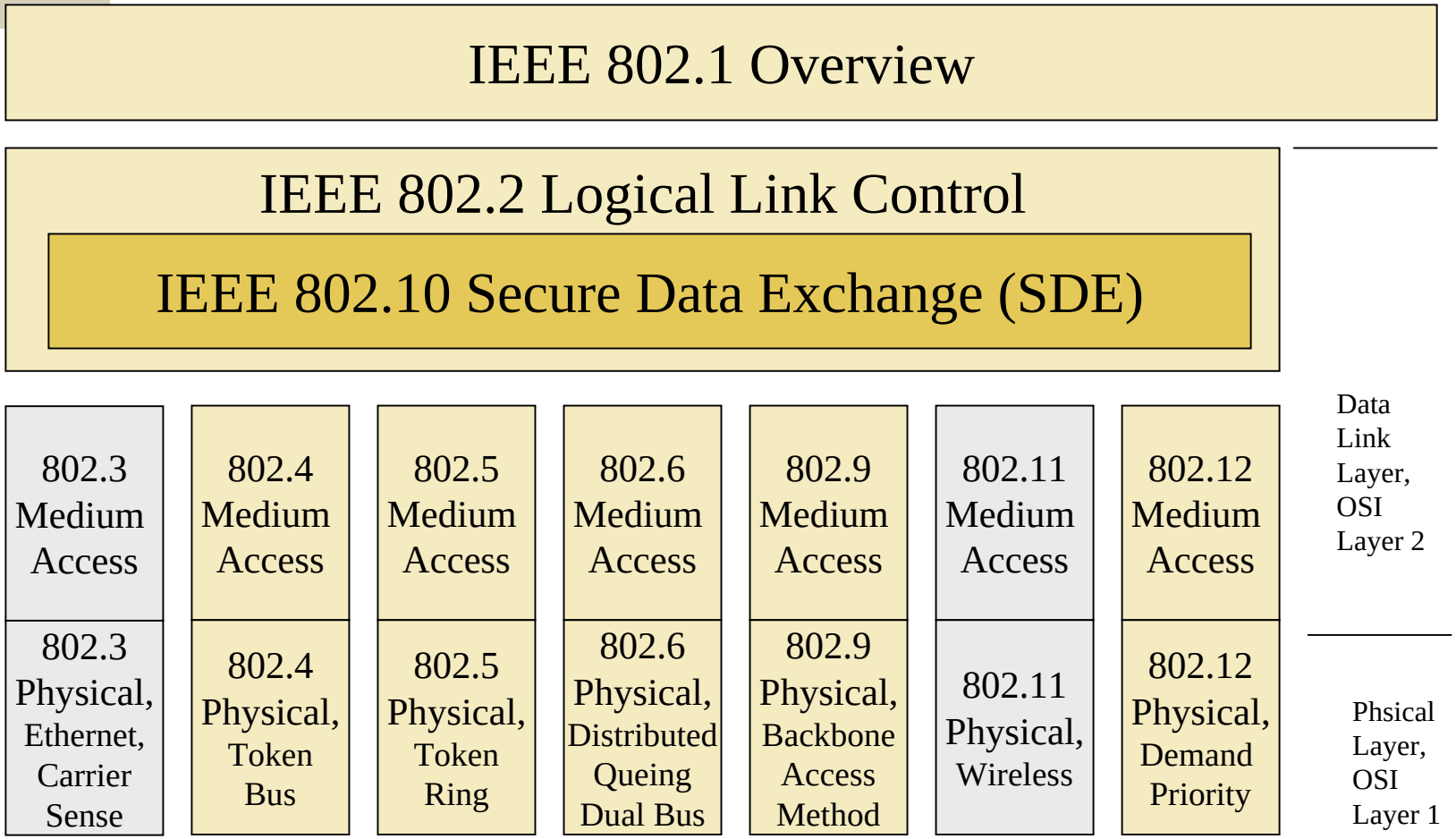
802.11 WLAN Security

Mauri Kangas, Helsinki University of Technology, 20.04.2004

Contents

- IEEE 802.10 Secure Data Exchange (SDE)
- IEEE 802.11 Security based on Wired Equivalent Privacy (WEP)
- IEEE 802.1x Port-Based Network Access Control
- IEEE 802.11i Standards-Based Wireless Security

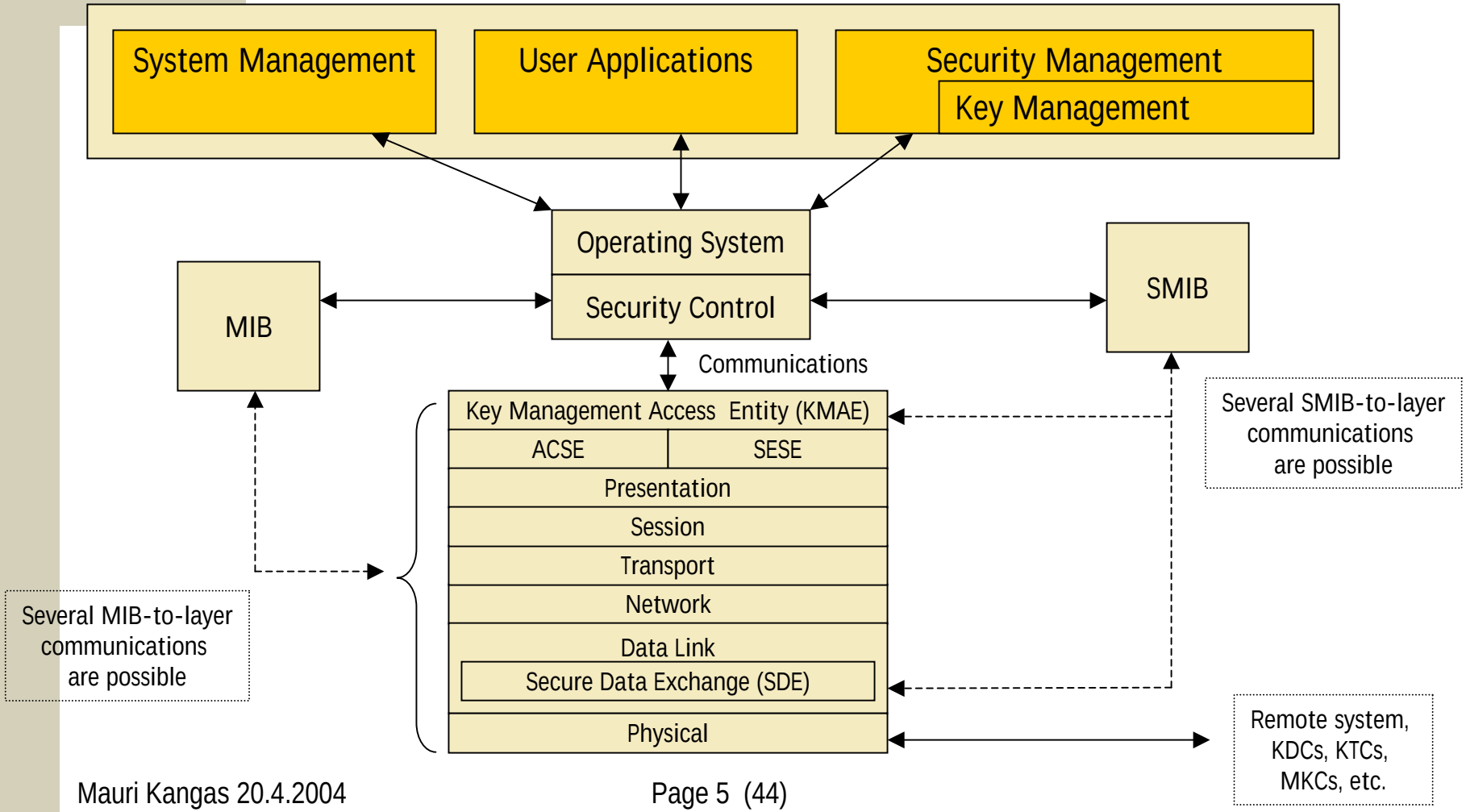
IEEE802.10 SDE in IEEE802 Reference Model



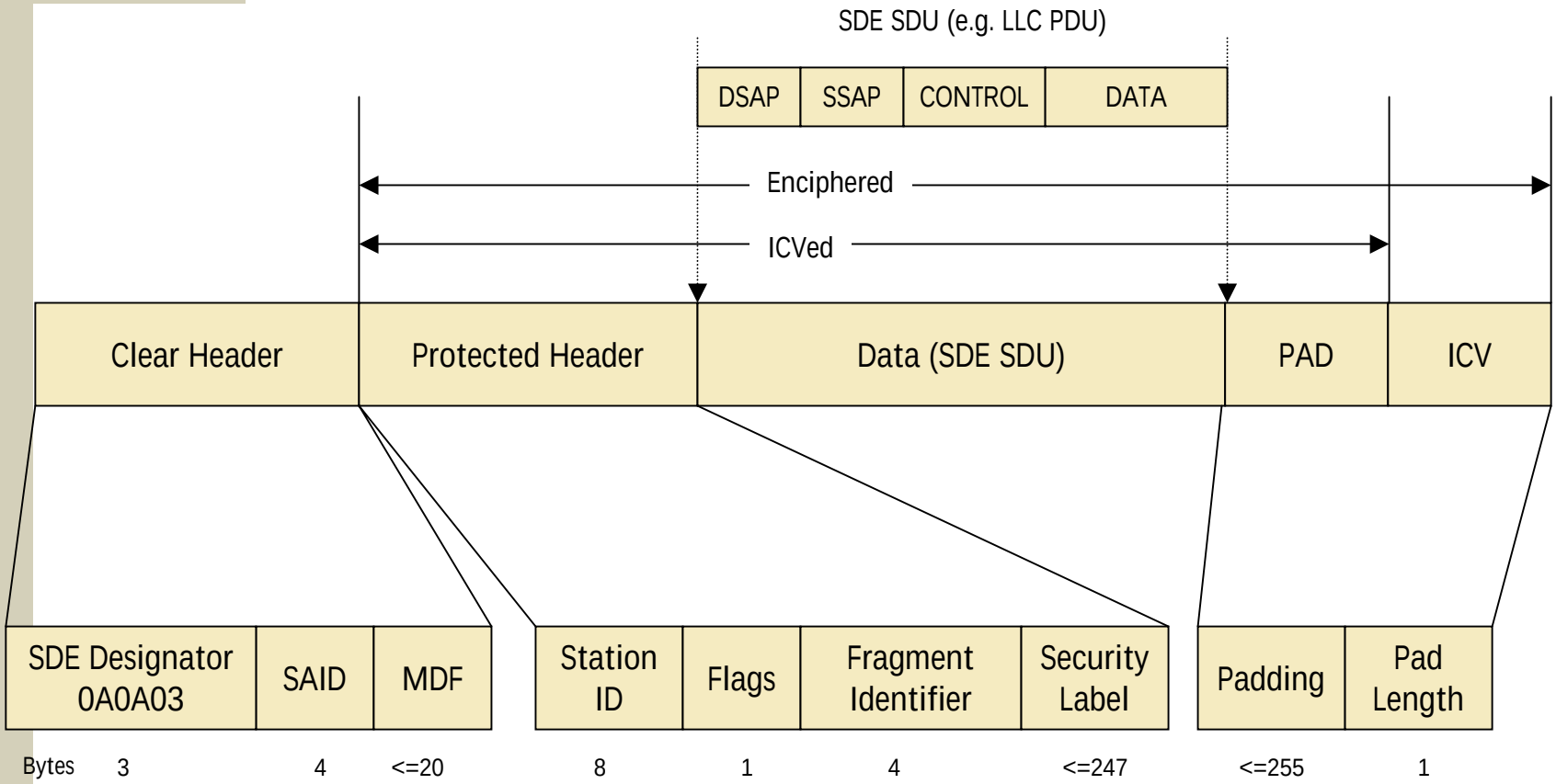
SDE Security Services

- **Data Confidentiality:** encipherment of the SDE SDU allows the use multiple cryptographic algorithm, but key change must happen out-of-band
- **Connectionless Integrity:** using integrity check value use and insertion into the SDE SDU
- **Data Origin Authentication:** achieved by the use of key management, Station ID is placed in the protected header part of the message. Data origin authentication can only provided in conjunction with the integrity service.
- **Access Control:** is provided by one or combination of the following: key management, system management, and the labelling of SDE SDUs. The SDE entity's use of security associations supports management's access control decisions. The SDE entity cannot deliver PDU unless a security association exist

Standard for Interoperable LAN/MAN Security (SILS) in the OSI Architecture



Structure of the SDE PDU



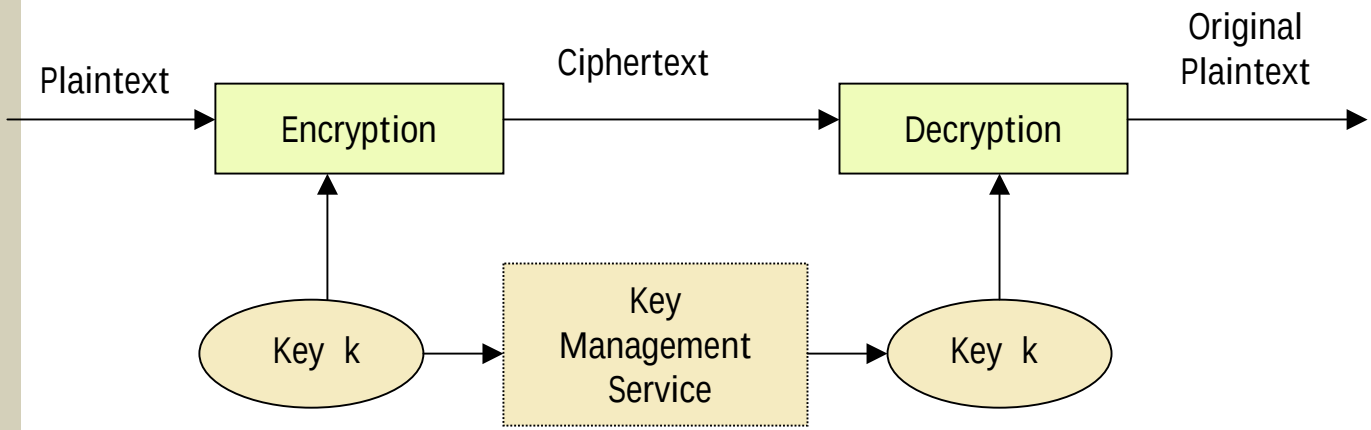
IEEE 802.11 Security - The Wired Equivalent Privacy (WEP) Algorithm

- Specified in IEEE 802.11 standard
  - ✓ Exchange of keys is not defined in IEEE 802.11; key distribution must be done with out-of-band communication
- Reasonably strong - security based on the following parameters:
  - ✓ Difficulty of discovering the secret key
  - ✓ Length of the secret key
  - ✓ Frequency of changing keys and the IV
- Self-synchronizing
  - ✓ WEP is self-synchronizing for each message
- Efficient
  - ✓ Can be implemented in hardware or software
- Exportable
  - ✓ Target has been to make it exportable from U.S., but there is no guarantee of that
- Optional
  - ✓ The use of WEP algorithm is optional in 802.11

The WEP Algorithm - Theory of Operation

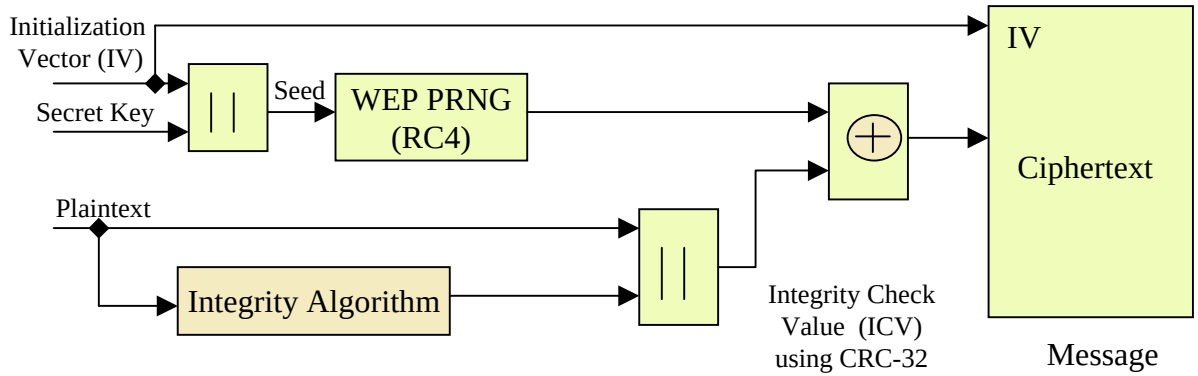
- Encryption: $E_k(P) = C$ P = Plaintext, C = Ciphertext
- Decryption: $D_k(C) = P$ P = Plaintext, C = Ciphertext
- Symmetric encryption/decryption process with the same key:

  $\Rightarrow D_k(E_k(P)) = P$

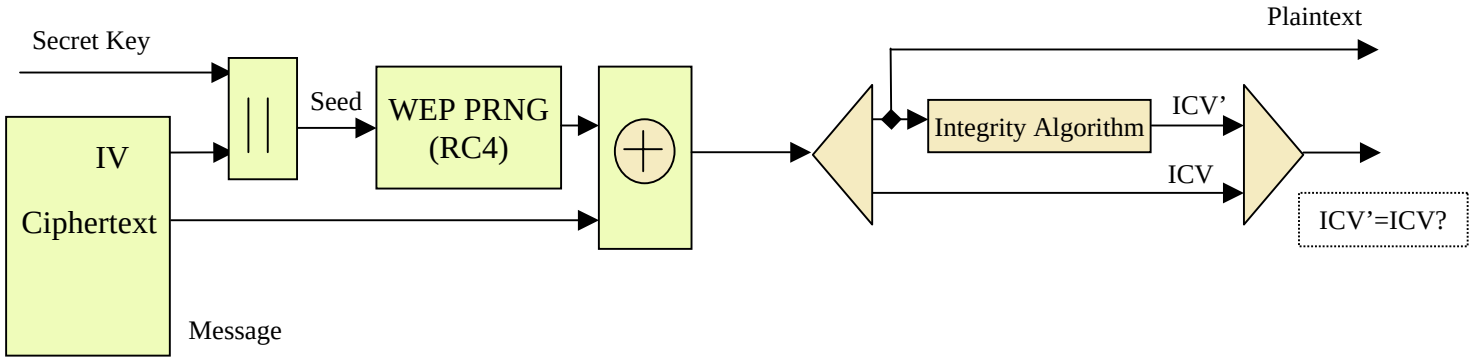


WEP Block Diagram

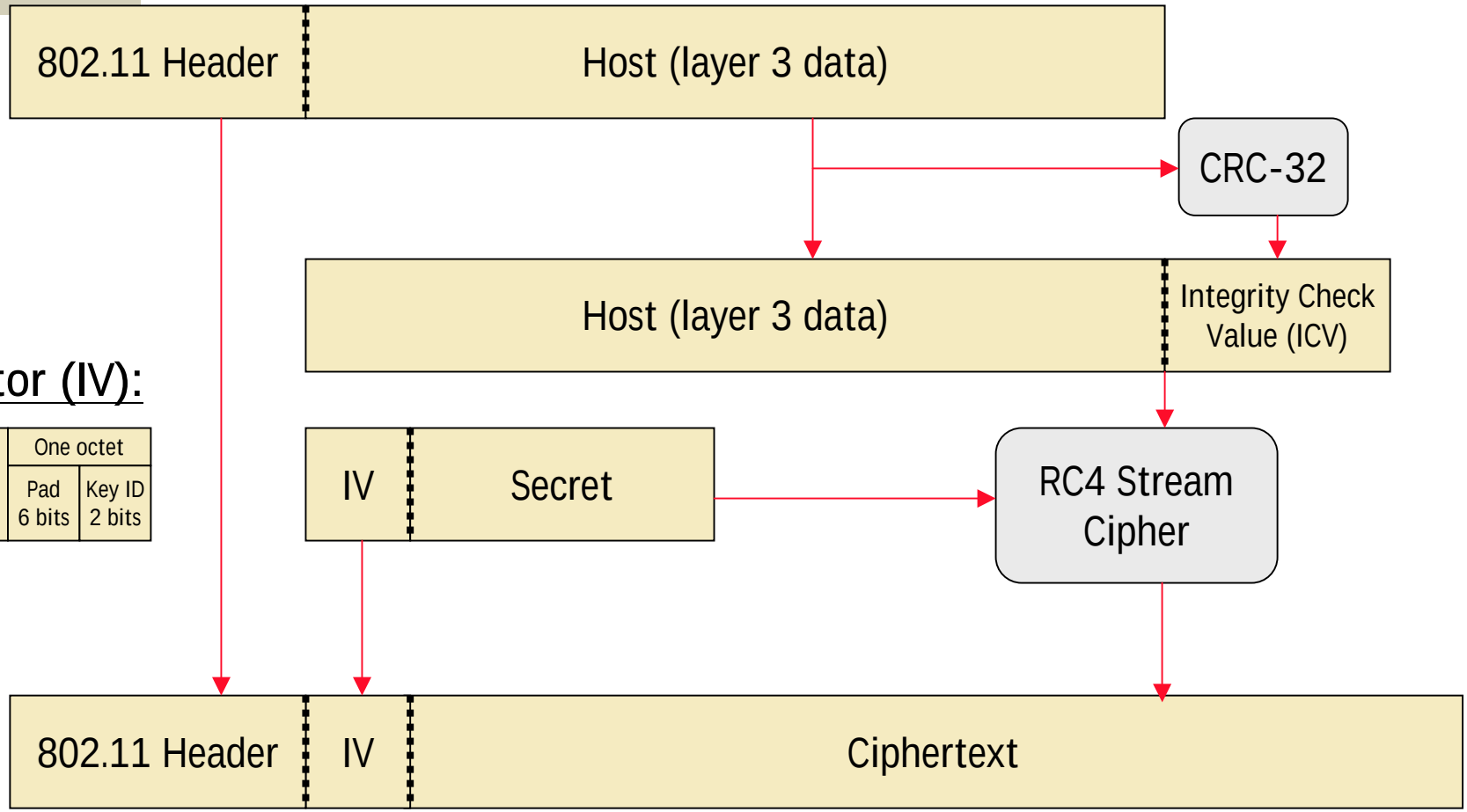
Encipherment:



Decipherment:



WEP Frame Body Extension in Encryption



InitVector (IV):

Init Vector (3)	One octet	
	Pad 6 bits	Key ID 2 bits

RC4 Stream Cipher

- RC4 consists of two parts: A key scheduling algorithm KSA, which turns a random key (whose typical size is 40-256 bits) into a initial permutation S of {0, 1, ..., N-1}, and an output generation part PRGA which uses permutation to generate a pseudo-random output sequence.
- The PRGA initializes two indices **i** and **j** to 0, and then loops over four simple operations which increment **i** as a counter, increment **j** pseudo randomly, exchange the two values of S pointed to by **i** and **j**, and output the value of S pointed to by **S[i] + S[j]** (modulo N). Note that every entry of S is swapped at least once (possibly with itself) within any N consecutive rounds, and thus the permutation S evolves fairly rapidly during the output generation process.
- The KSA consist of N loops that are similar to the PRGA round operation. It initializes S to be identity permutation and **i** and **j** to 0, and applies the PRGA round operation N times, stepping **i** across S, and updating **j** by adding **S[i]** and the next word of the key (in cyclic order).

```
KSA(K):
  Initialization:
        For i = 0,...,N-1
              S[i] = i
        j = 0
  Scrambling:
        For i = 0,...,N-1
              j = j + S[i] + K[i mod L]
              Swap( S[i], S[j] )

PRGA(K):
  Initialization:
        i = 0
        j = 0
  Generation loop:
        i = i + 1
        j = j + S[i]
        Swap( S[i], S[j] )
        Output z = S[ S[i] + S[j] ]
```

(Note: K is the concatenation of IV and WEP-key, L is length of key in bytes)

IEEE 802.11 Authentication

- Open System Authentication: communicating parties simply exchange their identities, which is the basis for authentication
- Shared Key Authentication:
  - ✓ A send MAC-auth-message with “Shared Key”-algorithm-ID and station A identifier
  - ✓ B responses with authentication frame that includes 128-bit ***challenge text***, which is generated by using WEP PRNG
  - ✓ A transmits the authentication frame, which includes the ***challenge text enciphered with WEP***
  - ✓ B receives the frame and decrypts it using WEP and the secret key shared with A. If the decryption results the original challenge text, B informs A about success, otherwise failure is communicated.

Weaknesses of WEP

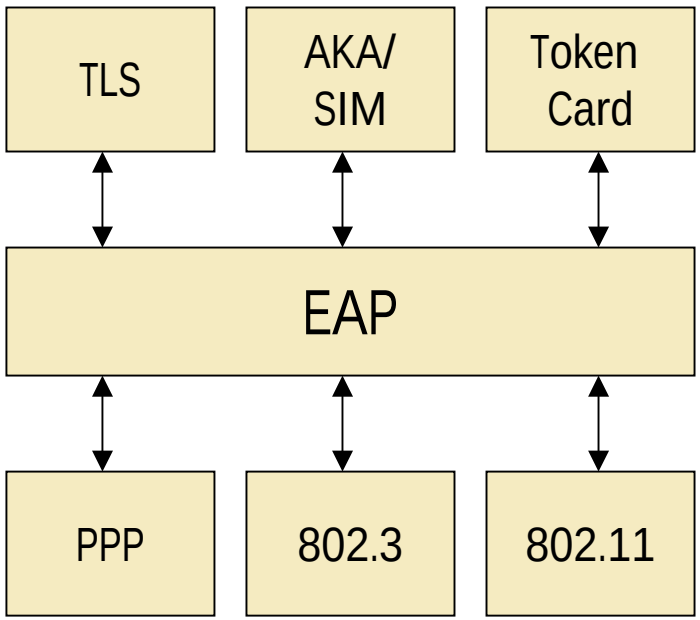
1. **WEP key recovery** - WEP uses the same WEP key and a different IV to encrypt data. The IV has only a limited range (0 to 16777215) to choose from. Eventually, the same IVs may be used over and over again.
2. **Unauthorized decryption and the violation of data integrity** – Once the WEP key is revealed, a hacker may transform the ciphertext into its original form and understand the meaning of the data. Based on the understanding of the algorithm, a hacker may use the cracked WEP key to modify the ciphertext and forward the changed message to the receiver.
3. **Poor key management** – A proper WEP key is typed into a wireless device associated in a wireless network to enable the WEP. Unfortunately, there are no mechanisms to renew the stored WEP key.
4. **No access point authentication** – WEP only provides a method for network interface cards (NICs) to authenticate access points. There is no way for access points to authenticate the NICs.
5. **Short 40-bit encryption scheme** - At the time of its introduction, WEP employed a necessarily short 40-bit encryption scheme:
   - Wi-Fi security had to be weak if the specification was to be adopted as an international standard and if products were to be freely exported (U.S. export laws requested max. 40-bit key)

IEEE 802.1x - Port-Based Network Access Control

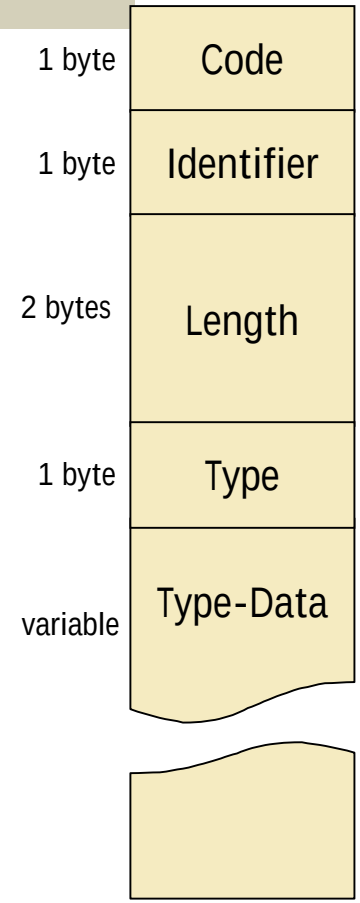
- IEEE 802.1x is a port-based authentication protocol for Ethernet networks
- The weaknesses in WEP usage for authentication in wireless IEEE 802.11 networks initiated a search for better solutions based on 802.1x specification
- 802.1x is based on IETF's Extensible Authentication Protocol (EAP)
  - ✓ EAP has been used as the basis for a number of network authentication extensions

The Extensible Authentication Protocol (EAP)

- Specified in RFC 2284
- Initially developed for PPP protocol, which is most widely used together with EAP
- EAP is a simple encapsulation, which can run over any link layer protocol and it can use any number of authentication protocols



EAP Requests and Responses – Message Format



Code = one byte that identifies the type of EAP packet;
1=request, 2=response

Identifier = an unsigned number to match requests with responses

---

Length = the length of the entire packet

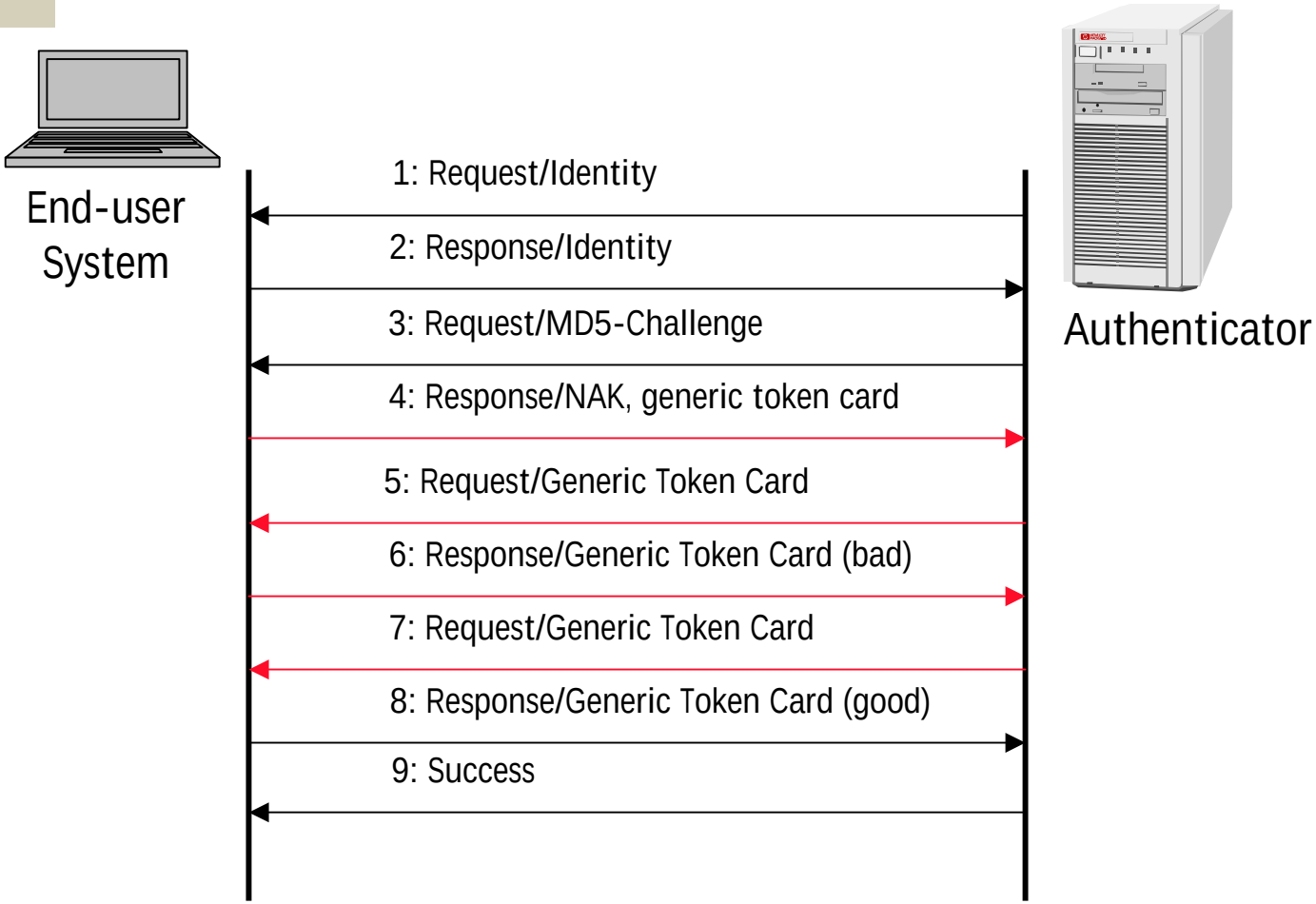
Type = type of the request or response

Type-Data = a variable length data field; interpretation according to type

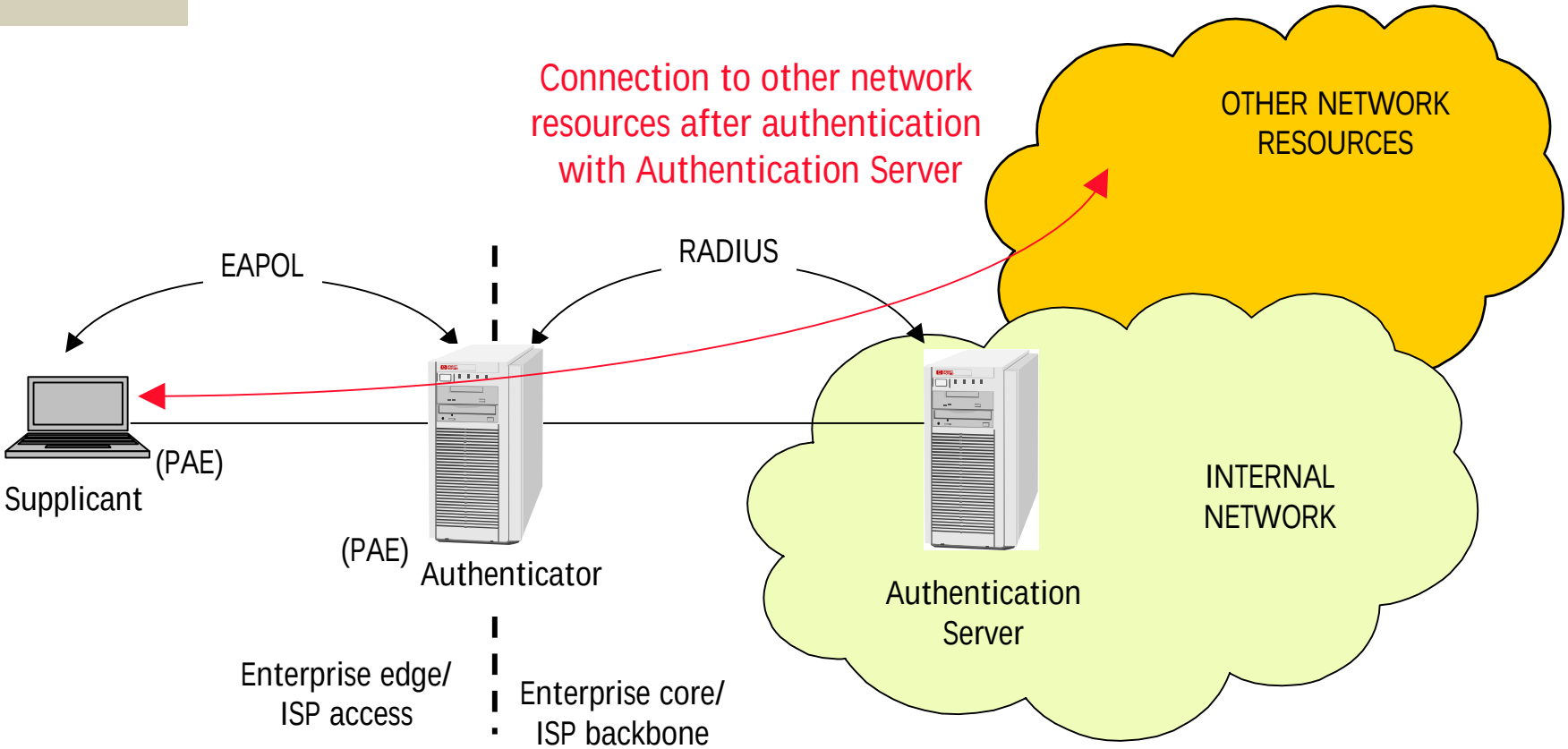
EAP Requests and Responses – Messages

- Identity: authenticator uses this as initial request
- Notification: authenticator can use this to send message to the user
- NAK: negative acknowledge is used to propose another authentication method
- MD-5 Challenge: Request contains a challenge to the end user; all implementations must support MD5, but user may propose some other authentication method
- One-Time Password (OTP): OTP system used in EAP is defined in RFC 1938; authenticator sends OTP challenge string to the user
- Generic Token Card: The request contain Generic Token Card information necessary for authentication
- TLS: for additional security RFC 2716 defines the use of Transport Layer Security (TLS) for authentication; EAP-TLS can ensure that the client is communicating with a legitimate authenticator

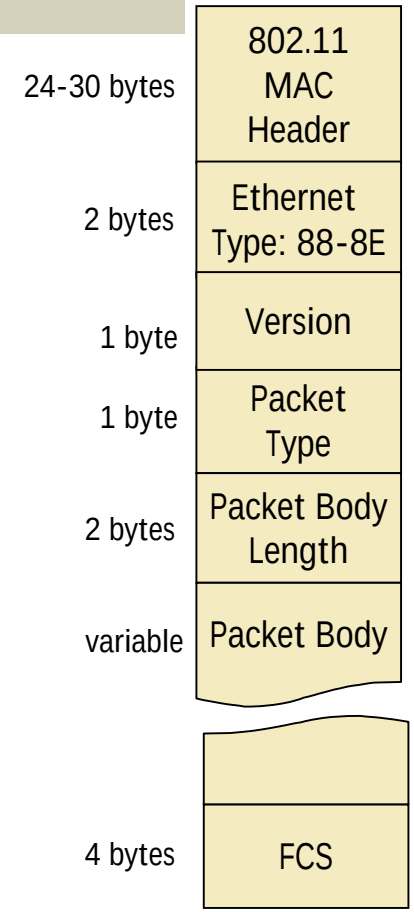
Sample EAP Exchange



802.1x Network Port Authentication - Architecture



EAPOL Encapsulation



802.11 MAC Header with length of 24-30 bytes

Ethernet type = EAPOL: 88-8E

Version = only version 1 is standardized today (Apr-2004)

Packet Type = EAPOL is extension to EAP and defines a few additional Messages to adapt EAP to port-based LAN environment

---

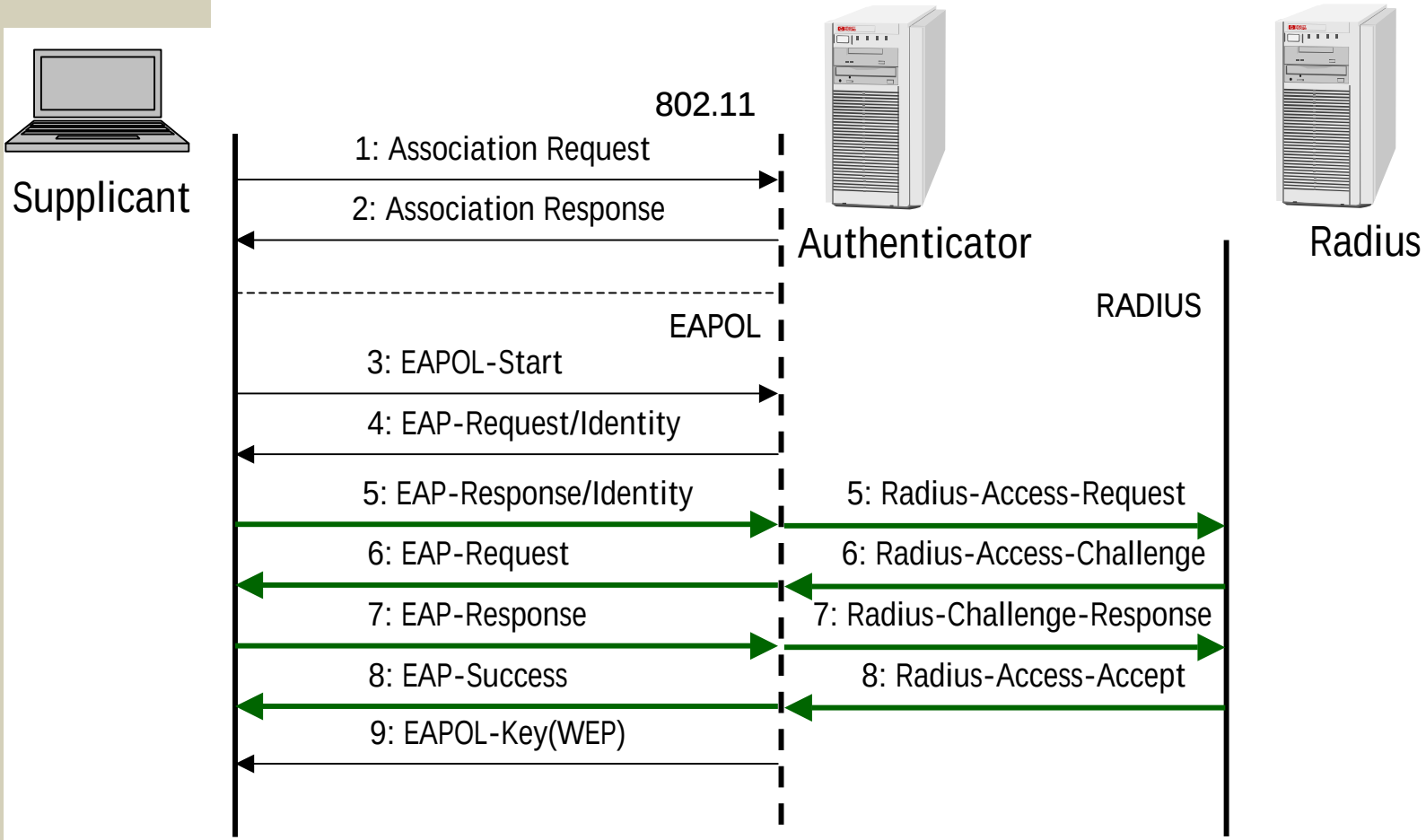
Packet Body Length = number of bytes in the packet body

Packet Body = a variable length data field, which exist in all other except EAPOL-Start and EAPOL-Logoff messages

---

FCS = frame checksum

802.1x on Wireless LANs – EAPOL Exhange



IEEE 802.11i Standards-Based Wireless Security

- 802.11i addresses the weaknesses of the WEP-based wireless security
- A lot of 802.11i tools have been released or announced and products are starting to appear
- The components of 802.11i include:
  - ✓ IEEE 802.1x port-based authentication framework
  - ✓ Temporal Key Integrity Protocol (TKIP) and Michael Message Integrity Check (MIC)
  - ✓ Advanced Encryption Standard (AES) encryption algorithm (to replace WEP's RC4 encryption)
  - ✓ Key hierarchy and management features
  - ✓ Cipher and authentication negotiation
  - ✓ Fast handoff, and secure de-authentication and disassociation
  - ✓ Roaming support

802.11i-WPA, interim solution

- WPA = Wi-Fi Protected Access
- In November 2002 Wi-Fi Alliance announced WPA, which is based on those components of 802.11i standard that are stable and may be deployed in existing 802.11 networks
- The initial release of WPA addresses AP-based 802.11 networks; ad-hoc networks will be addressed in the final standard
- The following components of 802.11i are included in the initial WPA release:
  - ✓ 802.1x authentication framework
  - ✓ TKIP (continues to use RC4 encryption) and Michael Message Integrity Check (MIC)
  - ✓ Key hierarchy and management
  - ✓ Cipher and authentication negotiation
- WPA specifies 802.1x/RADIUS implementation and a pre-shared key implementation

What is Temporary Key Integrity Protocol (TKIP)?

- TKIP enables secure, dynamic key generation and exchange.
- TKIP continues to use the RC4 encryption engine used by WEP, but provides the following important improvements over WEP:
  - ✓ Dynamic keys — Allows per-session and per-packet dynamic ciphering keys.
  - ✓ Message integrity checking (MIC) to ensure that the message has not been tampered with during transmission.
  - ✓ 48-bit IV hashing — Longer IV avoids the weaknesses of the shorter 24-bit WEP RC4 key.
  - ✓ Correction of WEP security vulnerability in which the IV is sent in clear text over the wireless connection.
- AES is being considered as a future replacement for RC4 in TKIP.

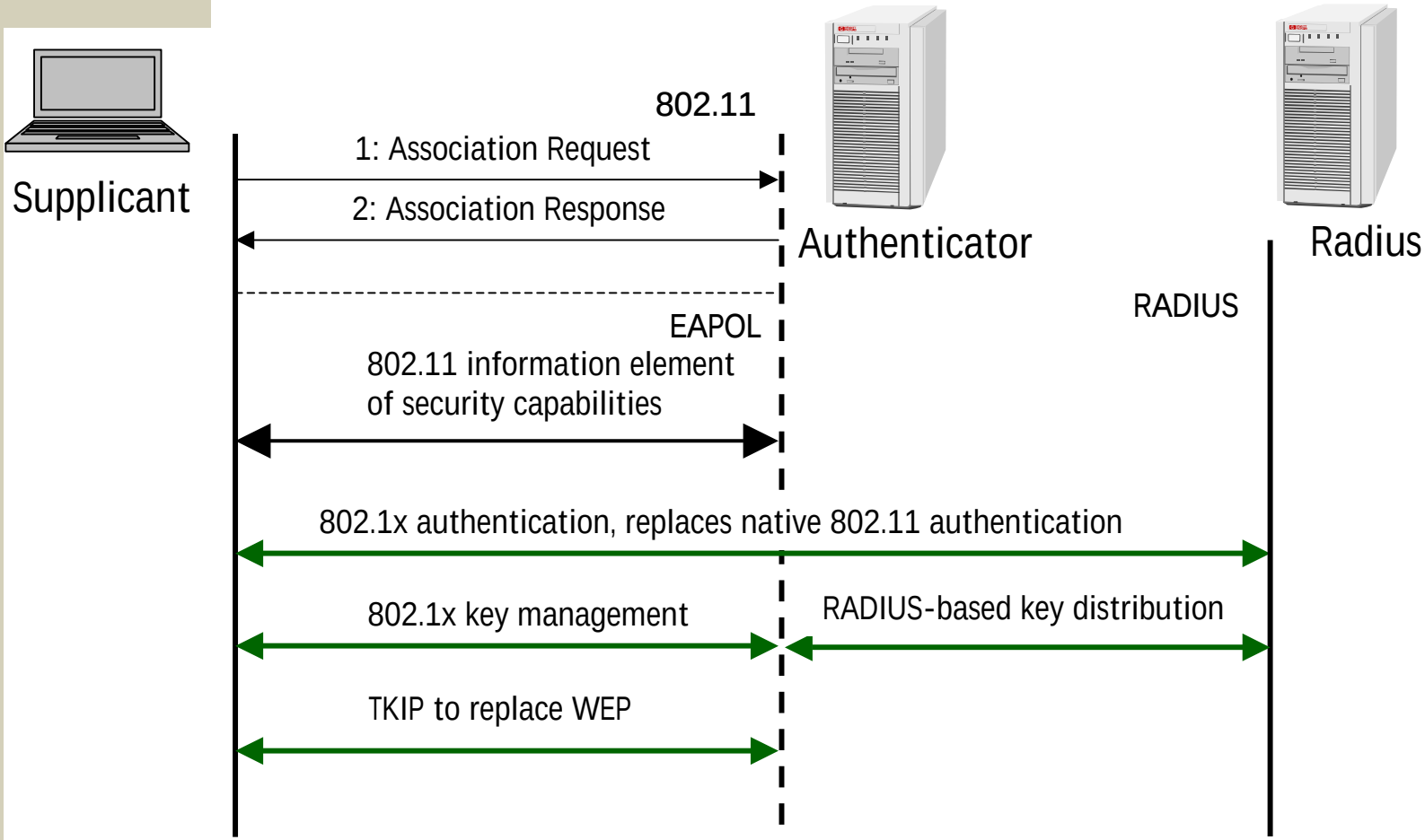
Key Hierarchy and Management in WPA

- WPA provides for more-secure and better key creation and management.
- This capability helps to safeguard against known key attacks.
- Client keys received via 802.1X key messages are used to derive base keys that are, in turn, used to derive per-packet keys.
- The master and base keys are not used to directly encrypt the data

Cipher and Authentication Negotiation in WPA

- WPA improves interoperability by requiring APs to “announce” their supported ciphers and authentication mechanisms.
- Clients wishing to authenticate to the AP via WPA can receive this announcement and respond appropriately (via a policy-based decision).
- In addition, the client can now choose the most secure cipher and authentication mechanism that it and the AP both support.

WPA 802.1x/RADIUS Solution



Preshared Key in WPA

- In the home, small office, or even some enterprises, WPA can be used in a pre-shared key mode that does not require an authentication server (or 802.1X).
- Access to the Internet and the rest of the wireless network services is allowed only if the pre-shared key of the computer matches that of the AP.
- This approach offers the setup simplicity of the WEP key, but uses the stronger TKIP encryption.
- The WPA pre-shared key differs dramatically from the WEP key. Under WPA, the pre-shared key is used only in the initial setup of the dynamic TKIP key exchange. As
- This base key is never sent over the air or used to directly encrypt the data stream. In contrast, the WEP key is static until manually changed by the user or administrator.

IEEE 802.11i-WPA Wrap-up

- Recent industry efforts are bringing more-robust native security to Wi-Fi networks. The basic 802.11 security solutions that are available “out of the box”—SSID, MAC address filtering, and WEP—are soon to be strengthened by replacing important components of WEP with WPA via software upgrades to the wireless client systems and APs.
- This solution will provide suitable security for both small home or business networks and larger networks.
- 802.1X- and/or VPN-based solutions provide more scalable solutions for large enterprise networks or networks that require more robust security.

IEEE 802.11i: AES Cipher Pseudo-Code

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin
    byte state[4,Nb]
    state = in
    AddRoundKey(state, w[0, Nb-1])
    for round = 1 step 1 to Nr–1
            SubBytes(state)
            ShiftRows(state)
            MixColumns(state)
            AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
    end for
            SubBytes(state)
            ShiftRows(state)
            AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])
    out = state
end
```

- Size of the input block, output block and state is always 128 bits (N_b=4).
- Key length can be 128, 192 or 256 bits, and the number of rounds (N_r) is 10, 12, and 14 respectively.
- In addition to the encryption pseudo-code on the left, there is need for key expansion routines, which will take 4*(N_r+1) rounds

AES – SubBytes() (1)

The **SubBytes()** transformation is a non-linear byte substitution that operates independently on each byte of the State using a substitution table (S-box). This S-box (Fig. 7), which is invertible, is constructed by composing two transformations:

1. Take the multiplicative inverse in the finite field GF(2^8), described in Sec. 4.2; the element {00} is mapped to itself.
2. Apply the following affine transformation (over GF(2)):

$$b_i' = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i \qquad (5.1)$$

for $0 \le i < 8$, where b_i is the i^{th} bit of the byte, and c_i is the i^{th} bit of a byte c with the value {63} or {01100011}. Here and elsewhere, a prime on a variable (e.g., b') indicates that the variable is to be updated with the value on the right.

In matrix form, the affine transformation element of the S-box can be expressed as:

AES – SubBytes() (2)

$$\begin{bmatrix} b'_0 \\ b'_1 \\ b'_2 \\ b'_3 \\ b'_4 \\ b'_5 \\ b'_6 \\ b'_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \qquad (5.2)$$

Figure 6 illustrates the effect of the `SubBytes()` transformation on the State.

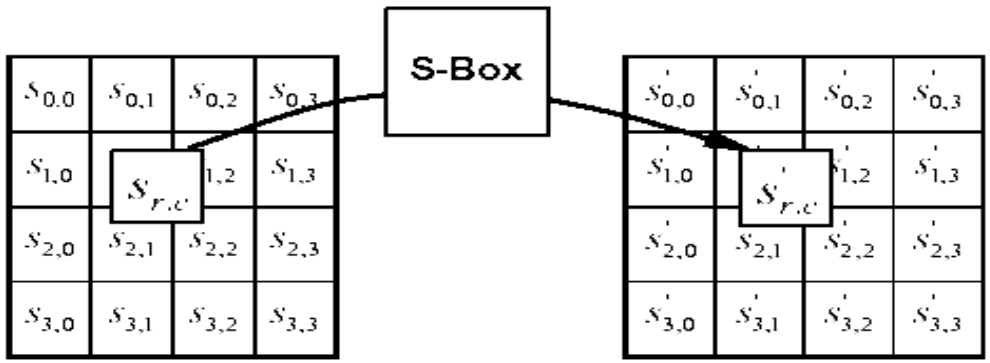


Figure 6. `SubBytes()` applies the S-box to each byte of the State.

AES – S-box used in SubBytes()

The S-box used in the **SubBytes()** transformation is presented in hexadecimal form in Fig. 7.

For example, if $s_{1,1} = \{53\}$, then the substitution value would be determined by the intersection of the row with index '5' and the column with index '3' in Fig. 7. This would result in $s'_{1,1}$ having a value of {ed}.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

AES – ShiftRows()

In the **ShiftRows()** transformation, the bytes in the last three rows of the State are cyclically shifted over different numbers of bytes (offsets). The first row, $r = 0$, is not shifted.

Specifically, the **ShiftRows()** transformation proceeds as follows:

$$s'_{r,c} = s_{r,(c+shift(r,Nb)) \bmod Nb} \quad \text{for } 0 < r < 4 \quad \text{and} \quad 0 \le c < Nb, \qquad (5.3)$$

where the shift value $shift(r,Nb)$ depends on the row number, r, as follows (recall that $Nb = 4$):

$$shift(1,4) = 1; \quad shift(2,4) = 2; \quad shift(3,4) = 3. \qquad (5.4)$$

This has the effect of moving bytes to "lower" positions in the row (i.e., lower values of c in a given row), while the "lowest" bytes wrap around into the "top" of the row (i.e., higher values of c in a given row).

Figure 8 illustrates the **ShiftRows()** transformation.

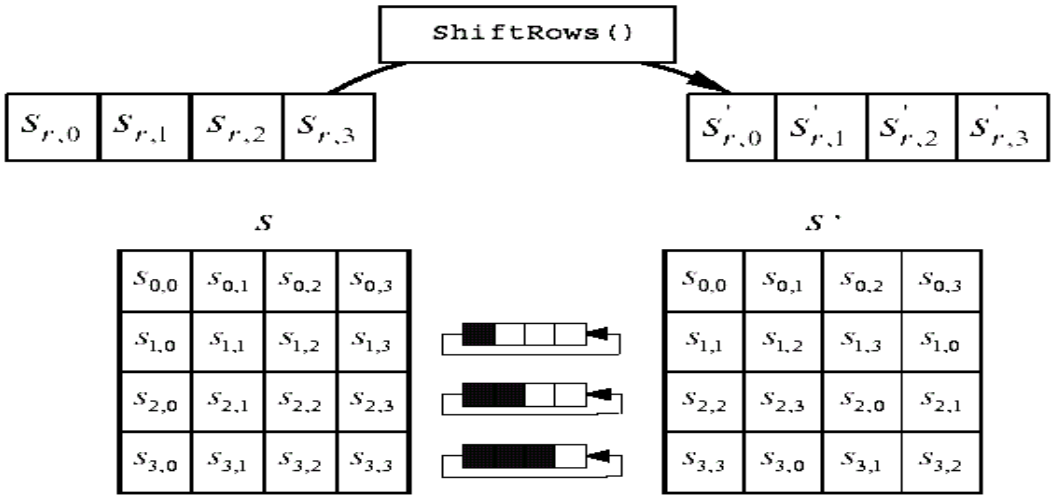


Figure 8. ShiftRows() cyclically shifts the last three rows in the State.

AES – MixColumns()

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < Nb. \qquad (5.6)$$

As a result of this multiplication, the four bytes in a column are replaced by the following:

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c}).$$

Figure 9 illustrates the **MixColumns()** transformation.

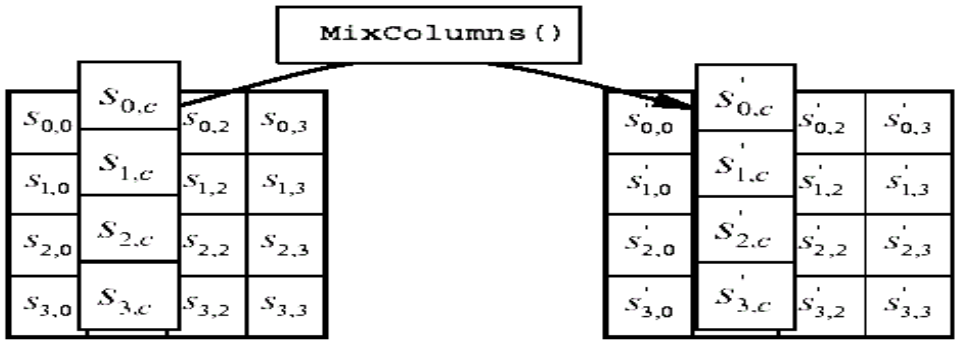


Figure 9. MixColumns() operates on the State column-by-column.

AES – AddRoundKey()

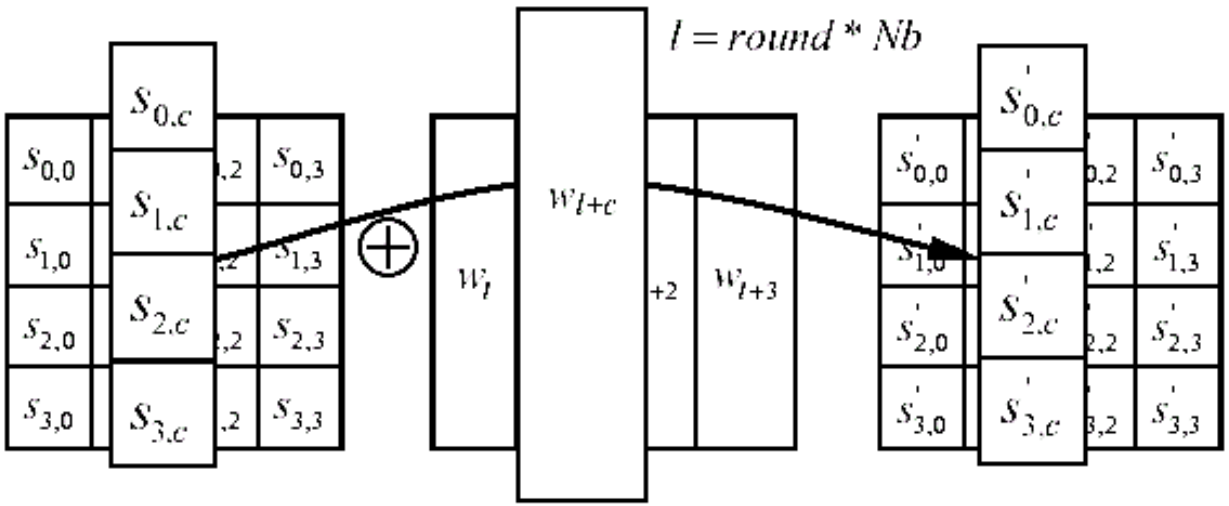
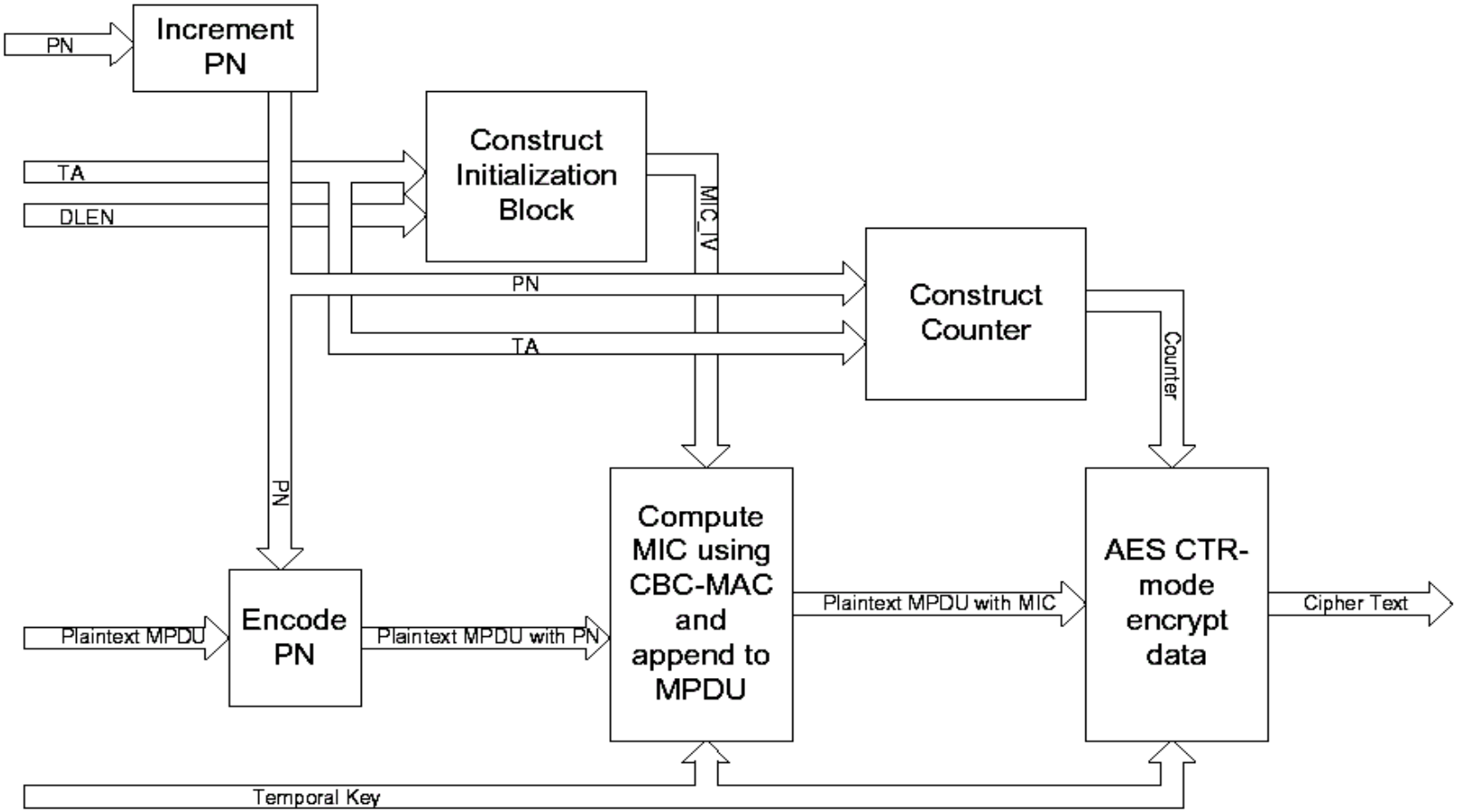
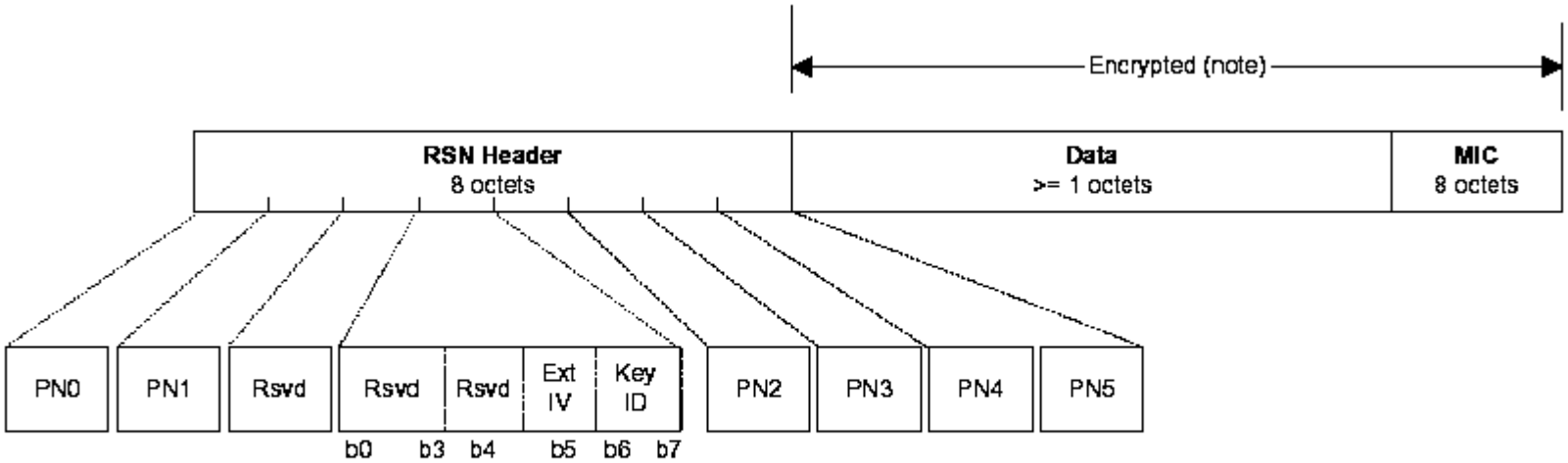


Figure 10. `AddRoundKey()` XORs each column of the State with a word from the key schedule.

Counter-Mode/CBC-MAC Protocol (CCMP) Encapsulation using AES-CTR Encryption

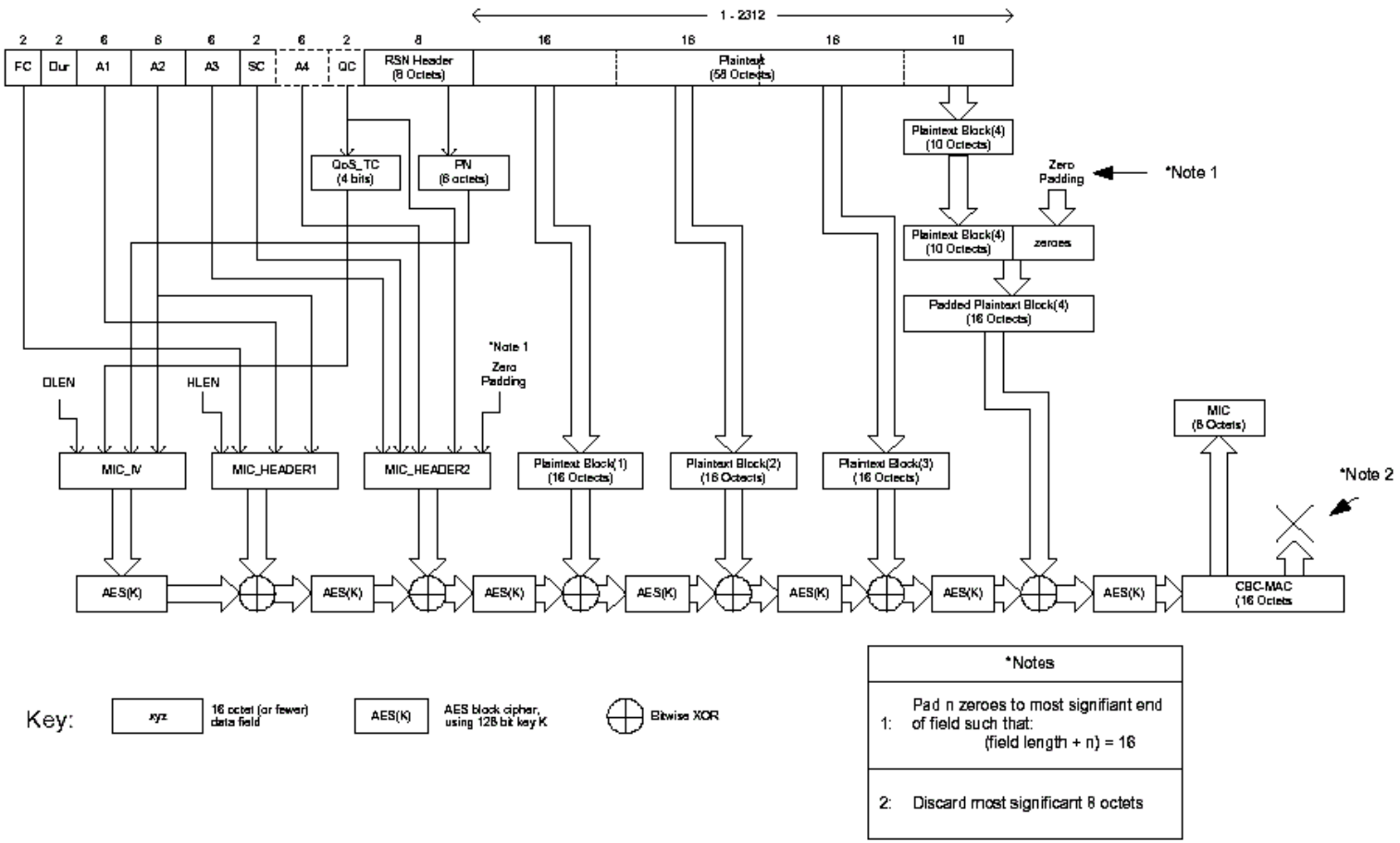


AES-CCMP Data Packet

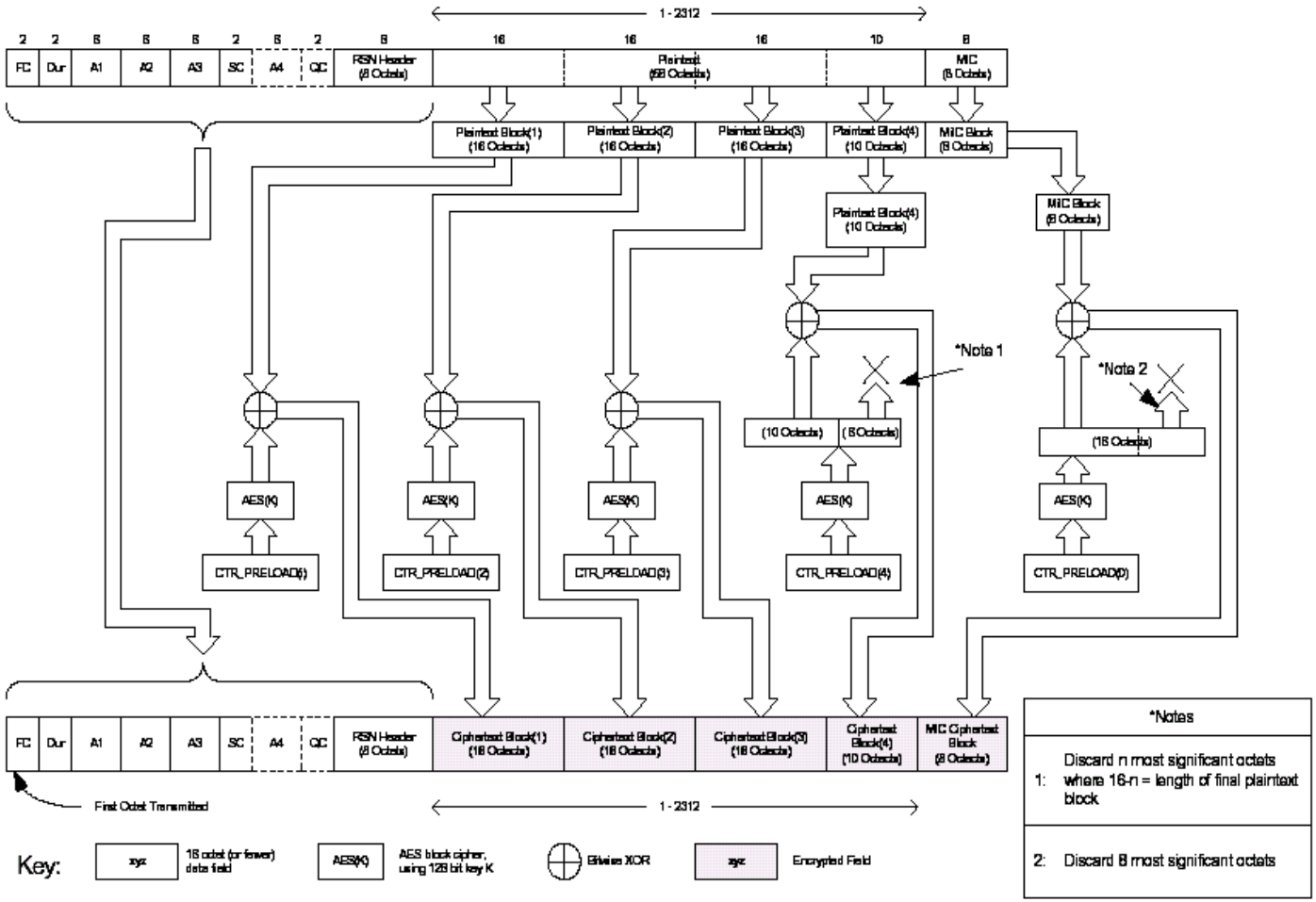


Note: The encipherment process has expanded the original MPDU size by 16 octets, 4 for the PN0-1 / Key ID field, 4 for the PN2-5 field and 8 for the Message Integrity Code (MIC).

AES-CCMP MIC Computation



AES-CCMP CTR-Mode Encryption



WRAP and AES-OCB (second choice in 802.11i)

- WRAP (Wireless Robust Authentication Protocol) privacy is based on 128-bit AES in OCB (Offset Code Book) mode:
  - ✓ Once the key has been derived and its associated state *initializ*ed, the IEEE 802.11 MAC uses the WRAP encapsulation algorithm with the key and the state to protect all unicast MSDUs
- WRAP Encapsulation contains the following steps:
  - ✓ Select the appropriate context based on the MSDU
  - ✓ Increment block count and the appropriate replay counter, based on the MSDU service class
  - ✓ Construct the Replay-Counter field of the final WRAP-protected MSDU payload
  - ✓ Construct the OCB nonce using the Replay-Counter, MSDU service class, and source MAC address
  - ✓ Construct an associated data block from the destination MAC address
  - ✓ AES-OCB encrypt the MSDU and associated data
  - ✓ Construct the MSDU payload from the replay counter, OCB encrypted data, and the OCB tag.

Some 802.11 Security Conclusions

- WEP is widely considered as broken
- WPA is an interim solution to the WEP vulnerability
- WPA uses a subset of 802.11i features and is generally believed as a major security improvement in wireless environment.
- WPA supports existing wireless infrastructure: Vendors can transit to the WPA standard through a software or firmware upgrade.
- 802.11i, “the final solution to wireless security”, is expected to provide the robust security required for wireless environment in the future.
- 802.11i based on AES-CCMP requires significant hardware changes
- VPN Security is another way to approach security problematics

References

1. Matthew S. Gast: 802.11 Wireless Networks, The Definitive Guide, O'Reilly 2002.
2. Bruce Potter, Bob Fleck: 802.11 Security, Securing Wireless Networks, O'Reilly 2003.
3. IEEE P802.11i, Draft Supplement Standard for Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: Wireless Medium Access Control (MAC) and Physical Layer Specifications: Specification for Enhanced Security, November 2002.
4. IEEE 802.11 Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY), IEEE 1999.
5. IEEE 802.10-1999, IEEE Standards for Local and Metropolitan Area Networks: Standard for Interoperable LAN/MAN Security (SILS) (Clause 2), IEEE 1998.
6. IEEE 802.10c-1998, IEEE Standards for Local and Metropolitan Area Networks: Supplement to Standard for Interoperable LAN/MAN Security (SILS), Key management (Clause 3), IEEE 1998.
7. Bruce Potter: Wireless Security's Future, IEEE Security & Privacy 2003.
8. FIPS: Advanced Encryption Standard (AES), FIP-197, November 2001.
9. DELL Computer Corporation: Wireless Security in 802.11 (Wi-Fi) Networks, White Paper, January 2003.
10. William Stallings: Wireless Communications and Networks, 2002.
11. Stanley Wong: The Evolution of Wireless Security in 802.11 Networks: WEP, WPA and 802.11 Standards, SANS Institute, May 2003.

Homework

- What security improvements are implemented in 802.1x and 802.11i compared to the original 802.11 WEP security?