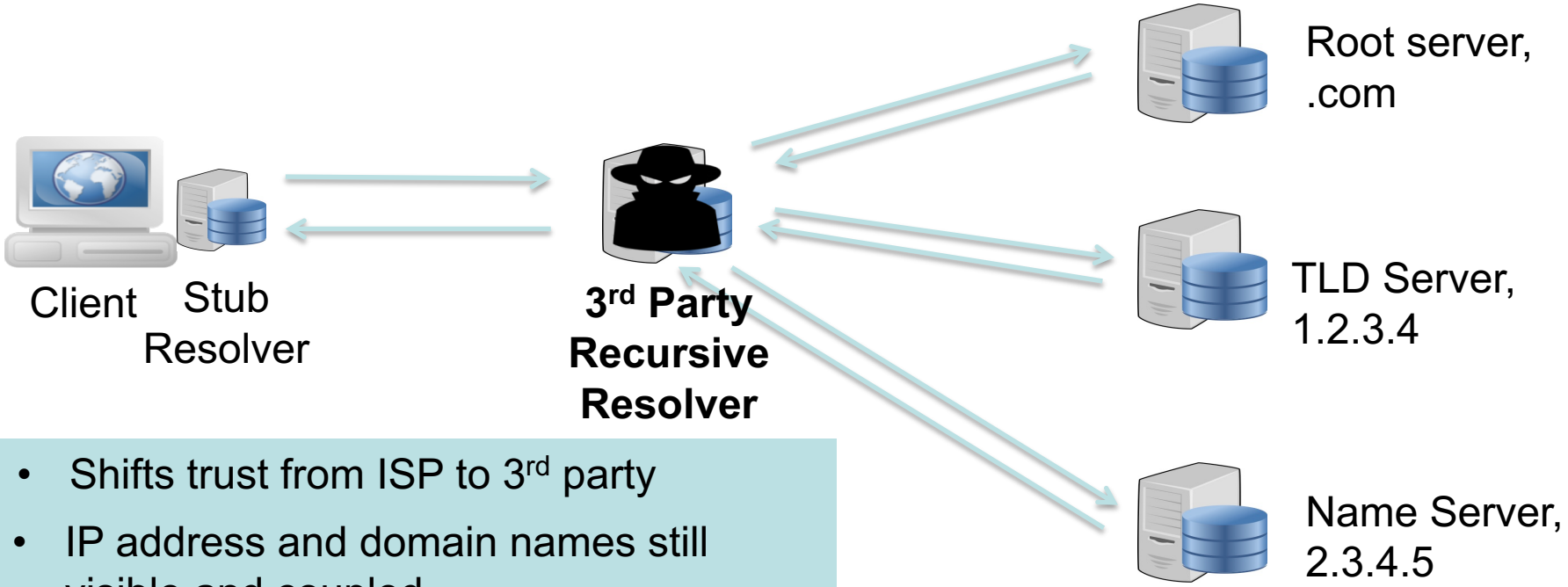
Oblivious DNS

Annie Edmundson, Paul Schmitt, Nick Feamster
Princeton University

Allison Mankin
Salesforce

Strawman: Change DNS Providers



- Shifts trust from ISP to 3rd party
- IP address and domain names still visible and coupled
- Still vulnerable to monitoring & data requests

Oblivious DNS (ODNS)

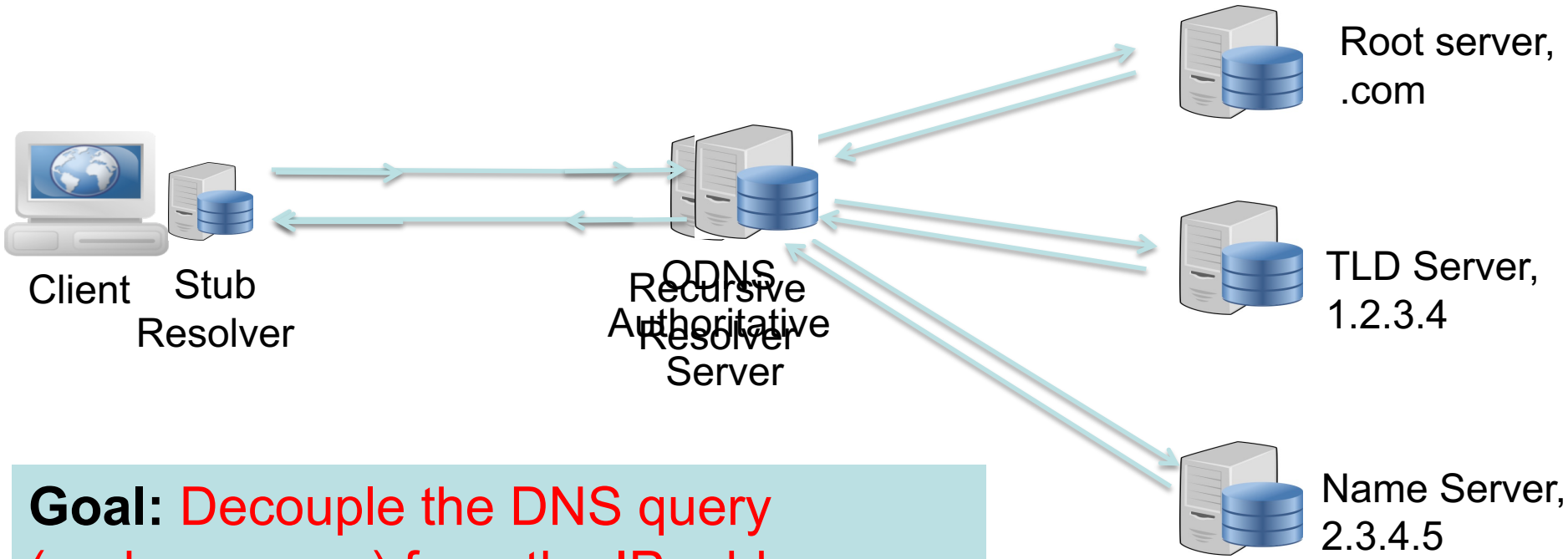
Goal*: Decouple the DNS query (and response) from the IP address that issued the query*

1. Obfuscate the DNS query before sending it to the local recursive resolver

2. Generate a referral to an ODNS authoritative server that can decipher the query

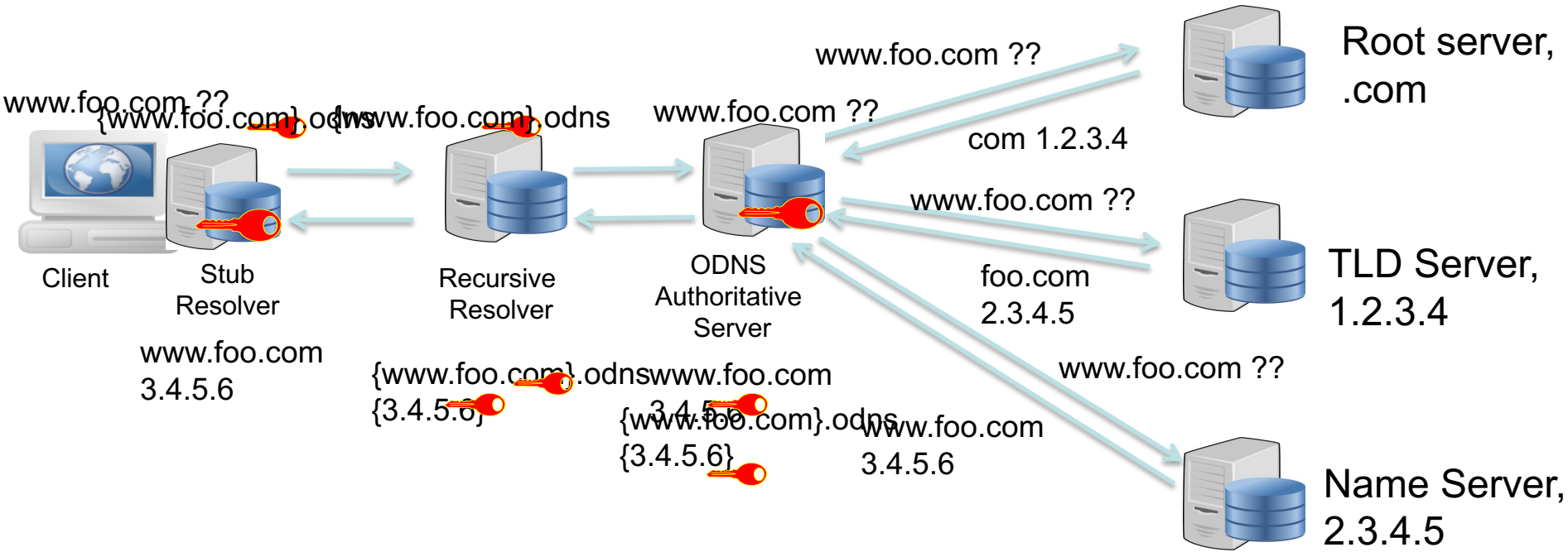
3. ODNS authoritative server can see the DNS query, but not the IP address of the requesting client

Oblivious DNS

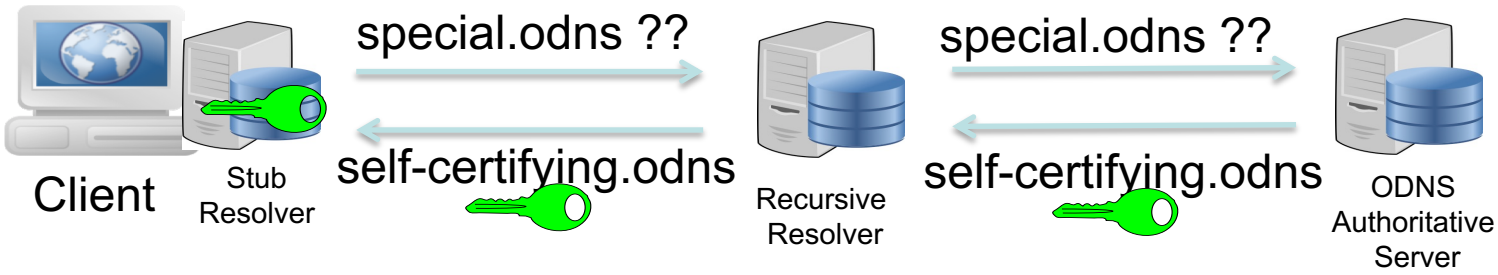


Goal: Decouple the DNS query (and response) from the IP address that issued the query

ODNS Queries & Responses



Distributing ODNS Keys to Clients



ODNS authoritative servers are replicated and anycasted, resulting in selection of the closest ODNS authoritative server

Changes/Additions to DNS

- **Stub resolver**
  - Session key generation
  - Encryption of session key with authoritative PK
  - Domain name augmentation
  - Appends session key in additional section
- **Authoritative DNS server**
  - Decrypt session key and query
  - Forward recursive query as before

Ongoing Implementation Efforts

- Prototype implementation in Go w/Go DNS library
- Some initial progress with Unbound at Hackathon

- Implementation detail: Ciphertext of encrypted QNAME too large for 0.6% of names in lookup trace

Performance Evaluation: ODNS

- Overhead of **cryptographic operations**
- Additional latency for **DNS lookups**
- Additional **Web page load time**
- Reduced caching at recursive resolver

Practical Considerations

- EDNS0 Client Subnet
  - **Challenge:** Local recursive can pass on client IP address in query
  - **Solution:** Local recursive should strip EDNS0 CS
- OPT Records and Query Length
  - **Challenge:** Keys are big. Encrypted query/session key can't go in OPT because most resolvers strip it!
  - **Near-term Solution:** QNAME (4 x 63 bytes)
    - 16-byte AES keys, ECIES encrypted key (44 bytes)
    - We use base64 encoding for encrypted domain & key (drawback: no 0x20 encoding)

Which Recursives Can ODNS Use Today?

- No EDNS0 Client Subnet, No 0x20...

	Open Recursive Resolver (IP)	EDNS0 Client Subnet	0x20
	Cloudflare (1.1.1.1)	No	Yes
	Google (8.8.8.8)	Yes	No
✓	Quad9 (9.9.9.9)	No	No
✓	Level3 (209.244.0.3)	No	No
✓	OpenDNS Home (208.67.222.222)	No	No
	Verisign (64.6.64.6)	No	Yes
✓	Norton ConnectSafe (199.85.126.10)	No	No
	Dyn (216.146.35.35)	Yes	No
✓	Comodo Secure DNS (8.26.56.26)	No	No
	Fourth Estate (45.77.165.194)	Yes	No
✓	DNS.WATCH (84.200.69.80)	No	No
	GreenTeamDNS (81.218.119.11)	Yes	No
✓	SafeDNS (195.46.39.39)	No	No
✓	FreeDNS (37.235.1.174)	No	No
✓	Hurricane Electric (74.82.42.42)	No	No
✓	Ultra (156.154.71.1)	No	No

ODNS: Summary

- ODNS protects privacy by decoupling clients' identities from their queries
- Implementation and evaluation show feasibility and low overhead
- ODNS is compatible with existing recursive resolvers and name servers