Privacy Enhancing Technologies (BSc)

Lecture IX: Tor

Univ.-Prof. Dr. Dominique Schröder

January 20, 2025

Overview

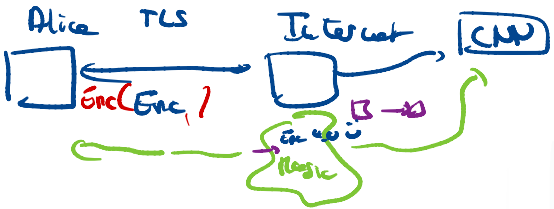
KW	Type	Lectures
42	**Lecture**	Internet and Privacy in the Web
43	**Lecture**	Fingerprinting & Censorship
44	**Exercise**	Introduction to Rust & Challenge Start
45	**Lecture**	Cryptographic Preliminaries
46	**Lecture**	Public Key Cryptography and Security Reductions
47	**Exercise**	Proofs and Theoretical Deepening
48	**Lecture**	TLS
49	**Exam**	**Midterm Exam**
50	**Exercise**	Challenge Hands-On
51	**Lecture**	Messaging I
3	**Lecture**	Messaging II (Forward Security, Post Compromise Security)
4	**Lecture**	**Tor**
5	**Lecture**	Q & A
6	**Exam**	**Final Exam**
9	**Exam**	**Reexam** (Both for Midterm and Final)

This Lecture's Agenda

- What is Tor?
- Core components of Tor
- How Tor Works
- Tor Hidden Services
- Stories about Tor

What is Tor?

What is Tor?



- Tor is a worldwide overlay anonymity network designed to hide users' IP addresses and provide online privacy.
- Approximately **2-5 million users** access Tor daily.
- Built on the principle of **onion routing**, where data is **encrypted in multiple layers** before being transmitted through a series of relays.

History and Audience of Tor

History:

- Original paper: *"Tor: The Second-Generation Onion Router"* (Usenix Security 2004).

Who Uses Tor?

- Journalists needing secure communication.
- Law enforcement for undercover operations.
- Academics and researchers studying online privacy.
- IT security professionals monitoring anonymity networks.
- CAPTCHA enthusiasts bypassing tracking mechanisms =).

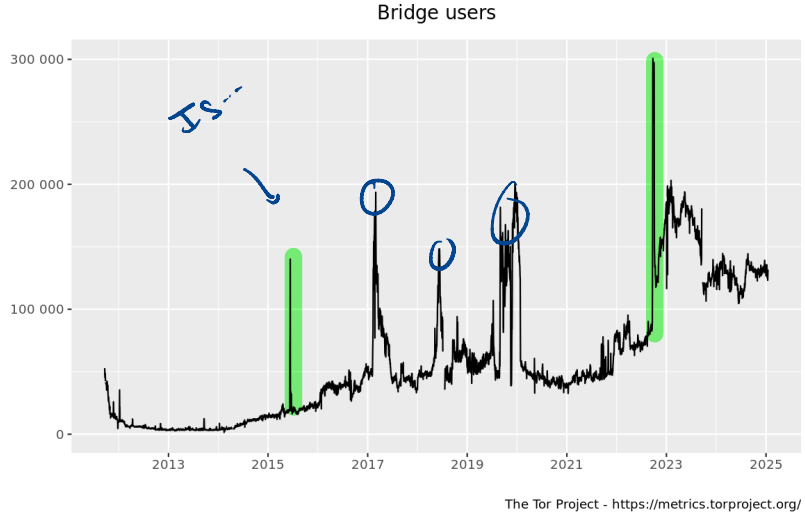
Tor in Numbers

- Approximately **1.95 million daily users** (as of late 2024).
- **7,500+ relays** operate worldwide, including **2,500+ exit relays**.
- Tor handles **hundreds of gigabits per second** of encrypted traffic.
- The network's structure and traffic data are *openly available* through Tor Metrics.

Major user locations:

- Russia: **63,161** (**48.39%**)
- Iran: **13,648** (**10.46%**)
- United States: **11,493** (**8.80%**)
- Germany: **4,668** (**3.58%**)
- China: **2,709** (**2.08%**)

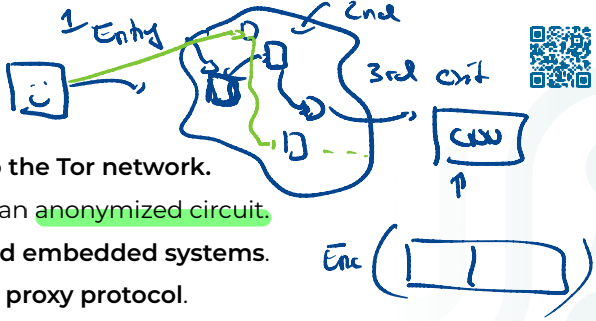
Using Bridges to Avoid Censorship



Core Components of Tor

Core Components of Tor

Tor Client



The Tor Client is the entry point for users into the Tor network.

- Acts as a **proxy** that routes traffic through an anonymized circuit.
- Available on **desktops, mobile devices, and embedded systems**.
- Supports applications that use the **SOCKS proxy protocol**.

How It Works:

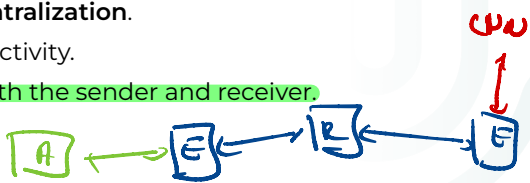
- Establishes an encrypted connection to a randomly selected **entry node**.
- Dynamically builds a **three-hop circuit** through the network.
- Periodically **rebuilds circuits** to prevent long-term correlation attacks.
- Encrypts data in multiple layers using **onion encryption**.

Core Components of Tor

Tor Relays

Tor Relays are volunteer-operated nodes that forward encrypted traffic across the Tor network.

- They increase **anonymity, security, and decentralization**.
- Unlike VPNs, relays **do not store logs** of user activity.
- They ensure that no single entity can track both the sender and receiver.



Key Characteristics:

- Traffic is **encrypted multiple times** before reaching its destination.
- Each relay in the circuit only knows its **direct neighbor**, not the full path.
- More participating relays improve the **network's speed and anonymity**.
- Volunteers contribute bandwidth to strengthen the **resistance to censorship**.

Core Components of Tor

Directory Authority (1)

The **Directory Authority** are special trusted servers that manage and maintain the Tor network's directory.

- There are **only 9 Directory Authorities** worldwide, operated by trusted entities.
- These servers store and publish the **network consensus**, which lists all active nodes and their attributes.
- Clients download this consensus when connecting to ensure they use an **up-to-date network map**.
- **Voting occurs every hour** to determine which nodes are included.

Core Components of Tor

Directory Authority (2)

How Directory Authorities Work:

- Each **node** in the system self-registers with these trusted servers.
- The authorities perform checks to **validate uptime, functionality, and compliance**.
- They prevent **malicious actors** from introducing untrusted infrastructure.
- Nodes marked as **"bad"** (due to censorship circumvention abuse, instability, or compromise) are removed from the list.

Security and Redundancy:

- Directory Authorities use **digital signatures** to prevent tampering.
- The system is **decentralized**—no single entity can fully control it.
- Even if some of these authorities go offline, the network can continue functioning.

Core Components of Tor

Tor Cells

Tor Cells are fixed-size packets used for communication within the Tor network.

- Each cell is **512 bytes** in size, ensuring uniform traffic flow.
- Cells are the fundamental units used for routing encrypted data through relays.
- Each cell consists of a **header** and a **payload**.

Header (5 Bytes)	**Payload** (507 Bytes)

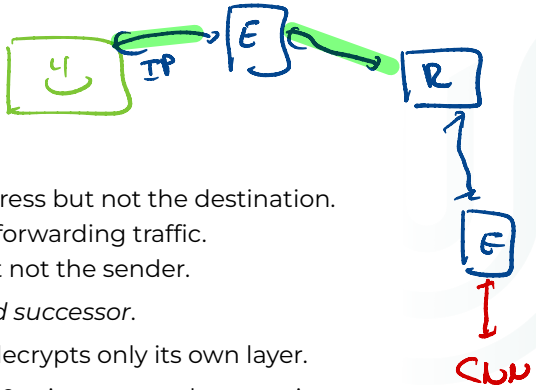
Total Size: **512 Bytes**

Types of Cells:

- **Relay Cells**: Carry encrypted traffic between clients and destinations.
- **Control Cells**: Used for circuit management and key negotiation.

Core Components of Tor

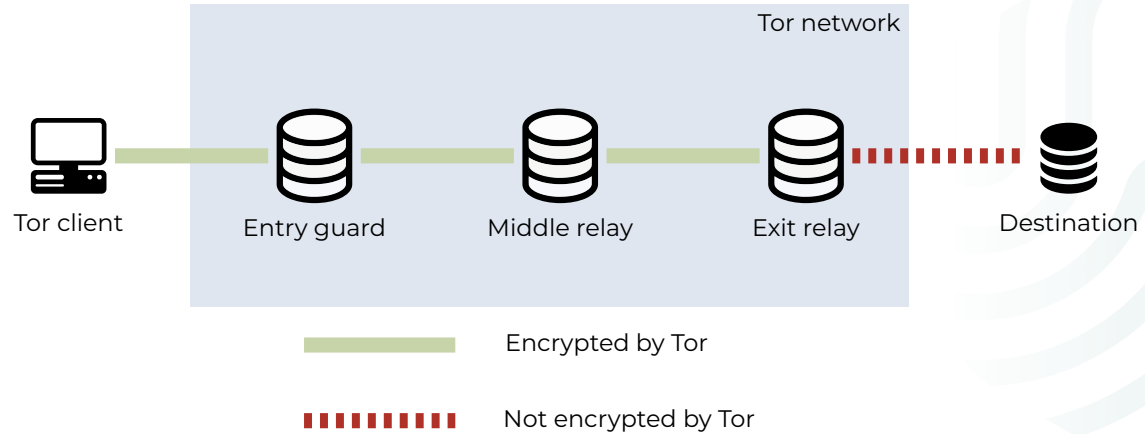
Tor Circuits (1)



- A *Tor circuit* consists of three relays:
  - **Entry Guard**: Knows the user's IP address but not the destination.
  - **Middle Relay**: Ensures anonymity by forwarding traffic.
  - **Exit Relay**: Knows the destination but not the sender.
- Each relay *only knows its predecessor and successor*.
- Traffic is *encrypted in layers*—each relay decrypts only its own layer.
- Circuits are *randomly rebuilt* after about 10 minutes to enhance privacy.

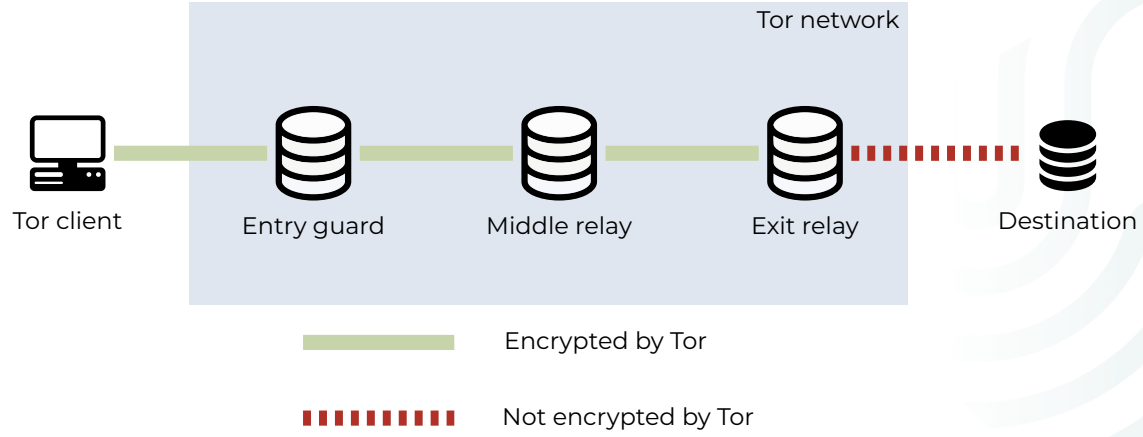
Core Components of Tor

Tor Circuits (2)



How Tor Works

A TOR Circuit



Establishing a Tor Circuit

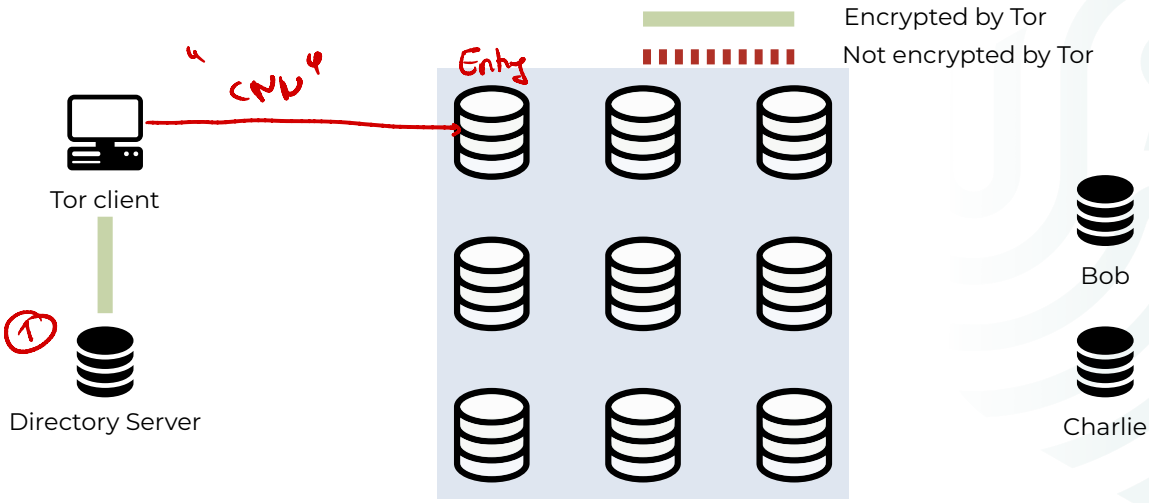
Downloading the Network Consensus (1)

The first step in using Tor is retrieving information about available nodes.

- Alice's **Tor client contacts a Directory Authority** to fetch the latest list of Tor nodes.
- The **Directory Authority** maintains a signed "network consensus" file with:
  - IP addresses and public keys of available nodes.
  - Relay bandwidth and uptime statistics.
  - Information about node roles (e.g., guard, middle, exit).
- This ensures Alice's client has an **updated, trusted list** of relays before building a circuit.

Establishing a Tor Circuit

Downloading the Network Consensus (2)



Establishing a Tor Circuit

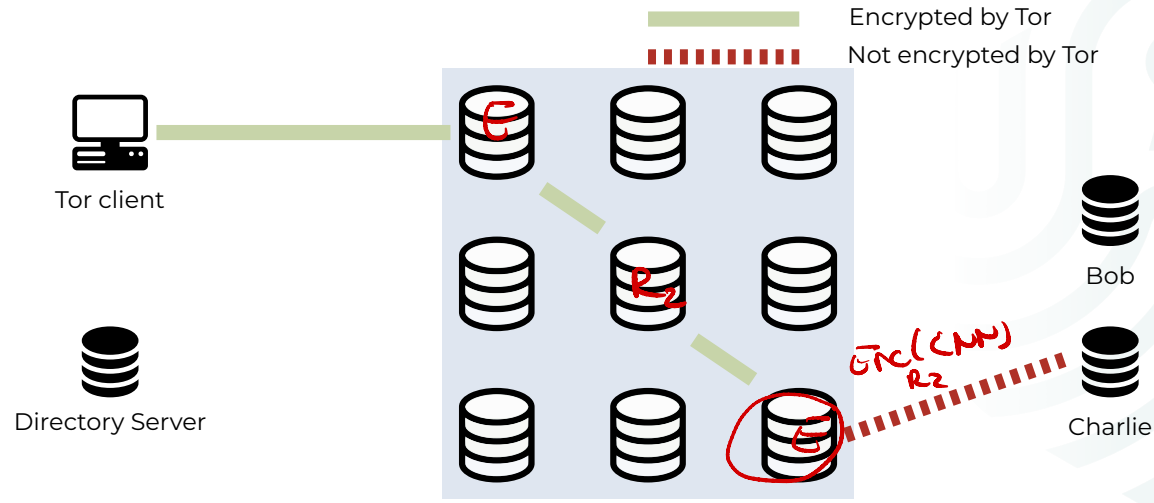
Creating the Circuit

Alice's Tor client builds a circuit through the network.

- The client selects **three random nodes** from the list:
  - **First Node (Entry)**: Knows Alice's real IP address but not the destination.
  - **Second Node (Middle)**: Ensures no single node knows both Alice and her destination.
  - **Third Node (Exit)**: Knows the destination but not Alice's IP.
- The client negotiates a **separate AES encryption key** with each node.
- This ensures **each relay only decrypts one layer** before forwarding the data.
- The path remains fixed for **about 10 minutes** before a new circuit is created.

Establishing a Tor Circuit

Creating the Circuit (2)



Establishing a Tor Circuit

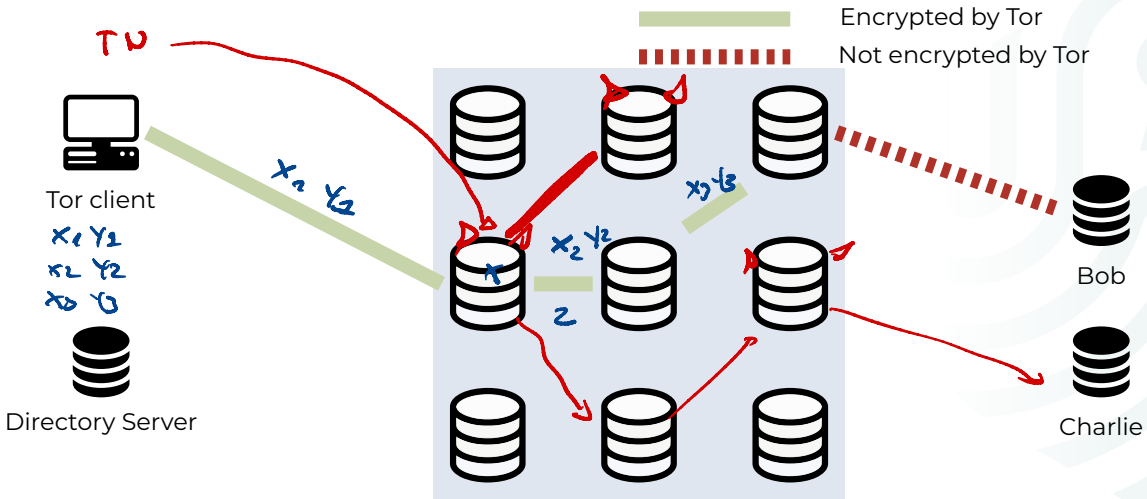
Sending the Request (1)

Once the circuit is built, Alice's request reaches the destination.

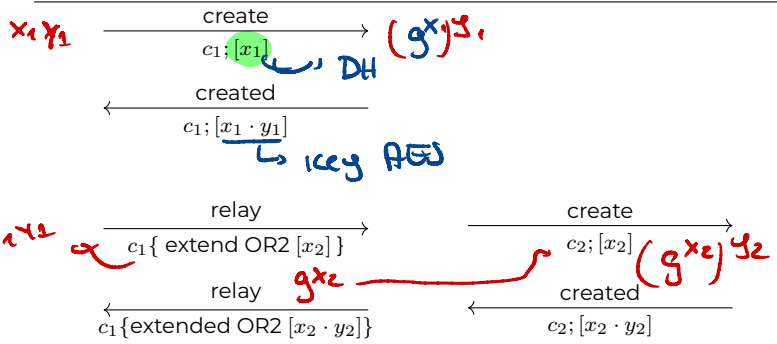
- Alice's encrypted request travels through the circuit.
- The **Exit Node decrypts the final layer** and sends the request to the website.
- The website sees the **Exit Node's IP address** instead of Alice's.
- When the website responds, data is sent **back through the same circuit**, getting encrypted at each hop.
- Eventually, the response reaches Alice **fully decrypted** at the last step.
- A different circuit may be chosen for the next request to **improve anonymity**.

Establishing a Tor Circuit

Sending the Request (2)

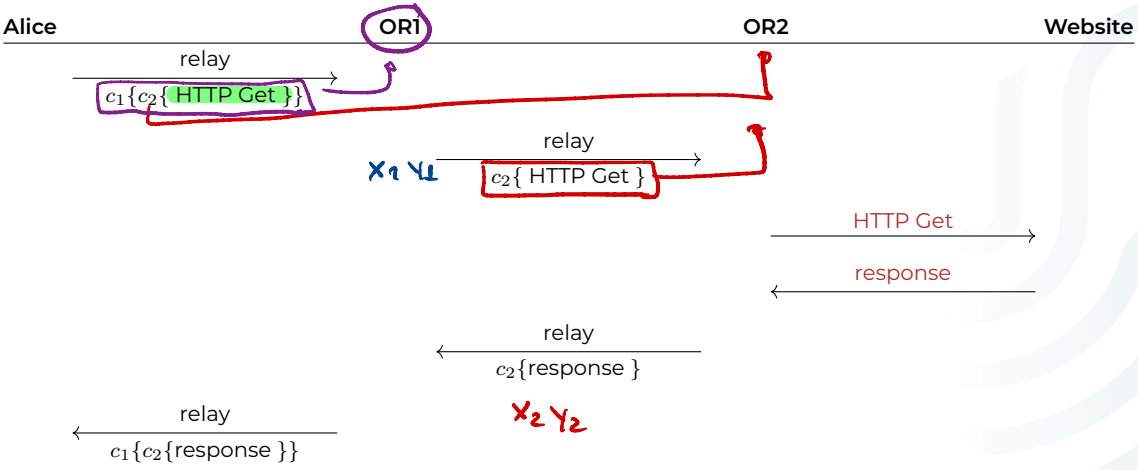


Establishing the Circuit



- Link between Alice-OR1 and OR1-OR2 are TLS encrypted.
- $c_i\{\}$ means, that the content is AES-encrypted under the symmetric key of the channel c_1.

Accessing the Website



- Link between OR2 and Website might not be TLS encrypted.

Tor Hidden Services

Tor vs. Tor Hidden Services

Standard Tor Usage:

- The client connects to a **server** via an anonymous Tor circuit.
- The client's **IP address is hidden**, but the destination server is still known.
- Used primarily for **anonymous browsing** (e.g., accessing websites without revealing the user's IP).

Tor Hidden Services (Onion Services):

- Both the **client and the server** remain anonymous.
- Communication happens through **rendezvous points**, preventing direct exposure of IPs.
- The website is hosted within the **Tor network**, with an address ending in `.onion`.
- Used for **privacy-focused services**, including secure messaging, anonymous marketplaces, and censorship-resistant websites.

Tor Hidden Services (1)

Establishing the Hidden Service (1)

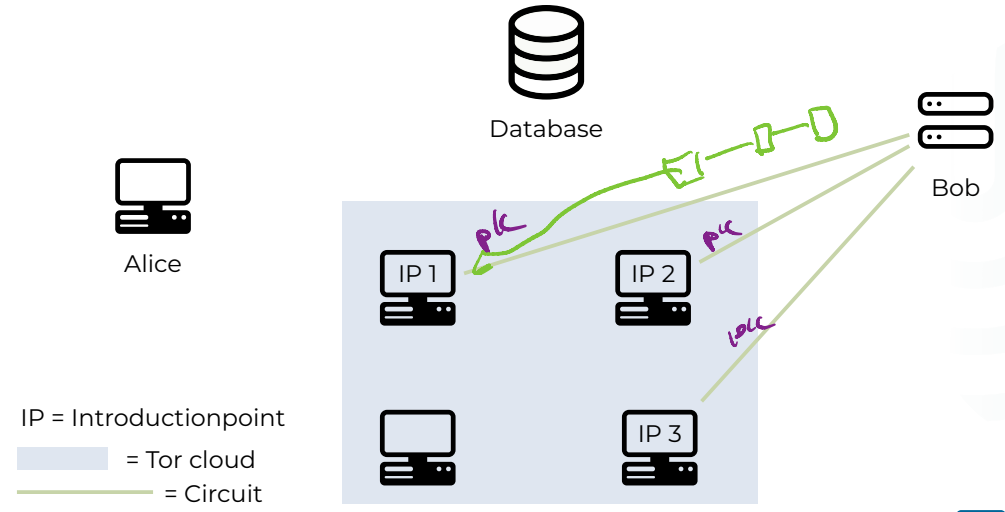
- A **server** that wants to provide a hidden service selects several Tor nodes as **introduction points**.
- The server establishes **encrypted circuits** to these introduction points.
- The server **generates a public-private key pair** and shares its **public key** with the introduction points.
- The introduction points remain **passive**, waiting for connection requests from clients.
- The server **publishes its .onion address** (generated using a hash function from its public key) to the **distributed hash table (DHT)**.

Purpose of This Step:

- Ensures the **service remains anonymous** and does not expose its IP.
- Allows users to find the service using its **unique .onion identifier**.
- Prepares the system for **secure client connections** via rendezvous points.

Tor Hidden Services (1)

Establishing the Hidden Service (2)



Tor Hidden Services (2)

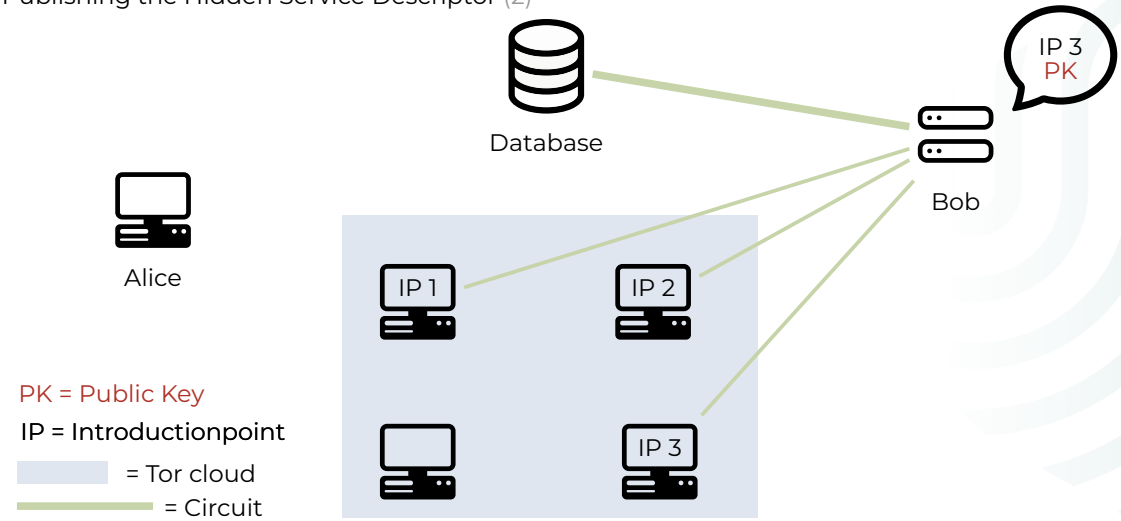
Publishing the Hidden Service Descriptor (1)

- The hidden service **creates a descriptor** containing:
  - Its **public key**.
  - A list of **introduction points** (Tor relays chosen in Step 1).
  - A **cryptographic signature** to verify authenticity.
- This descriptor is uploaded to a **distributed hash table (DHT)**.
- The **descriptor is signed with the hidden service's private key** to prevent tampering.
- The .onion address is derived from the **hash of the public key**, resulting in a **16-character name** (e.g., `xyz.onion`).
- Clients looking for the hidden service **query the DHT** to obtain the service descriptor.

Once the Descriptor is uploaded, the Hidden Service is discoverable by clients.

Tor Hidden Services (2)

Publishing the Hidden Service Descriptor (2)



Tor Hidden Services (3)

Publishing the Hidden Service Descriptor (2)

Why is This Step Important?

- Ensures that **only the legitimate service** can register its introduction points.
- Prevents **man-in-the-middle attacks** by verifying the descriptor signature.
- The **distributed nature of the DHT** enhances **resilience** against censorship.

Key Security Enhancements in v3 Onion Services:

- Uses **Ed25519 elliptic-curve cryptography** for stronger authentication.
- The **longer address (56 chars)** increases resistance to enumeration attacks.
- Improved **descriptor encryption** prevents leakage of metadata.
- Uses **SHA3-based public key hashing** for better security.

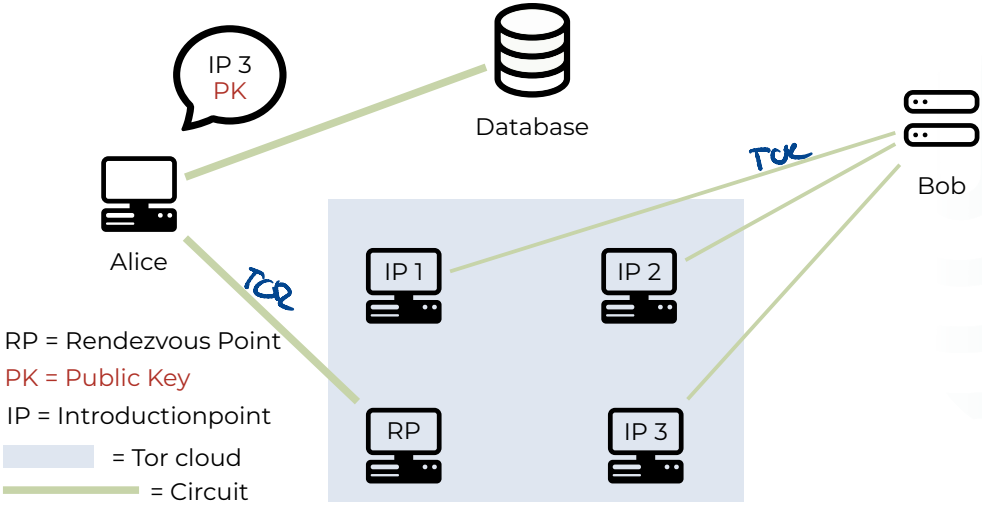
Tor Hidden Services (3)

Client Discovery and Connection Initialization (1)

- The hidden service **advertises its descriptor** (including its introduction points) to the **distributed hash table (DHT)**.
- A client must **know the .onion address** of the service to initiate a connection.
- The client retrieves the **service descriptor** from the DHT, which includes:
  - The **public key** of the service.
  - A list of **available introduction points**.
  - A cryptographic **signature verifying authenticity**.
- The client selects an **introduction point** and sends a **request encrypted with the service's public key**.
- The client builds a separate Tor circuit to a randomly chosen **relay** and asks it to act as a **rendezvous point**.
- The client generates and shares a **one-time secret** with the rendezvous point.

Tor Hidden Services (3)

Client Discovery and Connection Initialization (2)



Tor Hidden Services (3)

Client Discovery and Connection Initialization (3)

Why This Step Matters:

- Ensures the service remains **hidden and anonymous**.
- Uses **end-to-end encryption** to prevent eavesdropping.
- The **one-time secret** prevents replay attacks and unauthorized connections.
- Clients and services never communicate directly—**rendezvous points** maintain privacy.

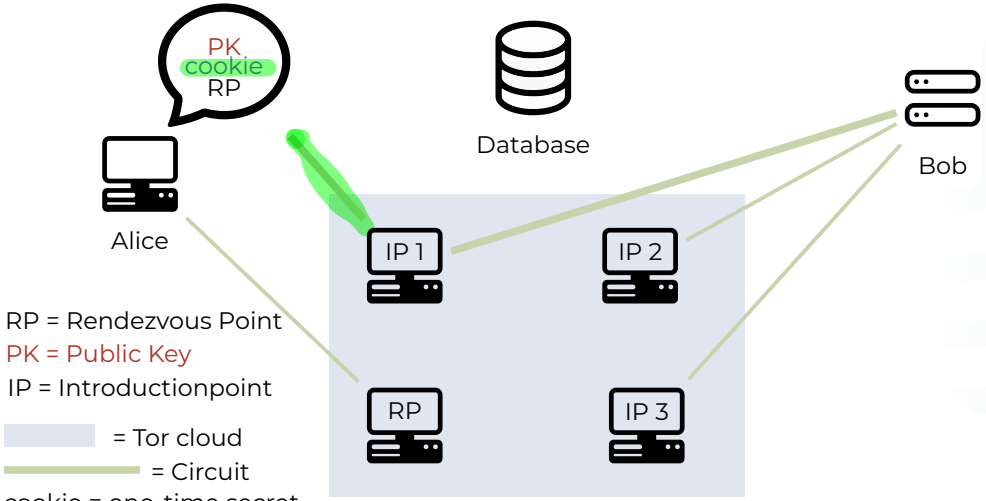
Tor Hidden Services (4)

Client Sends Introduction Message (1)

- When the **service descriptor** is available and the **rendezvous point** is ready, the client prepares an **introduction message**.
- This message is **encrypted with the hidden service's public key** and contains:
  - The **address of the rendezvous point**.
  - The **one-time secret** for authentication.
- The client sends this encrypted introduction message to one of the **introduction points**, requesting it to be relayed to the hidden service.
- The hidden service, upon receiving the introduction message, **decrypts it using its private key** and learns the client's **rendezvous point**.
- The hidden service then creates its own **Tor circuit to the rendezvous point** to establish a two-way communication channel.

Tor Hidden Services (4)

Client Sends Introduction Message (2)



Tor Hidden Services (4)

Client Sends Introduction Message (3)

Why This Step Matters:

- Communication remains **fully encrypted and anonymous**.
- The client and hidden service never communicate **directly**, only through Tor circuits.
- The hidden service **cannot see the client's IP address**, and vice versa.
- Ensures that only an **authenticated client** can request a connection.

Tor Hidden Services (5)

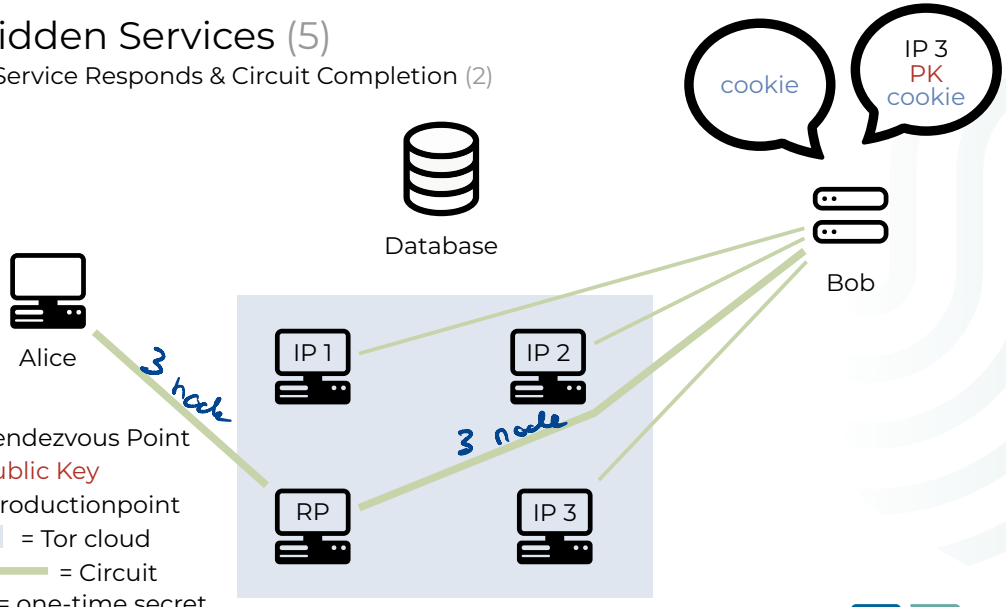
Hidden Service Responds & Circuit Completion (1)

- The hidden service **decrypts the introduction message** using its private key.
- It extracts:
  - The **address of the rendezvous point**.
  - The **temporary session key** (shared secret) for communication.
- The hidden service then builds a **Tor circuit to the rendezvous point**.
- It transmits the **temporary session key** to the rendezvous point, signaling readiness.
- The client and hidden service now both have established **separate Tor circuits** to the same rendezvous point.

→ **The rendezvous point is now ready to relay data between the client and the hidden service.**

Tor Hidden Services (5)

Hidden Service Responds & Circuit Completion (2)



Tor Hidden Services (6)

Data Exchange Through the Rendezvous Point (1)

- If the **rendezvous circuit** is established successfully, the **rendezvous point notifies the client**.
- The rendezvous point acts as a **passive relay**, forwarding **fully encrypted messages** between the client and the hidden service.
- Neither the client nor the service ever communicates **directly**—all traffic passes through **the rendezvous point**.

Tor Hidden Services (6)

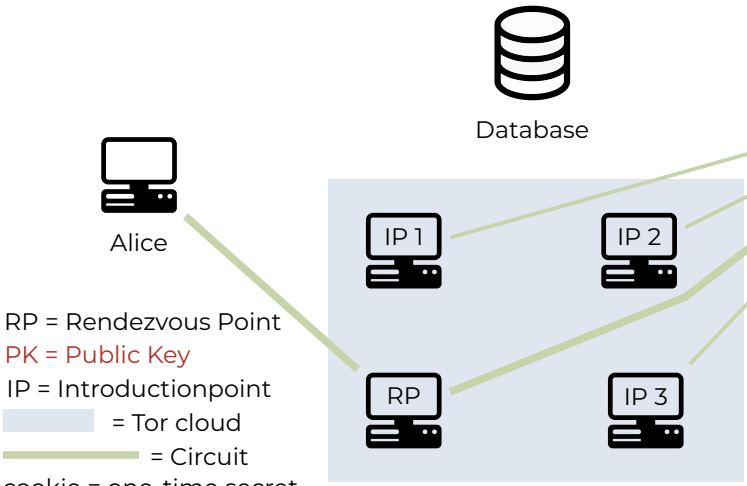
Data Exchange Through the Rendezvous Point (2)

Circuit Structure:

- The complete communication path consists of **six Tor relays**:
  - **Three nodes chosen by the client:**
    - The first node (Entry Guard) sees the client's IP but not the destination.
    - The middle relay forwards traffic.
    - The third relay (Rendezvous Point) is where the circuits from the client and service meet.
  - **Three nodes chosen by the hidden service:**
    - The hidden service selects its own three Tor relays.
    - The final relay in its circuit is also the rendezvous point.
- At no point does a single relay know both the client and the hidden service's identity.

Tor Hidden Services (6)

Data Exchange Through the Rendezvous Point (3)



Security of Tor

Security of Tor

Overview of Tor's Security Model

Tor aims to provide anonymity by obscuring the link between users and their online activities.

- Uses **onion encryption** to protect communication across multiple hops.
- Employs **directory authorities** to mitigate Sybil attacks.
- Relies on **distributed, volunteer-operated nodes** to decentralize control.
- Circuits are **dynamically changed** to minimize traffic analysis risks.

Security Goals:

- Prevents websites from discovering a user's **real IP address**.
- Protects against **network surveillance** and traffic analysis.
- Enables **censorship-resistant** access to information.
- Ensures end-to-end **encryption within the Tor network**.

Security of Tor

Threat Model & Limitations

Tor's security is based on assumptions about the adversary's capabilities.

- **Entry node risk:** The first relay knows the user's real IP address.
- **Exit relay exposure:** The final relay sees the unencrypted destination.
- **Traffic correlation attacks:** Analyzing packet timing and volume can de-anonymize users.
- **Sybil attacks:** Malicious entities may introduce many relays to deanonymize traffic.
- **Global passive adversaries:** Entities monitoring large parts of the internet may correlate traffic flows.

Key Considerations:

- **HTTPS is necessary** to protect data integrity through the exit node.
- **Guard nodes reduce** the risk of passive monitoring attacks.
- Tor does **not protect against endpoint compromise** (e.g., malware).

Security of Tor

Common Attacks on Tor

Tor faces various attack vectors that can compromise anonymity:

- **Traffic Correlation Attacks:**
  - Observing packet timing and volume at entry and exit nodes.
  - Possible with adversaries controlling large portions of the network.
- **Sybil Attacks (Large-Scale Node Injection):**
  - Malicious actors deploy numerous Tor relays to increase chances of controlling entry and exit nodes.
  - The more relays controlled, the greater the likelihood of tracking connections.
- **End-to-End Confirmation Attacks:**
  - An attacker operates both an entry and exit relay and correlates traffic.
  - Used to de-anonymize hidden services or targeted users.
- **Website Fingerprinting:**
  - Even if encrypted, unique traffic patterns can reveal visited websites.
  - Possible countermeasures: padding and traffic obfuscation.

Security of Tor

Defenses Against Attacks

Tor employs several mechanisms to protect user anonymity:

- **Guard Nodes:**
  - Users stick to a few long-term entry nodes, making traffic correlation harder.
  - Reduces exposure to malicious entry relays.
- **Rendezvous Points for Hidden Services:**
  - Prevents direct connections between clients and services.
  - Avoids revealing IP addresses of both parties.
- **Padding and Traffic Obfuscation:**
  - Standardizes packet sizes to prevent fingerprinting.
  - Future improvements include circuit-level padding mechanisms.
- **Encrypted Directory Authority Communications:**
  - Prevents adversaries from easily listing all available Tor relays.
  - Makes it harder to execute large-scale Sybil attacks.

Security of Tor

Best Practices for Tor Users

Users can take additional measures to strengthen their anonymity:

- Always use **HTTPS** when accessing websites over Tor.
- Disable **JavaScript and browser plugins** (e.g., Flash, WebRTC).
- Use **Tails OS** or a **hardened Linux environment**.
- Avoid logging into personal accounts (e.g., Gmail, Facebook) over Tor.
- Be cautious when downloading and opening files while connected to Tor.
- Consider using **Tor bridges** to bypass censorship in restricted areas.

Key Takeaway: Tor is only as strong as its users' operational security.

Stories About Tor

Stories About Tor

1. Operation Bayonet – The Fall of AlphaBay

- AlphaBay was the largest darknet marketplace, offering drugs, weapons, and stolen data.
- In 2017, law enforcement agencies took it down in a coordinated global operation.
- The FBI exploited an unencrypted login page to deanonymize its administrator.
- 

2. The Norwegian Police's Undercover Investigation

- Journalists from VG uncovered a police sting operation in child abuse forums on the dark web.
- Norwegian officers infiltrated the forums, collected evidence, and dismantled criminal networks.
- The case highlighted how law enforcement can use Tor for undercover investigations.
- 

Stories about Tor

3. SecureDrop – Whistleblower Protection

- Many news organizations, including The Guardian and The Washington Post, use SecureDrop.
- SecureDrop allows whistleblowers to anonymously submit sensitive information via Tor.
- Used by sources like Edward Snowden to expose mass surveillance programs.
- 

4. Tor in Authoritarian Regimes

- Tor has played a crucial role in bypassing censorship in countries like Iran and China.
- Activists use it to communicate securely and access restricted information.
- Some governments attempt to block Tor, leading to the development of Tor bridges.
- 

Feedback

Your *anonymous feedback* is invaluable to us, and we are truly grateful for your time and insights.

Thank you for helping us improve!

https://tuwel.tuwien.ac.at/mod/feedback/view.php?id=2459264

PRIVACY
ENHANCING
TECHNOLOGIES