EDU Tutorial:

DNS Privacy

Sara Dickinson
Sinodun
sara@sinodun.com

EDU Tutorial @ IETF_97 Seoul (Nov 2017)

Overview

- Goal:
  - Give audience historical background on why DNS Privacy is an important topic
    - Internet Privacy - presented by dkg
- Chart progress during last 3-4 years (DPRIVE)
- Present current status and tools

Internet Privacy

Daniel Kahn Gillmor
ACLU

DNS Privacy

- A brief history

IETF Privacy activity

March 2011	**I-D:** Privacy Considerations for Internet Protocols (IAB)
June 2013	Snowdon revelations
July 2013	**RFC6973:** Privacy Considerations for Internet Protocols
May 2014	**RFC7258**: **Pervasive Monitoring is an Attack**
August 2015	**RFC7624**: Confidentiality in the Face of Pervasive Surveillance: A Threat model and Problem Statement
	Much other ongoing work…..

What timing!

RFC 7258

“**PM is an attack on the privacy of Internet users and organisations**.”

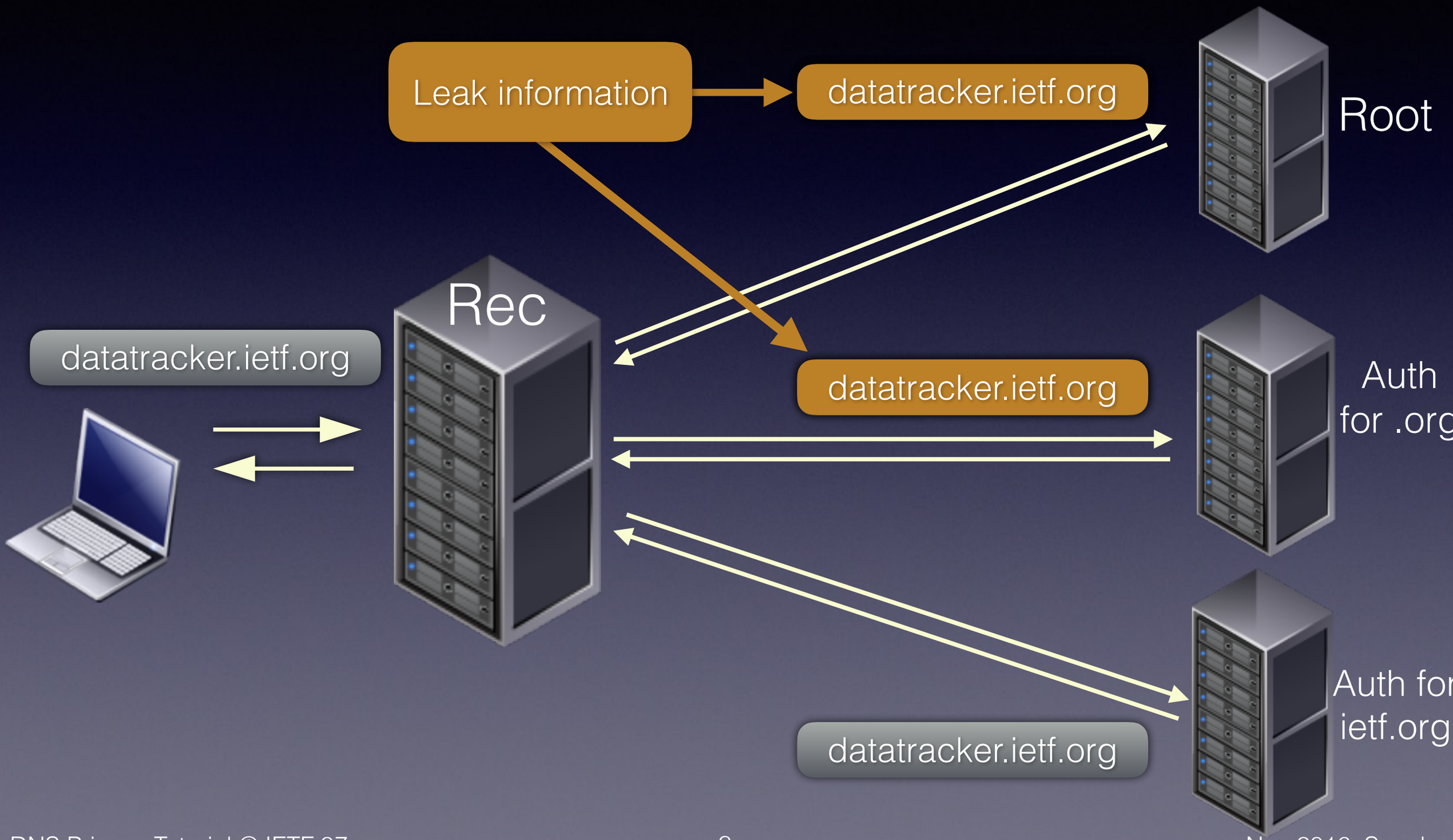
“…that needs to be **mitigated** where possible, **via the design of protocols** that make PM significantly more expensive or infeasible**.** “

DNS Privacy in 2013?

- DNS [RFC1034/5 - 1987] - original design availability, redundancy and speed! (DNS is an enabler)
- DNS standards:
  - UDP (99% of traffic to root)
  - TCP only for ‘fallback’ when UDP MTU exceeded and XFR (support only mandatory from 2010)
- Perception: The DNS is public, right? It is not sensitive/personal information….it doesn’t need to be protected/encrypted

DNS sent in clear text
-> NSA: ‘MORECOWBELL’

DNS Disclosure Example 1



DNS Privacy in 2013?

- **RFC6891**: Extension Mechanisms for DNS (EDNS0)

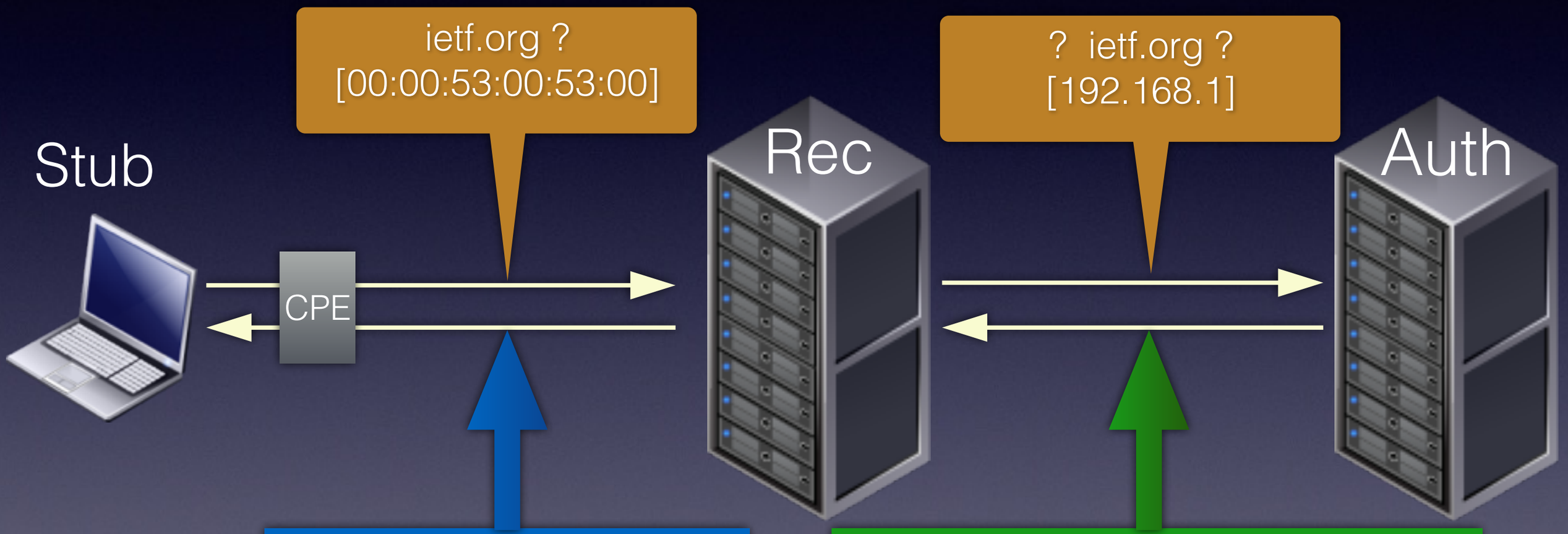
Intended to enhance DNS protocol capabilities

- But…. mechanism enabled addition of end-user data **into** DNS queries (non-standard options)

ISP justification: Parental Filtering (per device)

CDN justification: Faster content (geo location)

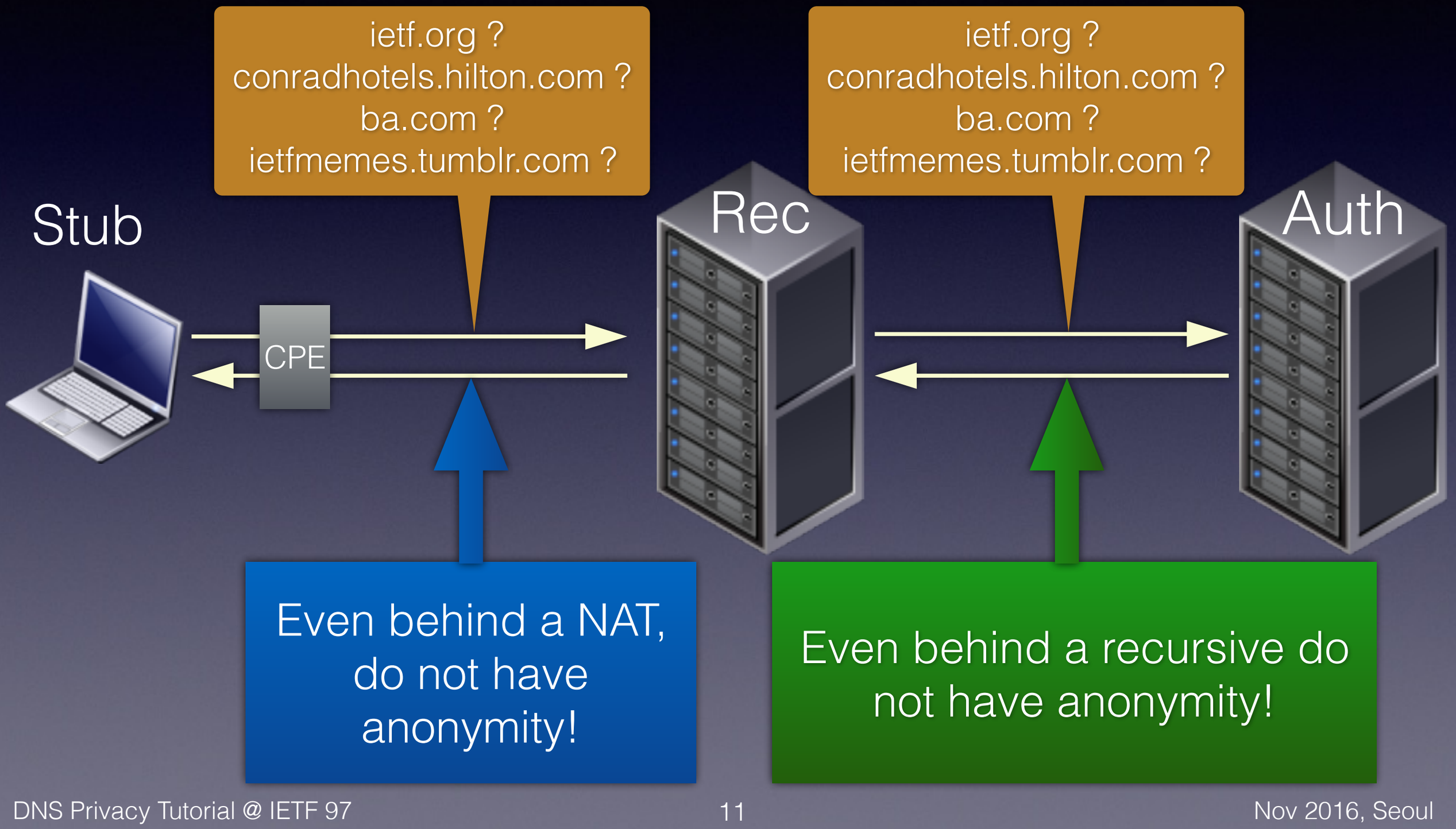
DNS Disclosure Example 2



[User src address]
MAC address or id
in DNS query

Client Subnet (RFC7871)
contains source subnet
in DNS query

DNS Disclosure Example 2



DNS Disclosure Example 3

- (AUTH) Who monitors or has access here?
- (UNAUTH) How safe is this data?

Who monitors or has access here?

- When at home…
- When in a coffee shop…

Who monitors or has access here?

DNS - complications

- Basic problem is leakage of meta data
  - Allows re-identification of individuals
- Even without user meta data traffic analysis is possible based just on timings and cache snooping
- DNS Filtering is becoming more prevalent

DNS Risk Matrix

	In-Flight		At Rest	
Risk	**Stub => Rec**	**Rec => Auth**	**At Recursive**	**At Authoritative**
Passive Monitoring	●	●		
Active Monitoring	●	●		
Other Disclosure Risks e.g. Data breaches			●	●

DNS Service Discovery

- Devices advertise services on local network (DNS, mDNS)
- Other devices then discover the service and use it

Alice's Images	. _imageStore._tcp	. local
Alice's Mobile Phone	. _presence._tcp	. local
Alice's Notebook	. _presence._tcp	. local

DNS-SD Privacy

- Advertising leaks information about:
  - User - ‘name’, devices, services (user tracking)
    - Devices - services & attributes (port, priorities)
      - Device fingerprinting possible

=> Software or specific device identification

- Discovery leaks info about preferred services

DNS Privacy options (2013)

- DNSCurve

  Recursive-Auth

  - Daniel J. Bernstein, initial interest but not adoption
- DNSCrypt

  Stub-Recursive

  - Many implementations, several open DNSCrypt Resolvers (OpenDNS), [Yandex browser]

- **Authentication** with some privacy

  Anti-spoofing, anti DoS

- Documented but not standard

DNS Privacy options (2014)

- Run a local resolver (Unbound)
- DNSTrigger (NLNet Labs)
- Client software to enable DNSSEC
- Used TLS on port 443 as last ditch attempt to enable DNSSEC (DNS-over-TLS impl)

Goal was DNSSEC, not Privacy!

DPRIVE WG

et al.

DPRIVE WG

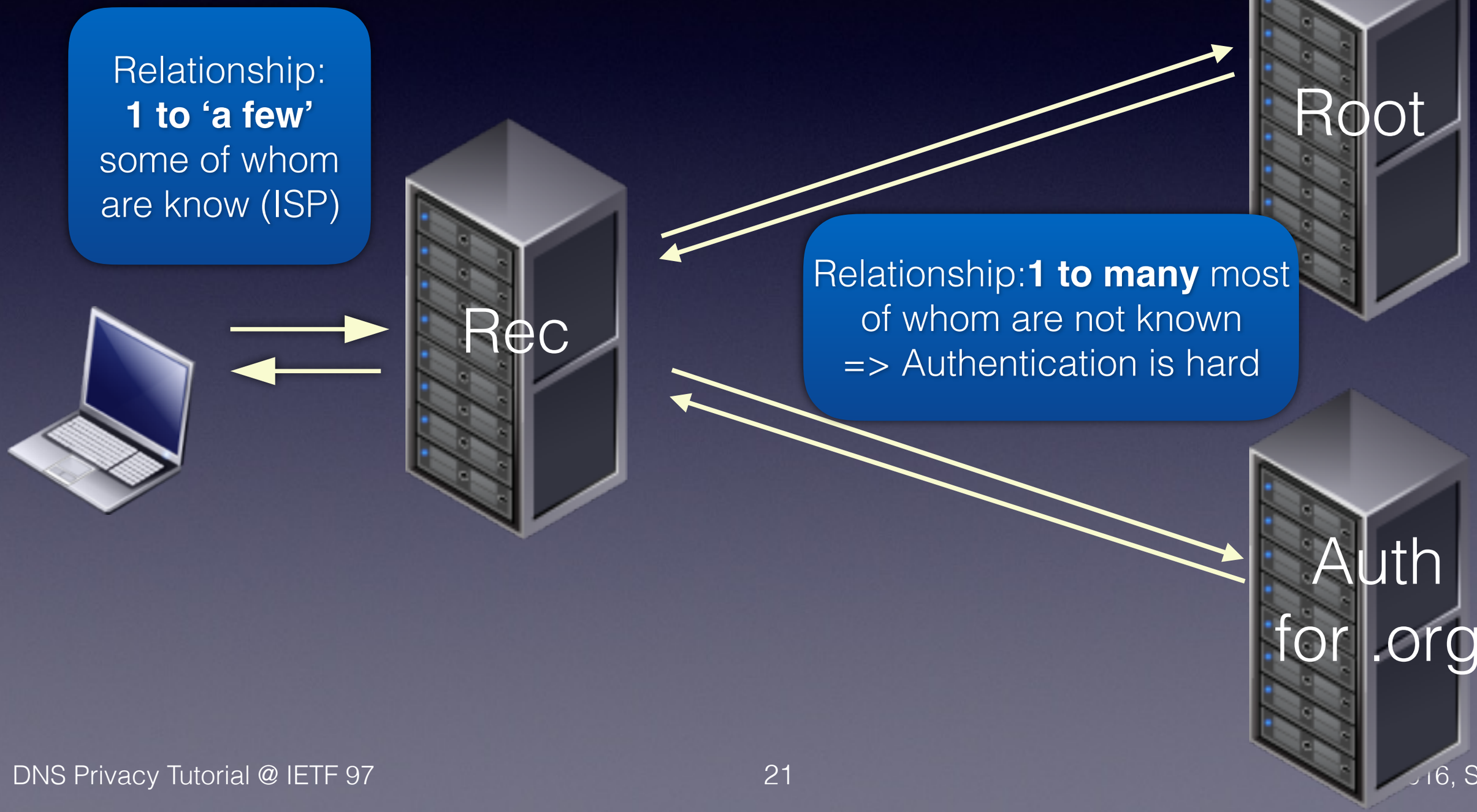
- DPRIVE WG create in 2014

Charter: Primary Focus is
Stub to recursive

- **Why not tackle whole problem?**
  - Don’t boil the ocean, stepwise solution
  - Stub to Rec reveals most information
  - Rec to Auth is a particularly hard problem

DNS Privacy problem

Relationship:
1 to 'a few'
some of whom
are know (ISP)



Relationship:**1 to many** most
of whom are not known
=> Authentication is hard

RFC 7626 - DNS Privacy Considerations

Worth a read - many interesting issues here!

- Problem statement: Expert coverage of risks throughout DNS ecosystem
- **Rebuts "alleged public nature of DNS data"**
  - The data may be public, but a DNS '**transaction**' is not/should not be.

"A typical example from outside the DNS world is: the web site of Alcoholics Anonymous is public; the fact that you visit it should not be."

Choices, choices…

- So… we know the problem but what mechanism to use for encrypting DNS?
  - STARTTLS
  - TLS
  - DTLS
  - Confidential DNS draft

Drafts submitted on all these solutions to the working group

Encryption Options

	Pros	Cons
STARTTLS	• Port 53 • Known technique • Incrementation deployment	• Downgrade attack on negotiation • Port 53 - middleboxes blocking? • Latency from negotiation
TLS (new port)	• New DNS port (no interference with port 53) • Existing implementations	• New port assignment • Scalability?
DTLS (new port)	• UDP based • Not as widely used/ deployed	• Truncation of DNS messages (just like UDP) ➡️Fallback to TLS or clear text ❌Can't be standalone solution

Encrypted DNS ‘TODO’ list

- Get a new port
- DNS-over-TCP/TLS: Address issues in standards and implementations
- Tackle authentication of DNS servers (bootstrap problem)
- What about traffic analysis of encrypted traffic - msg size & timing still tell a lot!

Get a new port!

- One does not simply get a new port…
- Oct 2015 - **853** is the magic number

Your request has been processed. We have assigned the following system port number as an early allocations per RFC7120, with the DPRIVE Chairs as the point of contact:

domain-s	853	tcp	DNS query-response protocol run over TLS/DTLS
domain-s	853	udp	DNS query-response protocol run over TLS/DTLS

DNS + TCP/TLS?

- DNS-over-TCP history:
  - typical DNS clients do ‘one-shot’ TCP
    - DNS servers have **very** basic TCP capabilities
    - No attention paid to TCP tuning, robustness
    - Performance tools based on one-shot TCP

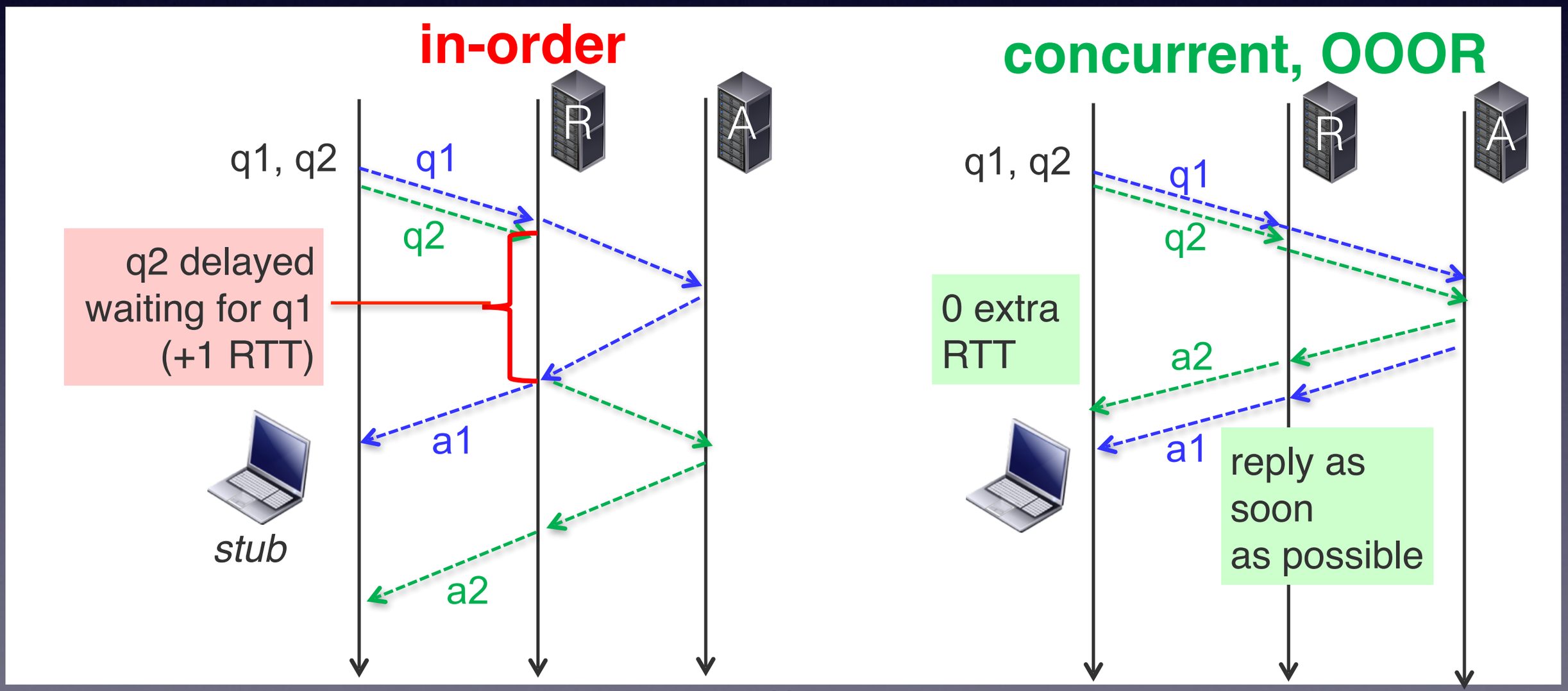
Fix DNS-over-TCP/TLS

Goal	How?
Optimise set up & resumption	TFO Fast Open TLS session resumption [TLS 1.3]
Amortise cost of TCP/TLS setup	**RFC7766** (bis of RFC5966) - March 2016: Client pipelining (not one-shot!), Server concurrent processing, Out-of-order responses **RFC7828:** Persistent connections (Keepalive)
Servers handle many connections robustly	Learn from HTTP world!

Performance (RFC7766)

Client - pipeline requests, keep connection open and handle out-of-order response

Server - concurrent processing of requests sending of out of order responses



Authentication in DNS-over-(D)TLS

2 Usage Profiles:

- Strict
  - "Do or do not. There is no try."
- Opportunistic
  - "Success is stumbling from failure to failure with no loss of enthusiasm"

(Encrypt & Authenticate) or Nothing

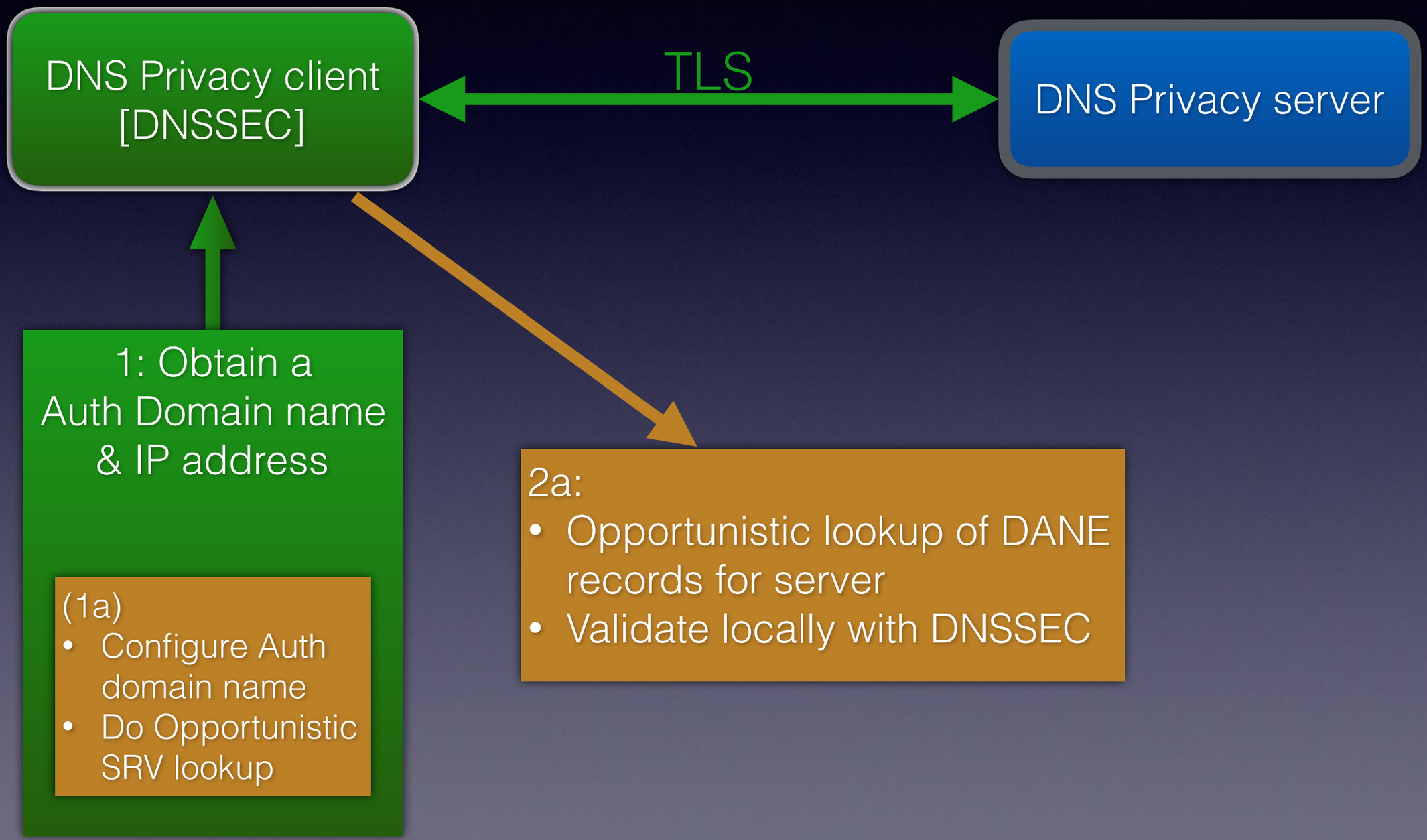
Try in order:

- Encrypt & Authenticate then
- Encrypt then
- Clear text

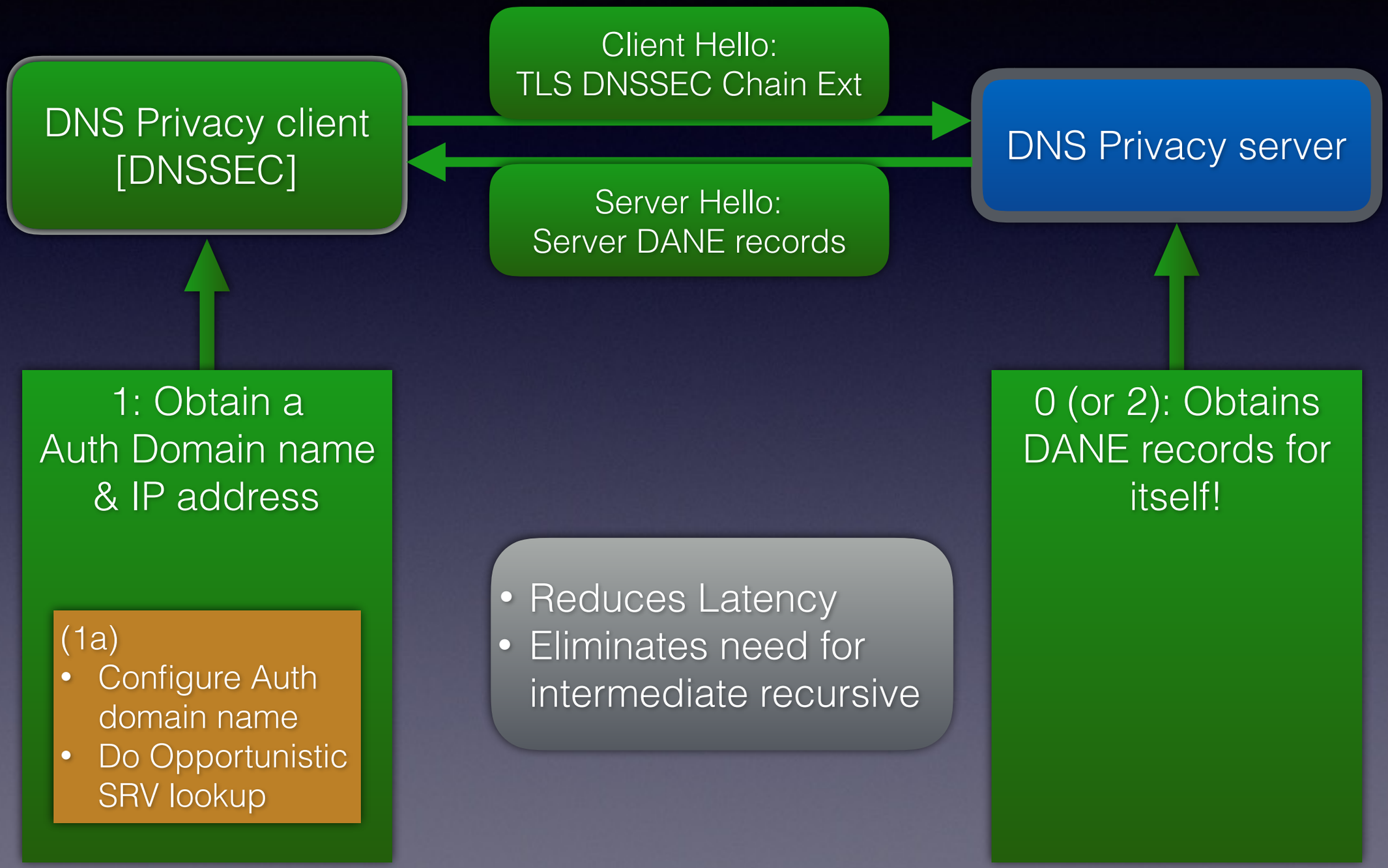
Authentication in DNS-over-(D)TLS

- Authentication based on config of either:
    - Authentication domain name
    - SPKI pinset
- Shouldn’t DNS use DANE…? Well - even better:
    - I-D: TLS DNSSEC Chain Extension

DNS Auth using DANE



TLS DNSSEC Chain Extension



DPRIVE Solution Documents (stub to recursive)

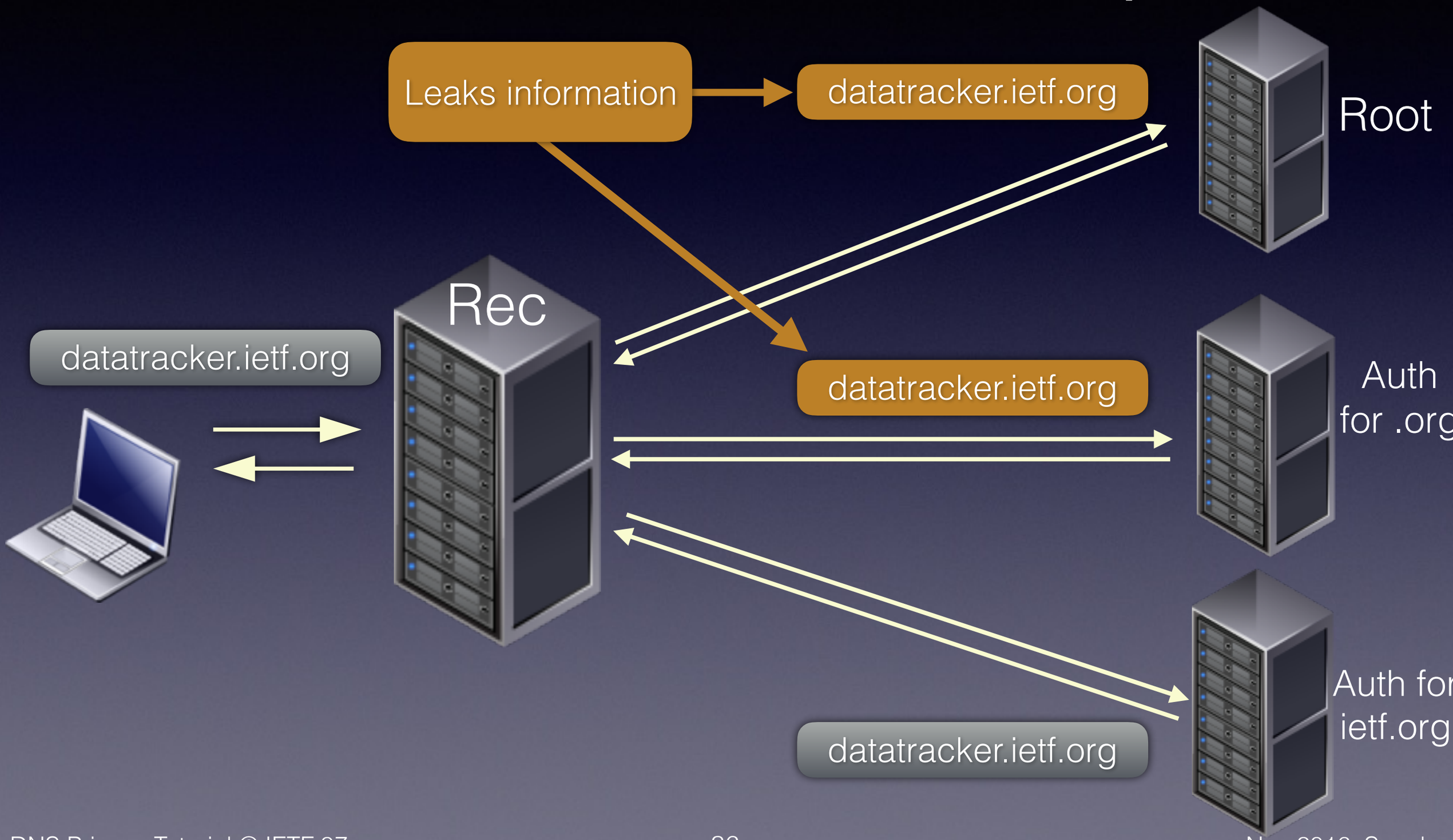
Document	Date	Topic
RFC7858	May 2016	DNS-over-TLS
RFC7830	May 2016	EDNS0 Padding Option
draft-ietf-dprive-dnsodtls*	Completed WGLC	DNS-over-DTLS
draft-ietf-dprive-dtls-and-tls-profiles	In WGLC	Authentication for DNS-over-(D)TLS

*Intended status: Experimental

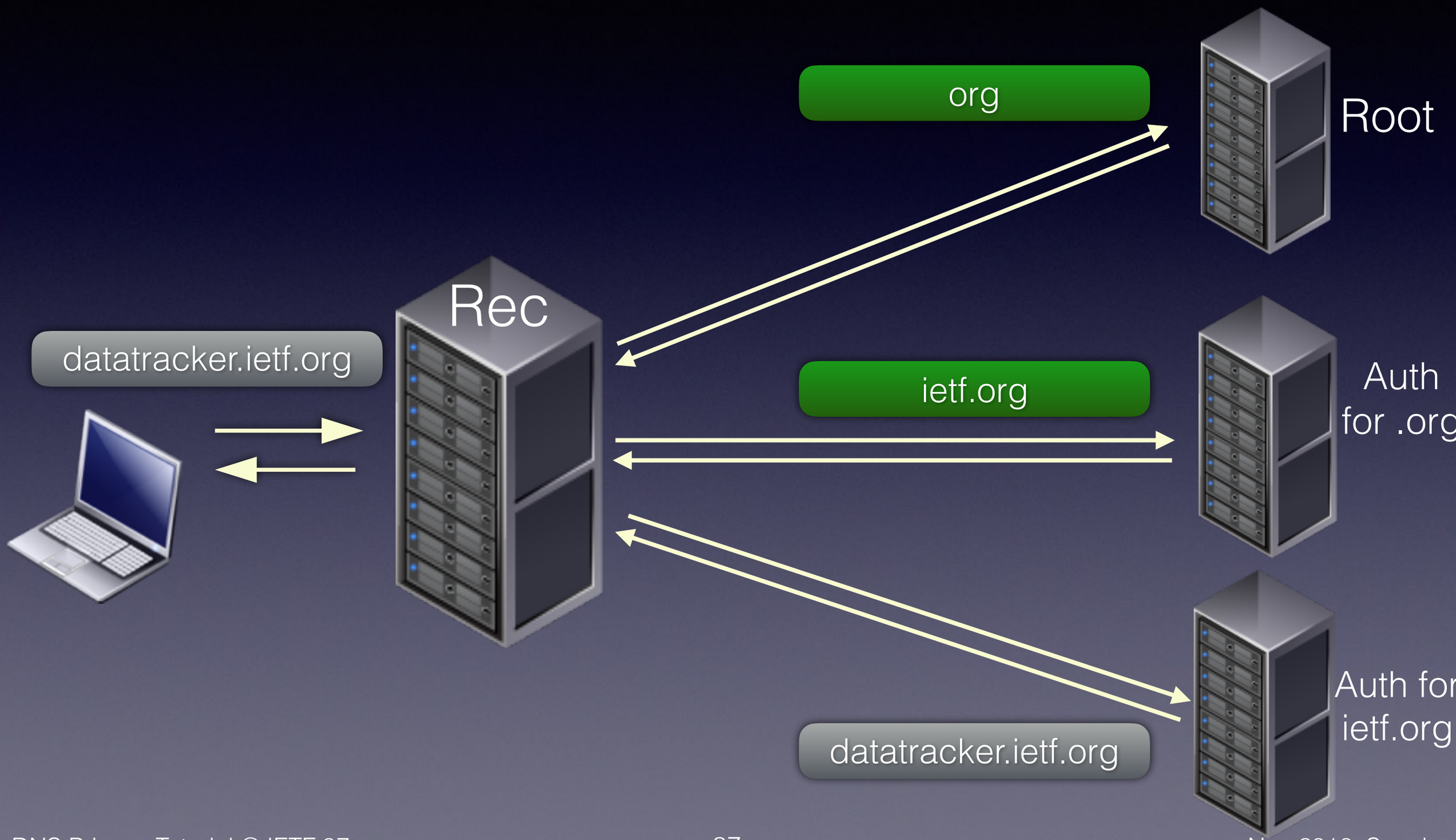
What about Recursive to Authoritative?

- DPRIVE - Re-charter? WG this Friday!
  - I-D: Next step for DPRIVE: resolver-to-auth link
    - Presents 6 authentication options/models
    - Data on DNS-over-(D)TLS
- DNSOP - RFC7816: QNAME Minimisation

DNS Disclosure Example 1



QNAME Minimisation



Data handling policies

- Do you read the small print of your ISPs contract?
- More work/research needed in this area
  - Monitoring of government policy and practice
  - Transparency from providers on policy and breaches
  - Methods for de-identification of user data (e.g. DITL)
  - ‘PassiveDNS’ data used for research/security

DNS-over-HTTP(S)

- DNS-over-HTTP(S) has been around a while…
  - I-D: Review of DNS-over-HTTP
- Privacy (HTTPS authentication)
- Bypass port 53 interference (middlebox, captive portals)
- Higher level API

DNS-over-HTTP(S)

- Google: DNS-over-HTTPS (non-standard)
- I-D: DNS wire-format over HTTP
  - “Servers and clients SHOULD use TLS for communication.”
- I-D: DNS Queries over HTTPS
- Non-WG Mailing list and Bar BOF here (Tuesday)

Risk Mitigation Matrix

	In-Flight		At Rest	
Risk	**Stub => Rec**	**Rec => Auth**	**At Recursive**	**At Authoritative**
Passive monitoring	**Encryption (e.g. TLS, HTTPS)**	**QNAME Minimization**		
Active monitoring	**Authentication & Encryption**			
Other Disclosure Risks e.g. Data breaches			**Data Best Practices (Policies) e.g. De-identification**	

DNS-SD

- I-D: Privacy Extensions for DNS-SD - adopted by WG
- 3 step design
  1. Offline pairing mechanism (shared secret)
  2. Discovery of the “Private Discovery Service”
  3. Actual Service Discovery (enc & auth conn)

Implementation Status

RECURSIVE

Recursive implementations

Features		Recursive resolver		
		Unbound	BIND	Knot Res
TCP/TLS Features	TCP fast open			
	Process pipelined queries			
	Provide OOOR			
	EDNS0 Keepalive			
TLS Features	TLS on port 853			
	Provide server certificate			
	EDNS0 Padding			
Rec => Auth	QNAME Minimisation			

- Dark Green: Latest stable release supports this
- Light Green: Patch available
- Yellow: Patch/work in progress, or requires building a patched dependency
- Purple: Workaround available
- Grey: Not applicable or not yet planned

RECURSIVE

Alternative server side solutions

- Pure TLS load balancer
  - NGINX, HAProxy
  - BIND article on using stunnel
- dnsdist from PowerDNS would be great…
  - But no support yet

Disadvantages
- server must still have decent TCP capabilities
- DNS specific access control is missing
- pass through of edns0-tcp-keepalive option

Stub implementations

Features		Stub			
		Idns	digit	getdns	BIND (dig)
TCP/TLS Features	TCP fast open				
	Connection reuse				
	Pipelining of queries				
	Process OOOR				
	EDNS0 Keepalive				
TLS Features	TLS on port 853				
	Authentication of server				
	EDNS0 Padding				

- Dark Green: Latest stable release supports this
- Light Green: Patch available
- Yellow: Patch/work in progress, or requires building a patched dependancy
- Grey: Not applicable or not yet planned

** getdns uses libunbound in recursive mode*

Implementation Status

- Increasing uptake of better DNS-over-TCP, QNAME minimisation
- Several implementations of DNS-over-TLS
- None yet of DNS-over-DTLS
- BII has DNS-over-HTTP implementation

Key is enabling end users and application developers to easily adopt DNS Privacy

Deployment Status

RECURSIVE

DNS-over-TLS Servers

Hosted by	Software
NLnet Labs	Unbound
OARC	Unbound
Surfnet (Sinodun)	Bind + HAProxy Bind + nginx
dkg	Knot Resolver
IETF?	

Find details at: DNS Test Servers

STUB

getdns
Unbound security

- Modern **async DNSSEC** enabled API
  - https://getdnsapi.net
- Written in C, various bindings (Python, Java,…)
- DNS-over-TLS, validating DNSSEC stub
- ‘Stubby’ now available for testing

Stubby

- A privacy enabling stub resolver (based on getdns_query tool)
- 1.1.0-alpha3
  - Run as daemon handling requests
  - Configure OS DNS resolution to point at 127.0.0.1

Stubby In Action

- Reads config from /etc/stubby.conf
    - domain name and SPKI pinset authentication
    - Strict and Opportunistic profiles
- How to build and use Stubby
- Demos available: Sara, Willem Toorop, Allison Mankin

Stubby in Action

```
{ resolution_type: GETDNS_RESOLUTION_STUB
, dns_transport_list: [ GETDNS_TRANSPORT_TLS ]
, upstream_recursive_servers:
  [ { address_data: 145.100.185.16
    , tls_auth_name: "dnsovertls1.sinodun.com"
    , tls_pubkey_pinset:
      [ { digest: "sha256"
        , value: 0x659B41EB08DCC70EE9D624E6219C76EE31954DA1548B0C8519EAE5228CB24150
      } ]
   } ]
, tls_authentication: GETDNS_AUTHENTICATION_REQUIRED
, listen_addresses: [ 127.0.0.1, 0::1 ]
, idle_timeout: 10000
}
```

```
[01:14:33.667974] GETDNS_DAEMON:   145.100.185.15 : Conn init
[01:15:30.746646] GETDNS_DAEMON:   145.100.185.15 : Conn closed: Conn stats     - Resp=36,Timeouts=0,Auth=Success,Keepalive(ms)=10000
[01:15:30.746687] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Resp=36,Timeouts=0,Best_auth=Success,Conns=1
[01:15:30.746698] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Conn_fails=0,Conn_shutdowns=0,Backoffs=0
[01:15:36.567899] GETDNS_DAEMON:   145.100.185.15 : Conn init
[01:16:32.377446] GETDNS_DAEMON:   145.100.185.15 : Conn closed: Conn stats     - Resp=233,Timeouts=0,Auth=Success,Keepalive(ms)=10000
[01:16:32.377545] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Resp=269,Timeouts=0,Best_auth=Success,Conns=2
[01:16:32.377578] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Conn_fails=0,Conn_shutdowns=0,Backoffs=0
[01:16:41.664881] GETDNS_DAEMON:   145.100.185.15 : Conn init
[01:16:59.188199] GETDNS_DAEMON:   145.100.185.15 : Conn closed: Conn stats     - Resp=13,Timeouts=0,Auth=Success,Keepalive(ms)=10000
[01:16:59.188265] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Resp=282,Timeouts=0,Best_auth=Success,Conns=3
[01:16:59.188284] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Conn_fails=0,Conn_shutdowns=0,Backoffs=0
[01:17:07.794347] GETDNS_DAEMON:   145.100.185.15 : Conn init
[01:17:18.745280] GETDNS_DAEMON:   145.100.185.15 : Conn closed: Conn stats     - Resp=1,Timeouts=0,Auth=Success,Keepalive(ms)=10000
[01:17:18.745350] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Resp=283,Timeouts=0,Best_auth=Success,Conns=4
[01:17:18.745372] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Conn_fails=0,Conn_shutdowns=0,Backoffs=0
[01:17:45.707624] GETDNS_DAEMON:   145.100.185.15 : Conn init
[01:17:56.670120] GETDNS_DAEMON:   145.100.185.15 : Conn closed: Conn stats     - Resp=1,Timeouts=0,Auth=Success,Keepalive(ms)=10000
[01:17:56.670188] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Resp=284,Timeouts=0,Best_auth=Success,Conns=5
[01:17:56.670211] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Conn_fails=0,Conn_shutdowns=0,Backoffs=0
[01:18:05.323299] GETDNS_DAEMON:   145.100.185.15 : Conn init
[01:18:16.207892] GETDNS_DAEMON:   145.100.185.15 : Conn closed: Conn stats     - Resp=2,Timeouts=0,Auth=Success,Keepalive(ms)=10000
[01:18:16.207974] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Resp=286,Timeouts=0,Best_auth=Success,Conns=6
[01:18:16.207997] GETDNS_DAEMON:   145.100.185.15 :               Upstream stats - Conn_fails=0,Conn_shutdowns=0,Backoffs=0
```

Ongoing and Future work

- Hacking this weekend at the IETF 97 Hackathon
    - lots of work on Stubby and test servers
- OS integration of client solutions
- More complete recursive implementations
- Increased deployment
- More DPRIVE work: Recursive to Auth….

Summary

- DNS Privacy is a real problem and more relevant than ever
- Active work on the large solution space
- Can test DNS Privacy today using Stubby & current test recursive servers
- More DNS Privacy services on the way…

Thank you!

Any Questions?

https://www.surveymonkey.com/r/97privacy