Anonymous Credentials and Social Mechanisms for Censorship Circumvention

What is Tor?

How Tor Works

Tor Bridges

- Secret entrances to the Tor network
- Must be distributed out-of-band
- DPI or an active adversary is required to identify Bridges
- Distributed via a centralised system called BridgeDB

Arms Race

Arms Race

- Since 2010: China's GFW began active probing Tor Bridges

  - Observe Tor client's TCP connection to the Bridge

  - For Tor<0.2.3.17-beta, identification was based upon Tor's unique ciphersuite list

  - A seemingly random machine from somewhere in China (possibly using IP-spoofing) will connect to the Bridge's IP:port and attempt to complete the first couple steps of the handshake

  - The Bridge is blocked by IP:port

  - The GFW sometimes spoofs a RST from Bridge to the client

Arms Race

- 2012: Ethiopia began blocking all TLS by looking for the client HELLO.
- Any packet with the string "TLS_DHE_RSA_WITH_AES_256_CBC_SHA" in it is dropped. If you pick "TLS_DHE_RSA_WITH_AES_128_CBC_SHA" instead, or split the ciphersuite list, it works.

Pluggable Transports, obfsproxy, & other unpronouceables

obfs4proxy

- Tor's NTor handshake with public keys obfuscated via the Elligator 2 mapping
- Link layer uses NaCl secret boxes (Poly1305, Xsalsa20)

Simple Formulae

- Make the handshake as uniform as possible
- Use some pre-shared key material for authentication of the server and encrypt starting with the client's first message

This just sweeps the problems under the rug

Bridge Distribution

```
Bridge obfs4 106.187.37.158:62421 50182425F17DEF0B51B0790188D2E04E300314B7
cert=pKDDKPfTYDJjX2tJbm6z/CW3+dnEg1vw3YjofAw2fbDnHJ2Rc7/yTAFg/1RiyoMme5Dgcw iat-
mode=0

Bridge obfs4 178.209.52.110:443 67E72FF33D7D41BF11C569646A0A7B4B188340DF
cert=Z+cv8z19Qb8RxWlkagp7SxiDQN++b7D2Tntowhf+j4D15/kLuj3EoSSGvuREGPc3h60Ofw iat-
mode=0

Bridge obfs4 83.212.101.3:41213 A09D536DD1752D542E1FBB3C9CE4449D51298239
cert=lPRQ/MXdD1t5SRZ9MquYQNT9m5DV757jtdXdlePmRCudUU9CFUOX1Tm7/meFSyPOsud7Cw iat-
mode=0
```

Proof-of-Work doesn't work

If the adversary is so omnipotent, then what can't they do?

If the adversary is so omnipotent, then what can't they do?

Make friends!

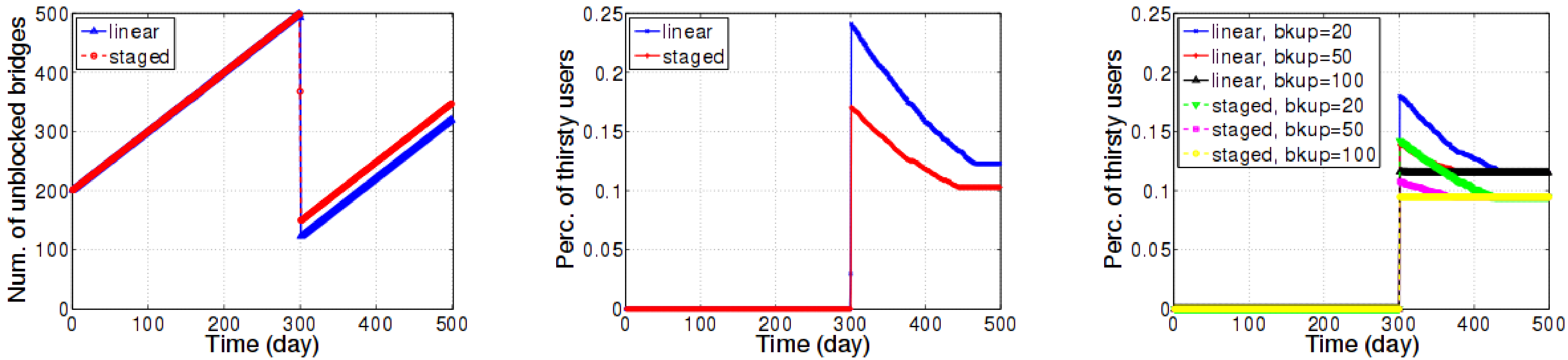
Social Graph Leakage
is
Bad News Bears

rBridge

Wang, Q., Lin, Z., Borisov, N., & Hopper, N. (2013, February). rBridge: User Reputation based Tor Bridge Distribution with Privacy Preservation. In *NDSS*.

- Users are given “brownie points” for “good behaviour”
- Users with enough brownie points might win the chance to invite their friends
- Censors lock themselves out of the system via their own bad behaviour. Also nobody wants to be friends with those losers anyway.

The Best Strategy for Censors



(a) Unblocked bridges (b) Thirsty users (c) Thirsty users with backup bridges

Figure 4: Event-driven blocking ($f = 5\%$)

Some Minor Problems

Some odd crypto choices, silly mistakes, and efficiency sacrifices for very little added privacy

- K-TAA signature scheme   O_o'
- Pedersen commitments on vectors
- Oblivious Transfer
- Ad-hoc anonymous credential construction from k-TAA signatures and Camenisch-Stadler NIZK proof of discrete logarithm.

Redesign

- CL Anonymous Credentials

  Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., & Shacham, H. (2009). Randomizable proofs and delegatable anonymous credentials. In Advances in Cryptology-CRYPTO 2009 (pp. 108-125). Springer Berlin Heidelberg.

  Uses Boneh-Boyen signatures and rerandomised Groth-Sahai NIZK proofs of satisfiability of a pairing-product equation to construct delegatable authentication tokens with attributes.

Redesign

- CL Anonymous Credentials

  Removes the need for k-TAA signatures, fixes the issues with making Pedersen commitments on vectors of independent variables, and provides a slightly less insane alternative to the credential constructions.

Redesign

$$
\pi_2 = \mathsf{NIPK}\left\{
\begin{array}{l}
(x, \Phi, C_\Phi, e_\Phi, s_\Phi, s'_\Phi, r_\Phi^{(1)}, r_\Phi^{(2)}, \delta_\Phi^{(1)}, \delta_\Phi^{(2)},\\
B_u, \tau_u, \phi_u, C_u, e_u, s_u, s'_u, r_u^{(1)}, r_u^{(2)}, \delta_u^{(1)}, \delta_u^{(2)},\\
\bar{\Phi}, \bar{s}'_\Phi, \bar{\phi}_u, \bar{s}'_u) :\\
\quad \bigwedge_{j=1}^{m} \left[b_j \neq z^{B_u} \right] \wedge\\
\quad C_u = g_1^{s'_u} g_2^{x} g_3^{B_u} g_4^{\tau_u} g_5^{\phi_u} \wedge\\
\quad A_u^{(1)} = g_1^{r_u^{(1)}} g_2^{r_u^{(2)}} \wedge\\
\quad (A_u^{(1)})^{c_u} = g_1^{\delta_u^{(1)}} g_2^{\delta_u^{(2)}} \wedge\\
\quad \frac{\hat{e}(A_u^{(2)}, pk)}{\hat{e}(g_0, h)} = \hat{e}(A_u^{(2)}, h)^{-c_u} \hat{e}(g_2, y)^{r_u^{(1)}}\\
\qquad \hat{e}(g_2, h)^{\delta_u^{(1)}} \hat{e}(g_1, h)^{s_u} \hat{e}(g_2, h)^{x}\\
\qquad \hat{e}(g_3, h)^{B_u} \hat{e}(g_4, h)^{\tau_u} \hat{e}(g_5, h)^{\phi_u} \wedge\\
\quad C_\Phi = g_1^{s'_\Phi} g_2^{x} g_3^{\Phi} \wedge\\
\quad A_\Phi^{(1)} = g_1^{r_\Phi^{(1)}} g_2^{r_\Phi^{(2)}} \wedge\\
\quad (A_\Phi^{(1)})^{c_\Phi} = g_1^{\delta_\Phi^{(1)}} g_2^{\delta_\Phi^{(2)}} \wedge\\
\quad \frac{\hat{e}(A_\Phi^{(2)}, pk)}{\hat{e}(g_0, h)} = \hat{e}(A_\Phi^{(2)}, h)^{-c_\Phi} \hat{e}(g_2, y)^{r_\Phi^{(1)}}\\
\qquad \hat{e}(g_2, h)^{\delta_\Phi^{(1)}} \hat{e}(g_1, h)^{s_\Phi} \hat{e}(g_2, h)^{x} \hat{e}(g_3, h)^{\Phi} \wedge\\
\quad \kappa_\Phi = z^{s_\Phi} \wedge\\
\quad t_u = T_{cur} - \tau_u \wedge\\
\quad \Big[\Big(t_u < T_0 \wedge \bar{\phi}_u = 0 \Big) \vee\\
\qquad \Big(t_u \geq T_0 \wedge t_u \leq T_1 \wedge \bar{\phi}_u = \rho(t - T_0) \Big) \vee\\
\qquad \Big(t_u > T_1 \wedge \bar{\phi}_u = \rho(T_1 - T_0) \Big) \Big] \wedge\\
\quad \bar{\Phi} = \Phi + \bar{\phi}_u - \phi_u \wedge\\
\quad \bar{C}_u = g_1^{\bar{s}'_u} g_2^{x} g_3^{B_u} g_4^{\tau_u} g_5^{\bar{\phi}_u} \wedge\\
\quad \bar{C}_\Phi = g_1^{\bar{s}'_\Phi} g_2^{x} g_3^{\bar{\Phi}} \wedge
\end{array}
\right\}
$$

$$
\pi_3 = \mathsf{NIPK}\left\{
\begin{array}{l}
(x, \Phi, C_\Phi, e_\Phi, s_\Phi, s'_\Phi, r_\Phi^{(1)}, r_\Phi^{(2)}, \delta_\Phi^{(1)}, \delta_\Phi^{(2)}, \tau_b,\\
\phi_b, C_b, e_b, s_b, s'_b, r_b^{(1)}, r_b^{(2)}, \delta_b^{(1)}, \delta_b^{(2)}, \bar{\Phi}, \bar{s}'_\Phi) :\\
\quad C_b = g_1^{s'_b} g_2^{x} g_3^{B_b} g_4^{\tau_b} g_5^{\phi_b} \wedge\\
\quad A_b^{(1)} = g_1^{r_b^{(1)}} g_2^{r_b^{(2)}} \wedge\\
\quad (A_b^{(1)})^{c_b} = g_1^{\delta_b^{(1)}} g_2^{\delta_b^{(2)}} \wedge\\
\quad \frac{\hat{e}(A_b^{(2)}, pk)}{\hat{e}(g_0, h)} = \hat{e}(A_b^{(2)}, h)^{-c_b} \hat{e}(g_2, y)^{r_b^{(1)}}\\
\qquad \hat{e}(g_2, h)^{\delta_b^{(1)}} \hat{e}(g_1, h)^{s_b} \hat{e}(g_2, h)^{x}\\
\qquad \hat{e}(g_3, h)^{B_b} \hat{e}(g_4, h)^{\tau_b} \hat{e}(g_5, h)^{\phi_b} \wedge\\
\quad \kappa_b = z^{s_b} \wedge\\
\quad C_\Phi = g_1^{s'_\Phi} g_2^{x} g_3^{\Phi} \wedge\\
\quad A_\Phi^{(1)} = g_1^{r_\Phi^{(1)}} g_2^{r_\Phi^{(2)}} \wedge\\
\quad (A_\Phi^{(1)})^{c_\Phi} = g_1^{\delta_\Phi^{(1)}} g_2^{\delta_\Phi^{(2)}} \wedge\\
\quad \frac{\hat{e}(A_\Phi^{(2)}, pk)}{\hat{e}(g_0, h)} = \hat{e}(A_\Phi^{(2)}, h)^{-c_\Phi} \hat{e}(g_2, y)^{r_\Phi^{(1)}}\\
\qquad \hat{e}(g_2, h)^{\delta_\Phi^{(1)}} \hat{e}(g_1, h)^{s_\Phi} \hat{e}(g_2, h)^{x} \hat{e}(g_3, h)^{\Phi} \wedge\\
\quad \kappa_\Phi = z^{s_\Phi} \wedge\\
\quad t_b = \beta_b - \tau_b \wedge\\
\quad \Big[\Big(t_b < T_0 \wedge \bar{\phi}_b = 0 \Big) \vee\\
\qquad \Big(t_b \geq T_0 \wedge t_b \leq T_1 \wedge \bar{\phi}_b = \rho(t_b - T_0) \Big) \vee\\
\qquad \Big(t_b > T_1 \wedge \bar{\phi}_b = \rho(T_1 - T_0) \Big) \Big] \wedge\\
\quad \bar{\Phi} = \Phi + \bar{\phi}_b - \phi_b - \phi^{-} \wedge\\
\quad \bar{\Phi} > 0\\
\quad \bar{C}_\Phi = g_1^{\bar{s}'_\Phi} g_2^{x} g_3^{\bar{\Phi}} \wedge
\end{array}
\right\}
$$

Remove Oblivious Transfer

- rBridge uses n-out-of-m OT to hide which Bridges are distributed to a client at the time of distribution.
- Another messy construction for an additional proof of inequality of openings to commitments to chosen Bridge and some previous Bridge to avoid duplicates.
- In the end, the client tells the server which Bridge it has when it reports the Bridge was blocked.

Open Questions

- What do we mean when we say “A Bridge is blocked”? Does “blocked in China” mean “blocked in Iran”? What if China sells data on Tor Bridges to Iran?
- Simpler anonymous credential constructions which don't require pairings? E.g. based upon algebraic MACs or Diffie-Hellman. Apparently pairings aren't cool enough or something and cryptographers keep needing to invent cool stuff to one-up each other.