Rifana

16/11/18
Friday

Module - 6

Web Security

E-mail Security

Security Services for E-mail:

The following are desirable security services that an e-m system might have:

1. Privacy - The ability to keep anyone but the intended rec pient from reading the msg.
2. Authentication - Assurrance to the recipient about the iden of the center.
3. Integrity - Assurrance that the msg has not been modif
4. Non-repudiation - The ability of the recipient to prove to third party that the center really sent the msg.
5. Proof of Submission - Verification to the center that the m was handed over to the mail delivery s/m.
6. Proof of Delivery - Verification that the recipient received msg.
7. Anonymity - The ability to send a msg so that the recip ent cannot find out the identity of the center.
8. Containment - The ability to keep the information from leaking out of a particular region.

9. Audit - Ability to record events having a security relevance.

10. Accounting - Ability to keep s/m usage statistics.

11. Self-destruct - The possibility that allows Alice to send a msg to Bob which Bob cannot forward or stored.

12. Message Sequence Integrity - Assurance that the entire sequence of msgs arrived in the order transmitted.

Establishing Keys

Security services are provided by cryptography. Cryptography requires keys.

(a) Establishing public key

Suppose Alice wants to obtain Bob's public key. The traditional methods of Alice visiting Bob personally or Alice telephoning Bob to obtain the key are inconvenient and risky.

The following practice is often employed:

Step 1: Alice first gets the public key of atleast one entity she trust.

Step 2: She obtains the other key which are signed by with the entity she trust with the msg like "x's public key y" such a signed msg is called a certificate

Step 3: After collecting several such certificates, Alice will have a chain of certificates starting from the entity she trust and ending with a certificate which certifies Bob's key.

The software package Pretty Good Privacy (PGP) provides such a service.

(b) Establishing Secret Keys

Secret key is the key used for encryption and decryption in symmetric key cryptography.

The services of a Key Distribution Centre (KDC) are used to establish a shared secret key.

Privacy

Assume that Alice wants to send a msg to Bob which only Bob can read. Alice may encrypt the msg using Bob's public key or by using a secret key Alice shares with Bob. This method has the following problem:

1. Let the msg be very long.

   Assume that the msg has to be send to be many persons in such a case msg has to be encrypted as many times as the no. of recipients.

2. Public key encryption is very inefficient and time consuming

In practice the following procedure is followed:

(1) Alice chooses a random secret key 's'.
(2) Alice encrypt the msg with 's'.
(3) Alice encrypt s with Bob's key.
(4) Alice sends both quantities to Bob.

In this sch, the msg need be encrypted only once with the key 'S'. But S has to be encrypted as many times as the no. of recipients.

Authentication

Suppose Alice sends a msg to Bob. To authenticate the msg, the following procedure is adopted:

(1) Alice computes the hash value of the msg using some cryptographic hash function.

(2) Alice signs the hash value with her private key.

(3) Alice sends the msg together with the signature to Bob.

Integrity

Methods for authentication can also be used to prove the integrity of msgs.

Non-repudiation

Repudiation is the act of denying that Alice send the msg. If the msg sch provides non-repudiation it means that if Alice sents a msg to Bob, Alice cannot later deny that she sent the msg.

The usual authentication method which includes the private key of the center provides non-repudiation also. Only someone with knowledge of the secret key could have signed the hash value. Anyone knowing the public key can verify the signature

19/11/18
Monday
4mp

Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) is an encryption program that provides cryptographic privacy and authentication for data communication. PGP is used for signing, encrypting & decrypting texts, e-mails, files, folders and whole desk partitions.

Phil Zimmerman developed PGP in 1991.

Features

1. PGP is available free world wide web and runs on a variety of platforms.
2. It is based on algorithms that are connected extreme secure like RSA, DSA, Diffie-Hellman key exchange, 3-DES, SHA-256.
3. It has a wide range of applicability: from corporation to individuals.
4. It is not controlled by any government organisation

Services

The operations of PGP consists of the following services:

1. Authentication
2. Confidentiality
3. Compression
4. E-mail compactability

Authentication

Procedure:

a) Sender creates a msg.

b) SHA-1 is used to generate a hash value of the msg.

c) Hash value is encrypted with RSA using senders private key.

d) The result is attached to the msg.

e) Receiver uses RSA with senders public key to decrypt the msg and recover the hash value.

f) Receiver compares the received hash value with the computed hash value. If they match, the msg is accepted as authenticated.

Confidentality

a) Sender generates a msg and a random number has the session key.

b) Msg is encrypted with the session key.

c) Session key is encrypted with RSA using receipients public key and is attached to the msg.

d) Receipient uses the private key to decrypt and recover the session key.

e) The session key is used to decrypt the msg.

Compression

As default PGP compresses the msg after appending the nature but before encryption. PGP uses the ZIP algorithm to compress msgs.

E-mail Compactibility

The output of an encryption scheme may be a stream of bytes. But many e-mail sims support only ASCII characters. PGP has a service for converting an encrypted msg to a stream of ASCII characters using base-64 (radix-64) conversion.

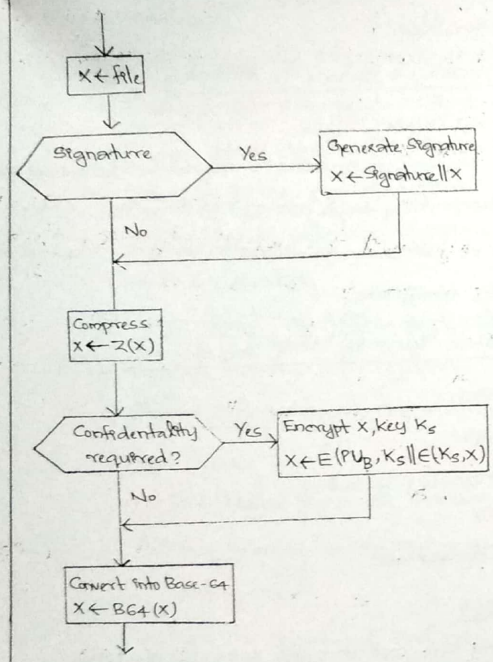
In the base-64 sim, the numbers 0 to 63 are represented by ASCII characters as follows:

0-25, 26-51, 52-61, 62, 63
A-Z, a-z, 0-9, +, /

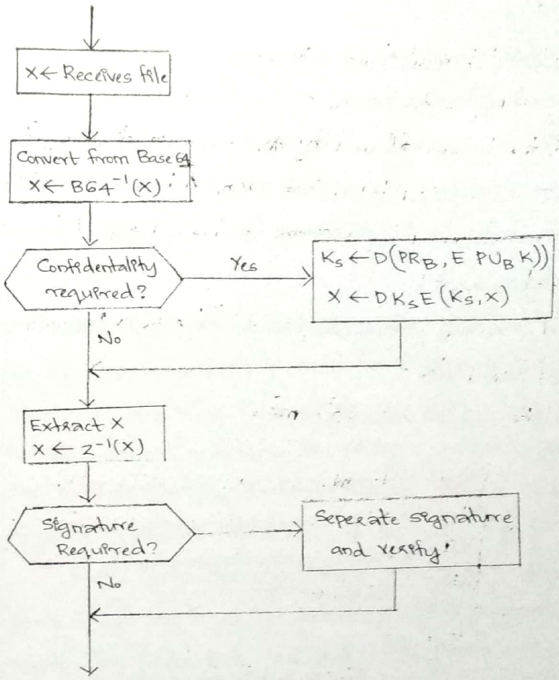
6-bits are required to represent a base-64 digit. 3 bytes can be represented by base-64 digit

byte 1	byte 2	byte 3	
******|	**|****	****|**	******
6 bits	6 bits	6 bits	6-bits
base-64 digits	base-64 digits	base-64 digits	base-64 digits

Transmission of PGP messages



Reception of PGP messages by B



22/11/18 Thursday

S/MIME

S/MIME stands for Secure/Multipurpose Internet Mail Extension.

S/MIME is a security enhancement for the MIME email format based on RSH data security.

MIME is an extension of SMTP (Simple Mail Transfer Protocol).

Traditional E-mail format Standard

According to this format a msg consists of a few lines, called the header followed by the body of the msg (the text). Both are separated by a bank line. A header consists of a keyword followed by a comma, followed by the key words values/argument.

The mostly commonly used keywords are "from", "to", "subject" and date.

Here is an example msg:-

Header
- Date : November 22, 2018 02.15.18 PM IST
- From : "Manju" <manjumheaven@gmail.com">
- To : principal@vidyaacademy.ac.in
- Subject – Example msg

Blank Line ——

Text
- Sir,
  Sorry, this is a test message.
- Your's Sincerely

MIME

The following are the limitations of SMTP:

1. SMTP cannot transmit binary files, image files, video files etc.
2. SMTP can transmit files containing only ASCII characters.
3. SMTP can transmit only msgs of a certain sizes. It reject larger files.

MIME specifications include the following:

1. Five new msg header fields.
2. Definitions of content formats to standardize representation for supporting email having multimedia content.
3. Definitions of transfer encodings to protect alterations of msgs by the email s/m.

The new header fields in MIME are the following:

1. MIME-version
2. Content-Type
3. Content-Transfer Encoding
4. Content-ID
5. Content-Description

Content-Types

The following are content types available in MIME.

Text : ASCII characters
Multipart : Contains different parts
Message : Some message
Image : JPeg or gif file
Video : MPEG file
Audio :
Application : General binary file

MIME transfer encodings are encodings in systems like base-64 encodings.

S/MIME provides the following functions:-

1) Enveloped data: encrypted content of any type

2) Signed data: signature is formed by taking a hash value of the msg and encrypting it using senders private key. The content + signature is encoded using base-64 encoding

3) Clear-signed data: Clear text with digital signature encoded in base-64 encoding

4) Signed and enveloped data: S/MIME uses the following algorithms.

(a) DSS (Digital Signature Standard) for digital signatures

(b) Diffie Hellman for key exchange algorithm for generating keys.

(c) RSA for signatures and encryption.

Summary of S/MIME

SMTP

Header fields: From, To, Date, Subject

Limitation: Only ASCII, On size, No binary files

MIME

Additional header fields

Content type → Audio, Video, Application, etc

Transfer encoding

MIME version ID

S/MIME

Enveloped

Signed data

Clear-signed data

Signed & enveloped data

- DSS
- Diffie Hellman
- RSA

IP Security

IPv4

Stands for Internet Protocol version 4. It is the 4th version of Internet Protocol; the protocol for transmitting packets across networks. IPv4 is a connectionless protocol for use in packet switched networks. This protocol doesnot guarantee delivery of msgs. It also doesnot assure proper sequencing of msgs or avoidance of duplicate delivery.

In a connectionless communication, a msg can be send from one end to another without prior arrangement. The device at one end transmits data addressed to other without first ensuring that the recipient is available & ready to receive data.

In connection oriented communication, the communicating parties must first exchange a data channel before the exchange of data.

Packet switching is the method of grouping data that is transmitted over a n/w into packets. A packet consists of a header and a payload. Data in the header is used to direct the packet to it's destination. Payload contains the content of the data.

Addressing:

* IPv4 uses 32 bit addresses
* Addresses are written in dotted-decimal-notation.
* Example: 10101100 00010000 11111110 00000001

8bits	8bits	8bits	8bits
172	16	254	1

Dotted decimal notation: 172.16.256.1

Header

IPv4 packet header consists of 13 fields of which 12 are required. The 13th field called options is "option".

Version Number	Header Length	Type of Service	Total Length	
Identification			Flags	Fragment Offset
Time-to-Live	Protocol	Header Checksum		
Source IP address				
Destination IP address				
Options				

- Header Length - size of the header (changes with the size of "options".
- Type of service - this is to be adopted for transmitting the packet.
- Total length - the size of the entire packet in octets.
- Flags - used to identify fragments.
- Fragment offset - the offset of a particular fragment relative to the beginning of the first fragment. The first fragment has offset 0.
- Time to live - this specifies how long a fragment should remain. After the expiry of this time the fragment destroys itself.
- Protocol - this specifies the protocol used in the data portion. It indicates what follows the IP header. For eg: If an Authentication Header (AH) follows the IP header, we assign the value 50 to the field "protocol".
- Header checksum - used for error checking the header.
- Options - This field is not commonly used.

IPv6

Stands for Internet Protocol version 6. It is the most recent version of Internet Protocol. It was developed by IETF (Internet Engineering Task Force) to deal with the problem of address exhaustion. It became an Internet Standard only on 14 July 2017.

IPv6 packet

A packet consists of a header containing information for addressing and routing & payload consisting of user data. IPv6 packets are transmitted over the link layer (link layer is the combination of layer 1 & 2 in the OSI model). Routers donot fragment IPv6 packets as they do in IPv4.

Addressing

An IPv6 address is 8 groups of 4 hexa decimal digits. Each hexa decimal digit is represented by 4 bits. The groups are separated by colons

Example:

ff00: 2001 : 0ab1 : fa23 : 80ab : 7cdf : 598e : 6cfa

The digits are case-insensitive. However, IETF recommends lower case letters.

Header format

Version	Traffic Class	Flow Label	32 bits
Payload Length	Next header	HOP Limit	32 bits
Source Address			32 x 4 = 128 bits
Destination Address			32 x 4 = 128 bits

Traffic class

This indicates the priority for transmitting packets. Eg; high priority streaming data

Flow Label

Field to indicate whether a packet should be relabeled

Next header

This is equivalent to the 'protocol field' in IPv4.

HOP Limit

This is replacement for 'Time to Live' field in IPv4. This value is decremented by 1 at each forwarding node packet is discarded when it becomes 0

IP Security

IPsec is an IETF standard for realtime communication security. This protocol authenticates and encrypts the packet of data send over an internet n/w. IPsec includes protocol for the following:

1) Establishing mutal authentication b/w agents
2) Negotiation of cryptographic keys
3) Protecting data flows b/w a pair of host, between a pair of security gateways and b/w a host and gateway.
4) Data origin authentication, data integrity, data confidentiality and replay protection.

The main components of IPsec are the following:

(i) Authentication header (AH)

(ii) Encapsulating security payload (ESP)

(iii) Internet key exchange.

Security Association

An IPsec Security Association (SA) is a cryptographically protected connection. Associated with each end of the security association is a key and other information such as the identity of the other end and cryptographic services being used.

Authentication Header (AH)

The AH header has the following format:

Field	No. of octets
Next values	1
Payload length	1
Unused	2
Security Parameter Index (SPI)	4
Sequence Number	4
Authentication Data	variable

Next header: Same as "protocol" field in IPv4.

Payload : : Size of the AH header

SPI : These parameters are used to identify the security associations.

Authentication data: Data for checking integrity.

Encapsulating Security Payload (ESP)

ESP allows encryption and integrity protection. If we use encryption we must use ESP. If we want integrity protect only we may use ESP or AH. Technically ESP always does encryption. If we don't want encryption we may use the special "Null encryption algorithm".

The presence of a ESP header is indicated by having the "protocol" or the "next header" field equal to 50.

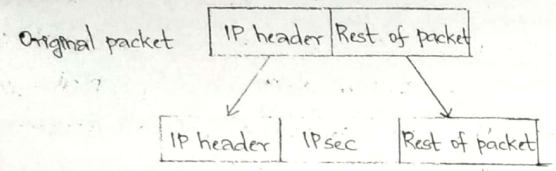
The various fields in ESP are as follows:

Field	No. of octets
SPI	4
Sequence Number	4
Initialization vector	variable
Data	"
Padding	"
Padding length	1
Next header	1
Authentication data	variable

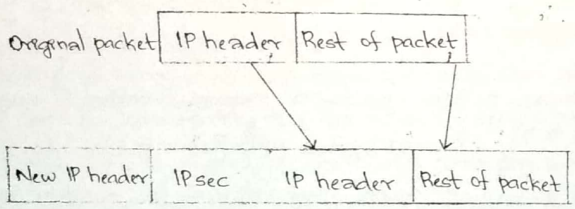
Tunnel, Transport Modes

In IPsec specification talk about two modes of applying the IPsec protection to a packet.

Transport Mode



Tunnel Mode

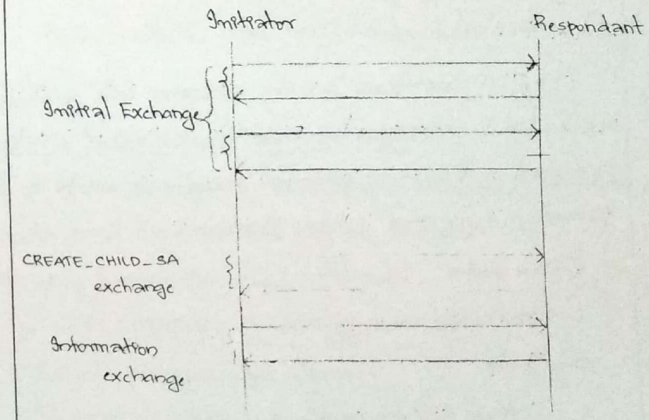


Internet Key Exchange(IKE)

The key management portion of IPsec involves the determination and distribution of secret keys. A common requirement is 4 keys for communication between two parties. There are special protocols for exchanging keys. IKEv2 is the second version of internet key exchanges protocol. The following are the important features of this protocol:

1. It is a refinement of the Diffie-Hellman key exchange algorithm.
2. It has a mechanism to prevent clogging attacks(clogging is spending too much time on computation of key
3. It enables two parties to setup a group of keys.
4. It has a mechanism to prevent replay attacks.
5. It authenticates the key exchange to prevent man-in-the-middle attack.

The procedure of key exchange involves exchange of messages in pairs.



1. The first two pairs of msgs, known as the initial exchange are used to exchange information regarding cryptographic algorithm and other security parameters. This exchange results in setting up a spcl security associations.
2. The next pair of msgs, known as CREATE_CHILD_SA msg, is used to establish further security associations for protecting traffic.
3. The final pair of msgs known as the information exchange, is used to exchange management information & notifications.

SSL/TLS

- SSL/TLS stands for Secure Socket Layer/Transport Layer Security.
- Basic Protocol

SSL/TLS partitions octects streams into records in to header and cryptographic protection, to provide a reliable, encrypted & integrity protected stream of octets to the application. There are 4 types of records.

1. User data
2. Handshake messages
3. Alerts
4. Change cipher specifications.

30/11/18
Friday

In the basic protocol, the client (Alice) initiates contact with the server (Bob). The contact is established by exchanging a sequence of message:

Message 1 (A to B): Alice without identifying herself says that she would like to talk to Bob. She gives a list of cryptographic algorithms she supports. She also gives a random number R_{Alice}.

Message 2 (B to A): Bob sends Alice his certificate and a random no R_{Bob} and he also chooses an algorithm.

Message 3 (A to B): Alice chooses a random number S and sends to Bob encrypted with Bob's public key. She also sends the hash value of the master secret key K.

Message 4 (B to A): Bob proves that he knows the key by sending the hash values of all the messages encrypted with appropriate keys.

Computing the Keys

The random number S in message 3 is called the pre-master key. The master secret key K is a function of S, R_{Alice} and R_{Bob}.

Client Authentication

In the basic protocol, Alice (client) authenticates Bob (server). But Bob doesnot know to whom he is talking.

SSL/TLS protocols can be used for client authentication also. This is done by having Bob send a "certificate request" in message 2. Alice, when she sees the request, sends her certificate and signature of the previous messages.

Public Key Infrastructure (PKI):

The PKI as deployed today, the client comes configured with public keys of various trusted organisations. The user at the client machine can modify this list. When the server sends a certificate, if it is signed by one of the certifying authorities in the client list, the client accepts the certificate. If the server sends the certificate signed by someone not on the list, the user is presented with a pop up window informing that the certificate could not be verified.

Attacks fixed by SSLv3

(1) In SSLv2, there was no integrity protection for initial messages. So an attacker could remove the list of cryptographic algorithms and replace it with weaker cryptographic algorithms. This is known as the "downgrade attack". This was fixed by adding a spcl msg to the end of initial message, in which each side sends a hash value of the messages in the exchange.

(2) In SSLv2, an attacker would send a premature closing message. This is known as the "truncation attack". In SSLv3, introduce a method to fix this attack.

Exportability

This refers to the exportability of SSL protocols. They are related to government regulations regarding restriction of export of cryptographic algorithms. For ex: the US government limited exportable cryptographic keys to 40 bit. But SSLv2 supported 128 bit keys.

Some methods have been developed to remove these restrictions.

Secure Electronic Transactions (SET)

SET is a security specification designed to protect credit card transactions on the Internet. It is a set of security protocols & formats.

SET provides 3 services:

1. Secure Communication Channel
2. Trust using digital signatures.
3. Privacy.

The SET protocols are designed to meet the following requirements.

1. Provide confidentiality of payment and ordering information.
2. Ensure integrity of all transmitted data.
3. Provide authentication that a card holder is a legitimate user of a credit card.
4. Provide authentication that a merchant can accept credit card transaction through it's relationship with a financial institution.
5. Ensure the use of best security practices & s/m design techniques.
6. Create a protocol that neither depends on transport security mechanisms nor prevents their use.
7. Facilitate and encourage interoperability among s/w. h/w providers.

SET participants are:

1. Card holder
2. Merchant
3. Issuer
4. Acquirer
5. Payment gateway
6. Certification authority.

Events required for a transaction:

1. Customer opens an account.
2. Customer receives a certificate.
3. Merchants acquire certificate.
4. Customer places an order.
5. The merchant is verified.
6. The order and the payment are send.
7. The merchant request payment authorization.
8. Merchant confirms the order.
9. Merchant provides goods or service.
10. Merchant requests payment.