

Anonymity

Christian Grothoff

`christian@grothoff.org`
`http://grothoff.org/christian/`

“You look at this and you say this is insane. It’s insane. And if it is only Hollywood that has to deal with this, OK, that’s fine. Let them be insane. The problem is their insane rules are now being applied to the whole world. This insanity of control is expanding as everything you do touches copyrights”

–Lawrence Lessig

Agenda

- Definitions and Metrics
- Techniques, Research Proposals and Systems
 - Dining Cryptographers, Mixes, Mixminion, PipeNet, Busses, Mute, Ants, StealthNet
 - **OFF**
 - P5, APFS, Crowds, Hordes

Off System

- Claims to be “anonymous”, but has no mechanisms
- Splits files in 128k blocks, XORs them for encryption (possibly also with a key)
- URLs specify which blocks (and keys) need to be XOR'ed for decryption
- Same encrypted block becomes part of many files

“Owner-Free refers both to the fact that nobody owns the system as a whole and nobody can own any of the data blocks stored in the system.” –Off Introduction

Our Legal System: What Colour are your bits?¹

- “In Paranoia, everything has a colour-coded security level (...) and everybody has a clearance on the same scale. You are not allowed to touch, or have any dealings with, anything that exceeds your clearance. If you’re a Red Troubleshooter, you’re not allowed to walk through an Orange door. Formally, you’re not really supposed to even know about the existence of anything above your clearance. Anyone who breaks the rules is a Commie Mutant Traitor, subject to the death penalty.”

¹<http://ansuz.sooke.bc.ca/lawpoli/colour/2004061001.php>

What Colour are your bits?

- “And very much of intellectual property law comes down to rules regarding intangible attributes of bits — Who created the bits? Where did they come from? Where are they going? Are they copies of other bits? Those questions are perhaps answerable by “metadata”, but metadata suggests to me additional bits attached to the bits in question, and I’d like to emphasize that I’m talking here about something that is not properly captured by bits at all and actually cannot be, ever. Let’s call it “Colour”, because it turns out to behave a lot like the colour-coded security clearances of the Paranoia universe.”

What Colour are your bits?

- “Random numbers have a Colour different from that of non-random numbers. (...) all cryptographers understand that it’s not the numbers that matter when you’re talking about randomness. What matters is where the numbers came from — that is, exactly, their Colour.”

Off System & Colour

- Off: Our bits are the XOR of two (coloured) bits, so they have lost their restrictions (no copyright)

Off System & Colour

- Off: Our bits are the XOR of two coloured bits, so they have no colour (no copyright)
- Law: Your bits are derived from two coloured bits, so you need **both** clearances

Crowds

M. Reiter and A. Rubin introduced *Crowds: Anonymity for Web Transactions*:

- in onion routing, the sender selects a path through the mix
- in Crowds, each member of the crowd chooses the next hop
- primary application is web-surfing

Crowds: features

- fewer public key operations than in a mix-net
- bi-directional communication (replies)
- efficiency and high scalability
- simple protocol

Crowds: non-goals

- no anonymity against local eavesdropper
- no responder anonymity
- no effort to defend against denial-of-service attacks (especially not against routers tampering with the indirected data)

Crowds: design

- node joins crowd (by signing up with central *blender* server), crowd forms one path with a single key for the entire path
- multiple, chained proxies, each proxy either exits or extends with probability $p_f > \frac{1}{2}$
- reply is routed back on the same path

Crowds: local eavesdropper

- there is no noise in the system
- ⇒ eavesdropper can see that a connection is initiated
- request is routed along static path with one session key
- ⇒ eavesdropper needs one node on the path for full exposure

Crowds: collaborating jondos

Suppose c out of n jondos are collaborating and p_f is the indirection probability. **Theorem 5.2:** If

$$n \geq \frac{p_f}{p_f - \frac{1}{2}} \cdot (c + 1)$$

the probability that the a collaborating jondo is the first node that the initiator connects to is lower than 50%.

Crowds: An attack²

The adversary may be able to deduce the initiator over time

- if an adversary controls one or more members of the crowd and
- if the protocol has multiple interactions between initiator and responder that can be correlated and that take different paths, since the initiator has a higher probability to be the sender of a query than all other nodes

²See also: M. Wright, M. Adler, B. Levine and C. Shields: *An Analysis of the Degradation of Anonymous Protocols*

Crowds: solution

- try to use only one static path
- paths must change when new jondos join
- *solution*: new jondos must join in groups, controlled by the central registration server

Crowds: scalability

Since the amount of traffic a node receives depends only on p_f , not on the size n of the crowd, scalability is great.

The requirement that all paths must be re-formed whenever nodes join is *much* worse, especially since the anonymity of the system depends on large crowds.

Crowds: choosing p_f

- The network load on an individual jondo does not change at all if that jondo changes the parameter p_f .
- Since the last jondo on a path must decrypt, it is optimal for CPU load to choose $p_f = 0$.
- If a jondo chooses $p_f = 1$, this is optimal for the rest of the crowd (jondo becomes a proxy!).
- If the jondo then additionally follows the Crowd requirement to indirect its own requests, they are trivially detectable and the jondo itself is exposed.

Crowds

Do you see any problems?

Tarzan

M. Freedman, E. Sit, J. Cates and R. Morris wrote *Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer*:

- generic Crowds implementation (tunnel)
- main difference from Crowds: onion routing style encryption
- goal: middleware (integrate into kernel!)

Hordes

C. Shields and B. Levine introduced a variant of Crowds:
A Protocol for Anonymous Communication Over the Internet.

- use the common Crowds protocol on the forward-path, but with link-to-link encryption
- use multicast (groups) for the response

⇒ faster response (no indirections)

⇒ less state in the jondos (really?)

Hordes

Do you see any problems?

P5: P2P Personal Privacy Protocol

R. Sherwood and B. Bahattacharjee describe P5, a *Protocol for Scalable Anonymous Communication* over the Internet. Notice that:

- P5 is **not** described as a true P2P system
- P5 uses mixes (onion routing) to achieve sender anonymity
- *broadcast* in P5 means application level broadcast

P5: claim to fame

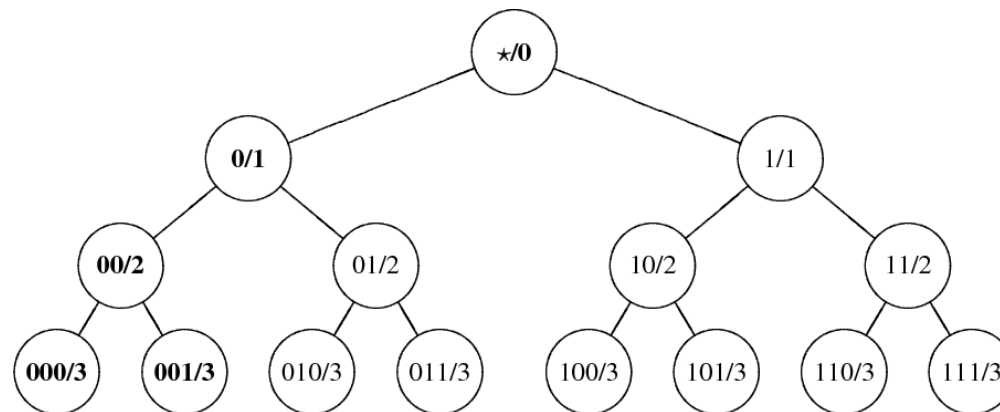
- sender-, receiver-, sender-receiver anonymity
- individual participants can trade-off anonymity for efficiency
- scalability

P5: modeled after global broadcast

- Initiator knows public key of the recipient
- Broadcast message encrypted with that key, all other nodes receive only garbage
- Initiator anonymity can be achieved by participating in a mix (see onion routing)

P5: broadcast tree

Instead of a global broadcast, broadcast groups (b/m) are used. A message to $(00/2)$ goes to the nodes shown in boldface:



b is chosen by hashing the user's public key, m randomly by the user under consideration of a local security policy.

P5: routing

- The broadcast tree is typically *not* used for communication
 - Peers create additional *routing keys*, use those for joining additional groups in the broadcast tree
 - Peers advertise the set of channels they have joined in their groups
- ⇒ Routing keys generate “lateral” edges in the tree, enabling DHT-style greedy routing

P5: Trade-offs

- Communicating parties may choose to give additional information about their choice of m to each other (after initial communication via $(b/0)$)
- Fixed amounts of noise are sent to mask activity, all packets have the same size

P5

Do you see any problems?

P5: Mob attack

- Peers can choose larger exposed value of m for efficiency
 - Malicious mob can join same channel as A and reduce efficiency of the channel
- ⇒ A might then expose additional bits of b , reducing anonymity

P5: Topology

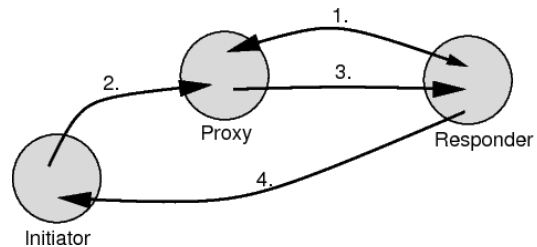
- “In our simulations, each user can determine the number of people in a group by consulting an *oracle* which maintains an up to date list of channel memberships”
- “In implementation, this information can be maintained in a secure distributed manner, either by the underlying application-layer multicast primitive, or at a well-known centralized *topology server*”

APFS: Responder Anonymity

V. Scarlata, B. Levine and C. Shields describe how to achieve responder anonymity in systems like crowds, hordes or onion routing in *Responder Anonymity and Anonymous Peer-to-Peer File Sharing*:

- two techniques for responder anonymity: with a proxy or with multicast
- Napster-style lookup-servers that are anonymous and dynamically created

APFS: proxy



1. Responder anonymously establishes connection to proxy
2. Initiator anonymously sends request to proxy
3. Proxy forwards requests to responder.
4. Responder anonymously establishes connection to initiator (optional, Horde-style multicast or without initiator anonymity)

APFS: multicast

1. Responder joins multicast group.
2. Initiator anonymously sends message to multicast group.
3. Responder anonymously responds (again Horde-style multicast or initiator not anonymous)

APFS: Napster revival

1. If we have responder anonymity, we can make nodes play the role of Napster servers.
2. If anonymity degrades over time, the servers can resign (and others can jump in).
3. A *coordinator* host keeps track of servers

APFS

Do you see any problems?

DC-Net

- Based on Dining Cryptographers
 - Instead of pairwise coin-flip, use secure PRNG with shared seed
 - Transmission slots and reservations protect against collisions
- ⇒ Best anonymity guarantees (for given n)
- ⇒ Traffic is $O(n^2)$ (n broadcasts/round)

CliqueNet / HerbivoreFS

E. Sirer, M. Polte and M. Robson designed *CliqueNet: A Self-Organizing Scalable, Peer-to-Peer Anonymous Communication Substrate*:

- based on DC-Nets
- more scalable, peer-to-peer and robust
- presumably supports TCP on top of CliqueNet
- Published 2004, to be released soon...

CliqueNet: goals

CliqueNet claims to achieve:

- **Strong Anonymity**, building on information-theoretic guarantees of DC-nets.
- **High Scalability**, no significant performance loss if more nodes join.
- **Robustness**, CliqueNet provides irrefutable, non-forgable proofs to identify disruptive nodes.

CliqueNet: The ideas

- automatically partition the network into smaller DC-nets of sizes between 3 and 5 participants
- some nodes, *ambassadors*, join multiple cliques for communication between cliques
- malicious hosts are detected and framed in a “distributed database”
- exponential back-off of round frequency if nothing is transmitted

Herbivore: details

- Pastry used for global organization
- Proof-of-work required for node to select virtual position
- Nodes maintain local (!?) *strike table* with misbehavior used to eliminate nodes from a clique
- Replication of documents used to defeat intersection attacks for file-sharing

CliqueNet / Herbivore

Do you see any problems?

Copyright

Copyright (C) 2010 Christian Grothoff

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.