Exakte Mustersuche mit

SYMMETRIC SEARCHABLE ENCRYPTION

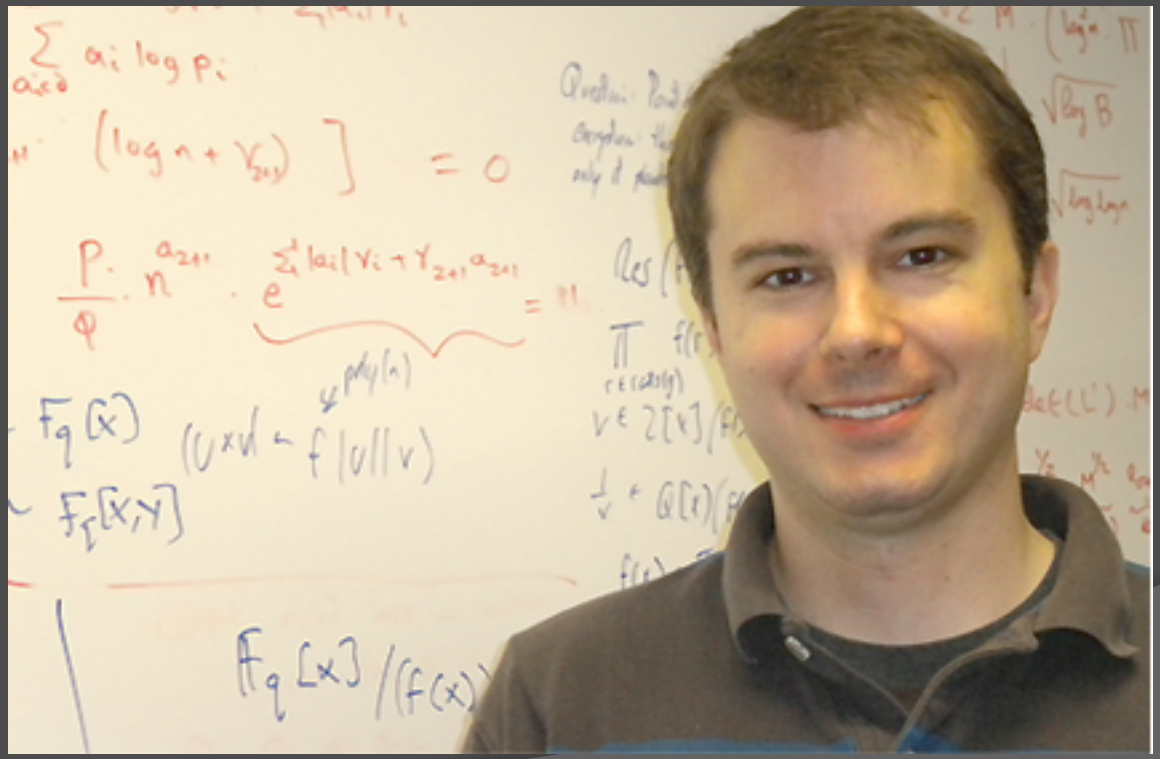
Rolf Haynberg | bob@aliceandbob.com | GPN 13 | 30.05.2013

- Dirk Achenbach
  - Institut für Kryptographie und Sicherheit (IKS)
  - Karlsruhe Institute of Technology (KIT)

- Jochen Rill
  - Cand. Dipl.-Inform.
  - Karlsruhe Institute of Technology (KIT)

Craig Gentry 2009

- Fully homomorphic encryption using ideal lattices



Vielen Dank für Eure Aufmerksamkeit!

Spielverderber: Performanz

Spezialisierung

- Partially Homomorphic Encryption
- Private Information Retrieval (PIR)
- Oblivious RAM
- MPC
- ...
- Searchable Encryption
  - Keyword-Based-Approach
  - Index-Based-Approach
  - ...
  - Exact Pattern Matching (Volltextsuche)

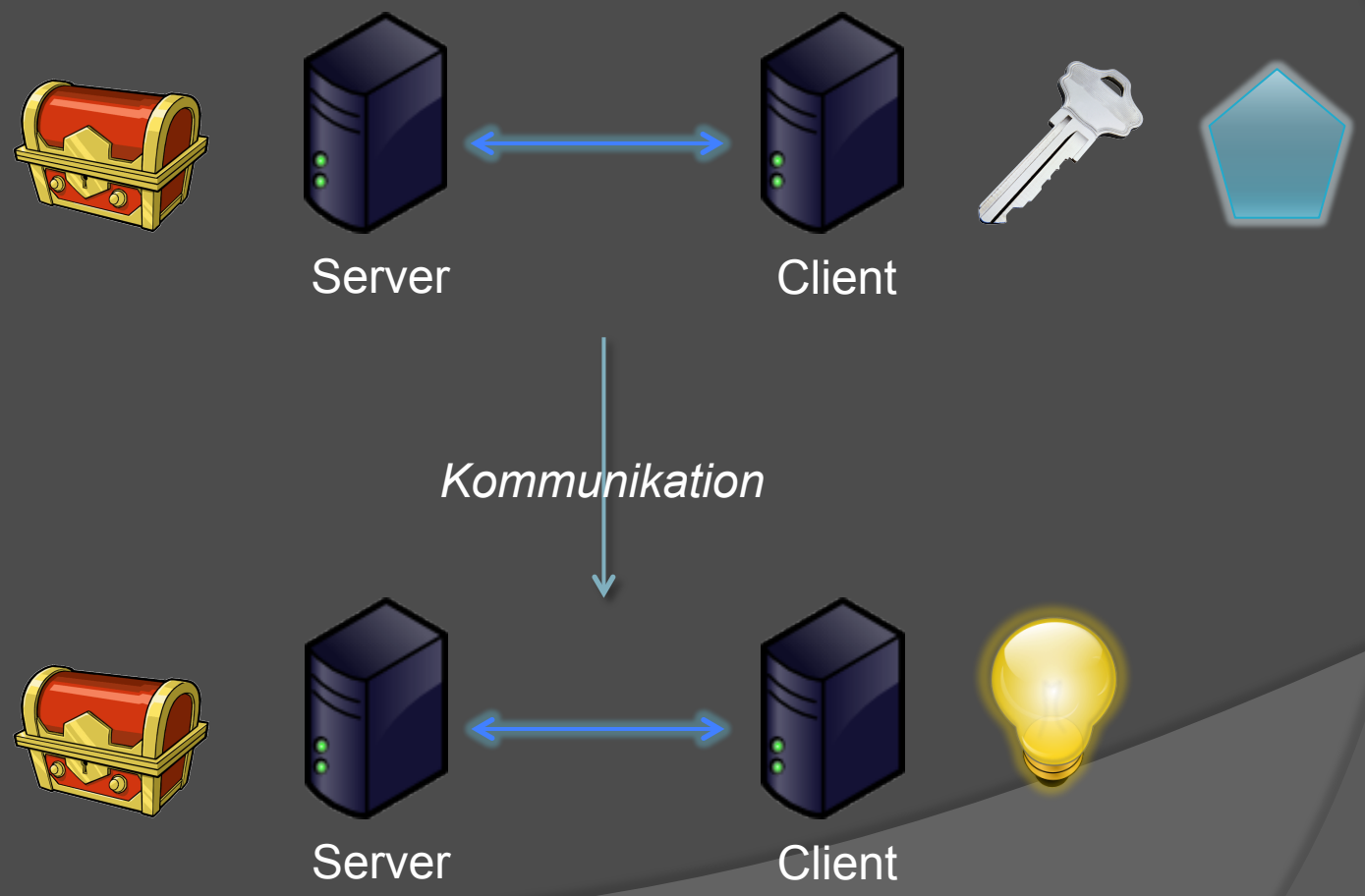
Agenda

1. Definition eines neuen kryptografischen Problems
2. Konstruktion einer Lösung
3. Ergebnisse / Benchmarking

Definition der Problemstellung

- Definition Volltextsuche
- Festlegung Angreifermodell
- Festlegung Sicherheitsbegriff
  - Pattern-Privacy
  - Data-Privacy
  - Result-Privacy

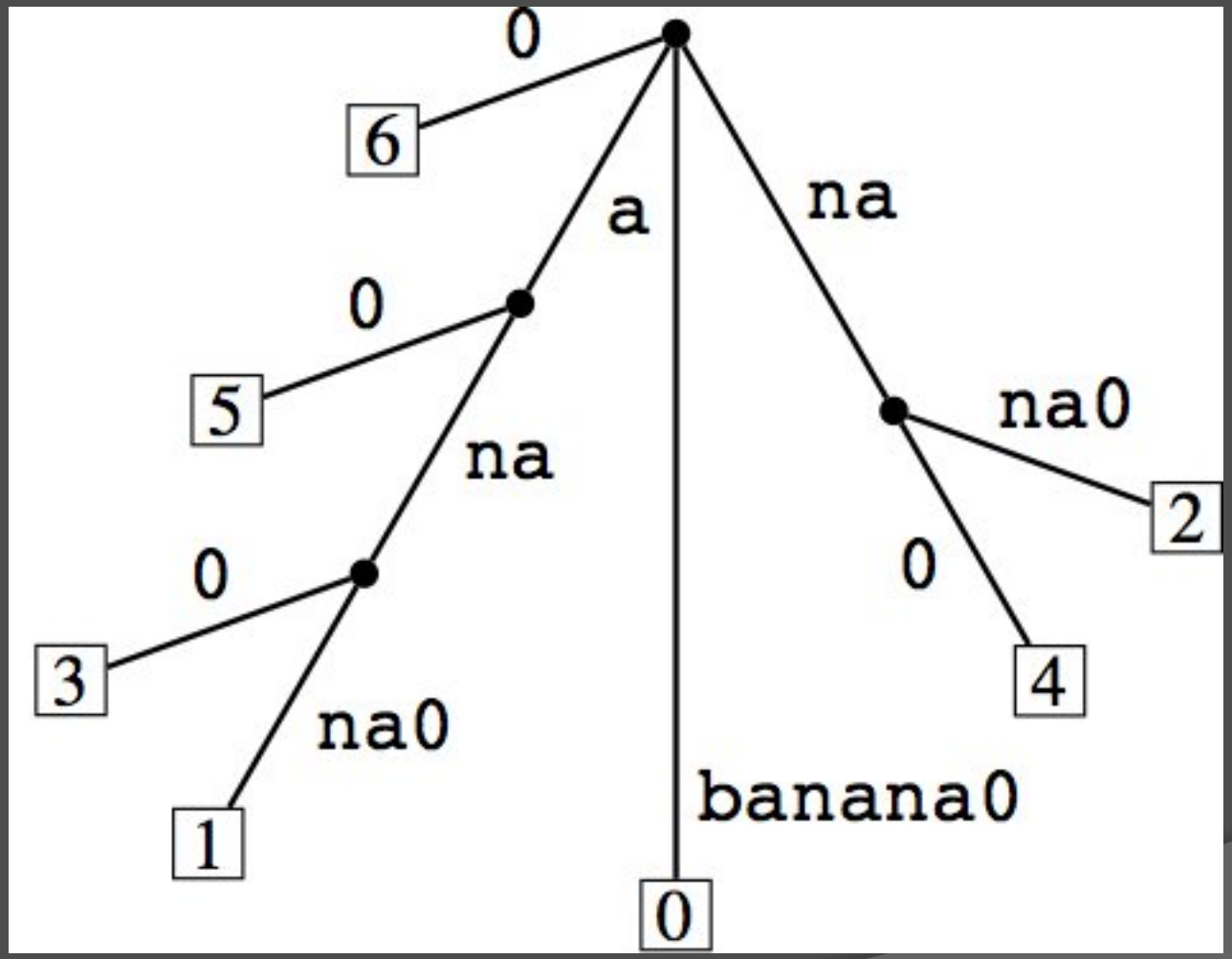
Sicherheitsbegriff



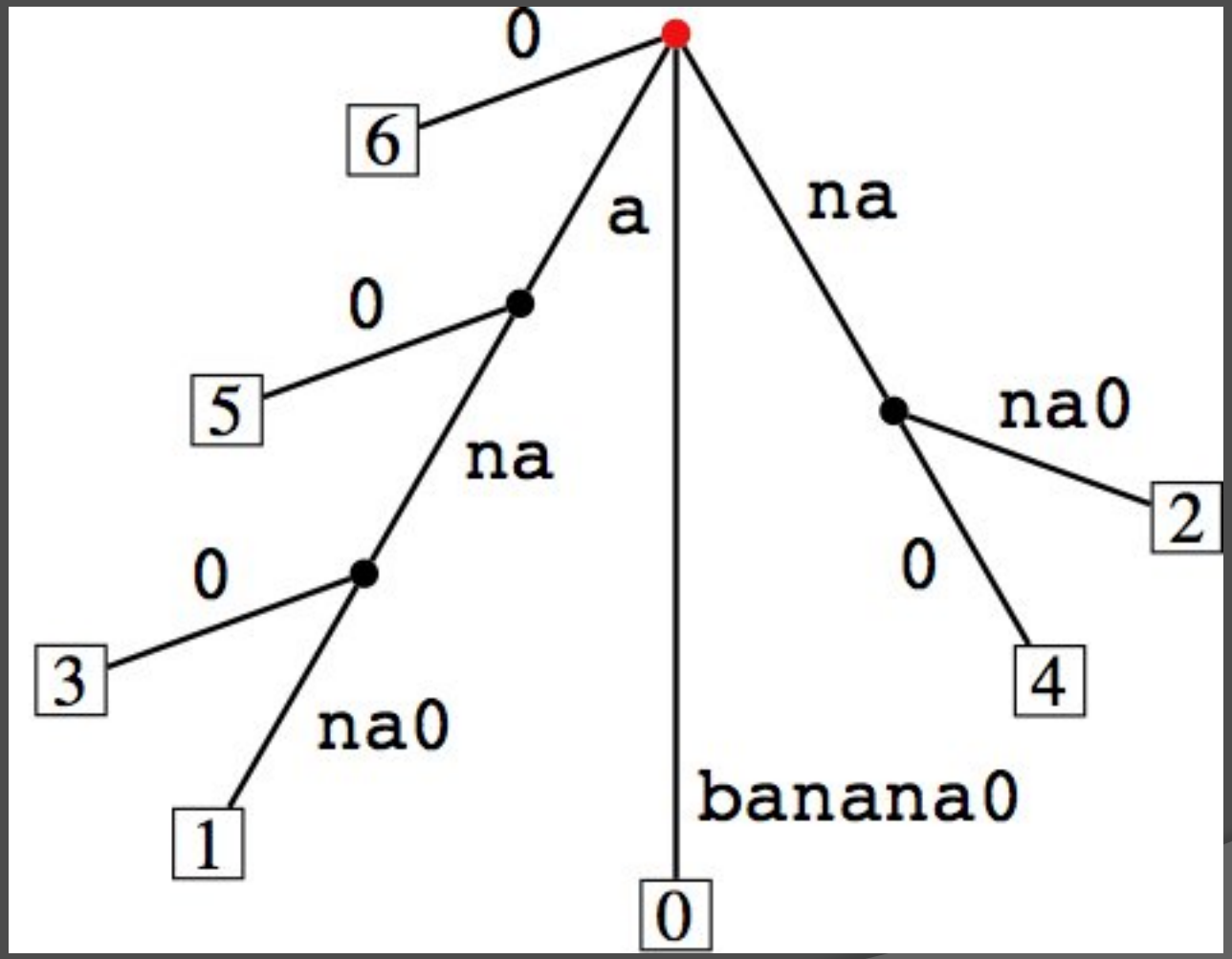
Konstruktion einer Lösung

- Volltextsuche
  - Hier: Nutze Vorberechnung
  - Trade-Off: Time vs. Space
  - O(m)
  - Aufbauend auf Suffix-Trees
  - Nutzen Directed Acyclic Word Graphs (DAWG)

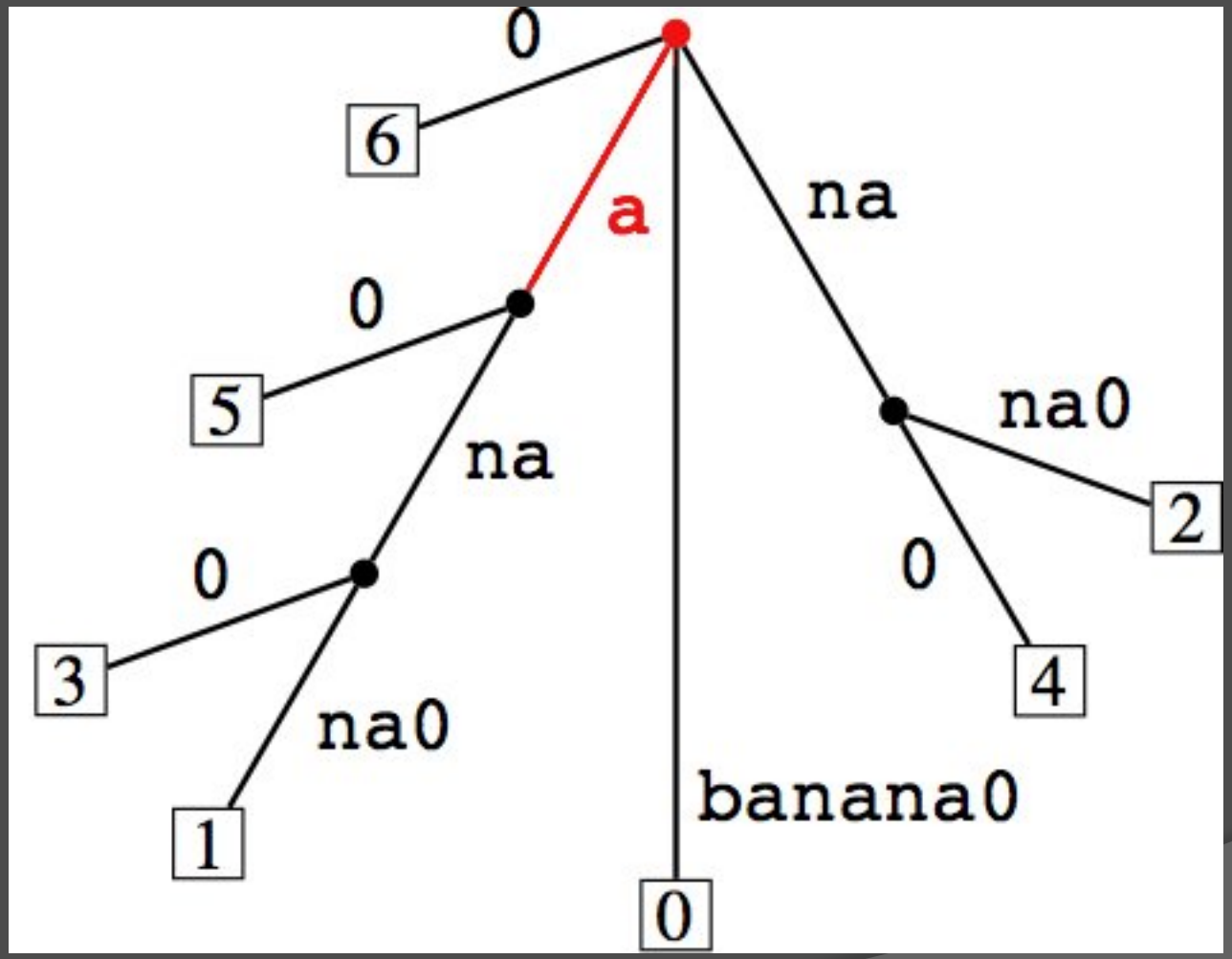
Suffix-Tree



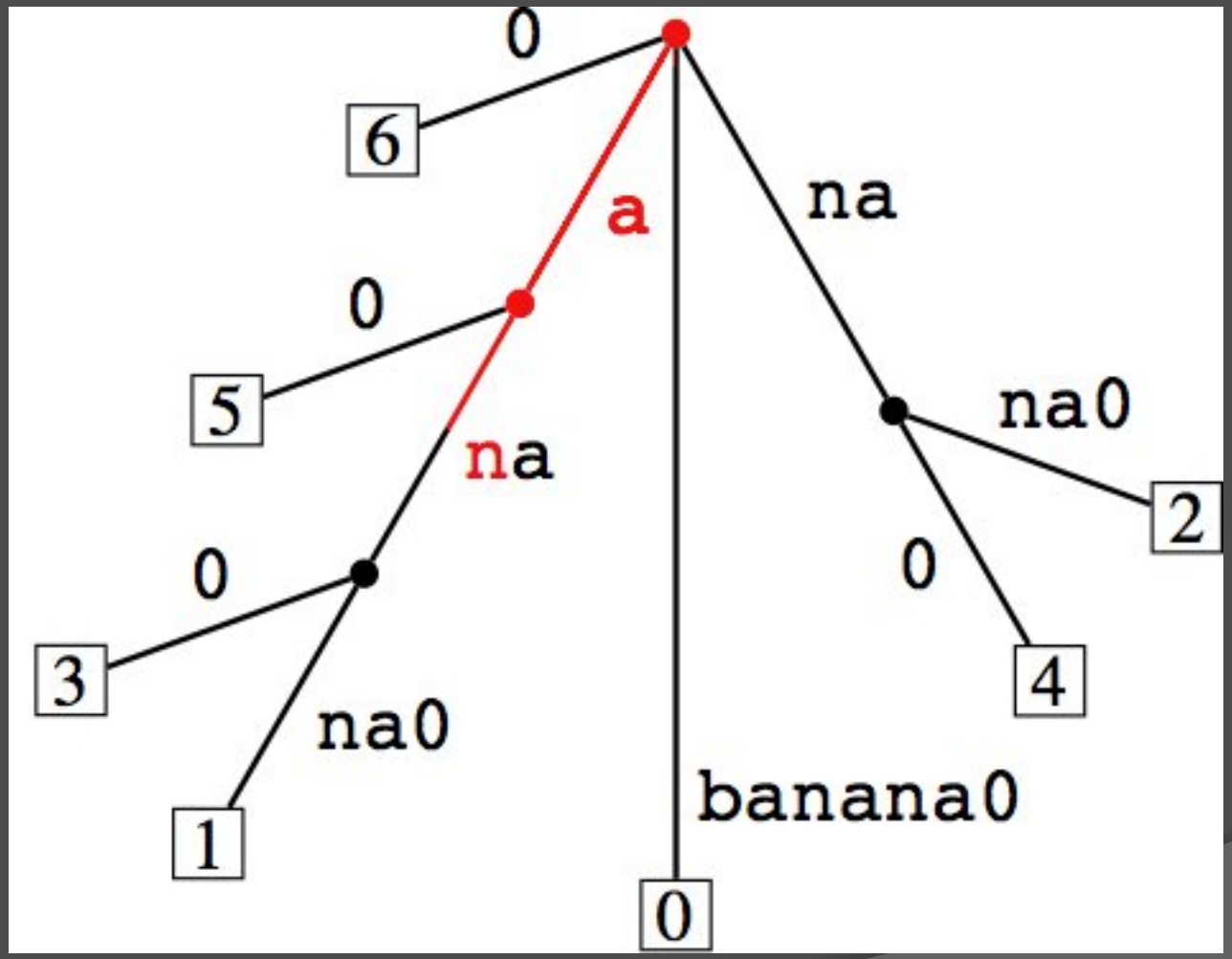
Suffix-Tree



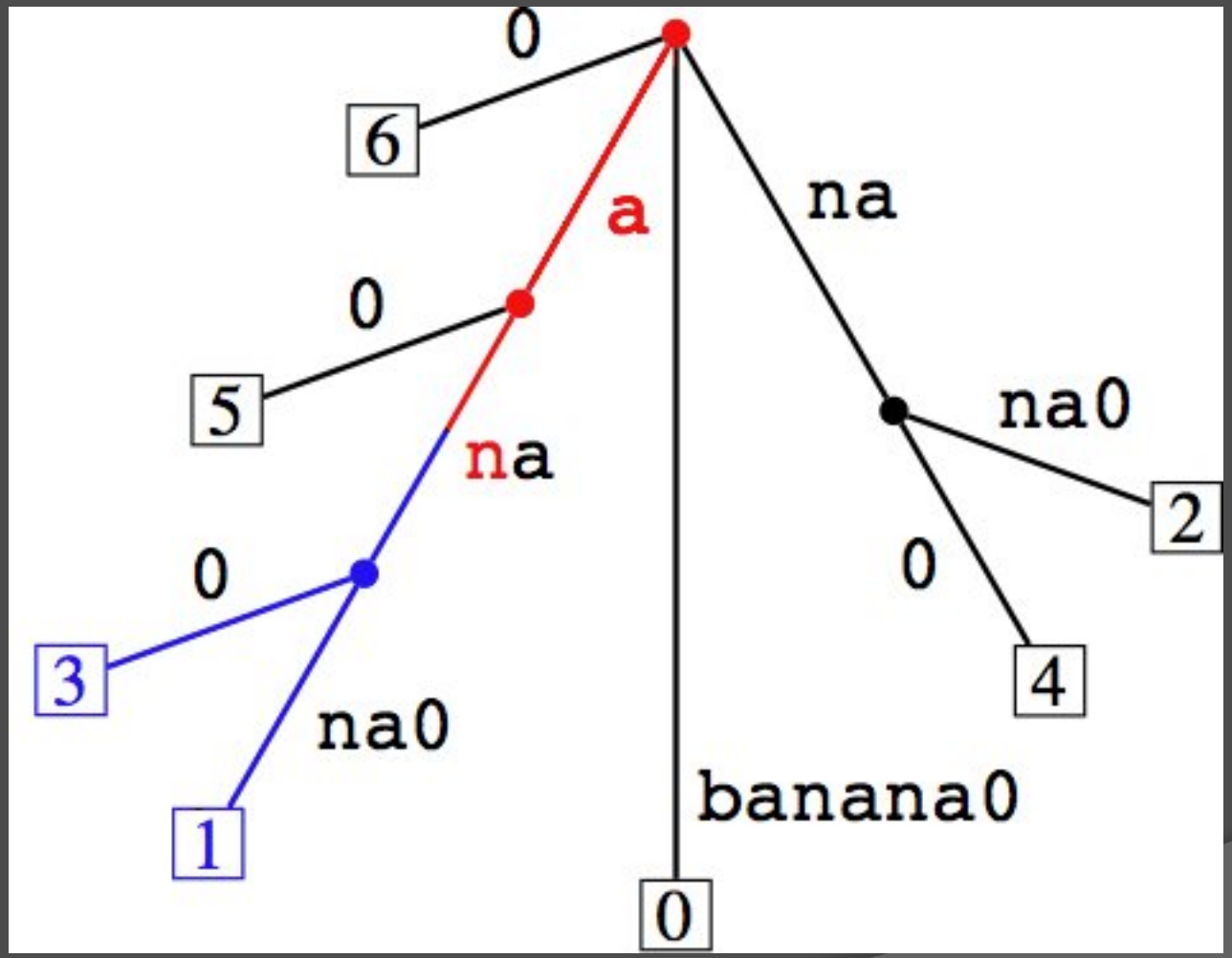
Suffix-Tree



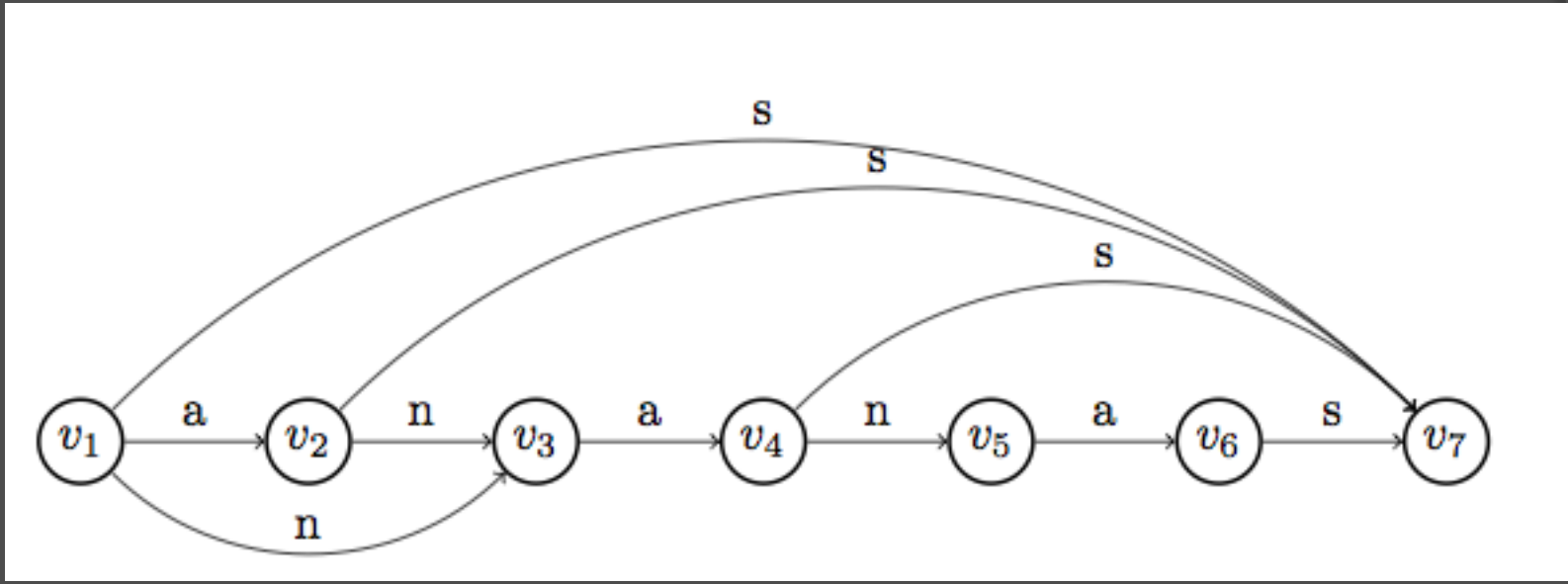
Suffix-Tree



Suffix-Tree

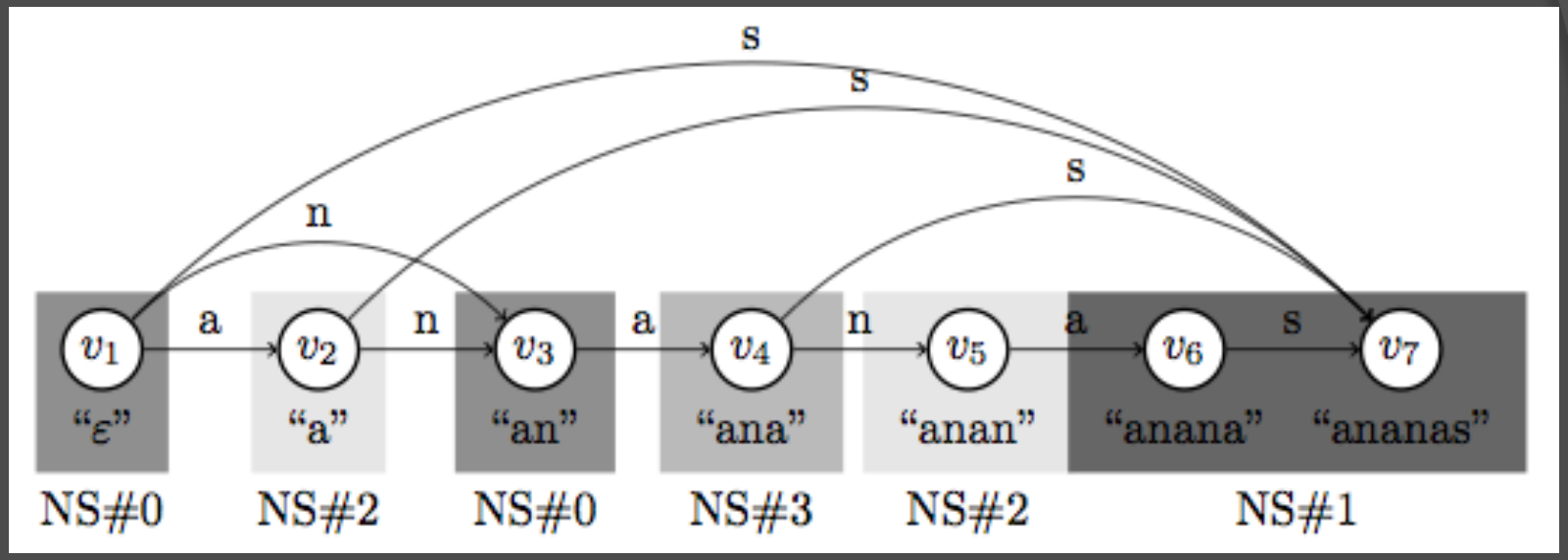


DAWG



S=ananas

DAWG: Nodesets



S=ananas

Was sonst noch zu tun ist

- Nodeset-Referenzen einfügen
- Pfad für decryption markieren
- Anzahl Vorkomnisse einfügen
- Nodesets padden
- Blöcke verschlüsseln

Wie sucht man?

1. Lade Block 0
2. Entschlüssele Block 0
3. Entscheide welchem Pfad zu folgen ist
4. Lade Block *b* der Zielknoten enthält
5. Entschlüssle *b*
6. GOTO 3

Benchmarking

- Implementierung in Java
- Benchmarked gegen triviale Lösung
- Data: 60 MB Mailbox (Enron Corpus)
- Client: Dual Core 1,3 Ghz, 4GB
- Server: Amazon S3, 1&1 SmartDrive
- Connection: IP over 3G,
  1,6 Mbit throughput, 310ms round trip

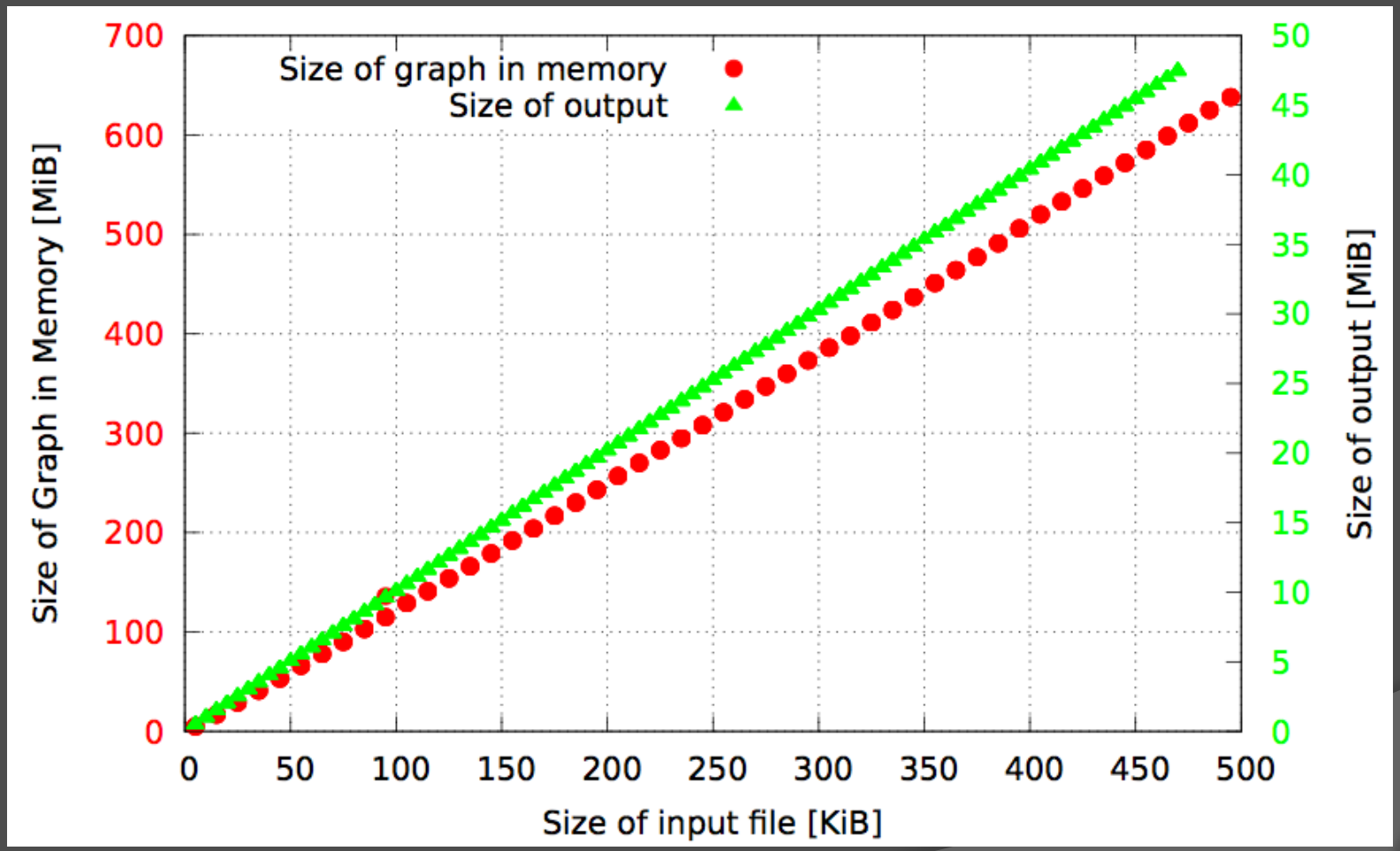
Time

$\|w\|$	*Amazon* [*sec*]	*Amazon trivial* [*sec*]	*Harddisk* [*sec*]
2	10.90	202.11	0.32
4	12.97	196.98	0.44
8	18.72	203.70	0.47
16	29.60	197.91	0.64
32	50.23	197.52	0.77
64	104.29	196.12	1.20

When are we good?

- Low throughput
- Short patterns (e.g. < 10 chars)
- Large data corpus

Speaking of large Data



What can be done?

Contact us if you want to help!

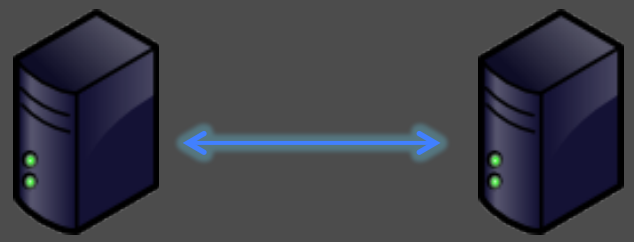
- Improve / Engineer DAWG construction
  - Perhaps in-place?
  - Lighter prog. language? LISP?
- Improve Searching
  - Non-Blocking-IO
  - Use Stream-Cipher
- Engineer Construction
  - Reduce Padding
  - Caching
  - Stop searching after match

Q&A

Spezialisierung

MPC

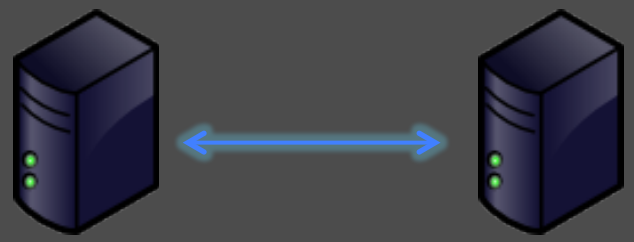
Rechnen,
Teil der Daten



Rechnen,
Teil der Daten

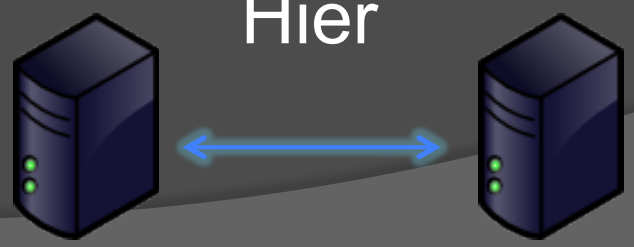
Homomorphe Verschlüsselung

Rechnen,
Daten



Hier

Rechnen



Daten