Lecture 12: Open Problems in Transport Security

C. 1600 - Fall 2023
Corrigan-Gibbs & Zeldovich
MIT
Plan

- What encryption leaks
  * Who’s talking to whom
  * How much they’re saying
- Availability → also not protected

- Why elimination leakage isn’t enough
  * Server compromise
  * Private Info Retrieval

Logistics

* Midterm Wed–this room, this time
  Open laptop, no network

* Monday: Guest speaker from BU Tech Law Clinic

(Setup laptop)

(For Mirai demo)
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
   - Denial of service

2. Even if we had a perfect encrypted pipe, why that's not good enough
   - Server compromise
   ... more in next module
Problem: Not attacker sees who is talking to whom
- From this “metadata” can infer your political beliefs, religion, medical condition, travel plans, kids’ school,
- Also allows attacker to selectively censor your traffic

⇒ To route IP packets, routers need to know src/dst IP addresses (+ DNS + TLS leakage)
⇒ In-network attacker learns exactly who you’re talking to (could talk to everyone all the time but...)

- For example, ifendpoint is http://nytimes.com, DNS resolver will have to route packets between
- Source 1.2.3.4 and Destination 151.101.1.164
Attempt at a solution: Tor

Idea:
- Bounce traffic around internet
- Hope attacker can't see too much of it
- Give up on precise sec defn'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)

nytimes.com/index.html
1.56 MB
76 asset reqs
CSS, JS, fonts,

nytimes.com/tips
41.92 KB
15 asset reqs
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 le thiy 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)
  - Messages are of fixed length
  - Some latency OK
  - Total bytes sent per user is small
  - Few comm partners per user.

Potential for system that protects metadata with strong formal security guarantee.

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

Maybe you will figure it out!
Encryption does not protect availability

- Denial of service
  - We already talked about censorship but there are other reasons: often $$$

- Basic idea: Swamp service w/ fake traffic
  - Real users can't get through
- Don't have to break enc/auth to cause damage!

- Old School: Traffic amplification

\[ \text{(spoof src addr)} \rightarrow \text{Big answer} \rightarrow \text{target} \]

Mitigation: Detect spoofed src at network egress

- DDoS = distributed attack, often w/ hijacked machines

  - E.g. Mirai botnet - IoT devices
  - Reach TB/s throughput (~400 M reqs/sec)
  - IS can process 10K req/s on a machine $\Rightarrow$ 40K machines

- No great plan: Use CDNs to absorb traffic
  - Economy of scale
  - Try to make devices harder to compromise
Encryption doesn’t protect against server compromise.

A strong encrypted pipe protects your innermost secrets...

- Your search queries leak your real conditions, religious 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)

\[ \begin{align*}
\text{query} & \quad \text{answer} \\
\{0, \ldots, N\} & \quad \{0, 1\}
\end{align*} \]

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 \( i \).

\( \triangledown \) 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 semi-secure scheme \((\text{Enc}, \text{Dec})\)
  - Cannot be CCA secure... why?
- Msg space is \( M = \mathbb{Z}_p \) \(
\) (ints \( \pmod{p} \))
- Extra property that for all \( k \in \mathcal{K} \) \( \leftarrow \) key space

For all \( m, \hat{m} \in \mathcal{M} \)

\[
\text{Enc}(k, m) \times \text{Enc}(k, \hat{m}) = \text{Enc}(k, m + \hat{m})
\]

\( \leftarrow \) Some op on texts

\( \Rightarrow \) Can add msgs under encryption

\( \Rightarrow \) Can multiply by constants: \( \text{Enc}(k, m) \times \text{Enc}(k, \hat{m}) = \text{Enc}(k, 2m) \)

\( \Rightarrow \) Can compute matrix-vector product of enc

vector \& public matrix

\[
D \times \text{Enc}(\hat{m}) = \text{Enc}(D \cdot \hat{m})
\]

(Can build from DDH assumption \( \ldots \) essentially ElGamal)
Construction of PIR

- View $N$-bit DB as $\sqrt{N}$-by-$\sqrt{N}$ matrix
- Client wants a bit in column $i = 1, \ldots, \sqrt{N}^3$

Client

\[
\begin{align*}
1 \leq k \leq N \\
M = \begin{bmatrix}
0 & 0 & 0 & \cdots & 1 & 0 & \cdots \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
0 & 0 & 0 & \cdots & 0 & 0 & \cdots \\
\end{bmatrix}
\end{align*}
\]

\[ q = \text{Enc}(k, m) \]

\[ \text{ans} = D \cdot \overline{q} \]

\[ \text{Dec}(k, \text{ans}) = D \cdot \overline{m} \]

\[ = D \cdot \begin{bmatrix} 0 & 0 & \cdots & 1 & 0 & \cdots \end{bmatrix} \]

\[ = \text{ith col of DB} \]

Server

\[ x_1, x_2, \ldots, x_{\sqrt{N}^3} \]

\[
D
\]

Correctness: By construction

Security: By sec of enc scheme

Comm cost: $\sqrt{N}$ cts $< N$

Server comp cost is high...

...but promising!
Additively Homomorphic Encryption

Group $G$, generator $g$ [As used in ElGamal encryption; order $q$]

Msg space $\{0, \ldots, M-1\}$ \(M = \text{poly}(\lambda)\)

msgs are not "too big"

$\text{Gen}(\lambda) \rightarrow (sk, pk)$

\[ x \leftarrow \{1, \ldots, q\} \]
\[ (sk, pk) \leftarrow (x, g^x) \]

$\text{Enc}(pk, m) \rightarrow ct$

\[ r \leftarrow \{1, \ldots, q\} \]
\[ ct = (g^r, g^m \, (pk)^r) \]

$\text{Dec}(sk, ct) \rightarrow m$

\[ (R, T) \leftarrow ct \]
\[ V \leftarrow T^{sk} \cdot R^{-1} \in G \]

find $m$ s.t. $g^m = V$

by brute force output $m$

Msg cannot be too large...
Why this scheme is additively homomorphic:

\[ (g^r, g^m (pk)^r) \leftarrow \text{Enc}(m_{j \cdot r}) \]

\[ (g^\hat{r}, g^{\hat{m}} (pk)^{\hat{r}}) \leftarrow \text{Enc}(m_{j \cdot \hat{r}}) \]

\[ (g^{r+\hat{r}}, g^{m+\hat{m}} (pk)^{r+\hat{r}}) \leftarrow \text{Enc}(m+\hat{m}, r+\hat{r}) \]
The End.