

University of Maryland, College Park  
Dept. of Computer Science  
CMSC818K, Spring 2009

March 3, 2009  
Cody Dunne

M. Waldman, A. D. Rubin and L. F. Cranor

Publius: a robust, tamper-evident, censorship-resistant web publishing system

*SSYM '00: Proceedings of the 9th USENIX Security Symposium, USENIX Association, 2000, 59-72*

S. Hazel and B. Wiley

Achord: A Variant of the Chord Lookup Service for Use in Censorship Resistant Peer-to-Peer Publishing Systems

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

I. Clarke, O. Sandberg, B. Wiley and T. Hong

Freenet: A distributed anonymous information storage and retrieval system

*Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Springer Berlin / Heidelberg, 2001, 2009/2001, 46-66*

M. J. Freedman, E. Sit, J. Cates and R. Morris

Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

University of Maryland, College Park  
Dept. of Computer Science  
CMSC818K, Spring 2009

March 3, 2009  
Cody Dunne

M. Waldman, A. D. Rubin and L. F. Cranor

Publius: a robust, tamper-evident, censorship-resistant web publishing system

*SSYM '00: Proceedings of the 9th USENIX Security Symposium, USENIX Association, 2000, 59-72*

S. Hazel and B. Wiley

Achord: A Variant of the Chord Lookup Service for Use in Censorship Resistant Peer-to-Peer Publishing Systems

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

I. Clarke, O. Sandberg, B. Wiley and T. Hong

Freenet: A distributed anonymous information storage and retrieval system

*Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Springer Berlin / Heidelberg, 2001, 2009/2001, 46-66*

M. J. Freedman, E. Sit, J. Cates and R. Morris

Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

# Outline

- Overall Goals
- Design Goals
- Overview
- Secret Sharing
- Operations (Publish, Retrieve, Delete, Update)
- Implementation Issues
- Related Work
- Conclusion

## Overall Goals

- Highly resistant to censorship
- Provide publishers with high degree of anonymity

## Design Goals

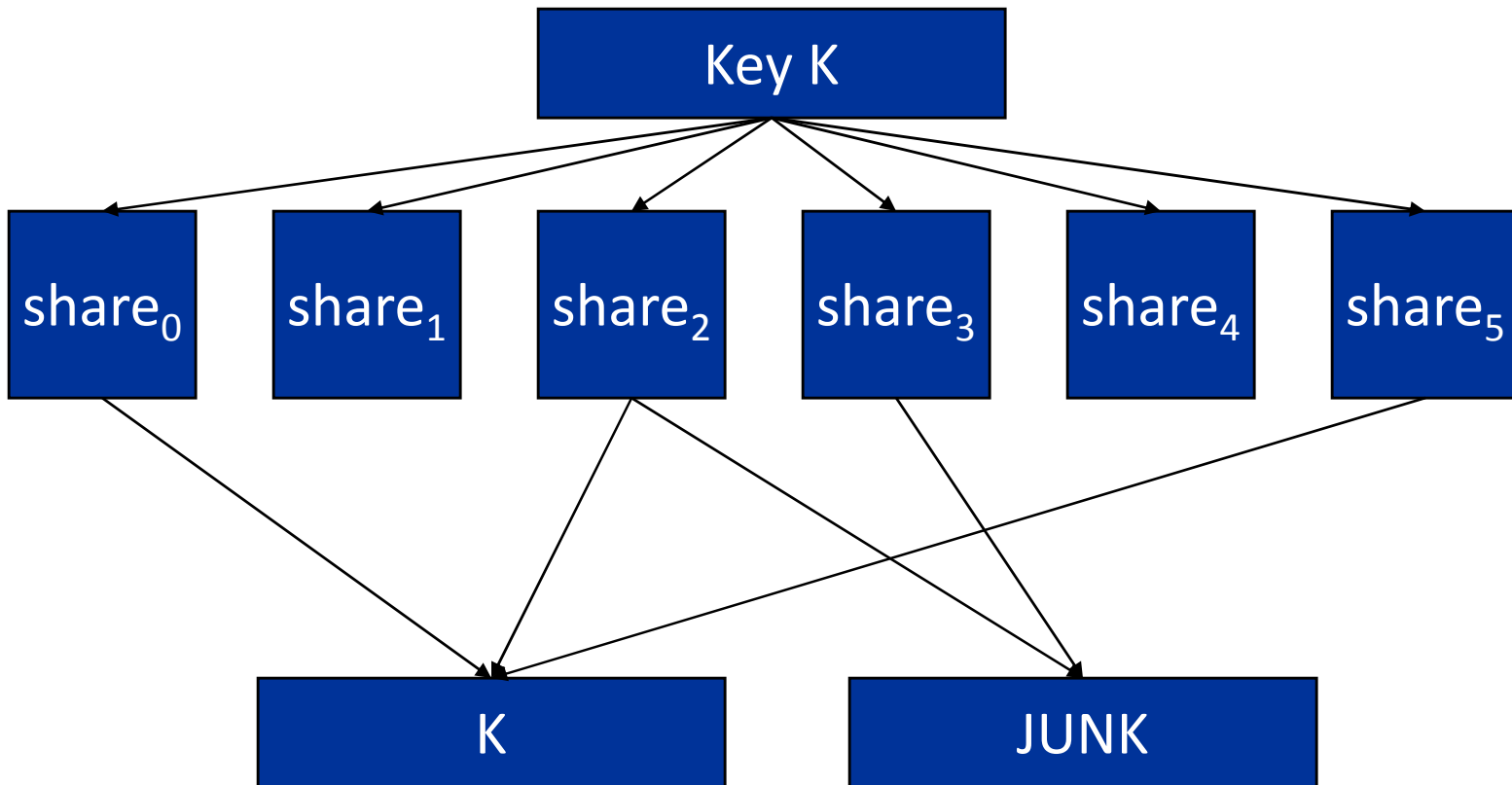
1. Censorship resistant (modification or deletion)
2. Tamper evident
3. Source anonymous (needs anonymous transport)
4. Updatable (or deletable)
5. Deniable (server)
6. Fault tolerant (malicious or faulty)
7. Persistent
8. Extensible (features or participants)
9. Freely available

# Overview

- Participants
  - *Publishers* post Publius content to the web
  - *Servers* host random-looking content (static list of size  $m$ )
  - *Retrievers* browse Publius content on the web
- State
  - System-wide list of servers
- Operations
  - Publish
  - Retrieve
  - Delete
  - Update
- Implementation
  - Proxy server for web browser
  - CGI script for server

# Secret sharing

A (3, 6) threshold scheme



## Secret sharing (Shamir)

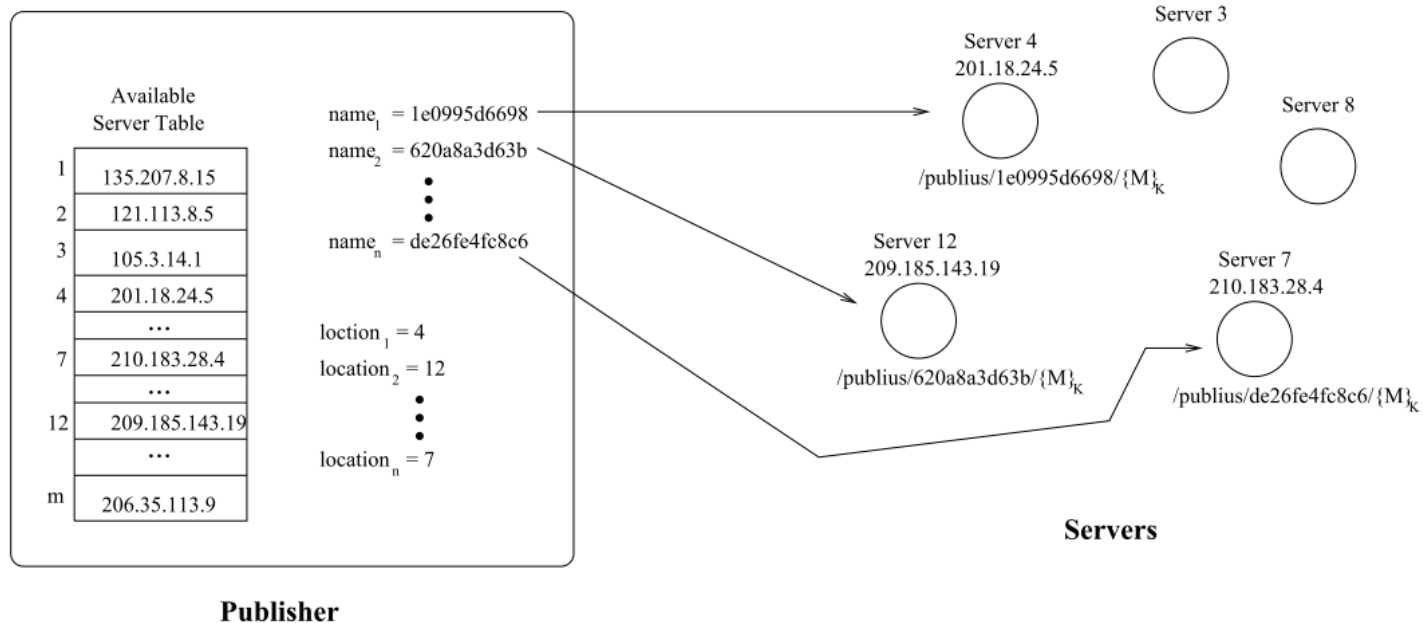
- $(k, n)$  threshold scheme to encrypt  $K$
- $n$  pieces of the key,  $k$  required to recreate  $K$
- Use a polynomial of  $k-1$  power
- $f(x) = a_0 + a_1x + a_2x^2 + \dots + a_{k-1}x^{k-1}$
- Set  $a_0 = K$  and pick  $a_1 \dots a_{k-1}$  randomly
- Give every participant a pair of input and output for  $f$
- Any group of  $k$  pieces is enough to interpolate  $f$

# Publish

- Make password file: password= md5(server\_domain\_name · PW))
- Generate random symmetric key K
- Encrypt message:  $\{M\}_k$
- Split K into n shares, share<sub>0</sub> -share<sub>n-1</sub>
  - Shamir's w/(k,n) threshold
- name<sub>i</sub> = wrap(md5(M · share<sub>i</sub>))
  - wrap = md5(xor of each half)
- location<sub>i</sub> = (name<sub>i</sub> mod m) + 1
  - m servers in the list
  - Need d unique values, or restart w/new K
  - d = minimum # unique servers for content
  - k < d < m
  - For each K create n = ceiling(d\*ln(d)) (optimal using Coupon Collectors approach)

## Publish (cont...)

- Use each location<sub>i</sub> as an index into the server list
- On the server, in directory name<sub>i</sub>, place {M}<sub>k</sub>, share<sub>i</sub>, password
- Given M, K, and m the locations of all shares are determined (URL)



## Retrieve

- Parse each  $\text{name}_i$  from URL
- $\text{location}_i = (\text{name}_i \bmod m) + 1$
- Choose  $k$  locations arbitrarily
- Choose one to get  $\{M\}_k$ , get each  $\text{share}_i$  from the rest
- Rebuild  $K$  & decrypt  $M$
- Test each  $\text{name}_i$  by redoing  $\text{name}_i = \text{wrap}(\text{md5}(M \cdot \text{share}_i))$  and comparing to URL
- Try different set  $k$  if you fail
- Alternate methods:
  - Try all  $n \cdot \binom{n}{k}$  combinations of shares and documents
  - Retrieve all  $n$  of the shares and use Gemmel & Sudan's method that accounts for some corrupt shares

## Delete

- Send password = md5(server\_domain\_name · PW))  
and name<sub>i</sub> to each server
- Compare to stored & delete if ok

## Update

- Change content w/o changing URL
- Publish new file and PW
- Send password = md5(server\_domain\_name · PW)), name<sub>i</sub>, and the new URL to each of the old servers
- Each creates update file with URL and deletes old file
- Subsequent retrieves of the old file return the new URL, so get >k of them and compare, then retrieve new URL
- Update flag in URL can specify non-updatable content

## Implementation Issues

- URLs specially encoded for proxy server, with list of all the name<sub>i</sub> concatenated w/options
- Mutually hyperlinked files need to be published in correct order
  - Get around cycles through multiple updates, modeled in DAG
- Eliminate extensions during publish and instead prepend them to file

## Related Work

- Almost all the related work mentioned has been monetized, bought up, closed, or was published and went nowhere
- Publius ceased development <2004

## Conclusion

- Pros
  - Relatively difficult to modify
  - Identity protected as long as external connection based anonymity is used
  - Simple proxy & CGI script tools available freely online
  - Very well written intro to the problem

## Conclusion (cont...)

- Cons
  - Deletion/Corruption attacks
    - Compromise of  $\geq n-k+1$  servers guarantees censorship
    - Similarly, (Clarke et al, '01) say, “since the identity of the servers themselves is not anonymized, an attacker might remove information by forcing the closure of  $n-k+1$  servers.”
    - Legal, DDOS, “Rubber-Hose Cryptanalysis” on publisher
  - Update attacks
    - Compromise of  $\geq k$  servers allows possible update file attack
    - $\geq n-k+1$  servers allows guaranteed update attack
  - DOS (fill servers)
    - Try to limit size, use HashCash or other CPU-based schemes, limit by IP address. Real \$\$\$ would be tough
    - IP address limits may not be possible with some anonymous transport services
  - No connection based anonymity

## Conclusion (cont...)

- Cons (cont...)

- Perpetual leakage / relies on  $m$  static servers

- If you allow for a dynamic system, you need a way to handle losing shares
      - Need to notice & tell others about missing/corrupt shares
      - Secret share refresh (Herzberg et al, '95)
    - Changing the number of servers,  $m$ , breaks all locations
      - $\text{location}_i = (\text{name}_i \bmod m) + 1$
    - Changing server domain name breaks all stored passwords
      - $\text{password} = \text{md5}(\text{server\_domain\_name} \cdot \text{PW})$
    - Long update chains for frequently updated files. With perpetual leakage, it would increase the danger of losing track of updates

University of Maryland, College Park  
Dept. of Computer Science  
CMSC818K, Spring 2009

March 3, 2009  
Cody Dunne

M. Waldman, A. D. Rubin and L. F. Cranor

Publius: a robust, tamper-evident, censorship-resistant web publishing system

*SSYM '00: Proceedings of the 9th USENIX Security Symposium, USENIX Association, 2000, 59-72*

S. Hazel and B. Wiley

Achord: A Variant of the Chord Lookup Service for Use in Censorship Resistant Peer-to-Peer Publishing Systems

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

I. Clarke, O. Sandberg, B. Wiley and T. Hong

Freenet: A distributed anonymous information storage and retrieval system

*Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Springer Berlin / Heidelberg, 2001, 2009/2001, 46-66*

M. J. Freedman, E. Sit, J. Cates and R. Morris

Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

# Outline

- Overview
- Design
- Restrictions
- Conclusions

## Overview

- Censorship resistance requires that each component is compatible
- The lookup algorithm must be scalable and have these properties:
  1. Insert w/o revealing identity
  2. Retrieve w/o identifying the requestor
  3. Difficult to get your node to hold a particular item
  4. **Difficult to identify the node(s) holding a particular item**

## Overview (cont...)

- Chord is attractive for distributed lookup
  - Provable performance and correctness
  - Guarantee finding an item if it exists with  $O(\log N)$  messages
- At odds with 4<sup>th</sup> property of censorship-resistance
  - Maps from keys to nodes
  - Network stabilization protocol looks at lots of nodes
- Achord is based on chord

## Design

- Prevent nodes from choosing ID
  - Assigned m-bit identity chosen in consistent way:  
sha1(IP\_address)
  - Provides for Property 3
- Limit each node's knowledge of the network to the  $O(\log n)$  node finger table
  - Provides some measure of Property 4

## Design (cont...)

- Chord: two methods for `find_successor`
  - Iterative: contact each node directly
  - Recursive: only contact node in finger table that most closely precedes target, and it recurses for you
- Achord severely restricts to certain cases
- Achord maps keys to values, instead of to nodes
  - `connect_to_successor` lookups based on recursive `find_successor`
  - Request: returns value through tunnel instead of node
  - Insert: pushes value to that node for insert
  - Provides some measure of anonymity for Properties 1 & 2
    - Can also use external measures

## Stabilization

- For speed & reliability, nodes must have their successors and predecessors set properly and their finger tables should be reasonably accurate
- Modified with new restrictions:

## Restrictions

- Only the node with ID  $n$  is allowed to call `find_successor(n)`
- Only iterative version used
  - At most  $O(\log N)$  nodes learn of our existence
- A node  $n$  is only allowed to access the predecessor of node  $n'$  if  $n$  was a previous value of  $n'.predecessor$ , and  $n$  hasn't accessed since it was changed from  $n$

## Restrictions (cont...)

- Finger tables updated using `find_best_match`, which returns new IP only if it is a closer match to one of the slots in  $n$  finger table than  $n'$ 
  - Limits IP harvesting by one node
    - Static: at most  $O(k \log N)$  IPs by any one node, where  $k$  is the finger table size
    - Dynamic: depends on join rate – all nodes find out about new ones over time

## Conclusions

- Pros
  - More anonymous than basic Chord
- Cons
  - Use lots of IPs to increase probability of getting a particular key
  - Node receiving a request can more accurately judge distance than in Freenet
  - Could use lots of IPs to harvest IPs faster

University of Maryland, College Park  
Dept. of Computer Science  
CMSC818K, Spring 2009

March 3, 2009  
Cody Dunne

M. Waldman, A. D. Rubin and L. F. Cranor

Publius: a robust, tamper-evident, censorship-resistant web publishing system

*SSYM '00: Proceedings of the 9th USENIX Security Symposium, USENIX Association, 2000, 59-72*

S. Hazel and B. Wiley

Achord: A Variant of the Chord Lookup Service for Use in Censorship Resistant Peer-to-Peer Publishing Systems

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

I. Clarke, O. Sandberg, B. Wiley and T. Hong

Freenet: A distributed anonymous information storage and retrieval system

*Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Springer Berlin / Heidelberg, 2001, 2009/2001, 46-66*

M. J. Freedman, E. Sit, J. Cates and R. Morris

Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

# Outline

- Goals
- Overview
- Conclusions

## Goals

- Anonymity for producers & consumers
- Deniability for servers
- Resistance to censorship
- Efficient dynamic storage & routing
- Decentralized

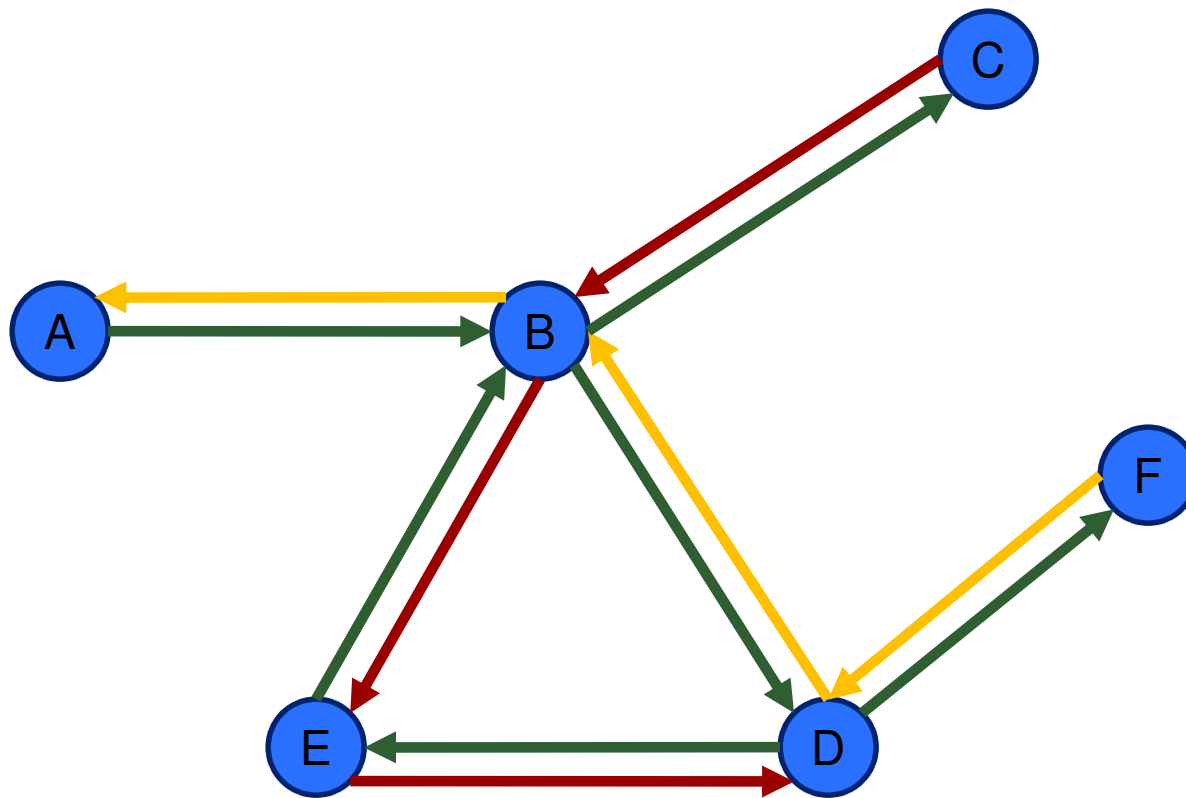
## Overview

- Network of identical nodes
  - Pool storage space
  - Route requests
- Infeasible to find origin or destination of transfer
- Difficult for node operator to determine or be held responsible for hosted content

## Overview (cont...)

- Chaum's Mixes for communication
- Globally Unique Identifiers for files (SHA-1)
- Steepest-ascent hill-climbing search
  - Ask your best guess from routing table, recursively
  - Once found, cache on the way back
- On file insert, routed towards GUID
  - Check if GUID exists, if not, write
  - Cached along the way

Demo of A requesting a file stored at F



# Conclusions

- Pros
  - Test implementation showed promise, time has proven
  - Large user base, still working & much improved
- Cons
  - No guarantees of data lifetime
  - Table 1

System	Attacker	Sender anonymity	Key anonymity
Basic Freenet	local eavesdropper	exposed	exposed
	collaborating nodes	beyond suspicion	exposed
Freenet + pre-routing	local eavesdropper	exposed	beyond suspicion
	collaborating nodes	beyond suspicion	exposed

**Table 1.** Anonymity properties of Freenet.

University of Maryland, College Park  
Dept. of Computer Science  
CMSC818K, Spring 2009

March 3, 2009  
Cody Dunne

M. Waldman, A. D. Rubin and L. F. Cranor

Publius: a robust, tamper-evident, censorship-resistant web publishing system

*SSYM '00: Proceedings of the 9th USENIX Security Symposium, USENIX Association, 2000, 59-72*

S. Hazel and B. Wiley

Achord: A Variant of the Chord Lookup Service for Use in Censorship Resistant Peer-to-Peer Publishing Systems

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

I. Clarke, O. Sandberg, B. Wiley and T. Hong

Freenet: A distributed anonymous information storage and retrieval system

*Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Springer Berlin / Heidelberg, 2001, 2009/2001, 46-66*

M. J. Freedman, E. Sit, J. Cates and R. Morris

Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

# Outline

- Overview
- Conclusions

# Overview

- Anonymous network layer – generic IP forwarding
- Each user runs client for local traffic and NAT
- Sequences of mix relays chosen from pool of users
  - Chosen randomly via Chord ring or based on criteria
  - Client picks chain & layers encryption
  - Establishes tunnel (control packet) then routes data through it (data packet)
  - NAT going in and going out for maximum anonymity

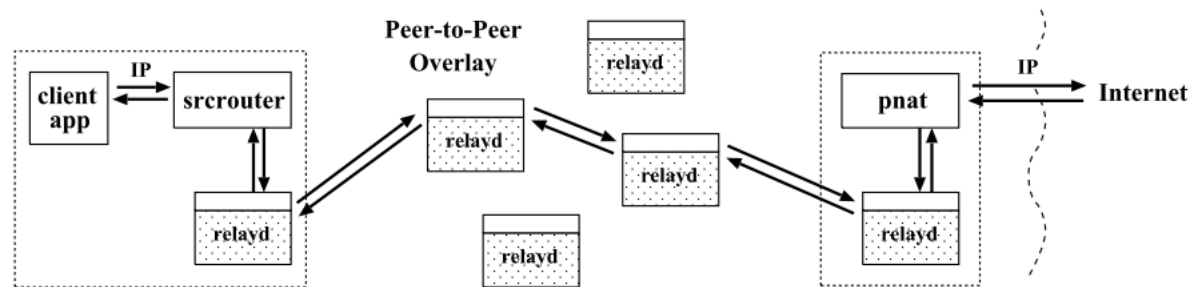


Figure 1: Tarzan Architecture Overview

# Conclusions

- Pros
  - Sender anonymity: can't determine if node is sending or relaying
  - Very low probability of choosing a fully compromised route
  - Prevents edge analysis
  - Reasonable tunnels quick to set up
- Cons
  - Built on UDP: best-effort
    - No reliability or authentication

University of Maryland, College Park  
Dept. of Computer Science  
CMSC818K, Spring 2009

March 3, 2009  
Cody Dunne

M. Waldman, A. D. Rubin and L. F. Cranor

Publius: a robust, tamper-evident, censorship-resistant web publishing system

*SSYM '00: Proceedings of the 9th USENIX Security Symposium, USENIX Association, 2000, 59-72*

S. Hazel and B. Wiley

Achord: A Variant of the Chord Lookup Service for Use in Censorship Resistant Peer-to-Peer Publishing Systems

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*

I. Clarke, O. Sandberg, B. Wiley and T. Hong

Freenet: A distributed anonymous information storage and retrieval system

*Designing Privacy Enhancing Technologies: International Workshop on Design Issues in Anonymity and Unobservability, Springer Berlin / Heidelberg, 2001, 2009/2001, 46-66*

M. J. Freedman, E. Sit, J. Cates and R. Morris

Introducing Tarzan, a Peer-to-Peer Anonymizing Network Layer

*IPTPS '02: Electronic Proceedings for the 1st International Workshop on Peer-to-Peer Systems, 2002*