Tor: The Onion Router: Part 2

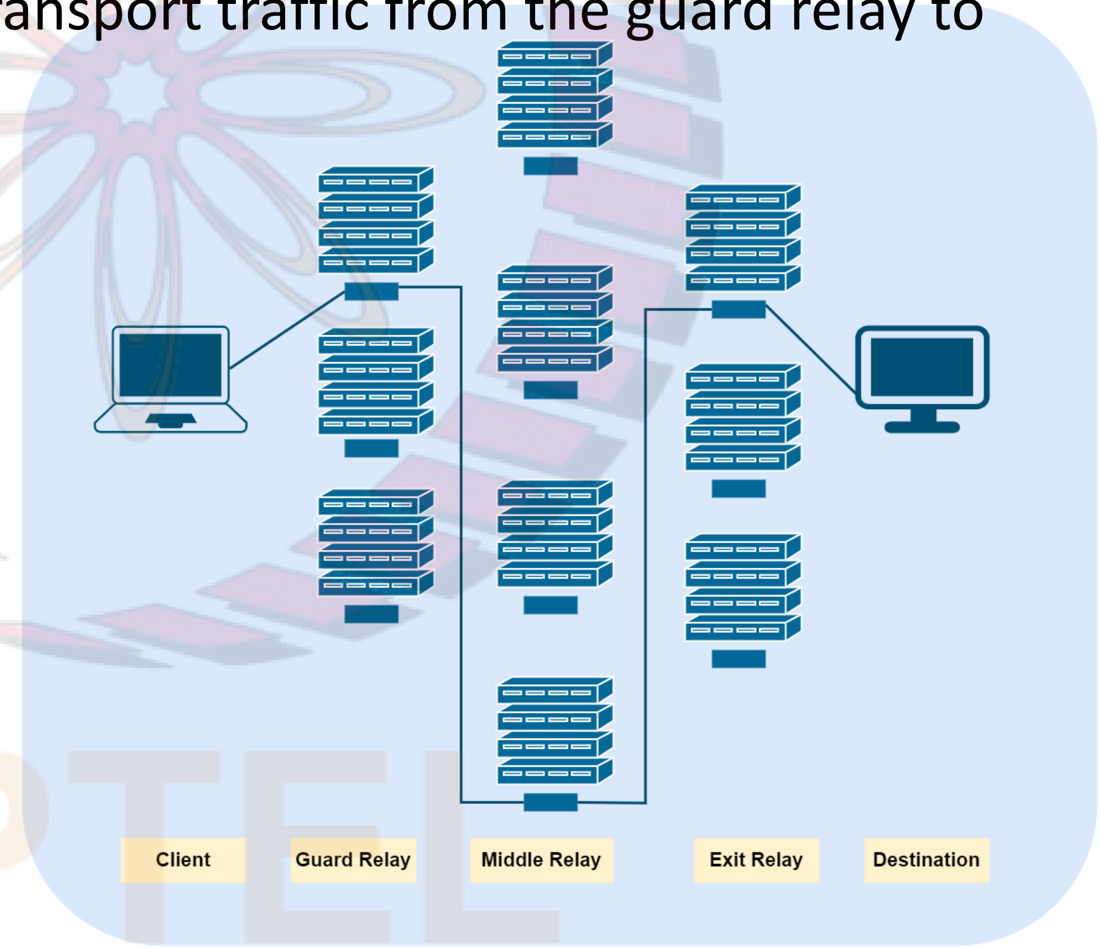
Gaurav S. Kasbekar
Dept. of Electrical Engineering
IIT Bombay

References

- J. Kurose, K. Ross, "*Computer Networking: A Top Down Approach*", Sixth Edition, Pearson Education, 2013
- Dingledine, R., Mathewson, N. and Syverson, P., 2004. "*Tor: The second-generation onion router*", Naval Research Lab Washington DC
- Reed, M.G., Syverson, P.F. and Goldschlag, D.M., 1998. Anonymous connections and onion routing. *IEEE Journal on Selected areas in Communications*, *16*(4), pp.482-494.
- The Tor Project:
  - https://www.torproject.org/
- How Tor Works: Parts One, Two and Three by Jordan Wright
  - https://jordan-wright.com/blog/tags/tor/

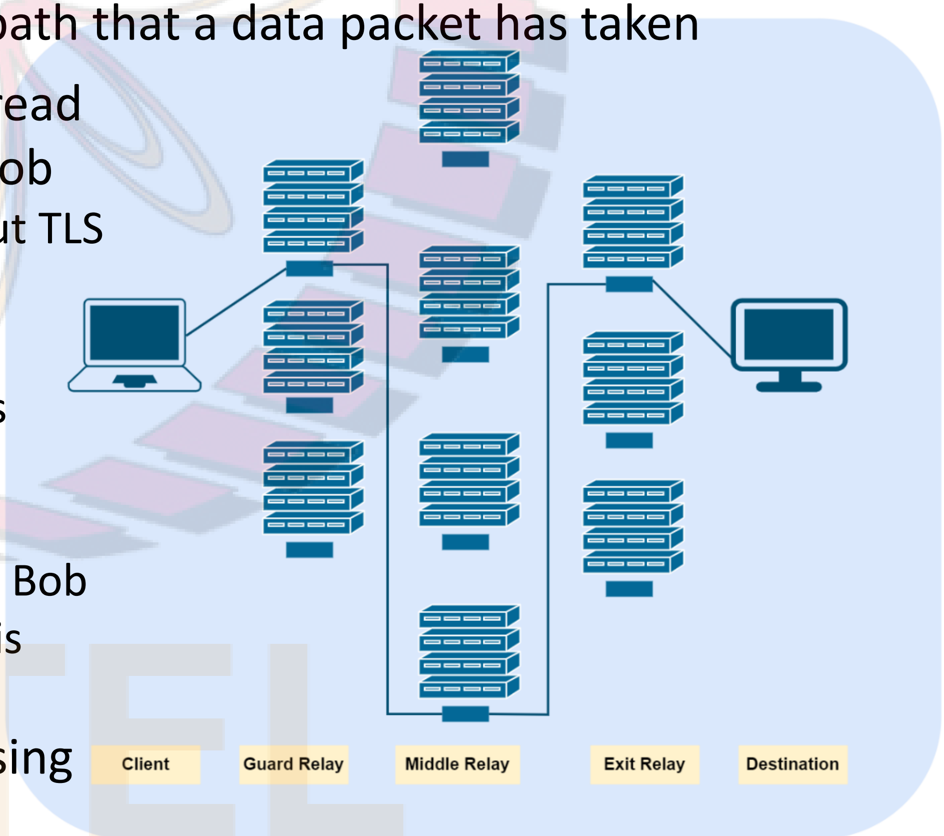
Terminology

- Recall: Tor routes connections through three relays
- **Entry/Guard Relay**
  - ❑ this is the entry point to the Tor network
- **Middle Relay**
  - ❑ middle nodes used to transport traffic from the guard relay to the exit relay
  - ❑ this prevents the guard and exit relay from knowing each other
- **Exit Relay**
  - ❑ these relays send traffic to the final destination intended by the client



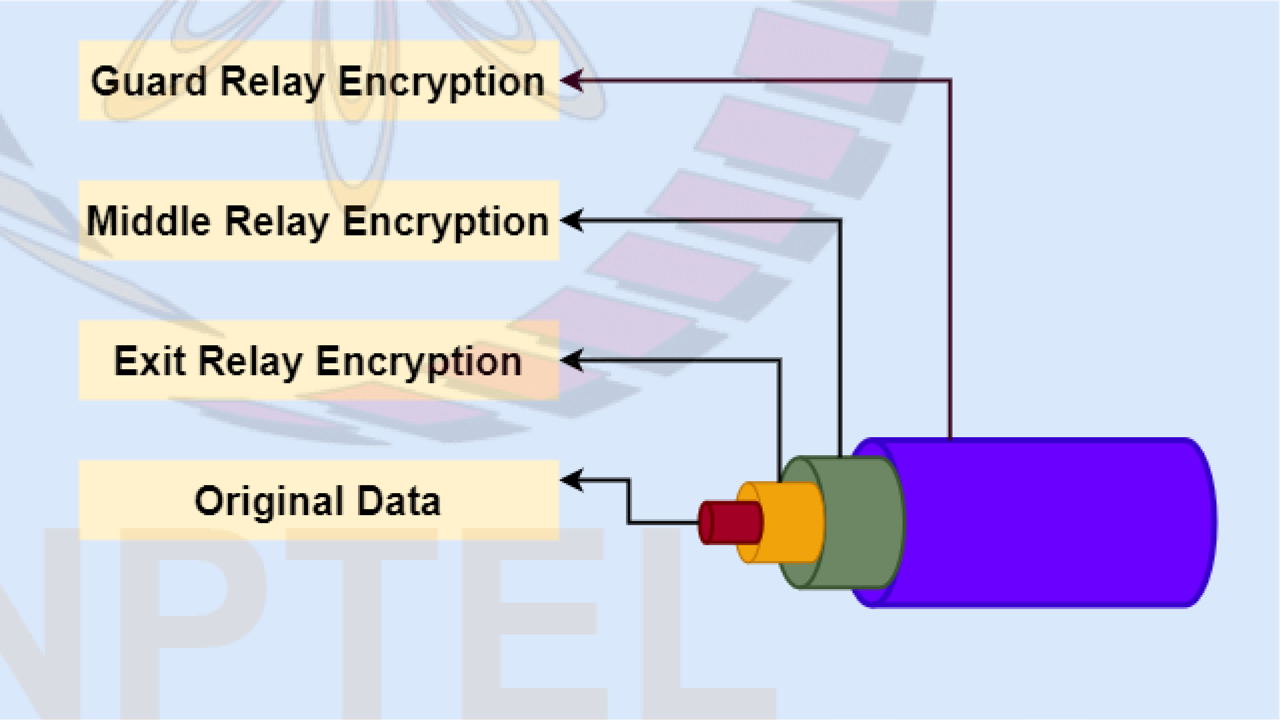
Onion Routing

- Recall: Tor allows independent individuals (volunteers) to contribute relays to its relay pool
  - ❑ so *some relays may be malicious*
  - ❑ no individual relay should know as to which two parties are communicating
  - ❑ no relay should be able to read the data they are communicating
- Tor is designed to put as little trust in relays as possible
- In particular, the design ensures that each relay on path between Alice and Bob knows only:
  - ❑ which node gave it data
  - ❑ and which node it is giving data to
- No individual relay knows the complete path that a data packet has taken
- Also, the entry and middle relays cannot read the data that is being sent from Alice to Bob
  - ❑ even if, e.g., Alice and Bob use http without TLS or any other encryption
- The third relay:
  - ❑ knows the destination's (Bob) IP address
  - ❑ however, it does not know who is communicating with Bob
  - ❑ can read the data being exchanged with Bob if it is not encrypted, and cannot read it if it is encrypted (e.g., if TLS is used)
- The above properties are implemented using *"Onion Routing"*



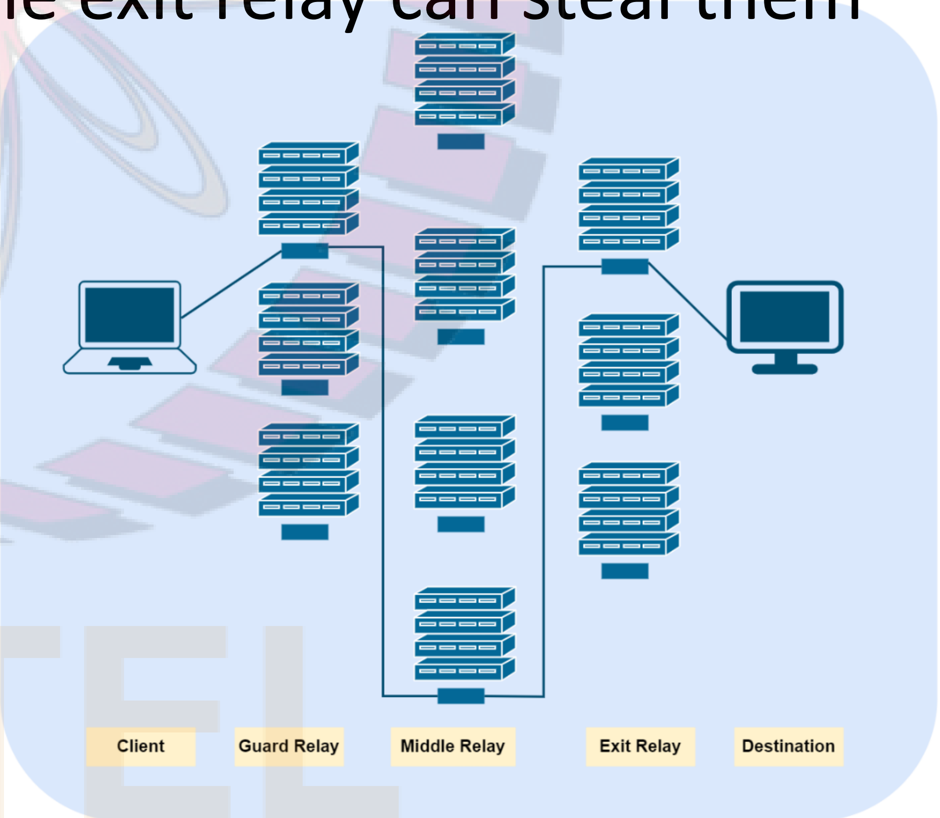
Onion Routing (contd.)

- When a client sends data over a connection through the Tor network, it:
  - ❑ encrypts the original data (including the header, which contains the destination address) in such a way that only the exit relay can decrypt it
  - ❑ then adds the address of the exit relay to the encrypted data and encrypts the result again in such a way that only the middle relay can decrypt it
  - ❑ then adds the address of the middle relay to the encrypted data and encrypts the result once more in such a way that only the guard relay can decrypt it
  - ❑ sends this package to the guard relay
- Note that the original data is wrapped in layers of encryption like the layers of an onion
- By doing this, each relay only has the information it needs to know:
  - ❑ which node it got the encrypted data from
  - ❑ and which node to send it to next



Need for Encryption

- Note that exit relays can read the original data sent by the client, since they have to pass that data to the destination
- So, e.g., if credentials are passed over HTTP, FTP, or other plaintext protocols, the exit relay can steal them
- This can be defended against by:
  - ❑ensuring that the data exchange between the client and destination is performed using a secure protocol such as TLS



Questions That Need to Be Addressed

- How is a circuit established from Alice to Bob via three relays?
  - ❑Note that each relay should only know the identities of the nodes before and after it in the circuit
- Suppose Alice establishes a circuit to Bob via three relays, how is data sent from Bob to Alice?
  - ❑Note that Bob does not know Alice's IP address
- How are encryption keys established between Alice and each relay?
- We consider a *simplified version of Tor* and discuss how the above questions are addressed

Circuit Establishment and Key Establishment

- Each relay has an RSA public-private key pair and corresponding certificate
- Also, there is a TLS connection between every pair of relays, over which data can be encrypted and sent
- Suppose Alice wants to establish a circuit with Bob via the relays R1, R2 and R3
  - ❑ Note that Alice cannot directly communicate with R2 or R3 since they should not know the identity of Alice
- Alice sends a request to R1 to create a connection with it
  - ❑ a TLS connection is established between Alice and R1
  - ❑ symmetric keys for encryption are agreed upon between Alice and R1
  - ❑ subsequently, all communication between Alice and R1 is encrypted
  - ❑ an ID, say (A,R1), is assigned to the connection between Alice and R1
- Then Alice sends a request to R1, over the established connection, specifying the address of the next relay, R2, and requesting for the circuit to be extended to R2
  - ❑ R1 sets up a connection with R2 and assigns an ID, say (R1,R2), to the connection between R1 and R2; *note that R1 does not reveal identity of Alice to R2*
  - ❑ R1 maintains an association between the IDs (A,R1) and (R1,R2) on the incoming and outgoing connections, respectively
  - ❑ Alice selects symmetric keys for encryption, encrypts them using R2's public key, and sends them to R2 via R1; thus, symmetric keys are established between Alice and R2
- In this manner, the circuit is extended hop by hop to R3 and symmetric keys are agreed upon
  - ❑ each relay knows the identities of only the nodes before and after it on the circuit
- R3 sets up a connection with Bob, assigns an ID, say (R3,Bob), to it and maintains an association between the IDs (R2,R3) and (R3,Bob)

Sending Data From Bob to Alice

- Suppose Alice has established a circuit to Bob via three relays, R1, R2 and R3, as described above
- How is data sent from Bob to Alice?
  - Note that Bob does not know Alice's IP address
- Bob sends data over the connection with ID (R3, Bob) to R3
- R3 knows that the ID (R3, Bob) corresponds to the ID (R2,R3)
- R3 forwards the data to R2 over the connection with ID (R2,R3) and so on, until the data reaches Alice
- How should data be encrypted as it travels on the path R3-R2-R1-Alice?
  - The data is encrypted in the same way as the data sent in the forward direction (from Alice to R3), with 3 layers
  - However, in this case the layers of encryption are added one by one, like an onion having its peels put back on: 1 layer each is added by R3, R2 and R1

Limitations of Tor

- Tor does not provide protection against end-to-end timing attacks:
  - some attackers spy on multiple parts of the Internet and use sophisticated statistical techniques to track the communications patterns of many different organizations and individuals
  - in particular, if an attacker can watch the traffic coming out of Alice’s computer
  - and also the traffic arriving at her chosen destination, say Bob’s computer, then
  - the attacker can use statistical analysis to discover that they are part of the same circuit
- If a user, Alice, does not want to reveal her identity to the destination, Bob:
  - she needs to ensure that the data she sends to the destination does not contain any information that reveals her identity
  - e.g., she should not type her name or address in web forms or send any information that reveal’s her computer’s configuration