Privacy: anonymity, and confidentiality, anti-censorship.

Anonymous routing.

Confidentiality + Anonymity
- Censorship resistance (and privacy)

Goal: ability to send confidential anonymous messages. Only the source and destination know the message content. Network intermediaries do not know who is communicating.

Anonymous routing on top of overlay - Tor
1. Uses private, public keys to establish symmetric keys.
2. Uses symmetric keys to construct onion routes.

Problem: - trusted database of nodes.
- key management? Big challenge.

Can you do it without keys? - can use random overlays to get both confidentiality and anonymity.

Confidentiality: information elision.

\[ I \rightarrow I, I_1 \rightarrow \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \rightarrow \begin{bmatrix} I_1' \\ I_2' \end{bmatrix} \]

Forward: \( I, a_{11}, a_{12} \) on path 1; only D can recover.
\( I, a_{21}, a_{22} \) on path 2.
Anonymity \[\rightarrow\] 1. route set-up
\[\leftarrow\] 2. data transmission.

How to do route set-up?

Anonymous routes: each node knows next hop,
- no one else knows a node's next hop
- send next hop info in a confidential manner.

Reuse nodes to send multiple pieces as long as pieces belong to different messages.

Dealing with churn \[\rightarrow\] use new coding in addition to source coding.

Censor:
- Restrictive govt, corporate firewall.
- Discovery attacks: notice unusual looking traffic, suspicious web access patterns, use of communication s/w.
- Disruptive attacks: block access to certain websites, attempts to block access to circumvention s/w.
Design goals for censorship resistance: circumvent so that both forms of these attacks are fruitless or unsuccessful.
- 
  - Deniability: can't confirm a client is accessing censored stuff
  - That den: should not arouse suspicion
  - Competence for servers: can't discover a server that is serving censored content to avoid blocking.
  - Communication robustness: should be robust in the face of censor disruption
  - Reasonable performance

- Use unframed proxy on localhost (squid)
- Upstream request is sequence of messages.
- Downstream response in image

Problems today

1. SSL fingerprinting
2. SSL looks suspicious
3. Not attempt to central servers
4. SSL can be blocked.
Downstream:  - Embed data in the less useful portion of image.
  - Decided by shared secret
  - Need to change cover image to use a webcam.

Upstream comm:

```
req. C
http://cs.wisc.edu

"/reqprom" http://www.2can.com
```

Mapping function: page on responder → public page
- Should be secure → critical to deliverability.

Several candidates: countries vs. bandwidth.

- Odd/even links: sequence may ask for any one of half of the links.
  1 bit per visible http request.
- Lenta modulo \( k \):
  any one of \( k \) links, \( \log(k) \) bits.
- Static mapping: strange browsing → poor throughput
  bandwidth \( M \) bits per request.
Range mapping: high-bandwidth, dynamic mapping
   -> Web-surfing 20-questions style.

Assume: set of answered questions commonly requested.

Responder tells requester:
1. Split strings for ranges in this set.
2. Mapping between splits and visible requests.

Requester sends a visible http request:

<table>
<thead>
<tr>
<th>Visible</th>
<th>Split String</th>
<th>Visible</th>
<th>Split String</th>
</tr>
</thead>
<tbody>
<tr>
<td>people.html</td>
<td>-</td>
<td>alexsa</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>microsop</td>
<td>-</td>
</tr>
<tr>
<td>publication.html</td>
<td>-</td>
<td>yahoo</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>jaffb</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>brian</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td>.microsoft</td>
<td>-</td>
</tr>
</tbody>
</table>

This assumes that any page on the responder is equally likely... NOT TRUE.

Diagram:

Idea can be applied once achieved and expanded.