

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 GUNet: 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`



(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

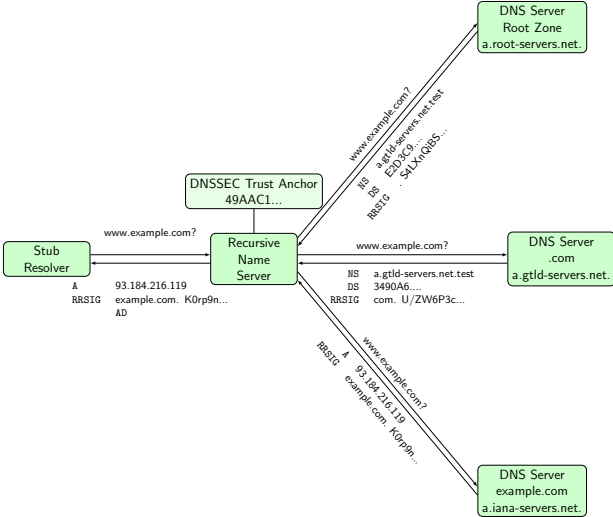
(U) New Hotness

- (TS//SI//REL) QUANTUMBISCUIT
 - Redirection based on keyword
 - 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

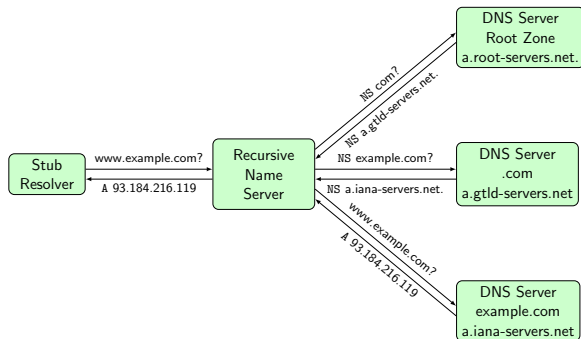


TOP SECRET//COMINT//REL TO USA, FVEY//20320108

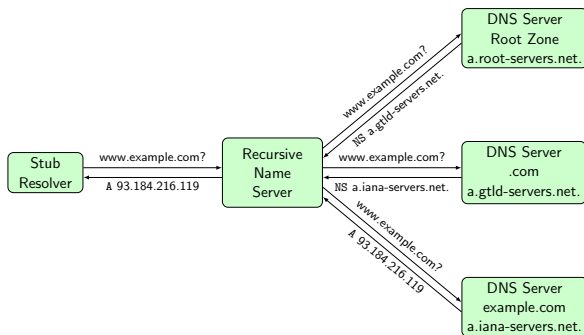
DNSSEC



Query Name Minimization



DNS over TLS



The Textbook Version of the Internet

Layering, \approx 1990

	HTTPS
DNS	TLS
UDP	TCP
IPv4	
Ethernet	
Phys. Layer	

The Textbook Version of the Internet

Layering, \approx 1990

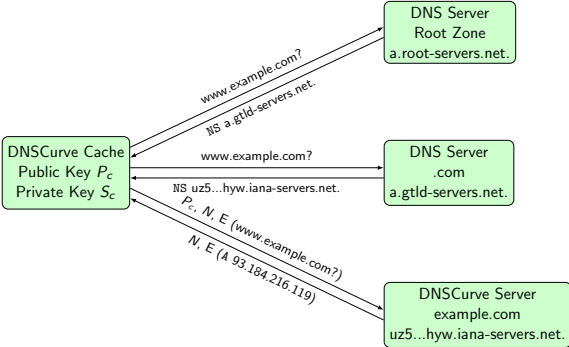
	HTTPS
DNS	TLS
UDP	TCP
IPv4	
Ethernet	
Phys. Layer	

"Layering", \approx 2020

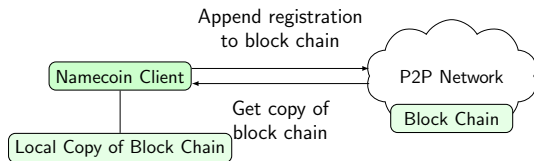
HTTPS	libmicrohttpd
TLS-with-DANE	libgnutls
DNS-over-TLS	libunbound
TLS*	libnss
TCP	Linux
IPv6	Linux
Ethernet	
Phys. Layer	

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

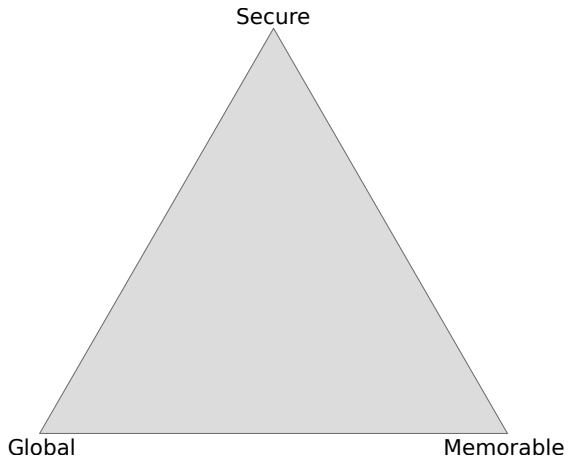
DNSCurve



Namecoin

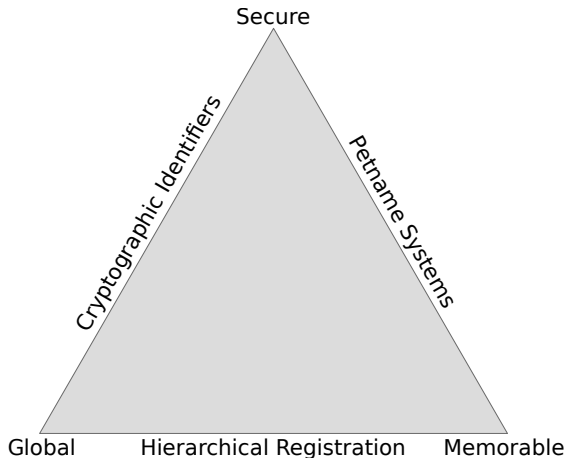


Zooko's Triangle



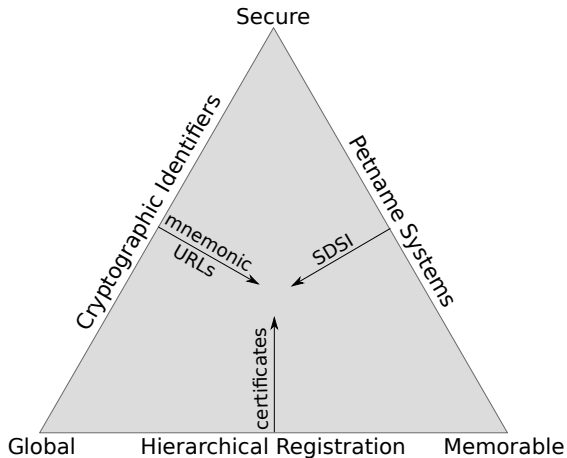
A name system can only fulfill **two!**

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

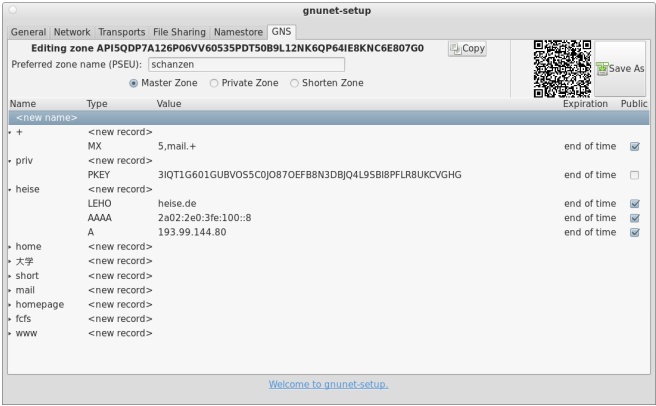
The GNU Name System¹

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

¹Joint work with Martin Schanzenbach and Matthias Wachs

Zone Management: like in DNS

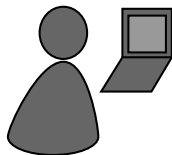


The screenshot shows the 'gnetnet-setup' application window. At the top, there are tabs for 'General', 'Network', 'Transports', 'File Sharing', 'Namestore', and 'GNS'. The main title is 'Editing zone API5QDP7A126P06VV60535PDT50B9L12NK6QP64IE8KNC6E807G0'. Below this, there is a text field for 'Preferred zone name (PSEU):' containing 'schanzen'. To the right of this field is a 'Copy' button and a QR code. Below the text field are three radio buttons: 'Master Zone' (selected), 'Private Zone', and 'Shorten Zone'. To the right of the QR code is a 'Save As' button.

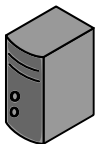
Name	Type	Value	Expiration	Public
<new name>				
+	<new record>			
	MX	5,mail.+	end of time	<input checked="" type="checkbox"/>
priv	<new record>			
	PKEY	3IQT1G601GUBVO55CJO87OEFB8N3DBJQ4L95BI8PFLR8UKCVGHG	end of time	<input type="checkbox"/>
heise	<new record>			
	LEHO	heise.de	end of time	<input checked="" type="checkbox"/>
	AAAA	2a02:2e0:3fe:100::8	end of time	<input checked="" type="checkbox"/>
	A	193.99.144.80	end of time	<input checked="" type="checkbox"/>
home	<new record>			
大学	<new record>			
short	<new record>			
mail	<new record>			
homepage	<new record>			
fcs	<new record>			
www	<new record>			

At the bottom of the window, there is a blue link: [Welcome to gnetnet-setup.](#)


Name resolution in GNS



Bob



Bob's webserver

Local Zone: K_{pub}^{Bob}		
www	A	5.6.7.8
		


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

Secure introduction



TUM

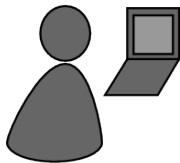


 **Bob Builder, Ph.D.**

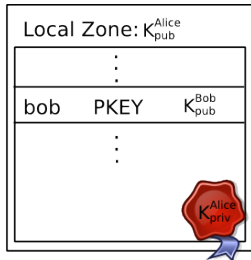
Address: Country, Street Name 23
Phone: 555-12345
Mobile: 666-54321
Mail: bob@H2R84L4JIL3G5C.zkey

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

Delegation

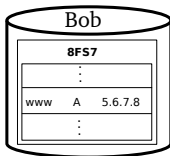
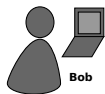


Alice

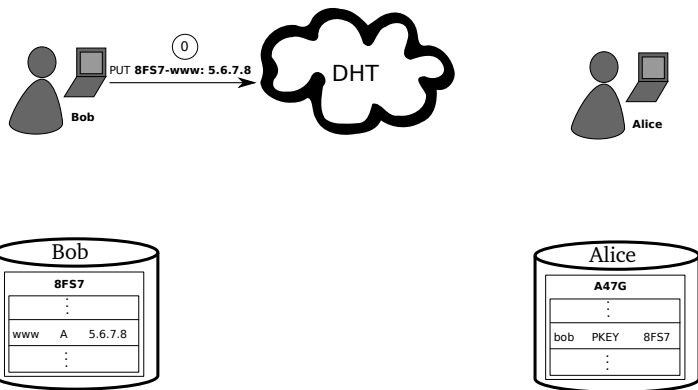


- ▶ Alice learns Bob's public key
- ▶ Alice creates delegation to zone K_{pub}^{Bob} under label **bob**
- ▶ Alice can reach Bob's webserver via **www.bob.gnu**

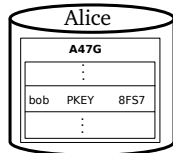
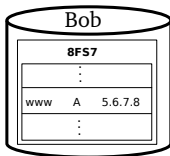
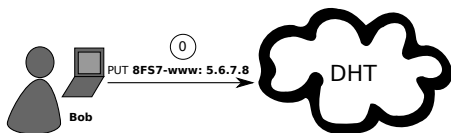
Name Resolution



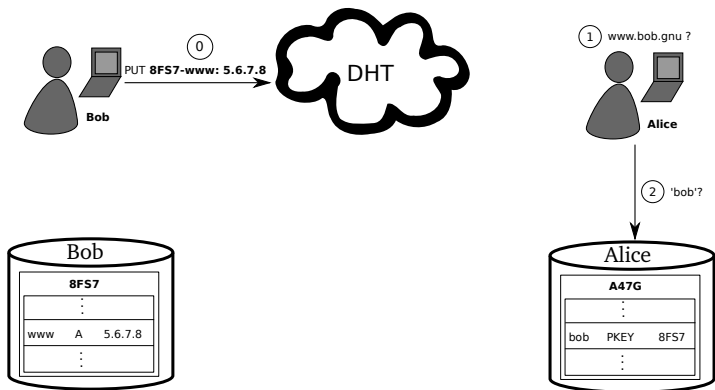
Name Resolution



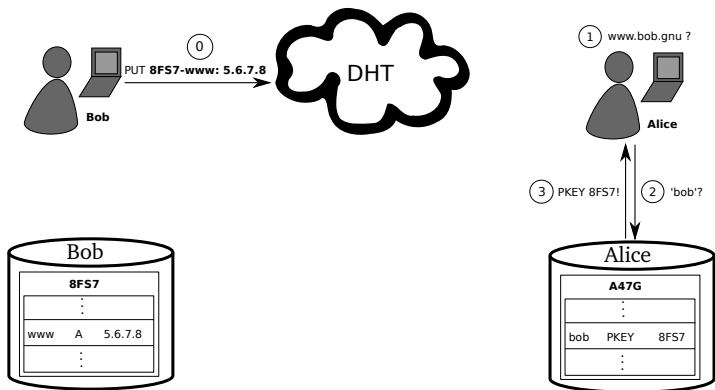
Name Resolution



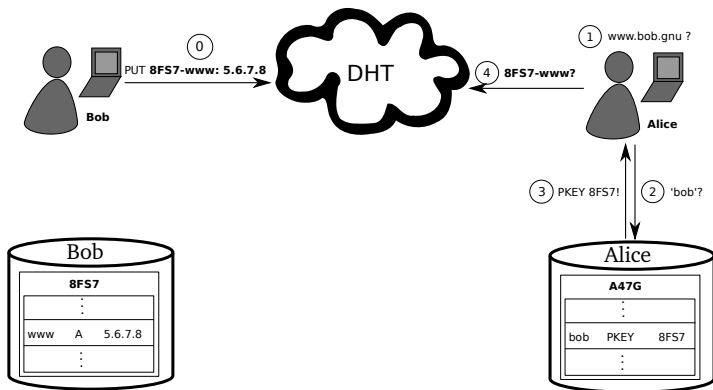
Name Resolution



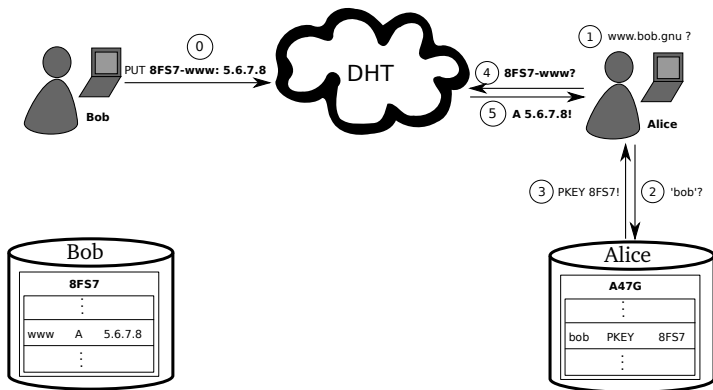
Name Resolution



Name Resolution



Name Resolution



GNS as PKI (via DANE/TLSA)

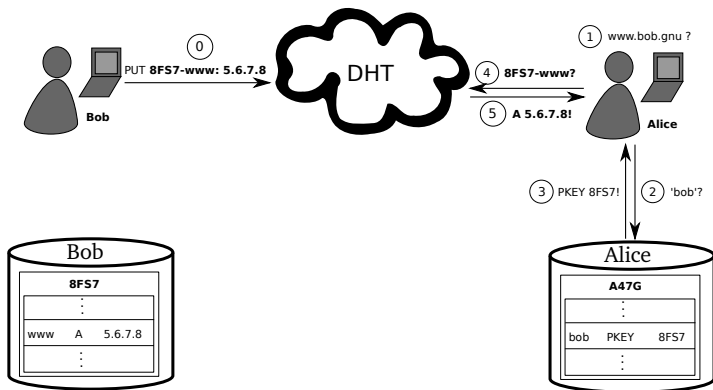
The screenshot shows a web browser window with the address bar displaying `https://freedom.gnu`. A security warning dialog is overlaid on the page. The dialog has a title bar that says "freedom.gnu" and "Identity verified". It contains several sections: "Permissions" and "Connection" tabs, a green lock icon indicating identity verification by GNS CA, a green lock icon indicating 256-bit encryption, and a section titled "Site information" stating "You have never visited this site before today". The background website has a red header with navigation links: "Why", "Licenses", "Education", "Software", "Documentation", and "Help". Below the header is a blue section titled "What is GNU?".

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



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 $R_{P,I}$ as $B_{P,I}$ under key $q_{P,I}$

$$h := H(I, P) \quad (1)$$

$$d := h \cdot x \pmod n \quad (2)$$

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

$$q_{P,I} := H(dG) \quad (4)$$

Query Privacy: Cryptography

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

$$h := H(I, P) \quad (1)$$

$$d := h \cdot x \pmod n \quad (2)$$

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

$$q_{P,I} := H(dG) \quad (4)$$

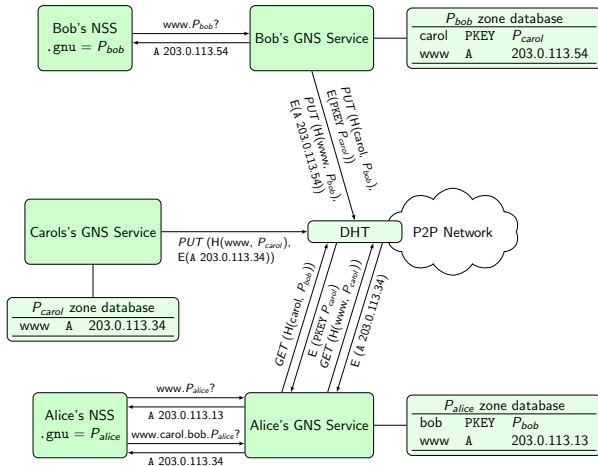
Searching for records under label I in zone P

$$h := H(I, P) \quad (5)$$

$$q_{P,I} := H(hP) = H(hxG) = H(dG) \Rightarrow \text{obtain } B_{P,I} \quad (6)$$

$$R_{P,I} = D_{HKDF(I,P)}(B_{P,I}) \quad (7)$$

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
- ⇒ Expensive operation ⇒ 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
- ⇒ Expensive operation ⇒ proof-of-work

Revocation Magic

- ▶ Peers maybe offline during initial flood
 - ▶ Network might be temporarily partitioned
- ⇒ Need to reconcile 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!



TUM



 **Bob Builder, Ph.D.**

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

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"
- ▶ → Bob's friends can resolve his records via **.petname.gnu*
- ▶ → 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 ...

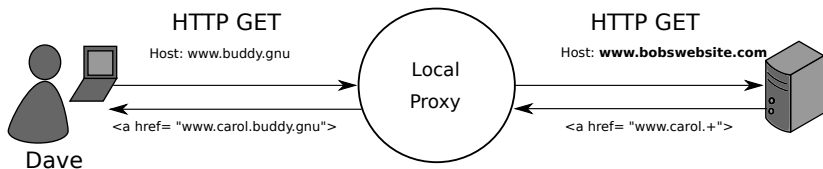
... we need support for Virtual Hosting!

... we need support for SSL!

Solution: Client Side SOCKS Proxy

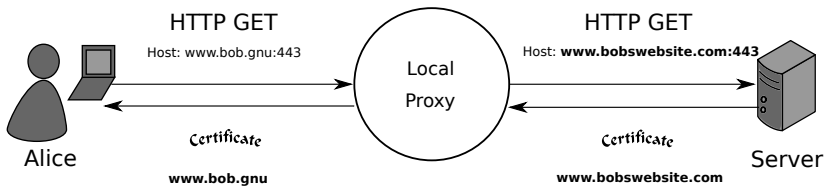
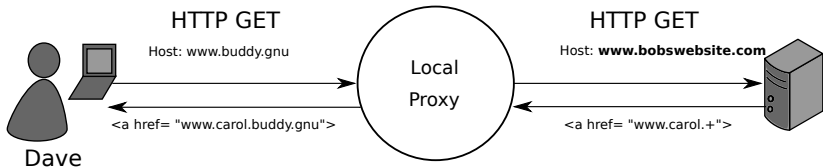
Legacy Hostname (LEHO) Records

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 GUNet 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 GUNet since 0.9.3
- ▶ Crypto changed to Curve25519 in 0.10.0
- ▶ Internationalized Domain Names are supported

Current State

- ▶ GNS part of GUNet 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. *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.