CoNe Freiburg

Peer-to-Peer Networks

Anonymity
9th Week

Albert-Ludwigs-Universität Freiburg
Department of Computer Science
Computer Networks and Telematics
Christian Schindelhauer
Summer 2008

Motivation

- **Society**
  - Free speech is only possible if the speaker does not suffer negative consequences
  - Thus, only an anonymous speaker has truly free speech
- **Copyright infringement**
  - Copying items is the best (and most) a computer can do
  - Copyright laws restrict copying
  - Users of file sharing systems do not want to be penalized for their participation or behavior
- **Dictatorships**
  - A prerequisite for any oppressing system is the control of information and opinions
  - Authors, journalists, civil rights activists like all citizens should be able to openly publish documents without the fear of penalty
- **Democracies**
  - In many democratic states certain statements or documents are illegitimate, e.g.
    - (anti-) religious statements
    - insults (against the royalty)
    - certain sexual contents
    - political statements (e.g. for fascism, communism, separation, revolution)
- **A anonymizing P2P network should secure the privacy and anonymity of each user without endangering other users**

Attacks

- **Denial-of-Service Attacks (DoS)**
  - or distributed denial of service attacks (DDoS)
  - one or many peers ask for a document
  - peers are slowed down or blocked completely
- **Sybil Attacks**
  - one attacker produces many fake peers under new IP addresses
  - or the attacker controls a bot-net
- **Use of protocol weaknesses**
- **Infiltration by malign peers**
  - Byzantine Generals
- **Timing attacks**
  - messages are slowed down
  - communication line is slowed down
  - a connection between sender and receiver can be established
- **Poisoning Attacks**
  - provide false information
  - wrong routing tables, wrong index files etc.
- **Eclipse Attack**
  - attack the environment of a peer
  - disconnect the peer
  - build a fake environment

Cryptography in a Nutshelf

- **Symmetric Cryptography**
  - AES
  - Affine Cryptosystems
- **Public-Key Cryptography**
  - RSA
  - ElGamal
- **Digital Signatures**
- **Public-Key-Exchange**
  - Diffie-Hellman

- **Interactive Proof Systems**
  - Zero-Knowledge-Proofs
  - Secret Sharing
  - Secure Multi-Party Computation

Diffie-Hellman Key Exchange

- **Diffie, Hellman 1978**
- **Goal**
  - Exchange a secret key between two participants A and B while all communication is surveilled
- **Initiator A**
  - choose prime number p
  - and a generator g
    - i.e. 1, g, g^2, ..., g^{p-2} mod p are all different
  - publishes p and g
- **Key Exchange**
  - A picks a random number a
  - B picks a random number b
  - A sends $x=g^a$ mod p
  - B sends $y=g^b$ mod p
  - A computes $K=y^a$ mod p
  - B computes $K=x^b$ mod p
- **Both parties know $K = x^{ab}$ mod p**
- **Scheme relies on the difficulty to compute the discrete logarithm mod p**
  - Now A and B can use K as secret key for symmetric cryptography

Zero-Knowledge Proofs

- **Without revealing secret information it is possible to prove the knowledge of a fact to a partner**
- **For this an interactive protocol is used**
- **Two parties**
  - The prover
  - The verifier
- **Correctness of the protocol follows when the verifier can reproduce the results without the help of the prover**
- **There are Zero-Knowledge proofs for all problems in PSPACE**
  - „IP = PSPACE“, Adi Shamir 1992

Zero-Knowledge Proofs for Hamiltonian Cycles

- **Both parties know a graph G**
- **The prover P knows a hamiltonian cycle**
- **The verifier V asks P for a proof**
  - but P does not want show his solution
- **Perform the following rounds for some time until V is convinced**
  - P renumbers all nodes and produces a new graph H with the same edges and the corresponding hamiltonian cycle
  - P sends the new graph to V
  - V asks one of the following questions to P
    - Prove that G is isomorphic to H
      - i.e. the same but renumbered graph
      - then V sends the renumbering table
    - Proof that you have a Hamiltonian cycle in H
      - then V sends the new Hamiltonian cycle
- V checks the correctness of each case but does not know the Hamiltonian cycle in the original graph

Peer-to-Peer Networks

Secret Sharing
1979

Blakley‘s Secret Sharing

- **Geroge Blakley, 1979**
- **Task**
  - n persons have to share a secret
  - only when k of n persons are present the secret is allowed to be revealed
- **Blakley‘s scheme**
  - in a k-dimensional space the intersection of k non-parallel k-1-dimensional spaces define a point
  - this point is the information
  - with k-1 sub-spaces one gets only a line
- **Construction**
  - A third (trusted) instance generate for a point n in R^k k non-parallel k-1-dimensional hyper-spaces

Shamir's Secret Sharing Systems

- **Adi Shamir, 1979**
- **Task**
  - n persons have to share a secret s
  - only k out of n persons should be able to reveal this secret
- **Construction of a trusted third party**
  - chooses random numbers $a_1,...,a_{k-1}$
  - defines

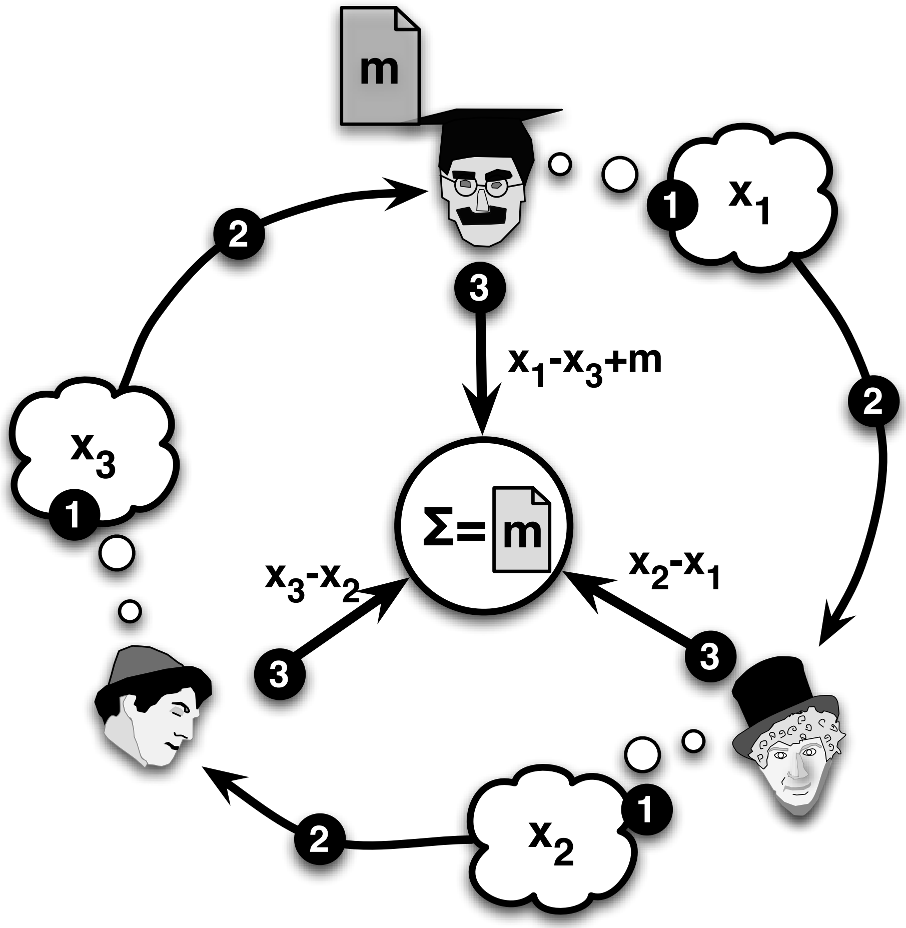
$$f(x) = s + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1}$$

  - chooses random $x_1, x_2, ..., x_n$
  - sends $(x_i, f(x_i))$ to player i
- **If k persons meet**
  - then they can compute the function f by the fundamental theorem of algebra
    - a polynomial of degree d is determined by d+1 values
  - for this they exchange their values and compute by interpolation
    - (e.g. using Lagrange polynoms)
- **If k-1 persons meet**
  - they cannot compute the secret at all
  - every value of s remains possible
- **Usually, Shamir's and Blakley's scheme are used in finite fields**
  - i.e. Galois fields (known from CRC)
  - this simplifies the computation and avoids rounding errors in the context of floating numbers

Chaum‘s Dining Cryptographers 1988

Dining Cryptographers

- **Anonymous publications without any tracing possibility**
- **$n \geq 3$ cryptographers sit at a round table**
  - neighbored cryptographers can communicate secretly
- **Each peer chooses secret number x_i and communicates it to the right neighbor**
- **If i wants to send a message m**
  - he publishes $s_i = x_i - x_{i-1} + m$
- **else**
  - he publishes $s_i = x_i - x_{i-1}$
- **Now they compute the sum $s=s_1+...+s_n$**
  - if s=0 then there is no message
  - else the sum of all messages

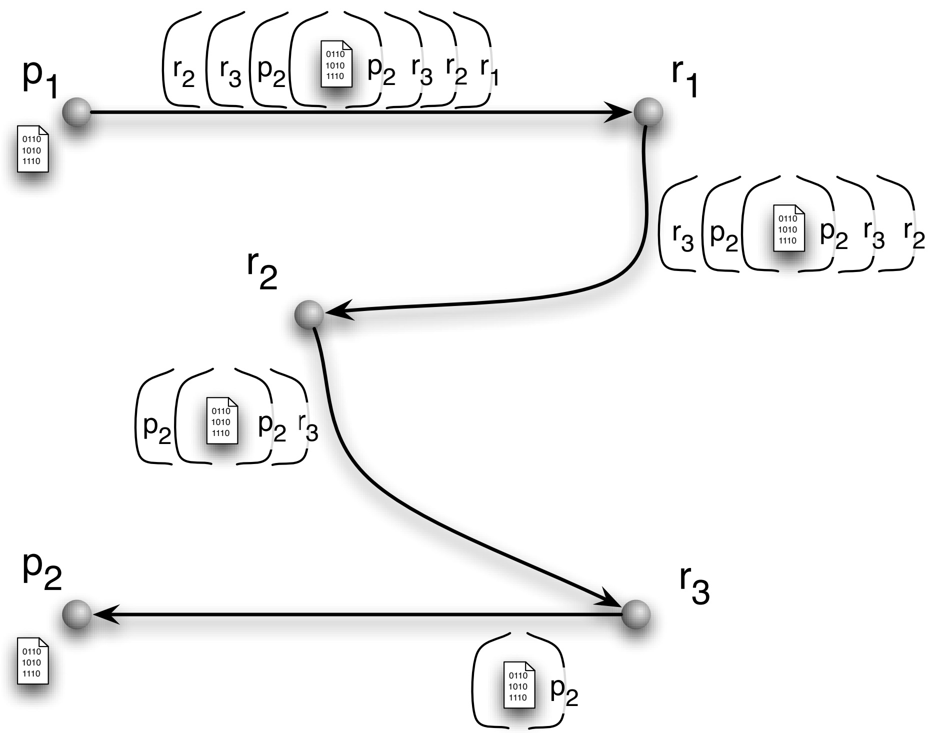


Peer-to-Peer Networks

Chaum Mixes
1981

Chaum's Mix-Cascades

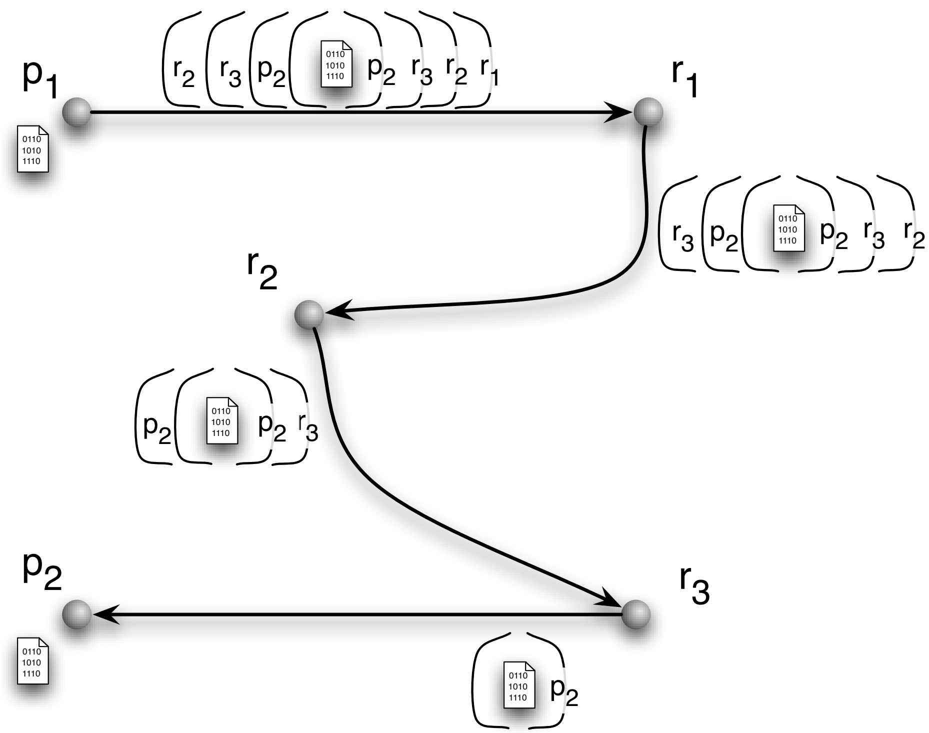
- **All peers**
  - publish the public keys
  - are known in the network
- **The sender p_1 now chooses a route**
  - p_1, r_1, r_2, r_3, ..., p_2
- **The sender encrypts m according to the public keys from**
  - p_2, ... r_3, r_2, r_1
  - and sends the message
  - $f(pk_{k1},(r_2,f(pk_{r2}...f(pk_{rk},(p_2,f(pk_{p2},m)))...)))))$
  - to r_1
- **r_1 encrypts the code, deciphers the next hop r_2 and sends it to him**
- **...**
- **until p_2 receives the message and deciphers it**



Computer Networks and Telematics
Albert-Ludwigs-Universität Freiburg
Christian Schindelhauer

Chaum‘s Mix Cascades

- **No peer on the route**
  - knows its position on the route
  - can decrypt the message
  - knows the final destination
- **The receiver does not know the sender**
- **In addition peers may voluntarily add detour routes to the message**
- **Chaum‘s Mix Cascades**
  - aka. Mix Networks or Mixes
  - is safe against all sort of attacks,
  - but not against traffic analysis



Peer-to-Peer Networks

Onion Routing
1998

TOR - Onion Routers

- **David Goldschlag, Michael Reed, and Paul Syverson, 1998**
- **Goal**
  - Preserve private sphere of sender and receiver of a message
  - Safety of the transmitted message
- **Prerequisite**
  - special infrastructure (Onion Routers)
    - all except some smaller number of exceptions cooperate
- **Method**
  - Mix Cascades (Chaum)
  - Message is sent from source to the target using proxies (Onion Routers)
  - Onion Routers unpredictably choose other routers as intermediate routers
  - Between sender, Onion Routers, and receiver the message is encrypted using symmetric cryptography
  - Every Onion Router only knows the next station
  - The message is encoded like an onion
- **TOR is meant as an infrastructure improvement of the Internet**
  - not meant as a peer-to-peer network
  - yet, often used from peer-to-peer networks

Other Work based on Onion Routing

- **Crowds**
  - Reiter & Rubin 1997
  - anonymous web-surfing based on Onion Routers
- **Hordes**
  - Shields, Levine 2000
  - uses sub-groups to improve Onion Routing
- **Tarzan**
  - Freedman, 2002
  - A Peer-to-Peer Anonymizing Network Layer
  - uses UDP messages and Chaum Mixes in group to anonymize Internet traffic
  - adds fake traffic against timing attacks

Peer-to-Peer Networks

FreeHaven
2000

Free-Haven

- **Dingledine, Freedman, Molnar, Sniffen Kamin 2000**
- **Goal**
  - Peer-to-Peer based Distributed Data Storage robust against attack from strong adversaries
  - Attacker tries to destroy data
- **Design**
  - Community of servers providing storage
  - Documents are stored using secret sharing on the servers
  - Document shares are exchanged between servers
  - For retrieval the shares are collected (using secured communication) from the storing servers
  - The addresses of the servers storing the data shares must be asked at a separate peer
    - which does not know the content nor the original keyword
  - All communication based on Onion Routing

Free-Haven

- **Operations**
  - Uploading documents
  - Downloading documents
    - preserving the authenticity of the documents
  - Expiration date for documents
    - until then a document is safe
    - then it must be erased
  - Adding new peer servers
  - Mechanisms to deteckt inactive or dead servers

- **Free-Haven**
  - provides anonymity to publishers, readers, servers and to the document
  - no query anonymity
    - others might know the topics a peer is interested
  - relies on trustable servers

Morphmix

- **Peer-to-Morphmix**
  - Rennhard, Plattner 2002
  - P2P-server system for anonymous web-surfing
  - encrypts with symmetric keys for efficiency
  - secret-key exchange with Diffie-Hellman public-key exchange using two additional nodes

- **Peer a uses peer b and w from its neighborhood to prepare connection between a and c**

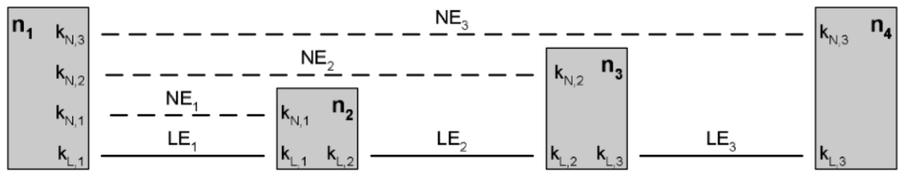


Figure 1: Layers of encryption.

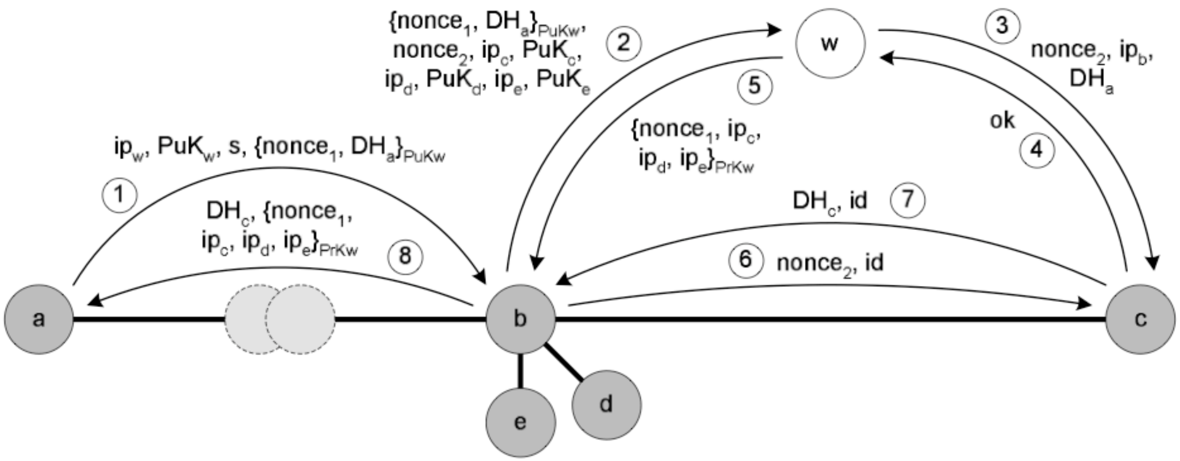


Figure 2: Setting up the nested encryption

Peer-to-Peer Networks

FreeNet
2000

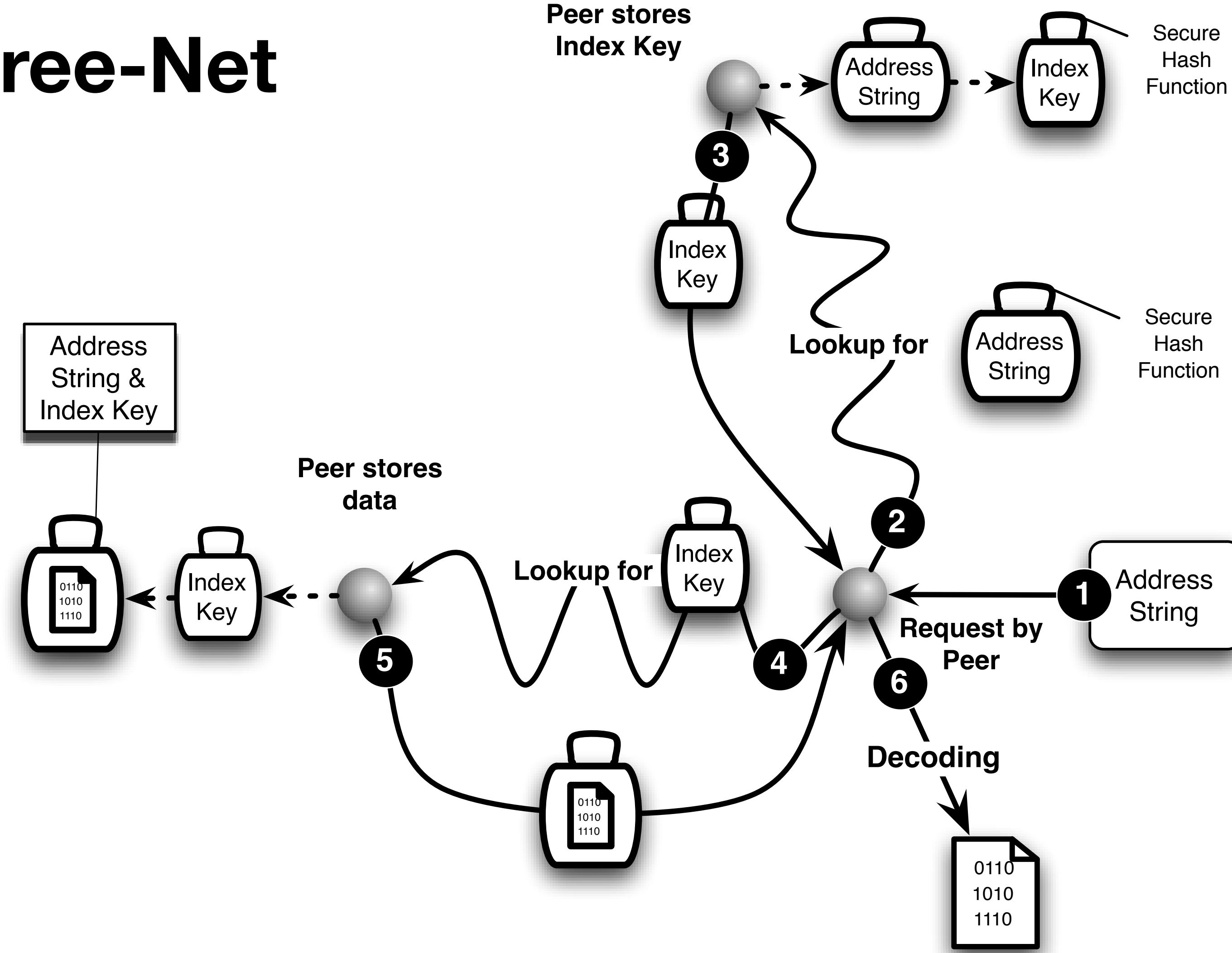
Free-Net

- **Ian Clarke, Oskar Sandberg, Brandon Wiley, Theodore Hong, 2000**
- **Goal**
  - peer-to-peer network
  - allows publication, replication, data lookup
  - anonymity of authors and readers
- **Files**
  - are encoding location independent
    - by encrypted and pseudonymously signed index files
    - author cannot be identified
  - are secured against unauthorized change or deletion
  - are encoded by keys unknown by the storage peer
    - secret keys are stored elsewhere
  - are replicated
    - on the look up path
  - and erased using “Least Recently Used” (LRU) principle

Free-Net

- **Network Structure**
  - is similar to Gnutella
  - Free-Net is like Gnutella Pareto distributed
- **Storing Files**
  - Each file can be found, decoded and read using the encoded address string and the signed subspace key
  - Each file is stored together with the information of the index key but without the encoded address string
  - The storage peer cannot read his files
    - unless he tries out all possible keywords (dictionary attack)
- **Storing of index files**
  - The address string coded by a cryptographic secure hash function leads to the corresponding peer
    - who stores the index data
      - address string
      - and signed subspace key
  - Using this index file the original file can be found

Free-Net


Peer stores
Index Key
Address
String
Index
Key
Secure
Hash
Function
3
Index
Key
Lookup for
Address
String
Secure
Hash
Function
Address
String &
Index Key
Peer stores
data
0110
1010
1110
Index
Key
Lookup for
Index
Key
2
1
Address
String
5
4
Request by
Peer
6
0110
1010
1110
Decoding
0110
1010
1110

Free-Net

- **Lookup**
  - steepest-ascent hill-climbing
    - lookup is forwarded to the peer whose ID is closest to the search index
  - with TTL field
    - i.e. hop limit
- **Files are moved to new peers**
  - when the keyword of the file is similar to the neighbor‘s ID
- **New links**
  - are created if during a lookup close similarities between peer IDs are discovered

Efficiency of Free-Net

- **Network structure of Free-Net is similar to Gnutella**

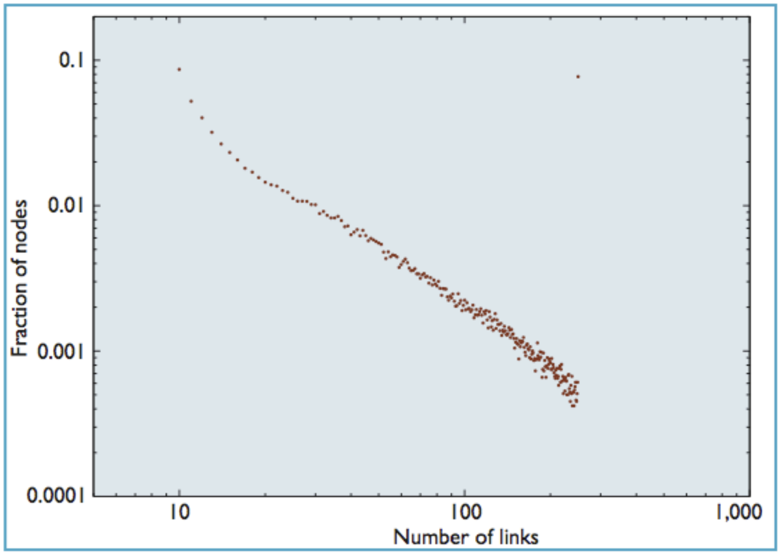


Figure 2. Degree distribution among Freenet nodes. The network shows a close fit to a power-law distribution.

- **The lookup time is polynomial on the average**

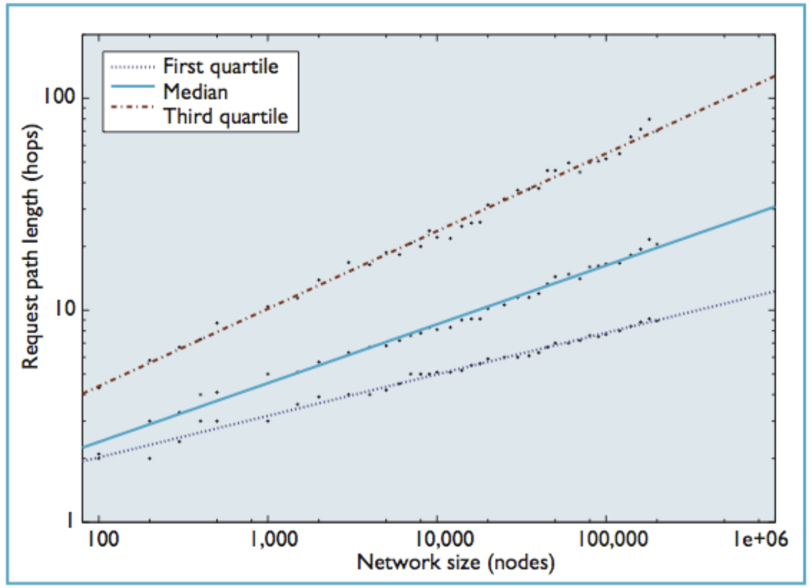


Figure 3. Request path length versus network size. The median path length in the network scales as $N^{0.28}$.

Dark-Net & Friend-to-Friend

- **Dark-Net is a private Peer-to-Peer Network**
  - Members can trust all other members
  - E.g.
    - friends (in real life)
    - sports club
- **Dark-Net control access by**
  - secret addresses,
  - secret software,
  - authentication using password, or
  - central authentication
- **Example:**
  - WASTE
    - P2P-Filesharing up to 50 members
    - by Nullsoft (Gnutella)
  - CSpace
    - using Kademlia

CoNe
Freiburg

Peer-to-Peer Networks

End of 9th Week

Albert-Ludwigs-Universität Freiburg
Department of Computer Science
Computer Networks and Telematics
Christian Schindelhauer
Summer 2008