The GNU name system

Christian Grothoff

Inria Rennes Bretagne Atlantique

11.7.2016

“The Domain Name System is the Achilles heel of the Web.” –Tim Berners-Lee
Trouble at the root

- ICANN asserts ccTLDs are not property to avoid seizure of .ir by US court
- ICANN approves .xxx despite objections from US conservative groups
- IETF approves .onion, but rejects .bit
- EU objects to US/AU/NZ plans for .wine to safeguard EU geographic indications system
- The Pirate Bay constantly changes its gTLD domain name due to censorship

Controlling gTLDs is about money & power.
Trouble in operations

▶ DNS remains a major source of traffic amplification for DDoS
▶ DNS censorship (i.e. by China) causes collateral damage in other countries
▶ DNS is part of the mass surveillance apparatus (MCB)
▶ DNS is abused for the offensive cyber war (QUANTUMDNS)

Band aid solutions\(^1\) will **not** fix this.

---

\(^1\)DNS-over-TLS, DNSSEC, DPRIVE, ...
The GNU name system

- 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
Zone management: like in DNS

![Image of gnunet-setup interface]

<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>priv</td>
<td>&lt;new record&gt;</td>
<td>PKEY</td>
<td>end of time</td>
<td></td>
</tr>
<tr>
<td>heise</td>
<td>&lt;new record&gt;</td>
<td>heise.de</td>
<td>end of time</td>
<td>✔</td>
</tr>
<tr>
<td>leho</td>
<td>&lt;new record&gt;</td>
<td>2a02:2e0:3fe:100::8</td>
<td>end of time</td>
<td>✔</td>
</tr>
<tr>
<td>home</td>
<td>&lt;new record&gt;</td>
<td>193.99.144.80</td>
<td>end of time</td>
<td>✔</td>
</tr>
<tr>
<td>大学</td>
<td>&lt;new record&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>short</td>
<td>&lt;new record&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>mail</td>
<td>&lt;new record&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>homepage</td>
<td>&lt;new record&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fcfs</td>
<td>&lt;new record&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>www</td>
<td>&lt;new record&gt;</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Name resolution in GNS

- 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 $K_{pub}^{Bob}$ 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

DHT

Alice

Bob

Alice

8FS7

www A 5.6.7.8

A47G

bob PKEY 8FS7
Name resolution

Bob

PUT 8FS7-www: 5.6.7.8

DHT

Alice

1 www.bob.gnu ?

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

www.bob.gnu ?

1

'bob'?

2
Name resolution

Bob

DHT

www.bob.gnu

Alice

PUT 8FS7-www: 5.6.7.8

0

1 www.bob.gnu ?

1 www.bob.gnu ?

PKEY 8FS7!

2 'bob'?

3 PKEY 8FS7!

Alice

Bob

8FS7

www A 5.6.7.8

A47G

bob PKEY 8FS7

...
Name resolution

1. Alice asks for `www.bob.gnu`.
2. Alice queries the DHT for `bob`.
3. Alice returns `PKEY 8FS7`.
4. Bob puts `8FS7-www: 5.6.7.8` into the DHT.

Bob's database:

| 8FS7 | ... | www | A | 5.6.7.8 | ... |

Alice's database:

| A47G | ... | bob | PKEY | 8FS7 | ... |
Name resolution

Bob

PUT 8FS7-www: 5.6.7.8

DHT

Alice

www.bob.gnu ?

'bob'? PKEY 8FS7!

8FS7-www?

A 5.6.7.8!

PUT 8FS7-www: 5.6.7.8

www      A      5.6.7.8

Bob

Alice

8FS7

www A 5.6.7.8

A47G

bob PKEY 8FS7

...
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 gui 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

Bob

PUT 8FS7-www: 5.6.7.8

DHT

Alice

www.bob.gnu

'bob'?

PKEY 8FS7!

8FS7-www?

A 5.6.7.8!

www      A      5.6.7.8

Bob

Alice

8FS7


A47G

bob     PKEY       8FS7   

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 $q_{P,l}$

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

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

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

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

\[
h := H(I, P) \quad (5)
\]
\[
q_{P,l} := H(hP) = H(hxG) = H(dG) \Rightarrow \text{obtain } B_{P,l} \quad (6)
\]
\[
R_{P,l} = D_{HKDF(I,P)}(B_{P,l}) \quad (7)
\]
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
Next steps: go global

- “.gnu” is personal, fine for an address book or OSN
- But, everyone likes global names! “.fr” is global.
Next steps: go global

- “.gnu” is personal, fine for an address book or OSN
- But, everyone likes global names! “.fr” is global.
  - Scale DHT implementation to deal with millions of records
Next steps: go global

- “.gnu” is personal, fine for an address book or OSN
- But, everyone likes global names! “.fr” is global.
  - Scale DHT implementation to deal with millions of records
  - Use NSEC/NSEC3 to XFR “.fr” into GNS zone

⇒ Globally unique identifiers
⇒ No out-of-bailiwick lookups
⇒ Privacy
⇒ Censorship-resistance
Next steps: go global

- “.gnu” is personal, fine for an address book or OSN
- But, everyone likes global names! “.fr” is global.
  - Scale DHT implementation to deal with millions of records
  - Use NSEC/NSEC3 to XFR “.fr” into GNS zone
  - Hijack “.fr” via NSS like we hijack “.gnu” today

⇒ Globally unique identifiers
⇒ No out-of-bailiwick lookups
⇒ Privacy
⇒ Censorship-resistance
Next steps: go global

- “.gnu” is personal, fine for an address book or OSN
- But, everyone likes global names! “.fr” is global.
  - Scale DHT implementation to deal with millions of records
  - Use NSEC/NSEC3 to XFR “.fr” into GNS zone
  - Hijack “.fr” via NSS like we hijack “.gnu” today
  - Great: no ICANN/IETF approval for ccTLD needed!
  - Scale to for all gTLDs supporting DNSSEC

⇒ Globally unique identifiers
⇒ No out-of-bailiwick lookups
⇒ Privacy
⇒ Censorship-resistance
Conclusion

- Plan to obsolete the obsolete DNS protocol
- No root, no exclusive hierarchy, no control issues
- Delegation allows using zones of other users
- Trust paths explicit, trust agility
- Privacy-enhanced queries, censorship-resistant
- Reliable revocation
Do you have any questions?


