School of Computer Science and Engineering, UNSW

UNSW AUSTRALIA

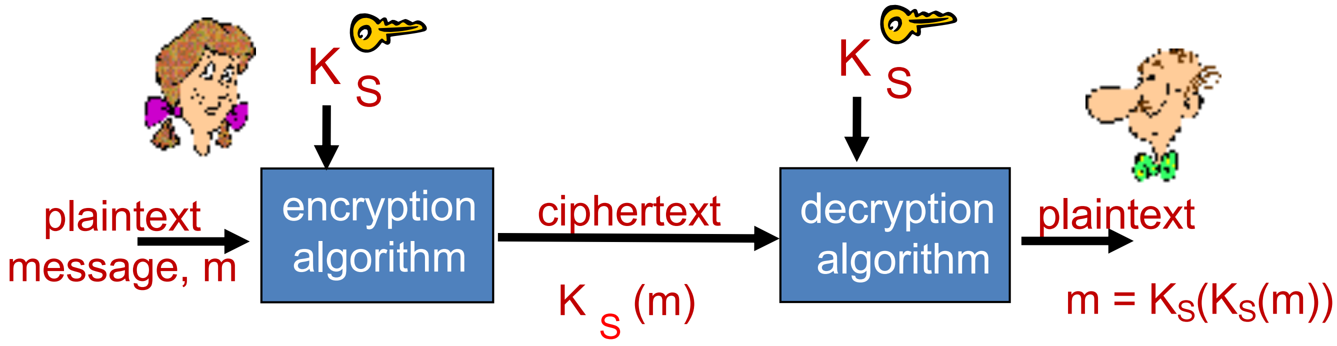
WK02-01: Symmetric Ciphers

Securing Fixed and Wireless Networks, COMP4337/9337

Never Stand Still

Sanjay Jha, Nadeem Ahmed

Symmetric Key Cryptography



Let’s look deeper into symmetric ciphers

Two types of symmetric ciphers

- Block ciphers
  - Break plaintext message in equal-size blocks
  - Encrypt each block as a unit
  - Used in many Internet protocols (PGP-secure email, SSL (secure Transport layer), IPsec (secure Network layer)
- Stream ciphers
  - Encrypt one bit at time
    - In practice we operate on byte boundries
  - Used in secure WLAN

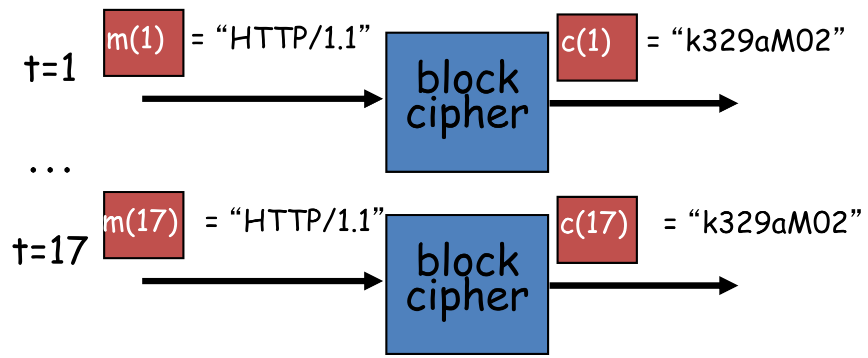
Block Cipher

- Processed as k bit blocks
- 1-to-1 mapping is used to map k-bit block of plaintext to k-bit block of ciphertext
- E.g: k=3 (see table)
  - 010110001111 => 101000111001
- Possible permutations = 8! (40,320)
- To prevent brute force attacks
  - Choose large k (64, 128, etc)
- Full-block ciphers not scalable
  - E.g., for k = 64, a table with 2^{64} entries required
  - instead use function that simulates a randomly permuted table

Input	Output
000	110
111	001
001	111
010	101
011	100
100	011
101	010
110	000

Cipher Block Chaining

- Cipher block: if input block repeated, will produce same cipher text:



- Sender creates a random k-bit number r(i) for ith block and calculates
  - $c(i) = K_S(\ m(i) \oplus r(i)\ )$
- Sends c(1), r(1), c(2), r(2), c(3), r(3)….
  - r(i) sent in clear, but Ks not known to attackers.
  - Need to transmit twice as many bits as before.

CBC Example

- Example: sent text 010010010 if no CBC, sent txt = 101101101
  - 1-to-1 mapping (see table earlier)
- Let's use the following random bits
  - r(001), r2(111), r3(100)
- First, we XOR the plain text with the above random bits:
  - E.g 010 XOR 001 = 011
  - Now do table lookup for 011 -> 100
- Use above technique to generate cipher text c(1) = 100, c(2) = 010, c(3) = 000:
  - All three outputs different even though same plain text 010.

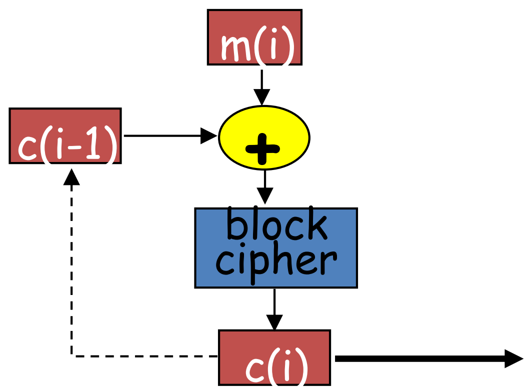
CBC: Sender

- Send only one random value along with the very first message block
- XOR ith input block, m(i), with previous block of cipher text, c(i-1)
  - c(0) is an Initialisation Vector transmitted to receiver in clear
  - First block:

    $c(1) = K_S(m(1) \oplus c(0))$

  - Subsequent blocks:

    $c(i) = K_S(m(i) \oplus c(i\text{-}1))$

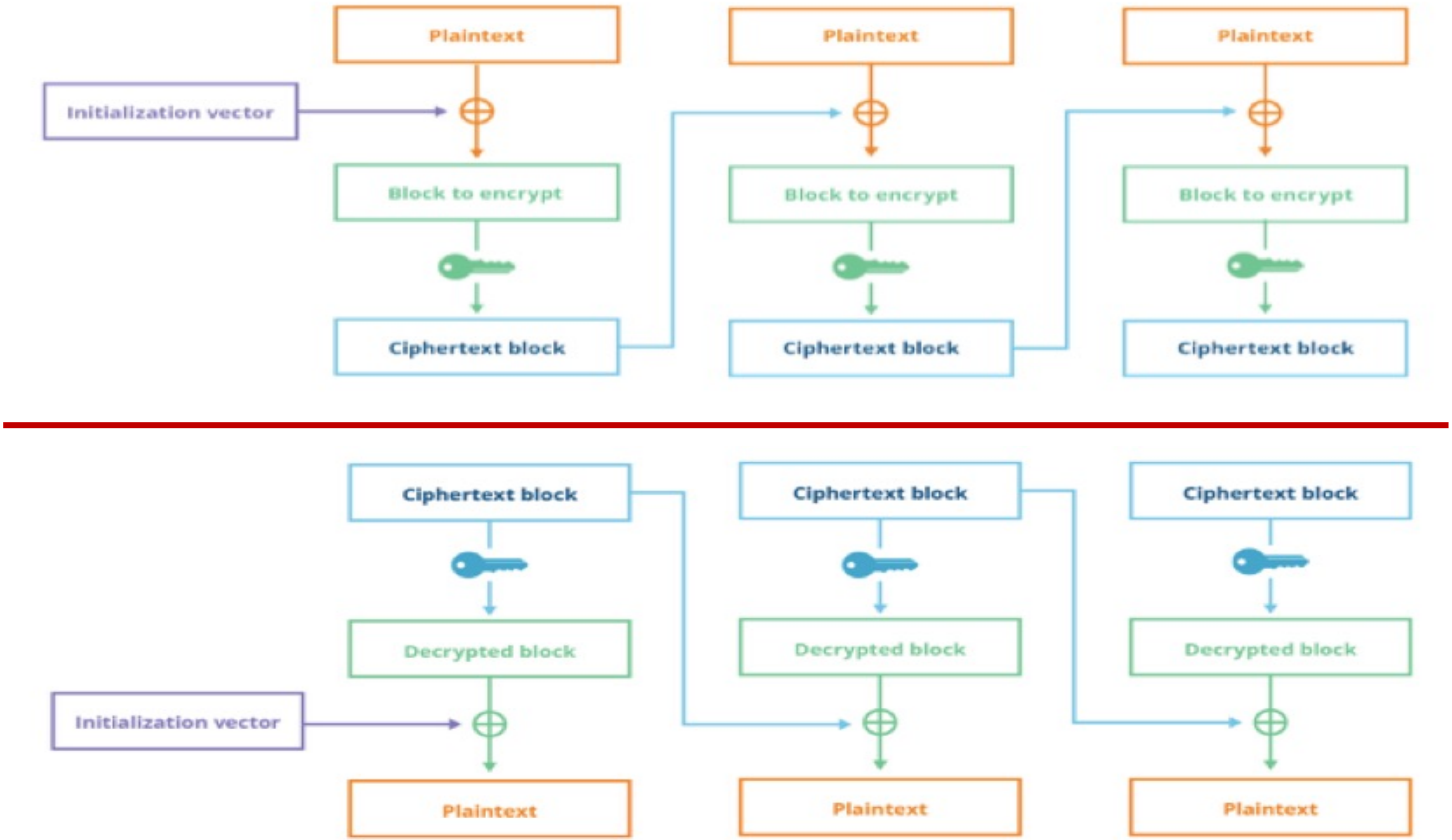


CBC: Receiver

- How to recover m(i)?
  - Decrypt with Ks to get $s(i) = Ks(c(i)) = m(i) \oplus c(i-1)$
  - Now the receiver knows c(i-1), it can get

    $m(i) = s(i) \oplus c(i-1)$
- IV sent only once
  - Intruder can't do much with IV since it does not have Ks
- CBC has important consequence for designing secure network protocols

CBC Operations



Block Ciphers

- Block cipher needs to wait for one block before processing it
  - Often require padding to align the input with the key mapping
- Operates on a single block of plaintext
  - 64 bits for Data Encryption Standard (DES)
  - 128 bits for Advanced Encryption Standard (AES)
- Computationally infeasible to break block cipher by brute-force decryption
  - DES use key size of 56 bits
  - AES supports key sizes of 128,192 and 256 bits

- Pay attention to the input block sizes and the key sizes

DES – NIST 1993

- 56-bit symmetric key, 64-bit plaintext input
- Block cipher with cipher block chaining

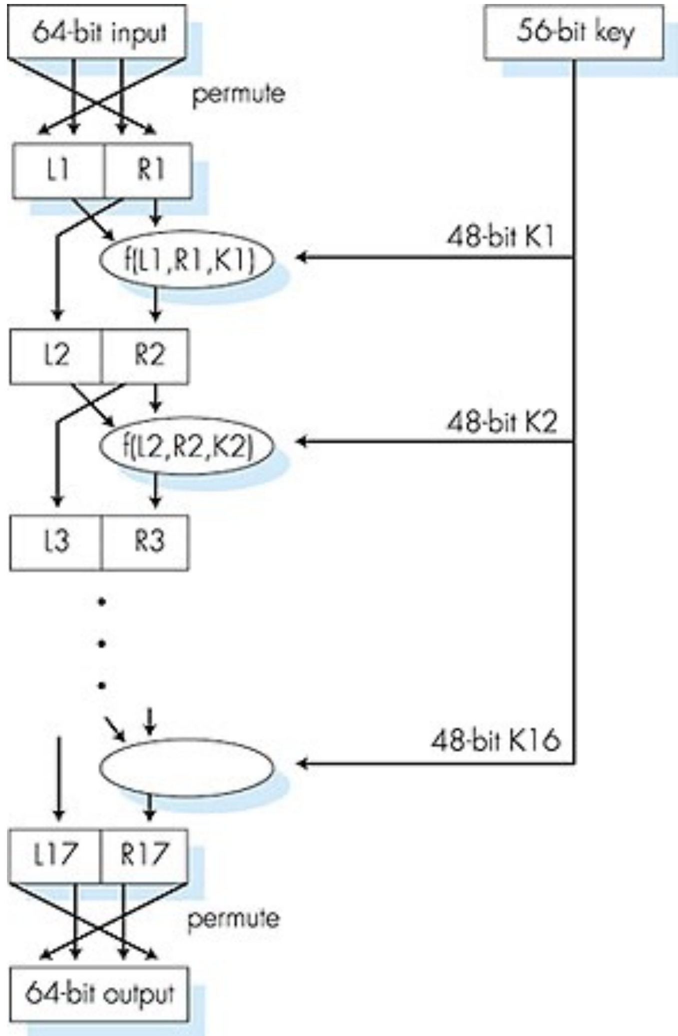
DES operation

initial permutation

16 identical "rounds" of function application, each using different 48 bits of key

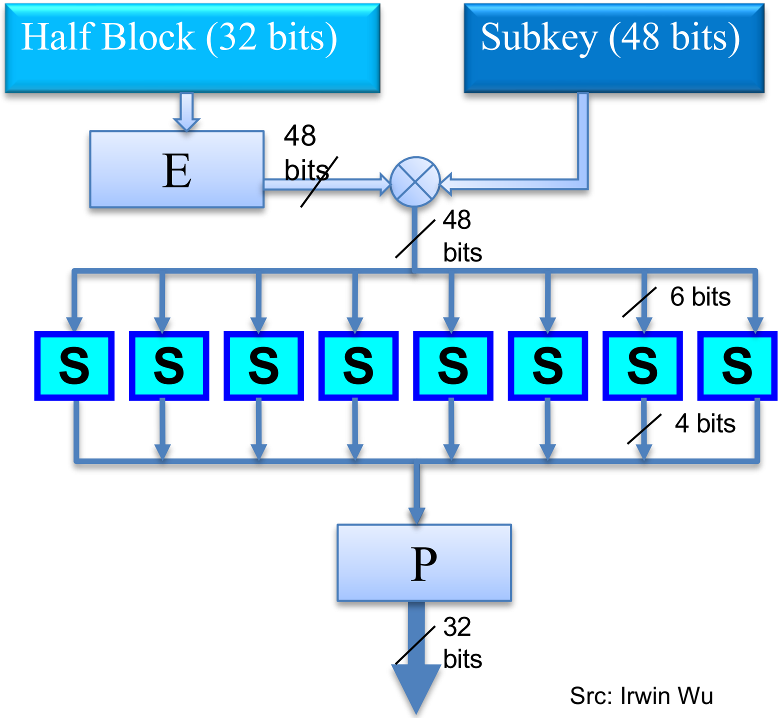
final permutation

No need to memorise this



Feistel (F) function

- Expansion (E): the 32-bit half-block is expanded to 48 bits using the expansion permutation
- Key mixing:
  - Sixteen 48-bit subkeys: one for each round from 56 bit key
  - Derived from the main key using the key schedule
  - E is combined with a subkey using an XOR operation
- S-box: Substitution
  - Transforms input bits using substitution tables to provide diffusion
  - Spread plaintext bits throughout ciphertext
  - Small change in either the key or the plaintext should cause a drastic change in the ciphertext (avalanche effect)
- P: permutation function
  - P yields a 32-bit output from a 32-bit input by permuting the bits of the input



No need to memorise this ☺

Attacks on DES

- Brute Force Attacks
  - DES Challenge I: 1997, 56-bit-key-encrypted phrase decrypted in 96 days
  - DES challenge II-1: Early 1998, decrypted in 39 days
  - DES Challenge II-2: July 1998 , 56 hours
  - DES Challenge III: Jan 1999, 22 hours and 15 minutes
- No known good analytic attack

Symmetric key crypto: 3DES

- 3DES: 1998
  - Triple Data Encryption Algorithm (TDEA)
  - Uses 64-bit input
  - Encrypt-Decrypt-Encrypt with 3 (different) keys
    - 168-bit key size
    - Ciphertext = $E_{K3}(D_{K2}(E_{K1}(\text{plaintext})))$
    - Plaintext = $D_{K1}(E_{K2}(D_{K3}(\text{ciphertext})))$
  - Three times slower than DES

Symmetric key crypto: 3DES

- Sweet32 attack on 3DES (self-read, examinable)

https://sweet32.info/SWEET32_CCS16.pdf

  – A birthday attack on long-lived TDEA sessions
  – Waiting for collision to happen
    - Same ciphertext produced using CBC mode
  – CVE-2016-2183 – a major security vulnerability (https://nvd.nist.gov/vuln/detail/CVE-2016-2183)
  – For 64-bit block size, collision happens after encrypting 2^{32} blocks with the same key

- 3DES disallowed at start of 2024

Bits of Security

- The security of a cryptographic primitive is expressed as “bits of security”
  - n-bit security means that the attacker would have to perform 2^n operations to break the security key.
  - It is different to a security claim
  - Useful for protocol comparison
- For symmetric ciphers
  - The security claim is typically equal to the size of the key
    - Considering a brute-force attack
  - TDEA has a security claim of 168 bits but provides 112 bits of security.
    - Meet-in-the-middle attack (not examinable)

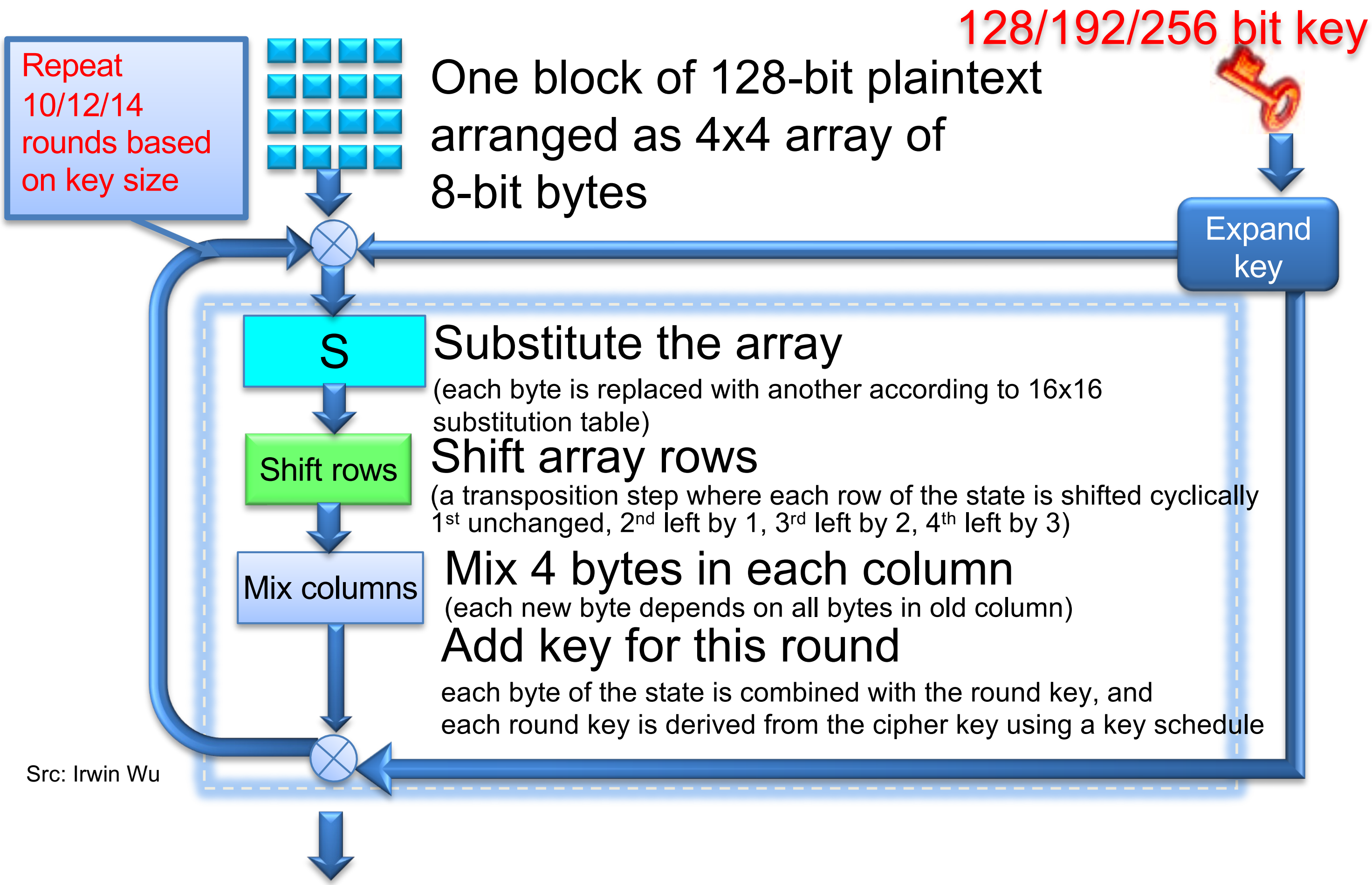
Symmetric Key Block Ciphers

- The security of a block cipher : key size k and block size n
  - Exhaustive search of the key, with complexity 2^k.
    - See the bits of security
  - Block size n controls the amount of data that can be encrypted using the same key. Should be secure up to 2^n.
  - Block ciphers are in fact unsafe with more than $2^{n/2}$ blocks of message (the birthday bound).
  - Security affected by the number of blocks processed with one set of keys.

AES: Advanced Encryption Standard

- Symmetric-key NIST standard, replaced DES (Nov 2001)
- Processes data in 128-bit blocks
  - The birthday bound corresponds to 256 EB
- 128, 192, or 256-bit keys
  - Brute force decryption (try each key) takes 149 trillion years for AES
    - Universe lifetime: 100 billion years

AES -Structure

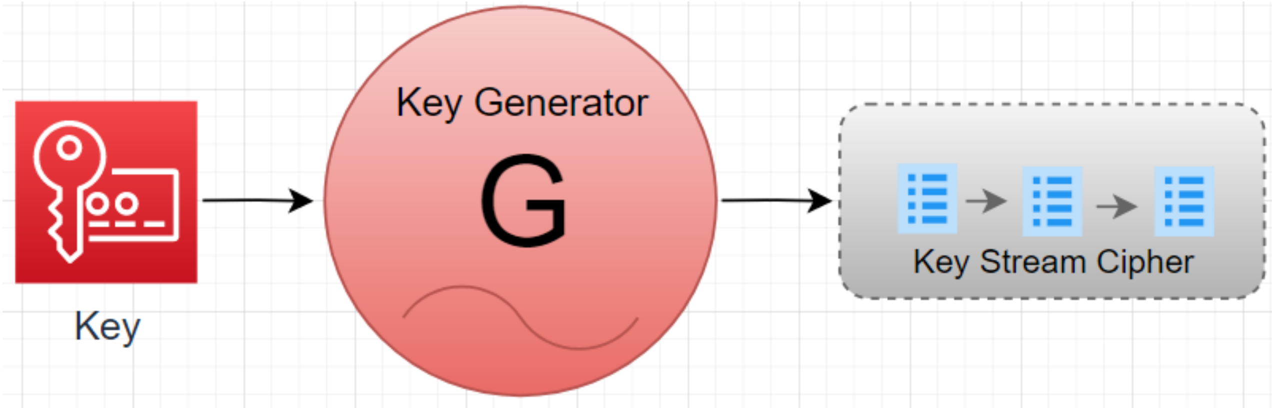


No need to **memorise** this ☺

AES Confidentiality Modes

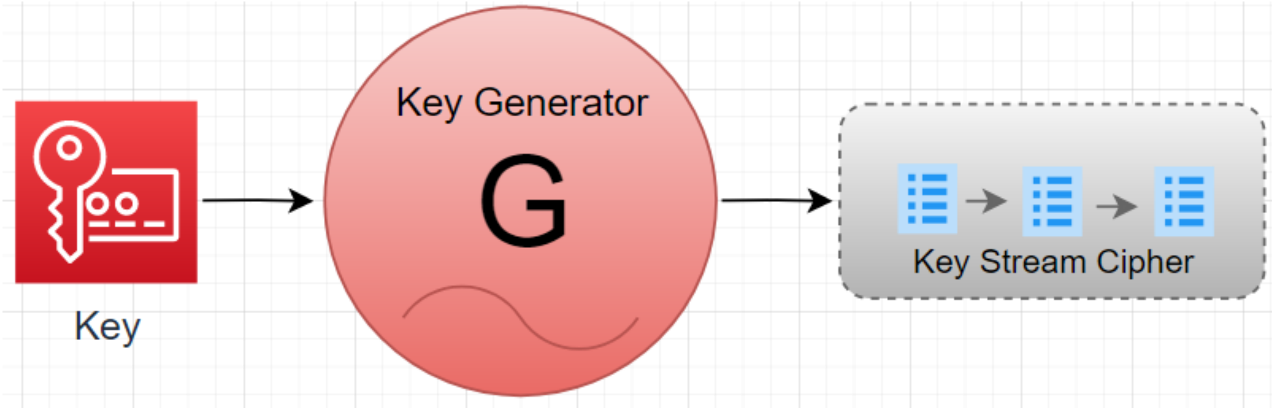
- Five confidentiality modes of operation
- Electronic Codebook (ECB)
  - Split plaintext into blocks, encrypt each one separately using the block cipher
  - Can be done in parallel
  - Message repetitions may show in ciphertext
- Cipher Block Chaining (CBC) mode
  - Split plaintext into blocks, XOR each block with the result of encrypting previous blocks
- Cipher Feedback (CFB), Output Feedback (OFB), and Counter (CTR) modes

Stream Ciphers



- Process message bit by bit (as a stream)
  - Ideal for real-time communication
  - A keystream must not be reused; otherwise, the encrypted messages can be recovered
    - XoR of two ciphertexts created using the same keystream reveals the XoR of the plaintexts

Stream Ciphers



- Expand a short key into a pseudo-random key stream
- Combine each byte of keystream with a byte of plaintext to get ciphertext:
  - m(i) = i th unit of message
  - ks(i) = i th unit of keystream
  - c(i) = i th unit of ciphertext
  - $c(i) = ks(i) \oplus m(i)$ ($\oplus$ = exclusive or)
  - $m(i) = ks(i) \oplus c(i)$

Rivest Cipher 4 – RC4

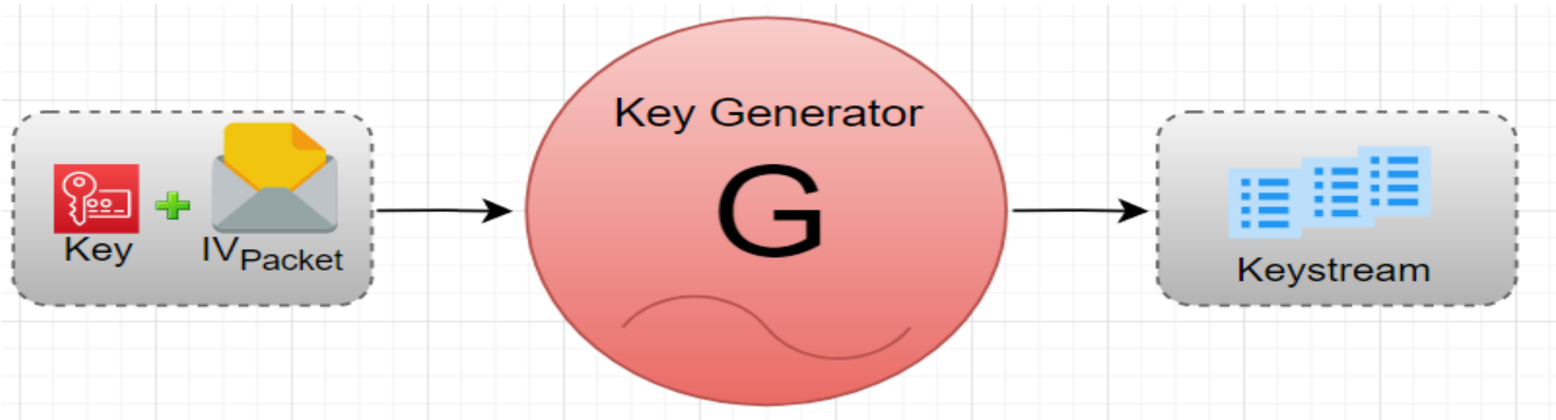
- Rivest Cipher 4: Designed by Ron Rivest
  - A proprietary cipher owned by RSA.com
    - No longer a trade secret
  - Ideal for software implementation, as it requires only byte manipulations
- Variable key size (40 to 2048 bits), byte-oriented stream cipher
- Widely used
  - SSL, Wireless WEP and WPA

Wired Equivalent Privacy – WEP

- How to design a flawed security protocol!
- Provide security equivalent to Wired Network
  - Problem starts with this thinking!
- Symmetric key crypto
  - confidentiality
  - end host authorisation
  - data integrity
- Efficient
  - implementable in hardware or software

Symmetric Cipher and Packet Independence

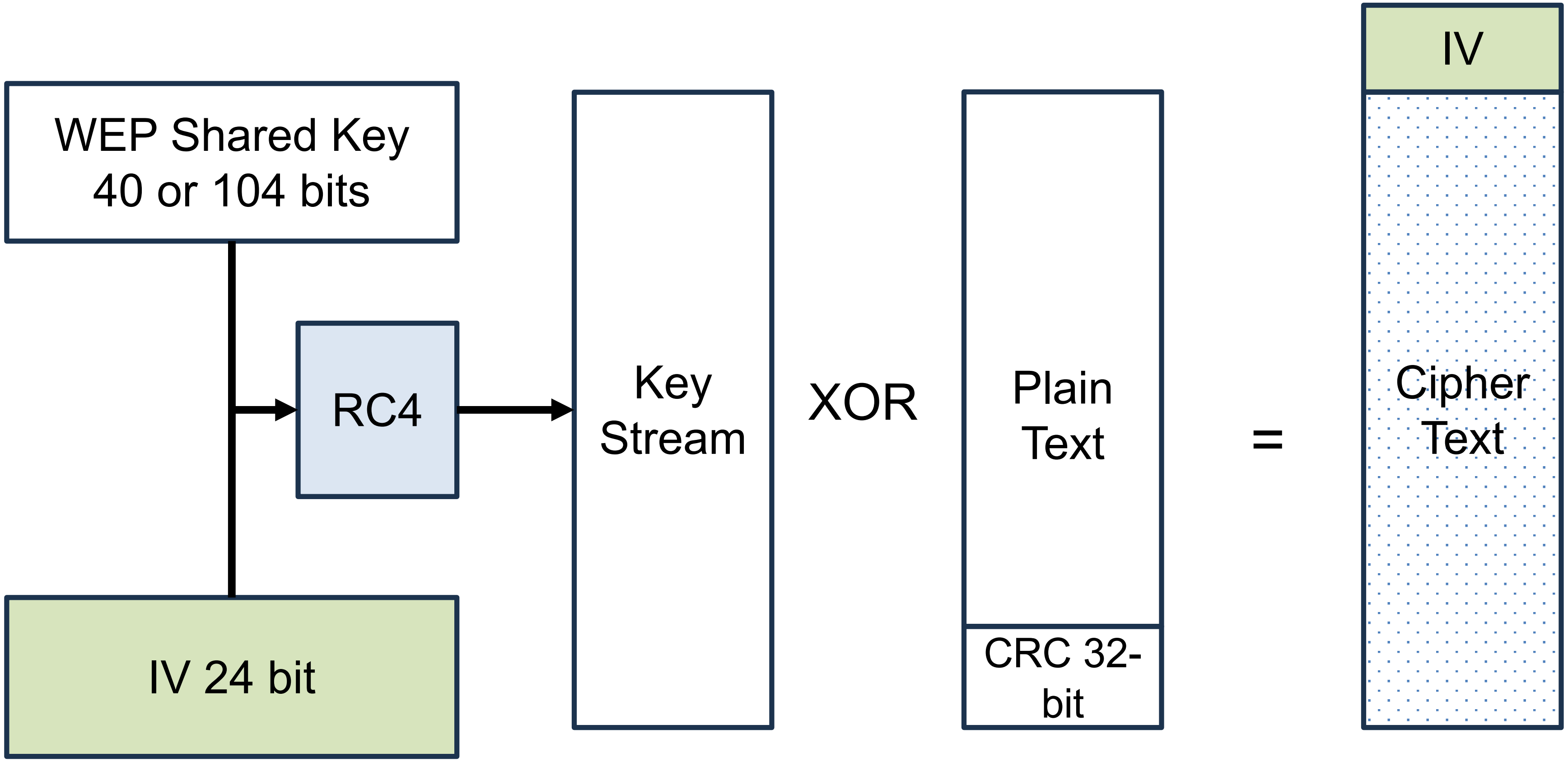
- Design goal: each frame (packet) separately encrypted
  - Ensure that keys are not repeated. i.e., every single frame or packet requires the generation of a new stream.
- WEP approach: initialize keystream with key + new IV for each frame



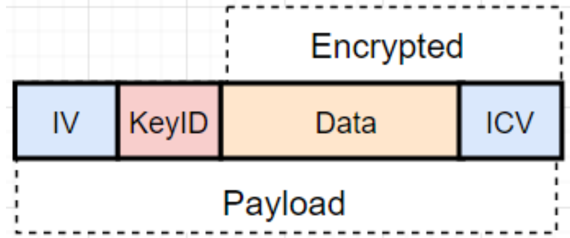
WEP Pre-shared Key

- Set a password on the AP and then enter on all devices
  - WEP PSK is calculated using a key derivation function (KDF) [see PBKDF2].
- Not possible to authenticate individuals
  - hard to distinguish who is using service - needs extra work.
- A key compromise for one user (or a user leaving the organization) means that every device needs to change to a new key
  - Must be distributed to all users securely

WEP Encryption (1)

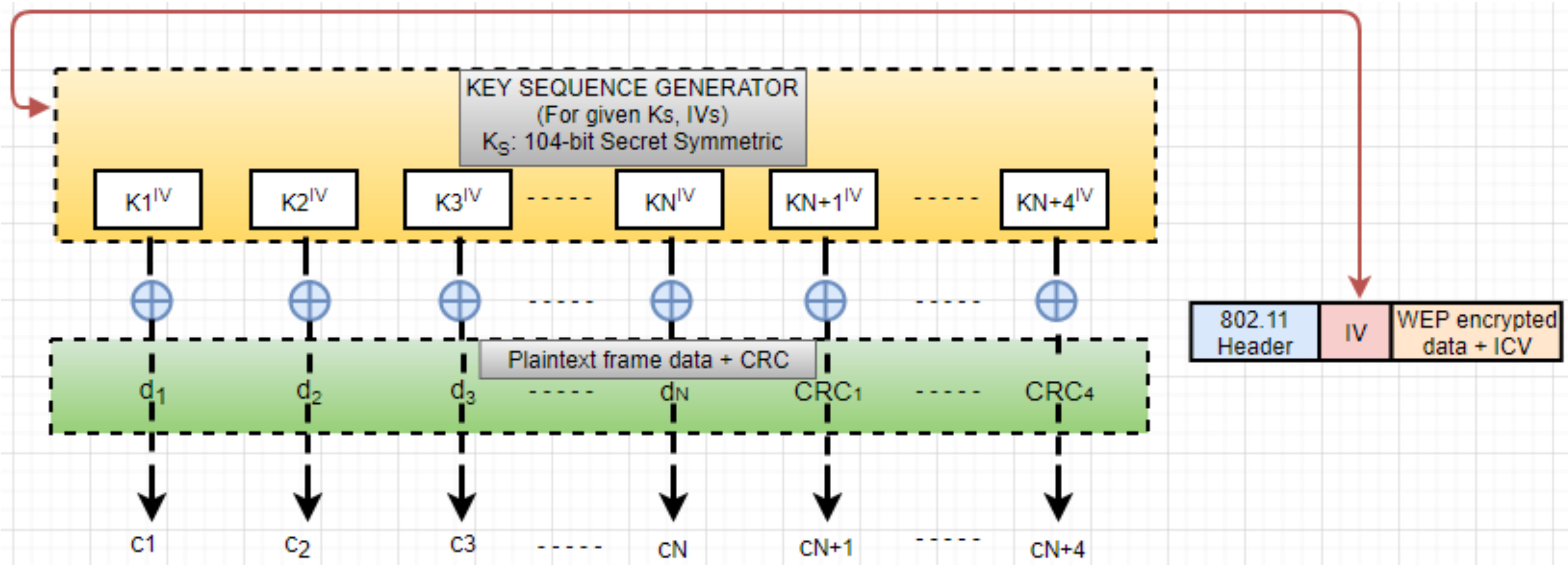


WEP Encryption (2)



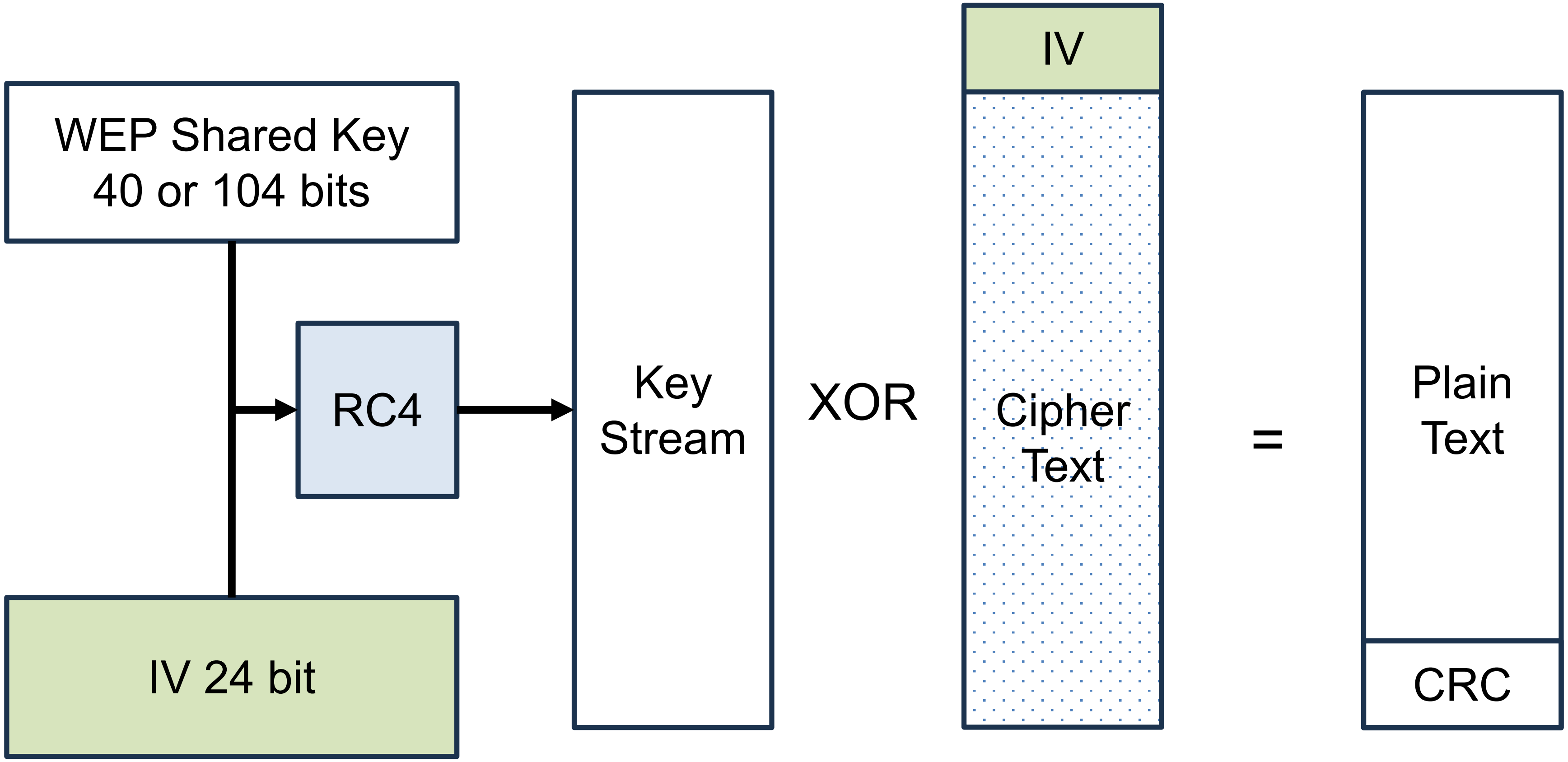
- Sender calculates Integrity Check Value (ICV) over data
  - for data integrity: uses CRC-32, four-bytes
- Each side has 104-bit shared key (can be 40-bit as well)
- Sender creates 24-bit initialization vector (IV), appends to key: gives 128-bit key
- Sender also appends KeyID (in 8-bit field) – Why?
- 128-bit key input into pseudo random number generator (PRNG) e.g. RC4 to get keystream
- Data in frame + ICV is encrypted with RC4:
  - Bytes of keystream are XORed with bytes of data & ICV
  - IV & KeyID are appended to encrypted data to create payload

WEP Encryption (3)

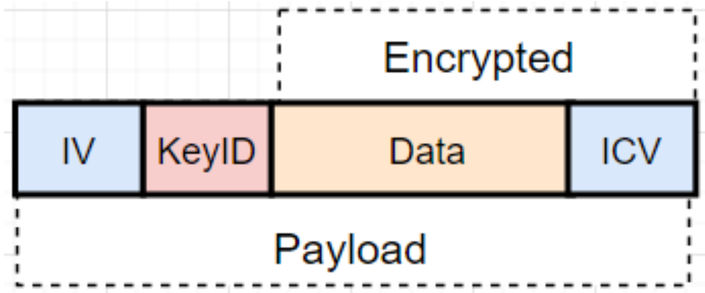


New IV for each frame

WEP Decryption (1)



WEP Decryption (2)



- Receiver extracts IV (received in plaintext)
- Inputs IV, shared secret key into pseudo random generator, gets keystream
- XORs keystream with encrypted data to decrypt data + ICV
- Verifies integrity of data with ICV
  - Note: message integrity approach used here is CRC-32 different from MAC (message authentication code) and signatures (using PKI).

Problems with Linear Checksum

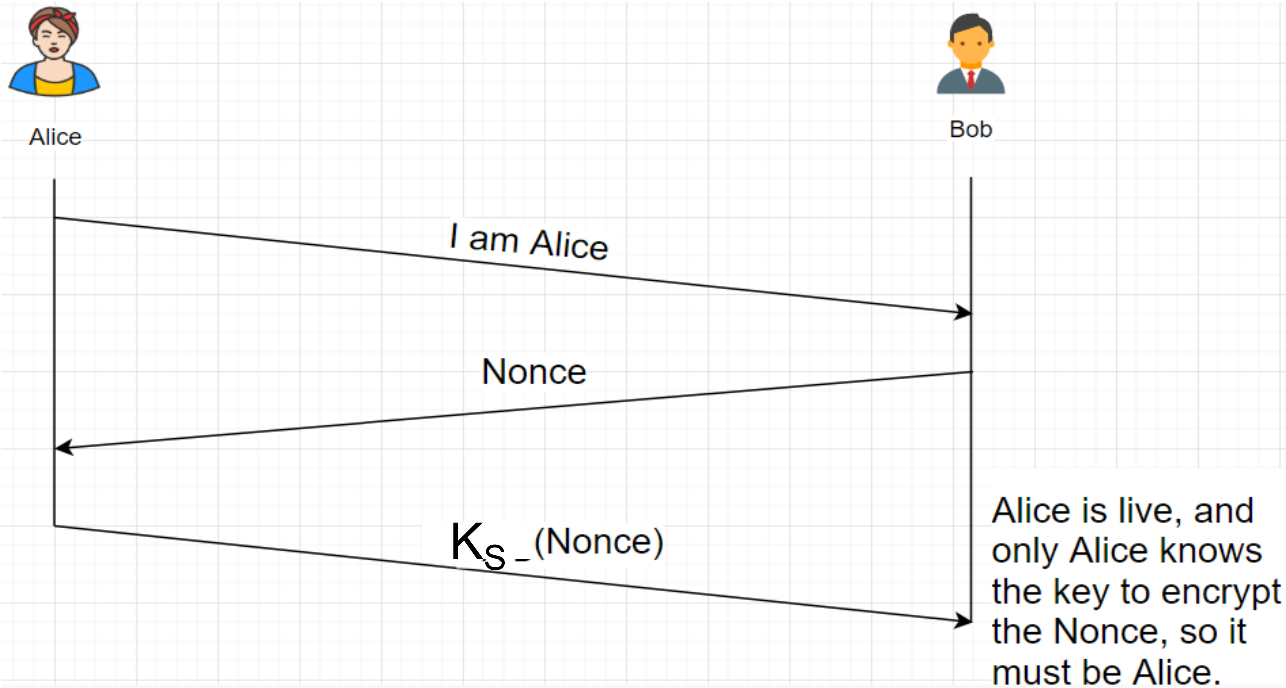
- Encrypted CRC-32 used as Integrity Check Vector (ICV)
  - Fine for random errors, but not malicious ones

- CRC-32 is linear
  - Possible to compute the bit difference of two CRCs based on the bit difference of the two messages
  - Flipping bit n in the message results in a deterministic set of bits in the CRC that must be flipped to produce a correct checksum on the modified message.
  - An attacker can flip arbitrary bits in an encrypted message and correctly adjust the checksum so that the resulting message appears valid.

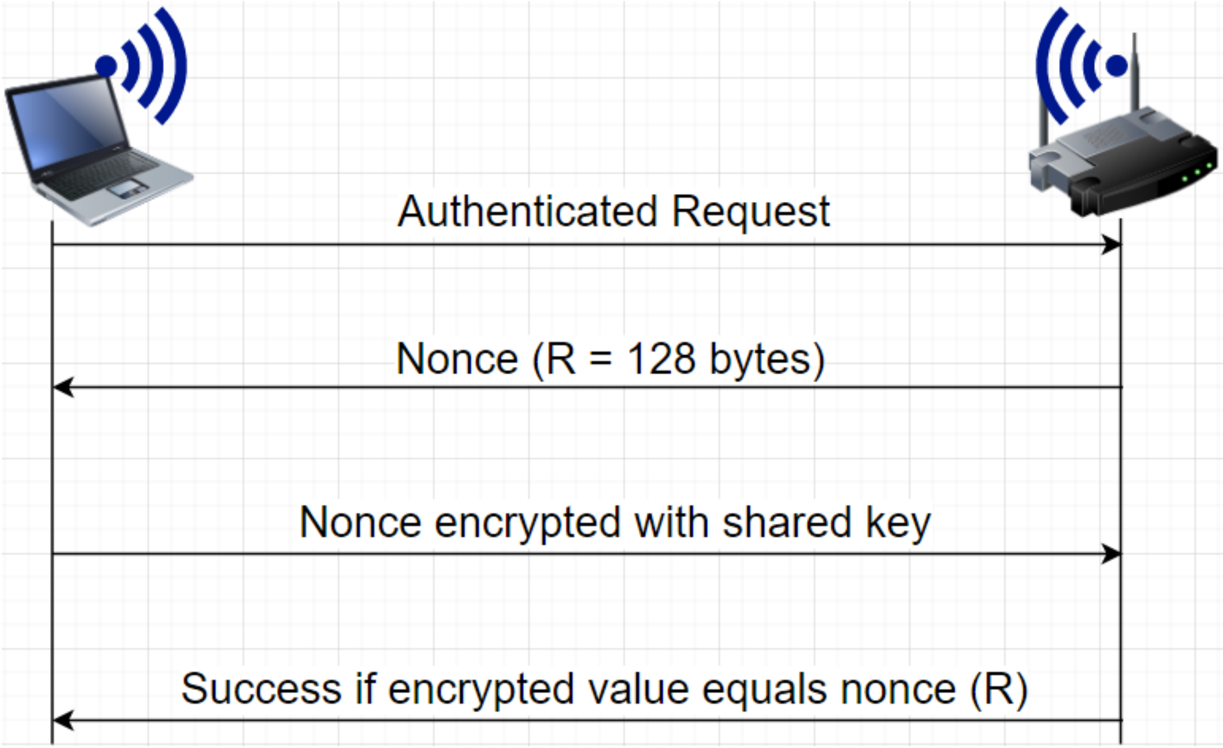
End-point Authentication W/Nonce

Nonce: number (R) used only *once –in-a-lifetime*

How to prove Alice "live": Bob sends Alice *nonce*, R. Alice must return R, encrypted with shared secret key



WEP Authentication



Notes:

- not all APs do it, even if WEP is being used
- AP indicates if authentication is necessary in beacon frame
- done before association

Breaking 802.11 WEP Encryption

Security hole:

- 24-bit IV, one IV per frame, -> IVs eventually reused
  - ~16 Million IVs at high speed exhausted in 2 hours
  - Can inject own packets to speed up
  - There are also *weak IVs* that makes it easy to discover the key

IV transmitted in ***plaintext*** -> IV reuse detected

Attack :

Trudy causes Alice to encrypt known plaintext d_1 d_2 d_3 d_4 …
Trudy sees: $c_i = d_i$ XOR kIV_i
Trudy knows c_i d_i, so can compute $kIV_i = d_i$ XOR c_i
Trudy knows encrypting key sequence kIV_1 kIV_2 kIV_3 …
Next time IV is re-used, Trudy can decrypt!

Fluhrer, Mantin and Shamir (FMS) Attack

- For 50% success rate, capture around 5 Million packets on average
- Due to inherent weakness in RC4, output of encrypting with first few bytes of key not random
- Certain key values generate predictable pattern of encrypted data
  - Associated packets have IVs that are "weak"
  - Initially determine first bytes of key through IVs and then get the rest through statistical analysis
- Encrypted ARP packets can be captured and replayed to get encrypted ARP response

- Fluhrer, S., Mantin, I., and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4", Selected Areas of Cryptography: SAC 2001, Lecture Notes in Computer Science Vol. 2259, pp 1-24, 2001.

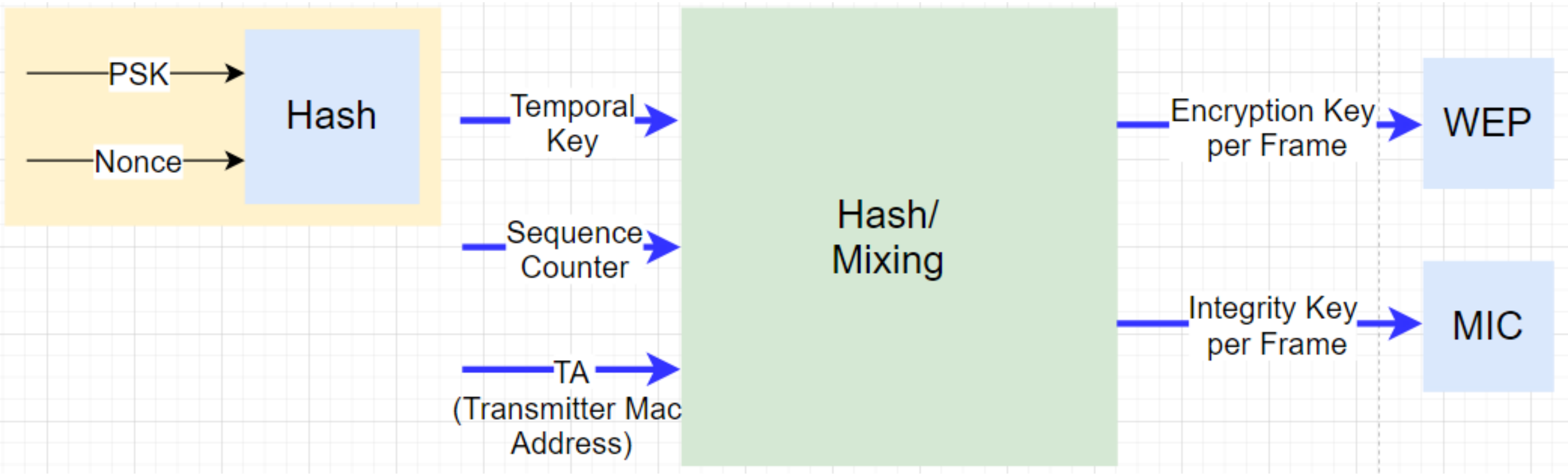
Wi-Fi Protected Access (WPA)

- WPA temporary solution to fix WEP while WPA2 developed
- WPA compatible with existing hardware that supported WEP
- WPA uses Temporal Key Integrity Protocol (TKIP)
  - RC4 for compatibility
  - Every packet encrypted with unique encryption key

WPA - New Features

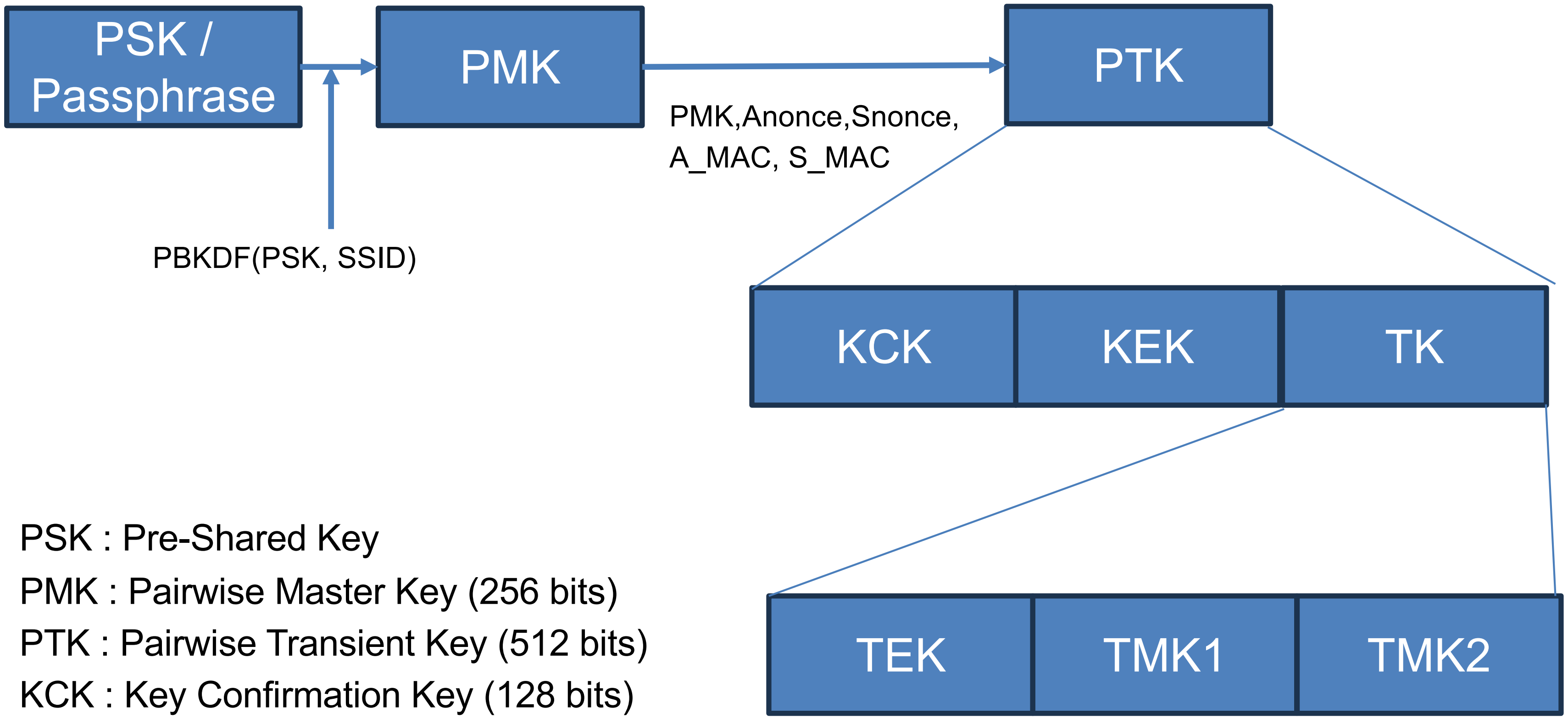
- Stronger Integrity than in WEP:
  - Special purpose Message Integrity Code (MIC) as opposed to WEP CRC
- To prevent FMS-style attacks
  - A new per-frame key is constructed using a cryptographic hash
- Temporal Key Integrity Protocol (TKIP) uses a cryptographic mixing function to combine a temporal key, the TA (transmitter MAC address), and the sequence counter into the WEP seed (128 bits)
  - Pre Shared Key (PSK) aka WPA-Personal similar to WEP-Key
    - However, it is not used for encryption
  - Instead, PSK serves as the seed for hashing the per-frame key

WPA Contd.



- TKIP changes the per packet key completely after every single packet
  - One key for encryption (128 bits)
  - One key for integrity (64 bits)

WPA Personal (TKIP) : Keys and keys everywhere



- PSK : Pre-Shared Key
- PMK : Pairwise Master Key (256 bits)
- PTK : Pairwise Transient Key (512 bits)
- KCK : Key Confirmation Key (128 bits)
- KEK : Key Encryption Key (128 bits)
- TK : Temporal Key (256 bits)
- TEK : Temporal Encryption Key (128 bits)
- TMK1 and 2: Temporal MIC Keys 1 and 2 (64 bits each)
- GTK: Group Temporal Key

WPA Contd.

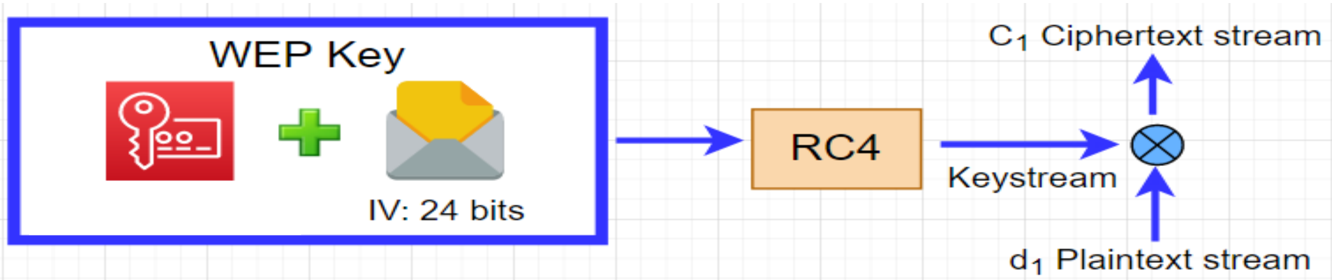
- WPA-Personal goes through a four-way handshake
  - Step 0: Both Client and AP derive Pairwise Master Key (PMK) from the PSK
  - Step 1: AP sends Anonce to client
  - Step 2a: Client derive Pairwise Transient Key PTK
    - PMK+Anonce+Snonce+ClientMac+APMac
  - Step 2b: Client sends Snonce and MIC to AP
  - Step 3a : AP calculates the same PTK as client
  - Step 3b: AP send GTK (Group key) and MIC to client
  - Step 4: Client sends ACK of key installation

WPA Contd.

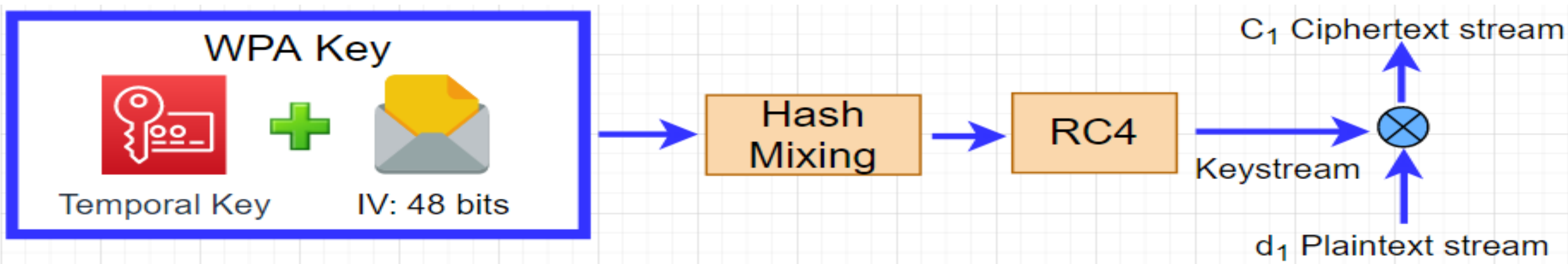
- The WEP IV is extended to 48 bits, and used as a packet sequence counter
  - A per packet sequence counter is used to prevent replay attacks
  - If a packet is received out of order, it is dropped by the receiving station

Recap: WEP vs WPA Security

- WEP IV extended to 48-bit IV
  - Reuse > 100 years for replay of the same IV
  - Used as packet sequence counter to prevent replay attacks



- In WPA, every packet encrypted with unique encryption key



WPA PSK Weakness – No Exam

- WPA, using the Temporal Key Integrity Protocol, was cracked by Erik Tews and Martin Beck

- Thomas Roth demonstrated at the 2011 Black Hat conference that WPA PSKs can be cracked quickly and easily using Amazon's Elastic Compute Cloud (EC2) service
  - He cracked his neighbor's WPA password in 20 minutes using a dictionary attack and a list of 70 million words – Not recommended
  - The attack only required one instance of Roth's self-made Cloud Cracking Suite (CCS) tool running in the cloud
  - It reached about 50,000 PSKs/s

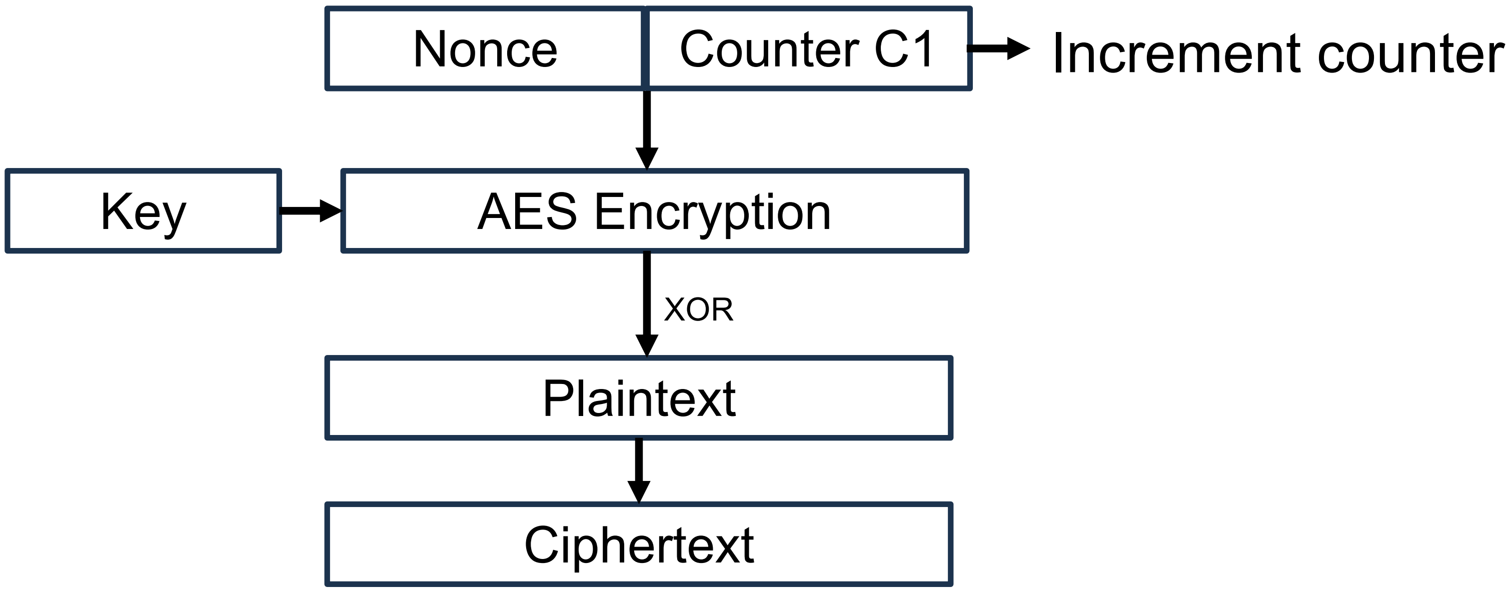
WPA2 (2004)

- New AP hardware
  - RC4 off-load hardware doesn’t do AES
- 128-bit block size
- AES-CCMP
  - AES instead of RC4 in TKIP and WEP
  - CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol)
  - Authenticated Encryption with Associated Data (AEAD)
    - Encryption and authentication combined
    - Additional data such as sequence no, port no etc.
  - PTK is 384 bits in AES-CCMP

WPA2 - Encryption

AES in CTR mode for encryption

- Initialize counter & nonce
- Each plaintext block is XORed with AES encrypted values to produce ciphertext
- Counter is incremented for each block



WPA2 – Integrity and Authentication

CBC-MAC for data integrity and authentication

- Plaintext block with additional data is fed to AES in CBC mode
  - XORing each block with the output of the previous ciphertext
- Last step produces MAC ensuring authentication and integrity.

- Receiver re-calculates the MAC for verification

WPA2 vs WPA

	Encryption	Authentication
WPA-Personal	TKIP	PSK
WPA-Enterprise	TKIP	802.1x/EAP
WPA2-Personal	AES-CCMP TKIP	PSK
WPA2-Enterprise	AES-CCMP TKIP	802.1x/EAP

802.1x: Port based authentication

EAP: Extensible Authentication Protocol

WPA3

- WPA3 introduced in 2018
  - WPA3-Personal mode uses a 128-bit encryption
  - WPA3-Enterprise uses 192-bit encryption.
    - We will look at WPA-3 Enterprise along with Enterprise network security (802.1X) in later weeks

  - Simplified security for IoT devices (EasyConnect)
    - Use QR code or NFC to securely onboard devices

  - Introduces Opportunistic Wireless Encryption (OWE)
    - Encryption between client and AP for open networks

WPA3

- Forward secrecy
  - Each session uses unique encryption key
  - A previous session is not compromised even if long-term keys are compromised

- PSK replaced by Simultaneous Authentication of Equals (SAE)
  - Based on the IETF Dragonfly key exchange.
  - Protection against brute force attacks

- Require use of Protected Management Frames
  - Ensures management frames are encrypted and authenticated
  - Protects against classic de-authentication attacks

Breaking WPA2/3 (self-read, examinable)

- KRACK
  - Key Reinstallation Attack (2017)
  - Attack against the 4-way handshake of the WPA2 protocol
  - AES-CCMP: Replay and decrypt packets
  - WPA-TKIP: Replay, decrypt and forge packets

- DragonBlood – Attack on DragonFly key exchange for WPA3

Only read parts of the papers that explains how the attack is launched and what vulnerabilities are exploited.

Open Wifi Security Challenge

- Openly accessible networks (OpenSSID) such as at airports or restaurants, there may neither be PSKs nor certificates
- Captive Portals* check your authenticity at logon time (often protected with SSL to protect against eavesdropping on your password)
- Only authenticated clients will receive service as packet filtering is deployed to only allow accessing the logon page until successful authentication
- Once logon authentication has been checked: no further security measures
  - No protection for your user data

*A captive portal is a web page to which a client is redirected when they connect to a guest SSID.

Open Wi-fi Security Challenge (2)

- You can deploy your own measures, e.g. VPN or SSL
- Configuration is often tedious or not even supported by communication partner
- Performance is affected because of additional (per-packet) overhead
  - Plus: your session can be stolen by using your MAC & IP addresses!
- Read about WiFi Certified Enhanced Open (not examinable)

Opportunistic Wireless Encryption (OWE)

- The client and AP use a pairwise secret derived from an initial Diffie–Hellman key exchange (DHKE).
  - Essentially agree upon a shared key to be used for encryption of traffic between the two end-points.
- IETF RFC 1180 has details of OWE
  - Describes how do use DHKE elements during Wi-fi Association.

- No prior authentication is needed, it provides “ENCRYPTION”
  - Improvement over previous no security.

- Captive Portal could authenticate – as discussed in previous foil

Acknowledgements

- Acknowledgement: foils are adapted mainly from **Introduction to Computer Networks and Cybersecurity by Wu and Irwin, CRC Press (Chapter 21)**
- Some foils are also from Günter Schäfer, **Security in Fixed and Wireless Networks**, Wiley (new edition available in German only, English in 2015)
- A few foils are from Adrian Perrig (ETH)

- Refer to Cybok Network Security KA Section:7 for brief summary