Why is anonymity so hard?

Roger Dingledine

The Free Haven Project

Many people need anonymity

- Political dissidents in oppressive countries
- Governments want to do operations secretly.
- Corporations are vulnerable to traffic analysis (corporate espionage) — VPNs, encryption don't cut it.
- Individuals are tracked and profiled daily. Imagine what they'll have in your dossier in twenty years.
- (If that doesn't scare you, think of your kids.)

A MIX node

- Messages change appearance after decryption
- Each MIX batches and reorders messages
- Messages are all the same length
- Store and forward (slow) to maintain anonymity sets

A MIX cascade

Free-route MIX networks

- User picks a path through the network
- Goal is to hide message's *path*
- Needs dummy traffic (inefficient, poorly understood) to protect against global adversaries (lots of traffic may work too?)
- Example: Mixmaster

Zero KnowledgeB48toFrledomoNB48gewno

Onion Routing

- Connection-oriented (low latency)
- Long-term connections betwee7 Onion Routers link padding betwee7 the routers
- Aims for security against traffic analysis, not traffic confirmation
- Users should run node, or anonymize connection to first node, for best privacy

Some technical problems for Onion Routing:

Convenient/Usable Proxies

- Currently we have an *application proxy* for each protoco610w which feeds into the *onion proxy*. Users should run both.

- But we really ought to intercept all traffic – otherwise we need to modify applications so they don't leak info.

- ...and nobody will use it if we need all these proxies (not true: p2p systems?)

Oh yeah, and I wrote the Onion Routing code

- It’s GPLed ... but it’s complicated.
- Send me mail and I’ll point you to it.

Ideal threat model

- Global passive adversary – can observe *everything*
- Owns half the nodes

Link padding and topology

- Remember that our goal is to hide the *path*
- Without link padding, adversary can observe when new connections start, and where they go.
- n^2 link padding is insane, but anything less seems unsafe.
- Open problem: what's the right compromise?

Timing attacks

- If the adversary owns two nodes on your path, he can recognize that they're on the same path

- Works passively (anab(1.t4(des)-.id1.t4(dw(o)3(pat)-1(c)hab(1.t4(

Tagging attacks

- Onion routing uses a stream cipher to encrypt the data stream going in each direction.

- An adversary owning a node – or a link! – can flip a byte in the data stream and look for an anomalous byte at the exit point (say, when it talks to a webserver).

- This sort of thing is generally solved by including a hash, but it's more complex than that.

Anonymity is hard for economic/social reasons too

- Anonymity requires *inefficiencies* in computation/F1-6(bandwidth,) T

But trust bottlenecks can break everything

- Nodes with more traffic must be more trusted
- Adversary who wants more traffic should provide good service
-

Strong anonymity requires distributed trust

- An anonymity system can’t be just for one entity
- (even a large corporation or government)
- So you must carry traffic for others to protect yourself
- But those others don’t want to trust their traffic to just one entity either

Pseudospoofing: volunteers are a danger too

- Are half your nodes run by a single ba6 guy?
- Global PKD to ensure unique identities? No.
- Decentralize6 trust flow algorithms? Not yet.
- Still a major open problem for dynamic decentralized anonymity

Need to manage incentives well

Even customization and preferential service are risky (1)

- It’s tempting to let users choose security and robustness parameters

- Eg, how many replicas of my file should I create?
  or how many pieces should I break my file into?

- But a file replicated many times stands out.

An example: Directory servers

- Distribute location, capabilities, key info, performance stats
- A single directory server is a point of failure
- Redundant directory servers: must be (provably!) synchronized to avoid partitioning attacks
- Can distinguish between clients that use static lists and clients that update frequently

Directory servers (2)

Conclusion: we’re screwed

- Usability is a *security* objective: anonymity systems are nothing without users.
- It’s critical that we integrate privacy into the systems we use to interact.
- But it’s hard enough to build a killer app.
  It’s going to be really really hard to solve all the factors at once.
- Our current directions aren’t going to work, from an incentive and usability perspective. We need to rethink.

A point of light: Mixminion

- High-latency free-route mix network
- Fixes many of the problems with Mixmaster
-

Another point of light: synchro919w systems

- Each message has a deadline by which the node must pass it on
- Length of pathw iw fixed, pathw might even be public
- Anonymity iw now based on size of batch at widest point, even for free-route systems
- Improves flo35(de)-ing/trickle attacks
-

Privacy Enhancing Technologies workshop

March 26-28, 2003

Dresden, Germany

http://petworkshop.org/