Secure Name Resolution

Christian Grothoff

Berner Fachhochschule

10.11.2017

"The Domain Name System is the Achilles heel of the Web." –Tim Berners-Lee

Background: Efficient Set Union

(based on "What's the difference? Efficient Set Reconciliation without Prior Context", Eppstein et al., SIGCOMM'11)

- Alice and Bob have sets A and B

- The sets are very large
- ...but their symmetric difference $\delta := |(A - B) \cup (B - A)|$ is small

- Now Alice wants to know $B - A$ (the elements she is missing)
- ...and Bob $A - B$ (the elements he is missing)

- How can Alice and Bob do this efficiently?
  - w.r.t. communication and computation

Bad Solution

- Naive approach: Alice sends A to Bob, Bob sends $B - A$ back to Alice
- . . . or vice versa.

- Communication cost: $O(|A| + |B|)$:(
- Ideally, we want to do it in $O(\delta)$.
- First improvement: Do not send elements of A and B, but send/request hashes. Still does not improve complexity :(

- We need some more fancy data structure!

Bloom Filters

Constant size data structure that “summarizes” a set.

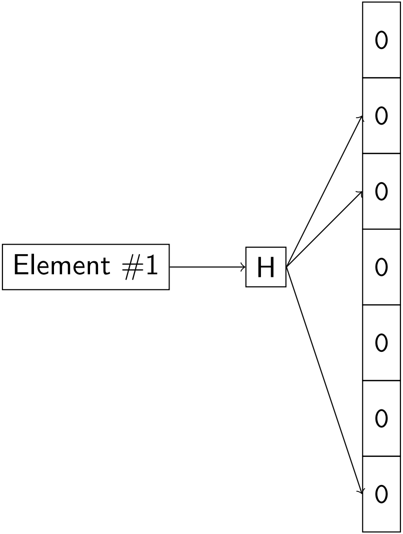
Operations:

$d = \mathit{NewBF}(\mathit{size})$ Create a new, empty bloom filter.

$\mathit{Insert}(d, e)$ Insert element e into the BF d.

$b = \mathit{Contains}(d, e)$ Check if BF d contains element e.
$b \in \{$“Definitely not in set”, “Probably in set”$\}$

BF: Insert



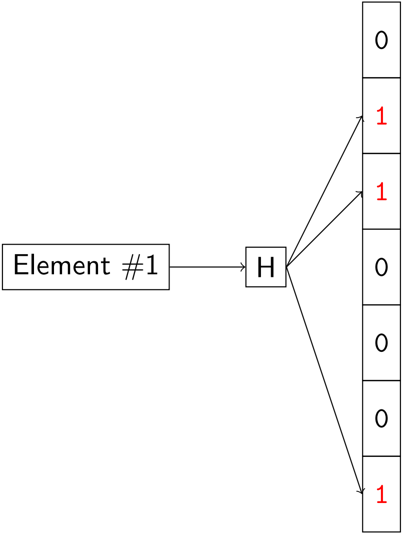
Element #1

H

0
0
0
0
0
0
0

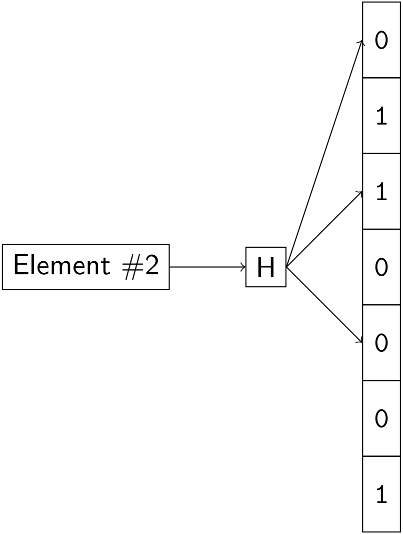
$H(\text{Element }\#1) = (2, 3, 7)$

BF: Insert



$H(\text{Element }\#1) = (2, 3, 7)$

BF: Insert



Element #2

H

0

1

1

0

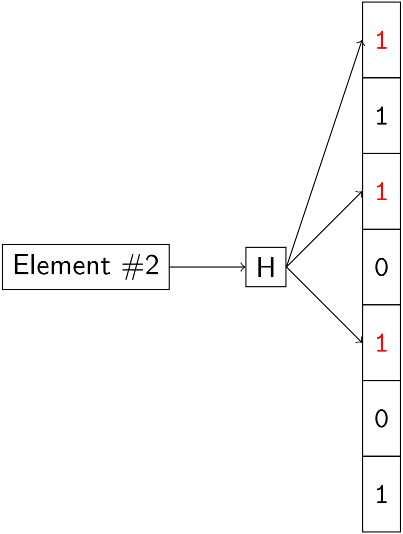
0

0

1

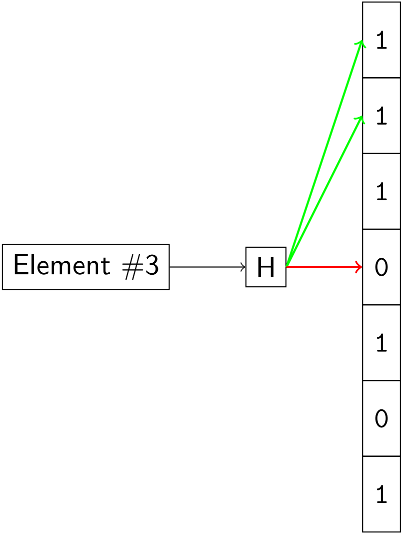
$H(\text{Element } \#1) = (2, 3, 7)$
$H(\text{Element } \#2) = (1, 3, 5)$

BF: Insert



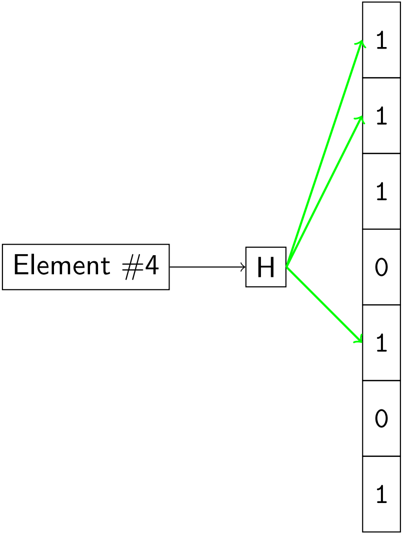
$H(\text{Element } \#1) = (2, 3, 7)$
$H(\text{Element } \#2) = (1, 3, 5)$

BF: Membership Test



$H(\text{Element } \#1) = (2, 3, 7)$
$H(\text{Element } \#2) = (1, 3, 5)$

BF: Membership Test (false positive)



Element #4

H

1

1

1

0

1

0

1

$H(\text{Element } \#1) = (2, 3, 7)$
$H(\text{Element } \#2) = (1, 3, 5)$

Counting Bloom Filters

BF where buckets hold a **positive integer**.

Additional Operation:

Remove(d, e) Remove element from the CBF d.

⇒ False negatives only when removing a non-existing element.

Invertible Bloom Filters

Similar to CBF, but

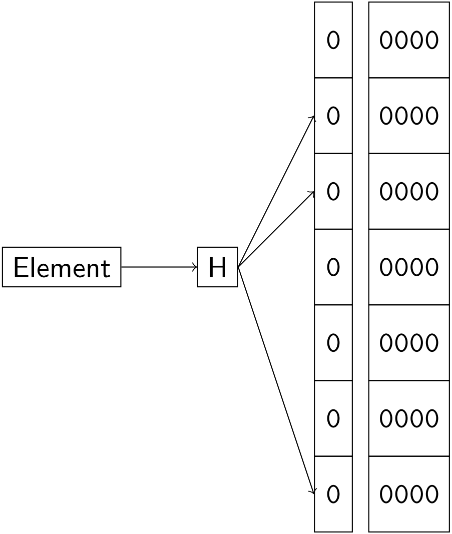
- Allow **negative counts**
- Additionaly store **(XOR-)sum of hashes** in buckets.

Additional Operations:

$(e, r) = \mathit{Extract}(d)$ Extract an element (e) from the IBF d, with result code $r \in \{\mathit{left}, \mathit{right}, \mathit{done}, \mathit{fail}\}$

$d' = \mathit{SymDiff}(d_1, d_2)$ Create an IBF that represents the symmetric difference of d_1 and d_2.

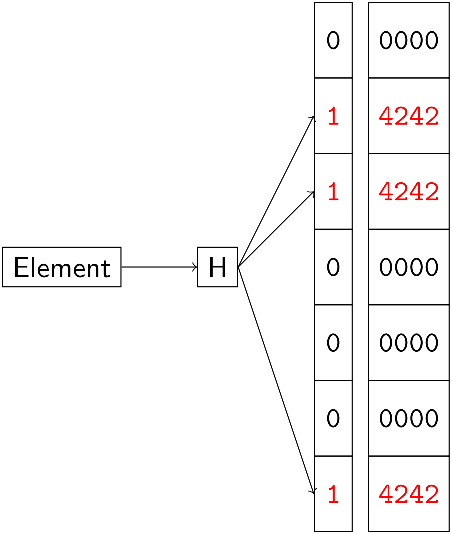
IBF: Insert



0	0000
0	0000
0	0000
0	0000
0	0000
0	0000
0	0000

$H(\text{Element } \#1) = (2, 3, 7)$
$H'(\text{Element } \#1) = 4242$

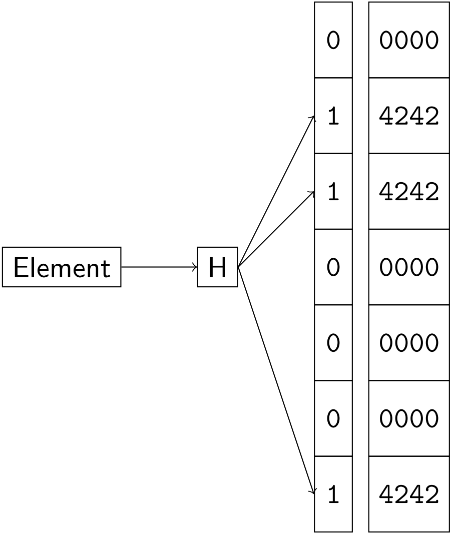
IBF: Insert



0	0000
1	4242
1	4242
0	0000
0	0000
0	0000
1	4242

$H(\text{Element }\#1) = (2, 3, 7)$
$H'(\text{Element }\#1) = 4242$

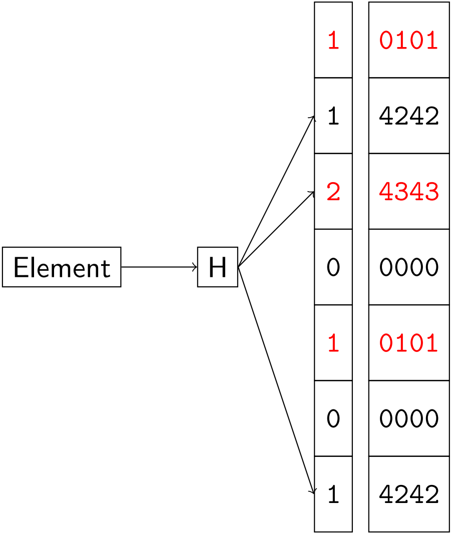
IBF: Insert



0	0000
1	4242
1	4242
0	0000
0	0000
0	0000
1	4242

$H(\text{Element } \#1) = (2, 3, 7)$
$H'(\text{Element } \#1) = 4242$
$H(\text{Element } \#2) = (1, 3, 5)$
$H'(\text{Element } \#2) = 0101$

IBF: Insert



1	0101
1	4242
2	4343
0	0000
1	0101
0	0000
1	4242

$H(\text{Element } \#1) = (2, 3, 7)$

$H'(\text{Element } \#1) = 4242$

$H(\text{Element } \#2) = (1, 3, 5)$

$H'(\text{Element } \#2) = 0101$

IBF: Extract

1	0101	pure bucket
1	4242	
2	4343	
0	0000	
1	0101	
0	0000	
1	4242	

- Pure bucket ⇒ extractable element hash
- Extraction ⇒ more pure buckets (hopefully/probably)
- Less elements ⇒ more chance for pure buckets

Symmetric Difference

We can directly compute the symmetric difference without extraction.

- Subtract counts
- XOR hashes

The Set Union Protocol

1. Create IBFs
2. Compute SymDiff
3. Extract element hashes

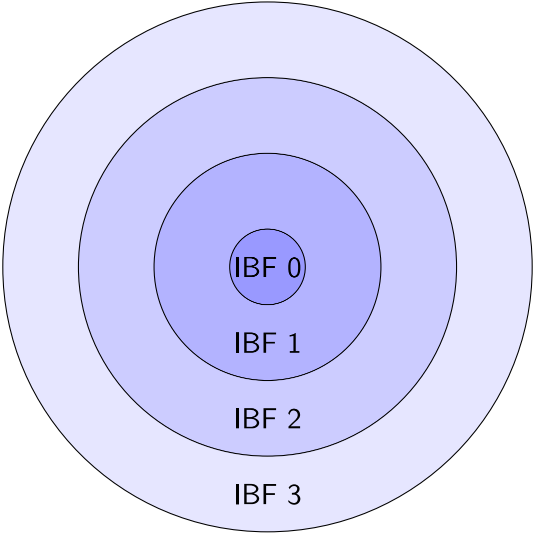
- Amount of communication and computation only depends on δ, not $|A| + |B|$:)
- How do we choose the initial size of the IBF?
- $\Rightarrow$ Do difference estimation first!

Difference Estimation

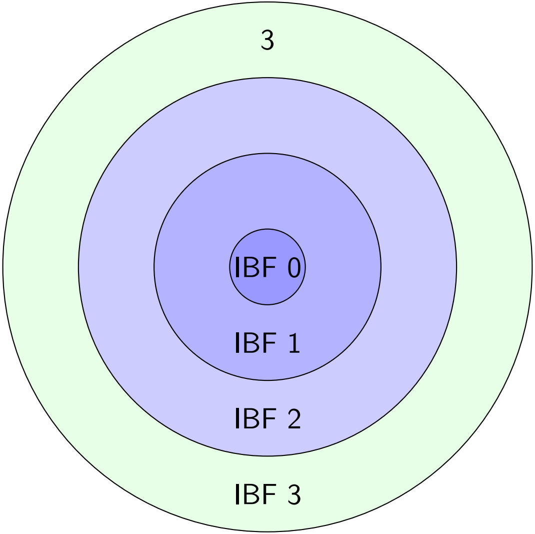
- We need an estimator that is accurate for small differences
- Idea: re-use IBFs for difference estimation:

1. Alice and Bob create fixed number of constant-size IBFs by sampling their set. The collection of IBFs is called a Strata Estimator (SE).
    - Stratum 0 contains 1/2 of all elements
    - Stratum 1 contains 1/4 of all elements
    - Stratum n contains $1/(2^n)$ all elements
2. Alice receives Bob's strata estimator
3. Alice computes $SE_{diff} = SymDiff(SE_{\text{Alice}}, SE_{\text{Bob}})$
    - by pair-wise *SymDiff* of all IBFs in the SE
4. Alice estimates the size of SE_{diff}.

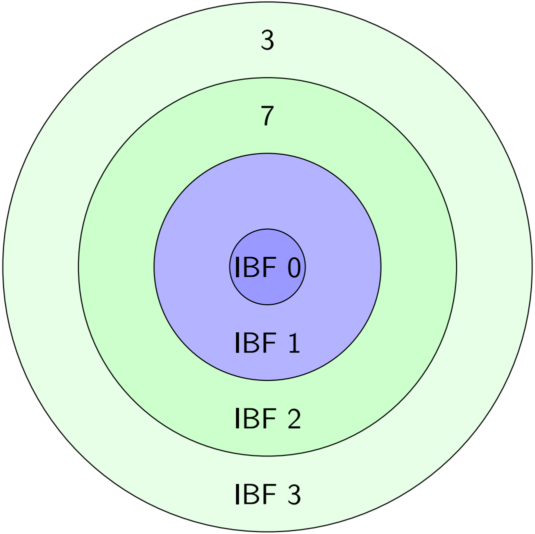
Strata Estimator



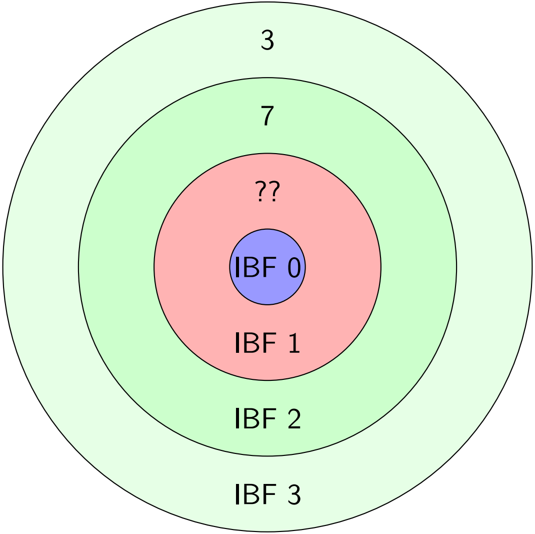
Strata Estimator



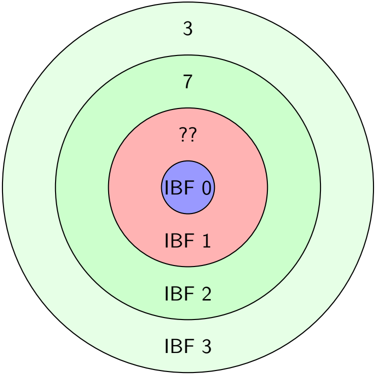
Strata Estimator



Strata Estimator



Estimation



Estimate as $(3+7)\cdot 2^1$.
(Number of extracted hashes scaled by 2^{r-1} for r failed rounds of strata decoding.)

The Complete Protocol

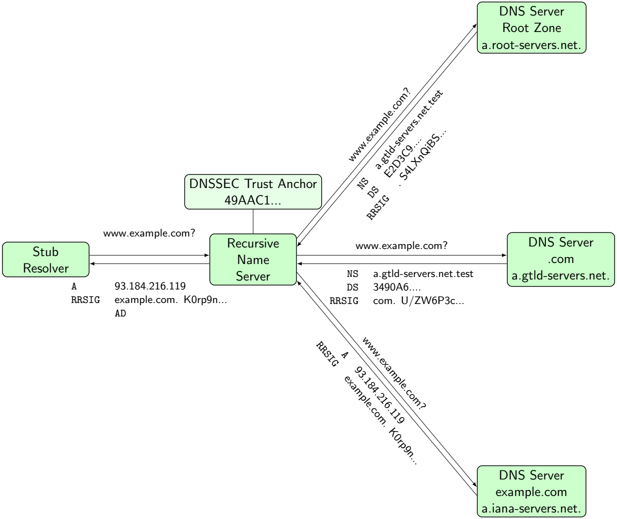
1. Alice sends SE_{Alice} to Bob
2. Bob estimates the set difference δ
3. Bob computes IBF_{Bob} with size δ and sends it to Alice
4. Alice computes IBF_{Alice}
5. Alice computes $IBF_{diff} = SymDiff(IBF_{\text{Alice}}, IBF_{\text{Bob}})$
6. Alice extracts element hashes from IBF_{diff}.
   - $b = left \Rightarrow$ Send element to to Bob
   - $b = right \Rightarrow$ Send element request to to Bob
   - $b = fail \Rightarrow$ Send larger IBF (double the size) to Bob, go to (3.) with switched roles
   - $b = done \Rightarrow$ We're done ...

Break

Security Goals for Name Systems

- Query origin anonymity
- Data origin authentication and integrity protection
- Zone confidentiality
- Query and response privacy
- Censorship resistance
- Traffic amplification resistance
- Availability

Reminder: DNSSEC



DNS Server
Root Zone
a.root-servers.net.

DNSSEC Trust Anchor
49AAC1...

Stub
Resolver

Recursive
Name
Server

DNS Server
.com
a.gtld-servers.net.

DNS Server
example.com
a.iana-servers.net.

www.example.com?

```
A       93.184.216.119
RRSIG   example.com. K0rp9n...
        AD
```

www.example.com?

```
NS      a.gtld-servers.net.test
DS      E2D3C9....
RRSIG   . S4LXnQiBS...
```

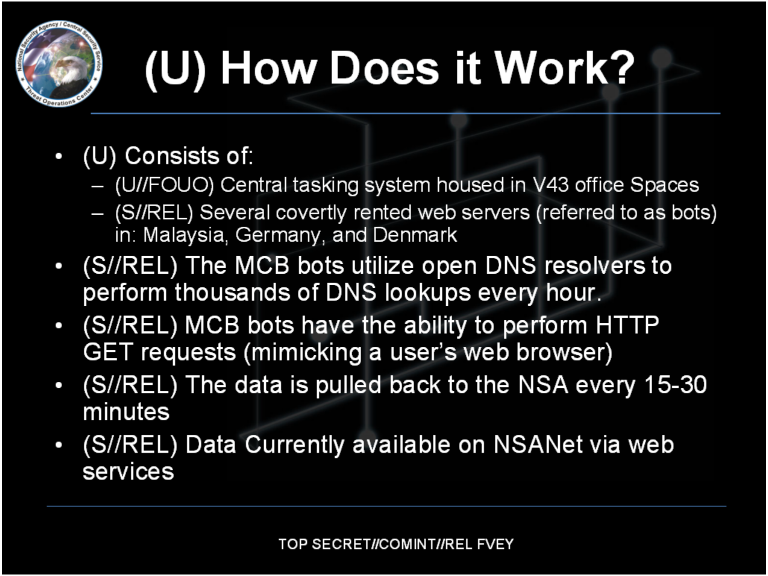
www.example.com?

```
NS      a.gtld-servers.net.test
DS      3490A6....
RRSIG   com. U/ZW6P3c...
```

www.example.com?

```
A       93.184.216.119
RRSIG   example.com. K0rp9n...
```

Exemplary Attacks: MORECOWBELL

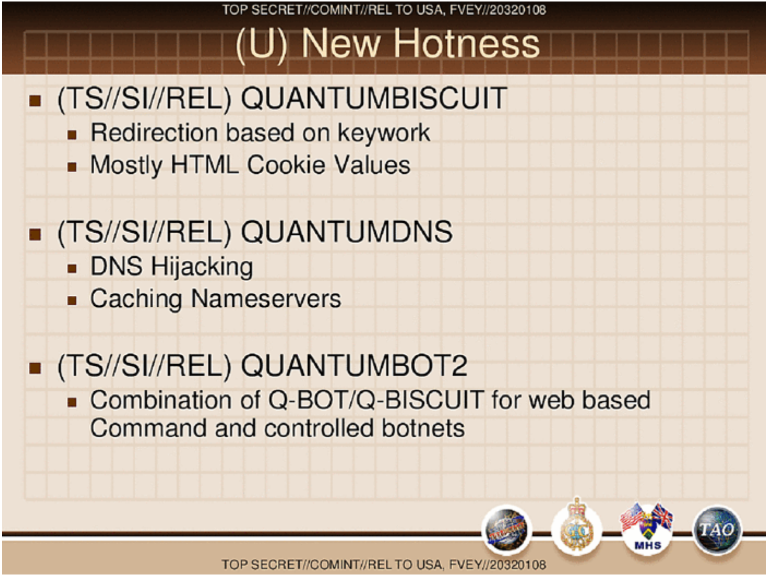


(U) How Does it Work?

- (U) Consists of:
  - (U//FOUO) Central tasking system housed in V43 office Spaces
  - (S//REL) Several covertly rented web servers (referred to as bots) in: Malaysia, Germany, and Denmark
- (S//REL) The MCB bots utilize open DNS resolvers to perform thousands of DNS lookups every hour.
- (S//REL) MCB bots have the ability to perform HTTP GET requests (mimicking a user's web browser)
- (S//REL) The data is pulled back to the NSA every 15-30 minutes
- (S//REL) Data Currently available on NSANet via web services

TOP SECRET//COMINT//REL FVEY

Exemplary Attacks: QUANTUMDNS



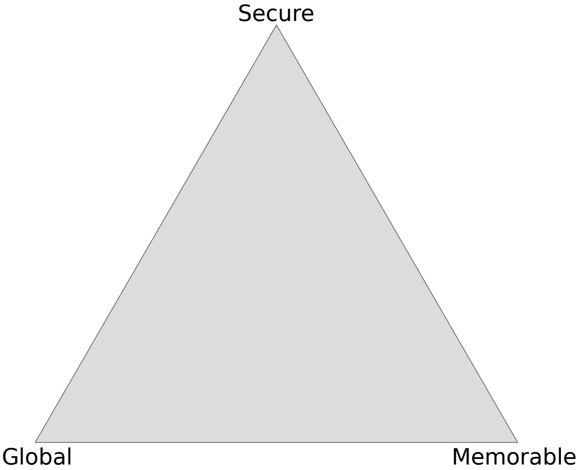
TOP SECRET//COMINT//REL TO USA, FVEY//20320108

(U) New Hotness

- (TS//SI//REL) QUANTUMBISCUIT
  - Redirection based on keywork
  - Mostly HTML Cookie Values
- (TS//SI//REL) QUANTUMDNS
  - DNS Hijacking
  - Caching Nameservers
- (TS//SI//REL) QUANTUMBOT2
  - Combination of Q-BOT/Q-BISCUIT for web based Command and controlled botnets

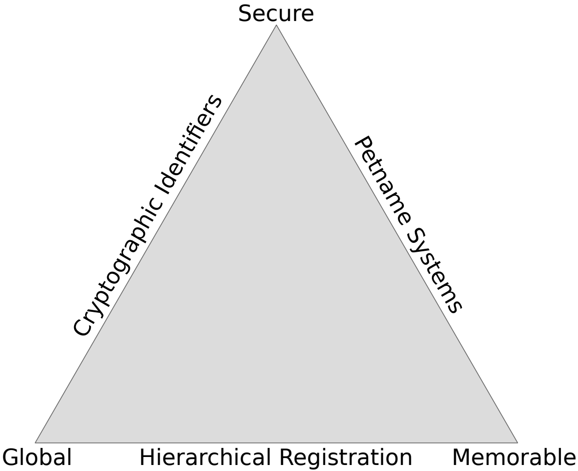
TOP SECRET//COMINT//REL TO USA, FVEY//20320108

Zooko's Triangle



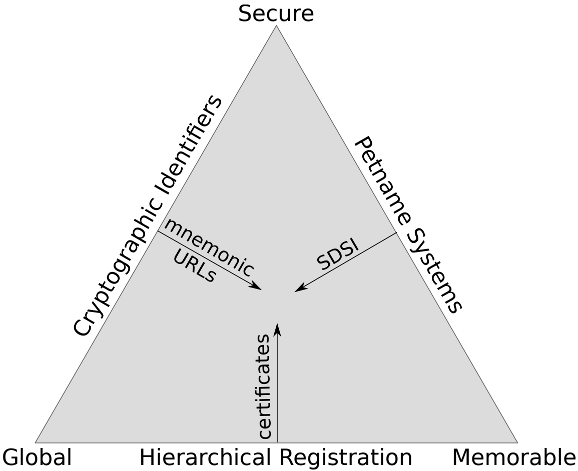
A name system can only fulfill **two**!

Zooko's Triangle

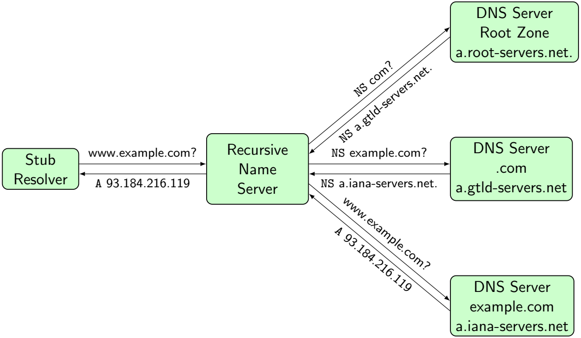


DNS, ".onion" IDs and `/etc/hosts/` are representative designs.

Zooko's Triangle



Query Name Minimization



Stub Resolver

www.example.com?

A 93.184.216.119

Recursive Name Server

NS com?

NS a.gtld-servers.net.

DNS Server Root Zone a.root-servers.net.

NS example.com?

NS a.iana-servers.net.

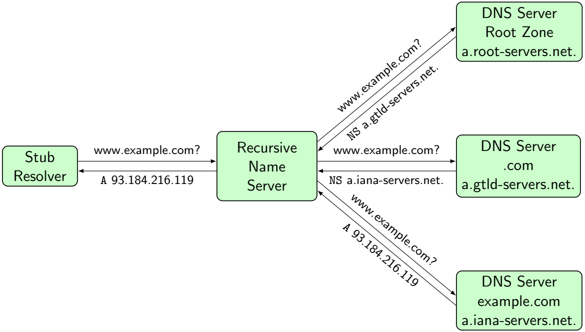
DNS Server .com a.gtld-servers.net

www.example.com?

A 93.184.216.119

DNS Server example.com a.iana-servers.net

DNS over TLS


Stub Resolver
www.example.com?
A 93.184.216.119
Recursive Name Server
www.example.com?
NS a.gtld-servers.net.
DNS Server Root Zone a.root-servers.net.
www.example.com?
NS a.iana-servers.net.
DNS Server .com a.gtld-servers.net.
www.example.com?
A 93.184.216.119
DNS Server example.com a.iana-servers.net.

The Textbook Version of the Internet

Layering, ≈ 1990

	HTTPS
DNS	TLS
UDP	TCP
IPv4	
Ethernet	
Phys. Layer	

The Textbook Version of the Internet

Layering, ≈ 1990

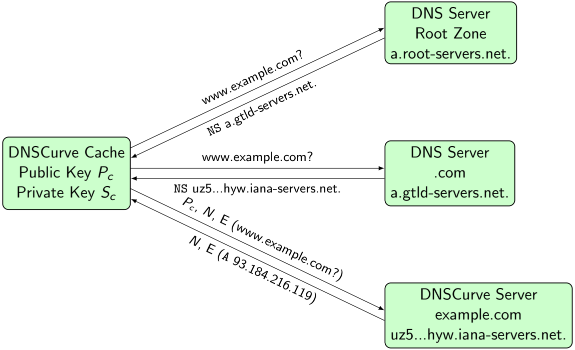
	HTTPS
DNS	TLS
UDP	TCP
IPv4	
Ethernet	
Phys. Layer	

“Layering”, ≈ 2020

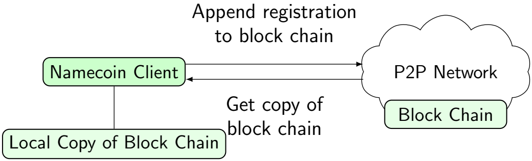
HTTPS	libmicrohttpd
TLS-with-DANE	libgnutls
DNS-over-TLS	libunbound
TLS*	libnss
TCP	Linux
IPv6	Linux
Ethernet	
Phys. Layer	

* = castrated version without RFC 6125 or RFC 6394, possibly NULL cipher, see TLS profiles draft.

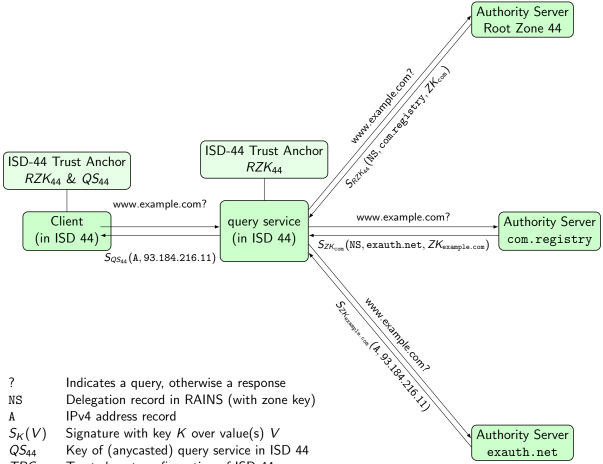
DNSCurve


DNS Server
Root Zone
a.root-servers.net.
www.example.com?
NS a.gtld-servers.net.
DNSCurve Cache
Public Key P_c
Private Key S_c
www.example.com?
DNS Server
.com
a.gtld-servers.net.
NS uz5...hyw.iana-servers.net.
P_c, N, E (www.example.com?)
N, E (A 93.184.216.119)
DNSCurve Server
example.com
uz5...hyw.iana-servers.net.

Namecoin

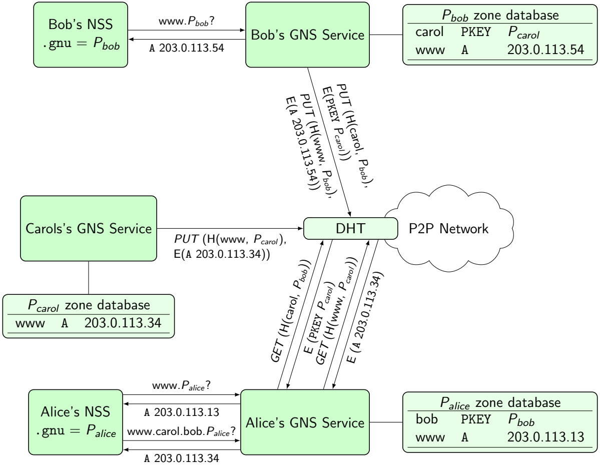

Append registration
to block chain
Namecoin Client
P2P Network
Block Chain
Get copy of
block chain
Local Copy of Block Chain

RAINS



?	Indicates a query, otherwise a response
`NS`	Delegation record in RAINS (with zone key)
`A`	IPv4 address record
$S_K(V)$	Signature with key K over value(s) V
QS_{44}	Key of (anycasted) query service in ISD 44
TRC_{44}	Trusted root configuration of ISD 44
RZK_{44}	Root zone key of ISD 44
ZK_{name}	Zone key of authority for “name”

The GNU Name System (GNS)



Bob's NSS

.gnu = P_{bob}

www.P_{bob}?

A 203.0.113.54

Bob's GNS Service

P_{bob} zone database

carol	PKEY	P_{carol}
www	A	203.0.113.54

PUT (H(carol, P_{bob}), E(PKEY P_{carol}))

PUT (H(www, P_{bob}), E(A 203.0.113.54))

Carols's GNS Service

PUT (H(www, P_{carol}), E(A 203.0.113.34))

DHT

P2P Network

P_{carol} zone database

www	A	203.0.113.34

GET (H(carol, P_{bob}))

E (PKEY P_{carol})

GET (H(www, P_{carol}))

E (A 203.0.113.34)

Alice's NSS

.gnu = P_{alice}

www.P_{alice}?

A 203.0.113.13

www.carol.bob.P_{alice}?

A 203.0.113.34

Alice's GNS Service

P_{alice} zone database

bob	PKEY	P_{bob}
www	A	203.0.113.13

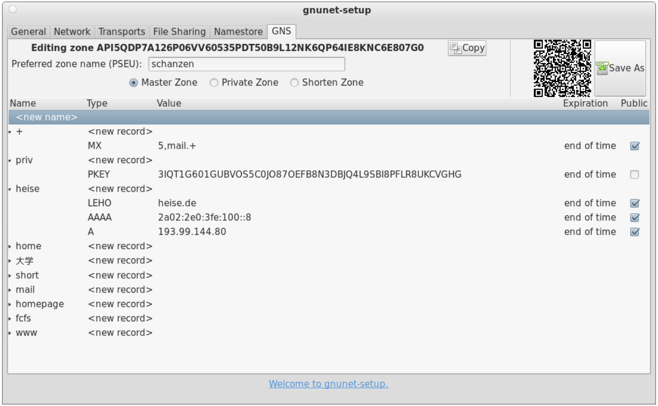
The GNU Name System[1]

Properties of GNS

- Decentralized name system with secure memorable names
- Delegation used to achieve transitivity
- Also supports globally unique, secure identifiers
- Achieves query and response privacy
- Provides alternative public key infrastructure
- Interoperable with DNS

[1] Joint work with Martin Schanzenbach and Matthias Wachs

Zone Management: like in DNS



gnunet-setup

Editing zone API5QDP7A126P06VV60535PDT50B9L12NK6QP64IE8KNC6E807G0

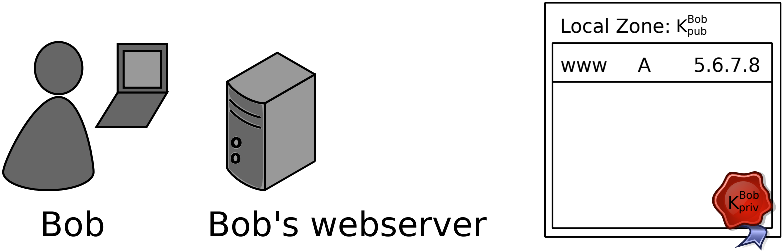
Preferred zone name (PSEU): schanzen

● Master Zone ○ Private Zone ○ Shorten Zone

Name	Type	Value	Expiration	Public
<new name>				
+	<new record>			
	MX	5,mail.+	end of time	☑
priv	<new record>			
	PKEY	3IQT1G601GUBVOS5C0JO87OEFB8N3DBJQ4L9SBI8PFLR8UKCVGHG	end of time	□
heise	<new record>			
	LEHO	heise.de	end of time	☑
	AAAA	2a02:2e0:3fe:100::8	end of time	☑
	A	193.99.144.80	end of time	☑
home	<new record>			
大学	<new record>			
short	<new record>			
mail	<new record>			
homepage	<new record>			
fcfs	<new record>			
www	<new record>			

Welcome to gnunet-setup.

Name resolution in GNS

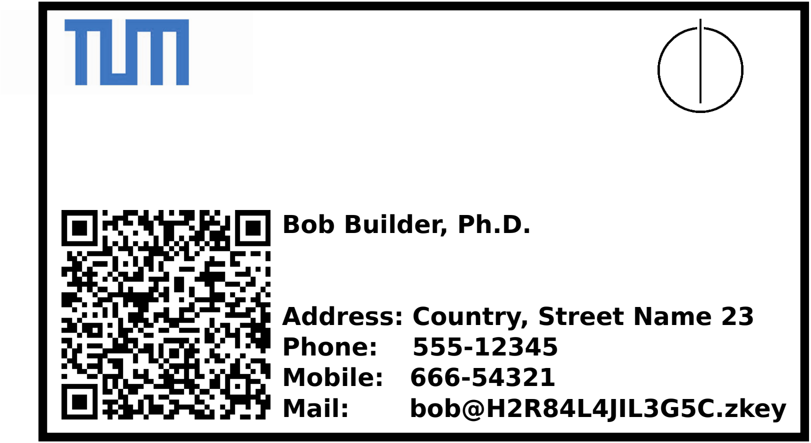


Local Zone: K_{pub}^{Bob}

www	A	5.6.7.8

- Bob can locally reach his webserver via **www.gnu**

Secure introduction

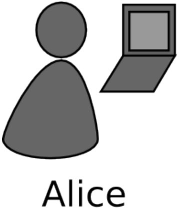


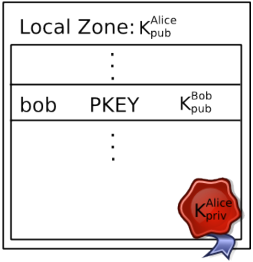
Bob Builder, Ph.D.

Address: Country, Street Name 23
Phone: 555-12345
Mobile: 666-54321
Mail: bob@H2R84L4JIL3G5C.zkey

- Bob gives his public key to his **friends**, possibly via QR code

Delegation




Local Zone: K_{pub}^{Alice}

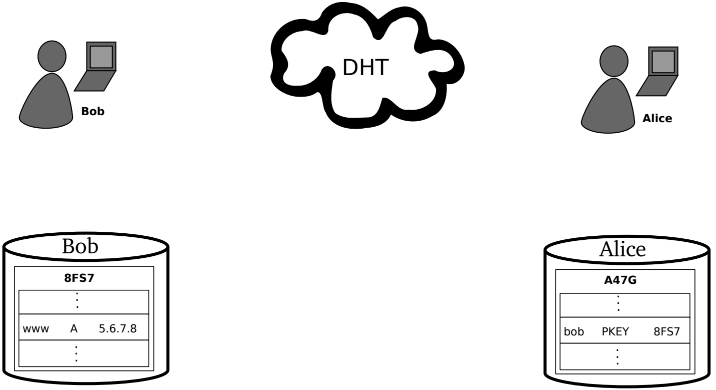
⋮

bob PKEY K_{pub}^{Bob}

⋮

- Alice learns Bob's public key
- Alice creates delegation to zone K_{pub}^{Bob} under label **bob**
- Alice can reach Bob's webserver via **www.bob.gnu**

Name Resolution



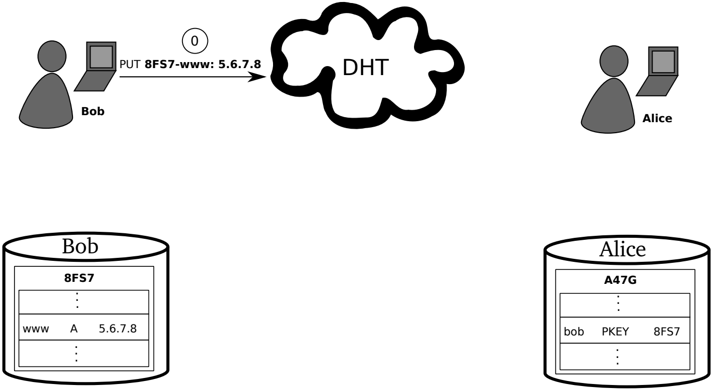
8FS7

	⋮	
www	A	5.6.7.8
	⋮	

A47G

	⋮	
bob	PKEY	8FS7
	⋮	

Name Resolution



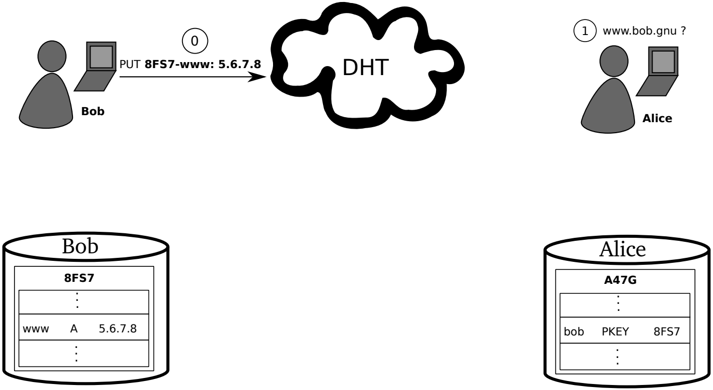
8FS7

	⋮	
www	A	5.6.7.8
	⋮	

A47G

	⋮	
bob	PKEY	8FS7
	⋮	

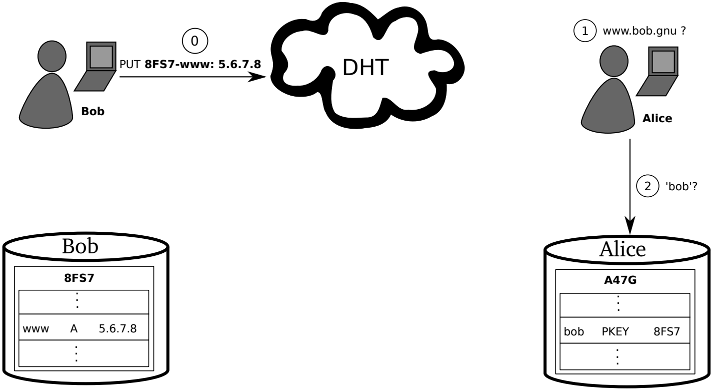
Name Resolution



8FS7		
	⋮	
www	A	5.6.7.8
	⋮	

A47G		
	⋮	
bob	PKEY	8FS7
	⋮	

Name Resolution



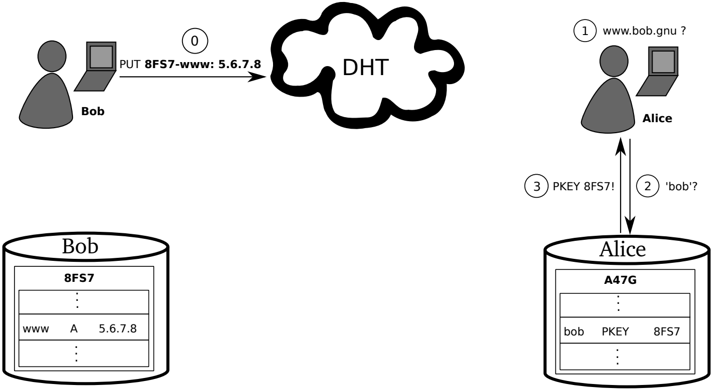
8FS7

⋮		
www	A	5.6.7.8
⋮		

A47G

⋮		
bob	PKEY	8FS7
⋮		

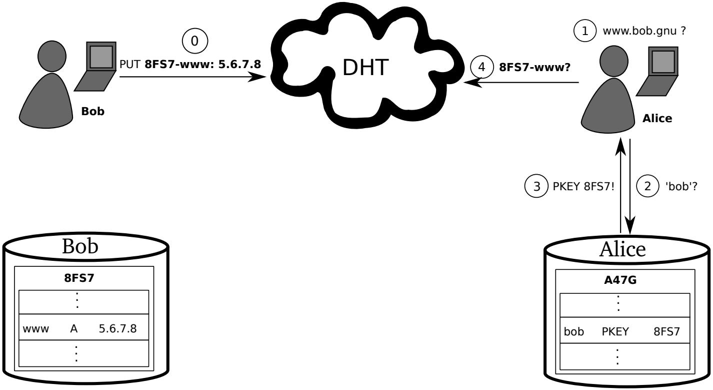
Name Resolution



	⋮	
www	A	5.6.7.8
	⋮	

	⋮	
bob	PKEY	8FS7
	⋮	

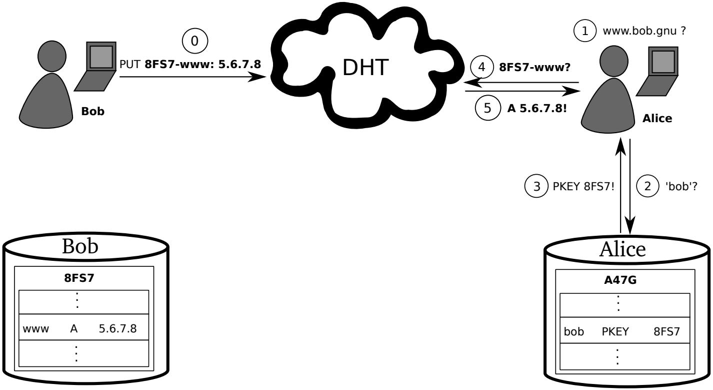
Name Resolution



8FS7		
	⋮	
www	A	5.6.7.8
	⋮	

A47G		
	⋮	
bob	PKEY	8FS7
	⋮	

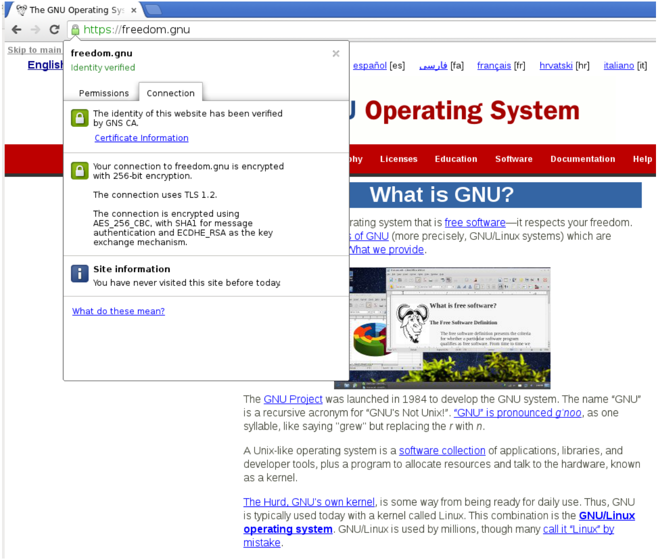
Name Resolution



8FS7		
⋮		
www	A	5.6.7.8
⋮		

A47G		
⋮		
bob	PKEY	8FS7
⋮		

GNS as PKI (via DANE/TLSA)



freedom.gnu
Identity verified

The identity of this website has been verified by GNS CA.

Certificate Information

Your connection to freedom.gnu is encrypted with 256-bit encryption.

The connection uses TLS 1.2.

The connection is encrypted using AES_256_CBC, with SHA1 for message authentication and ECDHE_RSA as the key exchange mechanism.

Site information

You have never visited this site before today.

What do these mean?

U Operating System

What is GNU?

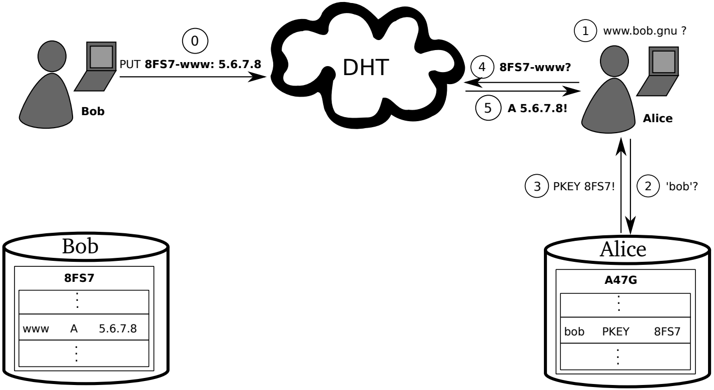
rating system that is free software—it respects your freedom. s of GNU (more precisely, GNU/Linux systems) which are What we provide.

The GNU Project was launched in 1984 to develop the GNU system. The name “GNU” is a recursive acronym for “GNU's Not Unix!”. “GNU” is pronounced *g'noo*, as one syllable, like saying "grew" but replacing the *r* with *n*.

A Unix-like operating system is a software collection of applications, libraries, and developer tools, plus a program to allocate resources and talk to the hardware, known as a kernel.

The Hurd, GNU's own kernel, is some way from being ready for daily use. Thus, GNU is typically used today with a kernel called Linux. This combination is the **GNU/Linux operating system**. GNU/Linux is used by millions, though many call it “Linux” by mistake.

Privacy Issue: DHT



0 PUT **8FS7-www: 5.6.7.8**

DHT

1 www.bob.gnu ?

4 **8FS7-www?**

5 **A 5.6.7.8!**

Bob

Alice

3 PKEY 8FS7!

2 'bob'?

Bob

8FS7		
⋮		
www	A	5.6.7.8
⋮		

Alice

A47G		
⋮		
bob	PKEY	8FS7
⋮		

Query Privacy: Terminology

G generator in ECC curve, a point

o size of ECC group, $o := |G|$, o prime

x private ECC key of zone ($x \in \mathbb{Z}_o$)

P public key of zone, a point $P := xG$

l label for record in a zone ($l \in \mathbb{Z}_o$)

$R_{P,l}$ set of records for label l in zone P

$q_{P,l}$ query hash (hash code for DHT lookup)

$B_{P,l}$ block with encrypted information for label l in zone P published in the DHT under $q_{P,l}$

Query Privacy: Cryptography

Publishing records $R_{P,l}$ as $B_{P,l}$ under key $q_{P,l}$

$$h := H(l, P) \tag{1}$$

$$d := h \cdot x \mod o \tag{2}$$

$$B_{P,l} := S_d(E_{HKDF(l,P)}(R_{P,l})), dG \tag{3}$$

$$q_{P,l} := H(dG) \tag{4}$$

Query Privacy: Cryptography

Publishing records $R_{P,l}$ as $B_{P,l}$ under key $q_{P,l}$

$$\begin{aligned} h &:= H(l, P) && (1)\\ d &:= h \cdot x \mod o && (2)\\ B_{P,l} &:= S_d(E_{HKDF(l,P)}(R_{P,l})), dG && (3)\\ q_{P,l} &:= H(dG) && (4) \end{aligned}$$

Searching for records under label l in zone P

$$\begin{aligned} h &:= H(l, P) && (5)\\ q_{P,l} &:= H(hP) = H(hxG) = H(dG) \Rightarrow \texttt{obtain } B_{P,l} && (6)\\ R_{P,l} &= D_{HKDF(l,P)}(B_{P,l}) && (7) \end{aligned}$$

The “.zkey” Zone

- “.zkey” is another pTLD, in addition to “.gnu”
- In “LABEL.zkey”, the “LABEL” is a public key of a zone
- “alice.bob.*KEY*.zkey” is perfectly legal

⇒ Globally unique identifiers

Key Revocation

- Revocation message signed with private key (ECDSA)
- Flooded on all links in P2P overlay, stored forever
- Efficient set reconciliation used when peers connect
- Expensive proof-of-work used to limit DoS-potential
- Proof-of-work can be calculated ahead of time
- Revocation messages can be stored off-line if desired

Shadow Records

- Records change
- Expiration time controls validity, like in DNS
- DHT propagation has higher delays, compared to DNS

Shadow Records

- Records change
- Expiration time controls validity, like in DNS
- DHT propagation has higher delays, compared to DNS
- SHADOW is a flag in a record
- Shadow records are only valid if no other, non-expired record of the same type exists

NICKnames

- “alice.bob.carol.dave.gnu” is a bit long for Edward (“.gnu”)
- Also, we need to trust Bob, Carol and Dave (for each lookup)
- Finally, Alice would have liked to be called Krista (just Bob calls her Alice)

NICKnames

- "alice.bob.carol.dave.gnu" is a bit long for Edward (".gnu")
- Also, we need to trust Bob, Carol and Dave (for each lookup)
- Finally, Alice would have liked to be called Krista (just Bob calls her Alice)
- "NICK" records allow Krista to specify her preferred NICKname
- GNS adds a "NICK" record to each record set automatically
- Edward learns the "NICK", and software could automatically create "krista.gnu"

NICKnames

- "alice.bob.carol.dave.gnu" is a bit long for Edward (".gnu")
- Also, we need to trust Bob, Carol and Dave (for each lookup)
- Finally, Alice would have liked to be called Krista (just Bob calls her Alice)
- "NICK" records allow Krista to specify her preferred NICKname
- GNS adds a "NICK" record to each record set automatically
- Edward learns the "NICK", and software could automatically create "krista.gnu"
- Memorable, short trust path in the future! TOFU!
- Krista better pick a reasonably unique NICK.

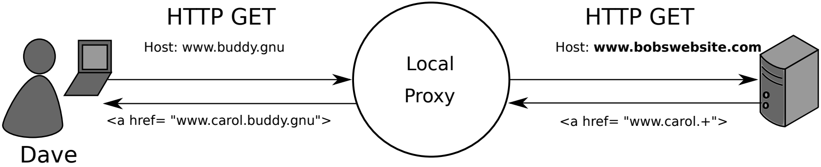
Relative Names

- GNS records can contain “.+”
- CNAME: “server1.+”
- MX: “mail.+”
- “.+” stands for “relative to current zone”

Supporting this for links in browsers would be nice, too.

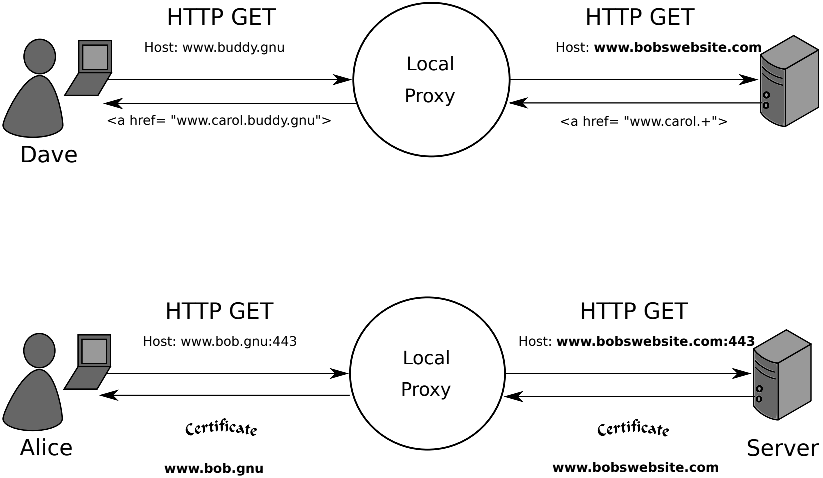
Legacy Hostname (LEHO) Records

LEHO records give a hint about the DNS name the server expects.



Legacy Hostname (LEHO) Records

LEHO records give a hint about the DNS name the server expects.



DNS Delegation

- Delegate to DNS using GNS2DNS records
- GNS2DNS record specifies:
  - Name of DNS resolver (i.e. "ns1.example.com" or "piratedns.+")
  - DNS domain to continue resolution in (i.e. "example.com" or "piratebay.org")
- GNS will first resolve DNS resolver name to A/AAAA record
- GNS will then resolve "*left.of.gns2dns.*example.com" using DNS

Fun GNS Record Types

- DNS CERT: store your GPG public key
- GNUNET VPN: TCP/IP services hosted in GNUnet
- GNUNET PHONE: have a conversation

Application Integration

- SOCKS proxy (`gnunet-gns-proxy`)
- NSS plugin
- GNS (C) API
- GNS (IPC) protocol
- GNS command-line tool

Summary

- Interoperable with DNS
- Globally unique identifiers with “.zkey”
- Delegation allows using zones of other users
- Trust paths explicit, trust agility
- Simplified key exchange compared to Web-of-Trust
- Privacy-enhanced queries, censorship-resistant
- Reliable revocation

Privacy summary

Method	Defense against MiTM	Zone privacy	Privacy vs. network	Privacy vs. operator	Traffic amplification resistance	Censorship resistance	Ease of migration
DNS	✗	✓	✗	✗	✗	✗	✓
DNSSEC	✓	✗	✗	✗	✗	✗	✗*
DNSCurve	✓	✓	✓	✗	✓	✗	✗
DNS-over-TLS	✓	n/a	✓	✗	✓	✗	✗
Namecoin	✓	✗	✓	✓	✓	✓	✗
RAINS	✓	✗	✓	✗	✓	✗	✗
GNS	✓	✓	✓	✓	✓	✓	✗

*EDNS0

Key management summary

	Suitable for personal use	Memorable	Decentralised	Modern cryptography	Understandable	Exposes metadata	Transitive
DNS	✗	✓	✗	✗	✗	✗	✓
DNSSEC	✗	✓	✗	✗	✗	✗	✓
DNSCurve	✗	✓	✗	✓	✗	✗	✓
DNS-over-TLS	✗	✓	✗	✗	✗	✗	✓
TLS-X.509	✗	✓	✗	✗	✗	✗	✓
Web of Trust	✓	✗	✓	✗	✗	✗	✓
TOFU	✓	✗	✓		✓	✓	✗
Namecoin	✗	✓	✗	✓	✓	✗	✓
RAINS	✗	✓	✗	✓	✓	✗	✓
GNS	✓	✓	✓	✓	✓	✓	✓

Ongoing and Future Work (Project 2, BS theses)

- Optimze GNUnet DHT
- Import ".fr" TLD into GNS and hijack it!
- Implement & evaluate bounded Eppstein set reconciliation
- Integrate GNS with Tor

Conclusion

- Query name minimization is low-cost, low-benefit approach, but should clearly be done
- Simple encryption schemes offer medium-cost, medium-benefit approach
- GNU Name System performance depends on the DHT
  ⇒ need to invest more in DHT design & implementation

DNS	globalist
DNSSEC	authoritarian
Namecoin	libertarian (US)
RAINS	nationalist
GNS	anarchist

In which world do you want to live?

Do you have any questions?

References:

- Nathan Evans and Christian Grothoff. *R5N. Randomized Recursive Routing for Restricted-Route Networks*. **5th International Conference on Network and System Security**, 2011.
- Matthias Wachs, Martin Schanzenbach and Christian Grothoff. *On the Feasibility of a Censorship Resistant Decentralized Name System*. **6th International Symposium on Foundations & Practice of Security**, 2013.
- M. Schanzenbach *Design and Implementation of a Censorship Resistant and Fully Decentralized Name System*. **Master's Thesis (TUM)**, 2012.