The GNU Name System

Christian Grothoff

Technische Universität München

16.08.2014

“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
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
The GNU Name System

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
Apropos DNS

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

▶ NS of net is a.gtld-servers.net
Apropos DNS

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
Apropos DNS

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
Apropos DNS

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
Apropos DNS

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
Apropos DNS

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
Apropos DNS

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
Apropos DNS
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
Apropos DNS

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
Apropos DNS

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
Apropos DNS

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
Apropos DNS

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
Apropos DNS

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 lriz.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
Apropos DNS

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
Zone Management: like in DNS

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Value</th>
<th>Expiration</th>
<th>Public</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>&lt;new record&gt;</td>
<td>5.mail.+</td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>prv</td>
<td>&lt;new record&gt;</td>
<td>3IQT1G601GUBV055C0J0870EF8B8N3DBJQ4L95B15FLR8UKCVGHG</td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>heise</td>
<td>PKEY</td>
<td>3IQT1G601GUBV055C0J0870EF8B8N3DBJQ4L95B15FLR8UKCVGHG</td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>LEHO</td>
<td>heise.de</td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AAAA</td>
<td>2a02:2e0:3fe:100::8</td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>193.99.144.80</td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>home</td>
<td>&lt;new record&gt;</td>
<td></td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>大学</td>
<td>&lt;new record&gt;</td>
<td></td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>&lt;new record&gt;</td>
<td></td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>mail</td>
<td>&lt;new record&gt;</td>
<td></td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>homepage</td>
<td>&lt;new record&gt;</td>
<td></td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>fcfs</td>
<td>&lt;new record&gt;</td>
<td></td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>www</td>
<td>&lt;new record&gt;</td>
<td></td>
<td>end of time</td>
<td></td>
</tr>
</tbody>
</table>
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

- Alice learns Bob’s public key
- Alice creates delegation to zone $K^{Bob}_{pub}$ under label **bob**
- Alice can reach Bob’s webserver via **www.bob.gnu**
Name Resolution

Bob

DHT

Alice

Bob

<table>
<thead>
<tr>
<th>8FS7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>www  A 5.6.7.8</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Alice

<table>
<thead>
<tr>
<th>A47G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>bob  PKEY 8FS7</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

...
Name Resolution

Bob

PUT 8FS7-www: 5.6.7.8

DHT

Alice

Bob

8FS7

www A 5.6.7.8

Alice

A47G

bob PKEY 8FS7
Name Resolution

Bob

PUT 8FS7-www: 5.6.7.8

DHT

Alice

1 www.bob.gnu ?

Bob

<table>
<thead>
<tr>
<th>8FS7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>www  A</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Alice

<table>
<thead>
<tr>
<th>A47G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>bob  PKEY 8FS7</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Name Resolution

1. Bob PUTs `8FS7-www: 5.6.7.8` into the DHT.
4. Alice can now access the website `www.bob.gnu`.
Name Resolution

Bob

PUT 8FS7-www: 5.6.7.8

DHT

Alice

www.bob.gnu?

1

www
A
5.6.7.8

8FS7

Bob

A47G

...
...

PKEY 8FS7!

Alice

bob
PKEY
8FS7

...
...

8FS7
Name Resolution

1. User Bob makes a request to put a record in the DHT.
   - Request: `PUT 8FS7-www: 5.6.7.8`.

2. The DHT stores the request and returns a key `bob`.
   - Response: `8FS7-www?`

3. User Alice retrieves the record using the key `bob`.
   - Request: `PKEY 8FS7!`

4. The DHT returns the record to Alice.
   - Response: `bob PKEY 8FS7`
Name Resolution

Bob

PUT 8FS7-www: 5.6.7.8

DHT

Alice

PUT 8FS7-www: 5.6.7.8

A 5.6.7.8!

Alice

PKEY 8FS7!

2 'bob'?

Bob

8FS7

www A 5.6.7.8

Alice

A47G

bob PKEY 8FS7

www.bob.gnu ?

1

Bob Alice

DHT

'bob'?2 3 PKEY 8FS7!

8FS7-www?

4

A 5.6.7.8!

5
GNS as PKI (via DANE/TLSA)

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 ˈɡɪən, 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

1. www.bob.gnu
2. 'bob'
3. PKEY 8FS7!
4. 8FS7-www?
5. A 5.6.7.8!

PUT 8FS7-www: 5.6.7.8

Bob

<table>
<thead>
<tr>
<th>8FS7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>www  A  5.6.7.8</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Alice

<table>
<thead>
<tr>
<th>A47G</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>bob  PKEY  8FS7</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

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

\begin{align*}
h & : = H(l, P) \\
d & : = h \cdot x \mod n \\
B_{P,l} & : = S_d(E_{HKDF}(l, P)(R_{P,l})), dG \\
q_{P,l} & : = H(dG)
\end{align*}
Query Privacy: Cryptography

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

\[
h : = H(l, P) \tag{1}
\]
\[
d : = h \cdot x \mod n \tag{2}
\]
\[
B_{P,l} : = S_d(E_{HKDF(l,P)}(R_{P,l})), dG \tag{3}
\]
\[
q_{P,l} : = H(dG) \tag{4}
\]

Searching for records under label $l$ in zone $P$

\[
h : = H(l, P) \tag{5}
\]
\[
q_{P,l} : = H(hP) = H(h \cdot xG) = H(dG) \Rightarrow \text{obtain } B_{P,l} \tag{6}
\]
\[
R_{P,l} = D_{HKDF(l,P)}(B_{P,l}) \tag{7}
\]
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 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!

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)
“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 **GNU**net **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
- BOX: proper support for TLSA (and SRV)
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)
- gnunet-conversation uses *reverse lookup* for caller ID
BOX records

- TLSA records in DNS are under a special name
- Performing a second lookup is **bad**
BOX records

- TLSA records in DNS are under a special name
- Performing a second lookup is **bad**

⇒ GNS BOX records include TLSA information under primary label!
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)
- SOCKS proxy is known to be problematic
Conclusion

- Decentralization is necessary
- Encryption requires a PKI
- GNS is a modern PKI designed for privacy
- Please consider adding GNS support to your code!
Conclusion

- Decentralization is necessary
- Encryption requires a PKI
- GNS is a modern PKI designed for privacy
- Please consider adding GNS support to your code!
Do you have any questions?

References:


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

- Memorable
- Global
- Secure

⇒ Availability of names (registration rate) is restricted
⇒ Adversary must not have 51% compute power
Namecoin

- Memorable:
Namecoin

- Memorable: Check
- Global:
Namecoin

- Memorable: Check
- Global: Check
- Secure:
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