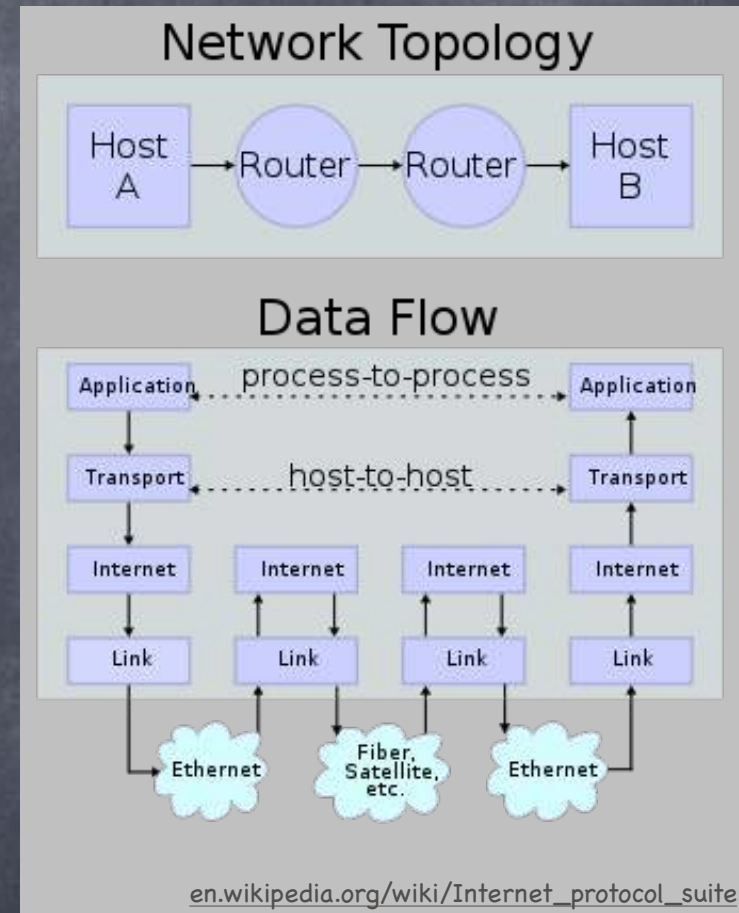


# IPsec, BGPsec, DNSSEC

Lecture 17

# Internet Protocol Suite

- Link layer: ethernet, wifi etc.
- Internet layer: addressing and routing
- Transport layer: Setting up channels (between ports), with traffic control, error-correction etc.
  - TCP/IP for these two
- Application layer: e.g., web, e-mail
  - Also, protocols to help with routing
    - Domain Name Service (DNS): names → IP addresses
    - Border Gateway Protocol (BGP): Routing across "Autonomous Systems" (AS)
    - Various protocols for routing within an AS



# TCP/IP

- Designed (in the 70's) assuming cooperating nodes
  - Focus on speed, scalability, inter-operability. No authentication, no encryption.
- Recall: TLS can implement secure channels even if the lower layers of the network (provided by TCP/IP) are adversarial
  - But if the network is arbitrarily adversarial, TLS cannot prevent Denial of Service
- IPsec — and authenticated versions of DNS, BGP — to make the network less adversarial
  - Importantly, implement authenticated channels. (IPsec also provides the option of encryption.)
  - Like TLS, no anonymisation of traffic



# IPsec

- Four components:

- **Internet Key Exchange (IKE)**: public-key phase to establish symmetric keys for the remaining components.
  - Relies on certificates (from certificate authorities)
  - Uses Diffie-Hellman key-exchange
- **Authentication Header (AH)**: MAC
  - On top of the entire IP packet (including headers)
  - Uses HMAC with SHA2, SHA1 or MD5 as the compression function. (Collision in compression function not known to translate to an attack on HMAC.)
- **Encapsulating Security Payload (ESP)**: SKE
  - AH on top of ESP: Encrypt-then-MAC ✓
- **IP Payload Compression**

# BGP

- All IP addresses are distributed among ~56000 ASes, including large (Tier 1) internet service providers, smaller ISPs, large and small institutions and corporations
- Inter-AS routing based on what they advertise to each other
  - Each AS re-advertises routes that it already learned
- Each AS uses a (business or optimisation) policy to choose a route from many advertised to it
  - A corrupt AS can send bogus routing information to another AS, and make it forward packets to it
    - The corrupt AS may analyse or drop (some of) the traffic sent to it
  - Several examples of incidents, sometimes resulting from misconfiguration, leading to outages

# BGPsec

- An important class of attacks is when an AS advertises that it has an IP range (i.e., IP prefix) within it
  - AS “originates” the IP range
  - Makes it more likely for another AS to use this route to the targeted IP range
    - Even more likely, if it announces route to sub-ranges, as ASes typically favour more specific IP ranges that contain the destination IP
- Route Origin Authorisation (ROA): require a certificate from an authority when claiming to originate an IP range
  - Uses “Resource Public Key Infrastructure,” rooted at “Regional Internet Registries”



# BGPsec

- Using Route Origin Authorisation does not validate the entire path being advertised
- BGPsec requires each step in the path to be authorised, by the destination of that step (except the last step to an IP range, which is certified by an authority, as in ROA)
  - If the authorities (Regional Internet Registries) are trusted (and their keys known), and an honest AS will certify only edges from honest neighbours, then an honest AS will not accept an “invalid” route to an IP originating in another honest AS
- Cannot prevent ASes from advertising legitimate paths and then dropping traffic routed through them
- Or colluding ASes to pretend there is an edge between them

# DNS

- Domain names (an.example.com) need to be translated to IP addresses (32 bit IPv4 address like 93.184.216.34 or 128 bit IPv6 address 9abc:def0:1234:5678:90ab:cdef:0123:4567)
- Solution: Domain Name servers which respond to a domain name query with an IP address
- Most internet activities (web browsing, email, VoIP communication, IOT activity) start with a DNS lookup
- Multiple security concerns: Authenticity, Privacy and Distributed Denial of Service



# DNS Security

- Problem 1: Any one can respond to a DNS query!
  - Causes DoS. Facilitates traffic analysis. And, if no transport layer authentication, serious problem, which will never be detected!
- Problem 2: The content of the DNS queries reveal a lot about user activity
- Easy fix for both: DNS-over-TLS (not common yet)
  - Ensures that the responses are from actual name servers

# DNS Security

- Problem 3: The actual name servers could be corrupt
  - In particular, can respond with wrong information
- Solution: Require name servers to store and return signed records, signed by a zone-owner
  - Called DNSSEC (two versions NSEC and NSEC3)
  - Note: Provides authenticity — but not secrecy — even without TLS
  - Note: Does not provide secrecy against the name server itself, even with TLS

# DNSSEC

- NSEC: store and return signed records, signed by the zone-owner
  - But what if the name server says no record available?
  - Need to verify that!
  - Simple idea: server should return two consecutive entries (in sorted order) and show that they are consecutive
    - Zone-owner signs not just individual records, but also pairs of adjacent records
- New concern: Zone enumeration
  - Information gathering is a typical first step in an attack
  - Individual DNS records are not meant to be secret. But, we do not want DNS to help an adversary recover all domain names in a zone from an honest name server.



# DNSSEC

- NSEC3: Tries to prevent zone enumeration using a simple variation on NSEC
  - Signed record pairs use  $H(\text{domain-name})$ , instead of domain name, where  $H$  is meant to be a random oracle
  - Default hash function used is SHA1
- Still allows enumerating  $H(\text{domain-name})$
- Then, can use an offline attack for zone-enumeration (as domain names are structured, and may be guessed)
- Question: An efficient way to prove that an entry is missing, without revealing anything else?
  - NSEC5: A proposal using “Verifiable Random Functions” (VRF)

Still in the current standard, from 2013, though SHA1 considered weak since 2005

# VRF

- VRF is a PRF, with an additional public-key
  - Remains pseudorandom even given public-key (for honest keys)
  - SK allows one to give a Zero-Knowledge proof that  $F_{SK}(x) = y$ , without revealing SK. Proof can be verified using a PK.
- NSEC5 proposes using a Random Oracle based VRF (assuming DDH)
  - $(SK, PK) = (y, Y=g^y)$  and  $F_y(Q) = H'(C^y)$ , where  $C=H(Q)$
  - By DDH,  $C^y$  pseudorandom group element given  $g, Y, C$  ( $Y$  and  $C$  random group elements).  $H'$  extracts a pseudorandom bit string.
  - Proof includes  $D=C^y$  and a **ZK proof of equality of discrete logs** for  $(g, Y)$  and  $(C, D)$ . i.e.,  $\exists y$  s.t.  $g^y = Y$  and  $C^y = D$ 
    - HVZK, made non-interactive using the Fiat-Shamir heuristic
    - $(C, D)$  can also be simulated as  $(g^r, Y^r)$  since  $H$  random oracle

# Honest-Verifier ZK Proofs

- ZK PoK to prove **equality of discrete logs** for  $((g,Y),(C,D))$ ,  
i.e.,  $\exists r \ Y = g^r$  and  $D = C^r$  [Chaum-Pederson]
  - Can be used to prove equality of two El Gamal encryptions  $(A,B)$  &  $(A',B')$  w.r.t public-key  $(g,Y)$ : set  $(C,D) := (A/A', B/B')$
- $P \rightarrow V$ :  $(U,M) := (g^u, C^u)$ ;  $V \rightarrow P$ :  $v$  ;  $P \rightarrow V$ :  $w := rv + u$  ;  
**V checks**:  $g^w = Y^v U$  and  $C^w = D^v M$
- Proof of Knowledge:
  - $g^w = Y^v U, C^w = D^v M \Rightarrow w = rv + u = r'v + u'$   
where  $U = g^u, M = g^{u'}$  and  $Y = g^r, D = C^{r'}$
  - If after sending  $(U,M)$   $P$  could respond to two different values of  $v$ :  $rv_1 + u = r'v_1 + u'$  and  $rv_2 + u = r'v_2 + u'$ , then  $r = r'$
- HVZK: simulation picks  $w, v$  first and sets  $U = g^w / A^v, M = C^w / D^v$



# VRF and NSEC5

- Using a VRF to protect against zone-enumeration
- Instead of  $H(\text{domain name})$ , use  $F_{SK}(\text{domain name})$ 
  - For a missing entry for a query  $Q$ , return:
    - $Y$ , and a VRF proof that  $F_{SK}(Q) = Y$
    - A pair of consecutive entries  $(Y_1, Y_2)$ , signed by zone-owner, such that  $Y_1 < Y < Y_2$
- Name server needs the VRF key  $SK$  (generated by the zone-owner) to compute  $F_{SK}(Q)$  and the proof. But does not have access to the signing key.
- Adversary querying an honest name server learns the presence/absence of an entry (and an upper bound on the total number of entries)
- Corrupt name server learns all entries, and can also refuse to answer queries, but it cannot give a wrong response

# DNSSEC

- Root Zone Signing Key (ZSK) is currently managed by Verisign
- The corresponding public key is signed by ICANN's Key Signing Key (KSK)
- ZSK renewed frequently (about twice every month), and gets signed in batches once every 3 months, in an elaborate Key Signing Ceremony
  - "Activation data" needed to use KSK in the ceremony is 3-out-of-7 secret-shared
  - KSK backed up encrypted, and the encryption key is 5-out-of-7 secret-shared

# Summary

- IETF Standards for securing the internet
  - TLS for transport layer security
  - Extensions that aim to add security to the original (insecure) protocols used at the internet layer
    - IPsec, BGPsec, DNSSEC
- Also IEEE 802 standards at the link layer: MACsec (MAC meets MAC!), protocols extending IETF's "Extensible Authentication Protocol" (EAP) like WPA2
- Complex standards that focus on efficiency, convenience, backward compatibility (given the millions of devices using older protocols), feasibility of deployment etc.