Pretty Bad Privacy
Pitfalls of DNS Encryption

Haya Shulman

Fraunhofer SIT
- and -
Fachbereich Informatik
Technische Universität Darmstadt

CASED

Fraunhofer SIT

OUTLINE

- **Domain Name System (DNS) and privacy concerns**
- **Privacy for DNS through encryption**
- **Interoperability with existing infrastructure**
- **Protocol support**

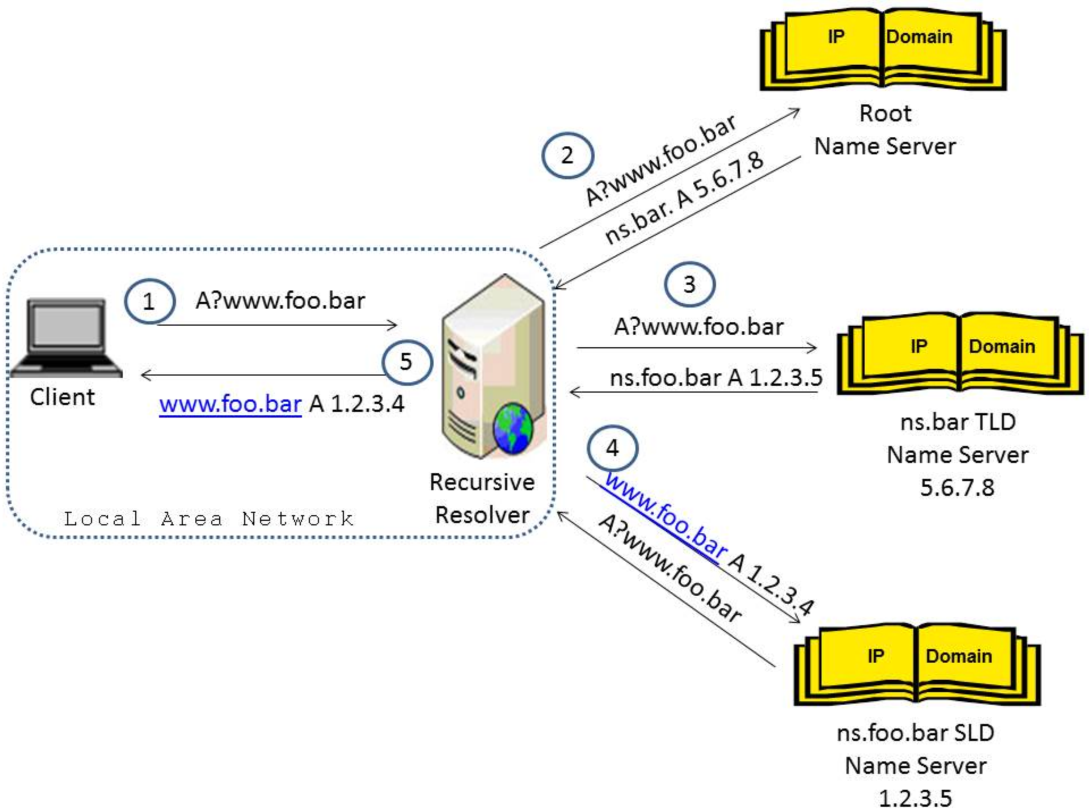
Domain Name System (DNS)

Lookup services

- Locate resources via names
- Security mechanisms: black lists, policies, security mechanisms (DANE, SPF, ROVER, ...)

Properties

- Authentication and integrity
- Availability
- Privacy



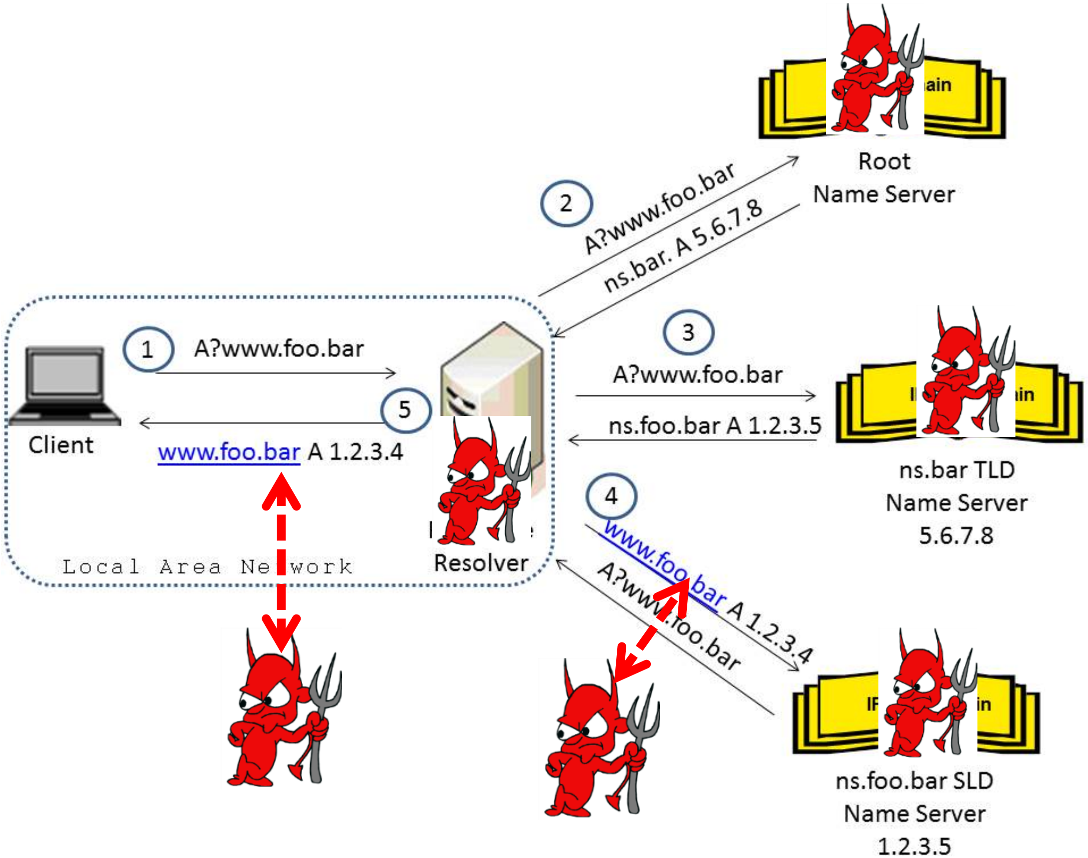
Threats: Monitoring and Surveillance

Cleartext DNS packets monitored, collected, logged

- Research
- Operational purposes
- Financial gain: tailored ads
- Intelligence collection
- Censorship

Attackers

- Eavesdroppers
- DNS/ networks operators
- Third party service providers



See [Bortzmeyer2013] for discussion of threats and privacy issues

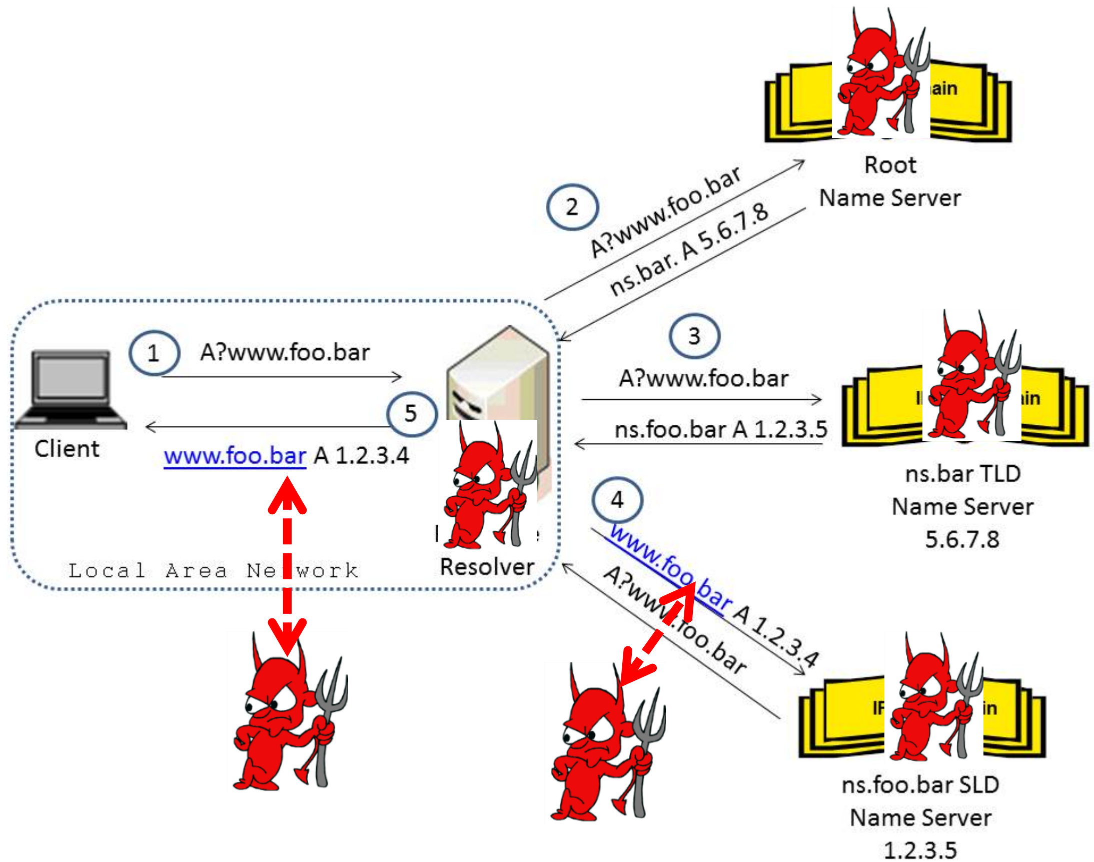
Threats: Monitoring and Surveillance

Cleartext DNS packets monitored, collected, logged

- Research
- Operational purposes
- Financial gain: tailored ads
- Intelligence collection
- Censorship

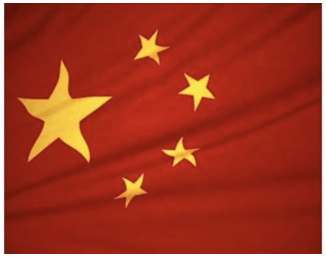
Attackers

- Eavesdroppers
- DNS/ networks operators
- Third party service providers



Data in DNS is public!!

Privacy for DNS?



DNS data is public!… but

- `www.cows.xxx, twitter.com`,…
- VoIP (looking up phone number)
- Sensitive personal information: OS, apps, habits
- More: retrieving certificates, lookup directory service

Large effort within research and operations communities to protect DNS privacy

- Number of proposals, encryption most promising
- On a standardisation track
- Already supported in some software

Encryption of DNS Packets

Selected Proposals

- **DNSCurve/DNSCrypt**
  - Bernstein, Dempsky
  - OpenDNS, DJBDNS
- **DNS over TLS**
  - Unbound (Nlnet Lab)
  - TDNS (Zhu et al, Hoffman et al)
- **Opportunistic encryption with `Encrypt` RR**
  - Wijngaards+Wiley

Differences

- **What is protected**
  - Channel vs DNS record
- **Adoption requirements**
  - Changes to DNS message format
  - Changes to DNS software
  - New server port

OUTLINE

- Domain Name System (DNS) and privacy concerns
- **Privacy for DNS through encryption**
- Interoperability with existing infrastructure
- Protocol support

Does Encryption Provide Privacy for DNS?

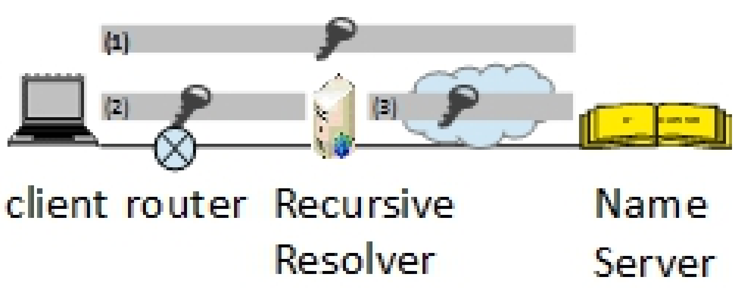
- **Destination IP address in DNS request leaks server‘s identity**
  - Correlation between IP and zone file
  - Often may suffice, e.g., xxx

- **But → zone coresidence**
  - More than 80% of name servers host more than 4 zone files
  - Some more than 500 zone files
  - Guessing by destination IP address does not provide significant advantage

- **But → side channels**
  - Generic (latency, packets‘ sizes)
  - DNS specific (transitive trust)

Attacker Model and Side Channels

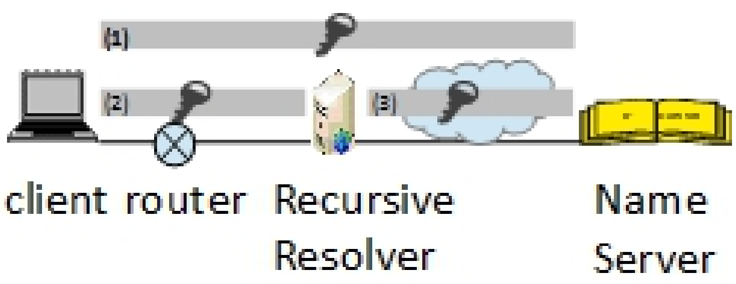
- **Scenario (2): [client] – [attacker] – [recursive]**
  - <u>Threat</u>: WiFi, compromised (home) router,…
  - Recursive caching resolver is trusted
  - Attacker does not see destination IP address of name server
  - Attacker sees request/response timing, sizes
    - Can differentiate cached vs non-cached responses
    - Use (request → response) latency /size to **guess target name server**



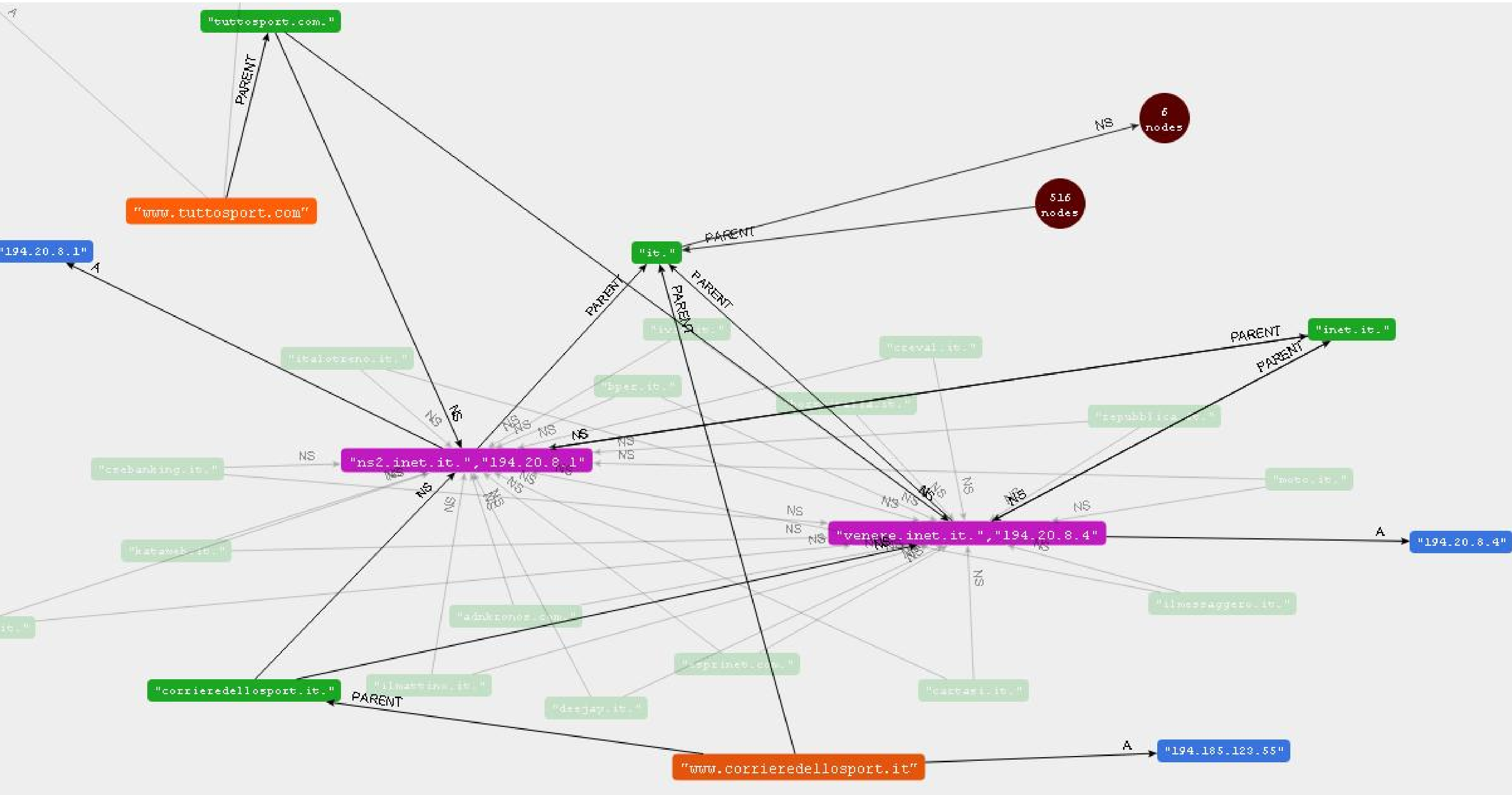
Attacker Model and Side Channels

- **Scenario (1): [client/recursive] – [attacker] – [name server]**
  - Threat: malicious network/DNS operator, eavesdropper
  - Attacker sees request/response timing, sizes, transitive trust dependencies
  - Cache cannot be utilised (end-to-end encryption)
  - Use queries‘ pattern + request → response latency/size to **guess DNS query**

- **Scenario (2+3): [client] – [attacker] – [recursive] – [attacker] – [name server]**
- Threat: malicious network operator, eavesdropper, WiFi, compromised router
- Use queries‘ pattern + request → response latency/size to **guess DNS query**



Transitive Trust Dependencies



Deanonymisation Utilising Transitive Trust Dependencies

- **Preprocessing (offline) phase**
  Query domains (e.g., 1M-top Alexa), construct graph (connected components)
  - For every query, add edges to all dependent queries (we use neo4j)
  - Add weights to edges to track queries‛ order
  - Flush cache after each query
- **Attack phase (single request)**
  - Upon queries from a client, record the pattern
  - Lookup a matching pattern in DB
- **Attack phase (concurrent requests with responses)**
  - Use timing to identify dependent requests
  - Correlate requests with responses via ports

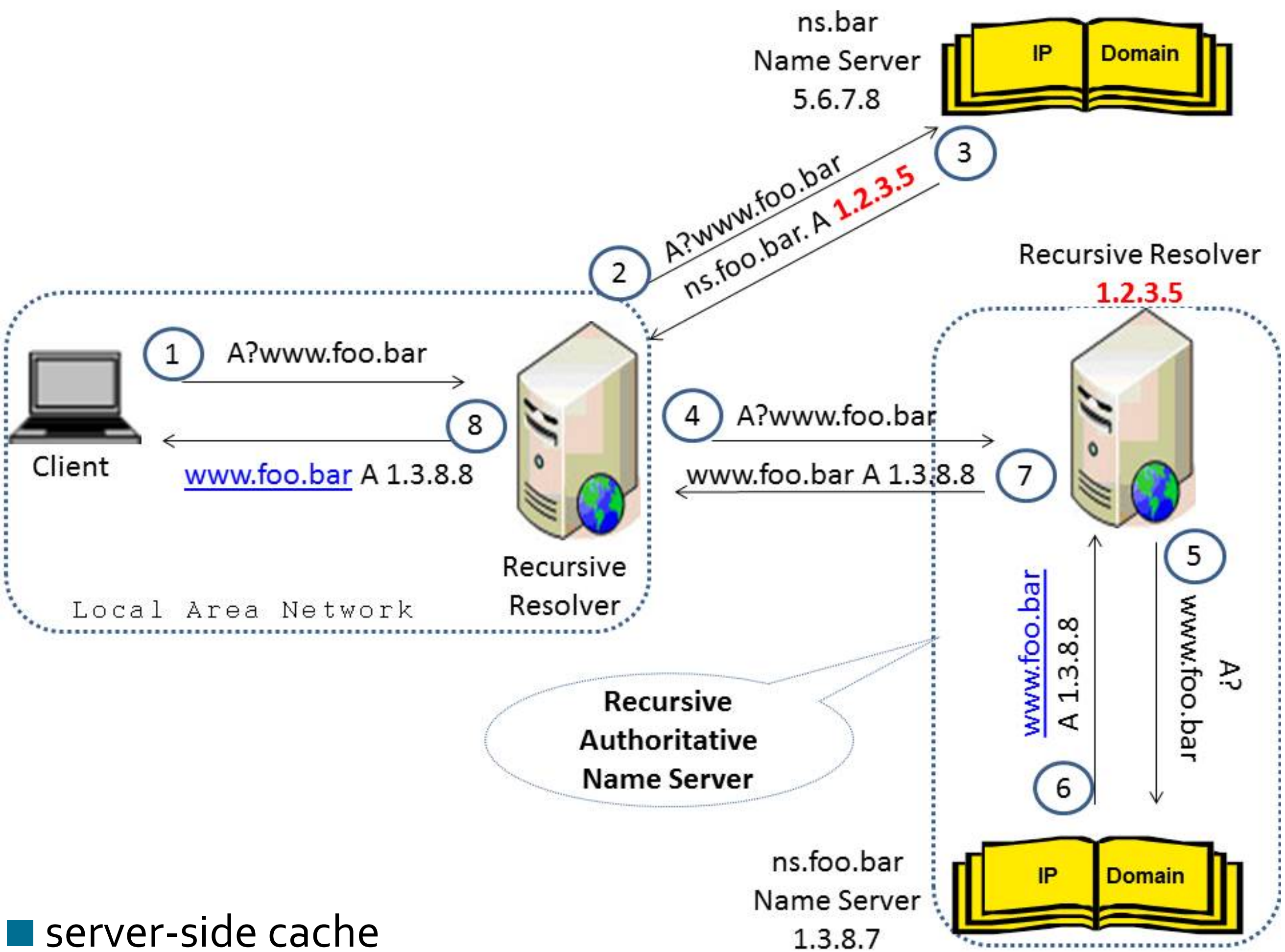
Deanonymisation Utilising Transitive Trust Dependencies

- **The cache is warm** – some queries are not sent (responded from cache)
- Subgraph matching with partial information
- Resolvers may vary in
  - caching policies
  - server selection algorithm
  - latencies
  - DNS records (e.g., CDN)
- Dependencies graph produced at **preprocessing phase** may differ from dependencies produced by a different (victim) resolver
- Use multiple (geographically) **distributed vantage points** to construct the DB
  During attack phase, match against all copies and use the most accurate result

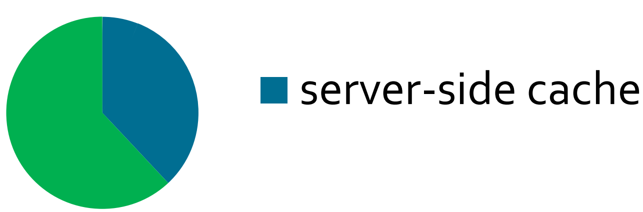
OUTLINE

- Domain Name System (DNS) and privacy concerns
- Privacy for DNS through encryption
- **Interoperability with existing infrastructure**
- Protocol support

Server-Side Caching Resolvers

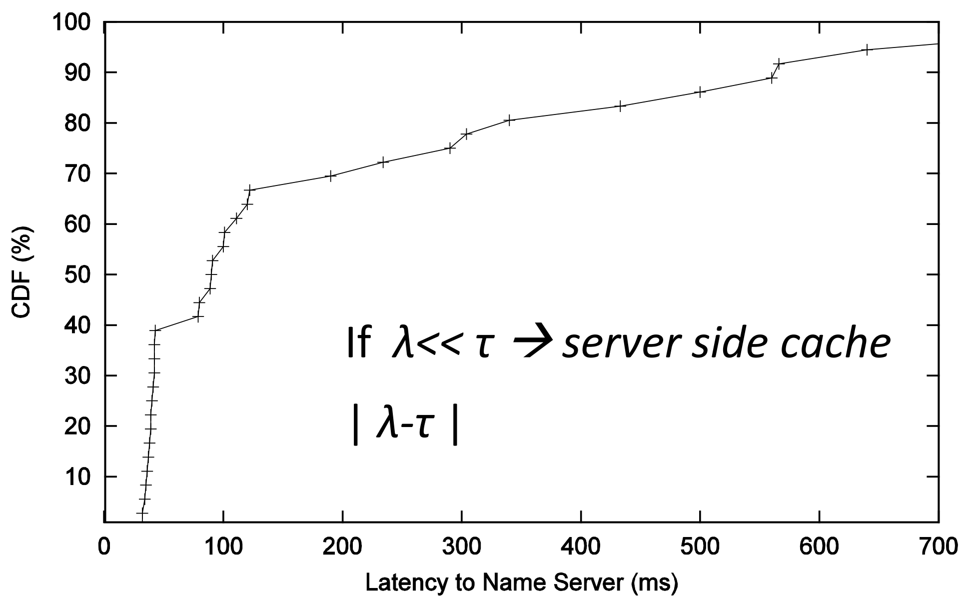


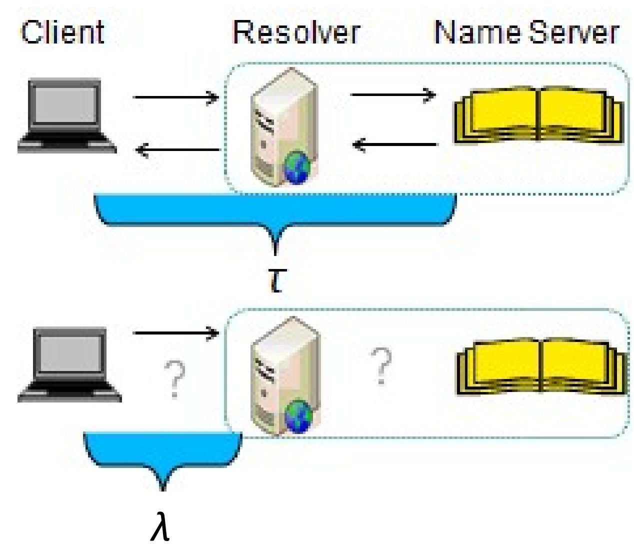
Alexa-50K



3rd Party Server-Side Caching Resolvers

- **Which name server to forward the DNS request to?**
  - Request is encrypted
  - Proxy does not have corresponding decryption key
  - Proxies are not trusted – operated by 3rd partiess





OUTLINE

- Domain Name System (DNS) and privacy concerns
- Privacy for DNS through encryption
- Interoperability with existing infrastructure
- **Protocol support**

TCP Support

- Proposals for encryption assume support of TCP
- Failures on client-side : 17% failures, [Geoff Huston 2013]
- Our study shows failures also on servers: SERVFAIL, timeouts, RST,…
  - On third party proxy
  - On name servers
- Requires careful study of TCP
- Failure cannot be distinguished from a downgrade attack
  - Attacker can cause fall-back to UDP

Fatal Failures with TCP on Name Server Side

- After TCP handshake, DNS request is responded with RST+ ICMP(type=3, code=10)
  server cannot answer (administratively prohibited)
  for instance: **`edns-chtn.cht.com.tw 202.39.168.132`**
- After TCP handshake, DNS request is responded with ACK then RST
  for instance: **`gerek.accv.es 195.77.23.35`**
- Server keeps resending SYN+ACK
  for instance: **`ns7.utoronto.ca 162.243.71.42`**
- After TCP handshake, DNS request is responded with RST
  for instance: **`dns1.hessen.de 141.90.2.53`**
- TCP window fluctuations: SYN+ACK with window 0, then SYN+ACK with window > 0 (e.g., 4096)
  for instance: **`beloit.edu 144.89.40.1`**
- After TCP handshake, DNS request is responded with ACK+FIN
  for instance: **`a.ns.207.148.in-addr.arpa 148.207.1.1`**
- After TCP handshake, DNS request is responded with multiple small segments
  e.g., segments of size < 100bytes for response length 557 bytes
  for instance: **`ns.CWRU.Edu 129.22.4.1`**
- After TCP handshake, server sends SYN+ACK, then silent
  for instance: **`cnsa.vita.virginia.gov 166.67.65.169`**

Large number of popular domains affected

Conclusions

Encryption is important

- Ensures privacy
- Prevents attacks against DNS
- But, important to study/consider obstacles and challenges

But, requires careful evaluation

- Infrastructure compatibility
- Protocol support

Future work and considerations

- Outsourcing is an increasing trend → how to handle third party proxies?
- Support of basic protocols :TCP → which version?
- DNS and side channels: timing, sizes, domains dependencies, browsers' prefetching,...

Questions?

Thank you!