Performance Evaluation of the First Open Source Implementation of the ODNS Protocol

Dashuai Wu (Speaker), Shibo Cui, Baojun Liu

Tsinghua University, Network and Information Security Lab (NISL)

Deliang Chang

QI-ANXIN Technology Research Institute

ICANN DNS Symposium, September 2024, Remote

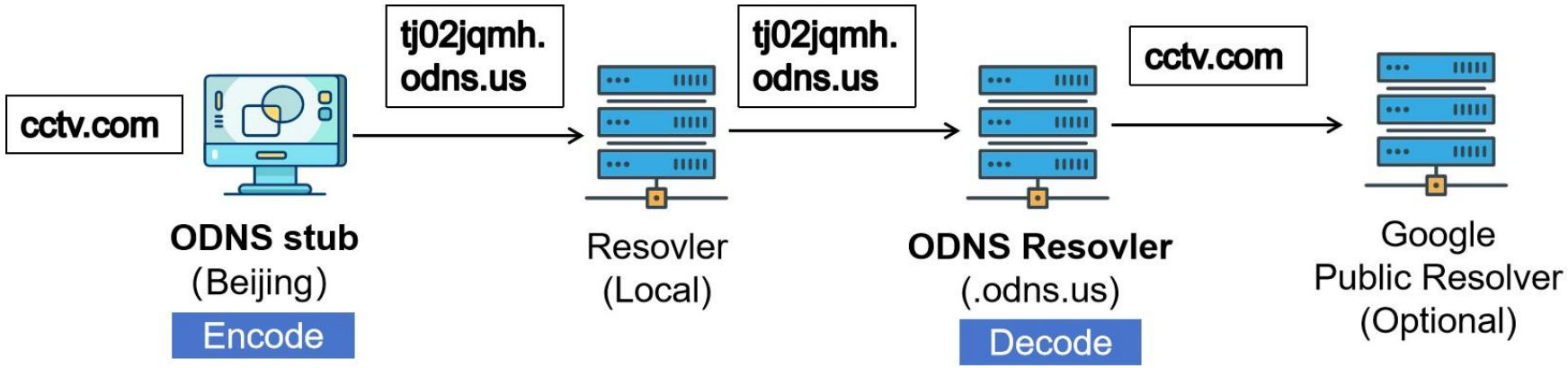
Privacy vulnerabilities in Domain Name System

- DNS maps domain names to IP addresses and other related information
- Obtaining the binary pair (DNS query name, Client IP) can reveal significant information about Internet users' online activities, such as browsing habits, types of websites of interest and frequency of network usage, etc.
- Encrypting DNS query name can partially address this issue, preventing eavesdroppers along the path from obtaining DNS query name
  - As implemented in protocols such as DNS over TLS [RFC7858], DNS over DTLS [RFC8094], DNS over HTTPS [RFC8484], and DNS over QUIC [RFC9250], etc.
- However, resolver and its service provider can still obtain the full binary pair
- To address this issue, NO single party should obtain the full binary pair

Anonymization DNS scheme

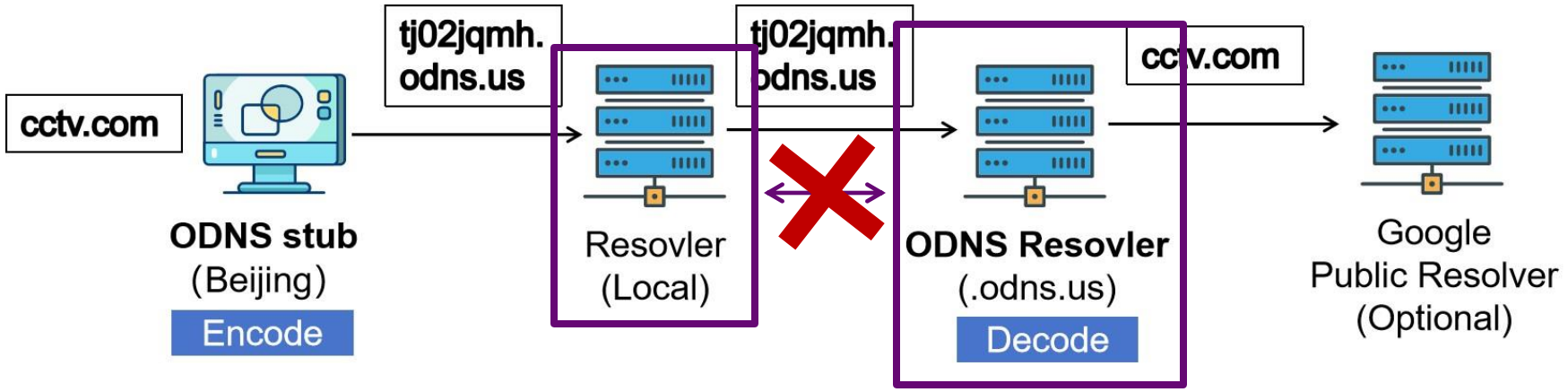
- Oblivious DNS (ODNS)
  - First using relay-based concept into encrypted DNS schemes
  - Introducing ODNS Resolver, which acts like a recursive resolver for original DNS queries and like an authoritative server for encoded DNS queries
  - Using Local Resolver as a natural relay
  - I-D.draft-annee-dprive-oblivious-dns-00: only 1 version, expired
- Anonymized DNSCrypt (ADNSCrypt)
  - Introducing an intermediate server, as a simple extention to standard DNSCrypt
- Oblivious DNS over HTTPS (ODoH) [RFC9230]
  - Introducing an proxy server, using HTTPS as the message channel
- DNS over Oblivious HTTP (DoOH) [RFC9458]

How ODNS Works

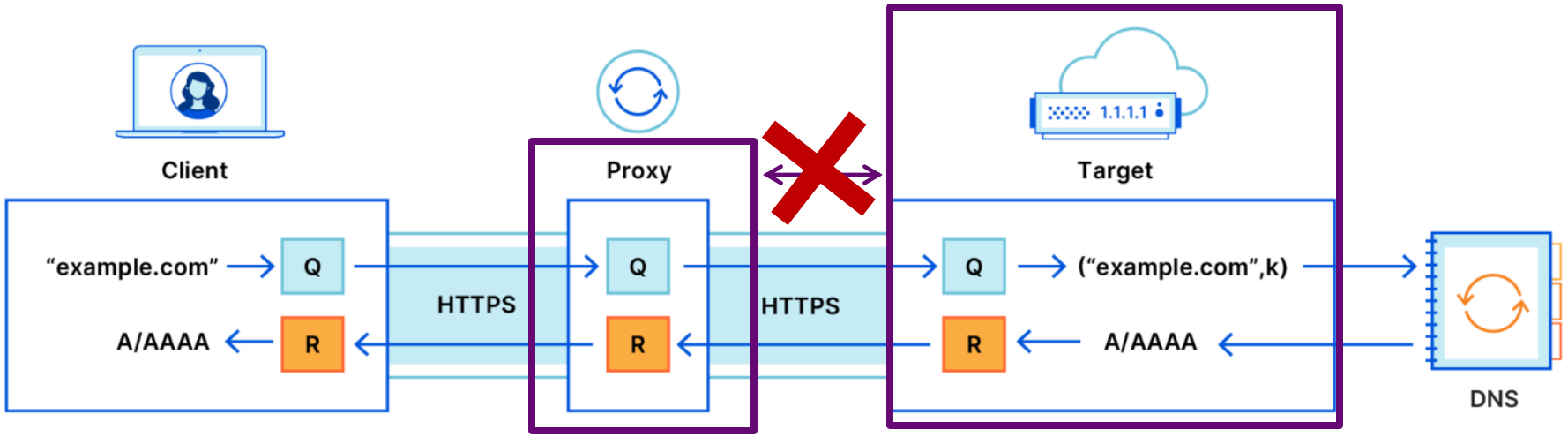


- 3 Main Roles: ODNS stub, Local Resolver, and ODNS Resolver
- Step 1: ODNS stub uses Hybrid Encryption Scheme to encode the domain name queried into a specific authoritative namespace (e.g. *.odns.us).
- Step 2: Resolver forwards the ODNS query to the ODNS resolver
- Step 3: ODNS Resolver decode the query, and then acts as a typical recursive resolver

ODNS vs ODoH: Simillar but Different



ODNS Workflow



ODoH Workflow (From: https://blog.cloudflare.com/oblivious-dns/)

Why we are interested in ODNS and What we have done

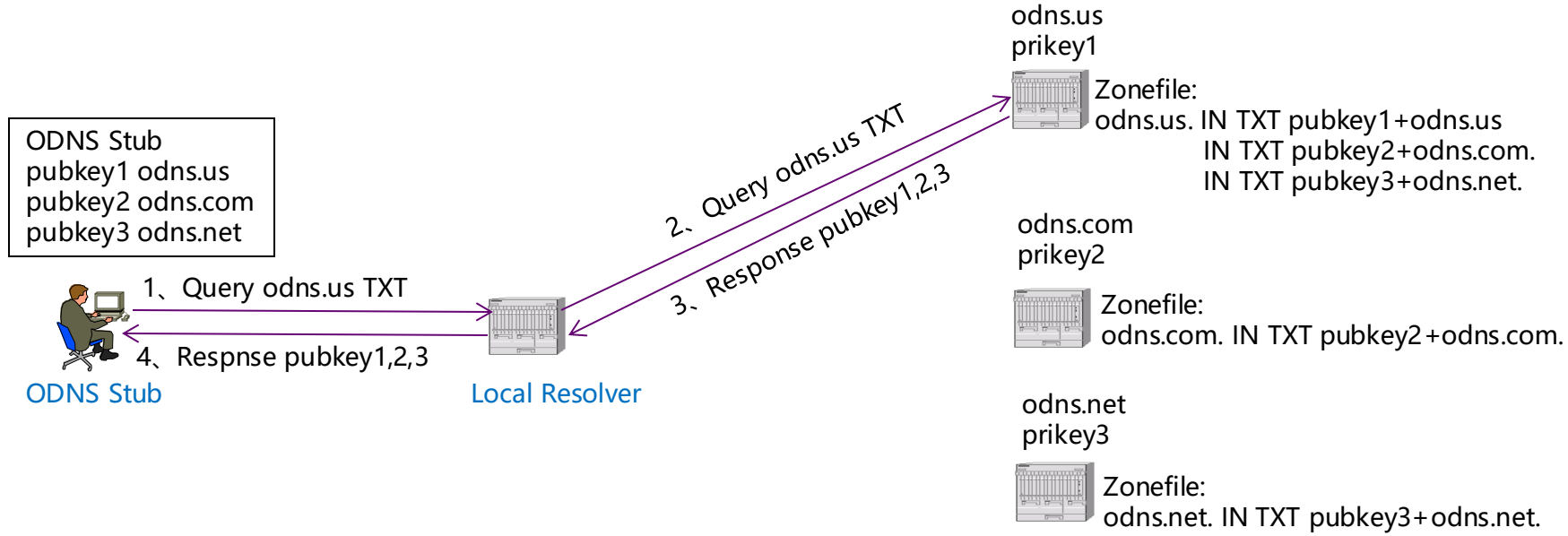
- ODNS is compatible with existing DNS infrastructure, while ADNSCrypt, ODoH are not. The former can directly use Local Resolver or Public Resolver as a relay, while the latter require a dedicated relay
- ODNS relies solely on the assumption that ODNS Resolver and Local Resolver do not colluding, so it's more resistant to collusion risks compared to ODoH
- Only need a proxy layer at stub and resolver, resulting in low implementation and deployment difficulty
- As far as we know, we did not find a concrete implementation of ODNS, so we develop it as a plugin based on CoreDNS using Golang
- Further, we are integrating ODNS into DNS software products together with 114DNS, a large DNS service provider in China

ODNS Execution - Overview

- Some specific details while implementing ODNS:
  - Discovery & key distribution
  - Query request: define encryption algorithms, encoding, and query format
  - Answer response: define decryption algorithms, decoding and answer format
- Performance Optimization
  - Smooth resolver key update

ODNS Execution - Discovery & key distribution

- ODNS Discovery & key distribution
  - The ODNS resolver publishes public key and domain information (which may include other ODNS resolvers) in the TXT records of its base domain.
  - The ODNS stubs requests and caches the public key and domain information from the pre-configured ODNS resolver.

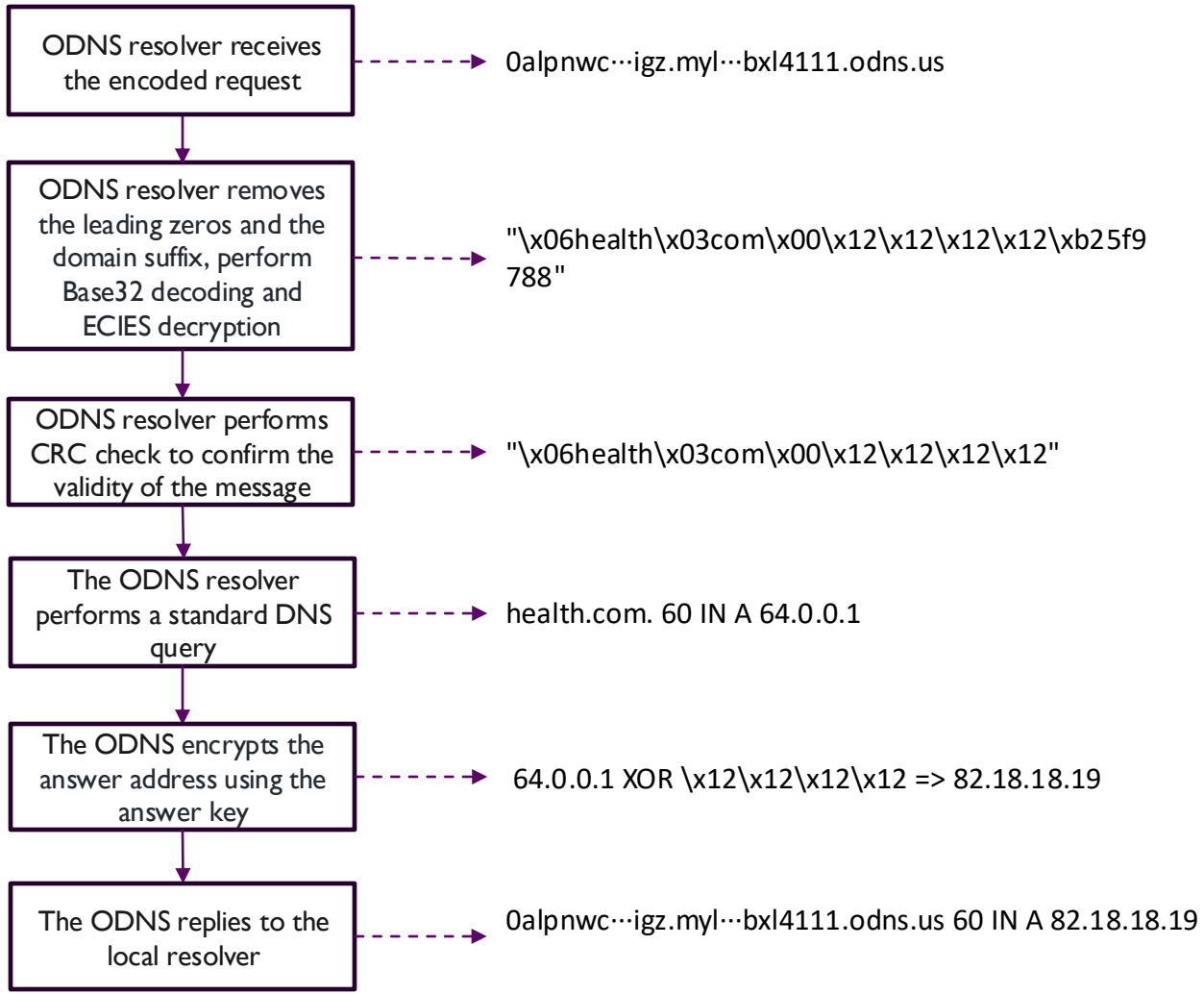


ODNS Execution - Query Request

- ODNS Query Request
  - ODNS use ECIES with point compression
  - The ODNS stub concatenates the various parts needed for the query and encrypts them using ECIES.
    - The query domain name
    - A key for encrypting answer
    - A CRC for checking integrity
  - The ODNS stub concatenated this encoded by Base32 with the ODNS resolver's base domain and sends the query to any local DNS resolver.
  - The local resolver will ultimately query the ODNS resolver, which is authoritative for the base domain.

ODNS Execution - Answer Response

- ODNS Answer Response
  - The ODNS resolver decrypts the encrypted query and decapsulates the message to obtain the answer encrypting key
  - The ODNS resolver performs a standard DNS query, encrypts the obtained answer, and replies to the local resolver.
  - The ODNS stub decrypts the obtained response using the answer key



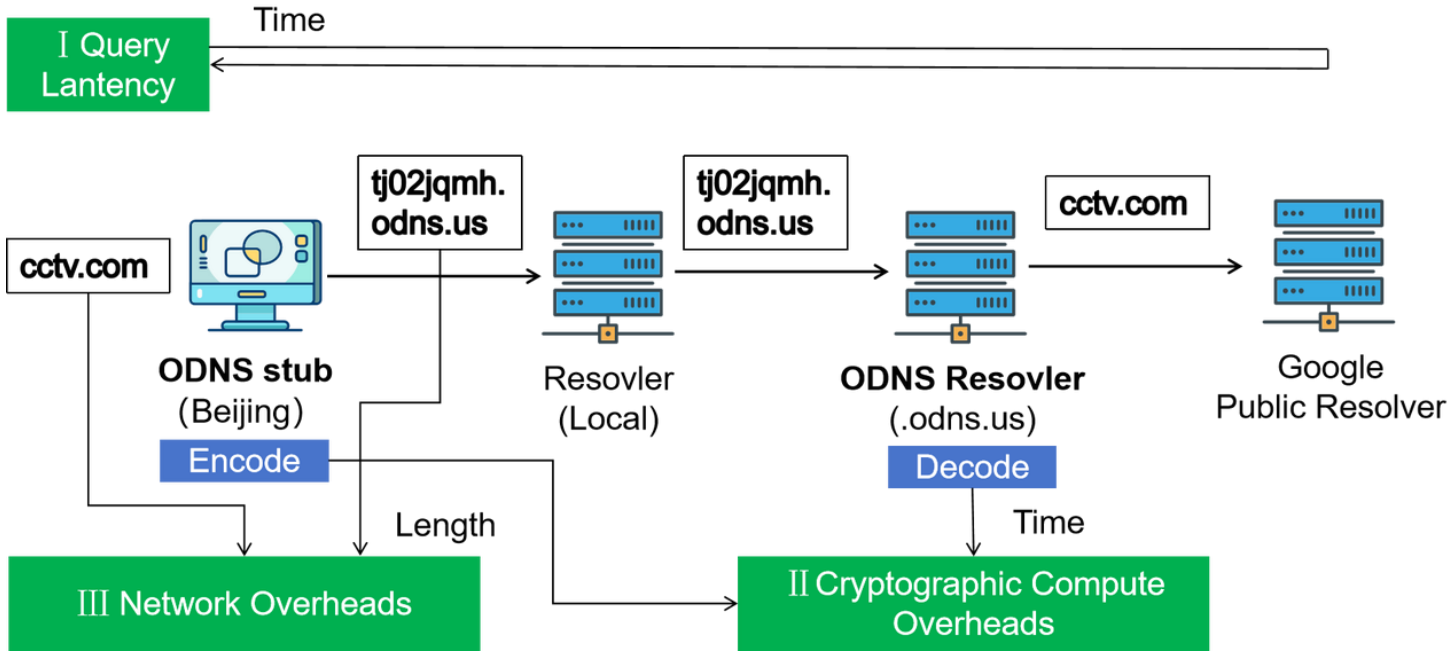
ODNS Execution – Performance Optimization

- ODNS Performance Optimization
  - Problem: The biggest performance bottleneck is the ECDH key exchange.
    - If the stub generates a new pair of private and public keys for each request sent, the resolver must perform an ECDH key exchange for each request it processes.
  - Solution: The stub can use a single pair of private and public keys for a certain time period (e.g., 1 hour). All requests sent during this period are encrypted using this private key.
  - With above optimization, a single 40-core CPU server can handle 1,000,000 QPS.

ODNS Execution – Smooth resolver key update

- Smoothly update the server public keys
  - Problem: If the stub continues to encrypt data using an expired public key after resolver updating the public key, the resolver will decrypt incorrect results and will be unable to respond correctly.
  - Solution: If the resolver encounters a decryption error, it will return 255.255.255.255 as the resolved address. When the stub detects this special result, it should update the public key (by re-querying the TXT record) and then resend the request.

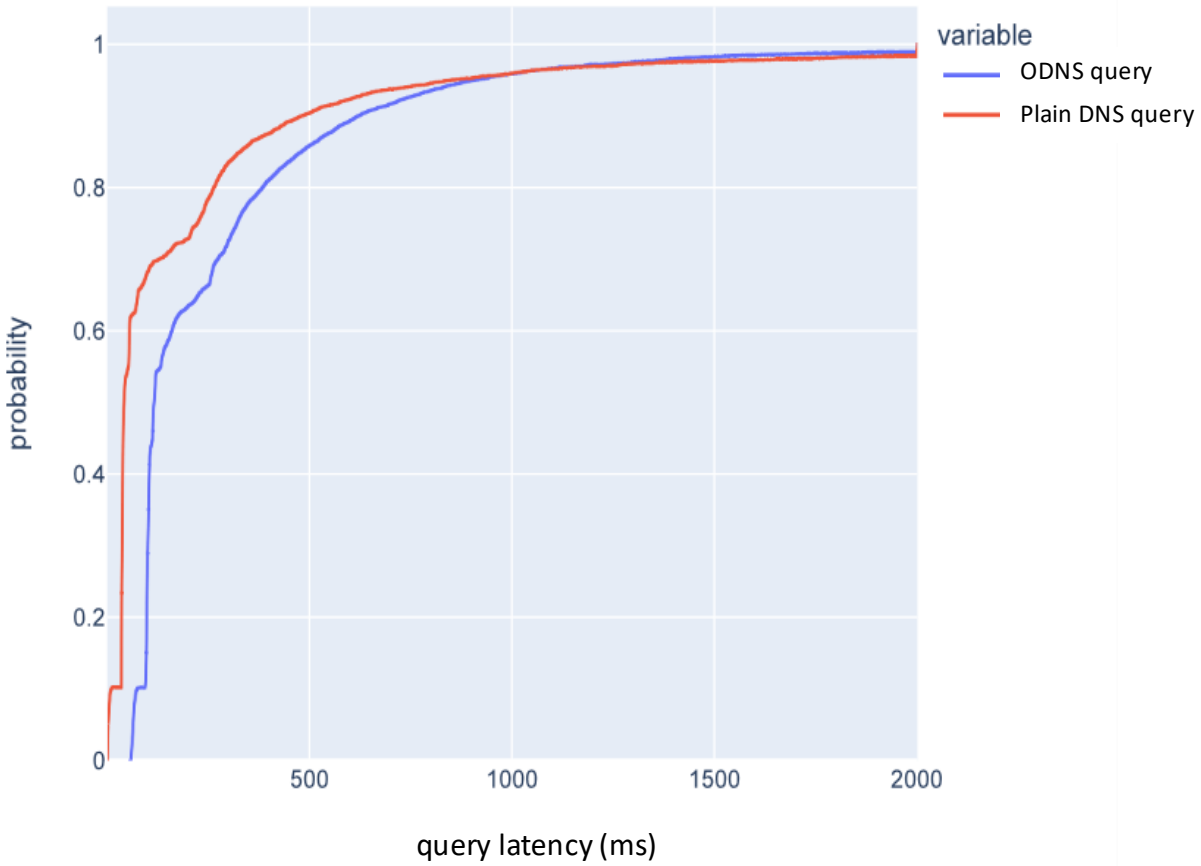
ODNS Performance - Overview



- We operate and maintain 9 ODNS resolvers worldwide deployed on VPS's.
- Our primary focus is on the end-to-end latency of DNS queries using ODNS and the additional overheads of cryptographic compute and network compared to plaintext DNS.

ODNS Performance - Query Latency

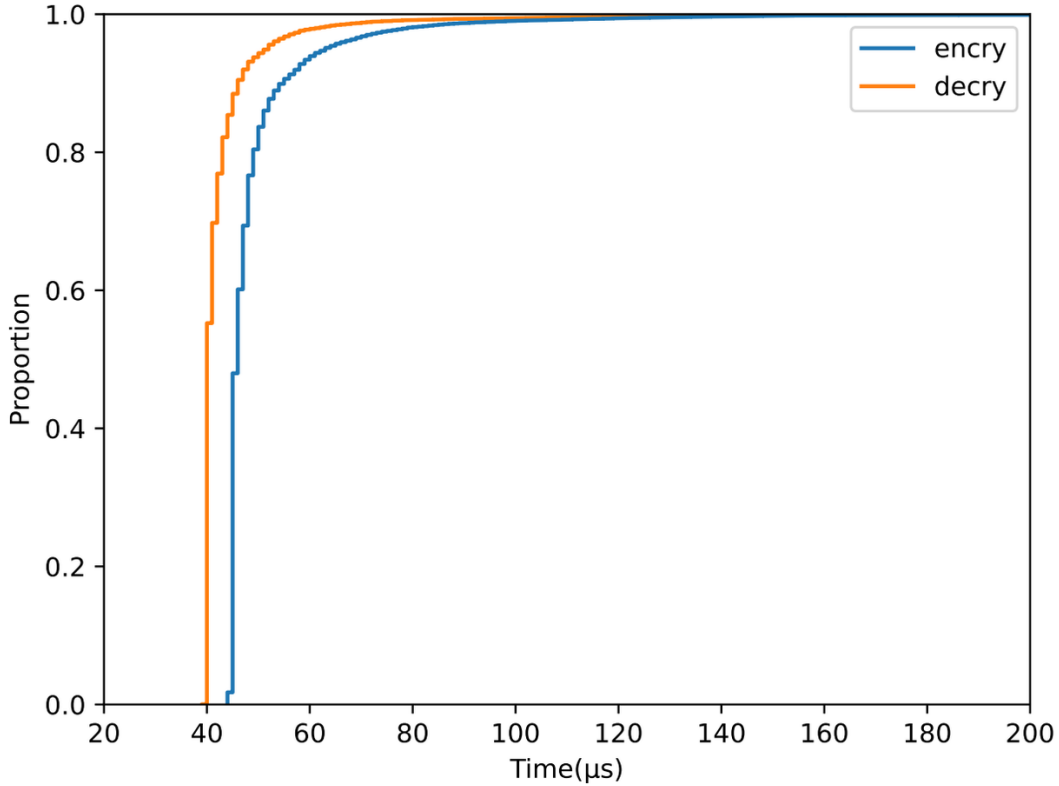
- Query 10k domains from the Tranco top 1M list
- ODNS
  - User stub and dedicated resolvers located in Beijing
  - Both use ISP resolver as next hop
  - Pre-query to cache public key for user and ODNS resolver address for ISP resolver
- Plaintext DNS
  - Use ISP resolver
- 80% of ODNS queries' latency < **500**ms
- ODNS's average additional overhead = **85.24**ms
  - Due to the extra network transmission between the public DNS resolvers and the ODNS resolvers.



CDF of ODNS query latency

ODNS Performance - Cryptographic Compute

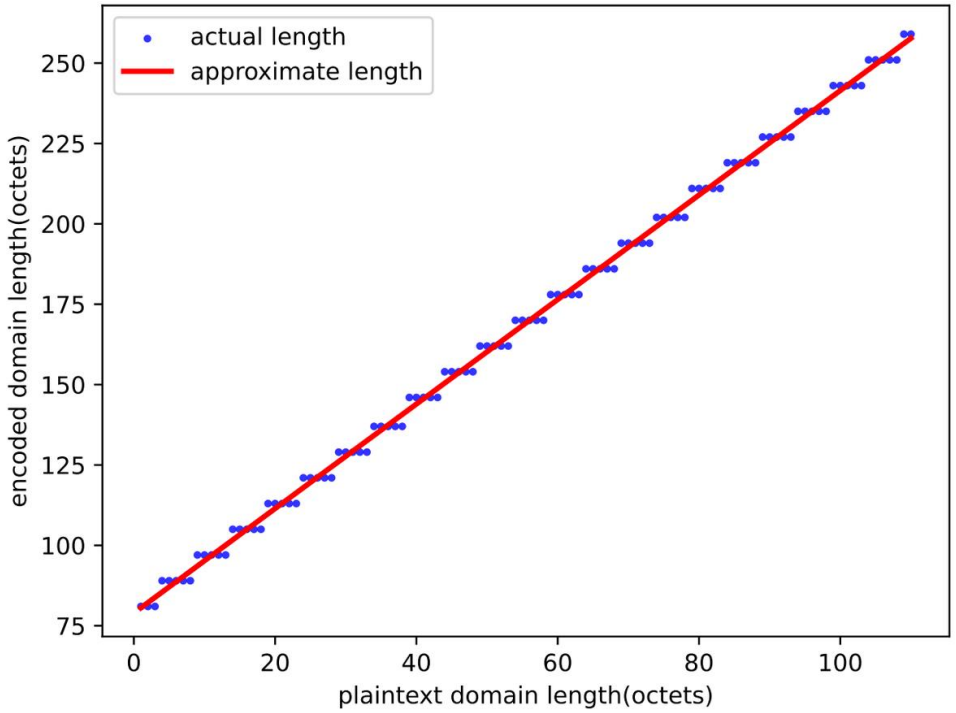
- Encoding/decoding 10k domains from the Tranco top 1M list
- ECIDS(Elliptic Curve Integrated Encryption Scheme)
  - Curve: Secp256r1
  - KDF: SHA256 HKDF
  - ENC: AES-128-CTR
  - Base32 encoding
- Encoding/decoding: 99/75μs (99th percentile)
  - One encoding and one decoding during the entire process
  - Only encrypt the domain name field



CDF of ODNS Cryptographic Compute time

ODNS Performance - Network

- Encoding 10k domains from the Tranco top 1M list
- The length of the encoded domain name: $1.625L + J + 65$
  - L is the length of plaintext domain name
  - J is the length of ODNS resolver's domain suffix
  - L can be a maximum of 108 bytes if J is 14
  - Sufficient in practice
- DNS compression
  - Only one encoding overhead in the DNS message
- Average size of plaintext/ODNS query: 31.6 / 119.3 bytes
  - Same for response



Length of domain before and after encoding

Conclusion

- Implement ODNS to improve DNS security and privacy
- Evaluate the performance of ODNS, including query latency, cryptographic computation and network overheads
- Compared to simillar protocols, the inherent advantages of ODNS, such as compatibility with DNS infrastruction and resistance to collusion, are still appealing
- We are working with 114DNS to implement ODNS as an optional feature, and operating a series of ODNS servers worldwide for experimental purposes

清華大學
Tsinghua University

Thanks for listening!

Dashuai Wu (Speaker), Shibo Cui, Baojun Liu

Tsinghua University, Network and Information Security Lab (NISL)

Deliang Chang

QI-ANXIN Technology Research Institute

ICANN DNS Symposium, September 2024, Remote