TOR-CLI-Routing Project

Authors: Tusshar Rana 100426359

Alex Babenko 100442814

Date: 21 November 2024

Abstract

The TOR-CLI-Routing Project aims to establish a secure communication channel using Tor for anonymity and RSA encryption for security. This paper details the setup, implementation, and benefits of using Tor and RSA in a client-server architecture. The project demonstrates how to route traffic through Tor and encrypt messages to ensure privacy and security.

Introduction

In today's digital age, secure communication is paramount. The TOR-CLI-Routing Project addresses this need by combining the anonymity provided by Tor with the robust security of RSA encryption. This project involves setting up a client-server architecture where the client routes its traffic through Tor and encrypts messages using the server's public key. The server decrypts these messages using its private key. This paper provides a comprehensive breakdown of the code, the rationale behind the chosen technologies, and the potential applications of this setup.

TOR

Tor (The Onion Router) is a free, open-source software that provides anonymous communication on the internet by routing traffic through a global network of volunteer-operated servers. Tor's unique ability to anonymize data flow makes it valuable for individuals seeking privacy, such as journalists, activists, or anyone navigating restrictive regimes. Its key purpose is to protect against network surveillance and traffic analysis, which can expose users' identities and online activities (Tor Project, 2024; IdentityIQ, 2023).

How Tor Works

1. **Onion Routing**
   The cornerstone of Tor's design is onion routing, a method that wraps internet data in multiple layers of encryption, akin to the layers of an onion. At each stage (or relay), one layer of encryption is removed, allowing data to move anonymously through the network. The relays can only decrypt enough to know the next step, ensuring no single relay knows the entire path or the source of the data (Tor Project, 2024).

2. **Circuit Creation**
   When a user connects to Tor, a random path, or circuit, through the network is established. This circuit involves three key nodes:

   - **Entry Node:** The entry node knows the user's IP address but does not have visibility into the final destination.
   - **Middle Relay:** This node acts as an intermediary, forwarding traffic while remaining unaware of both original source and final destination.
   - **Exit Node:** The final relay decrypts the last layer of encryption and sends the traffic to its intended destination. While it knows the destination, it does not know the original source of the request (IdentityIQ, 2023; ITP, 2023).

3. **Encryption**
   The encryption at each step ensures that no single node can see both the origin and destination for the all the future nodes. This makes it extremely difficult for adversaries to trace the data back to the user, as each hop obscures the details further (IdentityIQ, 2023).

Components of Tor

1. **Tor Browser**
   The most common way users access the Tor network is through the Tor Browser, which is a modified version of Firefox. This browser is built to prioritize privacy, blocking tracking technologies and ensuring that users remain anonymous while browsing. It also prevents sites from seeing users' physical locations (Tor Project, 2024).

2. **Tor Network**
   The Tor network is a decentralized collection of thousands of volunteer-operated relays. These relays work together to maintain the anonymity of users by obfuscating the path that data travels from sender to receiver (ITP, 2023).

3. **Nyx**

Nyx is a command-line application for monitoring Tor, offering insights into network activity, bandwidth usage, and more. It provides detailed statistics for relay operators and advanced users, displaying circuits, streams, and resource consumption. Nyx helps manage and visualize Tor's operations, making troubleshooting and performance monitoring easier (Tor Project, n.d.).

Benefits of Using Tor

1. **Anonymity**
   Tor is designed to provide anonymity by routing traffic through multiple relays and concealing users' IP addresses. This is critical for users who wish to avoid surveillance, such as journalists working in authoritarian regimes or activists in need of secure communication channels (Tor Project, 2024).

2. **Bypassing Censorship**
   Another significant advantage is Tor's ability to bypass censorship. Many users in countries with restrictive internet policies rely on Tor to access blocked websites, allowing them to circumvent government restrictions and communicate freely .

3. **Enhanced Security**
   Tor's multi-layer encryption provides an additional layer of security against surveillance. However, Tor does not encrypt the last packet that

will be send from the Exit node/router, while Tor enhances privacy, it does not guarantee total protection from all forms of cyber threats, such as malware or phishing attacks (IdentityIQ, 2023).

Challenges and Drawbacks

1. **Slower Connection Speeds**
   Due to the multiple layers of encryption and routing through several relays, Tor users often experience slower browsing speeds compared to conventional internet use. This trade-off is the result of prioritizing privacy over performance (IdentityIQ, 2023).
2. **Potential for Malicious Exit Nodes**
   While most nodes are run by volunteers committed to protecting privacy, there is always the risk of malicious exit nodes, which can potentially intercept unencrypted traffic. This is why it's important to use HTTPS whenever possible when using Tor (IdentityIQ, 2023).
3. **Potential for All the Nodes to be Malicious**
   Although Tor is designed to ensure privacy and anonymity by routing traffic through a network of relays, there is a theoretical risk that all of the nodes in the network could be compromised or malicious. Since Tor relies on the trustworthiness of individual volunteer-run nodes (entry, relay, and exit nodes), it is possible that an adversary with control over multiple nodes could compromise user privacy by tracking or even altering traffic

RSA Encryption

RSA encryption is a widely implemented public-key cryptosystem that enables secure communication by using two keys: a **public key** for encryption and a **private key** for decryption. The system ensures that even if an attacker intercepts the encrypted message, they cannot decrypt it without the corresponding private key (Schneier, 2015).

How RSA Works Without Equations:

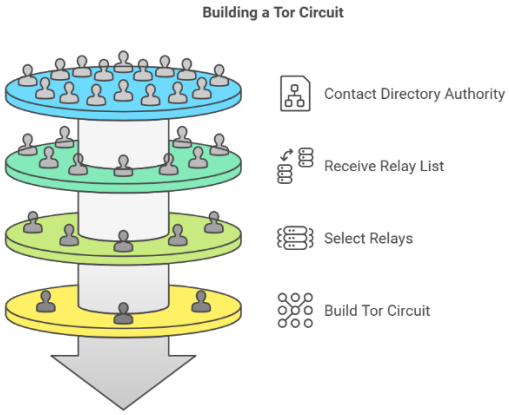
1. **Key Generation**: RSA generates two keys—a public key, which is shared and used by senders to encrypt messages, and a private key, kept secret by the receiver to decrypt incoming messages. This system ensures that unauthorized users cannot access the original message, even if they know the public key (Rivest, Shamir, & Adleman, 1978).

2. **Encryption**: In the encryption process, the message is encrypted using the recipient's public key, meaning only the holder of the private key can decrypt it. This method ensures that even if the communication is intercepted, the content remains unreadable (Schneier, 2015).
3. **Decryption**: The recipient uses the private key to decrypt the message, ensuring that the intended party can access the information. The security of RSA lies in the fact that it's computationally infeasible to derive the private key from the public key alone (Anderson, 2020).
4. **Security and Use Cases**: RSA is commonly employed in secure web browsing through protocols like HTTPS, ensuring the confidentiality of data such as credit card information and personal credentials. It is also used for digital signatures, confirming the sender's identity and message integrity (Rivest et al., 1978).

How Tor works

Ask Tor Directory :

- When a user connects to the Tor network, the Tor client first contacts a **Directory Authority** (a server that holds up-to-date information about Tor nodes).
- The Directory Authority sends the **list of available relays** (nodes) to the client. The client then selects a few relays to build a **Tor circuit**.
- The relays are chosen randomly, and the list is updated frequently to ensure anonymity (YouTube, 2023).



Negotiate Keys :

- After choosing three relays, the client performs **key negotiation** with each relay.
- This is done in layers:
    - The client negotiates a shared secret (symmetric key) with the first relay, then with the second (via the first), and finally with the third relay.
- This layered encryption ensures that each relay only knows its neighboring nodes, not the full path, enhancing privacy.
- The encryption process forms the **"onion" layers** (YouTube, 2023).

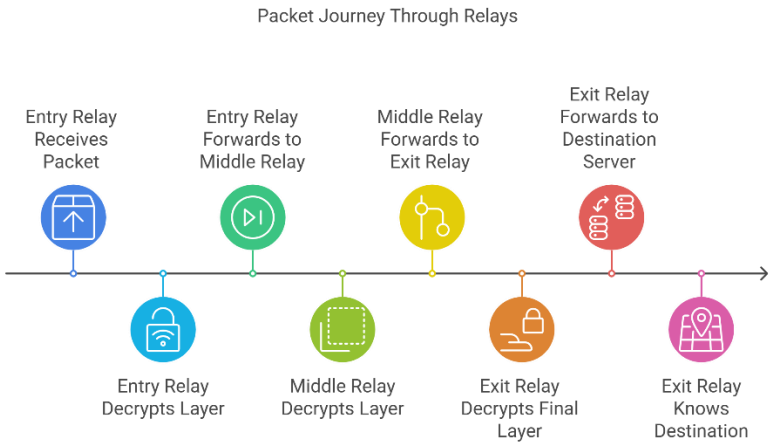
Encrypt a Packet :

- Once the keys are established, the client begins sending packets.
- The client first **encrypts the packet** using the key for the third (exit) relay.
- Then, it encrypts this already-encrypted packet with the key for the second (middle) relay.
- Finally, it encrypts everything with the key for the first (entry) relay.

- The relays decrypt the packet layer by layer as it passes through them, ensuring that no single relay knows both the origin and destination (YouTube, 2023).

How the Packet Would Go (Client -> Server) :

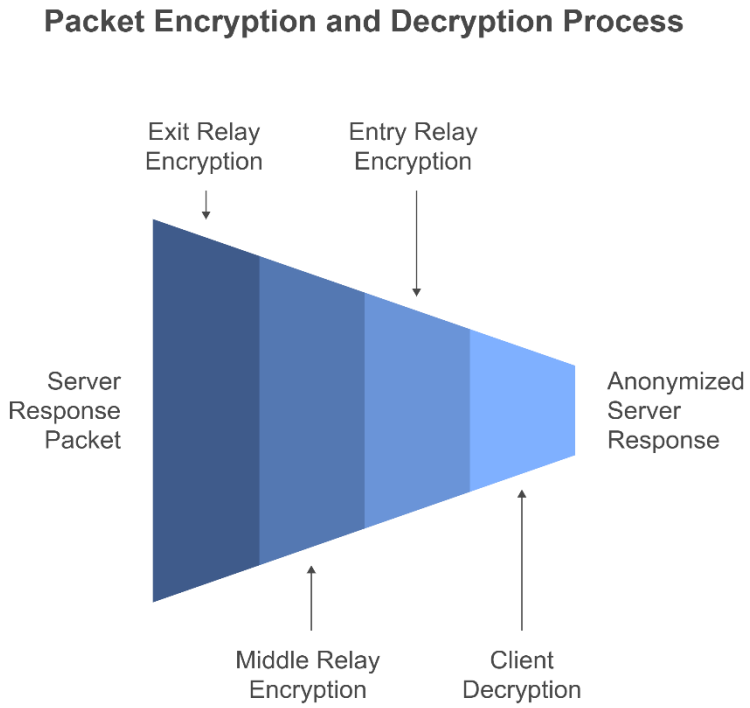
- The encrypted packet first goes to the **entry relay**, which decrypts one layer and forwards it to the middle relay.
- The **middle relay** decrypts another layer and sends it to the exit relay.
- Finally, the **exit relay** decrypts the last layer and forwards the packet to the destination server.
- Only the exit relay knows the final destination (e.g., a website), but it doesn't know who the client is, as that information is hidden by the encryption layers (YouTube, 2023).



How the Packet Would Go Back (Server -> Client) :

- When the server responds, the packet takes the same path back through the relays, but in reverse.
- The packet is encrypted by the exit relay, then sent to the middle relay.
- The middle relay adds a layer of encryption and forwards it to the entry relay.
- The entry relay adds the final layer of encryption and sends the fully encrypted packet to the client.

- The client then decrypts all the layers to read the server's response.
- This ensures that the server doesn't know who the client is, maintaining the anonymity (YouTube, 2023).



Conclusion

The TOR-CLI-Routing Project demonstrates the effective use of Tor and RSA encryption to establish a secure and anonymous communication channel. By routing traffic through Tor and encrypting messages with RSA, the project ensures both privacy and security. This setup can be applied to various scenarios where secure and anonymous communication is essential. Future work could involve adding more features, such as message integrity checks and support for additional encryption

References:

- Schneier, B. (2015). *Applied Cryptography: Protocols, Algorithms, and Source Code in C* (2nd ed.). Wiley.
- Rivest, R., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*.
- Anderson, R. (2020). *Security Engineering: A Guide to Building Dependable Distributed Systems* (3rd ed.). Wiley.
- IdentityIQ. (2023). What is Tor and how does onion routing work? Retrieved from www.identityiq.com
- Tor Project. (2024). The Tor Project. Retrieved from www.torproject.org
- ITP. (2023). Demystifying the dark web: An introduction to Tor and onion routing. https://itp.nyu.edu/networks/explanations/demystifying-the-dark-web-an-introduction-to-tor-and-onion-routing/
- YouTube. (2023). *How Tor works.* [Video]. YouTube. https://www.youtube.com/watch?v=gIkzx7-s2RU
- Tor Project. (n.d.). *Nyx - Terminal (command-line) status monitor for Tor*. Retrieved from https://nyx.torproject.org