4. Würzburger Workshop

Considering Security in Distributed Hash Tables

Heiko Niedermayer, Klaus Wehrle, Thomas Schreiner, Georg Carle
Wilhelm-Schickard-Institute for Computer Science
University of Tübingen

Übersicht

- Verteilte Hash-Tabellen (DHTs)
- Sicherheitsziele und Anwendungen
- Ansätze für Vertraulichkeit und Integrität
- Vergleich DHT-Routing <-> IP-Routing
- Angriffe auf DHTs
- Lösungen
- Zusammenfassung

Verteilte Hash-Tabellen (DHTs)

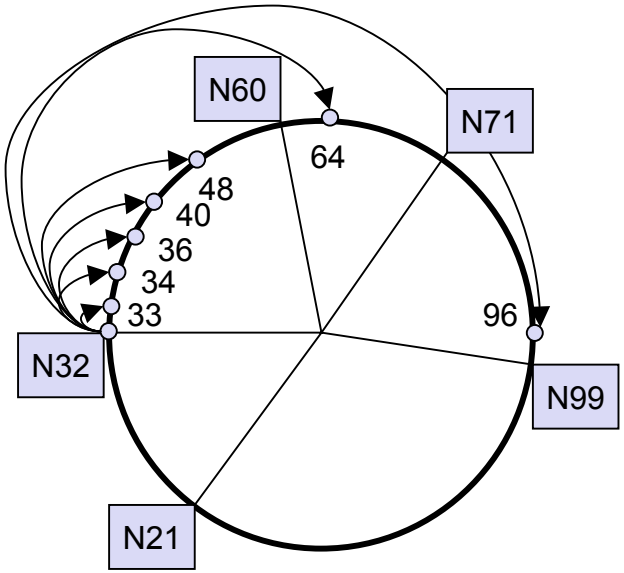
Peer-to-Peer-Netze

- verteilt
- autonom
- keine zentrale Instanz

Verteilte Hash Tabellen

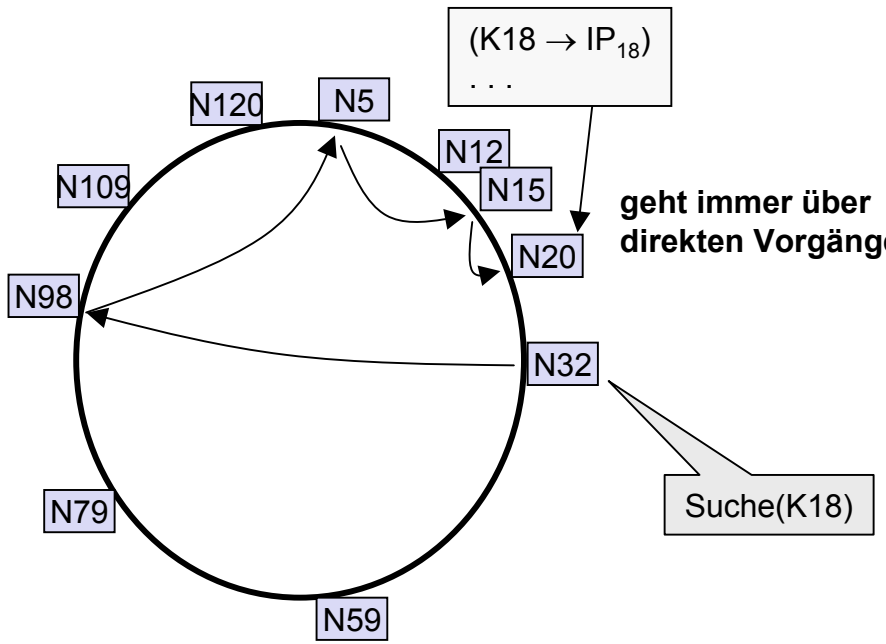
- Strukturierte Peer-to-Peer-Netze
- Zugriff über ID (Addressraum)
- speichert Key/Value-Paare
- i.a. O(log n) Hops zum Ziel
- Skalierbarkeit

Routing in Chord



Finger-Table		
i	**Ziel**	**Zeiger**
0	K33	N60
1	K34	N60
2	K36	N60
3	K40	N60
4	K48	N60
5	K64	N71
6	K96	N21

Bsp.: Suche nach K18 (M=7)



geht immer über direkten Vorgänger

Sicherheitsziele

Bisher

- Annahme: Trusted Web of Peers

Sicherheitsziele

- Vertraulichkeit
- Integrität
- Zurechenbarkeit
- Verfügbarkeit
- Kontrollierter Zugang
- Privatheit (Anonymität)

Jetzt

- Sicherheitsziele (gefordert?, erreicht?) mit nichtvertrauenswürdigen Peers

Sicherheitsziele und Anwendungen

Filesharing

- Integrität
- Verfügbarkeit

Verteiltes Dateisystem

- Vertraulichkeit
- Integrität
- Zurechenbarkeit
- Verfügbarkeit
- Kontrollierter Zugang

Instant Messaging

- Verfügbarkeit
- Kontrollierter Zugang

Klassische Sicherheitsziele und Ansätze I

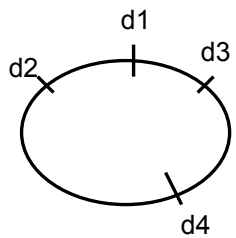
Vertraulichkeit

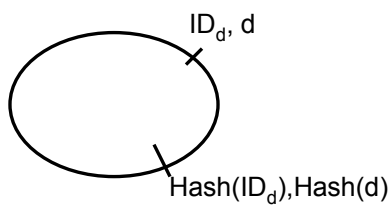
- gegenüber dem Internet
  - mögliche Ansätze:
    - SSL
    - IPSec
    - Verschlüsselung auf DHT-Ebene
- gegenüber der DHT
  - mögliche Ansätze:
    - Daten verschlüsselt ablegen
    - Daten verteilt ablegen (Secret Sharing), erst mit anderen Teilstücken zusammen ergeben Daten Sinn
    - Problem der Schlüsselverteilung und der Wahl geeigneter Verfahren

Klassische Sicherheitsziele und Ansätze II

Integrität

- Integrität der Nachrichten
  - mögliche Ansätze:
    - Message Authentication Codes (z.B. IPSec, SSL)
    - Digitale Signaturen
- Integrität der Daten
  - mögliche Ansätze:
    - Secret Sharing
      brauche k aus m Fragmenten zur Rekonstruktion der Daten,
      weitere Fragmente zur Überprüfung, Prüfsumme
    - Daten und Prüfsumme an verschiedenen Stellen ablegen, z.B.
      - Daten an ID
      - Prüfsumme an Hash(ID)
        - » Prüfsumme am besten mit kryptographischer Hashfunktion, z.B. SHA, MD5,..





IP vs DHT

IP	DHT
Addressraum durch Infrastruktur und zentrale Instanz gegeben	Knoten kann prinzipiell jede ID bekommen und an entsprechender Stelle im Netz positioniert werden
um an bestimmte Position zu kommen => geeignete Rechner müssen übernommen werden	um an bestimmte Position zu kommen => geeignete ID wählen
Routing über dedizierte Hardware der Provider	Routing über Clients, verwendet IP-Routing von Hop zu Hop

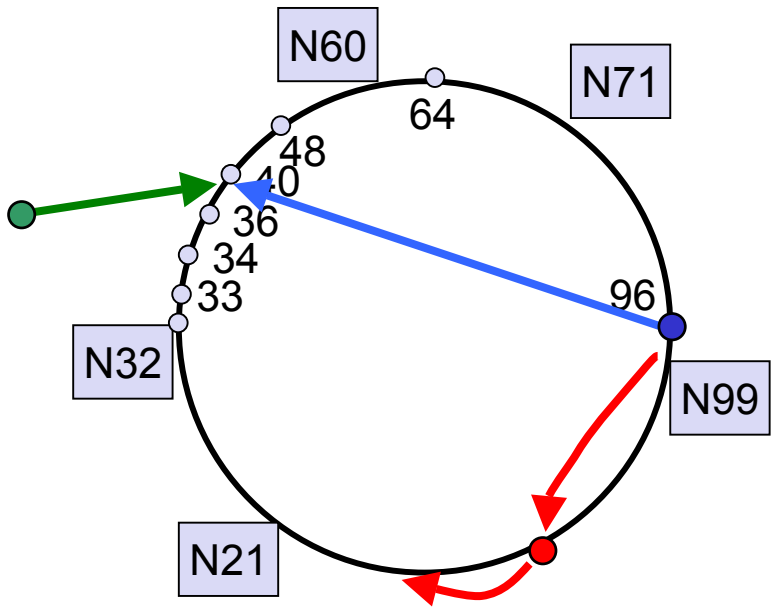
Angriffe auf DHTs

Senden falscher Nachrichten

- nicht zur DHT gehörender Knoten kann Nachrichten schicken
- Knoten in der DHT kann falsche Nachrichten schicken

Weitere aktive Angriffe

- Modifizieren von Nachrichten
- Löschen von Nachrichten
- Fehlerhaft Antworten
- Abhören von Nachrichten

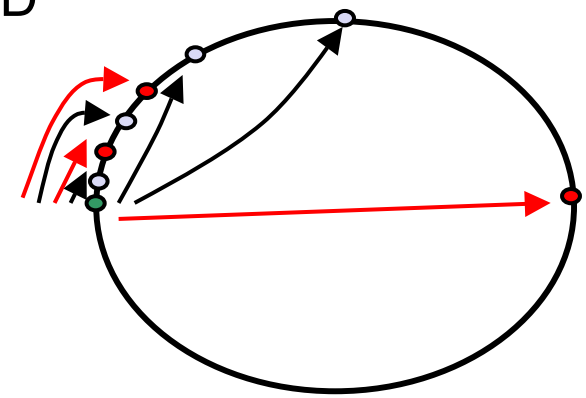


Knoten muss auf Pfad liegen
⇒ Angriff auf Routing?

Angriffe mittels DHT-Routing

Angriff auf Finger-Tabelle eines fragenden Knotens

- Annahme: ID des Knotens bekannt
- Einträge ergeben sich nach Regeln aus Knoten-ID
- an entsprechenden Stellen im ID-Space kann Angreifer Knoten positionieren
  => korrektes Routing = Routing über Angreifer



Lösungsansätze auf Client-Seite:

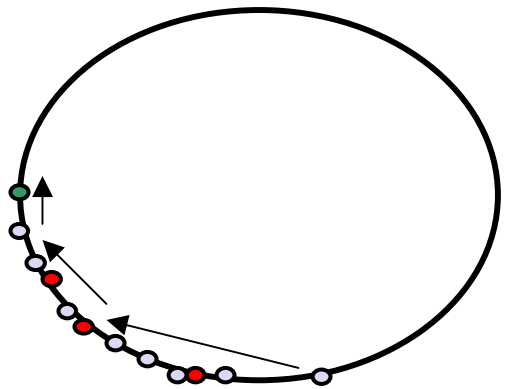
- Finger-Tabelle nicht exakt nach Vorschrift
  - unter Knoten mit ähnlicher ID den auswählen, dessen IP sich maximal möglich von den IPs anderer Einträge unterscheidet
  - ähnlich Proximity Routing
- Randomisierung des 1. Hops
  - Annahme: Angreifer besetzt nicht alle Einträge oder will gezielt einen Weg angreifen

Angriffe mittels DHT-Routing

Angriff auf Zielknoten mit bestimmter ID (Inhalt)

- bei Chord: alle Nachrichten gehen über Angreifer, wenn sich dieser als **Vorgängerknoten** positioniert
- allgemein: je näher am Zielknoten, desto höher die Wahrscheinlichkeit, dass Nachrichten über Angreifer gehen

Je näher am Ziel
=> desto kleiner der Fortschritt
im ID-Space
=> desto weniger potenzielle Knoten
für Weiterleitung



(Lösungs)-Ansatz für anfragenden Knoten

- Randomisierung des 1. Hops verhindert gezielten Angriff auf bestimmten Pfad

Lösungsmöglichkeiten zum Angriff auf DHT-Routing

Lösungsideen auf Ebene der DHT

- **Einschränkung der Wahlfreiheit der ID**
  - Überprüfung der Beziehung IP -> ID bzw. IP + Port -> ID
  - schwerer für Angreifer sich geeignet zu platzieren
- **Verhinderung mehrerer Instanzen**
  - mittels Login über zentrale Instanz
  - durch feste Bindung der ID an die IP
    - Annahme: Wahlfreiheit der IP nur im lokalen Netzbereich
    - Problem: NAT
      - Abschwächung: kleinen Bereich von Ports zulassen und Ports miteinbeziehen

Wie finde ich jemanden in DHT?

Jemand = IP?

= IP-Bereich?

Einfach

- Rechner fragen, welche ID er hat (wenn dies nicht verhindert wird)
- ID aus IP des Knotens berechnen (wenn ID sich direkt aus IP ergibt und diese bekannt ist)

Schwieriger

- durch Mithören (wenn Angreifer entsprechenden Rechner kontrolliert)
- durch Scannen der DHT

Kartierung einer DHT

Ziel: Zuordnung von ID und IP

Algorithmus

Algorithm 1 Map Generation

1: $n_a \leftarrow nodeID(attacker)$
2: $k \leftarrow n_a + 1$
3: **while** $respondingnode \neq n_a$ **do**
4: $\quad respondingnode \leftarrow lookup(k)$
5: $\quad$ print 'Node found: ', $respondingnode$
6: $\quad k \leftarrow respondingnode + 1$
7: **end while**

- Aufwand in DHT: O(n log(n))
- in kleinen und mittleren DHTs oder für begrenzte Adressbereiche durchführbar

Zusammenfassung

IP	DHT/Overlays
Adressen gegeben durch Infrastruktur	Adressen unabhängig von Infrastruktur
eingeschränkte Wählbarkeit der IP	ID prinzipiell beliebig wählbar
feste Position im Netz	beliebige Position im Netz

Take-Home-Message

- DHTs haben eigene Sicherheitsprobleme wegen
  - Dezentralität
  - Unabhängigkeit vom Infrastrukturnetz
  - Knoten kann beliebige Position im Netz haben
- Anwendungsentwickler müssen sich darüber im klaren sein und ggf. Lösungen entwickeln und implementieren

Ende

Fragen?