1 Crypto Recap

1.1 Objectives:

- **Confidentiality**
- **Integrity**
- **Authenticity**
- **Availability**
- Authorization
- Non-Repudiation, Accountability
- Freshness
- Anonymity, Unlinkability
- Intervenability, Contro
- Transparency

1.2 Confidentiality-Encryption

1.2.1 Symmetric Ciphers

- Secret key for en- and decryption
- Much more efficient
- **Block cipher:** encryps a plaintext block of fixed len e.g.: *Advanced Encryption Standard (AES)*
- **Stream cipher:** encrypts a bitstream e.g.:*ChaCha20*

1.2.2 Asymmetric Ciphers

- Public key for encryption
- Private key for decryption
- Ex.: RSA-based encryption

1.3 Integrity, Authenticity-Signatures, MACs

1.3.1 MACs

- Symmetric cryptography
- Protects data integrity & authenticity
- Ex.: Hash-based MAC

1.3.2 Digital Signatures

- Asymmetric cryptography
  - **Signing with** private key
- Protects data integrity & authenticity
- Provices non-repudiation

1.4 Block Cipher Modes of Operation

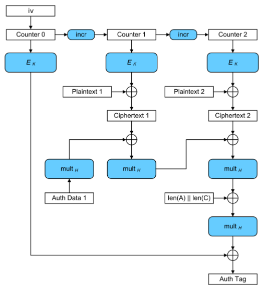
1.4.1 Electronic Code Book (ECB)

- Each plaintext block is encrypted seperatly
- Inherintly insecure! -> Smae block = Same cipher

1.4.2 Cipher Block Chainning (CBC)

- Plaintext is chained to previous ciphertext by XOR and encrypted afterwards
- Difficult to apply securely -> implementations often vulnerable

1.4.3 Galois Counter Mode (GCM)



2 Tranport Layer Security (TLS)

2.1 TLS handshake protocol

- Parameter Negotiation
- Key exchange
- Authentication

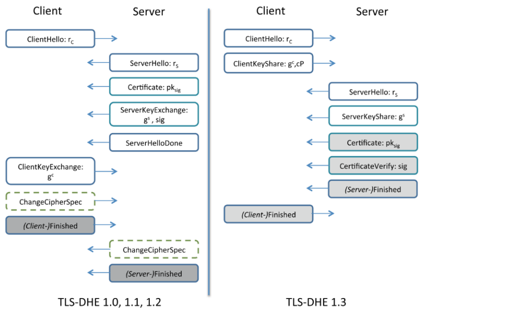
2.2 TLS record protocol

- Protection of integrity, authenticiy and confidentiality
- **Symmetric Cryptography:** e.g., block cipher, usually AES

2.3 Attacks

- Attacks on the record layer
- Attacks on the session key
- Attacks on the private server key
  - Attacks on implmentations: Timing Attacks, Heratbleed, Invalid Point Attacks
- Attacks on TLs eco system
  - Attacks on certificates and the PKI
  - Attacks on the browser GUI

2.4 Overview TLS 1.0-1.2 & TLS 1.3



2.5 HKDF Key Derivation Function

2.6 Forward Secrecy

- TLS 1.3 using certificate-based Authentification: **forward secrecy**
- TLS 1.3 using pre-shared keys (PSK):
  - PSK using *elliptic-curve* Diffie-Hellman: **forward secrecy**
  - PSK without EC-DHE (symmetric-only) and zero-round-trip data: **no forward secrecy**

2.7 Datagramm TLS

- DTLS is identical to TLS where possible
- DTLS has to introcue new mechanisms
  - Explicit sequence numbers
  - Retransmission timer for handshake message
  - Message re-ordering for the handshake
  - Fragmentation of large handshake messages into serveral DTLS records
  - Optional replay detection
  - Invalid records are discarded (silently)
  - Denial-of-Service countermeasures: statless cookies, HelloRetryRequest

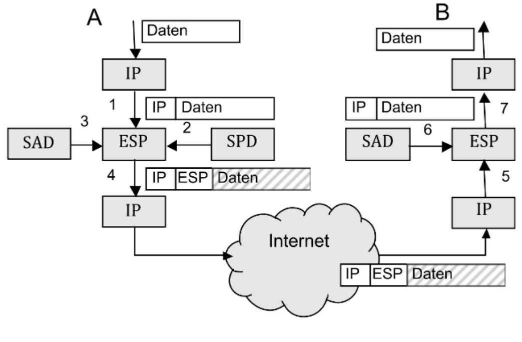
3 Network Layer Security: IPsec

3.1 TLS limitations

- Does not protect all traffic between hosts
  - Protects only a single connection; new connection => new key agreement or session resumption necessary
  - Non-TCP traffic (resp. non-UDP trafic for DTLS) cannto be protected
- Does not protect the TCP layer (RST attacks on TCP are possible which terminate TLS connection)
- Application specific: applications have to be modified to use TLS/DTLS
- Using a TLS-based tunel for VPNs has disadvangates (TCP) (DTLS good option tho)

3.2 Overview IPsec

- Goal: Protect IP packets cryptographically
  - Confidentiality
  - Integrity & Authenticity
- Seperation of packet protection from key exchange
  - Protocols for packet encryption and authentification: symmetric crypto, implemented in the TCP/IP stack
  - Key Exchange completly unrelated: asymmetric (+symmetric) crypto, implemented as network daemon



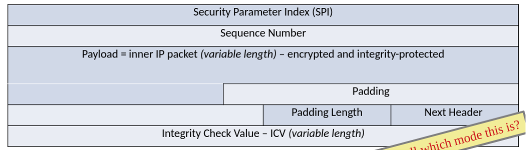
3.3 Packet Formats and Operational Modes

3.3.1 Authentification Header (AH)

- Integrity & Authenticity only
- Partially protects öuterIP header
  - Variable fields cannot be includes
    - Flags, Fragment Offset, TTL, Header Checksume
  - Other fields are included in MAC computation
    - Version, IP header length, total length, identification, Protocol (must be AH), source/dest Address

3.3.2 Encapsulating Security Payload (ESP)

- Confidentiality, Integrity & Authenticity
- Does not protect öuterIP header



- SPI: SA-Identifier
- Next Header: type of payload data (e.g. IPV4)
- Integrity Check Value: data for authentification / integrity protection
  - Message Authentification Code (MAC)
- SAs are negotiated in the key exchange
- Sender side: Packet must be categorized to determine which SA applies (By parameters such as destination=)
- Receiver side: Determines SA from IPsec header of received packet
- SA Parameters:
  - IPsec protocol (AH or ESP)
  - Authentification alrogrithm and key
  - Encryption algorithm and key
  - IPsec mode (transport or tunnel)
  - SA lifetimne
  - Sequence number: current counter
  - On reciver side: SSliding Window"for replay protection

3.4 Scenarios

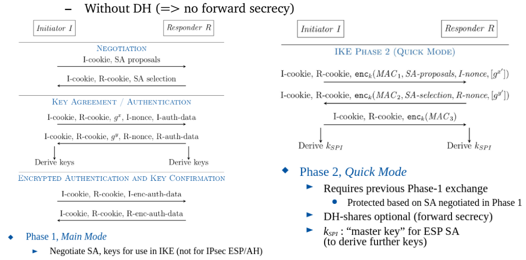
- Host to host
- Host to gateway
- Gateway to gateway

3.5 The Internet Key Exchange (IKE)

- So far, keys are already exchanges, SAs negotiated... but how?
- Authentification of the communication endpoints
- Dynamic negotiation fo algorithms and parameters
- Key establishment
  - Forward Secrecy
- Resistance to DoW attacks
- Efficiency (computation, messages, round trips)

3.6 IKEv1

- Phase 1: main mode vs. aggressive mode
- Authentification modes:
  - Digital Signatures
  - Public key encryption(PKE): two variants (rarely used)
  - Pre-shared keys(PSK)
- Phase 2 (quick mode)
  - With DH (=> forward secrecy)
  - Without DH (=> no forward secrecy)

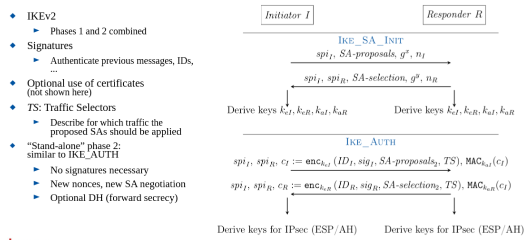


3.6.1 Discussion

- Complicated
  - Information from serveral (complex) RFCs required
  - Many options variants
- Generic
  - Clean seperation of different phases and functionalities
    - Can be advantagous in theory, but leads to inefficincies
  - Intended as general handshake and key agreement protocol, not only for IPsec
    - In practice: only used for IPsec
  - Ineficient
    - Requires too many round trips
  - Inadequate DoS protection
    - IKEv1 uses stateful cookies

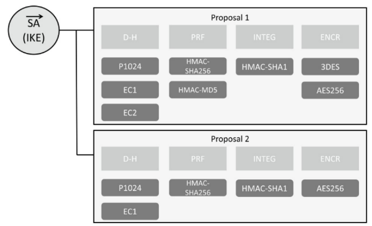
3.7 IKEv2

- Phase 2 (partially) combindes with Phase 1
  - More efficient thatn IKEv1
  - Initial negotiation of IPSec SA included
- Additional SAs can be negotiaded in furhter Phase 2 protocol runs
- Simplified specification
  - Essential information in one RFC
  - No distinction main mode vs. aggressive mode
- DoS protection using statless cookies: optional



3.8 SA Proposals

- Serverl proposals possible
- Proposal contains transforms
- Diffenten options for each transform possible
  - Exactly one transform has to be selected



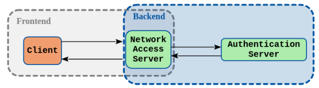
4 Security on Layer 2 (MAC Layer, Ethernet)

4.1 Attacks

- MAC address spoofing
- ARP spoofing
- ARP cache poisoning

4.2 Network Access Control

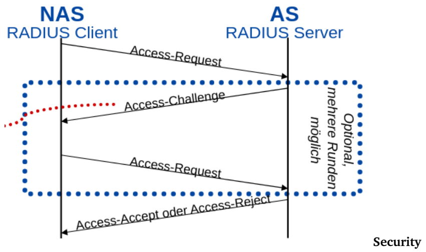
- **Frontend** protocols: Cient -> Network Access Server
  - Point-to-Point connections: PPP
  - LAN: PPPoE, EAPoL
  - WLAN: WPA/WPA2/WPA3, EAPoL
- **Backend** protocols: Network Access Server –> Authentication Server AAA protocols (Authentication, Authorization, Accounting)
  - Radius, Diameter
  - TACACS+



4.2.1 Radius - Remote Authentification Dial-In User Service

- Backend protocol
- Dial-In Server
  - E.g. DSL
  - Company infrastructure

Packet Format



Security

- Old protocol
  - Tries to protect sesitive data only
  - Insecure cryptography
- Uses only over secure protocols
  - Usually TLS
- RADIUS successor: Diameter
  - Fmore flexible, extensible, application-aware
  - SSecurity"features removed, no more insecure crypto of RADIUS => To be used over secure protocols TLS/DTLS/IPsec

4.2.2 Point-to-Point Protocol (PPP)

- **PPP:** Frontend protocol to connect two devices
  - Typically used for dial-up connections (e.g. DSL)
  - Transported over Layer 2 protocol (e.g. ethernet)
- Authentification mechanisms
  - PAP: username/password
  - CHAP: Challange Handshake Authentification Protocol
  - EAP: Extensible Authentification Protocol
- **PPPoE:** PPP over Ethernet (e.g. used for DSL connections)
  - Discovery: Client negotiates session with network access server
  - PPP Session: PPP frames are encapsulated in ethernet frames; potentially several hops

4.2.3 Extensible Authentification Protocol (EAP)

- Extensible authentification framework
  - Fronted protocol for lcient authentification
  - Server send request (e.g. challenge, ident. request, ..)
  - Client responds
  - Server sends success or failure message
  - EAP messages can be transported in RADIUS messages (as RADIUS attribute) for backend communication
- EAPoL: EAP over LAN

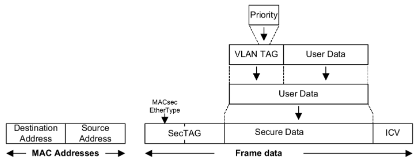
4.2.4 Port Based Network Access Control (PNAC)

- Authentification of clients befre they can use the network (LAN)
  - Client connect to the network
    - ⋆ Authentification traffic is allowed
  - Switch port can only be used for other purposes after successful authentification
  - Re-authentification after timeouts, link-down, etc.
- PNAC uses EAPoL
- Terminology: Port Access Entities (PAE)
  - Client: supplicant
  - Netowrk access server: authenticator
  - Attention: "port"

4.3 MACsec

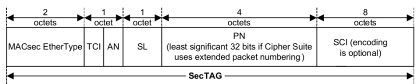
- cryptographically protect the ethernet layer
- Based on MACsec Key Agreeement (MKA)
  - After successful authentification (typically EAP)
  - Derives key from CAK - connectivity association key (pre-shared)
  - Key can be set up by EAP (e.g. EAP-TLS)

4.3.1 Frame



- Data cen be encryped and/or authenticated
- AES-GCM is supported

Security TAG (SecTAG)



- TCI: TAG Control Information
- AN: Association Number
- SL: Short Length (payload length; for short frames only)
- PN: Packet Number
- SCI: Secre Channel Identifier (edentify SA, when CA needs ore than 4)
  - SA is identified by SCI (optional) and AN

4.4 Summary

- Security on Layer 2 can protect all higher Layer
  - But: only in same network
  - Does nt work accross Layer 3 routers
- Network Access Control protects access to the network
  - Client Authentification
  - RADIUS/DIAMETER
  - EAP
  - Port-Based Access Control
- MACsec protects traffic
  - Encryption & authentification

5 Wireless Security

5.1 WiFi Security

5.1.1 Historic Overview

- 1999: WEP (Wired Equivilaent Privacy)
  - Goal: äs secure as a wired LAN"
  - Insecure, various attacks known
- 2003: WPA (WiFi Protected Access)
  - Improved protocols; most known attacks on WEP prevented
  - Enterprice Mode
  - But: Requirement ofr hardware-compatibility with WEP devices => Encryption improved, but still based on obsolete stream cipher
- 2004: WPA2 (still used)
  - Similar to WPA, but AES-based encryption: AES-CCMP
- 2018: WPA3 (supported by new devices)
  - Serveral improvements: Prevention of offline-attacks on pre-shared keys, forward secrecy, encryption for öpen"WLANs

5.1.2 WPA/WPA2 Security

- Personal Mode
  - Pre-Shared Keys
- Enterprise Mode
  - EAP-TLS, PEAP, EAP-TTLS

AES-CCMP

- Authenticated Encryption with Associated Data (AED)
- AES-CCM
  - Authentication: CBC-MAC
  - Encryption: Counter Mode (CTR)
  - MAC and encyprion: computed simultaneously

4-Way Handshake

- Based on Pairwise Master Key (PMK)
  - Personal Mode (WPA-PSK): Computed from passphrase and SSID (as ßalt") using PBKDF2 (password-based key derivation function)
  - Enterprise Mode: Established by key exchange protocol (e.g. EAP-TLS, PEAP)
- 4-Way HS to derive Pairwise Transient Key (PTK)
  - Exchance nonces
  - PTK is derived by hashing PMK, nonces, MAC addrs
  - Furhter key (for differnet purposes) derived from PTK
  - Client gets Group Temporary Key from AP (encrypted)
  - Message Integrity Code(MIC): MACs for integrity protection, key confirmation
- KRACK (2018): Key reinstallation attacks => meanwhile prevented by software/firmware updates
- Problem: Offline attacks against passphrase

5.1.3 WPA 3 Improvements

- Mandatory Protected Managment Frames
  - Prevents deauthentication attacks (DoS)
- Replace PSK Authentication with SAW protocol
  - Simultaneous Authentication among Equals (SAE): "Dragonfly"handshake
  - Prevents offline attacks on passphrase
  - Based on elliptic curve cryptography by default
- Forward Secrecy based on Diffie-Hellman
- 192-bit Security Mode (optional)
  - AES-256 (GCM)
  - SHA-384
  - 284-bit elliptic curves or RSA with at least 3K bits

5.1.4 Simultaneous Authentication among Equals

- SAE "Dragonflyäuthenicates participants and establishes PMK
  - Based on passphrase and (EC-)Diffie-Hellman
  - Can be initiated simultaneously by both parties (useful for mesh networking)
- 4-Way Handshake
  - Esablishes PTK basen on PMK
  - Same as in WPA2
  - But now: PMK with much higher entropy => Offlione attacks not practical
- Hash-to-Group: "Hunting and Pecking"
  - Generate point on elliptic curve from passphrase (and MAC addresses, etc.)
  - Cryptographic has function generates pseudo random numbers (by including a counter in the input)
    - ⋆ Both parties must use the exact same inputs in the same order
  - Fixed procedure to derive x-coordinate
    - ⋆ Check if point on curve can be generated
    - ⋆ If check fails: increase coutner and try again
- Auth-Commit messages
  - Exchange ECDH shares
- Auth-Confirm messages
  - Key confirmation, authentication of messages

5.2 Bluetooth Security

- Authentication: device authentication, no user authentication
- Pairing/bondig: create shared keys; used in connections later on
- Confidentiality: encryption of BT communication
- Message Integrity: MACs (authenticated encryption) to protect BT communication
- Authorization: control access to resources (based on devices, not users)
- Security Modes
  - Mode 1: no security
  - Mode 2: service level (only for backward compatibility)
  - Mode 3: link-level enforces security (only for backward compatibility)
  - Mode 4: authenticated link key using SSecure Connections", based on device pairing
- Eavesdroppting not trivial: Bluetooth uses frequency hoppting (not a security feature)

5.2.1 Device Pairing

- Authentication and generation of link key / long term key
- PIN/Legacy Pairing: enter PIN on both devices
  - Key generation based on PIN, device address, and random values
- Secure Simple Pairing (SSP): since Bluetooth 2.1
  - Numeric Comparison
    - ⋆ Compare 6-digit numbers
  - Passkey Entry
    - ⋆ Read 6-digit form one device, enter on the other one
  - Just Works
    - ⋆ User accepts connection without verification
  - Out of Band (OOB)
    - ⋆ Transmit data using other communication channels (e.g. NFC)

5.2.2 Simple Secure Pairing (SSP)

- Unauthenticated ECDH
- 2-Stage Authentication
  - Stage 2: depends on pariing method
  - Stage 2: Cryptographic authentication based on Stage 1 values and ECDH secret
- Key derivation to generate link key / long term key

5.2.3 Secure Authentication

- Paired (bonded) devices authenticate each other
- Challenge-Response scheme
  - 128-bit random challanges
  - Response: HMAC of BT addresses and challanges (using link key from pairing)
    - ⋆ Before Bluetooth 4.1: based on Bluetooth-specific algorithm E1
- Authenication failure: introcude delay (exponential back-off)

5.2.4 Confidentiality

- Bluetooth-specific stream cipher E0
  - Designed for efficiency
  - Serious attacks hve been published
  - "Practicalïn theory (but complex, hard to apply in practice)
- AES-CCM
  - Used since Bluetooth 4.1
  - Key derived from link key (pairing) and the authentication step

5.2.5 Privacy

- Privacy problem: Devices (users) can be identified by Bluetooth MAC addresses
- Mitigation: BLE private device addresses
  - Resolvable Private Address (RPA) is changed periodically
  - Identity Address remains constant (but is not transmitted over the air)
  - Identity Resolving Key to map RPA to Identity Address
  - Especially imprtant to discoverable devices (which advertice identity info)

5.2.6 5.x Security

- No major changes to security protocols and algorithms
- Bluetooth 5.0
  - PHY improvements, no relevant security changes
- Bluetooth 5.1
  - HCI support for debug keys (should not be relevant in production systems)
- Bluetooth 5.2: adds new features (Extended Attributes, Isochronous Communication, ...)
  - Isochronous communication: connection-oriented or connection-less
    - ⋆ Group communication: group keys need to be established
    - ⋆ Broadcast Authenication
- Bluetooth 5.3: Key Size Negotiation
  - Enables host to define minimun key size

5.2.7 BLUFFS

- BLUFFS: New attacks against bluetooth
  - Breaks Forward Secrecy and Future Secrecy
  - Enables man-in-the-middle attacks, impersonation if one session key compromised
  - Forces weak key: spec allows minimus of 7 Bytes entropy (56 Bits)
    - ⋆ Brute-force attack: offline, parallelizable
    - ⋆ Forces reuse of compromised key
  - Attack against bluetooth spec (BR/EDR: "Bluetooth Classic"versions 4.2 to 5.4): All compliant devices are affected
  - Published and presented at ACM CCS 2023

5.2.8 implementations Vulnerabilities

- BlueBorn(2017): Collection of implementation Vulnerabilities
  - On Windows, IOS, Linux, Android
  - Buffer overflow, integer overflows, ..
- Android (2018): implementation flaws in L2CAP and SMP
  - Remote Memory Disclousure
- BleedingTooth(2020): several bugin in Linux
  - Can even lead to arbitrary code execution in kernal mode
- Windows (2021): BT Driver Elevation of Privilege
- BrakTooth(2021)
  - Bluetooth controllers: SoC firmware Vulnerabilities(Link Manager)
  - Estimation 1400 bluetooth chips/modules affected

5.2.9 Summary

- Complex protocol stack, not easy to implement
- Many attacks in the past
  - on cryptography algorithms
- Bluetooth versions before 2.1 are basically completely insecure
- Bluetooth versions sinde 4.2 are relativly secure (...but: "BLUFFS"!)
  - But the implementations not necessarily!
- Bluetooth 5.2 architecture similar to 4.x
  - Introduces new features and minor security improvements

6 Security Mobile Networks