Internet Privacy

Internet Technologies and Applications

Contents

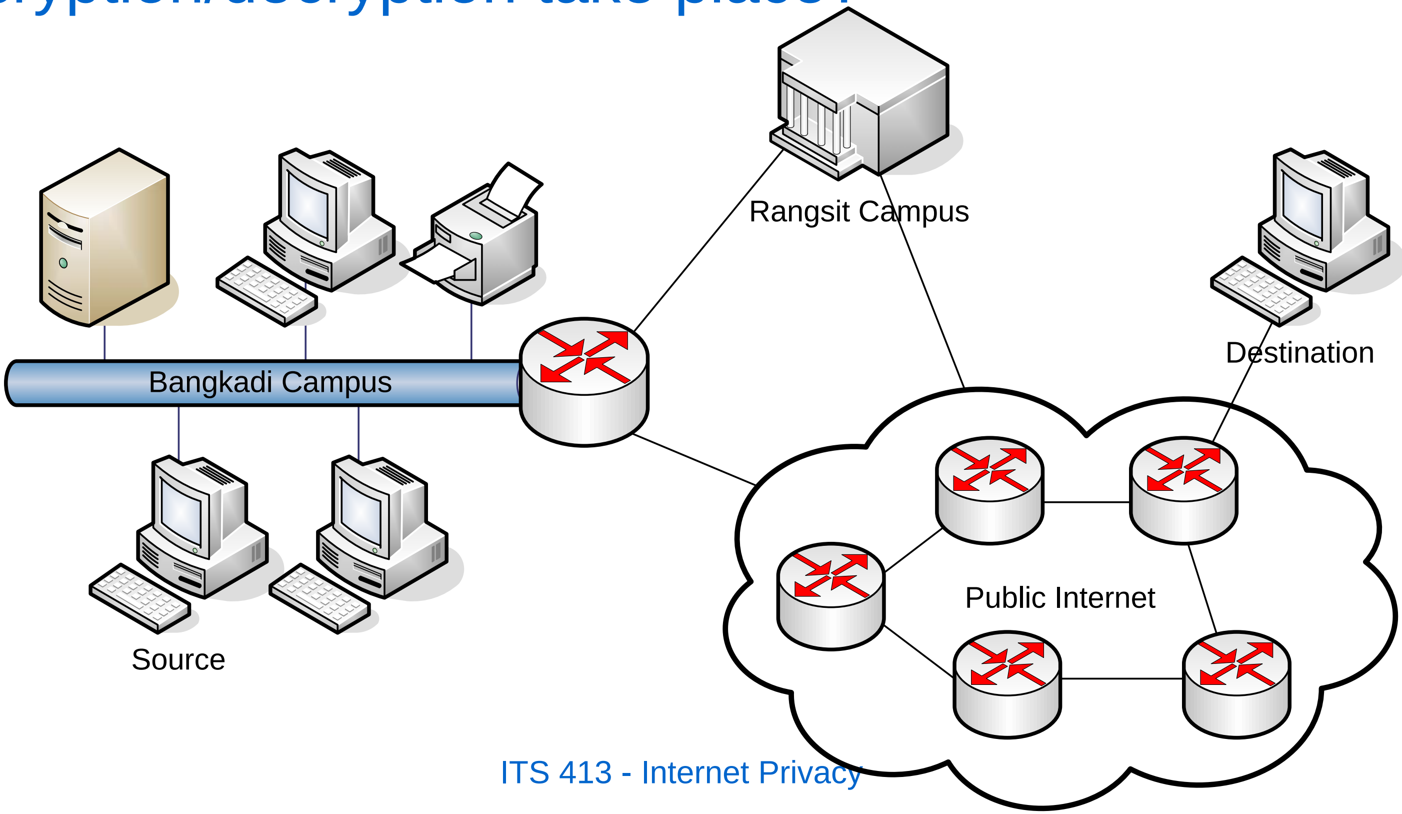
- What is Privacy?
- Internet Security Architecture
  - IPsec
- Anonymous Networking
  - Onion Routing and TOR

Privacy

- Keeping your messages private
  - Aim: stop other (unintended) people from hearing (understanding) your messages
  - Achieved using encryption of messages
    - Only the intended recipient knows the correct key to decrypt and obtain the original message
- Keep your behaviour private
  - Aim: stop people from learning who you are communicating with, when and how
    - Anonymous communications
  - Achieved by “mixing in” with other users; hiding the source/destination addresses

Where to Encrypt?

- Traditionally, encryption is used to provide confidentiality (privacy) of information
- For network communications, where should encryption/decryption take place?



Link versus End-to-End Encryption

- Link Encryption
- Encrypt/decrypt at endpoints of each link
- Requires many encrypt/decrypt devices
- Requires *all* links to use encryption
- Must decrypt/encrypt at each device in path
  - Message is vulnerable at switches (Layer 2 devices)
- E.g. ATM or MPLS switch has a unique key with each of its neighbour switches

- End-to-end Encryption
- Encrypt/decrypt at source/destination hosts
- Hosts do not have to rely on network operators Only data can be encrypted – header information is needed for routers/switches to determine where to send message
  - Vulnerable to traffic analysis
- Message vulnerable at gateways between systems e.g. Internet to phone network

Best to use a combination of link and end-to-end encryption!

IPsec

Network Layer Security

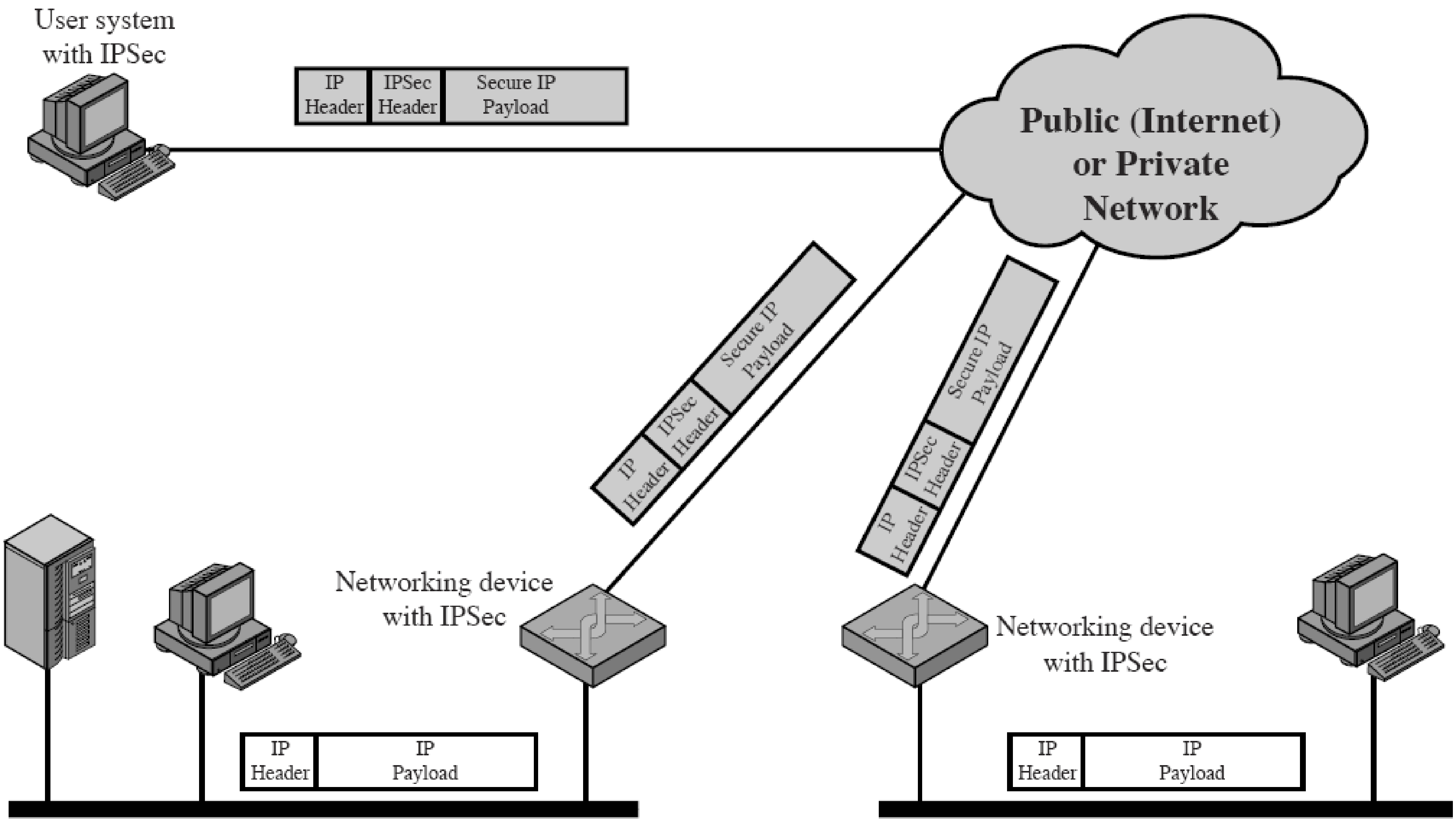
Internet Security Protocols/Standards

Internet Protocols	Security Protocols and Standards
HTTP, FTP, SSH, SMTP, DNS, DHCP, H.323, Messenger, BitTorrent, …	Secure Shell (SSH); Secure Electronic Transactions (SET), DNSSEC, HTTPS, Secure SMTP, PGP, S/MIME…
TCP, UDP	Secure Sockets Layer (SSL), also called Transport Layer Security (TLS)
IP	IPsec – optional addition to IPv4 (built-in with IPv6)
Ethernet, Wireless LAN, ADSL, SDH/PDH, ATM, …	

IPsec

- Internet Engineering Task Force (IETF) defined RFC 2401 (Internet security architecture)
  - IPsec is optional for IPv4 and mandatory in IPv6
    - Mandatory: implementations must support it; but users do not have to use it
  - Implemented as extension headers for IP
- Functionality offered by IPsec:
  - Authentication: verify the sender of IP datagrams
  - Confidentiality: encrypt contents of IP datagrams
  - Data Integrity: guarantee integrity of IP datagrams
  - Key Management: secure exchange of keys
- Allows all traffic to be encrypted at IP (network layer) level
  - Can provide security for all Internet applications (web browsers, email, e-commerce, …)
  - No need to change application or transport protocol software
    - Must have IPsec support on selected PCs, routers, firewalls

Example IPsec Scenario



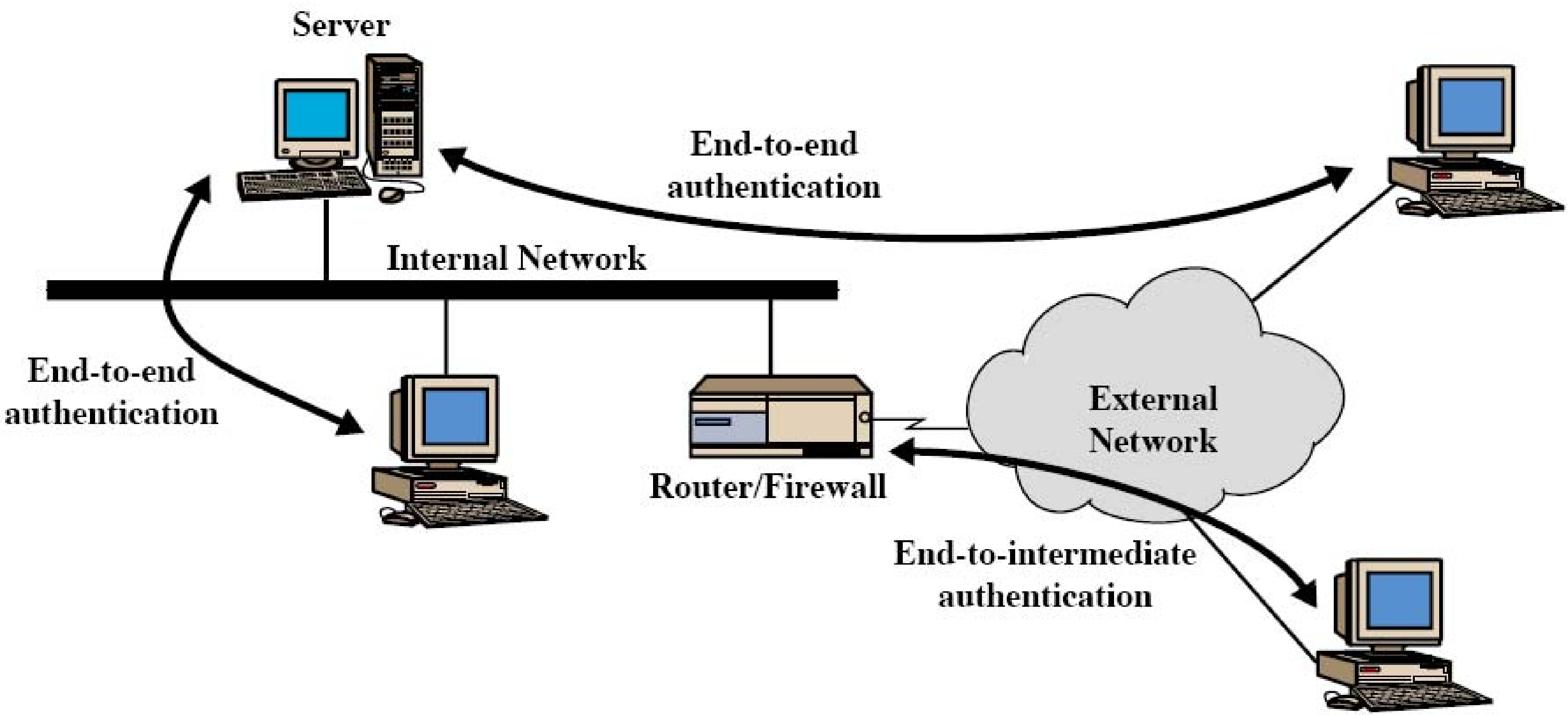
IPsec Components

- Security Association (SA)
  - Sender and receiver must establish relationship, called Security Association
  - Traffic sent within that SA is given services agreed upon between sender and receiver
- Encapsulating Security Payload (ESP)
  - Allows for encryption of payload (e.g. TCP packet), as well encryption plus authentication of payload
    - Encrypt using symmetric key algorithms
    - Authenticate, integrity check using Message Authentication Codes
- Authentication Header (AH)
  - Separate from ESP, allows for authentication-only of payload
    - Authenticate, integrity check using Message Authentication Codes
- Key Management
  - Mechanisms for exchanging keys
  - Two automated protocols
    - Oakley: based on Diffie-Hellman secret key exchange
    - Internet Security Association and Key Management Protocol (ISAKMP): framework for using different algorithms for key exchange

Protocol Modes

- Transport Mode
  - Apply encryption or authentication end-to-end
    - E.g. from PC to PC
  - Original IP header is not protected
    - Only protected TCP/UDP and application layer data
- Tunnel Mode
  - Apply encryption or authentication from intermediate device
    - E.g. from router to router or from router to PC
  - Original IP header is protected
    - Protect IP plus TCP/UDP plus application layer data
  - Often used for creating Virtual Private Networks (VPNs)

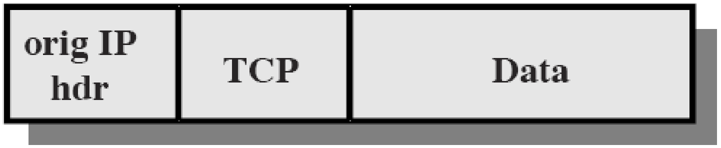
AH and Protocol Modes



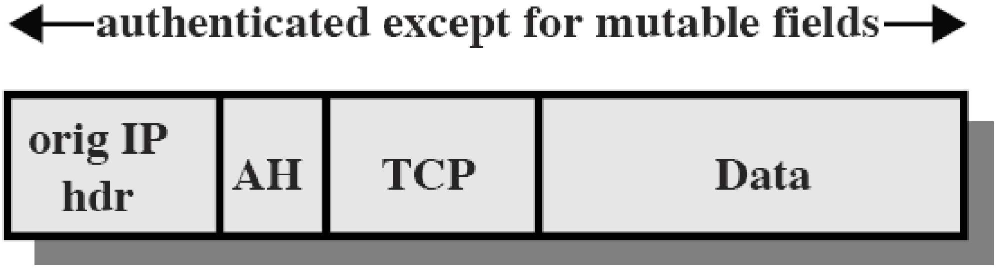
- Transport: end-to-end
- Tunnelling: end-to-intermediate, or intermediate-to-intermediate

AH and Protocol Modes

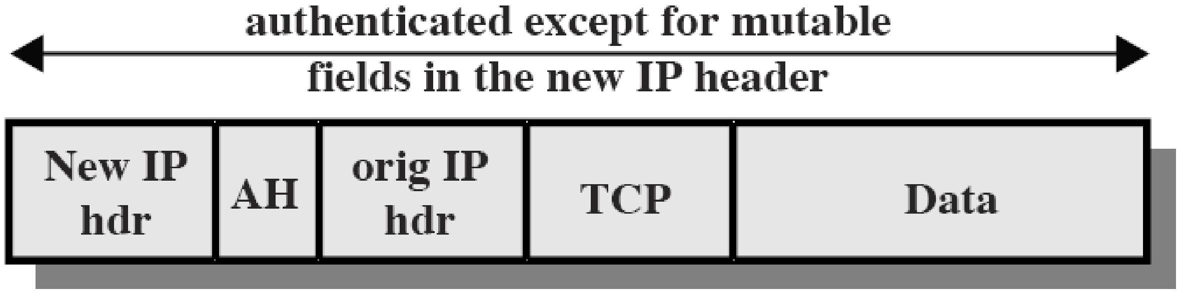
- Original IP datagram (before IPsec)



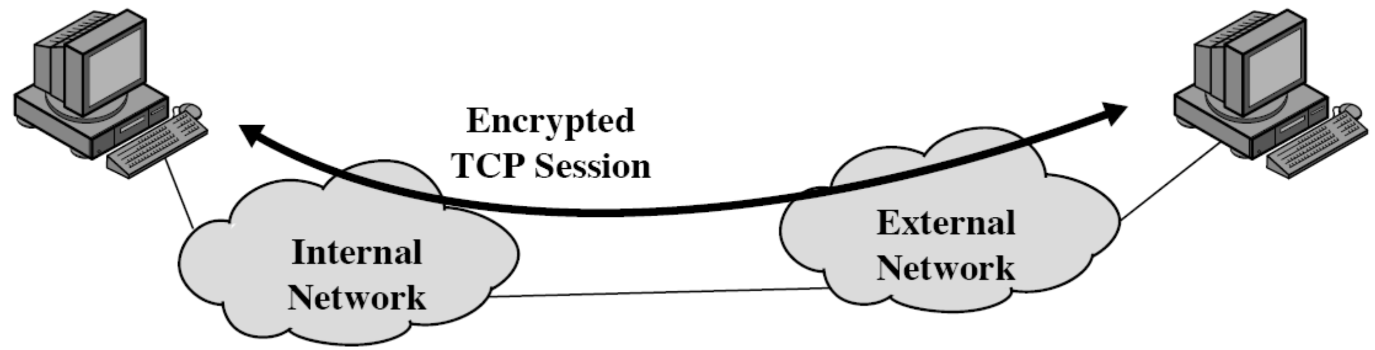
- AH with Transport Mode:



- AH with Tunnelling Mode:

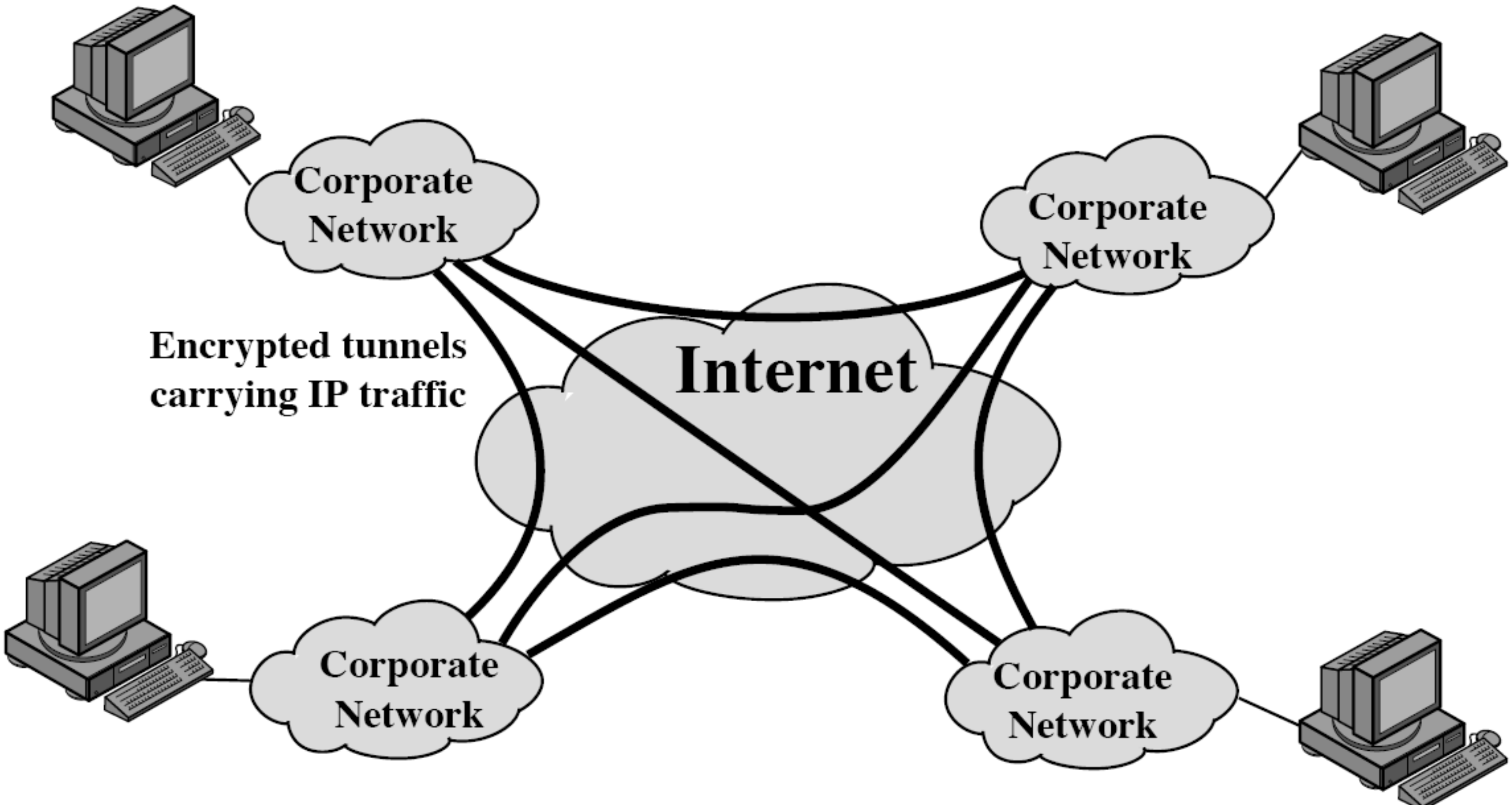


ESP and Transport Mode



- PCs support IPsec
- Encrypt traffic end-to-end; PC-to-PC

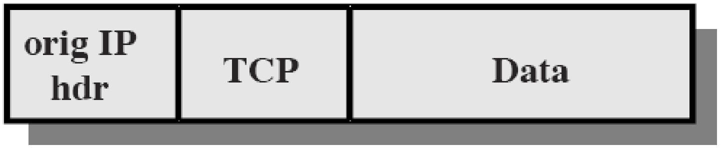
ESP and Tunnelling Mode



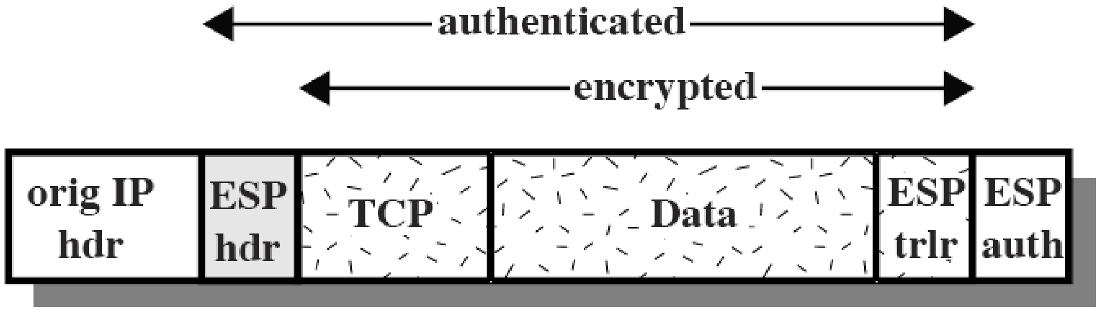
- Hosts/PCs send normal IP traffic (unencrypted)
- Routers at edge of local network creates an IPsec tunnel to other network

ESP and Protocol Models

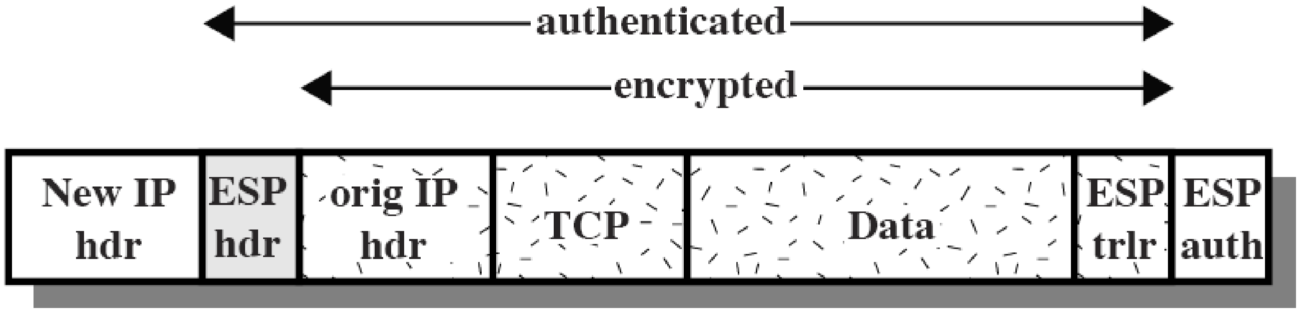
- Original IP datagram (before IPsec):



- ESP and Transport Mode:



- ESP and Tunnelling Mode



Summary of Protocol Modes

	Transport	**Tunnel**
AH	Authenticate IP payload and selected parts of IP header	Authenticates entire inner IP packet (payload plus header) and parts of outer header
ESP	Encrypts IP payload	Encrypts entire inner IP packet
ESP with Auth.	Encrypts IP payload; authenticates IP payload	Encrypts entire inner IP packet; authenticates inner IP packet

Summary of IPsec Services

	AH	ESP (encrypt only)	ESP (encrypt + auth.)
Access control	✓	✓	✓
Data integrity	✓		✓
Data origin authentication	✓		✓
Anti-replay	✓	✓	✓
Confidentiality		✓	✓
Limited traffic flow confidentiality		✓	✓

Applications of IPsec

- Connecting branches/offices securely over the Internet
  - Create a Virtual Private Network using IPsec from Office A to Office B
    - Use of Internet to connect offices is cheaper than dedicated lines (e.g. DSL, E1, ATM)
    - Use ESP in tunnelling mode
- Secure remote access over Internet
  - Employee connects from home/hotel via a ISP to office
    - VPN from user PC to office router
    - Use ESP in tunnelling mode
- Web sites and e-commerce applications
  - IPsec can be used as an alternative or complement to HTTPS and similar protocols
    - Use ESP in transport mode

Anonymous Services

Onion Routing and TOR

Who Needs Anonymity?

- Journalists, dissidents, whistleblowers
- Censorship resistant publishers/readers
- Socially sensitive communicants
  - E.g. chat rooms, web forums for abuse survivors, people with illnesses
- Law Enforcement
  - Anonymous tips, crime reporting
  - Surveillance and sting operations
- Companies
  - Are employers talking to job recruitment agencies?
  - Hide patterns of procurement and suppliers
  - Analysing competitors

Who Needs Anonymity?

- Governments
  - Hiding the source of queries and investigations
  - Sharing information without revealing all the parties
  - Elections and voting
- General public
  - Who are you sending email to?
  - What websites are you browsing?
  - Where do you work?
  - Where are you from?
  - What do you buy?
  - What organisations do you visit?
- Criminals

Encryption for Privacy

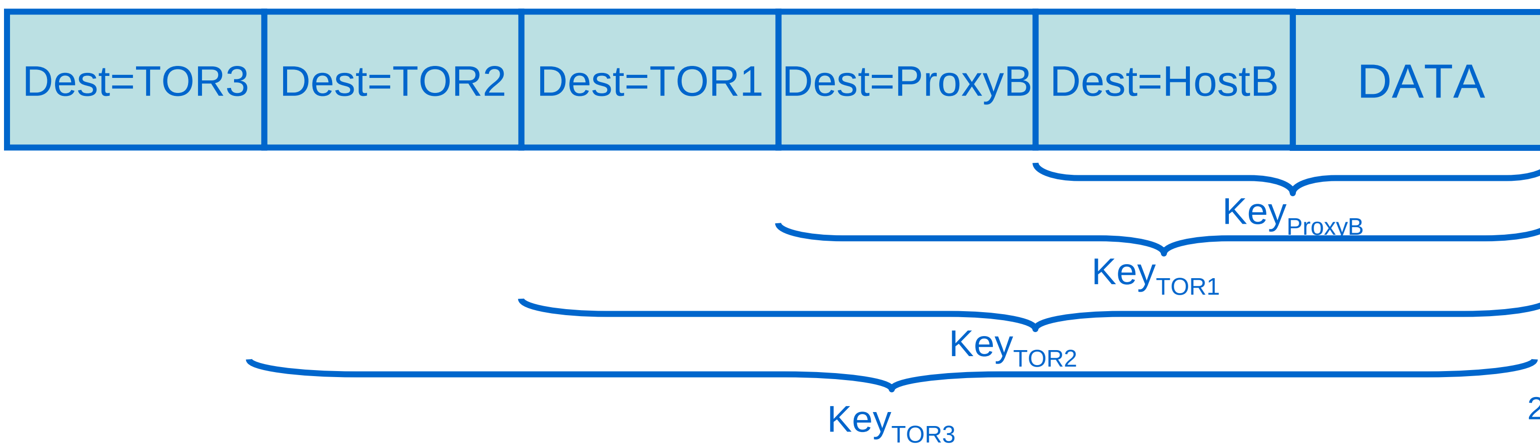
- Link-level encryption:
  - Example:
    - Encrypt from Laptop to WLAN AP over WLAN; then
    - Encrypt from AP to SIIT Bangkadi Router over Ethernet; then
    - Encrypt from SIIT Bangkadi to SIIT Rangsit over ADSL; then
    - …
  - Not suitable for Internet communications
    - Requires encryption across every link the path
    - Must trust intermediate routers
- End-to-end encryption:
  - TLS or IPsec: IP header not encrypted – anyone can see where you send the traffic
    - Tunnelling can hide the original source/destination, but requires tunnel end-points to be created (usually on same network as source/destination)
- Overlay Network and Application Level Encryption:
  - Create a virtual network on top of Internet for routing
  - Traffic between overlay routers is encrypted using application level encryption
    - Use multiple applications of encryption to hide source/destination

Onion Routing and TOR

- Onion Routing
  - Create an overlay network of onion routers in Internet
  - Aims to hide who is communicating
    - E.g. no-one else knows that A and B are communicating
- TOR = The Onion Router (or TOR Onion Routing)
  - Latest implementation of onion routing
  - TOR network contains:
    - Hundreds of routers
    - Hundreds of thousands of users
  - TOR is free:
    - Anyone can download proxy to become a user
    - And/or download the router to become an onion router

Onions

- Onions are messages
  - Each message is encrypted using multiple layers of encryption
  - Each router that forwards an onion, “peels” off a layer of encryption
  - Eventually the original plaintext message arrives at receiver
- Example (see diagrams):
  - DATA is to be sent from Source A to Destination B
  - Proxy A encrypts in this order:
    - M1 = DATA + Header (Dest = Host B), encrypted with Key_{ProxyB}
    - M2 = M1 + Header (Dest = Proxy B), encrypted with Key_{TOR1}
    - M3 = M2 + Header (Dest = TOR1), encrypted with Key_{TOR2}
    - M4 = M3 + Header (Dest = TOR2), encrypted with Key_{TOR3}
    - M5 = M4 + Header (Dest = TOR3)
  - Proxy A sends the following (M5) to TOR3:



Routing in TOR

- Every onion router maintains permanent connections with a set of neighbours
  - The network topology is fixed/static
    - This is ok since only relatively small number of routers (100's)
    - Possible to create full mesh – every router knows every other router
  - Set of directory servers maintain information about all routers
    - IP address, keys, policies
- A client selects a path through the network
  - Path is a set of onion routers
  - All messages on a connection will be sent via that path
  - (Remember: this is overlay of Internet; data between two onion routers may go through several IP routers)

Encryption Keys in TOR

- How does a Proxy obtain keys of TOR routers?
  - Proxy first obtains information about TOR routers from Directory Server
    - May include Public Key Certificates
  - Proxy then uses a key exchange protocol (e.g. Diffie Hellman) to exchange a shared, secret key with each TOR router
    - Example: KeyTOR2 is a secret key shared between TOR2 and ProxyA (no-one else knows this key)
- Data Encryption:
  - Messages (onions) sent to TORs are encrypted with the corresponding secret key
- Design trade-offs:
  - This uses public key encryption (slow) to exchange keys, and then uses symmetric key encryption (data) to encrypt data
  - Creates a session lasting several minutes
    - All data during the session will be encrypted with same secret key
    - Then a new key exchange will be performed and a new secret key used

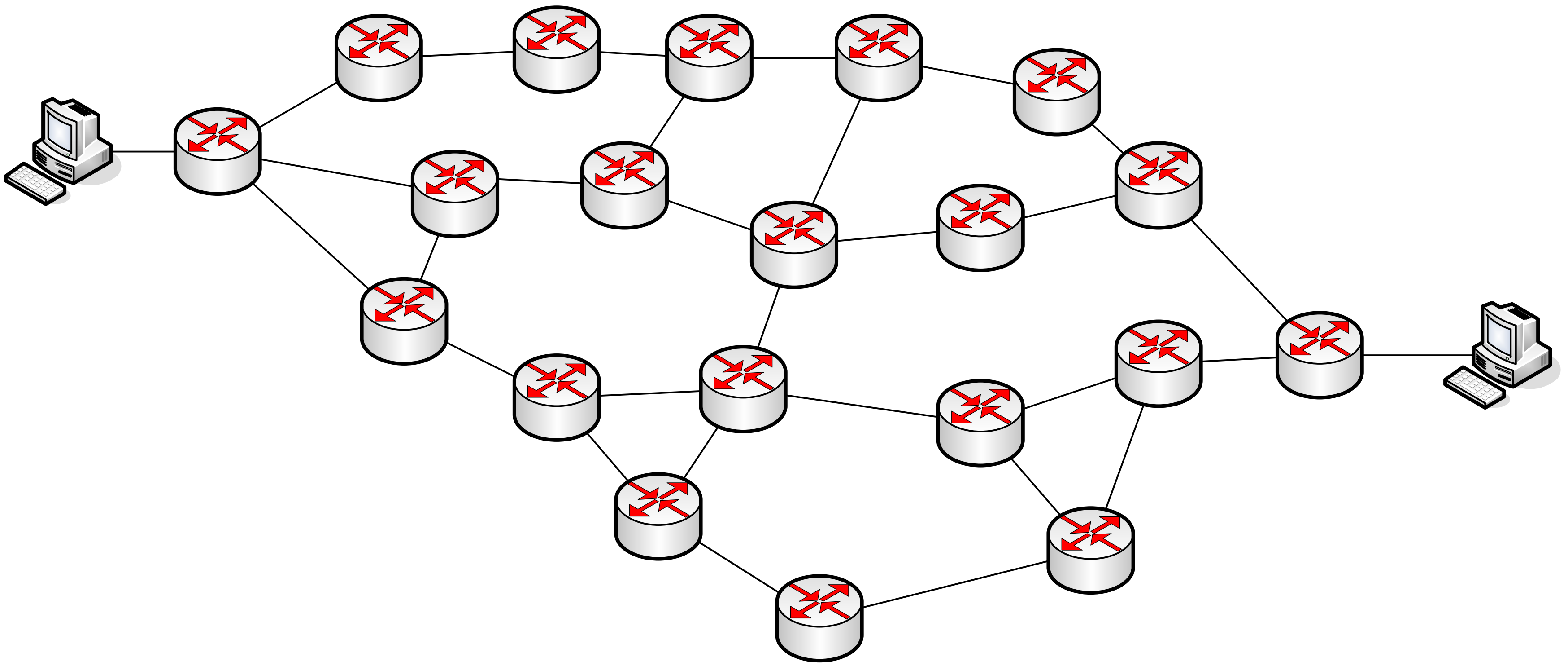
Connection Setup in TOR

- When Source A wants to communicate with Destination B:
  - Application on source A connects to Proxy A
    - Proxy is a TOR Router that also provides interface to TOR network for standard applications
      - E.g. Your web browser will connect to TOR Proxy the same way it connects to any web proxy – the proxy will then handle remaining connection to destination
    - Proxy can run on the same computer as Source A
  - Proxy A selects a path to Proxy B
    - All data in the session will go through the same set of TOR routers
    - Path does not have to be shortest path – may be random (as long as it is not too long)
    - Proxy learns about TOR routers from a publicly available TOR Directory Server
  - Proxy A exchanges secret keys with each TOR router (including Proxy B) along the path

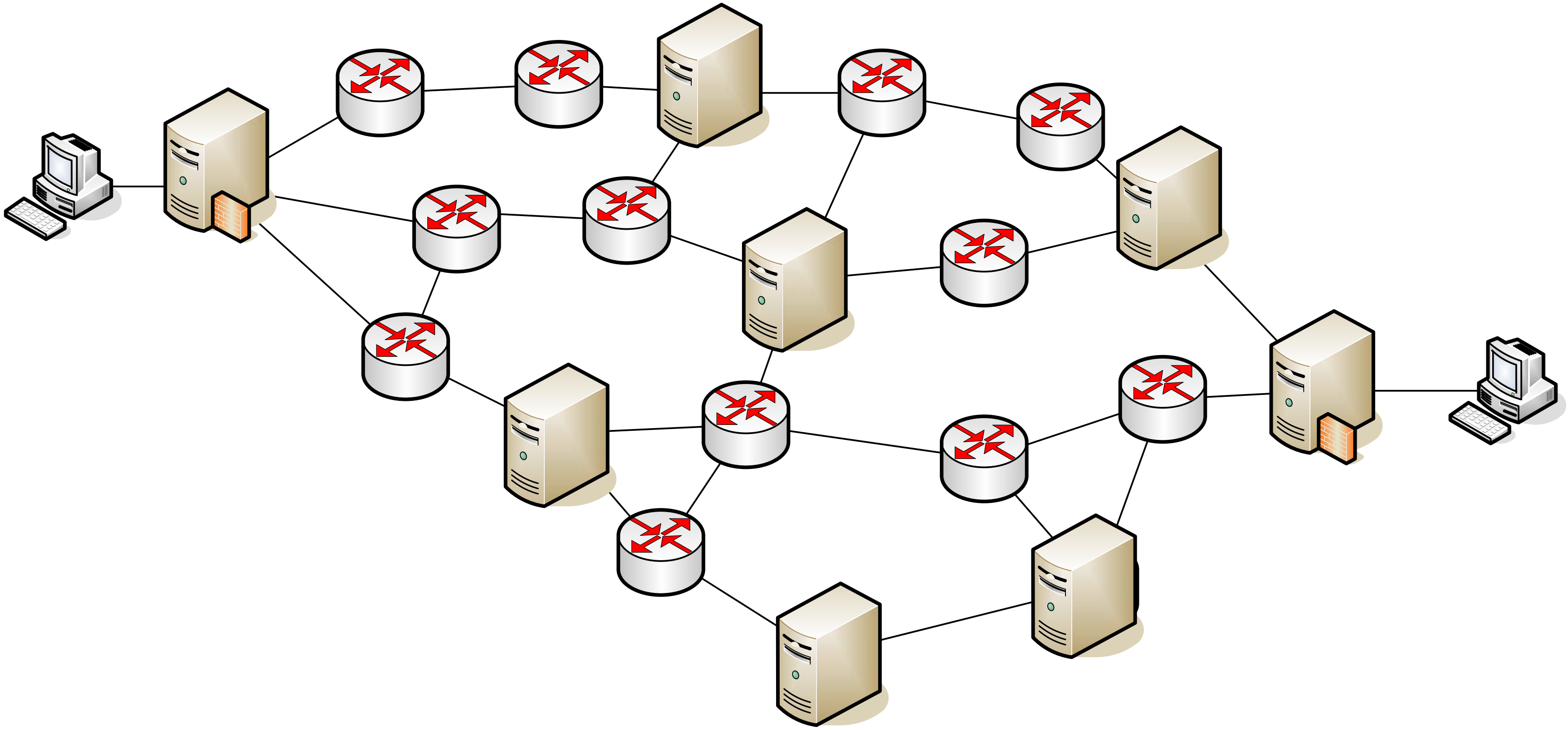
Sending Data in TOR

- When a source A sends to Destination B:
  - DATA is sent from Source A to Proxy A
    - May not be encrypted; this is just normal Internet communications
  - Proxy A creates the onion and sends to first TOR in path, e.g. TOR3
  - TOR3 decrypts the outer layer of onion to determine the next TOR in path (TOR2) and sends
  - Each subsequent TOR decrypts the outer layer and sends to next TOR
  - Finally, DATA is received by Proxy B and forward to Destination B
    - Again, this does not have to be encrypted

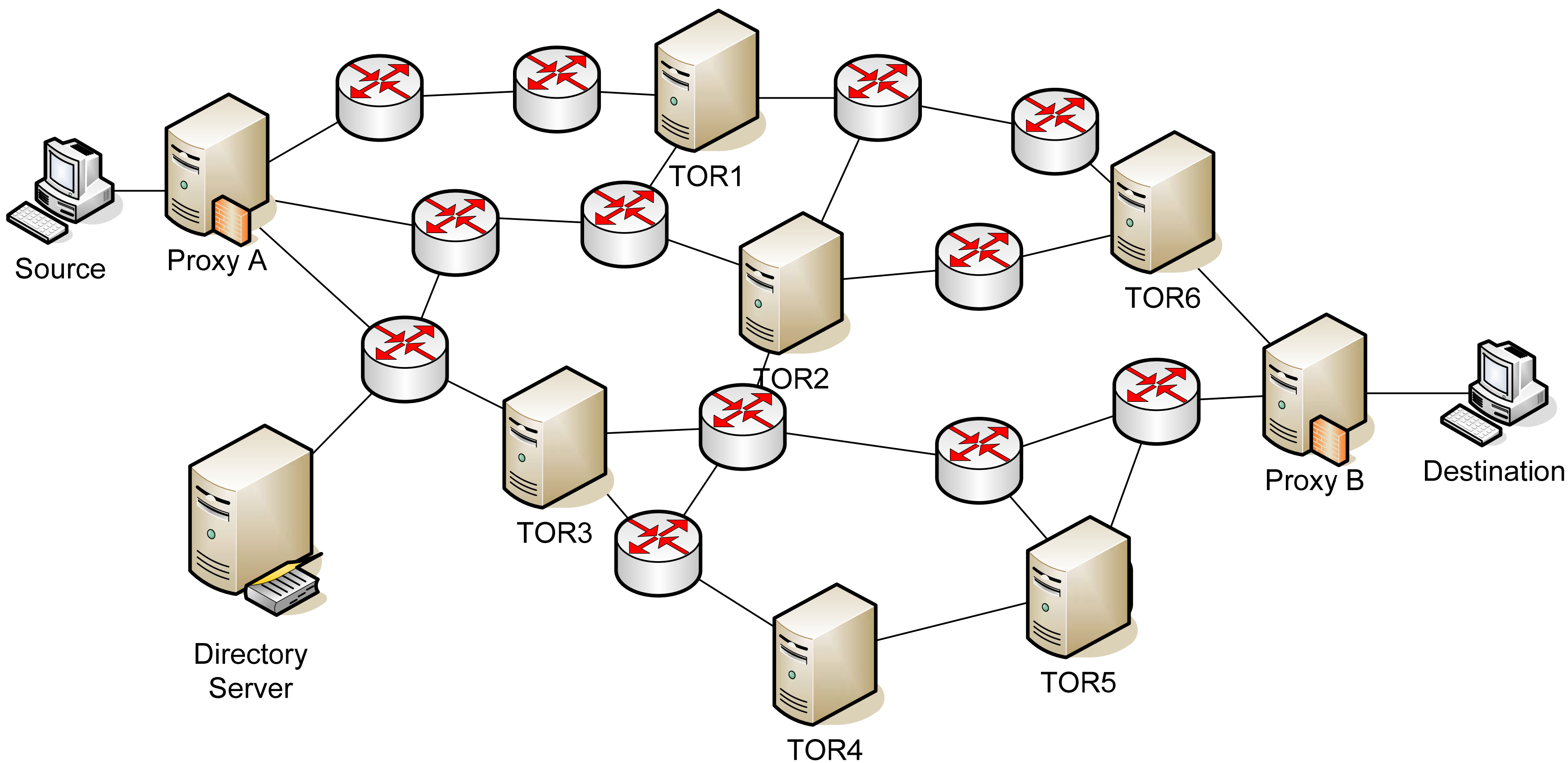
An internet



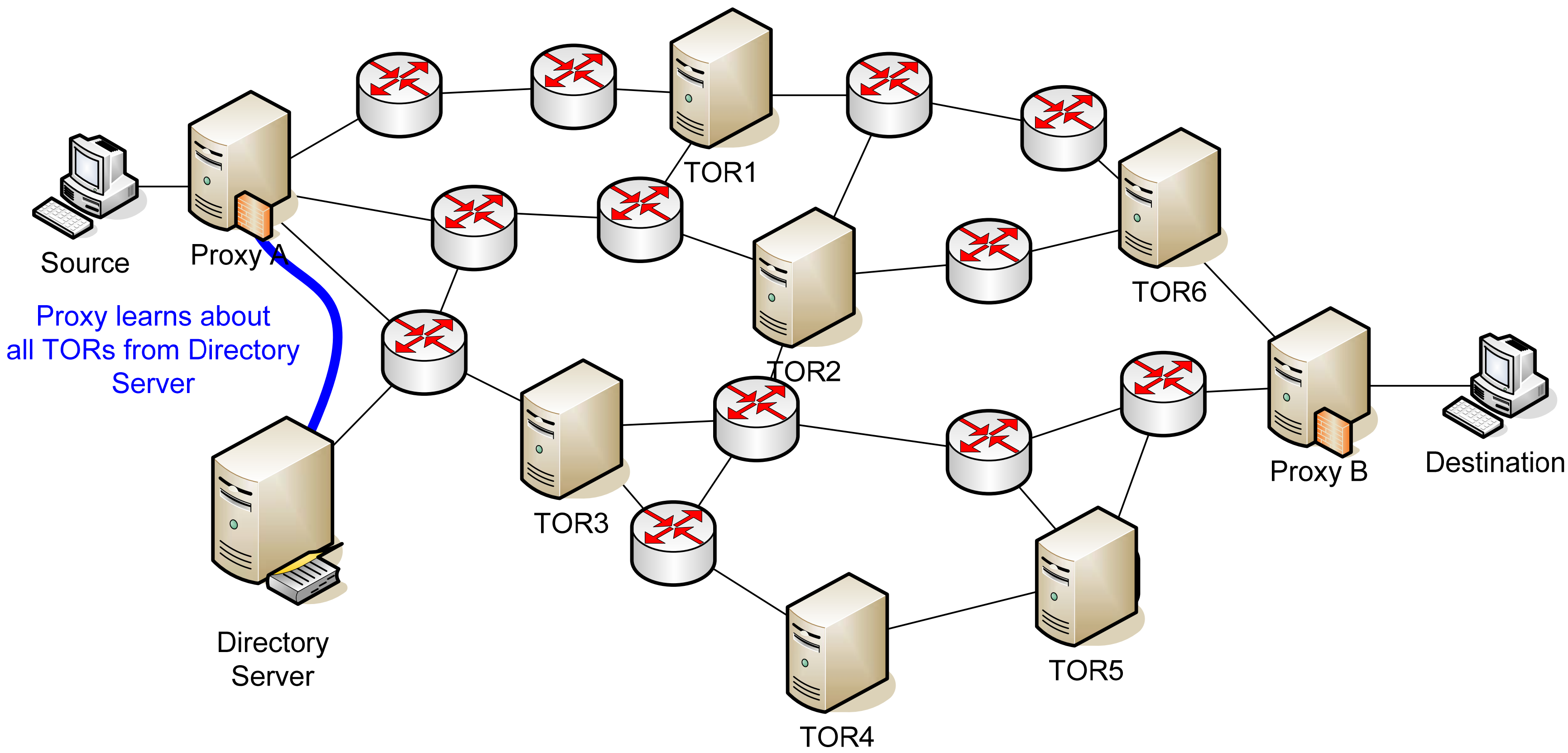
Overlay Network



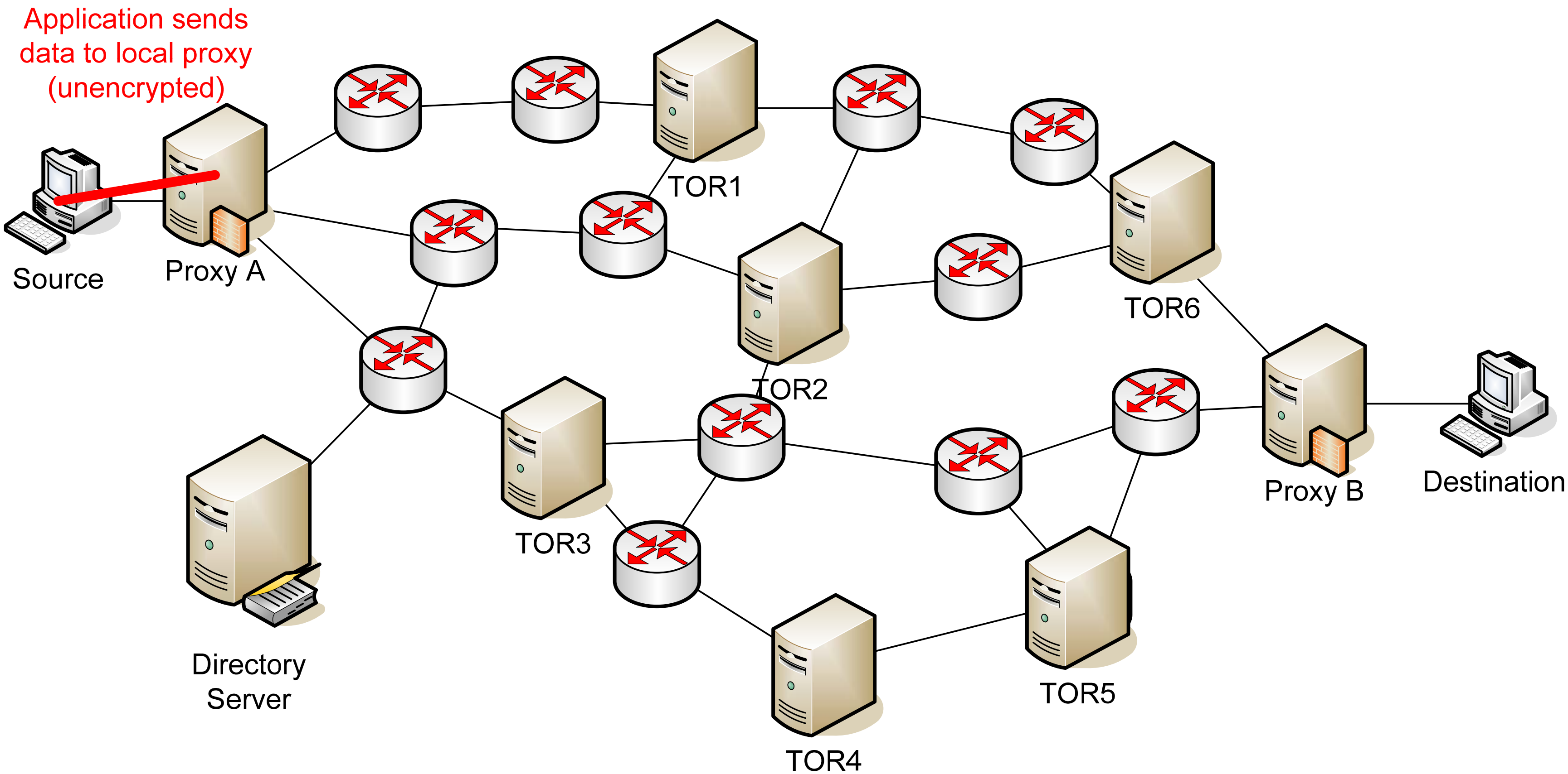
Onion Routing (TOR) Network



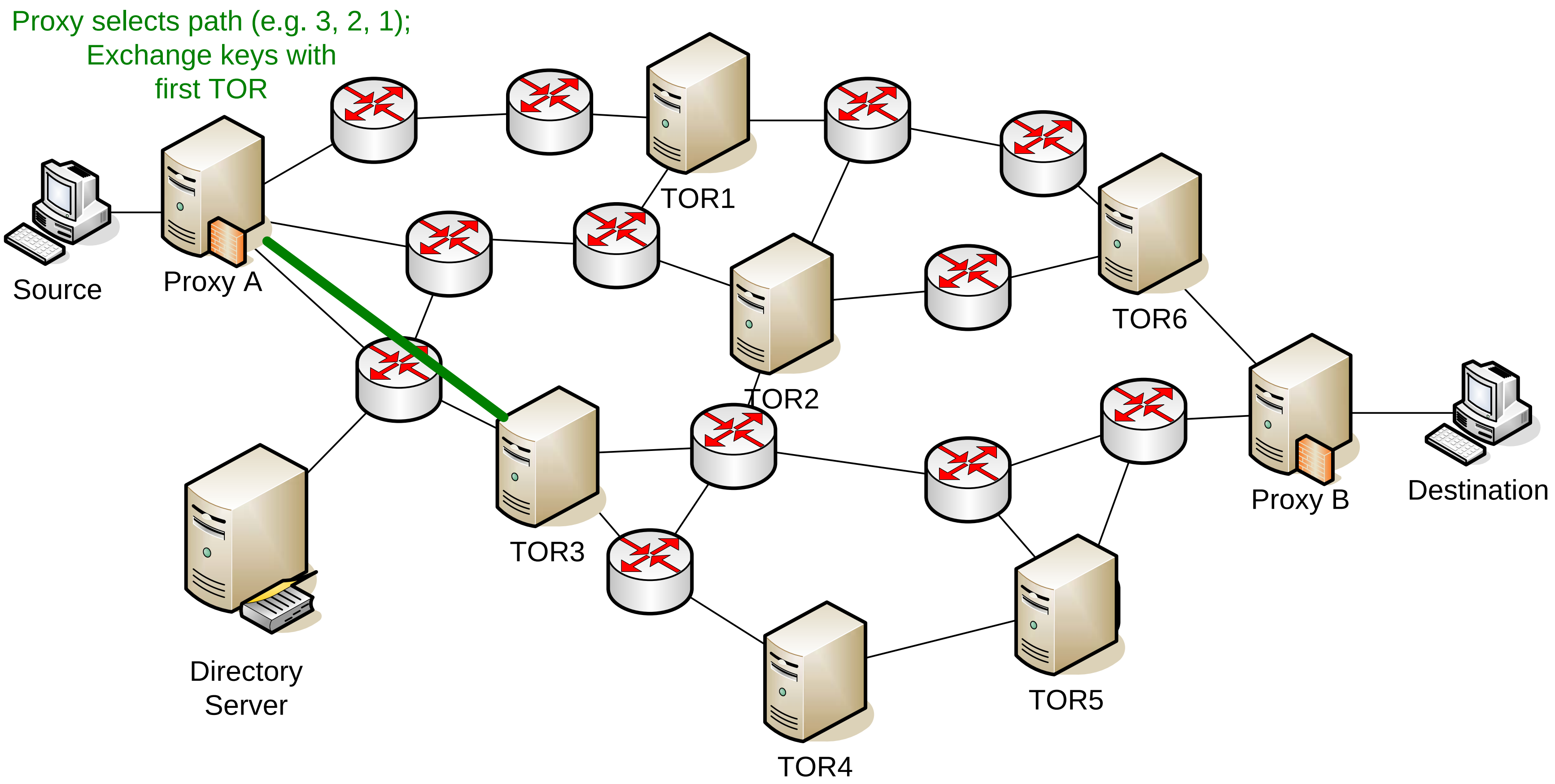
1. Proxy Learns Topology



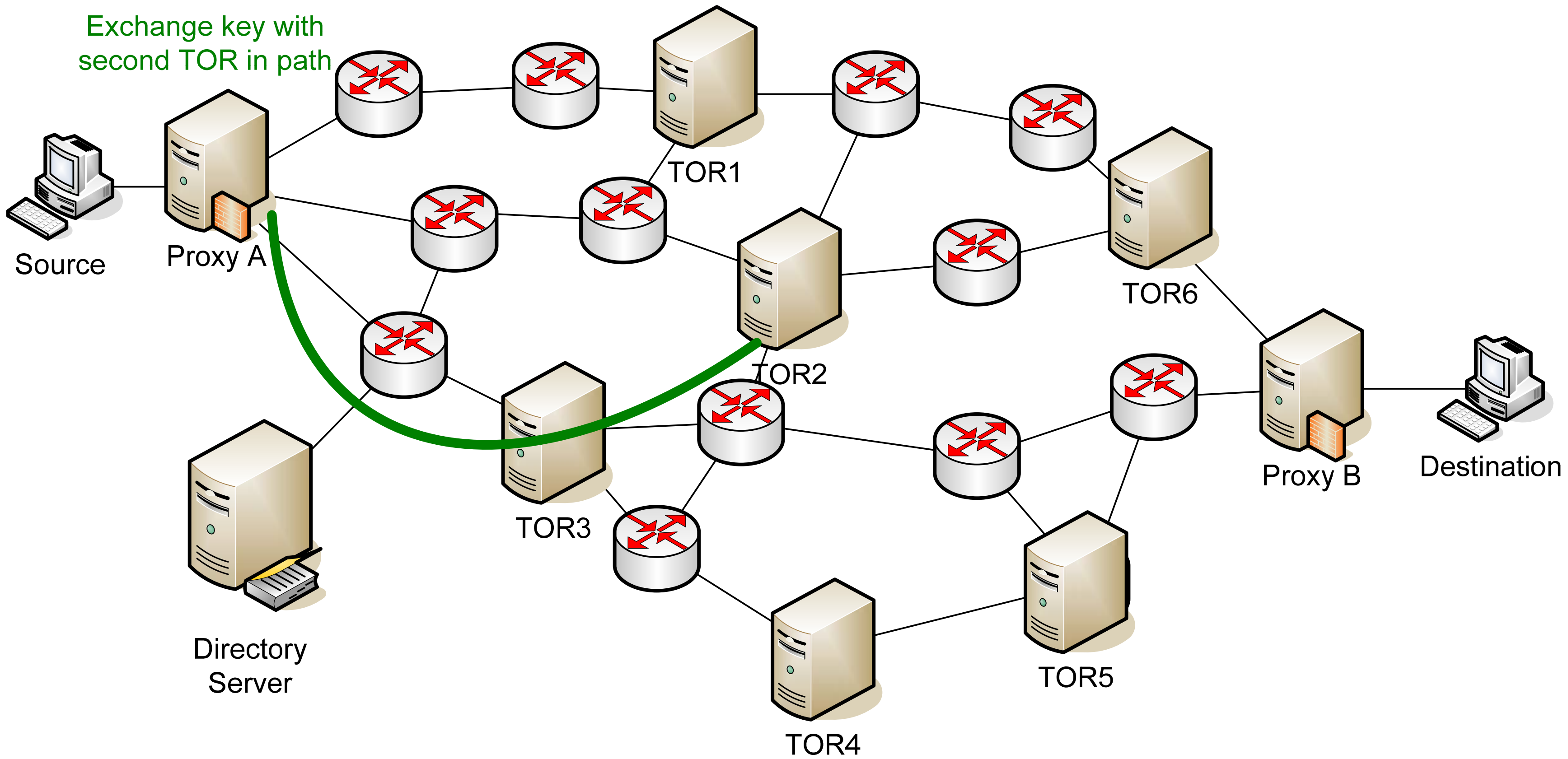
2. Source Sends DATA



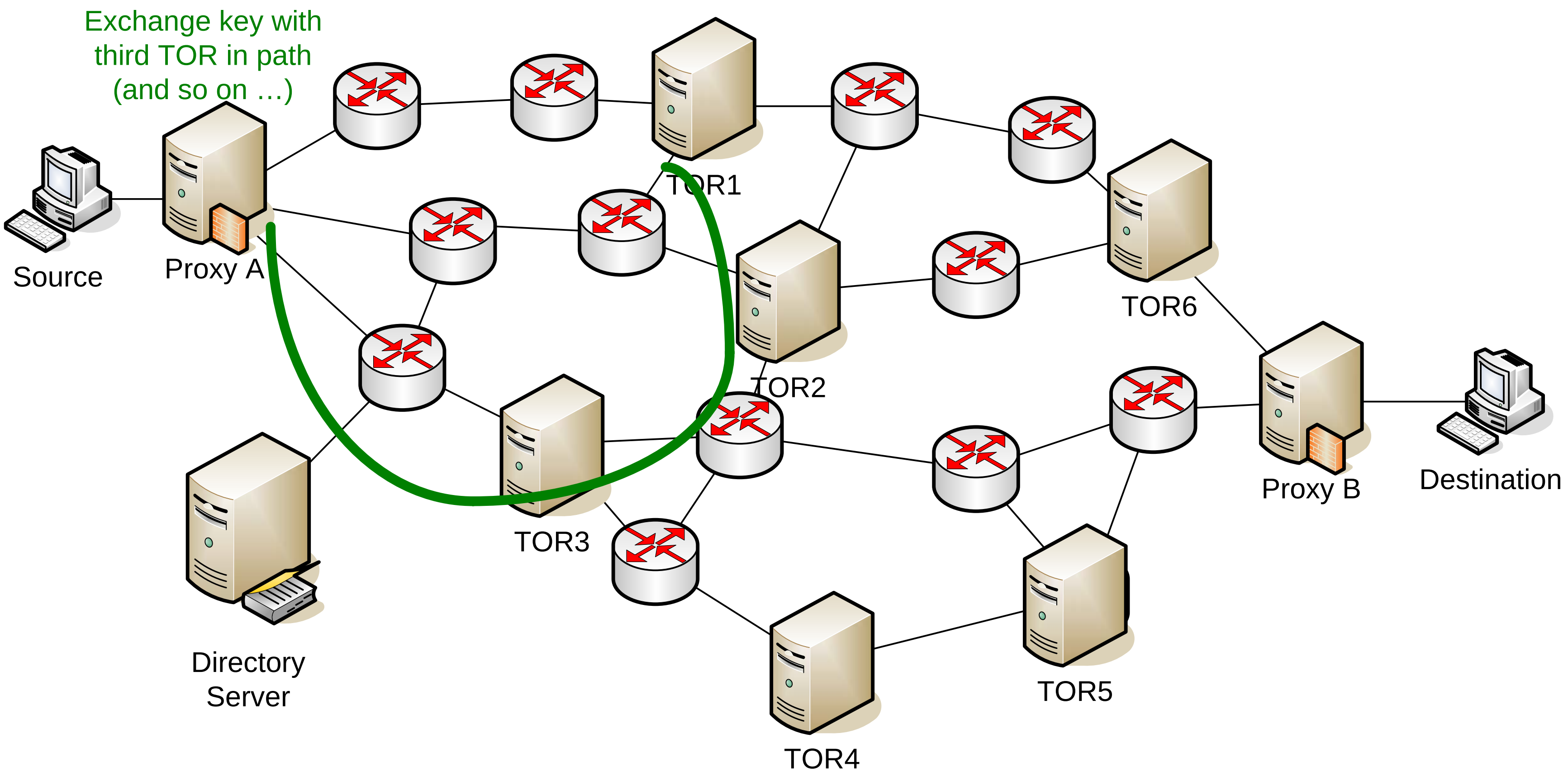
3. Proxy Selects Path and Key Exchange



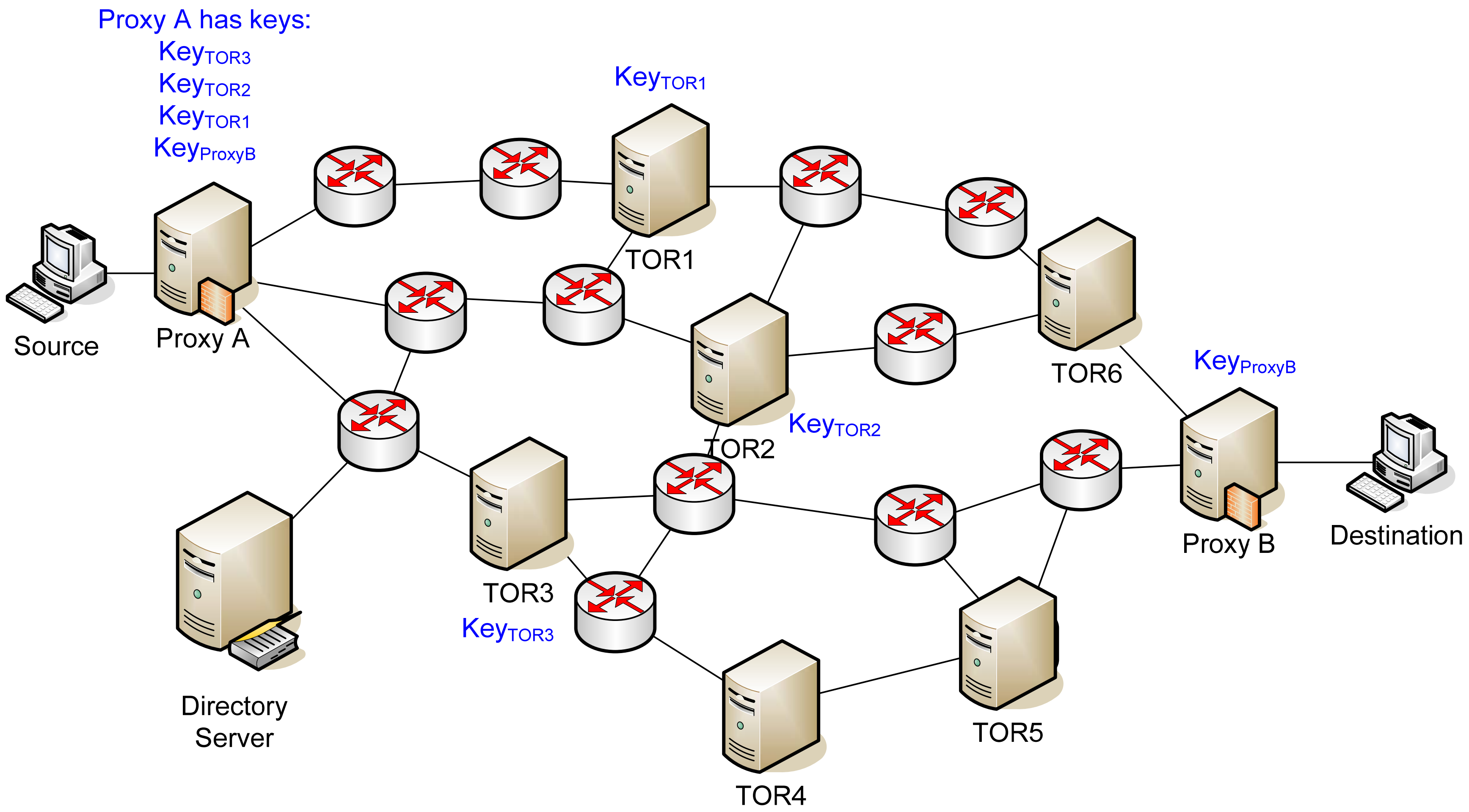
4. Exchange Keys with TOR2



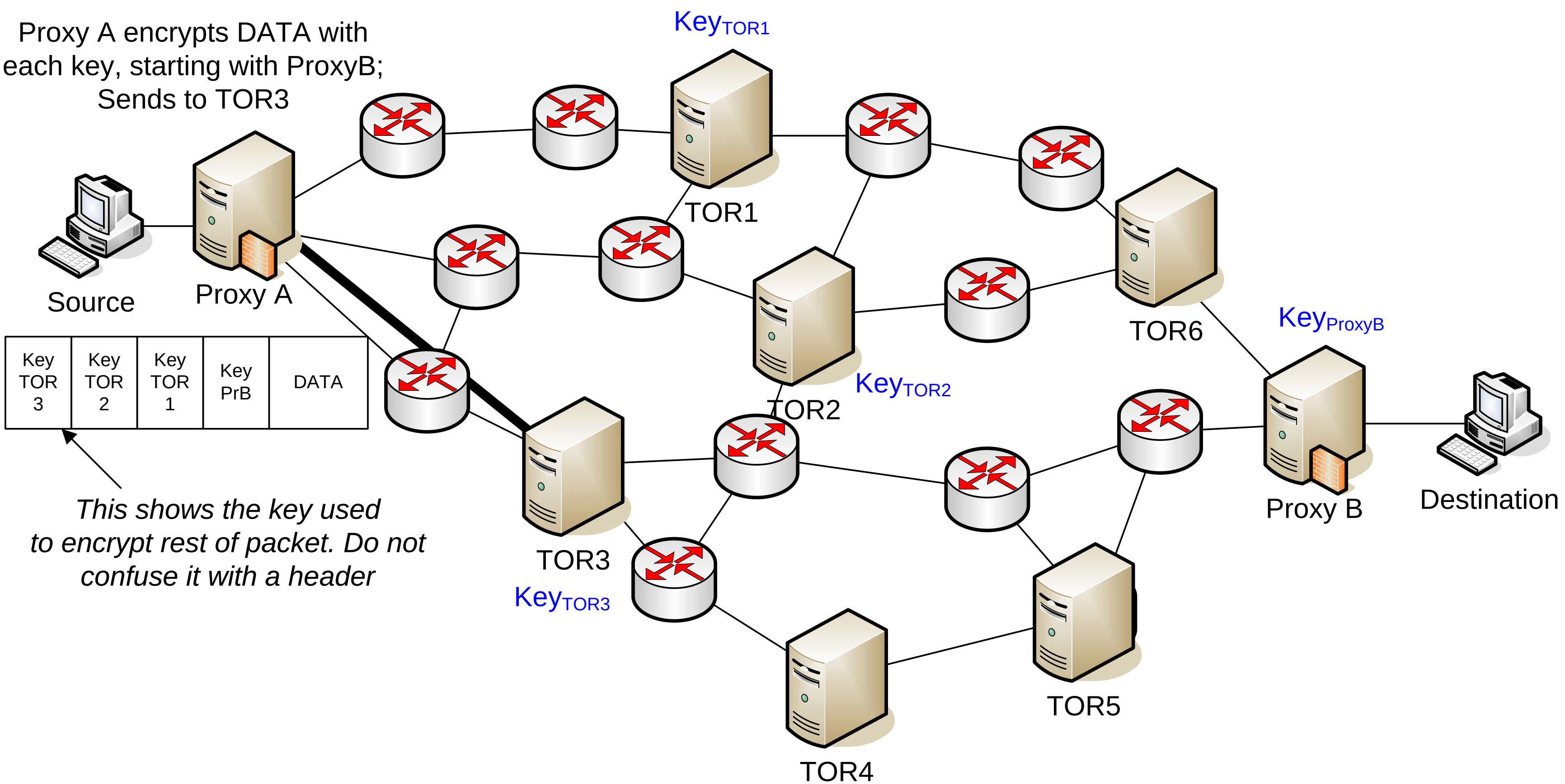
5. Exchange Keys with TOR1



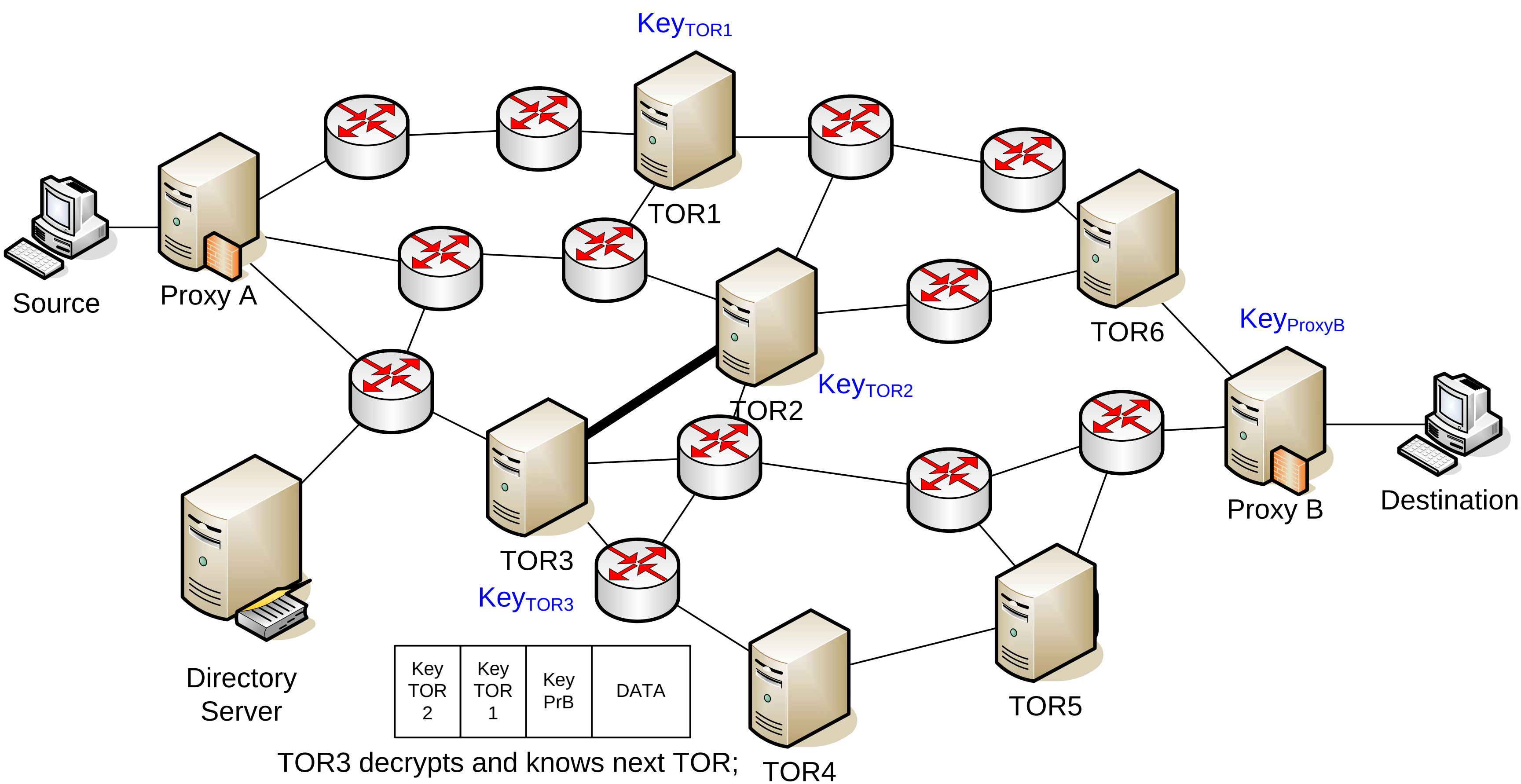
6. After Key Exchange



7. Proxy Sends DATA

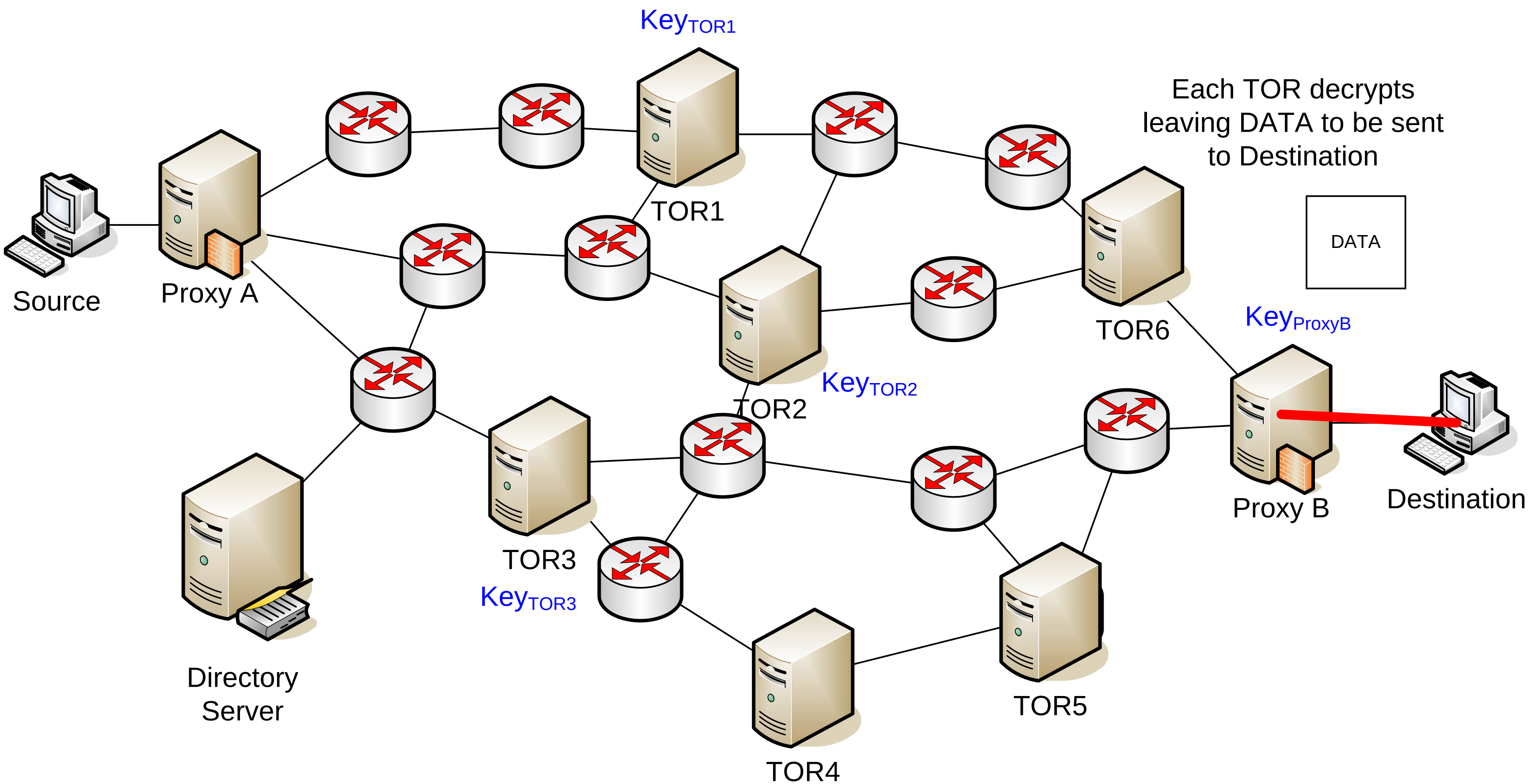


8. TOR3 Sends DATA



TOR3 decrypts and knows next TOR;
Sends to TOR2

9. Proxy B Sends DATA



How does TOR Provide Privacy?

- Routers are unaware of who the original source and final destination are:
  - As the message, including some headers, are encrypted, a TOR router only knows the TOR it received from and the TOR router it sends to
  - A path through more than two TOR routers (as well as using random paths and changing them often) makes it almost impossible for attacker to determine the original source and destination
  - Each TOR router MUST delete onions after short period of time
- TOR can also provide “Hidden Services”
  - A publisher of information (e.g. web server) can anonymously publish information for people to access
  - Requires Rendezvous points (no time to cover the details!)

Summary

- Privacy of Internet communications and behaviour are desired by many users
- Encryption is primary method for achieving private communications in Internet
  - IPsec in network layer solution (implemented in hosts and routers)
  - TLS/SSL is transport layer solution (implemented in hosts only)
- For web security, HTTP is used over TLS (HTTPS)
  - Provides confidentiality, authentication and integrity
- Privacy of behaviour is about hiding what you do and who you do it with!
  - Cookies are a means from enabling state-based web applications
  - But can be used to identify users and track their browsing habits
  - Without special protocols/applications, it is relatively easy to find out about behaviour of users on Internet
- Onion routing (TOR) is an overlay network and application that allows pair of users to hide their communications from others
  - No-one else knows that A and B are communicating
  - TOR does not hide the source from destination, e.g. B still knows A
  - However proxies can be used to remove identifiable information from Internet data, thereby providing some level of anonymous Internet communications