# *Anonymity Online*

**Halil Kemal TAŞKIN**

***hkt.me***

Middle East Technical University

Institute of Applied Mathematics

Department of Cryptography

# Notice

All the information and pictures in this presentation are gathered from public resources

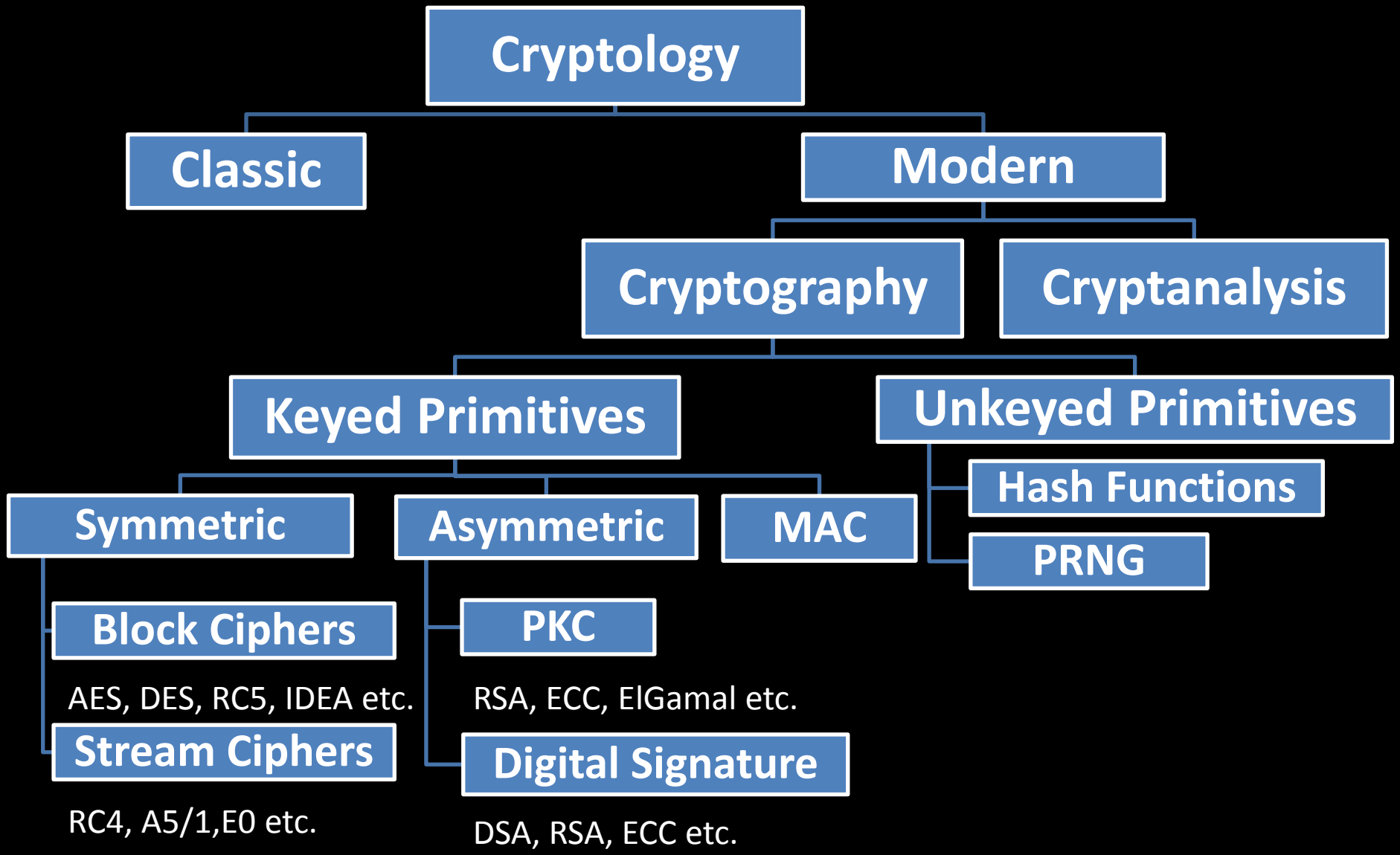
# Credits

- www.torproject.org
- www.i2p2.de
- Roger Dingledine
  - www.freehaven.net
- Paul Syverson
  - www.itd.nrl.navy.mil
  - www.onion-router.net

# Index

# Introduction

- Cryptography
- Computer Networks & Internet
  - How Internet works?
  - Network Security
  - Proxies
- Digital Concerns
  - Security
  - Privacy
  - Anonimity


Cryptology
Classic
Modern
Cryptography
Cryptanalysis
Keyed Primitives
Unkeyed Primitives
Symmetric
Asymmetric
MAC
Hash Functions
PRNG
Block Ciphers
PKC
AES, DES, RC5, IDEA etc.
RSA, ECC, ElGamal etc.
Stream Ciphers
Digital Signature
RC4, A5/1,E0 etc.
DSA, RSA, ECC etc.

# Internet

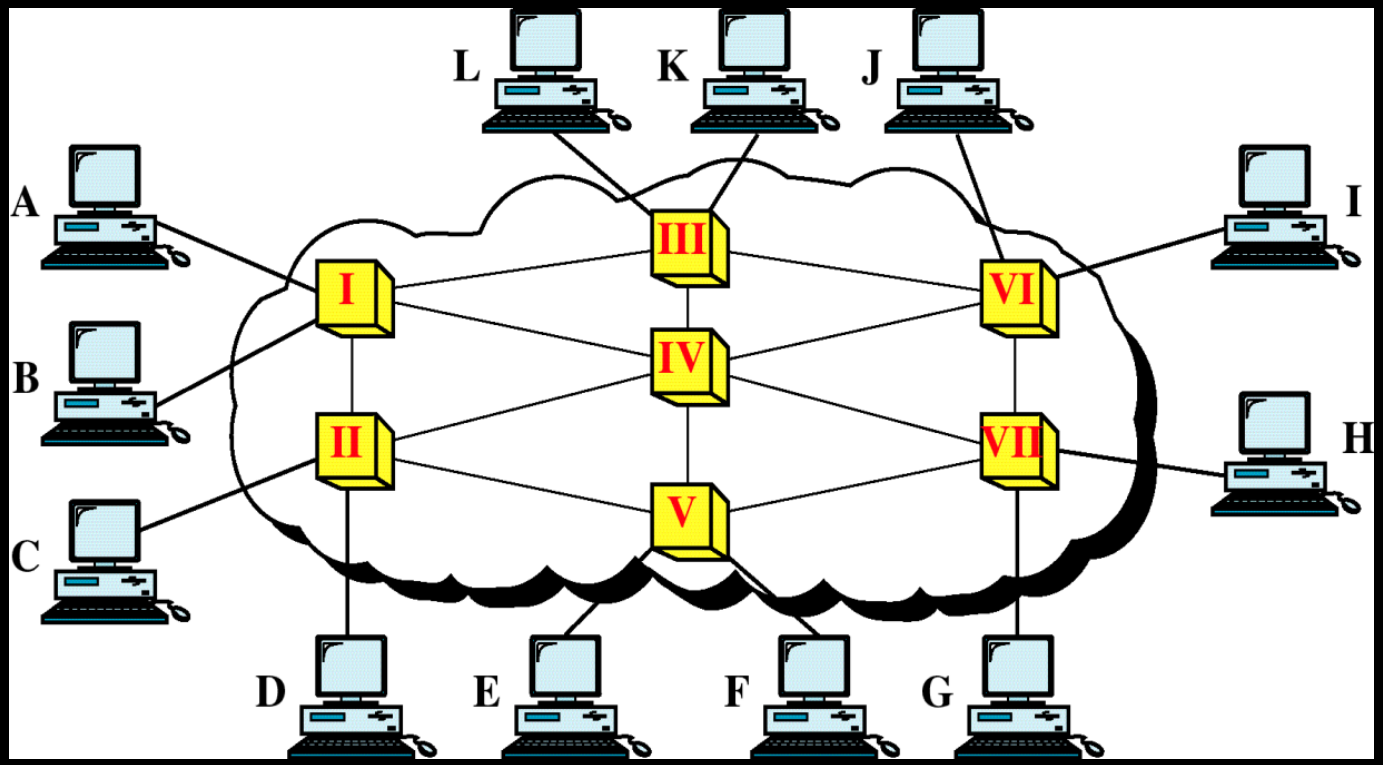
*A little history*

- As always, Alice and Bob want to communicate with each other.

- Circuit Switched Network
- Packet Switched Network

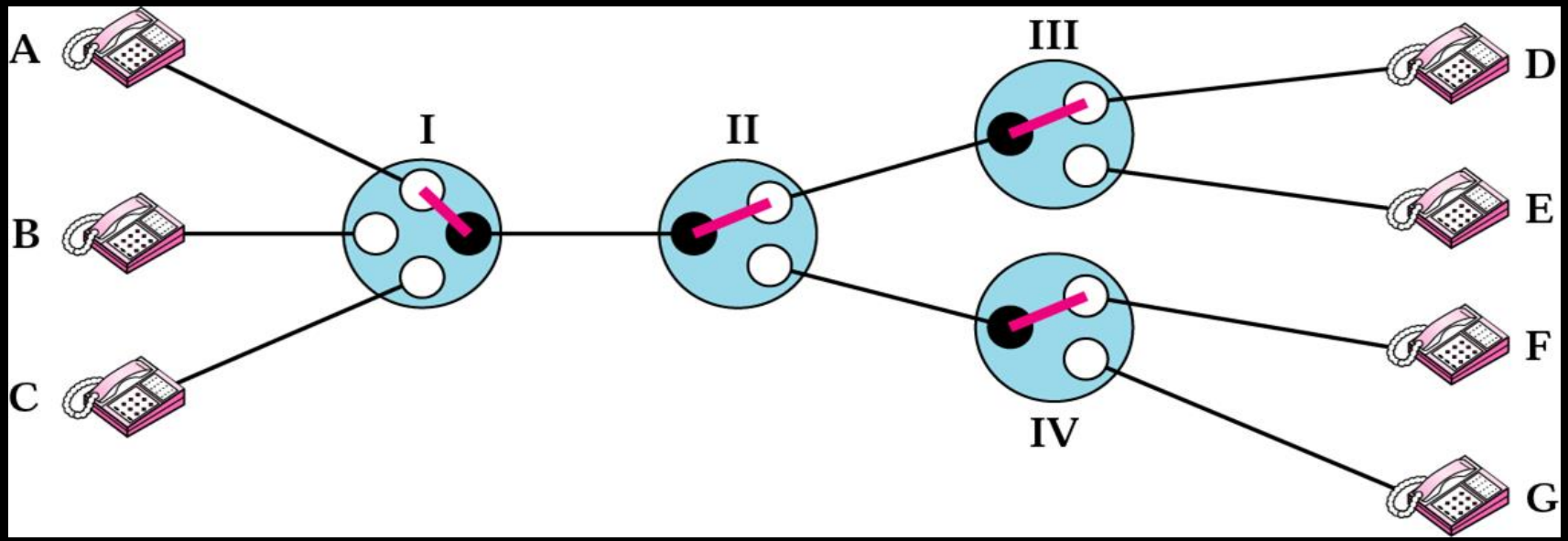
# Circuit Switched Network

- Developed by Bell, Hubbard and Sanders in 1878.
- Primarily designed for audio communication.
- Manual systems were used until 1920s.



# Circuit Switched Network

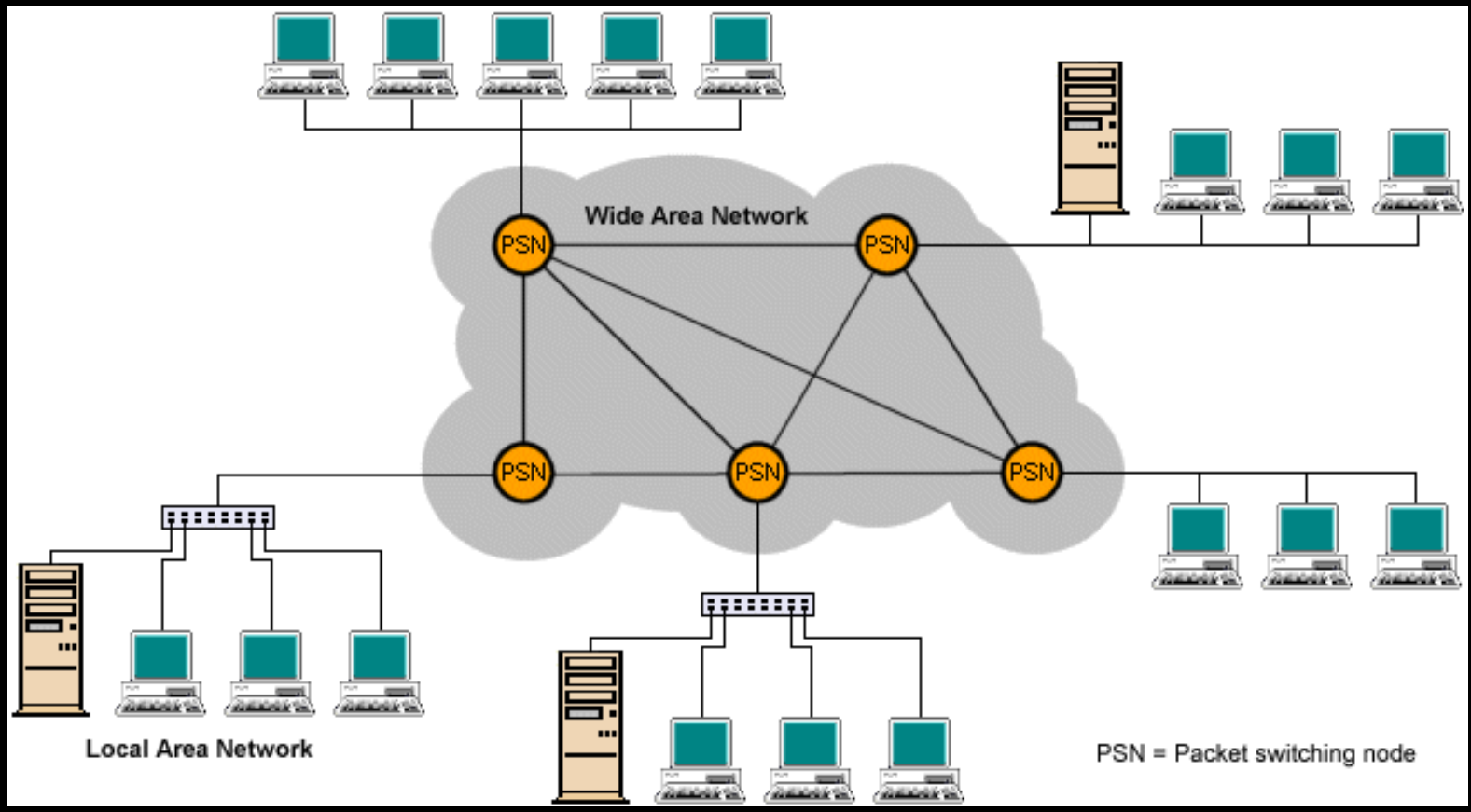
- Creates a direct physical connection between 2 devices such as phones or computers.



# Packet Switched Network

- First proposed for military uses in the early 1960s.
- Developed by  Advanced Research Projects Agency (ARPA) within the U.S. Department of Defense.
- First implementation appeared in 1968 known as ARPANET

# Packet Switched Network



# ARPANET

The initial ARPANET consisted of four nodes:

- University of California, Los Angeles (UCLA)
- Stanford Research Institute
- University of California, Santa Barbara (UCSB)
- University of Utah

# ARPANET

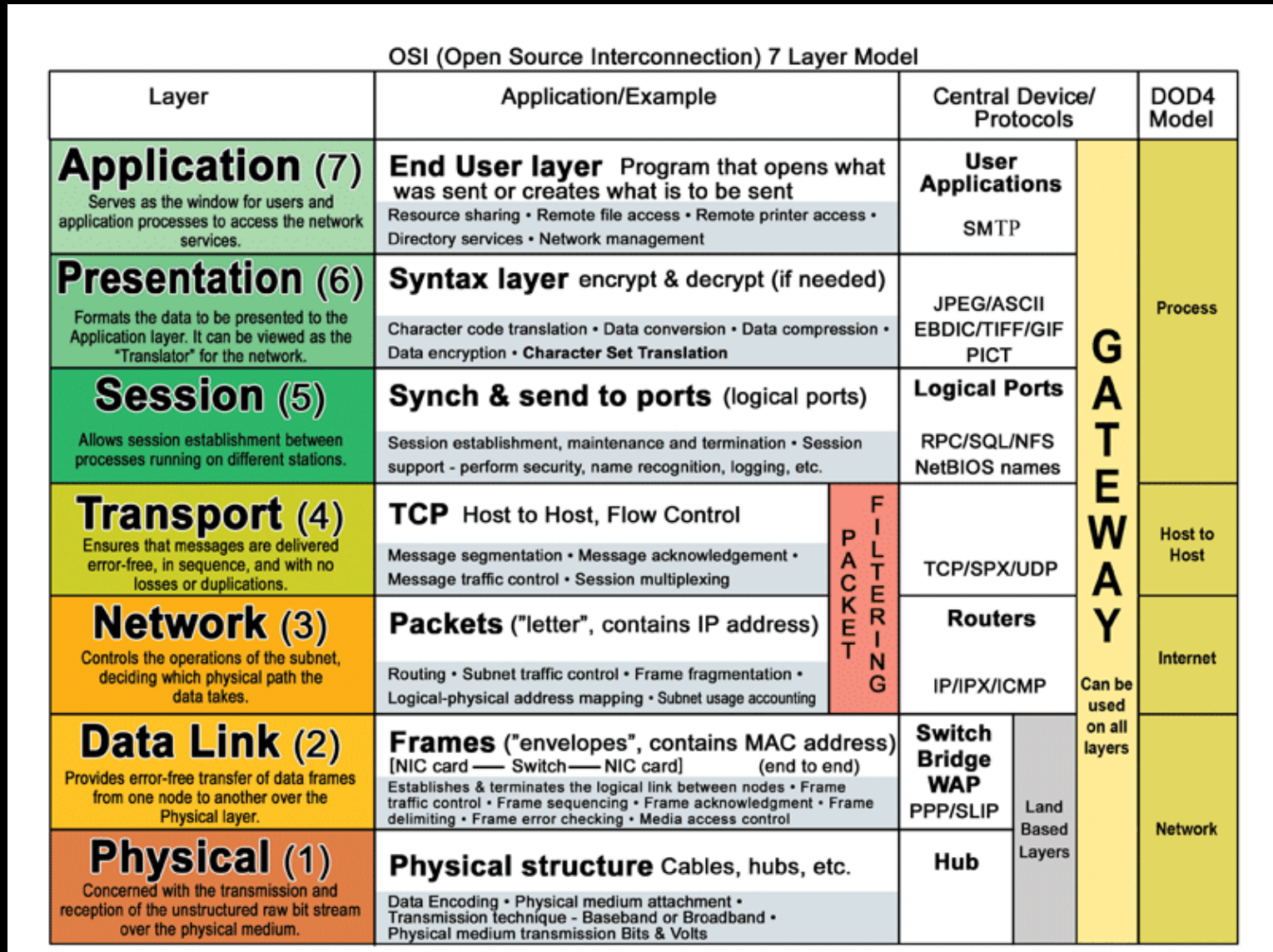
- First message was sent from UCLA to Stanford Research Institute at 22:30 on 29.10.1969
- It was supposed to be *login;*
- But, after sending the *l* and *o* characters, system crashed.
- Hence, *lo* was the first message sent over the ARPANET.

- This was the beginning of today's Internet.

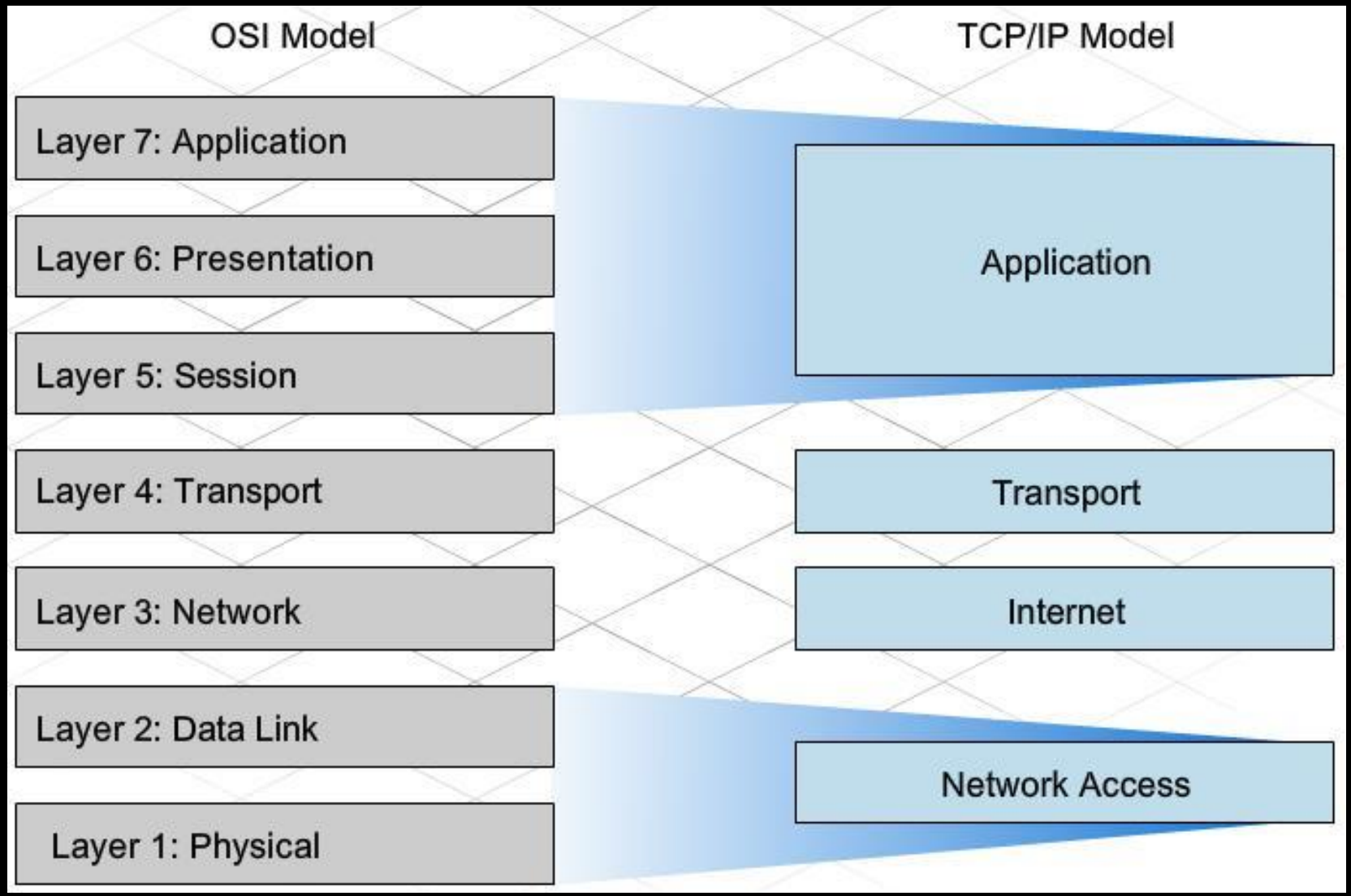
# Models

- Open Systems Interconnection (OSI) model
  - No protocol has yet been developed using this model. It just shows the abstraction layers of a network communication
  - So it is called as OSI Reference Model
- TCP/IP model
  - a.k.a Internet protocol suite
  - a.k.a DoD4 model
  - Set of communications protocols used for the Internet

# OSI


OSI (Open Source Interconnection) 7 Layer Model
Layer
Application/Example
Central Device/ Protocols
DOD4 Model
Application (7)
Serves as the window for users and application processes to access the network services.
End User layer Program that opens what was sent or creates what is to be sent
Resource sharing • Remote file access • Remote printer access • Directory services • Network management
User Applications
SMTP
Presentation (6)
Formats the data to be presented to the Application layer. It can be viewed as the "Translator" for the network.
Syntax layer encrypt & decrypt (if needed)
Character code translation • Data conversion • Data compression • Data encryption • Character Set Translation
JPEG/ASCII
EBDIC/TIFF/GIF
PICT
Session (5)
Allows session establishment between processes running on different stations.
Synch & send to ports (logical ports)
Session establishment, maintenance and termination • Session support - perform security, name recognition, logging, etc.
Logical Ports
RPC/SQL/NFS
NetBIOS names
Transport (4)
Ensures that messages are delivered error-free, in sequence, and with no losses or duplications.
TCP Host to Host, Flow Control
Message segmentation • Message acknowledgement • Message traffic control • Session multiplexing
PACKET FILTERING
TCP/SPX/UDP
Network (3)
Controls the operations of the subnet, deciding which physical path the data takes.
Packets ("letter", contains IP address)
Routing • Subnet traffic control • Frame fragmentation • Logical-physical address mapping • Subnet usage accounting
Routers
IP/IPX/ICMP
Data Link (2)
Provides error-free transfer of data frames from one node to another over the Physical layer.
Frames ("envelopes", contains MAC address)
[NIC card — Switch — NIC card] (end to end)
Establishes & terminates the logical link between nodes • Frame traffic control • Frame sequencing • Frame acknowledgment • Frame delimiting • Frame error checking • Media access control
Switch
Bridge
WAP
PPP/SLIP
Land Based Layers
Physical (1)
Concerned with the transmission and reception of the unstructured raw bit stream over the physical medium.
Physical structure Cables, hubs, etc.
Data Encoding • Physical medium attachment • Transmission technique - Baseband or Broadband • Physical medium transmission Bits & Volts
Hub
GATEWAY
Can be used on all layers
Process
Host to Host
Internet
Network

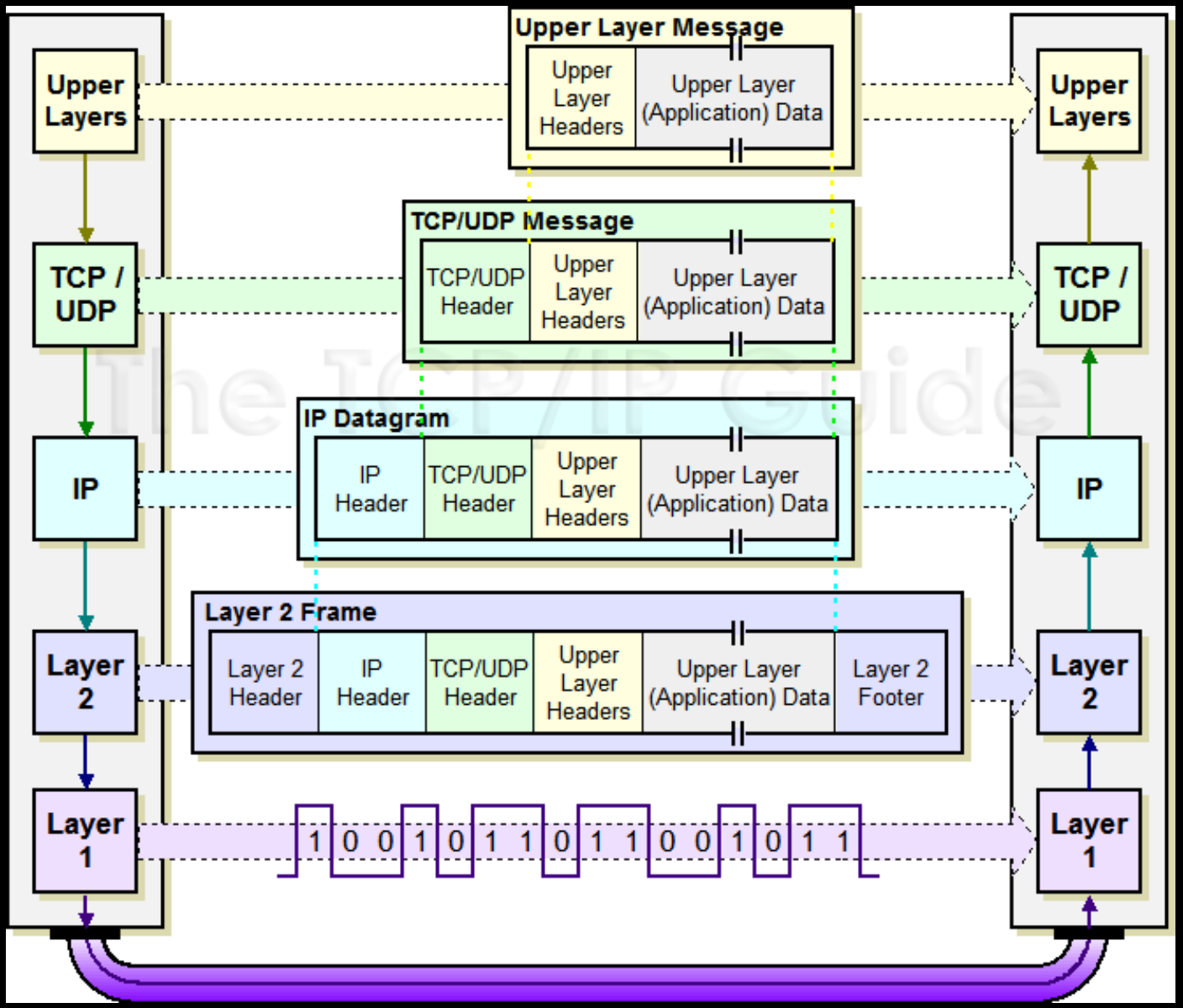
# OSI vs. TCP/IP


OSI Model
TCP/IP Model
Layer 7: Application
Layer 6: Presentation
Layer 5: Session
Layer 4: Transport
Layer 3: Network
Layer 2: Data Link
Layer 1: Physical
Application
Transport
Internet
Network Access

# TCP/IP Model

| | |
|---|---|
| Layer 4: Application | • HTTP (80), HTTPS (443), FTP (20,21), POP3 (110), SMTP (25), DNS (53), DHCP (67,68) |
| Layer 3: Transport | • TCP, UDP |
| Layer 2: Internet | • IP, ICMP, ARP |
| Layer 1: Link | • Twisted-Pair, Co-ax, Fiber, Radio<br>• Ethernet, Wi-Fi (802.11a/b/g/n) |

# TCP/IP Model


Upper Layer Message
Upper Layer Headers
Upper Layer (Application) Data
Upper Layers
TCP/UDP Message
TCP/UDP Header
Upper Layer Headers
Upper Layer (Application) Data
TCP / UDP
IP Datagram
IP Header
TCP/UDP Header
Upper Layer Headers
Upper Layer (Application) Data
IP
Layer 2 Frame
Layer 2 Header
IP Header
TCP/UDP Header
Upper Layer Headers
Upper Layer (Application) Data
Layer 2 Footer
Layer 2
Layer 1
1 0 0 1 0 1 1 0 1 1 0 0 1 0 1 1

# Example of TCP/IP Communication

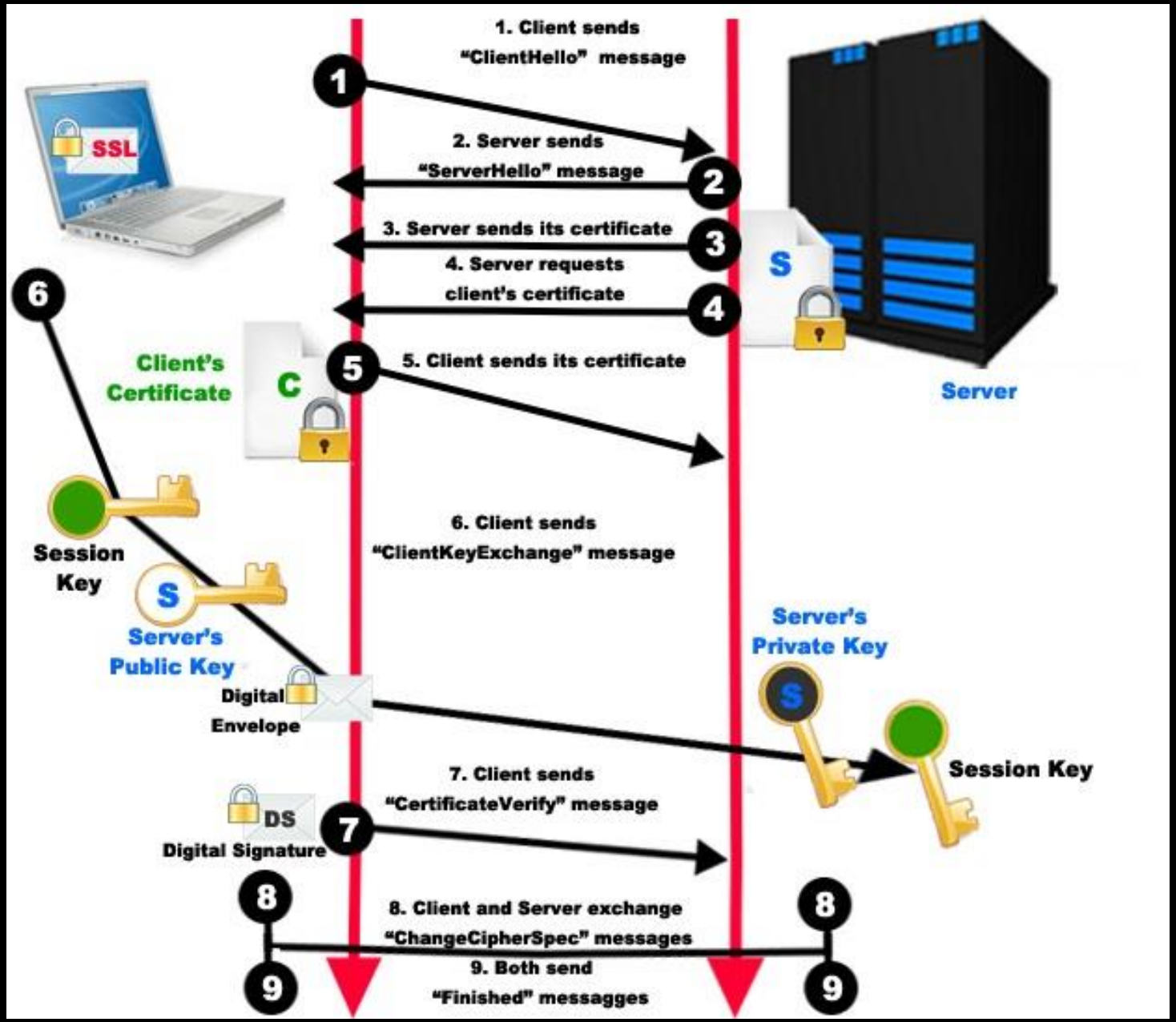
Create a TCP/IP Packet

- Create HTTP Packet
  - Header: «GET / HTTP/1.1\nHost: www.google.com\n\n»
  - Payload: Empty!
- Create TCP Packet
  - Header: Source Port, Destination Port, Checksum, Seq Numbers, Flags.
  - Payload: HTTP Packet
- Create IP Packet
  - Header: Source IP, Destination IP, Checksum, TTL, etc.
  - Payload: TCP Packet
- Create MAC Frame
  - Header: Source MAC, Destination MAC
  - Payload: IP Packet

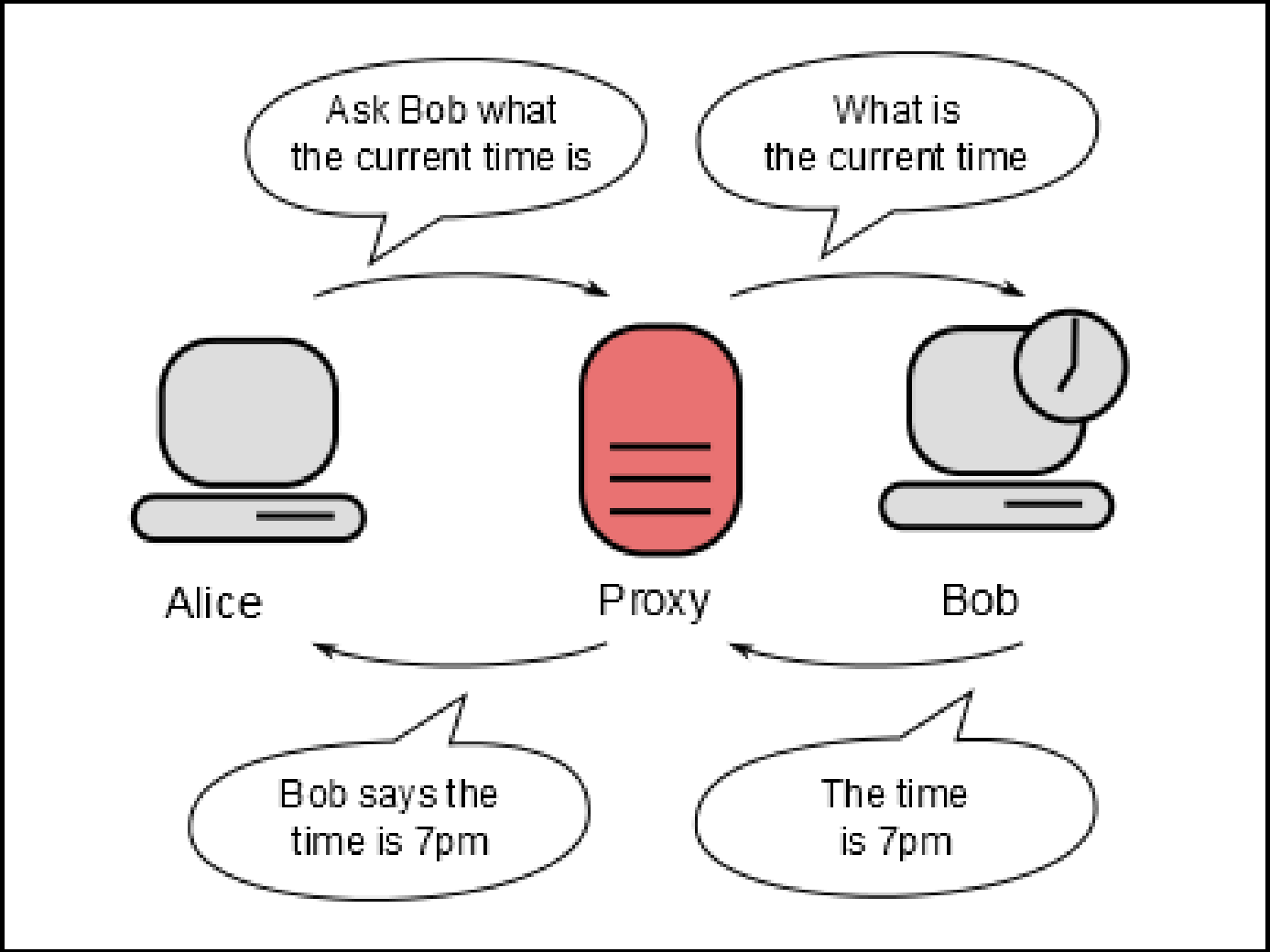
# Network Security

- There is no security in the basics of network communication.
- Design of TCP/IP model is lack of security concerns.
- There are two basic secure communication protocols in the current model
  - Layer 2.5: IPsec
  - Layer 3.5: SSL/TLS

# SSL/TLS


1. Client sends
"ClientHello" message
SSL
2. Server sends
"ServerHello" message
3. Server sends its certificate
4. Server requests
client's certificate
S
Server
6
Client's
Certificate
C
5
5. Client sends its certificate
6. Client sends
"ClientKeyExchange" message
Session
Key
S
Server's
Public Key
Server's
Private Key
Digital
Envelope
S
Session Key
7. Client sends
"CertificateVerify" message
DS
Digital Signature
7
8
8
8. Client and Server exchange
"ChangeCipherSpec" messages
9
9
9. Both send
"Finished" messagges

# Proxy


Ask Bob what
the current time is
What is
the current time
Alice
Proxy
Bob
Bob says the
time is 7pm
The time
is 7pm

# Proxy

Why is it used?

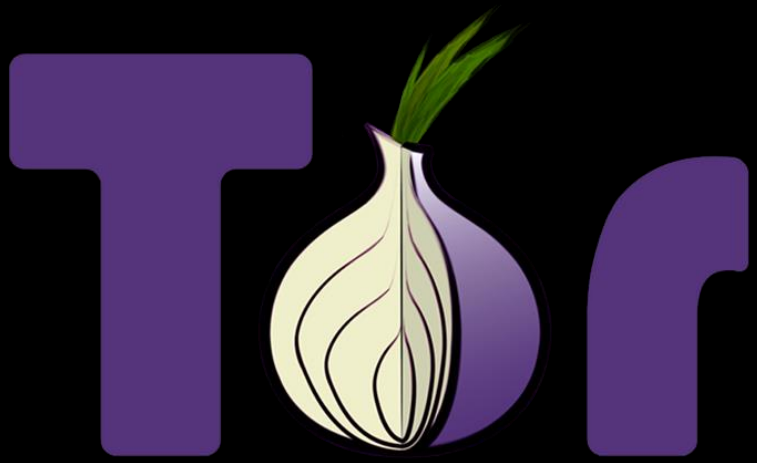
- Security
  - Abstraction of Computers
  - Against viruses/malware
- Access to restricted content
- Speed up access to resources
  - Save bandwidth
- Content Control
- Monitoring

  etc.

# Digital Concerns

- Security
  - Protection of communication from intrusion by an outside user.
- Privacy
  - Privacy is a person's right to control access to his or her personal information.
- Anonimity
  - Anonymity is a condition in which an individual's true identity is unknown.

# The Onion Router



# What is Tor?

- Tor was originally developed with the U.S. Navy for the primary purpose of protecting government communications.
- Tor aims to achieve online anonimity.
- Tor anonymizes the origin of your traffic!

# Who is using Tor?

- Normal people
- Governments
- Militaries and law-enforcement
- Journalists, bloggers, and their audience
- Activists and whistleblowers
- Business executives, IT professionals

- And criminals!

# Threat Model

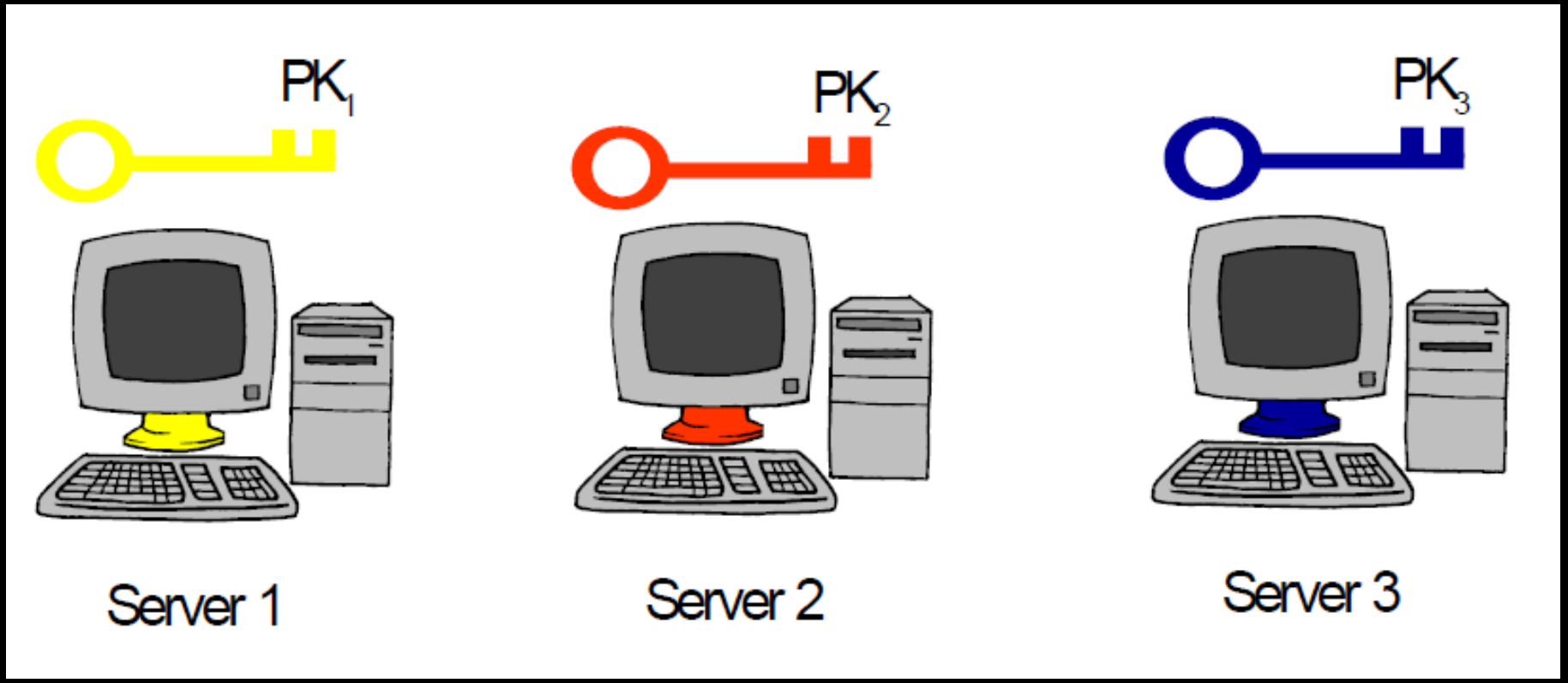
- Tor aims to protect the anonymity of its users from non-global adversaries.

- This means that the adversary has the ability to observe and control some part of the network, but not its totality.

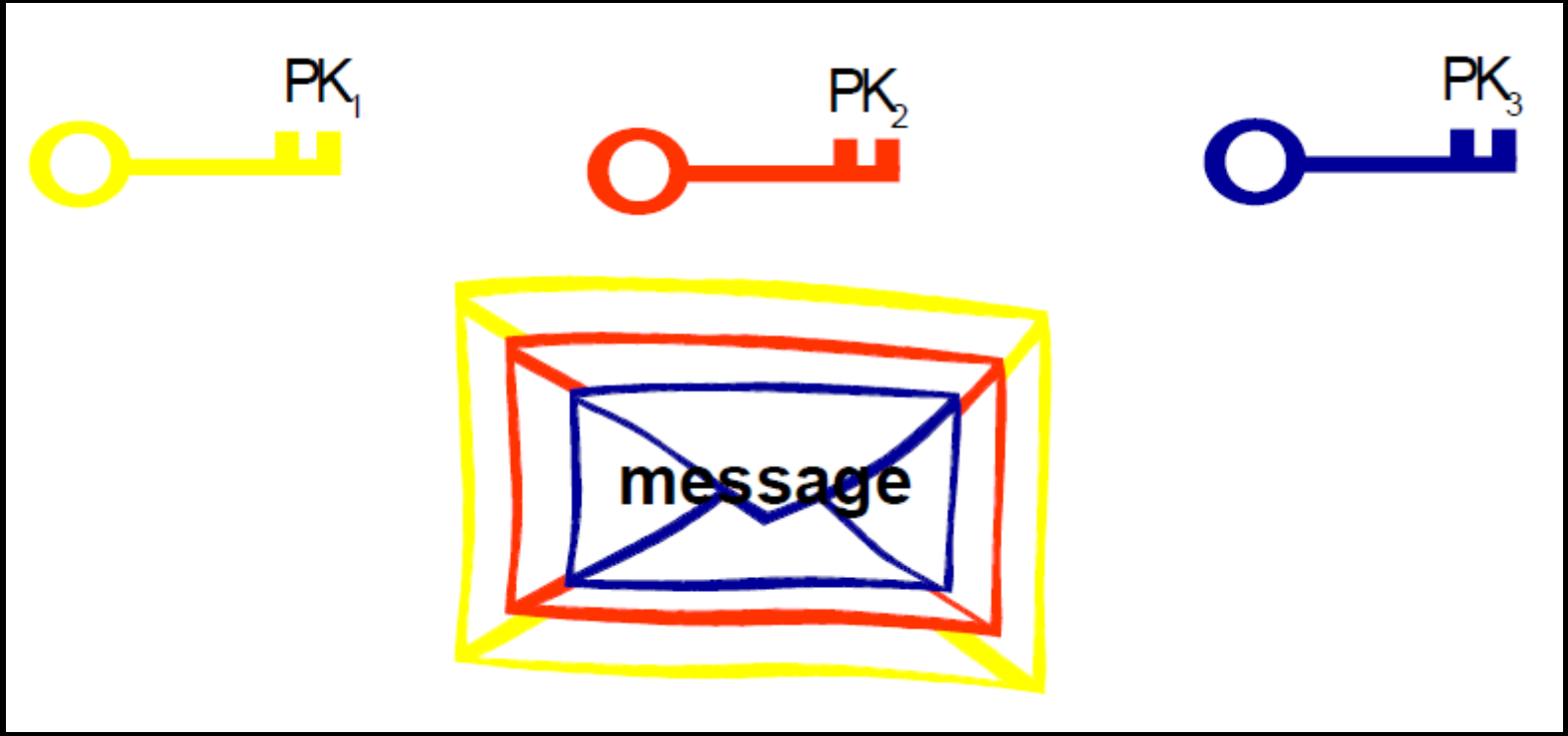
# Digital Mixes

- a.k.a Mix Networks
- Invented in 1981 by David Chaum the founder of IACR.
- The idea is to create hard-to-trace communications by using a chain of proxy servers.

# Mixing

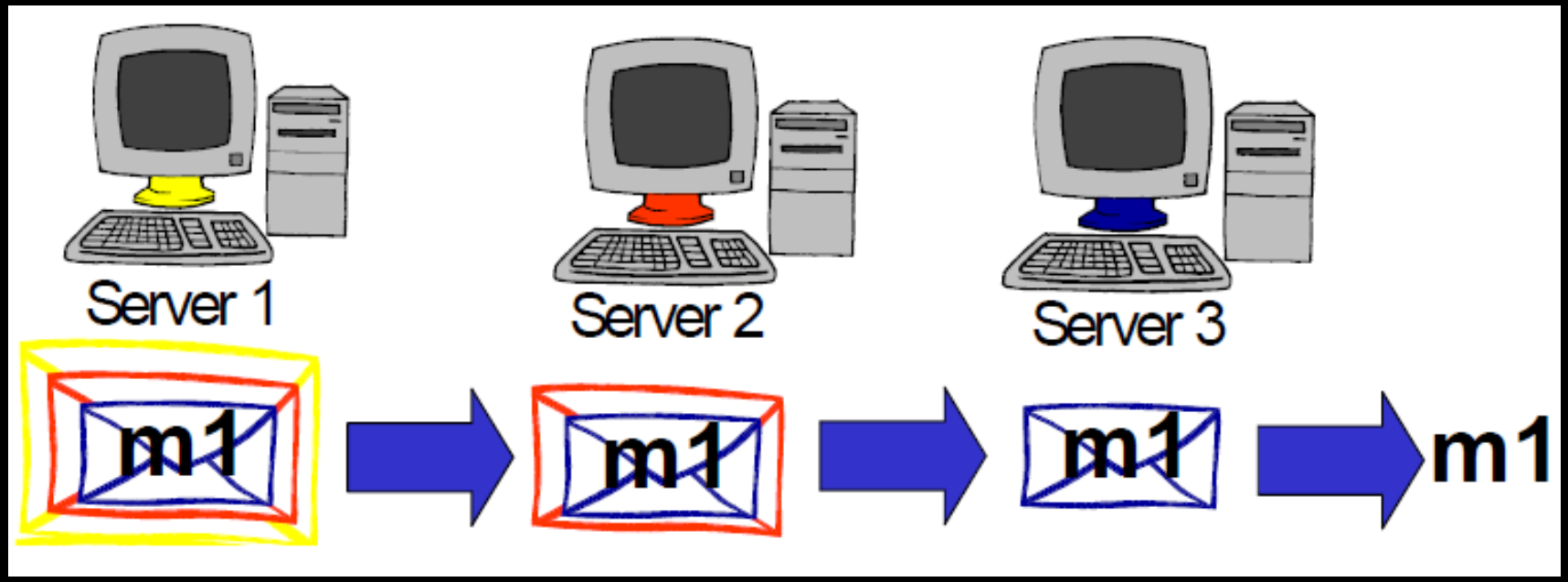


# Mixing




Ciphertext: $E_{PK1}[\ E_{PK2}[\ E_{PK3}[\ \text{message}\ ]\ ]\ ]$

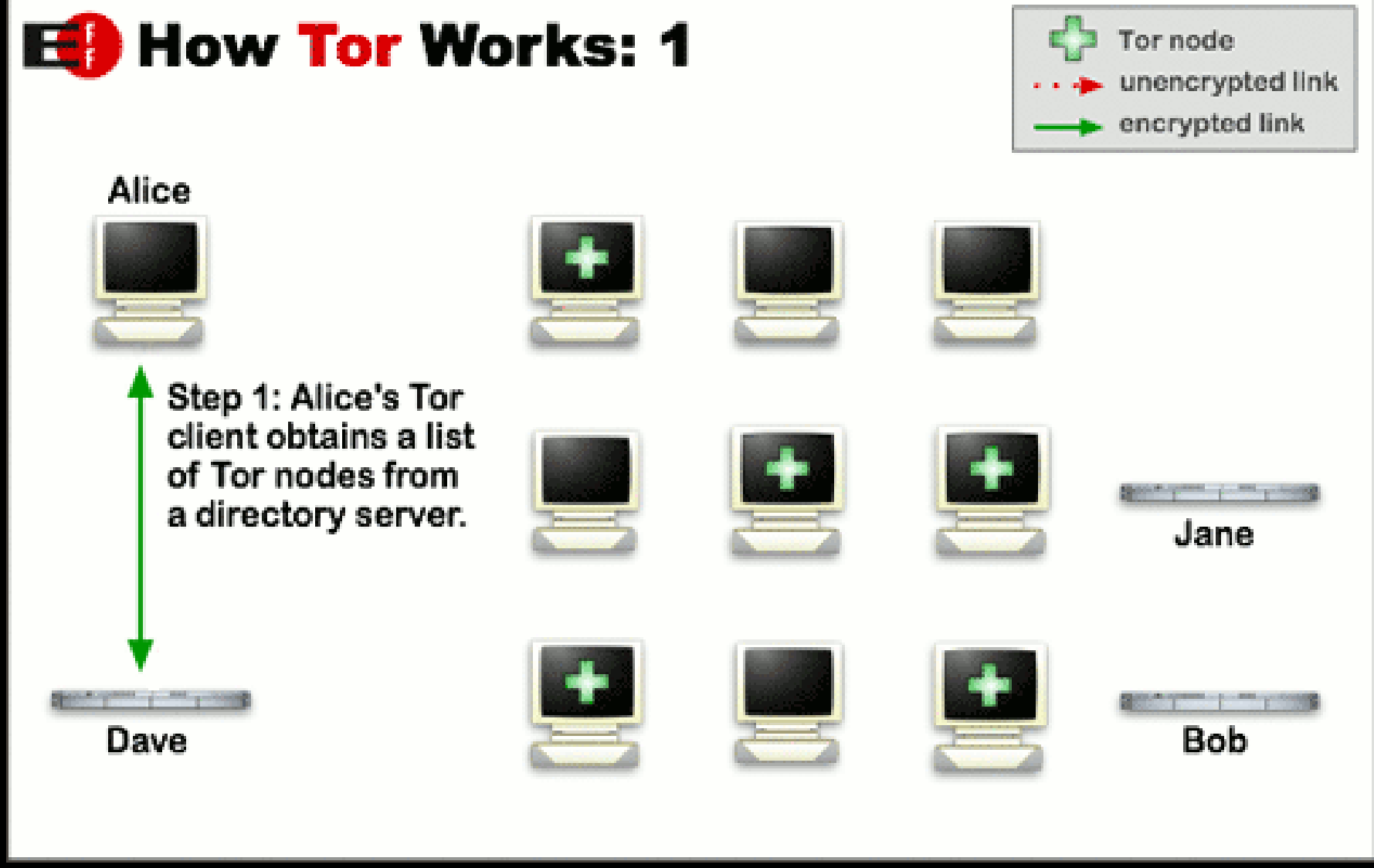
# Mixing



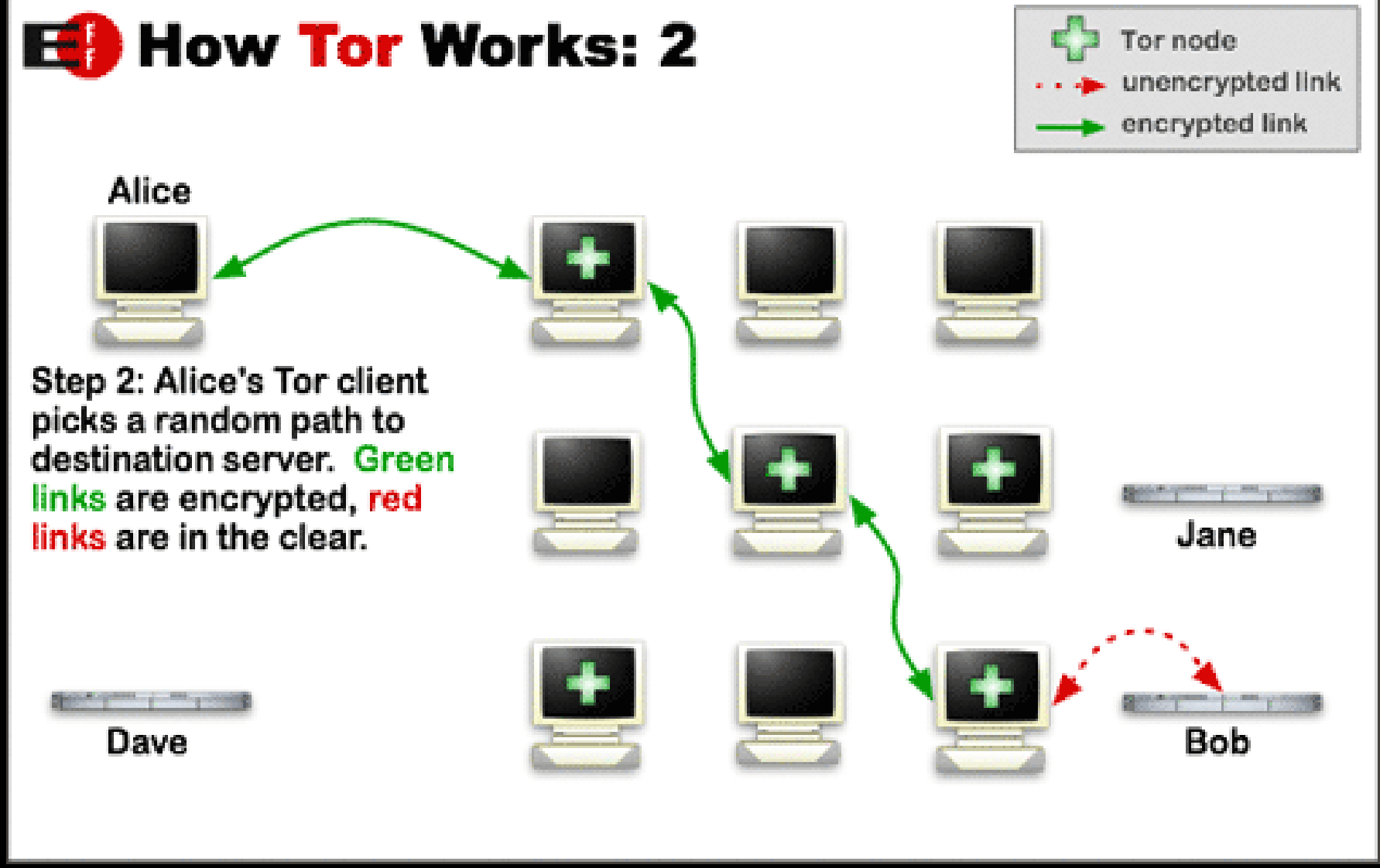
# Onion Routing

- Aim: Traffic Analysis Resistant Communication
- Protect the privacy of the sender and recipient of a message
- Combine Mixes and Proxies
- Use Hybrid Cryptosystem
  - PKC to establish circuits
  - Symmetric-key to transfer data

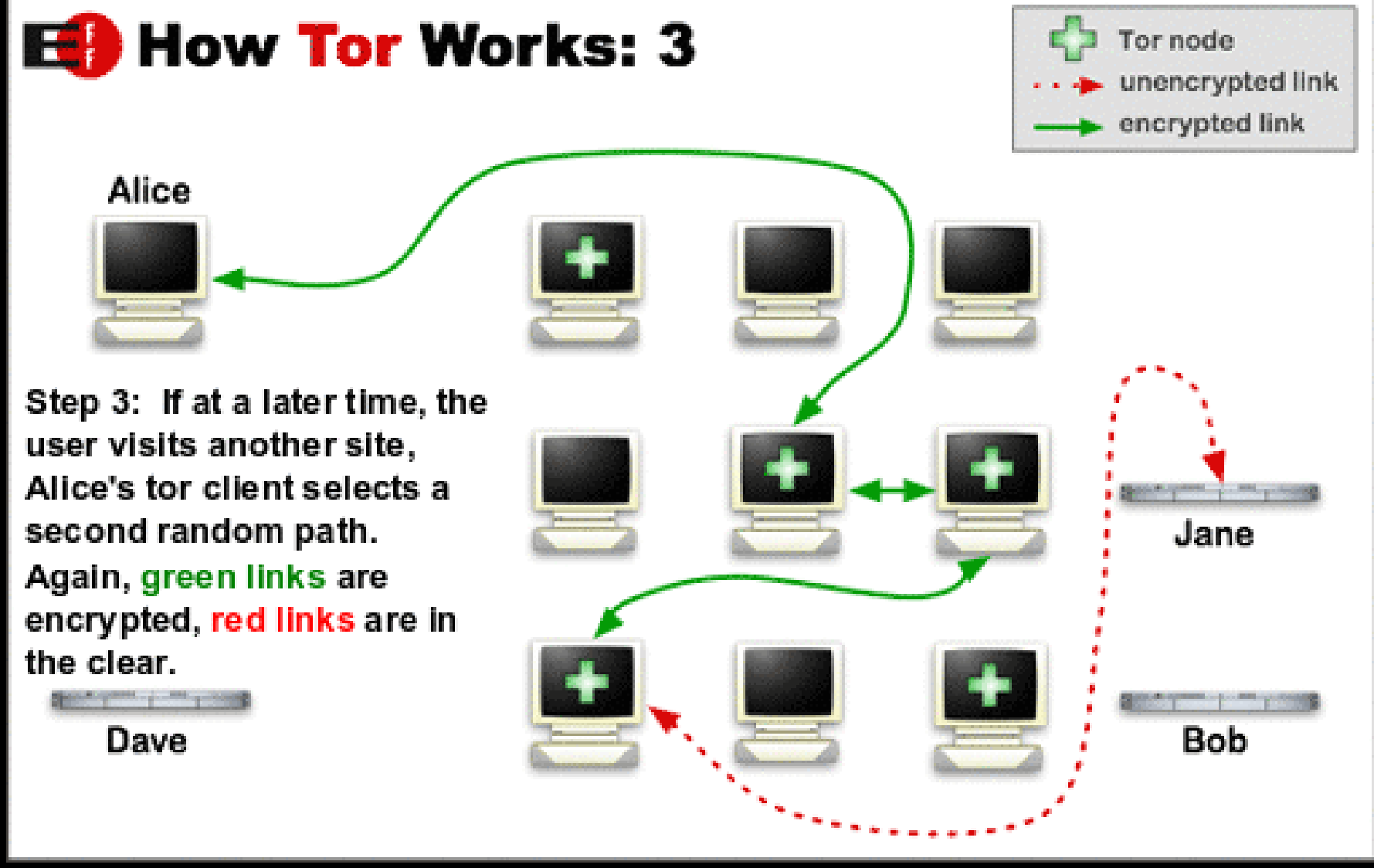
# How it works?


How Tor Works: 1
Tor node
unencrypted link
encrypted link
Alice
Step 1: Alice's Tor client obtains a list of Tor nodes from a directory server.
Jane
Dave
Bob

# How it works?


How Tor Works: 2
Tor node
unencrypted link
encrypted link
Alice
Step 2: Alice's Tor client picks a random path to destination server. Green links are encrypted, red links are in the clear.
Jane
Dave
Bob

# How it works?


How Tor Works: 3
Tor node
unencrypted link
encrypted link
Alice
Step 3: If at a later time, the user visits another site, Alice's tor client selects a second random path.
Again, green links are encrypted, red links are in the clear.
Dave
Jane
Bob

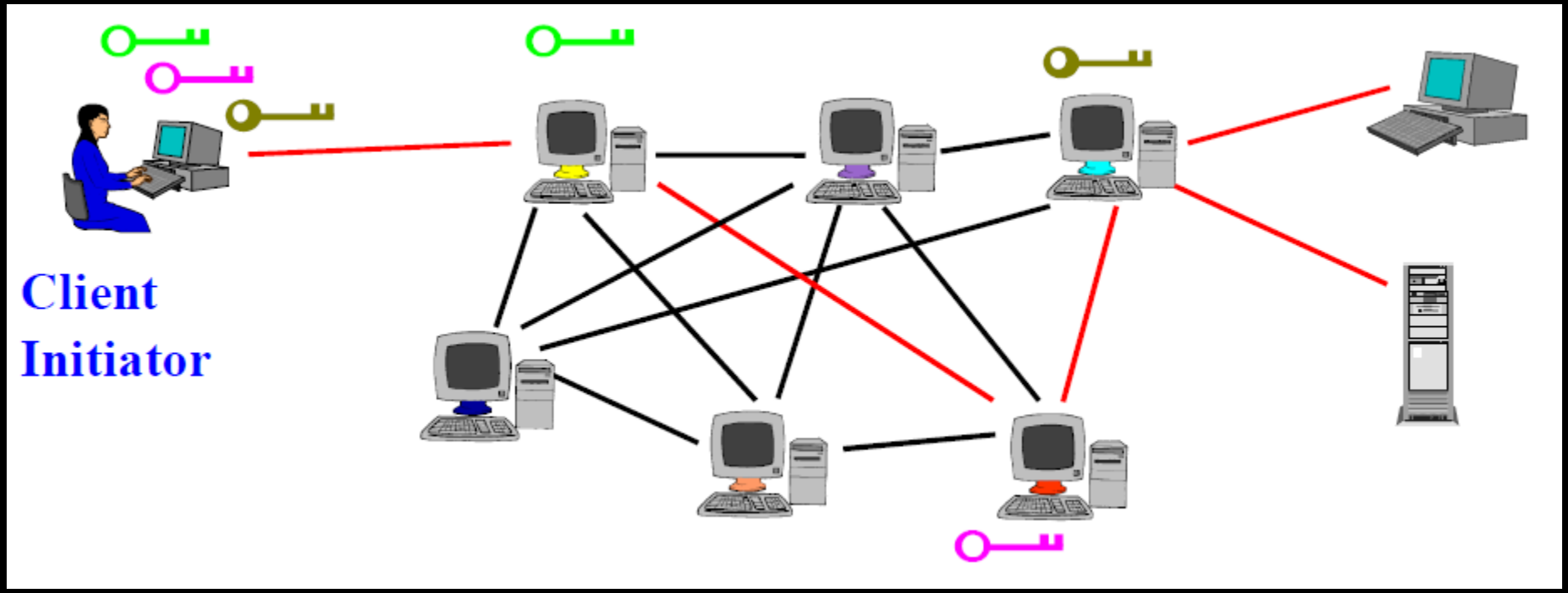
# Tor Specification

*Tor uses a stream cipher, a public-key cipher, the Diffie-Hellman protocol, and a hash function.*

- For a stream cipher, Tor uses 128-bit AES in counter mode, with an IV of all 0 bytes.
- For a public-key cipher, Tor uses RSA with 1024-bit keys and a fixed exponent of 65537.
- For Diffie-Hellman, Tor usea a generator (g) of 2.
  - For the modulus (p), Tor uses the 1024-bit safe prime from rfc2409 section 6.2
- For hash function, Tor uses SHA1

# Establish a Circuit

- Create and Extend Circuit



# Creating Circuits

When creating a circuit through the network, the circuit creator (OP) performs the following steps:

- Choose an onion router as an exit node $OR_{N.}$(N=5)

- Choose a chain of N-1 onion routers
  - $OR_1$...$OR_{N-1}$ to constitute the path, s.t. no router appears in the path twice.

- If not already connected to the first router in the chain, open a new connection to that router.

# Creating Circuits

- Choose a circID not already in use on the connection with the first router in the chain.
  - Send a CREATE cell along the connection, to be received by the first onion router.
- Wait until a CREATED cell is received
  - Finish the handshake
- For each subsequent onion router OR ($OR_2$ through $R_{N-1}$), extend the circuit to $R_N$.

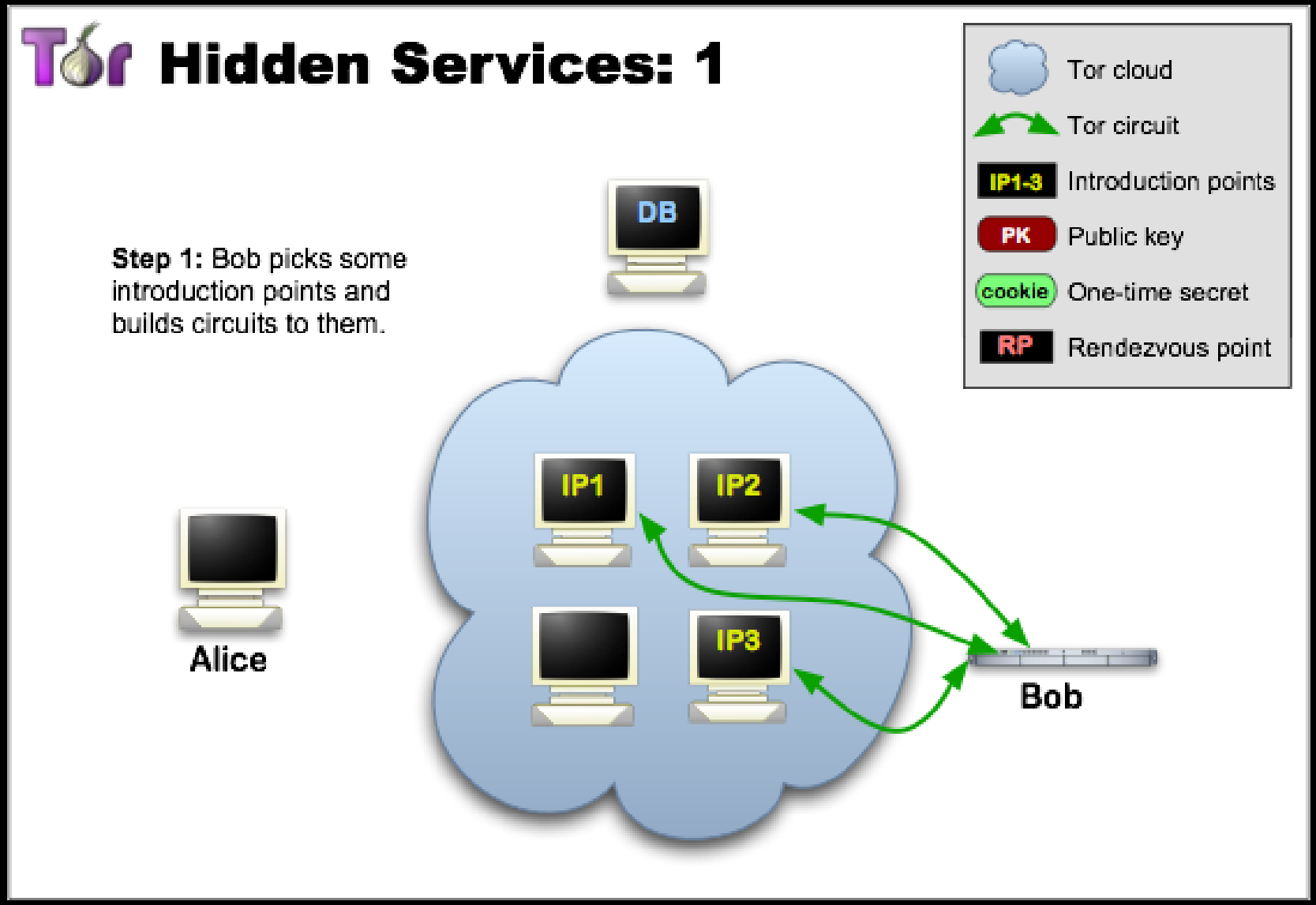
# How to Create a Circuit

- Create an SSL/TLS Connection
- Challenge-Response between the OP and $OR_1$
- OP Generates
  - Symmetric key: K
  - DH data: $g^x$
- OP sends to $OR_1$
  - $E_{PK1}$[**K || ($g^x$ part 1)**]
  - $E_K$(**$g^x$ part 2**)
- $OR_1$ responds
  - $g^y$
  - KH verification value from KDF-TOR
- OP computes $g^{xy}$
  - Use KDF-TOR to obtain shared secret.
  - Verify KH value

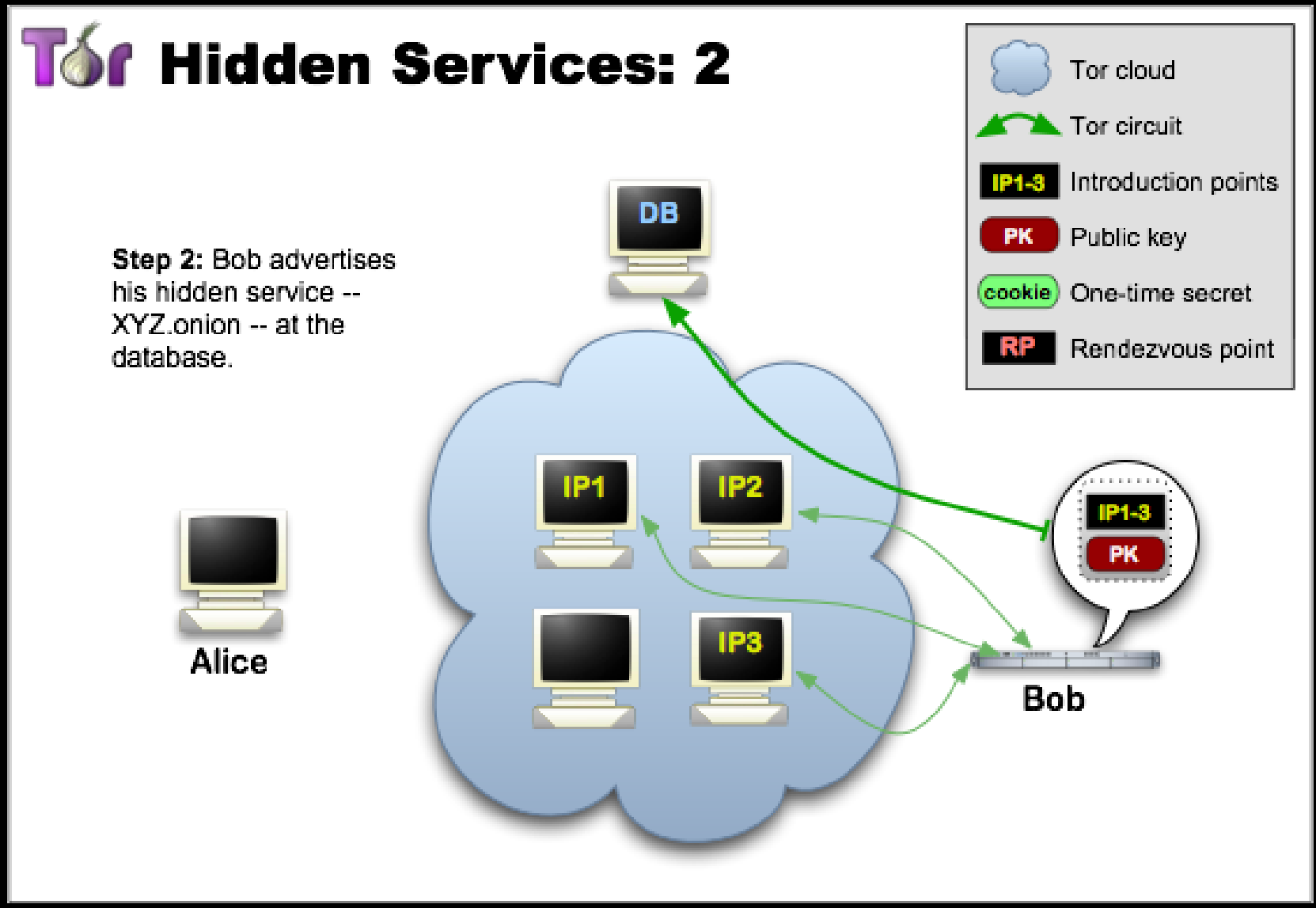
# Tor Hidden Services

- Tor makes it possible for users to hide their locations while offering various kinds of services, such as web publishing.
- Tor users can connect to these hidden services, each without knowing the other's network identity.
- Uses .onion TLDs

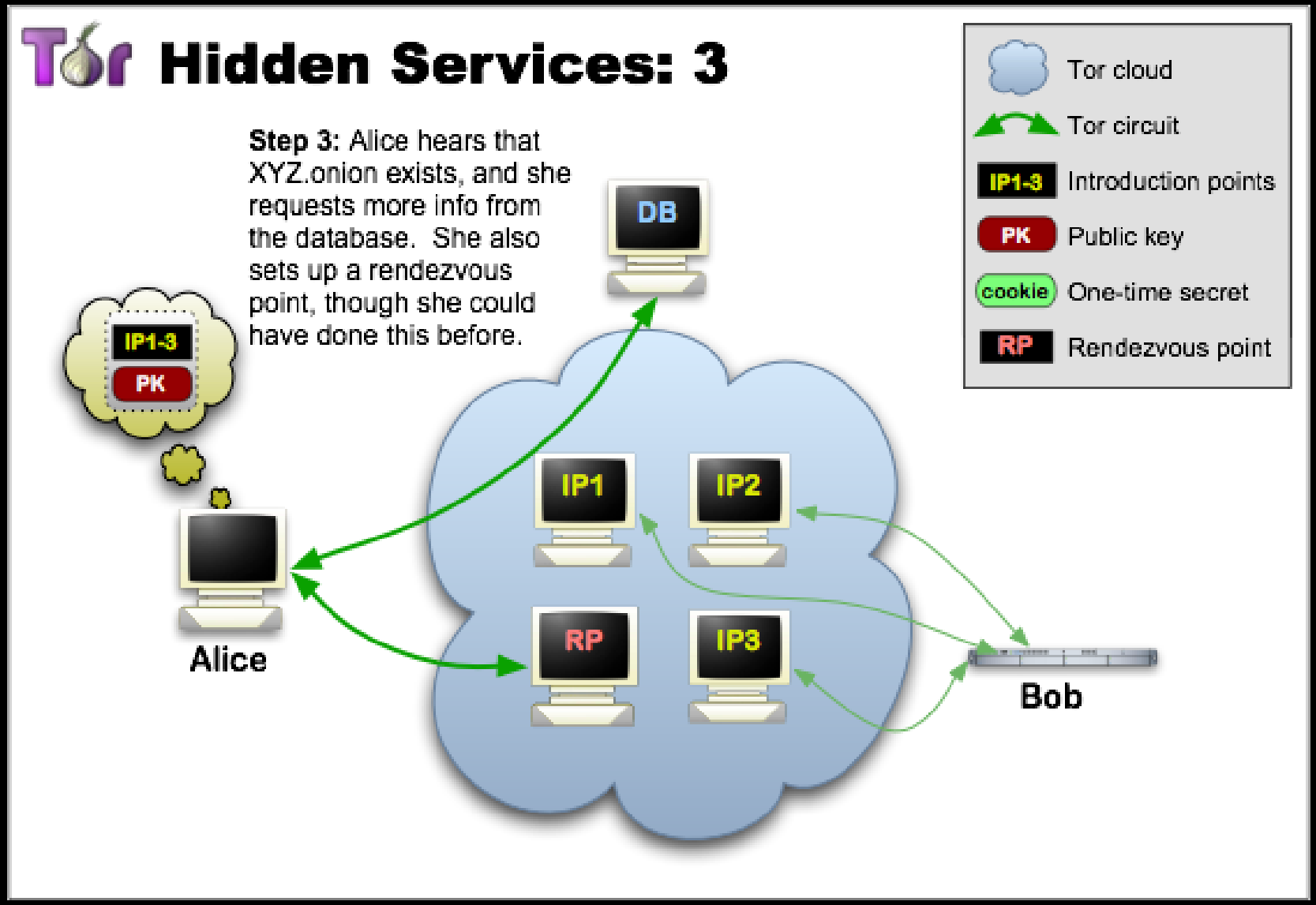
# Tor Hidden Services



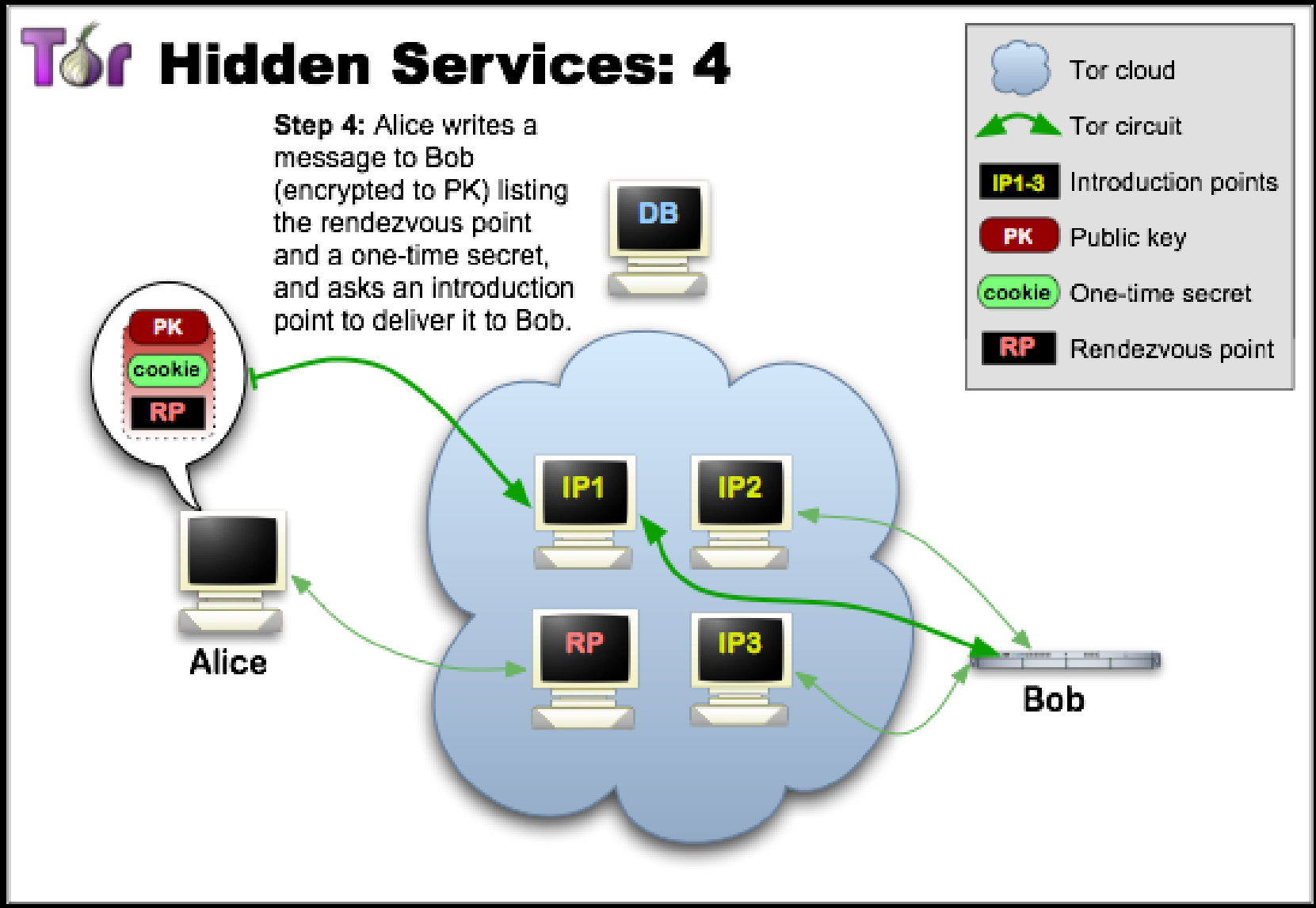
# Tor Hidden Services



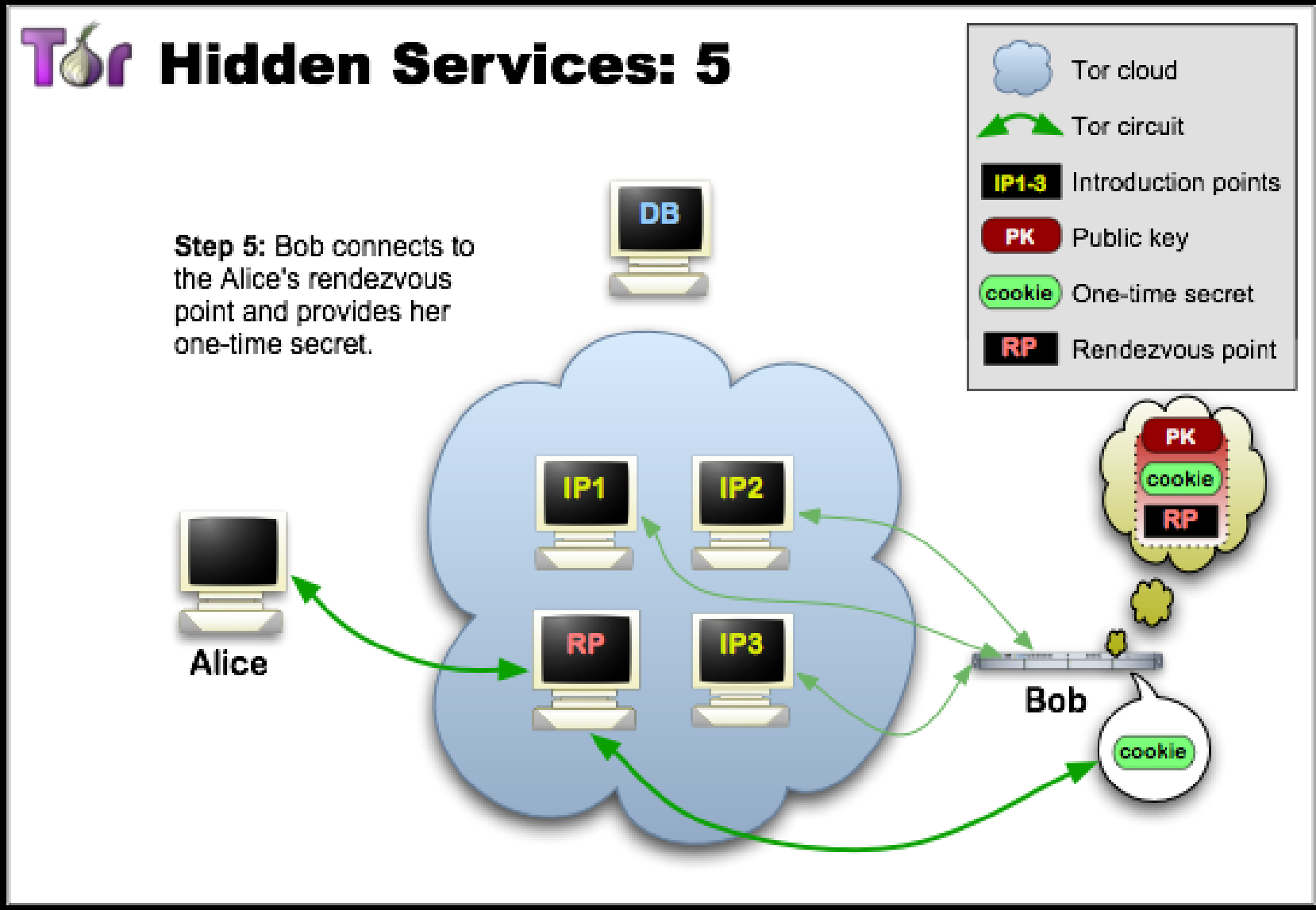
# Tor Hidden Services



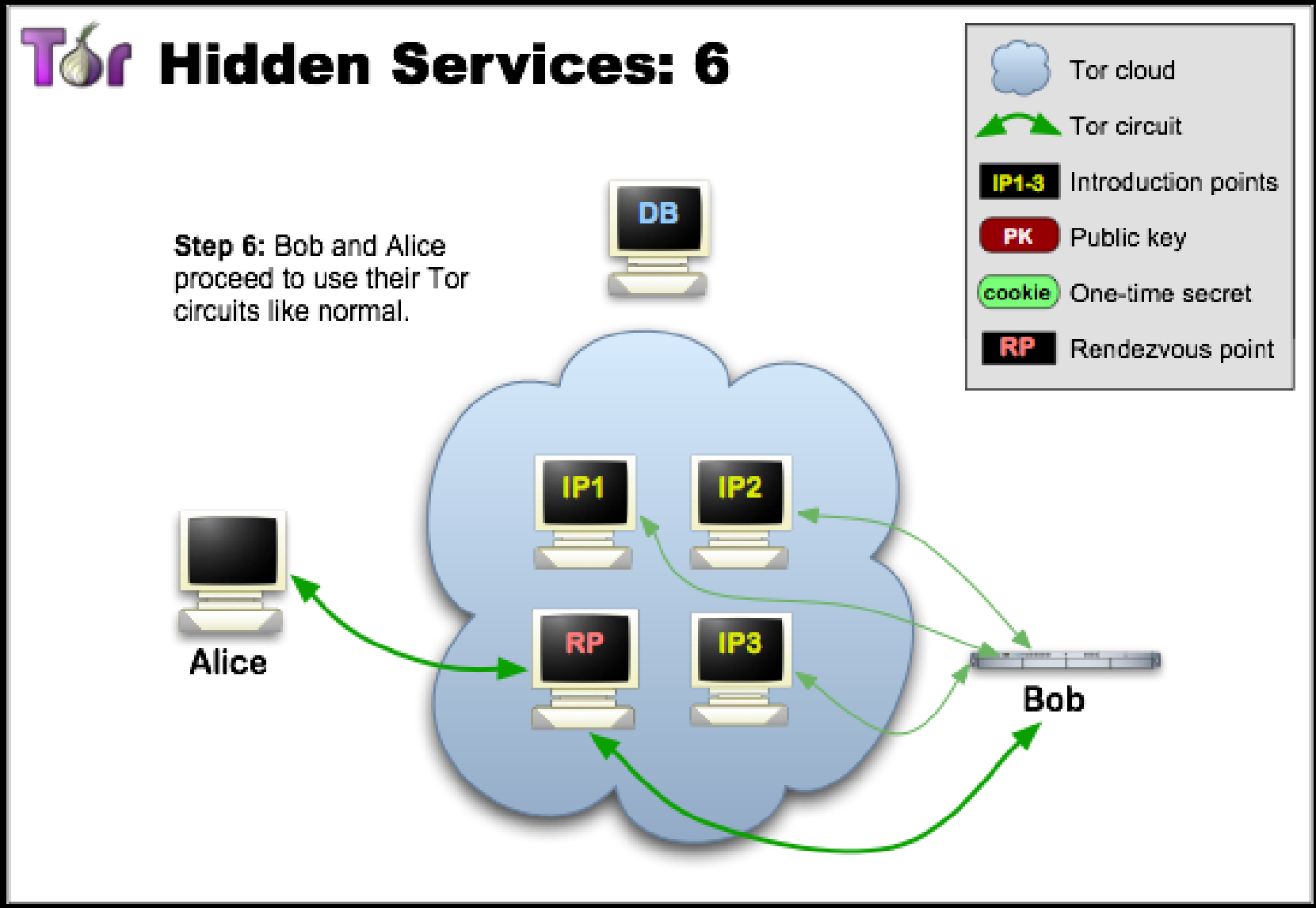
# Tor Hidden Services



# Tor Hidden Services



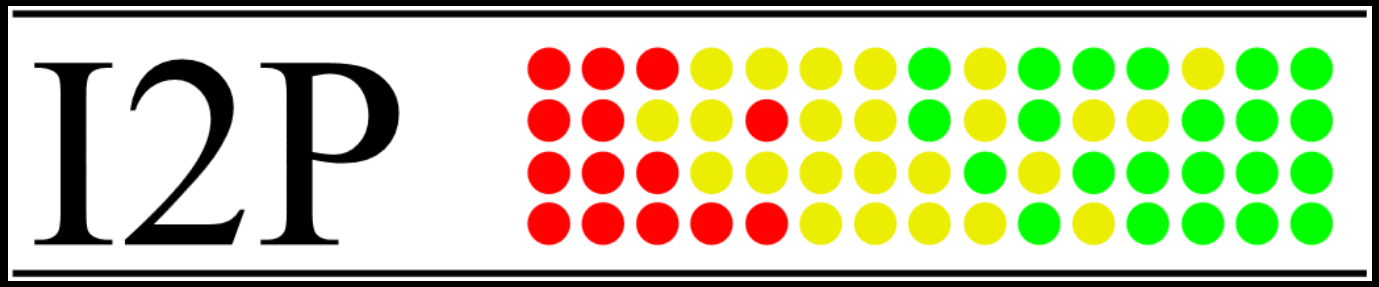
# Tor Hidden Services



# Tor Tools

- Tails
- Tor Browser Bundle (Windows / Linux / OS X)
- Vidalia
- Tor Cloud
- Tor2web
- Orbot: Tor on Android
- Onion Browser for iPhone

# Invisible Internet Project



# What is I2P?

- I2P is an anonymizing network, offering a simple layer that identity-sensitive applications can use to securely communicate.
- All data is wrapped with several layers of encryption, and the network is both distributed and dynamic, with no trusted parties.
- Uses .i2p TLDs.

# Threat Model

- Model based on Tor
- I2P software will not make you indistinguishable from people that don't use computers or who are not on the Internet.
- Instead, I2P is supposted to provide sufficient anonymity to meet the real needs.
  - Browse web pages
  - Exchange data
  - Fear of discovery by powerful organizations or states.

# Garlic Concept

When referring to I2P, the term "garlic" may mean one of three things:

- Layered Encryption
- Bundling multiple messages together
- ElGamal/AES Encryption

# Garlic Concept

- Garlic Routing
  - For building and routing through tunnels
- Garlic Bundling
  - For determining the success or failure of end to end message delivery
- Garlic Encryption
  - For publishing some network database entries (dampening the probability of a successful traffic analysis attack)

# Crypto @ I2P

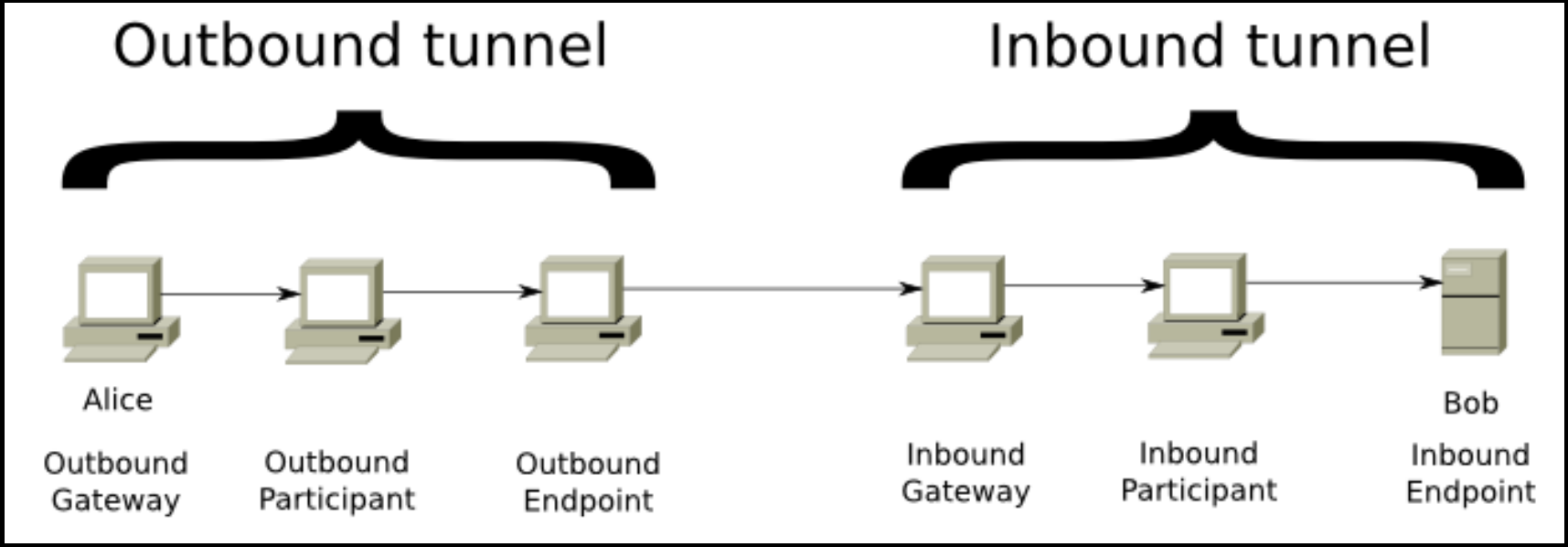
- ElGamal is used for asymmetric encryption.
- AES-256-CBC is used for symmetric encryption.
- 1024-bit DSA used for signing.
- DH is used for key agreement with RFC3526 2048bit MODP group (#14).
- SHA256 is used for hashing.

# How I2P works?

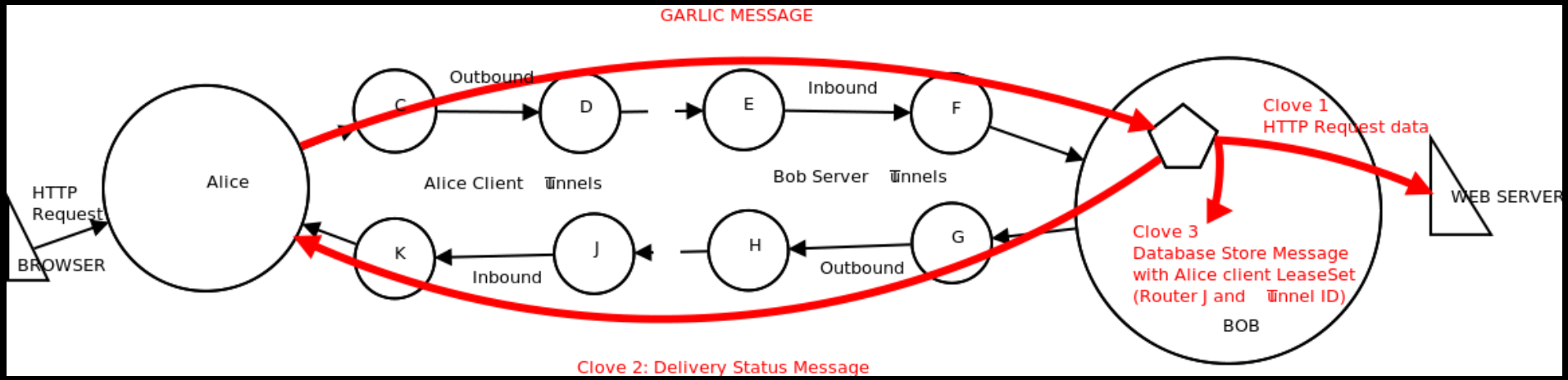
Two types of tunnels exist:

- **outbound** tunnels send messages away from the tunnel creator.
- **inbound** tunnels bring messages to the tunnel creator.

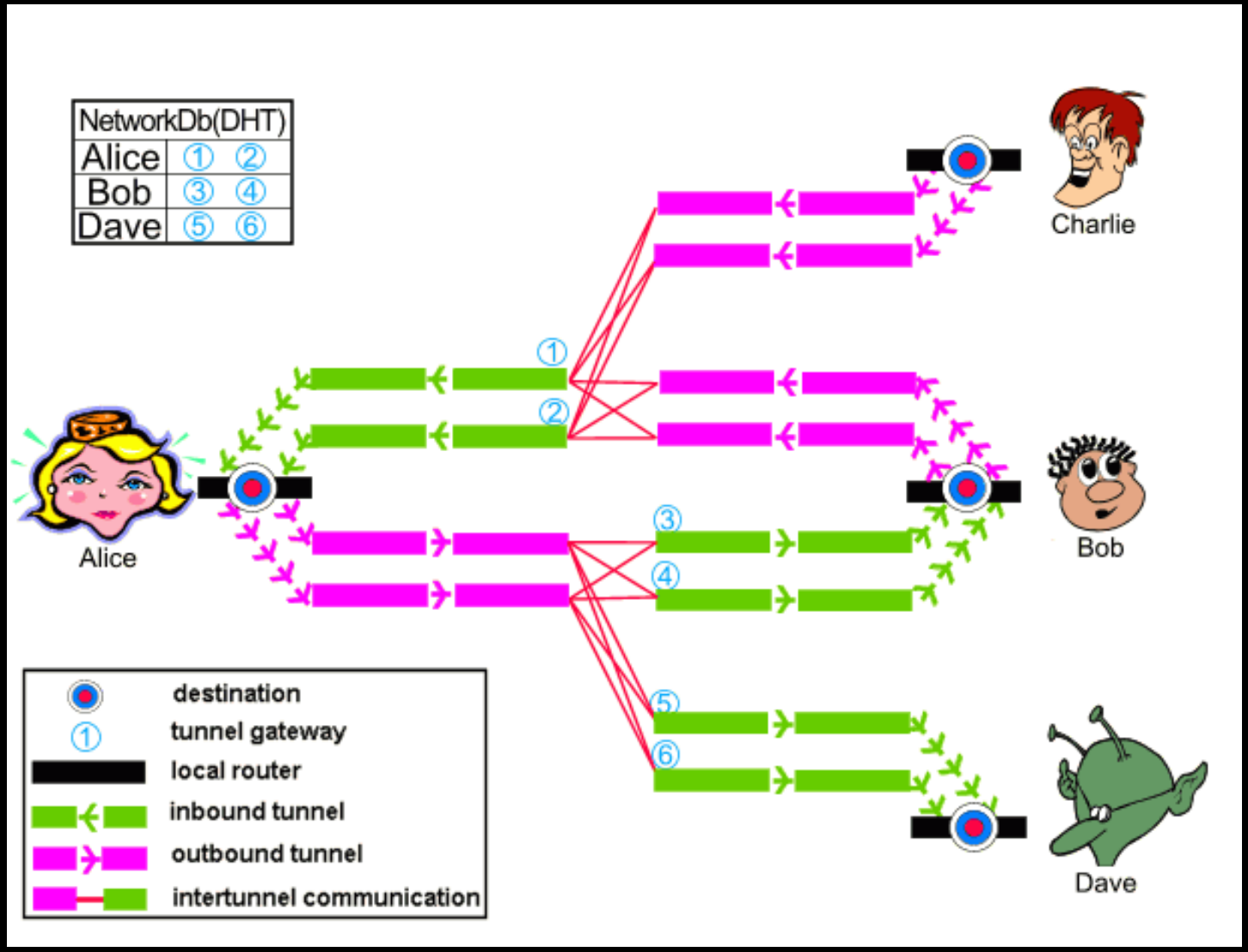
# How I2P works?



# Garlic Message



# I2P Example


NetworkDb(DHT)
Alice ① ②
Bob ③ ④
Dave ⑤ ⑥
Charlie
Alice
Bob
Dave
destination
tunnel gateway
local router
inbound tunnel
outbound tunnel
intertunnel communication

# I2P Services

- Invisible Internet & Public Web Browsing
- Anonymous Encrypted Web Hosting (eepsites)
- Anonymous E-Mail
- Anonymous File Transfer
- Anonymous Chat
- Forums & Blogging

# I2P Tools

- I2P (Windows / Linux / OS X)
  - routerconsole
  - susidns
  - susimail
  - i2psnark
  - i2ptunnel
    - I2P Webserver for eepsite
    - I2P HTTP Proxy
    - I2P HTTPS Proxy

# *Benefits of Tor over I2P*

- Much more visibility in the academic and hacker communities
- Has significant funding
- Designed and optimized for exit traffic, with a large number of exit nodes
- Centralized control reduces the complexity at each node

# *Benefits of I2P over Tor*

- Designed and optimized for hidden services, which are much faster than in Tor
- Fully distributed and self organizing
- Peers are selected by continuously profiling and ranking performance, rather than trusting claimed capacity
- Unidirectional tunnels instead of bidirectional circuits

# Tor & I2P Hands on

# References / Sources / Links

- Tor Project
  - www.torproject.org
- I2P
  - www.i2p2.de
- Onion Routing
  - www.onion-router.net
- The Free Haven Project
  - www.freehaven.net
- Bitcoin
  - www.bitcoin.org

# References / Sources / Links

- Roger Dingledine
  - www.freehaven.net/~arma
- Paul Syverson
  - www.syverson.org
- David Chaum
  - www.chaum.com

# References / Sources / Links

- A Brief History of the Internet
  - www.pbs.org/opb/nerds2.0.1
- Cipherspaces/Darknets An Overview of Attack Strategies
  - www.youtube.com/watch?v=xxKXkbohRZM
- 24C3: To be or I2P
  - www.youtube.com/watch?v=TsfdzfGZyu0

# Thanks for your attention

***Questions?***