Oblivious DoH

draft-pauly-dprive-oblivious-doh
Oblivious DoH supports proxying encrypted DNS queries between a client and resolver.
Oblivious DoH
Requirements

Client knowledge:

• Name and public key of target resolver
• Address of willing proxy

Privacy assumption: Targets and proxies do not collude

Privacy goal: Keep knowledge of DNS messages and stub IP separate to all (except the client stub)
Oblivious DoH

Protocol

Target and Resolver are best co-located
Oblivious DoH

Protocol

1) Stub discovers Target public key
Oblivious DoH

Protocol

2) Stub sends encrypted query to Target through Proxy
Oblivious DoH

2) Stub sends encrypted query to Target through Proxy
Oblivious DoH

Protocol

3) Target sends encrypted response to Stub through Proxy
Oblivious DoH
Relationship to other work

Connection-oriented proxies (CONNECT, SOCKS)

• Forces a trade-off between connection setup overhead and linkability caused by long-lived connections

Generalized anonymity networks (Tor)

• Non-negligible latency overhead [Muffett, NDSS 2021], heavier-weight solution

Oblivious HTTP proposals

• ODoH is a specific case of a generalized OHTTP, could eventually merge

• Scoping this to DNS as formulated allows proxies to be more confident that they are not operating as a fully open proxy, by limiting the content type and targets
Oblivious DoH

Deployment Questions

Key discovery?
- Will vary by deployment — there is no mandatory discovery mechanism

Who will proxy?
- Good Samaritans or entities acting on behalf of clients that proxy only to allowed targets

Non-collusion guarantees?
- No technical mechanism in place — not in scope
Oblivious DoH

Status

Several interoperable implementations exist and used in production

• Target support: odoh.cloudflare-dns.com


Initial measurements indicate that PLT and response latency not significantly impacted [Singanamalla et al., NDSS 2021]

• Continued experiments with more stub resolvers underway

Backing formal analysis in Tamarin
Is the WG interested in adopting this work?