Security of
Networks, Services, and Systems

802.11i, IPSec, DNS, BGP

Ricardo Morla

FEUP – SSR/M.EEC, SR/M.EIC

TOC

- 802.11
- IPSec
- DNS
- BGP

Securing 802.11 WiFi

- WEP
  - RC4 stream ciper, 64 or 128 bit key, does not change
- WPA
  - RC4 and TKIP for integrity, new 128 bit key for each packet
- WPA2, 802.11i
  - Counter mode (CCM) with AES
- WPA3
  - AES-256 in Galois counter mode, SHA-384

802.11i phases of operation

1. Discovery
2. Authentication
3. Key management

Then:

- Protected data transfer
- Connection termination

1. Discovery

- AP/STA security capability negotiation:
  - Confidentiality and integrity protocols
  - WEP, TKIP, CCMP (AES)
- Authentication method
  - 802.1X / EAPoL (EAP over LAN)
  - Pre-shared key
- Protocol:
  1. AP security capabilities sent in beacon frame or probe response
  2. STA sends association request with matching security capabilities
  3. AP replies with association response parameters or failure

2. Authentication - 802.1X only

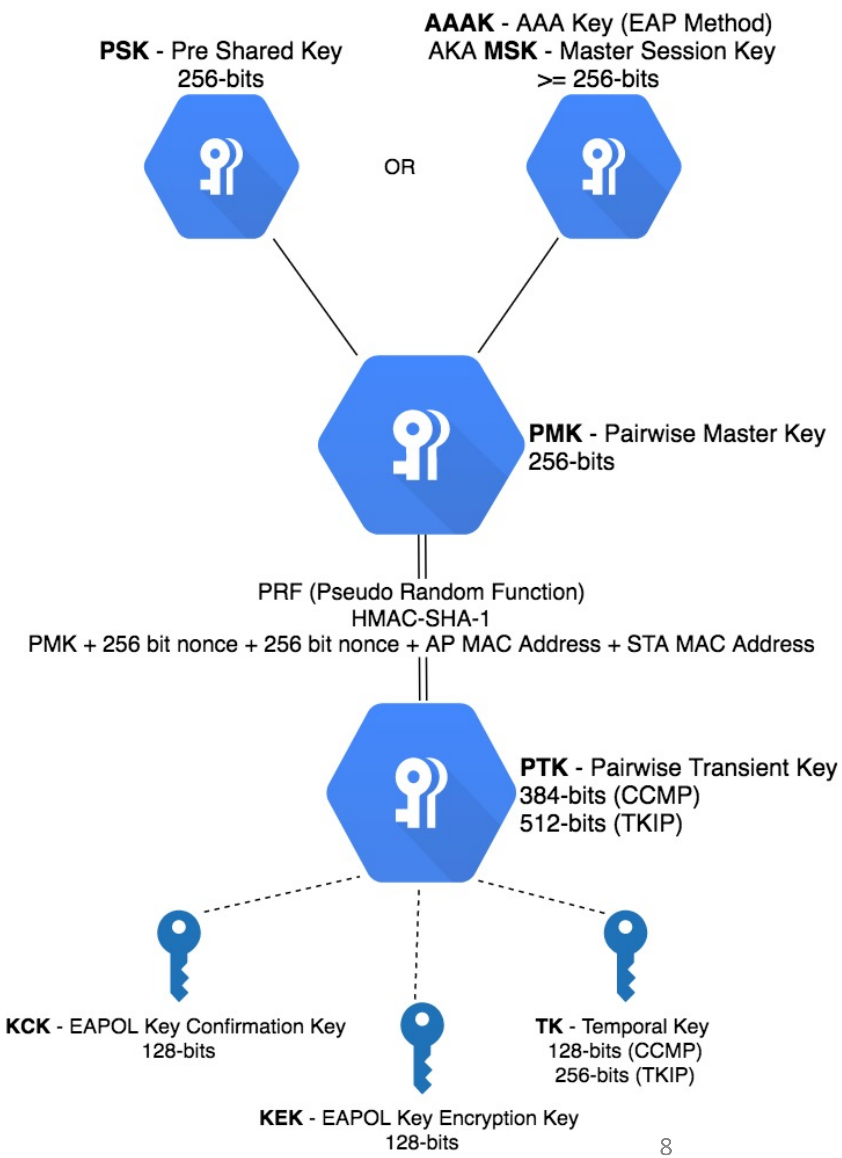
- Block non-authentication-related STA traffic
  - Control port: between supplicant (STA) and AS authentication server (RADIUS)
  - STA can only communicate with AS
- Authentication protocol negotiation (EAPoL):
  - AP sends 802.1X EAP requests identity, STA replies, AP forwards to AS
  - AS probes STA for authentication method, select method
  - STA/AS client authentication using selected authentication method
  - AS sends master session key (MSK) key to bootstrap encrypted communications

2. TLS-based Authentication Methods

- Server certificate used to authenticate server and to provide confidentiality
  - EAP-TLS: authenticate user by certificate
  - EAP-TTLS: establish tunnel, authenticate user by which ever method - could be clear text password, hash, etc
  - PEAP/MS-CHAPv2 : establish tunnel, use MS-CHAPv2 protocol for authentication with user password

3. Key Management

- Key structure
  1. Pre-shared (PSK) or Master Session Key (MSK) from authentication
  2. Pairwise Master Key PMK (equals PSK or truncation of MSK)
  3. Pairwise Transient Key (derived using four-way handshake)
- PTK consists of:
  - Key confirmation key KCK - used to generate MAC codes in 4-way handshake
  - Key encryption key KEK - used to encrypt additional data e.g. group key sent to client
  - Temporal key TK - used to encrypt network traffic
- Multicasting and Group Temporary Key GTK:
  - Sent by AP to all STA, encrypted with each STA's KEK



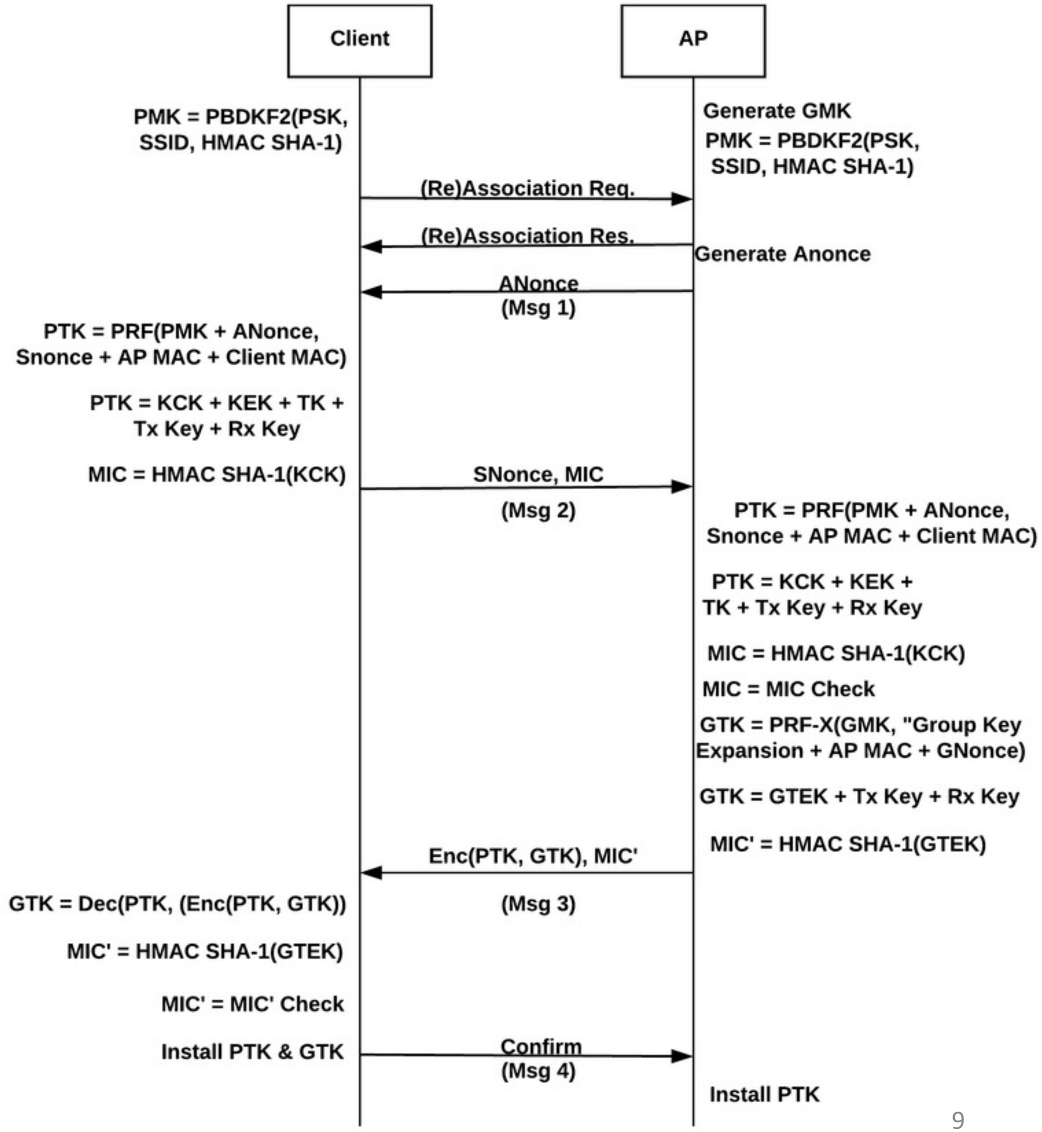
4-way handshake to derive PTK from PMK

1. AP → STA: AP_nonce and AP MAC address
2. STA generates STA_nonce

   * STA computes PTK (KCK,KEC,TK) as HMAC-SHA-1-128 of STA/AP MAC addresses, AP_nonce, and STA_nonce

   * STA → AP: STAnonce and KCK-keyed MAC of STA_nonce for authentication
3. AP can now also reliably compute PTK as HMAC-SHA-1-128 of the same data

   * AP → STA: GTK encrypted with KEK (AES)

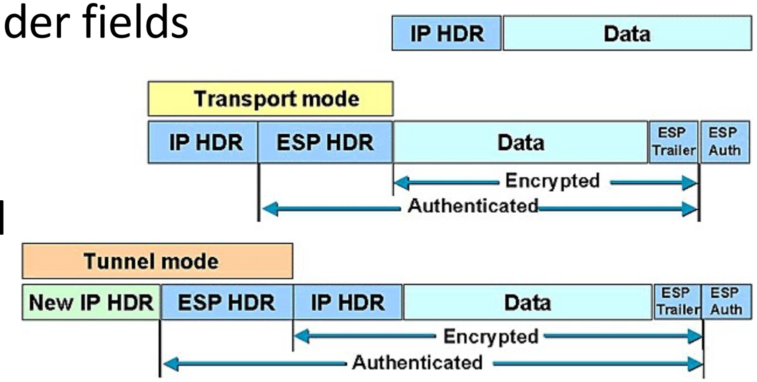


Securing the IP layer

- IP packets are notoriously insecure
  - Spoofing, man-in-the-middle, etc
- Securing the network by securing IP flows
  - IP flow: set of packets with same <Src IP, Dst IP>
  - Bi-directional flows: packets with Src and Dst IP addresses swapped
  - Like TCP flow which is the set of packets with same <Src IP, Dst IP, Src Port, Dst port, TCP protocol>
- How? RFC 4302
  - Cipher algorithm agreement, and key exchange
  - Typically two associations per bi-directional IP flow
  - Encrypt IP header, IP payload, entire IP packet

Modes of operation, Integrity, Confidentiality

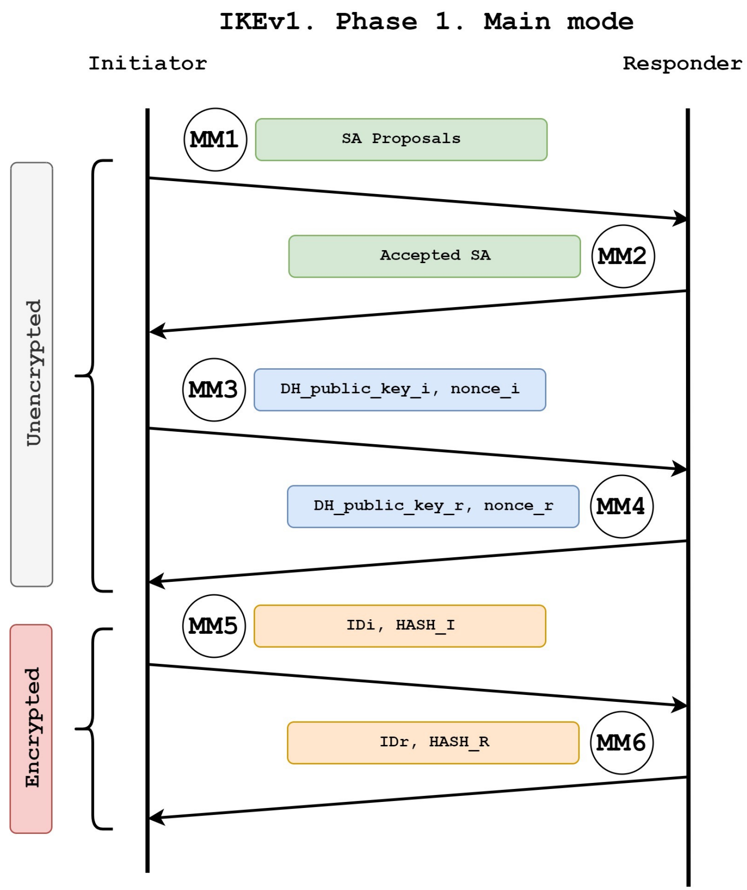
- Modes of operation
  - Transport - encrypts IP payload and integrity check IP header
    - NAT cannot change port number (confidentiality) and IP address (integrity)
  - Tunnel - encapsulate entire IP packet in new IPsec packet
- Authentication Header AH, IP proto #51
  - RFC 4302
  - Provides integrity for the IP packet except mutable header fields
- Encapsulating Security Payload ESP, IP proto #50
  - RFC 4303
  - Provides integrity and confidentiality for the IP payload



End point authentication
IKE Phase 1

1. A sends IKE policy proposals:
   RSA certificates, pre-shared secret, phase 2 cipher and HMAC
2. B sends accepted SA
3. A triggers DH key exchange + A-nonce
4. B replies with DH key exchange + B-nonce
5. A sends its certificate using cipher and HMAC
6. B sends its certificate using cipher and HMAC

After IKE phase 1, the two end points have authenticated each other and can start provisioning IPsec SA’s securely

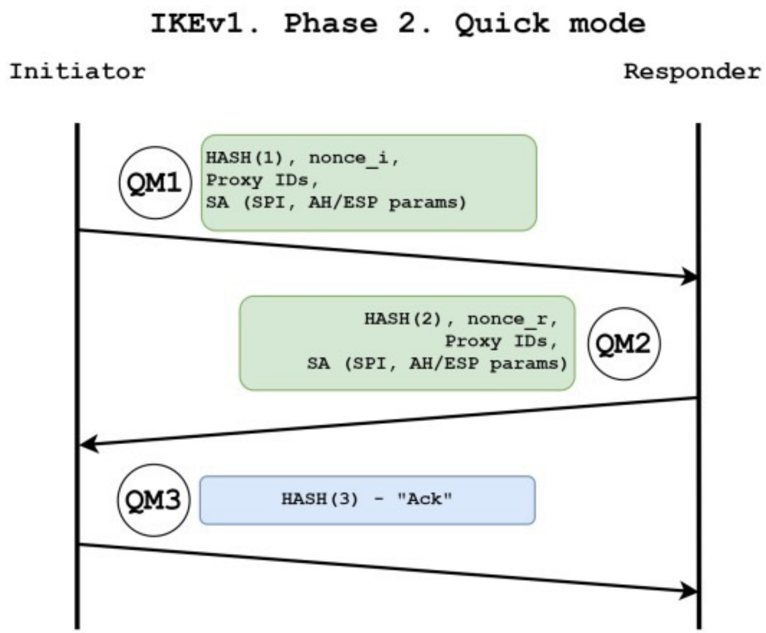


https://datatracker.ietf.org/doc/html/rfc7296

IP-flow security association – IKE Phase 2

1. A sends IPsec SA proposal with A-nonce and IP address in A + A-hash
2. B accepts and sends IPsec SA proposal with B-nonce and IP address in B + B-hash
3. A sends A+B-hash to confirm association

- SA keys are derived from DH shared secret in phase 1.
- Another DH exchange can be done for Perfect Forward Secrecy in SA encryption
- Hashes are over secret key, message id, nonces, and other information



Anti-replay

- Secure sequence numbers allow to prevent replay attacks
    - However, IP is connectionless and unreliable
- Dropping all packets except for N+1 after receiving packet N does not work
- To allow for dropped packets and out of order packets, IPsec uses a (typical) W=64 packet window
- Packets in the window are validated
- Packets to the left of the window are invalid (older)
- Packets to the right of the window are validated, and the window advances

Securing the Domain Name Service

- DNS over *
  - Confidentiality and integrity while in transit between client and resolver
  - Does not provide integrity of DNS responses
  - DNS over HTTPS (DoH)– RFC 8484
  - DNS over TLS (DoT) – RFC 7858
  - DNS over Datagram TLS – RFC 8094
  - DNS over Quic – draft-ietf-dprive-dnsoquic-07
- Securing the name-to-IP mapping
  - Next slide

https://dnsprivacy.org/the_solutions/

Securing the DNS mapping

- DNSSEC – RFC4033 etc
  - Integrity to the DNS query responses, can prevent poisoning
  - Relies on PKI to make the association between response and authoritative name server
  - Extends the set of DNS records to support integrity

https://link.springer.com/chapter/10.1007/978-3-030-75354-2_24

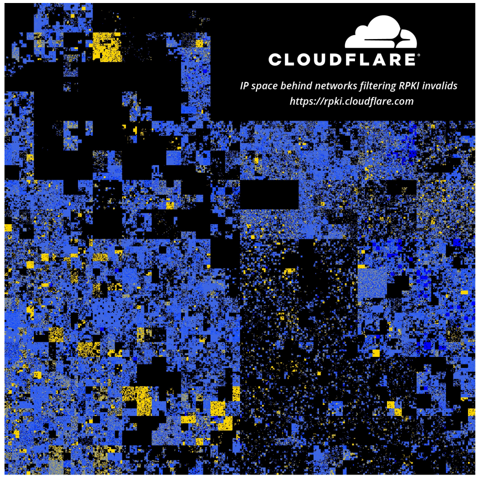
Securing BGP

- Securing the transport of control plane messages between adjacent routers
  - TCP-MD5 to secure the TCP header against TCP resets (weak, MD5, RFC 2385)
    - Reset TCP connection purges all routes learned from that peer router
    - TLS does not prevent TCP Resets
  - Using IPSec to connect the two routers instead of IP, not specific to BGP
  - Other protection
    - https://team-cymru.com/community-services/templates/secure-bgp-template/
- Securing the routes and paths
  - Next slide

https://www.wired.com/images_blogs/threatlevel/files/nist_on_bgp_security.pdf

Securing BGP routes

- Real world protection, filter out:
  - special use addresses, private AS, too long paths, too short prefixes (< /24)
  - customer prefixes in 'big' ASN's you know are not your customers (peerlocking)
- Securing the control plane
  - Origin authorization (RPKI)
  - First-hop authorization (RPKI enhanced)
  - Routing topology path verification (soBGP)
  - Full path verification (S-BGP, BGPsec, psBGP)



Yellow: BGP prefix secure

https://blog.cloudflare.com/is-bgp-safe-yet-rpki-routing-security-initiative/
https://www.sciencedirect.com/science/article/pii/S014036641731068X

Security of
Networks, Services, and Systems

`802.11i, IPSec, DNS, BGP`

Ricardo Morla

FEUP – SSR/M.EEC, SR/M.EIC