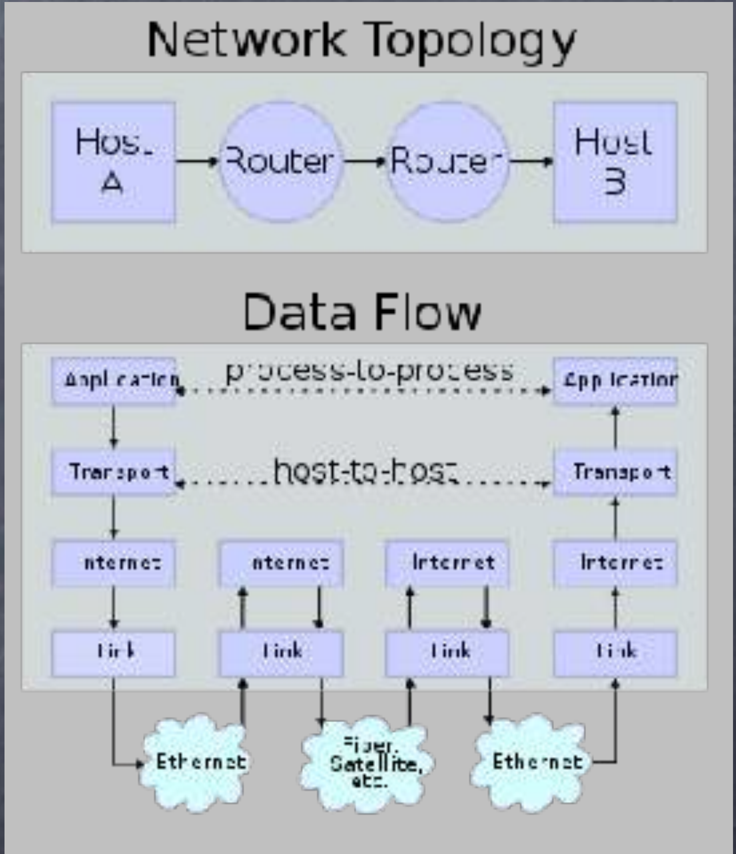
IPsec, BGPsec, DNSSEC

Lecture 18

And a bit of Zero-Knowledge Proofs

Internet Protocol Suite

- TCP/IP: Developed in the 70's
- IP: at the internet layer.
  - Handles addressing and routing
- TCP: at the transport layer.
  - Setting up channels (between ports), with traffic control, error-correction etc.
- Link layer (e.g., ethernet,wifi) and Application layer (e.g., web, e-mail) are too specific for TCP/IP
  - Interfaces: Media Access Controller (MAC) and ports



en.wikipedia.org/wiki/Internet_protocol_suite

Internet Protocol Suite

- Some important protocols at the application layer help IP
- Domain Name Service (DNS)
  - Translating names to IP addresses
- Routing: whom to forward a packet to
  - Two-level Routing
  - Border Gateway Protocol (BGP): Routing across "Autonomous Systems" (AS)
  - Routing within an AS: Various protocols

Internet Protocol Suite

- Originally, TCP/IP designed assuming cooperating nodes
  - Focus on speed, scalability, inter-operability. No authentication, no encryption.
- Transport Layer can implement secure channels even if the lower levels of the network are adversarial (TLS)
  - But if the network is arbitrarily adversarial, cannot prevent Denial of Service
  - Also, secure channels don't hide traffic (source/destination, rate of communication)
- IPsec — and authenticated versions of DNS, BGP — to make the network less adversarial. (But does not try to anonymise traffic.)
  - Importantly, implement authenticated channels. (IPsec also provides the option of encryption.)

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 distributed among ~56000 ASes, including large (Tier 1) internet service providers, smaller ISPs, large and small institutions and corporations
- Routing across "ASes" 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 Authorization (ROA): require a certificate from an authority when originating an IP range
  - Uses "Resource PKI," rooted at "Regional Internet Registries"
  - AS will accept only paths that end in a validated origin

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)
  - If Regional Internet Registries are trusted (and their keys known), then an honest AS will not use an "invalid" route
  - Cannot prevent ASes from advertising legitimate paths and then dropping traffic routed through them
  - Or colluding ASes to pretend there is a direct edge (one-hop path) between them

DNS

- Ddomain 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 with an IP address
- Problem: An adversary can respond to any DNS query!
  - Causes DoS. Facilitates traffic analysis. And, if no transport layer security, serious problem, which will never be detected!
  - Easy fix: DNS-over-TLS (not common yet)
- Additional Problem: Name servers could be corrupt!
- Solution: store and return signed records, signed by the zone-owner. Secure against corrupt name servers. (And, provides authenticity — but not secrecy — even without 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(domain-name), instead of domain name, where H is meant to be a random oracle
  - Default hash function used is SHA1!

    > Still in the current standard, from 2013, though SHA1 considered weak since 2005
- Still allows enumerating H(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?

DNSSEC

- Question: An efficient way to prove that an entry is missing, without revealing anything else?
- A recent proposal: NSEC5
  - Using "Verifiable Random Functions" (VRF)
- VRF is a PRF, with an additional public-key (SK & PK generated honestly)
  - Remains pseduorandom even given public-key
  - SK allows one to give proof that $F_{SK}(x) = y$, without revealing SK. Proof can be verified using a PK.
    - A Zero-Knowledge proof!
  - NSEC5 proposes a Random Oracle based VRF (assuming DDH)

DNSSEC

- Using a VRF to protect against zone-enumeration
- Instead of H(domain name), use F_{SK}(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 only learns the presence/absence of that entry
  - Corrupt name server learns all entries, and can also refuse to answer queries, but it cannot give a wrong response

VRF

- How to build a VRF?
  - Original notion [MRV'99] requires security even if PK is generated by the adversary
  - Constructions from RSA and bilinear pairings, with no random oracles
- NSEC5 based on the discrete log assumption and a random oracle based non-interactive ZK proof
  - (SK,PK) = $(y, Y=g^y)$ and $F_y(Q) = H'(\ C^y\ )$, where C=H(Q)
  - H' ensures pseudorandomness
  - 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 $C^y = D$

Honest-Verifier ZK Proofs

- ZK Proof of knowledge of discrete log of $A=g^r$
  - This can be used to prove knowledge of the message in an El Gamal encryption $(A,B) = (g^r, m Y^r)$
  - $P \rightarrow V$: $U := g^u$; $V \rightarrow P$: v ; $P \rightarrow V$: $w := rv + u$; V checks: $g^w = A^v U$
  - Proof of Knowledge:
    - Firstly, $g^w = A^v U \Rightarrow w = rv+u$, where $U = g^u$
    - If after sending U, P could respond to two different values of v: $w_1 = rv_1 + u$ and $w_2 = rv_2 + u$, then can solve for r
  - HVZK: simulation picks w, v first and sets $U = g^w/A^v$

HVZK and Special Soundness

- HVZK: Simulation for honest (passively corrupt) verifier
  - e.g. in PoK of discrete log, simulator picks (v,w) first and computes U (without knowing u). Relies on verifier to pick v independent of U.
- Special soundness: given (U,v,w) and (U,v′,w′) s.t. v≠v′ and both accepted by verifier, can derive a witness (in stand-alone setting)
  - e.g. solve r from w=rv+u and w′=rv′+u (given v,w,v′,w′)
  - Implies soundness: for each U s.t. prover has significant probability of being able to convince, can extract r from the prover with comparable probability (using "rewinding")

Honest-Verifier ZK Proofs

- ZK PoK to prove equality of discrete logs for ((g,Y),(C,D)), i.e., $Y = g^r$ 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→V: $(U,M) := (g^u,C^u)$; V→P: v ; P→V: $w := rv+u$; V checks: $g^w = Y^vU$ and $C^w = D^vM$
- Proof of Knowledge:
  - $g^w=Y^vU$, $C^w=D^vM$ $\Rightarrow$ $w = rv+u = r'v+u'$ 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 $U=g^w/A^v$, $M=C^w/D^v$

Fiat-Shamir Heuristic

- Limitation: Honest-Verifier ZK does not guarantee ZK when verifier is actively corrupt
  - Can be fixed by implementing the verifier using MPC
    - If verifier is a public-coin protocol -- i.e., only picks random elements publicly -- then MPC only to generate random coins
    - Fiat-Shamir Heuristic: random coins from verifier defined as R(trans), where R is a random oracle and trans is the transcript of the proof so far
      - Also, removes need for interaction!

VRF

- NSEC5 VRF based on the discrete log assumption and a random oracle based non-interactive ZK proof
  - (SK,PK) = (y, Y=g^y) and $F_y(Q)$ = H'(C^y), where C=H(Q)
  - H' ensures pseudorandomness
  - Proof includes D=C^y and a ZK proof of equality of discrete logs for (g,Y) and (C,D)
    - i.e., ∃y s.t. g^y = Y and C^y = D
  - HVZK made non-interactive using the Fiat-Shamir heuristic
  - (C,D) can be simulated as (g^r,Y^r) since H random oracle

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.