CS 458 / 658: Computer Security and Privacy

Module 5 - Security and Privacy of Internet Applications

Part 3 - Privacy-enhancing technologies (PETs)

Meng Xu *(University of Waterloo)*

Winter 2023

Outline

What is Privacy?

Q: What does privacy mean to you?

What is Privacy?

Q: What does privacy mean to you?

- Paper on PoPETs 2018: Turtles, Locks, and Bathrooms: Understanding Mental Models of Privacy Through Illustration
- Asked people of different ages in the US to draw a diagram on what privacy means to them, and here are a few illustrations.

Privacy as turtles

Fig. 63. "It's a turtle huddled up inside its shell." By John

Fig. 62. "Pearl oysters have something valuable to protect - the pearl. They can do so by simply 'closing the lid.' If only safeguarding the data in my laptop were that simple!" By Sharon, age 25.

Fig. 40. "A shield that protects me." By HAP, age 24

[0] All pictures from the PETS'18 paper

Privacy as locks

Fig. 10. "To me, privacy is fundamentally about feeling secure. Having the ability to control who has access to me, and to my information, makes me feel like I can control my privacy." By CJ, age 33

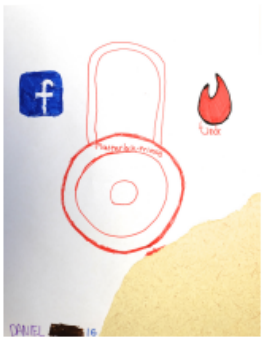


Fig. 11. By Daniel, age 16

Fig. 33. "Privacy means that the thoughts in my brain are locked away. What I know does not have to go into the world, which I put an X over." By Thomas, age 19

[0] All pictures from the PETS'18 paper

Privacy as bathrooms

Fig. 23. "This is me enjoying my privacy. This is the only time during the day, were I am truly alone and nothing bothers me. No man no children no dogs." By Cindy, age 54

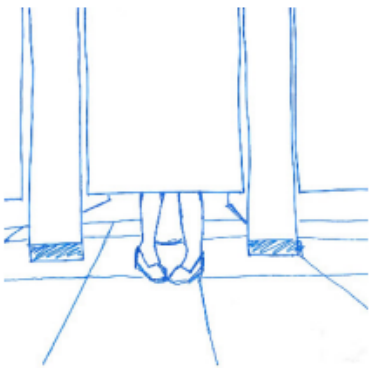


Fig. 38. By Rachel, age 20

Fig. 24. "No one come in when I am in the bathroom!" By Sydney, age 7

[0] All pictures from the PETS'18 paper

Privacy as filters

Fig. 45. "Green data (non-private) goes through; red does not (private data). Some yellow goes through (ambiguous)." By Ryan, age 36

Fig. 58. "Privacy is to me the ability to filter and control the information relevant to you that you release into the world (and having some confidence in the ability of the status of such information as private)." By Isadora, age 20

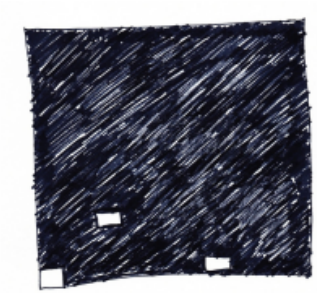


Fig. 74. "Privacy means my life is a black box, except for the items I choose to share with others." By Lauren, age 32

[0] All pictures from the PETS'18 paper

Privacy as controls

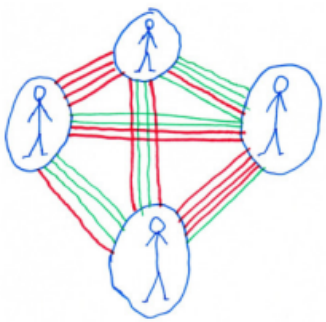


Fig. 46. 'Privacy is a network: I share what I want with whom I want and trust and what matches with those in the network, and don't share with those I don't want and trust to share with. Green = share. Red = don't." age 20s

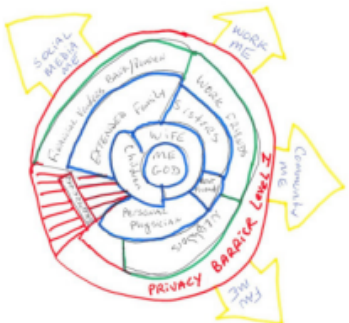


Fig. 47. "I give/receive based on my level of trust. Occasionally, I do not share with those I trust (i.e., my exception jail) as I do not trust what they will do with a specific piece of information. I accept that I must have a public persona." By Jim, age 51

Fig. 55. "There are bright sides, and there are dark sides. Some of them we'd love to share; some we don't, and they are called 'privacy."' By Evan, age 21

[0] All pictures from the PETS'18 paper

Privacy as tools

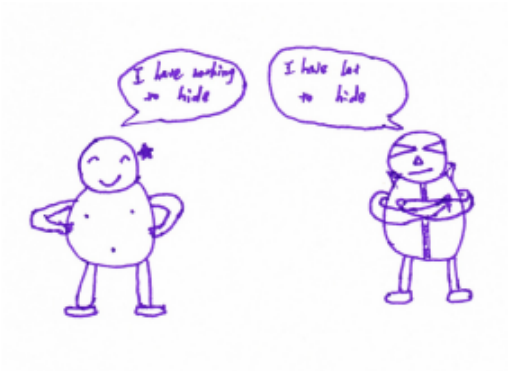


Fig. 54. By Lidong Wei

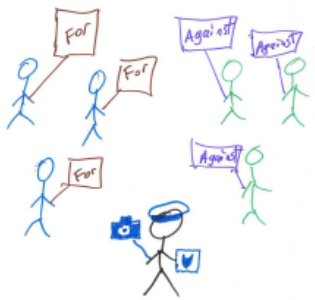


Fig. 30. "People should be able to express their views without surveillance & infiltration by the police." By anonymous, age "old"

Fig. 28. "The picture is of a diary that has a mechanism to keep it shut. There is no key available within the drawing, so that no one can open it. If no one can open it, privacy remains intact." By Karen, age 43

[0] All pictures from the PETS'18 paper

So, what is privacy?

So, what is privacy?

Privacy means something different to each person ("informational self-determination")

So, what is privacy?

Privacy means something different to each person ("informational self-determination")

There are two "types" of information that could be privacy-sensitive:

- Data: refers to contents of messages, contents of a database, etc.

So, what is privacy?

Privacy means something different to each person ("informational self-determination")

There are two "types" of information that could be privacy-sensitive:

- Data: refers to contents of messages, contents of a database, etc.
- Meta-data: any other information that is not data; for example, in communications, meta-data includes:
  - Who communicates with whom?
  - What time does Alice communicate with Bob?
  - How often does Alice communicate with Bob?
  - Where does Alice communicate from?
  - ...

So, what is privacy?

Privacy means something different to each person ("informational self-determination")

There are two "types" of information that could be privacy-sensitive:

- Data: refers to contents of messages, contents of a database, etc.
- Meta-data: any other information that is not data; for example, in communications, meta-data includes:
  - Who communicates with whom?
  - What time does Alice communicate with Bob?
  - How often does Alice communicate with Bob?
  - Where does Alice communicate from?
  - ...

* We can hide data using cryptography, but sometimes we need to leak some data to a potential adversary to get some utility from a service (many services have a privacy-utility trade-off).

So, what is privacy?

Privacy means something different to each person ("informational self-determination")

There are two "types" of information that could be privacy-sensitive:

- Data: refers to contents of messages, contents of a database, etc.
- Meta-data: any other information that is not data; for example, in communications, meta-data includes:
  - Who communicates with whom?
  - What time does Alice communicate with Bob?
  - How often does Alice communicate with Bob?
  - Where does Alice communicate from?
  - ...

* We can hide data using cryptography, but sometimes we need to leak some data to a potential adversary to get some utility from a service (many services have a privacy-utility trade-off).
* Protecting meta-data requires more than cryptography.

What we will cover

We need Privacy-Enhancing Technologies (PETs) to control data leakage, as well as to protect meta-data!

What we will cover

We need Privacy-Enhancing Technologies (PETs) to control data leakage, as well as to protect meta-data!

We'll only cover PETs that are related to two aspects of privacy:

- Anonymity in communications: how to hide who communicates with whom; we'll see remailers (mixes) and Tor.
- Data minimization: how to achieve a functionality while minimizing the amount of data collected; we'll see Private Information Retrieval (PIR)

Kerckhoff's principle, again

Remember this from the first lecture of this module?

Kerckhoff's principle

a cryptosystem should be secure, even if everything about the system, except the key, is public knowledge.

Shannon's maxim

one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them.

Kerckhoff's principle, again

Remember this from the first lecture of this module?

Kerckhoff's principle

a cryptosystem should be secure, even if everything about the system, except the key, is public knowledge.

Shannon's maxim

one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them.

We also need to keep these into account when designing PETs! We assume the adversary knows how the technology works, and we still want to get privacy with this assumption.

Outline

Anonymity

Q: What is the goal of being anonymous?

Anonymity

Q: What is the goal of being anonymous?

A: Hiding identities. More formally, "the state of not being identifiable within a set of subjects — the anonymity set" [a]

Anonymity

Q: What is the goal of being anonymous?

A: Hiding identities. More formally, "the state of not being identifiable within a set of subjects — the anonymity set" [a]

- A sender may be anonymous within a set of potential senders: the sender anonymity set.
- Same for recipients.

[a] from *Pfitzman et al*, "Anonymity, unobservability, and pseudonymity — a proposal for terminology"

Anonymity categorization

We can place transactions (both online and offline) on a continuum according to the level of *nymity* they represent, that is, how they refine the anonymity set:

- Verinymity: (Almost) unique information.
  - government ID, SIN, credit card number, address
- Persistent pseudonymity: a pseudonym or a "handle" that is used persistently by the same person
  - pseudonyms for blog posts, Twitter / Instagram usernames, etc
- Linkable anonymity:
  - prepaid phone cards, loyalty cards
- Unlinkable anonymity
  - cash payments, remailers, Tor (browser)

Anonymous communication systems

Anonymous communication systems are typically classified into:

- High-latency anonymous communication systems:

- Low-latency anonymous communication systems:

Anonymous communication systems

Anonymous communication systems are typically classified into:

- High-latency anonymous communication systems:
  - Provide protection against global passive adversaries
  - Have higher delays (fine for email)
  - We will see mixes (remailers).
- Low-latency anonymous communication systems:

Anonymous communication systems

Anonymous communication systems are typically classified into:

- High-latency anonymous communication systems:
  - Provide protection against global passive adversaries
  - Have higher delays (fine for email)
  - We will see mixes (remailers).
- Low-latency anonymous communication systems:
  - Do not protect against global passive adversaries, but are good against local adversaries.
  - Have lower delays (fine for browsing)
  - We will see Tor

Anonymous communication systems

Anonymous communication systems are typically classified into:

- High-latency anonymous communication systems:
  - Provide protection against global passive adversaries
  - Have higher delays (fine for email)
  - We will see mixes (remailers).
- Low-latency anonymous communication systems:
  - Do not protect against global passive adversaries, but are good against local adversaries.
  - Have lower delays (fine for browsing)
  - We will see Tor

The latency of high-latency anonymous communication systems has decreased significantly, they are not really "high-latency" anymore.

Anonymous system design decisions

- If you build a system at a certain level of anonymity, it is *easy* to modify it to have a *higher* level of anonymity, but *hard* to modify it to have a *lower* level.

- The lesson: design systems with a low level of anonymity fundamentally; adding more is easy.

Outline

Anonymity for email: remailers

How to send and receive emails without revealing your own email address?

Anonymity for email: remailers

How to send and receive emails without revealing your own email address?

- Anonymous remailers
  - If "From" is hidden, then who do you reply to?

Type 0 remailers

In the 1990s, there were very simple ("type 0") remailing services, the best known being `anon.penet.fi` (1993–1996)

Type 0 remailers

In the 1990s, there were very simple ("type 0") remailing services, the best known being `anon.penet.fi` (1993–1996)

Here is how it worked:

- Send email to `anon.penet.fi`
- It is forwarded to your intended recipient
- Your "From" address is changed to `anon43567@anon.penet.fi` (but your original address is stored in a table)

Type 0 remailers

In the 1990s, there were very simple ("type 0") remailing services, the best known being `anon.penet.fi` (1993–1996)

Here is how it worked:

- Send email to `anon.penet.fi`
- It is forwarded to your intended recipient
- Your "From" address is changed to `anon43567@anon.penet.fi` (but your original address is stored in a table)
- Replies to the anon address get mapped back to your real address and delivered to you

Type 0 remailers

In the 1990s, there were very simple ("type 0") remailing services, the best known being `anon.penet.fi` (1993–1996)

Here is how it worked:

- Send email to `anon.penet.fi`
- It is forwarded to your intended recipient
- Your "From" address is changed to `anon43567@anon.penet.fi` (but your original address is stored in a table)
- Replies to the anon address get mapped back to your real address and delivered to you

$\approx$ 10 000 emails per day ($\approx$ 700 000 users)

anon.penet.fi

Q: What is the assumption here?

anon.penet.fi

Q: What is the assumption here?

A: This works, as long as:

- No one's watching the Internet connections to or from `anon.penet.fi`
- The operator of `anon.penet.fi`, the machine (hardware), and the software all remain trustworthy and uncompromised
- The mapping of anon addresses to real addresses is kept secret

anon.penet.fi

Q: What is the assumption here?

A: This works, as long as:

- No one's watching the Internet connections to or from `anon.penet.fi`
- The operator of `anon.penet.fi`, the machine (hardware), and the software all remain trustworthy and uncompromised
- The mapping of anon addresses to real addresses is kept secret

Unfortunately, a lawsuit forced Julf (the operator) to turn over parts of the list, and he shut down the whole thing, since he could no longer legally protect it

Type I remailers

Cypherpunk (type I) remailers removed the central point of trust

- Messages are now sent through a "chain" of several remailers, with dozens to choose from
- Each step in the chain is encrypted to avoid observers following the messages through the chain
- Remailers also delay and reorder messages

Type I remailers

Cypherpunk (type I) remailers removed the central point of trust

- Messages are now sent through a "chain" of several remailers, with dozens to choose from
- Each step in the chain is encrypted to avoid observers following the messages through the chain
- Remailers also delay and reorder messages

Support for pseudonymity is dropped: no replies!

Type I remailers: how to chain remailers?

Type I remailers: how to chain remailers?

Step 1: choose remailers:

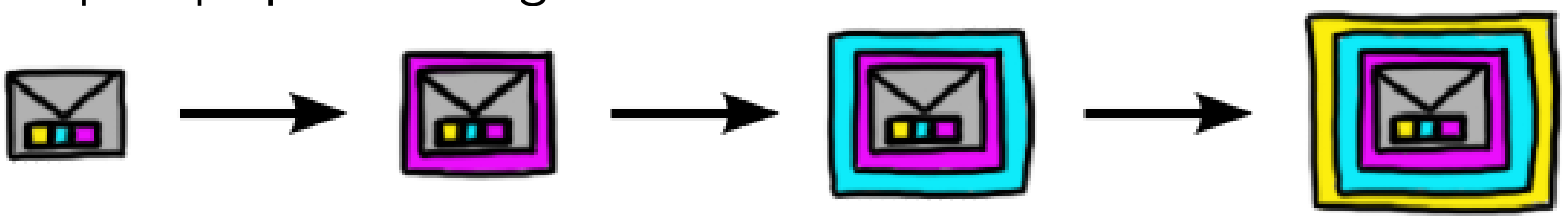




Type I remailers: how to chain remailers?

Step 1: choose remailers:

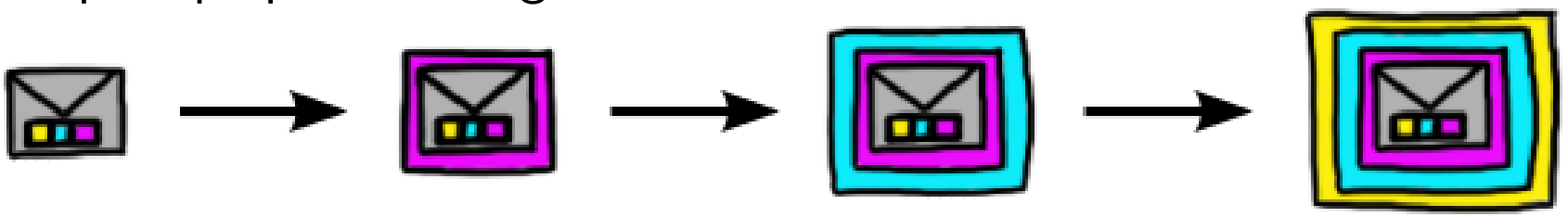
Step 2: prepare messages:



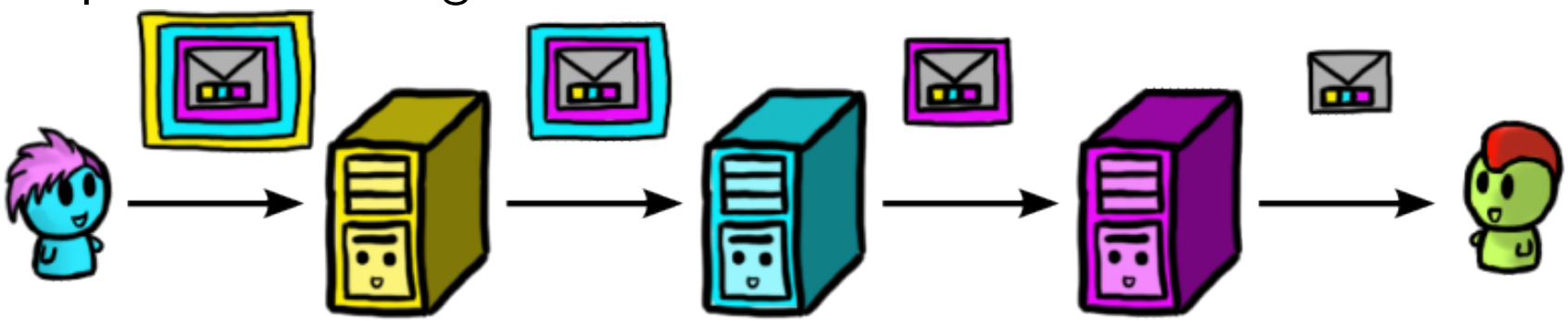
Type I remailers: how to chain remailers?

Step 1: choose remailers:

Step 2: prepare messages:



Step 3: send through the chain:



Nym servers / pseudonymous remailers

How to do replies? (i.e., recovering pseudonymity)

Nym servers / pseudonymous remailers

How to do replies? (i.e., recovering pseudonymity)

Q: How many of you ever sent passports to IRCC?

Nym servers / pseudonymous remailers

How to do replies? (i.e., recovering pseudonymity)

Q: How many of you ever sent passports to IRCC?

A:

Nym servers / pseudonymous remailers

How to do replies? (i.e., recovering pseudonymity)

- "nym servers" mapped pseudonyms to "reply blocks" that contained a nested encrypted chain of type I remailers.
- User A approaches a nym server with a chain of reply blocks for the nym server to relay back responses
- User A sends an anonymous mail B (via a chain of Type I remailers), including the chain of reply blocks
- User B responds to the nym server by attaching the response to the end of the reply blocks
- nym server relay the response back to user A by following the chain of reply blocks

Type II remailers

Mixmaster (type II) remailers appeared in the late 1990s

- Constant-length messages to avoid an observer watching "that big file" travel through the network
- Protections against replay attacks
- Improved message reordering

Requires a special email client to construct the message fragments

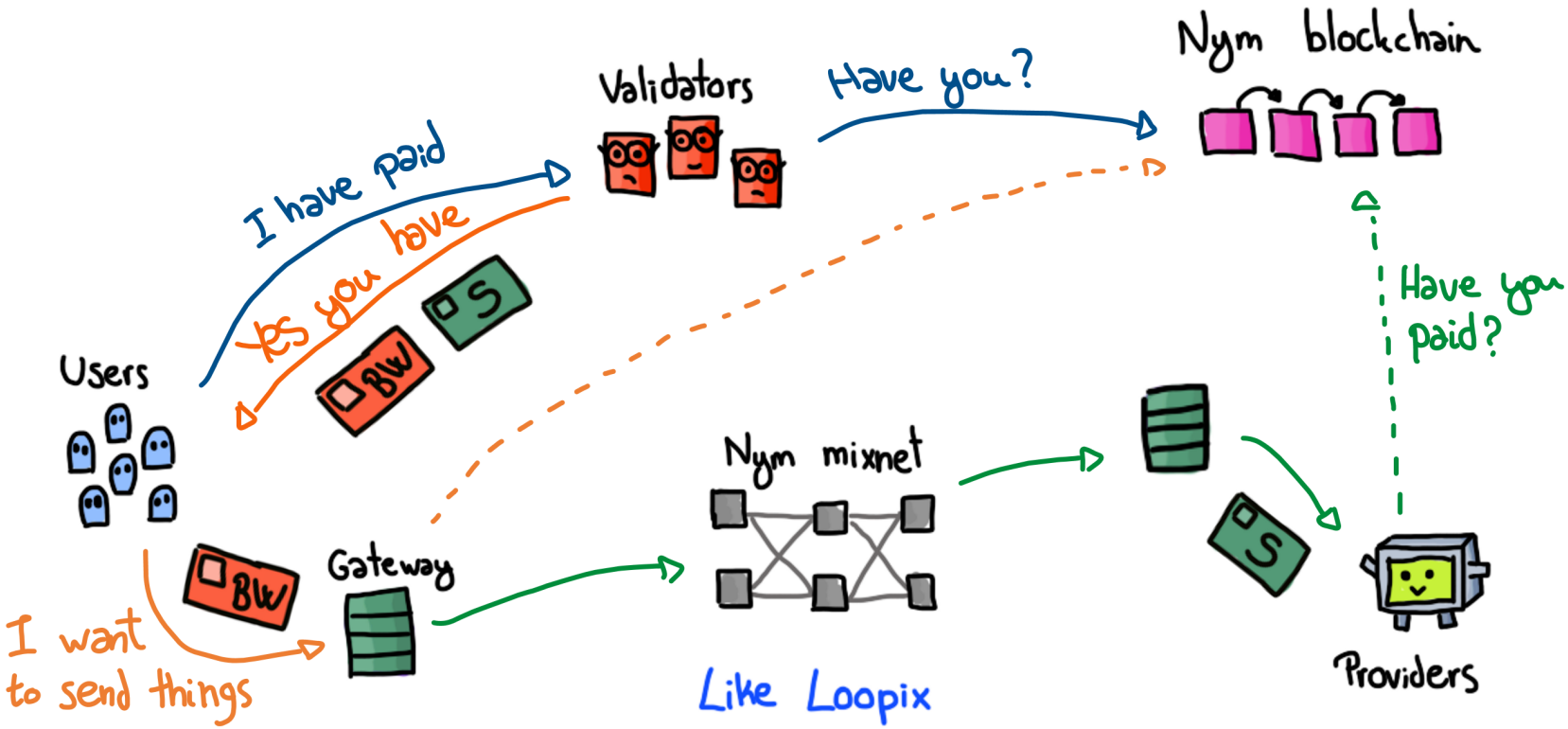
Type III remailers

Mixminion (type III) remailer appears in the 2000s

- Native (and much improved) support for pseudonymity
  - No longer reliant on type I reply blocks
  - Instead, relies on mix networks
- Improved protection against replay and key compromise attacks

But it's not very well deployed or mature, i.e., "you shouldn't trust Mixminion with your anonymity yet"

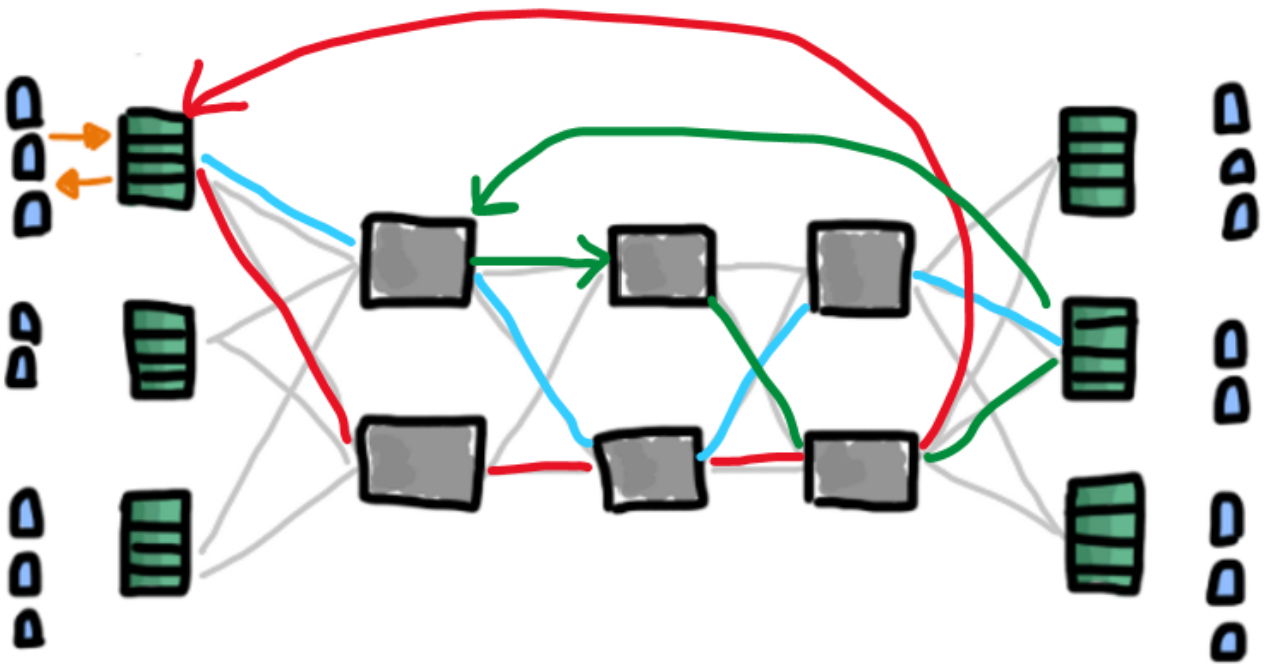
The Nym Network [Claudia Diaz, Harry Halpin, and Aggelos Kiayias (2021)]



The Nym Network

Nym is a new privacy infrastructure with three main components:

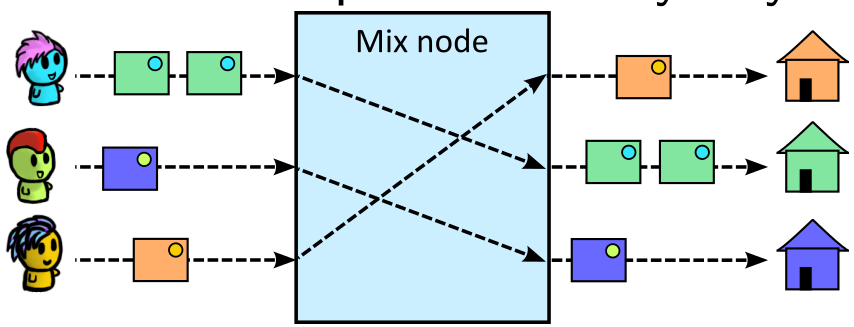
- The Nym mixnet: Loopix
- The Nym credentials
- The Nym token



Outline

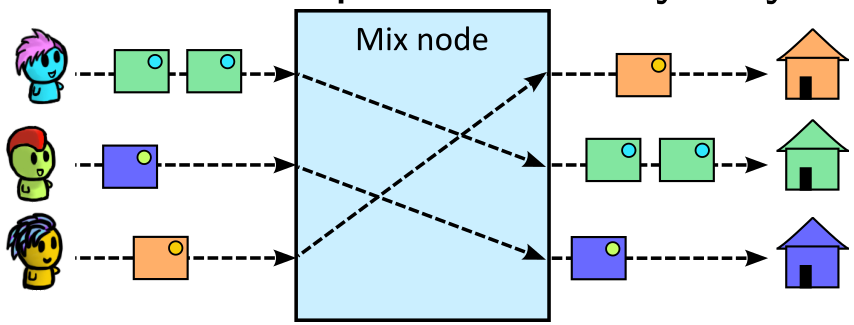
Mixes: basic operations

How do we provide anonymity?

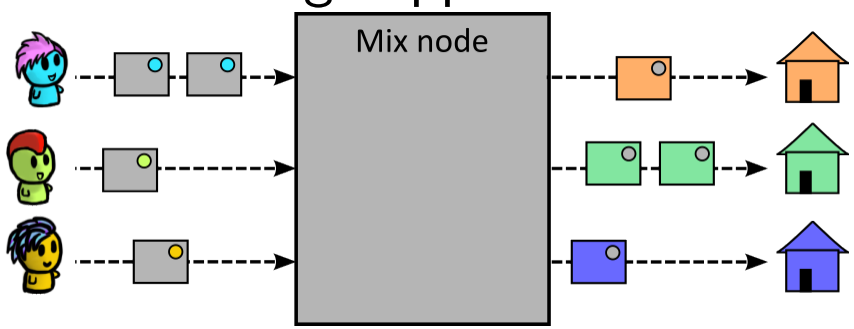


Mixes: basic operations

How do we provide anonymity?

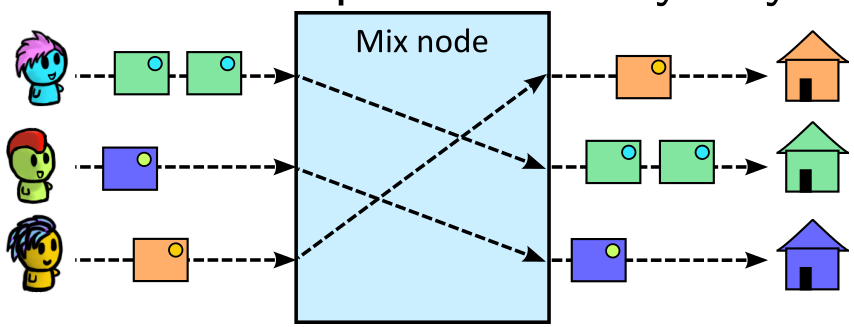


Change appearance!

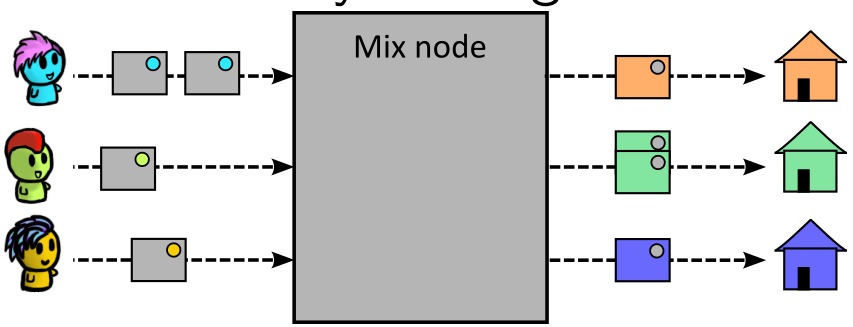


Mixes: basic operations

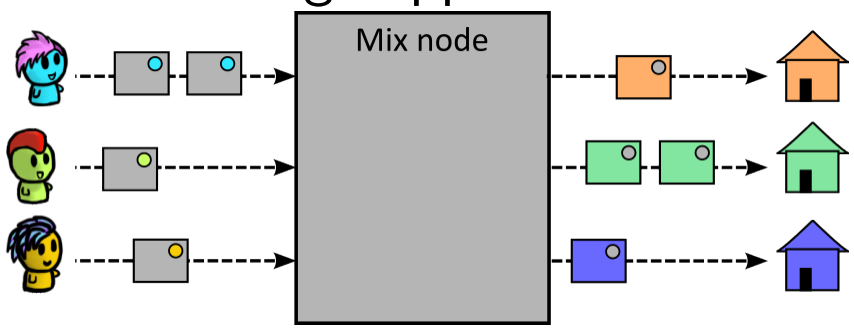
How do we provide anonymity?



Delay messages!

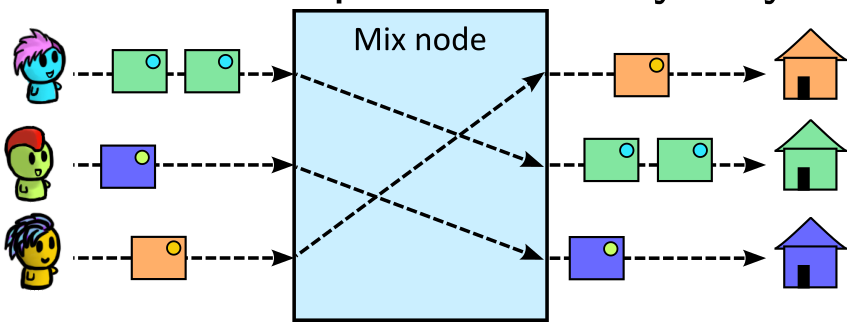


Change appearance!

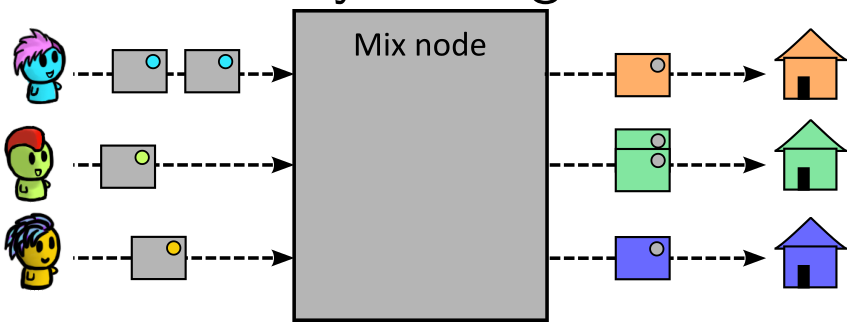


Mixes: basic operations

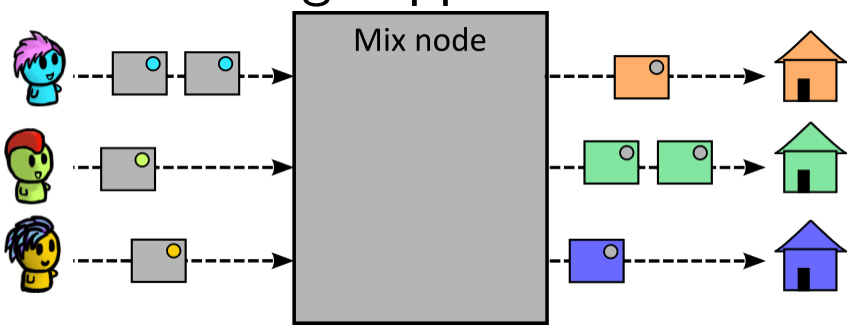
How do we provide anonymity?



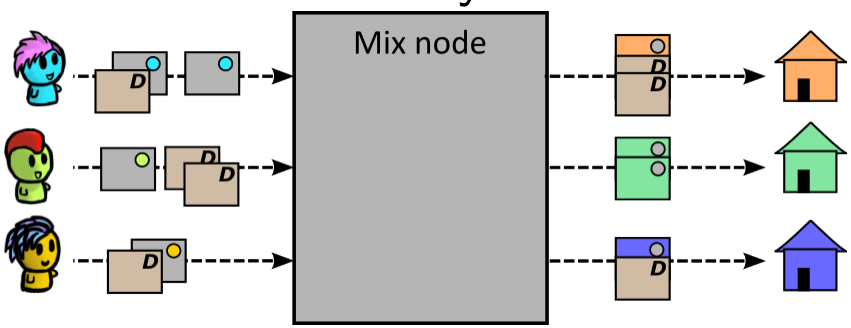
Delay messages!



Change appearance!

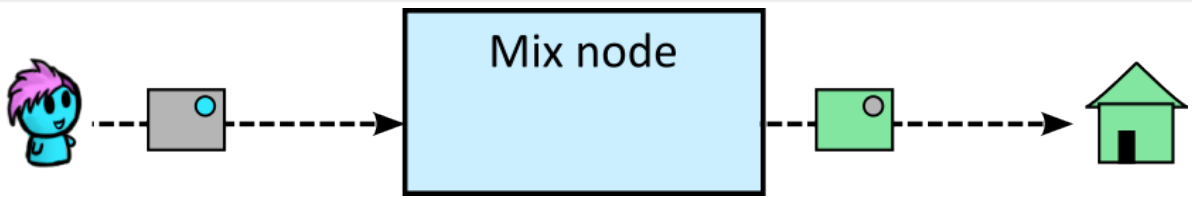


Add dummy traffic!



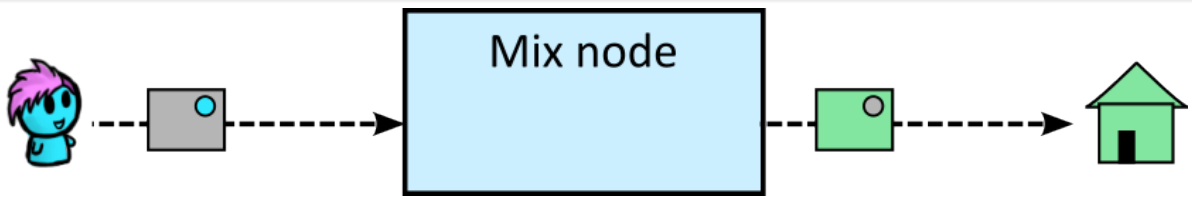
Operation 1: Changing Appearance

Q: How can we achieve this? (clue: we have some crypto tools!)



Operation 1: Changing Appearance

Q: How can we achieve this? (clue: we have some crypto tools!)



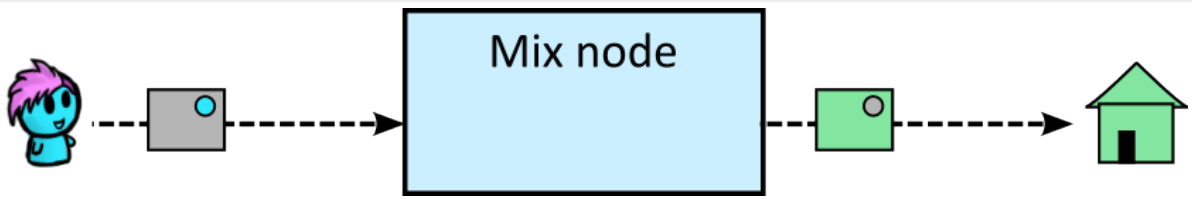
A: We can encrypt the output message with the Mix's key

$$\square = E_{K_{mix}}(\square)$$

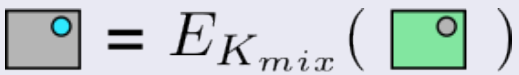
$$\square = E_{K_{Bob}}(m)$$

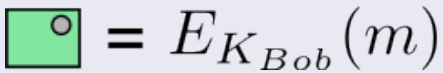
Operation 1: Changing Appearance

Q: How can we achieve this? (clue: we have some crypto tools!)



A: We can encrypt the output message with the Mix's key



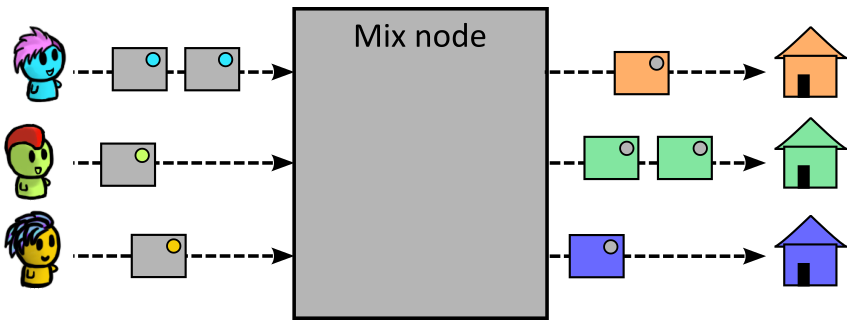


This "layered encryption" concept is called *onion routing*, and we will see it later in Tor.

Operation 2: Delaying Messages

Q: How do we do this?

- Do we add a random delay to each message?
- Do we add a deterministic delay to each message?
- Do we add a constant delay to each message?

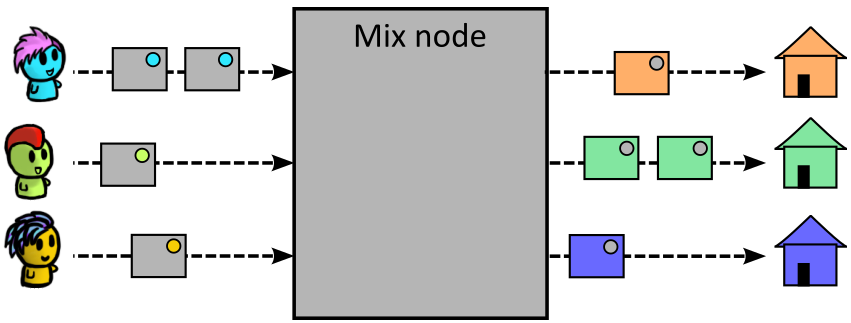


Notes on deterministic delay: it's not constant, it depends on the arrival time and/or other messages.

Operation 2: Delaying Messages

Q: How do we do this?

- Do we add a random delay to each message?
- Do we add a deterministic delay to each message?
- Do we add a constant delay to each message?



Notes on deterministic delay: it's not constant, it depends on the arrival time and/or other messages.

A: Yes. Yes. No.

We will see some examples on deterministic delay next!

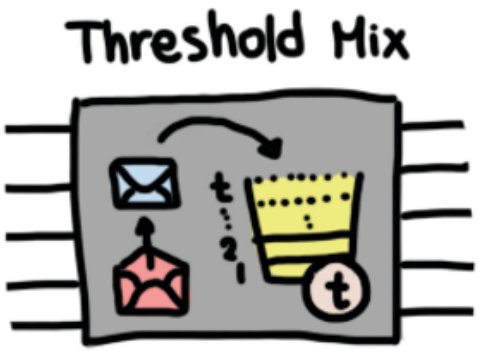
Delay tactics: threshold and timed

Some popular mixes types are threshold and timed mixes. These mixes gather messages until a flushing condition triggers. When this condition happens, this marks the end of a round.

Delay tactics: threshold and timed

Some popular mixes types are threshold and timed mixes. These mixes gather messages until a flushing condition triggers. When this condition happens, this marks the end of a round.

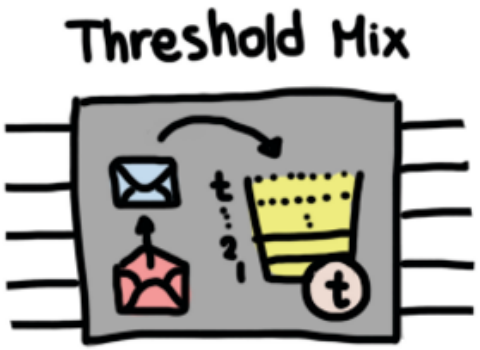
Threshold mix: it gathers t messages, then it flushes them.



Delay tactics: threshold and timed

Some popular mixes types are threshold and timed mixes. These mixes gather messages until a flushing condition triggers. When this condition happens, this marks the end of a round.

Threshold mix: it gathers t messages, then it flushes them.

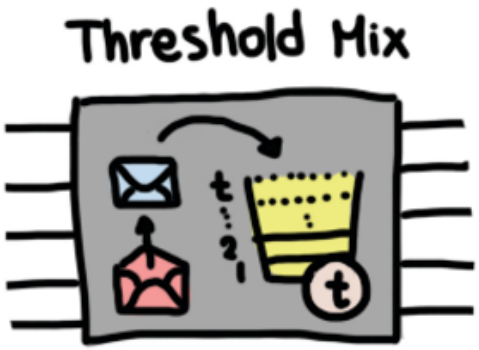


Timed mix: it gathers messages until a timer expires

Delay tactics: threshold and timed

Some popular mixes types are threshold and timed mixes. These mixes gather messages until a flushing condition triggers. When this condition happens, this marks the end of a round.

Threshold mix: it gathers t messages, then it flushes them.



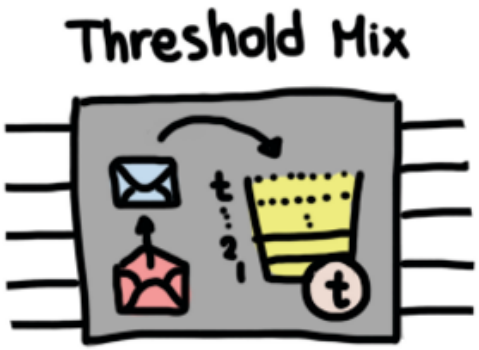
Timed mix: it gathers messages until a timer expires

Q: Which of the two is better?

Delay tactics: threshold and timed

Some popular mixes types are threshold and timed mixes. These mixes gather messages until a flushing condition triggers. When this condition happens, this marks the end of a round.

Threshold mix: it gathers t messages, then it flushes them.



Timed mix: it gathers messages until a timer expires

Q: Which of the two is better?

A: It depends... the threshold mix ensures a certain mixing size, the timed mix ensures a maximum message delay.

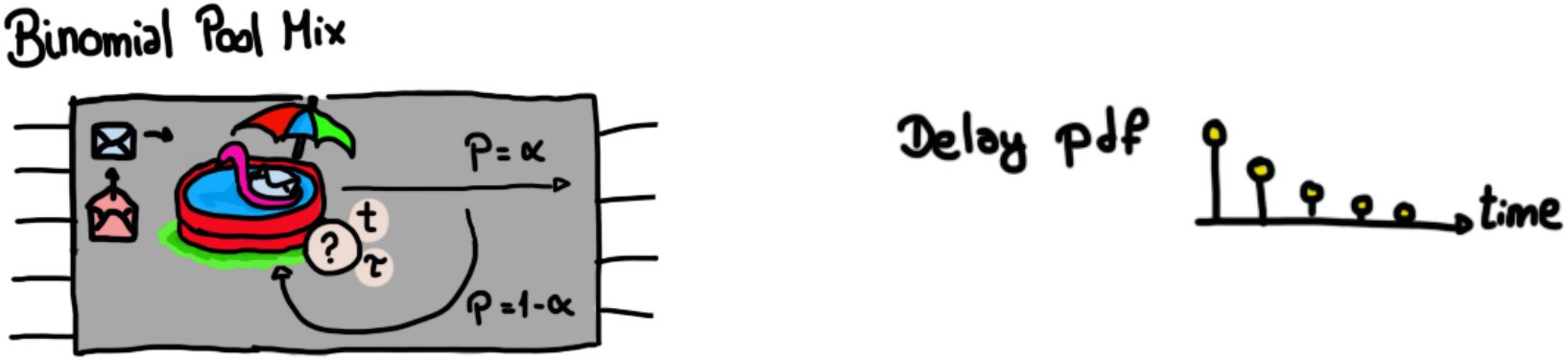
Pool mixes

When a (threshold/timed) mix keeps some messages inside after a round ends, it is called a *pool mix*.

Pool mixes

When a (threshold/timed) mix keeps some messages inside after a round ends, it is called a *pool mix*.

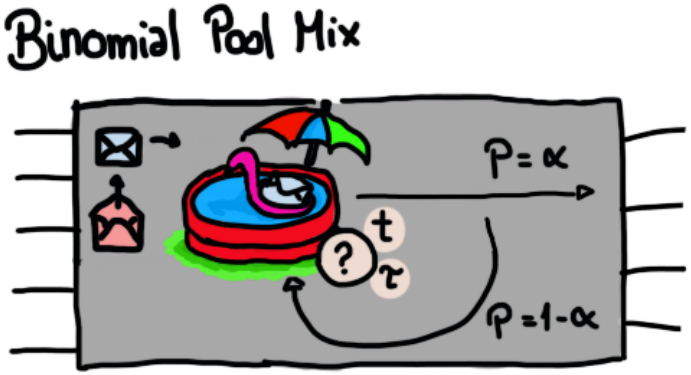
Example: the binomial pool mix forwards each message with probability α, and keeps it inside the mix with probability $1 - \alpha$.

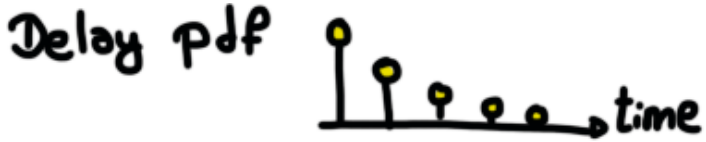


Pool mixes

When a (threshold/timed) mix keeps some messages inside after a round ends, it is called a *pool mix*.

Example: the binomial pool mix forwards each message with probability α, and keeps it inside the mix with probability $1 - \alpha$.



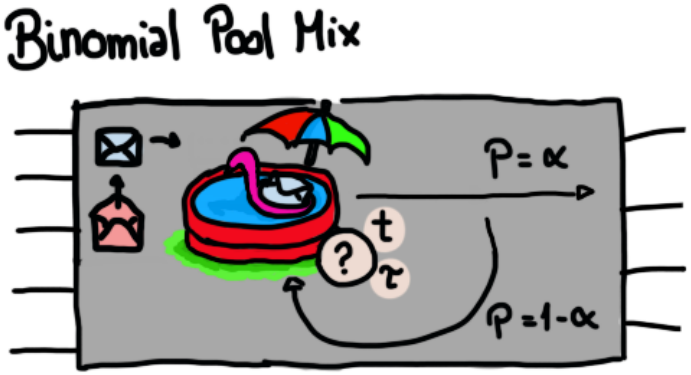


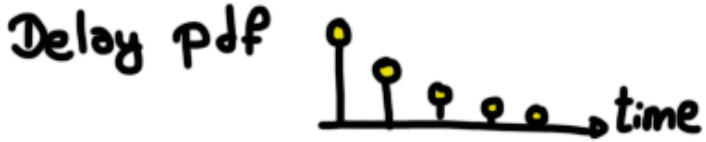
Q: What are the pros and cons of this?

Pool mixes

When a (threshold/timed) mix keeps some messages inside after a round ends, it is called a *pool mix*.

Example: the binomial pool mix forwards each message with probability α, and keeps it inside the mix with probability $1 - \alpha$.





Q: What are the pros and cons of this?

A: Pros: more anonymity; cons: more delay

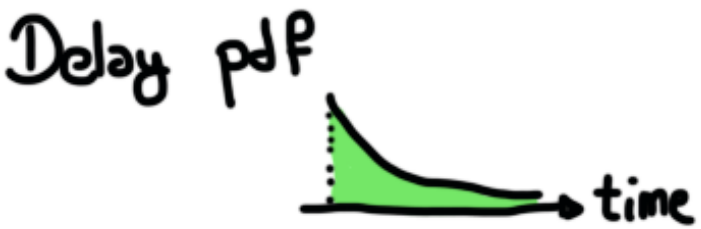
Continuous-time or Stop-and-Go (SG) mixes

Some mixes do not work on "batches" or "rounds", and instead delay each message independently: these are called continuous-time mixes or Stop-and-Go (SG) mixes.

Continuous-time or Stop-and-Go (SG) mixes

Some mixes do not work on "batches" or "rounds", and instead delay each message independently: these are called continuous-time mixes or Stop-and-Go (SG) mixes.

- Mixes that delay messages following an exponential distribution are very popular (Loopix, Nym).
- The user can choose the delay and include it in the message



Mixnets

Q: Why sending messages through a single mix is not a good idea?

Mixnets

Q: Why sending messages through a single mix is not a good idea?

A: There's a single point of failure, and the mix knows the message correspondence.

Mixnets

Q: Why sending messages through a single mix is not a good idea?

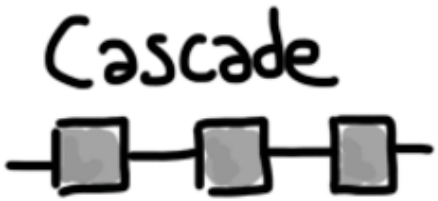
A: There's a single point of failure, and the mix knows the message correspondence.

We can chain mixes to create a mixnet.

Mixnets have different topologies, depending on which nodes a message can travel between.

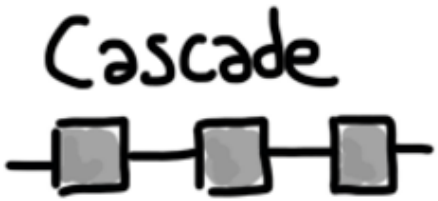
Mixnet topologies

- Cascade: one after the other



Mixnet topologies

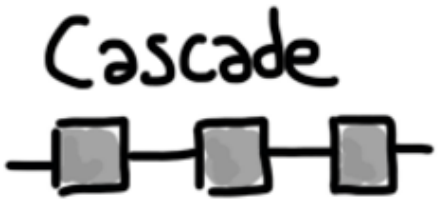
- Cascade: one after the other



- Freeroute: all of them are connected

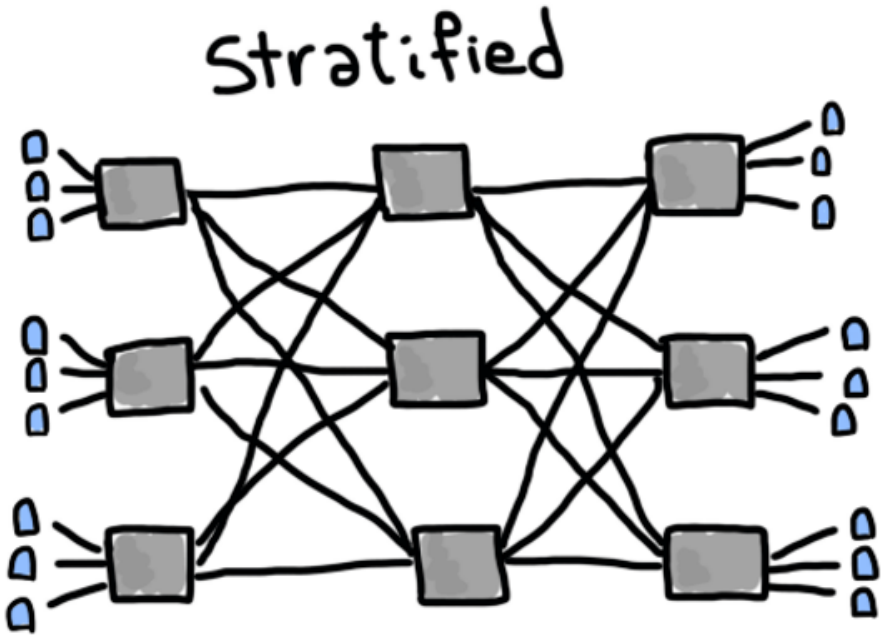
Mixnet topologies

- Cascade: one after the other



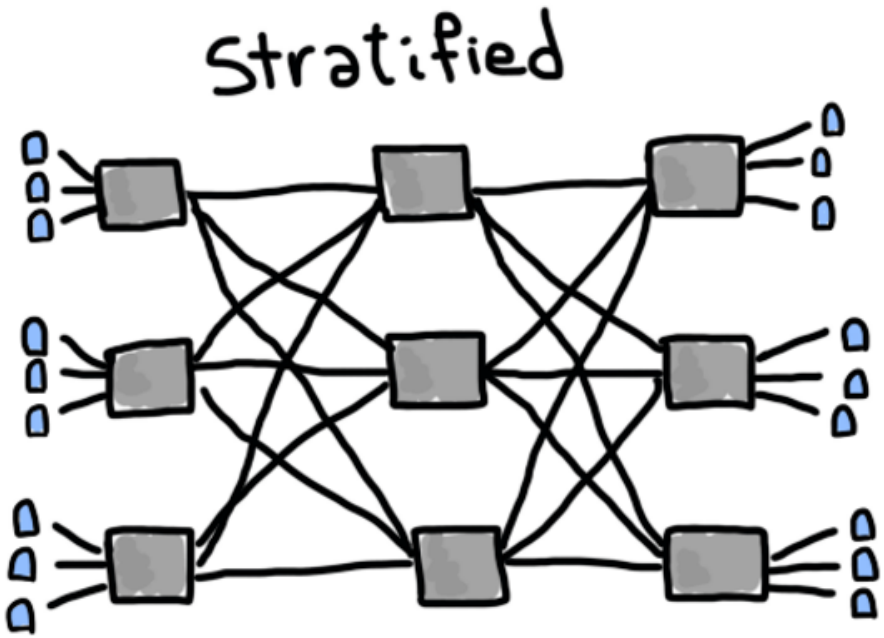
- Freeroute: all of them are connected

- Stratified: each layer is fully connected to the next layer



Operation 3: dummy messages

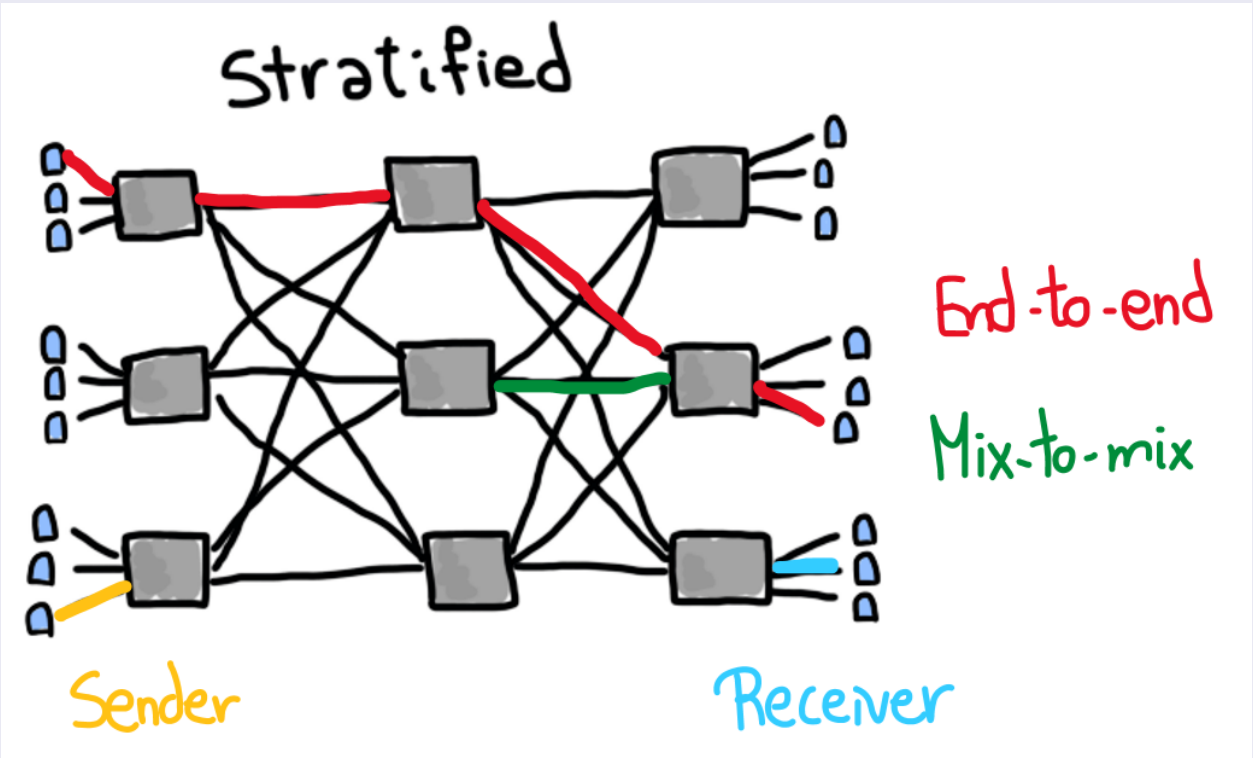
Q: Where do we add dummy traffic?



Operation 3: dummy messages

Q: Where do we add dummy traffic?

A: Anywhere! Everywhere!



Knowledge check

Q: What are the three basic operations of a mix node to provide anonymity?

Knowledge check

Q: What are the three basic operations of a mix node to provide anonymity?

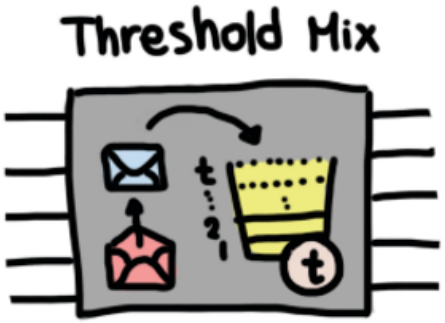
A: Change appearance, delay messages, add dummy traffic

Knowledge check

Q: What are the three basic operations of a mix node to provide anonymity?

A: Change appearance, delay messages, add dummy traffic

Q: Threshold mixes: pros and cons of increasing the threshold t?

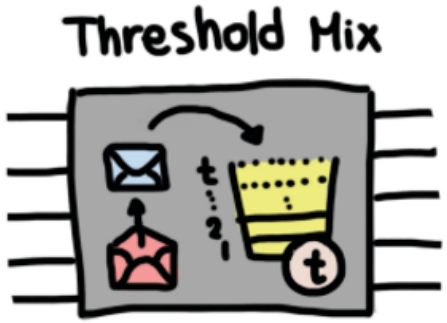


Knowledge check

Q: What are the three basic operations of a mix node to provide anonymity?

A: Change appearance, delay messages, add dummy traffic

Q: Threshold mixes: pros and cons of increasing the threshold t?



A: Increasing t improves anonymity but increases delay

Knowledge check

Q: Timed mixes: pros and cons of increasing the time τ?

Knowledge check

Q: Timed mixes: pros and cons of increasing the time τ?

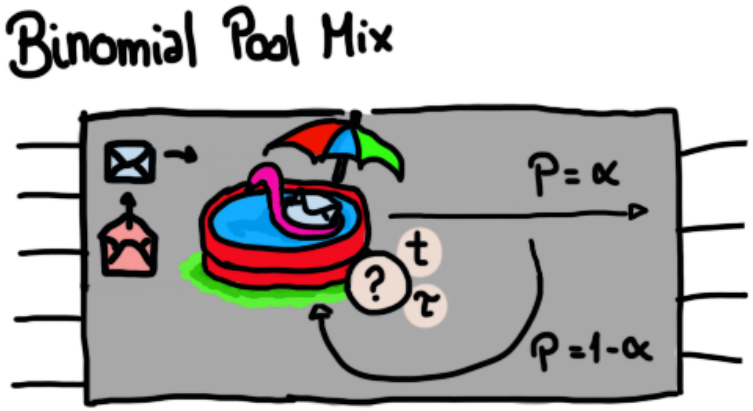
A: Increasing τ improves anonymity but increases delay

Knowledge check

Q: Timed mixes: pros and cons of increasing the time τ?

A: Increasing τ improves anonymity but increases delay

Q: Binomial pool mix: pros and cons of increasing the probability of forwarding a message α?

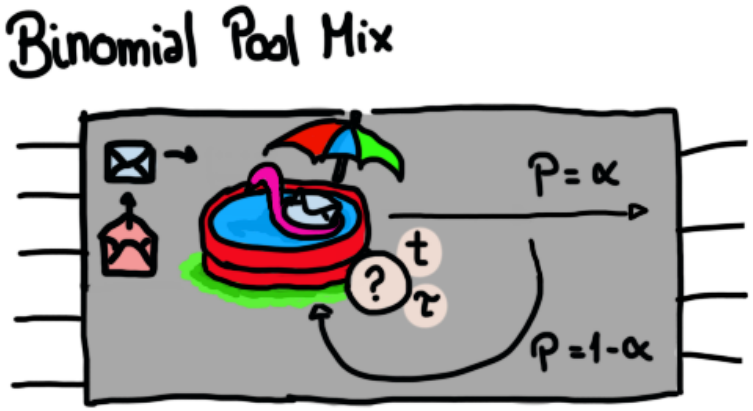


Knowledge check

Q: Timed mixes: pros and cons of increasing the time τ?

A: Increasing τ improves anonymity but increases delay

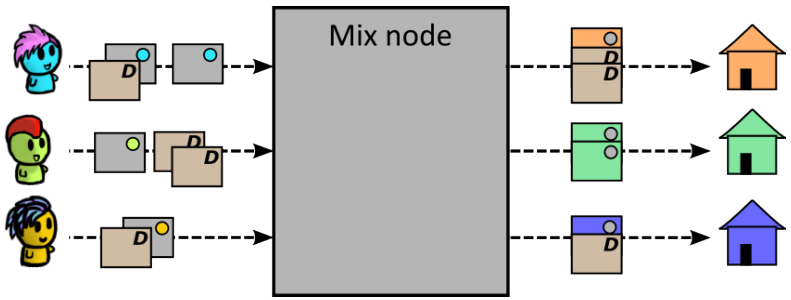
Q: Binomial pool mix: pros and cons of increasing the probability of forwarding a message α?



A: Increasing α decreases anonymity and delay

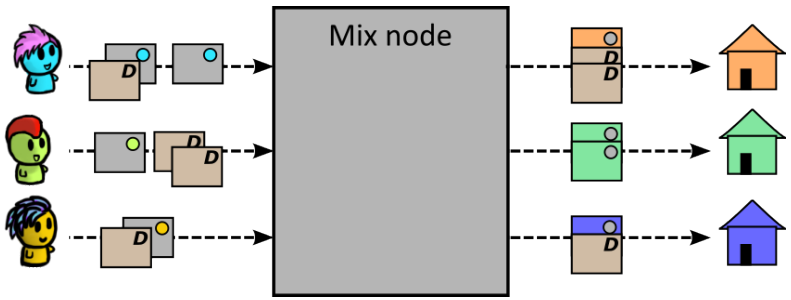
Knowledge check

Q: Dummy traffic: pros and cons of increasing the amount of dummy messages?



Knowledge check

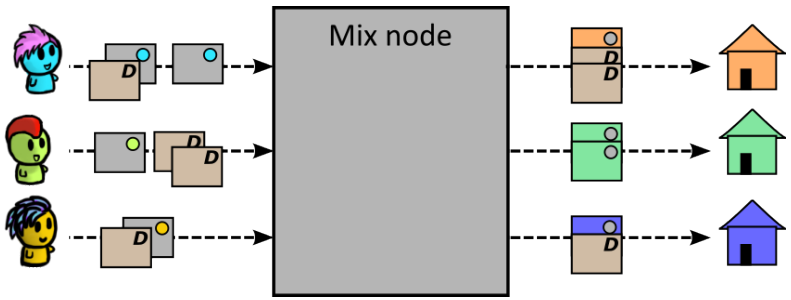
Q: Dummy traffic: pros and cons of increasing the amount of dummy messages?



A: More dummies require more bandwidth, but increase anonymity

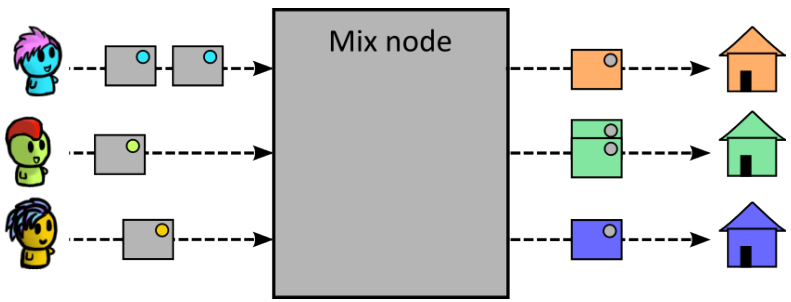
Knowledge check

Q: Dummy traffic: pros and cons of increasing the amount of dummy messages?



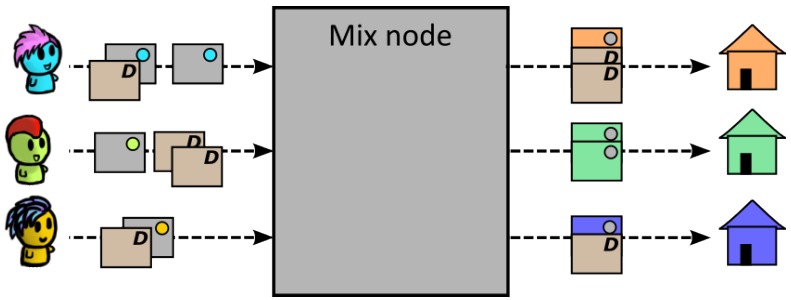
A: More dummies require more bandwidth, but increase anonymity

Q: What happens if the number of senders increases?



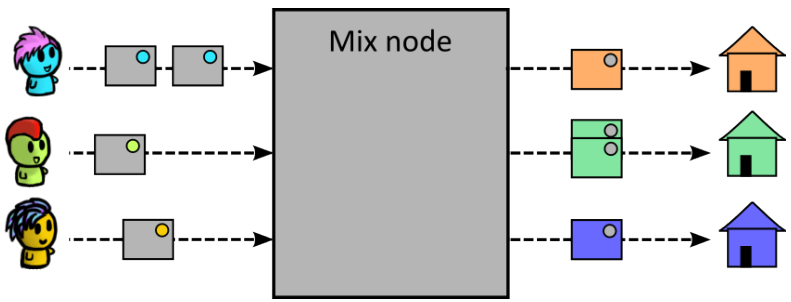
Knowledge check

Q: Dummy traffic: pros and cons of increasing the amount of dummy messages?



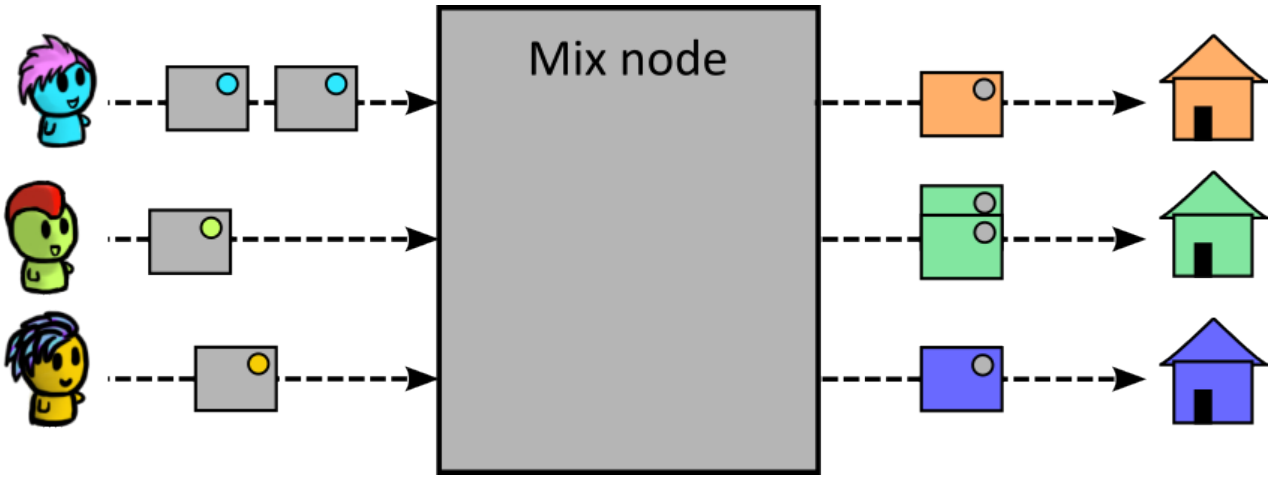
A: More dummies require more bandwidth, but increase anonymity

Q: What happens if the number of senders increases?



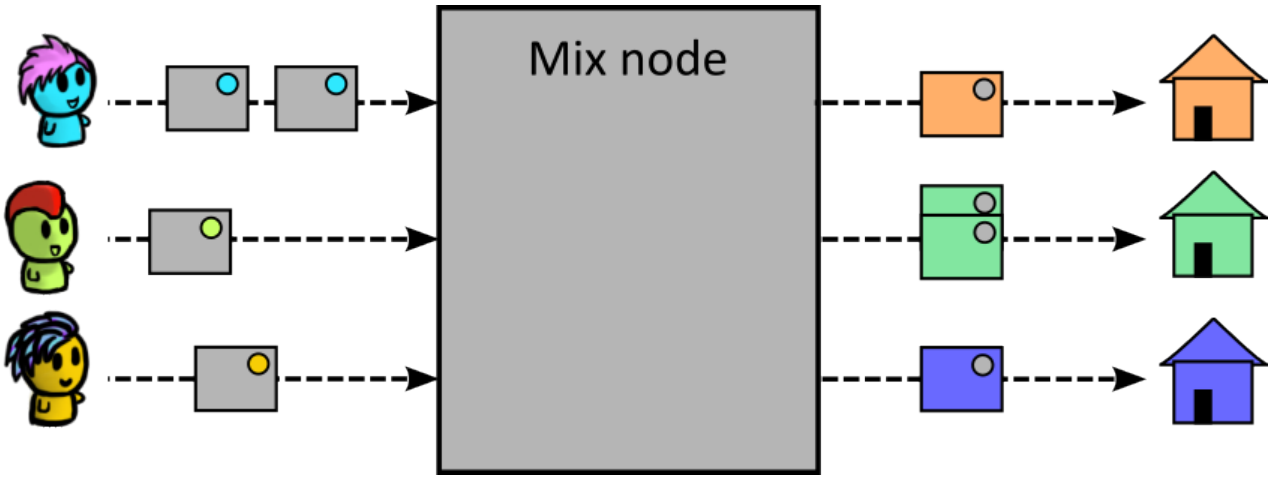
A: Depends on the actual mix/setting, but usually *anonymity loves company*. More people using the system usually improves its anonymity level.

Active adversary on a mix



Q: If you are an active adversary, how would you attack a mix? (e.g., a *threshold* mix)

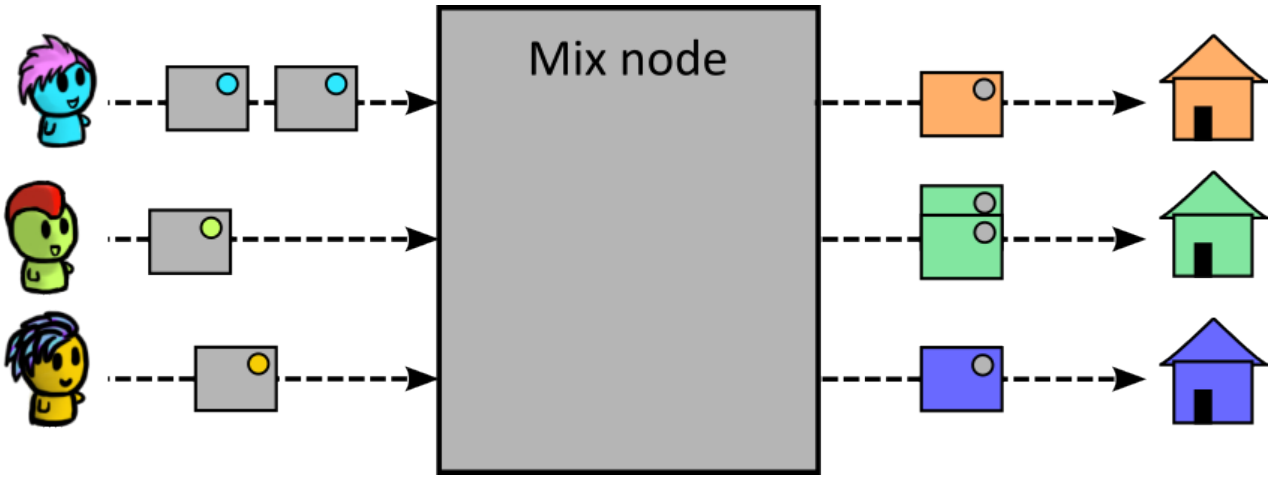
Active adversary on a mix



Q: If you are an active adversary, how would you attack a mix? (e.g., a *threshold* mix)

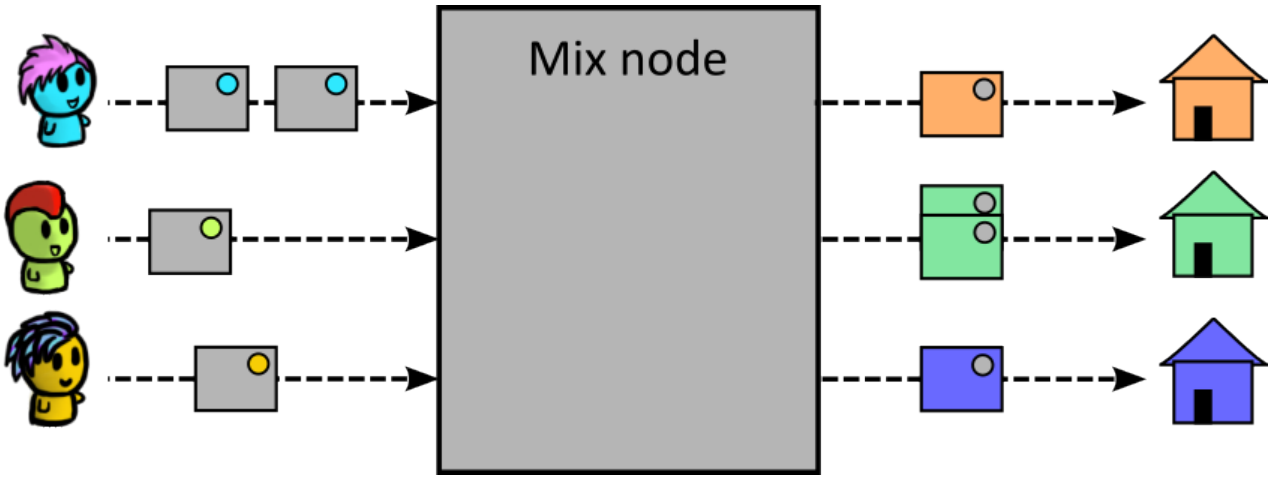
A: A possible attack: (*the* $n-1$ *attack*) Mallory sends all but one message (e.g., $t-1$ in this case). Mallory can identify her messages leaving the mix, so the remaining message has to be Alice's.

Short-term passive adversary on a mix



Q: If you are a passive adversary, how would you attack a mix (e.g., a *threshold* mix)

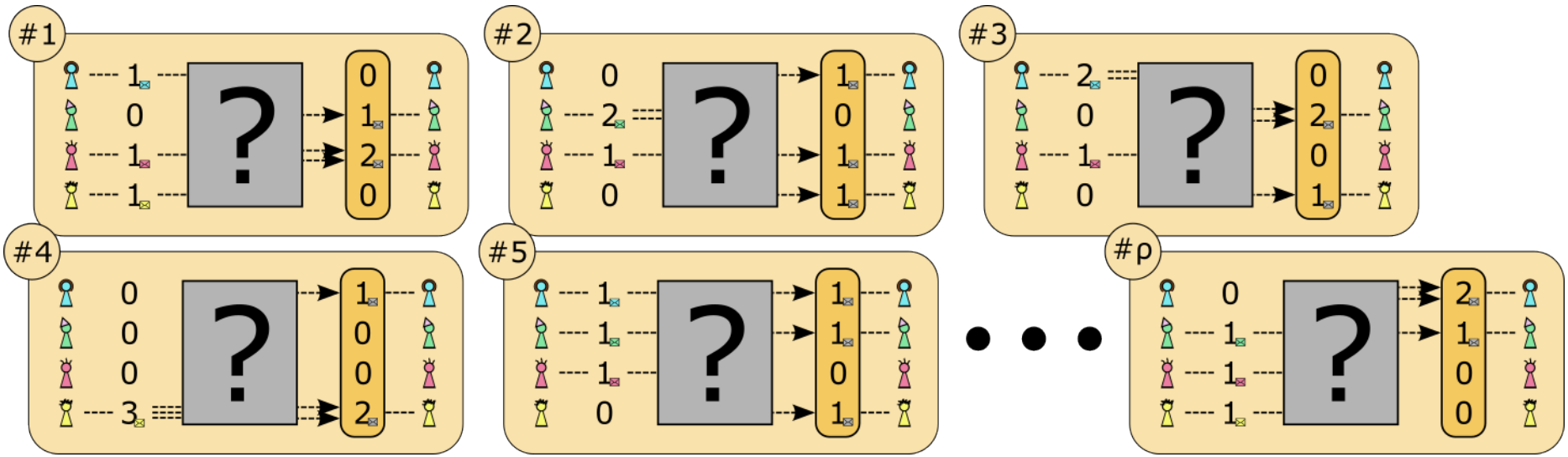
Short-term passive adversary on a mix



Q: If you are a passive adversary, how would you attack a mix (e.g., a *threshold* mix)

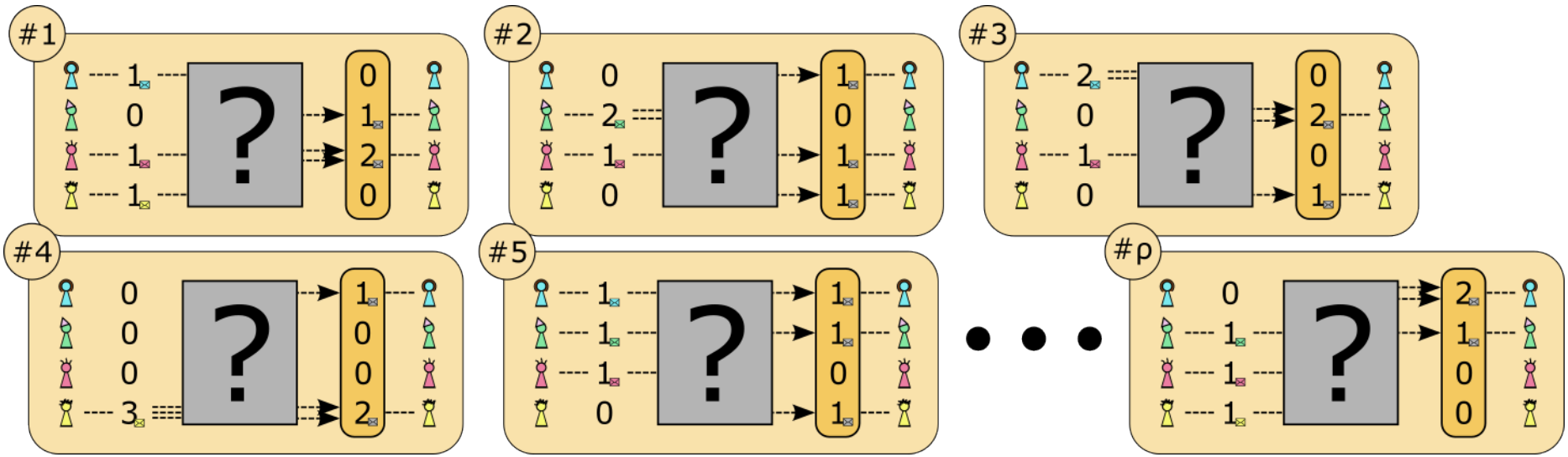
A: We can exploit sending behavior (e.g., Alice sent two messages, so "mostly likely" they were for the same recipient?)
But we can't do much more here...

Long-term passive adversary on a mix



Q: What if we are a passive adversary observing the mix for a long time? Who is Alice (first sender) most-likely sending messages to?

Long-term passive adversary on a mix



Q: What if we are a passive adversary observing the mix for a long time? Who is Alice (first sender) most-likely sending messages to?

A: Probably the second receiver...

Outline

Tor - purpose

Tor is a successful privacy enhancing technology that works at the transport layer with ≈2 million daily users

Tor - purpose

Tor is a successful privacy enhancing technology that works at the transport layer with ≈2 million daily users

Why do we need Tor when we have TLS?

- TLS protects data.
- We also want to protect metadata about the communication: e.g., IP addresses, browser fingerprints.

Tor - purpose

Tor is a successful privacy enhancing technology that works at the transport layer with ≈2 million daily users

Why do we need Tor when we have TLS?

- TLS protects data.
- We also want to protect metadata about the communication: e.g., IP addresses, browser fingerprints.

Tor is an anonymity network of nodes

- Scattered around the Internet are about 7,000 Tor nodes, also called Onion Routers

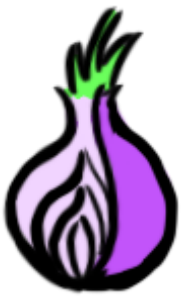
Tor makes internet browsing unlinkably anonymous. But Tor does not (and cannot) hide the existence of the transaction (website visit) altogether

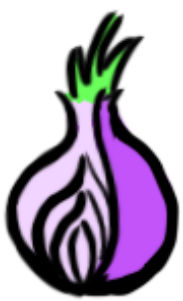
Build a Tor circuit

Alice wants to connect to a server without revealing her IP address

n_1

n_2



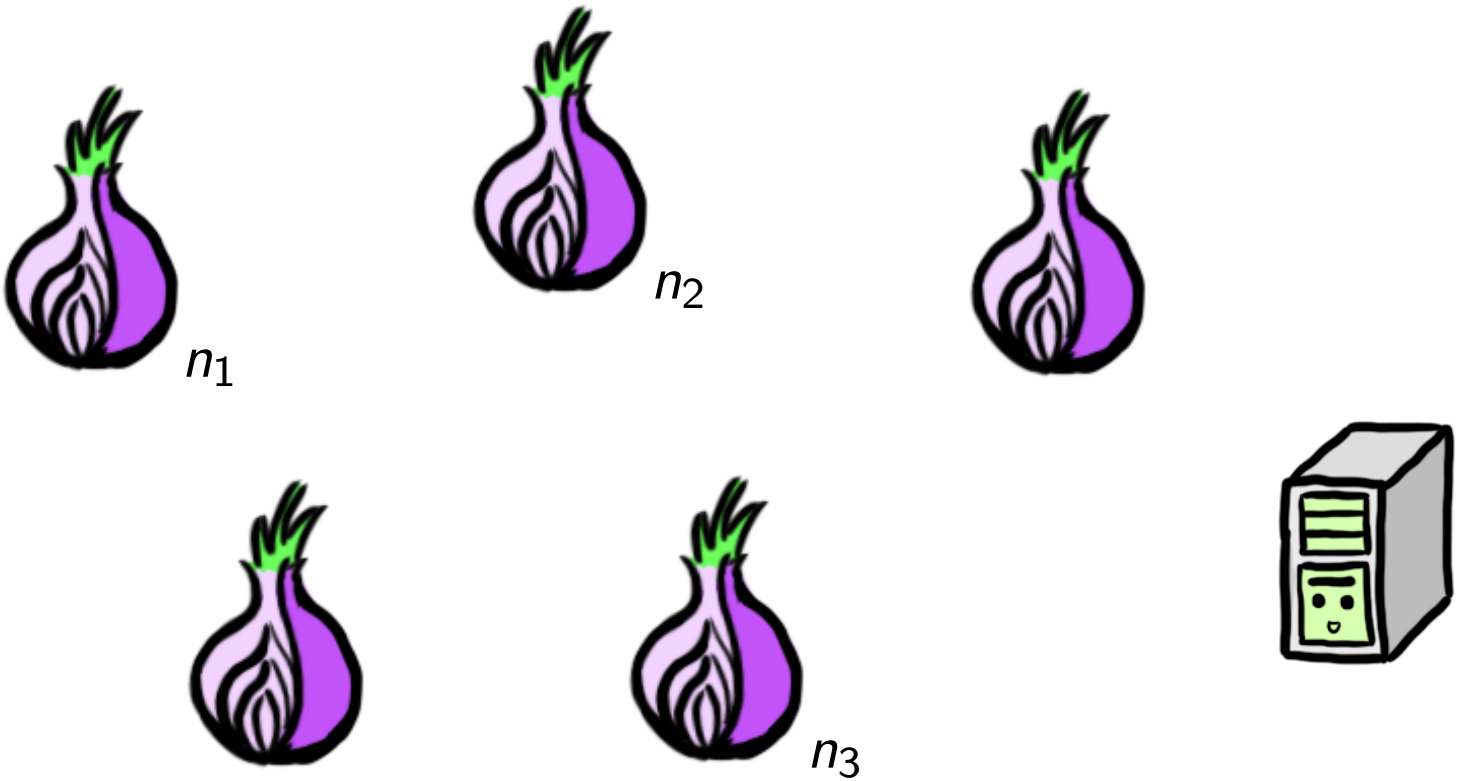


n_3

Alice has a global view of available Onion Routers

Build a Tor circuit

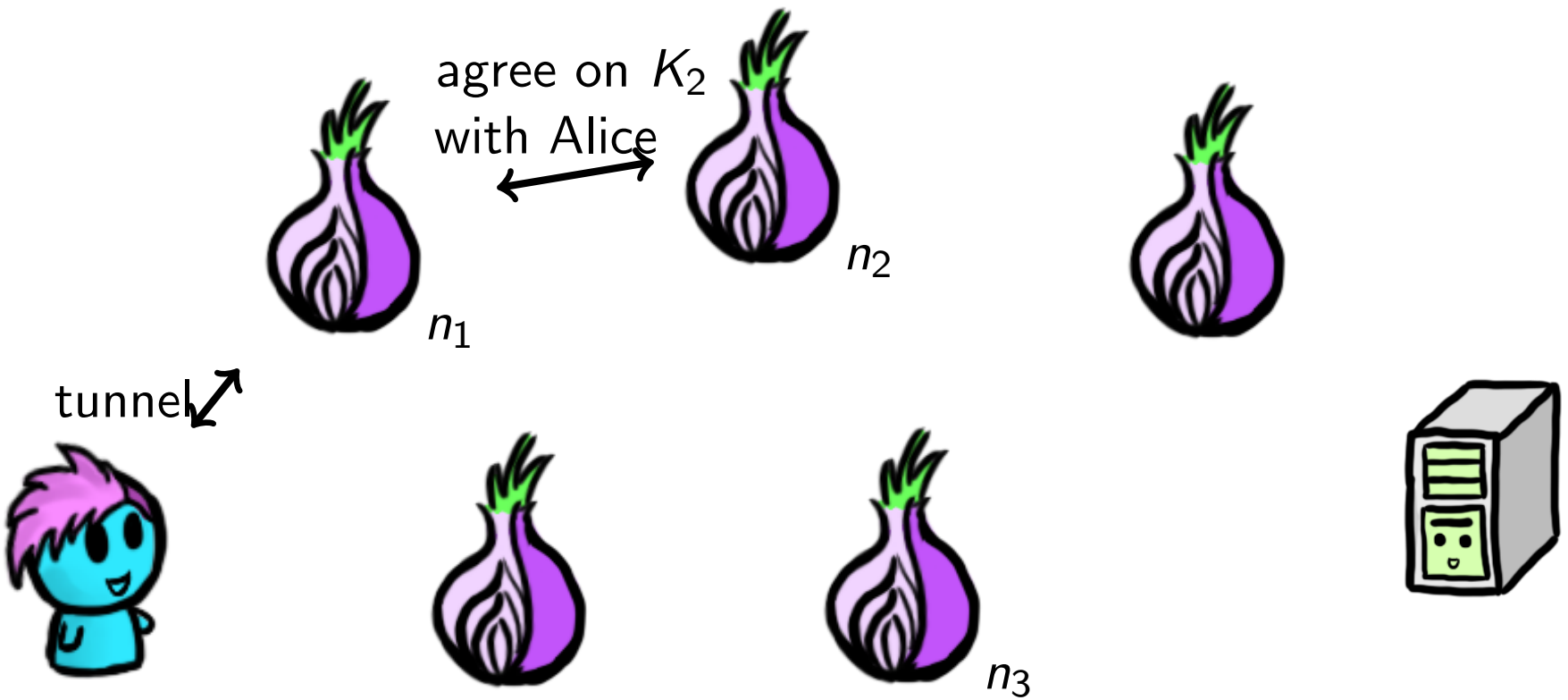
Alice picks one of the Tor nodes (n1) and uses public-key cryptography to establish an encrypted communication channel to it (much like TLS)



Result is a secret key K_1 shared by Alice and n1

Build a Tor circuit

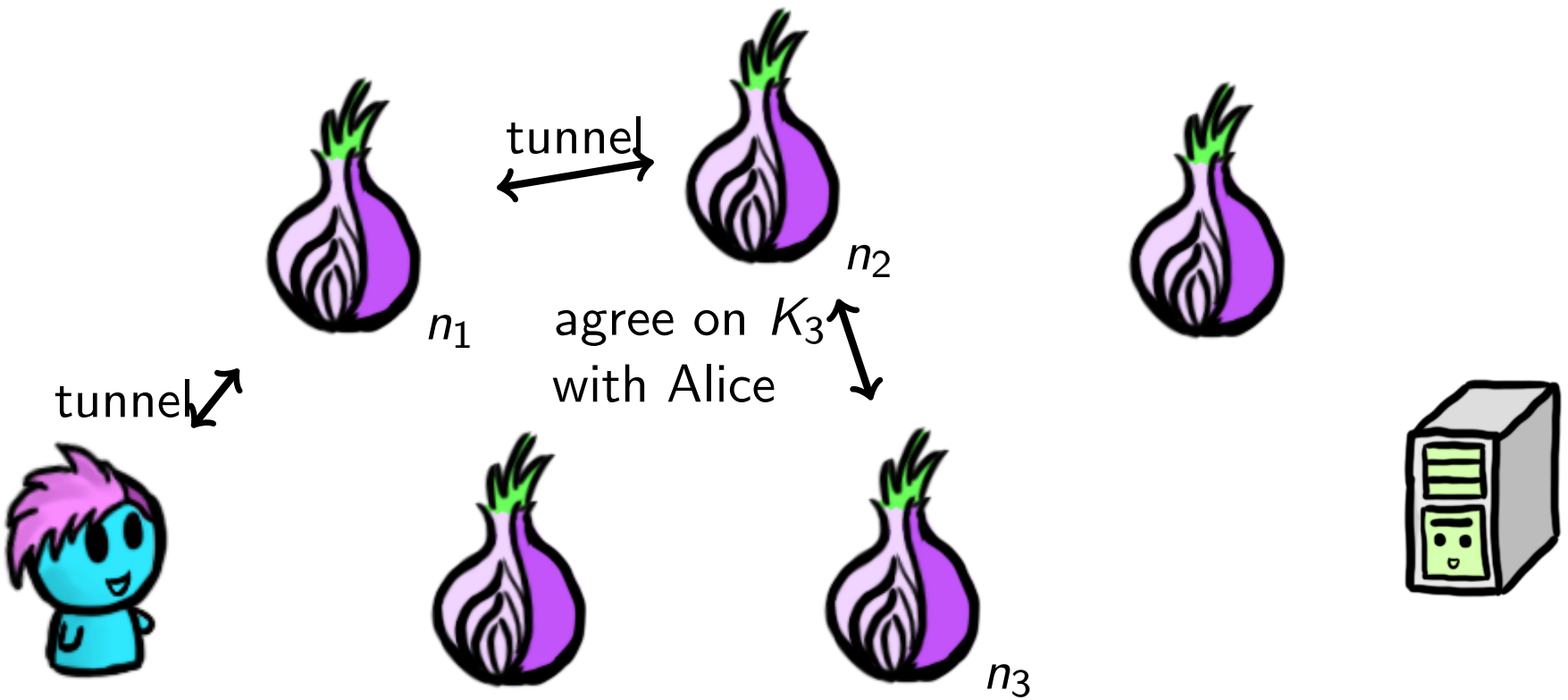
Alice tells n1 to contact a second node (n2), and establishes a new encrypted communication channel to n2, tunneled within the previous one to n1



Result is a secret key K_2 shared between Alice and n2, which is unknown to n1

Build a Tor circuit

Alice tells n2 to contact a third node (n3), and establishes a new encrypted communication channel to n3, tunneled within the previous one to n2

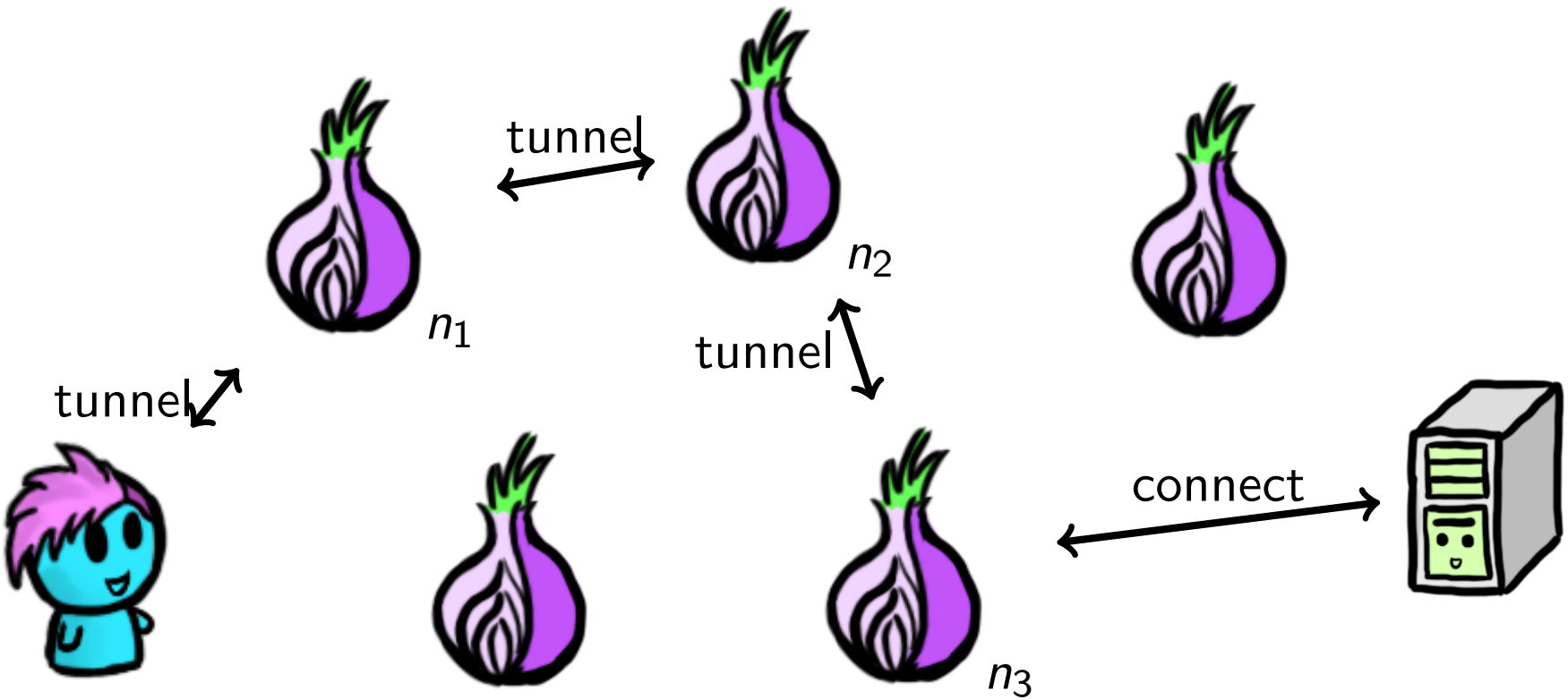


Result is a secret key K_3 shared between Alice and n3, which is unknown to n1 and n2

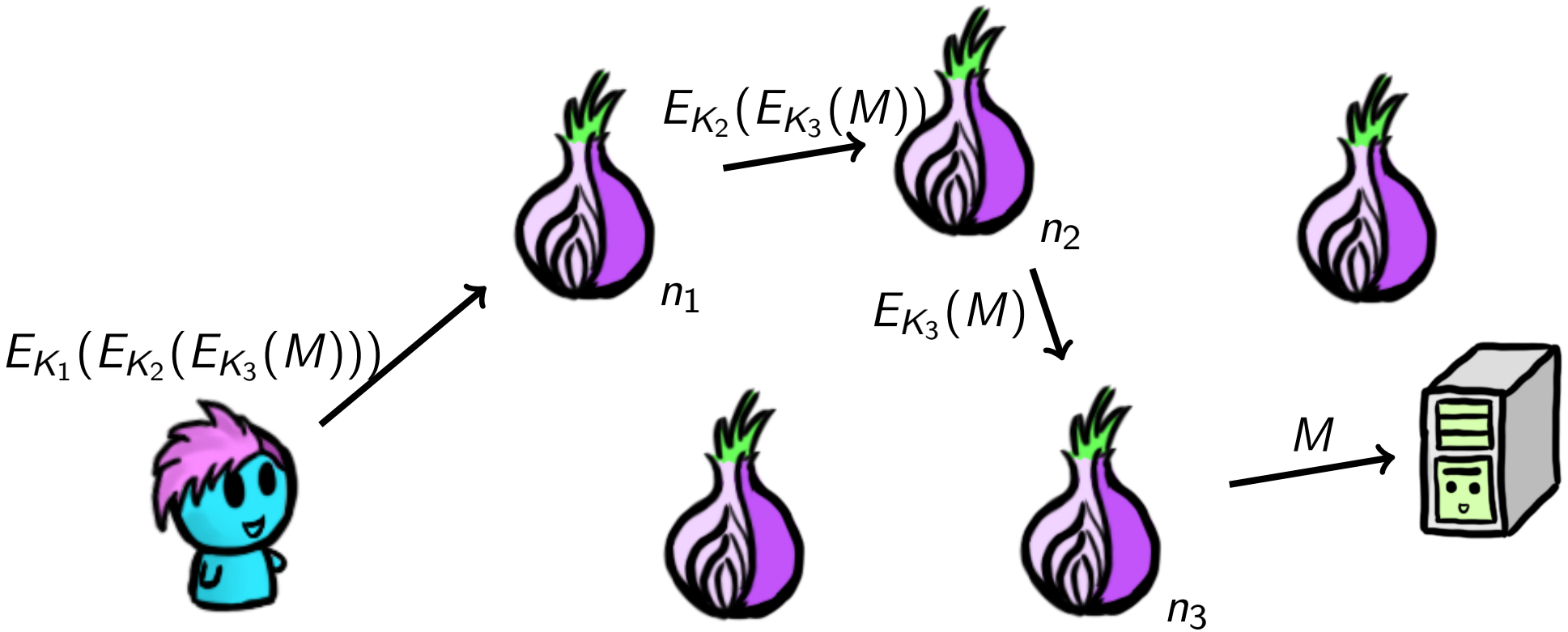
Build a Tor circuit

... And so on, for as many steps as she likes (usually 3) ...

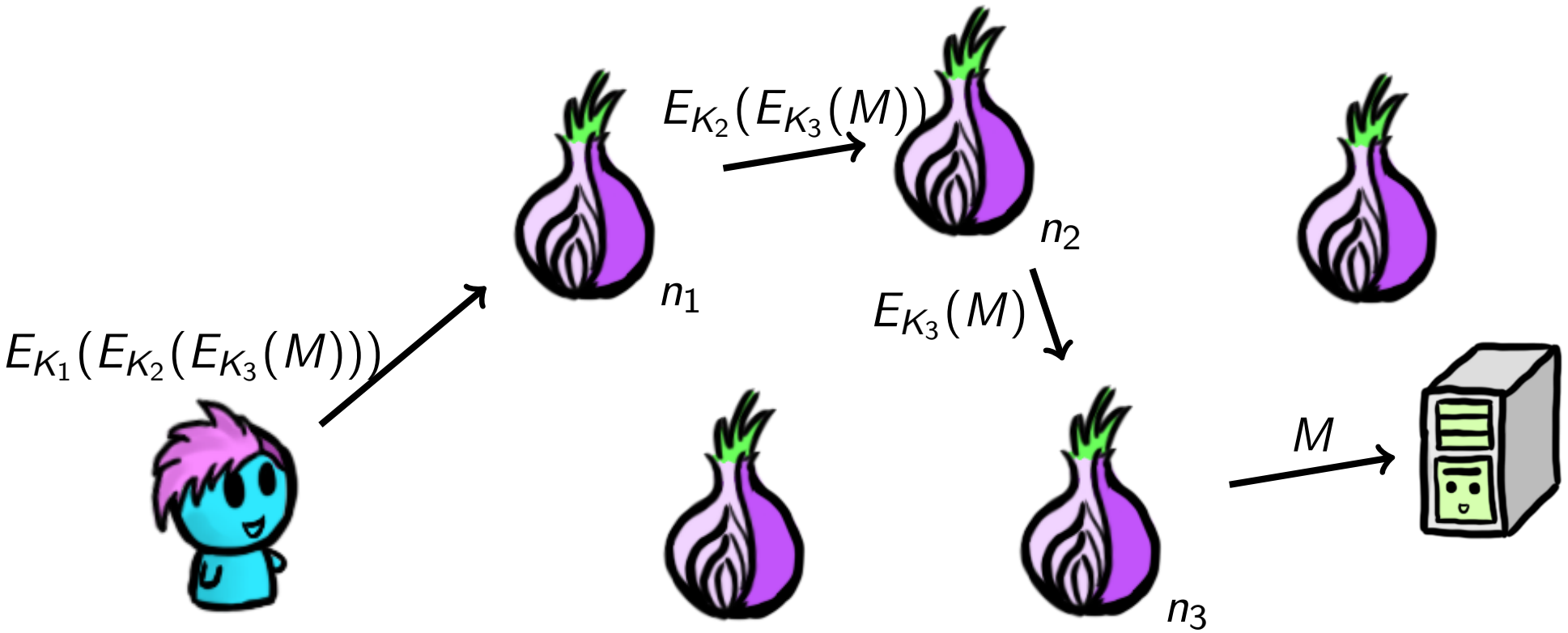
Alice tells the last node (within the layers of tunnels) to connect to the website



Sending messages with Tor



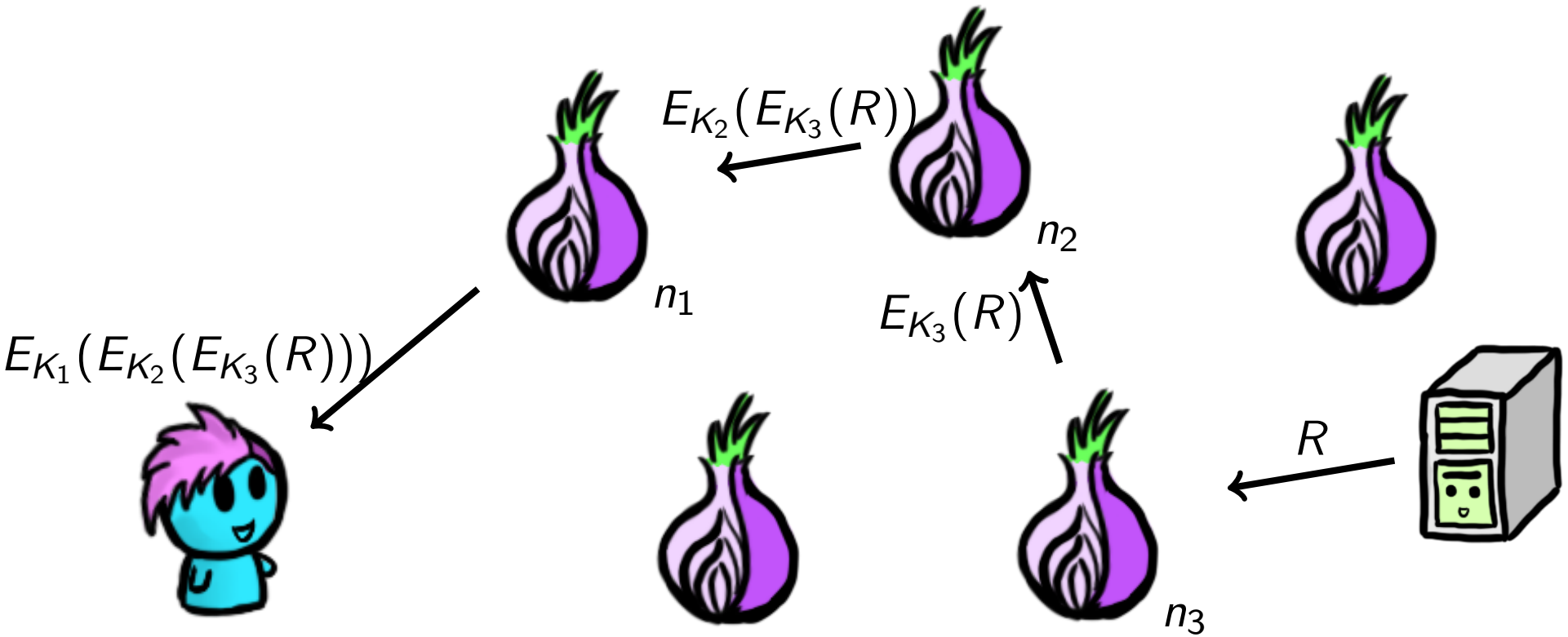
Sending messages with Tor



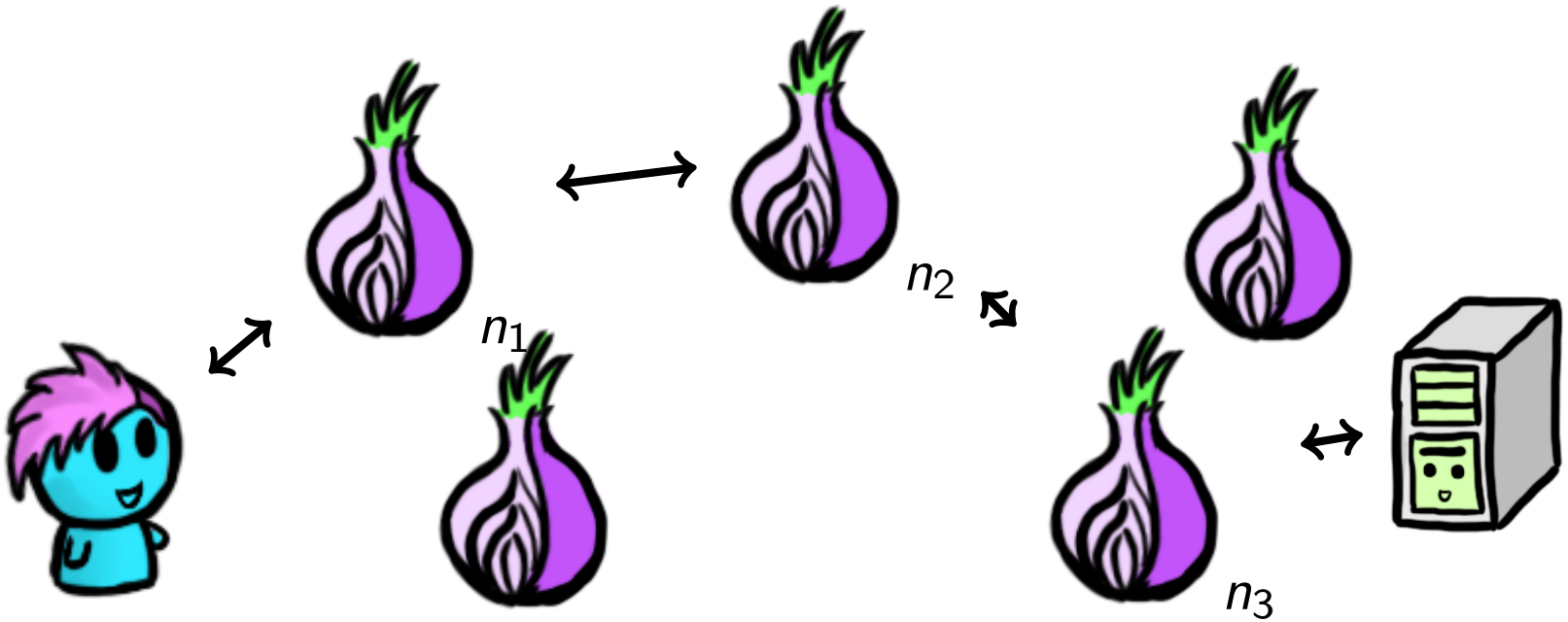
If connecting to a web server, M can also be encrypted (e.g., TLS)

Replies in Tor

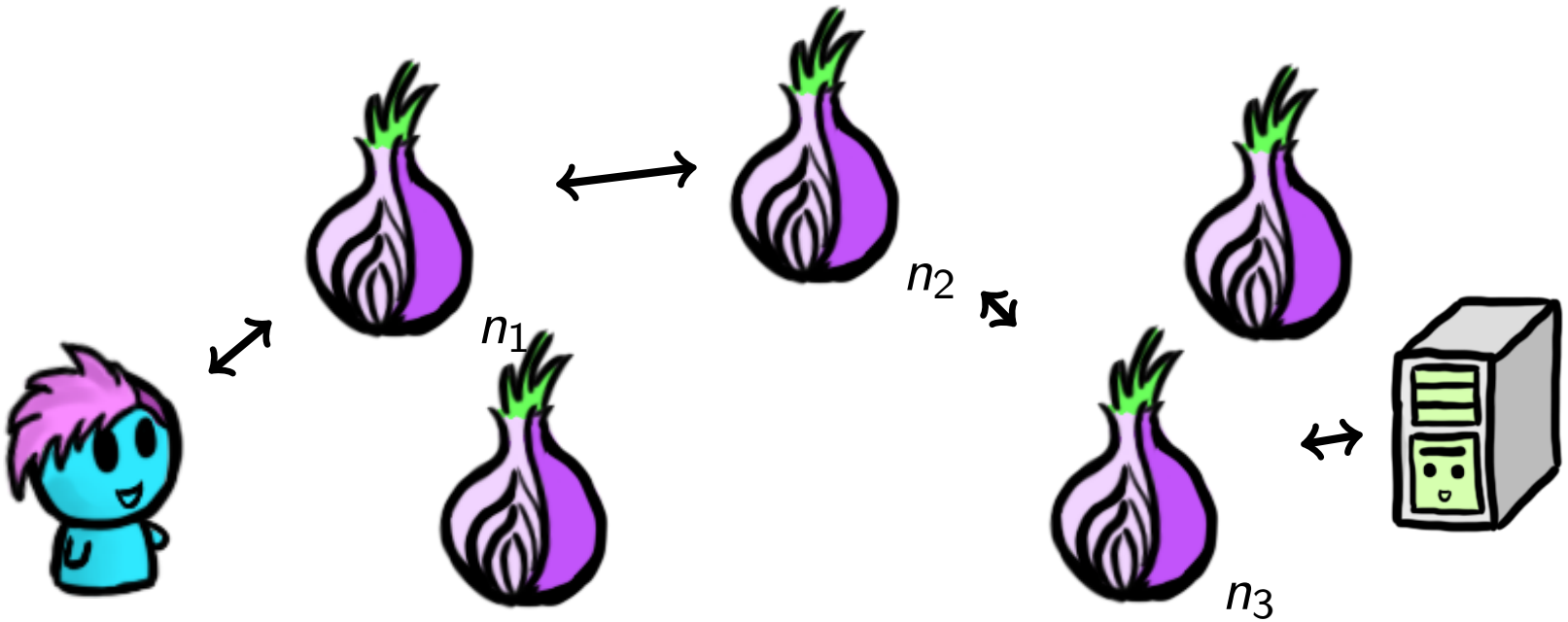
The server replies with R, sending it back to n3. The nodes encrypt the message back and Alice decrypts all the layers.



Who knows what?

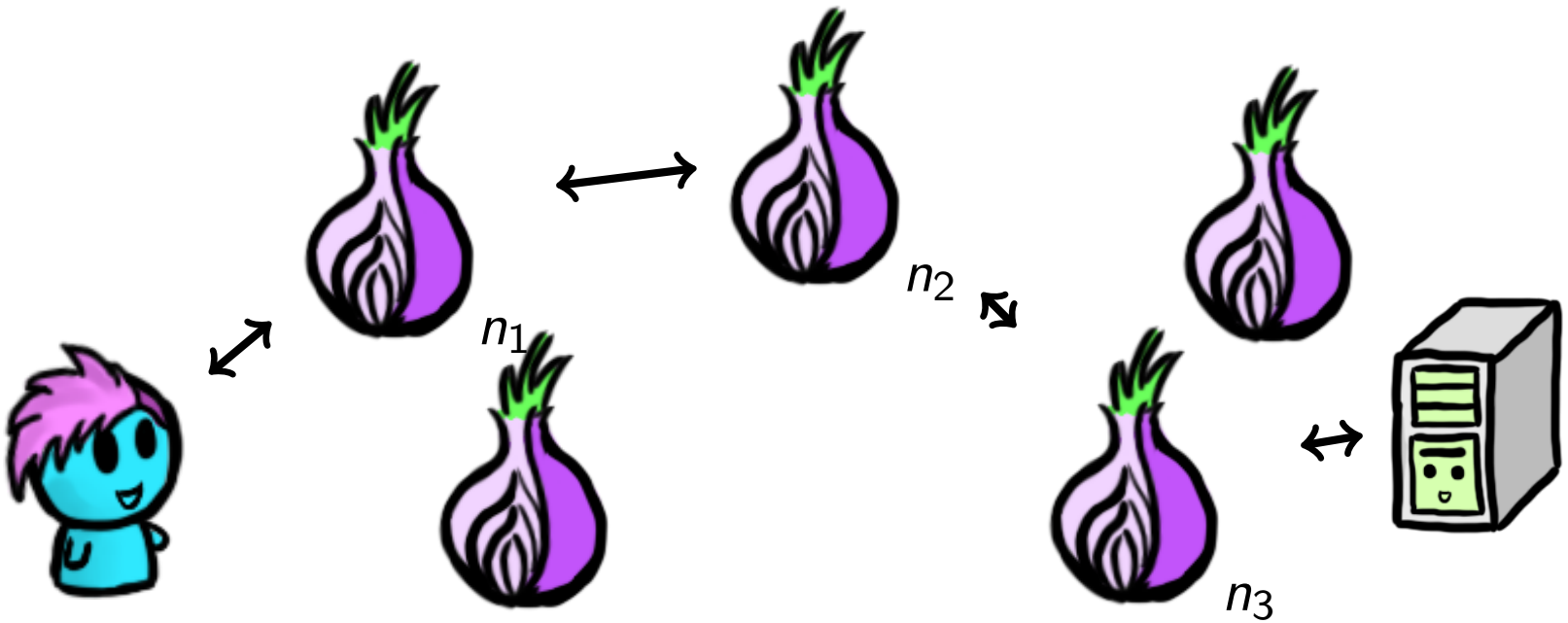


Who knows what?



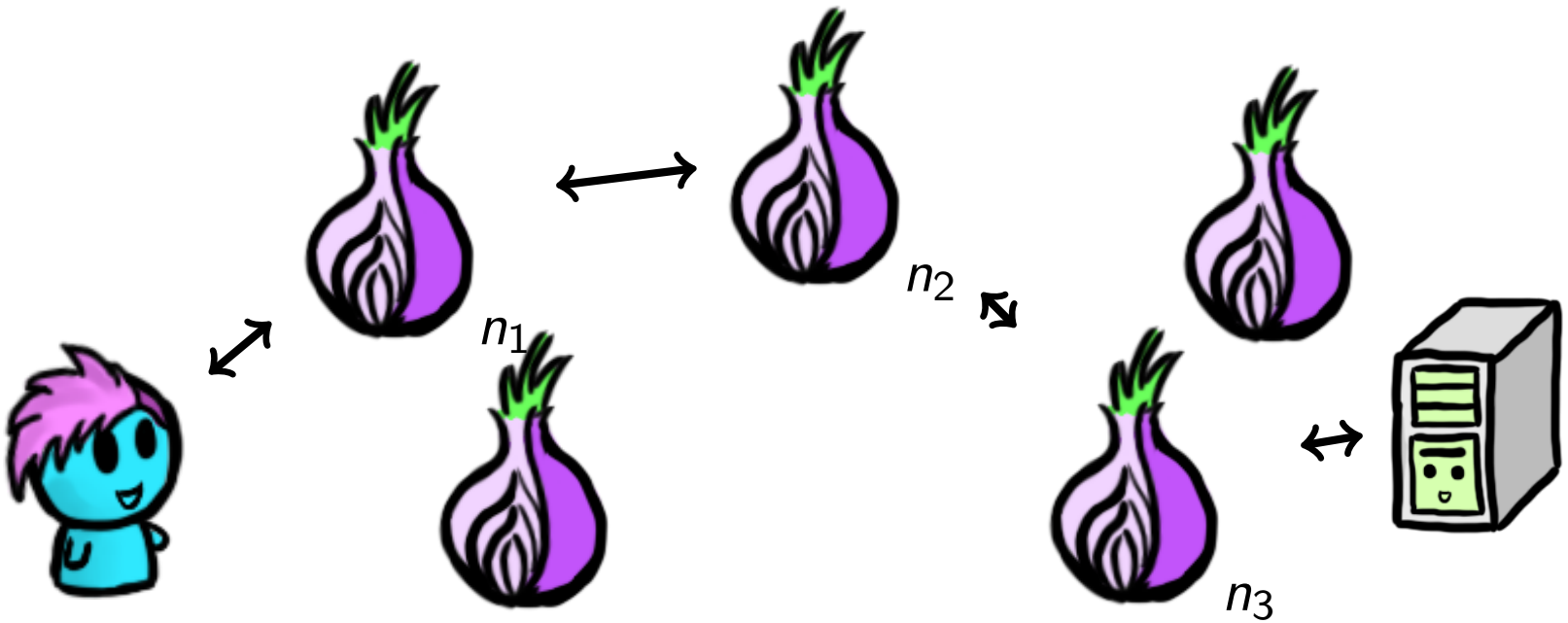
- Notice that node n1 knows that Alice is using Tor, and that her next node is n2, but does not know which website Alice is visiting
- Node n3 knows some Tor user (with previous node n2) is visiting a particular website, but doesn't know who
- The website itself only knows that it got a connection from Tor node n3

Adversaries



Q: Why must Alice choose all nodes, instead of letting each node pick the next one?

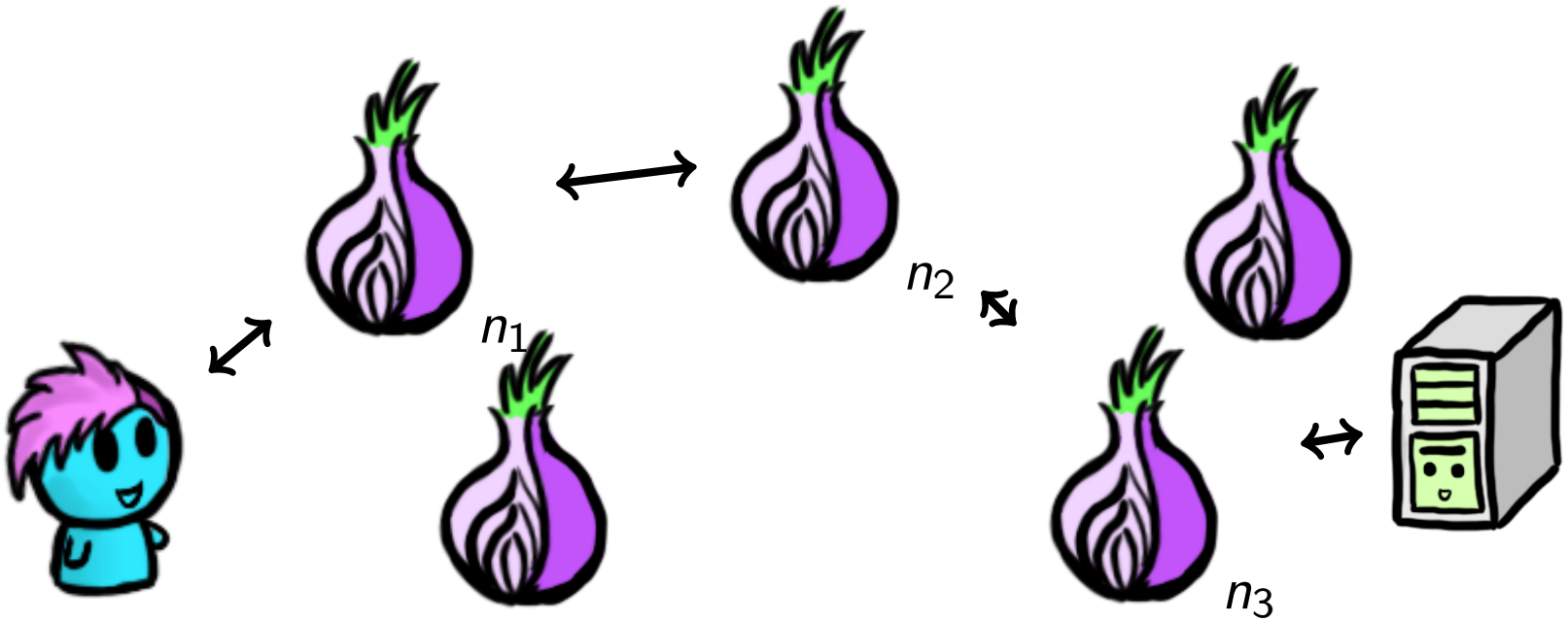
Adversaries



Q: Why must Alice choose all nodes, instead of letting each node pick the next one?

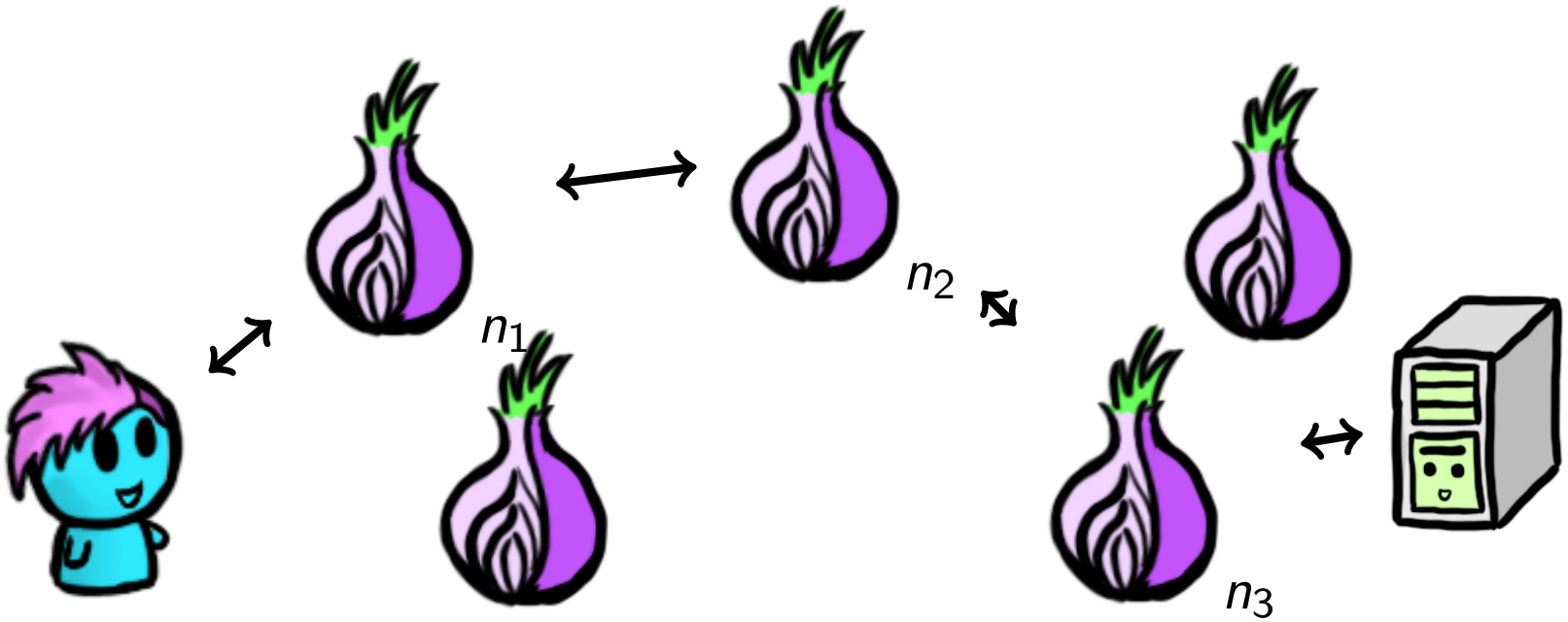
A: A malicious node would pick another malicious node. The user must have the ability to choose the nodes

Adversaries



Q: Why happens if Eve can inspect all network links? (a global passive adversary)

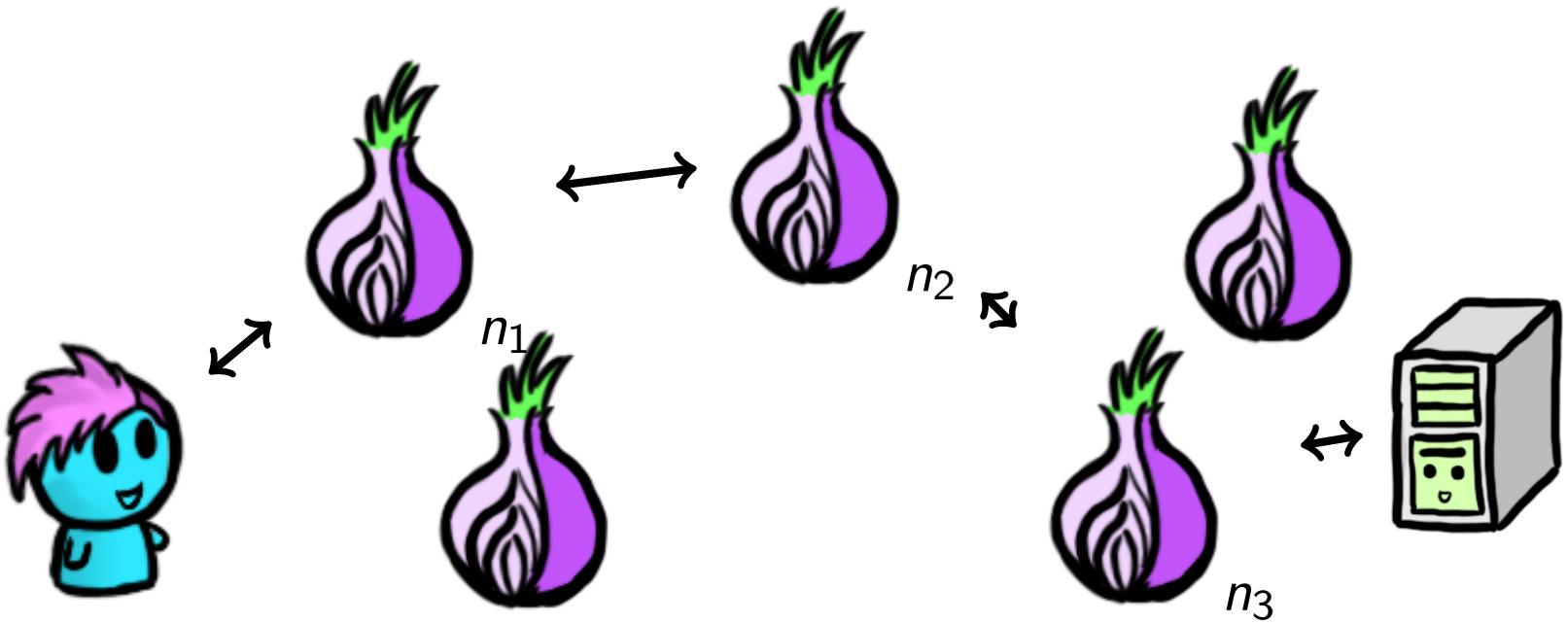
Adversaries



Q: Why happens if Eve can inspect all network links? (a global passive adversary)

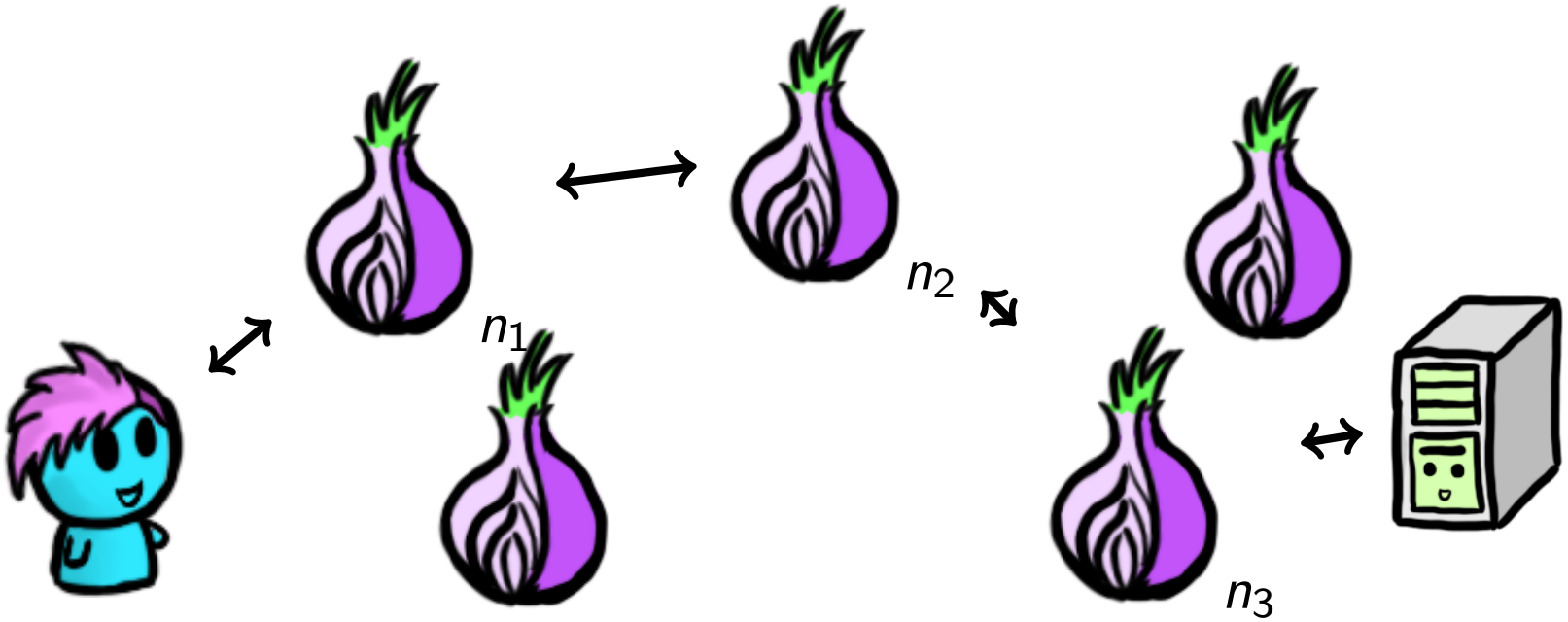
A: Tor does not protect against a global passive adversary. The adversary could de-anonymize Alice.

Adversaries



Q: What happens when Eve can inspect the incoming and outgoing traffic of a *single* node?

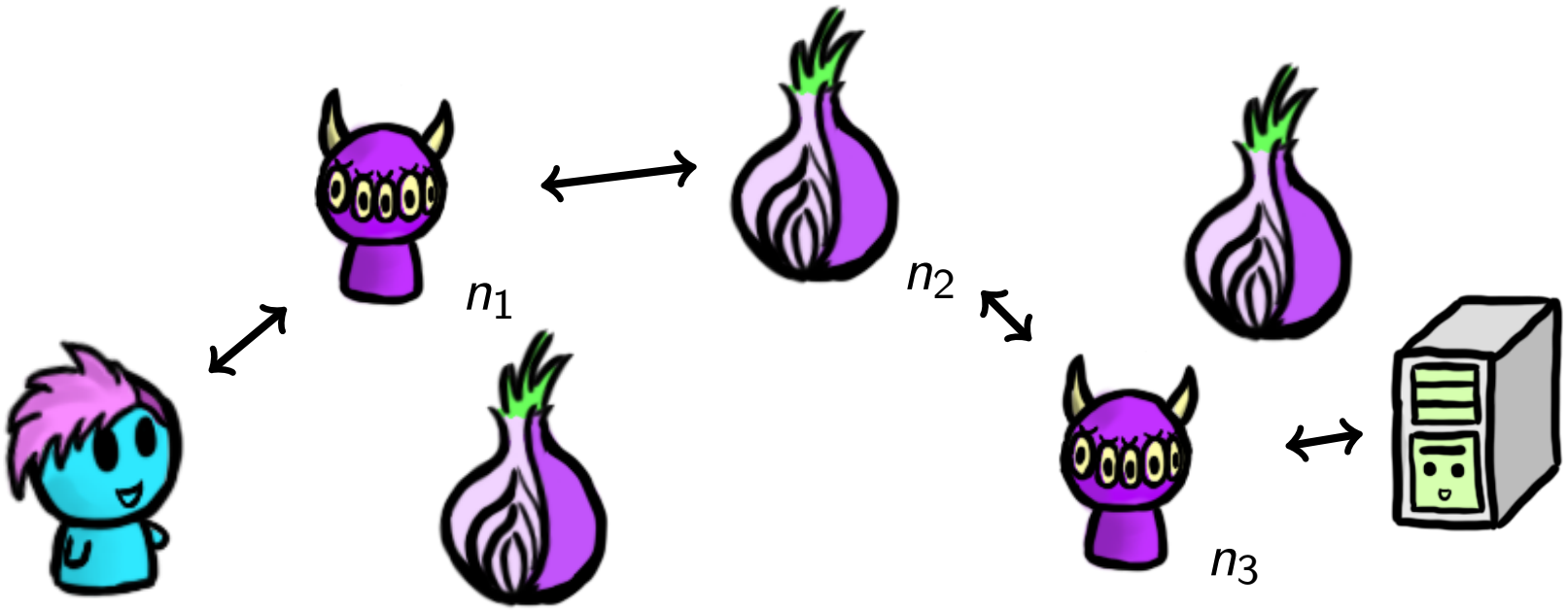
Adversaries



Q: What happens when Eve can inspect the incoming and outgoing traffic of a *single* node?

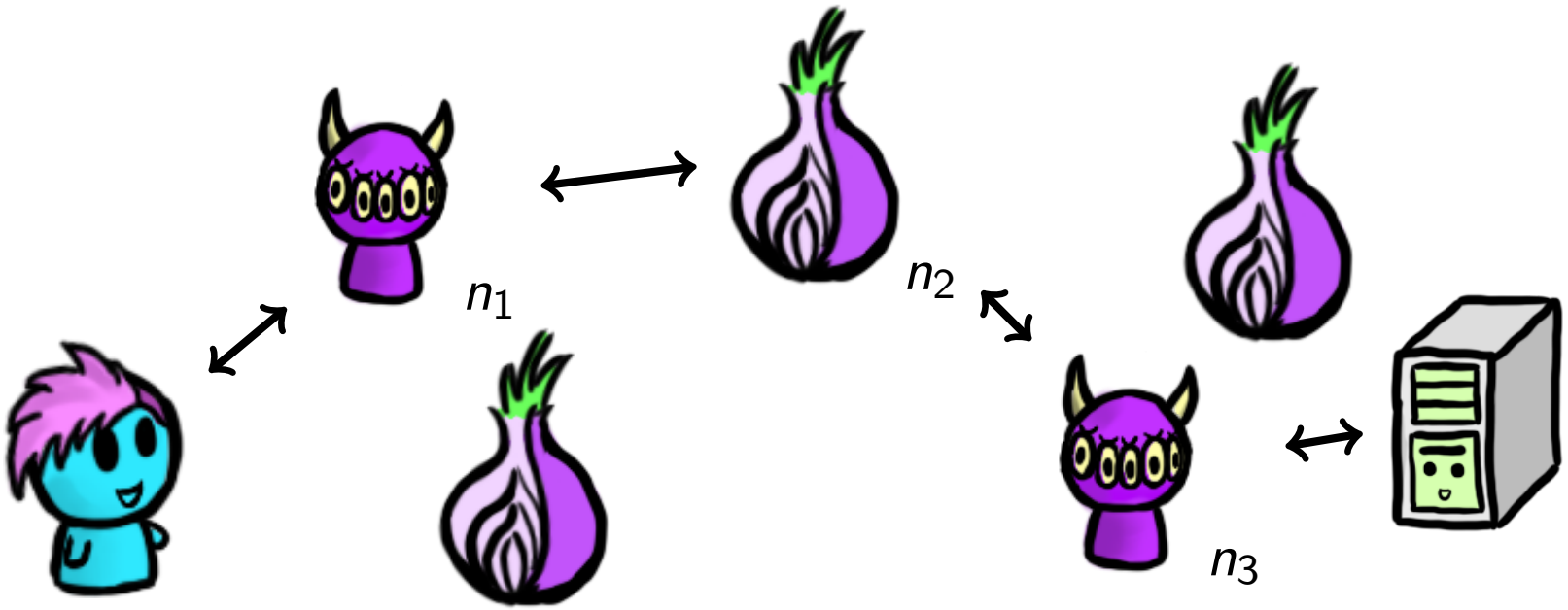
A: Alice is probably good

Adversaries



Q: What happens when Eve can inspect the incoming and outgoing traffic of the first and last nodes?

Adversaries



Q: What happens when Eve can inspect the incoming and outgoing traffic of the first and last nodes?

A: Traffic correlation attacks can easily de-anonymize Alice

Anonymity vs. pseudonymity

Tor provides for anonymity in TCP connections over the Internet, both unlinkably (long-term) and linkably (short-term)

What does this mean?

- There's no long-term identifier for a Tor user
- If a web server gets a connection from Tor today, and another one tomorrow, it won't be able to tell whether those are from the same person
- But two connections in quick succession from the same Tor node are more likely to come from the same person

Outline

Motivation

Simple scenario:

- Netflix stores its' movies in a database
  1. The Shawshank Redemption
  2. The Godfather
  3. The Dark Knight
  4. 12 Angry Men
  5. ...

- You request movies by index, say 1, 4, 2, ...

Motivation

Simple scenario:

- Netflix stores its' movies in a database
  1. The Shawshank Redemption
  2. The Godfather
  3. The Dark Knight
  4. 12 Angry Men
  5. ...
- You request movies by index, say 1, 4, 2, ...
- Netflix caches your selection and gradually builds a profile on your movie preferences

Motivation

Simple scenario:

- Netflix stores its' movies in a database
  1. The Shawshank Redemption
  2. The Godfather
  3. The Dark Knight
  4. 12 Angry Men
  5. ...
- You request movies by index, say 1, 4, 2, ...
- Netflix caches your selection and gradually builds a profile on your movie preferences
- But why? You has bought a Netflix license and so you should be able to access different movies

Definition

Goal: allow a user to query a database while hiding the identity of the data-items the user is after

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Non-private protocol

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Protocol:

- User: show me i
- Server: here is X_i

Non-private protocol

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Protocol:

- User: show me i
- Server: here is X_i

Analysis:

Q: Privacy and Bandwidth?

Non-private protocol

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Protocol:

- User: show me i
- Server: here is X_i

Analysis:

Q: Privacy and Bandwidth?

A: No privacy!
But very efficient, since we just receive 1 bit

Trivially-private protocol

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Protocol:

- User: show me *ALL*
- Server: here is $\{X_1, X_2, ..., X_n\}$

Trivially-private protocol

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Protocol:

- User: show me *ALL*
- Server: here is $\{X_1, X_2, ..., X_n\}$

Analysis:

Q: Privacy and Bandwidth?

A: Total privacy!
But very inefficient as we need all n bits transimitted

Trivially-private protocol

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Protocol:

- User: show me *ALL*
- Server: here is $\{X_1, X_2, ..., X_n\}$

Analysis:

Q: Privacy and Bandwidth?

A: Total privacy!
But very inefficient as we need all n bits transimitted

Sad news: if the server has unlimited computational power AND there is only a single copy of the database,
$\Longrightarrow$ n bits must be transferred!

“More” solutions?

"More" solutions?

- User asks for additional random indices
  - Drawback: balance information leak vs communication cost

"More" solutions?

- User asks for additional random indices
  - Drawback: balance information leak vs communication cost
- Anonymity
  - Note: this is in fact a different concern: it hides the identity of a user, not the fact that X_i is retrieved

Here is one protocol

Assume: two non-colluding servers with a copy of the dataset

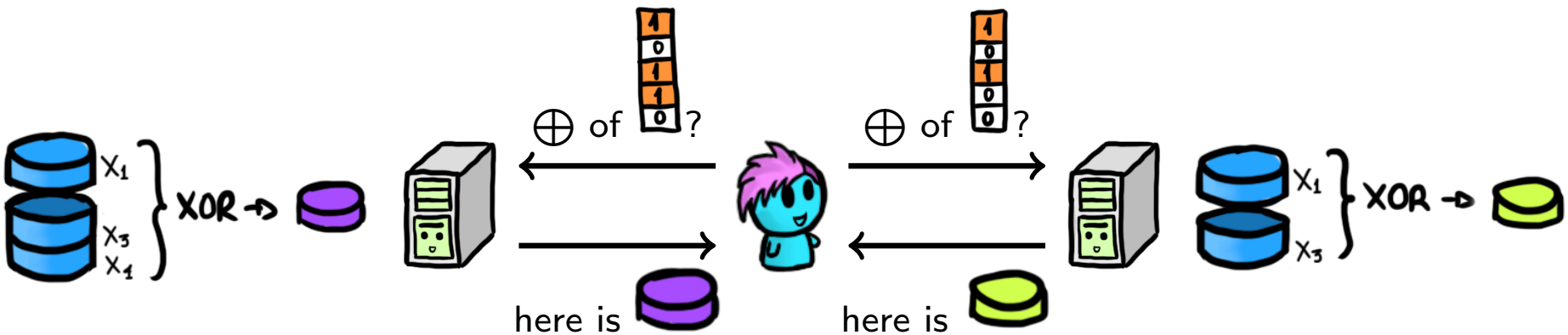
1. Alice wants X_4: she generates a random n-length binary vector and XOR's it with e_4 (where e_4 is an all-zero vector with it's $4th$ entry set to 1).

Here is one protocol

Assume: two non-colluding servers with a copy of the dataset

1. Alice wants X_4: she generates a random n-length binary vector and XOR's it with e_4 (where e_4 is an all-zero vector with it's $4th$ entry set to 1).

2. Alice queries the servers for XORs of entries:

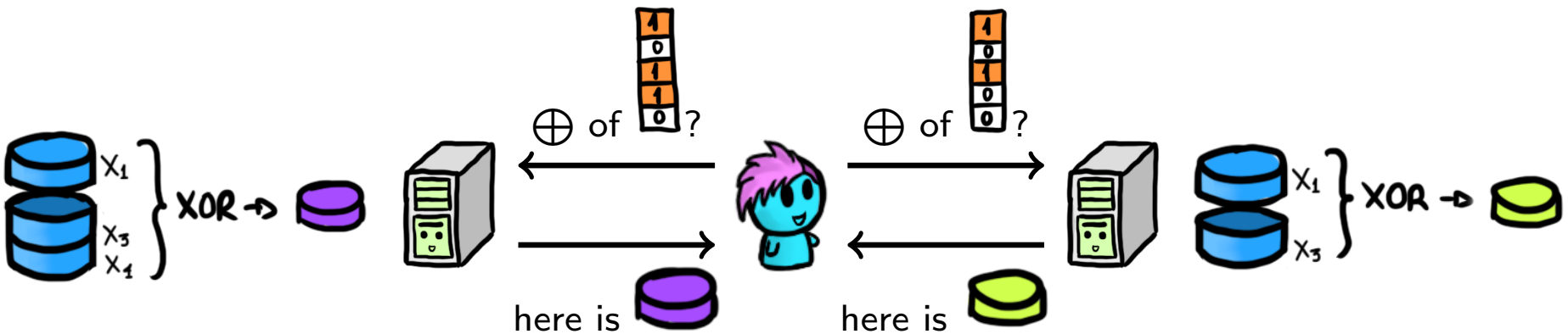


Here is one protocol

Assume: two non-colluding servers with a copy of the dataset

1. Alice wants X_4: she generates a random n-length binary vector and XOR's it with e_4 (where e_4 is an all-zero vector with it's $4th$ entry set to 1).

2. Alice queries the servers for XORs of entries:



3. Alice recovers the desired element:

Information-theoretic PIR (IT-PIR)

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Assumption: multiple (≥ 2) non-colluding servers

An example 2-server IT-PIR protocol:

- User $\rightarrow$ Server 1: $Q_1 \subset_R \{1, 2, ..., n\} \wedge i \notin Q_1$
- Server 1 $\rightarrow$ User: $R_1 = \bigoplus_{k \in Q_1} X_k$
- User $\rightarrow$ Server 2: $Q_2 = Q_1 \cup \{i\}$
- Server 2 $\rightarrow$ User: $R_2 = \bigoplus_{k \in Q_2} X_k$
- User derive $X_i = R_1 \oplus R_2$

Information-theoretic PIR (IT-PIR)

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Assumption: multiple (≥ 2) non-colluding servers

An example 2-server IT-PIR protocol:

- User $\rightarrow$ Server 1: $Q_1 \subset_R \{1, 2, ..., n\} \wedge i \notin Q_1$
- Server 1 $\rightarrow$ User: $R_1 = \bigoplus_{k \in Q_1} X_k$
- User $\rightarrow$ Server 2: $Q_2 = Q_1 \cup \{i\}$
- Server 2 $\rightarrow$ User: $R_2 = \bigoplus_{k \in Q_2} X_k$
- User derive $X_i = R_1 \oplus R_2$

Analysis:

- Probabilistic-based privacy $(1/|Q_2|)$
- # of bits: 1 ($\times$ 2 servers) + inexpensive computation

Computational PIR

Formal model:

- Server: holds an n-bit string $\{X_1, X_2, ..., X_n\}$
- User: wishes to retrieve X_i AND keep i private

Assumption: A single server with limited computational power

An example CPIR protocol:

- User chooses a large random number m
- User generates $n-1$ random quadratic residue (QR) mod m: $a_1, a_2, ..., a_{i-1}, a_{i+1}, ..., a_n$
- User generates a quadratic non-residue (QNR) mod m: b_i
- User $\rightarrow$ Server: $a_1, a_2, ..., a_{i-1}, b_i, a_{i+1}, ..., a_n$
- Server cannot distinguish between QRs and QNRs mod m, i.e., the request is just a series of random numbers: $u_1, u_2, ..., u_n$
- Server $\rightarrow$ User: $R = u_1^{X_1} \cdot u_2^{X_2} \cdot ... \cdot u_n^{X_n}$
- If R is a QR mod m, $X_i = 0$, else (R is a QNR mod m) $X_i = 1$

Comparison of CPIR and IT-PIR

CPIR

- Possible with a single server
- Server needs to perform intensive computations
- To break it, the server needs to solve a hard problem

IT-PIR

- Only possible with > 1 server.
- Server may need lightweight computations only
- To break it, the server needs to collude with other servers