

Lecture 13: Open Problems in Transport Security

G.1600 - Fall 2022
C-G, Kalai, Zeldovich
MIT



Plan

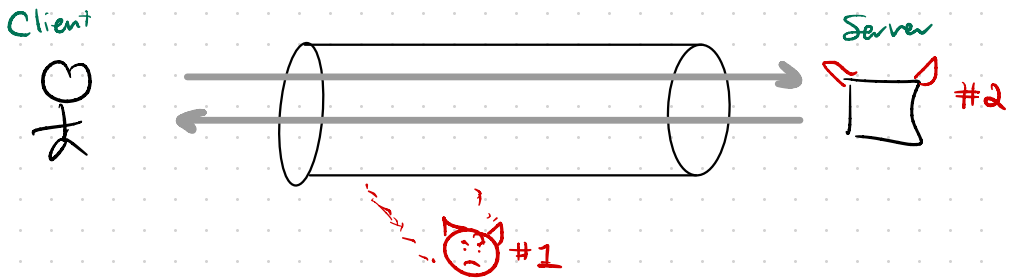
- What encryption leaks
 - * Who's talking to whom
 - * How much they're saying
- Why eliminating leakage isn't enough
 - * Server compromise
 - * Private Info Retrieval

Logistics

- * Midterm Wed - this room, this time
Open laptop, no network
- * Practice midterm on website
- * OH today 32-G970 3-4pm

Recap: Encryption in practice

- We have encryption, we have authentication.
- Using TLS, we can get "encrypted & auth pipe"
 - ↳ Convenient! Can run your favorite TCP-based protocol (HTTP, POP, SMTP, FTP, ...) over TLS to get confidentiality & authenticity.



Today

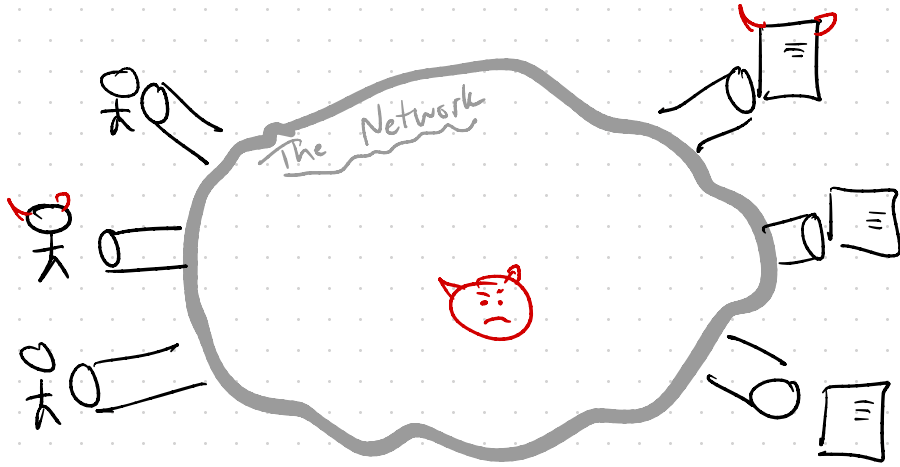
1. Imperfections in the "encrypted pipe"
↳ What it doesn't protect

- Metadata leakage

2. Even if we had a perfect encrypted pipe, why that's not good enough.

- Server compromise
- ... more in next module

"Best Possible" Security for Net Traffic

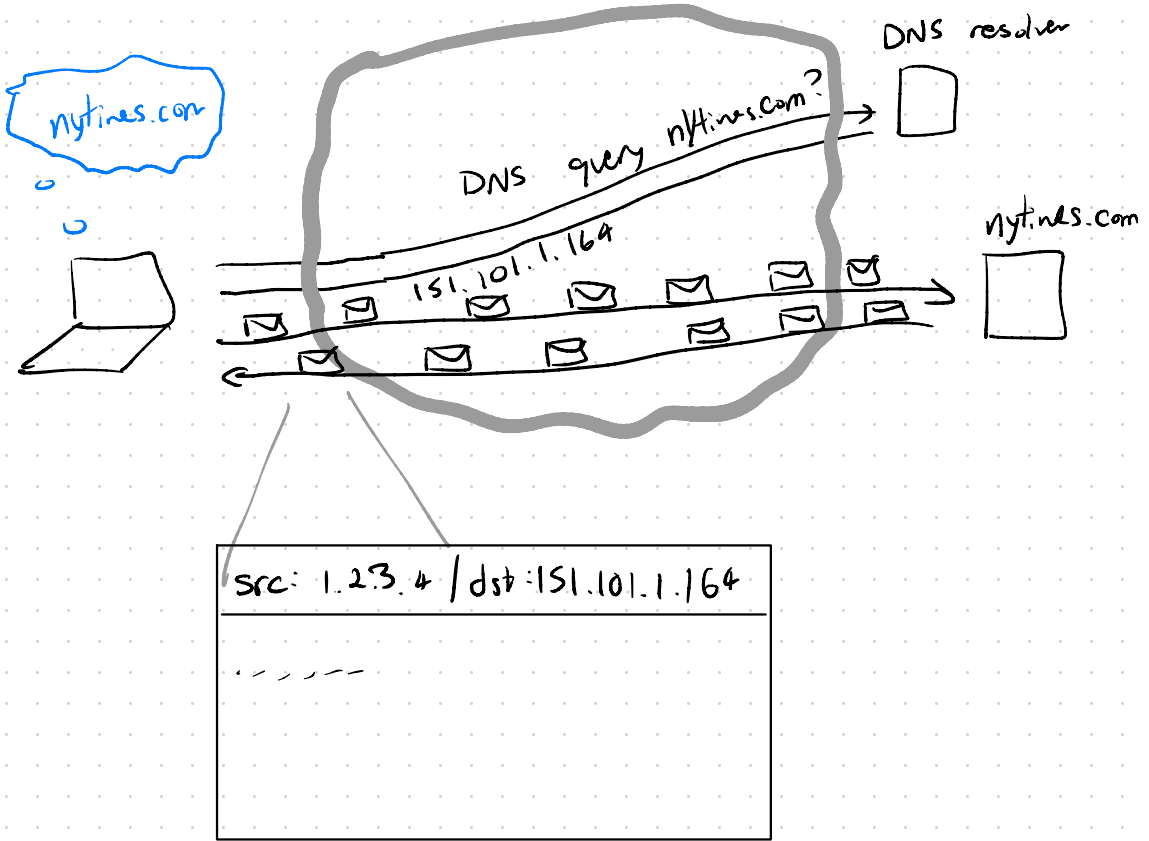


Attacker controlling some clients & servers should "learn nothing" about:

- * Which honest parties are communicating
- * what they're saying.

Problem: Net attacker sees who is talking to whom

- From this "metadata" can infer your political beliefs, religion, medical conditions, travel plans, kids' school, ...
- Also allows attacker to selectively censor your traffic



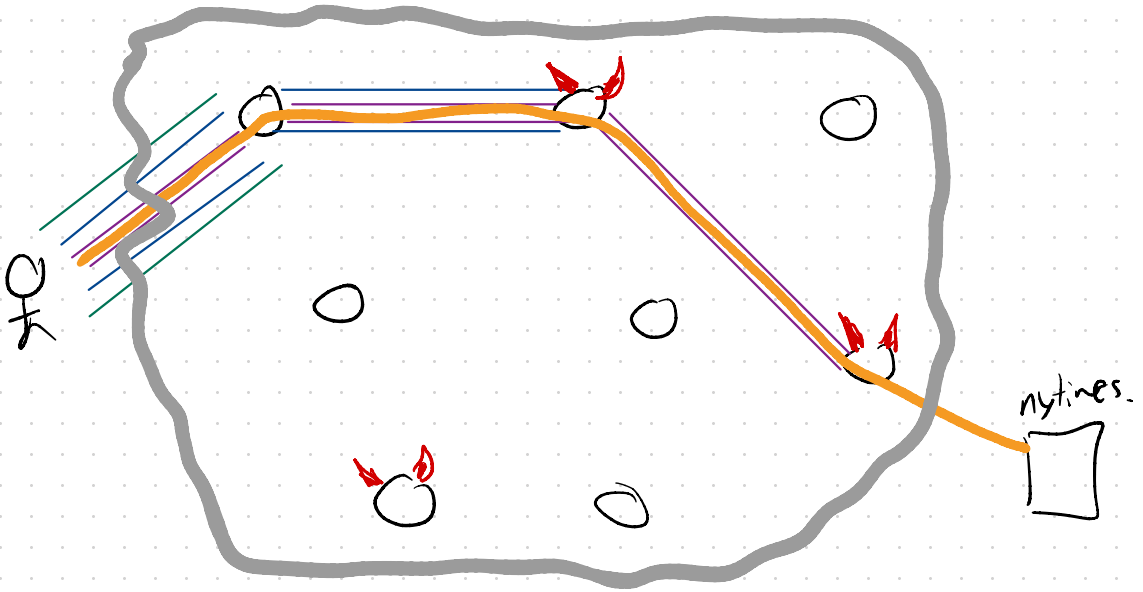
⇒ To route IP packets, routers need to know src/dst IP addresses

⇒ In-network attacker learns exactly who you're talking to (could talk to everyone all the time but...)

Attempt at a solution: Tor

Idea: * Bounce traffic around internet
* Hope attacker can't see too much of it
* Give up on precise sec def'n's

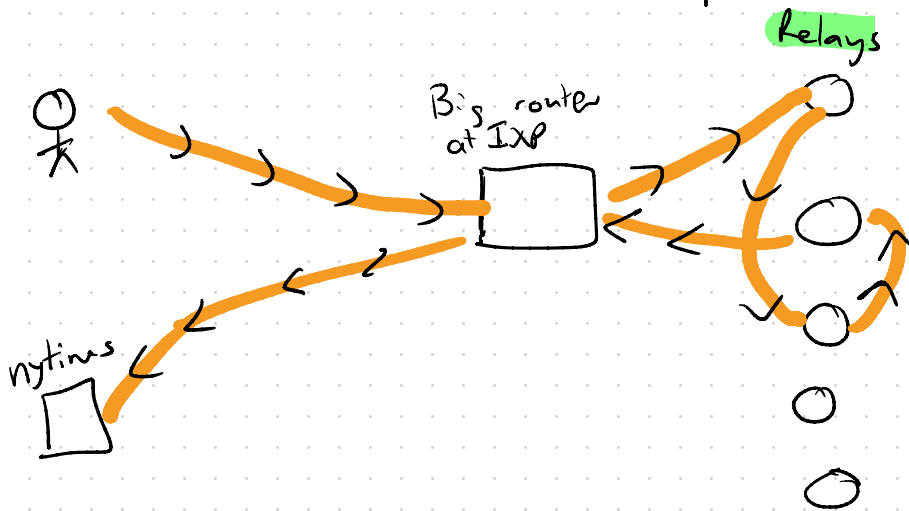
- Thousands of volunteer "relay" servers
- Build nested encrypted pipes (like TLS) through network. - 3 relays on path



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 Tor helps...



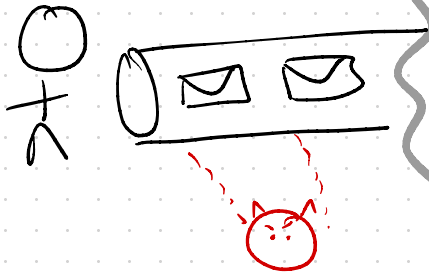
Also, might worry about sending traffic through computers run by randoms on Internet.
↳ Maybe no worse off?

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???

Problem: Attacker sees packet sizes & timing



Can learn whether you are:

- streaming a movie (which movie)
- using ssh (which commands)
- downloading a file (which file)
- browsing the web (which page)

<code>nytimes.com/index.html</code>	<code>nytimes.com/tips</code>
1.86 MB	41.92 KB
76 asset reqs	15 asset reqs

↖ CSS, img, JS, fonts, ...

Attempts at a solution...

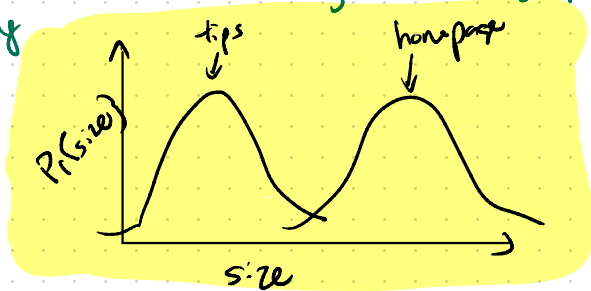
Padding: Make all nyt. pages exactly 50 MB long and make 100 reqs for assets?

- What about a page that needs 101 assets?
- Overhead is obscene!

Sort-of secure, but too \$\$\$

Random noise: Perturb length of each page & # of asset reqs
e.g. by adding random # of bytes.

- Still can leak the length since long pages remain long



- Given a handful of observations, can average out the noise.

Doesn't work at all... also \$

We are a long way from understanding how to protect communications **metadata**

↳ In contrast enc & sigs do an excellent job at protecting comm **data**.

One promising (?) direction:

- Try to solve the easier problem of metadata-hiding messaging
(Think: WhatsApp, iMessage, Signal)

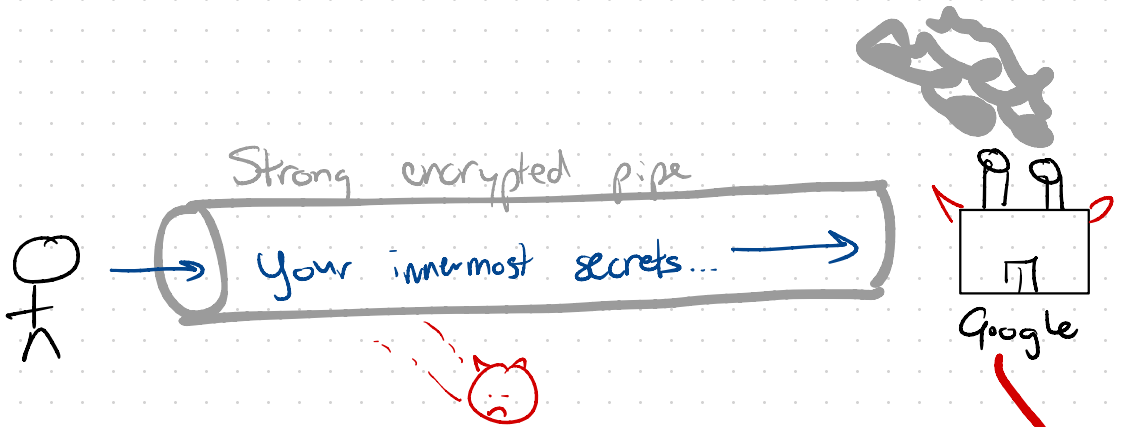
- + Msgs are \approx fixed length
- + Some latency ok
- + Total bytes sent per user is small
- + Few comm partners per user.

Potential for system that protects metadata w/
strong formal sec guarantee.

Many research papers, but still no deployed system...

Maybe you will figure it out!

Even ideal encryption isn't enough!



- * Your search queries leak your need conditions, religion, interest, beliefs, ...
- * Not to mention Gmail, Files, etc.

- Lose it in breach / compromise
- Sell your data
- Be compelled to turn it over to LEA, etc.

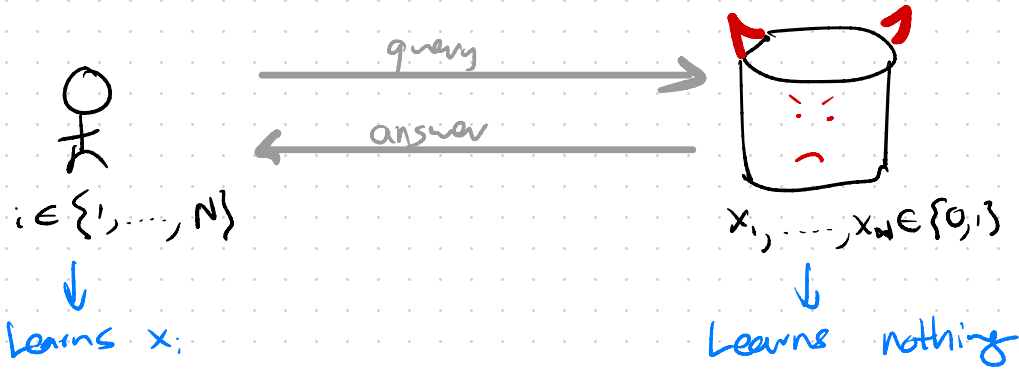
Next module focuses on platform security ...

- ↳ how to protect against server compromise by careful system design
- ↳ I'll give one example of how fancy crypto can help as well.

Private Info Retrieval

Goal: Read record from DB while hiding which record you read.

(Abstraction of private Google-search problem)



Correctness: Honest client interacting w/ honest server always outputs correct DB bit

Security: Client's query is a CPA-secure encryption of its index i .

(Why is CPA security good enough?)

Query "leaks nothing" about i .

↳ Client can fetch data from server w/o server learning what it fetched!

Surprise is that PIR is possible with
 $\ll N$ bits of communication.

We need one more tool:

Additively homomorphic encryption

- CPA-secure ~~secret key~~ enc scheme (Enc, Dec)
- Msg space is $\mathcal{M} = \mathbb{Z}_p \leftarrow \text{ints mod } p$
- Extra property, that for all $k \in \mathcal{K} \leftarrow \text{keyspace}$

for all $m, \hat{m} \in \mathcal{M}$

$$\text{Enc}(k, m) * \text{Enc}(k, \hat{m}) = \text{Enc}(k, m + \hat{m})$$

\leftarrow sum op on texts

\leftarrow addition mod p

\Rightarrow Can add msgs under encryption

\Rightarrow Can multiply by constants: $\text{Enc}(k, m) * \text{Enc}(k, c) = \text{Enc}(k, cm)$

\Rightarrow Can compute matrix-vector product of enc. vector \vec{b} public matrix



$$D * \text{Enc}(\vec{m}) = \text{Enc}(D \cdot \vec{m})$$

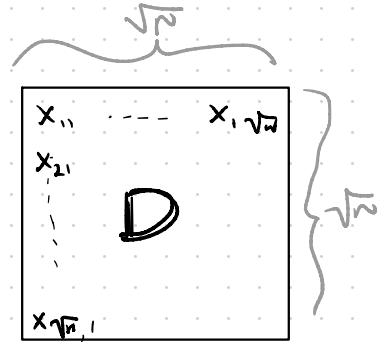
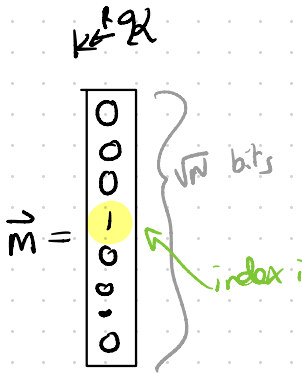
(Can build from DDH assumption ... essentially ElGamal)

Construction of PIR

- View N -bit DB as \sqrt{N} -by- \sqrt{N} matrix
- Client wants a bit in column $i \in \{1, \dots, \sqrt{N}\}$

Client

Server



$\vec{q} = \text{Enc}(k, \vec{m})$

\vec{q}

$\text{ans} = D \cdot \vec{q}$

$\text{ans} = \text{Enc}(k, D \cdot \vec{m})$

$\text{Dec}(k, \text{ans})$

$= D \cdot \vec{m}$

$= D \cdot$

0	0	0	0
0	1	0	0
0	0	0	0
0	0	0	0

$=$ ith col of DB.

Correctness:

By construction

Security:

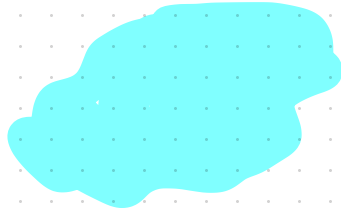
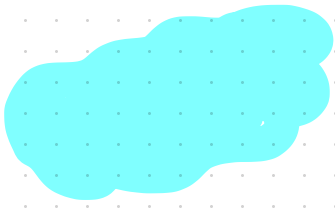
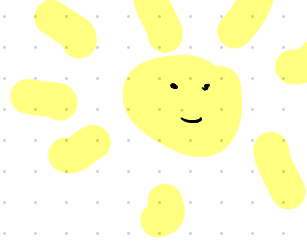
By sec of enc scheme

Comm cost:

\sqrt{N} cts $\ll N$

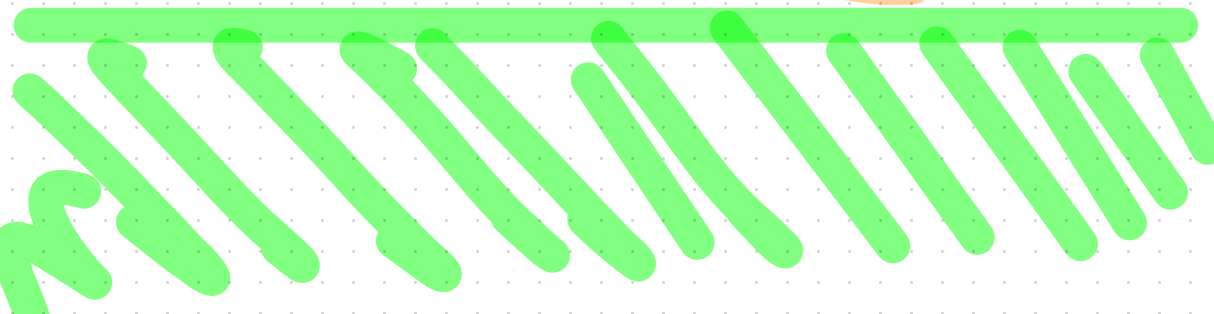
Server comp cost is high...

BUT Promising!



1. 1. 1.

2. 2. 2.



Additively Homomorphic Encryption

Group G , generator g {As used in ElGamal encryption.
order q

Msg space $\{0, \dots, M-1\}$ $M = \text{poly}(\lambda)$.
msgs are not "too big"

Gen() $\rightarrow (sk, pk)$

$x \xleftarrow{R} \{1, \dots, q\}$
 $(sk, pk) \leftarrow (x, g^x)$

Enc(pk, m) $\rightarrow ct$

$r \xleftarrow{R} \{1, \dots, q\}$
 $ct \leftarrow (g^r, g^m (pk)^r)$

Dec(sk, ct) $\rightarrow m$

$(R, T) \leftarrow ct$
 $V \leftarrow T^{sk} \cdot R^{-1} \in G$
find m s.t. $g^m = V$
by brute force
output m

Same as CPA-secure
ElGamal, with msg
"in the exponent."

Msg cannot be
too large...

Why this scheme is additively homomorphic:

$$(g^r, g^m \cdot (pk)^r) \leftarrow \text{Enc}(m, r)$$

$$(g^{\hat{r}}, g^{\hat{m}} \cdot (pk)^{\hat{r}}) \leftarrow \text{Enc}(\hat{m}, \hat{r})$$

$$(g^{r+\hat{r}}, g^{m+\hat{m}} \cdot (pk)^{r+\hat{r}}) \leftarrow \text{Enc}(m+\hat{m}, r+\hat{r})$$