To be or I2P

An introduction into anonymous communication with I2P

Jens Kubieziel <jens@kubieziel.de>

24th Chaos Communication Congress, 2007-12-28

Overview

- (obviously) an anonymizing network
- development started in february 2003
- actual version: 0.6.1.30
- message based network
- offers end-to-end encryption

Terminology

I2P uses a special terminology describing parts of their network. These will be explained in the following slides.

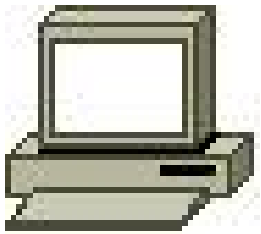
Terminology

I2P uses a special terminology describing parts of their network. These will be explained in the following slides.

- router
- tunnel
- gateway
- endpoint
- netDb

router

A router is simply the software which participates in the network.



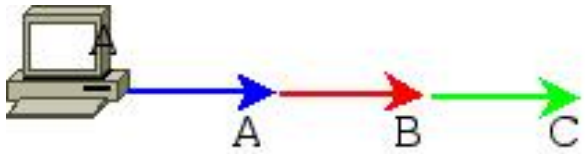
tunnel

A tunnel is a *unidirectional* path through several routers. So I2P knows about *inbound* and *outbound* tunnels.

tunnel

A tunnel is a *unidirectional* path through several routers. So I2P knows about *inbound* and *outbound* tunnels.
Every router has several incoming connections (inbound tunnels) and outgoing connections (outbound tunnels).

tunnel



A tunnel is a *unidirectional* path through several routers. So I2P knows about *inbound* and *outbound* tunnels.
Every router has several incoming connections (inbound tunnels) and outgoing connections (outbound tunnels).
Tunnels use layered encryption as we know it from other protocols.

tunnel gateway

What it is

- first router in a tunnel

tunnel gateway

What it is

- first router in a tunnel
- outbound tunnel: creator of the tunnel

tunnel gateway

What it is

- first router in a tunnel
- outbound tunnel: creator of the tunnel
- inbound tunnel: first router of the tunnel

tunnel gateway
What it does

inbound tunnel:

tunnel gateway
What it does

inbound tunnel:

- receives messages from a peer
- forwards them along the tunnel

tunnel gateway

What it does

inbound tunnel:

- receives messages from a peer
- forwards them along the tunnel

outbound tunnel:

- encodes messages
- forwards them along the tunnel

tunnel endpoint

The tunnel endpoint is

- the creator (inbound) or

tunnel endpoint

The tunnel endpoint is

- the creator (inbound) or
- the last hop of the tunnel (outbound)

tunnel endpoint

The tunnel endpoint is

- the creator (inbound) or
- the last hop of the tunnel (outbound)

tunnel endpoint

The tunnel endpoint is

- the creator (inbound) or
- the last hop of the tunnel (outbound)

The endpoint is not nessarily the endpoint of the communication.

Network Database

- called NetDb

Network Database

- called NetDb
- pair of algorithms to share network metadata

Network Database

- called NetDb
- pair of algorithms to share network metadata
- distributed hash table (Kademlia)

Network Database

- called NetDb
- pair of algorithms to share network metadata
- distributed hash table (Kademlia)
- contains two types of metadata: `leaseSet` and `routerInfo`

NetDb

routerInfo

- used for contacting another router

NetDb

routerInfo

- used for contacting another router
- contains:
  - routers identity (2048 bit ElGamal, 1024 bit DSA)
  - contact address (IP `1.2.3.4` at port `1234`)
  - when it was published (42 hours ago)
  - set of text options (used for debugging)
  - signature of the data above

NetDb

leaseSet

- contains a group of tunnel gateways,

NetDb

leaseSet

- contains a group of tunnel gateways,
- a 4 byte tunnel-ID,

NetDb

leaseSet

- contains a group of tunnel gateways,
- a 4 byte tunnel-ID,
- expiry date of that tunnel,

NetDb

leaseSet

- contains a group of tunnel gateways,
- a 4 byte tunnel-ID,
- expiry date of that tunnel,
- additional pairs of encryption and signing keys

NetDb
Bootstrapping

- decentralized database $\Rightarrow$ no data available at startup

NetDb
Bootstrapping

- decentralized database $\Rightarrow$ no data available at startup
- use of floodfill peers besides Kademlia

NetDb
Bootstrapping

- decentralized database $\Rightarrow$ no data available at startup
- use of floodfill peers besides Kademlia
- request `routerInfo` of one peer

NetDb
Bootstrapping

- decentralized database $\Rightarrow$ no data available at startup
- use of floodfill peers besides Kademlia
- request `routerInfo` of one peer
- query the router for references to other routers

Garlic routing

Traditional anonymizing protocols use onion routing, I2P uses garlic routing.

How to send messages to other routers?

Now we know about the basic terminology of I2P.

How to send messages to other routers?

Now we know about the basic terminology of I2P. But how is a message send through the network?

Requesting NetDb

When Alice wants to talk to Bob, she examines the follwing steps:

1. request the NetDb for Bob's `leaseSet`

Requesting NetDb

When Alice wants to talk to Bob, she examines the follwing steps:

1. request the NetDb for Bob's `leaseSet`
2. NetDb sends Bob's inbound gateways plus above mentioned information

Forwarding the message

1. Alice chooses an outbound tunnel

Forwarding the message

1. Alice chooses an outbound tunnel
2. tunnel gateway encrypts the message and adds forwarding instructions

Forwarding the message

1. Alice chooses an outbound tunnel
2. tunnel gateway encrypts the message and adds forwarding instructions
3. forwards it along the tunnel

Forwarding the message

1. Alice chooses an outbound tunnel
2. tunnel gateway encrypts the message and adds forwarding instructions
3. forwards it along the tunnel
4. tunnel endpoint forwards it to Bob's tunnel gateway

Forwarding the message

1. Alice chooses an outbound tunnel
2. tunnel gateway encrypts the message and adds forwarding instructions
3. forwards it along the tunnel
4. tunnel endpoint forwards it to Bob's tunnel gateway
5. gateway forwards it to Bob's router

Answering a message

- basically the same like sending
- Alice sends her destination in her message

Threat model

- replay attack

Threat model

- replay attack
- tagging attack

Threat model

- replay attack
- tagging attack
- sybil attack

Threat model

- replay attack
- tagging attack
- sybil attack
- flooding attack

Applications

- E-Mail
- IRC
- eepsites
- filesharing