Decentralizing Privacy-Preserving Network Applications

Christian Grothoff

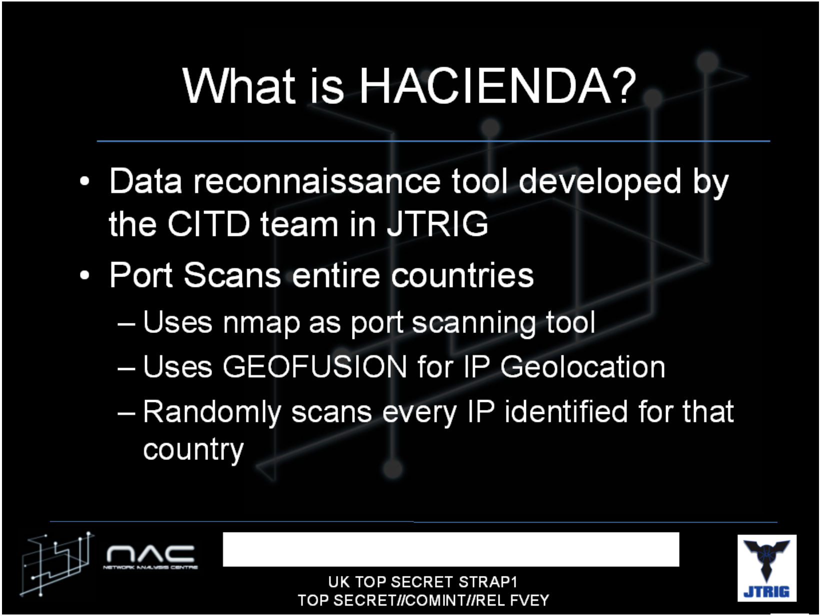
Inria Rennes Bretagne Atlantique

8.6.2017

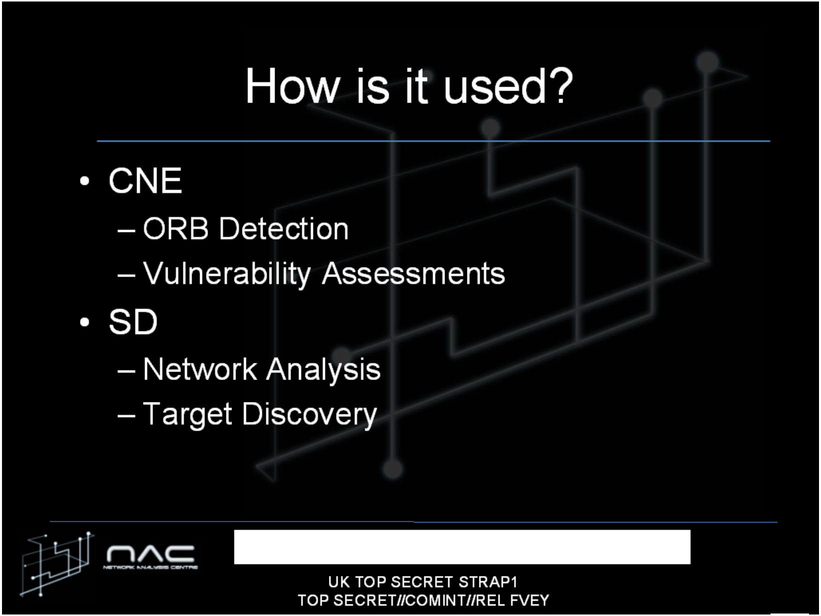
"Never doubt your ability to change the world." –Glenn Greenwald

The Internet is broken!

Example 1: Collateral Damage



Example 1: Collateral Damage



Example 1: Collateral Damage

TOP SECRET//COMINT

Communications Security Establishment | Centre de la sécurité des télécommunications

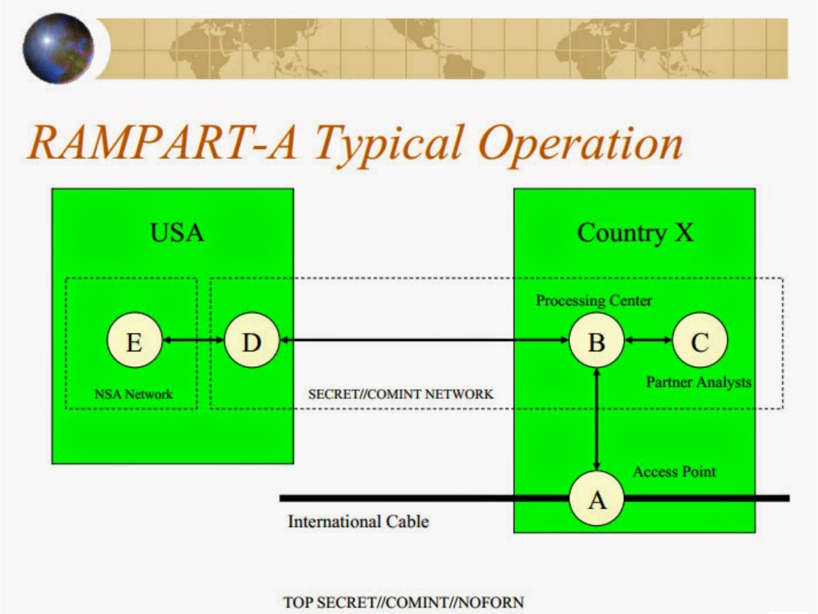
LANDMARK

- CSEC's Operational Relay Box (ORB) covert infrastructure used to provide an additional level of non-attribution; subsequently used for exploits and exfiltration
- 2-3 times/year, 1 day focused effort to acquire as many new ORBs as possible in as many non 5-Eyes countries as possible

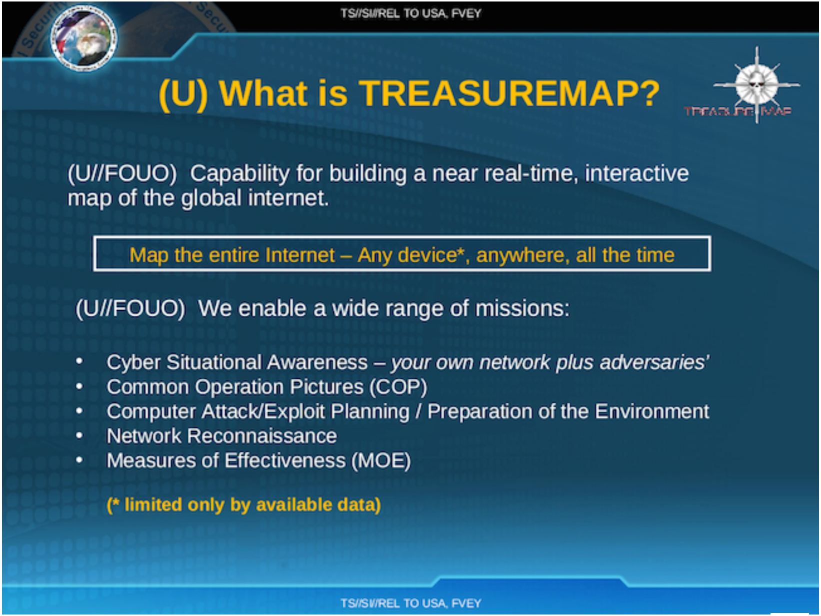
Canada

3

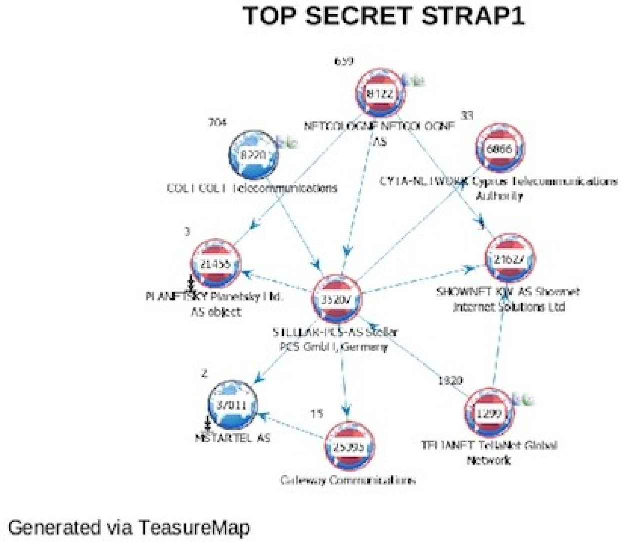
Example 2: Owning the Network



Example 2: Owning the Network



Example 2: Owning the Network



Why should you care?

If you are ...

- ... of any importance in the world, or
- ... a system or network administrator, or
- ... a security researcher, or
- ... in this room, or
- ... mistaken for any of the above,

Why should you care?

If you are ...

- ... of any importance in the world, or
- ... a system or network administrator, or
- ... a security researcher, or
- ... in this room, or
- ... mistaken for any of the above,

then you are probably a target.

So what if they listen to my calls?

- Kompromat — and you do not get to decide what is bad!
- Self-censorship
- Loss of business
- No privacy $\Rightarrow$ No free press $\Rightarrow$ No liberal democracy

So what if they listen to my calls?

- Kompromat — and you do not get to decide what is bad!
- Self-censorship
- Loss of business
- No privacy $\Rightarrow$ No free press $\Rightarrow$ No liberal democracy
- Security services also get you drunk, encourage you to drive, arrest you for drunken driving, and then ask you for your customer data.

The Internet is Broken

Administrators have power.

Power attracts Mexican drug cartels.

Adversary model: Mexican drug cartel

- They took your family, and will brutally kill them if you do not give them what they want.
- Under these circumstances, you must still not be able to assist, and the public system design must make that clear.
- Thus, the cartel has nothing to gain from abducting your family and will not bother with it.

System administrators are targets of such an adversary today.

Adversary model: Mexican drug cartel

- They took your family, and will brutally kill them if you do not give them what they want.
- Under these circumstances, you must still not be able to assist, and the public system design must make that clear.
- Thus, the cartel has nothing to gain from abducting your family and will not bother with it.

System administrators are targets of such an adversary today.

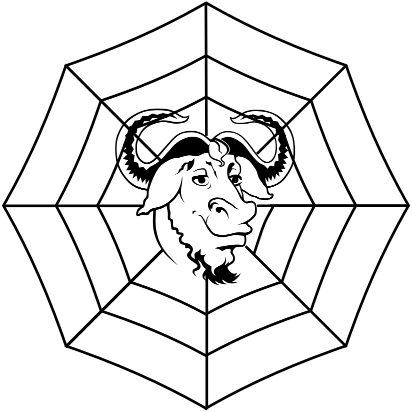
We need self-organizing networks!

The Internet is Broken by Design Choices!

Internet Design Goals (David Clark, 1988)

1. **Internet communication must continue despite loss of networks or gateways.**
2. The Internet must support multiple types of communications service.
3. The Internet architecture must accommodate a variety of networks.
4. The Internet architecture must permit *distributed management* of its resources.
5. The Internet architecture must be cost effective.
6. The Internet architecture must permit host attachment with a low level of effort.
7. **The resources used in the internet architecture must be accountable.**

Let's do something about it!



The Internet is Broken by Design Choices!

Internet Design Goals (David Clark, 1988)

1. **Internet communication must continue despite loss of networks or gateways.**
2. The Internet must support multiple types of communications service.
3. The Internet architecture must accommodate a variety of networks.
4. The Internet architecture must permit *distributed management* of its resources.
5. The Internet architecture must be cost effective.
6. The Internet architecture must permit host attachment with a low level of effort.
7. **The resources used in the internet architecture must be accountable.**

GNUnet Design Goals

1. GNUnet must be implemented as free software.
2. **The GNUnet must only disclose the minimal amount of information necessary.**
3. **The GNUnet must be decentralised and survive Byzantine failures in any position in the network.**
4. **The GNUnet must make it explicit to the user which entities must be trustworthy when establishing secured communications.**
5. **The GNUnet must use compartmentalization to protect sensitive information.**
6. The GNUnet must be open and permit new peers to join.
7. **The GNUnet must be self-organizing and not depend on administrators.**
8. The GNUnet must support a diverse range of applications and devices.
9. The GNUnet architecture must be cost effective.
10. **The GNUnet must provide incentives for peers to contribute more resources than they consume.**

Our Vision (Simplified)

Internet

Google
DNS/X.509
TCP/UDP
IP/BGP
Ethernet
Phys. Layer

Our Vision (Simplified)

Internet

Google
DNS/X.509
TCP/UDP
IP/BGP
Ethernet
Phys. Layer

HTTPS/TCP/WLAN/...

Our Vision (Simplified)

Internet

Google
DNS/X.509
TCP/UDP
IP/BGP
Ethernet
Phys. Layer

CORE (OTR)
HTTPS/TCP/WLAN/...

Our Vision (Simplified)

Internet

Google
DNS/X.509
TCP/UDP
IP/BGP
Ethernet
Phys. Layer

R^5N DHT
CORE (OTR)
HTTPS/TCP/WLAN/...

Our Vision (Simplified)

Internet

Google	
DNS/X.509	
TCP/UDP	CADET (Axolotl+SCTP)
IP/BGP	R^5N DHT
Ethernet	CORE (OTR)
Phys. Layer	HTTPS/TCP/WLAN/...

Our Vision (Simplified)

Internet

Google
DNS/X.509
TCP/UDP
IP/BGP
Ethernet
Phys. Layer

GNU Name System
CADET (Axolotl+SCTP)
R^5N DHT
CORE (OTR)
HTTPS/TCP/WLAN/...

Our Vision (Simplified)

Internet

Google
DNS/X.509
TCP/UDP
IP/BGP
Ethernet
Phys. Layer

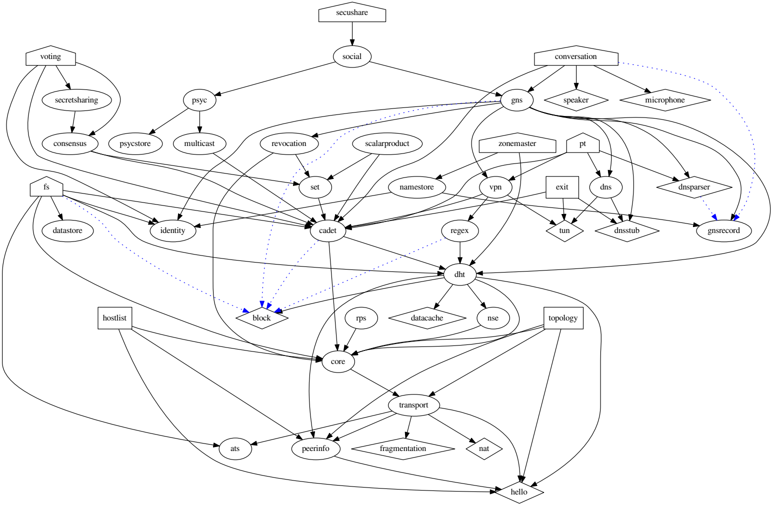
Applications
GNU Name System
CADET (Axolotl+SCTP)
R^5N DHT
CORE (OTR)
HTTPS/TCP/WLAN/...

Our Vision (Simplified)

Internet
Google
DNS/X.509
TCP/UDP
IP/BGP
Ethernet
Phys. Layer

GNUnet
Applications
GNU Name System
CADET (Axolotl+SCTP)
R^5N DHT
CORE (OTR)
HTTPS/TCP/WLAN/...

A real peer: Dependencies



Applications (being) built using GNUnet

- Anonymous and non-anonymous file-sharing
- IPv6–IPv4 protocol translator and tunnel
- GNU Name System: censorship-resistant replacement for DNS
- Conversation: secure, decentralised VoIP
- SecuShare, a social networking application
- GNU Taler: privacy-preserving payments
- ...

Summary

- This is **not** about the NSA
- Chinese, French, German, Russian agencies do the same
- This is about design goals

GNUnet is about designing network protocols to serve civil society.

Part I: The GNU Name System[1]

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

[1] Joint work with Martin Schanzenbach and Matthias Wachs

The GNU Name System (GNS)

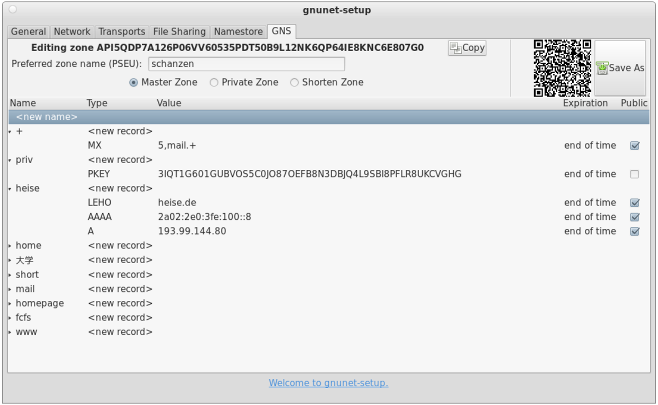
Properties of GNS

- 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

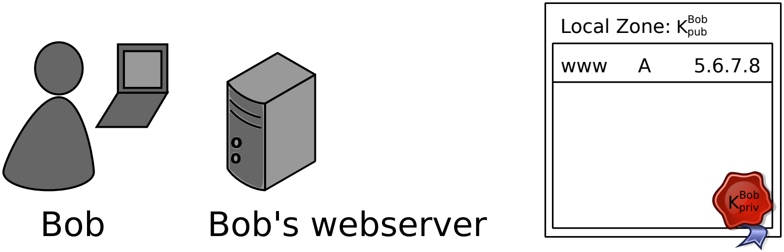
Uses for GNS in GNUnet

- Identify IP services hosted in the P2P network
- Identities in social networking applications

Zone management: like in DNS



Name resolution in GNS

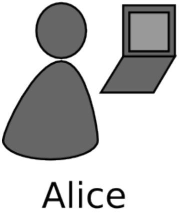


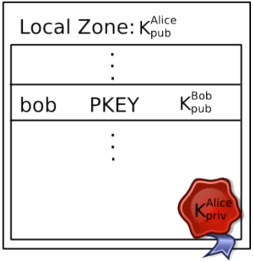
- Bob can locally reach his webserver via **www.gnu**

Secure introduction

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

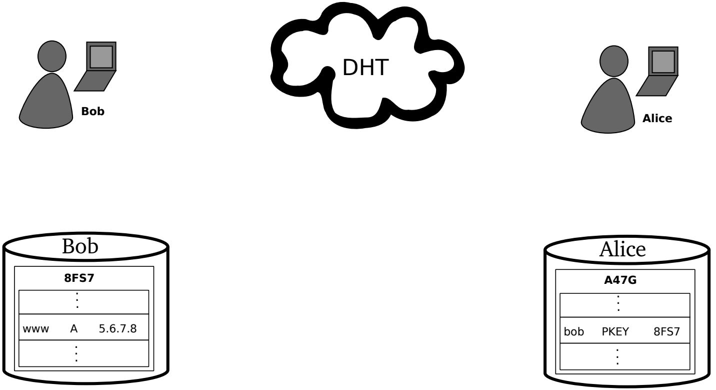
Delegation



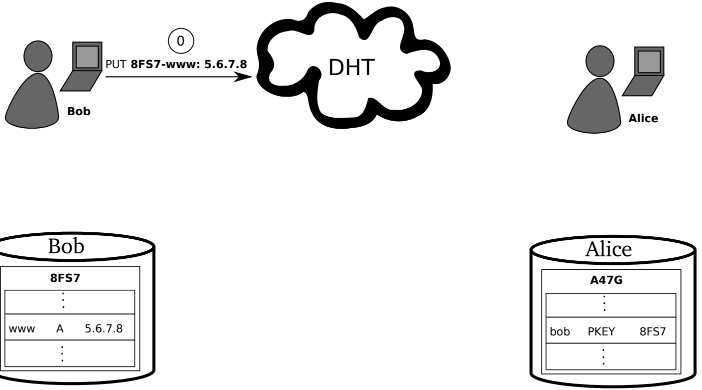


- Alice learns Bob's public key
- Alice creates delegation to zone K_{pub}^{Bob} under label **bob**
- Alice can reach Bob's webserver via **www.bob.gnu**

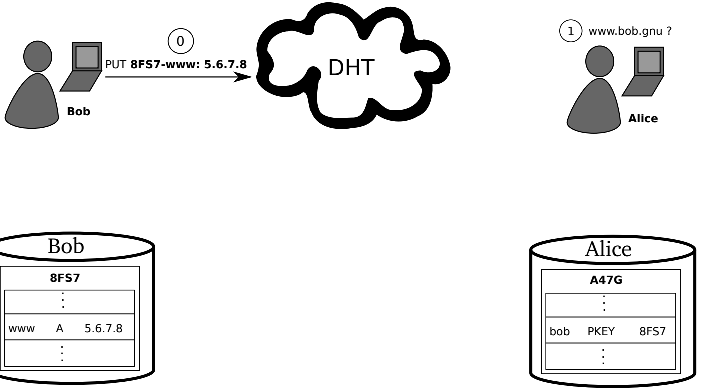
Name resolution



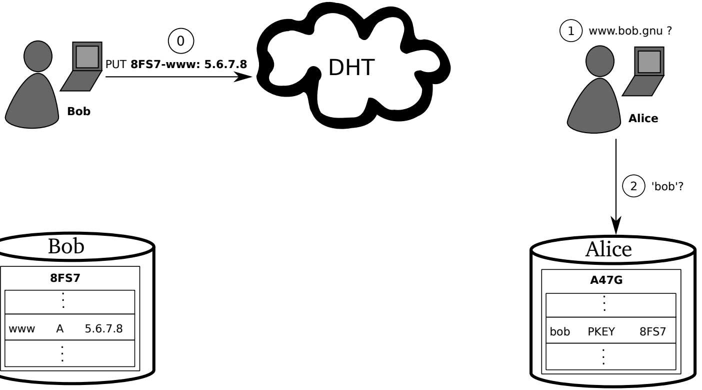
Name resolution



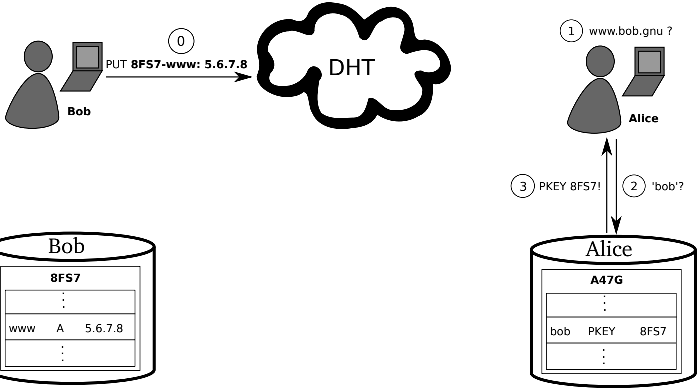
Name resolution



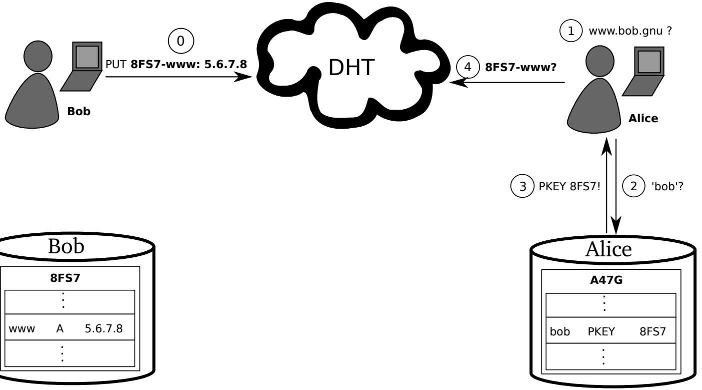
Name resolution



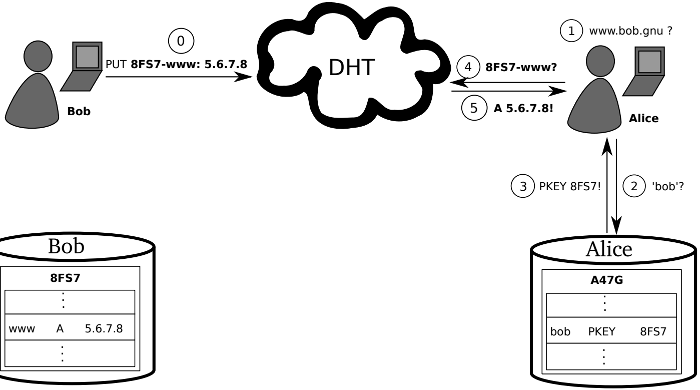
Name resolution



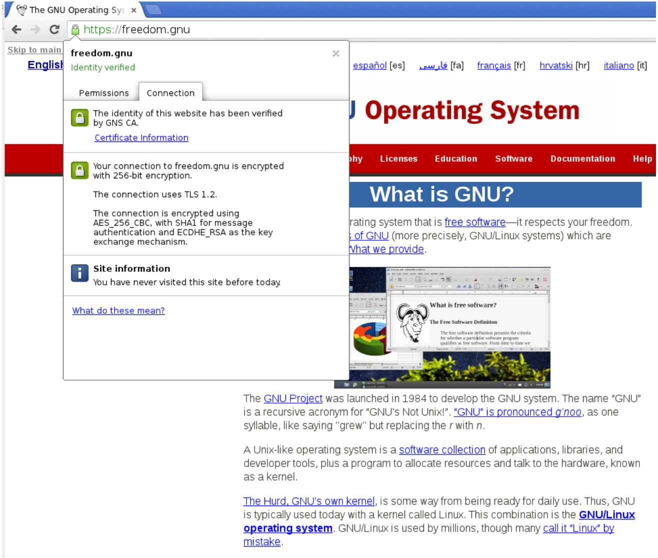
Name resolution



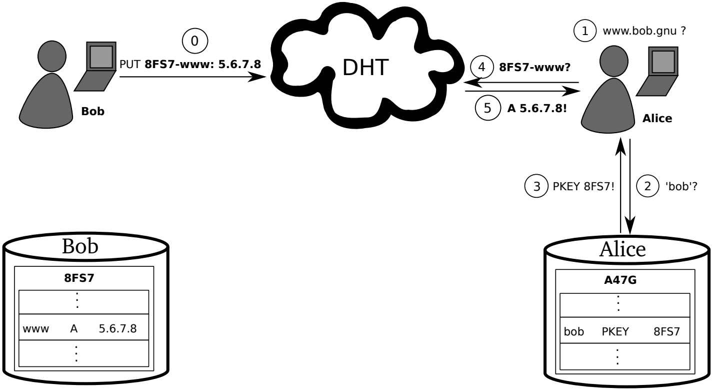
Name resolution



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 $q_{P,l}$

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

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

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

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

Summary

- Interoperable with DNS
- 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

Part II: Revisiting the Web-of-Trust[2]

"PGP assumes keys are too big and complicated to be managed by mortals, but then in practice it practically begs users to handle them anyway."

—Matthew Green

[2] Joint work with Álvaro García-Recuero and Jeffrey Burdges

Motivation

For email: differences of p≡p to other OpenPGP mail clients

- Keyservers are never used by default to prevent leakage of a peer's social graph (by signings and queries) and MITM attacks (re-encyption).
- The sender's public key is attached by default.
- The subject field gets encrypted by default (by moving it into the body).
- Instead of fingerprints, *Trustwords* (16-bit mappings of 4-digit hexablocks to words) are used.
- p≡p has a rating system and communicates (graphically) a *Privacy Status* with traffic lights semantics to the user.

The Web of Trust

Problem:

- Alice has certified many of her contacts and *flagged* some as *trusted* to check keys well.
- Bob has been certified by many of his contacts.
- Alice has **not** yet certified Bob, but wants to securely communicate with him.

The Web of Trust

Problem:

- Alice has certified many of her contacts and *flagged* some as *trusted* to check keys well.
- Bob has been certified by many of his contacts.
- Alice has **not** yet certified Bob, but wants to securely communicate with him.

Solution:

- Find paths in the certification graph from Alice to Bob.
- If sufficient number of short paths exist certifying the same key, trust it.

We will only consider paths with **one** intermediary.

The Web of Trust

Problem:

- Publishing who certified whom exposes the social graph.
- The "NSA kills based on meta data".

The Web of Trust

Problem:

- Publishing who certified whom exposes the social graph.
- The "NSA kills based on meta data".

Solution:

- Do not publish the graph.
- Have Alice and Bob collect their certificates locally.
- Use SMC protocol for

  private set intersection cardinality with signatures!

Straw-man version of protocol 1

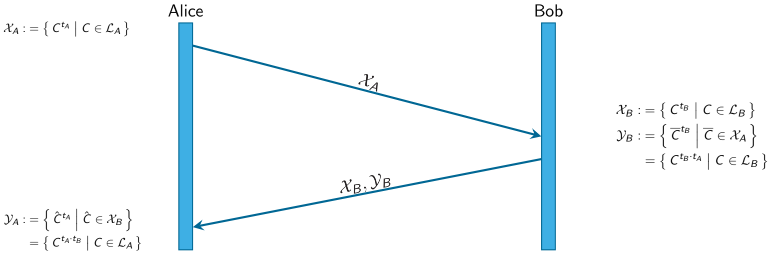
Problem: Alice wants to compute $n := |\mathcal{L}_A \cap \mathcal{L}_B|$

Suppose each user has a private key c_i and the corresponding public key is $C_i := g^{c_i}$ where g is the generator

The setup is as follows:

- $\mathcal{L}_A$: set of public keys representing Alice's subscriptions
- $\mathcal{L}_B$: set of public keys representing Bob's subscriptions
- Alice picks an ephemeral private scalar $t_A \in \mathbb{F}_p$
- Bob picks an ephemeral private scalar $t_B \in \mathbb{F}_p$

Straw-man version of protocol 1



Alice can get $|\mathcal{Y}_A \cap \mathcal{Y}_B|$ at linear cost.

Attacks against the Straw-man

If Bob controls two subscribers $C_1, C_2 \in \mathcal{L}_A$, he can:

- Detect relationship between $C_1^{t_A}$ and $C_2^{t_B}$
- Choose $K \subset \mathbb{F}_p$ and insert fakes:

$$\mathcal{X} := \bigcup_{k \in K} \left\{ C_1^k \right\}$$

$$\mathcal{Y} := \bigcup_{k \in K} \left\{ (C_1^{t_A})^k \right\}$$

so that Alice computes $n = |K|$.

Cut & choose version of protocol 1: Preliminaries

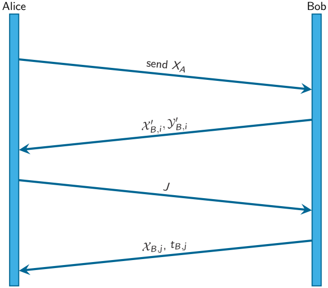
Assume a fixed system security parameter $\kappa \geq 1$.

Let Bob use secrets $t_{B,i}$ for $i \in \{1, \ldots, \kappa\}$, and let $\mathcal{X}_{B,i}$ and $\mathcal{Y}_{B,i}$ be blinded sets over the different $t_{B,i}$ as in the straw-man version.

For any list or set Z, define

$$Z' := \{h(x) | x \in Z\} \tag{8}$$

Cut & choose version of protocol 1



Protocol messages:

1. Alice sends:
   $\mathcal{X}_A :=$
   $\mathtt{sort}\,[\, C^{t_A} \mid C \in \mathcal{A}\,]$
2. Bob responds with commitments:
   $$\mathcal{X}'_{B,i}, \mathcal{Y}'_{B,i} \quad \textit{for} \;\; i \in 1, \ldots, \kappa$$
3. Alice picks a non-empty random subset $J \subseteq \{1, \ldots, \kappa\}$ and sends it to Bob.
4. Bob replies with $\mathcal{X}_{B,j}$ for $j \in J$, and $t_{B,j}$ for $j \notin J$.

Cut & choose version of protocol 1: Verification

For $j \notin J$, Alice checks the $t_{B,j}$ matches the commitment $\mathcal{Y}'_{B,j}$.

For $j \in J$, she verifies the commitment to $\mathcal{X}_{B,j}$ and computes:

$$\mathcal{Y}_{A,j} := \left\{ \hat{C}^{t_A} \;\middle|\; \hat{C} \in \mathcal{X}_{B,j} \right\} \tag{9}$$

To get the result, Alice computes:

$$n = |\mathcal{Y}'_{A,j} \cap \mathcal{Y}'_{B,j}| \tag{10}$$

Alice checks that the n values for all $j \in J$ agree.

Protocol 2: Private Set Intersection with Subscriber Signatures

- Suppose subscribers are willing to *sign* that they are subscribed.
- We still want the subscriptions to be private!
- BLS (Boneh et. al) signatures are compatible with our blinding.

$\Rightarrow$ Integrate them with our cut & choose version of the protocol.

Detailed protocol is in the paper.

Costs are linear in set size. Unlike prior work this needs no CA.

Part III: Lake[3]

[3] Joint work with Jeffrey Burdges

Asynchronous messaging

Email with GnuPG provides authenticity and confidentiality...

- ... but fails to protect meta-data
- ... and also fails to provide *forward secrecy* aka *key erasure*

Why forward secrecy?

Imagine Eve records your GnuPG encrypted emails *now*, say here:

If Eve *ever* compromises your private key in the *future*, then she can read the encrypted emails you sent *today*.

Forward secrecy

Synchronous messaging

XMPP/OtR over Tor

- Forward secrecy from OtR
- User-friendly key exchange
- Location protection (Tor)
- … but not asynchronous
- … and leaks meta-data
- … and not post-quantum

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

PWYA20120761354090000786404

SIGAD: US-984XN
PDDG: AX
CASE_NOTATION: P2BSQC110024003
DTG: 16MR1345Z12

Active User [redacted]
Active User IP Address [redacted]
Target User [redacted]
Target User IP Address [redacted]
Start Mar 16, 2012 13:40:04 GMT
Stop Mar 16, 2012 13:44:46 GMT

Other User IP Addresses
[redacted]

Time (GMT) From To Message
Mar 16, 2012 13:40:04 [redacted]
Mar 16, 2012 13:40:28
Mar 16, 2012 13:40:36
Mar 16, 2012 13:40:43
Mar 16, 2012 13:41:42
Mar 16, 2012 13:41:58 [redacted] **[OC: No decrypt available for this OTR encrypted message.]**
Mar 16, 2012 13:42:40 [redacted] **[OC: No decrypt available for this OTR encrypted message.]**
Mar 16, 2012 13:43:42 [redacted] **[OC: No decrypt available for this OTR encrypted message.]**
Mar 16, 2012 13:43:49 [redacted] **[OC: No decrypt available for this OTR encrypted message.]**
Mar 16, 2012 13:43:55 [redacted] **[OC: No decrypt available for this OTR encrypted message.]**
Mar 16, 2012 13:43:59 [redacted] **[OC: No decrypt available for this OTR encrypted message.]**
Mar 16, 2012 13:44:20 [redacted] **[OC: No decrypt available for this OTR encrypted message.]**
Mar 16, 2012 13:44:46 [redacted] **[OC: No decrypt available for this OTR encrypted message.]**

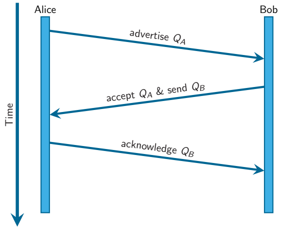
***

Why is OtR synchronous only?

We achieve *forward secrecy* through *key erasure* by negotiating an ephemeral session key using Diffie-Hellman (DH):

$$A^b = (g^a)^b = (g^b)^a = B^a \bmod p$$

$$d_A Q_B = d_A d_B G = d_B d_A G = d_B Q_A$$



Private keys:
d_A, d_B

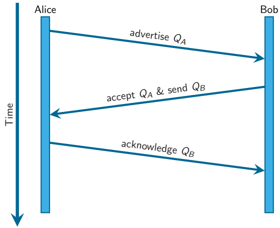
Public keys:
$Q_A = d_A G$
$Q_B = d_B G$

Why is OtR synchronous only?

We achieve *forward secrecy* through *key erasure* by negotiating an ephemeral session key using Diffie-Hellman (DH):

$$A^b = (g^a)^b = (g^b)^a = B^a \bmod p$$

$$d_A Q_B = d_A d_B G = d_B d_A G = d_B Q_A$$

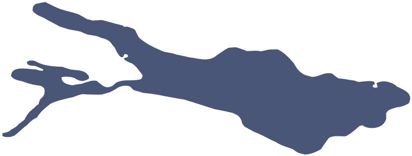


Private keys:
d_A, d_B

Public keys:
$Q_A = d_A G$
$Q_B = d_B G$

All three messages of the DH key exchange must complete before OtR can use a new ratchet key!

Project Lake[4]



[4] A lake is a big Pond.

Project Lake

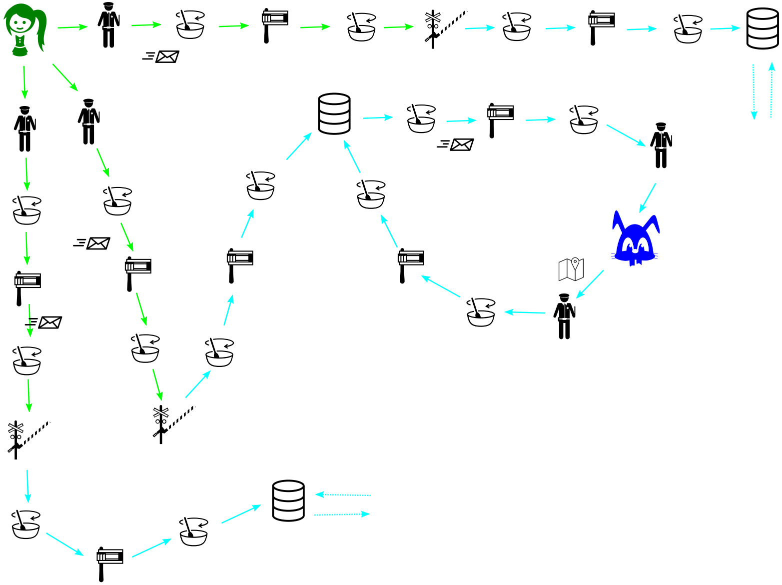
Layers:

<table>
<tr><td>MTA</td><td>IM</td></tr>
<tr><td colspan="2">p≡p</td></tr>
<tr><td colspan="2">Lake</td></tr>
<tr><td colspan="2">Xolotl</td></tr>
<tr><td>CADET</td><td>GNS</td></tr>
<tr><td colspan="2">GNUnet-CORE</td></tr>
<tr><td colspan="2">TCP/IP</td></tr>
<tr><td colspan="2">Ethernet</td></tr>
</table>

Properties:

- Endpoint **anonymity**
- Timing-attack resistance (mix, not circuit)
- No single point of failure: replicated mailbox
- Forward secrecy
- Post-quantum security
- Asynchronous delivery
- No meta-data leakage
- Off-the-record or on-the-record
- High latency

Lake Network Architecture



Asynchronous Mixing

Mixing vs. Onion Routing

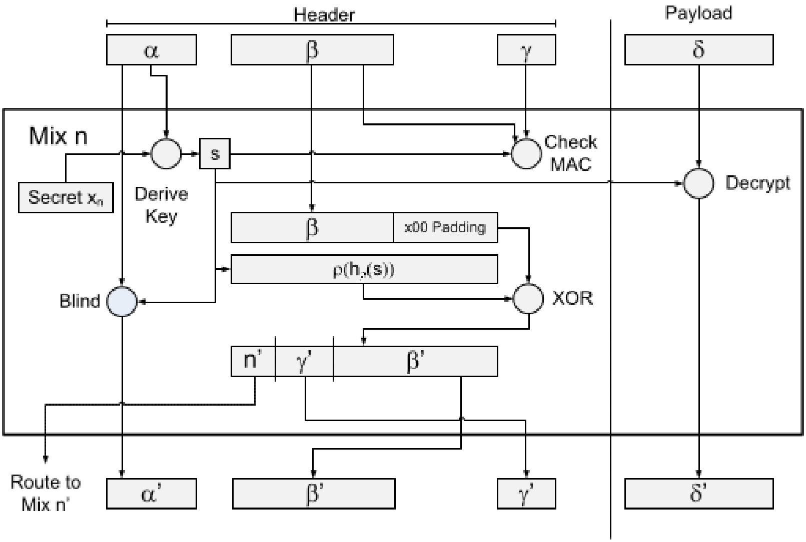
Onion routing:

- Source routing
- Circuit switching
- Low latency
- Vulnerable to timing attacks
- KX prevents replay attacks

Mixing:

- Source routing
- Packet switching
- High latency (message pool!)
- Timing attacks much harder
- Key rotation to prevent replay attacks

Sphinx by George Danezis and Ian Goldberg



The processing of a Sphinx message $((\alpha, \beta, \gamma), \delta)$ into $((\alpha', \beta', \gamma'), \delta')$

Sphinx properties

Provably secure in the universal composability model
[Camenisch & Lysyanskaya '05, Canetti '01]

1. Provides correct onion routing
2. Integrity, meaning immunity to long-path attacks
3. Security, including:
   - wrap-resistance[5]
   - indistinguishability of forward and reply messages

Replay protection implemented by Bloom filter (and key rotation).

[5] Prevents nodes from acting as decryption oracle.

Problem

Sphinx has forward secrecy only after key rotation.

- Long key lifetime:
  - Big Bloom filters to keep around to prevent replay attacks
  - Long window for key compromise
- Short key lifetime:
  - Limited delivery window after which messages are lost
  - Reduced mix effectiveness due to short time in pool
  - Loss of contact if reply addresses (SURBs) become invalid

Asynchronous Mixing with Forward Secrecy

Asynchronous Forward Secrecy with SCIMP

Idea of Silence Circle's SCIMP:

Replace key with its own hash.

Good:

New key in zero round trips.

Bad:

Once compromised, stays compromised.

Axolotl by Trevor Perrin and Moxie Marlenspike

Approach:

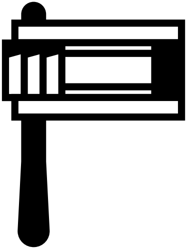
- Run DH whenever possible
- Iterate key by hashing otherwise
- Use TripleDH for authentication with deniability.

Result:

- Pseudonymous asynchronous KX
- Forward-secrecy
- Future secrecy
- Off-the-record
- Supports out-of-order messages
- Neutral against Shor's algorithm
- Formal security proof exists

```
MK      CK      RK      CK      MK
--      --      --      --      --
            ECDH(A0,B0)
                |
                |
    ECDH(A1,B0) +
               /|
              / |
             /  + ECDH(A1,B1)
     CK-A1-B0   |\
         |      | \
MK-0 ----+      |  \
         |      |   CK-A1-B1
MK-1 ----+      |       |
         |      |       +---- MK-0
MK-2 ----+      |       |
                |       +---- MK-1
    ECDH(A2,B1) +
               /|
              / |
             /  |
     CK-A2-B1   |
         |      + ECDH(A2,B2)
MK-0 ----+       \
                  \
                   \
                    CK-A2-B2
                        |
                        +---- MK-0
                        |
                        +---- MK-1
```

Xolotl ≈ Sphinx + Axolotl



Ratchet for Sphinx

Can we integrate a ratchet with Sphinx?

Axolotl does not work directly because:

- Relays never message users
- Cannot reuse curve elements

Idea:

- Users learn what messages made it eventually
- This is particularly true for replies

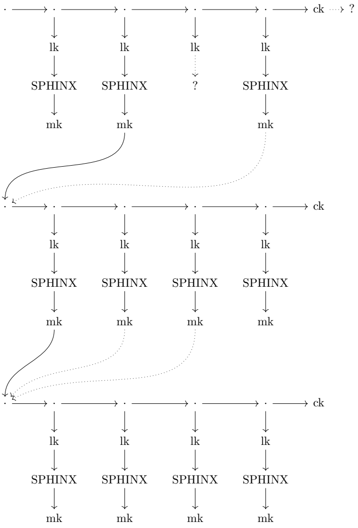
Client directs mix's ratchet state

Acknowledging ratchet state

Chain keys evolve like Axolotl, producing leaf keys.

Create message keys by hashing a leaf key with a Sphinx ECDH

$$\mathrm{mk} = H(\mathrm{lk}, H'(\mathrm{ECDH}(\mathrm{u}, \mathrm{r})))$$



Acknowledging ratchet state

Chain keys evolve like Axolotl, producing leaf keys.

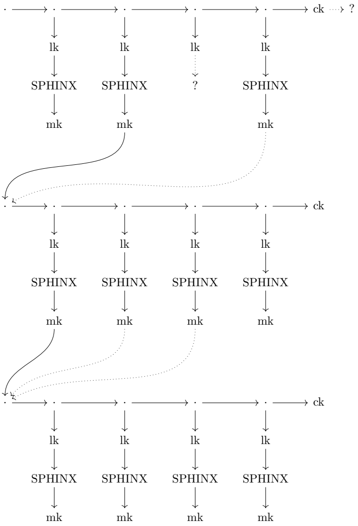
Create message keys by hashing a leaf key with a Sphinx ECDH

$$\mathrm{mk} = H(\mathrm{lk}, H'(\mathrm{ECDH}(\mathrm{u}, \mathrm{r})))$$

Packets identify the message key from which their chain started.

And their leaf key sequence no.

And parent max sequence no.



Ratchet placement

We cannot use the Xolotl ratchet for every mixnet hop:

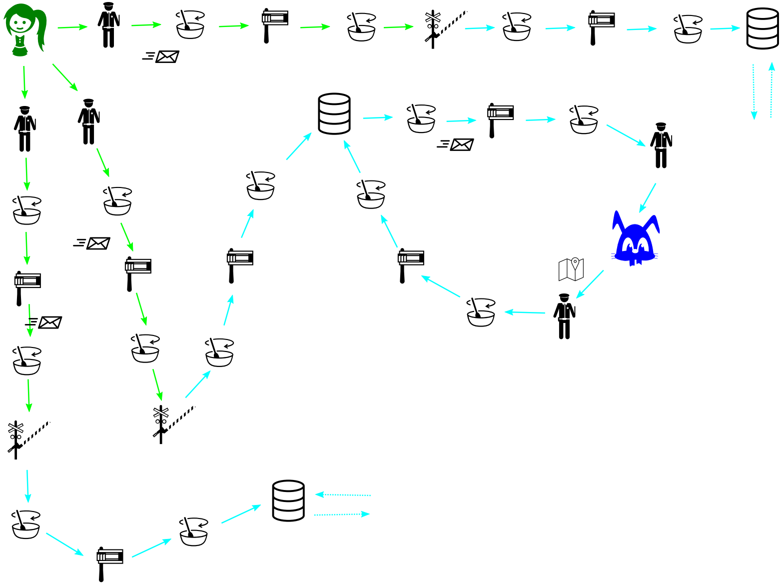
- Use of ratchet state results in pseudonymity
- Setup of post-quantum KX may be excessively expensive

Safe places:

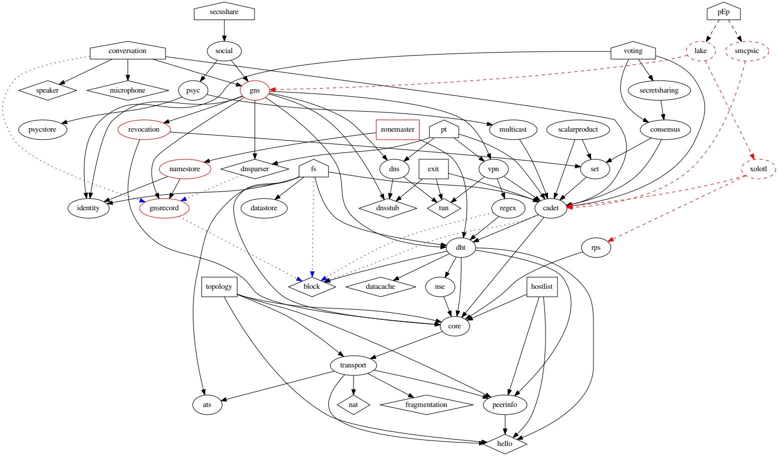
- Third hop out of a five hope circut (long-term ratchet)
- Guard node (while connection is maintained)

Other hops should use "ordinary" mix.

Lake Network Architecture



Summary


secushare
pEp
conversation
social
voting
lake
smcpsic
speaker
microphone
psyc
gns
secretsharing
psycstore
revocation
zonemaster
pt
multicast
scalarproduct
consensus
namestore
dnsparser
fs
dns
exit
vpn
set
xolotl
identity
gnsrecord
datastore
dnsstub
tun
regex
cadet
dht
rps
topology
block
datacache
nse
hostlist
core
transport
ats
nat
fragmentation
peerinfo
hello

Conclusion

There is hope!

Further reading

1. Christian Grothoff, Bart Polot and Carlo von Loesch. *The Internet is broken: Idealistic Ideas for Building a GNU Network*. **W3C/IAB Workshop on Strengthening the Internet Against Pervasive Monitoring (STRINT)**, 2014.
2. Nathan Evans and Christian Grothoff. *R^5N. Randomized Recursive Routing for Restricted-Route Networks*. **5th International Conference on Network and System Security**, 2011.
3. Matthias Wachs, Martin Schanzenbach and Christian Grothoff. *A Censorship-Resistant, Privacy-Enhancing and Fully Decentralized Name System*. **13th International Conference on Cryptology and Network Security**, 2014.
4. M. Schanzenbach *Design and Implementation of a Censorship Resistant and Fully Decentralized Name System*. **Master's Thesis (TUM)**, 2012.
5. Álvaro García-Recuero, Jeffrey Burdges and Christian Grothoff. *Privacy-Preserving Abuse Detection in Future Decentralised Online Social Networks.* **Data Privacy Management (DPM)**, pages 78–93, 2016.
6. Jeffrey Burdges and Christian Grothoff. *Xolotl-Lake*. Available in the future and in `lake.git`. 2018?