CSE 124
Overlay networks, VPNs, and IP security

March 1, 2016, UCSD
Prof. George Porter
Outline

1. DropBox-like “TritonTransfer”
2. Overlay networks
3. VPNs
4. IP network security
Part 1:
DropBox & TritonTransfer

* Some material courtesy of https://tech.dropbox.com/2014/07/streaming-file-synchronization/
Blocks and files

catpicture.jpg (52 KB)

b1 (16KB)  b2 (16KB)  b3 (16KB)  b4 (4KB)

SHA-256  SHA-256  SHA-256  SHA-256
 h1    h2    h3    h4
TritonTransfer RPC Example

Client

uploadFile("catpicture.jpg", [h1,h2,h3,h4])

[h1,h2,h3,h4]

uploadBlock(h1, b1)

OK

uploadBlock(h2, b2)

OK

uploadBlock(h3, b3)

OK

uploadBlock(h4, b4)

OK

uploadFile("catpicture.jpg", [h1,h2,h3,h4])

Server

[]
TritonTransfer RPC Example

Client

downloadFile("catpicture.jpg")

Server

[h1,h2,h3,h4]
downloadBlock(h1)
b1
downloadBlock(h2)
b2
downloadBlock(h3)
b3
downloadBlock(h4)
b4
downloadBlock(h5)

(file is now downloaded)

(Hash h5 not on server)

ERROR

Client

Server

www.websequencediagrams.com
TritonTransfer RPC Example

Client1

uploadFile("STARWARS.MOV", [h1,h2,...,h130])

[h1,h2,...,h130]

uploadBlock(h1)

OK

uploadBlock(h2)

OK

...

uploadBlock(h130)

OK

uploadFile("STARWARS.MOV", [h1,h2,...,h130])

[]

Server

Client2

uploadFile("STARWARS.MOV", [h1,h2,...,h130])

[]

Client1

Server

Client2

www.websequence diagrams.com
Optimizing downloads/uploads

• Common blocks only have to be calculated for files in the download directory
• Can save that state in secondary files if you want
  – E.g., myfile.dat → .myfile.dat
• Don’t need to keep state persistent across invocations of the server
  – But you can if you want
Part 2
Overlay networks
Abstractions and Overlays

Overlay Network

Physical Network
Overlays for routing: Why?

- Triangle inequality doesn’t hold in networks!
Forwarding traffic through tunneling
Tunneling example

What does R1’s routing table look like? R2’s?
Overlay Networks for routing

- Underlying network
  - Internet connectivity (IP Routing)
• Potential overlay connectivity
  – SF as root
• Determine edge weights
  – E.g., bandwidth, latency
Overlay Networks

• Build overlay connectivity
  – An application-layer distribution tree
Overlay Networks

• We have had overlay networks for at least the past ~2 decades
  – Mbone, 6bone, etc.

• Orig. idea: these would be experimental networks that would help with the transition to “production” networks

• Today, overlay networks are being explored as general-purpose networks
  – Driven by content distribution networks and P2P computing
Challenges to Building Overlay Networks

• What are some of the challenges to building overlays?
  – No central point of control
  – Scalability
  – Network performance tools
  – Building application-level peering that matches the topology of the underlying network

• Familiar story, but different level of abstraction
  – Can account for application-specific information rather than limited information available at network layer
  – Layer 7 versus layer 3 solution
Overlay networks for fault tolerance
Overlay networks for performance

Figure 1: Comparison of RTT for pairs of PlanetLab hosts whose point-to-point latencies were larger than 400 ms (high-latency paths). For the “excluding top n%” graphs, we removed the top n% of one-hop alternatives for each high-latency path from consideration, then used the best remaining one-hop.

Sontag et al., CoNEXT 2009
Part 3
VPNs and IP Security
Virtual private networks

• Can we get the benefits of separate networks while maintaining the benefits of sharing a common network infrastructure?

• What are the benefits of separate networks?
  – Security?
  – Performance?
  – Addressing?
  – ?
Virtual circuits

(a) Corporation X private network

(b) Corporation Y private network

Physical links

Virtual circuits
Making tunnels secure

• Option 1: Application layer
• Option 2: Transport layer
• Option 3: Network layer
• Option 4: Link layer
• Option 5: Physical layer
Problem

• Even with confidentiality there still remain threats for the website customer.
  – An adversary who can’t read the contents of your encrypted message might still be able to change a few bits in it, resulting in a valid order for, say, a completely different item or perhaps 1000 units of the item.
  – There are techniques to detect, if not prevent, such tampering.
  – A protocol that detects such message tampering provides data integrity.
  – The adversary could alternatively transmit an extra copy of your message in a replay attack.
Cryptographic Building Blocks

Symmetric-key encryption and decryption
• Principles of Ciphers
  – Most ciphers are block ciphers: they are defined to take as input a plaintext block of a certain fixed size, typically 64 to 128 bits.
  – Using a block cipher to encrypt each block independently—known as *electronic codebook (ECB) mode encryption*—has the weakness that a given plaintext block value will always result in the same ciphertext block.
  – Hence recurring block values in the plaintext are recognizable as such in the ciphertext, making it much easier for a cryptanalyst to break the cipher.
• Block Ciphers
  – A common mode of operation is cipher block chaining (CBC), in which each plaintext block is XORed with the previous block’s ciphertext before being encrypted.

• The result is that each block’s ciphertext depends in part on the preceding blocks, i.e. on its context. Since the first plaintext block has no preceding block, it is XORed with a random number.
  – That random number, called an initialization vector (IV), is included with the series of ciphertext blocks so that the first ciphertext block can be decrypted.
Cipher block chaining (CBC)

Plaintext block 0
Plaintext block 1
Plaintext block 2
Plaintext block 3

Encryption function

Initialization vector
(For block 0 only)

Blocks of ciphertext
Public-key encryption
Authentication using public keys

1. Message is encrypted using the private key.
2. The encrypted message is then decrypted using the public key.
3. The original message is recovered.
Cryptographic Building Blocks

– There are several common cryptographic hash algorithms, including MD5 (for Message Digest 5) and Secure Hash Algorithm 1 (SHA-1). MD5 outputs a 128-bit digest, and SHA-1 outputs a 160-bit digest.

– A digest encrypted with a public key algorithm but using the private key is called a digital signature because it provides nonrepudiation like a written signature.
Transport Layer Security (TLS)

Handshake protocol to establish TLS session

- List-of-algorithm-combinations, $N_C$
- Chosen-algorithm-combination, $N_S$
- Server authentication messages (depend on which algorithms are chosen)
- Client authentication messages (depend on which algorithms are chosen)
- HMAC of master secret and previous messages as seen by Client
- HMAC of master secret and previous messages as seen by Server
IPSec

• Support for IPsec, as the architecture is called, is optional in IPv4 but mandatory in IPv6.
• IPsec is really a framework (as opposed to a single protocol or system) for providing all the security services discussed throughout this chapter.
• IPsec provides three degrees of freedom.
  – First, it is highly modular, allowing users (or more likely, system administrators) to select from a variety of cryptographic algorithms and specialized security protocols.
  – Second, IPsec allows users to select from a large menu of security properties, including access control, integrity, authentication, originality, and confidentiality.
  – Third, IPsec can be used to protect “narrow” streams (e.g., packets belonging to a particular TCP connection being sent between a pair of hosts) or “wide” streams (e.g., all packets flowing between a pair of routers).
Transport vs. tunnel mode

• Transport:
  – Host-to-host secure connection
  – Encrypted, authenticated, or both

• Tunnel
  – Host-to-network or network-to-network
  – Entire IP packet tunneled in secure IPSec “envelope” to recovered at destination
Security in IPSec

• AH: Authentication header
  – Access control, message integrity, authentication, and antireplay protection
• ESP: Encapsulating Security Payload
  – Like AH, but with encryption too
• SA: Security association
  – Selection of algorithms, crypto, hashes, etc
• SPI: Security Parameters Index (SPI)
  – Per-connection index into SA database
• ISAKMP: Internet Security Association and Key Management Protocol
http://www.unixwiz.net/techtips/iguide-ipsec.html
IP “next” protocols

<table>
<thead>
<tr>
<th>Protocol code</th>
<th>Protocol Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ICMP — Internet Control Message Protocol</td>
</tr>
<tr>
<td>2</td>
<td>IGMP — Internet Group Management Protocol</td>
</tr>
<tr>
<td>4</td>
<td>IP within IP (a kind of encapsulation)</td>
</tr>
<tr>
<td>6</td>
<td>TCP — Transmission Control Protocol</td>
</tr>
<tr>
<td>17</td>
<td>UDP — User Datagram Protocol</td>
</tr>
<tr>
<td>41</td>
<td>IPv6 — next-generation TCP/IP</td>
</tr>
<tr>
<td>47</td>
<td>GRE — Generic Router Encapsulation (used by PPTP)</td>
</tr>
<tr>
<td>50</td>
<td>IPsec: ESP — Encapsulating Security Payload</td>
</tr>
<tr>
<td>51</td>
<td>IPsec: AH — Authentication Header</td>
</tr>
</tbody>
</table>
**IPSec in AH Transport Mode**

### Original IPv4 Datagram

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ver</td>
<td>version</td>
</tr>
<tr>
<td>hlen</td>
<td>header length</td>
</tr>
<tr>
<td>TOS</td>
<td>type of service</td>
</tr>
<tr>
<td>pkt len</td>
<td>packet length</td>
</tr>
<tr>
<td>ID</td>
<td>identification</td>
</tr>
<tr>
<td>flgs</td>
<td>flags</td>
</tr>
<tr>
<td>frag offset</td>
<td>fragment offset</td>
</tr>
<tr>
<td>TTL</td>
<td>time to live</td>
</tr>
<tr>
<td>proto = TCP</td>
<td>protocol number (TCP)</td>
</tr>
<tr>
<td>header checksum</td>
<td>checksum of header</td>
</tr>
</tbody>
</table>

**New IP type**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ver</td>
<td>version</td>
</tr>
<tr>
<td>hlen</td>
<td>header length</td>
</tr>
<tr>
<td>TOS</td>
<td>type of service</td>
</tr>
<tr>
<td>pkt len + AH size</td>
<td>packet length + AH size</td>
</tr>
<tr>
<td>ID</td>
<td>identification</td>
</tr>
<tr>
<td>flgs</td>
<td>flags</td>
</tr>
<tr>
<td>frag offset</td>
<td>fragment offset</td>
</tr>
<tr>
<td>TTL</td>
<td>time to live</td>
</tr>
<tr>
<td>proto = AH</td>
<td>protocol number (AH)</td>
</tr>
<tr>
<td>header checksum</td>
<td>checksum of header</td>
</tr>
</tbody>
</table>

**New IPv4 Datagram**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>src IP address</td>
<td>source IP address</td>
</tr>
<tr>
<td>dst IP address</td>
<td>destination IP address</td>
</tr>
<tr>
<td>TCP header (proto = 6)</td>
<td>TCP header</td>
</tr>
<tr>
<td>next = TCP</td>
<td>next protocol</td>
</tr>
<tr>
<td>AH len</td>
<td>AH length</td>
</tr>
<tr>
<td>Reserved</td>
<td>reserved space</td>
</tr>
<tr>
<td>SPI (Security Parameters Index)</td>
<td>SPI index</td>
</tr>
<tr>
<td>Sequence Number</td>
<td>sequence number</td>
</tr>
<tr>
<td>Authentication Data</td>
<td>authentication data (usually MD5 or SHA-1 hash)</td>
</tr>
<tr>
<td>TCP header (proto = 6)</td>
<td>TCP header</td>
</tr>
</tbody>
</table>

**Protected by AH Auth Data**
IPSec in AH Tunnel Mode

Original IPv4 Datagram

<table>
<thead>
<tr>
<th>ver</th>
<th>hlen</th>
<th>TOS</th>
<th>pkt len</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>flgs</td>
<td>frag offset</td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>proto=TCP</td>
<td>header cksum</td>
<td></td>
</tr>
<tr>
<td>src IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TCP header (proto = 6)

TCP payload

Protected by AH Auth Data

New IPv4 Datagram

<table>
<thead>
<tr>
<th>ver</th>
<th>hlen</th>
<th>TOS</th>
<th>pkt len + AH + IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>flgs</td>
<td>frag offset</td>
<td></td>
</tr>
<tr>
<td>TTL</td>
<td>proto=AH</td>
<td>header cksum</td>
<td></td>
</tr>
<tr>
<td>src IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dst IP address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>next=IP</td>
<td>AH len</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>SPI (Security Parameters Index)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence Number</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication Data (usually MD5 or SHA-1 hash)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TCP header (proto = 6)

TCP payload