Lecture 13: Open Problems in Transport Security

G. SOGO - Fall 2021
MIT
c-g, Devadas, Kalai, Zeldovich

Plan

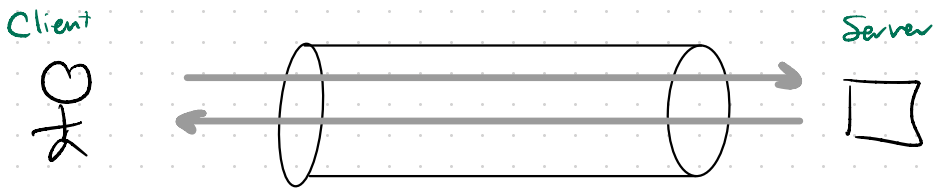
- Censorship circumvention
- Metadata-hiding messaging
- Metadata-hiding web browsing

Logistics

- Midterm 11/2 7:30-9:30pm in 50-340 (T/F, short answer, long ans)
- Practice problems out by Wed. Solns on Sunday.
- Lab 3 due Th 10/28

Recap: Encryption in practice

- We have encryption, we have authentication.
- Using TLS, we can get "encrypted & auth pipe"



As we discussed, TLS 1.3 satisfies eight (!) security properties.

↳ None of these implies anything about hiding the IP (network) address of the client/server.

↳ Attacker can potentially learn who you are talking to!

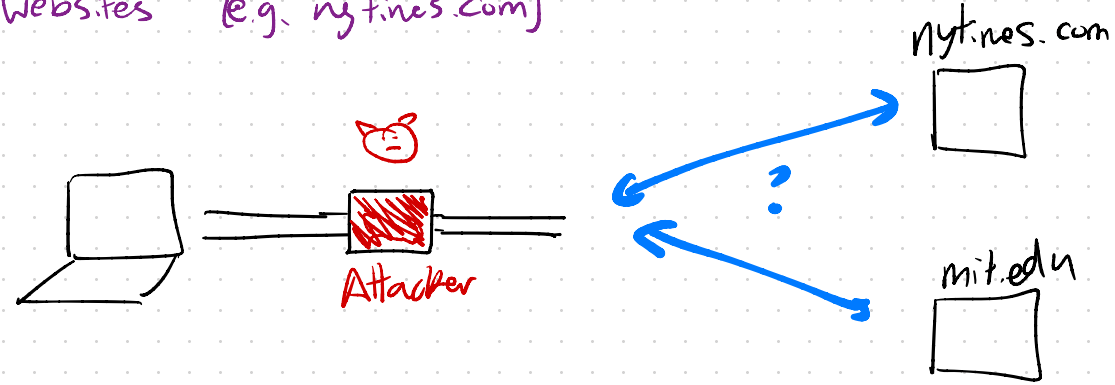
Two consequences:

1) Internet censorship
2) Mass surveillance

> There are many other open problems in comp sec & crypto — today will focus on metadata leakage.

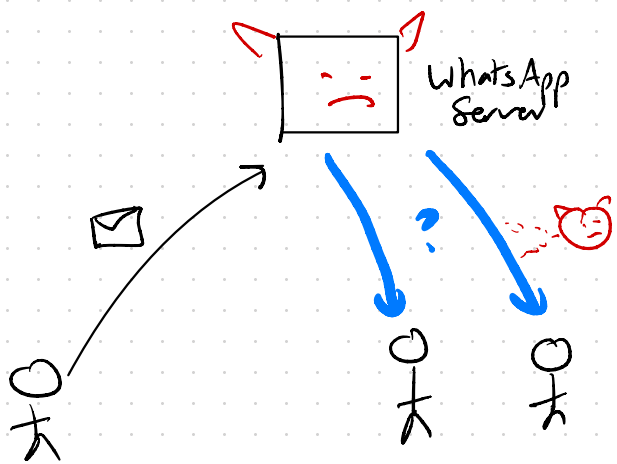
Setting 1: Web censorship

- Attacker controls a router (e.g. at national border)
- Attacker's goal: block access to certain websites (e.g. nytimes.com)



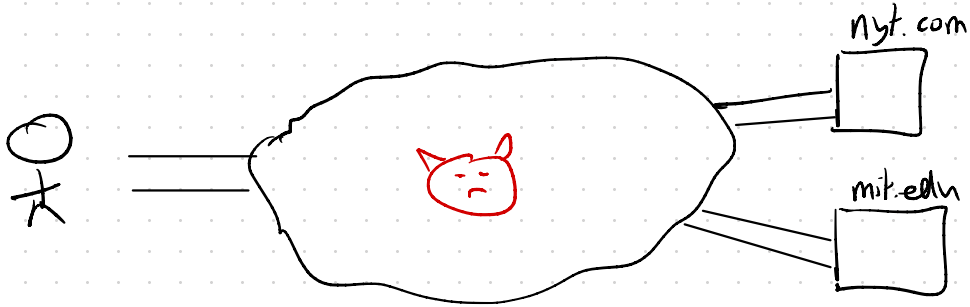
Setting 2: Metadata-hiding messaging

- Attacker controls network ("global adversary")
- Attacker may also control participants & servers
- Attacker's goal: figure out who talks to whom



Setting 3: Metadata-hiding web browsing

- Same as Setting #2 - just for web browsing.
- More challenging: more data, need to be backwards compatible, etc.



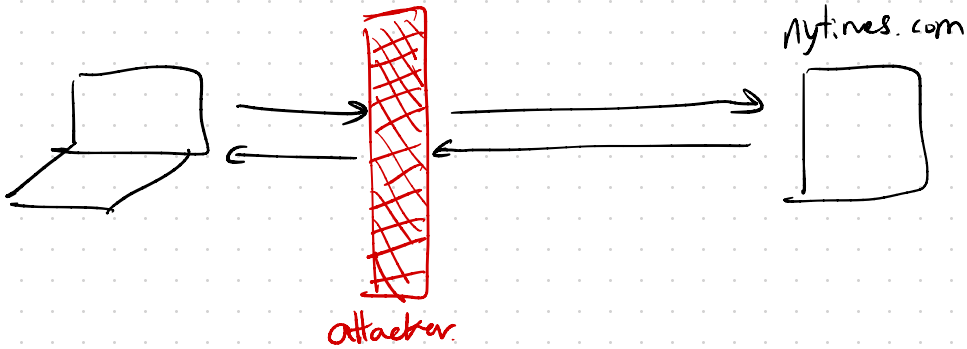
⇒ These really are <u>open</u> problems.

We won't have clean solutions.

<u>BUT</u>: A great area for research & further study.

Web Censorship

* Will talk about this first ~~bc~~ the problem/setting is simpler.
* Only gets messier from here...



Many countries aim to block access to certain websites in their borders.

* Armenia, Azerbaijan, Bangladesh, China, Cuba, Egypt, ..., Uzbekistan, Venezuela, Zimbabwe

↳ See "Freedom of the Internet" report from Freedom House

Attacker's goal: Block nytimes.com (e.g.) without blocking the entire Internet.

Usually (not always) too costly for govt to shut down all network access.

e.g Uganda Jan 2021 shutdown for ~4 days before elections.

Why is Internet censorship bad?

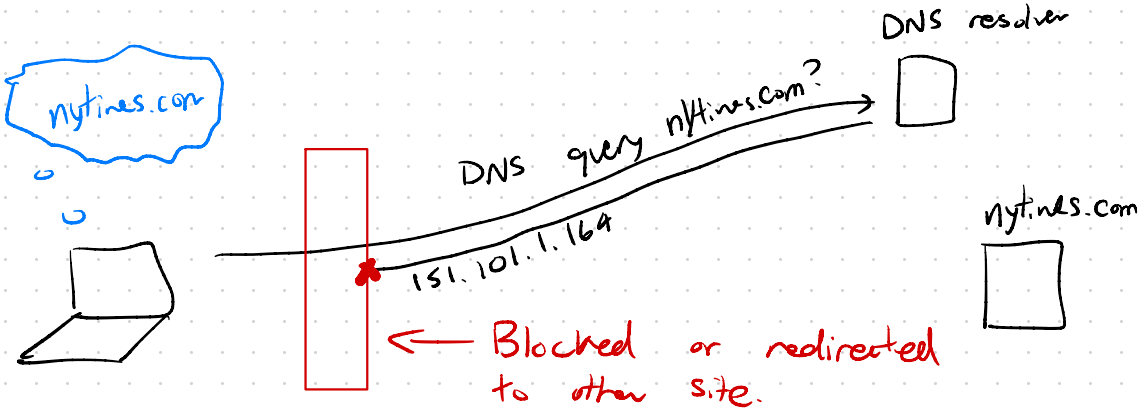
↳ Depends on who/where you are.

* K12 school?
* Country run by totalitarian govt?
* Corporate environment

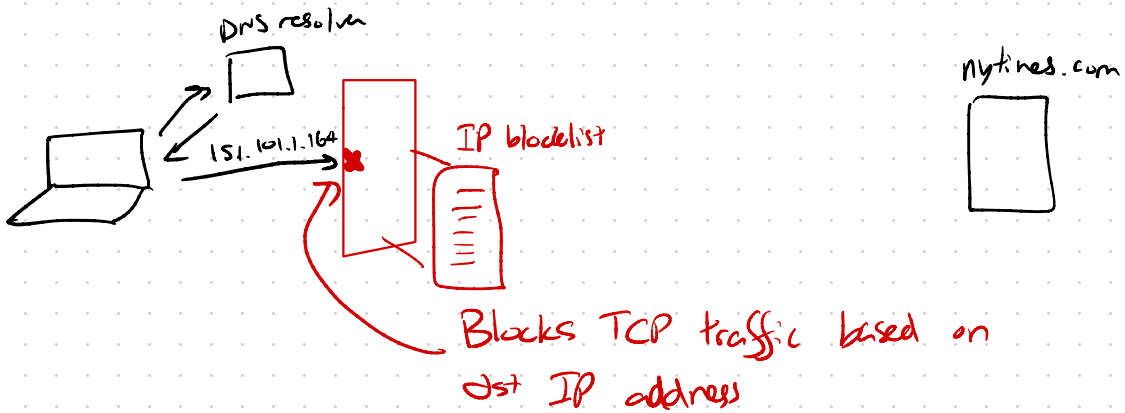
(Censorship tools for one environment often end up in another...)

Why doesn't TLS solve this problem?

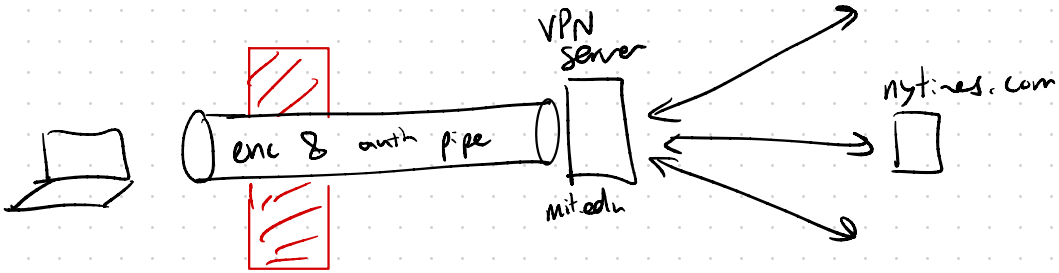
... before you even get to TLS.



... say you can get the DNS info locally (or via DNS-over-HTTPs, more commonly)



One Approach: Virtual private networks (VPNs)



↳ Attacker learns that client is using VPN via mit.edu.

↳ Can't easily learn which website client visits

Imperfect: Attacker can still learn a LOT via packet timings & sizes.

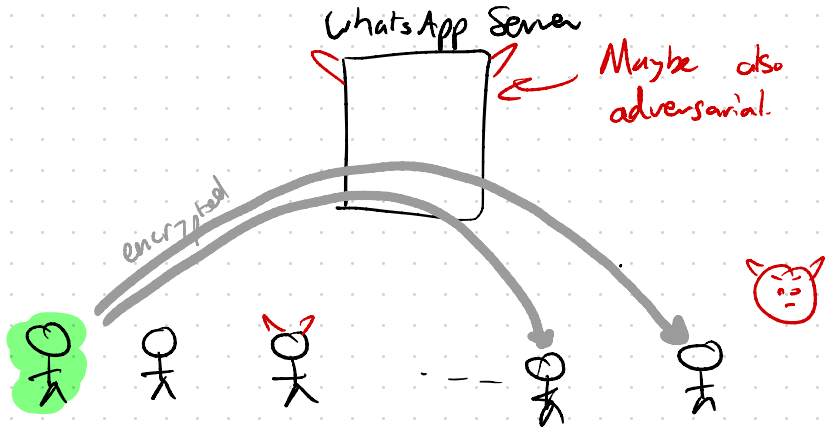
e.g. whether you are watching a video or using SSH or having a FaceTime call

Also! Can slow down or block VPN traffic to encourage people to switch back to non-VPN network.

Finally, in some settings (e.g K12 schools, DPRK), attacker may run software on the endpoint.

Hopeless...

Metadata - Hiding Messaging



Clients goal: "Hide who she is talking to." (metadata)

Informal (not like our CCA/CPA/EUF goals)

"Metadata" typically includes: endpoints, msg sizes, msg timings

Why want to hide your comms metadata?

Reveals your { medical conditions.
religion
travel plans
friends
vices
⋮

Even if attacker does not learn what you're saying, metadata can leak a LOT of sensitive info.

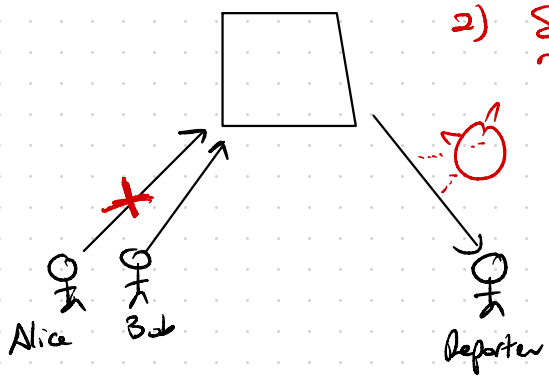
Why it's hard to achieve.

1. Even specifying/defining security is difficult.

- TLS has two parties, still very subtle to define security.
- Metadata-hiding properties involve ~ parties...
  - ↳ Many may be malicious!
  - ↳ Attacker may compromise them as protocol runs!
- Natural notions of security are impossible

e.g. Attacker wants to learn whether Alice or Bob is talking to reporter

1) Cut Alice's network
2) See if msg to reporter go through.



An attack?

Or not?

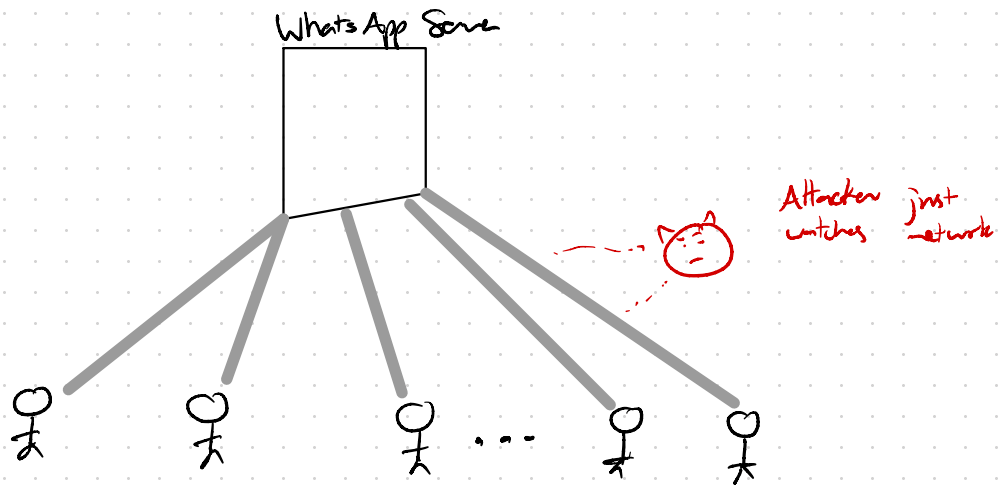
2. In any meaningful security defn, attacker is extremely powerful.

↳ Control many parties, delay msgs, watch network

↳ Makes defending against attacks difficult.

+ makes protocols super messy & complicated

First Attempt



Possible security goal:

For any pair of comm patterns

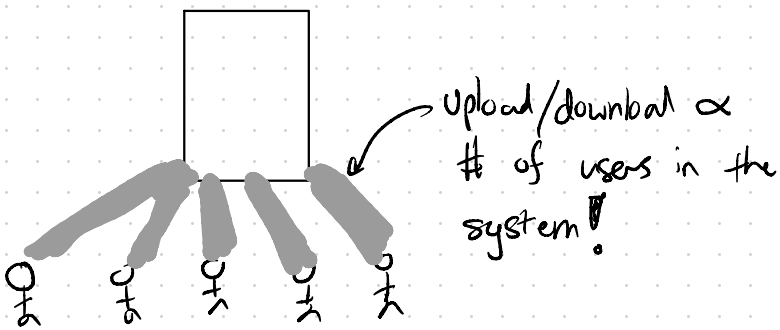
$$\left\{\begin{array}{l} \text{Alice} \to \text{Bob} \\ \text{Bob} \to \text{Carol} \\ \vdots \end{array}\right\}, \quad \left\{\begin{array}{l} \text{Alice} \to \text{Carol} \\ \text{Bob} \to \text{Dave} \end{array}\right\}$$

attacker's view of network is "the same" (comp-indist) Problem?

Problem: World in which everyone talks to Alice must "look the same" — in terms of net traffic — as world in which no one talks to Alice.

$\Rightarrow$ If correctness is perfect (Alice gets all of her msgs), seems to require a lot of b/w

↳ otherwise, attacker can break security.



Attempt #2: Weaker correctness? Alice only gets the first 10 msgs sent to her, and can only send ≤10 msgs.

BUT, we might want stronger security

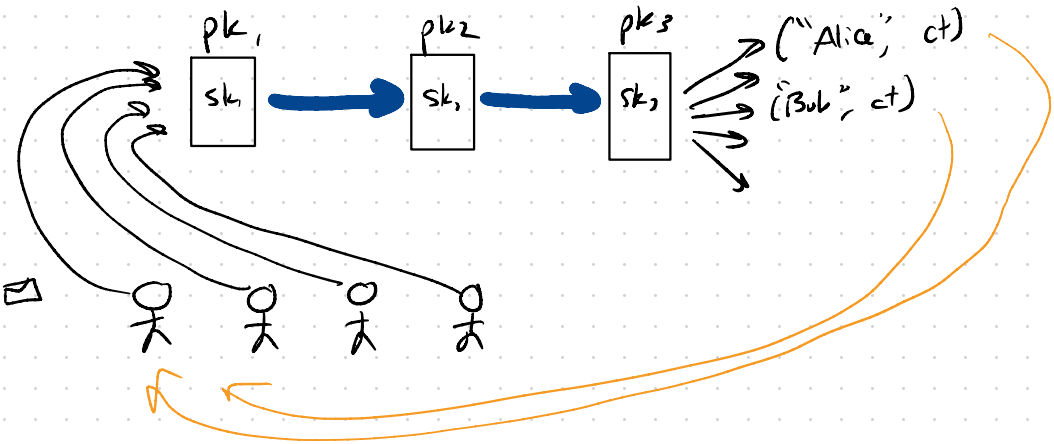
→ attacker controls some clients

→ attacker controls server

Idea: - Split the server into many servers
- Allow some leakage (# of msgs each user receives)
- Allow each client to send ≤ 1 msg per round
- Attacker can (passively) control all but 1 server.
- Require all possible worlds to be indist

A lot of restrictions!

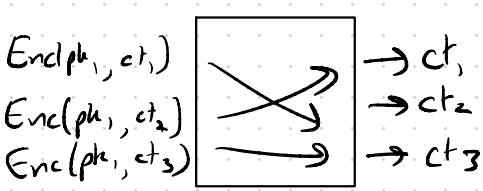
Mix nets (Chaum '81)



- To send msg to Bob, use CCA-secure PKE

$$\mathrm{Enc}(pk_1, \mathrm{Enc}(pk_2, \mathrm{Enc}(pk_3, \mathrm{Enc}(pk_{Bob}, \text{"msg"})))) = \boxed{}$$

- Each server shuffles and decrypts the batch of ctexts



- If passive attacker doesn't control one server, can't determine which msg went where.

Clever!

Why are we still not done?

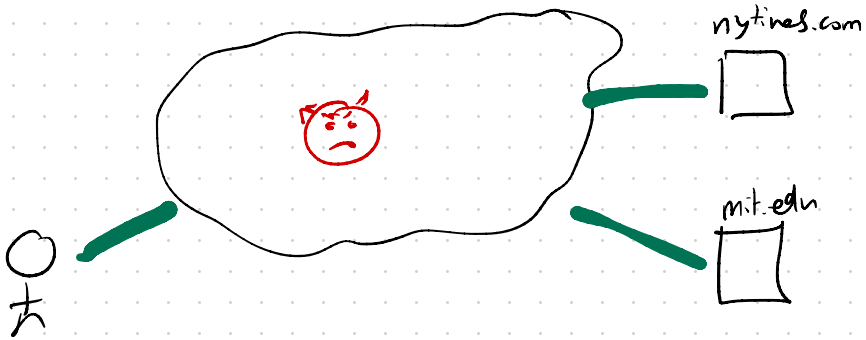
* Leakage might be too leaky { if Bob offline, does Alice get a msg?
* Active attacks make it all more complicated
* Every client needs to send in every round.
  ↳ Need to process msgs in a big batch
  (Won't work for billion-user systems)

⋮

Unclear to what extent these problems are artificial (we haven't come up with clever enough schema) vs. fundamental (inherent barriers to efficiency/security in realistic network/adv settings).

⇒ No system of this form (as far as I know) has made it into use at scale in practice.

Metadata-Hiding Web Browsing



Attacker controls network, wants to learn which website you're visiting.

↳ Like messaging but even more difficult.

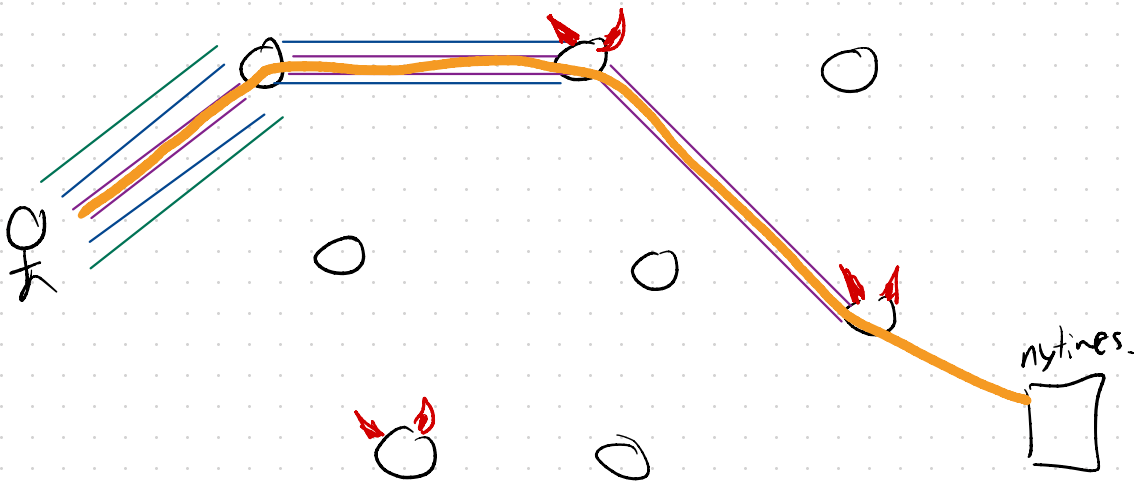
* Higher throughput, lower latency demands
* More diverse use cases (YouTube vs. NYTimes)

State of the art:

- Give up on hope to have precise security properties
- Bounce traffic around the Internet
- Hope attacker is not to. clever/powerful.

Tor

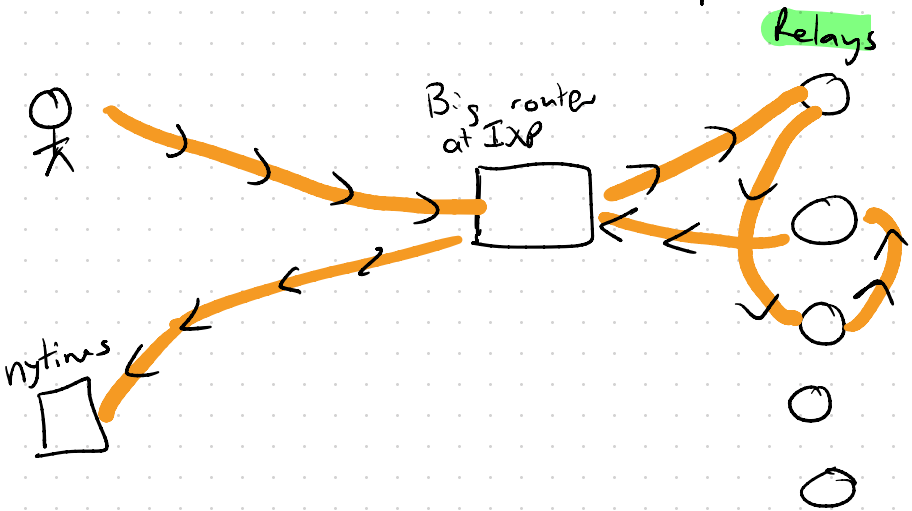
- Thousands of volunteer "relay" servers
- Build nested encrypted pipes (like TLS) through network. — 3 relays on path
- Like a VPN-in-VPN-in-VPN



Hope: If attacker is not too powerful, it will not be able to correlate input & output

* You can download & run Tor
* Millions of people use Tor daily (250 Gbps total)
* Even if security is imperfect, functionality is surprisingly good.

Unclear how much this helps...



Also, might worry about sending traffic through computers run by randoms on Internet.
↳ Maybe no worse off?

But, it's also plausible that Tor gives you much better privacy against net attackers than anything else does... just hard to know.

↳ Frustrating state of affairs...

Maybe YOU will come up with a better solution???


The
End