PRIVACY OF IDEAS IN PEER 2 PEER NETWORKS

Speaker: Dieter Brunotte

Proseminar „Peer-to-Peer Information Systems“

Overview

- Motivation
- Privacy of Ideas
- Basic Technics for Liane
- Liane
- Experiments
- Summary

Motivation

- At the search you leak your information need
  - especially for very specific Information

- The Problem is that others get potential information about unpublished ideas for research

-> need for anonymized search engines

Leakage

- Unwanted revealing of an idea is called **leakage**
- while performing a query you don‘t hide the query itself
- The query can be analized to get the current idea of the user
- Queries without results are very interessant because the idea seems to be a new one

Privacy of ideas

- **Definition:**
  A service assures Privacy of ideas if it can be fully used while not leaking information that can be easily assembled for learning the current ideas of a user.

- This is not the same as Anonymity
- also anonymous user can leak the idea
- need a method that hides the query

Basic Procedure for Privacy

- For avoiding privacy leaks we will have to

- (1) split the query into small subqueries
  - Every Query should have at least one document within the collection.
- (2) decorrelate the subqueries in time
- (3) anonymize sending and receiving of each query result
  - For this we use Tarzan.
- (4) build a final result from the results of the subqueries.

Tarzan

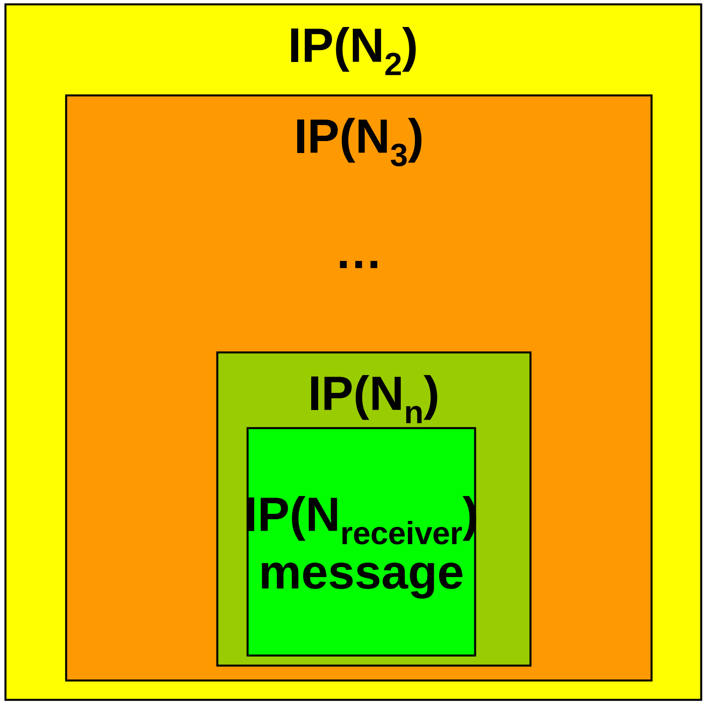
- offers anonymiced connections over P2P-networks using asymmetric encryption
  - every node has a public key

- a node N_{sender} who wants to send a message chooses n nodes $(N_1,...,N_n)$ from Tarzan network

- to send a anonymiced message the sender encrypts the message with the public keys of $N_1,...,N_n$ in n layers

Tarzan

- Tarzan message



IP(N_2)

IP(N_3)

…

IP(N_n)

IP($N_{receiver}$)
message

🟨 encrypted with pk of N_1

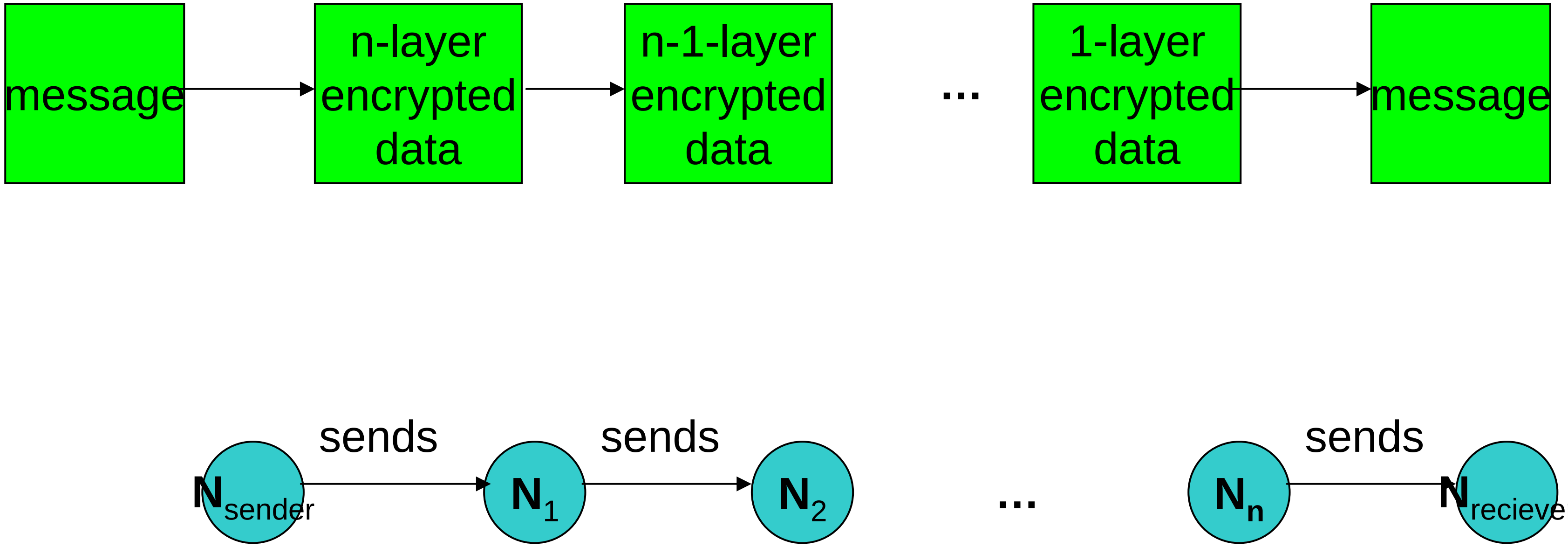
🟧 encrypted with pk of N_2

…

🟩 encrypted with pk of N_{n-1}

🟩 encrypted with pk of N_n

Tarzan


message
n-layer encrypted data
n-1-layer encrypted data
...
1-layer encrypted data
message
sends
sends
sends
N_{sender}
N_1
N_2
...
N_n
$N_{reciever}$

Tarzan: way back

- on the way back the same Tarzan-chain is used, but in the reverse direction
- every node adds an layer of encrytion
- decryption will be performed by N_{sender}

Inverted files

- „Bag of words“ assumption: Documents are sets of words
- for each word we have a list of document in which the word appears

  -> inverted files
- load the inverted files for the words of your query

Liane

- We combine Tarzan,Chord and inverted files to Liane
- Every inverted list of a word is stored at a Chord-Node
- for a query Q={q_1,q_2,...,q_m} the corresponding Chord-Nodes must be contacted to get the inverted files

Liane

- The queriing Peer must open |Q| anonymized connections using Tarzan to |Q| random-choosed Nodes

- This Peer perform the partial queries over the Chord-Ring to locate the inverted lists

- Result are the nodes containing the inverted list

- the inverted list are loaded again using Tarzan

Liane‘s weakness

- An attacker within the Liane network that owns many inverted lists can perform correlation attacks
  - theoretical attack
  - counter measure: dummy-queries, caching, ...

- Bigger Problem: Waste of resources
  - complexity of a Liane Query:

$$Comp_{dist} = O\left(c_{connect} \cdot |Q| \cdot \log N + c_{transfer} \cdot \sum_{\varphi \in Q} |InvListe(\varphi)| \right)$$

Optimization of Liane

- A query is not split into |Q| parts anymore
  - split the query in many subqueries with several terms
  - reduces size of the results of the subqueries
  - lower bandwidth
- low number of subqueries (many query terms) is a risk for leakage
- Optimization with cost model

Cost model for Liane

- We consider 2 main cost factors
  - c_{net} : cost of transferring a document reference over the network
  - c_{leak}: cost for the leakage of our idea

- We have costs of

$$c_{total}(Q) = c_{leak} \cdot P(leak) + c_{net} \cdot |RR|$$

- expressed as sum of subqueries

$$c_{total}(Q) = \sum_{j=1}^{m} c_{total}(Q_j)$$

Cost model for Liane

- We can compute P(leak)

$$P(leak_{Q_j}) = \left(1 - \prod_{q_i \in Q_j} \frac{|InvList\ (q_i)|}{|Coll.|}\right)^{|Coll.|} \approx e^{-|Coll.| \prod_{q_i \in Q_j} \frac{|InvList\ (q_i)|}{|Coll.|}}$$

- we get

$$c_{total}(Q_j) = c_{leak} \cdot e^{-|Coll.| \prod_{q_i \in Q_j} \frac{|InvList\ (q_i)|}{|Coll.|}} + c_{net} \cdot |Coll.| \cdot \prod_{q_i \in Q_j} \frac{|InvList\ (q_i)|}{|Coll.|}$$

Cost model

- Observations
  - the costs caused by leakage decreases exponentially with the number of documents found by the subquery
  - communication costs increase linearly with the number of documents found by the subqueries

- to minimize c_{total} we have to find a good Partitioning of the query into subqueries

Experiments: Setup

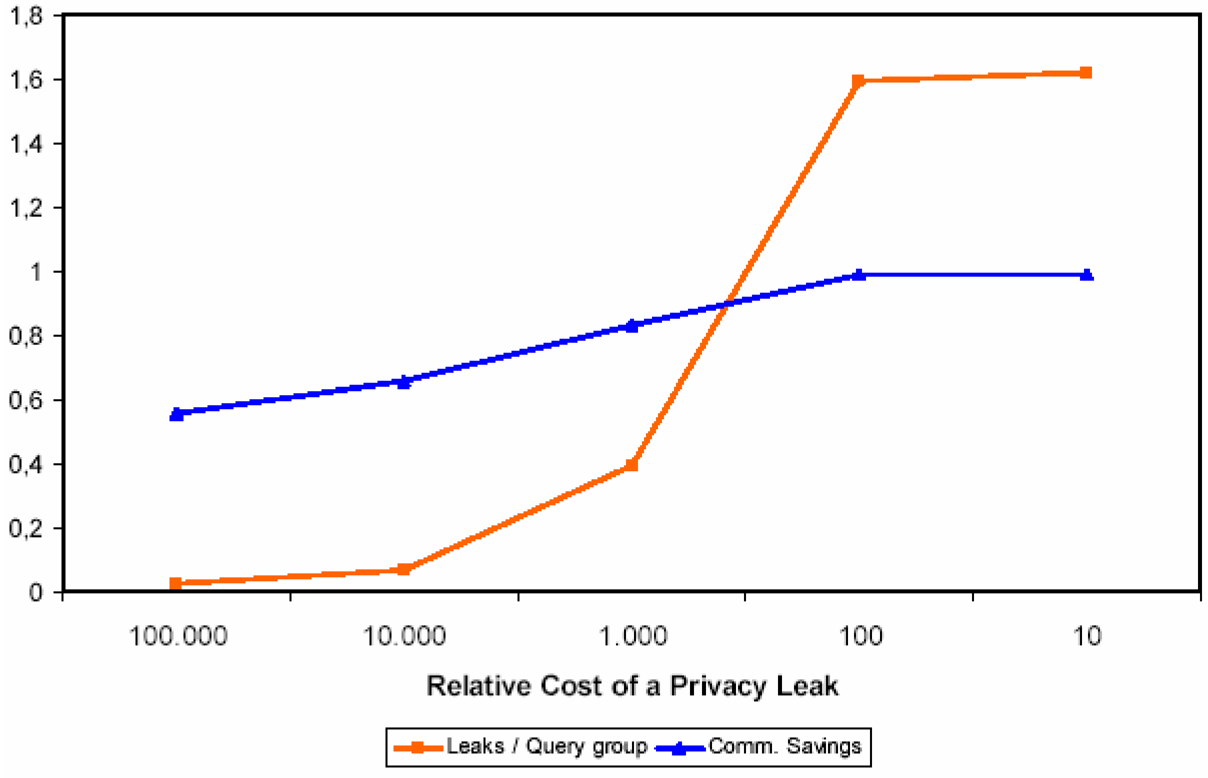
- Simulation of a PlanetP like P2P-network
- Document-Collection: 170000 News articles (Reuters Collection)
  - size of ~1-3 Kbyte per document
  - stopwords are removed
- The Queries contain:
  - n_k terms, that appear in at least k documents of the collection
  - n_{all} terms choosed from all terms of the collection

Experiments: Setup

- variation of k, n_{all}, n_k, c_{leak}/c_{net}
- use of a simple optimization-algorithm to divide the query into subqueries
- for every combination of values 1000 runs were averaged

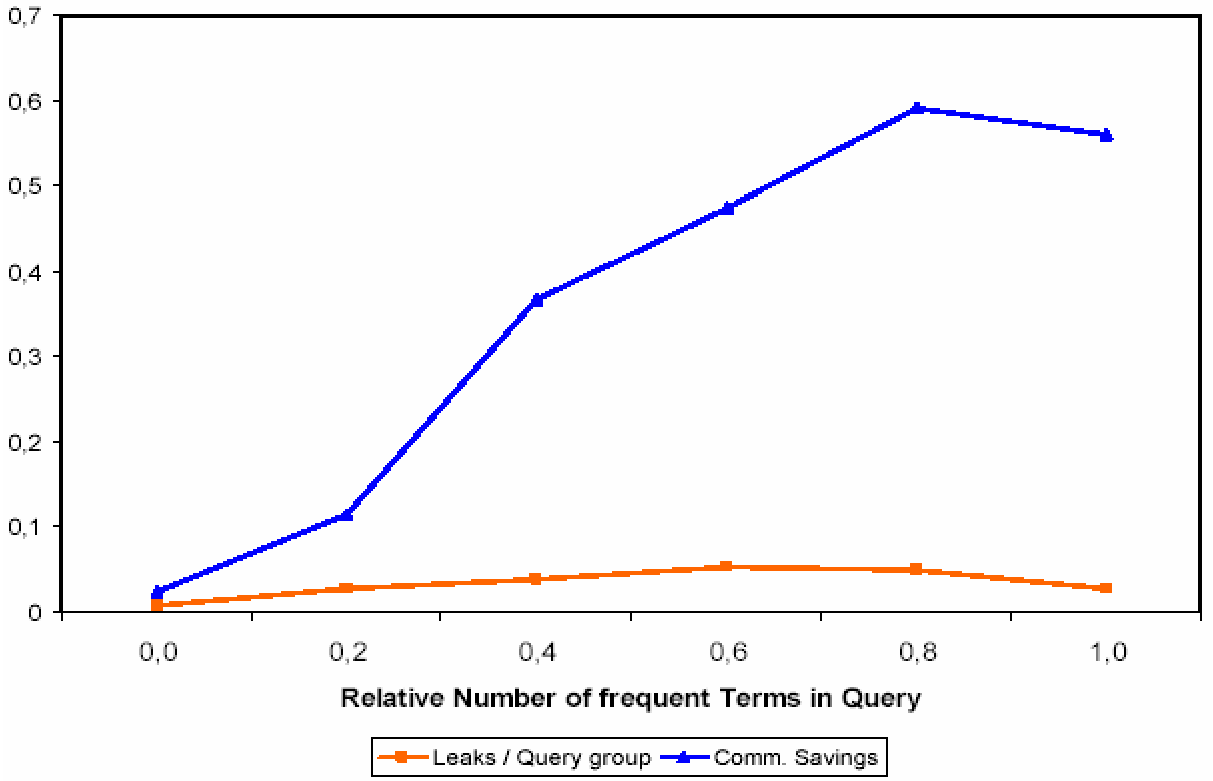
Variation of c_{leak}/c_{net}

- constant: k = 100, n_{all} = 0, n_k= 5, 5 query terms



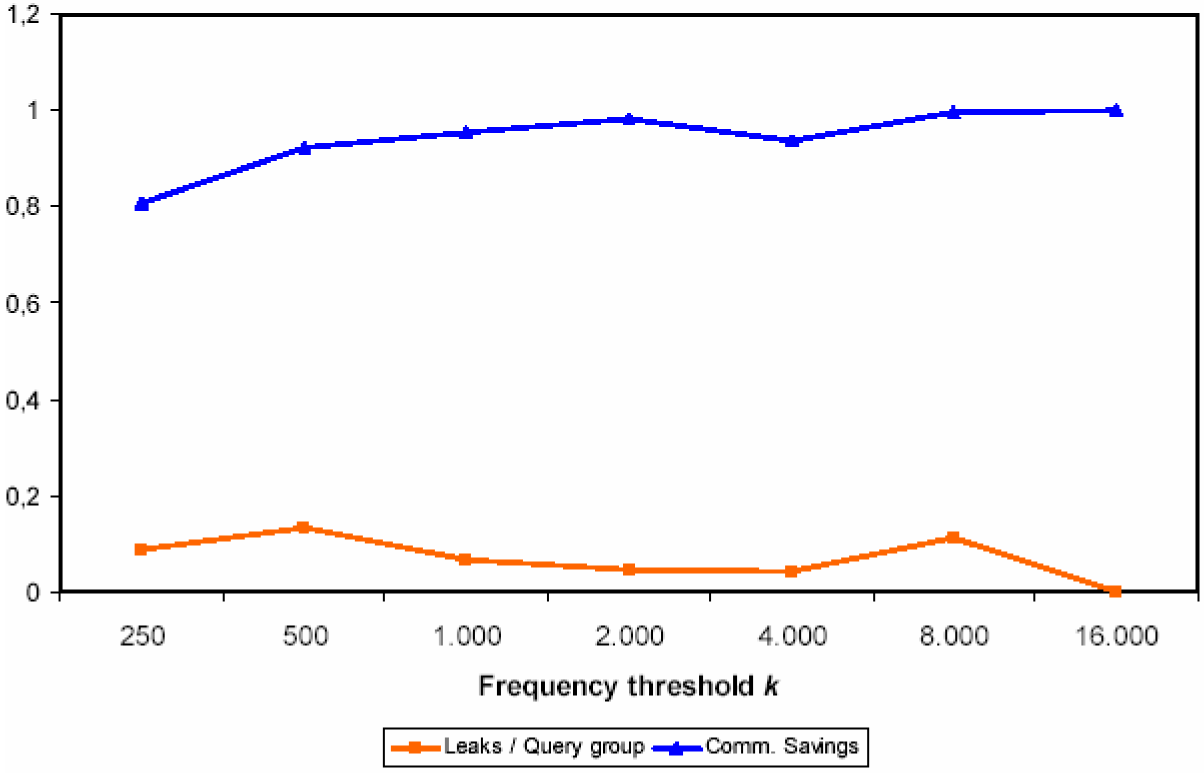
Variation of highfrequent terms

- constant: k = 100 , c_{leak}/c_{net} = 100000, 5 query terms



Variation of frequency-threshold k

- const.: $n_{all} = 0$, $n_k = 5$, $c_{leak}/c_{net} = 10000$, 5 query terms



Future work

- It misses experiments with real users to verify this definition of a new idea
- How many rare/high frequent terms are typical for a user query
- techniques for reducing the amount of data to send

Summary

- Privacy of ideas
  - definition of an new idea (empty query)
- Tarzan
  - anonymous data transmission
- Liane(=Tarzan + chord + inverted files)
  - System providing Privacy of ideas
  - Optimization with cost model
- Experiments on news article collection

End of Presentation

Questions !?

Thank for your attention!