

Darkening the heart of Phantom

# Overview

- Introduction
- Phantom Protocol
- Architecture
- Completed work
- Future work

# Introduction



Why care?

- whistle-blowers
- modern technology
- oppressive gov.
- youtube comments

What about bad guys!?

- nothing stopping them
- leveling the playing field

# Existing options

- Tor
- I2P
- experimental

# Tor Stats

- 1500 relay/exit nodes
- > 100,000 of users
- 126 countries
  - most used in Germany
- HTTP and BT

# Tor problems

- Designed for low latency
- Centralized
- Exit node

# Phantom Protocol

- new anon. protocol to combat the problems seen now

# Assumptions

- all traffic is eaves dropped on
- any node could be compromised
- no central authority
  - none!

# Goals

- Complete decentralization
- DoS resistance
- Theoretically secure
- end to end encryption
- Isolation from the Internet
- Protocol identification should be hard

# Goals

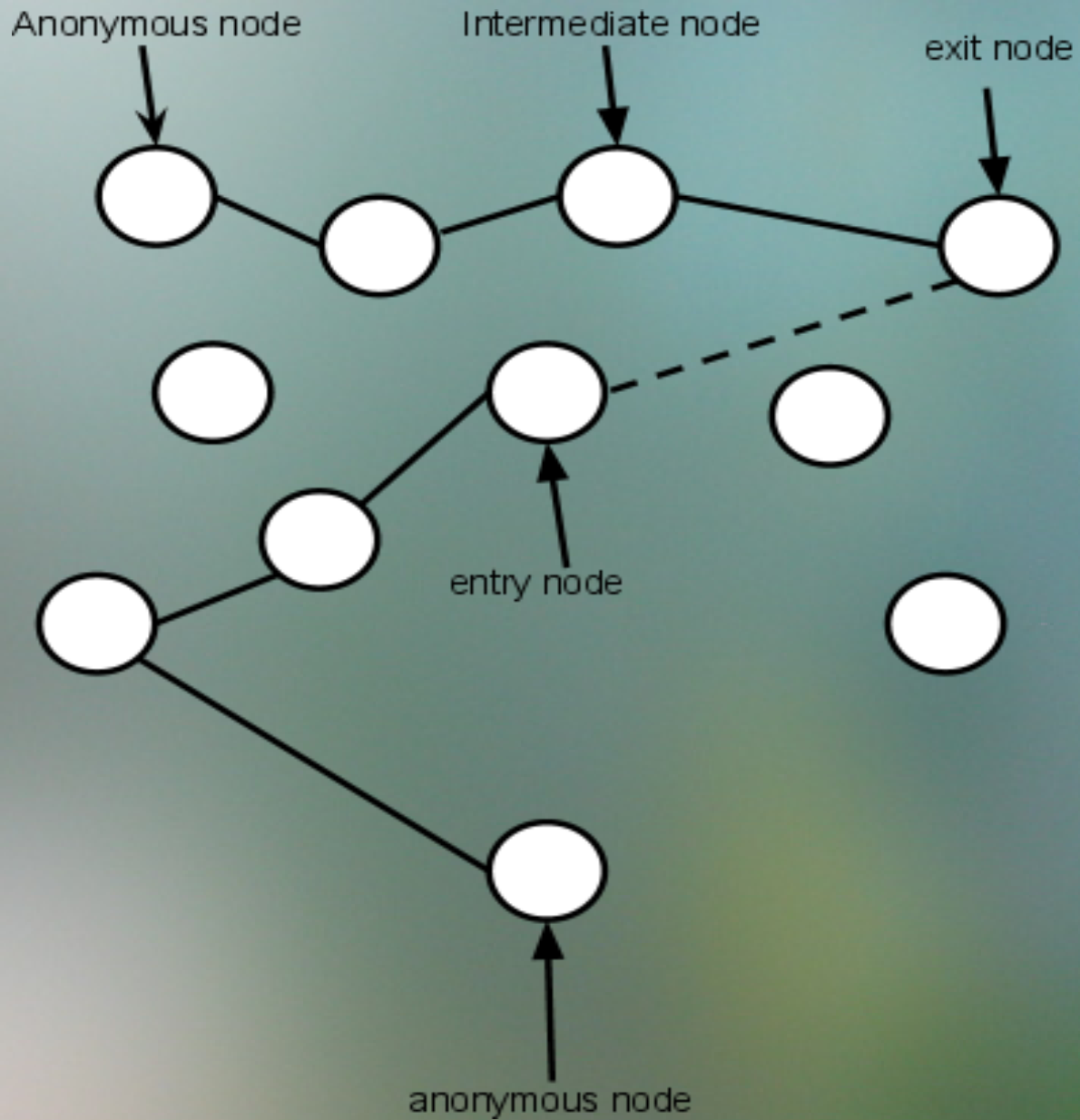
- High throughput
- Well abstracted

Additionally...

- Choose strong security over performance

# Big picture

- Nodes
  - Anonymous
  - Intermediate
  - exit
  - entry
- Paths
- Anonymous protocol (AP) address
- Tunnels



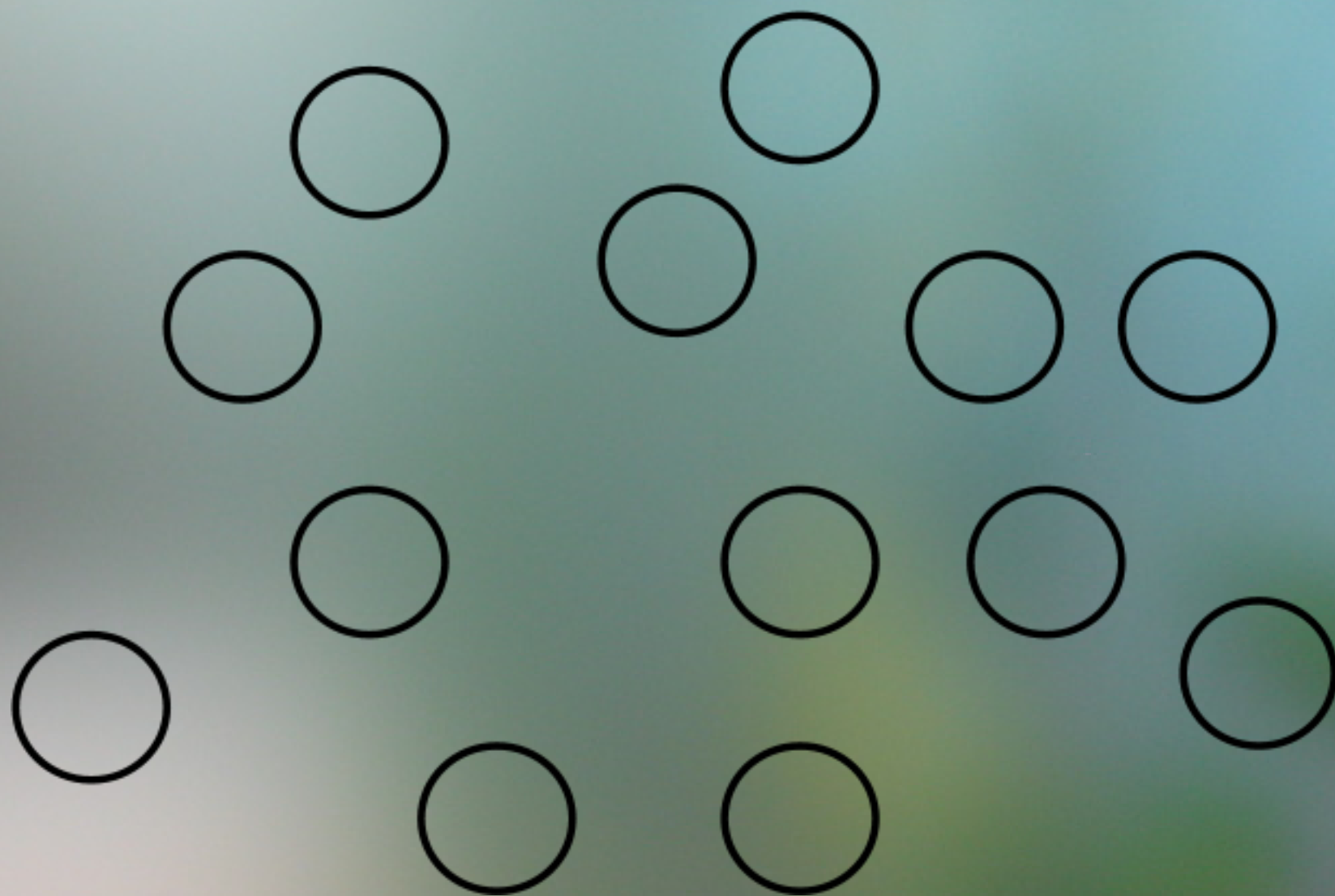
Terms illustrated

# Path creation

- Paths are not end to end
  - exit path
  - entry path
  - 2 paths needed for end to end
- Path is composed of nodes
  - X nodes
  - Y nodes

# Node placement

- Y nodes go between X nodes
  - buffer
- Y nodes are temporary
- X nodes are the final path
- Terminating X node



Process



# Advantages

- no central authority
- decisions to make about the path properties

# Tunnel creation

- anon. node sends a message to the exit node
- exit node sends a unique message back
- anon. node brute-forces keys
- sends information
- entry node does nearly the same

# Architecture

- details details details

# Design Decisions

- POSIX.1-2001 standard
- IPv6
- OpenSSL
- AES256
- RSA
- SSL
- SHA-1

# Design Decisions

- protocol buffers
- One server port
- TCP
  - not UDP
- Kademlia

# The DHT

- Kademlia with modifications
- Applies order and structure on the network
- Used to store contact information

# Kademlia useage

- Used by BitTorrent
- Overnet
- experimental protocols

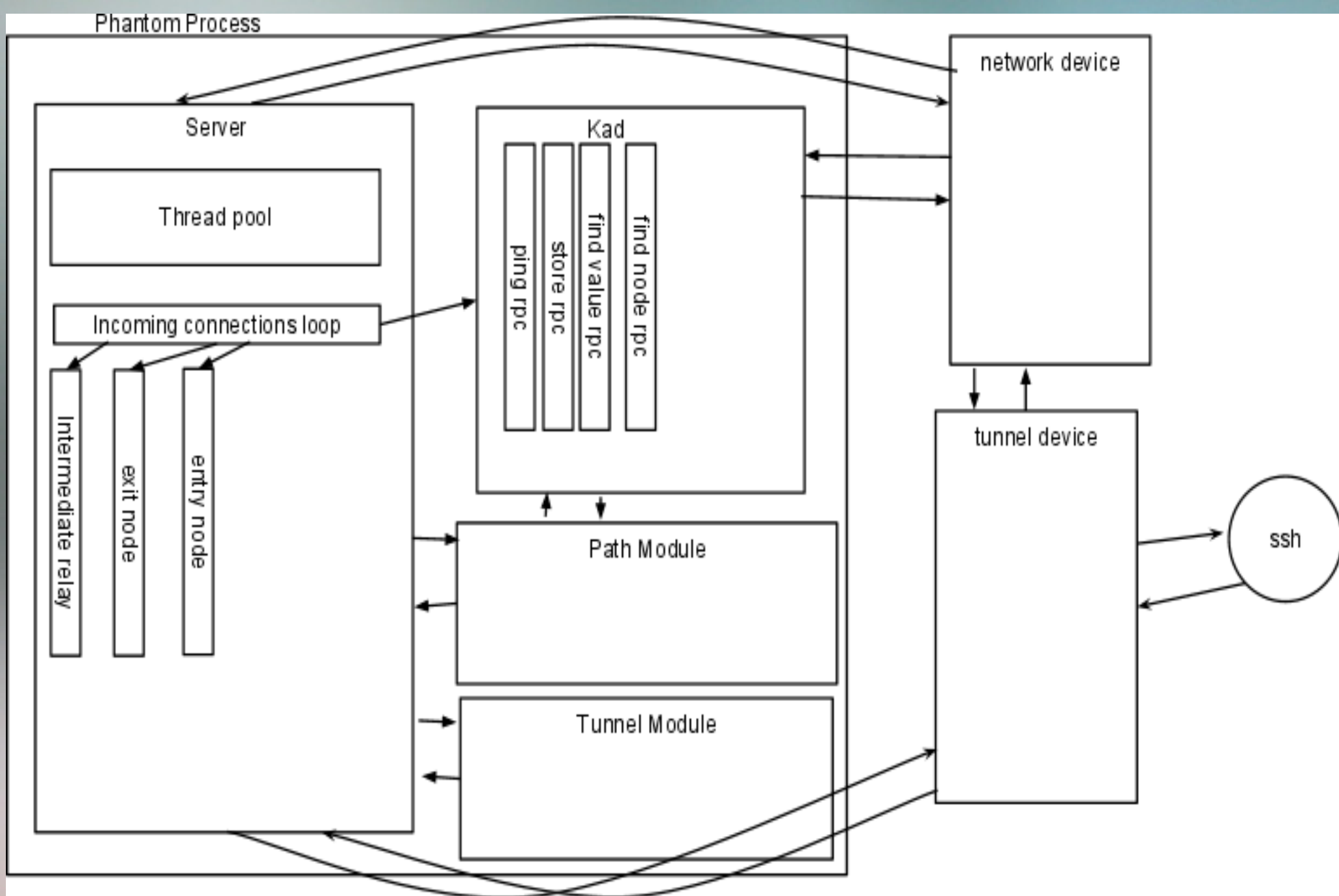
# Kademlia details

- All DHT traffic happens over SSL
- IDs are SHA-1 hashes
  - 160 bits and 160 buckets
- Buckets store 20 IDs

# Housekeeping

- bucket refresh: 3600 sec
- republish values: 3600 sec
- republish AP info: 86400 sec
- expire time: 86400 seconds





# Basic Architecture

## Completed work

- Testing architecture
- Path building changes
- DHT changes
- Security

# Testing architecture

- Testing Phantom is challenging
- Cloud architecture
- Eucalyptus
  - could be ported to AWS
- 20 nodes minimum
- built and configured with python and boto

# Scripting

- iPython
- tmux
- can do almost anything with the nodes and a python script

# Path building revisited

- AP routing entries not being republished
- signing the entries was necessary

# Publishing an AP

- choose AP
- add counter
- build path
- pack information
- sign it
- publish it anonymously

# Republish

- add to counter
- build path
- pack information
- sign it
- publish it anonymously

# Morphed into generic anonymous kademia RPCs

- added flags
- done on exit node
- communication happens through tunnel

# Key store problems

- possible process collisions
- insecure file operations
- predicable key names

# Key store update

- usage of tempfile()
  - not portable
  - POSIX
  - higher level I/O issues
- ID kept in memory
- no intra process interference
  - important for testing

# Housekeeping

- republishing possible -> housekeeping possible

# Bugs

- Fixed kademlia list sorting
- tunnels creation unreliabl

# Security

- Why the anon. kademlia RPC?
- DHT is subject to passive information leaks

# Attack scenario

- Attacker wants to know if address is being accessed
- Finds a certificate close to the hash of the interesting AP
- Answers kademlia find value requests
- records traffic

## Future work

- attack scenario is a symptom
- security is on going challenge

# Keeping the DHT secure

- structure and organization causes problems
- we need structure for efficient look ups
- attacker can find out our relay nodes
- how do we get the nodes anonymously then?



The problem

# Possible solutions

- Possible hybrid between structured and unstructured lookups
- Why am I speculating?
  - assuming 20% of nodes are compromised
  - tough problem
  - many attempts and no success thus far

# Other security things to consider

- Traffic analysis
- DoS
- Sybil

# Conclusions

- Dark Anonymous future
- A counterweight is needed to give people anonymity back

Thank you.