# Design of a blocking-resistant anonymity system

Roger Dingledine, Nick Mathewson
The Tor Project

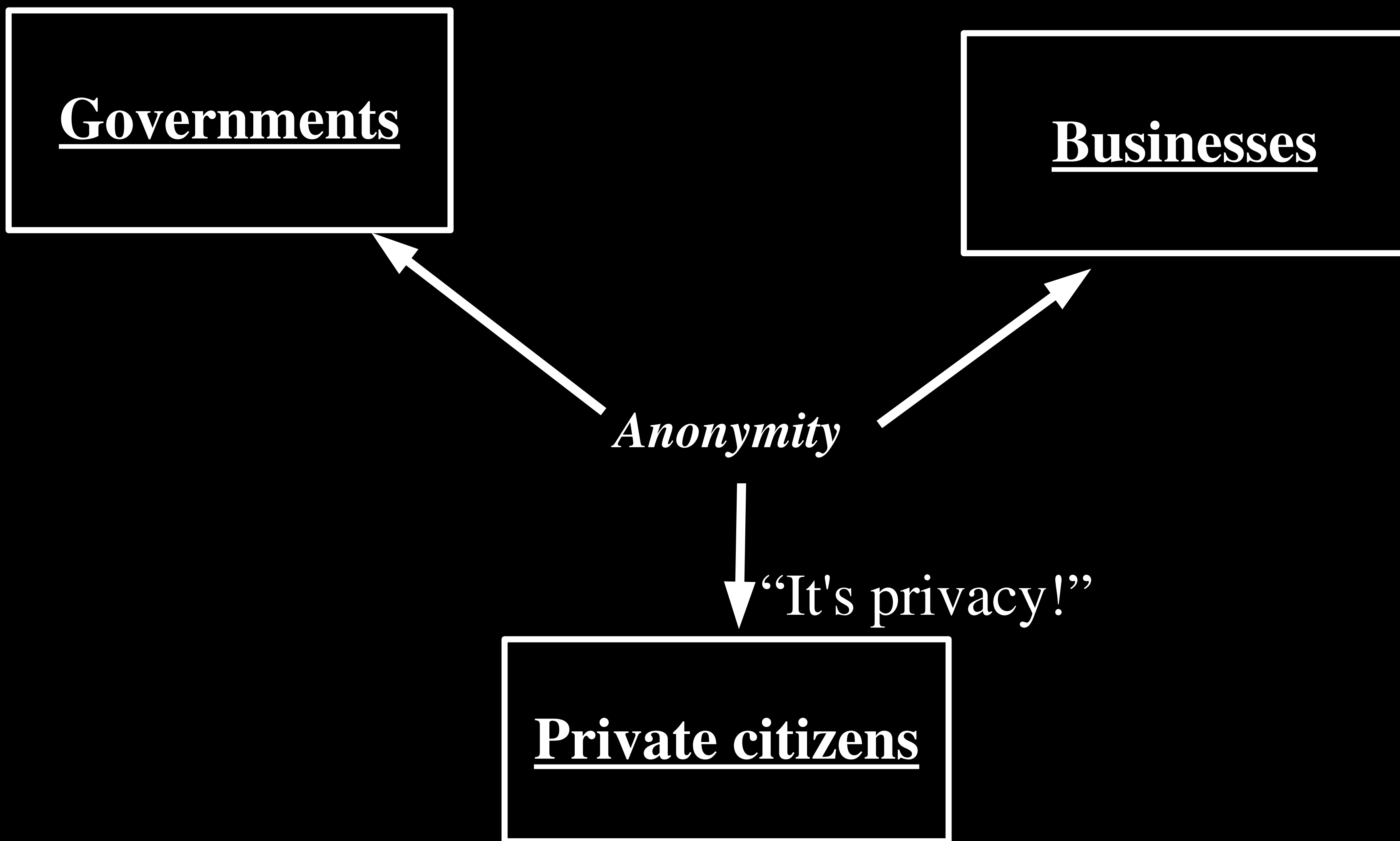
# Outline

- Crash course on Tor
- Goals for blocking resistance
- Assumptions (threat model)
- What Tor offers now
- Current proxy solutions
- What we need to add to Tor
- All the other issues that come up

# Tor: Big Picture

- Freely available (Open Source), unencumbered.
- Comes with a spec and full documentation: Dresden and Aachen implemented compatible Java Tor clients; researchers use it to study anonymity.
- Chosen as anonymity layer for EU PRIME project.
- 200000+ (?) active users.
- PC World magazine named Tor one of the Top 100 Products of 2005.

# Anonymity serves different interests for different user groups.



# Anonymity serves different interests for different user groups.

**Governments**

**Businesses**

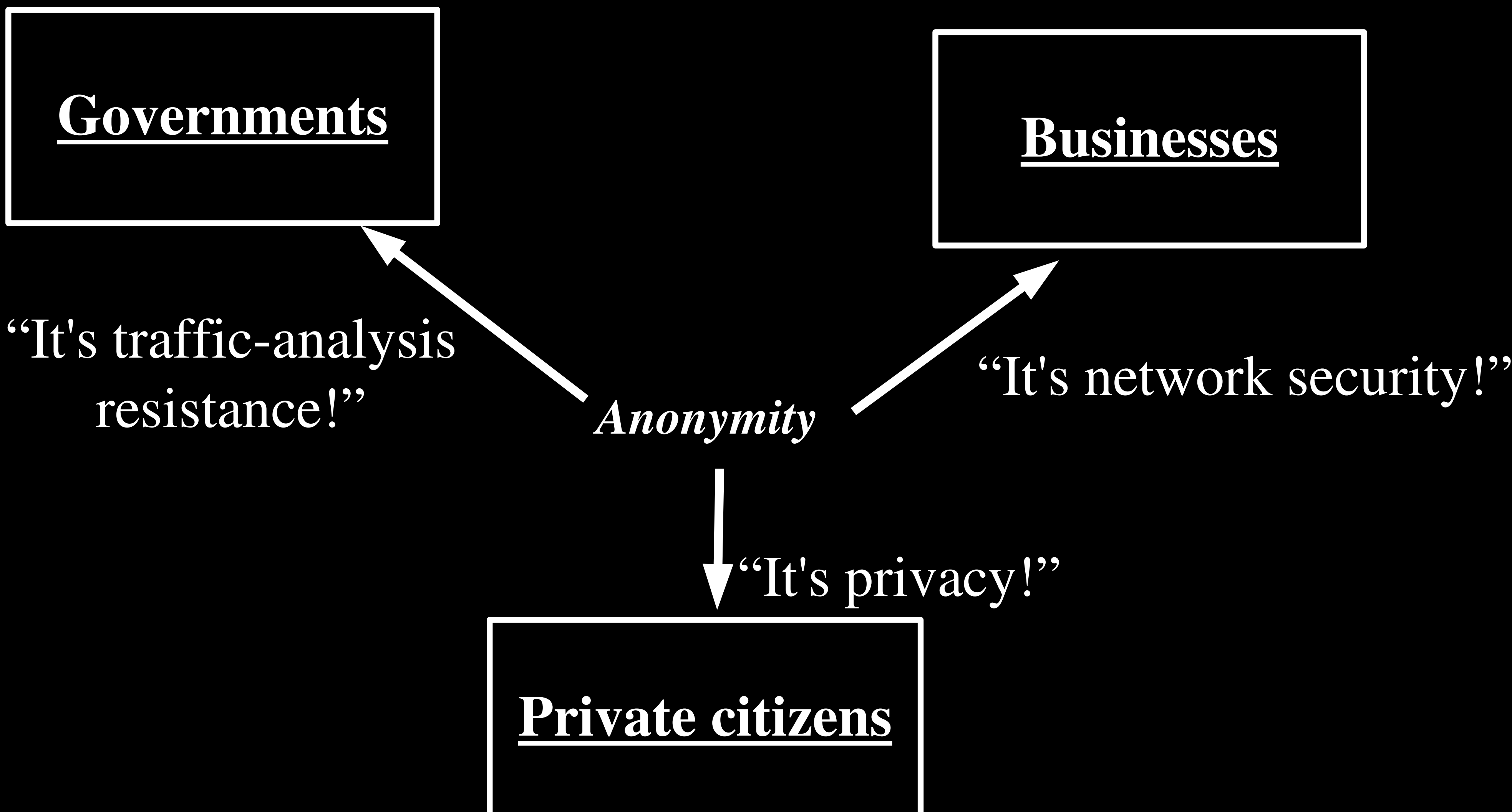
*Anonymity*

"It's network security!"

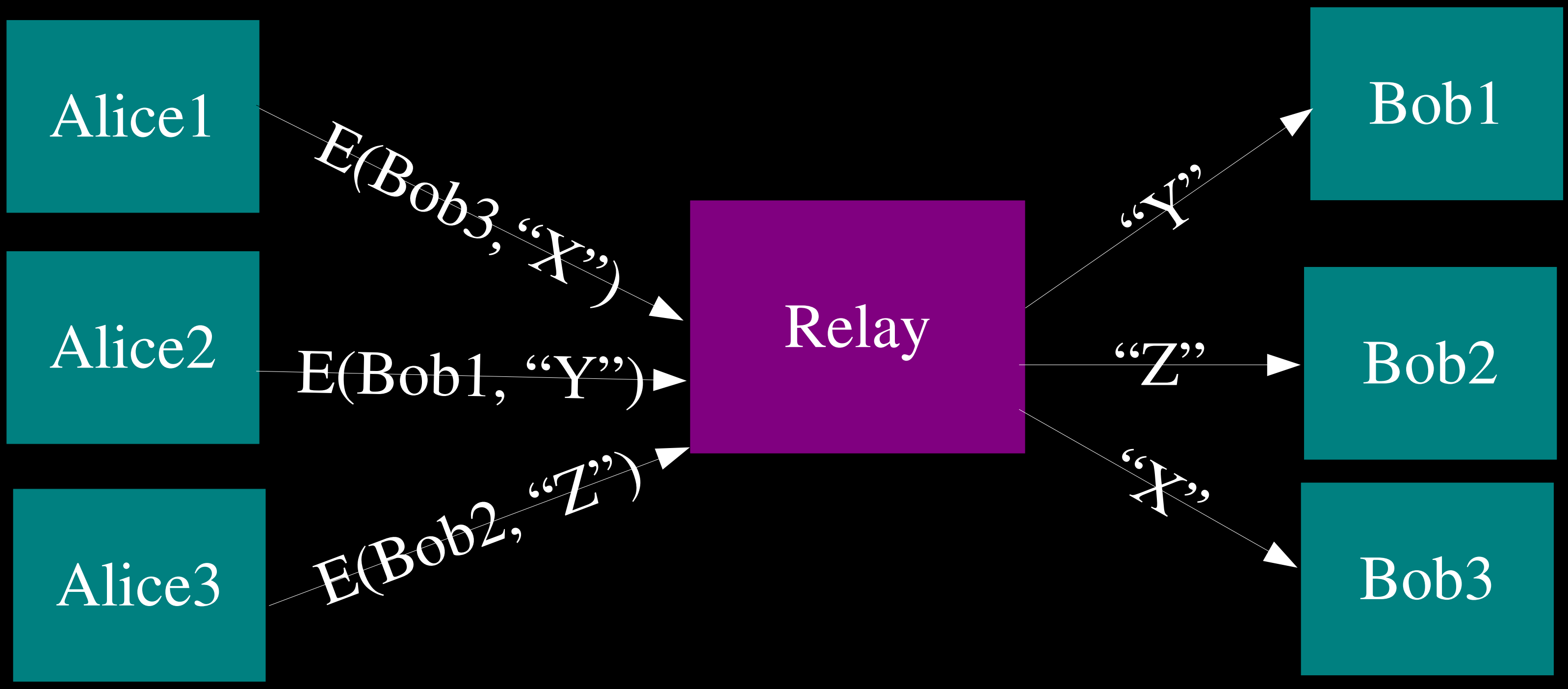
"It's privacy!"

**Private citizens**

# Anonymity serves different interests for different user groups.

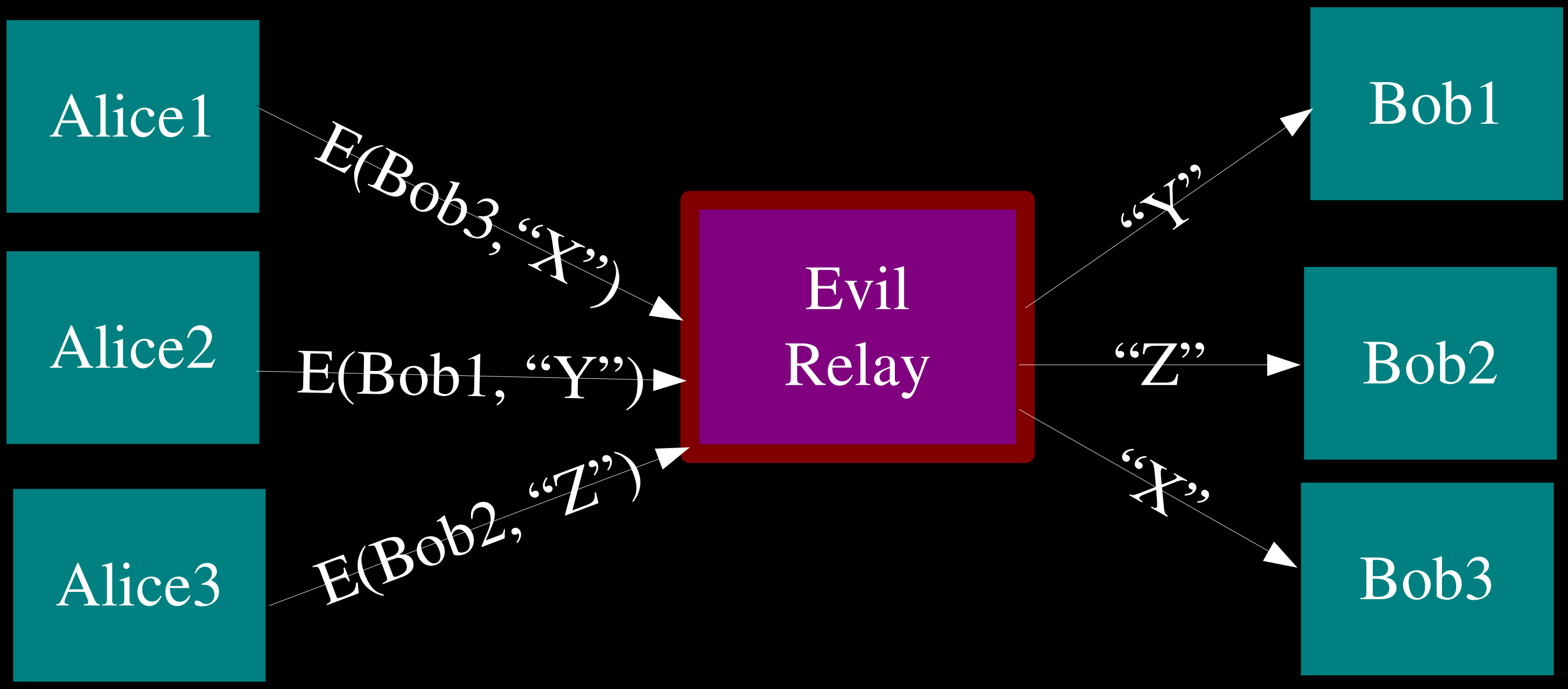


# The simplest designs use a single relay to hide connections.



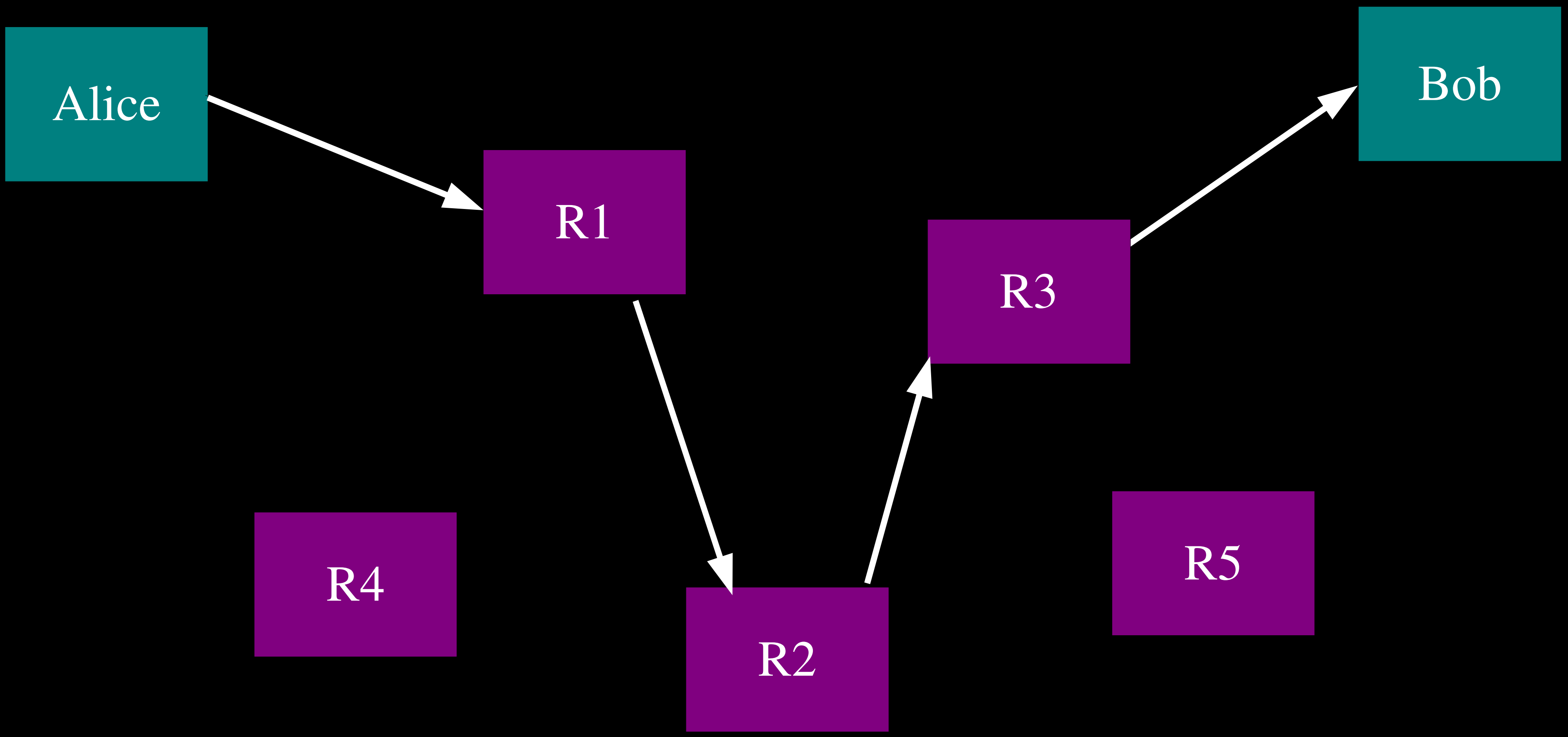

(example: some commercial proxy providers)

# But a single relay is a single point of failure.

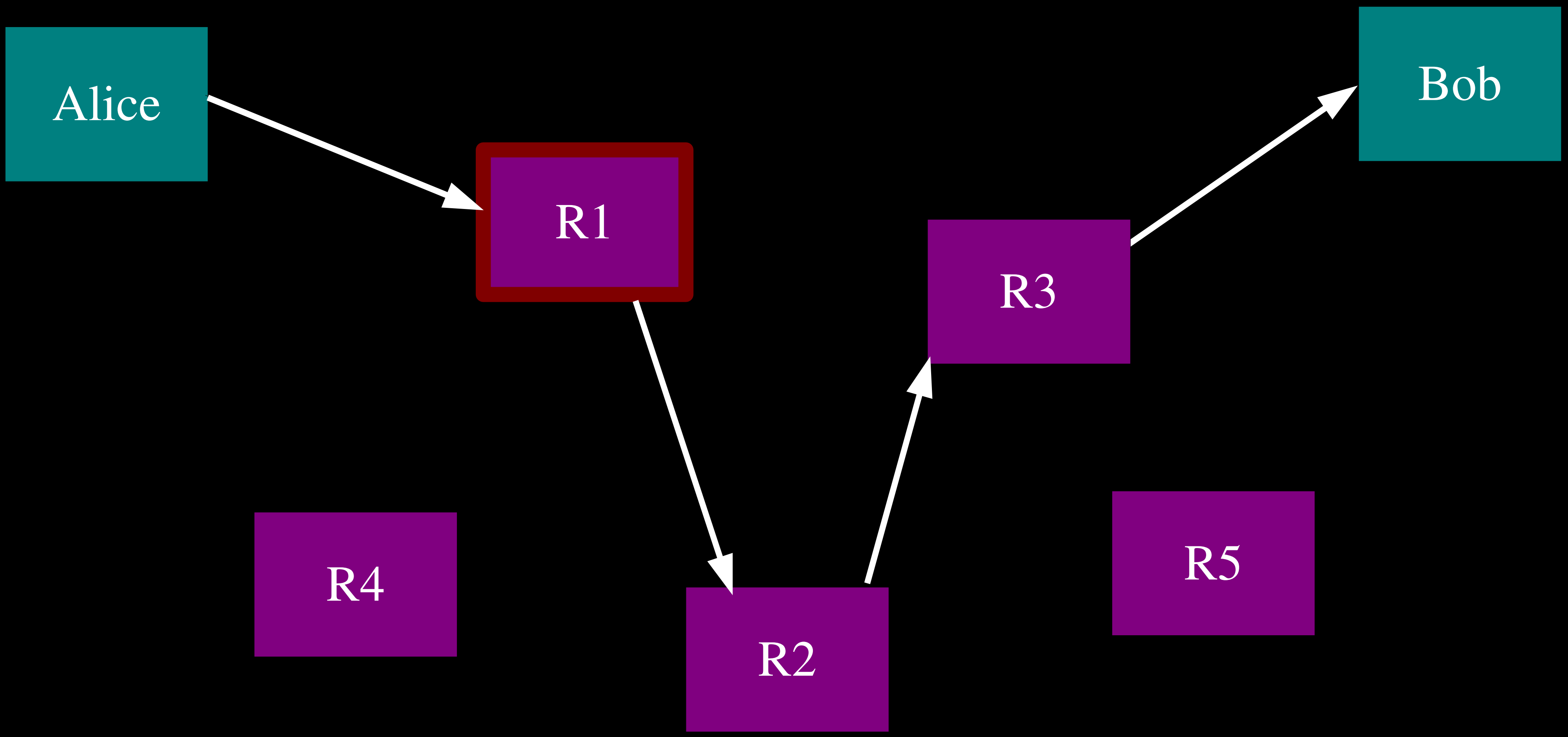



Eavesdropping on the relay works too.

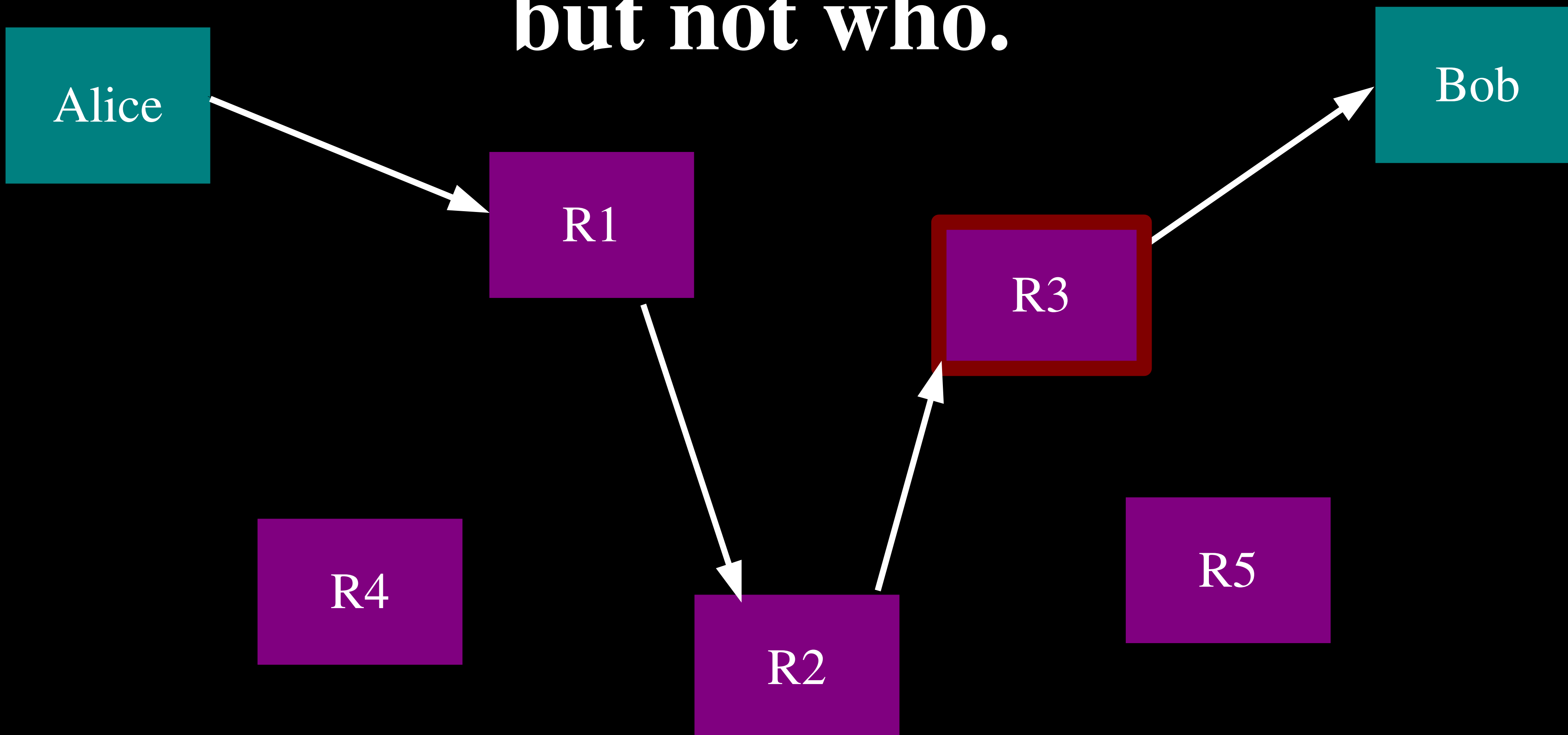
# So, add multiple relays so that no single one can betray Alice.



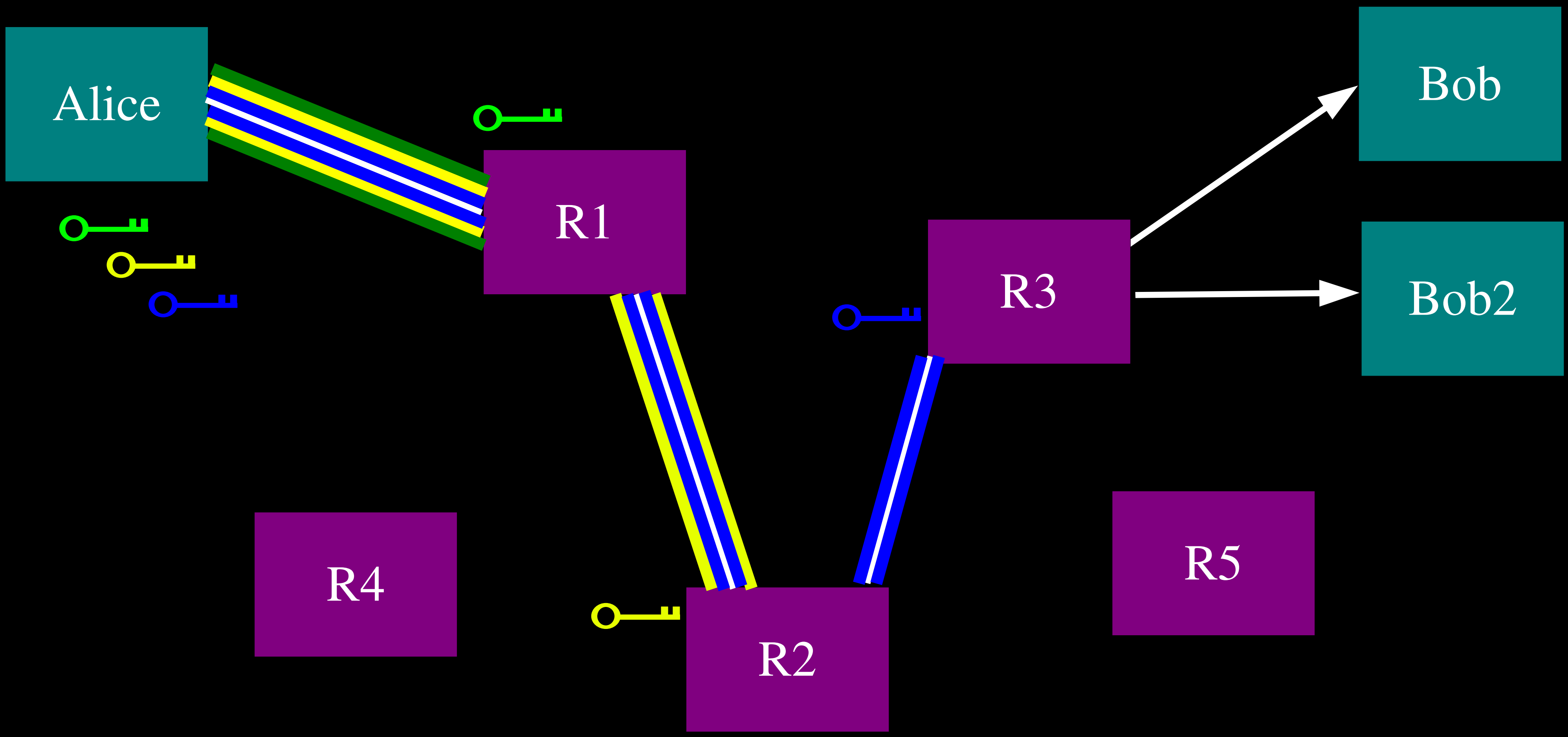
# A corrupt first hop can tell that Alice is talking, but not to whom.



# A corrupt final hop can tell that somebody is talking to Bob, but not who.



# Alice makes a session key with R1 ...And then tunnels to R2...and to R3



# Attackers can block users from connecting to the Tor network

- By blocking the directory authorities
- By blocking all the server IP addresses in the directory
- By filtering based on Tor's network fingerprint

# Goals

- Attract, and figure out how to use, more relay addresses
- Normalize Tor's network fingerprint
- Solve the discovery problem: how to find relay addresses safely
- Don't screw up our anonymity properties in the process

# Adversary assumptions aka Threat model

- Aim to defend against a strong attacker
  - so we inherit defense against weaker attackers
- Have a variety of users in mind
  - Citizens in China, Thailand, ...
  - Whistleblowers in corporate networks
  - Future oppressive situations
- Attackers will be in different stages of the arms race

# Attacker's goals (1)

- Restrict the flow of certain kinds of information
  - Embarrassing (rights violations, corruption)
  - Opposing (opposition movements, sites that organize protests)
- Chill behavior by *impression* that online activities are monitored

# Attacker's goals (2)

- Complete blocking is not a goal. It's not even necessary.
- Similarly, no need to shut down or block *every* circumvention tool. Just ones that are
  - popular and effective (the ones that work)
  - highly visible (make censors look bad to citizens -- and to bosses)

# Attacker's goals (3)

- Little reprisal against passive consumers of information.
  - Producers and distributors of information in greater danger.
- Censors (actually, govts) have economic, political, social incentives not to block the whole Internet.
  - But they don't mind collateral damage.

# Main network attacks

- Block by IP address / port at firewall
- Keyword searching in TCP packets
- Intercept DNS requests and give bogus responses or redirects

# Design assumptions (1)

- Network firewall has limited CPU and memory per connection
  - full steganography not needed, thankfully
- Time lag between attackers sharing notes
  - Most commonly by commercial providers of filtering tools
  - Insider threat not a worry initially

# Design assumptions (2)

- Censorship is not uniform even within each country, often due to different ISP policies
- Attacker can influence other countries and companies to help them censor or track users.

# Design assumptions (3)

- Assume the users aren't attacked by their hardware and software
  - No spyware installed, no cameras watching their screens, etc
- Assume the users can fetch a genuine copy of Tor: use GPG signatures, etc.

# Outline

- Goals
- Assumptions (threat model)
- ***What Tor offers now***
- Current proxy solutions
- What we need to add to Tor
- All the other issues that come up

# Tor gives three anonymity properties

- #1: A local network attacker can't learn, or influence, your destination
  - Clearly useful for blocking resistance
- #2: No single router can link you to your destination
  - The attacker can't sign up relays to trace users
- #3: The destination, or somebody watching it, can't learn your location
  - So they can't reveal you; or treat you differently.

# Other Tor design features (1)

- Well-analyzed, well-understood discovery mechanism: directory authorities.
- They automatically aggregate, test, and publish signed summaries of the available routers.
- Tor clients fetch these summaries to learn which routers have what properties.
- Directory information is cached throughout the Tor network.

# Other Tor design features (2)

- The list of dir authorities is not hard-wired.
- There are defaults, but you can easily specify your own to start using a different (or even overlapping!) Tor network.
- For example, somebody could run a separate Tor network in China.
- (But splitting up our users is bad for anonymity.)

# Other Tor design features (3)

- Tor automatically builds paths, and rebuilds and rotates them as needed.
- More broadly, Tor is just a tool to build paths given a set of routers.
- Harvard's “Blossom” project makes this flexibility more concrete:
  - It lets users view Internet resources from any point in the Blossom network.

# Other Tor design features (4)

- Tor separates the role of “internal relay” from the role of “exit relay”.
- Because we don't force all volunteers to play both roles, we end up with more relays.
- This increased diversity is what gives Tor users their anonymity.

# Other Tor design features (5)

- Tor is sustainable. It has a community of developers and volunteers.
- Commercial anonymity systems have flopped or constantly need more funding for bandwidth.
- Our sustainability is rooted in Tor's open design: clear documentation, modularity, and open source.

# Other Tor design features (6)

- Tor has an established user base of hundreds of thousands of people around the world.
- Ordinary citizens, activists, corporations, law enforcement, even govt and military users.
- This diversity contributes to sustainability.
- It also provides many many IP addresses!

# Anonymity is useful for censorship-resistance too!

- If a Chinese worker blogs about a problem at her factory, and she routes through her uncle's computer in Ohio to do it, ...?
- If any relay can expose dissident bloggers or compile profiles of user behavior, attacker should attack relays.
- ...Or just spread suspicion that they have, to chill users.

# Outline

- Goals
- Assumptions (threat model)
- What Tor offers now
- ***Current proxy solutions***
- What we need to add to Tor
- All the other issues that come up

# Relay versus Discovery

- There are two pieces to “proxying” schemes:
- a relay component: building circuits, sending traffic over them
- a discovery component: learning what routers are available

# Centrally-controller shared proxies

- Existing commercial anonymizers are based on a set of single-hop proxies.
- Typically characterized by two features:
  - They control and operate the proxies centrally.
  - Many different users get assigned to each proxy.
- Weak security compared to distributed-trust.
- But easier to deploy, and users don't need new software because they completely trust the proxy already.

# Independent personal proxies

- Circumventor, CGIProxy, Psiphon
- Same relay strategy, new discovery strategy: "Find a friend to install the relay for you."
- Great for blocking-resistance, but huge scalability question:
- How does the user in China find a volunteer in Ohio?
- How does the volunteer in Ohio find a user in China?

# Open proxies

- Google for “open proxy list”.
- Companies sell refined lists.
- Downsides:
  - Widely varying bandwidth, stability, reachability.
  - Legally questionable.
  - Not encrypted in most cases; keyword filtering still works.
  - “Too convenient” Are they run by the adversary?

# JAP and blocking-resistance

- Stefan Kopsell's paper from WPES 2004
- This is the idea that we started from in this blocking-resistance design.
- Uses the JAP anonymity network rather than Tor.
- Discovery is handled by making users solve a CAPTCHA in order to learn a relay address.

# Skype

- Port switching and encryption avoid the simple blocking and filtering attacks.
- Still has a central login server?

# ...and Tor itself

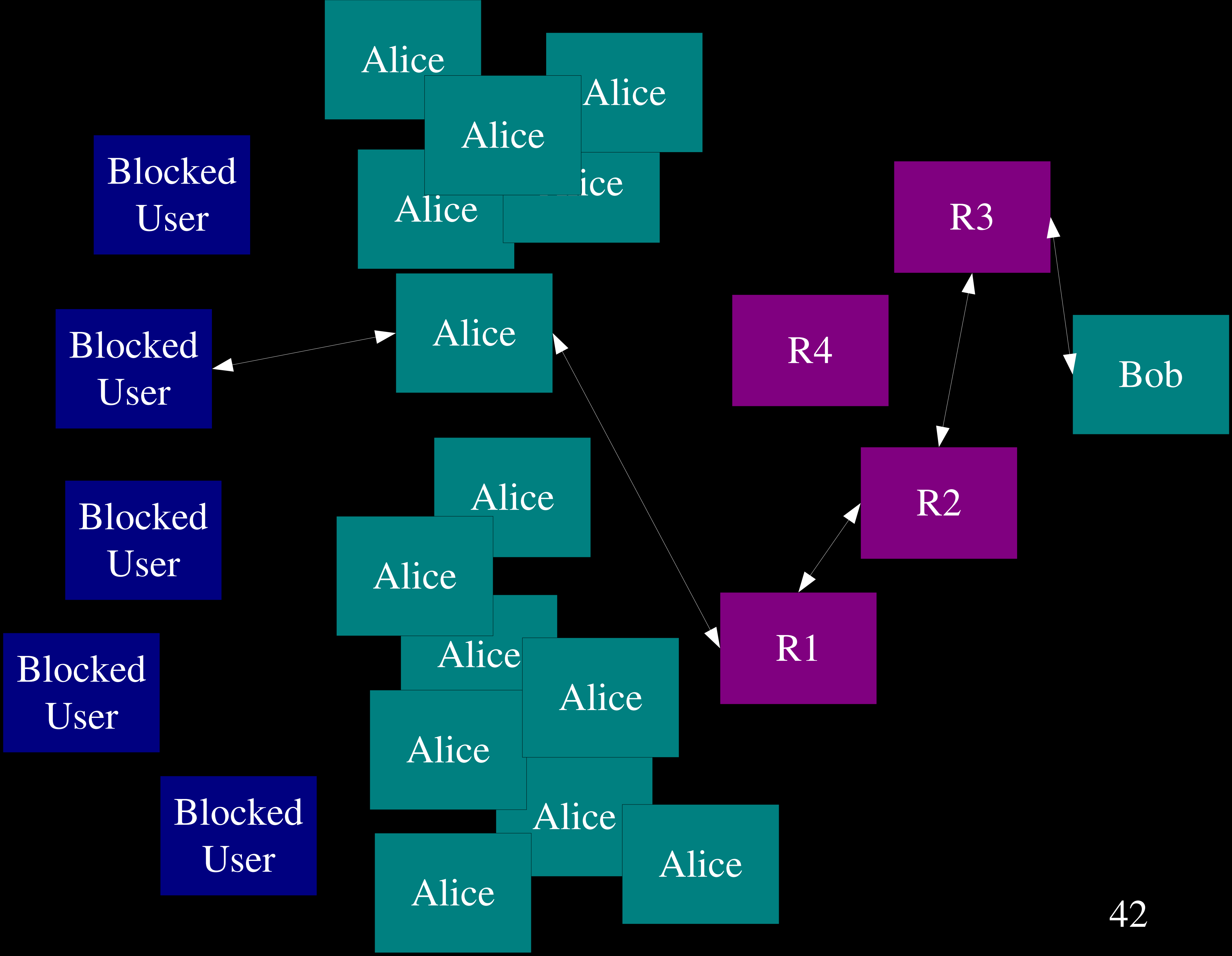
- Tor's website is blocked in many places, but not the Tor network. Why?
- Tens of thousands of users? “Nobody cares.”
- Perception: “Tor is for experts.”
- We haven't publicly threatened their control: “Tor is for civil liberties in free countries.”
- Realize that we're *already* in the arms race. These constraints teach us about priorities and capabilities of our various attackers.

# Outline

- Goals
- Assumptions (threat model)
- What Tor offers now
- Current proxy solutions
- ***What we need to add to Tor***
- All the other issues that come up

# Bridge relays

- Hundreds of thousands of Tor users, already self-selected for caring about privacy.
- Add a “Tor for Freedom” button to Vidalia (the most popular Tor GUI).
- Rate limit to 20KB/s?
- They can be internal relays, and don't have to be exit relays.


Alice
Alice
Alice
Alice
Blocked User
Blocked User
Blocked User
Blocked User
Blocked User
Alice
R3
R4
Bob
R2
Alice
Alice
Alice
Alice
R1
Alice
Alice
Alice
Alice

# Bridge directory authorities

- Specialized dir authorities that aggregate and track bridges, but don't provide a public list:
  - You can keep up-to-date about a bridge once you know its key, but can't just grab list of all bridges.
- Identity key and address for default bridge authorities ship with Tor.
- Bridges publish via Tor, in case somebody is monitoring the authority's network.

# One working bridge is enough

- Connect via that bridge to the bridge authority.
- ...and to the main Tor network.
- Remember, all of this happens in the background.
- “How to circumvent for all transactions (and trust the pages you get)”
  is now reduced to
  “How to learn about a working bridge”.

# Hiding Tor's network fingerprint

- [Skipping details for time]
- Get rid of plaintext HTTP (used by directories)
- Pick a good default port like 443.
- Make the TLS handshake look more like an ordinary HTTPS certificate exchange.
- Better understand timing and volume fingerprint attacks.

# Discovering working bridge relays

- Tor's modular design means we can separate the **relay** component from the **discovery** component.
- So we can use any discovery approach we like.
- Bridge relays change the problem from “How do I keep 1000 IP addresses public without letting the attacker learn them?”
- ...But alas, it's still going to be an arms race.

# Discovery: bootstrapping

- We assume users already have some way of bypassing the firewall to bootstrap.
- Open proxy servers, instant messaging, Skype, WoW, ...
- Or they know a friend who can.

# Independent bridges, no central discovery

- Like CGIProxy.
- Users could bootstrap by
  - knowing the bridge's operator, or
  - learning about the bridge from a local friend.
- “Telling a friend” has interesting incentives:
  - If he gets it blocked, you can't use it either now.
  - You're mapping your social network for the adversary.

# Families of bridges, no central discovery

- Volunteers run several bridges at once, or coordinate with other volunteers.
- The goal is that some bridges will be available at any given time.
- Each family has a bridge authority, to add new bridges to the pool, expire abandoned or blocked bridges, etc.
- Remember: this is all automated by the Tor client.

# Public bridges, central discovery

- What about bridges who don't know users?
  Or users who don't know any bridges?
- Divide bridges into pools based on identity key.
- Each pool corresponds to a *distribution strategy*. We start with eight strategies.
- Each strategy is designed to exercise a different scarce resource or property of the user.

# Distribution strategy #1

- Time-release bridge addresses.
- Divide available bridges into partitions, and each partition is deterministically available only in certain time windows.
- This pool will be first to get blocked, but
  - it will help to bootstrap until it *is* blocked
  - it won't be blocked by *every* adversary

# Distribution strategy #2

- Publish bridge addresses based on IP address of requester.
- Divide bridges into partitions, hash the requester's IP address, choose a random bridge from the appropriate partition.
- (Don't use entire IP address, just first 3 octets.)
- As a special case, treat all Tor exit IP addresses as being on the same network.

# Distribution strategy #3

- Combine time-based and location-based strategies.
- The bridge address provided in a given time slot is deterministic within the partition, rather than chosen randomly each time.
- So later requests during that time slot from a given network are given the same bridge address as the first request.

# Distribution strategy #4

- Use Circumventor's “mailing list trick”.
- Start a mailing list, let people sign up, send out a few new bridge addresses every few days.
- The adversary will block them, but not immediately.
- Every three or four days seems to be sufficient for Circumventor for now.

# Distribution strategy #5

- Users provide an email address and we mail them a bridge address.
- Limit one response per email address?
- Require a CAPTCHA.
  - We can leverage Yahoo and Gmail CAPTCHAs!

# Distribution strategy #6

- Social network reputation system.
- Pick some seeds (trusted people in blocked areas) and give them a few dozen bridge addresses and a few “delegation tokens”.
- Run a database near the bridge authority; Tor clients log in to learn more bridge addresses.
- Users can delegate trust to other people by giving them a token, which can be exchanged for a new account in the database.

# Distribution strategy #6 (cont)

- Accounts “in good standing” then accrue new bridge addresses and new tokens.
- How do we decide we like an account? If the bridges it knows don't end up blocked.
- Could track reputation between accounts, or use blinded tokens to prevent even the database from mapping the social network.
- Gets really messy. Future work.

# Distribution strategies #7 and #8

- Held in reserve, in case all our tricks fail at once and we need to deploy new strategies quickly.
- Please come up with new strategies and tell us! For example, SMS messages?
  For example, proof-of-work by encrypting the bridge address with a 40-bit key?

# Deploying all solutions at once

- Finally, we're not in the position of defender: We only need one scheme that works!
- The attacker must guess how to allocate his resources between all the discovery strategies.
- By deploying all of them at once, we make *all* of them more likely to succeed.

# Outline

- Goals
- Assumptions (threat model)
- What Tor offers now
- Current proxy solutions
- What we need to add to Tor
- ***All the other issues that come up***

# How do we learn if a bridge has been blocked? (1)

- Active testing via users
  - Can use Blossom-like system to build circuits through them to test.
  - If we pick random users to test from, the adversary should sign up users.
  - Even if we have trusted users, adversary can still discover them and then monitor them.

# How do we learn if a bridge has been blocked? (2)

- Passive testing via bridges
  - Bridges install GeoIP database, periodically report countries and traffic load.
  - But: If we don't see activity from Burma, does that mean it's blocked, or they're just asleep?

# How do we learn if a bridge has been blocked? (3)

- Different zones of the Internet are blocked in different ways – not just one per country.
- Lots of different possible locations for the fault: at bridge, at user, in between?
- Attacker could poison our bridge DB by signing up already-blocked bridges.
- Eventual solution will probably involve a combination of active and passive testing.

# Using Tor in oppressed areas

- Common assumption: risk of using Tor increases as firewall gets more restrictive.
- But as firewall gets more restrictive, more ordinary people use Tor too, for more mainstream activities.
- So the “median” use becomes more acceptable?

# Trusting local hardware/software

- Internet cafes
- USB-based Tor package
- CD-based Tor package (LiveCD)

# How many bridges do you need to know about to stay connected?

- Cablemodem / DSL bridges will disappear or move periodically.
- Already a tough problem with natural churn, but they can also get blocked.
- Related: how often should users fetch updates?

# Cablemodems don't usually run big websites

- So the attacker can just block all connections to Comcast, Verizon, ...
- We need to get bridges on both “consumer” and “producer” addresses.
- Also have to worry about economic pressure, E.g. from China on Verizon.

# Publicity attracts attention

- Many circumvention tools launch with huge media splashes. (The media loves this.)
- But publicity attracts attention of the censors.
- We threaten their *appearance* of control, so they must respond.
- We can control the pace of the arms race.

# Next steps

- Technical solutions won't solve the whole censorship problem. After all, firewalls are *socially* very successful in these countries.
- But a strong technical solution is still a critical puzzle piece.
- Next steps: deploy prototype bridges and bridge authorities, implement some discovery strategies, and get more intuition about what should come next.

# And Tor itself needs to survive

- Ongoing discussion around the world: is anonymity useful for the world?
- Data retention threatens privacy and safety, but won't catch the bad guys.
- We need your help!
  More Tor servers, more volunteers, more funding, ...