UNIT IV APPLICATION LAYER SECURITY

Electronic Mail Security: Pretty Good Privacy, S/MIME, Domain Keys Identified Mail. Wireless Network Security: Mobile Device Security

4.1 PRETTY GOOD PRIVACY

PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications.

In essence, Zimmermann has done the following:

1. Selected the best available cryptographic algorithms as building blocks.
2. Integrated these algorithms into a general-purpose application that is independent of operating system and processor.
3. Made the package and its documentation, including the source code, freely available via the Internet.
4. Entered into an agreement with a company to provide a fully compatible, low-cost commercial version of PGP.

PGP has grown explosively and is now widely used. A number of reasons can be cited for this growth:

1. It is available free worldwide in versions that run on a variety of platforms.
2. It is based on algorithms that are extremely secure.
3. It has a wide range of applicability.
4. It was not developed by, nor is it controlled by, any governmental or standards organization
5. PGP is now on an Internet standards track.

Operational Description

(i) Authentication

(ii) Confidentiality

(iii) Compression

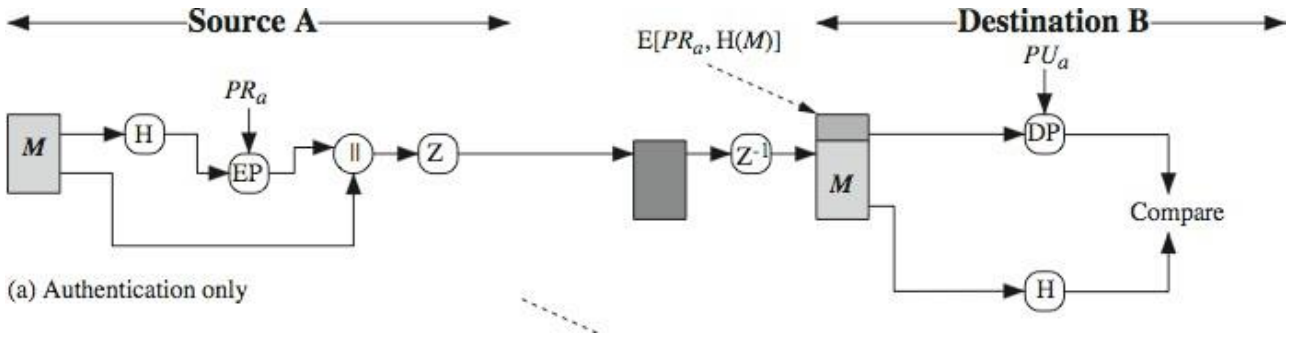
(iv) E-Mail compatibility

(v) Segmentation

(i) Authentication

- The sender creates a message.
- SHA-1 is used to generate a 160-bit hash code of the message.

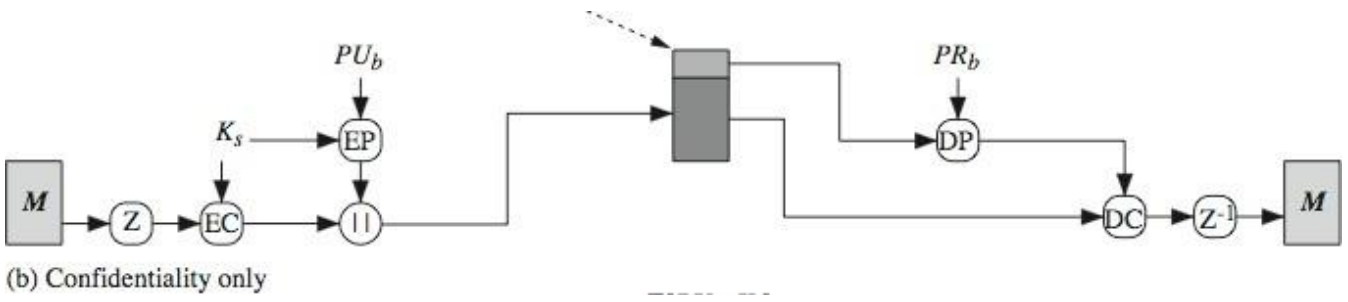
- The hash code is encrypted with RSA using the sender's private key, and the result is prepended to the message.
- The receiver uses RSA with the sender's public key to decrypt and recover the hash code.
- The receiver generates a new hash code for the message and compares it with the decrypted hash code. If the two match, the message is accepted as authentic.



(a) Authentication only

(ii) Confidentiality

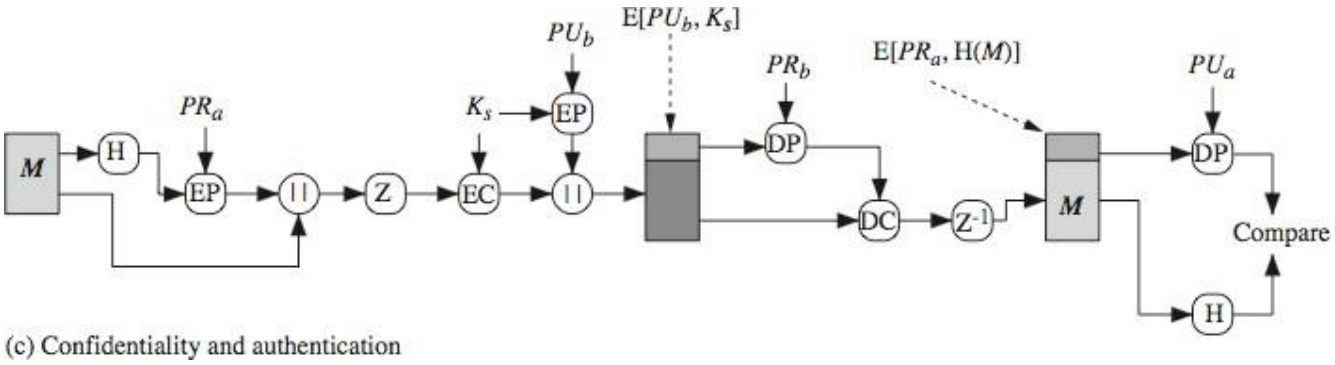
- The sender generates a message and a random 128-bit number to be used as a session key for this message only.
- The message is encrypted, using CAST-128 (or IDEA or 3DES) with the session key.
- The session key is encrypted with RSA, using the recipient's public key, and is prepended to the message.
- The receiver uses RSA with its private key to decrypt and recover the session key.
- The session key is used to decrypt the message.



(b) Confidentiality only

(iii) Confidentiality and Authentication

When both services are used, the sender first signs the message with its own private key, then encrypts the message with a session key, and then encrypts the session key with the recipient's public key.



(c) Confidentiality and authentication

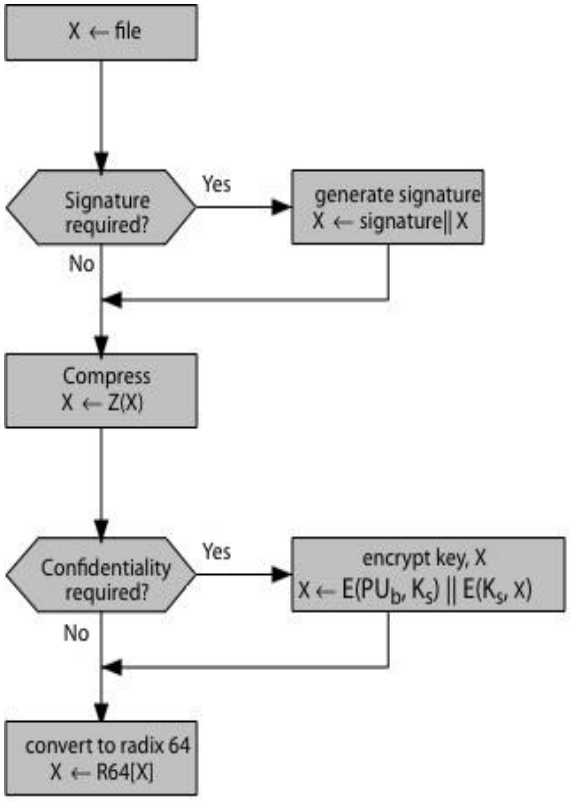
(iv) **Compression**

PGP compresses the message after applying the signature but before encryption. The compression algorithm is indicated by Z for compression and Z^{-1} for decompression. The signature is generated before compression for two reasons:

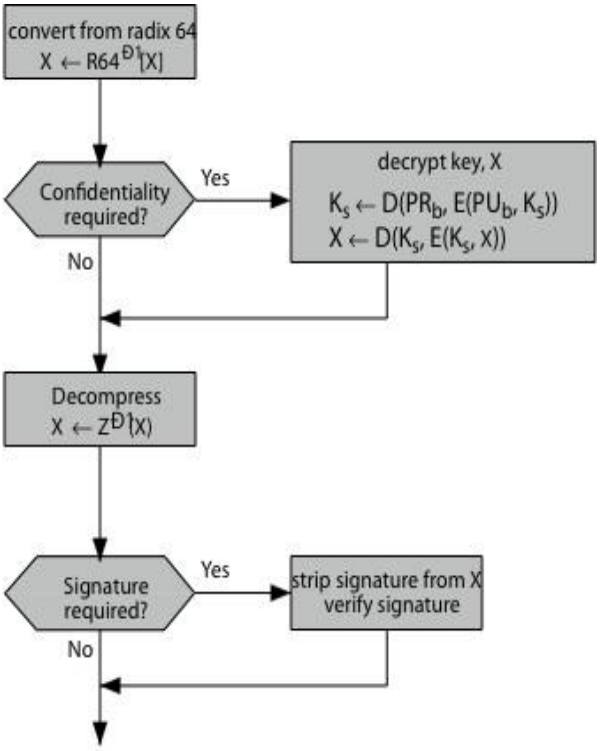
a) It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification.

b) Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty. The algorithm is not deterministic.

c) Message encryption is applied after compression to strengthen cryptographic security. Because the compressed message has less redundancy than the original plaintext, cryptanalysis is more difficult.



(a) Generic Transmission Diagram (from A)



(b) Generic Reception Diagram (to B)

v) E-mail Compatibility

When PGP is used, at least part of the block to be transmitted is encrypted. PGP provides the service of converting the raw 8-bit binary stream to a stream of printable ASCII. For this PGP using radix- 64 conversion. Each group of 3 octets of binary data is mapped in to 4 ASCII characters. The uses of radix 64 expands a message by 33%

(vi) Segmentation and reassembly

Any message longer than that must be broken up into smaller segments, each of which is mailed separately. At the receiving end, the PGP must strip off all e-mail header and retrieve the essential block. The header is separated from the body by a blank line, a header line consists of a keyword, followed by a colon followed by keyword's arguments.

Date:

From:

To:

Subjec:

PGP SESSION KEYS

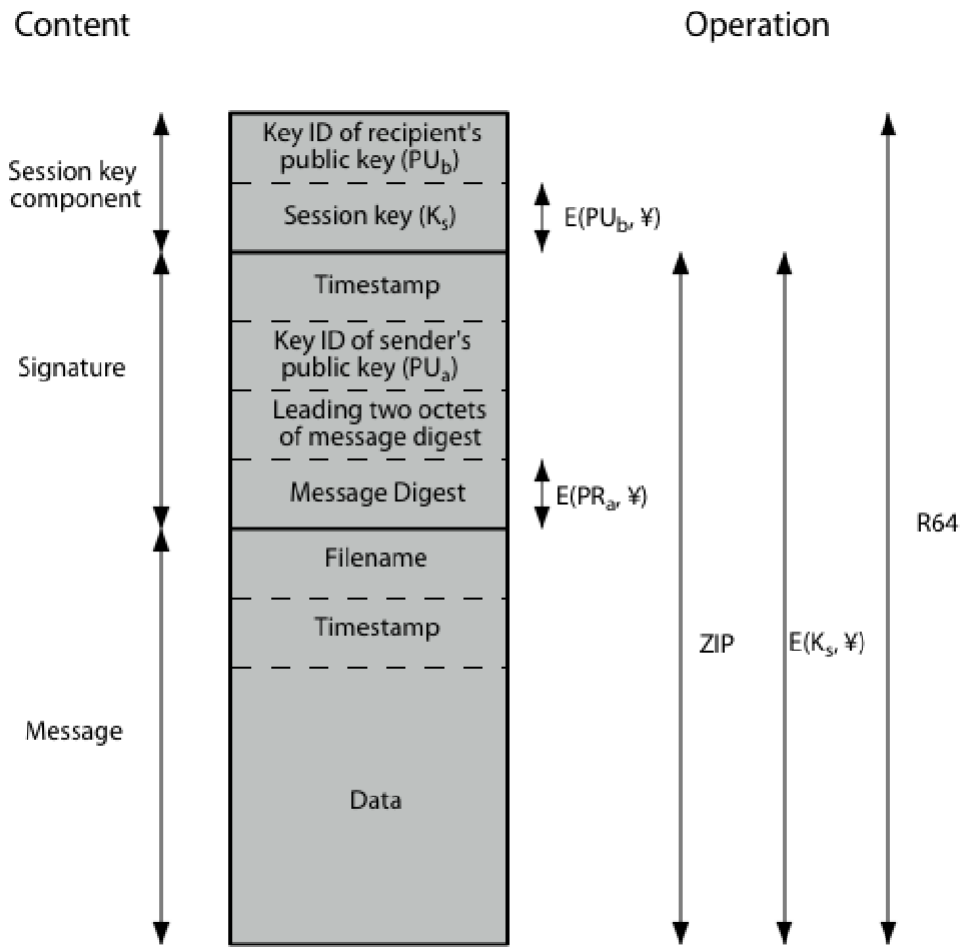
PGP makes use of four types of keys: one-time session symmetric keys, public keys, private keys, and passphrase-based symmetric keys.

Each session key is associated with a single message and is used only for the purpose of encrypting and decrypting that message, using a symmetric encryption algorithm, such as CAST-128 and IDEA with 128-bit keys; or 3DES with a 168-bit key.

PGP Public & Private Keys

- since many public/private keys may be in use, use a key identifier based on key

PGP Message Format



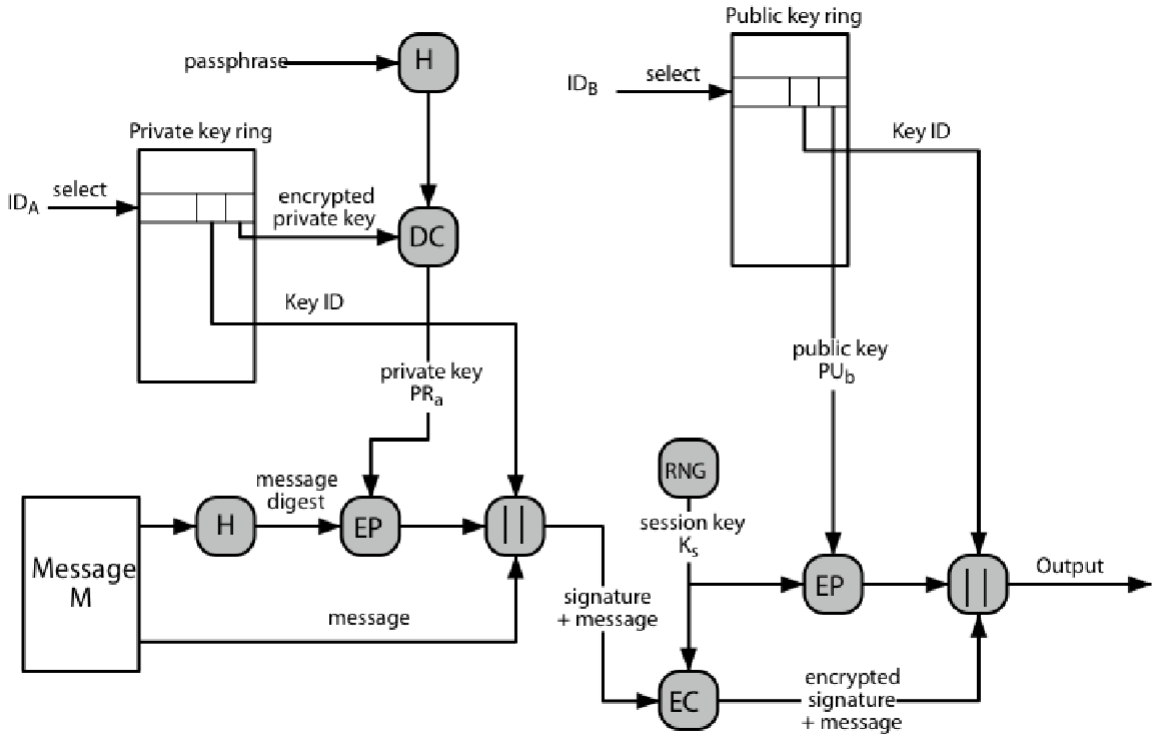
A message consists of three components:

- the message component,

- a signature (optional),
- and a session key component (optional).

The message component includes the actual data to be stored or transmitted, as well as a filename and a timestamp that specifies the time of creation. The signature component includes a timestamp, encrypted SHA-1 message digest, leading two digest octets for verification, and the Key ID of the sender's public key. The session key component includes the session key and the identifier of the recipient's public key that was used by the sender to encrypt the session key. The entire block is usually encoded with radix-64 encoding.

PGP Message Generation



The sending PGP entity performs the following steps:

1. Signing the message:
   a. PGP retrieves the sender's private key from the private-key ring
   b. PGP prompts the user for the passphrase to recover the unencrypted private key.
   c. The signature component of the message is constructed.
2. Encrypting the message:

a. PGP generates a session key and encrypts the message.

b. PGP retrieves the recipient's public key from the public-key ring

c. The session key component of the message is constructed.

4.2 S/MIME

S/MIME is a security enhancement to MIME. S/MIME will emerge as the industry standard for commercial and organizational use.

To understand the S/MIME, we need first to have a general understanding of the e-mail format RFC822.

RFC822

RFC 822 defines a format for text messages that are sent using e-mail. In RFC 822 messages are said to have an envelope and contents.

Envelop: Information needed for transmission and delivery is present.

Content: It contains the object to be delivered to the receiver.

Each line in the header consists of a keyword such as ***From, To, Subject, Date.***

The following are the limitations of SMTP/RFC 82 scheme.

- SMTP cannot transmit executable or other binary data.
- SMTP cannot transmit text data that includes natural language characters.
- SMTP server may reject mail message over a certain type

Overview of MIME

1. Five new message header fields are defined, which may be included in an RFC 822 header.

2. A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.

3. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

The five header fields defined in MIME are as follows

MIME-Version: This field must have a parameters value of 1.

Content-Type: This deals with the definition of variety of content types. In general content type specifies the type of data.

Content-Transfer-Encoding: Indicates the type of transformation that has been used to represent the body of the message.

Content-ID: Used to identify MIME entities.

Content-Description: A text description of the object within the body.

Content-Type

There are seven different major types of content and total of 15 subtypes.

Type	Subtype	Description
Text	Plain	Unformatted text;
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted
	Parallel	The multiple parts can be presented in parallel.
	Alternative	The different representation of the same
	Digest	Similar to Mixed, but the default type/subtype of
Message	rfc822	The body is itself an encapsulated message.
	Partial	Used to allow fragmentation of large mail items.
	External-	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN.
Applicatio	PostScript	Adobe Postscript.
	octet-stream	General binary data consisting of 8-bit bytes.

MIME Transfer Encodings

The objective is to provide reliable delivery across a largest range of environments.

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters.
binary	The lines are not necessarily short enough and non-ASCII
quoted-	Data being encoded are mostly ASCII text.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks
x-token	A named nonstandard encoding.

S/MIME functionality

S/MIME provides the following functions

Enveloped Data: This consists of encrypted content of any type and encrypted content encryption keys for one or more recipients.

Signed Data: A digital signature is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded
using base 64 encoding.

Clear-signed Data: The digital signature is encoded using base64. As a result recipients without
S/MIME capability can view the message content, although they cannot verify the signature.

Signed and enveloped data: Signed only and encrypted only entities may be nested, so that encrypted data may be signed.

S/MIME messages

S/MIME secures a MIME entity with a signature, encryption or both.

Type	**Subtype**	**smime**	**Description**
Multipart	Signed		A clear-signed message in two parts: one is the message and the other is the
Application	pkcs 7-mime	signedData	A signed S/MIME entity.
	pkcs 7-mime	envelopedDat	An encrypted S/MIME entity
	pkcs 7-mime	degen	s An entity containing only public- key
	pkcs 7-mime	Compressed	A compressed S/MIME
	pkcs 7-	signedData	The content type of the signature subpart

Content Type of S/MIME

Enveloped Data

The steps for preparing an enveloped Data are as follows

1. Generate a pseudo-random session key for a particular symmetric encryption algorithm.
2. For each recipient, encrypt the session key with the recipient's public RSA key.

3. For each recipient, prepare a block known as RecipientInfo that contains the sender's public- key certificate, an identifier for the algorithm used to encrypt the session key, and the encrypted session key.
4. Encrypt the message content with the session key.

SignedData

The steps for preparing a signedData MIME entity are as follows:

1. Select a message digest algorithm.
2. Compute the message digest, or hash function, of the content to be signed.
3. Encrypt the message digest with the signer's private key.
4. Prepare a block known as SignerInfo that contains the signer's public-key certificate, an identifierof the message digest algorithm, an identifier of the algorithm used to encrypt the message digest, and the encrypted message digest.

S/MIME Certificate Processing

S/MIME user has several key-management functions to perform:

Key generation: Must be capable of generating separate Diffie-Hellman and DSS key pairs and should be capable of generating RSA key pairs. Each key pair must be generated from a good source of nondeterministic random input and be protected in a secure fashion.

Registration: A user's public key must be registered with a certification authority in order to receive an X.509 public-key certificate.

Certificate storage and retrieval: The list of certificates could be maintained by the user or by some local administrative entity on behalf of a number of users.

VeriSign Certificates

VeriSign provides a service that is intended to be compatible with S/MIME and a variety of other applications. VeriSign issues X.509 certificates with the product name VeriSign Digital ID. Each digital ID contains

- Owner's public key
- Owner's name or alias
- Expiration date of the Digital ID

- Serial number of the Digital ID
- Name of the certification authority that issued the Digital ID

Enhanced Security Services

- Signed Recepits
- Security Labels
- Secure Mailing Lists

S/MIME Cryptographic Algorithms

- Hash functions: SHA-1 & MD5
- Digital signatures: DSS & RSA
- Session key encryption: ElGamal & RSA
- Message encryption: Triple-DES, RC2/40 and others

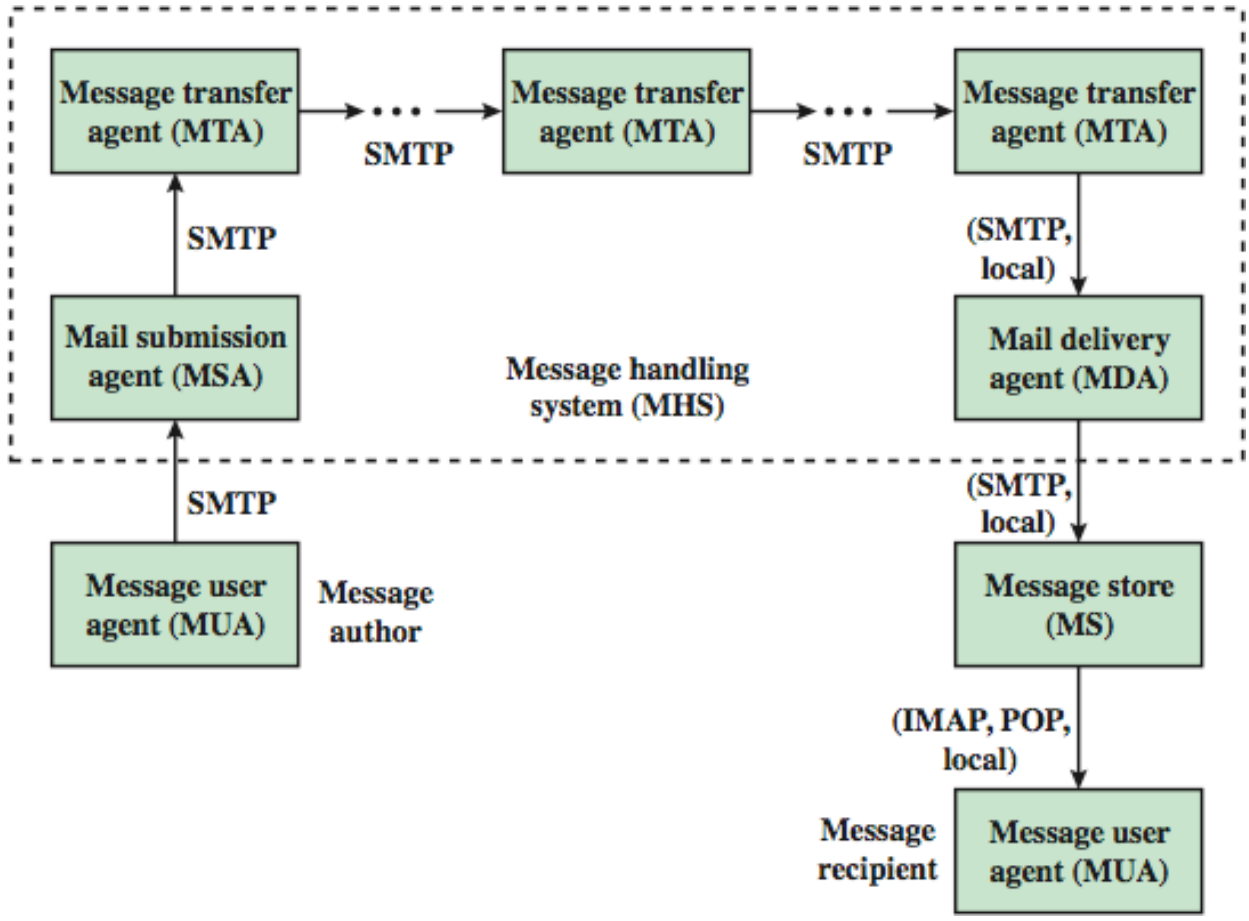
4.3 Domain Keys Identified Mail

- Domain Keys Identified Mail (DKIM) is a specification for cryptographically signing email messages, permitting a signing domain to claim responsibility for a message in the mail stream.

- Message recipients (or agents acting in their behalf) can verify the signature by querying the signer's domain directly to retrieve the appropriate public key, and thereby confirm that the message was attested to by a party in possession of the private key for the signing domain.

- DKIM is a proposed Internet Standard (RFC 4871: DomainKeys Identified Mail (DKIM) Signatures). DKIM has been widely adopted by a range of email providers, including corporations, government agencies, gmail, yahoo, and many Internet Service Providers (ISPs).

Internet Mail Architecture



- To understand to operation of DKIM, it is useful to have a basic grasp of the Internet mail architecture.
- At its most fundamental level, the Internet mail architecture consists of a user world in the form of Message User Agents (MUA), and the transfer world, in the form of the Message Handling Service (MHS), which is composed of Message Transfer Agents (MTA).
- A MUA is usually housed in the user's computer, and referred to as a client email program, or on a local network email server.
- The MHS accepts a message from one User and delivers it to one or more other users, creating a virtual MUA-to-MUA exchange environment.
- The MSA accepts the message submitted by an MUA and enforces the policies of the hosting domain and the requirements of Internet standards.

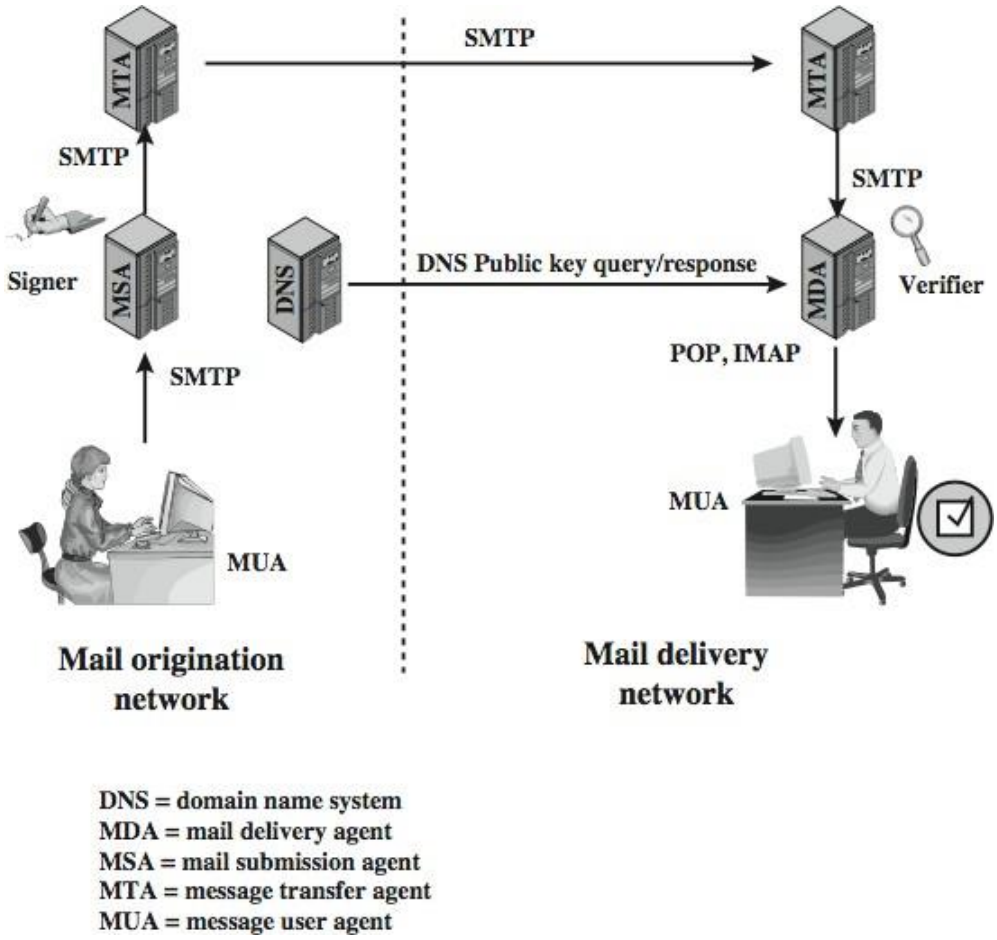
- This function may be co-located with the MUA or be a separate functional model.
- In the latter case, the Simple Mail Transfer Protocol (SMTP) is used between the MUA and the MSA.
- The MTA relays mail for one application-level hop. Relaying is performed by a sequence of MTAs, until the message reaches a destination MDA.
- The MDA is responsible for transferring the message from the MHS to the MS. An MS can be located on a remote server or on the same machine as the MUA.
- Typically, an MUA retrieves messages from a remote server using POP (Post Office Protocol) or IMAP (Internet Message Access Protocol).
- Also an administrative management domain (ADMD) is an Internet email provider.
- The Domain Name System (DNS) is a directory lookup service that provides a mapping between the name of a host on the Internet and its numerical address.

DKIM Strategy

- DKIM is designed to provide an email authentication technique transparent to the end user.
- In essence, a user's email message is signed by a private key of the administrative domain from which the email originates.
- The signature covers all of the content of the message and some of the RFC 5322 message headers.
- At the receiving end, the MDA can access the corresponding public key via a DNS and verify the signature, thus authenticating that the message comes from the claimed administrative domain.
- Thus, mail that originates from somewhere else but claims to come from a given domain will not pass the authentication test and can be rejected.
- This approach differs from that of S/MIME and PGP, which use the originator's private key to sign the content of the message, for various pragmatic reasons

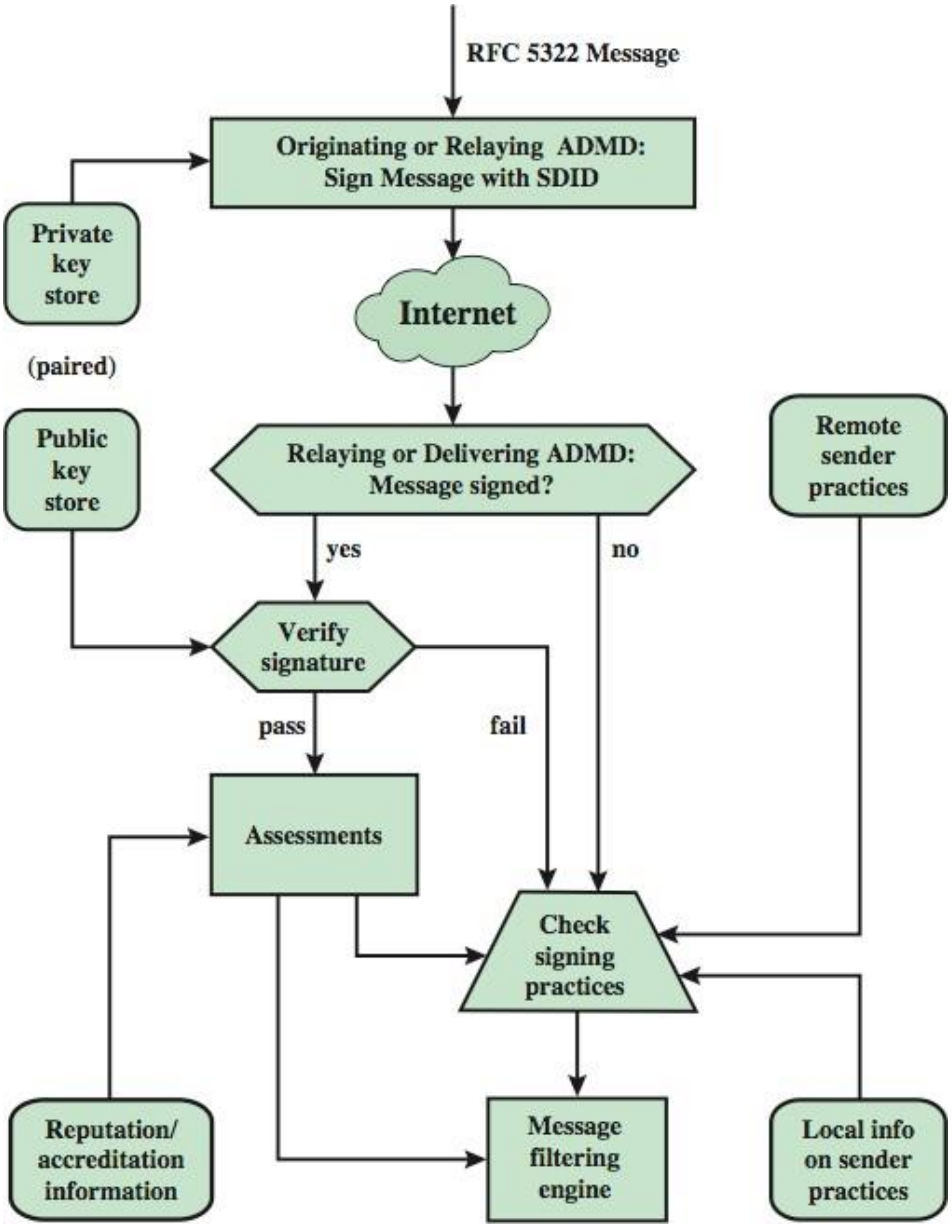
Figure shows a simple example of the operation of DKIM. An email message is generated by an email client program.

- The content of the message, plus selected RFC 5322 headers, is signed by the email provider using the provider's private key.
- The signer is associated with a domain, which could be a corporate local network, an ISP, or a public email facility such as gmail.
- The signed message then passes through the Internet via a sequence of MTAs. At the destination, the MDA retrieves the public key for the incoming signature and verifies the signature before passing the message on to the destination email client.
- The default signing algorithm is RSA with SHA-256. RSA with SHA-1 may also be used.



- In essence, a user's e-mail message is signed by a private key of the administrative domain from which the e-mail originates.
- At the receiving end, the MDA can access the corresponding public key via a DNS and verify the signature, thus authenticating that the message comes from the claimed administrative domain.
- Thus, mail that originates from somewhere else but claims to come from a given domain will not pass the authentication test and can be rejected.

DCIM Functional Flow



- Figure provides a more detailed look at the elements of DKIM operation. Basic message processing is divided between a signing Administrative Management Domain (ADMD) and a verifying ADMD.
- Signing is performed by an authorized module within the signing ADMD and uses private information from a Key Store.
- Verifying is performed by an authorized module within the verifying ADMD. The module verifies the signature or determines whether a particular signature was required.
- Verifying the signature uses public information from the Key Store. If the signature passes, reputation information is used to assess the signer and that information is passed to the message filtering system.
- If the signature fails or there is no signature using the author's domain, information about signing practices related to the author can be retrieved remotely and/or locally, and that information is passed to the message filtering system.
- The signature is inserted into the RFC 5322 message as an additional header entry, starting with the keyword DKIM-Signature.
- Before a message is signed, a process known as canonicalization is performed on both the header and body of the RFC 5322 message.
- Canonicalization is necessary to deal with the possibility of minor changes in the message made route. The signature includes a number of fields, as listed in the text.

4.4 Wireless Network Security

Wireless Security

Some of the key factors contributing to higher security risk of wireless networks include

- Channel
- Mobility

- Resources
- Accessibility

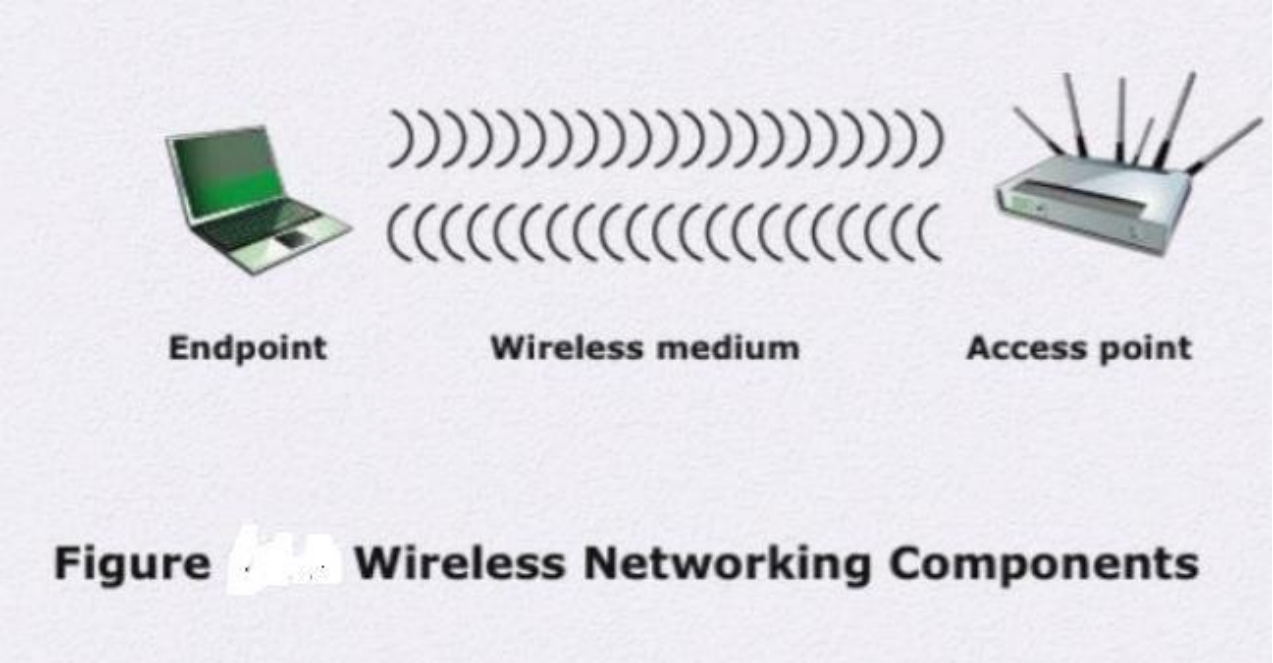


Figure Wireless Networking Components

Wireless Network Threats

- Accidental association
- Malicious Association
- Man in the middle attack
- DoS
- Identity theft

SECURITY MEASURES

Securing Wireless transmissions from Eavesdropping

- Signal hiding techniques
- Encryption

Securing Wireless Access Points

- Unauthorized Access to the network

Solutions

IEEE 802.1x standard for port based network access control

Securing wireless networks

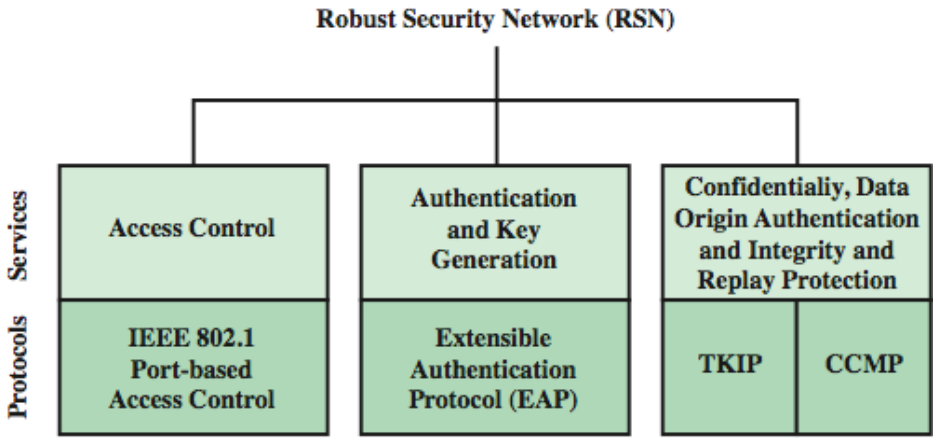
- Use encryption

- Use antivirus, antispyware software and firewall
- Turnoff Identifier broadcasting
- Allow only specific computers to access your wireless netw

IEEE 802.11i

- The differences between wired and wireless LANs suggest the increased need for robust security services and mechanisms for wireless LANs.
- For privacy, 802.11 defined the **Wired Equivalent Privacy (WEP)** algorithm. The privacy portion of the 802.11 standard contained major weaknesses. In order to accelerate the introduction of strong security into WLANs, the Wi-Fi Alliance circulated **Wi-Fi Protected Access (WPA)** as a Wi-Fi standard. The final form of the 802.11i standard is referred to as **Robust Security Network** (RSN). The Wi-Fi Alliance certifies vendors in compliance with the full 802.11i specification under the WPA 2 program.

802.11i RSN Services and Protocols



The 802.11i RSN security specification defines the following services:

- Authentication
- Access control
- Privacy with message integrity

802.11i Phases of Operation

Figure lists the cryptographic algorithms used for the 802.11i RSN security services.

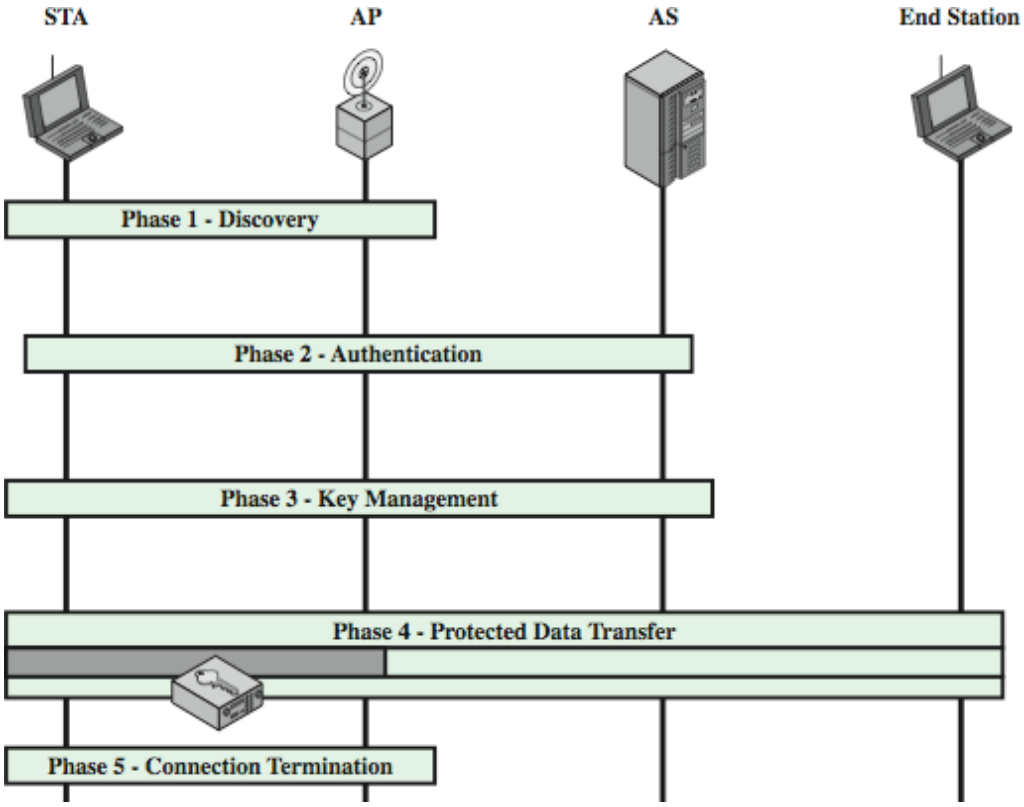
The operation of an IEEE 802.11i RSN can be broken down into five distinct phases of operation, as shown in Figure.

One new component is the authentication server (AS).

The five phase are:

• Discovery: An AP uses messages called Beacons and Probe Responses to advertise its IEEE 802.11i security policy. The STA uses these to identify an AP for a WLAN with which it wishes to communicate. The STA associates with the AP, which it uses to select the cipher suite and authentication mechanism when the Beacons and Probe Responses present a choice.

• Authentication: During this phase, the STA and AS prove their identities to each other. The AP blocks non-authentication traffic between the STA and AS until the authentication transaction is successful. The AP does not participate in the authentication transaction other than forwarding traffic between the STA and AS.



• Key generation and distribution: The AP and the STA perform several operations that cause cryptographic keys to be generated and placed on the AP and the STA. Frames are exchanged between the AP and STA only

• Protected data transfer: Frames are exchanged between the STA and the end station through the AP. As denoted by the shading and the encryption module icon, secure data transfer occurs between the STA and the AP only; security is not provided end-to-end.

• Connection termination: The AP and STA exchange frames. During this phase, the secure connection is torn down and the connection is restored to the original state.

802.11i Discovery and Authentication Phases

The discovery phase consists of three exchanges: Network and security capability discovery, Open system authentication, and Association.

802.11i Key Management Phase

This exchange is known as the 4-way handshake. The STA and AP use this handshake to confirm the existence of the PMK, verify the selection of the cipher suite, and derive a fresh PTK for the following data session. For group key distribution, the AP generates a GTK and distributes it to each STA in a multicast group.

802.11i Protected Data Transfer Phase

IEEE 802.11i defines two schemes for protecting 802.11 MPDU data message integrity and confidentiality:

- Temporal Key Integrity Protocol (TKIP)
- Counter Mode-CBC MAC Protocol (CCMP).

WAP (Wireless Application Protocol)

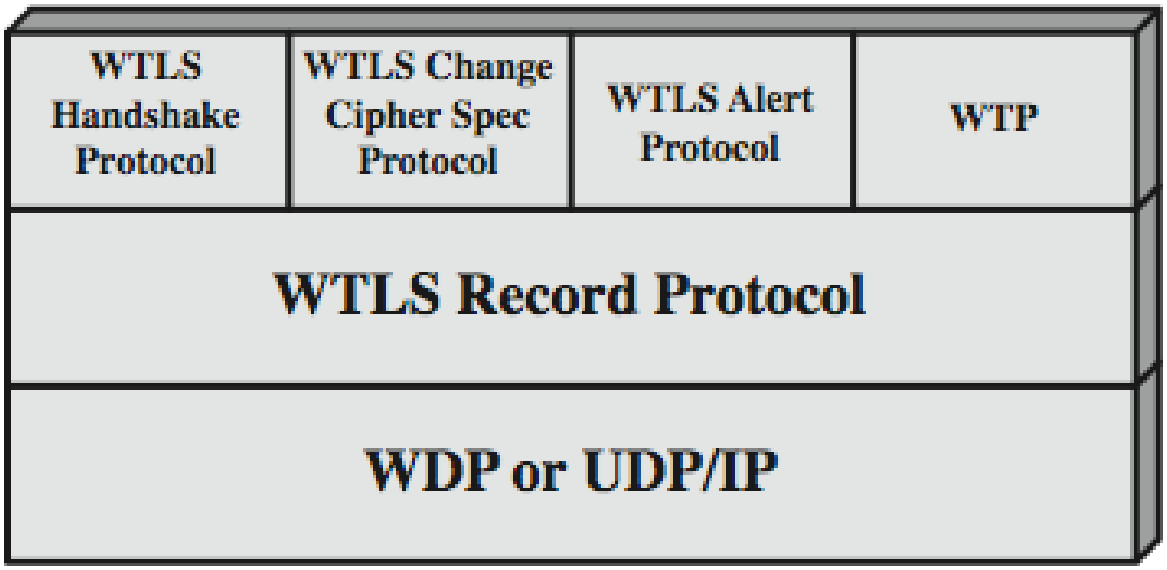
- a universal, open standard developed to provide mobile wireless users access to telephony and information services

Wireless Transport Layer Security (WTLS)

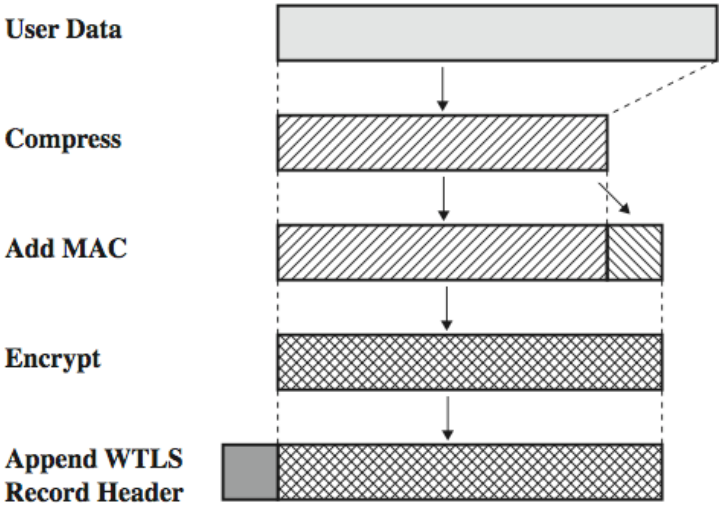
- provides security services between mobile device (client) and WAP gateway
    - provides data integrity, privacy, authentication, denial-of-service protection

- based on TLS
    - more efficient with fewer message exchanges
    - use WTLS between the client and gateway
    - use TLS between gateway and target server
- WAP gateway translates WTLS / TLS

WTLS Protocol Architecture



WTLS Record Protocol



4.5 Mobile Device Security

- Mobile devices have become an essential element for organizations as part of the overall network infrastructure
- Prior to the widespread use of smartphones, network security was based upon clearly defined perimeters that separated trusted internal networks from the untrusted Internet
- Due to massive changes, an organization's networks must now accommodate:
    - Growing use of new devices
    - Cloud-based applications
    - De-perimeterization
    - External business requirements

Security Threats

- Lack of physical security controls
- Use of untrusted mobile devices
- Use of untrusted Networks
- Use of untrusted Content
- Use of applications created by unknown parties
- Interaction with other systems
- Use of location services

MOBILE DEVICE SECURITY ELEMENTS

- Mobile device Configuration Server
    - Mobile device is configured with security mechanisms and parameters to confirm to organization security policy
- Application/ DB server
- Authentication/ Access Control Server
    - Used to verify device and user and establish limits on access
- Firewall

Mobile traffic is encrypted using SSL or IPSEC

Mobile Device Security Elements