Demo: PDNS: A Fully Privacy-Preserving DNS

Yunming Xiao[1], Chenkai Weng[1], Ruijie Yu[1], Peizhi Liu[1], Matteo Varvello[2], Aleksandar Kuzmanovic[1]
[1] Northwestern University, [2] Nokia Bell Labs
{yunming.xiao,ckweng,ruijieyu2022,peizhiliu2023,akuzma}@northwestern.edu,matteo.varvello@nokia.com

ABSTRACT

The Domain Name System (DNS) is a key component of Internet-based communication and its privacy has been neglected for years. Recently, DNS over HTTPS has improved the situation by fixing the issue of in-path middleboxes. Further progress has been made with proxy-based solutions such as Oblivious DoH, which separate a user's identity from their DNS queries. However, these solutions rely on non-collusion between DNS resolvers and proxy networks. This paper instead proposes PDNS, a new DNS extension that uses Private Information Retrieval to allow DNS resolvers to operate on blind queries, thereby eliminating any privacy leaks.

CCS CONCEPTS

• **Networks** → **Transport protocols**; **Naming and addressing**; **Network privacy and anonymity**; **Network security**; **Application layer protocols**; • **Computer systems organization** → **Dependable and fault-tolerant systems and networks**; • **Security and privacy** → **Cryptography**;

KEYWORDS

Domain Name Service; Private Information Retrieval

ACM Reference Format:
Yunming Xiao[1], Chenkai Weng[1], Ruijie Yu[1], Peizhi Liu[1], Matteo Varvello[2], Aleksandar Kuzmanovic[1]. 2023. Demo: PDNS: A Fully Privacy-Preserving DNS. In *ACM SIGCOMM 2023 Conference (ACM SIGCOMM '23), September 10, 2023, New York, NY, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3603269.3610860

1 INTRODUCTION

The DNS is the phonebook of the Internet [2] which maps IP addresses like "151.101.195.5" to human-friendly names like "cnn.com". Initially, DNS employs UDP or TCP and leaves the payloads as plaintext [4], allowing any middleboxes placed between a DNS client and *recursive resolver* (ReR) to monitor user activity. Twenty-eight years later, DNS-over-TLS (DoT) [6] and DNS-over-HTTPS (DoH) [10] solve this limitation by mean of end-to-end encryption. More recently, ODNS [19] moves a step further to protect user privacy from ReR by adding a proxy between the DNS client and ReR such that: (*i*) the proxy is blind with respect to an encrypted DNS query, (*ii*) the ReR is blind with respect to the client's identity (IP address). Assuming a non-colluding proxy and ReR, user privacy is enforced. However, non-collusion is hard to enforce and verify in reality. For example, it could be broken when faced with a court subpoena. Finally, ODNS still allows the ReR to gain insights into users collectively, e.g., answer questions like *"what is the most popular online newspaper, and its potential political affiliation, in a given region?"*

Table 1: Comparison of privacy-preserving properties of various DNS solutions.

Solution	Defend Pervasive Monitoring	Hide Individual Access Pattern	Hide Regional Access Pattern	Survive Non-Collusion Violation
DoUDP [4] / DoTCP [6]	No	No	No	N/A
DoT [12] / DoH [10]	Yes	No	No	N/A
DoT/DoH + Resolver Rotation [11, 18]	Yes	Yes*	No	N/A
ODNS [19] / ODoH [20]	Yes	Yes	No	No
ODNS/ODoH + Proxy Rotation [13]	Yes	Yes	Yes*	No
DoHoT [15, 16]	Yes	Yes	Yes*	Yes*
DNS with Multi-Server PIR	Yes	Yes	Yes	No
DNS with Single-Server PIR	Yes	Yes	Yes	Yes

The only way to guarantee full user privacy would be for a ReR to operate *in the blind, i.e.,* by resolving domain names without knowing what they are. The latter statement seems counter-intuitive, but in reality several techniques exist which allow such operations. These techniques fall in the branch of Private Information Retrieval (PIR), which is achieved by various cryptographic tools such as homomorphic encryption [5, 7–9, 17]. Indeed, private DNS is often cited as a motivating example in PIR research, but no practical implementation currently exists.

Table 1 summarizes the privacy-preserving properties of state-of-the-art DNS solutions. No existing solution but only single-server PIR has the potential to offer full privacy. The goal of this work is to fill the gap between PIR and DNS research. We do so by designing PDNS, a Privacy-Preserving DNS designed to *augment* rather than replace DNS, in a spirit similar to DoH/ODoH. To achieve our vision, we had to solve the following challenges.

PIR Selection and Optimization: Out of all the available PIR categories, we suggest utilizing the single-server stateless PIR schemes for DNS. These schemes do not require a non-collusion agreement, result in low costs for cache updates, and offer satisfactory running times for query processing. We benchmarked multiple such schemes and Spiral [14] stands out with the fastest running time, the shortest query size, and high-quality open-source implementation. To integrate Spiral into PDNS, we conducted research on the optimal DNS cache configuration for PIR. We also exploit multi-threading and low-level instruction support. We further implemented performance by leveraging hash collisions to purposely build large cache slots, which reduces the query time at the expense of more data being returned to the user.

ACM SIGCOMM '23, September 10, 2023, New York, NY, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0236-5/23/09.
https://doi.org/10.1145/3603269.3610860

Cache Population: PIR protocols assume that a database (or cache in DNS context) is either given or can be privately populated. This is not the case for DNS where the ReR is responsible to populate its cache based on the user request. Clearly, a *blind* ReR cannot perform such operation which should be tackled by the client instead. Still, the client cannot update the ReR cache or it would invalidate the system privacy. We propose EDNS(1), our own DNS extension which allows a client to communicate the IP address of its ReR in presence of cache misses, so that an authoritative name server can privately populate the ReR's cache.

We implement a prototype client and ReR of PDNS and extend the popular BIND9 [3] to support EDNS(1) at authoritative name servers. Our experiments show that PDNS outperforms the DoHoT and can achieve similar performance with ODoH with a small cache (64MB, up to 1.6M DNS records).

One final question remains: *what are the incentives for the adoption of PDNS?* For users, the extra privacy provided justifies the minor performance penalty. For the ReR, the extra cost is justified by unprecedented privacy guarantees, which could be offered at a premium. Participating authoritative name servers also have an incentive to support PDNS, as the additional traffic is offset by the increased privacy they can provide to their users, which is very valuable to domains offering sensitive content.

2 PRIVACY MODELS

We assume *untrusted* ReRs which may track DNS queries to violate user privacy. Similarly, we assume that DNS traffic can be intercepted by third parties, *i.e.,* middleboxes interposing between DNS clients and both ReRs and/or name servers. Finally, a ReR may also deliberately drop the DNS record of a sensitive domain from its cache, and infer which user might query such domain when the record is populated into the cache again.

We instead assume that name servers can be *trusted* with respect to user privacy. Indeed, root and TLD name servers are handled by trusted Internet authorities such as IANA (Internet Assigned Numbers Authority) [1]. Further, users can submit only partial domain names, e.g., querying only ".com" instead of the full domain name from the root name servers, and preserve some privacy when sending requests to them. With respect to authoritative name servers, their organizations have no incentives in using DNS traffic to infer a user visit, since they already have access to both IP and HTTP logs, either directly or provided by a content delivery network (CDN).

3 SYSTEM DESIGN

PDNS workflow consists of three main parts: initialization, query, and cache update (see Figure 1). Define N as the maximum number of PIR cache entries and X as the number of DNS records. The PIR cache entries are of the same length and each of them may be filled with multiple DNS records.

Initialization: Given as input a DNS cache $C := (c_1, \ldots, c_N)$, the PDNS ReR executes $\mathsf{SetupServer}(C, N) \rightarrow [C]$ to encode the cache. Upon registering to a PDNS ReR, the user executes $\mathsf{SetupUser}(N) \rightarrow (\mathsf{qk}, \mathsf{pk})$ to derive the query key qk and public key pk. The user sends pk to the ReR, who needs it to answer private DNS queries. This is a per-user key which can be shared across multiple ReRs, e.g., in the case of a DNS leveraging multiple physical machines for load balancing.

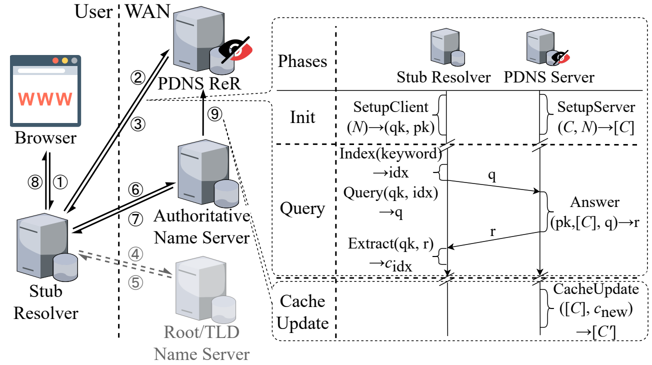

Figure 1: Visualization of PDNS and its workflow.

Query: A DNS query in PDNS implies the following steps:

(1) A user who wants to visit a domain d executes $\mathsf{Index}(\mathsf{d}) \rightarrow \mathsf{idx}$. The index idx points to a specific slot in ReR's cache, where the DNS record for d might be located. User executes $\mathsf{Query}(\mathsf{qk}, \mathsf{idx}) \rightarrow \mathsf{q}$ and sends the encrypted query q to the PDNS ReR.
(2) The PDNS ReR executes $\mathsf{Answer}(\mathsf{pk}, [C], \mathsf{q}) \rightarrow \mathsf{r}$. The output r is a ciphertext which encrypts the corresponding cache slot, if available, and is kept secret from ReR. Finally, the ReR sends r to user.
(3) User executes $\mathsf{Extract}(\mathsf{qk}, \mathsf{r}) \rightarrow c_{\mathsf{idx}}$. If c_{idx} contains a valid DNS record for the domain d, the DNS query is terminated. Otherwise, the user performs an iterative DNS lookup. Note that PDNS attempts to speed up such iterative DNS lookup by providing in c_{idx} the NS-record of d, or the IP address of the authoritative name server for d. The authoritative name server could optionally asks the user to prove the existence of cache miss (see § 4).

Cache update: The cache update happens after a cache miss is triggered and the user finishes an iterative DNS lookup for a domain d. The authoritative name server for d populates the PDNS ReR's cache by sending its most recent DNS record for d. As long as the authoritative name server does not collude with the ReR, user's privacy is maintained. This non-collusion requirement is not a violation of our privacy model; this is because the client-side iterative DNS lookup does not offer any more information to the organization behind a domain than what is already available via both IP and HTTP logs, as discussed in § 2.

4 FUTURE WORK

The PDNS construction above imposes new security challenges for DNS. Attackers can either congest authoritative name servers, or launch *reflection* attacks to congest or poison the cache of ReRs. We leave the defense of such security vulnerabilities to future work. Briefly, we plan to leverage a combination of the security properties of Spiral with digital signatures and PKI-based authentication to allow authoritative name servers to *validate* cache misses when needed, *i.e.,* when suspecting a potential attack.

REFERENCES

[1] Internet Assigned Numbers Authority. https://www.iana.org/.

[2] What is DNS? | How DNS works | Cloudflare. https://www.cloudflare.com/learning/dns/what-is-dns/.

[3] When to replace BIND DNS. https://bluecatnetworks.com/blog/when-to-replace-bind-dns/.

[4] Domain names - implementation and specification. RFC 1035, Nov. 1987.

[5] Z. Brakerski, C. Gentry, and V. Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS ’12, page 309–325, New York, NY, USA, 2012. ACM.

[6] J. Dickinson, S. Dickinson, R. Bellis, A. Mankin, and D. Wessels. DNS Transport over TCP - Implementation Requirements. RFC 7766, Mar. 2016.

[7] J. Fan and F. Vercauteren. Somewhat practical fully homomorphic encryption. *IACR Cryptology ePrint Archive*, 2012.

[8] C. Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford, CA, USA, 2009. AAI3382729.

[9] C. Gentry, A. Sahai, and B. Waters. Homomorphic encryption from learning with errors: Conceptually-simpler, asymptotically-faster, attribute-based. In R. Canetti and J. A. Garay, editors, *Advances in Cryptology – CRYPTO 2013*, pages 75–92, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[10] P. E. Hoffman and P. McManus. DNS Queries over HTTPS (DoH). RFC 8484, Oct. 2018.

[11] A. Hounsel, P. Schmitt, K. Borgolte, and N. Feamster. Encryption without centralization: distributing dns queries across recursive resolvers. In *Proceedings of the Applied Networking Research Workshop*, pages 62–68, 2021.

[12] Z. Hu, L. Zhu, J. Heidemann, A. Mankin, D. Wessels, and P. E. Hoffman. Specification for DNS over Transport Layer Security (TLS). RFC 7858, May 2016.

[13] J. Kurihara and T. Kubo. Mutualized oblivious dns (μ odns): Hiding a tree in the wild forest. *arXiv preprint arXiv:2104.13785*, 2021.

[14] S. J. Menon and D. J. Wu. Spiral: Fast, high-rate single-server PIR via FHE composition. In *IEEE S&P*, 2022.

[15] A. Muffett. DNS over Tor. https://developers.cloudflare.com/1.1.1.1/other-ways-to-use-1.1.1.1/dns-over-tor/.

[16] A. Muffett. DoHoT: making practical use of DNS over HTTPS over Tor. https://medium.com/@alecmuffett/dohot-making-practical-use-of-dns-over-https-over-tor-ef58d04ca06a.

[17] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 56(6), sep 2009.

[18] S. Rivera, V. K. Gurbani, S. Lagraa, A. K. Iannillo, and R. State. Leveraging ebpf to preserve user privacy for dns, dot, and doh queries. In *Proceedings of the 15th International Conference on Availability, Reliability and Security*, pages 1–10, 2020.

[19] P. Schmitt, A. Edmundson, A. Mankin, and N. Feamster. Oblivious dns: Practical privacy for dns queries. *Proceedings on Privacy Enhancing Technologies*, 2019(2):228–244, 2019.

[20] S. Singanamalla, S. Chunhapanya, J. Hoyland, M. Vavrusa, T. Verma, P. Wu, M. Fayed, K. Heimerl, N. Sullivan, and C. A. Wood. Oblivious DNS over HTTPS (odoh): A practical privacy enhancement to DNS. *Proc. Priv. Enhancing Technol.*, 2021(4):575–592, 2021.