Cryptographic Challenges in and around Tor

Nick Mathewson
The Tor Project
9 January 2013
Summary

- Very quick Tor overview
- Tor's cryptography, and how it's evolving
- Various opportunities for more Tor crypto work

Disclaimer:

This is not exhaustive; these are only our most interesting crypto needs, not all of them; these are not our most urgent needs in general.
Part 1: Tor overview
Ordinarily, traffic analysis and censorship are easy.
Ordinarily, traffic analysis and censorship are easy.
Tor makes traffic analysis and censorship harder...

`User` → Tor Network (abstract) → `Server`

`User` → Tor Network (abstract) → `Server`

`User` → Tor Network (abstract) → `Server`
...by using a network of relays to anonymize traffic.

(Use non-public entry relays to resist censorship.)
(But an end-to-end traffic correlation attack still works.)
Tor is the largest deployed network of its kind

- 3000 relays
- 1000 public bridges
- > 2 GiB/sec
- > 500,000 users each day (estimated)
  - (With a pretty broad diversity of interest)
Part 2: Tor could use better crypto
Tor uses TLS for its link protocol...
… with all the problems that entails.

- Easy to detect TLS variants based on:
  - Cipher choice
  - Certificate structure
  - List of extensions
- More secure: less common. Can't use any unpopular TLS feature.

(Did you know I have an effective veto over any new TLS features?)
Maybe other link protocols are better for anticensorship?

There are a number of these “Pluggable Transports” in development, but we need even more. *Even weak stego can help.*

...Do we still need “normal-looking” TLS?
Tor needs a one-way-authenticated handshake to build circuits

User

Relay A

Relay B

E(PK_A, g^{x1})

\(g^{y1}, H_1(g^{x1y1})\)

(Now have \(K_1 = KDF(g^{xy})\)
Tor needs a one-way-authenticated key exchange to build circuits

User

Enc(PK_A, g^x_1)

Relay A

\( g^y_1, H_1(g^{x_1y_1}) \)

(Now have \( K_1 = KDF(g^{x_1y_1}) \))

\( E_{K1}(Enc(PK_B,g^{x_2})) \)

Enc(PK_B,g^{x_2})

Relay B

\( g^y_2, H_1(g^{x_2y_2}) \)

E_{K1}(g^{y_2}, H_1(g^{x_2y_2}))

(Now have \( K_2 = KDF(g^{x_2y_2}) \))
We're replacing this protocol...

- Original protocol ("TAP") did hybrid encryption with RSA,DH-1024, badly. [Goldberg 2006]
- Replacement ("ntor") does *approximately*
  
  $C \rightarrow S: g^x$
  
  $S \rightarrow C: g^y, \ H1(inp=H( g^x \ g^y \ g^{xb} \ g^{xy} \ ...))$
  
  $K = KDF(H2(inp))$
  
  [Goldberg, Stebila, Ustaoglu 2011]
  
  (We're using DJB's curve25519 for DH group)
...and might replace it again

- Alternative ("ace") does approximately:
  C->S: $g^{x1}, g^{x2}$
  S->C: $g^y$
  $K = KDF(g^{bx1 + yx2})$
  [Backes, Kate, Mohammedi 2012]

- Best choices will depend on implementation tweaks.
- Can you do better?
We should replace our old relay cell protocol...

- Used for symmetric crypto once we have shared keys.

<table>
<thead>
<tr>
<th>Zeros (2)</th>
<th>Bad “MAC” (4)</th>
<th>Payload (503)</th>
</tr>
</thead>
</table>


We should replace our old relay cell protocol...

- Used for symmetric crypto once we have shared keys.

<table>
<thead>
<tr>
<th>Zeros (2)</th>
<th>Bad “MAC” (4)</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AES_CTR(Key1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AES_CTR(Key2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AES_CTR(Key3)</td>
<td></td>
</tr>
</tbody>
</table>
We should replace our old relay cell protocol...

- Used for symmetric crypto once we have shared keys.

<table>
<thead>
<tr>
<th>Zeros (2)</th>
<th>Bad “MAC” (4)</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AES_CTR(Key1)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AES_CTR(Key2)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AES_CTR(Key3)</td>
<td></td>
</tr>
</tbody>
</table>

To handle a cell:
- Remove a layer of encryption.
- If Zeros == 0, and “MAC” = H(Key3_M, Previous cells | Payload):
  - This cells is for us!
- Else, relay the cell
We should replace our old relay cell protocol...

- Used for symmetric crypto once we have shared keys.

<table>
<thead>
<tr>
<th>Zeros</th>
<th>Bad “MAC” (4)</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES CTR(Key1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES CTR(Key2)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AES CTR(Key3)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

But this is malleable!
Hang on, does it matter that it's malleable?

- Honest exit (probably) rejects M''
- Evil exit detects tag, but could just as easily do traffic correlation, for same result at less risk of detection.
- So, don't worry? (Dingledine, Mathewson, Syverson 2004)
Hang on, does it matter that it's malleable?

- Honest exit (probably) rejects $M''$
- Evil exit detects tag, but could just as easily do traffic correlation, for same result, at less risk of detection.
- Actually, it's not so clear-cut.
We could use an encrypt-and-mac structure

ENC(Payload,K1) → MAC1 → MAC2 → MAC3

ENC(..., K2)

ENC(..., K3)
We could use an encrypt-and-mac structure

ENC(Payload,K1) → MAC1 → MAC2 → MAC3

ENC(..., K2)

ENC(..., K3)

But that requires one MAC per hop, and leaks path length.
A chained wide-block cipher seems like a much better idea!

<table>
<thead>
<tr>
<th>Zeros</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WideBlock(Key1)</td>
</tr>
<tr>
<td></td>
<td>WideBlock(Key2)</td>
</tr>
<tr>
<td></td>
<td>WideBlock(Key3)</td>
</tr>
</tbody>
</table>

+
A chained wide-block cipher seems like a much better idea!

Any attempt to change the block renders the whole circuit unrecoverable.
What wide-block cipher to use?

- Not enough time to discuss all of them (LIONESS, CMC, XCB, HCTR, XTS, XEX, HCH, TET)
- Needs to be fast, proven, secure, easy-to-implement, non-patent-encumbered, side-channel-free,…
- One promising approach in progress by Bernstein, Sarkar, and Nandi – HFFH Feistel structure, fast, not yet finished.
- Other ideas?
Tor gets blocked too much.

- Some services mistake Tor for abuse
- Some services use IP blocking as a proxy for people-blocking, and can't not block Tor. (Wikipedia edits, some IRC nets.)

Can we do better?
Provide a way for users to make themselves blockable.

- Slightly expensive pseudonyms?
  - (Expensive how? SA model?)

- Anonymous blacklistable credentials? (Nymble, BNymble, BLACR, VERBS, Jack...)
  - Time to try this out in the wild?
  - What will we learn about their usability? Are they right?
There are more crypto issues in Tor

- Directory protocol
- Hidden service protocol
- Better DOS resistance
- SHA1, RSA1024 for node identity
Questions?

- See https://www.torproject.org/ for links to documentation, specifications, and more info about various Tor issues.
- See http://freehaven.net/anonbib/ for an incomplete but nonetheless useful anonymity bibliography.
- Grab me during a break for non-crypto Tor questions