The Pynchon Gate

L. Sassaman, B. Cohen, N. Mathewson
WPES’05

presented by B. Choi in cs6461
Computer Science
Michigan Tech
Motivations

• Pseudonymous mail retrieval
  – Users register their pseudonyms with “nym” server
  – The “nym” server can be part of or external to an anonymous communication system such as BABEL, Onion routing-based Mix-nets, Tarzan, ..
  – Nym server supports receiver anonymity
  – Either vulnerable to traffic analysis attacks or require a huge amount of cover traffic
  – PIR (private information retrieval) can be a solution
Goals

- Forward message security
  - Active and passive attackers
- Deployable and usable
  - Recruit many users!
Related work

- Chaum’81: reply blocks and return addresses
  - Time gap can lead to unreliability issues
  - Pseudonym management (multiple-use of reply blocks)
- Single-use reply blocks
  - Reliability issues still there
  - Intersection attacks possible
- Network level client anonymity
  - Widespread deployment in question: onion routing
Related Work

- Network-level server anonymity
  - Onion routing: rendezvous points
  - Sender doesn’t need to know receiver’s IP address
- Broadcast message and dead-drops
  - Send everything to everywhere
  - Scalability problem?
- Re-encryption mixes?
Pynchon Gate Overview

- Pynchon Gate: a group of servers
- Nym server receives emails for different pseudonyms
- Each cycle (24-hours) nym server passes the emails to collator.
- Collator batches them into indexed bucket pools and passes them to distributor nodes
- Distributors are independently operated (p2p)
Pynchon Gate Overview

- Pseudonym holder makes a series of requests to “k” chosen distributors
- Distributors cannot determine the pseudonym being requested
- Resistant to “k-1” collusion attacks
  - User identity to pseudonym
- Distributed-trust PIR-based message retrieval system!
  - Send everything to everywhere
Meta-index and pool bucket
Distributors and clients

- Independently operated (P2P)
- BitTorrent! (Bran Cohen)

Client
- Downloads the meta-index from a randomly chosen distributor
- Finds which index bucket to look at
- Downloads all the blocks form randomly chosen distributors (PIR)
- Repeats these up to a maximum volume
The PIR protocol

- Client retrieves a bucket from randomly chosen “K” distributors
Attacks

- Legal and hacking attacks
  - Dynamic key rotation
- Man-in-the-middle attack
  - TLS
- Replay attack
  - TLS
- Tagging and known-cleartext attack
  - TLS
Attacks

• Usage pattern and intersection attack
  – Hard to get usage pattern due to the cycle (24h)
  – Queries to distributors at a fixed interval
• Statistical disclosure against reply-block-based nym servers
System performance, scalability, optimizations

- Comparison to
  - Cypherpunk
  - Underhill
  - NNTP