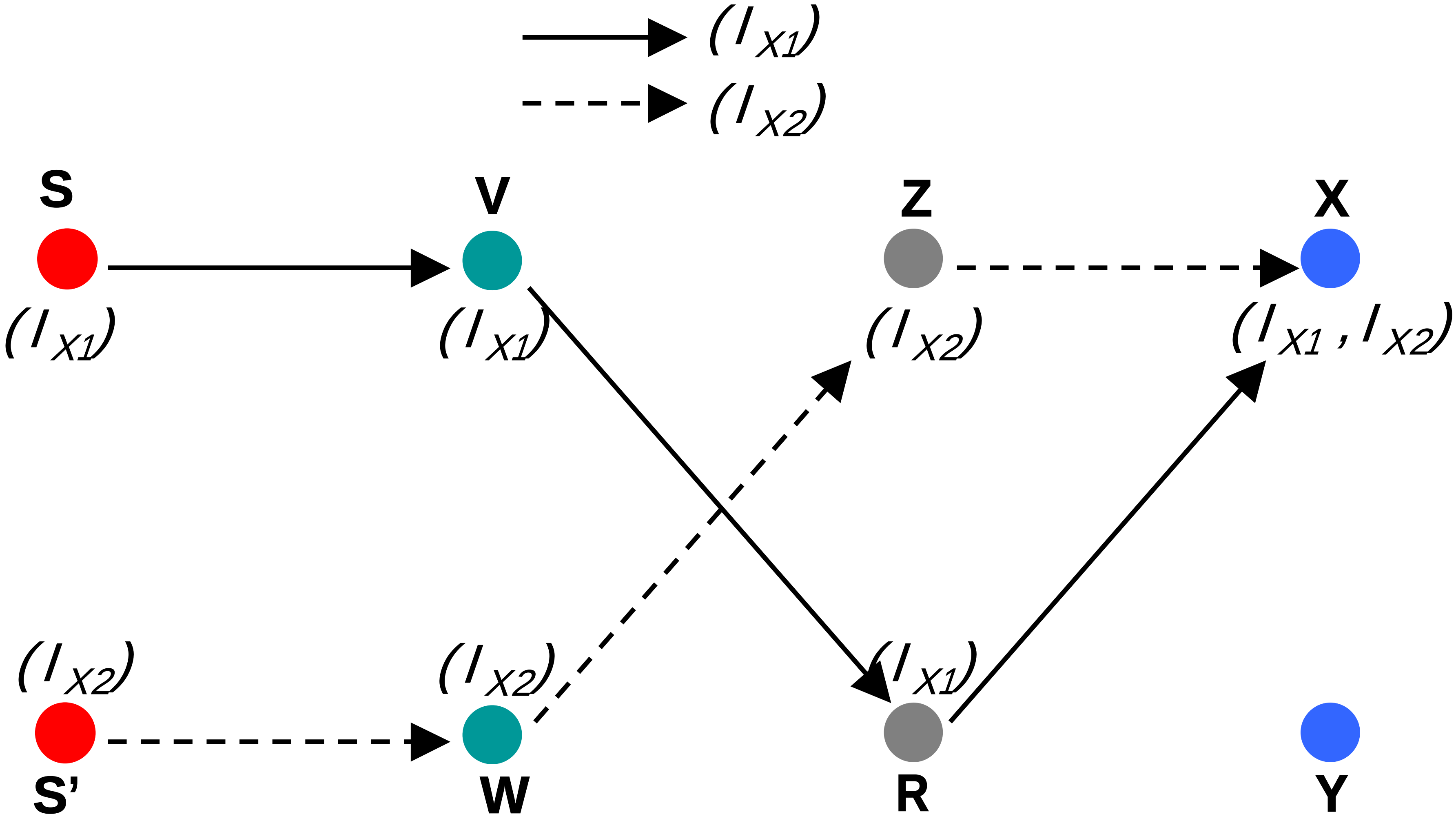
Slicing Protocol

- For each stage prior to X divide the *d* slices randomly between the *d* nodes in that stage



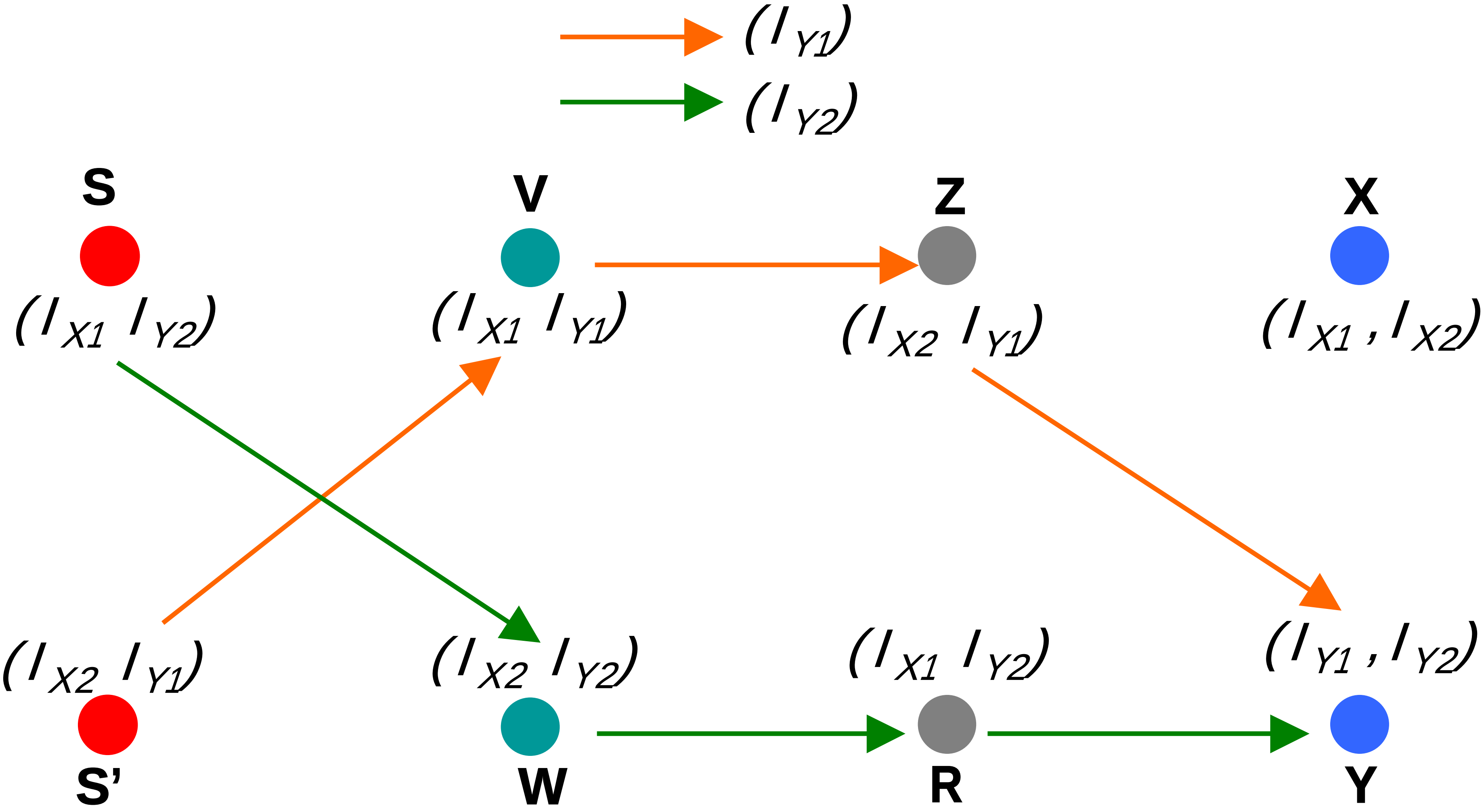
Slicing Protocol

- Slices are following node disjoint paths



Slicing Protocol

- Slices are following node disjoint paths

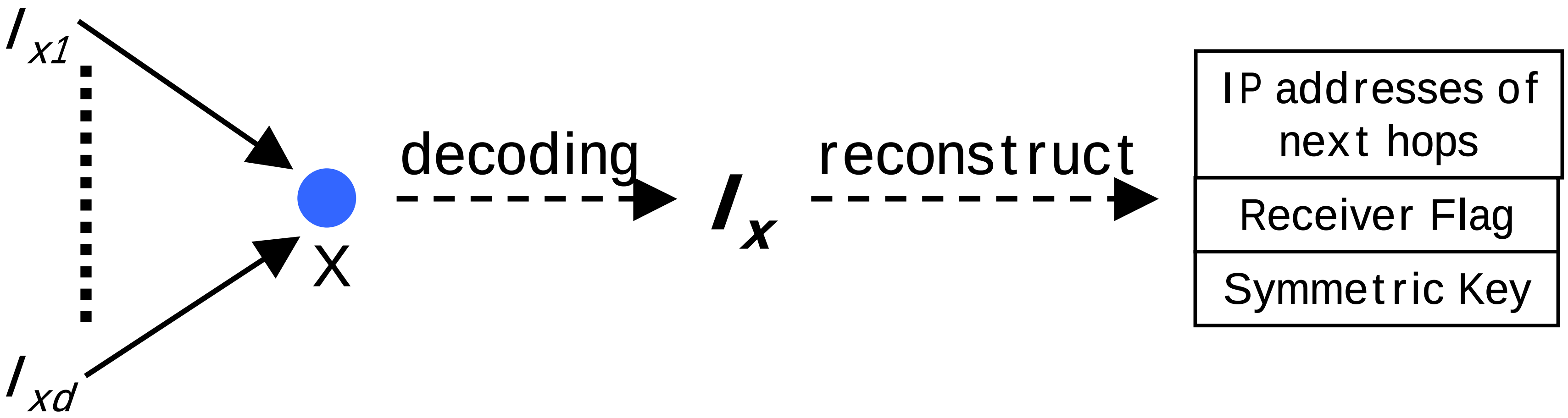


Slicing Protocol

- Source organizes $L*d$ relays into L stages of d nodes
- Source divides node information I into d random slices $(I_1 ,......, I_d)$
- Relay X gets the d random slices $(I_{x1} ,......, I_{xd})$
- If X is in stage k
  - Source goes to stages $k-1$ to 1
  - Assigns the d slices of node X randomly to the d nodes in that stage

Slicing Protocol - Decoding

- Node uses the *d* slices from its parents to decode its information



Slicing Protocol – Data Transmission

- Each node in the graph has a symmetric key assigned by the source
- Source uses normal ***onion routing*** to transmit data

Why this is exciting?

- No PKI → Truly distributed P2P anonymous overlays
- Scales to large number of nodes
- Simple matrix multiplications → Efficient anonymity

Practical anonymity

What we are doing…

- Resilience to node churn
- Anonymity similar to Chaum mixes (i.e., onion routing)
- Resilience to traffic analysis attacks
- Implementing it on Planetlab

To conclude...

Fundamentally new way to provide anonymity that does not need PKI