

Generic, Decentralized, Unstoppable Anonymity:

The Phantom Protocol

DEFCON 16 Presentation
Magnus Bråding 2008



Short Author Presentation

- Magnus Bråding
  - Swedish security researcher (Fortego Security)
  - 10+ years in the security business
  - Central contributor and driving force behind woodmann.com reverse engineering community

Project Background

(why is this interesting?)

- Big upswing in anti online privacy measures during last couple of years
  - Huge pressure from media companies
  - ISPs tracking and throttling arbitrary traffic
  - Data retention laws
  - Draconian laws for tracking and punishing P2P users
  - Abuse and misuse of global network blacklists, under the cover of being ”child porn” related, while in reality being much more arbitrary censorship
  - Recent EU law proposal to register, track and regulate all bloggers!
  - Dictatorships and other regimes with oppressed people censoring and tracking Internet use on an increasingly larger scale
- A huge upcoming demand for anonymity seems unavoidable!
- Existing anonymization solutions are in many ways not well suited for this upcoming demand and the circumstances surrounding it
- There is no real “standard” for anonymization, like BitTorrent is for P2P
- A perfect opportunity to get it right with a new solution, from the start!

Goals of the Project

- To be a good reference for future work within the field of anonymization
- To inspire further discussion about the optimal requirements for the future anonymization demand
- To be a starting point and inspiration for the design and development of a global de facto standard for generic anonymization
- *Not* to be a complete detailed specification ready to be implemented, but rather to be built upon

Limitations

- The protocol is designed to work in any network environment as long as no single attacker is able to eavesdrop *all* participating nodes in a correlated fashion, or directly controls a large majority of all nodes in the network
  - Such an attacker will still never be able to see *what* participating nodes are talking about though, only *who* they are communicating with
  - The protocol also contains built-in countermeasures to protect against attackers that are only able to monitor *parts* of the network

Further Assumptions and Directives

- Arbitrary random peers in the network are assumed to be compromised and/or adverse
- CPU power, network bandwidth, working memory and secondary storage resources are all relatively cheap, and will all be available in ever increasing quantity during coming years and thereafter
  - Thus, wherever a choice must be made between better security or better performance / lower resource consumption, the most secure alternative should be chosen (within reasonable bounds, of course)

Design Goals

Design Goal Overview

- Very important with well thought-out design goals, this is at least half the work in any successful project!
- The design goals are stipulated with the requirements and demand of today and the future in mind

Design Goal Overview

- Eight primary design goals:
  1. Complete decentralization
  2. Maximum DoS resistance
  3. Theoretically secure anonymization
  4. Theoretically secure end-to-end encryption
  5. Complete isolation from the ”normal” Internet
  6. Protection against protocol identification
  7. High Traffic Volume and Throughput Capability
  8. Generic, Well-Abstracted and Backward Compatible

Design Goal #1: Complete Decentralization

- No central or weak points can exist
- They *will* be targeted
  - Legally
  - Technically (DoS attacks, takedowns etc)
- Both ownership and technical design must be decentralized
  - Open/community owned design & source code

Design Goal #2: Maximum DoS Resistance

- The only way to stop a decentralized system without any legal owners is to DoS it
- It only takes one weakness, so defensive thinking must be applied throughout all levels of the design

Design Goal #3: Theoretically Secure Anonymization

- Nothing should be left to chance
- No security by obscurity
- All anonymization aspects should be able to be expressed as a risk probability or a theoretical (cryptographic) proof

Design Goal #4:
Theoretically Secure End-to-End Encryption

- Confidentiality is not only important by itself, but also directly important to anonymity!
  - Eavesdropped communication is highly likely to contain information of more or less identifying nature at some point!
- Even if someone would monitor and correlate all traffic at all points in the entire network, they should not be able to see *what* is communicated, no matter what

Design Goal #5: Isolation from the "Normal" Internet

- Users should not have to worry about Internet crimes being perpetrated from their own IP address
- An isolated network is necessary to be able to enforce end-to-end encryption for generic traffic
- Using an isolated network has many advantages, but not so many disadvantages in the end
- Out-proxies to the ”normal” Internet can still be implemented on the application level, selectively

Design Goal #6:
Protection against Protocol Identification

- Many powerful interests will lobby against a protocol like this, both to lawmakers and ISPs (who are already today filtering traffic)
- The harder it is made to positively identify the usage of the protocol, the harder it will be to track, filter and throttle it

Design Goal #7: High Volume / Throughput Capacity

- The traffic volume for ”normal usage” of the Internet increases every day
- More or less high speed / throughput is necessary for many Internet applications
- Popularity will be proportionally related to transfer speed and volume
- Anonymity is directly related to popularity

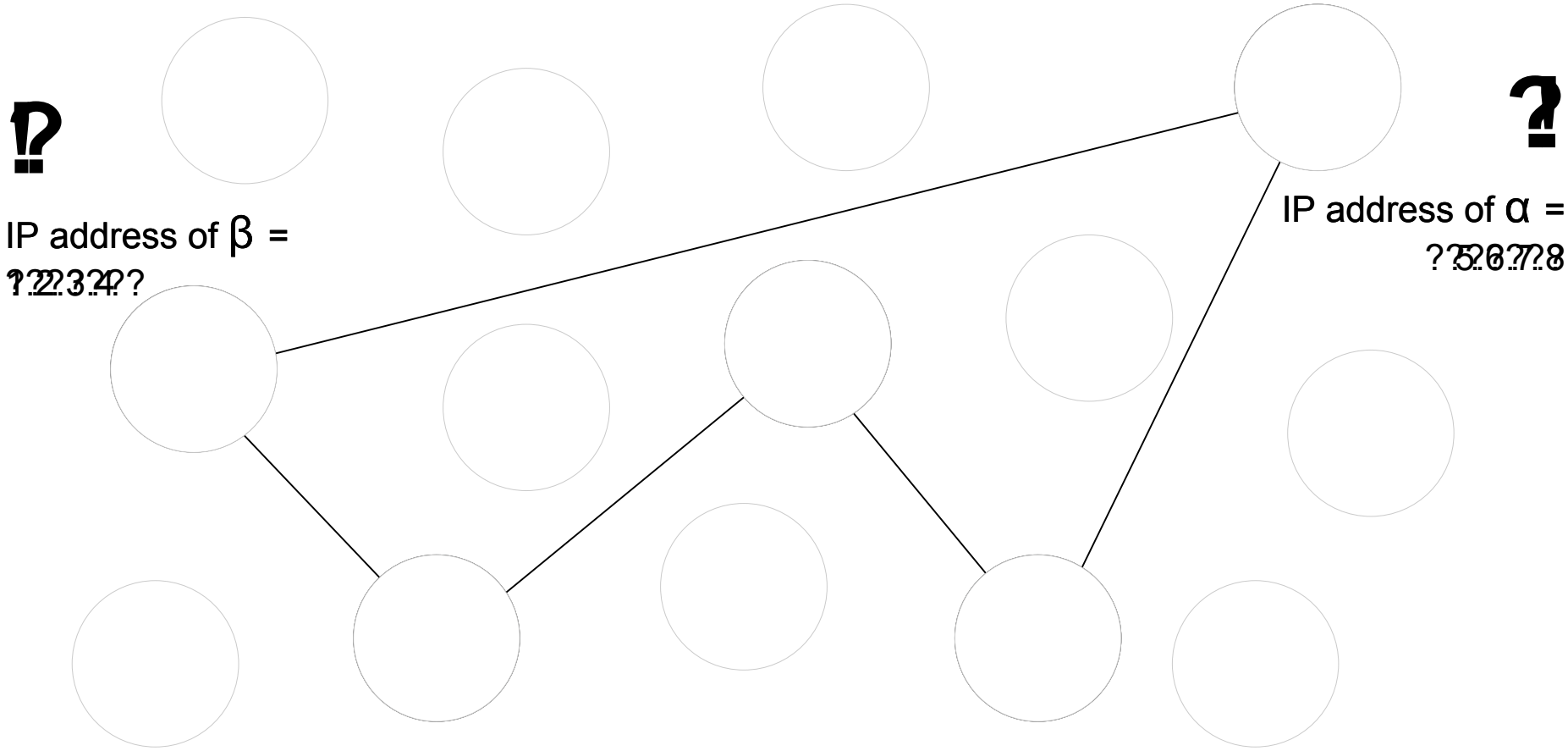
Design Goal #8:

Generic, Well-Abstracted and Backward Compatible

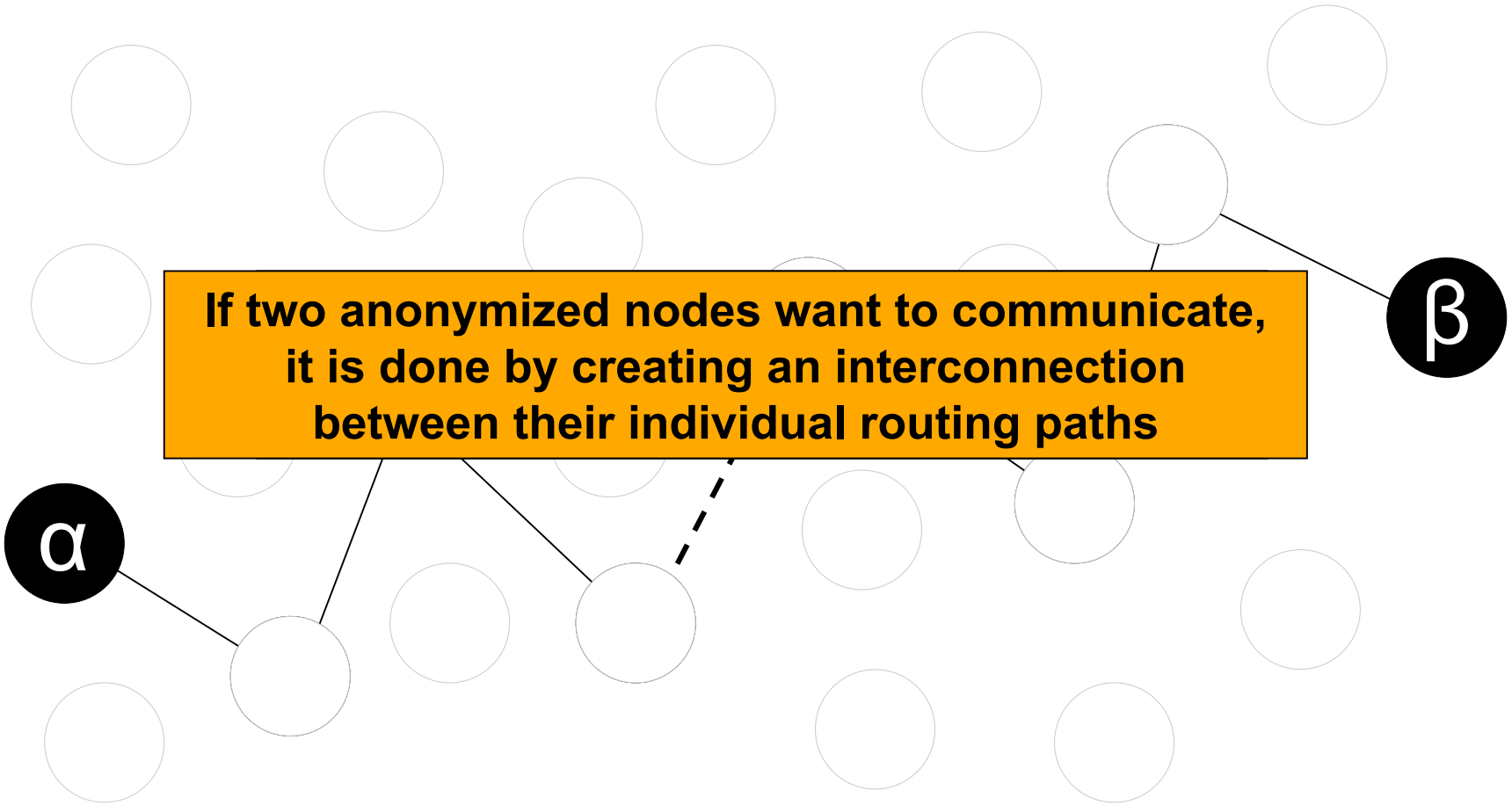
- A generic system is practically always superior to a specific system in the long run
- A well-abstracted system allows for efficient, distributed design and implementation
- A system compatible with all pre-existing network enabled applications will get a much quicker takeoff and community penetration, and will have *much* larger

A Bird's-Eye View

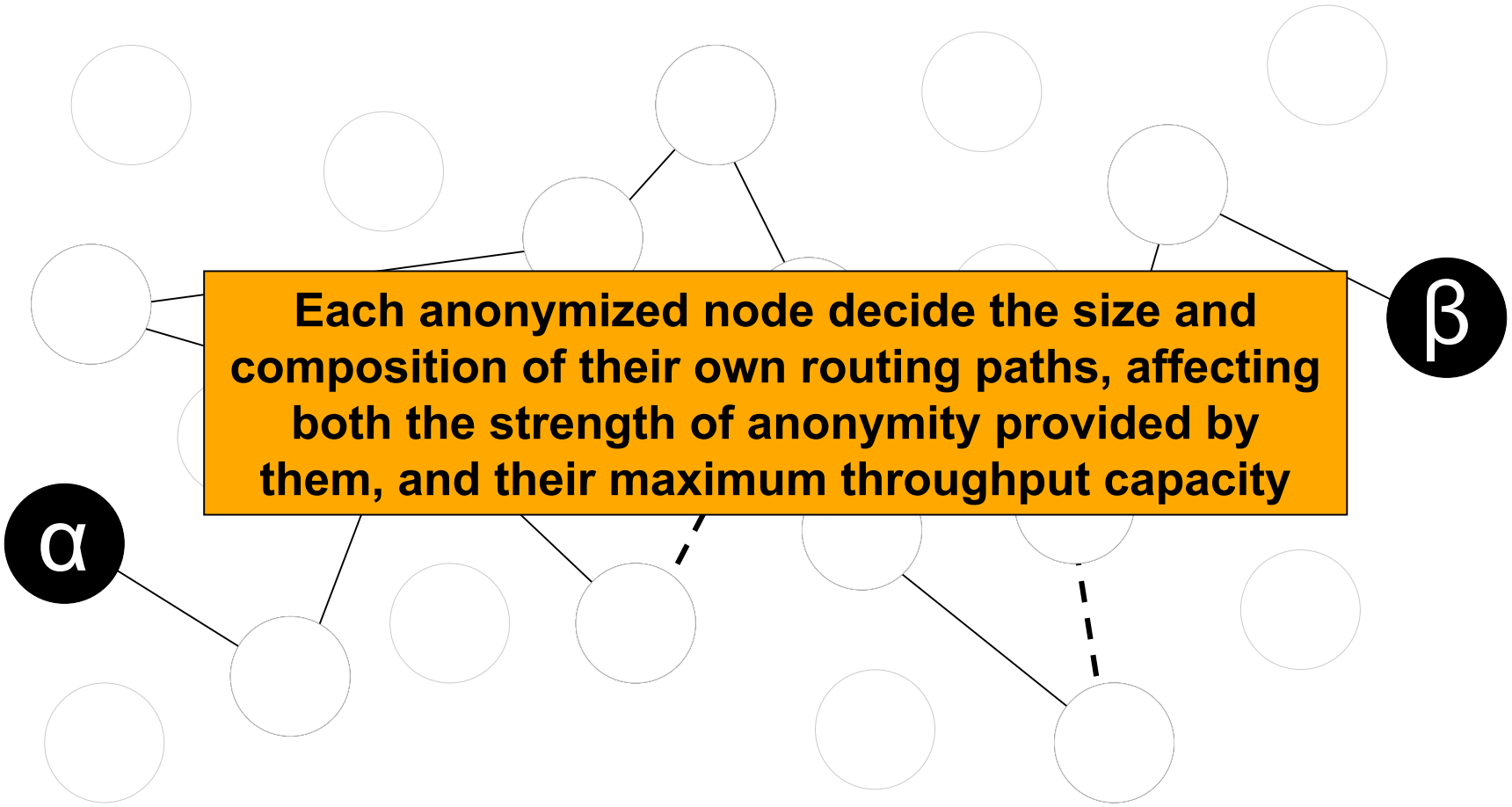
The Basic Idea



More About the Idea



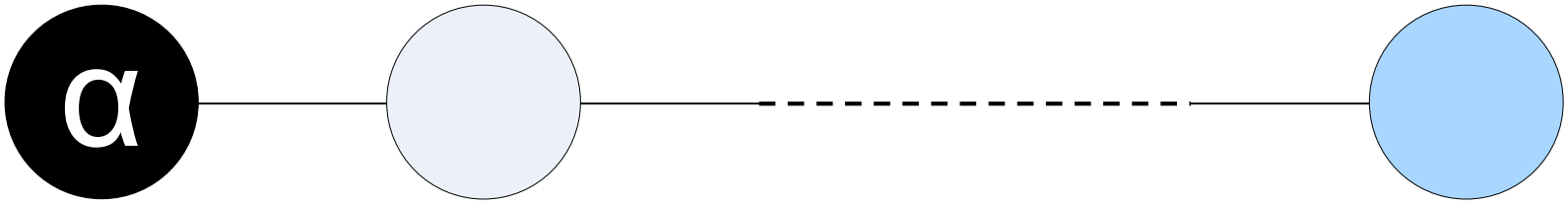
Routing Paths



High Level Design



Routing Path - Generalization



Anonymized node

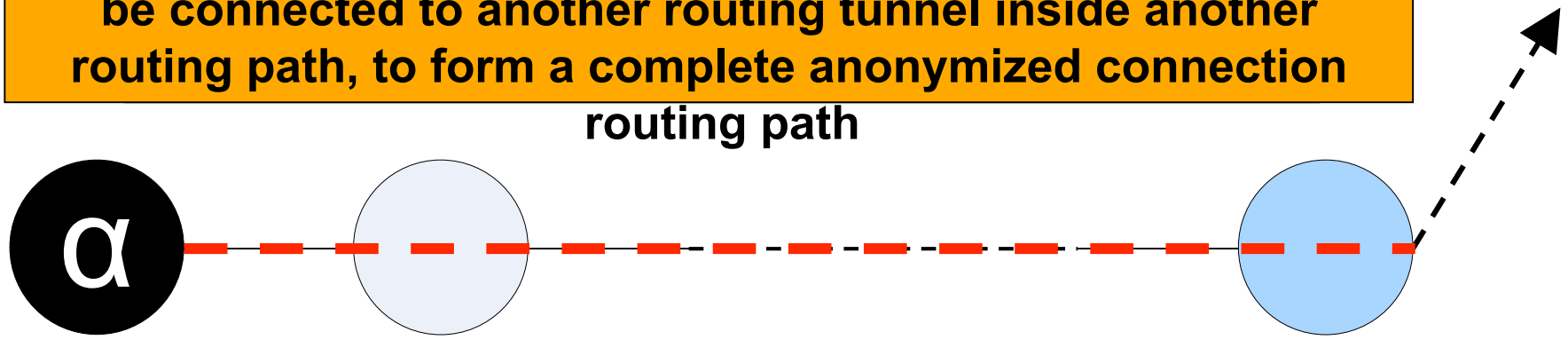
Intermediate node

Arbitrarily many more intermediate nodes

Terminating intermediate node

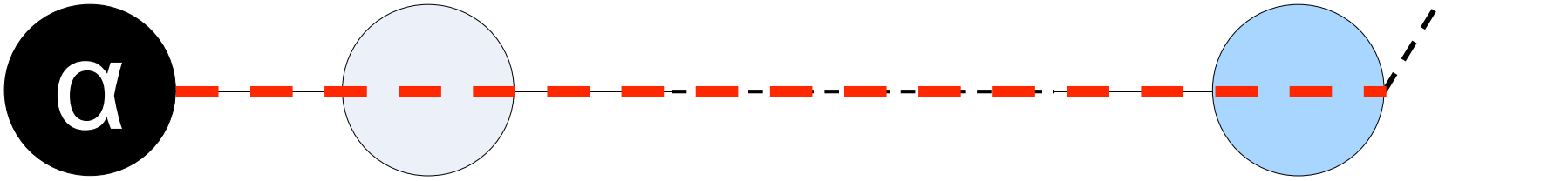
Routing Tunnels

Such a routing tunnel is set up relatively quick, and will then be connected to another routing tunnel inside another routing path, to form a complete anonymized connection



Routing Tunnels

Such a routing tunnel is set up relatively quick, and will then be connected to another routing tunnel inside another routing path, to form a complete anonymized connection



Anonymized node

Intermediate node

Arbitrarily many more intermediate nodes

Terminating intermediate node

AP Addresses

- ”Anonymous Protocol” addresses
- Equivalent to IP addresses in their format
- Equivalent to IP addresses in functionality, with the exception that they allow communication between two peers without automatically revealing their identity
- Backward compatible with IP applications

The Network Database

- Equivalent to the routing tables of the ”normal” Internet
- Distributed and decentralized database based on DHT (Distributed Hash Table) technology
  - Proven technology
  - Automatic resilience to constantly disappearing and newly joining nodes
  - Automatic resilience to malicious nodes of some kinds
- The network nodes *are* the database

Design Details

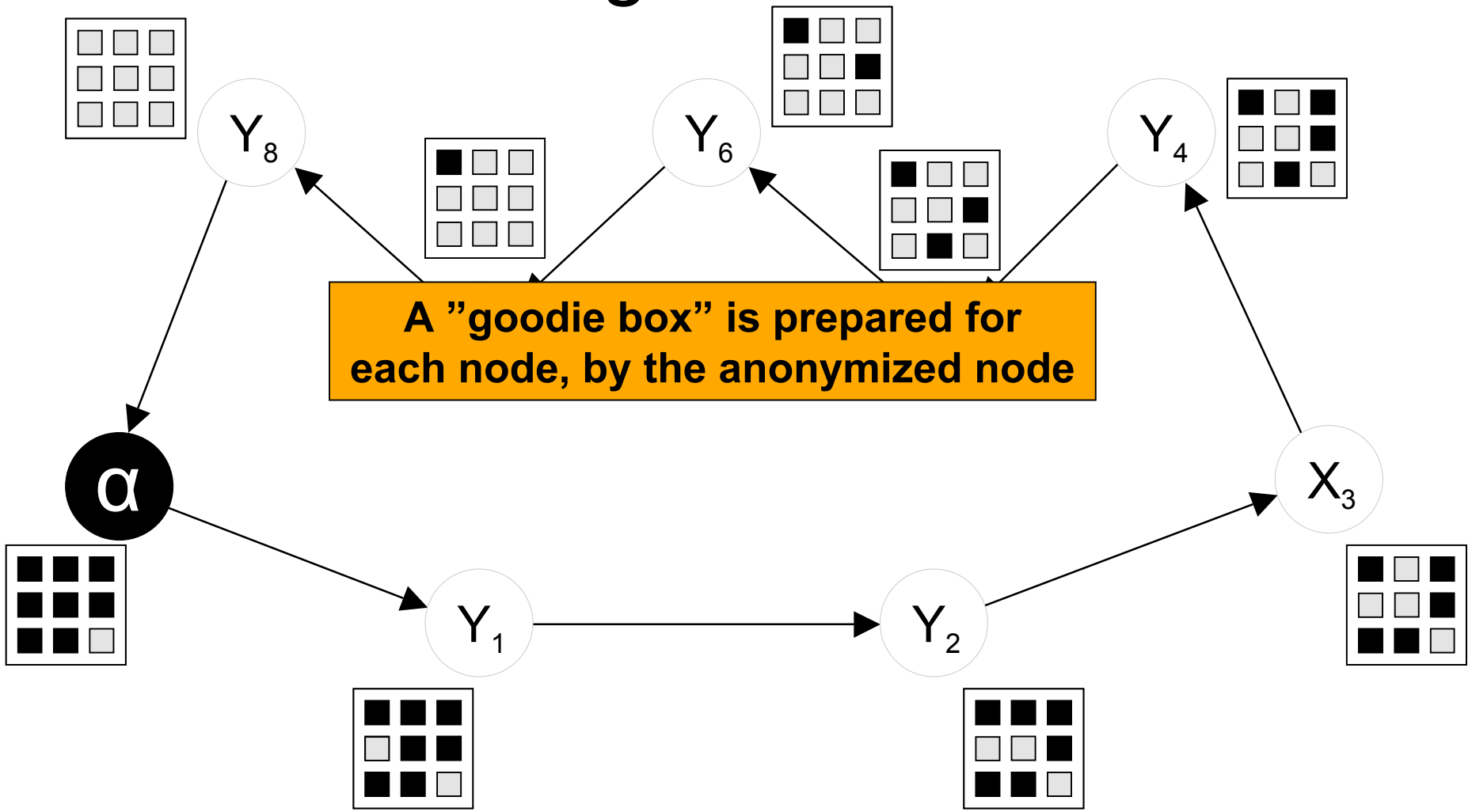
Secure Routing Path Establishment

The selection of the order of nodes in the sequence must obey the following rules:

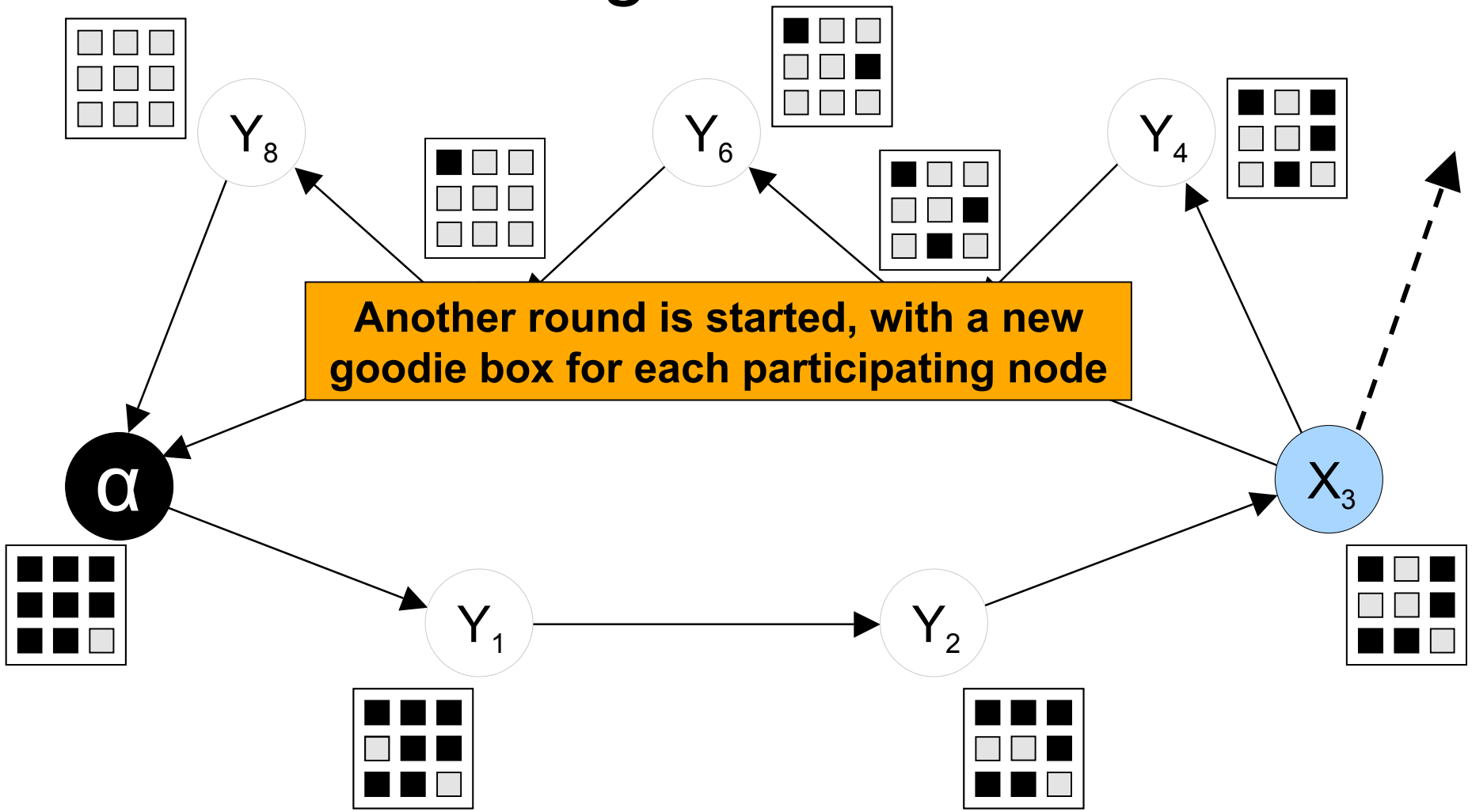
- **No two X-nodes can be adjacent to each other**
- **A Y-node should be located in one end of the sequence**
- **A number of Y-nodes equal to the total number of X-nodes minus one, should be located adjacent to each other in the other end of the sequence**
- **One end of the sequence should be chosen at random to be the beginning of the sequence**

Y_8

Secure Routing Path Establishment



Secure Routing Path Establishment



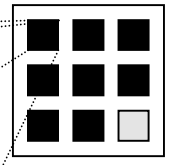
Secure Routing Path Establishment



The routing path is now securely established!

Repeat

The Goodie Box

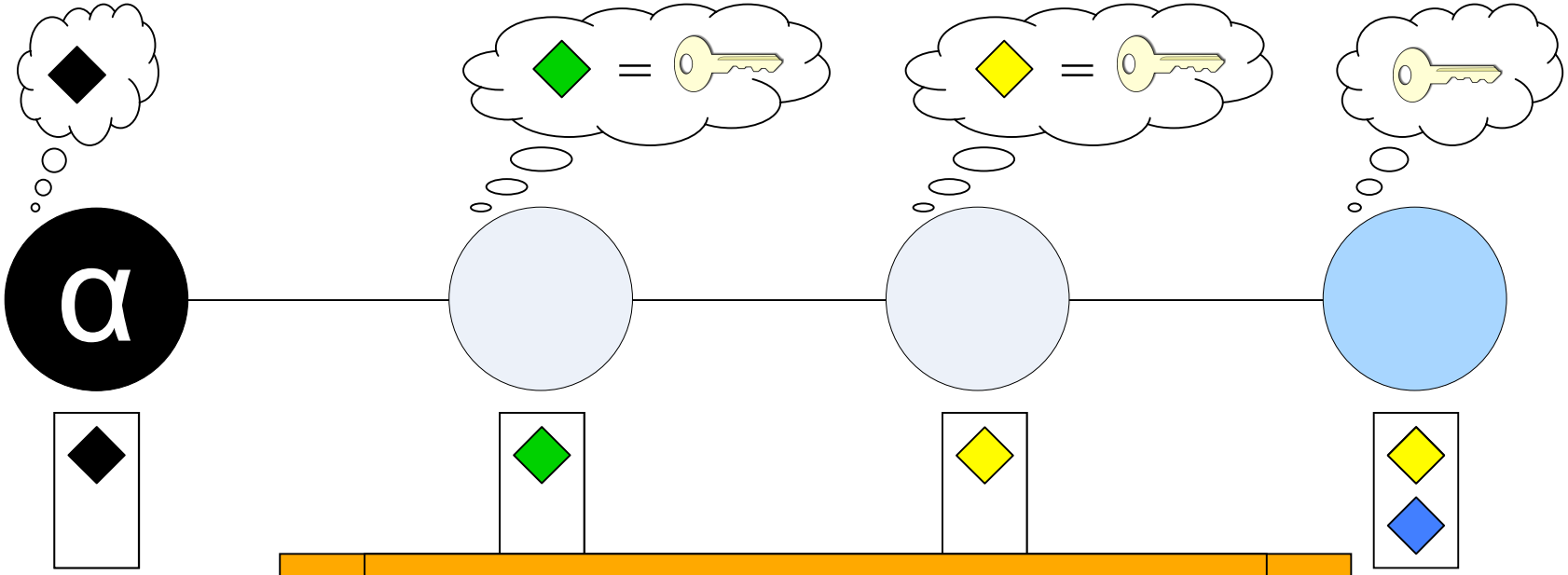


- The routing path construction certificate
- IP address and port number of next/previous nodes
- Random IDs of next/previous node connections
- Communication certificate of next/previous nodes
- Seeds and params for dummy package creation
- Seeds and params for stream encryption keys
- Flags
- A secure hash of the entire (encrypted) setup package array in currently expected state
- A secure cryptographic hash of the (decrypted) contents of the current setup package

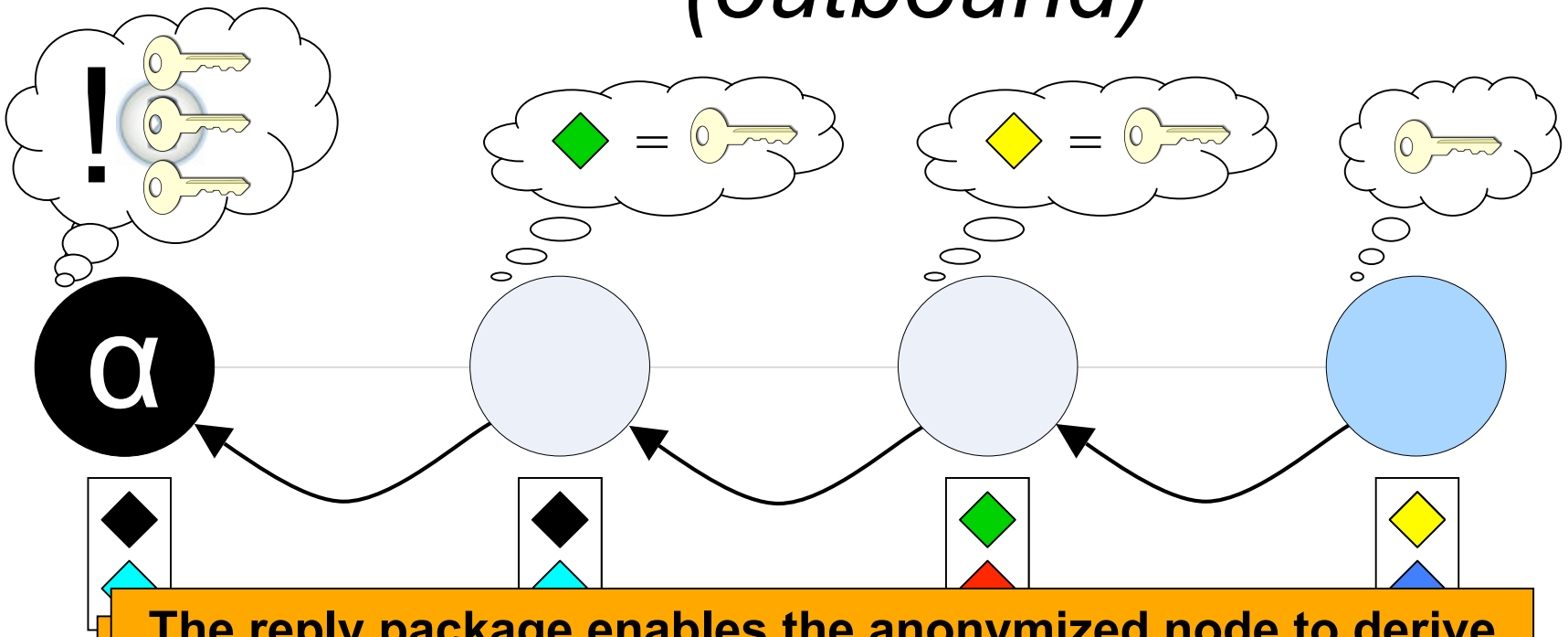
Second round extras:

- A signed routing table entry, for the AP address associated with the routing path

Secure Routing Tunnel Establishment (outbound)

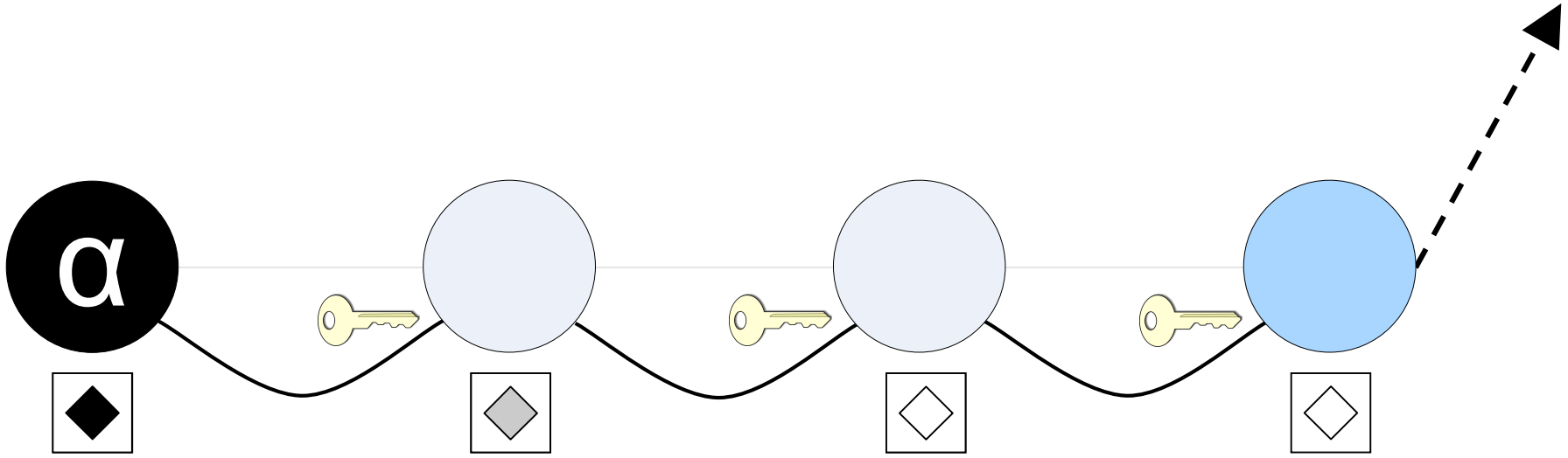


Secure Routing Tunnel Establishment (outbound)



The reply package enables the anonymized node to derive the keys of all the intermediary nodes, while it is impossible for any of them to derive any key with it themselves

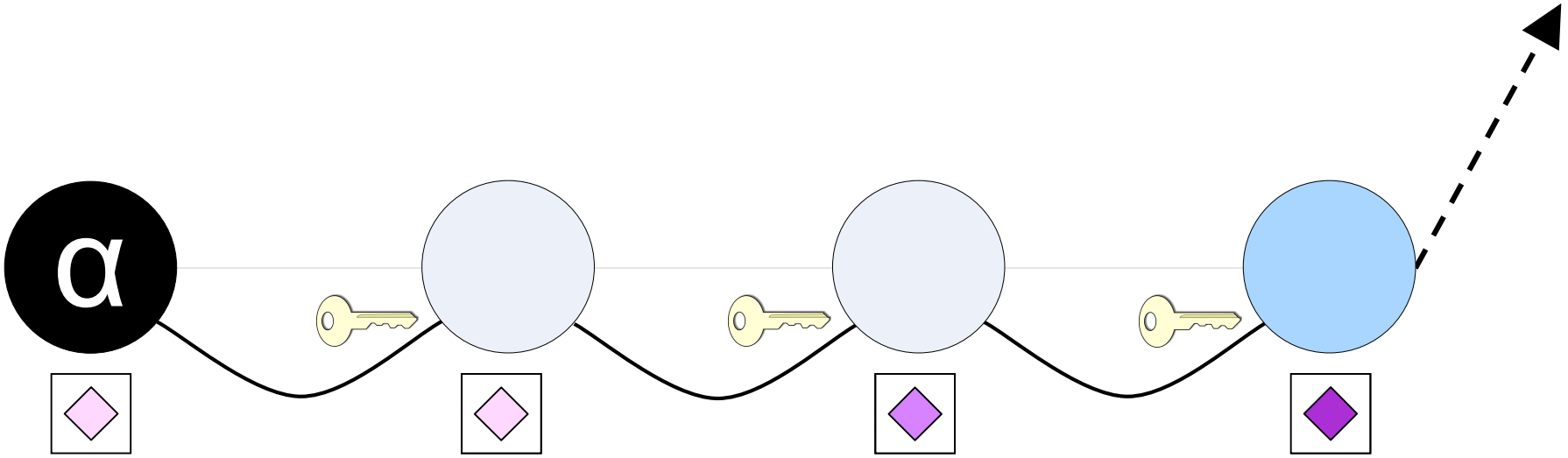
Secure Routing Tunnel Establishment (outbound)



The anonymized node informs the exit node

The exit node performs the connection, and confirms a successful connection back to the anonymized node

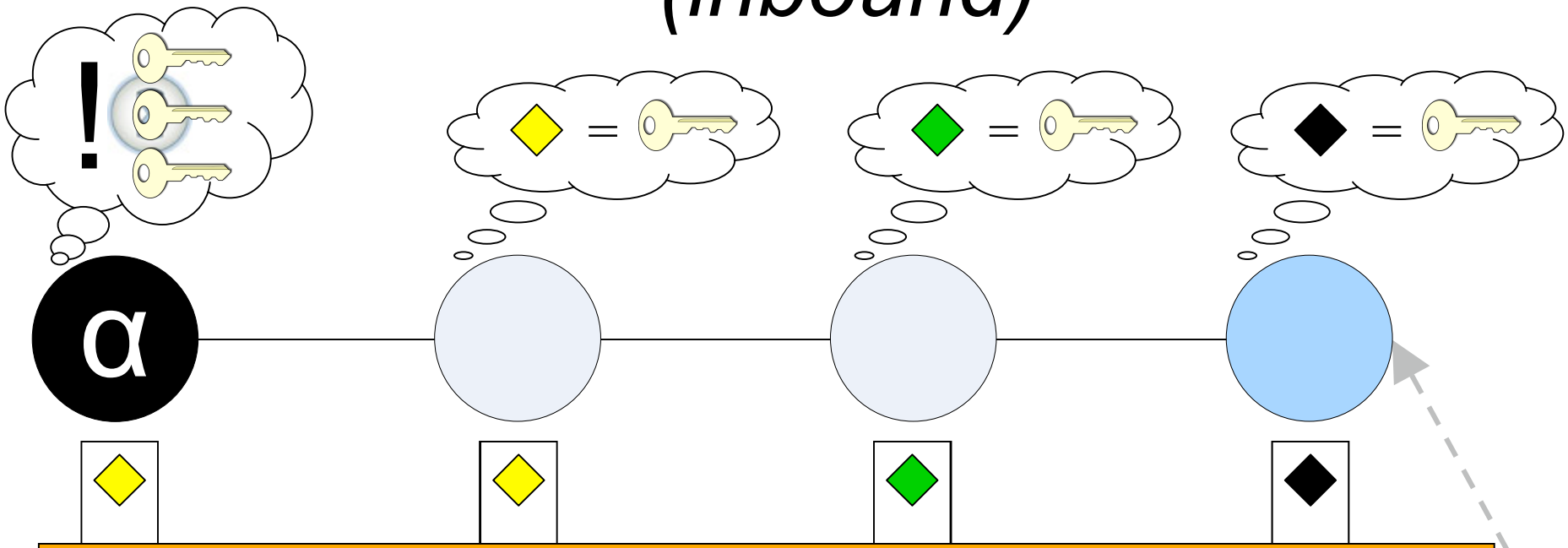
Secure Routing Tunnel Establishment (outbound)



The connection is fully established at both ends, and the application layer can now start communicating over it!

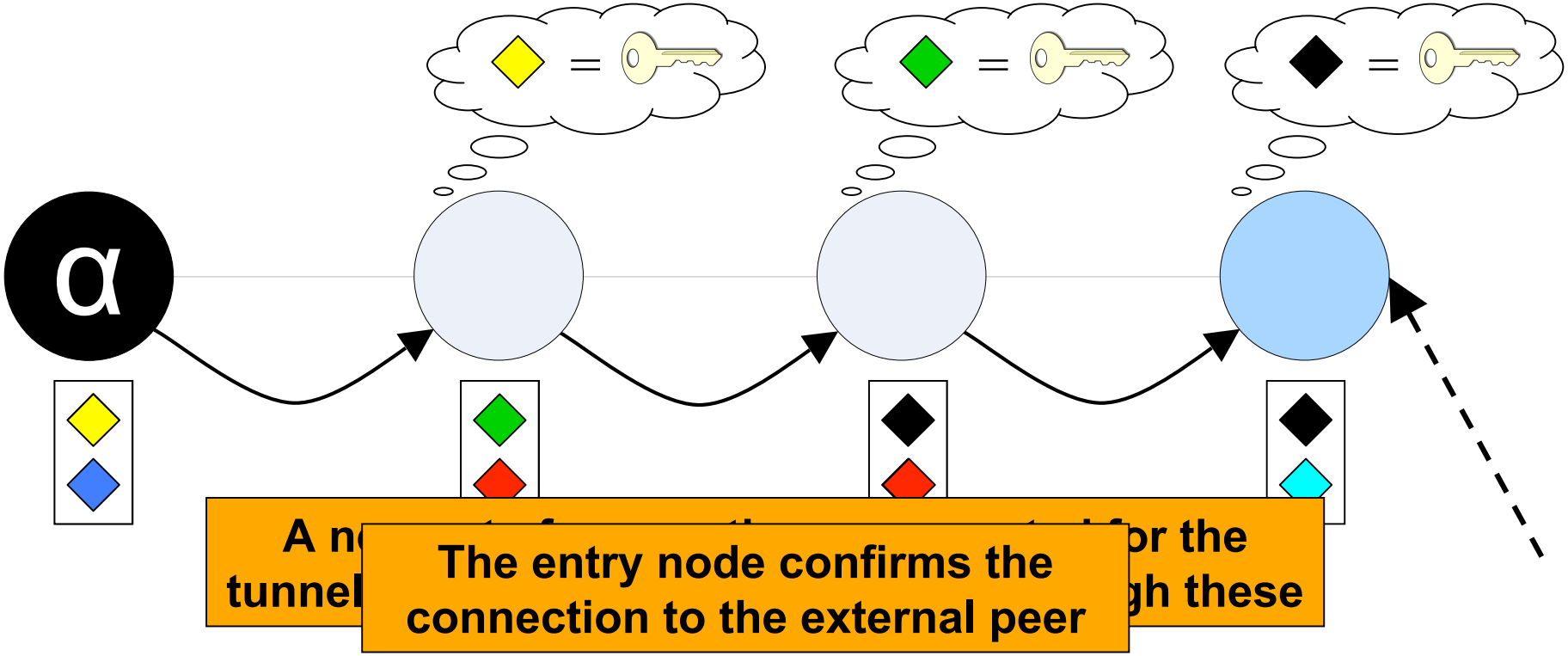
Repeat

Secure Routing Tunnel Establishment (inbound)

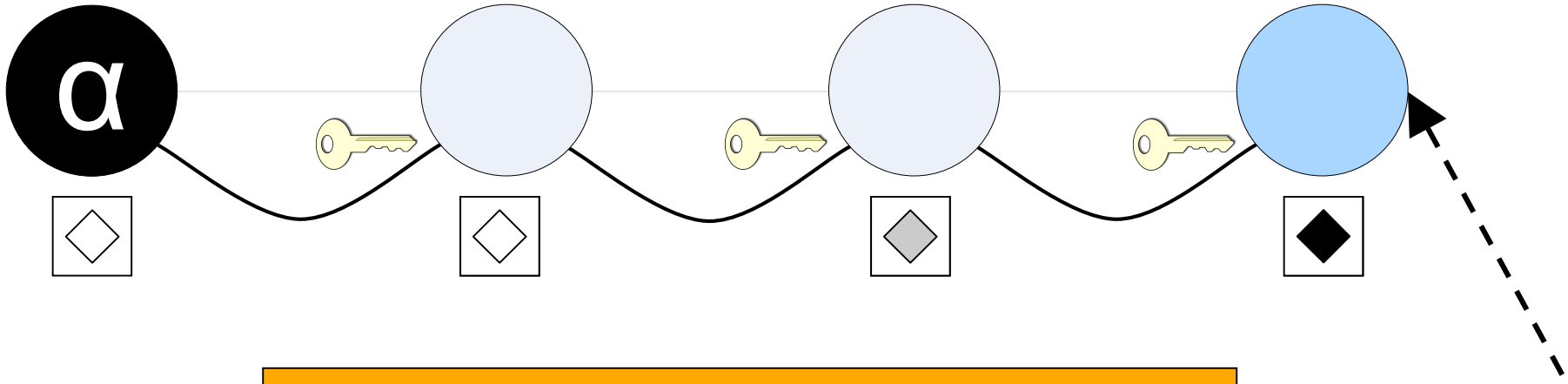


The initialization package enables the anonymized node to immediately derive the keys of all the intermediary nodes, while it is impossible for any of them to derive any key with it themselves

Secure Routing Tunnel Establishment (inbound)

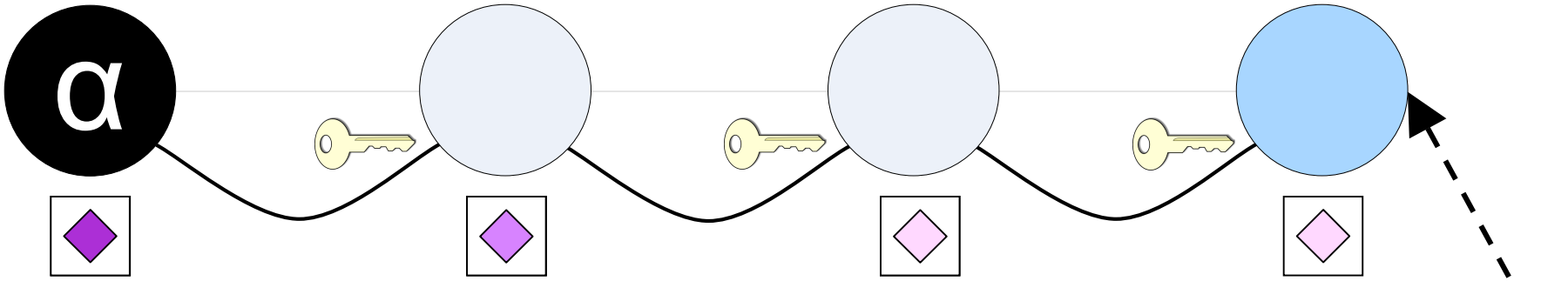


Secure Routing Tunnel Establishment (inbound)



It then confirms a successful connection back to the anonymized node

Secure Routing Tunnel Establishment (inbound)



To achie nnections
a though, a dummy package is first sent over the tunnel !

This symmetry is important!

Repeat

Secure End-to-End Encryption

- Once a full anonymized end-to-end connection has been established between two peers, double authenticated SSL can be used over it, as a final layer of encryption / authentication
- The used certificates can be stored in the network database, in the individual entries for each AP address

IP Backward Compatibility

- Identical format and functionality of IP
  - Address format
  - Port semantics
  - Connection semantics
- Binary hooks for all common network APIs
  - No need for any application author assistance
  - No need for any application source code
  - The application won't even know that its anonymized
- The common Internet DNS system can be used
- Simple to start supporting IPv6 and similar too

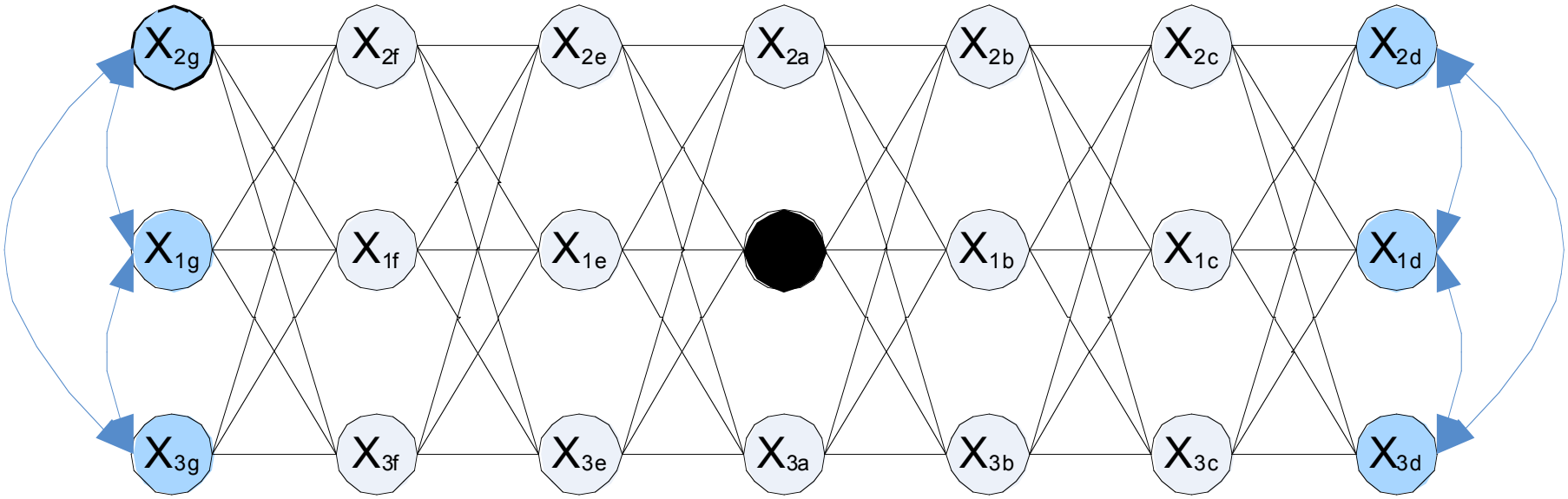
The Network Database

- Contains separate tables
  - Node IP address table, with associated info
  - Node AP address table, with associated info
- The database can be accessed through a specific strict API
- Voting algorithms, digital signatures and enforced entry expiry dates are used on top of the standard DHT technology in some cases, to help enforce permissions and protect from malicious manipulation of database contents and query results
- Resilient to ”net splits”

Manual Override Command Support

- Powerful emergency measure
  - Protection against DoS attacks
  - Restoration after possible more or less successful DoS attacks
  - Protection against known malicious nodes
- Signed commands can be flooded to all clients
  - Many DHT implementations natively support this feature
  - Commands signed by trusted party, e.g. project maintainers etc
  - Verification certificate hard coded into the client application
- Only commands for banning IP addresses, manually edit the network database etc, never affecting client computers!
- No real worry if signing keys would leak or be cracked
  - A minor update of the client could immediately be released, with a new key (verification certificate) hard coded into it, problem solved

High-Availability Routing Paths



Aftermath

Legal Aspects & Implications

- File sharing example:
  1. **Today: Lawsuits based on people connecting to a certain torrent**
  2. Lawsuits based on people using a certain file sharing program / protocol
  3. Lawsuits against endpoints in anonymization networks
  4. Lawsuits against routers on the Internet?
  5. Lawsuits based on people using a generic anonymization protocol
  6. Lawsuits based on people using cryptography?
  7. Lawsuits based on people using the Internet?

Legal Aspects & Implications

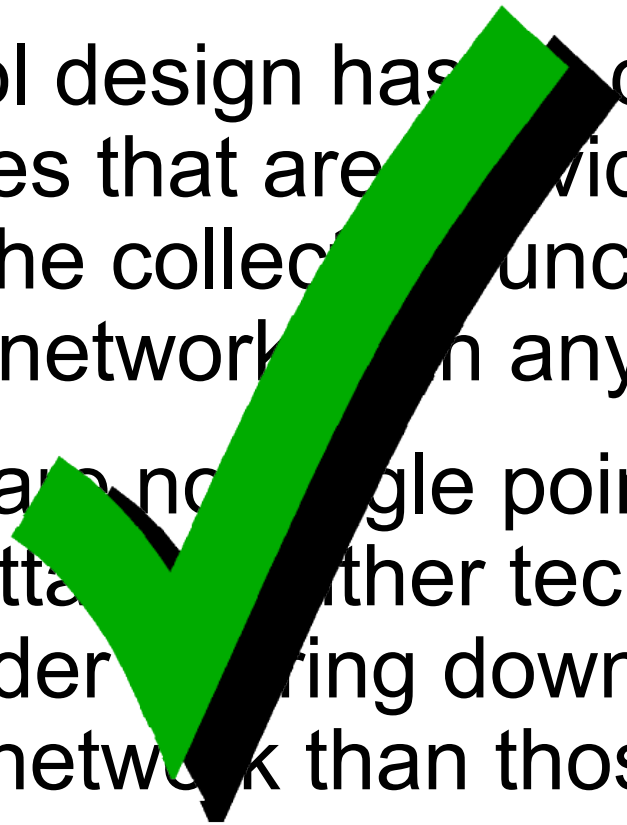
- License trickery?
  - A license for the main specification, saying that a certain EULA must accompany all implementations of the protocol
  - The EULA in turn, would say that through using the protocol implementation in question, the user:
    - Understands and agrees to that no node in the anonymous network can be held responsible for any of the data that is being routed through it, due to the simple fact that the user neither has any control over what such data may contain, nor any possibility whatsoever to access the data itself
    - Agrees to not use the protocol implementation to gather data that can or will be used in the process of filing a lawsuit against any of the network users that are just routing data
  - Probably won't work in many ways and several countries, but still an interesting line of thought to be investigated further

Review of Design Goals

- Review of our eight original design goals:
  1. Complete decentralization
  2. Maximum DoS resistance
  3. Theoretically secure anonymization
  4. Theoretically secure end-to-end encryption
  5. Complete isolation from the ”normal” Internet
  6. Protection against protocol identification
  7. High Traffic Volume and Throughput Capability
  8. Generic, Well-Abstracted and Backward Compatible

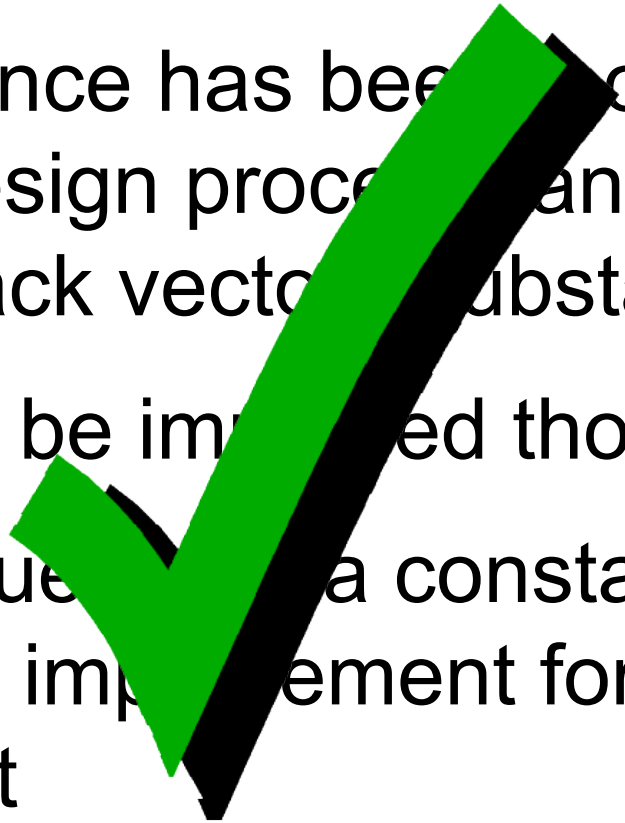
Review of Design Goal #1: Complete Decentralization

- The protocol design has no central points, or even nodes that are individually more valuable to the collective function of the anonymous network than any other
- Thus there are no single points of the network to attack either technically nor legally, in order bringing down any other parts of the network than those exact ones attacked



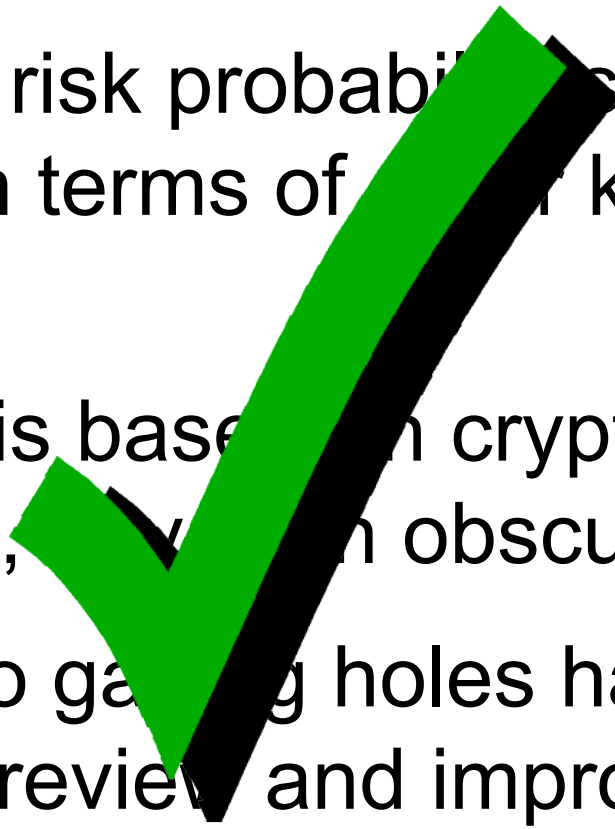
Review of Design Goal #2: Maximum DoS Resistance

- DoS resistance has bee concern during the entire design proc and has limited possible attack vecto ubstantially
- Can always be im ed though
- Must continue a constant area of concern and imp ement for future development



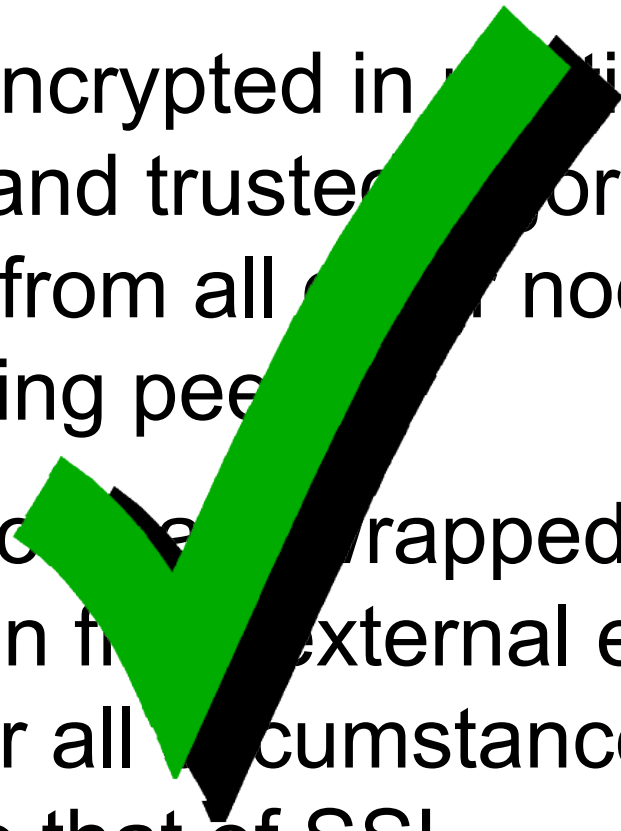
Review of Design Goal #3: Theoretically Secure Anonymization

- All involved risk probabilities can be expressed in terms of known probabilities
- All security is based on cryptography and randomness, not on obscurity or chance
- Hopefully no gaping holes have been left to chance, but review and improvements are of course needed, as always in security



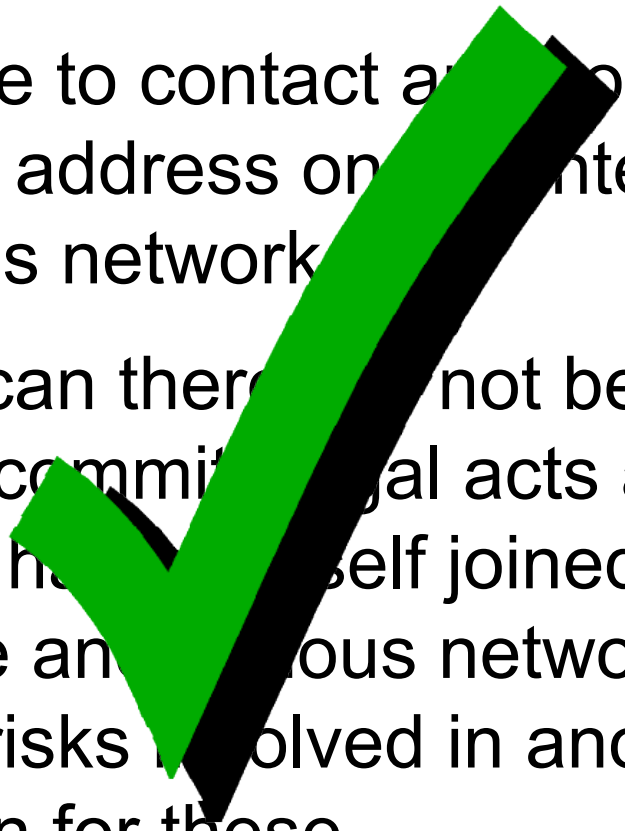
Review of Design Goal #4:
Theoretically Secure End-to-End Encryption

- All data is encrypted in multiple layers with well-known and trusted algorithms, protecting it from all other nodes except the communicating peers
- All connections are wrapped by SSL, so the protection from external eavesdroppers should under all circumstances be at least equivalent to that of SSL



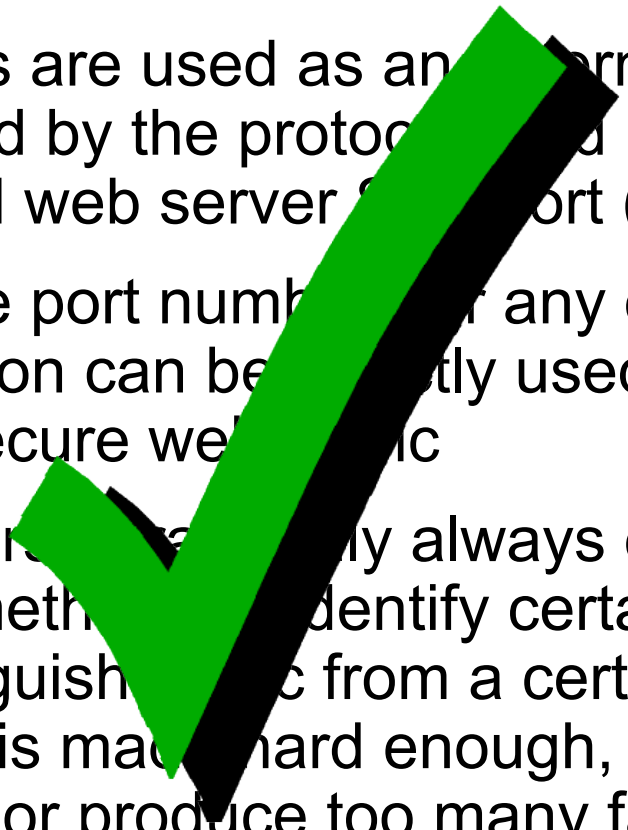
Review of Design Goal #5: Isolation from the "Normal" Internet

- It is impossible to contact a[nd] communicate with any regular IP address on the Internet from inside the anonymous network
- The network can therefore not be used to anonymously commit illegal acts against any computer that has itself joined and exposed services to the anonymous network, and thus accepted the risks involved in anonymous communication for these



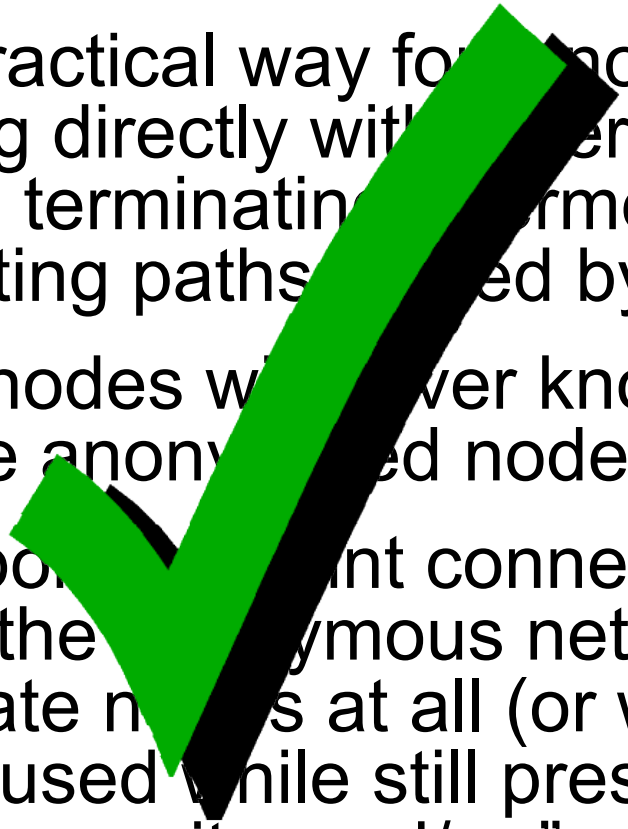
Review of Design Goal #6:
Protection against Protocol Identification

- SSL connections are used as an external shell for all connections used by the protocol and by default they also use the standard web server SSL port (tcp/443)
- Thus, neither the port number nor any of the contents of the communication can be directly used to distinguish it from common secure web traffic
- There are of course always enough advanced traffic analysis methods to identify certain kinds of traffic, or at least distinguish it from a certain other kind of traffic, but if this is made hard enough, it will take up too much resources or produce too many false positives to be practically or commercially viable



Review of Design Goal #7: High Volume / Throughput Capacity

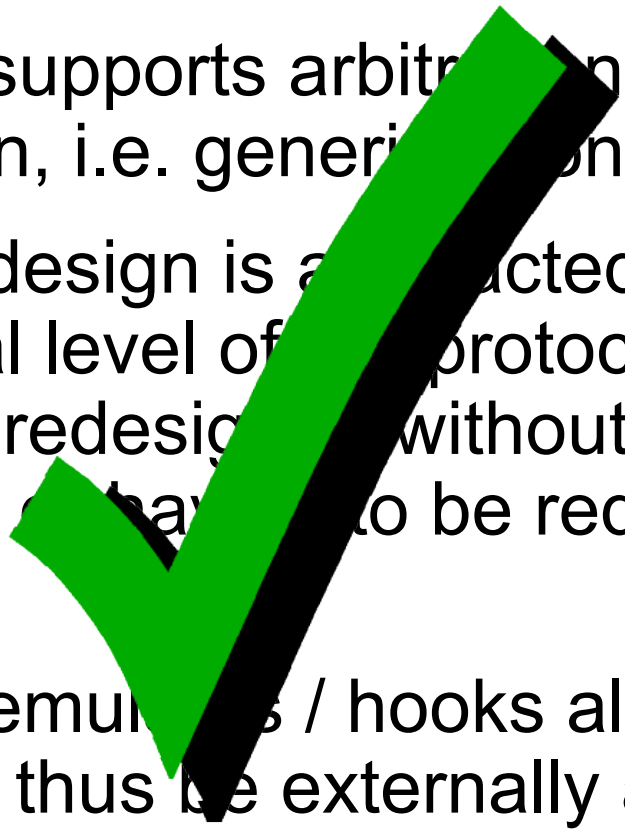
- There is no practical way for a node to know if it is communicating directly with a certain node, or rather with the terminating intermediate node of one of the routing paths created by this node
- Intermediate nodes will never know if they are adjacent to the anonymized node in a path or not
- Thus, single point connections between two nodes on the anonymous network, without any intermediate nodes at all (or with very few such), can be used while still preserving a great measure of anonymity, and/or ”reasonable doubt”



Review of Design Goal #8:

Generic, Well-Abstracted and Backward Compatible

- The protocol supports arbitrary network communication, i.e. generic anonymization
- The protocol design is abstracted in a way that each individual level of the protocol can be exchanged or redesigned without the other parts being affected or having to be redesigned at the same time
- The protocol emulates / hooks all TCP network APIs, and can thus be externally applied to any application that uses common TCP communication



Comparison with Other Anonymization Solutions

- Advantages of Phantom over TOR
  - Designed from the ground up with current and future practical anonymization needs and demand in mind
  - Compatible with all existing and future network enabled software, without any need for adaptations or upgrades
  - Higher throughput
  - No traffic volume limits
  - Isolated from the ”normal” Internet
  - End-to-end encryption
  - Better prevents positive protocol identification
  - Not vulnerable to ”DNS leak” attacks and similar

Comparison with Other Anonymization Solutions

- Advantages of Phantom over I2P
  - Compatible with all existing and future network enabled software, without any need for adaptations or upgrades
  - Higher throughput
  - End-to-end encryption
  - Better prevents positive traffic analysis identification

Comparison with Other Anonymization Solutions

- Advantages of Phantom over anonymized P2P
  - Less likely to be target of “general ban”
  - The generic nature of Phantom opens up infinitely much more potential than just binding the anonymization to a single application or usage area

Known Weaknesses

1. If all the nodes in a routing path are being controlled by the same attacker, this attacker can bind the anonymized node to the entry/exit node
   - No data can still be eavesdropped though, only what AP addresses it communicates with can be concluded
   - One very important detail is that it will be very hard for the attacker to conclusively know that its nodes actually constitute the entire path, since the last attacker controlled node will never be able to determine if it is actually communicating with the anonymized node itself, or with just yet another intermediate node in the routing path
   - The algorithms for routing path node selection can be optimized to minimize the risk of such a successful attack

Known Weaknesses

2. If an attacker monitors the traffic of *all* nodes in the network, it will be able to conclude the same thing as in the previous weakness, without even having to doubt where the routing paths end
   - This has been stated as a limitation from the start though
   - Some anonymization protocols try to counter such attacks by delaying data and sending out junk data, but this goes against the high throughput design goal of Phantom

Known Weaknesses

3. Individual intermediate nodes in a routing path could try to communicate their identity to other non-adjacent attacker controlled intermediate nodes in the same routing path, by means of different kinds of covert channels
   - Examples of such covert channels could be information encoding using timing or chunk size for communicated data
   - Could be countered to some degree by micro delays and data chunk size reorganization in intermediate nodes, but very hard to defend against completely
   - Again though, very hard for the attacker to conclusively know where in the path its nodes are located, since they will never be able to determine if they are communicating with another intermediate node or not, or even the direction of the path

Summary

- There is no complete specification of the Phantom protocol ready for immediate implementation
- The main goals of this project is rather to:
  - Explore the optimal requirements for an anonymization solution of today and future years
  - Provide examples of solutions for problems likely to be associated with these requirements
  - Inspire discussions about the design of such a system
  - Be the starting point of an open de facto standard for free, secure and ubiquitous Internet anonymization
- Please see the Phantom white paper for more details:
  - http://www.fortego.se/phantom.pdf

Future of Phantom

- A Google Code repository, wiki and discussion group has been reserved for the project, which will hopefully be able to work as a central coordinating location for future design, development and implementation of the Phantom protocol and the ideas inspired by it:

  http://code.google.com/p/phantom

Questions / Discussion

If you come up with a question later on, feel free to ask me over a beer, or to contact me by email!

magnus.brading@fortego.se