Distributed Decentralized Domain Name Service

Brendan Benshoof Andrew Rosen
Anu G. Bourgeois Robert W. Harrison

Georgia State University

May 27th, 2016

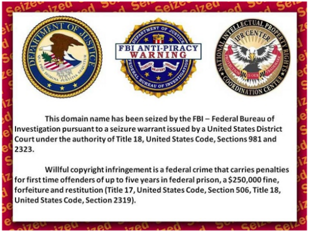
What is D^{3}NS

Distributed Decentralized Domain Name Service

- The goal is to create a secure, distributed DNS
- Requirements:
  - Decentralization
  - Authentication
  - Backwards Compatible

Motivation

- Recent events have demonstrated that centralized authorities are not as secure a previously hoped
  - There is little cryptographic protection against the subpoena
  - Poorly constructed laws targeting DNS
  - SOPA and PIPA would have resulted in DNS blocking and compromised security
- A distributed approach for authentication is much less vulnerable


DEPARTMENT OF JUSTICE
FBI ANTI-PIRACY WARNING
FEDERAL BUREAU OF INVESTIGATION
NATIONAL INTELLECTUAL PROPERTY RIGHTS COORDINATION CENTER
IPR CENTER
Seized

This domain name has been seized by the FBI – Federal Bureau of Investigation pursuant to a seizure warrant issued by a United States District Court under the authority of Title 18, United States Code, Sections 981 and 2323.

Willful copyright infringement is a federal crime that carries penalties for first time offenders of up to five years in federal prison, a $250,000 fine, forfeiture and restitution (Title 17, United States Code, Section 506, Title 18, United States Code, Section 2319).

Related Work

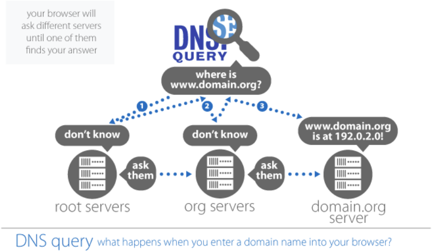
Cox *et al.*[1] developed DDNS:

- Motivated by problem of expertise
- Fault tolerant, load-balancing, and scalable
- Easier to administer
- Found higher latencies in a P2P-based DNS
- Incentive problem – why store records for others?

[1] Cox *et al.*, "Serving DNS using a Peer-to-Peer Lookup Service" in *Peer-to-Peer Systems*, pp. 155-165, Springer, 2002

Current DNS

- ICANN is the final arbiter on who owns what domain
- ICANN maintains and organizes the TLD authoritative name servers
- Third party verifiers act to authenticate DNS records


your browser will ask different servers until one of them finds your answer
DNS QUERY
where is www.domain.org?
1
2
3
don't know
don't know
www.domain.org is at 192.0.2.0!
ask them
ask them
root servers
org servers
domain.org server
DNS query what happens when you enter a domain name into your browser?

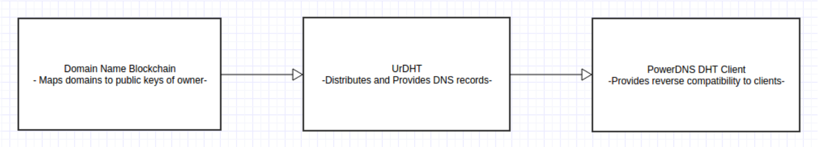
Source: ICANN

P2P-Based DNS

- The shared block chain is the final arbiter of who owns what
- The DHT organizes and maintains the authoritative TLD servers
- The block chain acts to authenticate DNS records

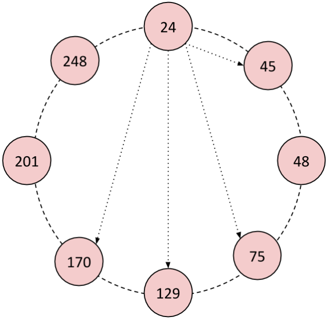
Components

- Domain Name Blockchain.
- Distributed Hash Table (UrDHT).
- DNS server frontend (PowerDNS)


Domain Name Blockchain
- Maps domains to public keys of owner-
UrDHT
-Distributes and Provides DNS records-
PowerDNS DHT Client
-Provides reverse compatibility to clients-

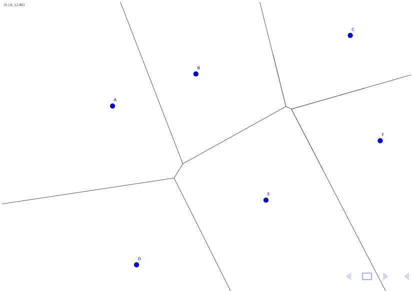
Distributed Hash Tables

- Means of organizing communication and responsibility in a P2P network
- Each peer is responsible for a verifiable span of hash values
- Facilitates one-to-one communication and one-to-many communication



UrDHT

- Uses Voronoi regions on an *n*-dimensional torus to assign responsibility.
- Can define how to compute the regions to emulate almost any DHT topology.
- Node responsibility:
  - Node is responsible for its space, defined by its neighbors.
  - If a node leaves/fails, each neighbors assumes that it is responsible until corrected by maintenance.



Fault Tolerance

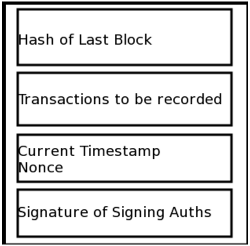
- Churn creates a period where I/O can fail.
- With UrDHT:
  - Reads of backed up data are successful.
  - Writes to the region are successful.
  - Reads of **new** data are vulnerable until it is backed up.
  - This means a much smaller window of vulnerability. Writes never fail.

Cool Things UrDHT Can Do

- Embed problem spaces into DHT topology
- Minimal latency based routing
- Basically turns routing into best-search first

DNS Blockchain

- Using a technique similar to Bitcoin, we can assign domain names as reward for mining new blocks and transfer domains between owners
- An ‘owner’ in this context is a public key
- These public keys can be used to verify stored DNS records by their signature records



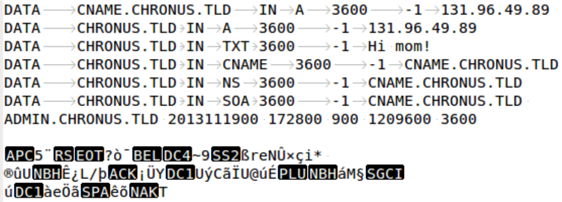
Hash of Last Block

Transactions to be recorded

Current Timestamp
Nonce

Signature of Signing Auths

Example Record

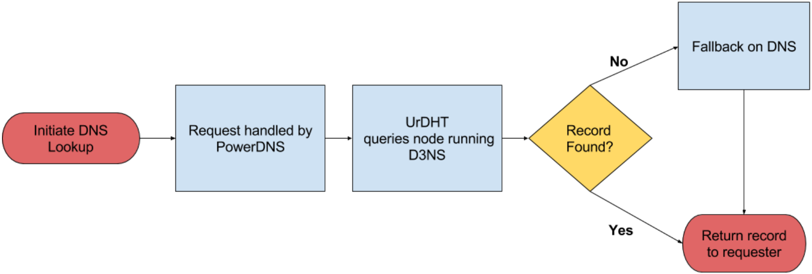


```
DATA    CNAME.CHRONUS.TLD   IN  A   3600    -1  131.96.49.89
DATA    CHRONUS.TLD IN  A   3600    -1  131.96.49.89
DATA    CHRONUS.TLD IN  TXT 3600    -1  Hi mom!
DATA    CHRONUS.TLD IN  CNAME   3600    -1  CNAME.CHRONUS.TLD
DATA    CHRONUS.TLD IN  NS  3600    -1  CNAME.CHRONUS.TLD
DATA    CHRONUS.TLD IN  SOA 3600    -1  CNAME.CHRONUS.TLD 
ADMIN.CHRONUS.TLD 2013111900 172800 900 1209600 3600

APC5¨RSEOT?ò¯BELDC4~9SS2ßreNÛ×çi* 
®ûUNBHÊ¿L/þACK¡ÜYDC1UýCãÏU@úÉPLUNBHáM§SGCI
úDC1àeÖãSPAêõNAKT
```

PowerDNS

- Well established authoritative DNS server software.
- Provides easy interface for custom applications.
- Serves the DNS requests for DHT client.


Initiate DNS Lookup
Request handled by PowerDNS
UrDHT queries node running D3NS
Record Found?
No
Fallback on DNS
Yes
Return record to requester

Man in the Middle In a DHT

- Need to have a distributed, reliable way to authenticate
- Given:
  - An existing network where nodes have exchanged keys securely
  - A new peer who wishes to join the network and share their public key

Prevention

- At least 2 members of the network interrogate the new peer for its public key
- Those interrogators compare their results
- If those results match:
    - The new peer creates an authentication record
    - The interrogators sign that record
    - The new record is distributed across the network
- If the results do not match:
    - An attack is detected and reported to the new peer by all authenticating servers.
    - A member of the network may make a ban of the compromised peer
    - Otherwise the joining process can be repeated.

Conclusions

- Deployable prototype of a decentralized and distributed top-level DNS
- Stronger robustness
- Fully reverse compatible
- Offers greater security
- Any organization can create their own secure verification server

Future Work

- Optimize caching
- Backup Comorbidity
- Further security measures