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

Introduction

The Domain Name System (DNS) is a key component of Internet-based communication, yet its privacy has been neglected for years. Recently, DNS over HTTPS (DoH)[1] has improved the situation by fixing the issue of in-path middle-boxes. Further progress has been made with proxy-based solutions such as Oblivious DoH (ODoH)[5], which separate a user's identity from their DNS queries. However, these solutions rely on non-collusion between DNS resolvers and proxy networks. Further, ODNS still allows the recursive resolver (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?*"

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. While the latter statement seems counter-intuitive, they can be realized through Private Information Retrieval (PIR)[7], leveraging a set of cryptographic tools such as homomorphic encryption. 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.

Solution	When Cache Updates	Performance	# Number of Slots (S=64B) 2^{16}	2^{18}	2^{20}	Slot Size (NumSlots=2^{20}) 128B	512B	2,048B
SimplePIR [56]	Update required for server and every user	Update (MB/user)	7.4	14.7	29.5	42.4	86.8	178.1
		Query Duration (ms)	4.04	8.51	19.07	30.29	80.87	256.38
		Query Comm. (KB)	7	14	28	41	84	173
SealPIR [28]	Server update only	Update (μs/slot)	2.21	2.19	2.19	4.22	16.36	78.1
		Query Duration (ms)	117	301	902	1,636	5,831	25,338
		Query Comm. (KB)	278	278	278	278	278	278
Spiral [73]	Server update only	Update (μs/slot)	37.62	62.28	31.36	31.34	63.63	298.31
		Query Duration (ms)	249	501	794	797	1,423	3,882
		Query Comm. (KB)	30	30	36	36	36	36

Figure 1. Summary and performance analysis of state-of-the-art single-server PIR solutions.

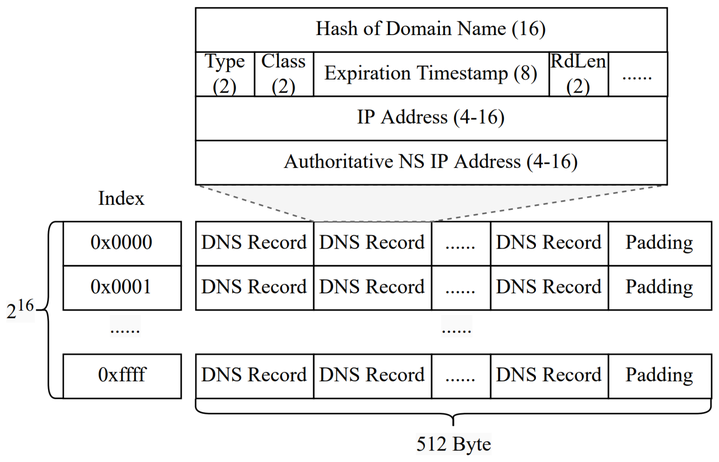

Figure 2. PDNS cache construction with large slot.

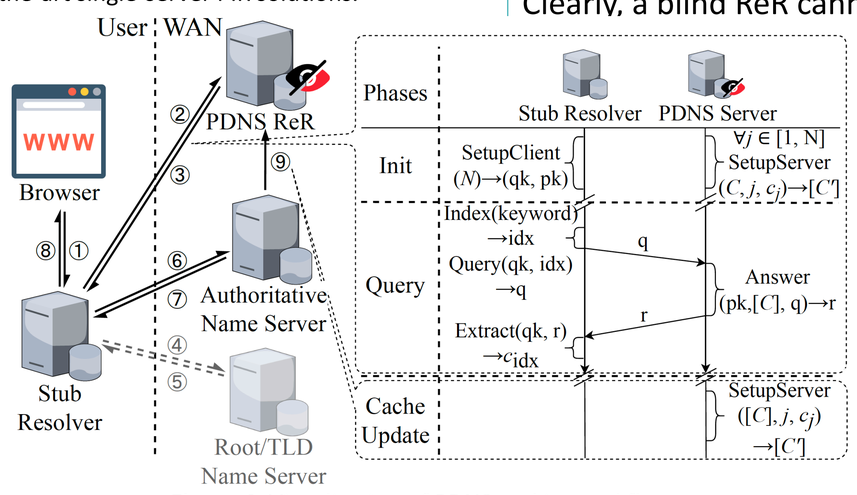

Figure 4. Visualization of PDNS and its workflows.

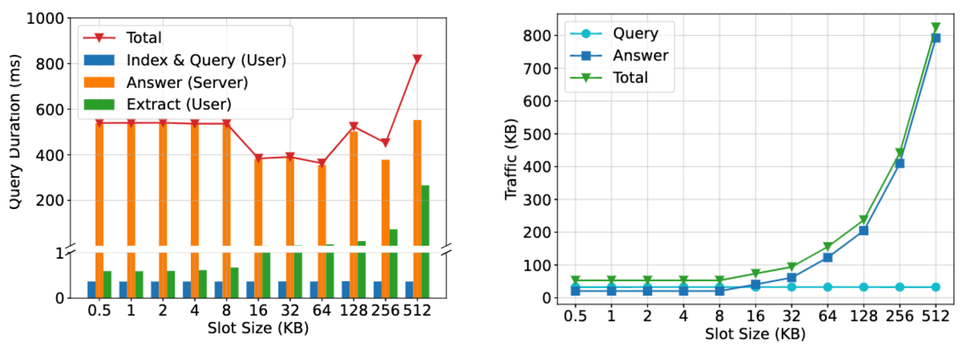

Figure 3. PIR optimization for PDNS. Given a 512MB cache, we have (a) Query duration as a function of slot size; (b) Query and answer traffic as a function of slot size.

Incentive

Users: Extra privacy provided justified for minor performance penalty (**Figure 5**) and subscription fees.

PDNS ReRs: The extra cost is justified by unprecedented privacy guarantees, which could be offered at a premium (est. $2-$5 dollars per user per month).

Participating Authoritative Name Servers: The additional traffic is offset by the increased privacy they can provide to their users, which is very valuable to domains offering sensitive content.

Solution	Defend Pervasive Monitoring	Hide Individual Access Pattern	Hide Regional Access Pattern	Survive Non-Collusion Violation
DoUDP/DoTCP	No	No	No	N/A
DoH[1]/DoT[2]	Yes	No	No	N/A
DoH/DoT + Rotation[3]	Yes	Yes*	No	N/A
ODNS[4]/ODoH[5]	Yes	Yes	No	No
ODNS/ODoH + Rotation	Yes	Yes	Yes*	No
DoHoT[6]	Yes	Yes	Yes*	Yes*
DNS + Multi-Server PIR	Yes	Yes	Yes	No
DNS + Single-Server PIR	**Yes**	**Yes**	**Yes**	**Yes**

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

Issues and Solutions

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[7] stands out with the fastest running time, the shortest query size, and high-quality open-source implementation (**Figure 1**).

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 (**Figure 2**), which reduces the query time at the expense of more data being returned to the user (**Figure 3**).

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, as shown in **Figure 4**. Still, the client cannot update the ReR cache or it would invalidate the system privacy. We propose EDNS-PR, our own EDNS(0) 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 (⑨ **in Figure 4**).

New Security Vulnerability. Following our new design, we need to protect both the authoritative name servers from DDoS attacks, and also protect PDNS ReR from malicious cache pollution.

For authoritative name server, we propose a "challenge" mechanism. Specifically, the name servers keeps track of the time at which it populated a domain record at a PDNS ReR. If a request is made before the cache is supposed to expire, name server challenges the user. The user needs to send its query, response, and a temporary private key to the name server to prove there was indeed a cache miss.

For PDNS ReR, they can validate the authoritative name servers by performing an iterative DNS lookup themselves.

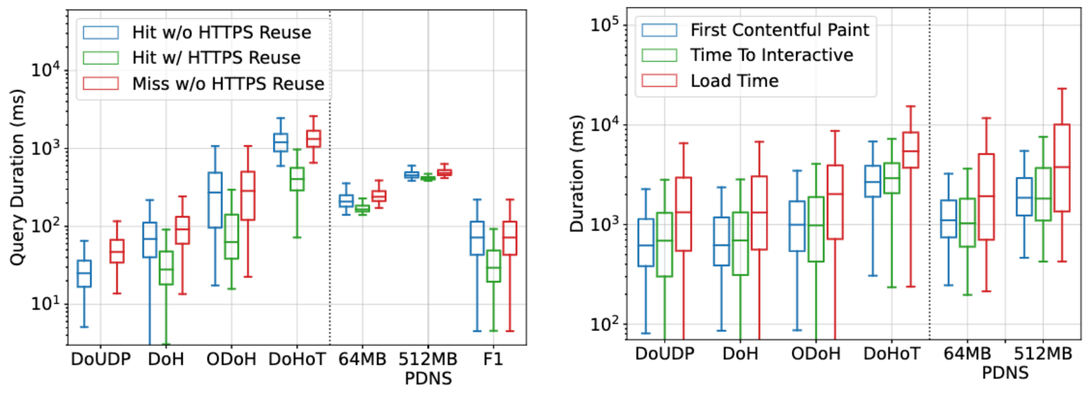

Figure 5. Performance evaluation of PDNS.* (a) Query duration; (b) Web performance.

Note

* F1[8] is a specialized hardware accelerator for PIR, which provides *at least* 1,000x speedups compared to CPU. The numbers for F1 is estimated based on such speedups.

Contact

Emails:
{yunming.xiao,ckweng,ruijieyu2022,peizhiliu2023}@u.northwestern.edu,
matteo.varvello@nokia.com,
akuzma@northwestern.edu
Website: https://yunmingxiao.github.io/projectsfolder/pdns/

References

1. P. E. Hoffman and P. McManus. "DNS Queries over HTTPS (DoH)". RFC 8484, 2018.
2. Hu et al. "Specification for DNS over Transport Layer Security (TLS)". RFC 7858, 2016.
3. Hounsel et al. "Encryption without centralization: distributing dns queries across recursive resolvers". In Proceedings of *ANRW*, 2021.
4. Schmitt et al. "Oblivious dns: Practical privacy for dns queries". Proc. Priv. Enhancing Technol., 2019.
5. Singanamalla et al. "Oblivious DNS over HTTPS (odoh): A practical privacy enhancement to DNS". Proc. Priv. Enhancing Technol., 2021.
6. A. Muffett. DNS over Tor. https://developers.cloudflare.com/1.1.1.1/other-ways-to-use-1.1.1.1/dns-over-tor/.
7. S. J. Menon and D. J. Wu. Spiral: "Fast, high-rate single-server PIR via FHE composition". In IEEE S&P, 2022.
8. Samardzic et al. "F1: A fast and programmable accelerator for fully homomorphic encryption". MICRO, 2021.