Stephen Tambussi
Student ID: 00001469512

COEN 329 (Network Technology)
Fall 2022

# Dark Internet

## *Audience*

This document covers the architecture, design and operation of the Dark Internet followed by a deep dive into the various technologies that comprise the Dark Internet. The Cybersecurity and Dark Internet section discusses the technical enhancements of the Dark Internet on computer security and Dark Internet weaknesses that threaten the anonymity of the network.

The reader is expected to have basic knowledge of computer networking to be able to understand the architecture and design of Dark Internet systems. Some background in the field of cybersecurity is desired to have a better understanding of the Dark Internet security strengths and weaknesses.

This document can be used by system architects, designers, developers, business analysts, academicians, researchers, and technical authors.

# *Table of Contents*

# *Table of Figures*

# *Table of Tables*

# 1

# 1. Introduction

**Darknets**, or the Dark Internet, are an overlay network within the overarching Internet that can only be accessed through specific software or authorization and typically uses a unique customized communication protocol. Darknet technologies enable the infamously illicit content found on the dark web, websites and communications purposely hidden from the world wide web. Since darknet is a broad term, there are a few different technologies that classify as darknets.

Some of the technologies that classify as darknets:

- Peer-to-peer file hosting services such as Freenet
- Anonymity proxy networks such as Tor and I2P

This report will cover both of the aforementioned darknet technologies. However, the focus will be on anonymity proxy networks, such as Tor and I2P, and their implications with respect to cybersecurity.

This report consists of four chapters. The first chapter provides an overview of darknet systems, their design and operation, and differences between the public Internet and the Dark Internet.

The second chapter will go more in-depth into the popular technologies that implement darknets. Starting with file hosting services, the chapter will then provide an in-depth analysis of anonymity proxy networks including their architecture, operation, and services. The specific software implementations covered are Freenet, Tor, and I2P. Particular focus will be on Tor and I2P since they are popular and widely used software.

The third chapter analyzes the association of darknet technologies with cybersecurity and the results of this relationship. Specifically, the various ways darknet technologies enhance online security and privacy are discussed. Common darknet weaknesses and the vulnerabilities of specific darknet implementations are also covered.

Finally, the last chapter discusses the conclusion of this report, the future aspects of darknet technology, and its influence on cybersecurity.

# 2

# 2. Understanding the Dark Internet

## 2.1 The Dark Internet and Internet

The Dark Internet, as mentioned before, is any overlay network within pre-existing networks, which is only accessible through specific software. These darknets provide the means for the notoriously illegal websites, services, and content to exist on the dark web. For clarification, the dark web is not the Dark Internet. They are two distinct, but closely related concepts. The dark web is the content and websites that exist on the Dark Internet.

Darknets grew out of a need for more secure communications across the Internet as they could be easily monitored and analyzed. As such, the Dark Internet is best understood as a security enhancement of existing Internet technologies. To better understand the relationship between the Dark Internet and the Internet, it is best to look at their conceptual differences.

Table 1 - Differences between the Internet and Dark Internet

| | **Internet** | **Dark Internet** |
|---|---|---|
| **Accessibility** | Highly accessible through popular web browsers such as Chrome, Firefox, etc. | Less accessible as it requires special software such as Tor or I2P which may be difficult to use for the average user |
| **Users** | Typically anyone that requires access to web content | Typical users are hackers, whistleblowers, government agencies and corporations, or anyone who desires to conceal their identity |
| **Anonymity** | Very little as IP addresses can be used to identify users | Almost completely anonymous except for a few security weaknesses |
| **Security** | Can be secure when connections utilize encryption technologies such as SSL or TLS | More secure than the public Internet, especially when security weaknesses are fixed |
| **Safety** | Relatively safe when practicing good Internet browsing behaviors | Can be dangerous, particularly for inexperienced users |

## 2.2 Design and Operation

The various darknet technologies available all have different implementations and design choices. However, every darknet implementation shares three characteristics of their network.

1. Darknets are mostly, if not completely, decentralized. Darknets use peer-to-peer methods where the data is stored across multiple connected computers in the network that can be geographically distributed. This is in contrast to conventional networks where the data is stored in centralized servers.
2. Darknets use the already existing infrastructure of the public Internet. They do not have their own physical infrastructure, but rather use custom software to separate their network from the public network.
3. Darknets use custom network protocols and ports to isolate users outside their network.

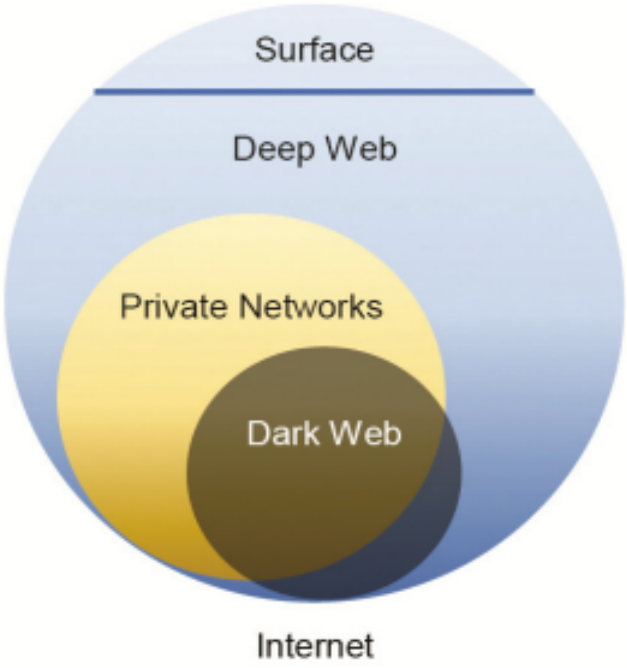


Figure 1 - Layers of the Internet

Figure 1 displays the visual representation of darknets in relation to the overarching Internet. The dark web, representative of the Dark Internet, is simply overlaid on top of the existing Internet using custom software to isolate itself. This concept is known as an overlay network. Every darknet system operates in this manner. Where systems differ is in the implementation or design of the custom software used to isolate their network.

As such, a key component of darknets is an overlay network. An overlay network is a computer network that is created on top of an existing network. They are blind to the underlying network and its services. Consequently, discovery and routing in overlay networks is performed at the application layer.

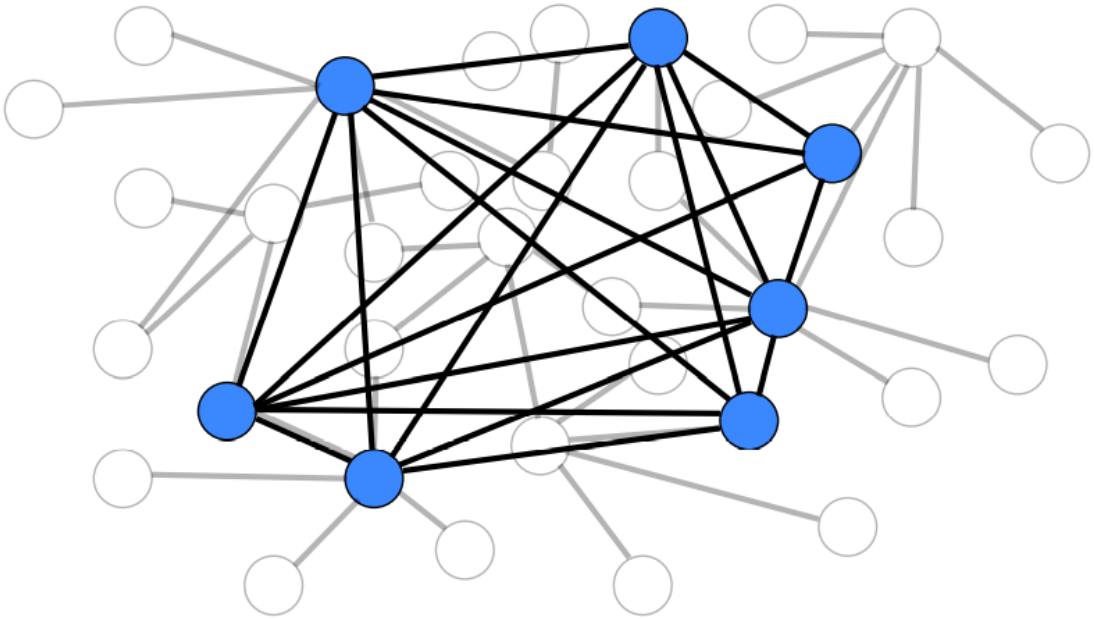

Figure 2 - An Overlay Network

The above figure shows how overlay networks are created on top of an existing network, but still connected to it. Nevertheless, overlay networks are logically separate from the underlying network and this characteristic is what enables the anonymity of darknets.

Similar to public networks, darknets also have certain functional requirements that they must fulfill. These five requirements are as follows:

1. Supports a method for enabling the user to input new information or data into the network.
2. Provides a distribution network that transmits copies of data to other users.
3. Contains widely available devices that allow users to download data from the network.
4. Implements a search mechanism to enable users to find data or information.
5. Incorporates a storage mechanism that can retain data for periods of time to facilitate fast retrieval and limit exposure of darknet nodes that input data into the network.

## 2.3 Darknet Implementations

There are many different software products available that utilize darknet technology to either improve security and privacy, provide a distributed system of data sharing, or a combination of both. The following software products are popular implementations of darknet technology.

**Freenet** is a peer-to-peer file sharing service. Using an implementation of darknet technology, it enables decentralized, secure, and anonymous uploading and downloading of data files across its network. Freenet achieves these goals by dividing and distributing the data for a single uploaded file across multiple nodes (connected devices) in the network. Each piece of the file is encrypted. When users download a file, intermediate nodes, which only have knowledge of their immediate neighbors, are used to pass on requests for the desired data. With this design, Freenet's network is capable of powerful anonymity in its communications.

Freenet provides multiple services beyond just anonymous file sharing. Featuring an application programming interface (API), many other services have been created that expand upon its base functionality.

1. Messaging Forums
2. Website hosting
3. Online chats

**I2P**, or the Invisible Internet Project, is an anonymity network. Beginning as a fork of Freenet, I2P evolved into its own project to rival Tor. Its implementation uses end-to-end encryption for all communication messages across its network with network endpoints acting as cryptographic identifiers. This enables senders and receivers to remain completely anonymous by concealing their IP addresses. Furthermore, each message is sent across the network through geographically distributed volunteer nodes and can take many different paths to reach its destination. Through this design, I2P enables highly secure and anonymous communications in its network.

Since I2P is designed as an anonymizing network layer, it can support many of the same services available in the public Internet. As such, I2P offers more services compared to Freenet.

1. Email
2. Website hosting
3. Instant messaging
4. File sharing
5. Messaging Forums

**Tor**, or The Onion Router, is a popular anonymity proxy network. As one of the most famous applications of darknet technology, it implements an anonymous communication technique known as onion routing. Essentially, each message is encrypted in a multi-layered manner and then randomly bounces through volunteer network nodes until it reaches its destination. In addition to this, Tor operates in such a way that, to the receiver of the message, it appears the origin point of the traffic is from the last node used in the transmission and not from the actual sender. In this way, Tor provides excellent privacy protection for its users.

Similar in concept to I2P with a few key differences, Tor supports connections to the public Internet along with its own custom services, known as onion services. I2P also offers custom services, but it is incapable of making connections outside of its own network. Therefore, Tor provides the same set of services as I2P, but with greater functionality due to its ability to connect to the public Internet anonymously.

# 3

# 3. Technologies of the Dark Internet

The creation of darknets stemmed from a singular goal, privacy. The Internet was not originally designed with privacy in mind. As a result of this, many of the later developments to improve privacy were added on top of existing technologies and susceptible to vulnerabilities. Darknets are an attempt to solve this lack of privacy while still maintaining compatibility with the foundational technologies of public networks.

Darknet software can be classified into two primary types, peer-to-peer file sharing and anonymity networks. The following section describes a file sharing software known as Freenet and its underlying implementation. Subsequent sections will discuss anonymity networks.

## 3.1 Peer-to-Peer File Sharing: Freenet

Freenet operates as an adaptive peer-to-peer network for the purpose of anonymous file distribution and access. As such, its architecture is designed in such a way to support this functionality.

The network consists of nodes that query one another for storage and retrieval of data files. These files are named by location-independent keys. Every node in the network has its own local storage space that it makes available to the network. In addition, each node keeps track of which nodes have which file through a system known as a distributed hash table (DHT). Through this, if a user requests a file, the network can find the node that possesses the requested data. Freenet is designed so that most users will also run nodes, both to improve security and increase the storage available to the network as a whole.

Every file in the Freenet network is identified by file keys generated through a hash function. These keys are used for every operation involving files, including searching, retrieval, and uploading. Additionally, Freenet uses key-based routing (KBR), in conjunction with a DHT, to find the closest node that has the requested data by matching the file key.

Requests in the Freenet network are passed along from node to node with each node deciding where to next send the request. Each request is encrypted to prevent a malicious node from intercepting the message. Since Freenet is designed for privacy, each node only has knowledge of its immediate upstream and downstream neighbors. As a result, the routing algorithms for data storage and retrieval adaptively adjust routes using only local knowledge of the network. Furthermore, each request has a hops-to-live value which is comparable to IP’s time-to-live. This value is decremented at every node to prevent infinite loops in the network.

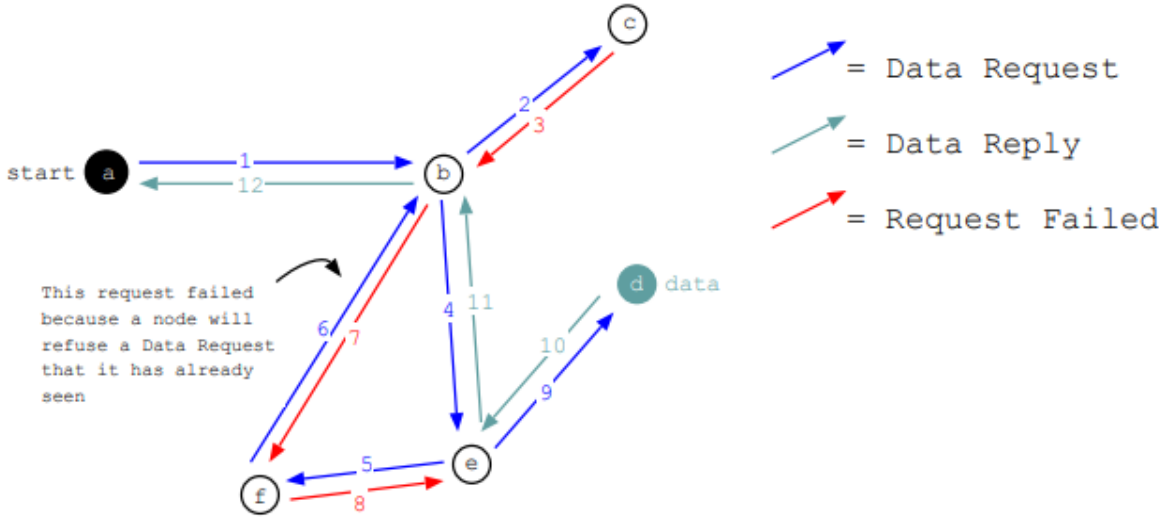


Figure 3 - Freenet Request Sequence

Figure 3 visualizes a typical request sequence for data in the network. The request message will be continually forwarded around the network, until the requested data has been found. At multiple points, a node may respond back with a “request failed” message due to its inability to contact any other nodes and lacking the desired data. A node may also reply with a “request failed” message because it has seen the request before to prevent a loop.

For data storage, the key for the file is first calculated. Then, an insert message is sent to the user’s node with the proposed key and hops-to-live value, which determines the amount of nodes to store the file on. If the file key is already in use by another file, the node will return the pre-existing file to indicate a collision. However, if the key is not already in use, the insert message will be forwarded to the node with the key closest to the proposed key.

Freenet was designed as a packet-oriented network with self-contained messages. Every message in the network has a transaction ID which allows nodes to keep track of inserts and requests. As such, Freenet operates at the application layer and works independently of the underlying transport protocol. Therefore, it is compatible with both the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) for its message transport.

## 3.2 Anonymity Proxy Networks: Tor and I2P

Tor is an anonymity proxy network that operates through a geographically distributed overlay network to hide a user's traffic data for the purpose of privacy. It does this through a technique known as onion routing.

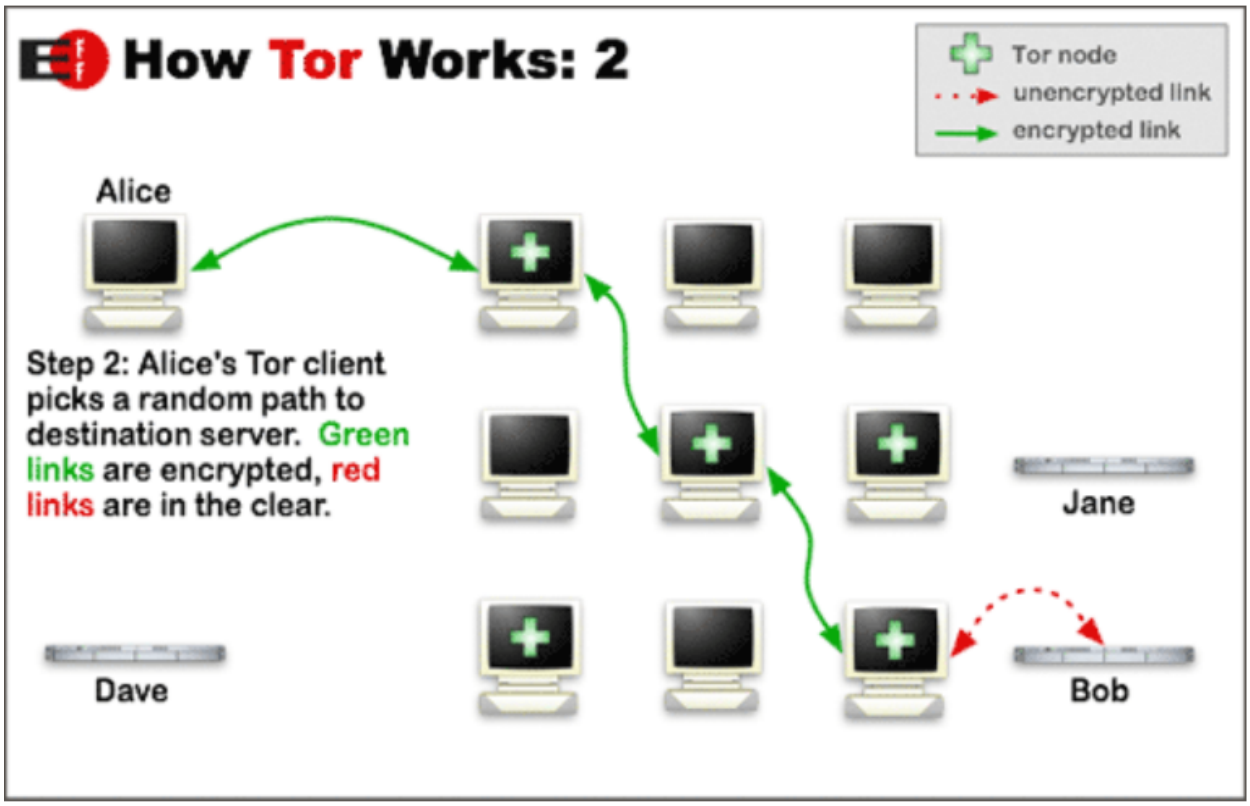


Figure 4 - Tor General Operation

Figure 4 shows a high-level overview of Tor's operation. The source node transmits the communication message through multiple encrypted links across the network's relays. The path the message will take is determined randomly to further conceal the origin of the message from traffic analysis. New connections within 10 minutes of the initial selection of the path will use that same path. After this time period, a new forwarding path will be selected to further help conceal a user's traffic patterns.

Regular Tor connections still leave the message decrypted at the end of the forwarding path as seen in the figure above. For that reason, Tor also supports a method that provides anonymity to not only users on the network, but also websites and other servers. These servers, which are configured to only receive connections through Tor, are called onion services. Each onion service has an associated onion address through which they are accessed. This onion address allows servers to keep their IP address, and therefore location, hidden. A Tor node uses these addresses by referencing a distributed hash table (DHT) for the

server's key and then using KBR to establish a connection to that server. In this way, onion services provide complete end-to-end encryption for users in the Tor network.

Similar to Tor, I2P is another anonymity network that conceals user's traffic through end-to-end encryption. This encryption for I2P is provided by a method known as garlic routing.

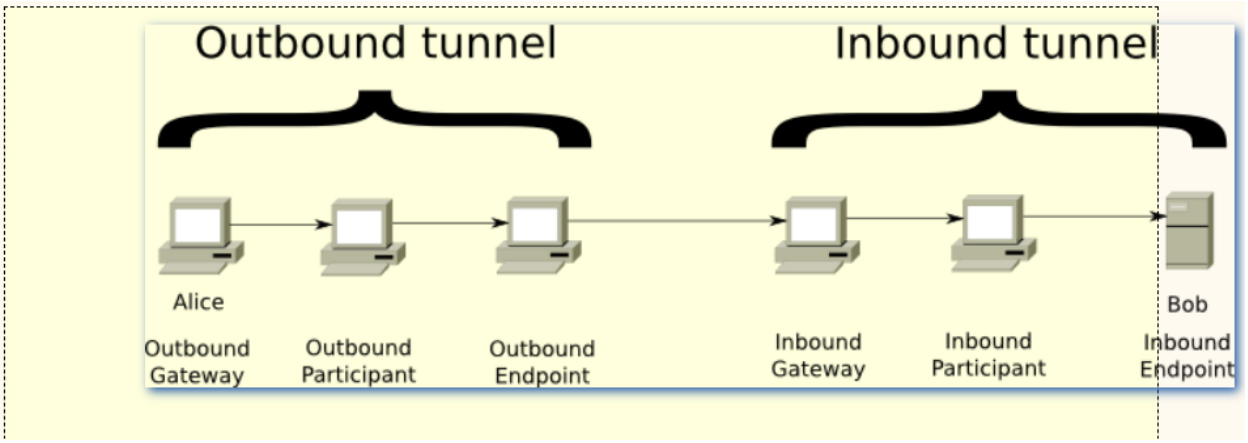


Figure 5 - I2P Tunnel Concept

In the above figure, the primary method of communication for I2P is shown. Peers in the I2P network communicate using a tunnel, which is a directed path through a list of selected routers. Tunnels are unidirectional, meaning that communication messages can only be sent in one direction. As such, to have two-way communication between peers, two separate tunnels are needed. Furthermore, layered encryption is implemented in such a fashion that each router only removes a single layer of the encryption. This decrypted information contains the address of the next router and the encrypted data to be forwarded.

A tunnel's entry point is known as the gateway, while its exit point is called the endpoint. The outbound tunnel transmits messages away from the tunnel creator and the inbound tunnel brings messages toward the tunnel creator. The combination of these two tunnel types enables users to communicate with each other.

I2P also includes a DHT network database known as netDb. This database is used to share network metadata. NetDb contains two types of metadata, routerInfo and leaseSets. RouterInfo provides routers in the network with the required information to contact another router such as public keys and transport addresses. LeaseSet gives routers the necessary data to contact a particular destination through specifying the tunnel gateway to the destination. Similar to Tor, I2P uses its DHT in combination with KBR to support secure communications across the network.

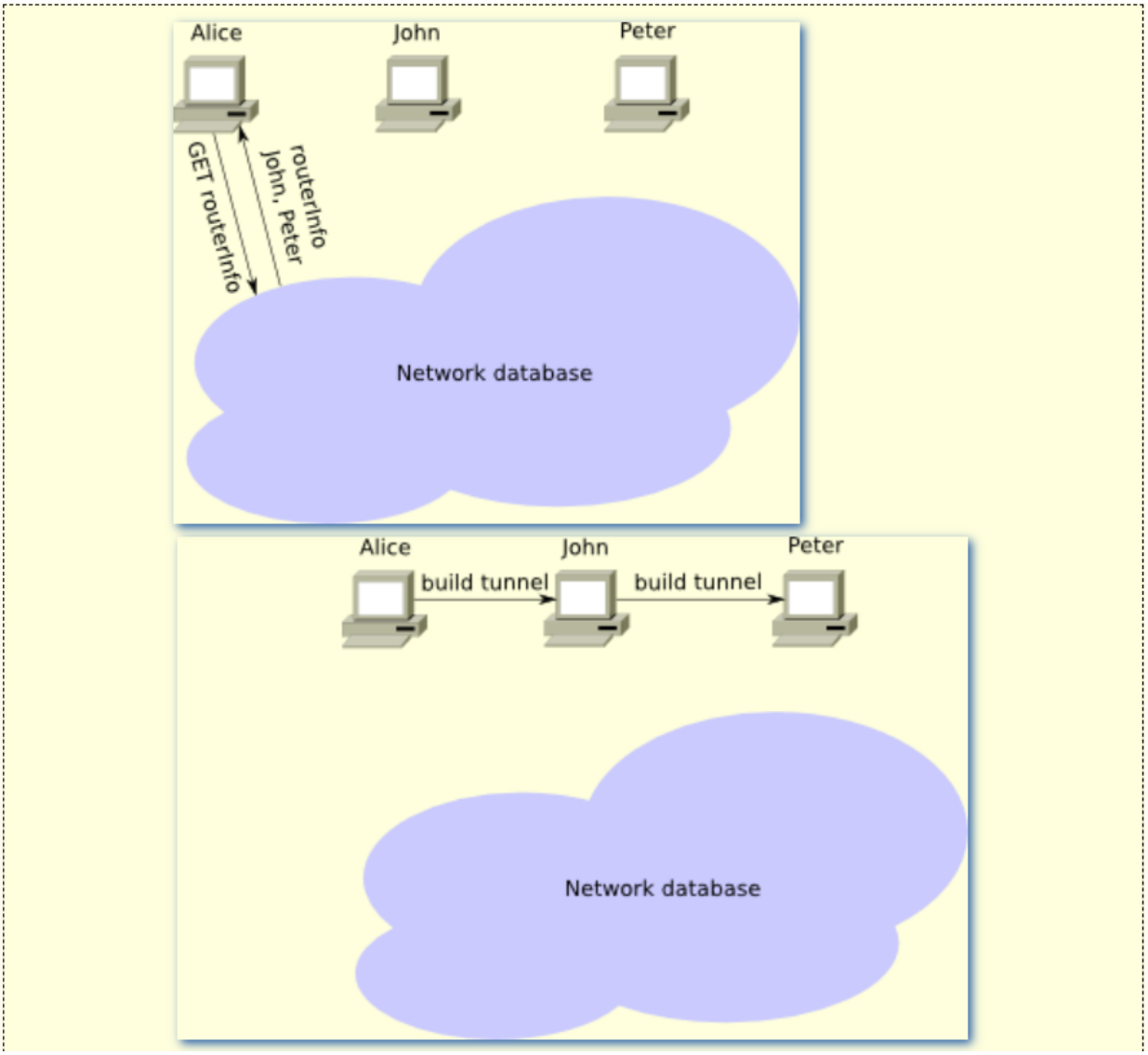


Figure 6 - Building Tunnels with DHT Database

As seen in the above figure, when a user wants to build a communication tunnel, they query the DHT network database to collect routerInfo. This routerInfo will provide them the necessary public keys and transport addresses to contact another router. The user can then send a build message to the first router in the desired path to request the creation of a tunnel. The first router in the path will forward the build message onward to subsequent routers until the tunnel has been constructed.

Tor and I2P are similar in many ways. Like Tor, I2P offers a comparable anonymous website hosting service known as eepsites with special addresses similar to Tor's onion address. Both network technologies are based on the concept of mix networks where messages are shuffled and encrypted at a router, limiting traceability of a connection. However, beneath the surface, there are some significant differences between their implementation and technology. The following sections describe their implementations and the differences between the two darknet technologies.

### 3.2.1 Onion Routing

Onion routing is the technique through which Tor achieves its anonymous communication goals. In this technique, each message in the network is encapsulated in layers of encryption, similar to the layers of an onion and hence the name. At each network node, known as onion routers, a single layer of the encryption is removed to reveal the message's next destination. In this way, the message sender maintains anonymity because each node only knows its previous and next node, not the entire message path.

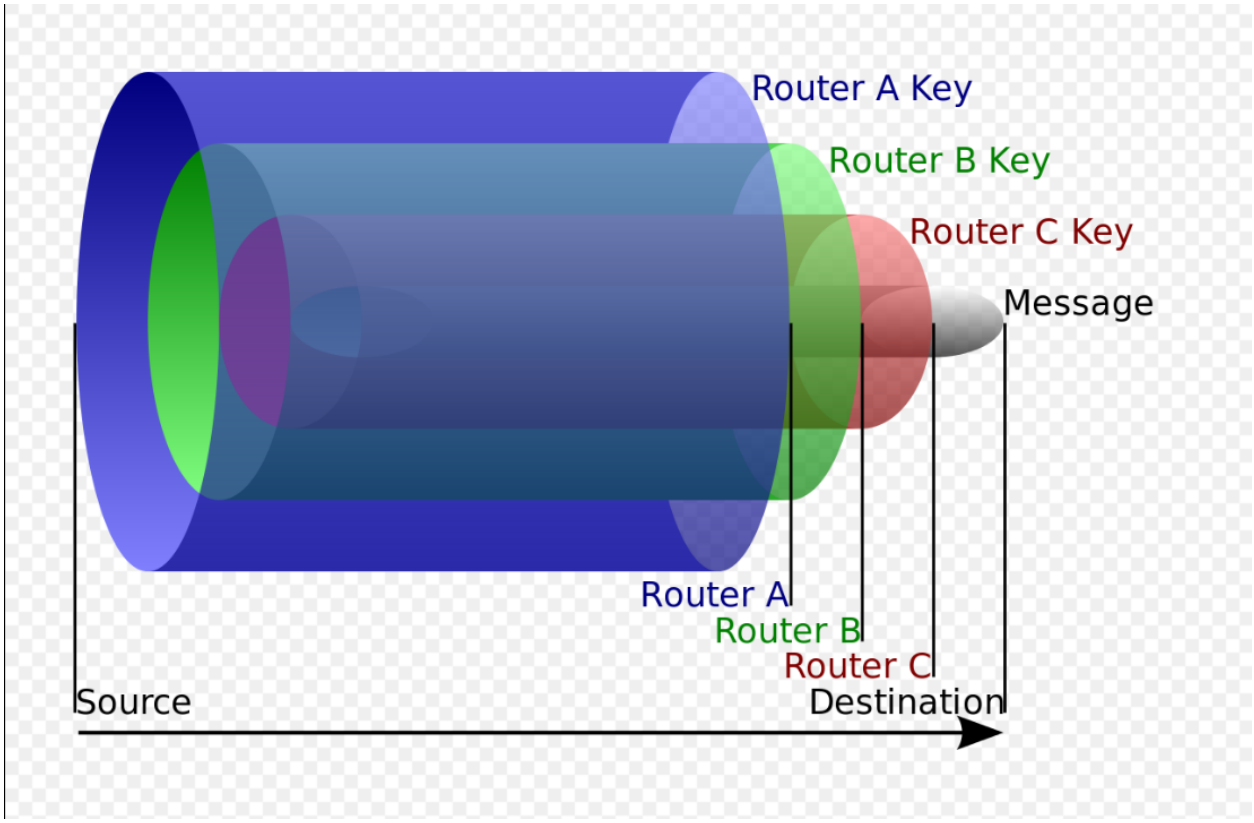


Figure 7 - Onion Layers

In the above figure, each successive router will remove a layer of encryption to determine the message's next destination and its sender. This encapsulated message is considered a data structure known as an onion. Even if the message at a router just came from the actual origin, the router is incapable of determining this and only sees the sender as another node in the network. Just before the message reaches its destination, the final router removes the last layer of encryption and transmits the original unencrypted message to its destination. Furthermore, the number of layers encapsulated around a message is determined by the amount of routers in the forwarding path.

The figure below displays the progression of an encrypted onion message as it goes through its path in the network. The sending node will negotiate a symmetric encryption key (the shared key) with each relay node. At the exit relay, the message is completely decrypted and sent to its destination.

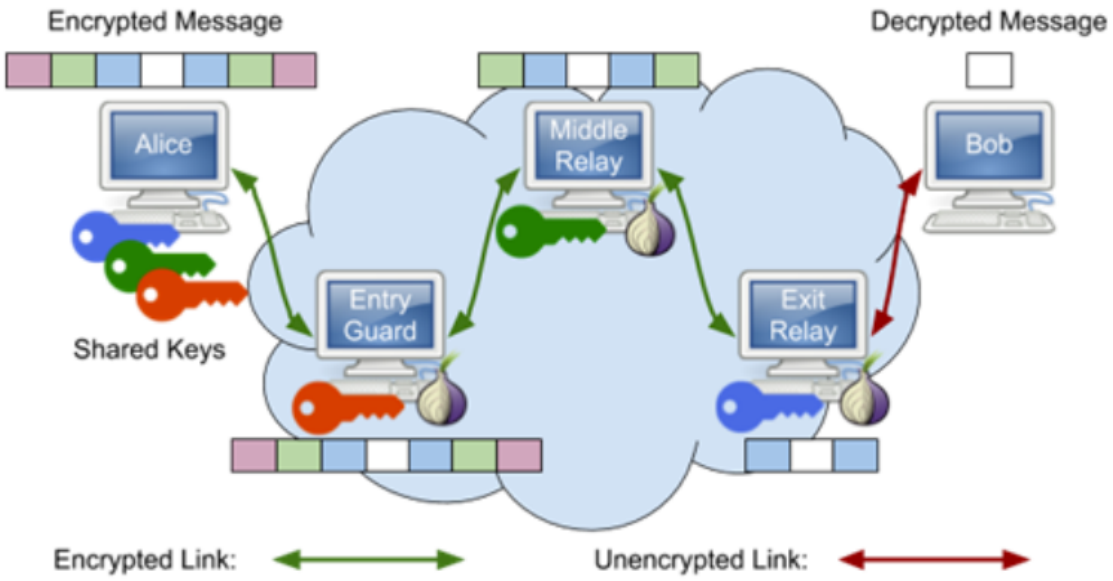


Figure 8 - Progression of Onion Routing

Despite the name, onion routing occurs in the application layer and not at the network layer. Onion routing depends on IP routing for forwarding the data passed through socket connections. As a result of this, the route that a message might take between onion routers is determined by the underlying IP network. Also, onion routing relies on connection-based communication to transmit data uncorrupted and in-order. Therefore, most, if not all, onion routing implementations utilize the Transmission Control Protocol (TCP), including Tor.

To better understand onion routing, it is necessary to observe the process for onion creation and transmission. The process is as follows:

1. The originator (sender) selects a group of nodes from a list that is provided by the directory node.
2. These nodes are arranged to form a path through which the message will be sent.
3. Through asymmetric key cryptography, the sender gets a public key from the directory node.
4. The sender uses this public key to send an encrypted message to the first node, known as the entry node, for establishing a connection and a shared key (session key).
5. From this encrypted link to the entry node, the sender will relay a message through the first node to a second node in the chain that only itself, and not the first node, is capable of decrypting.
6. After the second node receives the message, it will establish a connection with the first node and extend the encrypted link from the sender. The second node does not know whether the first node is the sender or not.
7. This link creation process can be continued to build larger chains in the network.
8. Once the chain is complete, the sender will transmit the message.

### 3.2.2 Garlic Routing

Garlic routing is I2P's equivalent of Tor's onion routing. In simple terms, it is a variant of onion routing and operates similarly. However, whereas onion routing encrypts one message for a single transmission, garlic routing encrypts multiple messages together for a single transmission. The benefit of this is two fold. First, garlic routing enhances resilience against traffic analysis attacks. Second, data transfer speeds are increased due to multiple messages being transmitted at once.

Messages in garlic routing are known as garlic messages, or just garlic for short. Each garlic message contains multiple cloves. These cloves are fully formed messages that have their own instructions for delivery. In I2P, messages are added into a garlic message when it is desired to conceal their information from other peers. Moreover, a garlic message is encrypted using a specific public key so that intermediary nodes are incapable of discovering the number of messages in the garlic, their content, and where cloves are destined. Through this technique, garlic routing provides complete end-to-end encryption for every message sent through its network.

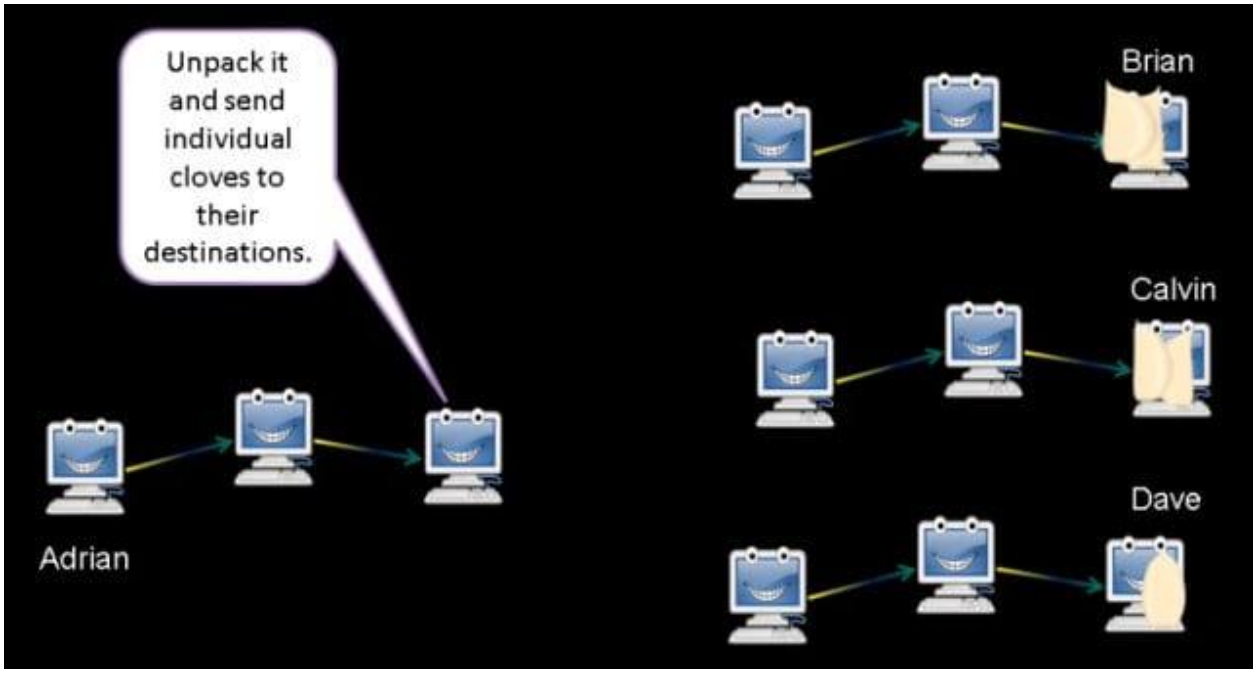


Figure 9 - Garlic Routing Visualized

Shown in the above figure, once a garlic message reaches the end of the tunnel, each clove in the garlic is unpacked and sent to their specified destination. This enhances security and resolves network transfer speed limitations that are present in Tor.

### 3.2.3 Tor vs. I2P

Tor and I2P have many similarities and differences. Both platforms share the same goal of total user anonymity across the network. However, the means by which they each achieve this goal are different.

Tor is generally intended to connect to a website in the public Internet and provides greater anonymity than using a more mainstream internet browser. As mentioned before, Tor also offers hidden services, known as onion services, which provide complete end-to-end encryption for the user connecting to the onion website. Nevertheless, Tor does not offer end-to-end encryption when connecting to a public Internet website and this leaves a user more vulnerable to being identified. On the other hand, I2P supports complete end-to-end encryption for every connection because it is only capable of connecting to its own hidden services and not the public Internet. I2P offers many types of hidden services including anonymous websites known as eepsites, email, file sharing, and others.

Another important difference is that I2P is more decentralized than Tor. Tor contains directory servers which maintain data on the status of each node in the network. Each server services a particular geographical area that is closest to the server's location. These servers are controlled by around 9 authority figures, who are people that created Tor originally, and are the entire root of trust in the Tor network. I2P does not have centralized servers for network metadata and instead completely distributes its metadata across the network.

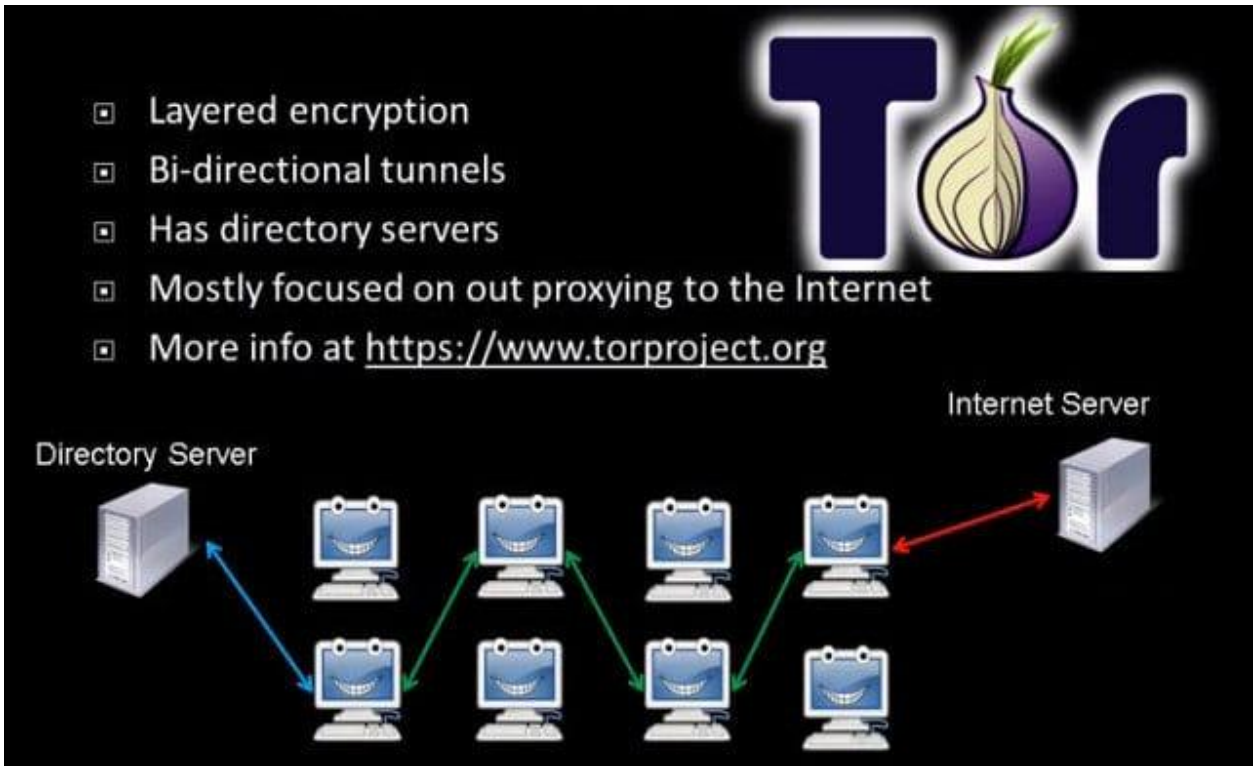


Figure 10 - Overview of Tor

The above figure summarizes the key technical points of the Tor network and its goals. The figure below does the same for I2P.

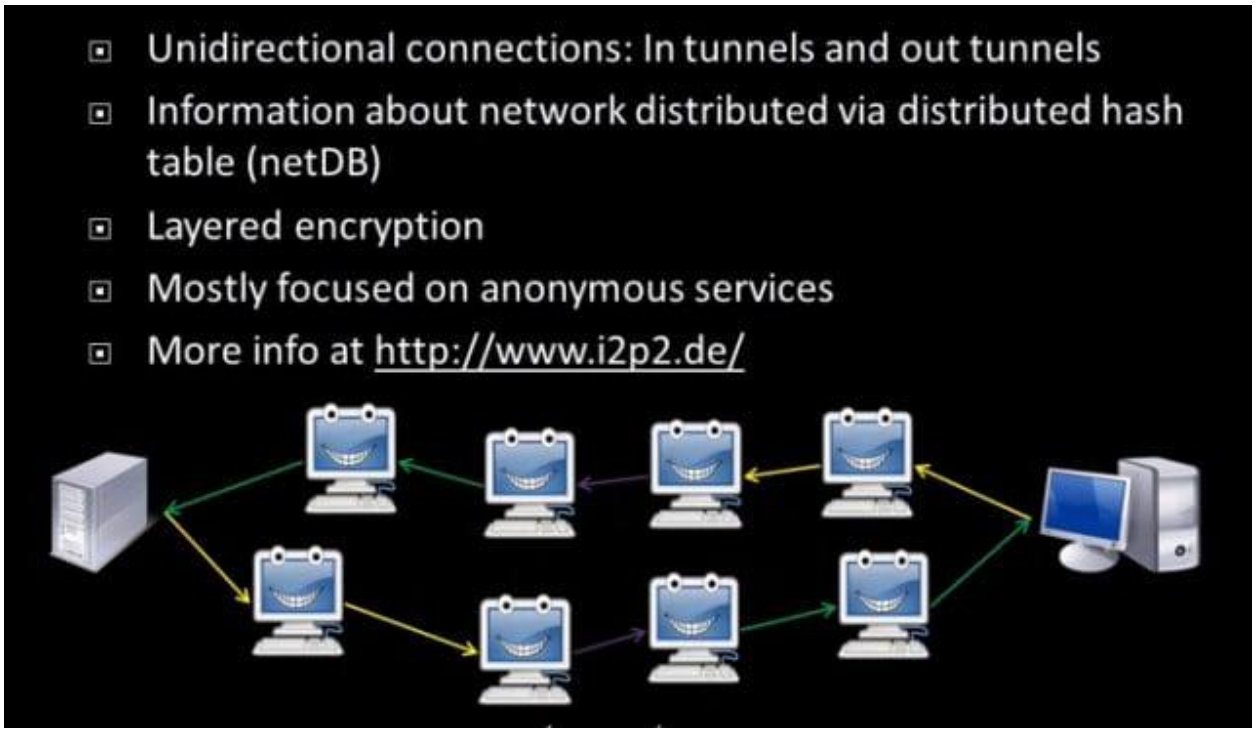


Figure 11 - Overview of I2P

Design-wise, I2P is conceptually a packet switched network, while Tor is a circuit switched network. I2P can route around congestion or other network failures and load balance data across all available resources. This makes I2P more efficient in communications especially when network resources are limited.

Regarding anonymity, I2P offers more protection against network attackers. As discussed earlier, I2P peers communicate through unidirectional tunnels. Consequently, an I2P tunnel reveals half the amount of traffic data as a Tor circuit, which is bidirectional. For example, an HTTP request and response would travel the same path in the Tor network, but in I2P the request packets would travel through one or more outbound tunnels and the response packets would return through one or more different inbound tunnels.

Both technologies have their strengths and weaknesses, but Tor still holds some key advantages over I2P. For one, Tor can connect to the public Internet and help anonymize the connection. I2P is incapable of doing this and can only connect to its own hidden services inside its network. Furthermore, I2P is less popular and therefore does not have the same level of software maturity as Tor. As a result of this, Tor offers many more features, software, and developer support over I2P. Another advantage is that due to Tor's popularity, it has been more thoroughly tested and checked for vulnerabilities in its core

implementation. Finally, I2P may encounter future issues of scalability since its network is small relative to Tor and its design is unproven.

To summarize, the following table highlights the key differences between Tor and I2P.

Table 2 - Tor vs. I2P

| | **Tor** | **I2P** |
|---|---|---|
| **Intended Usage** | Anonymized connections to the public Internet | Connections to its own hidden services |
| **Hidden Services Support** | Supports end-to-end encryption through onion services | Supports end-to-end encryption through hidden services such as eepsites |
| **Decentralized** | Mostly decentralized except for its directory servers | Completely decentralized as its network metadata is managed by a DHT database |
| **Tunneling** | Bidirectional | Unidirectional |
| **Network Type** | Circuit switched | Packet switched |
| **Maturity** | Highly popular with a large development community | Growing in popularity but still lacking in software maturity |
| **Proven Design** | Highly tested and widespread usage | Still needs greater adoption to test its network scalability |
| **Anonymity** | Weaker due to its bidirectional tunnels and exit node vulnerability when connecting to public Internet | Greater anonymity as it uses unidirectional tunnels and can only connect to its own hidden services inside its network |

# 4

# 4. Cybersecurity and the Dark Internet

A major aspect of the Dark Internet is, of course, its association with cybersecurity. Cybersecurity is increasingly crucial in modern times with massive networks supporting countless devices all transmitting potentially sensitive data. Without cybersecurity, malicious actors could steal sensitive data, disrupt systems that operate critical infrastructure, and manipulate users across the network. Darknets were designed to ensure anonymous communications across networks and therefore are an important component of cybersecurity to protect networked systems from malicious actors.

As such, it is important to discuss the specific enhancements darknets make to cybersecurity, but also the weaknesses in the technology that threaten user anonymity. The following sections will cover these topics and provide an overview of vulnerabilities for Tor and I2P, two of the most popular darknet implementations.

## 4.1 Security and Privacy Enhancements

The primary enhancement darknets make to cybersecurity is the anonymity of their communications. The Internet was not originally designed with consideration for privacy and so therefore regular traffic is easily traceable by third parties through traffic analysis and identification of a device's IP address. This enables hackers, governments, businesses or any other third party to determine the origin of a network communication and its destination. Through this, third parties can take a variety of actions including malicious attacks (hackers), content censorship or criminal prosecution (governments), and personal data collection (businesses). Accordingly, it is clear why users may want to anonymize their internet communications.

As it has been the focus of this report, there are a variety of different methods to anonymize network communications. However, many of these methods share similar characteristics. Freenet uses key-based routing and all data uploaded to the network is distributed and encrypted across its nodes. Similarly, Tor and I2P use key-based routing, but also utilize layered encryption and randomized routing to further secure a message as it travels across the network. Key-based routing (KBR) enables nodes to conceal their IP addresses when transmitting data and are used in conjunction with a DHT. Layered encryption prevents intermediate nodes from examining the message in-transit. Randomized routing helps limit third parties from tracing the path a message takes in the network. The combination of all these methods helps prevent third parties from determining the origin of communications, the paths they travel, and their content.

The results of these techniques are virtually complete privacy online. Of course, with the introduction of any new technology there are many positive and negative outcomes in its usage.

The security and privacy advantages of darknet technologies have provided the following benefits to its users.

1. Freedom of communication by anonymizing connections. This has helped users access censored content in their countries, aiding whistleblowers, and many other restricted information scenarios.
2. Help ensure confidentiality between communicating parties. For example, critical government data shared between its organizations needs to be secure from tampering or unintended access.
3. Stops personal data collection and limits surveillance of online activity from governments or companies.
4. Prevents digital stalking of a user from malicious actors in the network.

Nevertheless, these privacy advantages are attractive for malicious hackers and criminals as it prevents them from being identified. Darknet technologies have enabled the following crimes.

1. Copyright infringement with file sharing networks and their untraceability.
2. Propagating criminal data and software such as stolen information and malware.
3. Enabling the trade of illegal substances and services in online black markets.
4. Supporting terrorist activities by providing anonymous forums and communications through which they can coordinate.

Despite the risks inherent in darknet technologies, they still see active development and usage. Even still, every technology has its limitations and darknets are no different. Certain vulnerabilities exist that threaten the security and privacy benefits of darknet technologies.

## 4.2 Darknet Weaknesses

As with any piece of software, there are vulnerabilities that can be exploited by bad actors to compromise the integrity of a user's machine. Darknets do provide better protection against network surveillance and other privacy invasion methods, but they are still susceptible to these network threats, especially when their weaknesses are exploited.

A potential threat to darknet anonymity is traffic analysis where a malicious actor can examine transmitted messages to infer information about those messages from communication patterns, even when they are encrypted. This can result in the sending and receiving user being identified. Almost every known darknet implementation is vulnerable to some type of traffic analysis attack, including Tor. And while both Tor and I2P implement safeguards to minimize the threat of traffic analysis attacks, they are still possible.

This is not the only weakness of Tor. Being the most popular darknet implementation, Tor has had many vulnerabilities exposed. Another type of vulnerability that is inherent in Tor's design is the complete decryption of a transmitted message at its exit node. As seen above in previous figures, a sent message has its last layer of encryption removed at an exit node to be sent to its destination. A malicious exit node can exploit this to capture the unencrypted traffic and collect information about its source through both its payload and protocol data. This exit node vulnerability only occurs when using Tor to connect to the public Internet as onion services are completely end-to-end encrypted. It can also be mitigated by utilizing end-to-end encryption for the message itself with either Secure Sockets Layer (SSL) or Transport Layer Security (TLS).

A general network threat that even darknets are susceptible to is a Distributed Denial of Service (DDoS) attack. In this attack, attackers construct botnets, which are large groups of devices connected to the network and controlled by the attackers, and command them to transmit massive amounts of fake traffic into the network. Typically, this traffic is directed at one or more network servers with the goal of overloading them and therefore making them unusable. Evidently, it would appear that darknets such as Tor and I2P would be resilient to such attacks because they are decentralized. This is partially correct as they are resistant to common types of DDoS attacks, but still some attacks exist that are specialized for each darknet and can bring them to their knees.

One final weakness of darknet systems is human error in their usage. Many users seeking anonymity in their online activities have been identified solely because of their improper usage of darknet software like Tor. A simple mistake is all that is needed for a third party, such as a government or hacker, to identify the user through their traffic. As such there are certain behaviors that users should be aware of to ensure their anonymity when using darknet software.

- Only connect to websites that use HTTPS, a secure version of HTTP that is end-to-end encrypted. In Tor, this can help mitigate the exit node vulnerability.
- Disable Javascript when browsing. This will minimize the amount of browser code executed when viewing websites and can help protect against attacks that exploit Javascript vulnerabilities.
- Do not run torrenting services that use the BitTorrent protocol as these softwares work in such a way that a user's real IP address is more likely to be disclosed. Furthermore, they are already limited in transfer speed and the extra layer of encryption from darknets will only slow them down more.
- Keep darknet software up-to-date to always receive the latest security updates that patch vulnerabilities.
- Do not use usual public Internet services where online activity is recorded or tracked such as on social networking sites. While not directly identifiable, this data can be traced back to the user.
- Split internet usage between the public Internet and darknet browsers.

Ultimately, like any technology, there are limitations to the privacy granted by darknets. Traffic analysis is a serious threat to darknet anonymity that can be used to identify users. Tor experiences vulnerabilities inherent in its design that can be exploited to examine network messages in-transit. DDoS attacks and human error are other vulnerabilities that are applicable to all darknet technologies.

### 4.2.1 Tor and I2P Vulnerabilities

Tor and I2P are still in active development and as a result there are occasionally software bugs that introduce vulnerabilities. There are also flaws inherent in their designs that have been successfully exploited. However, many of these vulnerabilities and exploits have since been patched or are easily mitigated by end users through proper configuration and careful browsing behaviors. Even so, there is a set of vulnerabilities that can prove to be particularly dangerous to a user's anonymity.

A **DNS leak** is a vulnerability present in both Tor and I2P. In general, when a user searches the internet and selects a website to visit, the Domain Name System (DNS) is used to convert the website's human readable address (.com) into a machine readable IP address. If Tor or I2P are not configured correctly, then when a user visits either a public Internet site or hidden service website, a query to convert the human readable address will be sent to a DNS server. These queries are recorded and a third party monitoring the server can access this information. While a DNS server would not have data on the activity of the user, it would have data on the websites a user visited. This data is unencrypted and enables surveillance of a user's online browsing history. Essentially, this vulnerability results from a misconfiguration where communications are sent outside the confines of a darknet.

Figure 12 illustrates the DNS Leak vulnerability and how a monitored DNS server can be used by a third party to track the websites visited by a user. If the user does not use a proxy server for DNS, then the request will go directly to the DNS server unencrypted. Malicious actors can collect this data and use it to build a profile to assist them in identifying a user.

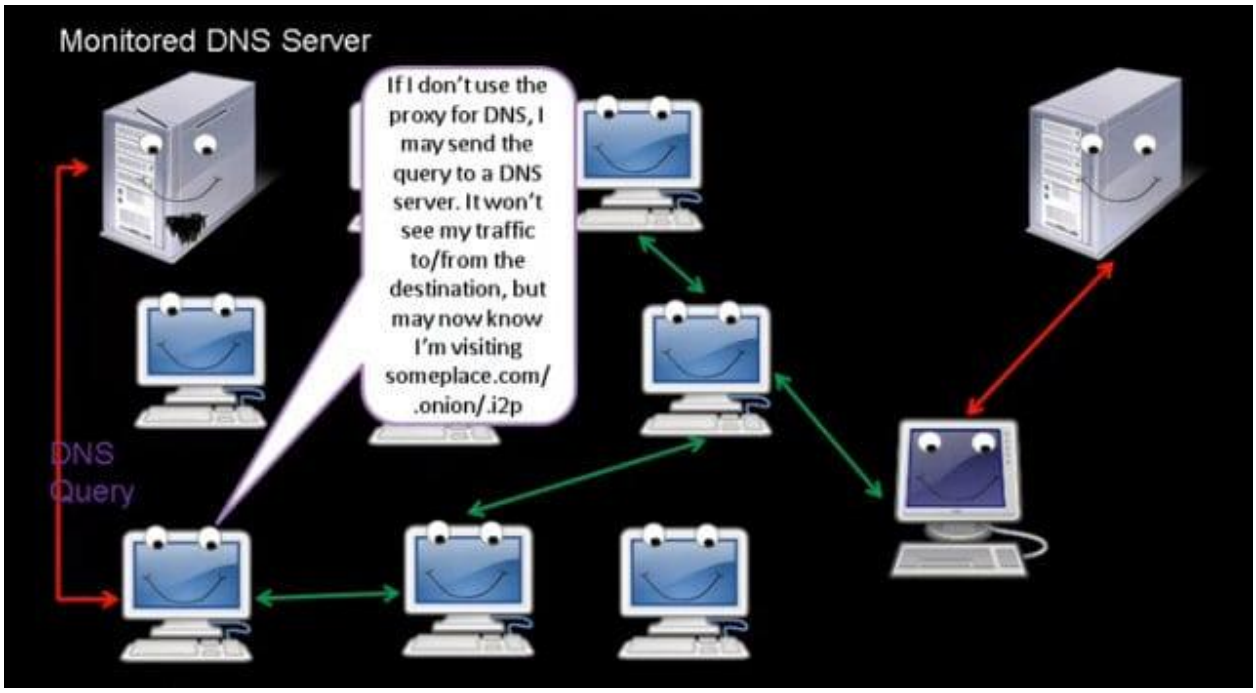


Figure 12 - Example of DNS Leak

To mitigate the DNS Leak vulnerability, every connection should be through a proxy server. In Tor and I2P, the default configuration should operate in this manner. However, it is good practice to verify that all traffic is going through the secured ports. For Tor specifically, users should validate that DNS is being used through SOCKS, a session layer protocol that forwards traffic between nodes through a proxy server.

**BitTorrent usage** introduces a vulnerability when used with Tor due to the incompatibility between the two technologies. As mentioned before, it is not safe practice to use the BitTorrent protocol over Tor and increases user risk of identification. BitTorrent uses a DHT for facilitating connections between peers. Connections to the DHT are through UDP, but Tor lacks support for UDP as it operates using TCP. Therefore, if a user is torrenting over Tor, the peer-to-peer traffic may be routed through the Tor network, but the DHT traffic is sent out via UDP and unencrypted. BitTorrent traffic has two primary attributes that can be used to determine the IP address of a user, a peer ID and port number. Malicious actors can analyze these attributes from the unencrypted connections over UDP to determine a user's IP address and identify them.

This vulnerability is illustrated below where a malicious exit node can discover the IP address of a user and circumvent the anonymity granted by Tor.

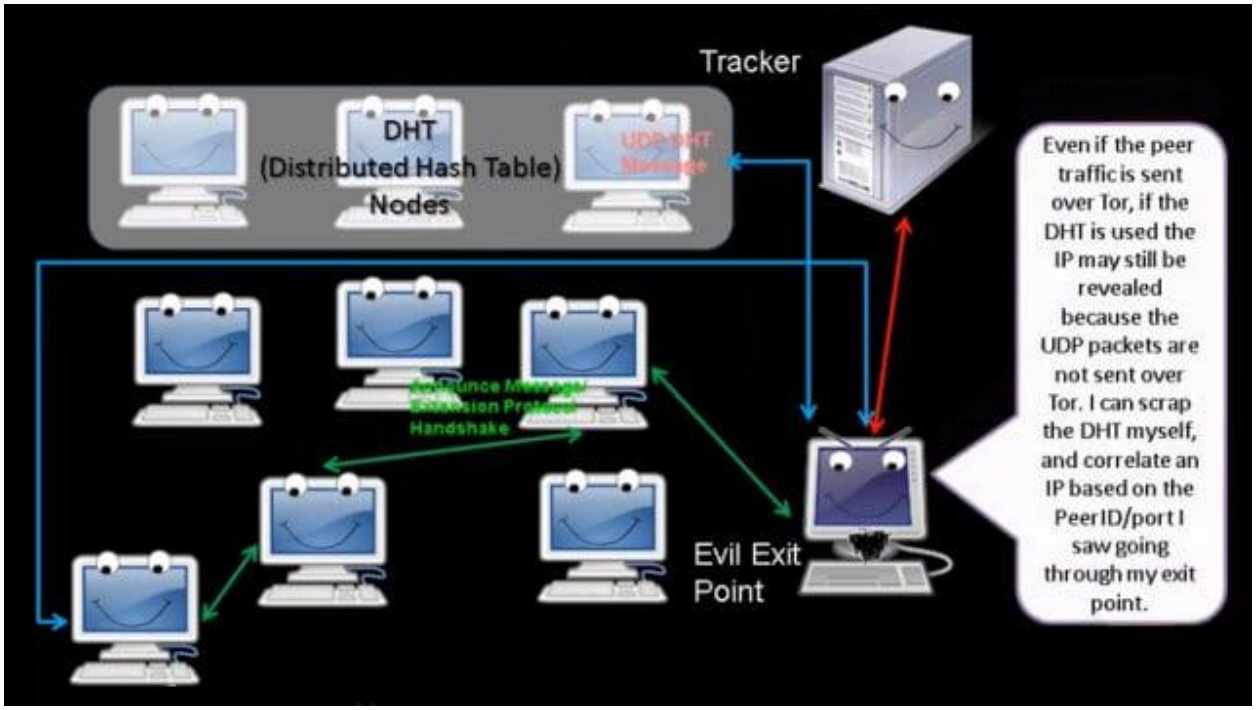


Figure 13 - BitTorrent over Tor Vulnerability

There are not many mitigations to the BitTorrent vulnerability as it exposes an inherent design flaw in the Tor network. The best solution is to simply not use BitTorrent over Tor and instead use a VPN with BitTorrent to achieve the same effect. Other options include using Tor designed file sharing applications or file sharing services offered by other darknets, such as I2P and Freenet.

**Clock-based vulnerability** is an issue found in both Tor and I2P. This issue cannot deduce the identity of a user alone, but it can be used in conjunction with other vulnerabilities to completely deanonymize them. Essentially, certain darknet protocols allow users to check a remote system's clock. If these clocks are synchronized to the local time of their geographical location, then this information can be used to determine where that system is in the world and further narrow down the potential users of a node.

This vulnerability is better illustrated below by figure 14 where a malicious user queries a node hosting a hidden service website for its time.

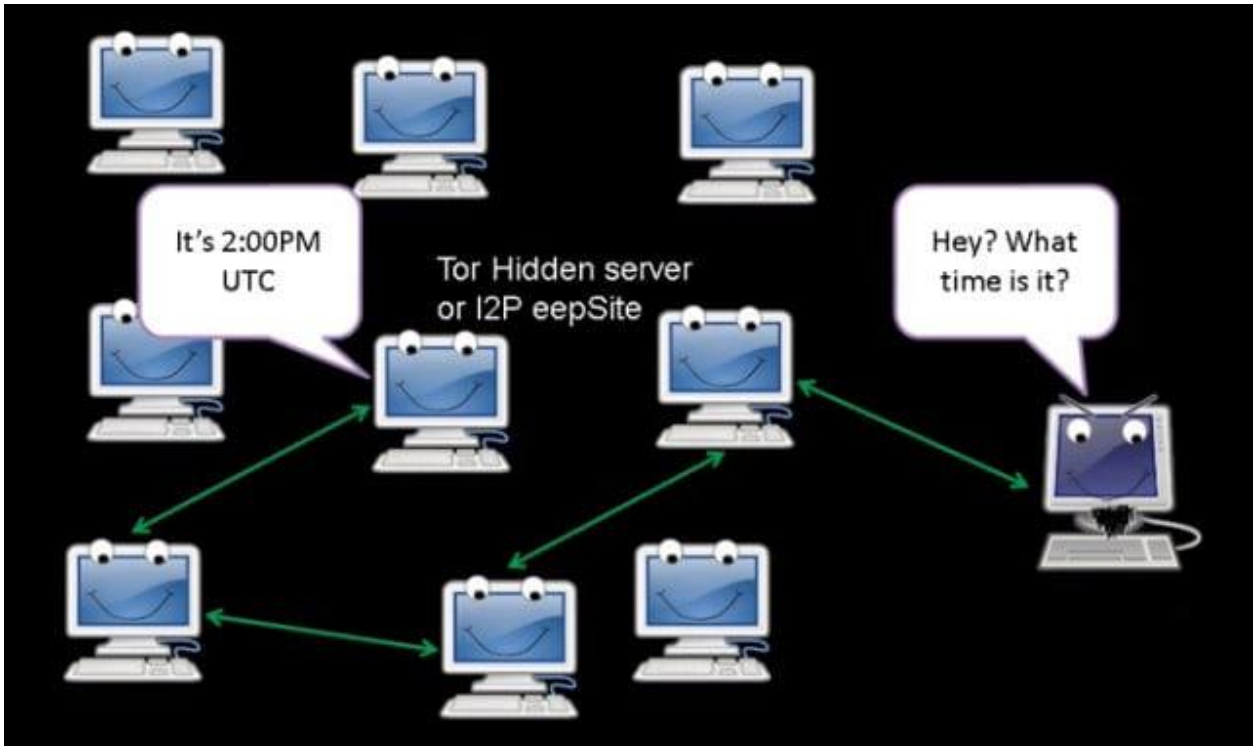


Figure 14 - Clock-based Vulnerability

To exploit the clock-based vulnerability, an attack could correlate the data found in the DHT about node connections and responses from hidden service websites. For example, a malicious actor would log every Tor or I2P user possible from the DHT and then determine if they are running a website on their node. This actor could then collect information about the running web services and the time on their clock for the user nodes that have websites. With this data, the malicious actor can then query a list of hidden service websites, such as an eepsite or onion site, for their time and see if any of the responses match with the information they have on user nodes. Through this method, a malicious actor can determine which nodes are running which websites.

Fortunately, attacks exploiting the clock-based vulnerability are rare. Network jitter introduces variation in the time difference between nodes, which makes it difficult to reliably discern the local time of a node. Some mitigations include setting clocks with a consistent and regularly used NTP server and others are implemented within a darknet protocol itself.

**Sybil attacks** are an exploit of a vulnerability present in Tor and I2P. Essentially, an attacker will set up multiple nodes that appear to be completely separate users in the network, but are in fact controlled by the attacker. This exploit is not necessarily an attack itself, but it can make other attacks much easier because of multiple colluding nodes.

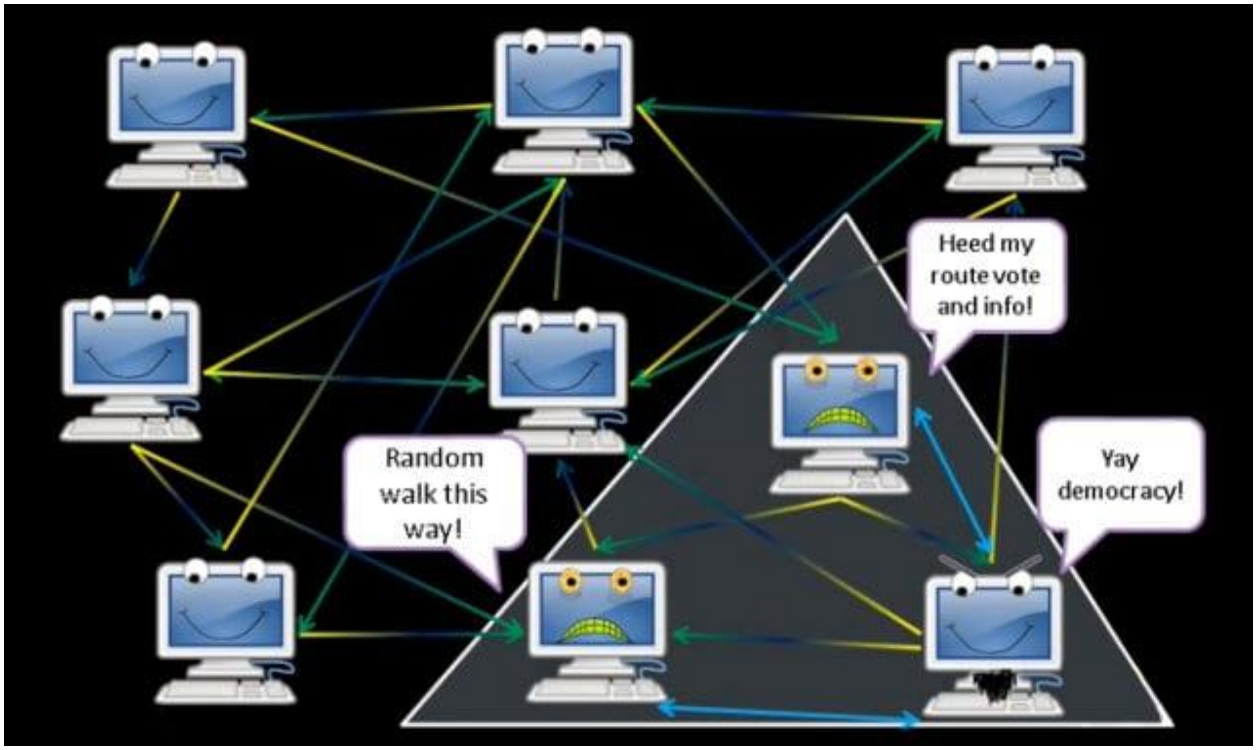


Figure 15 - Sybil Attack in a Network

The figure above visualizes the impacts of a sybil attack. The attacker creates two other nodes that are under their command and therefore they control a larger part of the network. One of the possible effects of a sybil attack is routing control, which can enable the attacker to redirect traffic to malicious exit nodes that log the unencrypted messages.

There are a few mitigation strategies to sybil attacks, but none of them are absolute fixes. One strategy is to make it computationally expensive to have more nodes in the network. A good example of this is in decentralized networks that use the Proof of Work (PoW) consensus algorithm such as Bitcoin. Another strategy, that is implemented in both Tor and I2P, is to restrict IP addresses in the same subnet from being in consecutive hops. The idea behind this is that colluding nodes would most likely have their own IP network.

A **correlation attack** is an exploit of a vulnerability that affects both I2P and Tor. Correlation attacks are a broad term for a group of techniques that use the network data available about specific nodes to infer the identity of a user. The previously mentioned clock-based vulnerability can be exploited by a correlation attack. This type of attack may not outright identify a user, but it is useful for narrowing down the possible identities of certain nodes in the network.

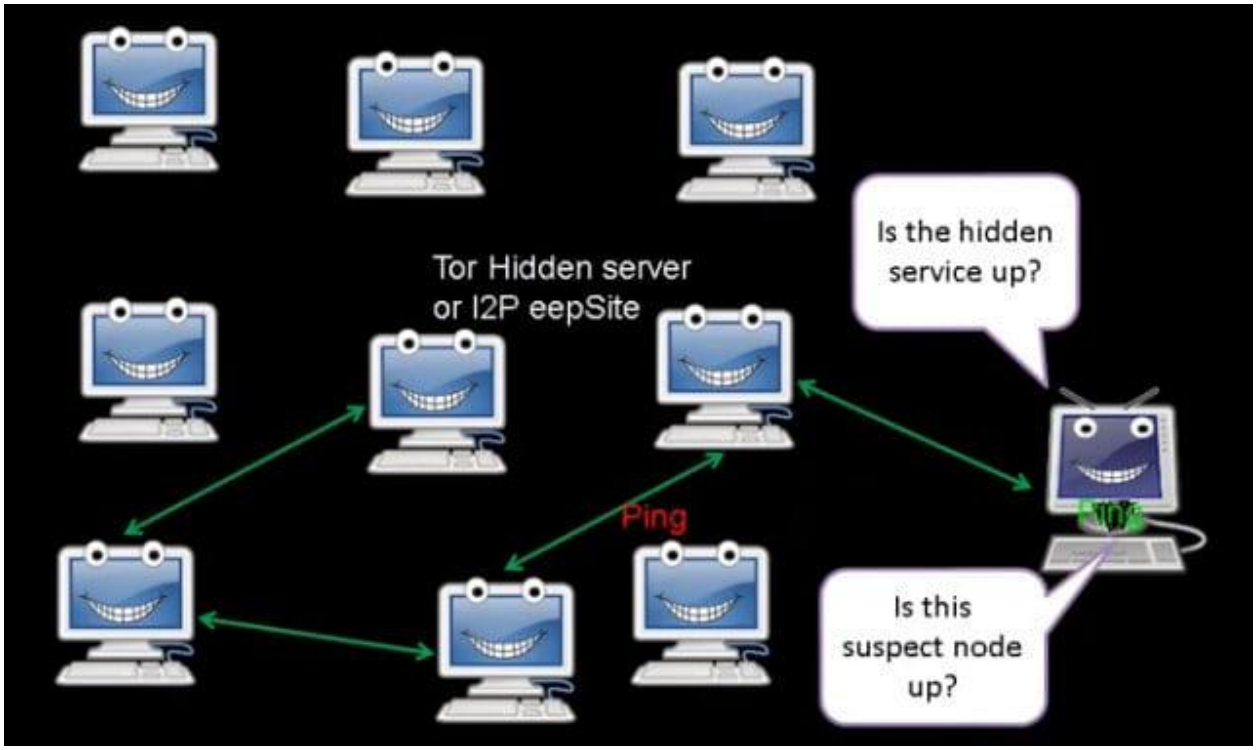


Figure 16 - Correlation Attack in a Network

Figure 16 displays a simple example of a correlation attack. In this example, an attacker selects a Tor onion website or I2P eepsite to surveil. If this site is down, the attacker can then check every node in the network to see if any of them are offline. This offline node can then be inferred to be the host of the hidden service website. Evidently, this type of correlation attack only works well in small networks with not many nodes.

Mitigations for correlation attacks include increasing the number of nodes in the network and limiting the amount of data available to nodes about the network. Increasing the number of nodes is a simple solution that is very effective because it raises the computational requirements to scan through every node. However, this is easier said than done as scaling up darknets introduces its own set of issues. The other mitigation is limiting how much data is given to nodes about the network. Tor and I2P both have methods to limit the amount of data for protection against correlation attacks.

**DDoS Starvation Attack** is a vulnerability exploit specific to I2P. A hostile user can create a significant number of nodes in the I2P network. These nodes each appear as separate entities, but are actually controlled by one user and as such this attack may be used in combination with a sybil attack. The hostile user can then direct the nodes to not provide any resources to the network. This will cause neighboring nodes to search through a larger network database or request the creation of more tunnels than is actually needed, effectively starving the network and limiting usability.

I2P has a mitigation strategy to this attack. The network maintains profiles on nodes, which classify them based on their network performance. These profiles help direct forwarding decisions by identifying underperforming nodes and either limiting their usage as forwarding paths or avoiding them entirely.

**DDoS Flooding Attack on Tor Directory Servers** exploits the vulnerability with Tor's centralized directory servers. As previously mentioned, almost all of the Tor network is completely decentralized except for its directory servers, which maintains status data on each node in the network. A general flooding attack is, as it sounds, transmitting a very large amount of traffic to a network server for the purpose of overloading it and rendering it unusable for legitimate users. Subsequently, when such an attack is applied to the Tor network at one or all of its directory servers, it will become increasingly unusable. These attacks can be directed at specific directory servers in a certain location to restrict Tor's usability for the users in that area. Only Tor is susceptible to this attack due to its centralized directory servers. I2P distributes its network metadata across the network so it is protected from flooding attacks.

These are just some of the vulnerabilities and exploits that are capable of eliminating the anonymity and usability of darknets. Many other attacks exist and more vulnerabilities are being discovered that malicious actors or government agencies can exploit to identify users. And as such, it is important to consider the future of darknet technology, not only in the context of cybersecurity, but also in terms of its scalability and usability.

# 5

# 5. Future of the Dark Internet

Dark Internet Technology will only continue to improve throughout the future. As computers increase in computational speed and network bandwidth improves, darknets will not only get faster, but also more secure.

The network encryption that is foundational to darknets requires fast processors, which can quickly execute the algorithms to encode messages for transmission. Since computer chips are significantly more powerful than before, this process is no longer the limiting factor for darknet scalability and speed. And as such, this increase in computational power can enable the application of stronger encryption methods to further secure darknet communications. Advancements in network transmission technology can also aid the scalability and speed of darknets since they fundamentally rely upon the public Internet networks.

Nevertheless, darknets still face several technical challenges that limit their overall scalability, dependability, and usability:

1. Decentralized networks are slower than traditional networks with centralized servers as data is distributed across many nodes. This is a limitation of darknet architecture, but increases in transfer speed will help alleviate this issue.
2. Darknets are more secure than traditional networks due to the inherent benefits in decentralization, but network security now requires a completely different way of thinking. Rather than a few centralized points of network vulnerability, darknets can have many vulnerable points distributed across the network.
3. Darknets do not have their own physical network and depend upon an underlying public network. This dependence introduces some risks as weaknesses present in the underlying network can affect the darknet.
4. UIs of darknet software have historically lacked usability as this software was developed by a small group of people. This issue has improved over time with the Tor Browser being highly usable, but there is still much work to be done.

Darknets not only face technical challenges, but they also face societal issues that limit their adoption into the mainstream. Primarily, darknets enable the perpetuation of cybercrimes as they offer anonymous communications to criminals who desire to conceal their identity. However, these cybercriminals represent only a small percentage of the total darknet user base and there is significant interest in darknet technologies from business groups and government agencies that seek to secure their critical communications.

Ultimately, regardless of the technical challenges and societal issues present in darknet technologies, future developments and adoption of darknets looks encouraging. Of course, many of the darknet advancements would stem from technological improvements in the underlying public networks, including increases in computational power and network transmission upgrades. Anonymous communications are one of the most critical components of darknets and ensuring this anonymity will be a challenge as network complexity grows, introducing more points of failure for attackers to exploit.

# Acronyms

| | |
|---|---|
| API | Application Programming Interface |
| DDoS | Distributed Denial of Service |
| DHT | Distributed Hash Table |
| DNS | Domain Name System |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| IP | Internet Protocol |
| KBR | Key-based Routing |
| NTP | Network Time Protocol |
| PoW | Proof of Work |
| SOCKS | Socket Secure |
| SSL | Secure Socket Layer |
| TCP | Transmission Control Protocol |
| TLS | Transport Layer Security |
| UDP | User Datagram Protocol |
| VPN | Virtual Private Network |

# References


i. Bassel AlKhatib, Randa Basheer, "*Crawling the Dark Web: A Conceptual Perspective, Challenges and Implementation*", JDIM.
ii. https://www.crowdstrike.com/cybersecurity-101/the-dark-web-explained/
iii. Peter Biddle, Paul England, Marcus Peinado, Bryan Willman, "*The Darknet and the Future of Content Distribution*", Microsoft Corporation.
iv. Michael Reed, Paul Syverson, David Goldschlag, "*Anonymous Connections and Onion Routing*", IEEE.
v. https://en.wikipedia.org/wiki/Dark_web
vi. https://en.wikipedia.org/wiki/Darknet
vii. https://en.wikipedia.org/wiki/Tor_(network)
viii. https://en.wikipedia.org/wiki/Onion_routing
ix. Ian Clarke, Oskar Sandberg, Brandon Wiley, Theodore Hong, "*Freenet: A Distributed Anonymous Information Storage and Retrieval System*"
x. https://arstechnica.com/information-technology/2015/01/under-the-hood-of-i2p-the-tor-alternative-that-reloaded-silk-road/
xi. https://geti2p.net/en/docs/how/tech-intro
xii. https://2019.www.torproject.org/about/overview.html.en
xiii. Matthew Wright, Micah Adler, Brian Neil Levine, "*The Predecessor Attack: An Analysis of a Threat to Anonymous Communications Systems*", ACM.
xiv. https://geti2p.net/en/docs/how/threat-model
xv. Peter Likarish, "*Tor Anonymity Network & Traffic Analysis*"
xvi. Roger Dingledine, Nick Mathewson, Paul Syverson, "*Tor: The Second-Generation Onion Router*", USENIX.
xvii. https://www.socinvestigation.com/the-tor-architecture-and-its-inherent-security-implications/
xviii. https://privacy-pc.com/articles/common-darknet-weaknesses-an-overview-of-attack-strategies.html