Strong anonymity: is it really worth it?

Roger Dingledine

The Free Haven Project

Talk overview

Many people need anonymity

Political dissidents in oppressive countries

Governments want to do operations secretly.

Corporations are vulnerable to tra c analysis (corporate espionage) | VPNs, encryption don't cut it.

Individuals are tracked and pro led daily. Imagine what they'll have in your dossier in twenty years.

(If that doesn't scare you, think of your kids.)

Single-hop proxies

Most popular, easiest to deploy

Single point of failure (legal, technical)

Adversary characteristics

External (wires) or Internal (participants)

Passive or Active

Local or Global

Static or Adaptive

A MIX node

Messages change appearance after decryption

Each Td[(Each Tdebatchages)-41andch Tdereoeam(Messagen)]TJ

A MIX cascade

Free-route MIX networks

User picks a path through the network

Goal is to hide message's *path*

Needs dummy tra c (ine cient, poorly understood) to protect against global adversaries

Example: Mixmaster

Crowds: anonymous web browsing

\Blending into a crowd"

Users forward requests within their crowd

At each forward, with probability p to no(thir)-416memebr;r to to

Onion Routing

Connection-oriented (low latency)

Long-term paths between Coo416(p((Onion)6417(Rers5(w)-415(C

Zero Knowledge's Freedom Network

Connection-oriented (low latency)

Paid ISPs to run Freedom nodes

Tunnelled all tra c (udp, tcp, icmp | everything) through the Freedom netw

But anonymity is hard

Anonymity requires *ine ciencies* in computation, bandwidth, storage

Unlike encryption, it's not enough for just one person to want anonymity | the infrastructure must participate

Other people provide your anonymity (noise)

The more noise, the more anonymous se42ihing in that noise is

You're always better o going where the noise is

More users is good

High tra c) better performance

Better performance) high tra c

Attracts more users: faster *and* more anonymous

But trust bottlenecks can break everything

Strong anonymity requires distribited trust

Can we get volunteers to run nodes?

Liability, especially for exit nodes

Having lots of nodes might work, but making an example of a few well-chosen nodes can scare everybody

We can allow nodes to set individual exit policies

Pseudospoo ng: volunteers are a danger too

Are half your nodes run by a single bad guy?

Global PKI to ensure unique identities? No.

Decentralized trust ow algorithms? Not yet.

Still a major open problem for dynamic decentralized anonymity systems

Even customization and preferential service are risky

It's tempting to let users choose security parameters

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

But a le replicated many times stands out.

Even customization and preferential service are risky

We'd like to let clients customize to barter better, e.g. in systems like MNet

We'd like to let users pay (or pay more) for better service or preferential treatment

But the hordes in the coach seats are better o anonymity-wise than those in rst class.

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 well-synchronized to avoid partitioning attacks

Can distinguish between clients that use static lists and clients that update frequently

Directory servers (cont)

But even if uniform client information, nodes can still do trickle attack: hold message until other clients have di erent information.

Introducing reputation means adversary has new avenue to manipulate client information

Tension between giving clients accurate timely information, and preventing adversary from manipulating client behavior

Conclusion: we're screwed

Usability is a *security* objective: anonymity systems are nothing without users.

It's critical that we i7tegrate privacy i7to the systems we use to i7teract.

But it's hard enough to build a killer app.