The GNU Name System: A Public Key Infrastructure for Social Movements in the Age of Universal Surveillance

Christian Grothoff

The GNUnet Project

28.04.2017

"Never doubt your ability to change the world." –Glenn Greenwald

The Internet

Virtually all Internet protocols are broken:

Ethernet	MAC spoofing, cleartext
IP	IP spoofing, cleartext
BGP	AS hijacking, cleartext
DNS	cache poisoning, cleartext
DNSSEC	cleartext, often no end-to-end authentication
TLS	100 CAs can certify anybody for anything
HTTP	too chatty, complex, slow
...	

The Internet

Virtually all Internet protocols are broken:

Ethernet	MAC spoofing, cleartext
IP	IP spoofing, cleartext
BGP	AS hijacking, cleartext
DNS	cache poisoning, cleartext
DNSSEC	cleartext, often no end-to-end authentication
TLS	100 CAs can certify anybody for anything
HTTP	too chatty, complex, slow
...	

Rule 1 for the GNUnet: Encrypt everything.

Encryption to the Rescue?

- Existing Internet PKIs are easily controlled:
  - DNSSEC root certificate
  - X.509 CAs (HTTPS certificates)
  - Major browser vendors (CA root stores!)

Encryption to the Rescue?

- Existing Internet PKIs are easily controlled:
  - DNSSEC root certificate
  - X.509 CAs (HTTPS certificates)
  - Major browser vendors (CA root stores!)
- Encryption does not help if PKI is compromised!

Encryption to the Rescue?

- Existing Internet PKIs are easily controlled:
  - DNSSEC root certificate
  - X.509 CAs (HTTPS certificates)
  - Major browser vendors (CA root stores!)
- Encryption does not help if PKI is compromised!
- PGP Web-of-Trust leaks social graph

How bad is it?

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`
- NS of `de.net` is `ns1.denic.de`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`
- NS of `de.net` is `ns1.denic.de`
- NS of **tum.de** is `dns1.lrz.de`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`
- NS of `de.net` is `ns1.denic.de`
- NS of **tum.de** is `dns1.lrz.de`
- NS of `lrz.de` is `dns1.lrz.de`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`
- NS of `de.net` is `ns1.denic.de`
- NS of **tum.de** is `dns1.lrz.de`
- NS of `lrz.de` is `dns1.lrz.de`
- NS of **in.tum.de** is `tuminfo1.informatik.tu-muenchen.de`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`
- NS of `de.net` is `ns1.denic.de`
- NS of **tum.de** is `dns1.lrz.de`
- NS of `lrz.de` is `dns1.lrz.de`
- NS of **in.tum.de** is `tuminfo1.informatik.tu-muenchen.de`
- NS of `tu-muenchen.de` is `ws-han1.wip-ip.dfn.de`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`
- NS of `de.net` is `ns1.denic.de`
- NS of **tum.de** is `dns1.lrz.de`
- NS of `lrz.de` is `dns1.lrz.de`
- NS of **in.tum.de** is `tuminfo1.informatik.tu-muenchen.de`
- NS of `tu-muenchen.de` is `ws-han1.wip-ip.dfn.de`
- NS of `dfn.de` is `ws-han1.wip-ip.dfn.de`

A DNS Lookup in 2014...

What would a simple DNS lookup do? Say for `taler.net`?

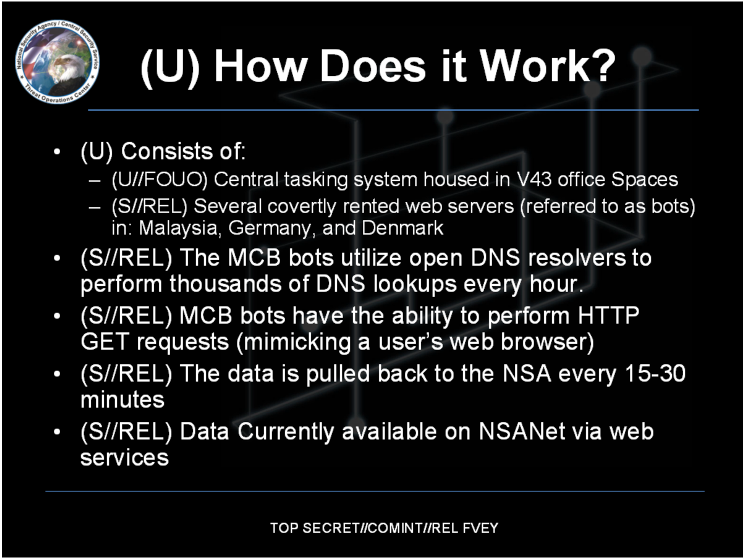
- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`
- NS of `de.net` is `ns1.denic.de`
- NS of **tum.de** is `dns1.lrz.de`
- NS of `lrz.de` is `dns1.lrz.de`
- NS of **in.tum.de** is `tuminfo1.informatik.tu-muenchen.de`
- NS of `tu-muenchen.de` is `ws-han1.wip-ip.dfn.de`
- NS of `dfn.de` is `ws-han1.wip-ip.dfn.de`
- NS of **net.in.tum.de** is `dns1.lrz.de`

A DNS Lookup in 2014...

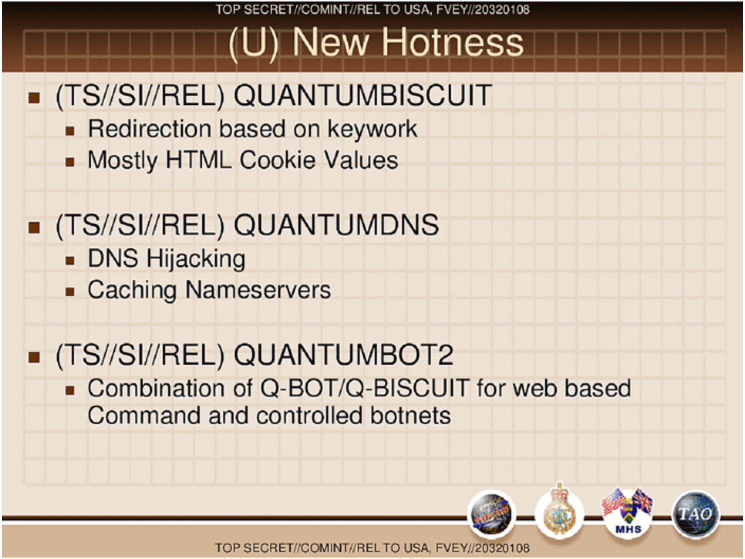
What would a simple DNS lookup do? Say for `taler.net`?

- NS of **net** is `a.gtld-servers.net`
- NS of **taler.net** is `dns1.name-services.com`
- NS of `com` is `a.gtld-servers.net`
- CNAME of **taler.net** is **pixel.net.in.tum.de**
- NS of **de** is `n.de.net`
- NS of **net** was `a.gtld-servers.net`
- NS of `de.net` is `ns1.denic.de`
- NS of **tum.de** is `dns1.lrz.de`
- NS of `lrz.de` is `dns1.lrz.de`
- NS of **in.tum.de** is `tuminfo1.informatik.tu-muenchen.de`
- NS of `tu-muenchen.de` is `ws-han1.wip-ip.dfn.de`
- NS of `dfn.de` is `ws-han1.wip-ip.dfn.de`
- NS of **net.in.tum.de** is `dns1.lrz.de`
- A of **pixel.net.in.tum.de** is 131.159.20.32

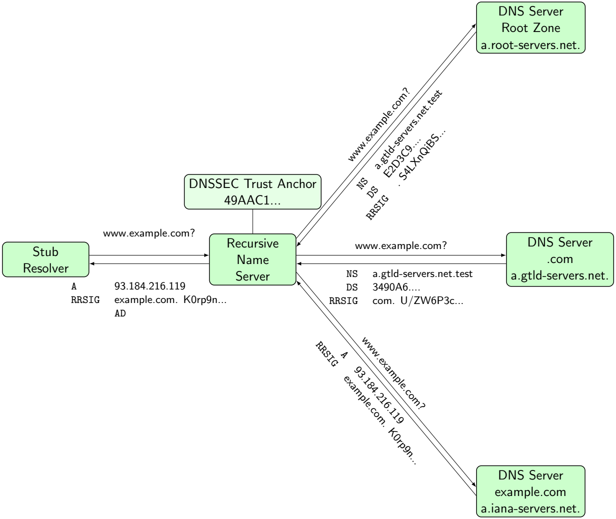
Exemplary Attacks: MORECOWBELL



Exemplary Attacks: QUANTUMDNS

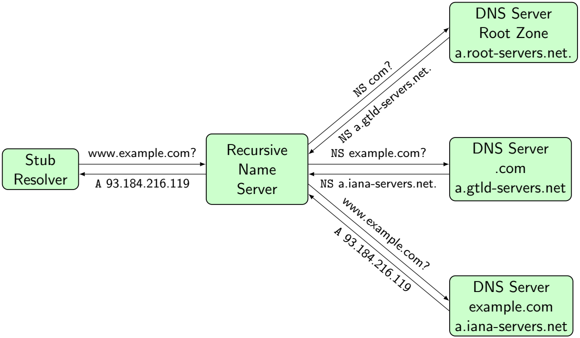


DNSSEC

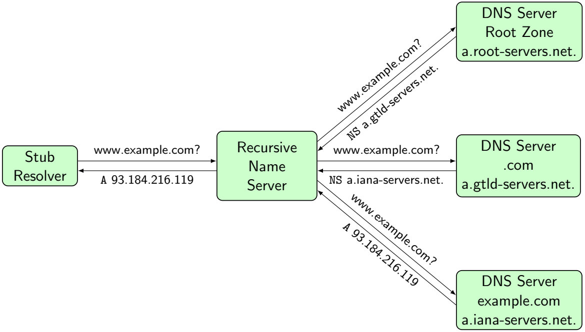


DNS Server
Root Zone
a.root-servers.net.
www.example.com?
NS a.gtld-servers.net.test
DS E2D3C9....
RRSIG . S4LXnQiBS...
DNSSEC Trust Anchor
49AAC1...
www.example.com?
Stub Resolver
Recursive Name Server
A 93.184.216.119
RRSIG example.com. K0rp9n...
AD
www.example.com?
DNS Server
.com
a.gtld-servers.net.
NS a.gtld-servers.net.test
DS 3490A6....
RRSIG com. U/ZW6P3c...
www.example.com?
A 93.184.216.119
RRSIG example.com. K0rp9n...
DNS Server
example.com
a.iana-servers.net.

Query Name Minimization



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

<table>
<tr><td></td><td>HTTPS</td></tr>
<tr><td>DNS</td><td>TLS</td></tr>
<tr><td>UDP</td><td>TCP</td></tr>
<tr><td colspan="2">IPv4</td></tr>
<tr><td colspan="2">Ethernet</td></tr>
<tr><td colspan="2">Phys. Layer</td></tr>
</table>

The Textbook Version of the Internet

Layering, ≈ 1990

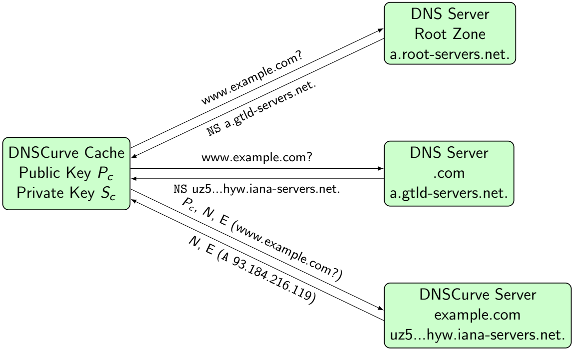
	HTTPS
DNS	TLS
UDP	TCP
IPv4	
Ethernet	
Phys. Layer	

"Layering", ≈ 2020

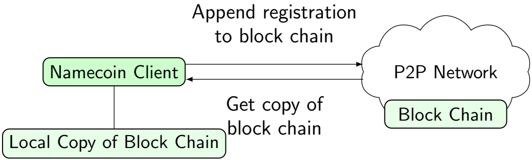
HTTPS	libmicrohttpd
TLS-with-DANE	libgnutls
DNS-over-TLS	libunbound
TLS*	libnss
TCP	Linux
IPv6	Linux
Ethernet	
Phys. Layer	

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

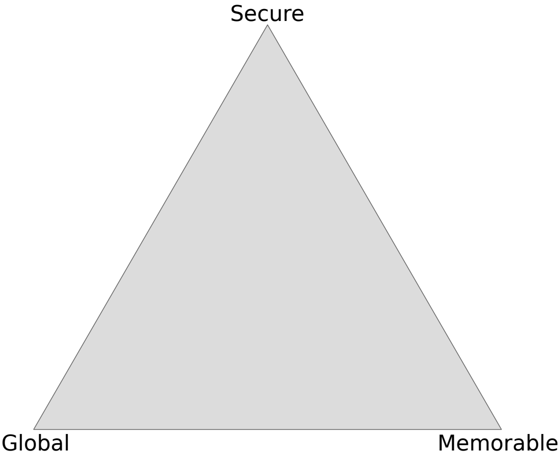
DNSCurve


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

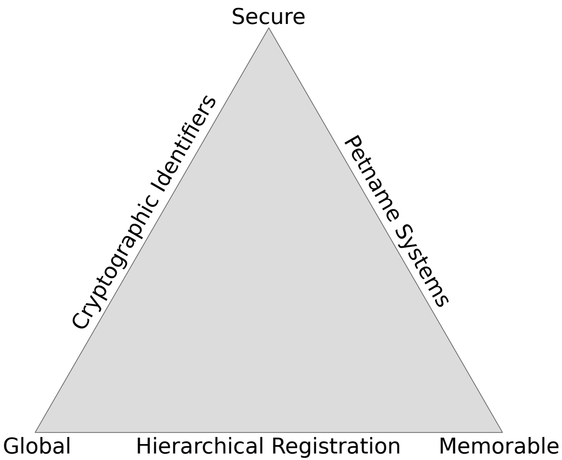

Append registration
to block chain
Namecoin Client
P2P Network
Block Chain
Get copy of
block chain
Local Copy of Block Chain

Zooko's Triangle



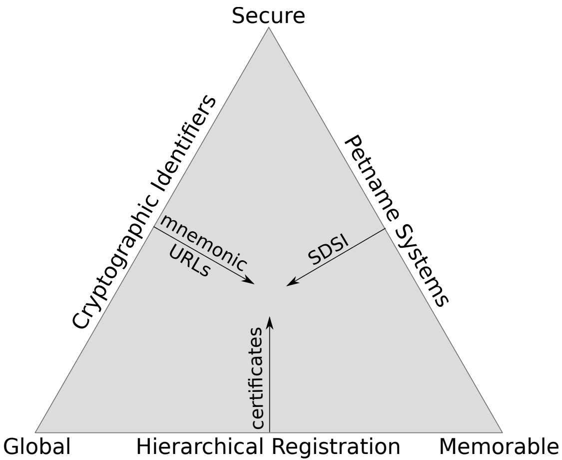
A name system can only fulfill **two**!

Zooko's Triangle



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

Zooko's Triangle



DNSSEC security is broken by design (adversary model!)

Namecoin

Namecoin

- Memorable:

Namecoin

- Memorable: Check
- Global:

Namecoin

- Memorable: Check
- Global: Check
- Secure:

Namecoin

- Memorable: Check
- Global: Check
- Secure: different adversary model!

Namecoin

- Memorable: Check
- Global: Check
- Secure: different adversary model!

⇒ Availability of names (registration rate) is restricted

Namecoin

- Memorable: Check
- Global: Check
- Secure: different adversary model!

⇒ Availability of names (registration rate) is restricted

⇒ Adversary must not have 51% compute power

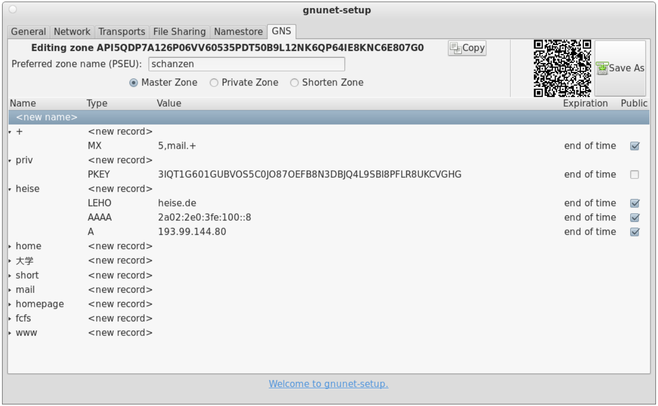
The GNU Name System[1]

Properties of GNS

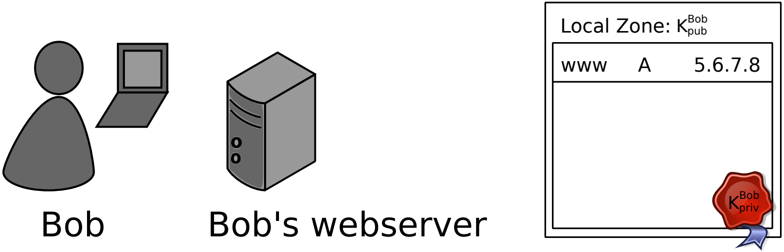
- Decentralized name system with secure memorable names
- Delegation used to achieve transitivity
- Achieves query and response privacy
- Provides alternative public key infrastructure
- Interoperable with DNS

[1] Joint work with Martin Schanzenbach and Matthias Wachs

Zone Management: like in DNS



Name resolution in GNS

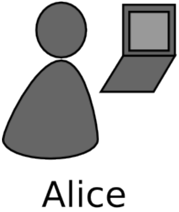


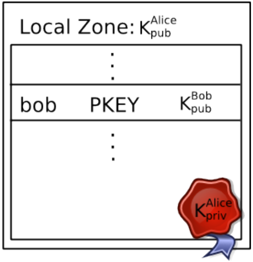
- Bob can locally reach his webserver via **www.gnu**

Secure introduction

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

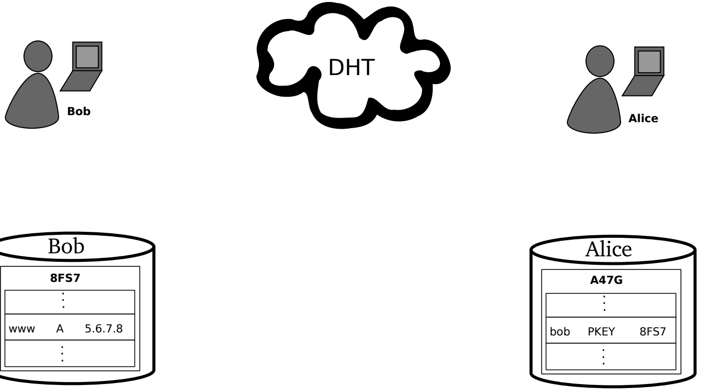
Delegation



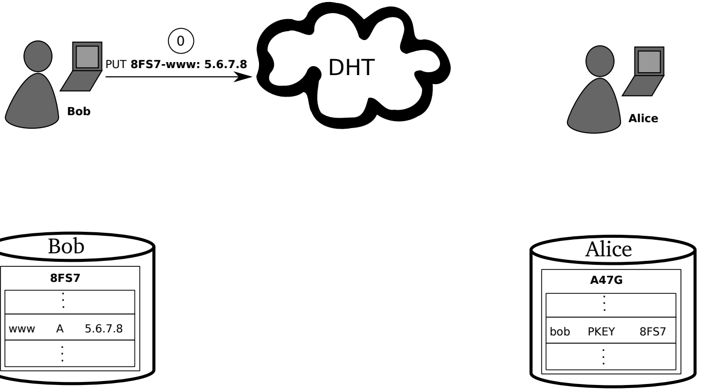


- Alice learns Bob's public key
- Alice creates delegation to zone K_{pub}^{Bob} under label **bob**
- Alice can reach Bob's webserver via **www.bob.gnu**

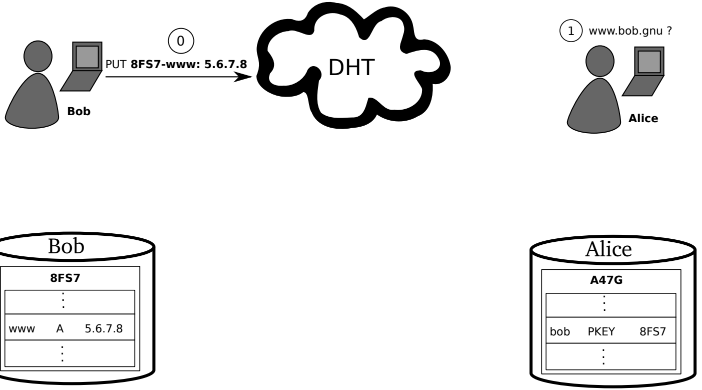
Name Resolution



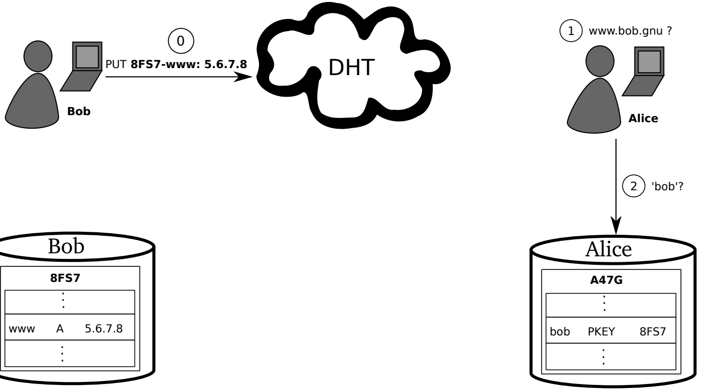
Name Resolution



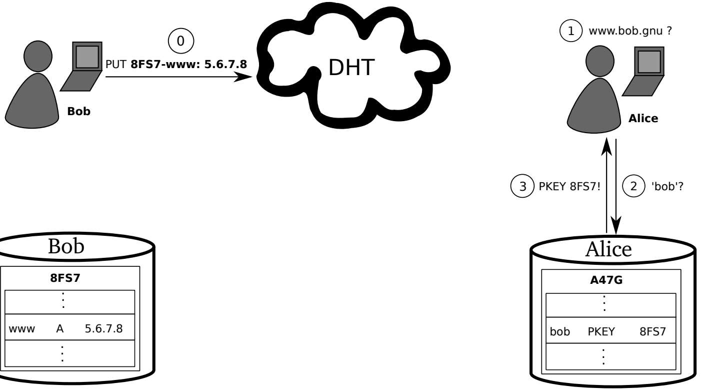
Name Resolution



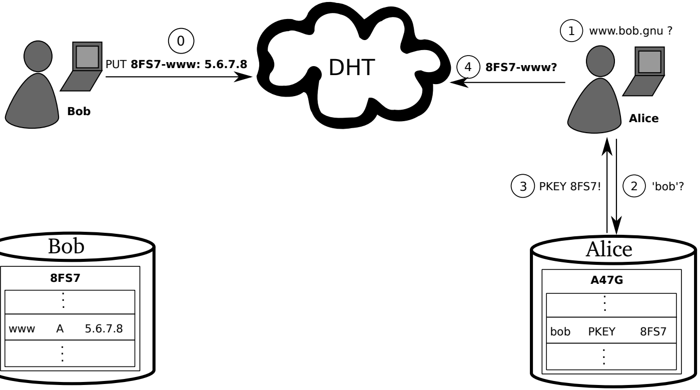
Name Resolution



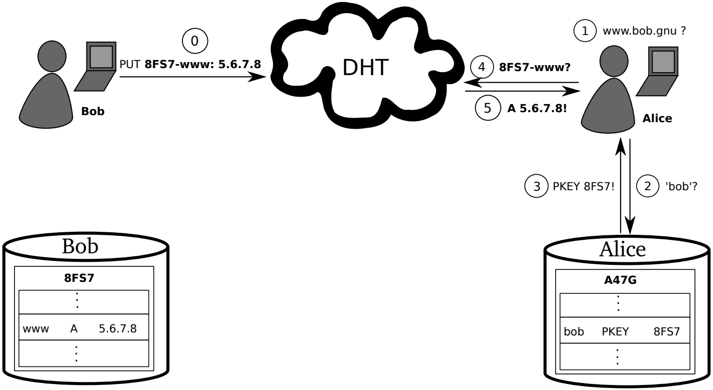
Name Resolution



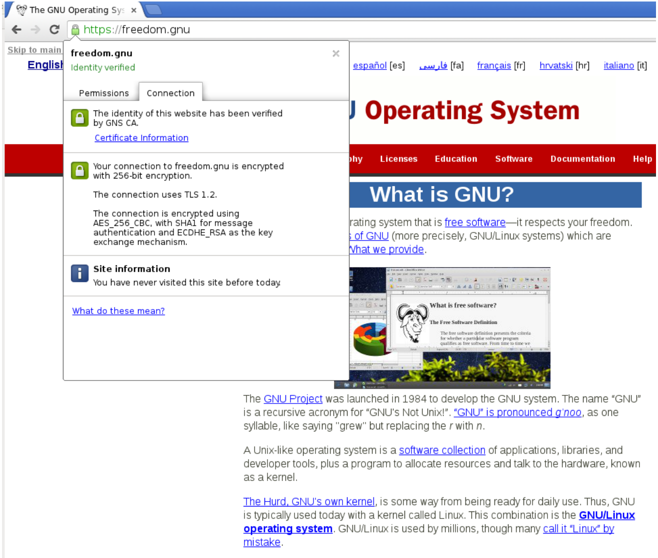
Name Resolution



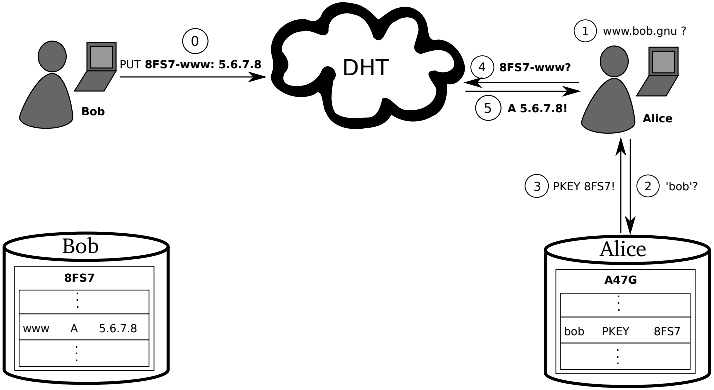
Name Resolution



GNS as PKI (via DANE/TLSA)



Privacy Issue: DHT



Query Privacy: Terminology

G	generator in ECC curve, a point
n	size of ECC group, $n := \|G\|$, n prime
x	private ECC key of zone ($x \in \mathbb{Z}_n$)
P	public key of zone, a point $P := xG$
l	label for record in a zone ($l \in \mathbb{Z}_n$)
$R_{P,l}$	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 $q_{P,l}$

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

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

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

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

Query Privacy: Cryptography

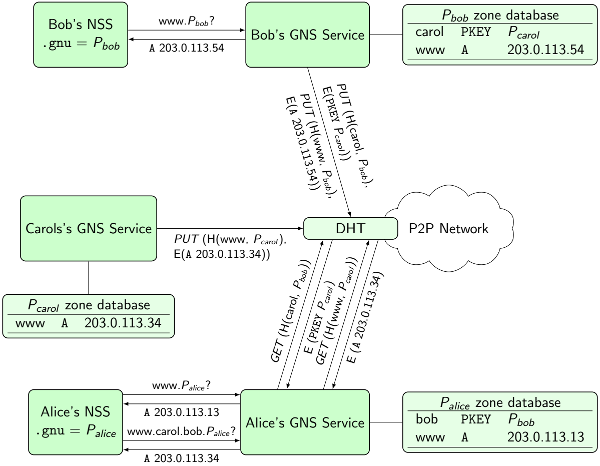
Publishing records $R_{P,l}$ as $B_{P,l}$ under key $q_{P,l}$

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

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 GNU Name System (GNS)



Revocation

Revocation Basics

- Revocation certificate (RC): message signed with private key
- Peer receives new valid RC, floods to all neighbours
- All peers store all valid RCs forever

$\Rightarrow$ Expensive operation $\Rightarrow$ proof-of-work

Revocation

Revocation Basics

- Revocation certificate (RC): message signed with private key
- Peer receives new valid RC, floods to all neighbours
- All peers store all valid RCs forever

$\Rightarrow$ Expensive operation $\Rightarrow$ proof-of-work

Revocation Magic

- Peers maybe offline during initial flood
- Network might be temporarily partitioned

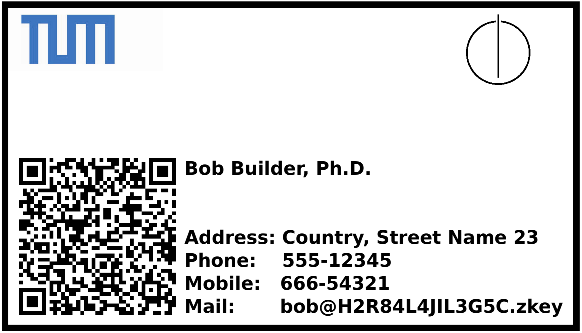
$\Rightarrow$ Need to reconsile revocation sets on connect

Whenever two peers establish a P2P connection, they must compute the set union of their RC sets!

The ".zkey" pTLD

- "LABELS.*PKEY*.zkey" format
- PKEY is the public key of the zone
- Works a bit like ".onion"

⇒ Globally unique identifiers!



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
- Eve learns the “NICK”, and GNS creates “krista.short.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
- Eve learns the “NICK”, and GNS creates “krista.short.gnu”
- Memorable, short trust path in the future! TOFU!
- Krista better pick a reasonably unique NICK.

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

Practical Concerns

- Name registration
- Support for browsing
- New record types
- Integration with applications
- State of the implementation

Registering a name in GNS

- Bob gives his PKEY to his **friends** via QR code
- or registers it at the **GNUnet fcfs** authority *pin.gnu* as "bob"
- $\rightarrow$ Bob's friends can resolve his records via *.*petname*.gnu
- $\rightarrow$ or *.bob.pin.gnu

From DNS to GNS

Names are not globally unique, but ...

... we need support for Virtual Hosting!

... we need support for SSL!

From DNS to GNS

Names are not globally unique, but ...

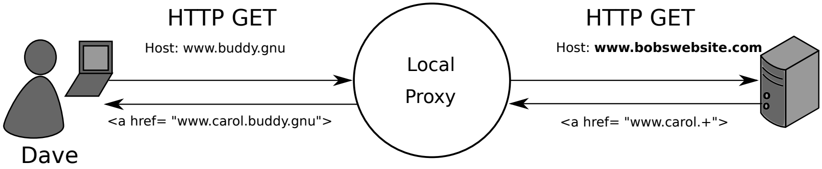
... we need support for Virtual Hosting!

... we need support for SSL!

Solution: Client Side SOCKS Proxy

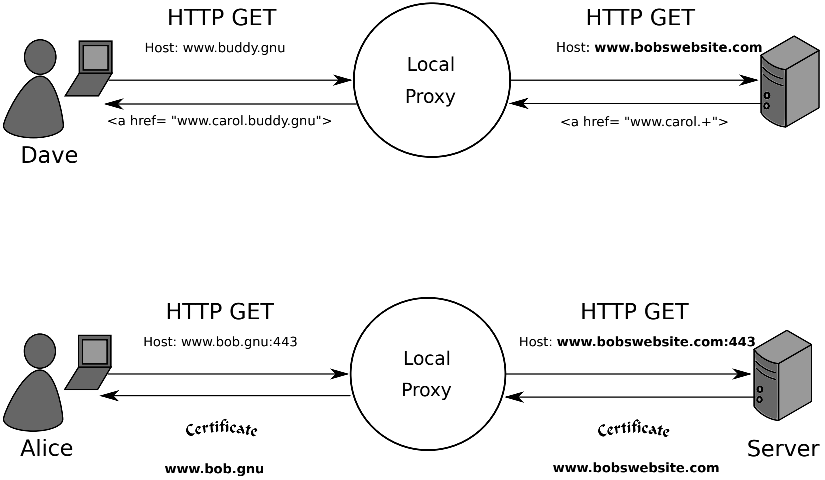
Legacy Hostname (LEHO) Records

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.



Long-Term Vision

- Integration with browser and HTTP server
- HTTP server receives "GNS-Zone: PKEY" instead of "Hostname"
- HTTP client uses "TLSA" record of GNS, instead of "LEHO"

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.

New Record Types

- PKEY: delegate to another GNS zone
- NICK: preferred names for shortening
- LEHO: legacy hostname

New Record Types

- PKEY: delegate to another GNS zone
- NICK: preferred names for shortening
- LEHO: legacy hostname
- GNS2DNS: delegate to DNS
- VPN: peers hosting TCP/IP services
- PHONE: call users using `gnunet-conversation`

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

VPN Delegation

- Delegates to GNUnet VPN
- VPN record specifies:
  - Identity of hosting peer (no anonymity!)
  - Service identifier (hash code)
- GNS can map VPN record to A/AAAA record of `gnunet-vpn` tunnel

PHONE service

- PHONE record specifies:
  - Identity of hosting peer (no anonymity yet!)
  - Line number (to support multiple phones per peer)

Application Integration

- SOCKS proxy (`gnunet-gns-proxy`)
- NSS plugin
- DNS packet interception (`gnunet-dns-service`)
- GNS (C) API
- GNS (IPC) protocol
- GNS command-line tool

Current State

- GNS part of GNUnet since 0.9.3
- Crypto changed to Curve25519 in 0.10.0
- Internationalized Domain Names are supported

Current State

- GNS part of GNUnet since 0.9.3
- Crypto changed to Curve25519 in 0.10.0
- Internationalized Domain Names are supported
- Installation is “non-trivial” (for your parents)
- Needs more work on reverse lookup

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	✓	✗	✓	✓	✓	✓	✗
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	✓	✗	✓		✓	✓	✗
SMP/PANDA	✓	✗	✓	✓	✓	✓	✗
Namecoin	✗	✓	✗	✓	✓	✗	✓
GNS	✓	✓	✓	✓	✓	✓	✓

Conclusion

- We have decentralized the PKI
- Privacy and security are preserved

Conclusion

- We have decentralized the PKI
- Privacy and security are preserved

Do you have any questions?

References:

- Nathan Evans and Christian Grothoff. *R^5N. 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.