Network Security

Professor Gaurav S. Kasbekar
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Week - 10
Lecture - 60
Tor: The Onion Router: Part 2

Hello, recall that in the previous lecture, we introduced Tor, The Onion Router. We will now discuss its operation in detail. To discuss the operation of Tor in detail, we introduce some terminology. Recall that in Tor, a source connects with a destination through three intermediate relays. An example is shown here.

This is the source, or the client, and this is the destination. The connection between the source and the destination is via three intermediate relays. This is the first relay, this is the second relay, and this is the third relay. The connection is shown here by these lines. So, this is the connection from the source to the first relay, then from the first relay to the second relay, and so on. This is the connection from the second relay to the third relay, and this is the connection from the third relay to the destination.

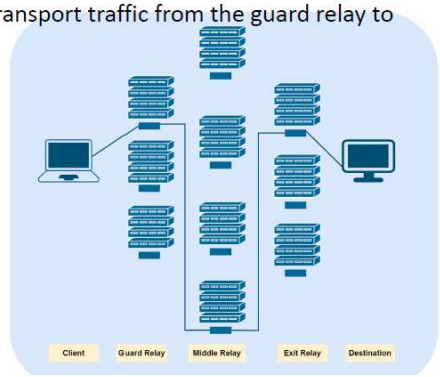
So, the first relay on this path is known as the entry relay or the guard relay. This is the entry point to the Tor network. The Tor network is shown in the middle here. All these are Tor relays. But the three relays used in this particular connection are this one, this one, and this one.

The first of these relays is known as the entry or the guard relay. Then the next relay on the path which is this one is the middle relay. The middle relay is used to transport traffic from the guard relay to the exit relay. We will see that the exit relay is the third relay on the path. So, this is the exit relay.

Since the middle node is used to transport traffic from the guard relay to the exit relay, this prevents the guard and exit relays from knowing each other. In this example, this relay does not know that it is communicating with this relay because the middle relay relays traffic between these two relays. Hence, these two relays don't know that they are communicating with each other. Then the exit relay is the third relay on the path that is shown here. This exit relay sends traffic to the final destination intended by the client.

So, as we can see here, the exit relay communicates with the end destination. This terminology will be used throughout this lecture. The first relay is the guard relay or the entry relay. The second relay is the middle relay and the third relay on the path is the exit relay. So, Tor as the name suggests, it uses what is known as onion routing.

- **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



Recall that Tor allows independent individuals called volunteers to contribute relays to its relay pool. So, some relays may be malicious. A particular individual who contributes a relay may want to actually compromise the anonymity of connections that are being conducted over Tor. Some relays may be malicious or mischievous. No individual relay should know as to which two parties are communicating.

Hence, the use of three relays as opposed to only one relay. So, if only one relay were used instead of three relays, then if that relay were malicious, it would know who is communicating with whom and could disclose this information to others. Hence, the use of three relays so that even if one of them is compromised even if one of them is malicious no one knows who is communicating with whom. No relay should be able to read the data being communicated. Tor is designed to place as little trust in relays as possible.

So, as we said, the use of at least three relays is precisely so that even if one of them is malicious, it cannot know who is communicating with whom and hence it cannot leak this information. In particular, the design ensures that each relay on the path between Alice and Bob only knows which node gave it data and which node it is giving data to. For example, this relay only knows the identity of this client and the identity of this relay but it does not know the identity of this relay or the identity of the destination. No individual relay knows the complete path a data packet has taken. Also the entry and middle relays cannot read the data that is being sent from Alice to Bob.

So, this link is encrypted, this link is also encrypted, this link is encrypted and as we saw earlier, this link is optionally encrypted. So, if Alice and Bob use HTTP without TLS or any other encryption, even then the entry and middle relays cannot read the data that is being sent from Alice to Bob. The third relay knows the destination's IP address, that is Bob's IP address, but it does not know who is communicating with Bob. The third relay can read the data being exchanged with Bob if it is not encrypted, and it cannot read it if it is encrypted, for example, if TLS is used. So, these properties are implemented using onion routing.

When a client sends data over a connection to the Tor network, that connection is shown over here. The encryption of the plaintext data generated by the source is shown over here. This is the original data. When a client sends data over a connection through the Tor network, it encrypts the original data, which includes the header, which contains the destination address, such that only the exit relay can decrypt it. So that encryption is shown here by the yellow layer.

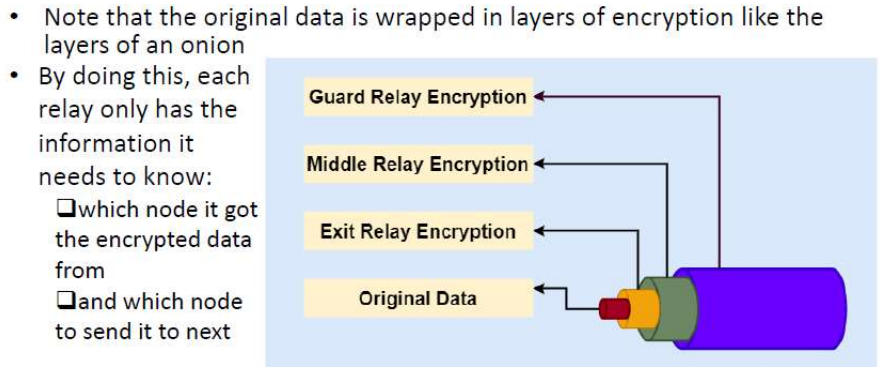
So, this is the exit relay encryption. The original data plus the header, which contains the destination's IP address, this is encrypted using this exit relay encryption. So, it is encrypted such that only the exit relay can decrypt it. Then after this exit relay encryption, the client adds the address of the exit relay to the encrypted data and then encrypts the result again such that only the middle relay can decrypt it. So that second encryption is shown over here.

This is the middle relay encryption. So, after doing the exit relay encryption, the client adds the header, which includes the identity of the exit relay to this information and then it encrypts the resulting information and that is the middle relay encryption. Then again the client adds the address of the middle relay to the encrypted data and encrypts the result once more such that only the guard relay can encrypt it. So, this third encryption is shown here by the purple layer. Finally, the client sends this package to the guard relay.

The client obviously knows the IP address of the guard relay. The client knows the IP addresses of all three relays as well as the destination. Hence, the client can send this package to the guard relay. Now, when the guard relay receives this package, it decrypts the outer encryption, that is the purple encryption. And once it decrypts this, it gets the IP address of the middle relay.

Recall that at this stage, we had added the address of the middle relay to the encrypted data and then encrypted it once more so that only the guard relay can decrypt it. Hence, when

the guard relay decrypts this purple layer of encryption, it gets the IP address of the middle relay. Hence, it can send the decrypted packet to the middle relay. This purple layer and the inner layers these are sent to the middle relay. Then the middle relay can decrypt this green layer of encryption and once it decrypts it, it gets the IP address of the exit relay. So, then the middle relay forwards that packet to the exit relay.



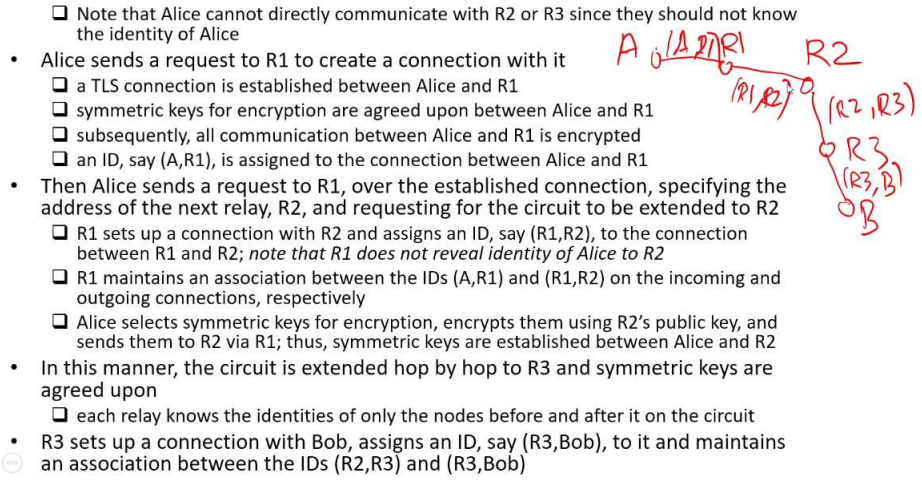
Then the exit relay receives it, decrypts this yellow layer of encryption, extracts the original data, and sends it to the destination. Notice that the original data is wrapped in layers of encryption, like the layers of an onion—hence the name onion routing. So, we can see that this is the first layer which is put onto it, then this is the second layer and this is the third layer. These are like the layers of an onion—hence the name onion routing. By doing this, each relay only has the information it needs to know which node it received the encrypted data from and which node it will send the data to next.

Note that exit relays can read the original data that is sent by the client because they have to pass that data to the destination. So, if credentials are passed over HTTP, FTP, or other plaintext protocols, the exit relay can steal these credentials. This can be defended against by ensuring that the data exchange between the client and the destination is performed using a secure protocol such as TLS. If TLS is used to encrypt the information, then even the exit relay is not able to read that information. Now, there are some questions that need to be addressed in the operation of Tor.

One question is: how is a circuit established from Alice to Bob via three relays? So, the source Alice cannot simply send packets to the three relays directly because the second relay, that is, the middle relay should not know the identity of Alice and the exit relay or the third relay also should not know the identity of Alice. So, it's challenging to establish a circuit from Alice to Bob via three relays. So, how is such a circuit established? 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? That is, in the return path, so how does Bob know where to send the data to? Notice that Bob only knows the identity of the exit relay, Bob does not know the identity of the middle relay, guard relay, or of Alice. So, how does Bob know how to send the return information back to Alice? Note that Bob does not know Alice's IP address.

How are encryption keys established between Alice and each relay? So, again we have this challenge that each relay should only know the identities of the nodes before and after it in the circuit. We now discuss how these questions are addressed in Tor. We consider a simplified version of Tor and discuss how these questions are addressed. First, we discuss circuit establishment and key establishment.



That is, we will discuss how to address the first question, that is how is the circuit established from Alice to Bob via three relays and how to address the third question, that is how are encryption keys established between Alice and each relay. Each relay has an RSA public-private key pair and the corresponding certificate. There is a TLS connection between each 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 because they should not know the identity of Alice.

So, Alice can only directly communicate with R1. And if Alice wants to send a message to R2, then it should be via the relay R1. Alice first sends a request to R1 to create a connection with it. So, a TLS connection is established between Alice and R1. Let's illustrate that.

This is Alice and first a connection is created between Alice and R1. So, this is a TLS connection between Alice and R1. Symmetric keys for encryption are agreed upon between

Alice and R1 using the usual procedure in TLS. So, using that procedure, symmetric keys for encryption are agreed upon between Alice and R1. So, Alice can now send encrypted data to R1.

Subsequently, all communication between Alice and R1 is encrypted. Now, an ID, say (A, R1) is assigned to the connection between Alice and R1. So, this connection is assigned an ID, let's call it (A, R1). So, (A, R1) is the identity of this connection. Then in the next step, Alice sends a request to R1 over this established connection specifying the address of the next relay which is R2 and requesting for the circuit to be extended to R2.

So, Alice sends a request to R1 requesting that the circuit be extended to the next relay, that is, R2 and Alice specifies the address of R2 to R1 that way R1 knows that it should establish the connection with R2. Alice informs the address of R2 to R1 and then R1 establishes a connection with 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 the identity of Alice to R2. So, let's label this connection between R1 and R2 by this identifier (R1, R2).

So, this is the identifier of the connection between R1 and R2. R1 maintains an association between these IDs, (A, R1) and (R1, R2) on the incoming and outgoing connections respectively. That way R1 knows that any information that is received over the connection with ID (A, R1) should be forwarded on the connection with ID (R1, R2) and conversely, any data that is coming on the return path from this connection with ID (R1, R2) should be sent on this connection with identifier (A, R1). So, that's the advantage of maintaining an association between these IDs (A, R1) and (R1, R2) on the incoming and outgoing connections. Then 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. So, Alice selects some random symmetric keys which will be used for securely communicating with R2. Then Alice sends these encrypted keys to R1. Then R1 forwards them over this connection to R2. So, at this point, symmetric keys have been established between Alice and R2.

So, Alice can send encrypted data to R2 and vice versa. R2 can send encrypted data to Alice. In this manner, the circuit is extended hop by hop to R3 and then symmetric keys are agreed upon between Alice and R3. So, this connection is extended to R3. An identifier is assigned to this connection.

This identifier is (R2, R3). And finally, from R3, the connection is extended to Bob. And this connection is assigned the ID (R3, Bob). So, each relay knows the identities of only the nodes before and after it on the circuit. R3 sets up a connection with Bob, that connection is shown over here, assigns an ID to it, say (R3, Bob) and maintains an association between the IDs (R2, R3) and (R3, Bob).

- 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

So, overall in this process, a connection has been established between Alice and Bob, but this has the feature that each relay only knows the identities of the relays before and after it on the path. Now, we have answered the first and third questions; the questions that we raised. We have answered the first question, how is the circuit established from Alice to Bob via three relays? And we answered the third question, how are encryption keys established between Alice and each relay? Now, we answer the second question, how is data sent from Bob to Alice on the return path?

Suppose Alice has established a circuit to Bob via three relays, R1, R2, and R3, as we discussed on the previous slide. How is data sent from Bob to Alice? To send data, we make use of the identifiers that we have assigned to different connections. Note that Bob does not know Alice's IP address. Bob sends data over the connection with ID (R3, Bob) to R3.

In the previous picture, Bob knows about the connection between itself and R3. So, Bob sends the data over this connection with identifier (R3, B). Now, R3 knows that the ID (R3, Bob) corresponds to the ID (R2, R3). So, R3 forwards the data to R2 over the connection with ID (R2, R3), and so on until the data reaches Alice. So, in this picture, R3 maintains an association between the connection with ID (R3, Bob) and the connection with ID (R2, R3). So, whatever data it receives from Bob, it forwards that data over the connection with ID (R2, R3).

Similarly, R2 has maintained an association between the connection with ID (R2, R3) and the connection with ID (R1, R2). Whatever data it receives from the connection with ID (R2, R3), it forwards on the connection with ID (R1, R2), and so on and so forth. So, this way, the data finally reaches Alice. So, we have answered the second question as well. How is data sent from Bob to Alice?

Now, 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 three layers. But in this case, the layers of encryption are added one by one like an onion having its peels put back on. One layer is added by R3, R2, and R1. So, it is an exercise for you to think about this and find out why these layers of encryption are added one by one.

At a high level, these are added to protect the identity of R3 from R1 and identity of R3 also from R1. So, I mean these layers of encryption are added so that R1 does not know the identity of R3. So, the message that is sent by R3 is encrypted by R2, so that R1 does not know R3's identity. It's an exercise for you to think about how adding these layers of encryption one by one by R3, R2, and R1 achieves anonymity. We have discussed the operation of Tor in detail.

Now, we discuss the limitations of Tor. Tor does not provide protection against end-to-end timing attacks. Some attackers spy on multiple parts of the internet and they use sophisticated statistical techniques to track the communications patterns of many different organizations and individuals. For example, consider an attacker who can watch the traffic coming out of Alice's computer. This is the internet and Alice is connected to the internet by a certain connection.

And this is the connection from Bob's computer to the internet. An attacker watches the traffic in different parts of the internet, and in particular, the attacker watches the traffic coming out of Alice's computer. The attacker also watches the traffic arriving at the chosen destination, say Bob's computer. So, the attacker also monitors the traffic that is flowing on this connection between the internet and Bob. Then, the attacker can use statistical analysis to discover that they are parts of the same circuit.

Consider the sequence of packets that are being sent from Alice into the internet and the sequence of packets that are being sent from the internet into Bob's computer. So, these are statistically similar. For example, the spacings between adjacent packets and the lengths of different packets, which are sent on this connection are similar to the spacings between different packets and lengths of different packets that are sent on this connection. So,

whatever packets are sent from Alice, after some delay, similar packets are sent from the internet to Bob. By observing such correlations, an attacker can find out that these packets are part of the same connection.

So, that way, an attacker can use such end-to-end timing attacks to find out who is communicating with whom. So, these techniques are challenging, and they require the attacker to monitor different parts of the internet. The monitoring cost is extremely high. But if the attacker has so many resources, then it can launch such an attack—an end-to-end timing attack. Another limitation of Tor is that if a user, Alice, does not want to reveal her identity to the destination, Bob, in that case, she needs to ensure that the data she sends to the destination does not contain any information that reveals her identity.

For example, she should not type her name or address in web forms or send any information that reveals her computer's configuration. So, Alice has to be careful in this respect. If she types her name or address in web forms or sends any information that reveals her computer's configuration, then her anonymity is compromised. So, that should not happen. Hence, Alice has to be careful while communicating with Bob.

- 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

These are some limitations of Tor. In summary, we discussed the operation of Tor. We discussed that a source Alice communicates with the destination Bob via three intermediate relays and we discussed how a circuit is established from Alice to the destination Bob via three intermediate relays and how are symmetric keys established between Alice and each relay. We also discussed how Bob sends information back to Alice despite not knowing Alice's ID. So, we discussed how data can be sent on the return path. And finally, we discussed the limitations of Tor.

This concludes our discussion of anonymous connections and onion routing. Thank you.