UNIT-4

***Email Privacy**: Pretty Good Privacy (PGP) and S/MIME. **IP Security:** IP Security Overview,IP Security Architecture, Authentication Header, Encapsulating Security Payload,CombiningSecurityAssociations andKeyManagement.*

PRETTYGOODPRIVACY

In virtually all distributed environments, electronic mail is the most heavily usednetwork-basedapplication.Butcurrentemailservicesareroughlylike"postcards",anyone who wants could pick it up and have a look as it's in transit or sitting in therecipients mailbox. PGP provides a confidentiality and authentication service that can beused forelectronicmailand filestorageapplications.Withtheexplosively growingreliance on electronic mail for every conceivable purpose, there grows a demand forauthentication and confidentiality services. The Pretty Good Privacy (PGP) secure emailprogram, is a remarkable phenomenon, has grown explosively and is now widely used.Largely the effort of a single person, Phil Zimmermann, who selected the best availablecryptoalgorithmstouse&integratedthemintoasingleprogram,PGPprovidesaconfidentiality and authentication service that can be used for electronic mail and filestorage applications. It is independent of government organizations and runs on a widerangeofsystems,inbothfree&commercialversions.*Thereare**five**importantservicesinPGP*

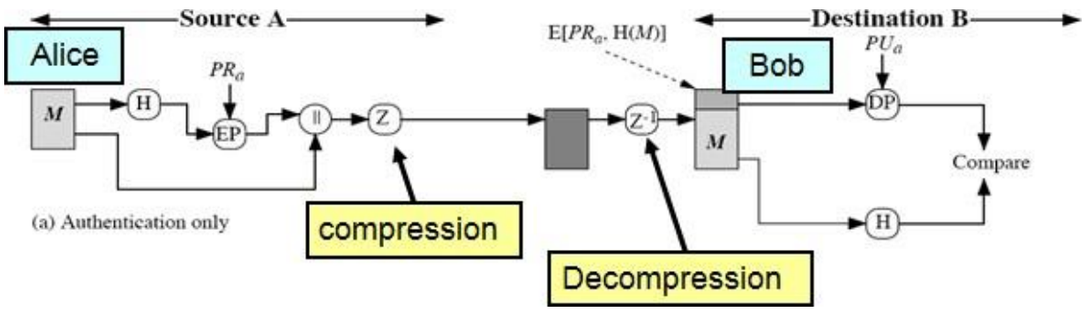
- *Authentication (Sign/Verify)Confidentiality*
- *(Encryption/Decryption)Compression*
- *Email*
- *compatibilitySegmentationandRe*
- *assembly*

Thelastthreeare**transparent** totheuser

PGPNotations:

Ks	=session key used in symmetric encryption scheme
PRa	=privatekeyofuserA,usedinpublic-key encryption scheme
PUa	=publickeyofuserA,usedinpublic-key encryption scheme
EP	=public-keyencryption
DP	=public-keydecryption
EC	=symmetricencryption
DC	=symmetric decryption
H	=hashfunction
||	=concatenation
Z	= compression using ZIP algorithm
R64	=conversion to radix 64 ASCIIformat

PGPOperation-Authentication



1. sendercreatesmessage

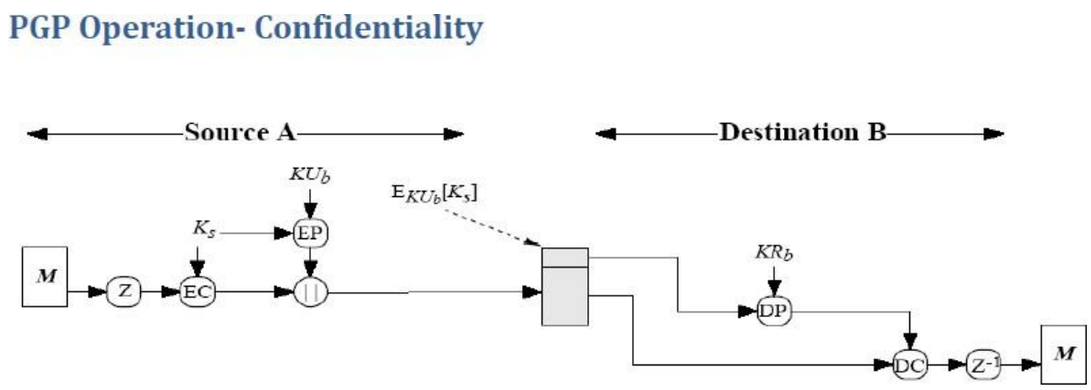
2. useSHA-1togenerate160-bithashofmessage

3. signedhashwithRSA usingsender'sprivatekey,andisattachedtomessage

4. receiverusesRSAwithsender'spublickeytodecryptandrecoverhashcode

5. receiver verifies received message using hash of it and compares with decrypted hashcode

PGPOperation-Confidentiality



Sender:

1. Generatesmessageandarandomnumber(sessionkey)onlyforthis message

2. EncryptsmessagewiththesessionkeyusingAES,3DES,IDEAorCAST-128

3. Encryptssessionkeyitselfwithrecipient'spublickeyusingRSA

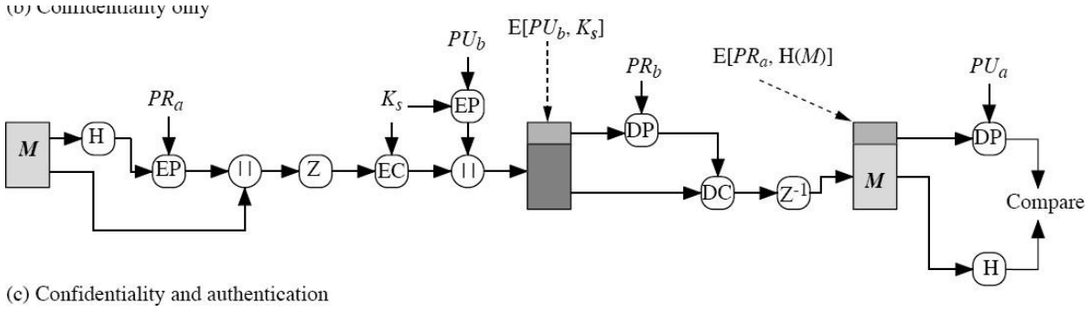
4. Attachesittomessage

Receiver:

1. Recoverssessionkeybydecryptingusinghis privatekey

2. DecryptsmessageusingthesessionkeyConfidentiality service provides no assurance to the receiver as to the identity of sender(i.e. no authentication). Only provides confidentiality for sender that only the recipientcanreadthemessage (andnoone else)

PGPOperation–Confidentiality&Authentication



(c) Confidentiality and authentication

can use both services on same message o create signature & attach to messageoencryptboth message & signature

oattachRSA/ElGamalencryptedsessionkey

o*iscalled**authenticatedconfidentiality***

PGPOperation–Compression

Asadefault,PGPcompressesthemessageafterapplyingthesignaturebutbeforeencryption. This has the benefit of saving space both for e-mail transmission and for filestorage.TheplacementofthecompressionalgorithM,indicatedbyZforcompressionandZ-1fordecompressionis critical. The compression algorithmusedisZIP.

ThesignatureisgeneratedbeforecompressionfortworeasonS:

1. sothatonecanstoreonlytheuncompressedmessagetogetherwithsignatureforlaterverificati on

2. Applying the hash function and signature after compression would constrain all PGPimplementationstothesameversionofthecompressionalgorithmasthePGPcompressional gorithmis not deterministic

Messageencryptionisappliedaftercompressiontostrengthencryptographicsecurity.Becau sethecompressedmessagehaslessredundancythantheoriginalplaintext, cryptanalysis is more difficult.

PGPOperation–EmailCompatibility

When PGP is used, at least part of the block to be transmitted is encrypted, and thusconsistsofastreamofarbitrary8-bitoctets.HowevermanyelectronicmailsystemsonlypermittheuseofASCIItext.Toaccommodat ethisrestriction,PGPprovidestheservice

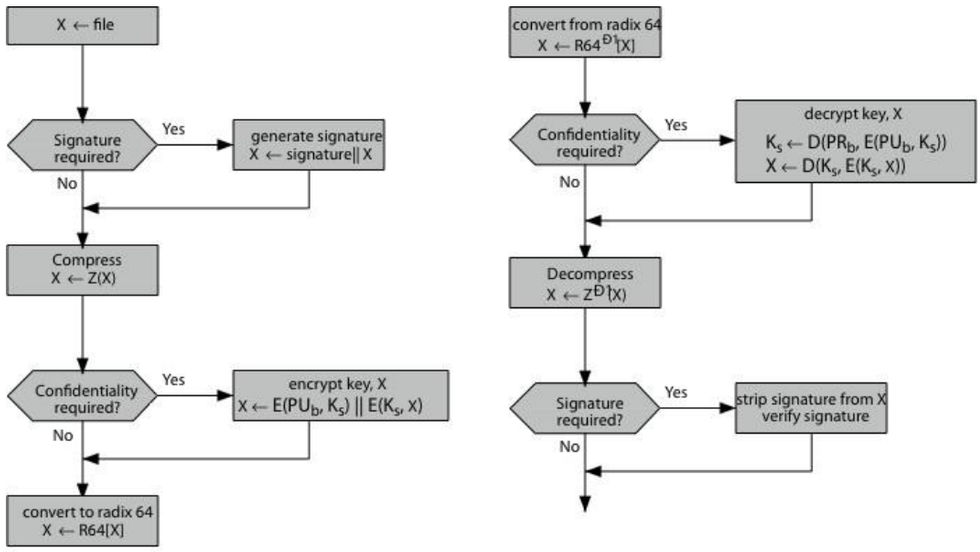
ofconvertingtheraw8-bitbinarystreamtoastreamofprintableASCIIcharacters.Itusesradix-64 conversion, in which each group of three octets of binary data is mapped intofour ASCII characters. This format also appends a CRC to detect transmission errors. Theuseofradix64expandsamessageby33%,butstillanoverallcompressionofaboutone-thirdcan be achieved.

PGPOperation-Segmentation/Reassembly

E-mailfacilitiesoftenarerestrictedtoamaximummessagelength.Forexample,manyofthe facilities accessible through the Internet impose a maximum length of 50,000 octets.Anymessagelongerthanthatmustbebrokenupintosmallersegments,eachofwhichismail ed separately. To accommodate this restriction,PGP automatically subdivides amessage that is too large into segments that are small enough to send via e-mail. Thesegmentationisdoneafterallofthe otherprocessing,includingtheradix-64conversion.Thus, the session key component and signature component appear only once, at thebeginningofthefirstsegment.Reassemblyatthereceivingendisrequiredbeforeverifyingsig nature ordecryption

PGPOperations– Summary

Function	Algorithms Used	Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key, and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key, and included with the message.
Compression	ZIP	A message may be compressed, for storage or transmission, using ZIP.
Email compatibility	Radix 64 conversion	To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversion.
Segmentation	—	To accommodate maximum message size limitations, PGP performs segmentation and reassembly.



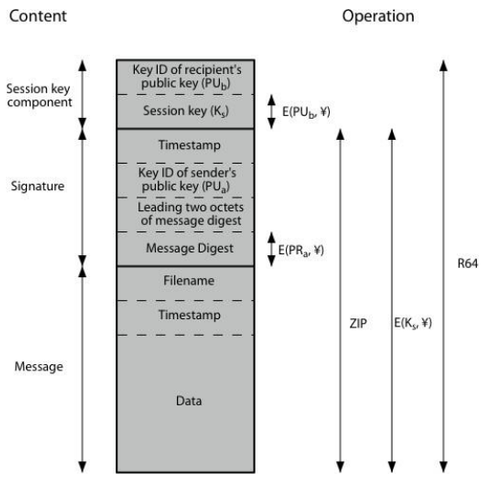
(a) Generic Transmission Diagram (from A)

(b) Generic Reception Diagram (to B)

PGPMessageFormat

Amessageconsistsofthreecomponents:themessagecomponent,asignature(optional),andasessionkeycomponent(optional).The*messagecomponent*includestheactualdatato be stored or transmitted, as well as a filename and a timestamp that specifies the timeofcreation.The *signaturecomponent*includesthe following:

- *Timestamp*:Thetimeatwhichthesignaturewasmade.
- *Messagedigest*:The160-bitSHA-1digest,encryptedwiththesender'sprivatesignaturekey.
- *Leading two octets of message digest*: To enable the recipient to determine if thecorrectpublickeywasusedtodecryptthemessagedigestforauthentication,bycomparing this plaintext copy of the first two octets with the first two octets of thedecrypted digest.Theseoctetsalsoserveasa16-bitframechecksequencefor themessage.
- *Key ID of sender's public key*: Identifies the public key that should be used to decryptthe message digest and, hence, identifies the private key that was used to encrypt themessagedigest



Notation:

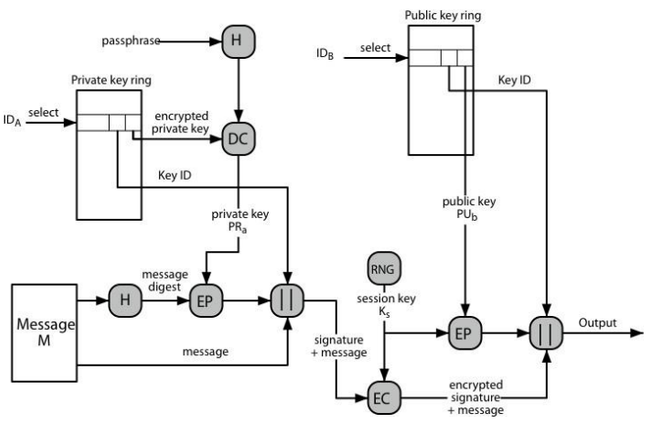
$E(PU_b, \bullet)$ = encryption with user b's public key
$E(PR_a, \bullet)$ = encryption with user a's private key
$E(K_s, \bullet)$ = encryption with session key
ZIP = Zip compression function
R64 = Radix-64 conversion function

The *session key component* includes the session key and the identifier of the recipient'spublic key that was used by the sender to encrypt the session key. The entire block isusuallyencodedwith radix-64encoding.

PGPMessageTransmissionandReception

Messagetransmission

The following figure shows the steps during message transmission assuming that themessageis to be bothsignedandencrypted.



ThesendingPGPentityperformsthefollowingsteps:

Signingthemessage

a. PGP retrieves the sender's private key from the private-key ring using your_userid asan index. If your_userid was not provided in the command, the first private key on theringis retrieved.

b. PGPpromptstheuserforthepassphrasetorecovertheunencryptedprivatekey.

c. Thesignaturecomponentofthemessageisconstructed

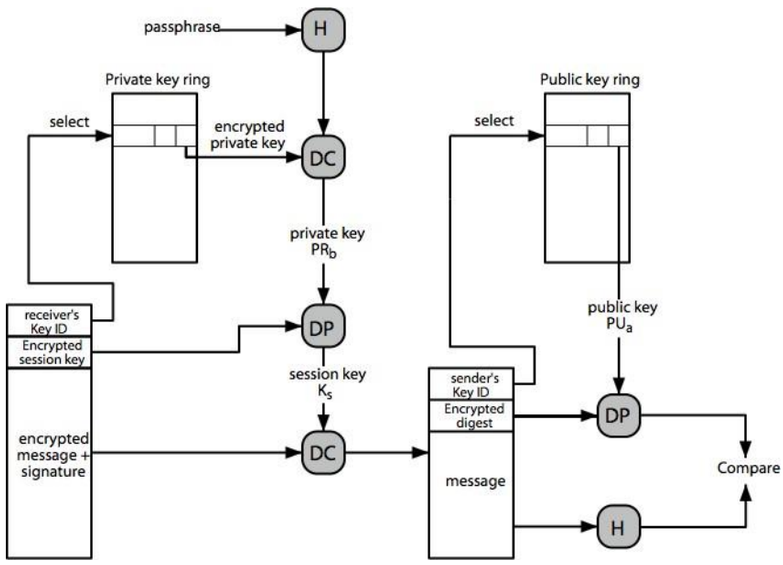
Encryptingthemessage

a. PGPgeneratesasessionkeyandencryptsthemessage.

b. PGPretrievestherecipient'spublickeyfromthepublic-keyringusingher_useridasanindex.

c. Thesessionkeycomponentofthemessageisconstructed.

MessageReception



ThereceivingPGPentityperformsthefollowingsteps:

Decryptingthemessage

a. PGPretrievesthereceiver'sprivatekeyfromtheprivate-keyring,usingtheKeyIDfieldinthesession key componentof themessageasan index.

b. PGPpromptstheuserforthepassphrasetorecovertheunencryptedprivatekey.

c. PGPthenrecoversthesessionkeyanddecryptsthemessage.

Authenticatingthemessage

a. PGPretrievesthesender'spublickeyfromthepublic-keyring,usingtheKeyIDfieldinthesignature keycomponentof themessageasan index.

b. PGPrecoversthetransmittedmessagedigest.

c. PGPcomputesthemessagedigestforthereceivedmessageandcomparesittothetransmitted messagedigest to authenticate.

S/MIME

S/MIME(Secure/MultipurposeInternetMailExtension)isasecurityenhancementtotheMIMEInternete-mail formatstandard,whichinturnprovidedsupportforvaryingcontenttypesandmulti-partmessagesoverthetextonlysupportintheoriginalInternetRFC822emailstandard.MIMEallowsencodingofbinarydatatotextualformfortransport over traditional RFC822 email systems. S/MIME is defined in a number ofdocuments, most importantly RFCs 3369, 3370, 3850 and 3851 and S/MIME support isnowincludedin manymodern mailagents.

RFC822

RFC 822 defines a format for text messages that are sent using electronic mail and it hasbeenthestandardforInternet-basedtextmailmessage.TheoverallstructureofamessagethatconformstoRFC822isverysimple.Amessageconsistsofsomenumberofheaderlines(theheader)followedbyunrestrictedtext(the body).Theheaderisseparated from the body by a blank line. A header line usually consists of a keyword,followed by a colon, followed by the keyword's arguments; the format allows a long lineto be broken up into several lines. The most frequently used keywords are *From, To*,*Subject*,and*Date*.

MultipurposeInternetMailExtensions

MIME is an extension to the RFC 822 framework that is intended to address some of theproblems and limitations of the use of SMTP (Simple Mail Transfer Protocol) or someother mail transfer protocol and RFC 822 for electronic mail. **Problems with RFC 822andSMTP**

• Executable files or other binary objects must be converted into ASCII. Various schemesexist(e.g.,Unix UUencode), butastandardis needed

• Textdatathatincludesspecialcharacters(e.g.,Hungariantext)cannotbetransmittedasSMTP is limitedto7-bit ASCII

• Someserversrejectmailmessagesoveracertainsize

• SomecommonproblemsexistwiththeSMTPimplementationswhichdonotadherecompletely tothe SMTPstandards definedinRFC821. Theyare:

- delete, add, or reorder CR and LF
- characterstruncate or wrap lines longer than 76
- charactersremove trailing white space (tabs and
- spaces)pad lines in a message to the same
- lengthconverttabcharactersintomultiplespaces

MIMEisintendedtoresolvetheseproblemsinamannerthatiscompatiblewithexistingRFC822implementationsandthespecificationisprovidedinRFC's2045through2049.

TheMIMEspecificationincludesthefollowingelements:

1. Fivenewmessageheaderfieldsaredefined,whichprovideinformationaboutthebodyofthemessage.
2. Anumberofcontentformatsaredefined,thusstandardizingrepresentationsthatsupportmultimedia electronic mail.
3. Transferencodingsaredefinedthatprotectthecontentfromalterationbythemailsystem.

MIME-Newheaderfields ThefiveheaderfieldsdefinedinMIMEareasfollows:

• *MIME-Version:*Musthavetheparametervalue1.0.Thisfieldindicatesthatthemessageconformsto RFCs 2045and2046.

• *Content-Type*: Describes the data contained in the body with sufficient detail that thereceiving user agent can pick an appropriate agent or mechanism to represent the datatothe userorotherwise dealwiththedatainanappropriate manner.

• *Content-Transfer-Encoding*: Indicates the type of transformation that has been used torepresentthebodyofthemessage inawaythatis acceptableformailtransport.

• Content-ID:UsedtoidentifyMIMEentitiesuniquelyinmultiplecontexts.

• *Content-Description*: A text description of the object with the body; this is useful whentheobject is not readable(e.g.,audio data).

MIMEContentTypesThebulkoftheMIMEspecificationisconcernedwiththedefinitionof a variety of content types. There are seven different major types of content and a totalof 15 subtypes. In general, a content type declares the general type of data, and thesubtype specifies a particular format for that type of data. For the text type of body, theprimary subtype is plain text, which is simply a string of ASCII characters or ISO 8859characters.Theenrichedsubtypeallowsgreaterformattingflexibility.Themultiparttypein dicates that the body contains multiple, independent parts. The Content-Type headerfieldincludesaparametercalledboundarythatdefinesthedelimiterbetweenbodyparts.T his boundary should not appear in any parts of the message. Each boundary starts on anewlineandconsistsoftwohyphensfollowedbytheboundaryvalue.Thefinalboundary,whichin dicatestheendofthelastpart,alsohasasuffixoftwohyphens.Withineachpart,theremaybeanopti onalordinaryMIMEheader.Therearefoursubtypesofthemultiparttype,allofwhichhave the sameoverall syntax.

Type	Subtype	Description
Text	Plain	Unformatted text; may be ASCII or ISO 8859.
	Enriched	Provides greater format flexibility.
Multipart	Mixed	The different parts are independent but are to be transmitted together. They should be presented to the receiver in the order that they appear in the mail message.
	Parallel	Differs from Mixed only in that no order is defined for delivering the parts to the receiver.
	Alternative	The different parts are alternative versions of the same information. They are ordered in increasing faithfulness to the original, and the recipient's mail system should display the "best" version to the user.
	Digest	Similar to Mixed, but the default type/subtype of each part is message/rfc822.
Message	rfc822	The body is itself an encapsulated message that conforms to RFC 822.
	Partial	Used to allow fragmentation of large mail items, in a way that is transparent to the recipient.
	External-body	Contains a pointer to an object that exists elsewhere.
Image	jpeg	The image is in JPEG format, JFIF encoding.
	gif	The image is in GIF format.
Video	mpeg	MPEG format.
Audio	Basic	Single-channel 8-bit ISDN mu-law encoding at a sample rate of 8 kHz.
Application	PostScript	Adobe Postscript.
	octet-stream	General binary data consisting of 8-bit bytes.

Themessagetypeprovidesanumberofimportantcapabilitiesin MIME.Themessage/rfc822subt ypeindicatesthatthebodyisanentiremessage,includingheader

and body. Despite the name of this subtype, the encapsulated message may be not only asimple RFC 822 message, but also any MIME message. The message/partial subtypeenablesfragmentationofalargemessageintoanumberofparts,whichmustbereassembled at the destination. For this subtype, three parameters are specified in theContent-Type:Message/Partialfield:anidcommontoallfragmentsofthesamemessage,a sequence number unique to each fragment, and the total number of fragments. Themessage/external-body subtype indicates that the actual data to be conveyed in thismessagearenotcontainedinthebody.Instead,thebodycontainstheinformationneededto access the data. The application type refers to other kinds of data, typically eitheruninterpretedbinarydataorinformationtobeprocessedbyamail-basedapplication.

MIME Transfer Encodings The other major component of the MIME specification, inaddition to content type specification, is a definition of transfer encodings for messagebodies.Theobjectiveistoprovidereliabledeliveryacrossthelargestrangeofenvironments.

MIME Transfer Encodings

7bit	The data are all represented by short lines of ASCII characters.
8bit	The lines are short, but there may be non-ASCII characters (octets with the high-order bit set).
binary	Not only may non-ASCII characters be present but the lines are not necessarily short enough for SMTP transport.
quoted-printable	Encodes the data in such a way that if the data being encoded are mostly ASCII text, the encoded form of the data remains largely recognizable by humans.
base64	Encodes data by mapping 6-bit blocks of input to 8-bit blocks of output, all of which are printable ASCII characters.
x-token	A named nonstandard encoding.

TheMIMEstandarddefinestwomethodsofencodingdata.TheContent-Transfer-Encoding field can actually take on six values. Three of these values (7bit, 8bit, andbinary)indicatethatnoencodinghasbeendonebutprovidesomeinformationaboutthenature of the data. Another Content-Transfer-Encoding value is x-token, which indicatesthat some other encoding scheme is used, for which a name is to be supplied. The twoactual encoding schemes defined are quoted-printable and base64. Two schemes aredefinedtoprovideachoicebetweenatransfertechniquethatisessentiallyhuman

readableandonethatissafeforalltypesofdatainawaythatisreasonablycompact.Thequoted-printabletransferencodingisusefulwhenthedataconsistslargelyofoctetsthatcorrespondtoprintableASCIIcharacters.Inessence,itrepresentsnonsafecharactersbythehexadecimalrepresentationoftheircodeandintroducesreversible(soft)linebreaksto limit message lines to 76 characters. The base64 transfer encoding, also known asradix-64 encoding, is a common one for encoding arbitrary binary data in such a way astobe invulnerableto theprocessingbymailtransportprograms.

CanonicalForm

AnimportantconceptinMIMEandS/MIMEisthatofcanonicalform.Canonicalformisaformat, appropriate to the content type, that is standardized for use between systems.This is in contrast to native form, which is a format that may be peculiar to a particularsystem.

Native Form	The body to be transmitted is created in the system's native format. The native character set is used and, where appropriate, local end-of-line conventions are used as well. The body may be a UNIX-style text file, or a Sun raster image, or a VMS indexed file, or audio data in a system-dependent format stored only in memory, or anything else that corresponds to the local model for the representation of some form of information. Fundamentally, the data is created in the "native" form that corresponds to the type specified by the media type.
Canonical Form	The entire body, including "out-of-band" information such as record lengths and possibly file attribute information, is converted to a universal canonical form. The specific media type of the body as well as its associated attributes dictate the nature of the canonical form that is used. Conversion to the proper canonical form may involve character set conversion, transformation of audio data, compression, or various other operations specific to the various media types. If character set conversion is involved, however, care must be taken to understand the semanties of the media type, which may have strong implications for any character set conversion (e.g. with regard to syntactically meaningful characters in a text subtype other than "plain").

S/MIMEFunctionality

S/MIMEhasaverysimilarfunctionalitytoPGP.Bothoffertheabilitytosignand/orencryptmessages.

Functions

S/MIMEprovidesthefollowingfunctions:

• **Envelopeddata**:Thisconsistsofencryptedcontentofanytypeandencrypted-contentencryptionkeys foroneormore recipients.

• **Signed data**: A digital signature is formed by taking the message digest of the contenttobesignedandthenencryptingthatwiththeprivatekeyofthesigner.Thecontentplussignature are then encoded using base64 encoding. A signed data message can only beviewedbyarecipient with S/MIMEcapability.

• **Clear-signed data**: As with signed data, a digital signature of the content is formed.However, in this case, only the digital signature is encoded using base64. As a result,recipientswithoutS/MIMEcapabilitycanviewthemessagecontent,althoughtheycannot verifythesignature.

• **Signed and enveloped data**: Signed-only and encrypted-only entities may be nested,so that encrypted data may be signed and signed data or clear-signed data may beencrypted.

IPSECURITYOVERVIEW

Definition:InternetProtocolsecurity(IPSec)isaframeworkofopenstandardsforprotecting communications over Internet Protocol (IP) networks through the use ofcryptographic security services. IPSec supports network-level peer authentication, dataoriginauthentication,dataintegrity,dataconfidentiality(encryption),andreplayprotection.

NeedforIPSec

In Computer Emergency Response Team (CERT)'s 2001 annual report it listed 52,000security incidents in which most serious types of attacks included **IP spoofing**, in whichintruderscreatepacketswithfalseIPaddressesandexploitapplicationsthatuseauthenticationbasedonIPandvariousformsof**eavesdroppingandpacketsniffing**,inwhichattackersreadtransmittedinformation,includinglogoninformationanddatabase

contents.Inresponsetotheseissues,theIABincludedauthenticationandencryptionasnecessarysecurityfeatures inthenext-generation IP i.e.IPv6.

ApplicationsofIPSec

IPSecprovidesthecapabilitytosecurecommunicationsacrossaLAN,acrossprivateandpublicwide areanetworks (WAN's),andacrosstheInternet.

• ***Secure branch office connectivity over the Internet***: A company can build a securevirtual private network over the Internet or over a public WAN. This enables a businesstorelyheavilyontheInternetandreduceitsneedforprivatenetworks,savingcostsandnetworkmanagementoverhead.

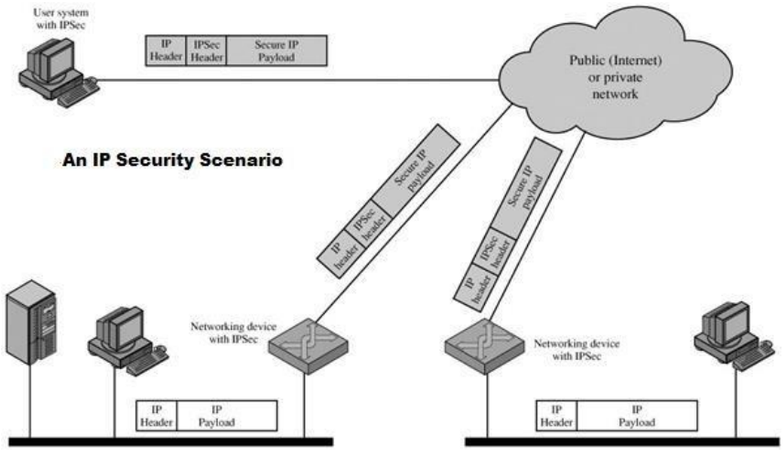
• ***Secure remote access over the Internet***: An end user whose system is equipped withIP security protocols can make a local call to an Internet service provider (ISP) and gainsecure access to a company network. This reduces the cost of toll charges for travellingemployeesandtelecommuters.

• ***Establishing extranet and intranet connectivity with partners***: IPSec can be used tosecurecommunicationwithotherorganizations,ensuringauthenticationandconfidentialityandprovidingakeyexchange mechanism.

• ***Enhancing electronic commerce security***: Even though some Web and electroniccommerce applications have built-in security protocols, the use of IPSec enhances thatsecurity.

The principal feature of IPSec enabling it to support varied applications is that it canencrypt and/or authenticate all traffic at IP level. Thus, all distributed applications,including remote logon, client/server, e-mail, file transfer, Web access, and so on, can besecured.

The following figure shows a typical scenario of IPSec usage. An organization maintains LANsatdispersed locations.NonsecureIPtrafficis conductedoneach LAN.



The IPSec protocols operate in networking devices, such as a router or firewall thatconnecteachLANtotheoutsideworld.TheIPSecnetworkingdevicewilltypicallyencrypt and compress all traffic going into the WAN, and decrypt and decompress trafficcoming from the WAN; these operations are transparent to workstations and servers onthe LAN. Secure transmission is also possible with individual users who dial into theWAN.SuchuserworkstationsmustimplementtheIPSecprotocolstoprovidesecurity.

BenefitsofIPSec

ThebenefitsofIPSecarelistedbelow:

• IPSecinafirewall/routerprovidesstrongsecuritytoalltrafficcrossingtheperimeter

• IPSecinafirewallisresistanttobypass

• IPSecisbelowtransportlayer(TCP,UDP),hencetransparenttoapplications

• IPSeccanbetransparenttoendusers

• IPSeccanprovidesecurityforindividualusersifneeded(usefulforoffsiteworkersandsettingup asecure virtualsubnetworkforsensitive applications)

RoutingApplications

IPSecalsoplaysavitalroleintheroutingarchitecturerequiredforinternetworking.Itassuresthat :

• routeradvertisementscomefromauthorizedrouters

• neighboradvertisementscomefromauthorizedrouters

• redirectmessagescomefromtheroutertowhichinitialpacketwassent

• Aroutingupdateisnotforged

IPSECURITYARCHITECTURE

TounderstandIPSecurityarchitecture,weexamineIPSecdocumentsfirstandthenmoveontoIPSec services andSecurityAssociations.

IPSecDocuments

The IPSec specification consists of numerous documents. The most important of these,issuedin Novemberof1998, areRFCs2401, 2402,2406, and2408:

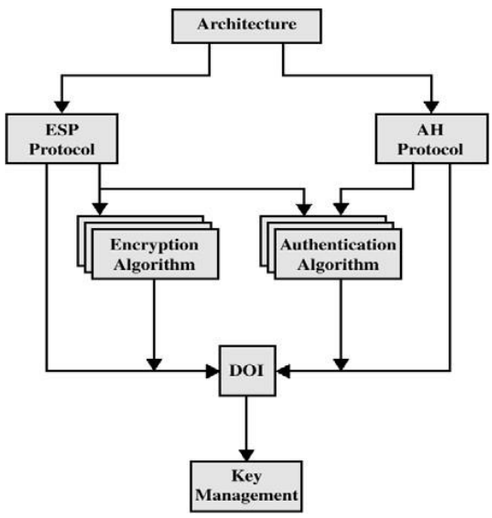
• RFC2401:Anoverviewofasecurityarchitecture

• RFC2402:DescriptionofapacketauthenticationextensiontoIPv4andIPv6

• RFC2406:DescriptionofapacketencryptionextensiontoIPv4andIPv6

• RFC2408:Specificationofkeymanagementcapabilities

SupportforthesefeaturesismandatoryforIPv6andoptionalforIPv4.Inbothcases,thesecurity features are implemented as extension headers that follow the main IP header.The extension header for authentication is known as the Authentication header; that forencryption is known as the Encapsulating Security Payload (ESP) header. In addition tothese four RFCs, a number of additional drafts have been published by the IP SecurityProtocol Working Group set up by the IETF. The documents are divided into sevengroups,asdepictedin followingfigure:



• **Architecture**:Coversthegeneralconcepts,securityrequirements,definitions,andmechanismsdefiningIPSec technology

• **EncapsulatingSecurityPayload(ESP):**Coversthepacketformatandgeneralissuesrelatedtotheuseoftheespforpacketencryptionand,optionally,authentication.

• **AuthenticationHeader(AH):**Coversthepacketformatandgeneralissuesrelatedtotheuseof AH forpacketauthentication.

•**Encryption Algorithm**: A set of documents that describe how various encryptionalgorithmsareused forESP.

• **AuthenticationAlgorithm:**AsetofdocumentsthatdescribehowvariousauthenticationalgorithmsareusedforAHandfortheauthenticationoptionofESP.

• **KeyManagement**:Documentsthatdescribekeymanagementschemes.

• **Domain of Interpretation (DOI):** Contains values needed for the other documents torelatetoeachother.Theseincludeidentifiersforapprovedencryptionandauthenticationalgorithms,aswellasoperationalparameters suchaskeylifetime.

IPSecServices

IPSecarchitecturemakesuseoftwomajorprotocols(i.e.,AuthenticationHeaderandESPprotocols) for providing security at IP level. This facilitates the system to beforehandchooseanalgorithmtobeimplemented,securityprotocolsneededandanycryptographickeysrequiredtoproviderequestedservices.TheIPSecservices areasfollows:

Connectionless Integrity:- Data integrity service is provided by IPSec via AH whichpreventsthedatafrombeingalteredduringtransmission.

Data Origin Authentication:- This IPSec service prevents the occurrence of replayattacks,address spoofingetc., whichcanbe fatal

AccessControl:-

Thecryptographickeysaredistributedandthetrafficflowiscontrolledinboth AHandESPprotocols,whichisdonetoaccomplishaccesscontroloverthedatatransmission.

Confidentiality:-

Confidentialityonthedatapacketisobtainedbyusinganencryptiontechniqueinwhichallthedata packetsaretransformed intociphertextpacketswhichareunreadableanddifficult tounderstand.

Limited Traffic Flow Confidentiality:- This facility or service provided by IPSecensures that the confidentiality is maintained on the number of packets transferred orreceived.This can be done usingpaddingin ESP.

ReplaypacketsRejection:-

Theduplicateorreplaypacketsareidentifiedanddiscardedusingthesequence numberfieldin both AHandESP.

	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✔	✔	✔
Connectionless integrity	✔		✔
Data origin authentication	✔		✔
Rejection of replayed packets	✔	✔	✔
Confidentiality		✔	✔
Limited traffic flow confidentiality		✔	✔

SECURITYASSOCIATIONS

Since IPSEC is designed to be able to use various security protocols, it uses SecurityAssociations (SA) to specify the protocols to be used. SA is a database record whichspecifiessecurityparameterscontrollingsecurityoperations.Theyarereferencedbythes endinghostandestablishedbythereceivinghost.AnindexparametercalledtheSecurity Parameters Index (SPI) is used. SAs are in one direction only and a second SAmust be established for the transmission to be bi-directional. A security association isuniquelyidentifiedbythree parameters:

- **Security Parameters Index (SPI):** A bit string assigned to this SA and having localsignificanceonly.TheSPIiscarriedinAHandESPheaderstoenablethereceivingsystemtosel ectthe SAunder whichareceivedpacketwillbe processed.
- **IP Destination Address**: Currently, only unicast addresses are allowed; this is theaddress of the destination endpoint of the SA, which may be an end user system or anetworksystemsuchasafirewall or router.
- **Security Protocol Identifier**: This indicates whether the association is an AH or ESPsecurityassociation.

SAParameters

In each IPSec implementation, there is a nominal Security Association Database thatdefinestheparametersassociatedwitheachSA.Asecurityassociationisnormallydefinedbyt hefollowingparameters:

• **Sequence Number Counter:** A 32-bit value used to generate the Sequence Numberfieldin AH orESP headers

• **Sequence Counter Overflow:** A flag indicating whether overflow of the SequenceNumberCountershouldgenerateanauditableeventandpreventfurthertransmission ofpacketson thisSA(requiredforallimplementations).

• **Anti-Replay Window:** Used to determine whether an inbound AH or ESP packet is areplay

• **AHInformation:**Authenticationalgorithm,keys,keylifetimes,andrelatedparametersbeingu sedwith AH (requiredforAH implementations).

• **ESP Information:** Encryption and authentication algorithm, keys, initialization values,keylifetimes,andrelatedparametersbeingusedwithESP(requiredforESPimplementati ons).

• **LifetimeofThisSecurityAssociation:**AtimeintervalorbytecountafterwhichanSAmustbere placedwithanewSA(andnewSPI)orterminated,plusanindicationofwhichofthese actions shouldoccur(requiredforallimplementations).

• **IPSecProtocolMode:**Tunnel,transport,orwildcard(requiredforallimplementations).The se modesare discussedlaterin this section.

• **Path MTU:** Any observed path maximum transmission unit (maximum size of a packetthat can be transmitted without fragmentation) and aging variables (required for allimplementations).

TransportandTunnelModes

BothAHandESPsupporttwomodesofuse:transportandtunnelmode.

	TransportModeSA	**Tunnel ModeSA**
AH	**Authenticates** IP payloadand selected portions of IPheaderandIPv6extensio n headers	**Authenticates** entire innerIP packet plus selectedportionsofouterIPh eader
ESP	**Encrypts** IP payload andanyIPv6extesion header	**Encrypts**innerIPpacket
ESPwithauthentication	**Encrypts** IP payload andany IPv6 extesion header.**Authenticates**IPp ayload butnoIPheader	**Encrypts** inner IP packet.**Authenticates** inner IPpacket

IPseccanbeused(bothAHpacketsandESPpackets)intwomodes

• **Transportmode**:theIPsecheaderisinsertedjustaftertheIPheader–thiscontainsthesecurityinformation,suchasSAidentifier,encryption, authentication

Typically used in end-to-end

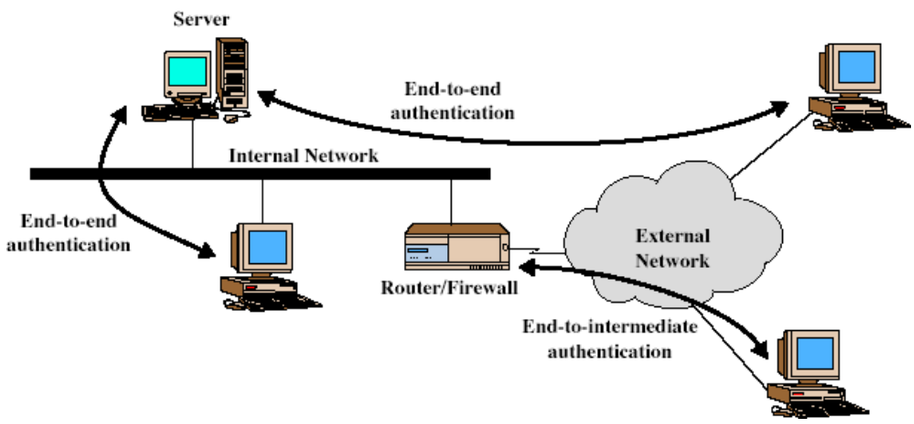
communicationIPheader notprotected

• **Tunnelmode**:theentireIPpacket,headerandall,isencapsulatedinthebodyofanewIPpacket with acompletelynewIPheader

Typically used in firewall-to-firewall

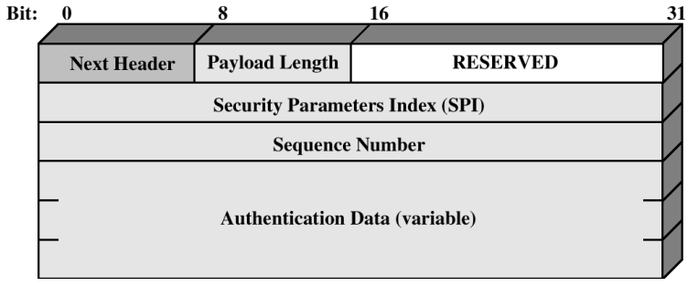
communicationProvidesprotectionforthewholeIPpacket

Noroutersalongthewaywillbeable(andwillnotneed)tocheckthecontentofthepackets



AUTHENTICATIONHEADER

The Authentication Header provides support for data integrity and authentication of IPpackets. The data integrity feature ensures that undetected modification to a packet'scontent in transit is not possible. The authentication feature enables an end system ornetworkdevicetoauthenticatetheuserorapplicationandfiltertrafficaccordingly;italsopreve nts the address spoofing attacks observed in today's Internet. The AH also guardsagainst the replay attack. Authentication is based on the use of a message authenticationcode (MAC), hence the two parties must share a secret key. The Authentication Headerconsistsofthefollowingfields:



IPSecAuthenticationHeader

• **NextHeader(8bits):**Identifiesthetypeofheaderimmediatelyfollowingthisheader.

• **Payload Length (8 bits):** Length of Authentication Header in 32-bit words, minus 2.For example, the default length of the authentication data field is 96 bits, or three 32-bitwords. With a three-word fixed header, there are a total of six words in the header, andthePayloadLengthfieldhas avalue of 4.

• **Reserved(16bits):**Forfutureuse.

• **SecurityParametersIndex(32bits):**Identifiesasecurityassociation.

• **SequenceNumber(32bits):**Amonotonicallyincreasingcountervalue,discussedlater.

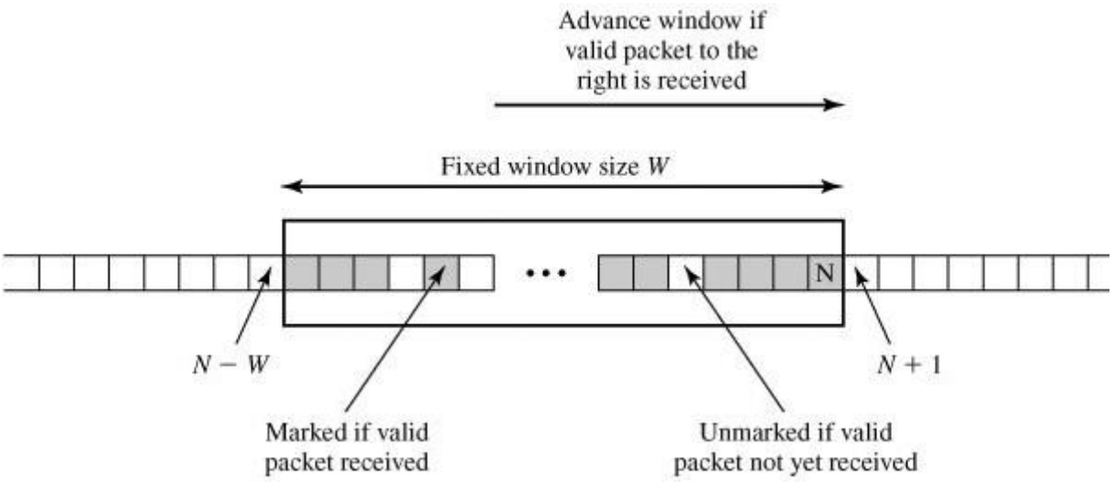
• **AuthenticationData(variable):**Avariable-lengthfield(mustbeanintegralnumberof32-bitwords)thatcontainstheIntegrityCheckValue(ICV),orMAC,forthispacket.

Anti-ReplayService

Anti-replay service is designed to overcome the problems faced due to replay attacks inwhich an intruder intervenes the packet being transferred, make one or more duplicatecopiesofthatauthenticatedpacketandthensendsthepacketstothedesireddestination ,thereby causing inconvenient processing at the destination node. The Sequence Numberfieldis designedtothwartsuch attacks.

WhenanewSAisestablished,thesenderinitializesasequencenumbercounterto0.Eachtime that a packet is sent on this SA, the sender increments the counter and places thevalue in the Sequence Number field. Thus, the first value to be used is 1. This value goeson increasing with respect to the number of packets being transmitted. The sequencenumber field in each packet represents the value of this counter. The maximum value ofthe sequence number field can go up to 2^{32}-1. If the limit of 2^{32}-1 is reached, the sendershouldterminate thisSAandnegotiate anew SAwithanewkey.

TheIPSecauthenticationdocumentdictatesthatthereceivershouldimplementawindow of size W, with a default of W = 64. The right edge of the window represents thehighest sequence number, N, so far received for a valid packet. For any packet with asequence number in the range from N-W+1 to N that has been correctly received (i.e.,properly authenticated), the corresponding slot in the window is marked as shown.Inboundprocessingproceedsasfollowswhen apacket is received:



AntireplayMechanism

1. If the received packet falls within the window and is new, the MAC is checked. If thepacketisauthenticated,thecorrespondingslotinthewindowismarked.
2. If the received packet is to the right of the window and is new, the MAC is checked. Ifthe packet is authenticated, the window is advanced so that this sequence number is therightedgeofthe window,andthecorrespondingslotinthewindowismarked.
3. If the received packet is to the left of the window, or if authentication fails, the packetisdiscarded;thisis an auditableevent.

IntegrityCheckValue

ICV is the value present in the authenticated data field of ESP/AH, which is used todetermine any undesired modifications made to the data during its transit. ICV can alsobe referred as MAC or part of MAC algorithm. MD5 hash code and SHA-1 hash code areimplementedalongwith HMACalgorithms i.e.,

- HMAC-MD5-96
- HMAC-SHA-1-96

In both cases, the full HMAC value is calculated but then truncated by using the first 96bits, which is the default length for the Authentication Data field. The MAC is calculatedover

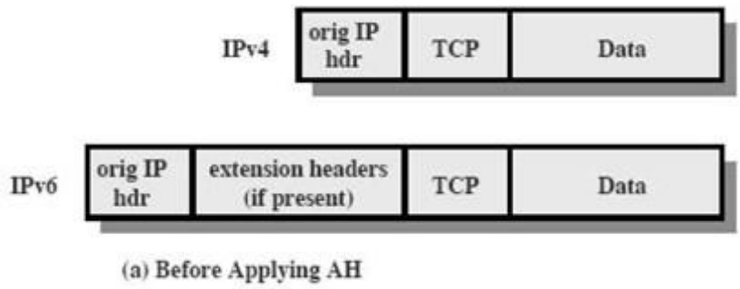
• IPheaderfieldsthateitherdonotchangeintransit(immutable)orthatarepredictableinvalueup onarrivalattheendpointfortheAHSA.Fieldsthatmaychangeintransitandwhosevalueonarrivali sunpredictablearesettozeroforpurposesofcalculationatbothsourceanddestination.

• The AH header other than the Authentication Data field. The Authentication Data fieldissetto zeroforpurposes ofcalculation atbothsourceanddestination.

• The entire upper-level protocol data, which is assumed to be immutable in transit (e.g.,aTCP segment oran innerIP packet in tunnelmode).

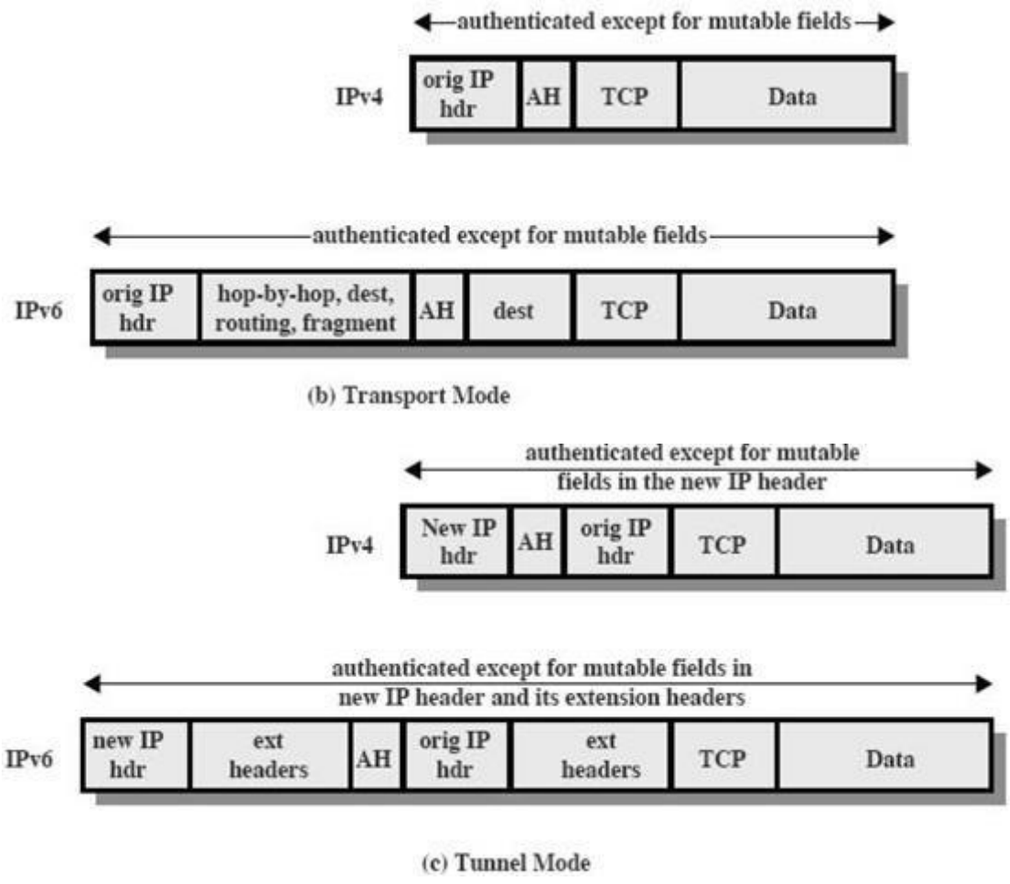
TransportandTunnelModes

ThefollowingfigureshowstypicalIPv4andIPv6packets.In thiscase,theIPpayloadisaTCP segment; it could also be a data unit for any other protocol that uses IP, such as UDPorICMP.



(a) Before Applying AH

For transport mode AH using IPv4, the AH is inserted after the original IP header andbeforetheIPpayload(e.g.,aTCPsegment)shownbelow.Authenticationcoverstheentirepack et,excludingmutablefieldsintheIPv4headerthataresettozeroforMACcalculation.Inthecontext ofIPv6,AHisviewedasanend-to-endpayload;thatis,itisnotexaminedorprocessedbyintermediaterouters.Therefore,theAHapp earsaftertheIPv6baseheaderandthehop-by-hop,routing,andfragmentextensionheaders.Thedestinationoptionsextensionheadercouldap pearbeforeoraftertheAHheader,depending on the semantics desired.Again, authentication covers the entire packet,excludingmutablefieldsthat aresettozeroforMACcalculation.



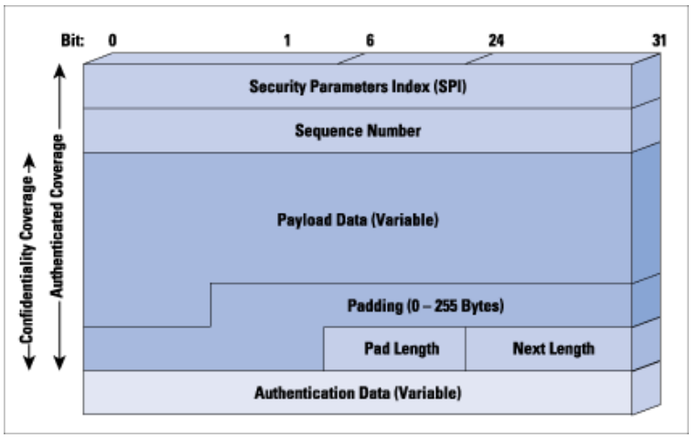
FortunnelmodeAH,theentireoriginalIPpacketisauthenticated,andtheAHisinsertedbetween the original IP header and a new outer IP header. The inner IP header carriesthe ultimate source and destination addresses, while an outer IP header may containdifferent IP addresses (e.g., addresses of firewalls or other security gateways). Withtunnel mode, the entire inner IP packet, including the entire inner IP header is protectedby AH. The outer IP header (and in the case of IPv6, the outer IP extension headers) isprotectedexceptformutableandunpredictable fields.

ENCAPSULATINGSECURITYPAYLOAD

TheEncapsulatingSecurityPayloadprovidesconfidentialityservices,includingconfidentialityo fmessagecontentsandlimitedtrafficflowconfidentiality.Asanoptionalfeature, ESP canalsoprovideanauthenticationservice.

ESPFormat

ThefollowingfigureshowstheformatofanESPpacket.Itcontainsthefollowingfields:



SecurityParametersIndex (32bits):Identifiesasecurityassociation.

• **Sequence Number** (32 bits): A monotonically increasing counter value; this providesananti-replayfunction,asdiscussedfor AH.

• **Payload Data (variable):** This is a transport-level segment (transport mode) or IPpacket(tunnelmode) that is protectedbyencryption.

• **Padding (0-255 bytes):** This field is used to make the length of the plaintext to be amultipleofsomedesirednumberofbytes.Itisalsoaddedtoprovideconfidentiality.

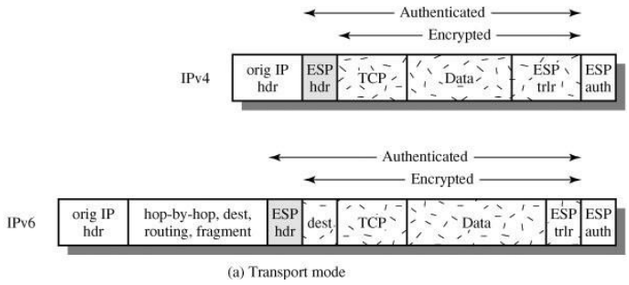
• **Pad Length (8 bits):** Indicates the number of pad bytes immediately preceding thisfield.

• **Next Header (8 bits):** Identifies the type of data contained in the payload data field byidentifying the first header in that payload (for example, an extension header in IPv6, oranupper-layer protocolsuchasTCP).

• **Authentication Data (variable):** A variable-length field (must be an integral numberof 32-bit words) that contains the Integrity Check Value computed over the ESP packetminustheAuthentication Datafield.

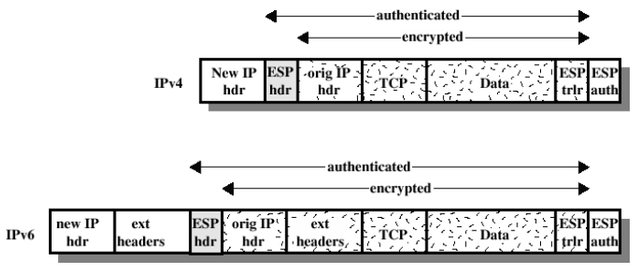
AddingencryptionmakesESPabitmorecomplicatedbecausetheencapsulation *surrounds*thepayloadratherthan*precedes*itaswithAH:ESPincludesheaderandtrailer

TransportModeESP



(a) Transport mode

TunnelModeESP

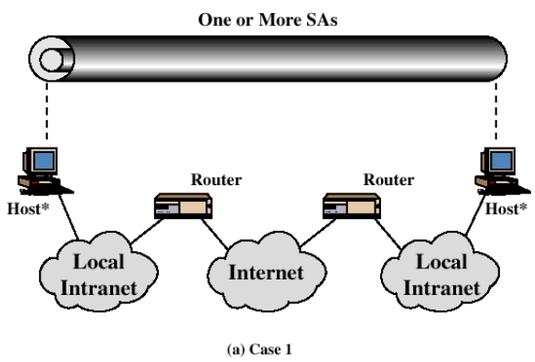


(b) Tunnel Mode

BasicCombinationsofSecurityAssociations

TheIPSecArchitecturedocumentlistsfourexamplesofcombinationsofSAsthatmustbesupported by compliant IPSec hosts (e.g., workstation, server) or security gateways (e.g.firewall,router).

case:-1



(a) Case 1

All security is provided between end systems that implement IPSec. For any two endsystems to communicate via an SA, they must share the appropriate secret keys. Amongthepossiblecombinations:

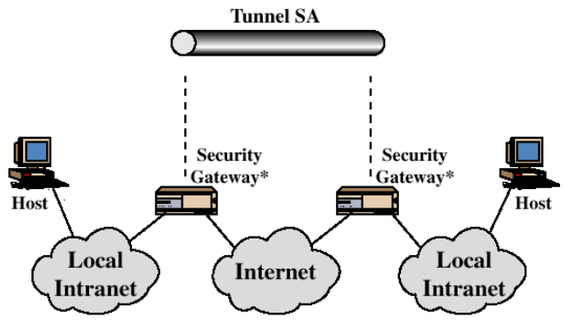
a) AHintransportmode

b) ESPintransportmode

c) ESPfollowedbyAHintransportmode(anESPSAinsideanAHSA)

d) Anyoneofa, b,orcinsideanAHorESPintunnelmode

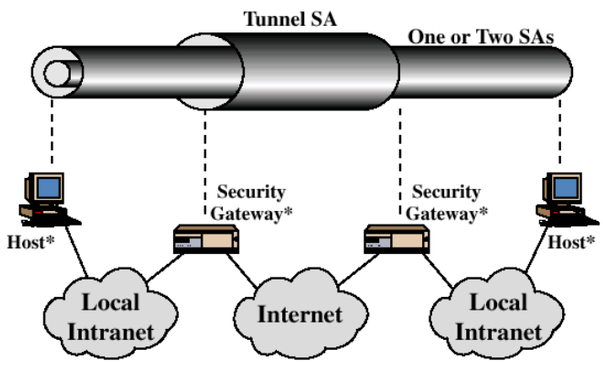
Case:-2



(b) Case 2

Securityisprovidedonlybetweengateways(routers,firewalls,etc.)andnohostsimplementIPSec.Thiscaseillustratessimplevirtualprivatenetworksupport.Thesecurity architecture document specifies that only a single tunnel SA is needed for thiscase. The tunnel could support AH, ESP, or ESP with the authentication option. NestedtunnelsarenotrequiredbecausetheIPSecservicesapplytotheentireinnerpacket.

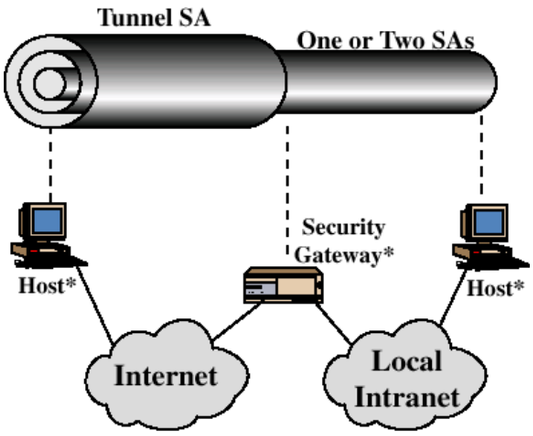
Case-3:-



(c) Case 3

The third combination is similar to the second, but in addition provides security even tonodes. This combination makes use of two tunnels first for gateway to gateway andsecondfornodetonode.Eitherauthenticationortheencryptionorbothcanbeprovidedby using gateway to gateway tunnel. An additional IPSec service is provided to theindividualnodes byusingnode to node tunnel.

Case:-4



(d) Case 4

This combination is suitable for serving remote users i.e., the end user sitting anywherein the world can use the internet to access the organizational workstations via thefirewall. This combination states that only one tunnel is needed for communicationbetweena remote userandanorganizationalfirewall.

KEYMANAGEMENT

The key management portion of IPSec involves the determination and distribution ofsecret keys. The IPSec Architecture document mandates support for two types of keymanagement:

• **Manual:** A system administrator manually configures each system with its own keysand with the keys of other communicating systems. This is practical for small, relativelystaticenvironments.

• **Automated:**Anautomatedsystemenablestheon-demandcreationofkeysforSAsandfacilitatestheuseofkeysinalargedistributedsystemwithane volvingconfiguration.

ThedefaultautomatedkeymanagementprotocolforIPSecisreferredtoasISAKMP/Oakleyandc onsists of thefollowingelements:

• **Oakley Key Determination Protocol**: Oakley is a key exchange protocol based on theDiffie-Hellman algorithm but providing added security. Oakley is generic in that it doesnotdictate specific formats.

• **Internet Security Association and Key Management Protocol (ISAKMP):** ISAKMPprovides a framework for Internet key management and provides the specific protocolsupport, includingformats, fornegotiation ofsecurityattributes.

OakleyKeyDeterminationProtocol

Oakley is a refinement of the Diffie-Hellman key exchange algorithm. The Diffie-Hellmanalgorithmhas twoattractivefeatures:

• Secretkeysarecreatedonlywhenneeded.Thereisnoneedtostoresecretkeysforalongperiodof time, exposing themtoincreasedvulnerability.

• Theexchangerequiresnopre-existinginfrastructureotherthananagreementontheglobalparameters.

However,Diffie-Hellmanhasgotsomeweaknesses:

• Noidentityinformationaboutthepartiesisprovided.

• Itispossibleforaman-in-the-middleattack

• Itiscomputationallyintensive.Asaresult,itisvulnerabletoacloggingattack,inwhichanopponent requestsahigh number of keys.

Oakleyisdesignedtoretaintheadvantagesof Diffie-Hellmanwhilecounteringitsweaknesses.

FeaturesofOakley

TheOakleyalgorithmis characterizedbyfiveimportantfeatures:

1. Itemploysamechanismknownascookiesto thwartcloggingattacks.
2. Itenablesthetwopartiestonegotiateagroup;this,inessence,specifiestheglobalparametersof theDiffie-Hellman keyexchange.
3. Itusesnoncestoensureagainstreplayattacks.
4. ItenablestheexchangeofDiffie-Hellmanpublickeyvalues.
5. ItauthenticatestheDiffie-Hellmanexchangetothwartman-in-the-middleattacks.

Incloggingattacks,anopponentforgesthesourceaddressofalegitimateuserandsendsapublicDiffie-Hellmankeytothevictim.Thevictimthenperformsamodularexponentiation to compute the secret key. Repeated messages of this type can clog thevictim's system with useless work. The **cookie exchange** requires that each side send apseudorandomnumber,thecookie,intheinitialmessage,whichtheothersideacknowledges. This acknowledgment must be repeated in the first message of the Diffie-Hellmankeyexchange.Therecommendedmethodforcreatingthecookieistoperforma

fast hash (e.g., MD5) over the IP Source and Destination addresses, the UDP Source andDestinationports,andalocallygeneratedsecretvalue.Oakleysupportstheuseofdifferent **groups** for the Diffie-Hellman key exchange. Each group includes the definitionof the two global parameters and the identity of the algorithm. Oakley employs **nonces**toensureagainstreplayattacks.Eachnonceisalocallygeneratedpseudorandomnumber . Nonces appear in responses and are encrypted during certain portions of theexchange to secure their use. Three different authentication methods can be used withOakleyaredigitalsignatures,public-keyencryptionandSymmetric-keyencryption.

Aggressive OakleyKeyExchange

Aggressive key exchange is a technique used for exchanging the message keys and is socalledbecauseonlythreemessagesareallowedtobeexchangedatanytime.

$\mathbf{I \rightarrow R:}$ $CKY_I, OK\_KEYX, GRP, g^x, EHAO, NIDP, ID_I, ID_R, N_I, S_{KI}[ID_I \| ID_R \| N_I \| GRP \| g^x \| EHAO]$

$\mathbf{R \rightarrow I:}$ $CKY_R, CKY_I, OK\_KEYX, GRP, g^y, EHAS, NIDP, ID_R, ID_I, N_R, N_I, S_{KR}[ID_R \| ID_I \| N_R \| N_I \| GRP \| g^y \| g^x \| EHAS]$

$\mathbf{I \rightarrow R:}$ $CKY_I, CKY_R, OK\_KEYX, GRP, g^x, EHAS, NIDP, ID_I, ID_R, N_I, N_R, S_{KI}[ID_I \| ID_R \| N_I \| N_R \| GRP \| g^x \| g^y \| EHAS]$

Notation:

I	=	Initiator
R	=	Responder
CKY_I, CKY_R	=	Initiator, responder cookies
OK_KEYX	=	Key exchange message type
GRP	=	Name of Diffie-Hellman group for this exchange
g^x, g^y	=	Public key of initiator, responder; g^{xy} = session key from this exchange
EHAO, EHAS	=	Encryption, hash authentication functions, offered and selected
NIDP	=	Indicates encryption is not used for remainder of this message
ID_I, ID_R	=	Identifier for initiator, responder
N_I, N_R	=	Random nonce supplied by initiator, responder for this exchange
$S_{KI}[X], S_{KR}[X]$	=	Indicates the signature over X using the private key (signing key) of intiator, responder

ExampleofAggressiveOakleyKeyExchange

In the first step, the initiator (I) transmits a cookie, the group to be used, and I's publicDiffie-Hellman key for this exchange. I also indicates the offered public-key encryption,hash, and authentication algorithms to be used in this exchange. Also included in thismessage are the identifiers of I and the responder (R) and I's nonce for this exchange.Finally, I appends a signature using I's private key that signs the two identifiers, thenonce, the group, the Diffie-Hellman public key, and the offered algorithms. When Rreceivesthemessage,RverifiesthesignatureusingI'spublicsigningkey.Racknowledgesthe message by echoing back I's cookie, identifier, and nonce, as well as the group. R alsoincludes in the message a cookie, R's Diffie-Hellman public key, the selected algorithms(which must be among the offered algorithms), R's identifier, and R's nonce for thisexchange.Finally,RappendsasignatureusingR'sprivatekeythatsignsthetwo

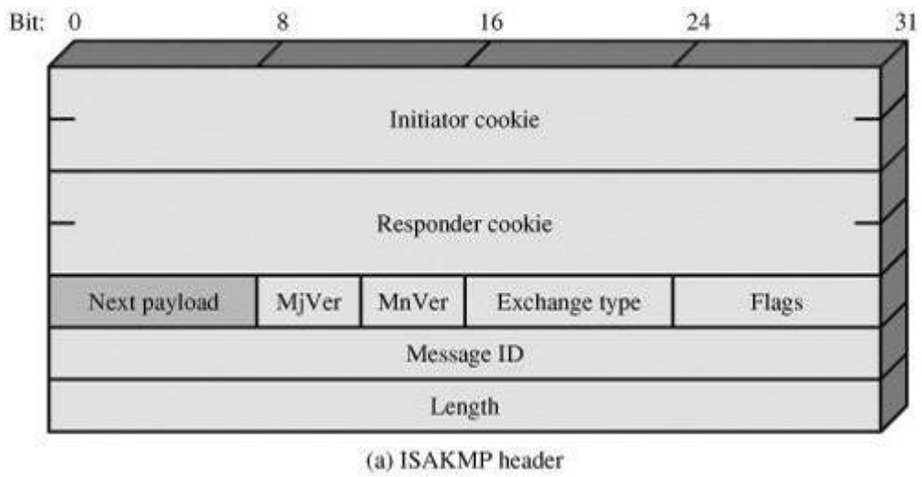
identifiers,thetwononces,thegroup,thetwoDiffie-Hellmanpublickeys,andtheselectedalgorithms.

When I receives the second message, I verifies the signature using R's public key. Thenoncevaluesinthemessageassurethatthisisnotareplayofanoldmessage.Tocompletetheexchange,ImustsendamessagebacktoRtoverifythatIhasreceivedR'spublickey.

ISAKMP

ISAKMPdefinesproceduresandpacketformatstoestablish,negotiate,modify,anddeletesecurityassociations.AspartofSAestablishment,ISAKMPdefinespayloadsforexchangingkeygeneration andauthenticationdata.

ISAKMPHeader Format



(a) ISAKMP header

AnISAKMPmessageconsistsofanISAKMPheaderfollowedbyoneormorepayloadsandmust follow UDP transport layer protocol for its implementation. The header format ofanISAKMP header is shownbelow:

- InitiatorCookie(64bits):CookieofentitythatinitiatedSAestablishment,SAnotification, orSAdeletion.
- ResponderCookie(64bits):Cookieofrespondingentity;nullinfirstmessagefrominitiator.
- NextPayload(8bits):Indicatesthetypeofthefirstpayloadinthemessage
- MajorVersion(4bits):Indicatesmajorversionof ISAKMPinuse.
- MinorVersion(4bits):Indicatesminorversioninuse.

• ExchangeType(8bits):Indicatesthetypeofexchange.Canbeinformational,aggressive, authenticationonly,identityprotection orbaseexchange(S).

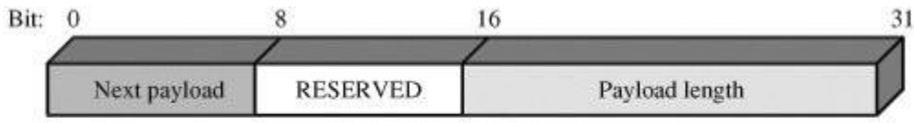
• Flags (8 bits): Indicates specific options set for this ISAKMP exchange. Two bits so fardefined:TheEncryptionbitissetifallpayloadsfollowingtheheaderareencryptedusingthe encryption algorithm for this SA. The Commit bit is used to ensure that encryptedmaterialis not receivedpriortocompletionof SAestablishment.

• MessageID(32bits):UniqueIDforthismessage.

• Length(32bits):Lengthoftotalmessage(headerplusallpayloads)inoctets.

ISAKMPPayloadTypes

AllISAKMPpayloadsbeginwiththesamegenericpayloadheadershownbelow.



(b) Generic payload header

TheNextPayloadfieldhasavalueof0ifthisisthelastpayloadinthemessage;otherwiseits value is the type of the next payload. The Payload Length field indicates the length inoctets of this payload, including the generic payload header. There are many differentISAKMPpayloadtypes.They are:

a. TheSApayloadisusedtobegintheestablishmentofanSA.TheDomainofInterpretation parameter identifies the DOI under which negotiation is taking place. TheSituationparameterdefinesthesecuritypolicyforthisnegotiation;inessence,thelevelsofsec urityrequiredforencryptionandconfidentialityarespecified(e.g.,sensitivitylevel,securitycom partment).

b. The Proposal payload contains information used during SA negotiation. The payloadindicates the protocol for this SA (ESP or AH) for which services and mechanisms arebeing negotiated. The payload also includes the sending entity's SPI and the number oftransforms.Eachtransformiscontainedinatransformpayload.

c. TheTransformpayloaddefinesasecuritytransformtobeusedtosecurethecommunicationsc hannelforthedesignatedprotocol.TheTransform#parameterservestoidentifythisparticularp ayloadsothattherespondermayuseittoindicateacceptance

of this transform. The Transform-ID and Attributes fields identify a specific transform(e.g., 3DES for ESP, HMAC-SHA-1-96 for AH) with its associated attributes (e.g., hashlength).

d. The Key Exchange payload can be used for a variety of key exchange techniques,including Oakley, Diffie-Hellman, and the RSA-based key exchange used by PGP. The KeyExchangedatafieldcontainsthedatarequiredtogenerateasessionkeyandisdependentonthe key exchange algorithmused.

e. The Identification payload is used to determine the identity of communicating peersand may be used for determining authenticity of information. Typically the ID Data fieldwillcontain an IPv4orIPv6address.

f. The Certificate payload transfers a public-key certificate. The Certificate Encoding fieldindicates the type of certificate or certificate-related information, which may includeSPKI,ARL,CRL,PGPinfoetc.AtanypointinanISAKMPexchange,thesendermayincludeaCertificateRequestpayloadtorequestthecertificateoftheothercommunicatingentity.

g. The Hash payload contains data generated by a hash function over some part of themessage and/or ISAKMP state. This payload may be used to verify the integrity of thedatainamessageortoauthenticate negotiatingentities.

h. The Signature payload contains data generated by a digital signature function oversomepartofthemessageand/orISAKMPstate.Thispayloadisusedtoverifytheintegrityofthedata inamessage andmaybe usedfornonrepudiation services.

i. TheNoncepayloadcontainsrandomdatausedtoguaranteelivenessduringanexchangeandprotectagainst replayattacks.

j. The Notification payload contains either error or status information associated withthisSAorthisSAnegotiation.SomeoftheISAKMPerrormessagesthathavebeendefinedareInvalidFlags,InvalidCookie, PayloadMalformedetc

k. The Delete payload indicates one or more SAs that the sender has deleted from itsdatabaseandthat thereforeare no longervalid.

ISAKMPExchanges

ISAKMP provides a framework for message exchange, with the payload types serving asthe building blocks. The specification identifies five default exchange types that shouldbesupported.

1. Base Exchange: allows key exchange and authentication material to be transmittedtogether.Thisminimizesthenumberofexchangesattheexpenseofnotprovidingidentityprotection.

(a) Base Exchange

(1)**I**⟶**R**: SA; NONCE	Begin ISAKMP-SA negotiation
(2)**R**⟶**E**: SA; NONCE	Basic SA agreed upon
(3)**I**⟶**R**: KE; ID_I AUTH	Key generated; Initiator identity verified by responder
(4)**R**⟶**E**: KE; ID_R AUTH	Responder identity verified by initiator; Key generated; SA established

The first two messages provide cookies and establish an SA with agreed protocol andtransforms;bothsidesuseanoncetoensureagainstreplayattacks.Thelasttwomessages exchange the key material and user IDs, with an authentication mechanismusedtoauthenticatekeys,identities,andthenoncesfromthefirsttwo messages.

2. IdentityProtectionExchange:expandstheBaseExchangetoprotecttheusers'identities.

(b) Identity Protection Exchange

(1)**I**⟶**R**: SA	Begin ISAKMP-SA negotiation
(2)**R**⟶**E**: SA	Basic SA agreed upon
(3)**I**⟶**R**: KE; NONCE	Key generated
(4)**R**⟶**E**: KE; NONCE	Key generated
(5)*__I__⟶**R**: ID_I; AUTH	Initiator identity verified by responder
(6)*__R__⟶**E**: ID_R; AUTH	Responder identity verified by initiator; SA established

The first two messages establish the SA. The next two messages perform key exchange,with nonces for replay protection. Once the session key has been computed, the twoparties

exchange encrypted messages that contain authentication information, such as digitalsignaturesandoptionallycertificates validatingthe publickeys.

3. AuthenticationOnlyExchange:usedtoperformmutualauthentication,withoutakeyexchange

The first two messages establish the SA. In addition, the responder uses the secondmessage to convey its ID and uses authentication to protect the message. The initiatorsendsthethirdmessage totransmit itsauthenticatedID.

4. AggressiveExchange:minimizesthenumberofexchangesattheexpenseofnotprovidingiden tityprotection.

(d) Aggressive Exchange

(1)**I**⟶**R**: SA; KE; NONCE; ID_I;	Begin ISAKMP-SA negotiation and key exchange
(2)**R**⟶**E**: SA; KE; NONCE; ID_R; AUTH	Initiator identity verified by responder; Key generated; Basic SA agreed upon
(3)*I⟶**R**: AUTH	Responder identity verified by initiator; SA established

In the first message, the initiator proposes an SA with associated offered protocol andtransform options. The initiator also begins the key exchange and provides its ID. In thesecondmessage,theresponderindicatesits acceptanceoftheSAwithaparticularprotocol and transform, completes the key exchange, and authenticates the transmittedinformation. In the third message, the initiator transmits an authentication result thatcoversthepreviousinformation,encryptedusingthesharedsecretsessionkey.

5. InformationalExchange:usedforone-waytransmittalofinformationforSAmanagement.

(e) Informational Exchange

(1)*I⟶**R**: N/D	Error or status notification, or deletion