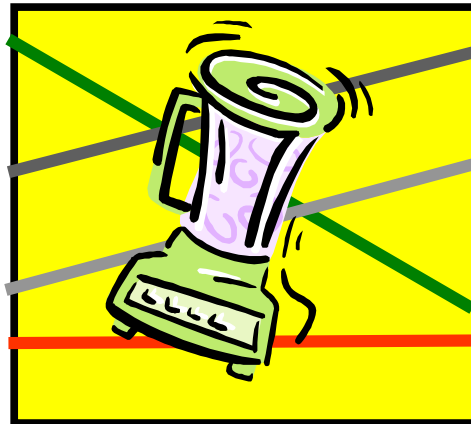


Universal Re-encryption: For Mix-Nets and Other Applications



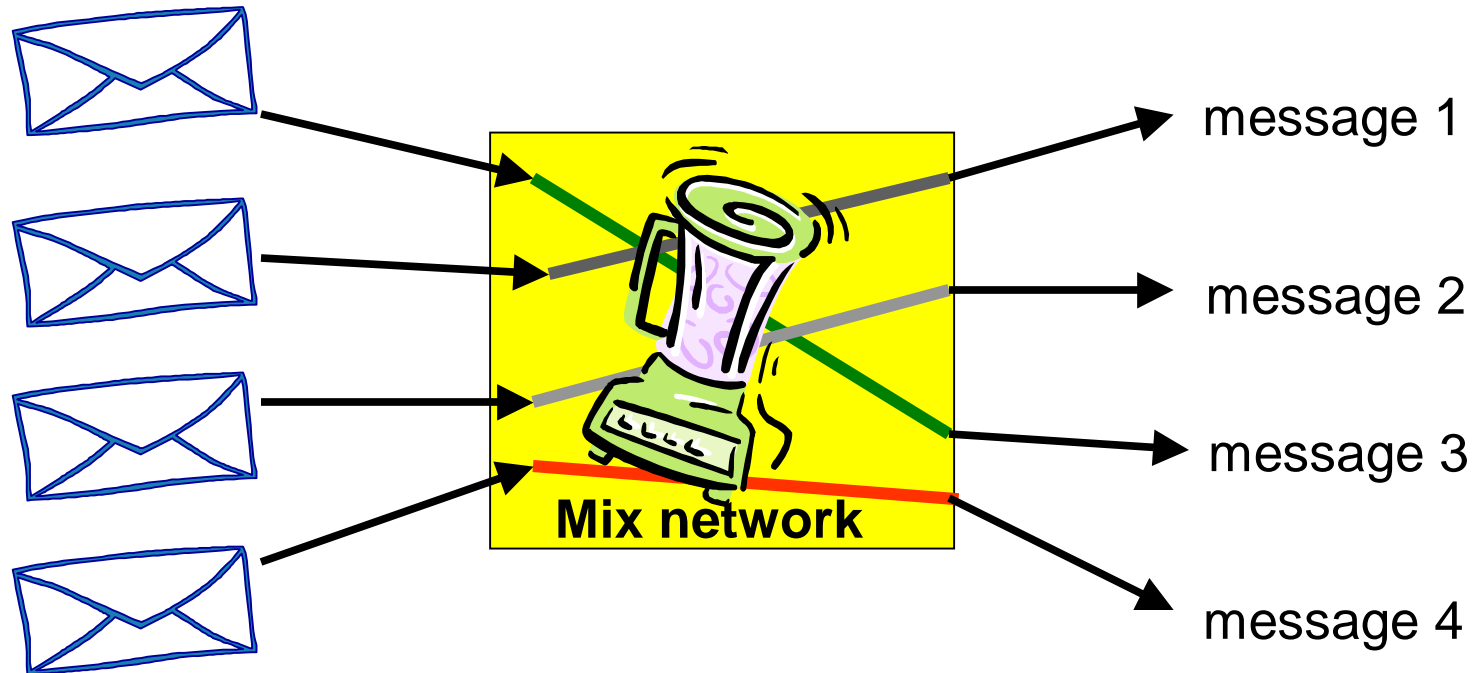
Philippe Golle
Stanford

Markus Jakobsson Ari Juels
RSA Labs

Paul Syverson
NRL

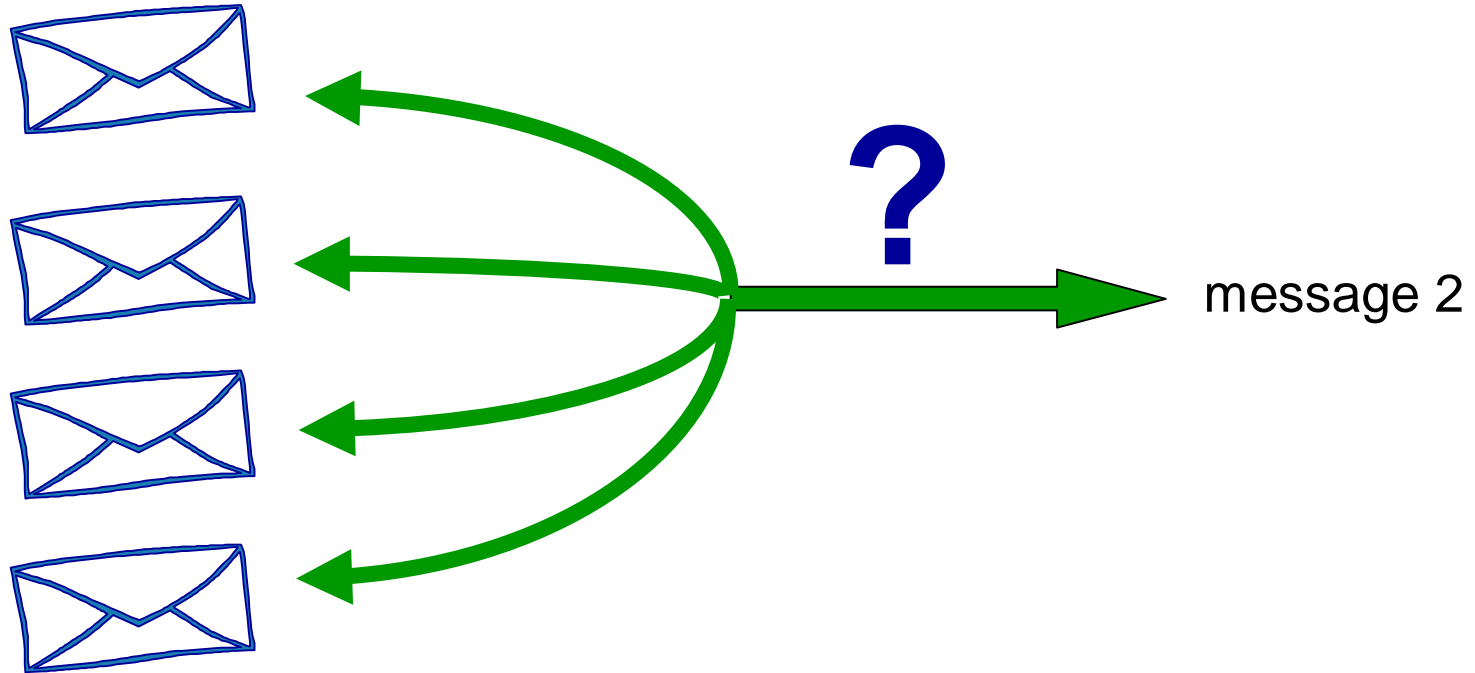


What does a mix network do?



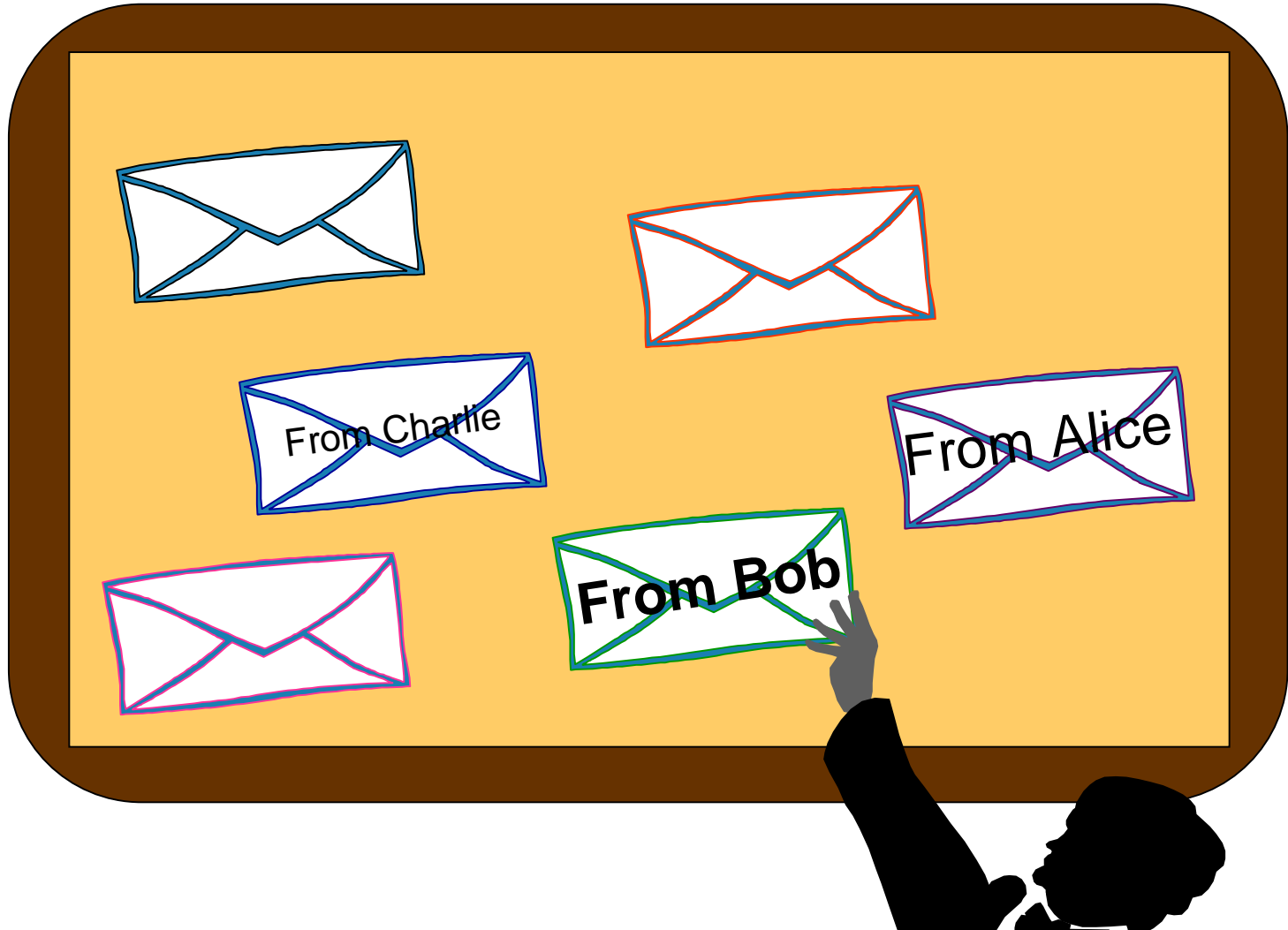
Randomly permutes and decrypts inputs

What does a mix network do?

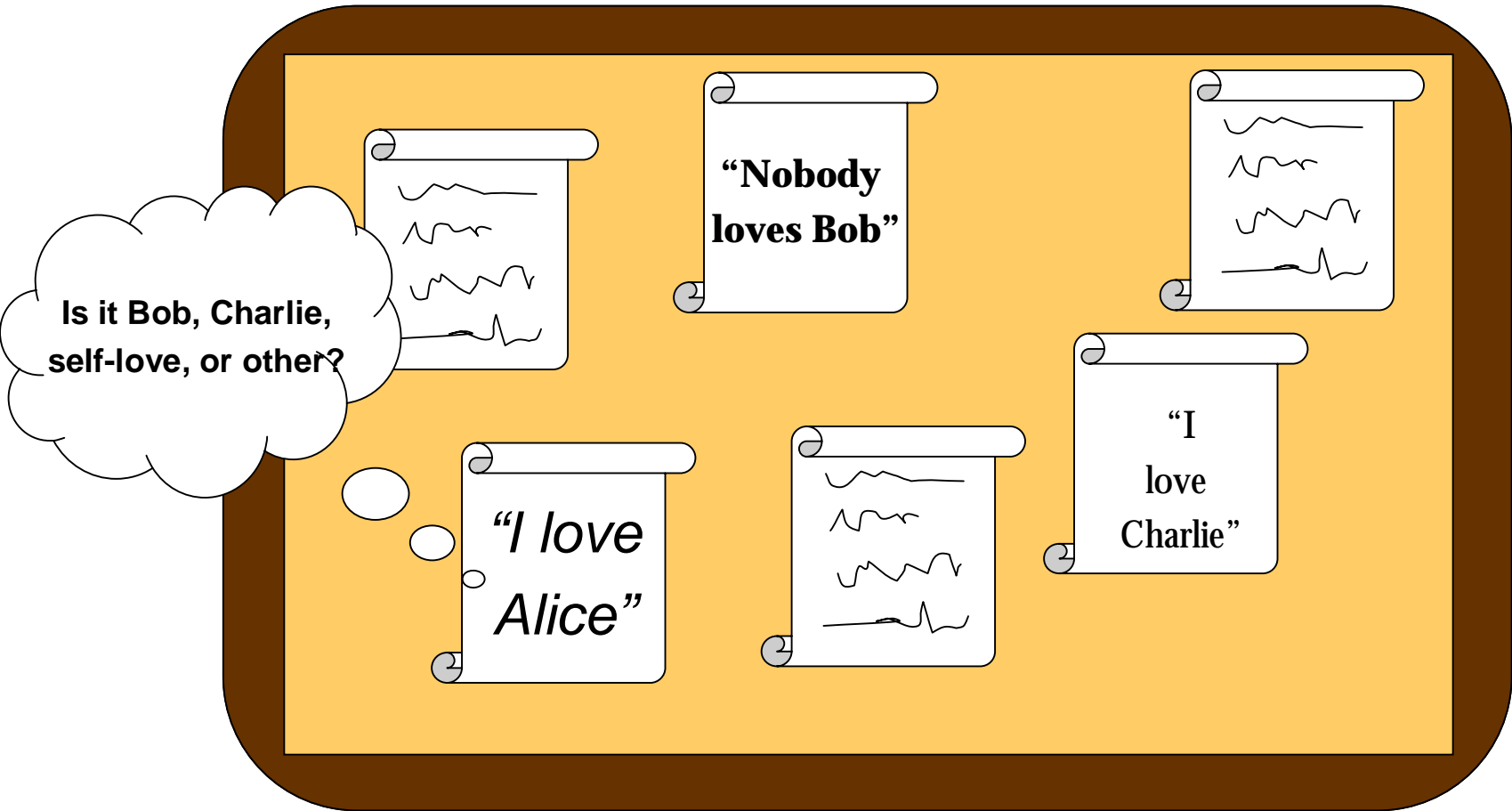


Key property: Adversary can't tell which ciphertext corresponds to a given message

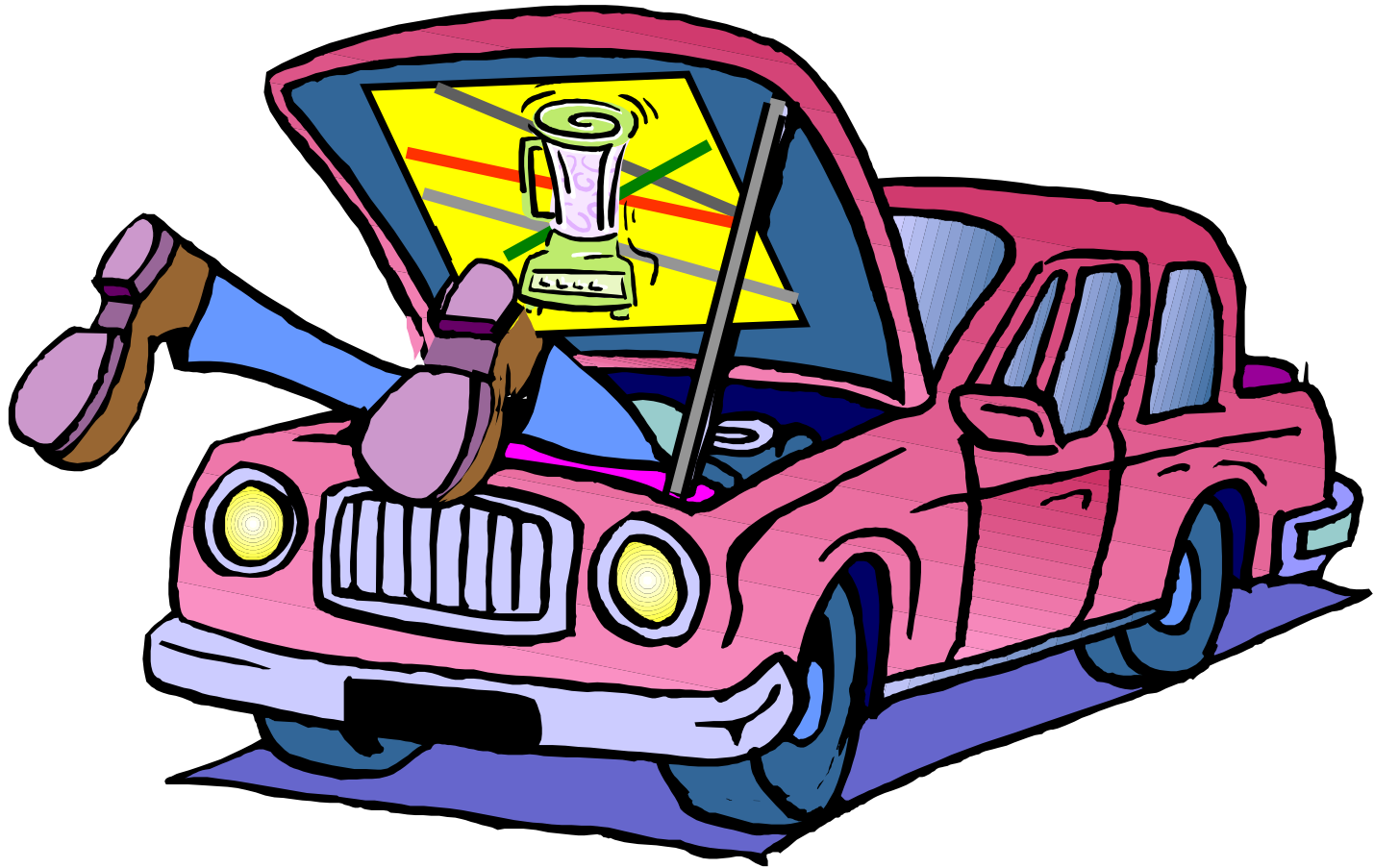
Example application: Anonymizing bulletin board or e-mail



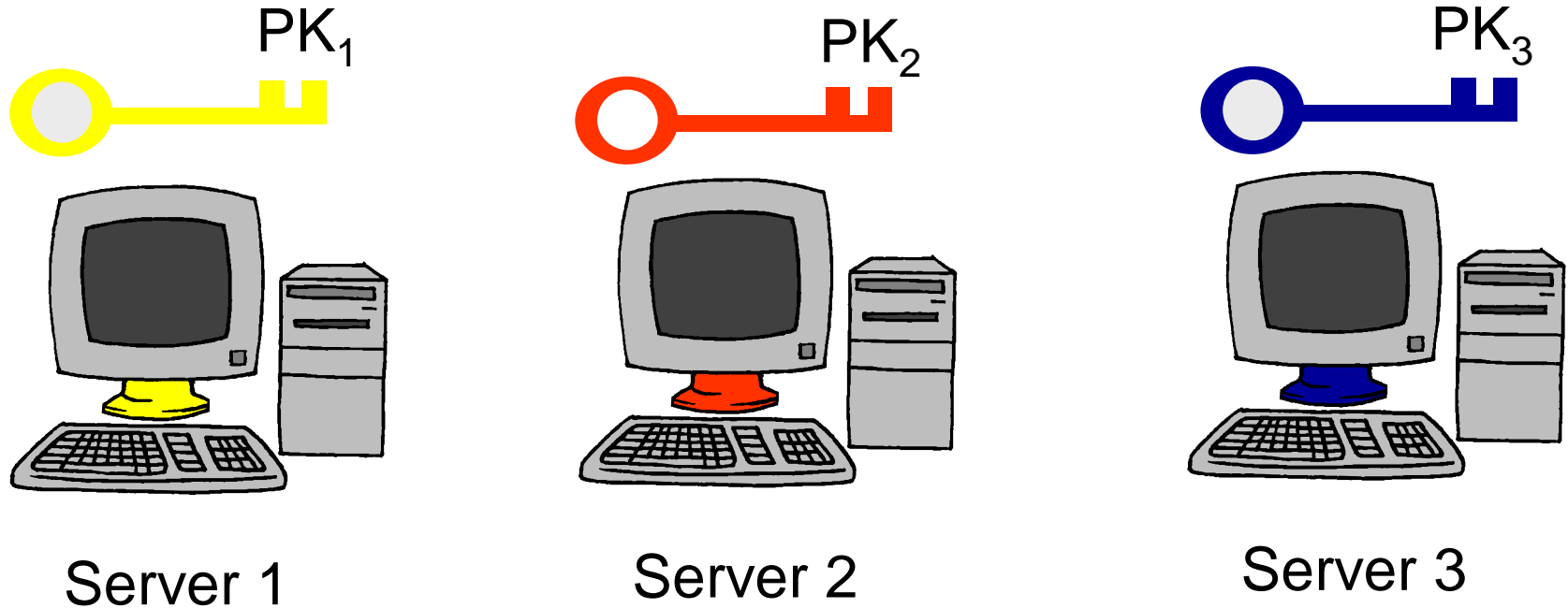
Example application: Anonymizing bulletin board or e-mail



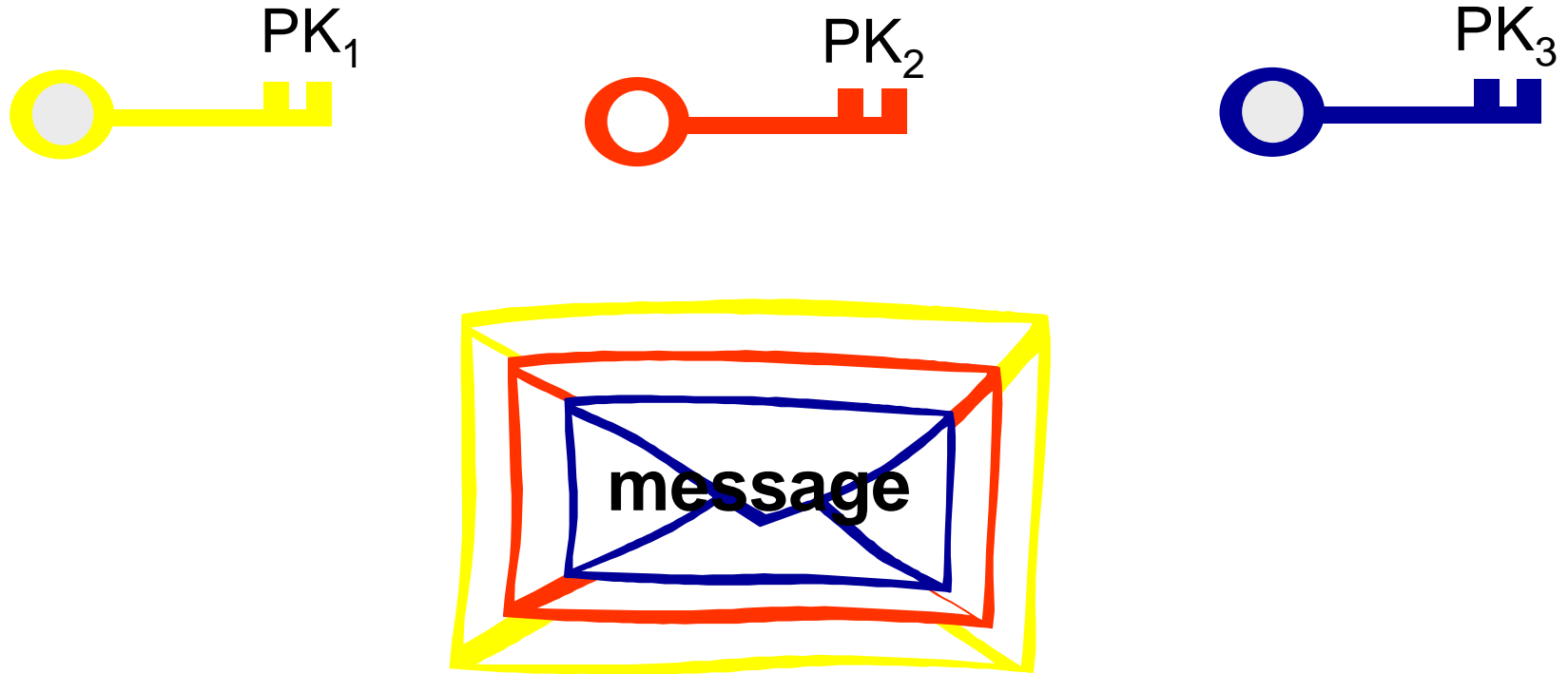
A look under the hood



Basic Mix (Chaum '81)

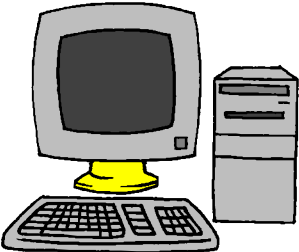


Encryption of Message



$$\text{Ciphertext} = E_{PK_1}[E_{PK_2}[E_{PK_3}[\text{message}]]]$$

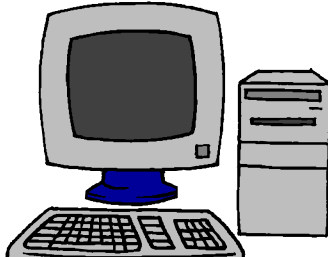
Basic Chaumian Mix



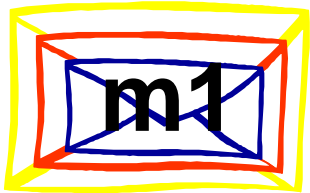
Server 1



Server 2



Server 3



decrypt
and
permute

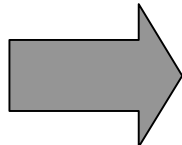
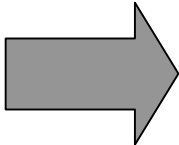
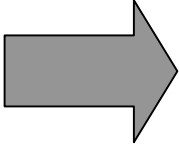
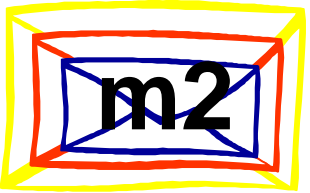


decrypt
and
permute

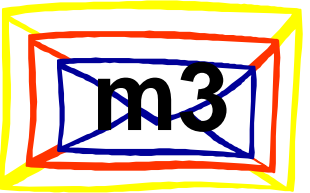


decrypt
and
permute

m2



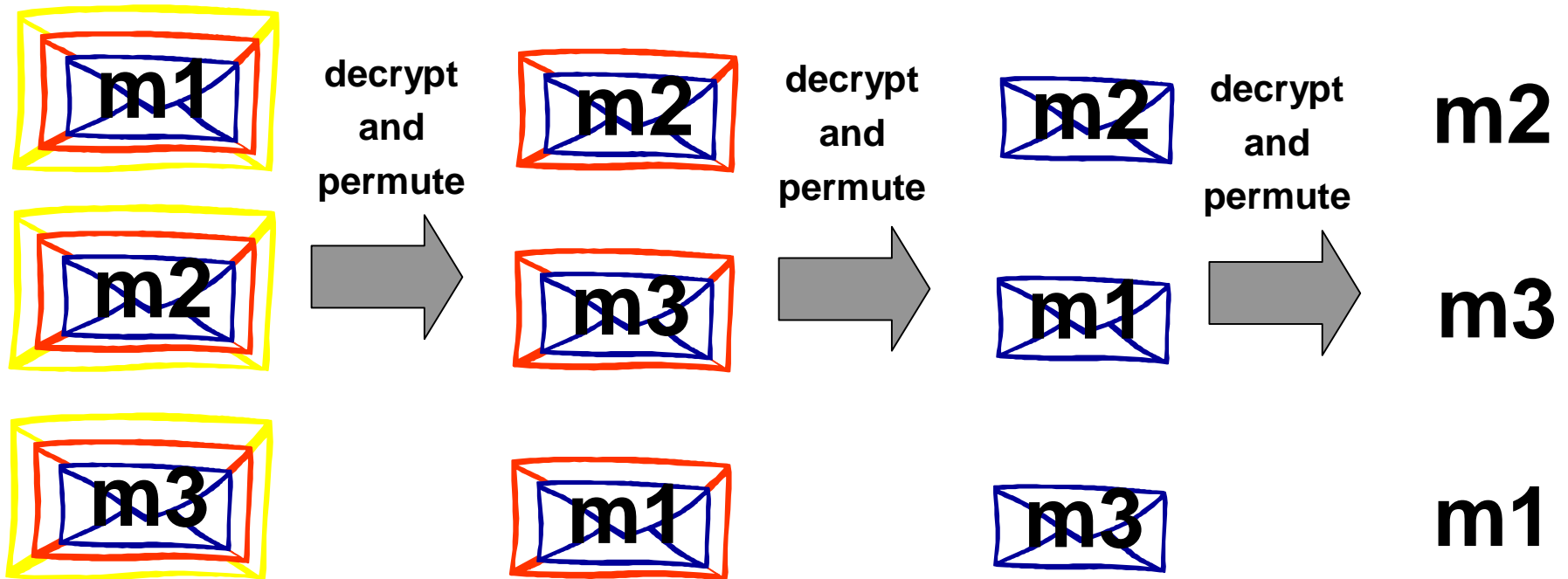
m3



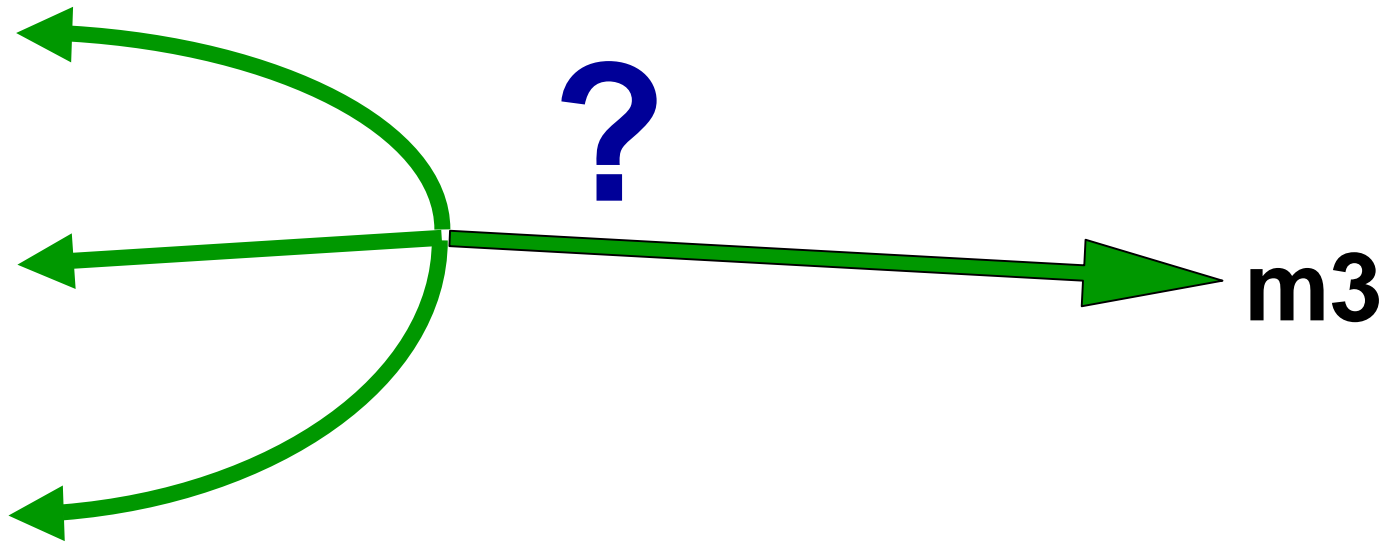
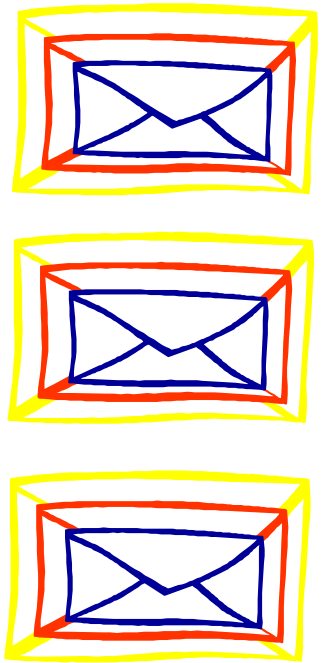
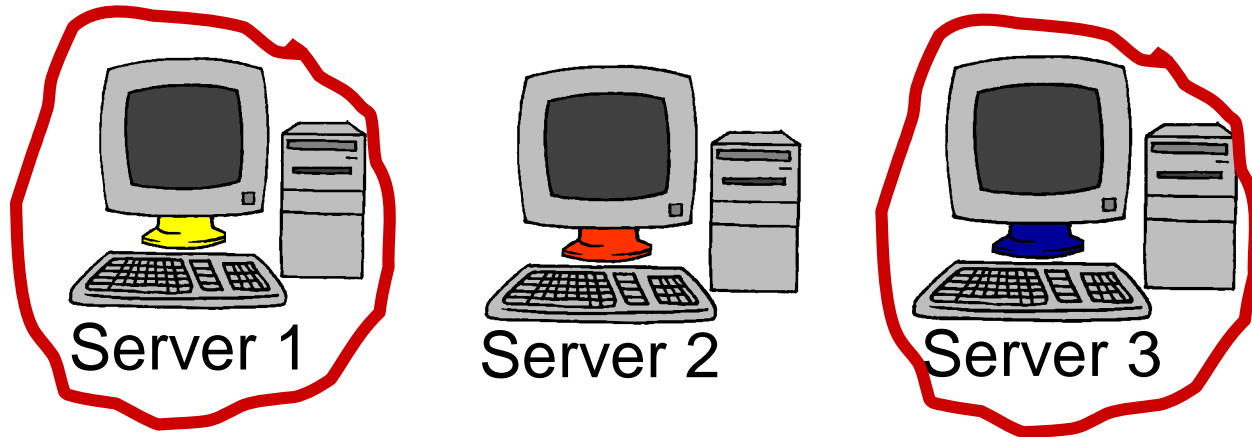
m1

Basic Chaumian Mix

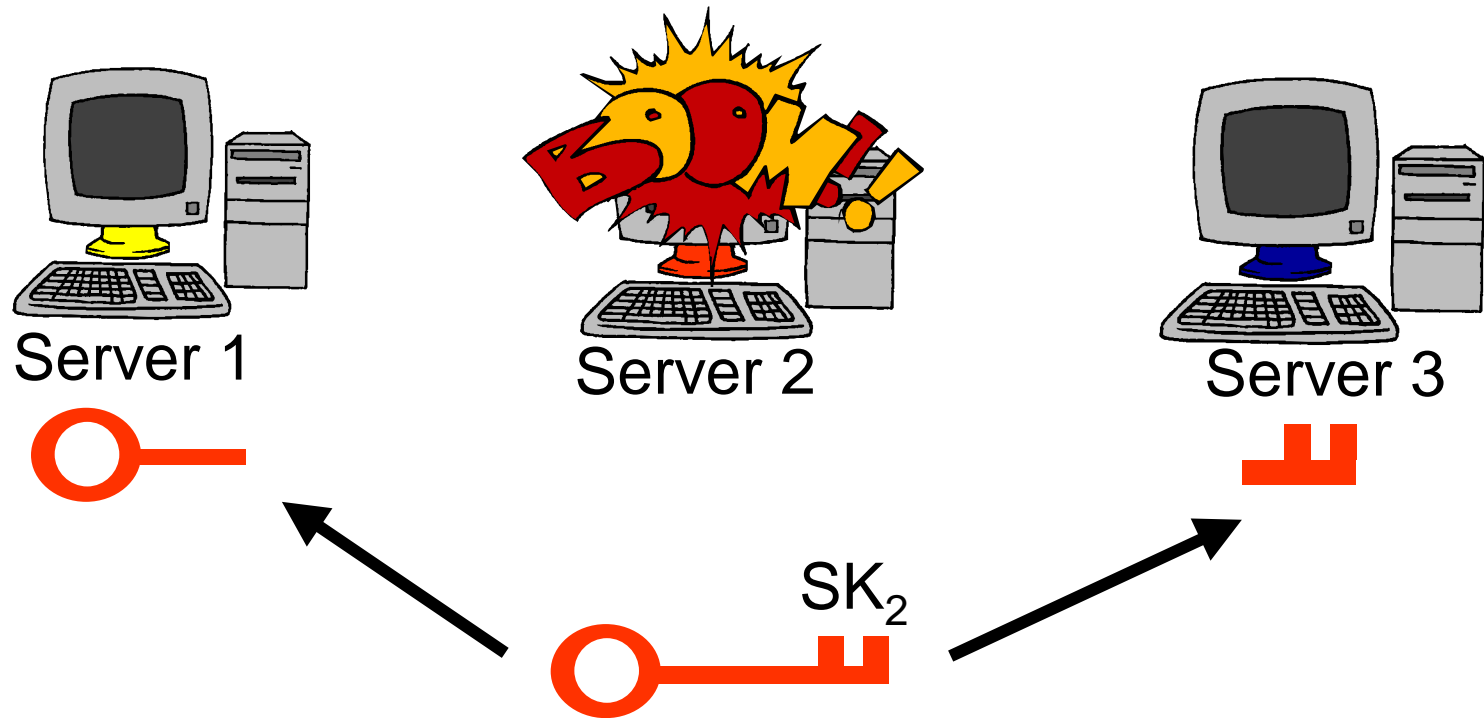
Observe: As long as one server is honest, privacy is preserved



Basic Chaumian Mix

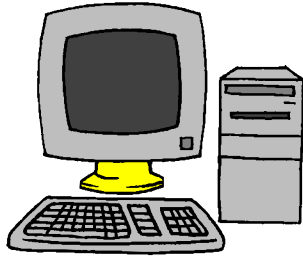


What if one server fails?



- Previous solution ideas:
 - Robustness: Share key among other mixes
 - Twinning
 - Splitting
 - Reliability: Track and use reputable mixes

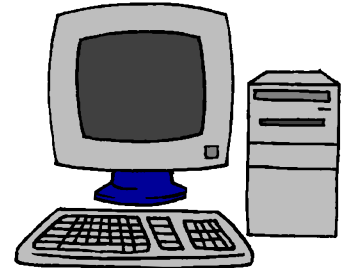
What if one server fails?



Server 1



Server 2



Server 3



- New Idea: Mixing without keys
 - No need to depend on any server (recovery mechanism)
 - No need to trust servers to protect keys
 - No need for PKI

A look further under the hood



Basic Re-encryption Mixnet



- Inputs are ciphertexts
- Outputs are a re-encryption of the inputs.
- El Gamal public key encryption:
 - Anyone can encrypt with the public key e
 - Those who know the secret key d can also decrypt
 - Malleable: can produce $E_2(m)$ from $E_1(m)$ without knowing d
 - Verifiable
 - Multiplicative homomorphism: given $E(m)$ and $E(m')$ I can produce $E(mm')$

Universal Re-encryption Mixnet



- Inputs are ciphertexts
- Outputs are a re-encryption of the inputs.
- El Gamal public key encryption:
 - Anyone can encrypt **without** the public key e
 - Those who know the secret key d can also decrypt
 - Messages encrypted with different keys are indistinguishable

ElGamal Encryption

- P, Q are prime, $P = 2Q + 1$
- G_Q subgroup of Z_P^* of order Q
- g generator of G_Q
- $x \in G_Q$ is private key
- $y = g^x \text{ mod } P$ is public key
- $E(m) = (g^r, my^r)$ where $r \in G_Q, r$ random
- $D(G, M) = M/G^x = my^r / g^{xr} = m$

ElGamal with Re-encryption

- Ciphertext (G, M)
- Re-encryption $(G', M') = (Gg^{r'}, My^{r'})$
 - Needs public key y but not private key
- $D(G', M') = M' / G'^x = my^{rr'} / g^{xrr'} = m$
- Introduced for voting
- Much work on efficient provable shuffles

Universal Re-encryption

- $(a,b) = (E[m]; E[1])$ E is ElGamal enc
- $(a',b') = (R[b,k]a; R[b,k'])$
 - $R[*,k]$ is re-encryption with random k
- $(E[m]', E[1]') = ([my^r y^{kr'}, g^r g^{kr'}], (y^{r'k'}, g^{r'k'}))$
- $D(E[m]') = M'/G'^x = my^{krr'} / g^{xkrr'} = m$

Symmetric-hybrid Encryption

- $U[k_1], U[1], e[k_1, m]$
 - $U[1]$ is universal blank
can be converted to $U[m_i]$
can be reused
 - $e[k_1, m]$ is symmetric encryption of m
- Final message
 $U[k_1], U[k_2], \dots, U[k_n], e[k_n, e[k_{n-1}, \dots, e[k_1, m] \dots]]$
- Can also do an asymmetric hybrid

Applications

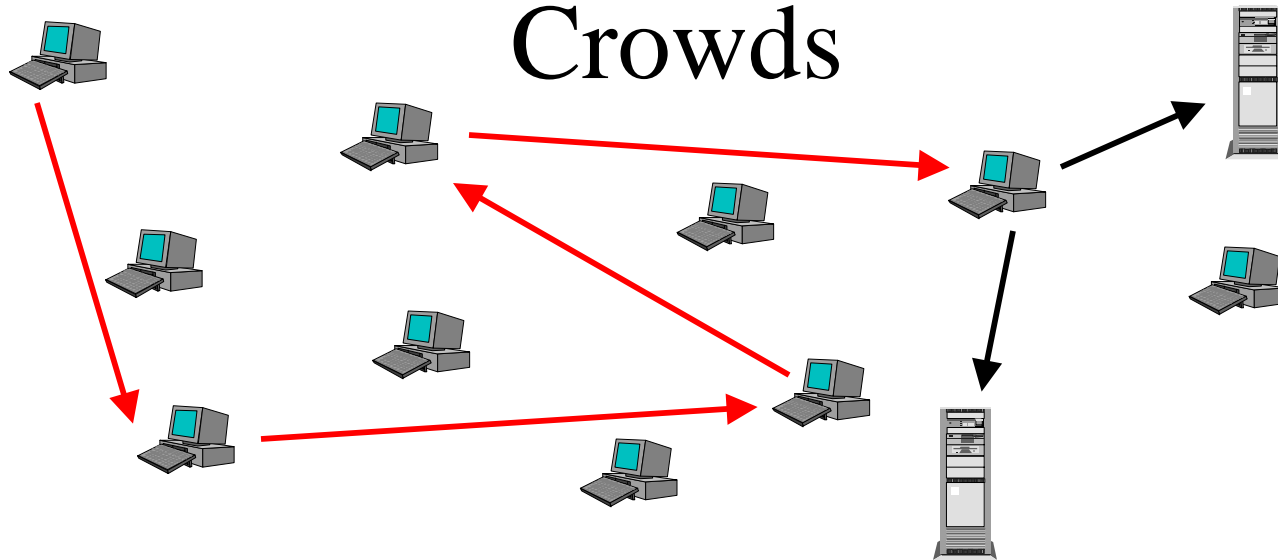
Universal Mixnets

- Any node can mix any message
- Nodes can be dynamic
- Network topology not significance
- No PKI and less trust of each node
- No robustness/reliability issues with node failure
- No overhead or threats from replay (universal semantic security)
- Can have free route re-encryption mixnets
 - With large anonymity sets

Anonymous Connections

- Use hybrids re-encryption to combine virtues of previous connection anonymity schemes
- Building a Crowd with Universal Onions

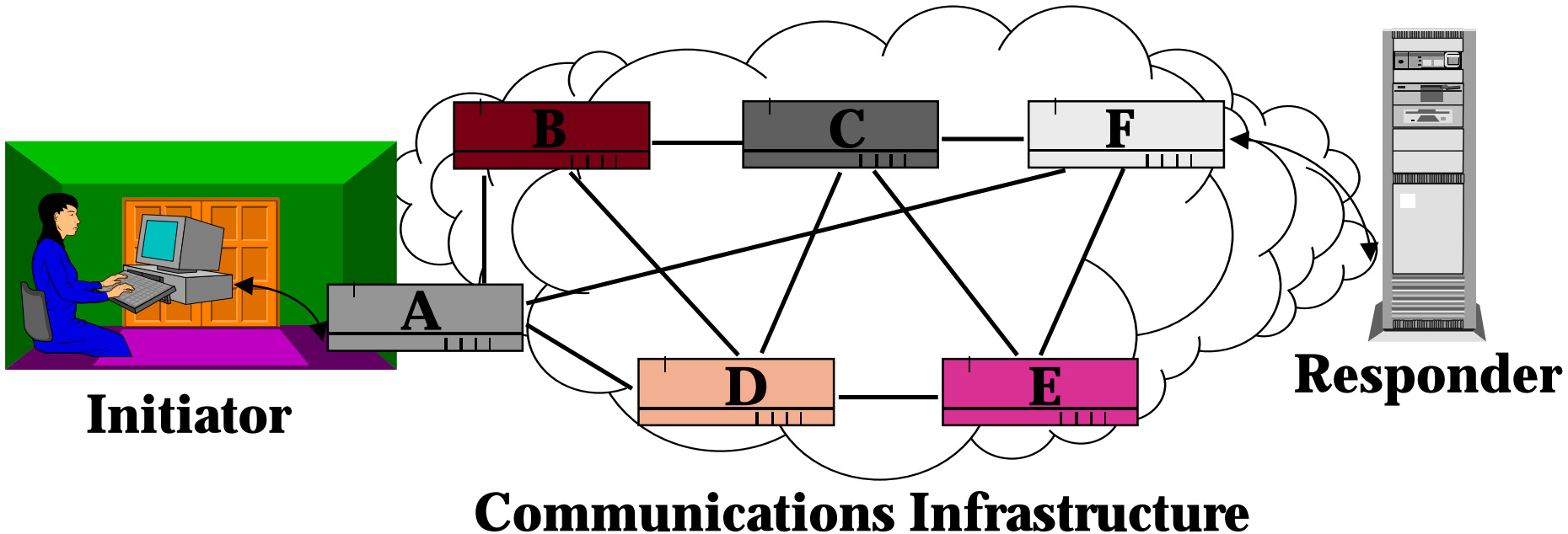
Crowds



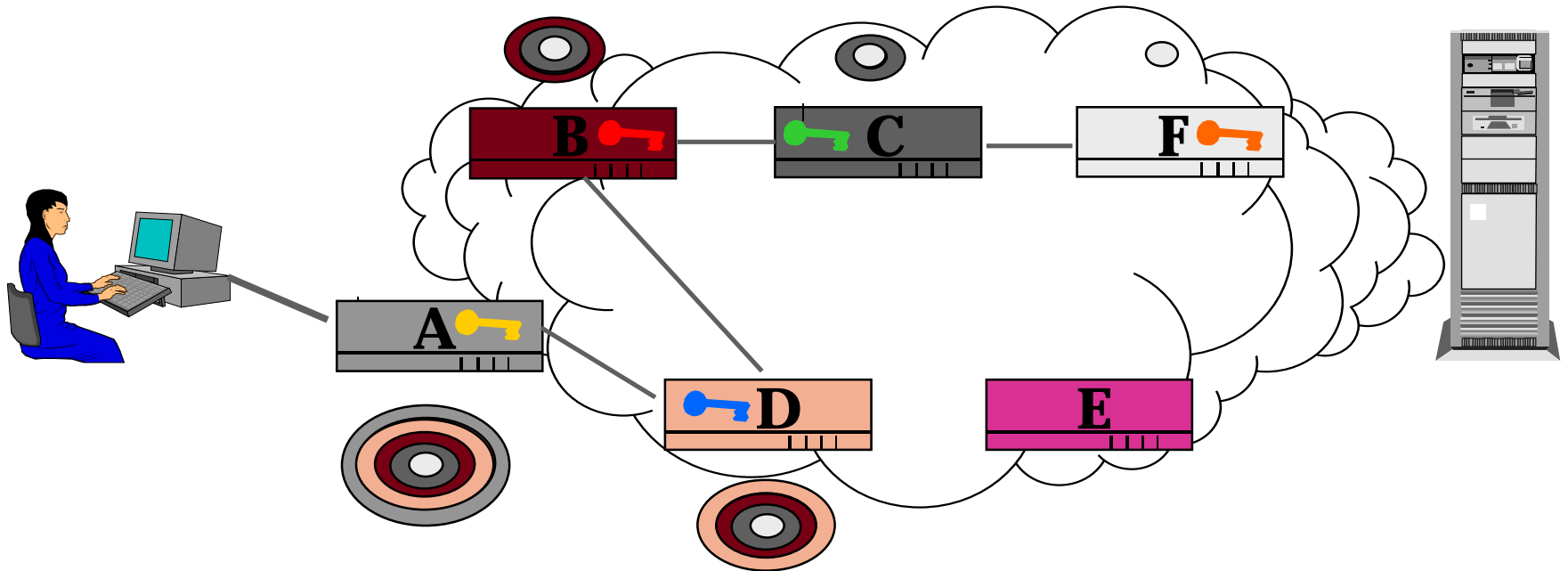
- User machines are the network (peer-to-peer scheme)
- “Blender” announces crowd composition to members
- “jondo” at machine flips weighted coin
 - If Heads forwards to random crowd member
 - If Tails connects to end Web address
- All jondos on path know path key
- All connections from a source use same path for lifetime of that crowd

Onion Routing

Proxies interface between the initiator's machine and the communications infrastructure.




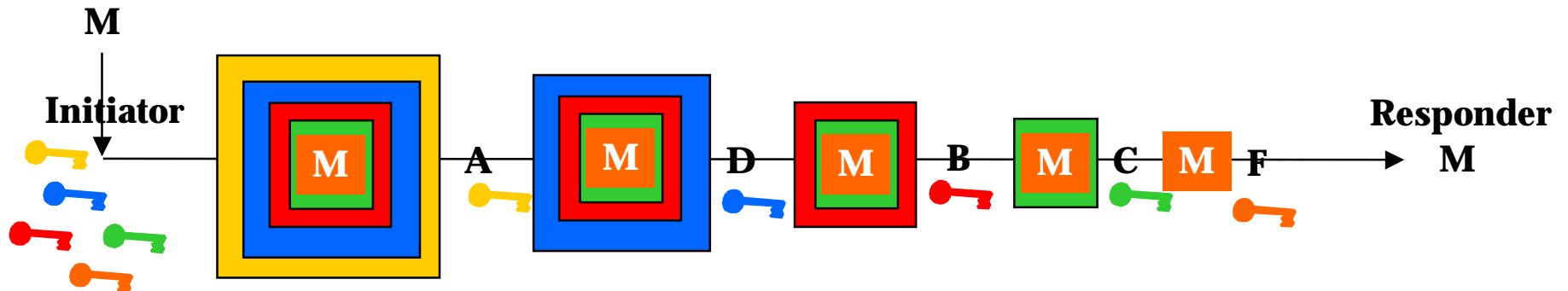
Connection Setup



- Initial proxy knows Onion Routing network topology, selects a route, and generates the onion
- Each layer of the onion identifies next hop in the route and contains the cryptographic keys to be used at that node.

Data Movement (Forward)

- Uses the keys distributed in the onion 
- Initiator's proxy pre-encrypts data cells for onion routers on route
- As data moves through an anonymous connection, it looks different at each onion router.
- Data cell M sent from an initiator to a responder via onion route:



[Actual size of data packet (cell) remains constant over the path]

Tradeoffs

- Crowds: Peer to peer
- Crowds: Low crypto overhead
- Crowds: No need for PKI
- Onions: Path key not shared
 - Connection anonymity not dependent on data anonymity
 - No pseudonymous profiling

Universal Mix Connections

- P2P network of universal mixes
- “Onion” is a hybrid with a universal blank
- Path formed as in crowds
- Don’t need to know who’s in the crowd
- Each node
 - creates and stores symmetric key
 - Signs it and universally encrypts it with blank
 - Passes it back to originator on connection
- Rest works the same as onion routing

RFID Tags

- EZ Pass automated toll payment
- Supermarket shipment tracking, stock monitoring, theft prevention
- Consumer stock monitoring, ordering
- Consumer theft-protection of belongings

RFID Tag Privacy

- “Kill” tags on purchased goods
 - Tags can no longer be used
- Put tags to sleep to later awaken
 - Doesn't allow continuous controlled access
 - Complexity (lots of keys) vs. trust for awakening

Universal Re-encryption for RFID Tag Privacy

- Alice at supermarket checkout.
- Uses PK_{Alice} from fidelity card.
- Cashier creates universal ciphertexts on Alice's purchase IDs.
- As Alice walks home passes readers that re-encrypt her tags or does it herself.
- Alice enters home, tags decrypted for home use.

Conclusions

- Universal Re-encryption: New primitive
- Applications
 - Reduced trust in mixes
 - Less complex mixnets (no PKI)
 - Better anonymous connections
 - Privacy preserving RFID tags
- Open
 - Properties: Universal Semantic Security, Existential Construction Resistance
 - More Applications