UNIT-1

1). INTRODUCTION TO INFORMATION SECURITY

This is the age of universal electronic connectivity, where the activities like hacking, viruses, electronic fraud are very common. Unless security measures are taken, a network conversation or a distributed application can be compromised easily.

Some examples are:

- Online purchases using a credit/debit card.
- A customer unknowingly being directed to a false website.
- A hackers ending a message to a person pretending to be some one else.

Network Security has been affected by two major developments over the last several decades. First one is introduction of computers into organizations and the second one being introduction of distributed systems and the use of networks and communication facilities for carrying data between users & computers. These two developments lead to 'computer security' and 'network security', where the computer security deals with collection of tools designed to protect data and to thwart hackers. Network security measures are needed to protect data during transmission. But keepin mind that, it is the information and our ability to access that information that we are really trying to protect and not the computers and networks.

2). WHY WE NEED INFORMATION SECURITY

Because there are threats.

Threat:A threat is an object, person, or other entity that represents a constant danger to an asset The 2007 CSI survey.

- 494computersecuritypractitioners
- 46%sufferedsecurityincidents
- 29%reported to law enforcement
- Averageannualloss$350,424
- 1/5suffered_targeted attack'
- The source of the greatest financial losses?
- Most prevalent security problem
- Insider abuse of network access
- Email

Threat Categories:

- Acts of human error or failure
- Compromises to intellectual property
- Deliberate acts of espionage or trespass
- Deliberate acts of information extortion
- Deliberate acts of sabotage or vandalism
- Deliberate act soft theft
- Deliberate software attack
- Forces of nature
- Deviations in quality of service
- Technical hardware failures or errors
- Technological obsolesce

3). SECURITY APPROACHES:

Definitions

Computer Security-generic name for the collection of tools designed to protect data and hackers

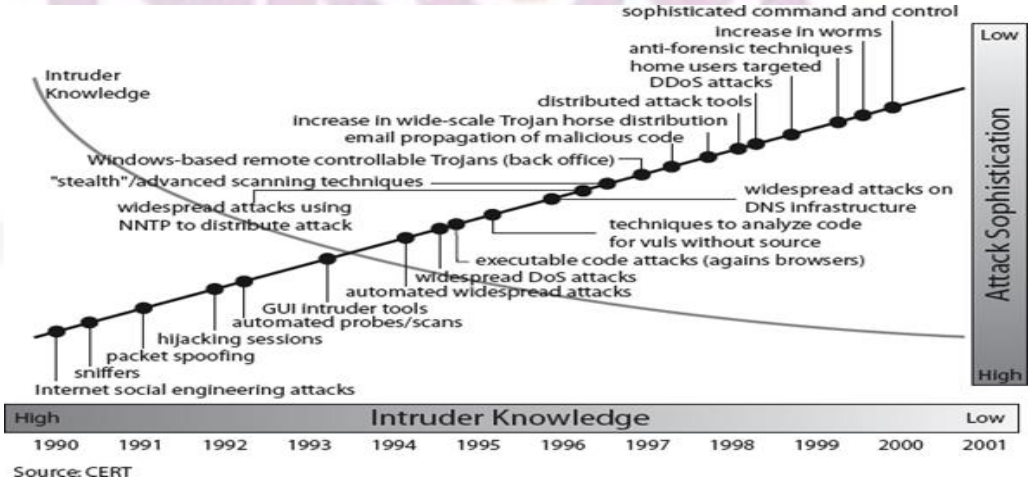
Network Security - measures to protect data during their transmission

Internet Security-measures to protect data during their transmission over a collection of interconnected networks

Our focus is on

Internet Security

Which consists of measures to deter, prevent, detect, and correct security violations that involve the transmission &storage of information



4.) PRINCIPLES OF SECURITY:

The CIA triad refers to an information security model made up of the three main components: confidentiality, integrity and availability. Each component represents a fundamental objective of information security.

ASPECTS OF SECURITY:

Consider the three aspects of information security:

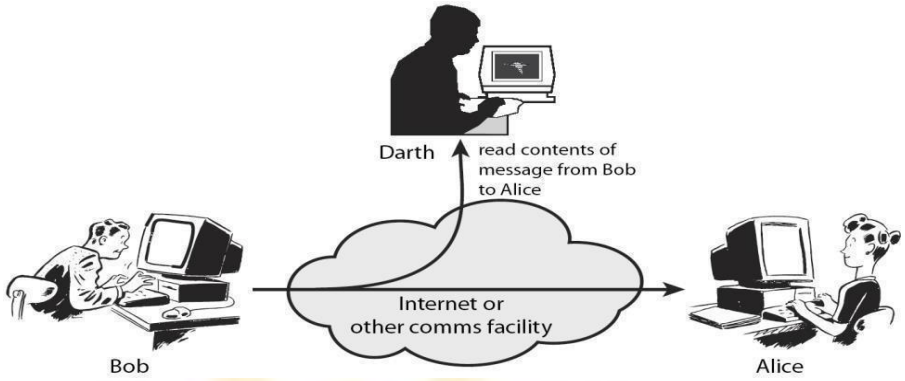
- Security attack
- Security mechanism
- Security service

TYPES OF SECURITY ATTACKS

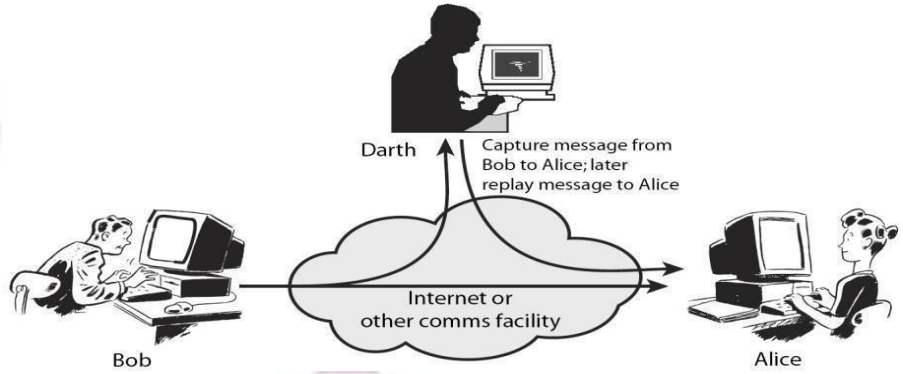
Any action that compromises the security of information owned by an organization.
Information security is about how to prevent attacks, or failing that, to detect attacks on information-based systems.
Often threats & attacks used to mean same thing have a wide range of attacks can focus on generic types of attacks.

- Passive
- Active

Passive Attack:



Active Attack:



INTERRUPTION

An asset of the system is destroyed or becomes unavailable or unusable. It is an attack on availability.

Examples:

- Destruction of some hardware
- Jamming wireless signals
- Disabling file management systems

INTERCEPTION

An unauthorized party gains access to an asset. Attack on confidentiality.

Examples:

- Eavesdropping
- Wire tapping to capture data in a network.
- Illicitly copying data or programs

MODIFICATION:

When an unauthorized party gains access and tampers an asset. Attack is on integrity.

Examples:

- Changing data file.
- Altering a program and the contents of a message.

FABRICATION:

An unauthorized party inserts a counterfit object into the system. Attack on authenticity also called impersonation.

Examples:

- Hackers gaining access to a personal email and sending messages.
- Insertion of records in data files.
- Insertion of spurious message in a network.

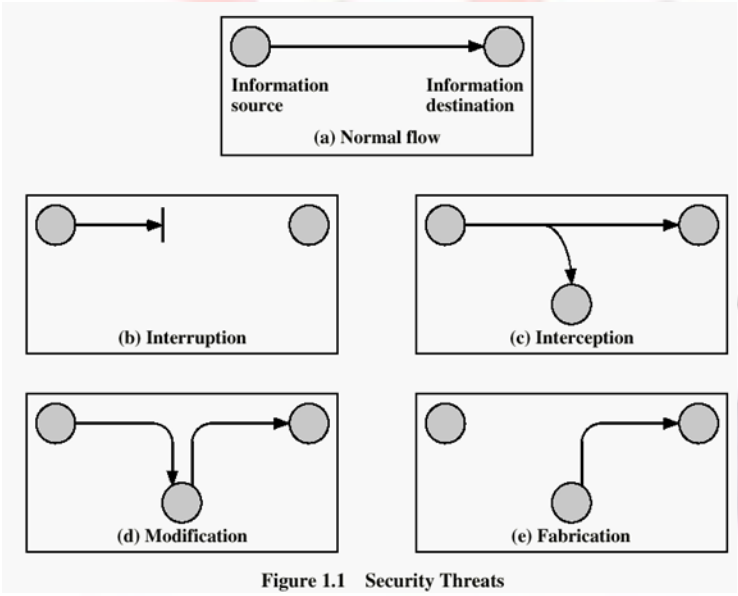


Figure 1.1 Security Threats

SECURITY SERVICES:

It is a processing or communication service that is provided by a systemtogiveaspecifickindofproductiontosystemresources.Securityservic esimplementsecuritypoliciesandareimplemented by security mechanisms.

CONFIDENTIALIY:

Confidentiality is the protection of transmitted data from passive attacks. It is used toprevent the disclosure of information to unauthorized individuals or systems. It hasbeendefinedas"ensuringthatinformationisaccessibleonlytothoseauthorizedtohaveaccess".The other aspect of confidentiality is the protection of traffic flow from analysis.**Ex:**Acredit card numberhastobesecuredduringonlinetransaction.

Authentication

This service assures that a communication is authentic. For a single messagetransmission, its function is to assure the recipient that the message is from intendedsource. For an ongoing interaction two aspects are involved. First, during connectioninitiation the service assures the authenticity of both parties. Second, the connectionbetween the two hosts is not interfered allowing a third party to masquerade as oneofthe twoparties.TwospecificauthenticationservicesdefinesinX.800are

Peer entity authentication: Verifies the identities of the peer entities involved incommunication. Provides use at time of connection establishment and during datatransmission.Providesconfidenceagainst a masqueradeora replayattack

Dataoriginauthentication: Assumestheauthenticityofsourceofdataunit,butdoesnotprovideprotectionagainstduplicationormodificationofdataunits. Supportsapplicationslikeelectronicmail,wherenopriorinteractionstakeplacebetweencommunicatingentities.

Integrity

Integritymeansthatdatacannotbemodifiedwithoutauthorization.Likeconfidentiality, it can be applied to a stream of messages, a single message or selectedfieldswithinamessage.Twotypesofintegrityservicesareavailable.Theyare

Connection-Oriented Integrity Service: This service deals with a stream ofmessages, assures that messages are received as sent, with no duplication, insertion,modification, reordering or replays. Destruction of data is also covered here. Hence, itattends toboth messagestreammodificationand denialofservice.

Connectionless-OrientedIntegrityService: Itdealswithindividualmessagesregardlessoflargercontext,providingprotectionagainstmessagemodificationonly.

An integrity service can be applied with orwithout recovery. Because it isrelated to active attacks, major concern will be detection rather than prevention. If aviolationisdetectedand theservice reports it,eitherhuman intervention orautomatedrecoverymachinesarerequiredtorecover.

Non-repudiation

Non-repudiationpreventseithersenderorreceiverfromdenyingatransmittedmessage.Thiscapabilityiscrucialtoe-commerce.Withoutitanindividualorentitycandenythathe,sheoritisresponsibleforatransaction,thereforenotfinanciallyliable.

AccessControl

Thisreferstotheabilitytocontrolthelevelofaccessthatindividualsorentities have to a network or system and how much information they can receive. It istheabilitytolimitandcontroltheaccesstohostsystemsandapplicationsviacommunicationlinks.Forthis,eachentitytryingtogain accessmustfirstbeidentifiedorauthenticated,sothat access rightscanbetailoredtotheindividuals.

Availability

Itisdefinedtobethepropertyofasystemorasystemresourcebeingaccessible and usable upon demand by an authorized system entity. The availability cansignificantly be affected by a variety of attacks, some amenable to automated countermeasures i.e authentication and encryption and others need some sort of physicalaction toprevent or recover from loss of availability of elements of a distributedsystem.

SECURITYMECHANISMS

AccordingtoX.800,thesecuritymechanismsaredividedintothoseimplementedinaspecificprotocollayerandthosethatarenotspecifictoanyparticular protocol layer or security service. X.800 also differentiates reversible &irreversibleenciphermentmechanisms.Areversibleenciphermentmechanismis simply an encryption algorithm that allows data to be encrypted and subsequentlydecrypted, whereas irreversible encipherment include hash algorithms and messageauthenticationcodesusedindigitalsignatureandmessageauthenticationapplications

SpecificSecurityMechanisms

Incorporatedinto

theappropriateprotocollayerinordertoprovidesomeoftheOSIsecurityservices,

Encipherment: It refers to the process of applying mathematical algorithms forconvertingdataintoaformthatisnotintelligible.Thisdependsonalgorithmuse dandencryptionkeys.

Digital Signature: The appended data or a cryptographic transformation applied toany data unit allowing to prove the source and integrity of the data unit and protectagainstforgery.

AccessControl: Avarietyoftechniquesusedforenforcingaccesspermissionstoth esystemresources.

DataIntegrity: Avarietyofmechanismsusedtoassuretheintegrityofadataunitor streamofdataunits.

AuthenticationExchange: Amechanismintendedtoensuretheidentityofanentit ybymeansofinformation exchange.

TrafficPadding: Theinsertionofbitsinto gapsinadatastreamtofrustratetrafficanalysisattempts.

RoutingControl: Enablesselectionofparticularphysicallysecureroutesforcertaind ataandallowsroutingchangesonce abreachofsecurityis suspected.

Notarization: Theuseofatrustedthirdpartytoassurecertainpropertiesofadataex change

PervasiveSecurityMechanisms

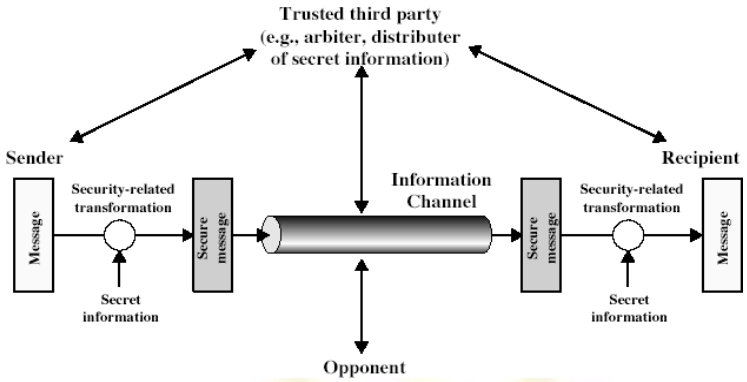
Thesearenotspecifictoanyparticular OSIsecurityserviceorprotocollayer.

Trusted Functionality: That which is perceived to b correct with respect to somecriteria **Security Level:** The marking bound to a resource (which may be a data unit)thatnamesordesignatesthesecurityattributesofthatresource.

Event Detection: It is the process of detecting all the events related tonetworksecurity. **Security AuditTrail**:Datacollectedandpotentiallyusedto facilitateasecurity audit, which is an independent review and examination of system recordsandactivities.

SecurityRecovery: Itdealswithrequestsfrommechanisms,suchaseventhandli ngand managementfunctions,and takesrecoveryactions.

MODELFORNETWORKSECURITY



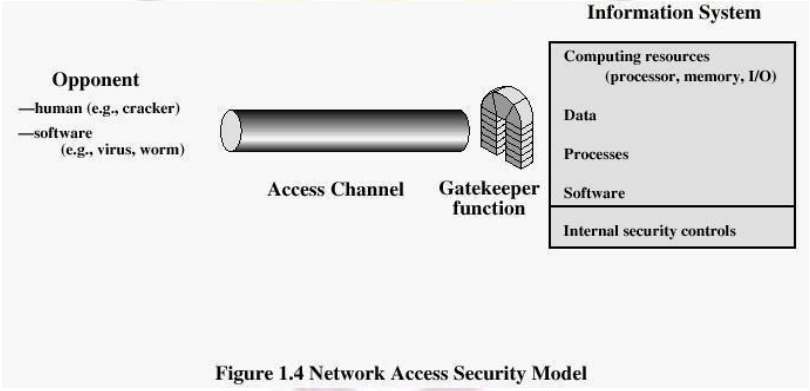


Figure 1.4 Network Access Security Model

Data is transmitted over network between two communicating parties, whomustcooperatefortheexchangetotakeplace.Alogicalinformationchannelise stablishedbydefiningaroutethroughtheinternetfromsourcetodestinationbyuseof communication protocols by the two parties. Whenever an opponent presents athreat to confidentiality, authenticity of information, security aspects come into

play.Twocomponentsarepresentinalmostallthesecurityprovidingtechniques.

A security-related transformation on the information to be sent making itunreadable by the opponent, and the addition of a code based on the contents of the message, used toverifytheidentityofsender.

Some secret information shared by the two principals and, it is hoped, unknowntotheopponent.Anexampleisanencryptionkeyusedinconjunctionwiththetr ansformationtoscramblethemessagebeforetransmissionandunscrambleitonreceptio n

A trusted thirdpartymay be neededto achievesecuretransmission. It isresponsiblefordistributingthesecret

informationtothetwoparties,whilekeepingitaway from any opponent. It also may be needed to settle disputes between the twoparties regarding authenticity of a message transmission. The general model showsthat

therearefourbasictasksindesigninga particularsecurityservice:

1. Designanalgorithmforperforming thesecurity-related transformation.Thealgorithmshouldbesuchthatanopponentcannotd efeatitspurpose
2. Generatethesecretinformationtobeusedwiththealgorithm
3. Developmethodsforthedistributionandsharingofthesecretinformation
4. Specify a protocol to be used by the two principals that makes use of the securityalgorithm and the secret information to achieve a particular security serviceVariousotherthreatstoinformationsystemlikeunwantedaccess stillexist.
5. The existenceofhackersattemptingtopenetratesystemsaccessibleoveranet workremains a concern. Another threat is placement of some logic in computer system affecting various applications and utility programs. This inserted code presents twokindsofthreats.

Informationaccessthreatsinterceptormodifydataonbehalfofuserswho shouldnothaveaccesstothatdata

Service threats exploit service flaws in computers to inhibit use by legitimateusers Viruses and worms are twoexamples of software attacks inserted into thesystem bymeans of a disk or alsoacross the network. The security mechanismsneededtocope with unwanted access fallintotwobroadcategories.

Somebasicterminologiesused

- **CIPHERTEXT**-thecodedmessage
- **CIPHER**-algorithmfortransformingplaintexttociphertext
- **KEY**-info usedincipherknownonlytosender/receiver
- **ENCIPHER(ENCRYPT)**-convertingplaintexttociphertext
- **DECIPHER(DECRYPT)**-recoveringciphertextfromplaintext
- **CRYPTOGRAPHY**-studyofencryptionprinciples/methods
- **CRYPTANALYSIS(CODEBREAKING)**-thestudyofprinciples/methodsofdecipheringciphertext*witho ut*knowingkey
- **CRYPTOLOGY**-thefieldofbothcryptographyandcryptanalysis

CRYPTOGRAPHY

Cryptographic systems are generally classified along 3

independentdimensions:

Typeofoperationsusedfortransformingplaintexttociphertext

Alltheencryptionalgorithmsareabasedontwogeneralprinciples:**substitution**, in which each element in the plaintext is mapped into another element,and **transposition**,inwhichelementsintheplaintextarerearranged.

Thenumberofkeysused

If the sender and receiver uses same key then it is said to be **symmetric key(or)singlekey(or)conventionalencryption**.Ifthesenderandreceiverusediff erentkeysthen itissaidtobe**publickeyencryption**.

Thewayinwhichtheplaintextisprocessed

A **block cipher** processes the input and block of elements at a time, producingoutput block for each input block. A **stream cipher** processes the input elementscontinuously,producingoutputelementoneat atime,asit goesalong.

CRYPTANALYSIS

The process of attempting to discover X or K or both is known as cryptanalysis.The strategy used by the cryptanalysis depends on the nature of the encryption schemeandtheinformationavailabletothecryptanalyst.**Therearevaroustypesofcry ptanalyticattacks**based onthe amount ofinformationknowntothe cryptanalyst.

Ciphertext only–Acopyofciphertextaloneisknowntothe cryptanalyst.

Known plaintext – The cryptanalyst has a copy of the cipher text and the correspondingplaintext.

Chosen plaintext – The cryptanalysts gains temporary access to the encryption machine.They cannot open it to find the key, however; they can encrypt a large number of suitablychosenplaintexts and trytousetheresultingciphertexts todeducethekey.

Chosen cipher text – The cryptanalyst obtains temporary access to the decryptionmachine, uses it to decrypt several string of symbols, and tries to use the results todeducethekey.

CLASSICALENCRYPTIONTECHNIQUES

Therearetwobasicbuildingblocksofallencryptiontechniques:substitutionandtranspositi on.

SUBSTITUTIONTECHNIQUES

A substitution technique is one in which the letters of plaintext are replaced by otherletters or by numbers or symbols. If the plaintext is viewed as a sequence of bits,

thensubstitutioninvolvesreplacingplaintextbitpatternswithciphertextbitpatterns
.

CAESARCIPHER

TheearliestknownuseofasubstitutioncipherandthesimplestwasbyJuliusCaesar.
The Caesar cipher involves replacing each letter of the alphabet with theletter standing 3 places further down the alphabet. e.g., plain text : pay more moneyCiphertext:SDBPRUHPRQHB
Notethatthealphabetiswrappedaround,sothatletterfollowing„z"is„a".
Foreachplaintextletterp,substitutetheciphertextlettercsuchthatC=E(p)=(p+ 3)mod26
AshiftmaybeanyamountsothatgeneralCaesaralgorithmisC =E(p) =(p+k) mod26 Wherek takeson avaluein therange1to25.

- ThedecryptionalgorithmissimplyP=D(C)=(C-k) mod26

MONOALPHABETICCIPHERS

Here, Plaintext characters are substituted by a different alphabet stream ofcharacters shifted to the right or left by n positions. When compared to the Caesarciphers,thesemonoalphabeticciphersaremoresecureaseachletteroftheciphertext can be any permutation of the 26 alphabetic characters leading to 26! orgreater than 4 x 10_{26} possible keys. But it is still vulnerable to cryptanalysis, when acryptanalyst is aware of the nature of the plaintext, he can find the regularities of thelanguage.Toovercome these attacks, multiple substitutions for a single letter are used. For example, a lettercan be substituted by different numerical cipher symbols such as 17, 54, 69….. etc.Even this method is not completely secure as each letter in the plain text affects onletterin theciphertext.Or,usingacommonkeywhichsubstituteseveryletteroftheplaintext.

Thekey*ABCDEFGHIIJKLMNOPQRSTUVWXYZQWERTYUIIOPASDFGHJKL ZXCVBNM*

Wouldencryptthemessage

*IIthinkthereforeIIam*into

OZIIOFAZIITKTYGKTOQD

Butanyattackerwouldsimplybreakthecipherbyusingfrequencyanalysisbyobser ving the number of times each letter occurs in the cipher text and then lookingupon the English letter frequency table. So, substitution cipher is completely
ruinedbytheseattacks.Monoalphabeticciphersareeasytobreakastheyreflectthefr equencyoftheoriginalalphabet.Acountermeasureistoprovidesubstitutes,known ashomophonesforasingleletter.

PLAYFAIR CIPHERS

Itisthebestknownmultiple – letterencryptioncipherwhichtreatsdigramsintheplaintextassingleunitsandtranslatestheseunitsintociphertextdigrams.ThePlayfairCipherisadigramsubstitutionciphero fferingarelativelyweakmethodof encryption. It was used for tactical purposes by British forces in the Second Boer Warand in World War I and for the same purpose by the Australians and Germans duringWorld War II. This was because Playfair is reasonably fast to use and requires nospecial equipment. A typical scenario for Playfair use would be to protect importantbut non-critical secrets during actual combat. By the time the enemy cryptanalystscould break the message, the information was useless to them. It is based around a5x5 matrix, a copy of which is held by both communicating parties, into which 25 ofthe 26 letters of the alphabet (normally either j and i are represented by the sameletterorxisignored)areplacedinarandomfashion.Forexample,theplaintextis *ShiSherryloves HeathLedger* andtheagreedkey is*sherry*. Thematrixwill bebuiltaccordingtothefollowingrules.

- inpairs,
- withoutpunctuation,
- AllJsarereplacedwithIs.

SHISHERRYLOVESHEATHLEDGER

- Doubleletterswhichoccurinapairmustbedividedbyan XoraZ.
- E.g.LITERALLYLITERALXLY

SH IS HE RX RY LO VE SH EA TH LE DG ER The alphabet square is prepared using,
a5*5matrix,norepetitionletters,noJsandkeyiswrittenfirstfollowedbytheremaining alphabetswithnoiandj.
SHERYAB CDFGIKLMNOPQTUVWXZ

Forthegenerationofciphertext,therearethreerulestobefollowedbyeachpairofletters.lettersappearonthesamerow：replacethemwiththelettersto heirimmediateright respectivelylettersappearonthesamecolumn：replacethemwiththelettersimmediatelybelowrespectively not on the same row or column：replace them with the letters on the same rowrespectivelybutattheotherpairof cornersoftherectangledefinedby
theoriginalpair.Basedontheabovethreerules,theciphertextobtainedforthegivenplaintextis

HEGHER DRYSIQWHHESCOYKRALRY

Anotherexamplewhichissimplerthantheaboveonecanbegivenas:

Here,keywordis*playfair*.Plaintext is*Hellotherehellothere* becomes ---------- *he lxlother*

ex.Applyingtherulesagain,foreachpair,Iftheyareinthesamerow,replaceeachwiththelette rtoitsright(mod5)

he KG

Ifthey areinthesamecolumn, replaceeachwiththeletter belowit(mod5)

lo RV

Otherwise,replaceeachwithletterwe'dgetifweswappedtheircolumnindices

lx YV

Sotheciphertextfor thegivenplaintextis**KGYVRVQMGIKU**

$$\begin{bmatrix} p & l & a & y & f \\ i & r & b & c & d \\ e & g & h & k & m \\ n & o & q & s & t \\ u & v & w & x & z \end{bmatrix}$$

To decrypt the message, just reverse the process. Shift up and left instead of downandright.Dropextrax'sandlocateanymissingI'sthatshouldbej's.Themessa gewillbe back into the original readable form. no longer used by military forces because ofthe advent of digital encryption devices. Playfair is now regarded as insecure for anypurpose because modern hand-held computers could easily break the cipher withinseconds.

HILLCIPHER

Itisalsoamultiletterencryptioncipher.Itinvolvessubstitutionof'*m*'ciphertext letters for '*m'* successive plaintext letters. For substitution purposes using'*m'* linear equations, each of the characters are assigned a numerical values i.e. a=0,b=1,c=2,d=3,…….z=25.Forexampleifm=3,thesystemcanbedefinedas:$\mathbf{c_1=(k_{11}p_1+k_{12}p_2+k_{13}p_3)\ mod\ 26\ c_2=(k_{21}p_1+k_{22}p_2+k_{23}p_3)\ mod\ 26\ c_3=(k_{31}p_1+k_{32}p_2+k_{33}p_3)\ mod\ 26}$ If we represent in matrix form, the above statements as matrices andcolumn vectors:

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} \quad mod\ 26$$

Thus, C = KP *mod26*, where C= Column vectors of length 3 P= Column vectors of length 3K=3x3 encryptionkeymatrix.Fordecryptionprocess,inverse ofmatrix**K** i.e. $\mathbf{K}_{-1}$is required which is defined by the equation $KK_{-1} = K_{-1}K = I$, where Iistheidentitymatrixthatcontainsonly0'sand1'sasitselements.Plaintextisrecovered byapplyingK_{-1} totheciphertext.Itisexpressedas$C = E_K(P)=KP$ *mod26*$P=D_K(C)=K_{-1}C$*mod26.*$=K_{-1}KP=IP=P$

Example: The plain text is I can't do it and the size of m is 3 and key K is chosen asfollowing

I can't do it

8 2 0 13 19 3 14 8 19

$$\begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix}$$

The encryption process is carried out as follows

$$\begin{pmatrix} 4 \\ 14 \\ 12 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix} \begin{pmatrix} 8 \\ 2 \\ 0 \end{pmatrix} \pmod{26}$$

$$\begin{pmatrix} 19 \\ 12 \\ 14 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix} \begin{pmatrix} 13 \\ 19 \\ 3 \end{pmatrix} \pmod{26}$$

$$\begin{pmatrix} 18 \\ 21 \\ 9 \end{pmatrix} = \begin{pmatrix} 9 & 18 & 10 \\ 16 & 21 & 1 \\ 5 & 12 & 23 \end{pmatrix} \begin{pmatrix} 14 \\ 8 \\ 19 \end{pmatrix} \pmod{26}$$

So, the encrypted text will be given as → **EOM TMY SVJ**

Themainadvantagesofhillcipheraregivenbelow:perfectlyhidessingle-letterfrequencies.

ItUseof**3x3**Hillcipherscanperfectly hideboththeSingleletterandtwo-letterfrequencyinformation.

Strongenoughagainsttheattacksmadeonlyontheciphertext.

But,itstillcanbeeasily brokeniftheattackisthroughaknownplaintext.

POLYALPHABETICCIPHERS

In order to make substitution ciphers more secure, more than one alphabet can beused. Such ciphers are called **polyalphabetic**, which means that the same letter of amessage can be represented by different letters when encoded. Such a one-to-manycorrespondence makes the use of frequency analysis much more difficult in order tocrack the code. We describe one such cipher named for *Blaise de Vigenere* a 16-thcentury Frenchman. The **Vigenere cipher** is a polyalphabetic cipher based on usingsuccessively shifted alphabets, a different shifted alphabet for each of the 26 Englishletters. The procedure is based on the tableau shown below and the use of a keyword.The letters of the keyword determine the shifted alphabets used in the encodingprocess.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

For the message COMPUTING GIVES INSIGHT and keyword LUCKY we proceed

byrepeatingthekeywordasmanytimesasneededabovethemessage,asfollows.

L	U	C	K	Y	L	U	C	K	Y	L	U	C	K	Y	L	U	C	K	Y	L
C	O	M	P	U	T	I	N	G	G	I	V	E	S	I	N	S	I	G	H	T

Encryption is simple: Given a key letter x and a plaintext letter y, the ciphertext letteris at the intersection of the row labeled *x* and the column labeled *y*; so for L, theciphertext letter would be N. So, the ciphertext for the given plaintext would be givenas:

L	U	C	K	Y	L	U	C	K	Y	L	U	C	K	Y	L	U	C	K	Y	L				
C	O	M	P	U	T	I	N	G	G	I	V	E	S	I	N	S	I	G	H	T				<==MESSAGE
N	I	O	Z	S	E	C	P	Q	E	T	P	G	C	G	Y	M	K	Q	F	E				<==Encoded Message

Decryptionisequallysimple:Thekeyletteragainidentifiestherowandposition of ciphertext letter in that row decides the column and the plaintext letter isat thetopofthatcolumn.Thestrengthof thiscipheristhatthere aremultipleciphetext letters for each plaintext letter, one for each unique letter of the keywordand thereby making the letter frequency information is obscured. Still, breaking thiscipher has been made possible because this reveals some mathematical principlesthat apply in cryptanalysis. To overcome the drawback of the periodic nature of thekeyword, a new technique is proposed which is referred as an autokey system, inwhich a key word is concatenated with the plaintext itself to provide a running key.ForexIntheaboveexample,thekeywouldbe*luckycomputinggivesin*Still,thiss cheme is vulnerable to cryptanalysis as both the key and plaintext share the samefrequency distribution of letters allowing a statistical technique to be applied. Thus,the ultimate defense against such a cryptanalysisis tochoosea keyword thatis as longas plaintext and has no statistical relationship to it. A new system which works onbinarydataratherthanlettersisgivenas
$\mathbf{C_i = p_i k_i}$where,p_i=ithbinarydigitofplaintextk_i=ithbinarydigitofkeyC_i=ith binarydigitofciphertext==exclusive-oroperation.BecauseofthepropertiesofXOR,decryptionisdonebyperformingthesamebit wiseoperation.
$\mathbf{p_i = C_i k_i}$Averylongbut,repeationkeywordisusedmakingcryptanalysisdifficult.

TRANSPOSITION TECHNIQUES

Allthetechniquesexaminedsofarinvolvethesubstitutionofaciphertextsym bolfora plaintext symbol. A very different kind of mapping is achieved by performing somesortofpermutationontheplaintextletters.Thistechniqueisreferredtoas atranspositioncipher.

Rail fenceis simplestofsuchcipher,inwhichtheplaintextiswritten downasasequenceofdiagonalsandthenreadoffasasequenceofrows.
Plaintext=meetattheschoolhouse

Toencipherthismessagewitharailfenceofdepth2,
Wewritethemessageasfollows:meatecolosetthshohue
TheencryptedmessageisMEATECOLOSETTHSHOHUE

Row Transposition Ciphers-A more complex scheme is to write the message in arectangle, row by row, and read the message off, column by column, but permute theorderofthecolumns.Theorderofcolumnsthenbecomesthekeyofthealgorithm.

.,plaintext=meetattheschoolhouseKey=4312567
PT=meeta tt heschool houseCT=ESOTCUEEHMHLAHSTOETO
Apuretranspositioncipheriseasilyrecognizedbecauseithasthesameletterfrequenciesastheoriginalplaintext.Thetranspositionciphercanbemadesignificantly more secure by performing more than one stage of transposition. Theresult is morecomplexpermutationthatis not easilyreconstructed.

ENCRYPTION AND DECRYPTION:

There are various reasons for using these processes, but a few important reasons are:

It helps protect confidential and sensitive data like passwords or other credentials.

- It helps maintain the integrity of users' data, as data won't constantly change and can be frequently verified.
- Helps to ensure that an attacker or third party doesn't alter the files or documents sent with the message.
- Prevents plagiarism and protects IP address.
- Helpful for proper network communication where an attacker can't access encrypted data.
- Helps protect PII/PHI data and securely explore the internet while maintaining compliant standards.

SYMMETRIC AND ASYMMETRIC KEY CRYPTOGRAPHY:

Symmetric Key Cryptography

In symmetric key cryptography, an individual key is used for both encryption and decryption. The sender needs the key to encrypt the plaintext and sends the cipher document to the receiver. The receiver used the similar key (or ruleset) to decrypt the message and recover the plaintext. Because an individual key is used for both functions, symmetric key cryptography is also known as symmetric encryption.

Symmetric key cryptography schemes are usually categorized such as stream ciphers or block ciphers. Stream ciphers works on a single bit (byte or computer word) at a time and execute some form of feedback structure so that the key is constantly changing.

Asymmetric cryptography

Asymmetric cryptography uses two keys for encryption and decryption. It depends on the technique of public and private keys. A public key, which is interchanged between higher than one user. Data is decrypted by a private key, which is not transformed. It is slower but more secure. The public key used in this encryption technique is applicable to everyone, but the private key used in it is not revealed

In asymmetric encryption, a message that is encrypted utilizing a public key can be decrypted by a private key, while if the message is encrypted by a private key can be decrypted by utilizing the public key. Asymmetric encryption is broadly used in dayto-day communication channels, particularly on the internet.

STEGANOGRAPHY

Aplaintextmessagemaybehiddeninanyoneofthetwoways.Themethodsofstegan ographyconcealtheexistenceofthemessage,whereasthemethodsofcryptography renderthemessageunintelligibletooutsidersbyvarioustransformations of the text. A simple form of steganography, but one that is timeconsuming to construct is one in which an arrangement of words or letters within anapparently innocuous text spells out the real message. e.g., (i) the sequence of firstletters of each word of the overall message spells out the real (hidden) message. (ii)Subset of the words of the overall message is used to convey the hidden

message.Variousothertechniqueshavebeenusedhistorically,someofthemare

- **Charactermarking**– selectedlettersofprintedortypewrittentextareoverwritten in pencil. The marks are ordinarily not visible unless the paper isheldtoanangleto brightlight.
- **Invisible ink** – a number of substances can be used for writing but leave novisibletrace untilheat orsome chemicalis appliedtothepaper.
- **Pin punctures** – small pin punctures on selected letters are ordinarily notvisibleunlessthepaperisheldinfront ofthelight.
- **Typewritten correction ribbon** – used between the lines typed with a blackribbon, the results of typing with the correction tape are visible only under astronglight.

DrawbacksofSteganography

- Requiresalotofoverheadto hidearelativelyfewbitsof information.
- Oncethesystemisdiscovered,itbecomesvirtuallyworthless.

key range and key size, possible types of attacks:

The concept of key range and key-size are related to each other. Key Range is total number of keys from smallest to largest available key. An attacker usually is armed with the knowledge of the cryptographic algorithm and the encrypted message, so only the actual key value remains the challenge for the attacker.

• If the key is found, the attacker can get original plaintext message. In the brute force attack, every possible key in the key-range is tried, until we get the right key.

• In the best case, the right key is found in the first attempt, in the worst case, the key is found in the last attempt. On an average, the right key is found after trying half of the possible keys in the key-range. Therefore by expanding the key range to a large extent, longer it will take for an attacker to find the key using brute-force attack.

• The concept of key range leads to the principle of key size. The strength of a cryptographic key is measured with the key size

• Key size is measured in bits and is represented using binary number system. Thus if the key range from 0 to 8, then the key size is 3 bits or in other words we can say if the size is bits then the key range is 0 to 256. Key size may be varying, depending upon the applications and the cryptographic algorithm being used, it can be 40 bits, 56 bits, 128 bits & so on. In order to protect the cipher-text against the brute-force attack, the key-size should be such that the attacker can not crack it within a specified amount of time.

From a practical viewpoint, a 40-bit key takes about 3 hours to crack, however a 41-bit key would take 6 hours and 42-bit key would take 12 hours & so on. This means every additional bit doubles the amount of time required to crack the key. We can assume that 128 bit key is quite safe, considering the capabilities of today's computers.

DescriptiveQuestions:

(a) 2MarksQuestions

1. Definesecurityattack,securitymechanismandsecurityservices.

Securityattack:anyactionthatcompromisesthesecurityofinformationownedbyanorganization.

Securitymechanism:amechanismthatisdesignedtodetect,preventorrecoverfromasecurityattack.

Securityservices:aservicethatenhancesthesecurityofthedataprocessingsystemsandtheinformationtransfersofanorganization.

2. Mentionthedifferenttypesofsecurityservices.

- Authentication
- Confidentiality
- Dataintegrity
- Nonrepudiation
- Accesscontrol
- Availability

3. Definepassiveattackandactiveattack.

Passiveattacksareinthenatureofeavesdropping,ormonitoringoftransmissions.Thetypesofpassiveattack are

- Releaseofmessagecontent
- Trafficanalysis

Activeattacksinvolvesomemodificationofdatastreamorcreationofafalsestream.Thetypesofactiveattackare

- Masquerade
- Replay
- Modification
- Denialofservice

4. Definethefollowingterms:

Plaintext:theoriginalmessagetobetransmitted.

Ciphertext:thecoded(encrypted)messageorthescrambledmessage.Encryption/ Enciphering :process ofconverting plain texttociphertext.Decryption/Deciphering:processofconvertingciphertext toplaintext.

5. Whatarethetwobasicfunctionsusedinencryptionalgorithms?

Thetwobasicfunctionsusedinencryptionalgorithmsare

- Substitution
- Transposition

6. DefineThreatandattack.

Threatisapossibledangerthatmightexploitavulnerabilitytobreachsecurityandthuscausepossibleharm.

Attackis anyattempttodestroy,expose,alter,disable,stealorgainunauthorizedaccesstoormakeunauthorizeduseofanasset

7. Whatarethetwoapproachestoattackingacipher?

Thetwoapproachestoattackacipher are:

1. Cryptanalysis
2. Brute-forceattack

8. DefineBrute-forceattack.

Theattacker trieseverypossiblekeyonapieceofciphertextuntilanintelligibletranslation into plaintext is obtained. On average, half of all possible keys must be tried to achievesuccess.

9. WhatisModificationofmessages

Modificationofmessagessimplymeansthatsomeportionofa legitimatemessageisaltered,orthat

messagesaredelayedorreordered,toproduceanunauthorizedeffect.

10. Whatismasquerade?

A **masquerade** takes place when one entity pretends to be a different entity. For example,authentication sequences can be captured and replayed after a valid authentication sequencehas taken place, thus enabling an authorized entity with few privileges toobtain extraprivilegesbyimpersonatinganentitythathasthoseprivileges.

11. WhatisReply?

Replay involves the passive capture of a data unit and its subsequent retransmission toproduceanunauthorizedeffect.

12. DefineDenialofservice.

Prevents or inhibits the normal use or management of communication facilities. Anotherform of service denial is the disruption of an entire network, either by disabling the networkoroverloadingitwithmessagessoastodegradeperformance.

13. Definethetwobasicbuildingblocksofencryptiontechniques.

- Substitutiontechnique–itisoneinwhichthelettersof theplaintextarereplacedbyotherlettersorbynumbersorsymbols.
- Transpositiontechnique– itisonewhichperformssomesortofpermutationontheplaintextletters.

14. Listoutthecomponentsofencryptionalgorithm.

- Plaintext
- Encryptionalgorithm
- Secretkey
- Ciphertext
- Decryptionalgorithm

15.Mentionfewmono-alphabeticandpoly-alphabeticciphers.

Mono-alphabeticciphers:-playfaircipher,hillcipher,CaesarcipherPoly-alphabeticciphers:-vigenerecipher,onetimepadcipher

16. Whatissteganography?Mentionfewtechniquesinit.

Steganographyisatechniqueforhidingtheoriginalmessage.Someoftherelatedtechniquesare

- Charactermarking
- Invisibleink
- Pinpunctures
- Typewritercorrectionribbon

17. MentionthefunctionsinvolvedinsimplifiedDES.

- Initialpermutation
- AcomplexfunctionF$_k$withakeyk1
- Switching
- AcomplexfunctionF$_k$withakeyk2
- Inversepermutation

18. Definestreamcipherandblockcipher.

Astreamcipherisonethatencryptsa digitaldata streamonebitoronebyteatatime.

Ablockcipherisoneinwhichablockofplaintextistreatedasawholeandusedtoproduceacipher textblock ofequalblock

19. Comparestreamcipherwithblockcipherwithexample.

Streamcipher: Processestheinputstreamcontinuouslyandproducingoneelementatati me.Example:Caesercipher.

Blockcipher: Processesthe inputoneblockofelementsata timeproducinganoutputblockforeachinputblock.Example:DES

20. WhataretheoperationsusedinAES?

- Substitutebytes
- ShiftRows
- MixColumns
- AddRoundKey

21. ConverttheGivenText"CRYPTOGRAPHY"intociphertextusingRa ilfenceTechnique.

Inrailfencetechniquetheplaintextiswrittendownasasequenceofdiagonalsandthenreadoffa sasequenceofrows.

CYTGAH RPORPY

TheciphertextisCYTGAHRPORPY.

22. Whataretheattacksthatcanbeperformedinthenetworks?

- Disclosure
- Trafficanalysis
- Masquerade
- Contentmodification
- Sequencemodification
- Timingmodification
- Sourcerepudiation
- Destinationrepudiation

23. WhatprimitiveoperationsareusedinRC4

- Keyexpansion

- Encryption
- Decryption

(b) 10MarksQuestions

1. ExplainindetailaboutOSIsecurityarchitecture.

2. A)ExplainthestepsinvolvedinRC4.
B)Discussdifferentblockciphermodesofoperation

3. A)How NISTdefines computersecurity? Give examples of recentcomputersecurityattackswhich you know.
B) AES consists of four functions in three layers. Which of the functions are primarily forconfusionandwhichareprimarilyfordiffusion?Whichofthelayersareforconfusion and whicharefordiffusion?Justifyyouranswers.OrExplainindetailabout AES

4. Drawamatrixthatshowstherelationshipbetweensecuritymechanismsandattacks.

5. A)Writeshortnotesonsecuritymechanisms.
B) Explainindetailaboutthe stepsinvolvedinDES.

6. Explainthefollowingsubstitutiontechniquesinnetworksecurity.
A) Caesar cipherB) Play fair cipherc) Polyalphabetic cipherOr
Explainaboutsubstitutionciphersindetailwithanexample.

7. A)whatismeantbysecurityattack?Explainvarioustypesofsecurityattacks.
B) Whatarethegoalsofsecurity?ExplainindetailaboutsecurityServices?

8. Explain classical encryption techniques (Steps involved in each encryption technique likeCaesarcipher,playfaircipher,hillcipher, vigenerecipher,onetimepadcipher,railfence,etc)

9. A)Explainaboutsteganography,transpositioncipher.
B)Explainanythreesubstitutionciphers with anexample

UNIT-2

Block Cipher Principles

Block ciphers are built in the Feistel cipher structure. Block cipher has a specific number of rounds and keys for generating ciphertext. For defining the complexity level of an algorithm few design principles are to be considered.

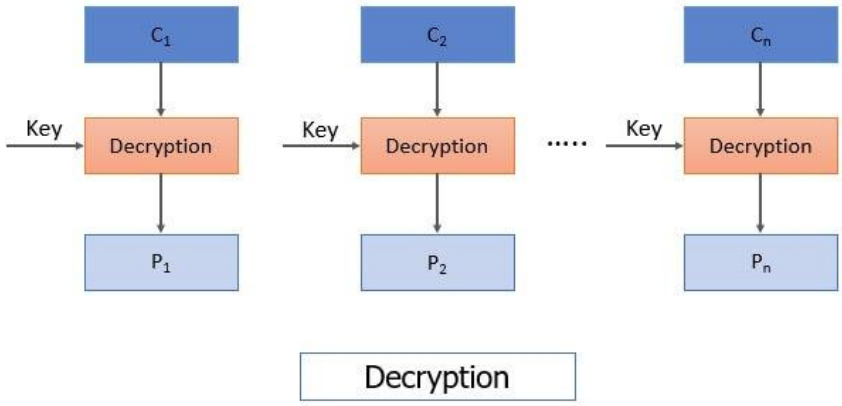
These are explained as following below:

1. Number of Rounds –
   The number of Rounds is regularly considered in design criteria, it just reflects the number of rounds to be suitable for an algorithm to make it more complex, in DES we have 16 rounds ensuring it to be more secure while in AES we have 10 rounds which makes it more secure.

2. Design of function F –
   The core part of the Feistel Block cipher structure is the Round Function. The complexity of cryptanalysis can be derived from the Round function i.e. the increasing level of complexity for the round function would be greatly contributing to an increase in complexity.

To increase the complexity of the round function, the avalanche effect is also included in the round function, as the change of a single bit in plain text would produce a mischievous output due to the presence of avalanche effect.

3. Key schedule algorithm –
   In Feistel Block cipher structure, each round would generate a sub-key for increasing the complexity of cryptanalysis. The Avalanche effect makes it more complex in deriving sub-key. Decryption must be done very carefully to get the actual output as the avalanche effect is present in it.

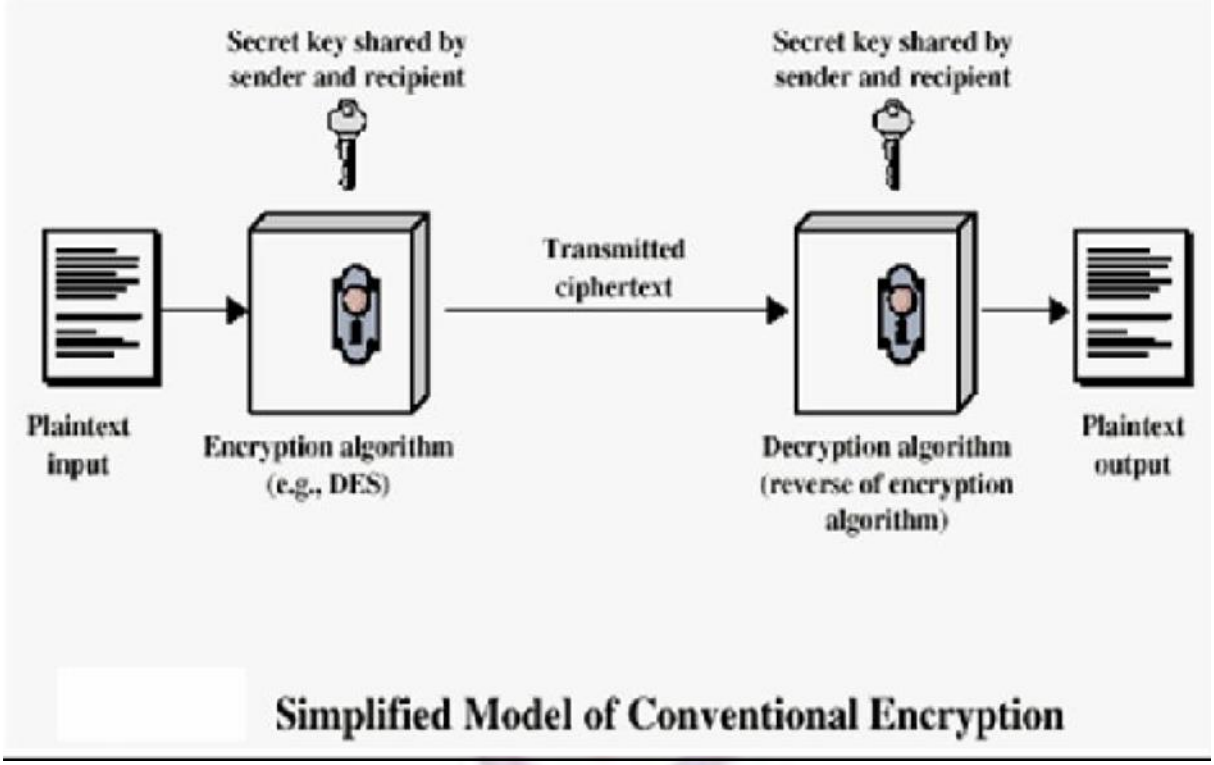


CONVENTIONALENCRYPTIONPRINCIPLES

AConventional/Symmetricencryptionschemehasfiveingredients

1. ***PlainText***:Thisistheoriginalmessageordatawhichis fedintothealgorithmas input.

2. ***Encryption Algorithm***: This encryption algorithm performs various substitutions andtransformations on theplaintext.
3. ***SecretKey***:Thekeyisanotherinputtothealgorithm.Thesubstituti onsandtransformations performedbyalgorithmdependonthekey.
4. ***CipherText*:**Thisisthescrambled(unreadable)messagewhichisoutputofthee ncryption algorithm. This cipher text is dependent on plaintext and secret key. For agivenplaintext,twodifferentkeys producetwodifferentciphertexts.
5. ***Decryption Algorithm***: This is the reverse of encryption algorithm. It takes the ciphertext andsecretkeyasinputsand outputstheplaintext.



Simplified Model of Conventional Encryption

The important point is that the security of conventional encryption depends on thesecrecy of the key, not the secrecy of the algorithm i.e. it is not necessary to keep thealgorithmsecret,butonlythekeyistobekeptsecret. Thisfeaturethatalgorithmneednot be kept secret made it feasible for wide spread use and enabled manufacturersdevelop low cost chip implementation of data encryption algorithms. With the use ofconventional algorithm, the principal security problem is maintaining the secrecy of thekey.

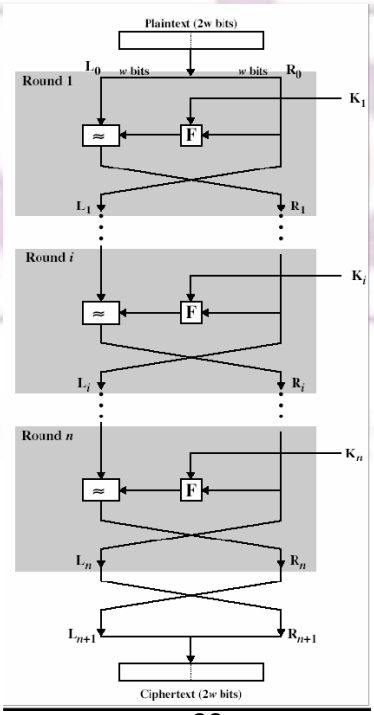
FEISTELCIPHERSTRUCTURE

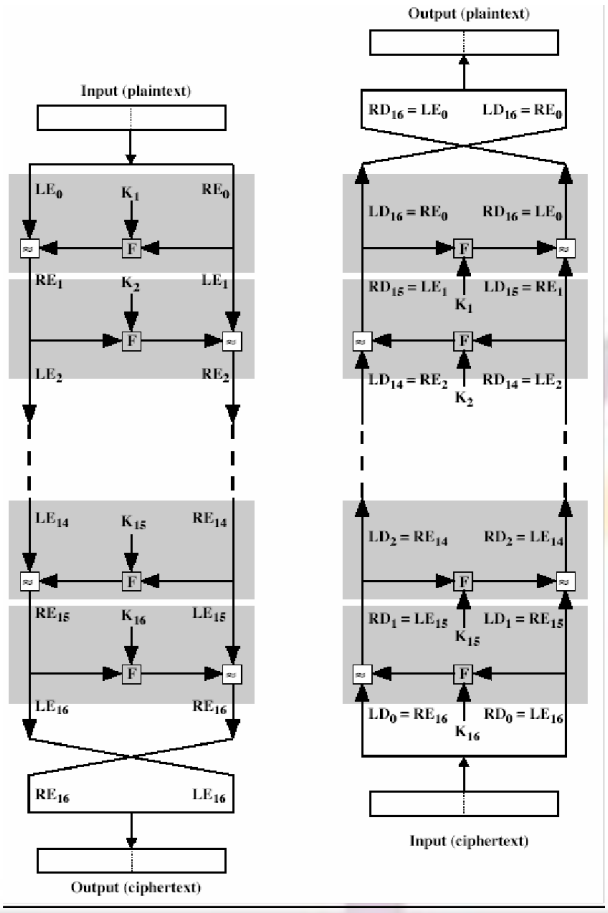
The input to the encryption algorithm are a plaintext block of length 2w bitsanda keyK.the plaintext block is divided intotwohalves L_0 and R_0.The twohalves ofthe data pass through „n" rounds of processing and then combine to produce theciphertext block. Each round „i" has inputs Li-1 and R_{i-1}, derived from the previousround,aswellasthesubkeyK_i,derivedfromtheoverallkeyK.ingeneral,thesubkey s$K_i$aredifferentfrom Kandfrom eachother.

Allroundshavethesamestructure.A substitutionisperformedonthelefthalfofthedata(assimilartoS-DES).ThisisdonebyapplyingaroundfunctionFtotherighthalfof the data and

then taking the XOR of the output of that function and the left half ofthe data. The round function has the same general structure for each round but isparameterized by the round subkey k_i. Following this substitution, a permutation isperformedthatconsistsoftheinterchangeofthetwohalvesofthedata.Thisstructureis a particular form of the substitution-permutation network. The exact realization ofa Feistelnetwork depends on the choice of the following parameters anddesignfeatures:

- **Blocksize**-Increasingsizeimprovessecurity,butslowscipher
- **Keysize**-
  Increasingsizeimprovessecurity,makesexhaustivekeysearching harder,butmayslowcipher
- **Numberofrounds**-Increasingnumberimprovessecurity,butslowscipher
- **Subkeygeneration**-
  Greatercomplexitycanmakeanalysisharder,butslowscipher
- **Round function** -Greatercomplexity canmakeanalysisharder, but slowscipher
- **Fastsoftwareen/decryption&easeofanalysis**-
  aremorerecentconcernsforpracticaluseandtesting





The process of decryption is essentially the same as the encryption process. The ruleis as follows: use the cipher text as input to the algorithm, but use the subkey k_i inreverse order.i.e.,k_n in the first round,k_{n-1} in second round and soon.For clarity, weuse the notation LE_i and RE_i for data traveling through the decryption algorithm. Thediagram below indicates that, at each round, the intermediate value of the decryptionprocess is same (equal) to the corresponding value of the encryption process with twohalvesofthevalueswapped..,$RE_i||LE_i$(or)equivalently$RD_{16-i}||LD_{16-i}$

After the last iteration of the encryption process, the two halves of the output areswapped, so that the cipher text is $RE_{16} || LE_{16}$. The output of that round is the ciphertext. Now take the cipher text and use it as input to the same algorithm. The input tothe first round is $RE_{16} || LE_{16,}$ which is equal to the 32-bit swap of the output of thesixteenthroundoftheencryptionprocess. Nowwewillseehowtheoutputofthefirstround of the decryption process is equal to a 32-bit swap of the input to the sixteenthroundoftheencryption process.

Firstconsidertheencryptionprocess,$LE_{16}=RE_{15}$

RE16=LE15(+)F(RE15,K16)Onthedecryptionside,$LD_1=RD_0=LE_{16}=RE_{15}RD_1=LD_{0(+)}F(RD_0,K_{16})=RE_{16}F(RE_{15},K_{16})=[LE_{15}F(RE_{15},K_{16})]F(RE_{15},K_{16})=LE_{15}$

Therefore,$LD_1=RE_{15}RD_1=LE_{15}$Ingeneral,forthei$_{th}$iterationoftheencryptionalgorithm,$LE_i=RE_{i-1}RE_i=LE_{i-1}F(RE_{i-1},K_i)$
Finally,theoutputofthelastroundofthedecryptionprocessis$RE_0||LE_0$.A32-bitswaprecoverstheoriginalplaintext.

DEFINITIONS

Encryption:Convertingatextintocodeorcipher.
Convertingcomputerdataandmessagesintosomething,incomprehensibleuseakey,sothatonlyaholderofthematchingkeycanreconvertthem.

ConventionalorSymmetricorSecretKeyorSingleKeyencryption:

Usesthesamekeyforencryption&decryption.

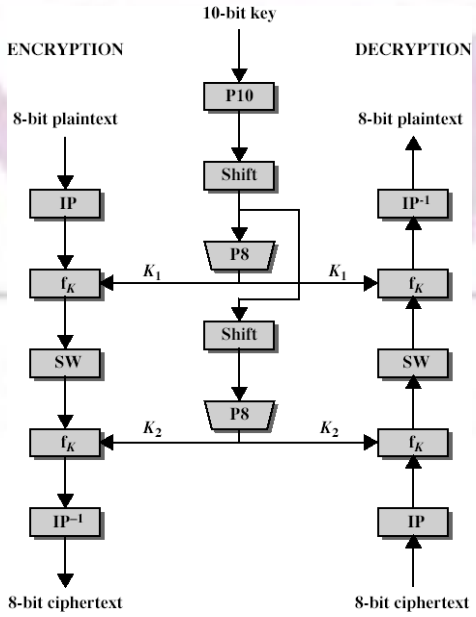
PublicKeyencryption: Usesdifferentkeysforencryption&decryption

ConventionalEncryption Principles

- Anencryptionschemehasfiveingredients:

1. Plaintext–Originalmessageordata.
2. Encryptionalgorithm–performssubstitutions&transformationsonplaintext.
3. SecretKey–exactsubstitutions&transformationsdependonthis
4. Ciphertext-outputiescrambledinput.
5. Decryptionalgorithm-convertsciphertextbacktoplaintext.

SIMPLIFIED DATA ENCRYPTION STANDARD (S-DES)



The figure above illustrates the overall structure of the simplified DES. The

S-DESencryptionalgorithmtakesan8-bitblockofplaintext(example:10111101)anda 10-bit key as input and produces an 8-bit block of ciphertext as output. The S-DESdecryption algorithm takes an 8-bit block of ciphertext and the same 10-bit key usedtoproducethatciphertextasinputandproducestheoriginal8-bitblockofplaintext.

Theencryptionalgorithminvolvesfivefunctions:

- aninitialpermutation(IP)
- acomplexfunctionlabeledf_k,whichinvolvesbothpermutation andsubstitutionoperationsanddependsonakeyinput
- asimplepermutationfunctionthatswitches(SW)thetwohalvesofthedata
- thefunction$f_k$again
- apermutationfunctionthatistheinverseoftheinitialpermutation

Thefunctionf_k takesasinputnotonlythedatapassingthroughtheencryptionalgorithm, but alsoan 8-bit key. Here a 10-bit key is used from which two 8-bitsubkeys are generated. The key is first subjected to a permutation (P10). Then a shiftoperationisperformed.The output ofthe shiftoperationthenpassesthrough apermutationfunctionthatproducesan8-bitoutput(P8) forthefirstsubkey(K1).Theoutput of the shift operation also feeds into another shift and another instance of P8toproducethesecondsubkey(K2).

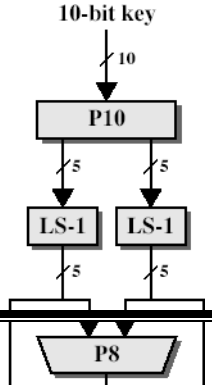
Theencryptionalgorithmcanbeexpressedasacompositioncomposition$_1$of functions:IP^{-1}ofK2oSWofk1oIP Whichcanalsobewrittenas

Ciphertext=$IP^{-1}(f_{K2}(SW(f_{k1}(IP(plaintext)))))$

K1=P8(Shift(P10(Key)))

K2=P8(Shift(shift(P10(Key))))Decryptioncanbeshownas

Plaintext=$IP^{-1}(f_{K1}(SW(f_{k2}(IP(ciphertext)))))$



S-DESdependsontheuseofa10-bitkeysharedbetweensenderandreceiver. From this key, two 8-bit subkeys are produced for use in particular stages of theencryptionanddecryptionalgorithm. First,permutethe key in the followingfashion.Let the 10-bit key be designated as (k1, K2, k3, k4, k5, k6, k7, k8, k9, k10). Then thepermutationP10isdefinedas:
P10(k1,K2,k3,k4,k5,k6,k7,k8,k9,k10)=(k3,k5,K2,k7,k4,k1010,k1,k9,k8,k6)
P10canbeconciselydefinedbythedisplay:

P10									
3	5	2	7	4	10	1	9	8	6

This table is read from left to right; each position in the table gives the identity of theinput bit that produces the output bit in that position. So the first output bit is bit 3 ofthe input; the second output bit is bit 5 of the input, and so on. For example, the key(1010000010) is permuted to (10000 01100). Next, perform a circular left shift (LS-1),orrotation,separatelyonthefirstfivebitsandthesecondfivebits.In ourexample,theresultis(0000111000).NextweapplyP8,whichpicksoutandperm utes8 ofthe10 bitsaccording tothefollowingrule:

P8							
6	3	7	4	8	5	10	9

The result is subkey 1 (K1). In our example, this yields (10100100). We then go backtothepair of5-bitstringsproducedbythetwoLS-1functionsandperformsacircularleft shift of 2 bit positions on each string. In our example, the value (00001 11000)becomes (00100 00011). Finally, P8 is applied again to produce K2. In our example,theresultis(01000011).

S-DESencryption

Encryptioninvolvesthesequentialapplicationoffivefunctions.

InitialandFinalPermutationsTheinputtothealgorithmisan8-bitblockofplaintext,whichwefirstpermuteusingtheIPfunction:

IP							
2	6	3	1	4	8	5	7

Thisretainsall8bitsoftheplaintextbutmixesthemup.Considertheplaintexttobe11 110011.

Permutedoutput=10111101

At theend ofthealgorithm,theinversepermutation isused:

IP–1							
4	1	3	5	7	2	8	6

TheFunctionf_k

ThemostcomplexcomponentofS-DESisthefunctionf_k,whichconsistsofacombinationofpermutationandsubstitution functions.Thefunctionscanbeexpressed as follows. Let L and R be the leftmost 4 bits and rightmost 4 bits of the 8-bit input to f K, and let F be a mapping (not necessarily one to one) from 4-bit stringsto4-bitstrings.

Thenwe letf_k(L,R)=(L(+)F(R,SK),R)

WhereSKisasubkeyand(+)isthebit-by-bitexclusive-ORfunction.

e.g., permuted output = 10111101 and suppose F (1101, SK) = (1110) for some keySK.ThenfK(10111101)= 10111110,1101 =01011101

WenowdescribethemappingF.Theinputisa4-bitnumber(n1n2n3 n4).Thefirstoperationisanexpansion/permutationoperation:

E/P							
4	1	2	3	2	3	4	1

R=1101E/Poutput=11101011Itisclearertodepicttheresultinthisfashion:

n_4	n_1	n_2	n_3
n_2	n_3	n_4	n_1

The8-bitsubkeyK1=(k11,k1212,k1313,k1414,k1515,k1616,k1717,k18)i sadded tothisvalueusingexclusive-OR:

$n_4 \oplus k_{11}$	$n_1 \oplus k_{12}$	$n_2 \oplus k_{13}$	$n_3 \oplus k_{14}$
$n_2 \oplus k_{15}$	$n_3 \oplus k_{16}$	$n_4 \oplus k_{17}$	$n_1 \oplus k_{18}$

Letusrenamethese8 bits:

$p_{0,0}$	$p_{0,1}$	$p_{0,2}$	$p_{0,3}$
$p_{1,0}$	$p_{1,1}$	$p_{1,2}$	$p_{1,3}$

The first 4 bits (first row of the preceding matrix) are fed into the s-box to produce a 2-bit output, and the remaining 4 bits are fed into s1 to produce another 2-bit output.These two boxes are defined as follows:

$$S0 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \end{array} \qquad S1 = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix} \end{array}$$

TheS-boxesoperateasfollows.Thefirstandfourthinputbitsaretreatedasa2-bitnumberthatspecifyarowoftheS-box,and thesecond and third input bits specify a column of the S-box. The entry inthat row and column, in base 2, is the 2-bit output. For example, if (p0,0p0,3) =) (00) and (p0,1 p0,2) = (10), then the output is from row 0,column 2 of S0, which is 3, or (11) in) binary. Similarly, (p1,0 p1,3) and(p1,1p1,2)areusedtoindexintoarowandcolumnofS1to produceanadditional 2 bits. Next, the 4 bits produced by S0 and S1 undergoafurtherpermutationasfollows:

P4			
2	4	3	1

TheoutputofP4isthe outputofthe functionF.

The Switch Function The function f K only alters the leftmost 4 bits of the input.

Theswitchfunction(SW)interchangestheleftandright4bitssothatthesecondinstanceoffKoperateson a different 4 bits. In this second instance, the E/P, S0, S1, and P4 functions are thesame. The keyinputisK2. Finallyapplyinverse permutationtogetthe ciphertext

DATAENCRYPTIONSTANDARD(DES)

The main standard for encrypting data was a symmetric algorithm known asthe Data Encryption Standard (DES). However, this has now been replaced by a newstandard known as the Advanced Encryption Standard (AES) which we will look atlater. DES is a 64 bit block cipher which means that it encrypts

data 64 bits at a time.This is contrasted to a stream cipher in which only one bit at a time (or sometimessmall groups of bits such as a byte) is encrypted. DES was the result of a researchprojectsetupbyInternationalBusinessMachines(IBM)Corporationinthe late1960's which resulted in a cipherknown as LUCIFER. In the early1970's it wasdecidedtocommercializeLUCIFERandanumberofsignificantchangeswerei ntroduced.IBMwasnotthe onlyone involvedinthese changesas theysoughttechnical advice from the National Security Agency (NSA) (other outside consultantswereinvolvedbutitislikelythattheNSAwerethemajorcontributorsfro matechnicalpointofview).ThealteredversionofLUCIFERwasputforwardasapro posal for the new national encryption standard requested by the National BureauofStandards(NBS)3.

Itwasfinallyadoptedin1977astheDataEncryptionStandard DES (FIPS PUB 46). Some of the changes made to LUCIFER have been the subject ofmuch controversy even to the present day. The most notable of these was the key size.LUCIFER used a key size of 128 bits however this was reduced to 56 bits for DES. Eventhough DES actually accepts a 64 bit keyas input, the remaining eight bits are usedfor parity checking and have no effect on DES's security. Outsiders were convincedthat the 56 bit key was an easy target for a brute force attack4 due to its extremelysmall size. The need for the parity checking scheme was also questioned withoutsatisfyinganswers.Anothercontroversialissuewas that the S-boxesusedwere designed under classified conditions and no reasons for their particular design wereever given. This led people toassume that theNSAhad introduceda "trapdoor"throughwhichtheycoulddecryptanydataencryptedbyDESevenwithoutk nowledge of the key. One startling discovery was that the S-boxes appeared to besecure against an attack known as Differential Cryptanalysis which was only publiclydiscovered by Biham and Shamir in 1990. This suggests that the NSA were aware ofthisattackin 1977;13yearsearlier!InfacttheDESdesignersclaimedthatthereasontheynevermadet hedesignspecifications for the S-boxes available was that they knew about a number of attacksthat weren't public knowledge at the time and they didn't want them leaking - this isquite a plausible claim as differential cryptanalysis has shown. However, despite allthis controversy, in 1994 NIST reaffirmed DES for government use for a further fiveyears for use in areas other than "classified". DES of course isn't the only symmetriccipher. There are many others, each with varying levels of complexity. Such ciphersinclude: IDEA, RC4, RC5, RC6 and the new Advanced Encryption Standard (AES). AESis an important algorithm

and was originally meant to replace DES (and its moresecurevarianttripleDES)asthestandardalgorithmfornon-classifiedmaterial.However as of 2003, AES with key sizes of 192 and 256 bits has been found to besecure enough to protect information up to top secret. Since its creation, AES hadunderdone intense scrutiny as one would expect for an algorithm that is to be used asthe standard. To date it has withstood all attacks but the search is still on and itremains to be seen whether or not this will last. We will look at AES later in thecourse.

INNER WORKINGOFDES

DES (and most of the other majorsymmetric ciphers) is based on a cipher known as theFeistelblockcipher.Itconsistsofanumberofroundswhereeachroundcontainsbit-shuffling,non-linearsubstitutions(S-boxes)andexclusiveORoperations.Aswithmostencryptionschemes, DES expectstwo inputs-the plaintext to be encrypted andthesecretkey.Themannerinwhichtheplaintextisaccepted,andthekeyarrangementusedfor encryptionanddecryption,bothdeterminethetypeofcipheritis.DESisthereforeasymmetric, 64 bit block cipher as it uses the same key for both encryption and decryption andonly operates on 64 bit blocks of data at a time5 (be they plaintext or ciphertext). The key sizeusedis56bits,howevera64bit(oreight-byte)keyisactuallyinput.Theleastsignificantbitofeach byte is either used for parity (odd for DES) or set arbitrarily and does not increase thesecurity in any way. All blocks are numbered from left to right which makes the eight bit ofeach byte the paritybit.

Onceaplain-textmessage isreceivedtobeencrypted,itisarranged into64bitblocksrequired for input. If the number of bits in the message is not evenly divisible by 64, then thelastblockwillbepadded.Multiplepermutationsandsubstitutionsareincorporatedth roughoutinordertoincrease thedifficultyofperformingacryptanalysisonthecipher.

OVERALL STRUCTURE:

Figure below shows the sequence of events that occur during an encryption operation. DES performs an initial permutation on the entire 64 bit block of data. It isthensplitinto2,32bitsub-blocks,LiandRiwhicharethenpassedintowhatis known as a round (see figure 2.3), of which there are 16 (the subscript i in Li and Riindicatesthe currentround). Eachoftheroundsareidentical andtheeffectsofincreasing theirnumberis twofold- thealgorithms security is

increased anditstemporalefficiencydecreased.Clearlythesearetwoconflictingoutcomesandacompromisemustbemade.ForDESthenumberchosenwas16,probablytoguarantee the elimination of any correlation between the ciphertext and either theplaintext or key6 . At the end of the 16th round, the 32 bit Li and Ri output quantitiesareswappedtocreatewhatisknownasthepre-output.This[R16,L16]concatenation is permuted using a function which is the exact inverse of the initialpermutation.Theoutputofthisfinalpermutationisthe64bitciphertext

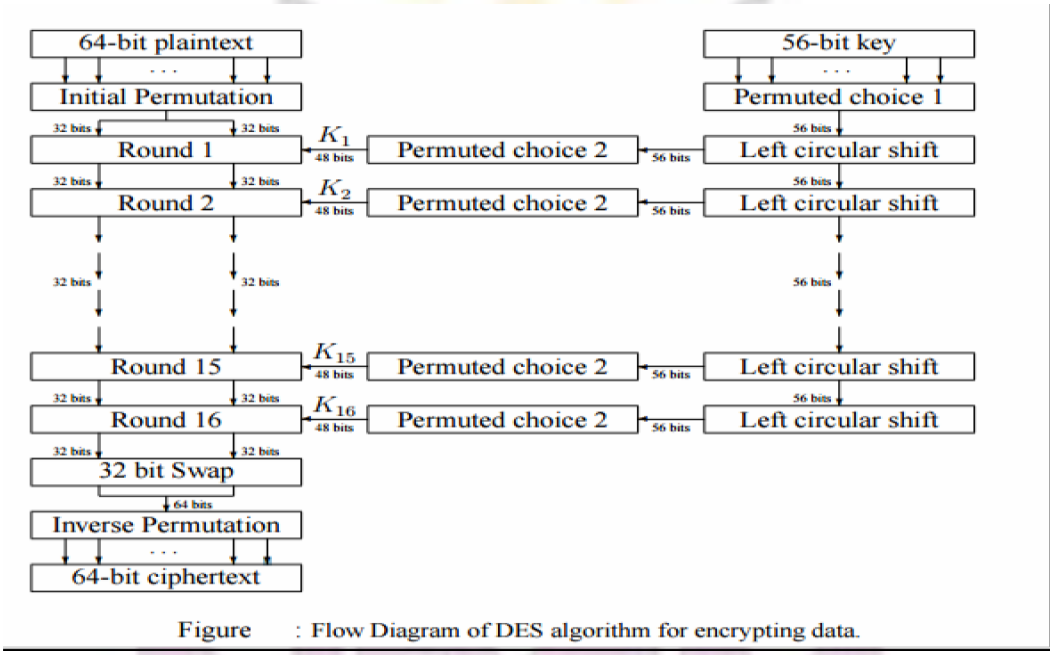


Figure : Flow Diagram of DES algorithm for encrypting data.

Sointotaltheprocessingoftheplaintextproceedsinthreephasesascanbeseenfromthelefthandsideoffigure

1. Initialpermutation(IP-definedintable2.1)rearrangingthebitstoformthe“permutedinput”.
2. Followedby16iterationsofthesamefunction(substitutionandpermutation).Theoutput of the last iteration consists of 64 bits which is a function of the plaintext andkey.Theleftandrighthalvesareswappedtoproducethepreoutput.
3.

(a) Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

(b) Inverse Initial Permutation (IP^{-1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

(c) Expansion Permutation (E)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

(d) Permutation Function (P)

16	7	20	21	29	12	28	17
1	15	23	26	5	18	31	10
2	8	24	14	32	27	3	9
19	13	30	6	22	11	4	25

Table 2.1: Permutation tables used in DES.

As the figure shows the input to each round consists of the theLi,Ripairanda48bitsubkeywhich is a shifted and contracted version of the original 56 bit key. The use of the keycanbeseenintherighthandportionoffigure2.2:•Initiallythekeyispassedthrough a permutation function (PC1 - defined in table 2.2) • For each of the 16 iterations, asubkey (Ki) is produced by a combination of a left circular shift and a permutation(PC2 - defined in table 2.2) which is the same for each iteration. However, the resultingsubkeyisdifferentforeachiterationbecause ofrepeatedshifts.

(a) Input Key

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56
57	58	59	60	61	62	63	64

(b) Permuted Choice One (PC-1)

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

(c) Permuted Choice Two (PC-2)

14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

(d) Schedule of Left Shifts

Round number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Table 2.2: DES key schedule.

DETAILS OF INDIVIDUAL ROUNDS

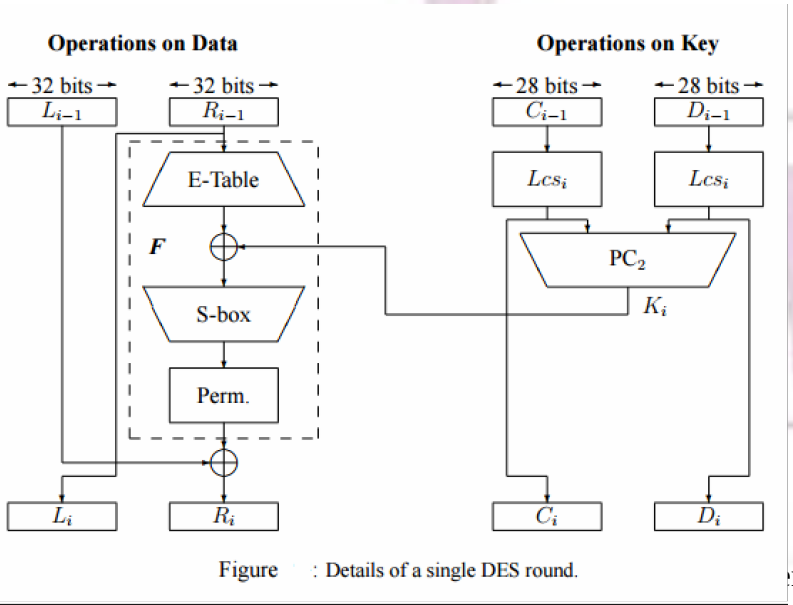


Figure : Details of a single DES round.

Tl rred to as the cipherfunction and is labeled F. This function accepts two different length inputs of 32 bits and 48bits and outputs a single 32 bit number. Both the data and key are operated on in parallel,however the operations are quite different. The 56 bit key is split into two 28 bit halves Ciand Di (C and D being chosen so as not to be confused with L and R). The value

of the keyused in any round is simply a left cyclic shift and a permuted contraction of that used in thepreviousround.

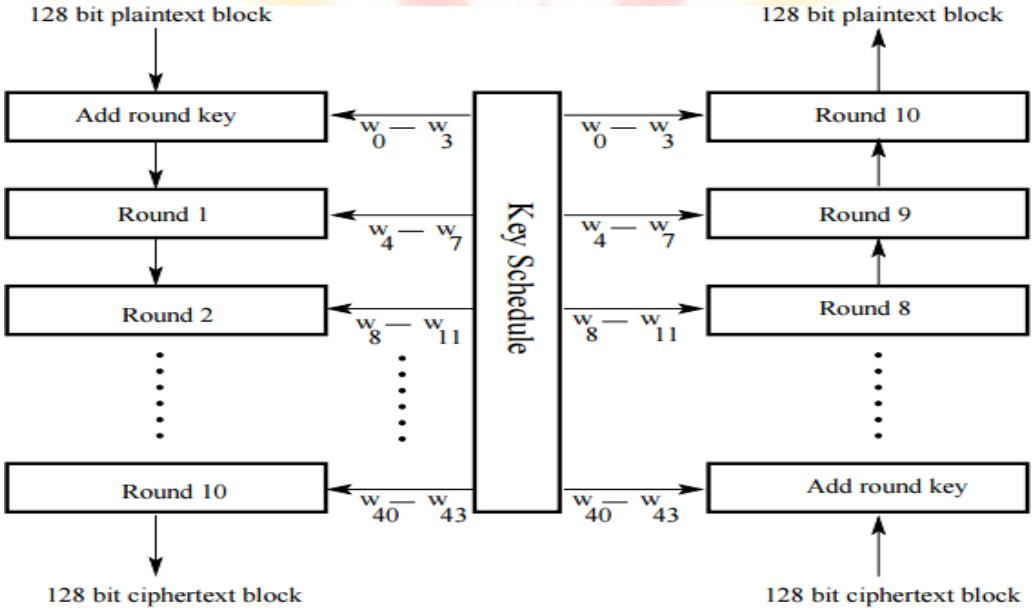
Mathematically, thiscan bewritten as

Ci=Lcsi(Ci-1)

Di=Lcsi(Di-1)

Ki=P C2(Ci,Di)

where Lcsi is the left cyclic shift for round i, Ci and Di are the outputs after the shifts, P C2(.) isafunctionwhichpermutesand compressesa56 bit numberinto a48 bit number and Ki isthe actual key used in round i. The number of shifts is either one or two and is determined bythe roundnumberi.Fori={1,2,9,16}
thenumberofshiftsisoneandforeveryotherrounditistwo



AES Encryption AES Decryption

ThealgorithmbeginswithanAddroundkeystagefollowedby9rounds offourstagesandatenthroundofthreestages.
This applies for both encryption and decryption with the exception that each stage ofa round the decryption algorithm is the inverse of its counterpart in the encryptionalgorithm.

Thefourstagesareasfollows:1. Substitutebytes2.Shiftrows3.MixColumns4.AddRoundKey

SubstituteBytes

• Thisstage(knownas SubBytes) issimply atablelookupusinga16×16matrixofbytevaluescalledans-box.

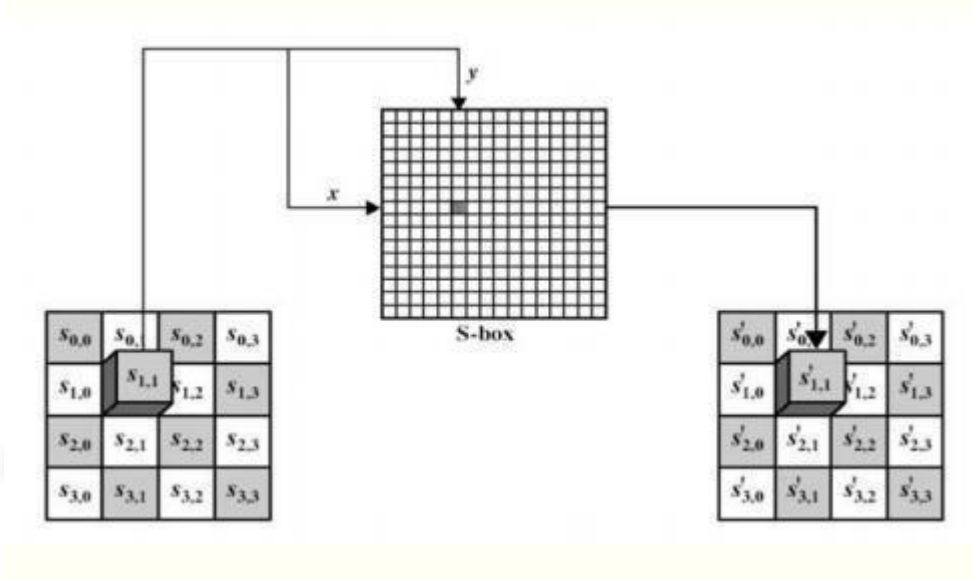
• Thismatrixconsistsofallthepossiblecombinationsofan8bitsequence(28 =16×16 =256).

• However,thes-boxisnotjustarandompermutationofthesevaluesandthereisawelldefined method forcreating thes-boxtables.

• The designersof Rijndaelshowed howthis wasdone unlikethes-boxesin DES forwhichnorationalewasgiven.Ourconcernwillbehowstateiseffectedineach round.

• For this particular round each byte is mapped into a new byte in the following way:the leftmost nibble of the byte is used to specify a particular row of the s-box and therightmostnibblespecifiesa column.

• For example, the byte {95} (curly brackets represent hex values in FIPS PUB 197)selectsrow9column5whichturnsouttocontainthevalue{2A}. Thisisthenusedtoupdatethestatematrix.

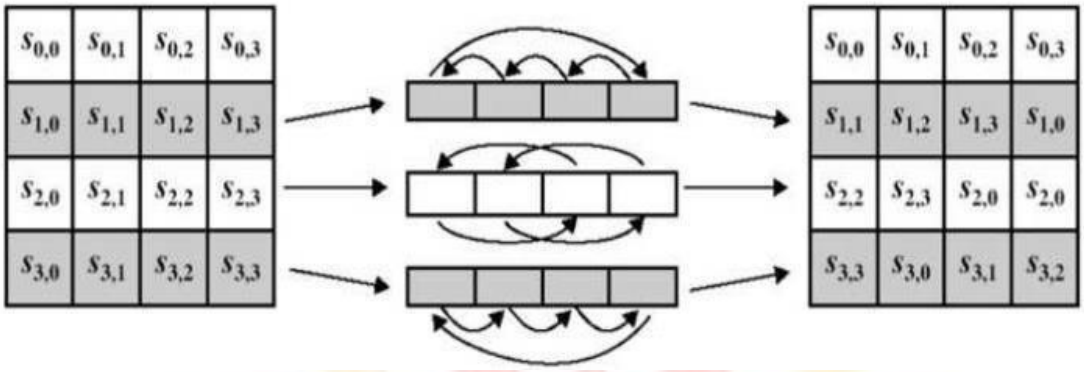


ShiftRowTransformation

• Thisstage(knownasShiftRows)isshowninfigurebelow.

• Simplepermutationannothingmore.

• It works as follow: – The first row of state is not altered. – The second row is shifted1bytestotheleftinacircularmanner.– Thethirdrowisshifted2bytestotheleftina circularmanner.– Thefourthrowisshifted3bytestotheleftinacircularmanner.



MIX COLUMNTRANSFORMATION

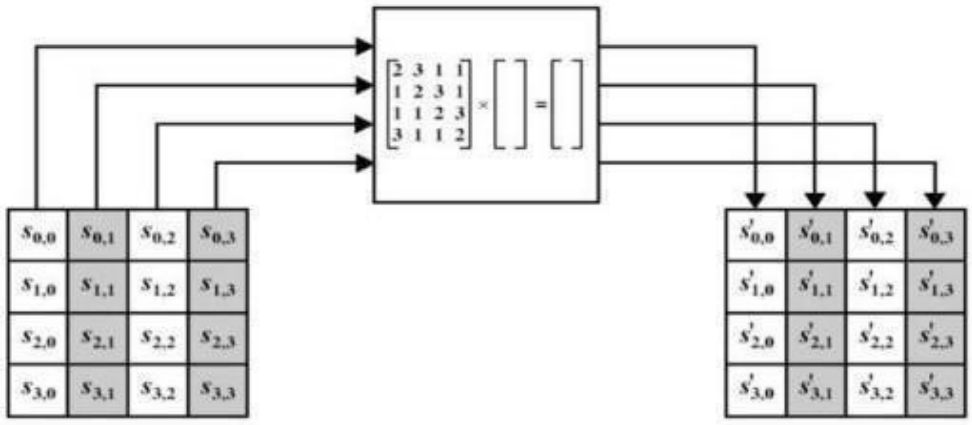
• Thisstage(knownasMixColumn)isbasicallyasubstitution

• Each columnis operated onindividually. Eachbyte ofa columnismapped into anewvaluethatisafunctionofallfourbytesin thecolumn.

• The transformationcanbedetermined bythe followingmatrixmultiplication onstate

• Eachelementoftheproductmatrixisthesumofproductsofelementsofonero wandonecolumn.

• Inthiscasetheindividualadditionsandmultiplicationsareperformedin GF(28). TheMixColumnstransformationofasinglecolumnj(0≤j≤3)ofstatecanbe



ADDROUNDKEYTRANSFORMATION

• In this stage(known as AddRoundKey)the 128 bits of stateare bitwise XORed withthe128bitsoftheroundkey.
• Theoperationisviewedasacolumnwiseoperationbetweenthe4bytesofastatecolumn andonewordoftheroundkey.
• This transformation is as simpleas possible whichhelpsinefficiencybut it alsoeffectseverybitofstate.
• TheAESkeyexpansionalgorithmtakesasinputa4-wordkeyandproducesalineararrayof44 words.Eachrounduses4ofthesewordsasshowninfigure.
• Eachwordcontains32byteswhichmeanseachsubkeyis128bitslong.Figure7showpseudocodeforgeneratingtheexpandedkeyfromtheactualkey.

BLOWFISHALGORITHM

- asymmetricblockcipher designedbyBruceSchneierin1993/94
- characteristics
  - fastimplementationon32-bitCPUs
  - compactinuseofmemory
  - simplestructureforanalysis/implementation
  - variablesecuritybyvaryingkeysize
- hasbeenimplementedinvariousproducts

BLOWFISHKEYSCHEDULE

- usesa32to448bitkey,32-bitwordsstoredinK-arrayK_j,jfrom1to14
- usedtogenerate
  - 1832-bitsubkeysstoredinParray,$P_1 \ldots P_{18}$
  - four8x32S-boxesstoredin$S_{i,j}$,eachwith25632-bitentries

1. initializeP-arrayandthen4S-boxesinorderusingthefractionalpartofpiP_1(leftmost3

2-bit),andsoon,,,$S_{4,255}$.

2. XOR P-array with key-Array (32-bit blocks) and reuse as needed:assume we have up tok_{10} then P_{10} XOR K_{10},, P_{11} XOR K_1 … P_{18} XORK_8

Encrypt64-bitblockofzeros,andusetheresulttoupdateP_1andP_2.

1. encrypting output form previous step using current P & S andreplaceP3 andP4.Thenencryptingcurrentoutputanduseitto
2. updatesuccessivepairsofP.
3. After updating all P's(last:P_{17} P_{18}), startupdatingS valuesusingtheencryptedoutputfrompreviousstep.

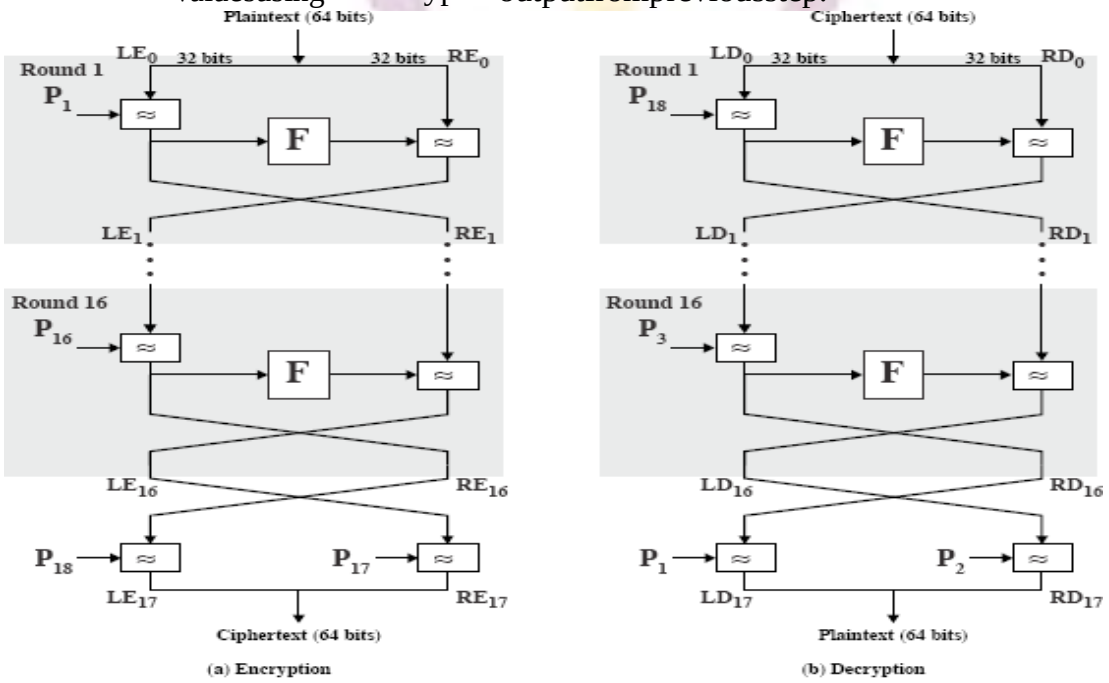


Figure 6.3 Blowfish Encryption and Decryption

- requires521encryptions,henceslowinre-keying
- Notsuitableforlimited-memoryapplications.

BLOWFISH ENCRYPTION

- usestwomainoperations:additionmodulo2^{32},andXOR
- dataisdividedintotwo32-bithalvesL_0&R_0

  for$i$=1to16do

  $R_i = L_{i-1}$XOR P_i;

  $R_i = L_{i-1}$XOR P_i;R_{i1}

$L_{17}=R_{16}XORP_{18}$;

RC5 Encryption Algorithm

RC5 is a symmetric key block encryption algorithm designed by Ron Rivest in 1994. It is notable for being simple, fast (on account of using only primitive computer operations like XOR, shift, etc.) and consumes less memory.

Example:

Key : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Plain Text : 00000000 00000000

Cipher Text : EEDBA521 6D8F4B15

RC5 is a block cipher and addresses two word blocks at a time.
Depending on input plain text block size, number of rounds and key size, various instances of RC5 can be defined and each instance is denoted as RC5-w/r/b where w=word size in bits, r=number of rounds and b=key size in bytes.
Allowed values are:

Parameter	Possible Value
block/word size (bits)	16, 32, 64
Number of Rounds	0 – 255
Key Size (bytes)	0 – 255

Note – Since at a time, RC5 uses 2 word blocks, the plain text block size can be 32, 64 or 128 bits.

Notation used in the algorithm:

Symbol	Operation
x <<< y	Cyclic left shift of x by y bits

Symbol	Operation
+	Two's complement addition of words where addition is modulo
^	Bit wise Exclusive-OR

Step-1: Initialization of constants P and Q.
RC5 makes use of 2 magic constants P and Q whose value is defined by the word size w.

Word Size (bits)	P (Hexadecimal)	Q (Hexadecimal)
16	b7e1	9e37
32	b7e15163	9e3779b9
64	b7e151628aed2a6b	9e3779b97f4a7c15

For any other word size, P and Q can be determined as:

P = Odd((e-2))

Q = Odd((-2))

Here, Odd(x) is the odd integer nearest to x, e is the base of natural logarithms and is the golden ratio.

Step-2: Converting secret key K from bytes to words.
Secret key K of size b bytes is used to initialize array L consisting of c words where c = b/u, u = w/8 and w = word size used for that particular instance of RC5. For example, if we choose w=32 bits and Key k is of size 96 bytes then, u=32/8=4, c=b/u=96/4=24.
L is pre initialized to 0 value before adding secret key K to it.

for i=b-1 to 0

L[i/u] = (L[u/i] <<< 8) + K[i]

Step-3: Initializing sub-key S.
Sub-key S of size t=2(r+1) is initialized using magic constants P and Q.

S[0] = P

for i = 1 to 2(r+1)-1

S[i] = S[i-1] + Q)

Step-4: Sub-key mixing.
The RC5 encryption algorithm uses Sub key S. L is merely, a temporary array formed on the basis of user entered secret key.
Mix in user's secret key with S and L.

i = j = 0

A = B = 0

do 3 * max(t, c) times:

A = S[i] = (S[i] + A + B) <<< 3

B = L[j] = (L[j] + A + B) <<< (A + B)

i = (i + 1) % t

j = (j + 1) % c

Step-5: Encryption.
We divide the input plain text block into two registers A and B each of size w bits. After undergoing the encryption process the result of A and B together forms the cipher text block.
RC5 Encryption Algorithm:

1. One time initialization of plain text blocks A and B by adding S[0] and S[1] to A and B respectively. These operations are mod .
2. XOR A and B. A=A^B
3. Cyclic left shift new value of A by B bits.
4. Add S[2*i] to the output of previous step. This is the new value of A.
5. XOR B with new value of A and store in B.
6. Cyclic left shift new value of B by A bits.
7. Add S[2*i+1] to the output of previous step. This is the new value of B.
8. Repeat entire procedure (except one time initialization) r times.

A = A + S[0]

B = B + S[1]

for i = 1 to r do:

A = ((A ^ B) <<< B) + S[2 * i]

B = ((B ^ A) <<< A) + S[2 * i + 1]

return A, B

Alternatively, RC5 Decryption can be defined as:

for i = r down to 1 do:

B = ((B - S[2 * i + 1]) >>> A) ^ A

A = ((A - S[2 * i]) >>> B) ^ B

B = B - S[1]

A = A - S[0]

return A, B

Simplified International Data Encryption Algorithm (IDEA)

In cryptography, block ciphers are very important in the designing of many cryptographic algorithms and are widely used to encrypt the bulk of data in chunks. By chunks, it means that the cipher takes a fixed size of the plaintext in the encryption process and generates a fixed size ciphertext using a fixed-length key. An algorithm's strength is determined by its key length.

The Simplified **International Data Encryption Algorithm (IDEA)** is a **symmetric key block** cipher that:

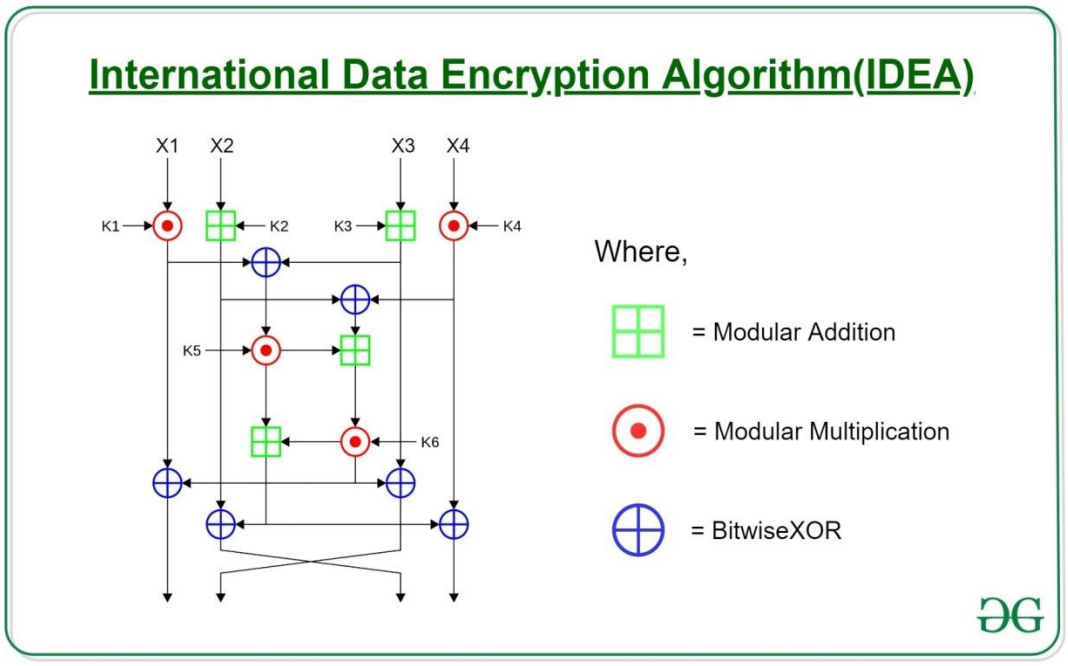
- uses a fixed-length plaintext of **16 bits** and
- encrypts them in **4 chunks of 4 bits** each
- to produce **16 bits ciphertext**.
- The length of the key used is **32 bits**.
- The key is also divided into 8 blocks of 4 bits each.

This algorithm involves a series of 4 identical complete rounds and 1 half-round. Each complete round involves a series of 14 steps that includes operations like:

- Bitwise XOR
- Addition modulo
- Multiplication modulo +1

After 4 complete rounds, the final "half-round" consists of only the first 4 out of the 14 steps previously used in the full rounds. To perform these rounds, each binary notation must be converted to its equivalent decimal notation, perform the operation and the result obtained should be converted back to the binary representation for the final result of that particular step.

Key Schedule: 6 subkeys of 4 bits out of the 8 subkeys are used in each complete round, while 4 are used in the half-round. So, 4.5 rounds require 28 subkeys. The given key, 'K', directly gives the first 8 subkeys. By rotating the main key left by 6 bits between each group of 8, further groups of 8 subkeys are created, implying less than one rotation per round for the key (3 rotations).



	K1	K2	K3	K4	K5	K6
Round 1	1101	1100	0110	1111	0011	1111
Round 2	0101	1001*	0001	1011	1100	1111
Round 3	1101	0110	0111	0111*	1111	0011
Round 4	1111	0101	1001	1101	1100	0110*
Round 4.5	1111	1101	0110	0111		

* denotes a shift of bits

Notations used in the 14 steps:

Symbol Operation

Symbol	Operation
*	Multiplication modulo +1
+	Addition modulo
^	Bitwise XOR

The 16-bit plaintext can be represented as **X1 || X2 || X3 || X4**, each of size 4 bits. The 32-bit key is broken into 8 subkeys denoted as K1 || K2 || K3 || K4 || K5 || K6 || K7 || K8, again of size 4 bits each. Each round of 14 steps uses the three algebraic operation-Addition modulo (2^4), Multiplication modulo (2^4)+1 and Bitwise XOR. The steps involved are as follows:

1. X1 * K1
2. X2 + K2
3. X3 + K3
4. X4 * K4
5. Step 1 ^ Step 3
6. Step 2 ^ Step 4
7. Step 5 * K5
8. Step 6 + Step 7
9. Step 8 * K6
10. Step 7 + Step 9
11. Step 1 ^ Step 9
12. Step 3 ^ Step 9
13. Step 2 ^ Step 10
14. Step 4 ^ Step 10

The input to the next round is Step 11 || Step 13 || Step 12 || Step 14, which becomes X1 || X2 || X3 || X4. This swap between 12 and 13 takes place after each complete round, except the last complete round (4th round), where the input to the final half round is Step 11 || Step 12 || Step 13 || Step 14.

After last complete round, the half-round is as follows:

1. X1 * K1
2. X2 + K2
3. X3 + K3
4. X4 * K4

The final output is obtained by concatenating the blocks.

Example:

Key: 1101 1100 0110 1111 0011 1111 0101 1001

Plaintext: 1001 1100 1010 1100

Ciphertext: 1011 1011 0100 1011

Explanation:
The explanation is only for 1st complete round (the remaining can be implemented similarly) and the last half round.

- **Round 1:**
  - From the plaintext: **X1 – 1001, X2 – 1100, X3 – 1010, X4 – 1100**
  - From the table above: **K1 – 1101, K2 – 1100, K3 – 0110, K4 – 1111, K5 – 0011, K6 – 1111**
  -

(1001(9) * 1101(13))(mod 17) = 1111(15)

(1100(12) + 1100(12))(mod 16) = 1000(8)

(1010(10) + 0110(6))(mod 16) = 0000(0)

(1100(12) * 1111(15))(mod 17) = 1010(10)

(1111(15) ^ 0000(0)) = 1111(15)

(1000(8) ^ 1010(10)) = 0010(2)

(1111(15) * 0011(3))(mod 17) = 1011(11)

(0010(2) + 1011(11))(mod 16) = 1101(13)

(1101(13) * 1111(15))(mod 17) = 1000(8)

(1011(11) + 1000(8))(mod 16) = 0011(3)

(1000(8) ^ 1111(15)) = 0111(7)

(1000(8) ^ 0000(0)) = 1000(8)

(0011(3) ^ 1000(8)) = 1011(11)

(0011(3) ^ 1010(10)) = 1001(9)

-
- **Round 1 Output**: 0111 1011 1000 1001 *(Step 12 and Step 13 results are interchanged)*
- **Round 2:**
    - From Round 1 output: **X1 – 0111, X2 – 1011, X3 – 1000, X4 – 1001**
    - From the table above: **K1 – 0101, K2 – 1001, K3 – 0001, K4 – 1011, K5 – 1100, K6 – 1111**
    - **Round 2 Output**: 0110 0110 1110 1100 *(Step 12 and Step 13 results are interchanged)*
- **Round 3:**
    - From Round 2 Output: **X1 – 0110, X2 – 0110, X3 – 1110, X4 – 1100**
    - From the table above: **K1 – 1101, K2 – 0110, K3 – 0111, K4 – 0111, K5 – 1111, K6 – 0011**
    - **Round 3 Output**: 0100 1110 1011 0010 *(Step 12 and Step 13 results are interchanged)*
- **Round 4:**
    - From Round 3 Output: **X1 – 0100, X2 – 1110, X3 – 1011, X4 – 0010**
    - From the table above: **K1 – 1111, K2 – 0101, K3 – 1001, K4 – 1101, K5 – 1100, K6 – 0110**
    - **Round 4 Output**: 0011 1110 1110 0100 *(Step 12 and Step 13 results are interchanged)*
- **Round 4.5:**
    - From Round 4 Output: **X1 – 0011, X2 – 1110, X3 – 1110, X4 – 0100**
    - From the table above: **K1 – 1111, K2 – 1101, K3 – 0110, K4 – 0111**

- **Round 4.5 Output**: 1011 1011 0100 1011 *(Step 2 and Step 3 results are **not** interchanged)*
-

(0011(3) * 1111(15))(mod 17) = 1011(11)

(1110(14) + 1101(13))(mod 16) = 1011(11)

(1110(14) + 0110(6))(mod 16) = 0100(4)

(0100(4) * 0111(7))(mod 17) = 1011(11)

-
- **Final Ciphertext** is **1011 1011 0100 1011**

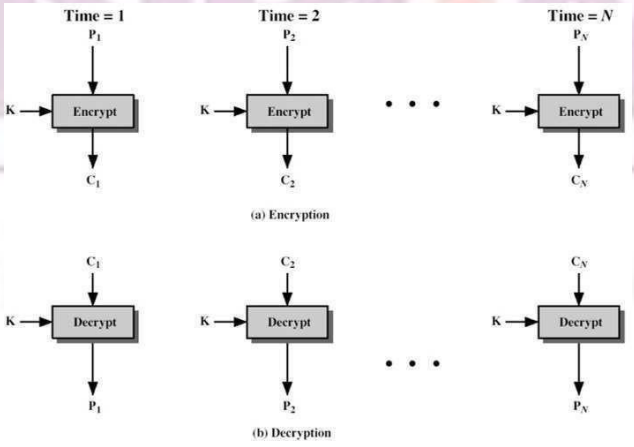
BLOCKCIPHEROPERATIONS

• Direct useof ablockcipherisinadvisable
• Enemycanbuild up"codebook"ofplaintext/ciphertextequivalents
• Beyond that, direct use only works on messages that are a multipleofthe cipherblocksizeinlength
• Solution: five standard Modes of Operation: Electronic Code Book(ECB), Cipher Block Chaining (CBC), Cipher Feedback (CFB), OutputFeedback(OFB),andCounter(CTR).

ElectronicCodeBook

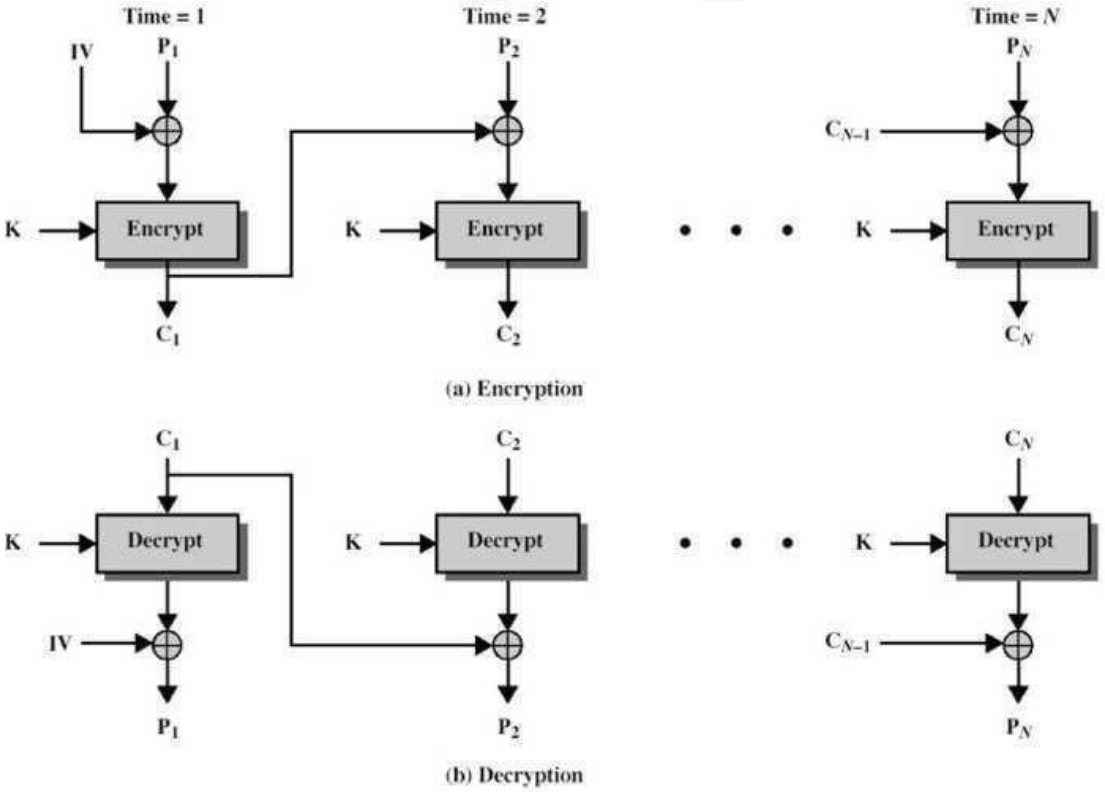
• Directuseof theblockcipher
• Used primarilytotransmitencryptedkeys
• Veryweakifusedfor general-purposeencryption;neveruseitfor afileoramessage.
• Attacker canbuildupcodebook;nosemanticsecurity

• Wewrite{P}k→Cto denote"encryptionofplaintextPwithkeyk toproduceciphertext



CIPHER BLOCK CHAINING:

- Wewouldlikethatsameplaintextblocksproducedifferentciphertextblocks.
- CipherBlockChaining(seefigure)allowsthisbyXORingeachplaintextwiththeCiphertext from the previous round (the first round using an Initialisation Vector(IV)).
- Asbefore,thesamekeyisusedforeachblock.
- Decryptionworksasshowninthefigurebecauseofthepropertiesofthe XORoperation,



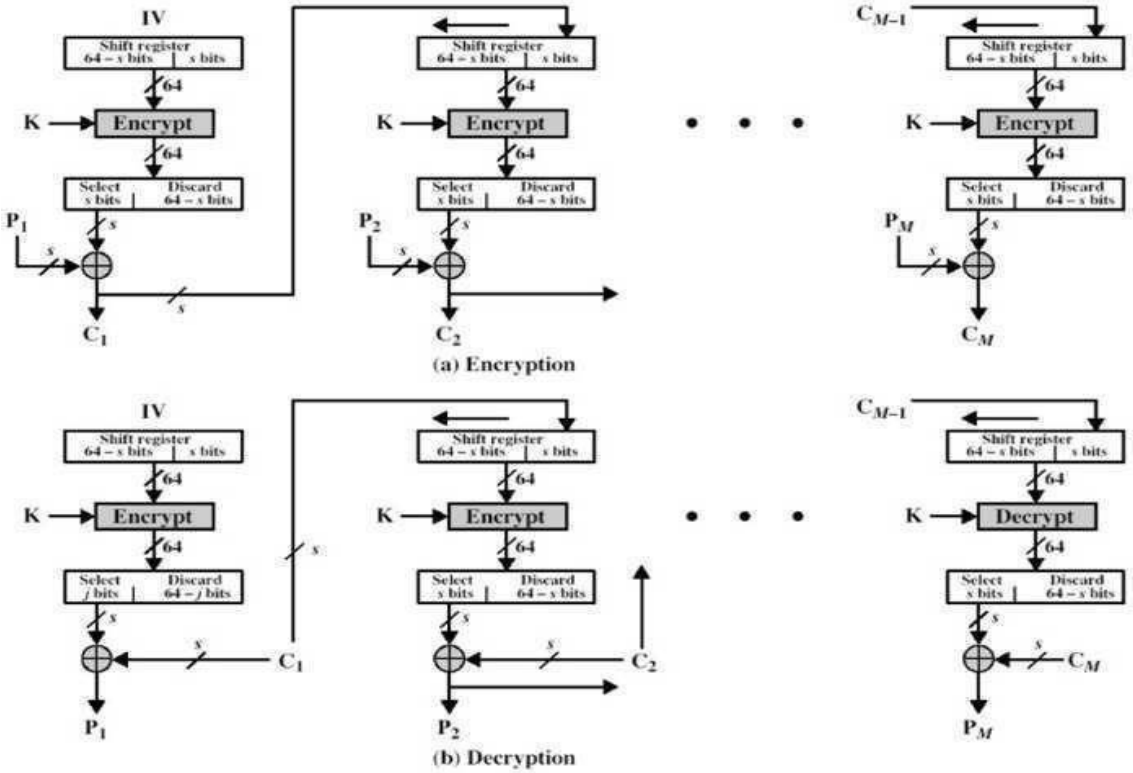
CipherFeedback(CFB)Mode

- TheCipherFeedbackandOutputFeedbackallowsablockciphertobeconvertedintoastreamcipher.
- Thiseliminatestheneedtopadamessagetobeanintegralnumberofblocks. Italsocanoperatein real time.
- FigureshowstheCFBscheme.
- Inthisfigureitassumedthattheunitoftransmissionissbits;acommonvalueiss=8.
- AswithCBC, theunitsofplaintextarechainedtogether,sothatthecipher textofanyplaintext

unit is a function of all the preceding plaintext (which is split into s bitsegments).

• The input to the encryption function is a shift register equal in length to the blockcipher of the algorithm (although the diagram shows 64 bits, which is block size usedbyDES,thiscanbeextendedtootherblocksizessuchasthe128bitsofAES).

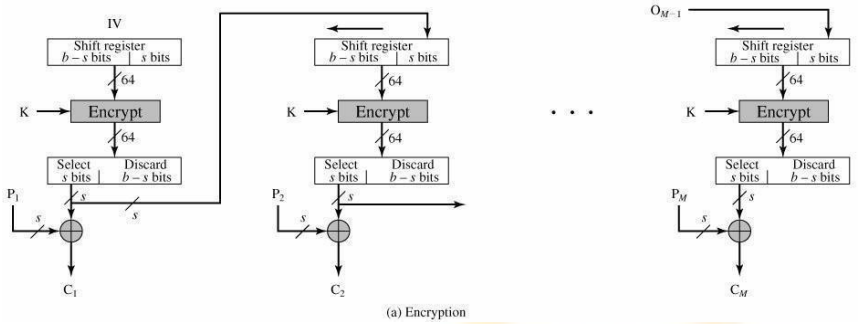
• ThisisinitiallysettosomeInitialisationVector(IV).

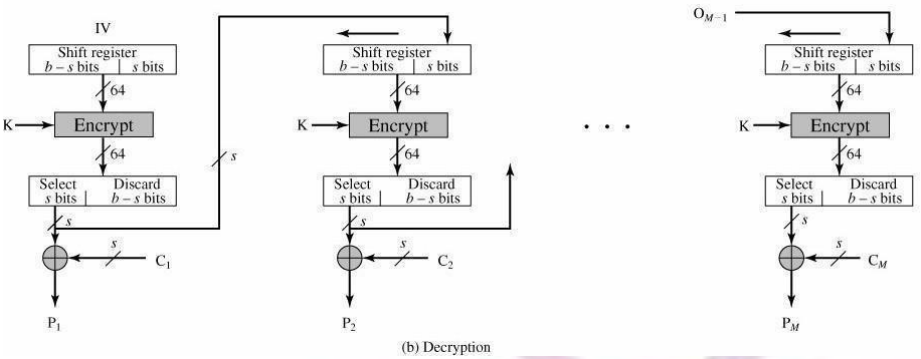


OUTPUT FEEDBACK(OFB) MODE

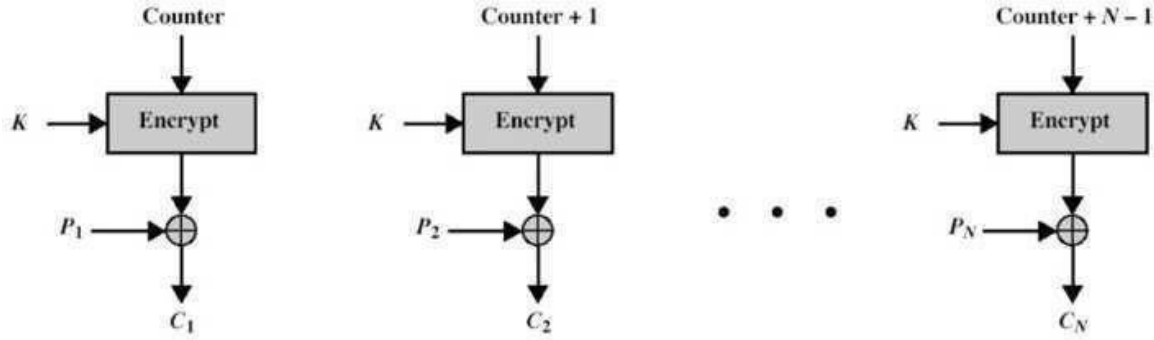
- The Output Feedback Mode is similarin structure to that of CFB,as seen in figure13.
- Ascanbeseen,itistheoutputoftheencryptionfunctionthatisfedbackt otheshift register in OFB, whereas in CFB the ciphertext unit is fed back to the shiftregister.
- OneadvantageoftheOFBmethodisthatbiterrorsintransmission do notpropagate.
- For example, ifabiterror occurs in C1onlytherecovered value of P1 is affected;subsequentplaintextunitsarenotcorrupted.

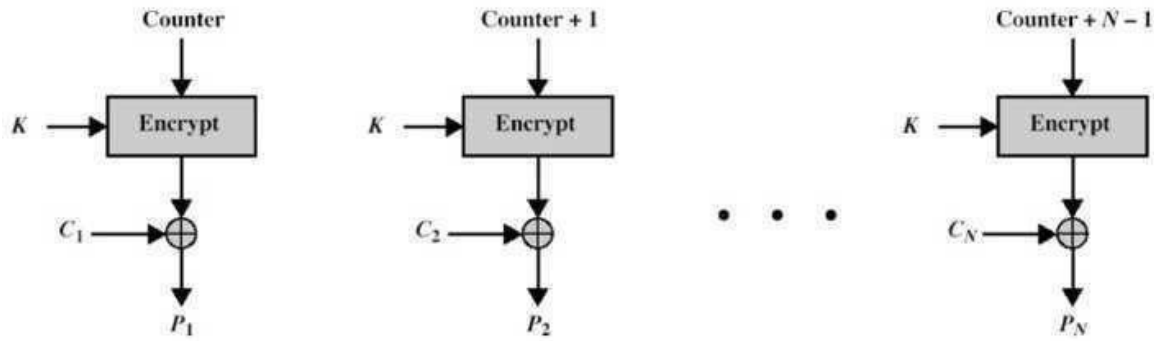
WithCFB,C1alsoservesasinputtotheshiftregisterandthereforecausesadd itionalcorruption downstream.





CounterMode





professor

Stream Ciphers

In stream cipher, one byte is encrypted at a time while in block cipher ~128 bits are encrypted at a time.

Initially, a key(k) will be supplied as input to pseudorandom bit generator and then it produces a random 8-bit output which is treated as keystream.

The resulted keystream will be of size 1 byte, i.e., 8 bits.

1. Stream Cipher follows the sequence of pseudorandom number stream.
2. One of the benefits of following stream cipher is to make cryptanalysis more difficult, so the number of bits chosen in the Keystream must be long in order to make cryptanalysis more difficult.
3. By making the key more longer it is also safe against brute force attacks.
4. The longer the key the stronger security is achieved, preventing any attack.
5. Keystream can be designed more efficiently by including more number of 1s and 0s, for making cryptanalysis more difficult.
6. Considerable benefit of a stream cipher is, it requires few lines of code compared to block cipher.

Encryption:

For Encryption,

- Plain Text and Keystream produces Cipher Text (Same keystream will be used for decryption.).
- The Plaintext will undergo XOR operation with keystream bit-by-bit and produces the Cipher Text.

Example –

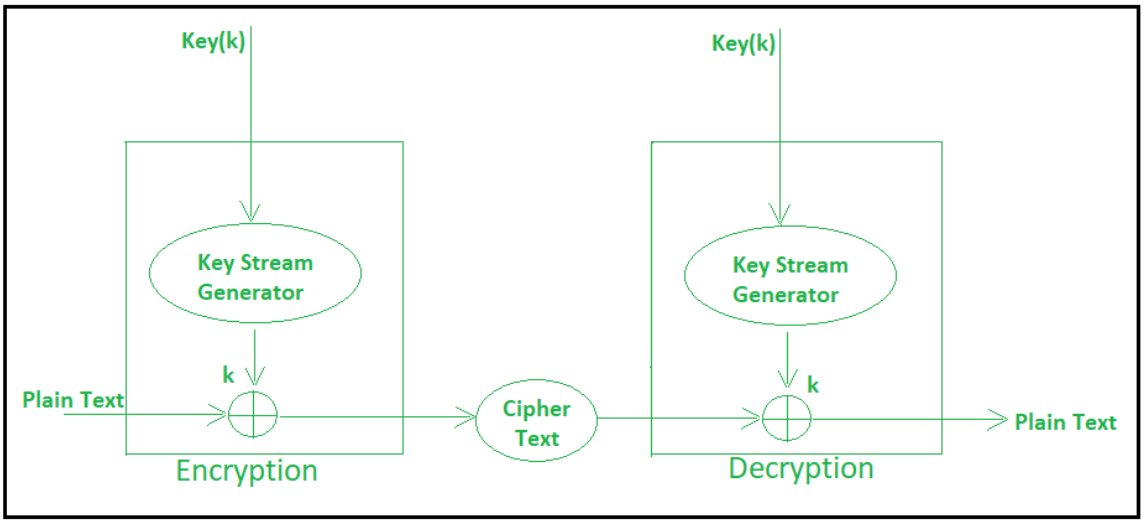
Plain Text: 10011001

Keystream: 11000011

``````````````````````

Cipher Text: 01011010

## Decryption:

For Decryption,

- Cipher Text and Keystream gives the original Plain Text (Same keystream will be used for encryption.).
- The Ciphertext will undergo XOR operation with keystream bit-by-bit and produces the actual Plain Text.

**Example –**

Cipher Text: 01011010

Keystream: 11000011

``````````````````````

Plain Text: 10011001

Decryption is just the reverse process of Encryption i.e. performing XOR with Cipher Text.



RC4

RC4 means Rivest Cipher 4 invented by Ron Rivest in 1987 for RSA Security. It is a Stream Ciphers. Stream Ciphers operate on a stream of data byte by byte. RC4 stream cipher is one of the most widely used stream ciphers because of its simplicity and speed of operation. It is a variable key-size stream cipher with byte-oriented operations. It uses either 64 bit or 128-bit key sizes. It is generally used in applications such as Secure Socket Layer (SSL), Transport Layer Security (TLS), and also used in IEEE 802.11 wireless LAN std.

Why Encryption Is Important?

Unauthorized data access can be prevented by encryption. If we perform encryption then third parties can not have access to data which we share or receive. The encryption is done by using a secret key, or we can say that by using a public key and private key. Both sender and receiver are having their public key and private key through which encryption of plain text and decryption of ciphertext is performed.

History of RC4 Encryption

RC4 was designed by Ron Rivest in 1987. He was working under RSA Security. Rivest Cipher 4 is an official name while it is also known as Ron's Code. Initially, RC4 was trade secret but once it's code spread in the public domain it was no more a trade secret. While Ron did not reveal the RC4 algorithm until 2014 when he described the history of RC4 in English Wikipedia.

Applications of RC4

RC4 is used in various applications such as WEP from 1997 and WPA from 2003. We also find applications of RC4 in SSL from 1995 and it is a successor of TLS from 1999. RC4 is used in varied applications because of its simplicity, speed, and simplified implementation in both software and hardware.

Types of RC4

There are various types of RC4 such as Spritz, RC4A, VMPC, and RC4A.

1. SPRITZ: Spritz can be used to build a cryptographic hash function, a deterministic random bit generator (DRBG), n an encryption algorithm that supports authenticated encryption with associated data (AEAD).
2. RC4A: Souraduyti Paul and Bart Preneel have proposed an RC4 variant, which they call RC4A, which is stronger than RC4.
3. VMPC: VMPC is another variant of RC4 which stands for Variably Modified Permutation Composition.

RC4A+: RC4A+ is a modified version of RC4 with a more complex three-phase key schedule which takes about three times as long as RC4 and a more complex output function which performs four additional lookups in the S array for each byte output, taking approximately 1.7 times as long as basic

RC4 Algorithm

The algorithm operates on a user-selected variable-length key(K) of 1 to 256 bytes (8 to 2048 bits), typically between 5 and 16 bytes. To generate a 256-byte state vector S, the master key is used.
The first step is the array initialization. It is a character array of size 256 i.e. S[256]. After that, for every element of the array, we initialize S[i] to i.

Code for array initialization:

```
Char S[256];

int i;

for(i=0;i<256;i++)

S[i] = i
```

The array will look like –

S[] = {0, 1, 2, 3, ------, 254, 255}

After this, we will run the KSA algorithm-KSA is going to use the secret key to scramble this array. KSA is a simple loop, in which we are having two variable i and j. We are using these variables to rearrange the array. Rearranging the array is done by using a secret key.

Code for KSA (Key Scheduling Algorithm) :

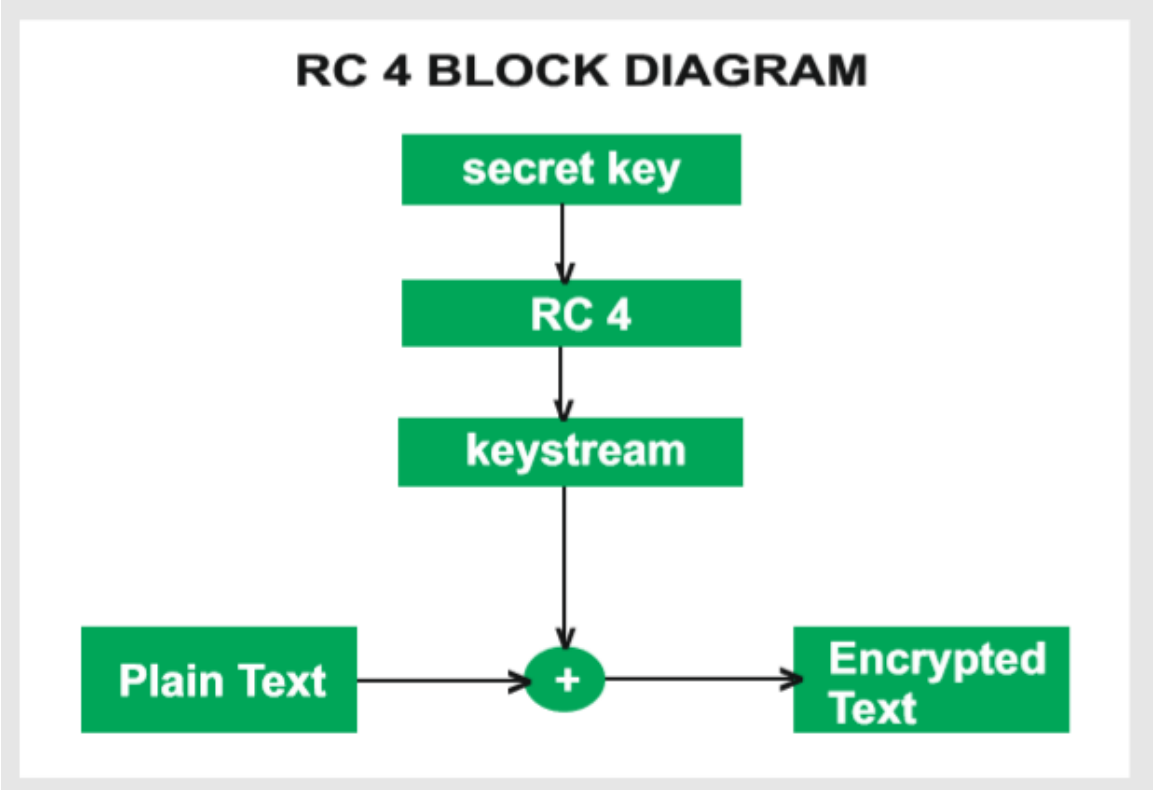
```
int i, j=0;

for(i=0;i<256;i++)
{
j=( j + S[i] + T[i]) mod 256;
Swap(S[i], S[j]);
}
```

KSA has been scrambled, S[256] array is used to generate the PRGA(Pseudo Random Generation Algorithm). This is the actual Keystream.

Code for PRGA (Pseudo Random Generation Algorithm):

```
i=j=0;
while(true)
{
i = ( i + 1 ) mod 256;
j = ( j + S[i] ) mod 256;
Swap( S[i], S[j] );
t = ( S[i] + S[j] ) mod 256 ;
k = S[t];
}
```

This is the next step of scrambling.



Working of RC4

Encryption Procedure

1. The user inputs a plain text file and a secret key.
2. The encryption engine then generates the keystream by using KSA and PRGA Algorithm.
3. This keystream is now XOR with the plain text, this XORing is done byte by byte to produce the encrypted text.
4. The encrypted text is then sent to the intended receiver, the intended receiver will then decrypted the text and after decryption, the receiver will get the original plain text.

Decryption Procedure

Decryption is achieved by doing the same byte-wise X-OR operation on the Ciphertext.

Example: Let A be the plain text and B be the keystream (A xor B) xor B = A

Advantages

1. RC4 stream ciphers are simple to use.
2. The speed of operation in RC4 is fast as compared to other ciphers.
3. RC4 stream ciphers are strong in coding and easy to implement.
4. RC4 stream ciphers do not require more memory.
5. RC4 stream ciphers are implemented on large streams of data.

Disadvantages

- If RC4 is not used with strong MAC then encryption is vulnerable to a bit-flipping attack.
- RC4 stream ciphers do not provide authentication.
- RC4 algorithm requires additional analysis before including new systems.
- RC4 stream ciphers cannot be implemented on small streams of data.
- RC4 fails to discard the beginning of output keystream or fails to use non-random or related keys for the algorithm.

ASSYMETRIC KEY CIPHERS

PRINCIPLES OF PUBLIC KEYCRYPTOSYSTEMS

The development of public-key cryptography is the greatest and perhaps theonly true revolution in the entire history of cryptography. It is *asymmetric*, involvingthe use of two separate keys, in contrast to symmetric encryption, which uses onlyonekey.Publickeyschemesareneithermorenorlesssecurethanprivatekey(security depends on the key size for both). Public-key cryptography *complementsratherthanreplaces*symmetriccryptography.Bothalsohaveissueswithkeydistribution,requiring theuseof some suitable protocol. The concept of

public-key cryptographyevolved from anattempttoattacktwoofthemostdifficultproblemsassociatedwithsymmetricencryption:

1.) ***key distribution***– how to have secure communicationsin general withouthavingtotrustaKDCwithyourkey

2.)***digitalsignatures***– howtoverifyamessagecomesintactfromtheclaimedsender

Public-key/two-key/asymmetriccryptographyinvolvestheuseof**two**keys:

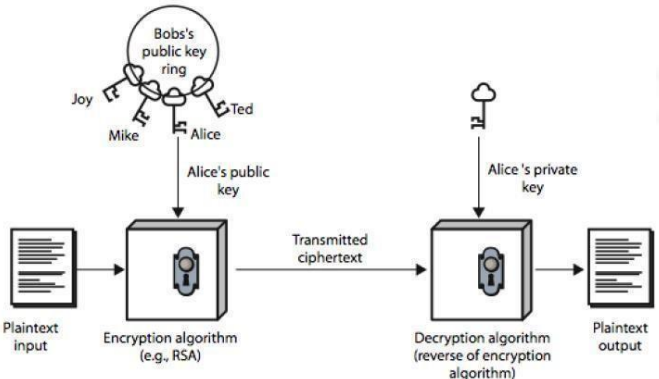
a*public-key*,whichmaybeknownby anybody,andcanbeusedto**encrypt**

- **messages**,and**verifysignatures**
- a*private-key*,knownonlytotherecipient,usedto**decryptmessages**,and **sign** (create)**signatures**.
- is **asymmetric**becausethosewhoencryptmessagesor verifysignaturescannotdecryptmessagesorcreatesignatures

Public-Keyalgorithmsrelyononekeyforencryptionandadifferentbutrelatedkeyfordecryption.Thesealgorithmshavethefollowingimportantcharacteristics:

- itiscomputationallyinfeasibletofinddecryptionkeyknowingonlyalgorithm&encryptionkey
- it is computationallyeasyto en/decryptmessages when therelevant(en/decrypt)keyisknown
- eitherofthetworelatedkeyscanbeusedforencryption,withtheotherusedfordecryption (forsomealgorithmslikeRSA)

The following figure illustrates public-key encryption process and shows that a public-keyencryptionschemehassixingredients:plaintext,encryptionalgorithm,public&privatekeys,ciphertext&decryption algorithm.



The essentialstepsinvolved in apublic-key encryption schemearegiven below:

1.)Eachusergeneratesa pairofkeystobeusedforencryptionanddecryption.

2.)Eachuserplacesoneofthetwokeysinapublicregisterandtheotherkeyiskeptprivate.

3.)IfBwantsto sendaconfidentialmessagetoA,BencryptsthemessageusingA'spublickey.

4.) When A receives the message, she decrypts it using her private key. Nobody elsecandecryptthemessagebecausethatcanonlybedoneusingA'sprivatekey(Deducing aprivatekeyshouldbeinfeasible).

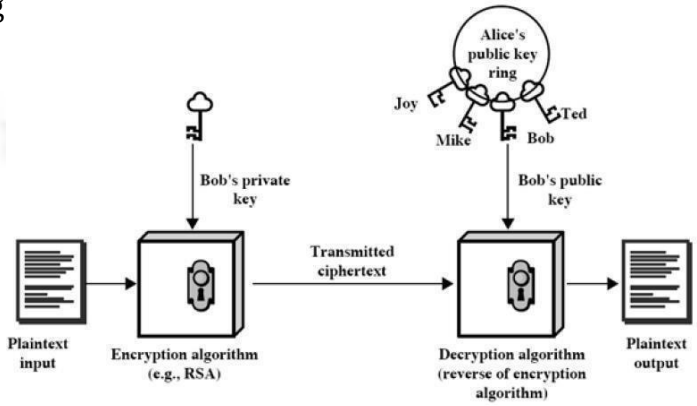
5.) If a user wishes to change his keys –generate another pair of keys and publish thepublicone:nointeractionwithotherusersisneeded.

NotationsusedinPublic-keycryptography:

- Thepublic keyofuserAwillbedenoted**KUA**.
- TheprivatekeyofuserAwillbedenoted**KRA**.
- EncryptionmethodwillbeafunctionE.
- DecryptionmethodwillbeafunctionD.
- If B wishes tosenda plainmessage XtoA,thenhesendsthecryptotext $Y=E(KU_A,X)$

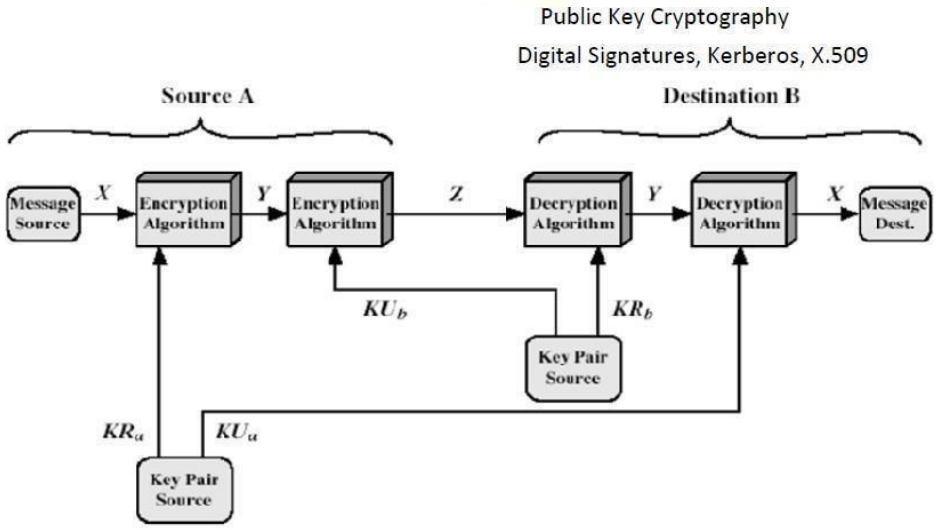
The intended receiverAwilldecryptthemessage:$D(KR_A,Y)=X$

The first attack on Public-key Cryptography is the attack on Authenticity. **An attackermayimpersonateuserB**:hesendsamessage$E(KU_A,X)$andclaimsinthemessagetobe B–A has no guarantee this is so. To overcome this, B will encrypt the message using hisprivatekey: $Y=E(KR_B,X)$.Receiver decryptsusingB'spublickeyKRB.Thisshowstheauthenticity of the sender because (supposedly) he is the only one who knows the privatekey.Theentireencryptedmessageservesasadigitalsignature.Thisschemeisdepictedin thefollowing:



Authentication

But, a drawback still exists. Anybody can decrypt the message using B's public key. So,secrecy or confidentiality is being compromised. One can provide both *authentication andconfidentiality*usingthepublic-keyschemetwice:



Public-Key Cryptosystem: Secrecy and Authentication

B encrypts X with his private key: Y=E(KRB,X) B encrypts Y with A's public key:Z=E(KUA,Y)
AwilldecryptZ(andsheistheonlyonecapableofdoingit):Y=D(KRA,Z)
A can now get the plaintext and ensure that it comes from B (he is theonly one whoknowshisprivatekey):decryptYusingB'spublickey:X=E(KUB,Y).

Applicationsforpublic-keycryptosystems

1.)**Encryption/decryption**:senderencryptsthemessagewiththereceiver'spublickey.

2.)**Digitalsignature**:sender"signs"themessage(orarepresentativepartofthemessage)usinghisprivatekey

3.)**Keyexchange**:twosidescooperatetoexchangeasecretkeyforlateruseinasecret-keycryptosystem.

ThemainrequirementsofPublic-key cryptography are:

1. ComputationallyeasyforapartyBtogenerateapair(publickeyKUb,privatekeyKRb).
2. EasyforsenderAtogenerateciphertext:

3. EasyforthereceiverBtodecryptciphertectusingprivatekey:
4. Computationallyinfeasible todetermineprivate key(KRb) knowingpublic key(KUb)
5. ComputationallyinfeasibletorecovermessageM,knowingKUbandciphertextC
6. Eitherofthetwokeyscanbeusedforencryption,withtheotheru sedfordecryption:

$$M=D_{KRb}[E_{KUb}(M)]=D_{KUb}[E_{KRb}(M)]$$

Easyisdefinedtomeanaproblemthatcanbesolvedinpolynomialtimeasafunctionofinputlength.A problemisinfeasibleiftheeffortto solveitgrowsfasterthanpolynomialtime as a function of input size. Public-key cryptosystems usually rely on difficult mathfunctions rather than S-P networks as classical cryptosystems. **One-way function** is one,easy to calculate in one direction, infeasible to calculate in the other direction (i.e., theinverse is infeasible to compute). **Trap-door function** is a difficult function that becomeseasyifsomeextrainformationisknown.Ouraimtofinda***trap-doorone-wayfunction***,which is easy to calculate in one direction and infeasible to calculate in the other directionunlesscertain additionalinformationisknown.

SecurityofPublic-keyschemes:

- Likeprivatekeyschemesbruteforce**exhaustivesearch**attackisalwaystheoreticallypossible.Butkeysusedaretoolarge(>512bits).
- Securityreliesona**largeenough**differenceindifficultybetween**easy**(en/decrypt)and**hard**(cryptanalyse)problems.Moregenerallythe**hard** problemisknown,itsjustmadetoohardtodoinpractise.
- Requirestheuseof**verylargenumbers,**henceis**slow**comparedto privatekey schemes

RSAALGORITHM

RSA is the best known, and by far the most widely used general public keyencryption algorithm, and was first published by Rivest, Shamir & Adleman of MIT in1978 [RIVE78]. Since that time RSA has reigned supreme as the most widely acceptedandimplementedgeneral-purposeapproachtopublic-keyencryption.TheRSAschemeisablockcipherinwhichtheplaintextandthecipher textareintegersbetween 0 and n- 1 for some fixed n and typical size for n is 1024 bits (or 309 decimaldigits). It is based on exponentiation in a finite (Galois) field over integers modulo

aprime,usinglargeintegers(eg.1024bits).Itssecurity isdueto thecostoffactoring large numbers. RSA involves a public-key and a private-key where the public key isknown to all and is used to encrypt data or message. The data or message which hasbeen encrypted using a public key can only be decryted by using its correspondingprivate-key.Eachusergeneratesakeypair

publicandprivatekeyusingthefollowingsteps:

- *eachuserselectstwolargeprimes atrandom-p,q*
- *computetheirsystemmodulusn=p.q*
- *calculateø(n),where**ø(n)=(p-1)(q-1)***
- *selectingatrandomtheencryptionkeye,where1<e<ø(n),and**gcd(e,ø(n))=1***
- *solvefollowingequationtofinddecryptionkeyd:**e.d=1modø(n)**and0≤d≤n*
- *publishtheirpublicencryptionkey:**KU={e,n}***
- *keepsecretprivatedecryptionkey**:KR={d,n}***

Both the sender and receiver must know the values of n and e, and only the receiverknowsthevalueofd.EncryptionandDecryptionaredoneusingthefollowin gequations.ToencryptamessageMthesender:

– obtains**publickey**ofrecipient*KU={e,n}*

– computes:**C=Memodn**,where0≤M<nTodecryptthecipherthetextCthe owner:

– usestheirprivatekey*KR={d,n}*

– computes:**M=Cd modn=(Me)dmodn=Medmodn**

Forthisalgorithmtobesatisfactory,thefollowingrequirementsaretobemet.

a) Itspossibletofindvaluesofe,d,nsuchthat**Med=Mmodn**forallM<n

b) ItisrelativelyeasytocalculateMeandCforallvaluesofM<n.

c) Itisimpossibletodeterminedgiveneandn

The way RSA works is based on Number theory: **Fermat's little theorem**: if **p**is prime and **a** is positive integer not divisible by **p**, then **ap-1 ≡ 1 mod p. Corollary**:Foranypositiveinteger**a**andprime**p**,**ap≡amodp.**

Fermat's theorem, as useful as will turn out to be does not provide us withintegers d,e we are looking for –Euler's theorem (a refinement of Fermat's) does.Euler's function associates to any positive integer **n,** a number **φ(n)**: the number ofpositiveintegerssmallerthan**n** andrelativelyprimeto**n.**Forexample, **φ(37)=36**i.e. **φ(p) = p-1** for any prime **p.** For any two primes **p,q**, **φ(pq)=(p-1)(q-1). Euler'stheorem**: for any relatively prime integers a,n we have **aφ(n)≡1 mod n. Corollary:F**oranyintegersa,nwehave**aφ(n)+1≡amodnCorollary:**Letp,qbetwo oddprimesandn=pq.Then:φ(n)=(p-1)(q-

1) For any integer m with 0<m<n, m(p-1)(q-1)+1 ≡ m mod n For any integers k,mwith 0<m<n, mk(p-1)(q-1)+1 ≡ m mod n Euler's theorem provides us the numbers d,esuchthatMed=Mmodn.Wehavetochoosed,esuchthated=kφ(n)+1,oreq uivalently,d≡e-1modφ(n)

AnexampleofRSAcanbegivenas,Selectprimes:*p*=17&*q*=11Compute*n*=*pq*=17×11=18 7

Compute ø(*n*)=(*p*–1)(*q*-1)=16×10=160 Select e *:* gcd(e,160)=1; choose *e*=7Determined*:de*=1mod160and*d*<160Valueisd=23since23×7=161=10×16 0+1

PublishpublickeyKU={7,187}

KeepsecretprivatekeyKR={23,187}Now,givenmessageM=88(nb.88 <187)encryption:C=887mod187=11

decryption:M=1123mod18 7=88AnotherexampleofRS Aisgivenas,

Let**p=11,q=13,e=11,m=7**

n=pqi.e. n=11*13=143

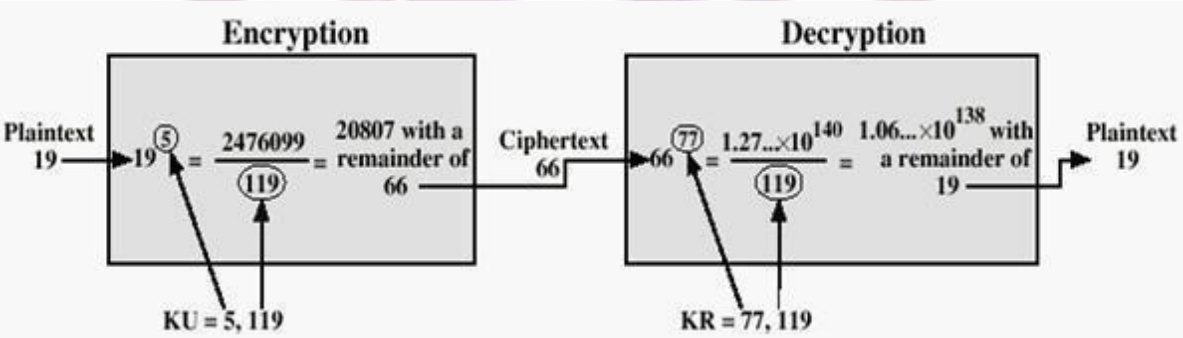
ø(n)=(p-1)(q-1)i.e.(11-1)(13-1)=120

e.d=1modø(n)i.e.11dmod120=1i.e.(11*11)mod120=1;sod=11publickey

:{11,143}andprivatekey:{11,143}

C=Memodn,sociphertext=711mod143=727833mod143;i.e.C=106

M=Cdmodn,plaintext=10611mod143=1008mod143;i.e.M=7



determinetwoprimesatrandom-p,q

- selecteithereordandcomputetheother

- meansmustbesufficientlylarge
- typicallyguessanduseprobabilistictest

SecurityofRSA

TherearethreemainapproachesofattackingRSAalgorithm.

Bruteforcekeysearch(infeasiblegivensizeofnumbers)Asexplainedbefore,inv olvestryingallpossibleprivatekeys.Bestdefenceisusinglargekeys.

Mathematicalattacks(basedondifficultyofcomputingø(N),byfactoringmodulus N)Thereareseveralapproaches,allequivalentineffecttofactoringtheproductoft woprimes.Someofthem aregivenas:

- factorN=p.q,hencefindø(N) andthend
- determineø(N)directlyandfindd
- findddirectly

Thepossibledefensewouldbeusinglargekeysandalsochoosinglargenumbersforp and q, which should differ only by a few bits and are also on the order of magnitude10_{75}to10_{100}.Andgcd(p-1,q-1)shouldbesmall.

Elgamal Encryption Algorithm

Elgamal encryption is a public-key cryptosystem. It uses asymmetric key encryption for communicating between two parties and encrypting the message. This cryptosystem is based on the difficulty of finding **discrete logarithm** in a cyclic group that is even if we know g^a and g^k, it is extremely difficult to compute g^{ak}.

Idea of Elgamal cryptosystem

Suppose Alice wants to communicate with Bob.

1. Bob generates public and private keys:
   - Bob chooses a very large number **q** and a cyclic group $\mathbf{F}_q$.
   - From the cyclic group $\mathbf{F}_q$, he choose any element **g** and an element **a** such that gcd(a, q) = 1.
   - Then he computes $h = g^a$.
   - Bob publishes **F**, $\mathbf{h = g^a}$, **q**, and **g** as his public key and retains **a** as private key.
2. Alice encrypts data using Bob's public key :
   - Alice selects an element **k** from cyclic group **F** such that gcd(k, q) = 1.
   - Then she computes $p = g^k$ and $s = h^k = g^{ak.}$
   - She multiples s with M.
   - Then she sends $(p, M*s) = (g^k, M*s)$.
3. Bob decrypts the message :

- Bob calculates $s' = p^a = g^{ak}$.
- He divides M*s by s' to obtain M as $s = s'$.

Following is the implementation of the ElGamal cryptosystem in Python Python3

```
# Python program to illustrate Elgamal encryption


import random
from math import pow


a = random.randint(2, 10)


def gcd(a, b):
    if a < b:
        return gcd(b, a)
    elif a % b == 0:
        return b;
    else:
        return gcd(b, a % b)


# Generating large random numbers
def gen_key(q):


    key = random.randint(pow(10, 20), q)
    while gcd(q, key) != 1:
        key = random.randint(pow(10, 20), q)


    return key
```

```
# Modular exponentiation
def power(a, b, c):
    x = 1
    y = a

    while b > 0:
        if b % 2 != 0:
            x = (x * y) % c;
        y = (y * y) % c
        b = int(b / 2)

    return x % c

# Asymmetric encryption
def encrypt(msg, q, h, g):

    en_msg = []

    k = gen_key(q)# Private key for sender
    s = power(h, k, q)
    p = power(g, k, q)

    for i in range(0, len(msg)):
        en_msg.append(msg[i])

    print("g^k used : ", p)
```

```
    print("g^ak used : ", s)
    for i in range(0, len(en_msg)):
        en_msg[i] = s * ord(en_msg[i])

    return en_msg, p

def decrypt(en_msg, p, key, q):

    dr_msg = []
    h = power(p, key, q)
    for i in range(0, len(en_msg)):
        dr_msg.append(chr(int(en_msg[i]/h)))

    return dr_msg

# Driver code
def main():

    msg = 'encryption'
    print("Original Message :", msg)

    q = random.randint(pow(10, 20), pow(10, 50))
    g = random.randint(2, q)

    key = gen_key(q)# Private key for receiver
    h = power(g, key, q)
    print("g used : ", g)
```

```
    print("g^a used : ", h)


    en_msg, p = encrypt(msg, q, h, g)
    dr_msg = decrypt(en_msg, p, key, q)
    dmsg = ''.join(dr_msg)
    print("Decrypted Message :", dmsg);


if __name__ == '__main__':
    main()
```

SampleOutput :

Original Message : encryption

g used : 586069695452241770718895237154794403533331590789O

g^a used : 471130975563936428955245483450621514465395805525 2

g^k used : 124751880895032276157890157407090919114125671267 82

g^ak used : 394487876321671361611533372266549063577567400682 95

Decrypted Message : encryption

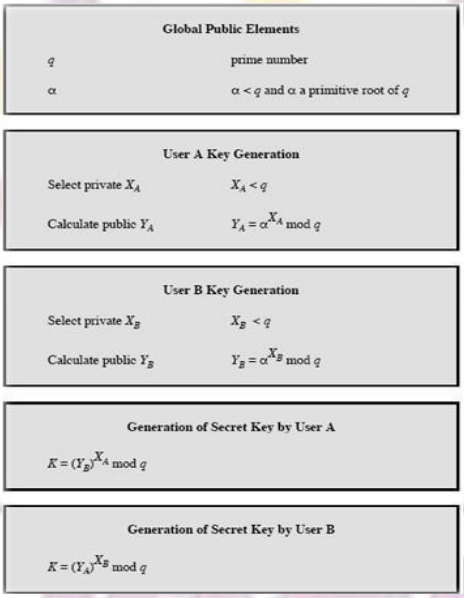
In this cryptosystem, the original message **M** is masked by multiplying $\mathbf{g^{ak}}$ to it. To remove the mask, a clue is given in form of $\mathbf{g^k}$. Unless someone knows **a**, he will not be able to retrieve **M**. This is because finding discrete log in a cyclic group is difficult and simplifying knowing g^a and g^k is not good enough to compute g^{ak}.

DIFFIE-HELLMANKEYEXCHANGE

Diffie-Hellman key exchange(**D-H**)is acryptographicprotocol that allowstwoparties that have no prior knowledge of each other to jointly establish a shared secretkey over an insecure communications channel. This key can then be used to encryptsubsequentcommunicationsusingasymmetrickeycipher.TheD-Halgorithmdependsforitseffectivenessonthedifficultyofcomputingdiscretelogarithms.

First, a primitive root of a prime number p, can be defined as one whose powersgenerate all the integers from 1 to p-1. If a is a primitive root of the prime number p,then the numbers, ***a mod p, a_2 mod p,..., a_{p-1} mod p***, are distinct and consist of theintegersfrom 1throughp1 in somepermutation. Foranyinteger***b***andaprimitiveroot***a***ofprimenumber***p***,wecanfindauniqueex ponent

isuch $b \equiv a^i \pmod{p}$ where $0 \leq i \leq (p-1)$ that .Theexponentiisreferre dtoasthediscrete logarithm of b for the base a, mod p. We express this value as ***$dlog_{a,p}$ (b).*** Thealgorithm issummarizedbelow:



For this scheme, there are two publicly known numbers: a prime number q and aninteger α that is a primitive root of q. Suppose the users A and B wish to exchange akey. User A selects a random integer X_A< q and computes Y_A = α^{XA} mod q. Similarly,userBindependentlyselectsarandomintegerXA<qandcomputesYB=αXB modq.

Each side keeps the X value private and makes the Y value available publicly to theother side. User A computes the key as K = $(Y_B)^{XA}$ mod q and user B computes the keyasK=$(Y_A)^{XB}$mod

DiscreteLogProblem

The (discrete) exponentiation problem is as follows: Given a base a, an exponent band a modulus p, calculate c such that $a_b \equiv c \pmod{p}$ and $0 \leq c < p$. It turns out that thisproblem is fairly easy and can be calculated "quickly" using

fast-exponentiation. Thediscrete log problem is the inverse problem: Given a base a, a result c ($0 \leq c < p$) and amodulusp,calculate the exponent b such that $ab \equiv c \pmod{p}$. It turns out that no one hasfound a quick way to solve this problem With DLP, if P had 300 digits, X_a and X_b havemore than 100 digits, it would take longer than the life of the universe to crack themethod.

ExamplesforD-Hkeydistributionscheme:

1) Letp=37andg=13.

Let Alice pick a = 10. Alice calculates 13_{10} (mod 37) which is 4 and sends that to Bob.Let Bob pick b = 7. Bob calculates 13_7 (mod 37) which is 32 and sends that to Alice.(Note: 6 and 7 are secret to Alice and Bob, respectively, but both 4 and 32 are knownbyall.)

10(mod37)whichis30,thesecretkey.

7(mod37)whichis30,thesamesecretkey.

2) Let p = 47 and g = 5. Let Alice pick a = 18. Alice calculates 5_{18} (mod 47) which is 2and sends that to Bob. Let Bob pick b = 22. Bob calculates 5_{22} (mod 47) which is 28 andsendsthattoAlice.

18(mod47)whichis24,thesecretkey.

22(mod47)whichis24,thesamesecretkey

Man-in-the-MiddleAttackonD-Hprotocol

SupposeAliceandBobwishtoexchangekeys,andDarthistheadversary.Theattac kproceedsasfollows:

1. Darthprepares forthe attack bygenerating tworandomprivatekeys X_{D1} and $X_{D2}$and thencomputingthecorresponding publickeysY_{D1}andY_{D2}.
2. Alicetransmits$Y_A$toBob.
3. Darth intercepts Y_A andtransmits Y_{D1} toBob. Darth alsocalculates K2 $=(Y_A)_{XD2}$modq.
4. BobreceivesY_{D1}andcalculatesK1=$(Y_{D1})_{XE}$modq.
5. Bobtransmits$X_A$toAlice.
6. DarthinterceptsX$_A$andtransmitsY_{D2}toAlice.DarthcalculatesK1=$(Y_B)_{XD1}$modq.
7. AlicereceivesY_{D2}andcalculatesK2=$(Y_{D2})_{XA}$modq.

At this point, Bob and Alice think that they share a secret key, but instead Bob

andDarthsharesecretkeyK1andAliceandDarthsharesecretkeyK2.Allfuturecom municationbetweenBobandAliceiscompromisedinthefollowingway:

1. Alicesendsanencryptedmessage M:E(K2,M).
2. Darthinterceptstheencryptedmessageanddecryptsit,torecoverM.

3. Darth sends Bob E(K1, M) or E(K1, M'), where M' is any message. In the first case,Darth simply wants to eavesdrop on the communication without altering it. In thesecondcase,Darthwantstomodifythemessagegoing toBob. Thekeyexchangeprotocolisvulnerabletosuchanattackbecauseitdoesnotaut henticate the participants. This vulnerability can be overcome with theuse ofdigitalsignaturesand public-keycertificates.

ELLIPTICCURVECRYPTOGRAPHY(ECC)

Elliptic curve cryptography (ECC) is an approach to public-key cryptography basedon the algebraic structure of elliptic curves over finite fields. The use of elliptic curvesin cryptography was suggested independently by Neal Koblitz and Victor S. Miller in1985.TheprincipalattractionofECCcomparedtoRSAisthatitappearstoofferequalsecur ityforafarsmallerbitsize,therebyreducingthe processingoverhead.

EllipticCurveoverGF(p)

LetGF(p)beafinitefield,p>3,andleta,b

- GF(p) are constant such that ***4a3 + 27b2 ≡ 0 (mod p).*** An elliptic curve,E(a,b)(GF(p)),isdefinedasthesetofpoints(x,y)cGF(p)*GF(p)whic hsatisfytheequation

y2≡x3+ax +b(modp),togetherwithaspecialpoint,O,*calledthepointatinfinity*.LetPan d Qbe twopoints onE(a,b)(GF(p))andOisthe pointatinfinity.

- P+O=O+P=P
- IfP=(x1,y1)then-P=(x1,-y1)andP+(-P)= O.
- IfP=(x1,y1)andQ=(x2,y2),andPandQ

arenotO.thenP+Q=(x3,y3)where

x3=λ 2-x1-x2

y3=λ(x1-x3)-y1and

λ=(y2-y1)/(x2-x1)ifP≠Q

λ=(3x12+a)/2y1ifP=Q

$$\lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1} \; for \_ x_1 \neq x_2 \\ \dfrac{3x_1^{\,2} + a}{2y_1} \; for \_ x_1 = x_2 \end{cases}$$

An elliptic curve may be defined over any finite field GF(q). For GF(2m), the curve has adifferent form:$-y^2 + xy = x^3 + ax^2 + b$, *where* $b \neq 0$.

CryptographywithEllipticCurves

The addition operation in ECC is the counterpart of modular multiplication in RSA, andmultiple addition is the counterpart of modular exponentiation. To form a cryptographicsystem using elliptic curves, some kind of hard problem such as discrete logarithm orfactorization of prime numbers is needed. Considering the equation, Q=kP, where Q,P arepoints in an elliptic curve, it is "easy" to compute Q given k,P , but "hard" to find k givenQ,P. This is known as the elliptic curve logarithm problem. k could be so large as to makebrute-forcefail.

ECCKeyExchange

Pick a prime number p= 2180 and elliptic curve parameters a and b for the equation$y^2 \equiv x^3 + ax + b \pmod p$ which defines the elliptic group of points Ep(a,b).

SelectgeneratorpointG=(x1,y1)inEp(a,b)suchthatthesmallestvalueforwhichnG =Obeaverylargeprimenumber.Ep(a,b)andGareparametersofthecryptosystemknow ntoallparticipants.Thefollowingstepstakeplace:

- A&BselectprivatekeysnA<n,nB<n
- computepublickeys:PA=nA×G,PB=nB×G
- Computesharedkey:K=nA×PB,K=nB×PA{samesinceK=nA×nB×G}

ECCEncryption/Decryption Aswithkeyexchangesystem,anencryption/decry ption system requires a point G and and elliptic group Ep(a,b) asparameters. First thing to be done is to encode the plaintext message m to be sent asanx-ypoint**Pm**.EachuserchoosesprivatekeynA<nandcomputespublickeyPA=nA×G . To encrypt and send a message to Pm to B, A chooses a random positiveintegerkandproducesthecipher text**Cm**consistingofthepairofpoints**Cm ={kG,Pm+kPb}.**here,AusesB'spublickey.Todecrypt the ciphertext, B multiplies the first point in the pair by B's secret key andsubtractstheresultfromthesecondpoint$\mathbf{Pm}+k\mathbf{Pb}-\mathbf{nB}(k\mathbf{G})=\mathbf{Pm}+k(\mathbf{nB}\mathbf{G}) - \mathbf{nB}(k\mathbf{G}) = \mathbf{Pm}$ A has masked the message **Pm** by adding k**Pb** to it. Nobody but Aknows the value of k, so even though **Pb** is a public key, nobody can remove the maskk**Pb.** For an attacker to recover the message, he has to compute k given G and kG,whichisassumedhard.

Securityof **ECC**To**protect** a128bitAESkeyitwouldtakeaRSAKeySizeof3072bitswhereasanECC KeySizeof256bits.

Computational Effort for Cryptanalysis of Elliptic Curve Cryptography Compared to RSA

Key Size	MIPS-Years
150	3.8×10^{10}
205	7.1×10^{18}
234	1.6×10^{28}

(a) Elliptic Curve Logarithms using the Pollard rho Method

Key Size	MIPS-Years
512	3×10^{4}
768	2×10^{8}
1024	3×10^{11}
1280	1×10^{14}
1536	3×10^{16}
2048	3×10^{20}

(b) Integer Factorization using the General Number Field Sieve

ApplicationsofECC:

- Wirelesscommunicationdevices
- Smartcards
- Webserversthatneed tohandlemanyencryptionsessions
- Any application where security is needed but lacks the power,storageandcomputationalpowerthatisnecessaryforourcurrent cryptosystems.

Knapsack Encryption Algorithm

Knapsack Encryption Algorithm is the first general public key cryptography algorithm. It is developed by **Ralph Merkle** and **Mertin Hellman** in 1978. As it is a Public key cryptography, it needs two different keys. One is Public key which is used for Encryption process and the other one is Private key which is used for Decryption process. In this algorithm we will two different knapsack problems in which one is easy and other one is hard. The easy knapsack is used as the private key and the hard knapsack is used as the public key. The easy knapsack is used to derived the hard knapsack.

For the easy knapsack, we will choose a **Super Increasing knapsack problem**. Super increasing knapsack is a sequence in which every next term is greater than the sum of all preceding terms.

Example –

{1, 2, 4, 10, 20, 40} is a super increasing as

1<2, 1+2<4, 1+2+4<10, 1+2+4+10<20 and 1+2+4+10+20<40.

1.Encryption :

As our knapsacks contain six values, so we will split our plain text in a groups of six:

100100 111100 101110

Multiply each values of public key with the corresponding values of each group and take their sum.

100100 {31, 62, 14, 90, 70, 30}

1x31+0x62+0x14+1x90+0x70+0x30 = 121

111100 {31, 62, 14, 90, 70, 30}

1x31+1x62+1x14+1x90+0x70+0x30 = 197

101110 {31, 62, 14, 90, 70, 30}

1x31+0x62+1x14+1x90+1x70+0x30 = 205

So, our cipher text is 121 197 205.

2.Decryption :

The receiver receive the cipher text which has to be decrypt. The receiver also knows the values of m and n.

So, first we need to find the , which is multiplicative inverse of n mod m i.e.,

n x mod(m) = 1

31 x mod(110) = 1

Now, we have to multiply 71 with each block of cipher text take modulo m.

121 x 71 mod(110) = 11

Then, we will have to make the sum of 11 from the values of private key {1, 2, 4, 10, 20, 40} i.e.,
1+10=11 so make that corresponding bits 1 and others 0 which is 100100. Similarly,

197 x 71 mod(110) = 17

1+2+4+10=17 = 111100

And, 205 x 71 mod(110) = 35

1+4+10+20=35 = 101110

After combining them we get the decoded text.

100100111100101110 which is our plai

DescriptiveQuestions:

(a) 2MarksQuestions

Definecryptanalysisandcryptology.

Cryptanalysis: techniques used for deciphering or decrypting a message without the knowledge oftheencipheringorencryptingdetailsissaidtobecryptanalysis.Cryptology:thestudyofcryptographyandcryptanalysistogetheriscalledcryptology.

2. **Mentionthevarioustypesofcryptanalyticattack.**
   - Knownplaintext
   - Ciphertextonly
   - Chosenplaintext
   - ChosenCiphertext

3. **Definesymmetrickeycryptographyandpublickeycryptography.**

Insymmetrickeycryptography,onlyonekeyisusedforencryptionanddecryption.

In public key cryptography, two keys (public key and private key) are used. When one keyis used for encryption, then the other must be used for decryption. The public key is known to alltheparticipantsbuttheprivatekeyiskeptsecretbytheowner.

4. **DefinePrimalityTest**.

A primality testing is a test to determine whether or not a given number is prime, asopposed to actually decomposing the number into its constituent prime factors (which is known asprimefactorization).

5. **DefineEuler'stotientfunction(usedinRSAalgorithm).**

Itisthenumber ofpositiveintegersthatarelessthan'n'andrelativelyprimeto'n'.Whenistheproductoftwo primenumbers(p&q)

ItisrepresentedasΦ(n)anditisexpressedasΦ(n)=Φ(pq)=(p-1)(q-1).

6. **WhatarethevariousapproachestoattackstheRSAalgorithm?**
   - Bruteforceattack
   - Mathematicalattacks

- Timingattacks

7. DefineEuler'stheoremandit'sapplication.

Euler'stheoremstatesthatfor
everyaandnthatarerelativelyprime:aϕ(n)≡≡1modn

8. Findgcd(1970,1066)usingEuclid'salgorithm?

gcd(1970,1066)=gcd(1066,1970mod1066)
=gcd(1066,904)=2

9. Findgcd(24140,16762)usingEuclid'salgorithm?
gcd(24140,16762)=gcd(16762,24140mod16762)
=gcd(16762,7378)=34

WhydoweneedDiffieHellmanalgorithm?

It is used for exchanging the secret keys between the sender and the receiver. It allows twousersto exchangeakeysecurely.

11.Whatis anellipticcurve?

Itisaplanealgebraiccurvedefinedbyanequationofthe form$y2=x3+ax+b$thatisnon-singularalsograph hasno cuspsorselfintersections

(b) 10MarksQuestions

1. Writeshortnoteson:i)Linearcongruenceii)Exponentialanddiscretelogarithm

2. A)ExplainindetailaboutElgamalCryptosystemandChineseRemaindertheorem
   B)InRSAsystem,thepublickeyofgivenusere=31,n=3599whatistheprivatekeyofuser?

3. A)Whatarethedifferentalgorithmsinwhichprimenumbersareused?
   B)Explain RSA algorithm.AndperformEncryption andDecryption using RSAp=3 q=11e=7M=5

4. ExplainDiffie-Hellmankeyexchangealgorithmindetail.

5. A)Whataretherequirementsofpublickeycryptography?
   B)Explainthesymmetrickeycryptographyandasymmetrickeycryptographywith anexample.

6. Findtheresultof$(x^5+x^2+x)tiply(x^7+x^4+x^3+x^2+x)$inGF$(2^8)$withirreduciblepolynomial$x^8+x^4+x^3+x+1$

7. ExplainprocessandAlgorithmofExtendedEuclideanforGCD?Givena=1 61b=28,findgcd(a,b)andthevaluesofsandtbyusingExtendedEuclideanAlg orithm?

8. A)ExplainEuler'sphiFunctionandFermat'slittletheorem?
   B)ExplainEllipticcurvecryptographyandMillerRabinalgorithmwithanexample problem 6.

UNIT-3

CRYPTOGRAPHIC HASH FUNCTIONS

MESSAGE AUTHENTICATION

Data is prone to various attacks. One of these attacks includes message authentication. This threat arises when the user does not have any information about the originator of the message. Message authentication can be achieved using cryptographic methods which further make use of keys.

- **Revelation:** It means releasing the content of the message to someone who does not have an appropriate cryptographic key.
- **Analysis of Traffic:** Determination of the pattern of traffic through the duration of connection and frequency of connections between different parties.
- **Deception:** Adding out of context messages from a fraudulent source into a communication network. This will lead to mistrust between the parties communicating and may also cause loss of critical data.
- **Modification in the Content:** Changing the content of a message. This includes inserting new information or deleting/changing the existing one.
- **Modification in the sequence:** Changing the order of messages between parties. This includes insertion, deletion, and reordering of messages.
- **Modification in the Timings:** This includes replay and delay of messages sent between different parties. This way session tracking is also disrupted.
- **Source Refusal:** When the source denies being the originator of a message.
- **Destination refusal:** When the receiver of the message denies the reception.

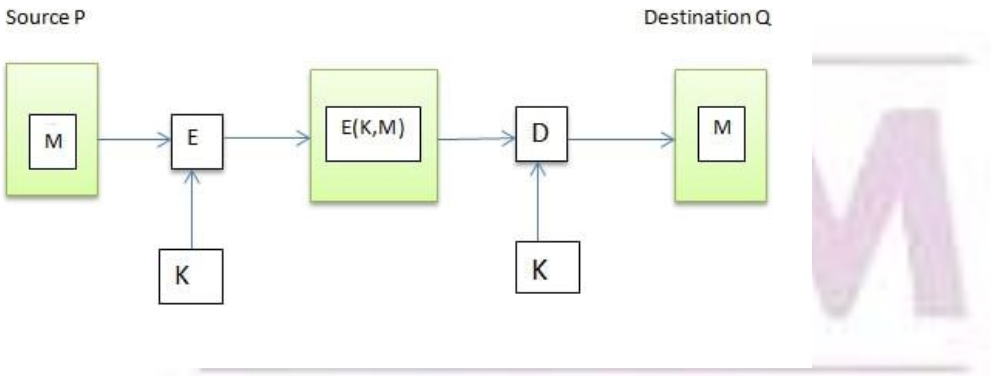
Message Authentication Functions:

All message authentication and digital signature mechanisms are based on two functionality levels:

- **Lower level:** At this level, there is a need for a function that produces an authenticator, which is the value that will further help in the authentication of a message.
- **Higher-level:** The lower level function is used here in order to help receivers verify the authenticity of messages.

These message authentication functions are divided into three classes:

- **Message encryption:** While sending data over the internet, there is always a risk of a Man in the middle(MITM) attack. A possible solution for this is to use message encryption. In message encryption, the data is first converted to a ciphertext and then sent any further. Message encryption can be done in two ways:
- **Symmetric Encryption:** Say we have to send the message M from a source P to destination Q. This message M can be encrypted using a secret key K that both P and Q share. Without this key K, no other person can get the plain text from the ciphertext. This maintains confidentiality. Further, Q can be sure that P has sent the message. This is because other than Q, P is the only party who possesses the key K and thus the ciphertext can be decrypted only by Q and no one else. This maintains authenticity. At a very basic level, symmetric encryption looks like this:



- **Public key Encryption:** Public key encryption is not as advanced as symmetric encryption as it provides confidentiality but not authentication. To provide both authentication and confidentiality, the private key is used.
- **Message authentication code (MAC):** A message authentication code is a security code that the user of a computer has to type in order to access any account or portal. These codes are recognized by the system so that it can grant access to the right user. These codes help in maintaining information integrity. It

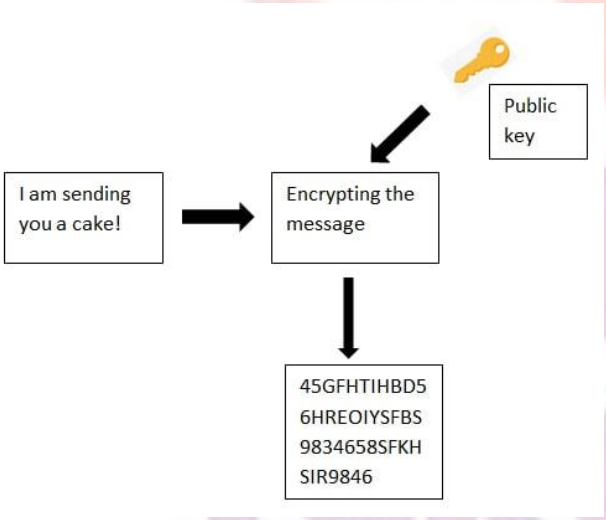
also confirms the authenticity of the message.

- **Hash function:** A hash function is nothing but a mathematical function that can convert a numeric value into another numeric value that is compressed. The input to this hash function can be of any length but the output is always of fixed length. The values that a hash function returns are called the message digest or hash values.

Measures to deal with these attacks:

Each of the above attacks has to be dealt with differently.

Message Confidentiality: To prevent the messages from being revealed, care must be taken during the transmission of messages. For this, the message should be encrypted before it is sent over the network.



- **Message Authentication:** To deal with the analysis of traffic and deception issues, message authentication is helpful. Here, the receiver can be sure of the real sender and his identity. To do this, these methods can be incorporated:
    - Parties should share secret codes that can be used at the time of identity authentication.
    - Digital signatures are helpful in the authentication.
    - A third party can be relied upon for verifying the authenticity of parties.
- **Digital Signatures:** Digital signatures provide help against a majority of these issues. With the help of digital signatures, content, sequence, and timing of the

messages can be easily monitored. Moreover, it also prevents denial of message transmission by the source.

- **Combination of protocols with Digital Signatures:** This is needed to deal with the denial of messages received. Here, the use of digital signature is not sufficient and it additionally needs protocols to support its monitoring.

MESSAGEAUTHENTICATION CODES

Message authentication is a procedure to verify that received messages comefrom the alleged source and have not been altered. Message authentication may alsoverifysequencingandtimeliness.Itisintendedagainsttheattackslikecontentmodification,sequencemodification,timingmodificationandrepudiation.Forrepudiation, concept of digital signatures is used to counter it. There are three classesby which different types of functions that may be used to produce an authenticator.Theyare:

Messageencryption–theciphertextservesas authenticator

Messageauthenticationcode(MAC)–apublicfunctionofthemessageandasecret key producing a fixed-length value to serve as authenticator. This does notprovideadigitalsignaturebecauseAand Bsharethesamekey.

Hashfunction–apublicfunctionmappinganarbitrarylengthmessageintoa fixed- length hash value to serve as authenticator. This does not provide a digitalsignaturebecausethereisnokey.

MESSAGEENCRYPTION:

Message encryption by itself can provide a measure of authentication. The analysisdiffers for conventional and public-key encryption schemes. The message must havecomefromthesenderitself,becausetheciphertextcanbedecryptedusing his(secretor public) key. Also, none of the bits in the message have been altered because anopponentdoes notknow how to manipulatethebits of the ciphertext to inducemeaningfulchangestotheplaintext.Oftenoneneedsalternativeauthenticationschemesthanjustencryptingthemessage.

Sometimesoneneedstoavoidencryptionoffullmessagesduetolegal requirements.

Encryptionandauthenticationmaybeseparatedinthesystemarchitecture.

Thedifferentwaysinwhichmessageencryptioncanprovideauthentication,confidentialityinbothsymmetricandasymmetricencryptiontechniquesisexplainedwith thetablebelow:

Confidentiality and Authentication Implications of Message Encryption

A → B: $E_K[M]$ •Provides confidentiality —Only A and B share K •Provides a degree of authentication —Could come only from A —Has not been altered in transit —Requires some formatting/redundancy •Does not provide signature —Receiver could forge message —Sender could deny message (a) Symmetric encryption
A → B: $E_{KU_b}[M]$ •Provides confidentiality —Only B has KR_b to decrypt •Provides no authentication —Any party could use KU_b to encrypt message and claim to be A (b) Public-key encryption: confidentiality
A → B: $E_{KR_a}[M]$ •Provides authentication and signature —Only A has KR_a to encrypt —Has not been altered in transit —Requires some formatting/redundancy —Any party can use KU_a to verify signature (c) Public-key encryption: authentication and signature
A → B: $E_{KU_b}[E_{KR_a}(M)]$ •Provides confidentiality because of KU_b •Provides authentication and signature because of Kr_a (d) Public-key encryption: confidentiality, authentication, and signature

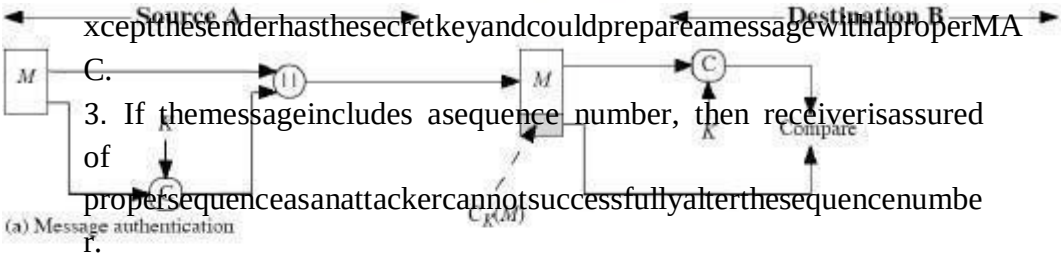
MESSAGEAUTHENTICATIONCODE

An alternative authentication technique involves the use of a secrete key to generate a small fixed- size block of data, known as cryptographic checksum or MAC,whichisappendedtothemessage.Thistechnique assumesthatboththe communicating parties say A and B share a common secret key K. When A has a messageto sendtoB,itcalculatesMACasafunctionCofkeyandmessagegivenas:

MAC=Ck(M) Themessage and the MAC are transmitted to the intended recipient, who upon receiving performsthe same calculation on the received message, using the

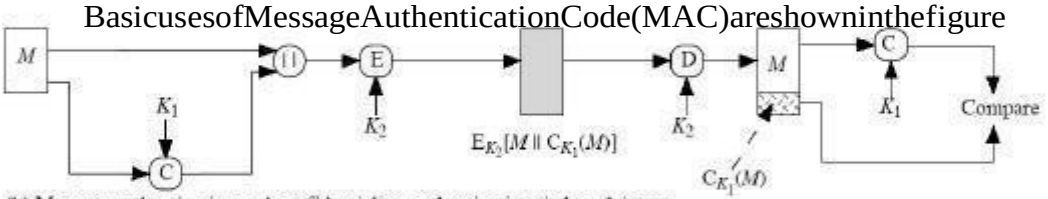
same secret key to generate anew MAC. The received MAC is compared to the calculated MAC and only if theymatch,then:

1. The receiver is assuredthatthemessage hasnotbeen altered: Any alternationsbeendonetheMAC'sdonotmatch.
2. Thereceiverisassuredthatthemessageisfromtheallegedsender:Noonee xceptthesenderhasthesecretkeyandcouldprepareamessagewithaproperMA C.
3. If themessageincludes asequence number, then receiverisassured of propersequenceasanattackercannotsuccessfullyalterthesequencenumbe r.

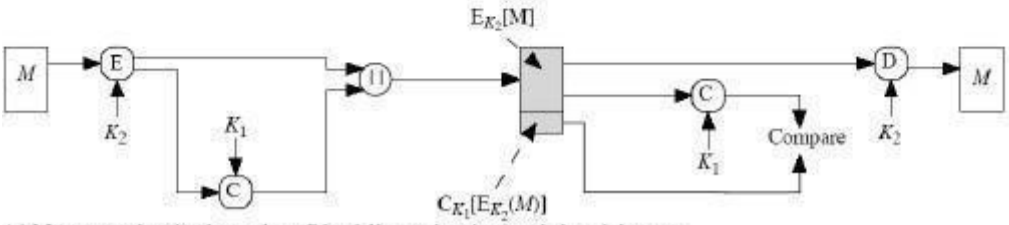


(a) Message authentication

BasicusesofMessageAuthenticationCode(MAC)areshowninthefigure



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

TherearethreedifferentsituationswhereuseofaMACisdesirable:

If a message is broadcast to several destinations in a network (such as a militarycontrolcenter),thenitischeaperand morereliabletohavejustonenoderesponsibletoevaluatetheauthenticity– messagewillbesentinplainwithanattachedauthenticator.

If one side has a heavy load, it cannot afford to decrypt all messages –it will justchecktheauthenticityofsomerandomlyselectedmessages.

Authentication of computer programs in plaintext is very attractive service as theyneed not be decrypted every time wasting of processor resources.

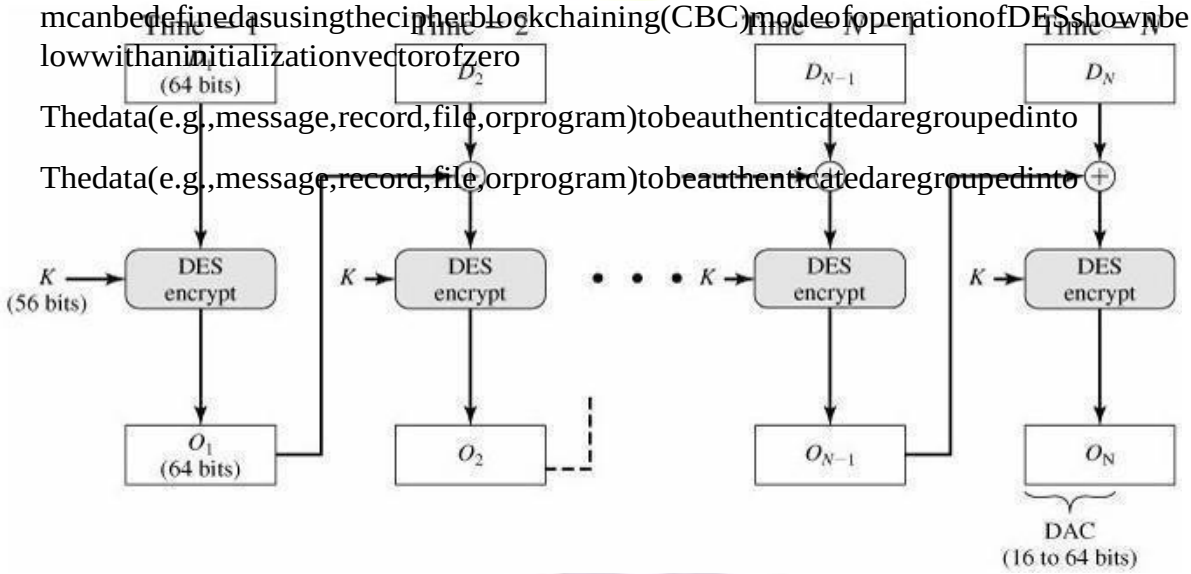
Integrity of theprogramcan alwaysbecheckedbyMAC.

MESSAGEAUTHENTICATION CODEBASED ON DES

TheDataAuthenticationAlgorithm,basedonDES,hasbeenoneofthemostwidelyused MACs for a number of years. The algorithm is both a FIPS publication (FIPS PUB113)and an ANSIstandard (X9.17).But,security weaknesses in this algorithm havebeendiscoveredanditisbeingreplacedbynewerandstrongeralgorithms.Thealgorith mcanbedefinedasusingthecipherblockchaining(CBC)modeofoperationofDESshownbe lowwithaninitializationvectorofzero

Thedata(e.g.,message,record,file,orprogram)tobeauthenticatedaregroupedinto

Thedata(e.g.,message,record,file,orprogram)tobeauthenticatedaregroupedinto



$O_1 = E(K, D_1)$

$O_2 = E(K, [D_2 \oplus O_1])$

$O_3 = (K, [D_3 \oplus O_2])$

•

•

•

$O_N = E(K, [D_N \oplus O_{N1}])$

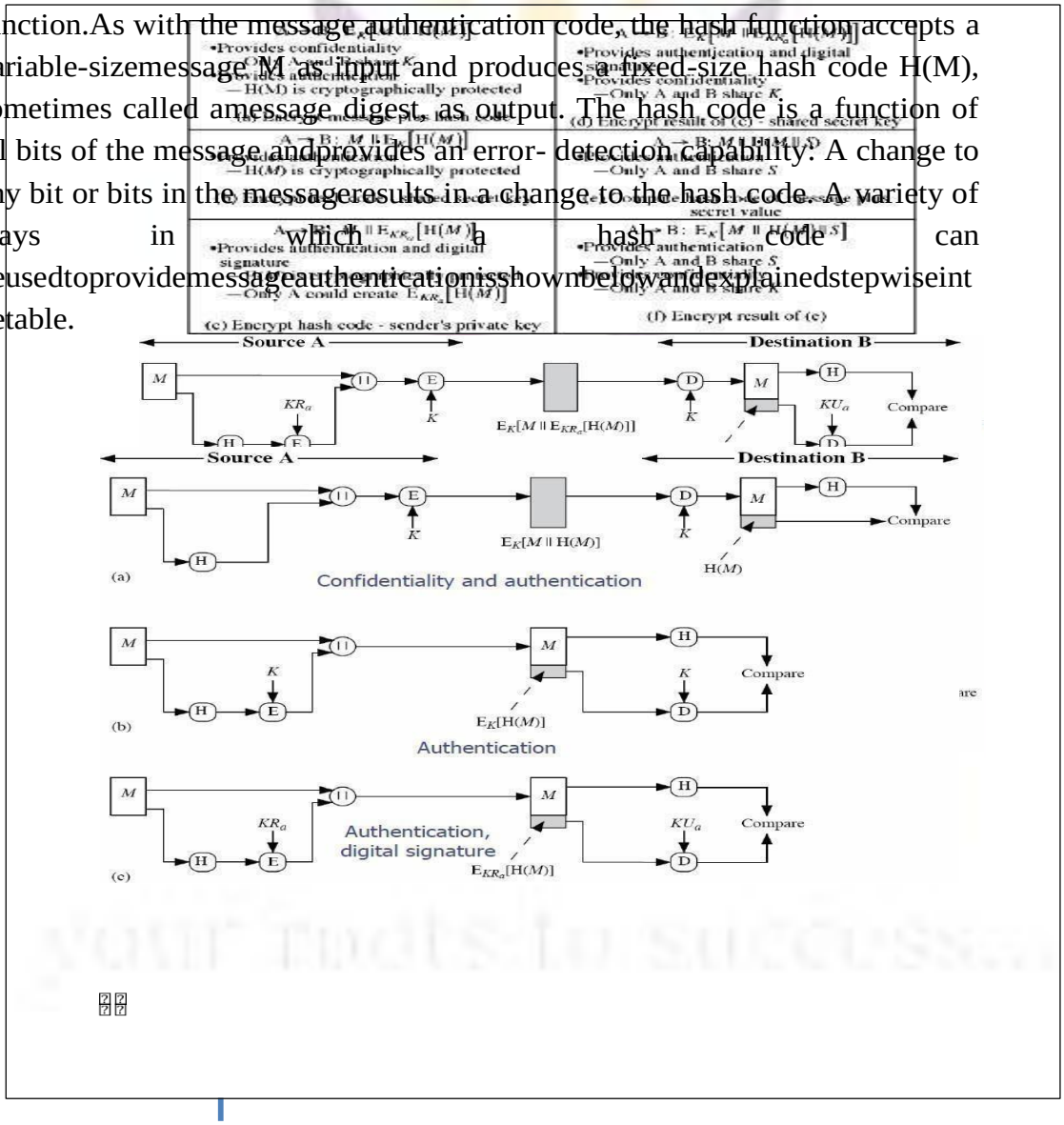
contiguous 64-bit blocks: D1, D2,..., DN. If necessary, the final block is padded on therightwith zeroes toform a full 64-bit block.Usingthe

DESencryption algorithm,E, andasecretkey,K,adataauthenticationcode(DAC)iscalculatedasfollows: TheDACconsistsofeithertheentireblockONortheleftmostMbitsoftheblock,with 16≤M≤64

UseofMACneedsasharedsecretkeybetweenthecommunicatingparties and also MAC does not provide digital signature. The following tablesummarizestheconfidentialityandauthenticationimplicationsoftheapproaches shownabove.

HASH FUNCTION

A variation on the message authentication code is the one-way hash function.As with the message authentication code, the hash function accepts a variable-sizemessage M as input and produces a fixed-size hash code H(M), sometimes called amessage digest, as output. The hash code is a function of all bits of the message andprovides an error- detection capability: A change to any bit or bits in the messageresults in a change to the hash code. A variety of ways in which a hash code can beusedtoprovidemessageauthenticationisshownbelowandexplainedstepwiseint hetable.



In case where confidentiality is not required methods b and c have an advantage over those that encrypt the entire message in less that computation is required. Growing interest for technique that avoid encryption is due to reason like encryption software is quiet slow and may be covered by patents. Also encryption hardware costs are not negligible and the algorithms are subjected to U.S export control. A fixed-length hash value h is generated by a function H that takes as input a messageofarbitrarylength**:h=H(M).**sendsMandH(M)

A authenticates themessage bycomputing H(M) and checking the matchB

Requirementsforahashfunction: Thepurposeofahashfunctionisto producea"fingerprint"of afile,message,or otherblockofdata.Tobeusedformessage authentication, the hash function H must have the following properties canbeappliedtoamessageofanysizeproducesfixed-lengthoutput

H ComputationallyeasytocomputeH(M)foranygivenM

ComputationallyinfeasibletofindMsuchthatH(M)=h,foragivenh, referredtoas The*one-wayproperty*

ComputationallyinfeasibletofindM'suchthatH(M')=H(M),foragivenM,referredtoas*weak collisionresistance*.

Computationallyinfeasible tofind M,M' with H(M)=H(M')(toresisttobirthdayattacks),referredtoas*strongcollisionresistance*.

Examplesofsimplehashfunctionsare:

- Bit-by-bitXORofplaintextblocks:h=D1⊕D2⊕…⊕DN
- RotatedXOR– beforeeachadditionthehashvalueisrotatedtotheleftwith1bit
- Cipherblockchainingtechniquewithoutasecretkey.

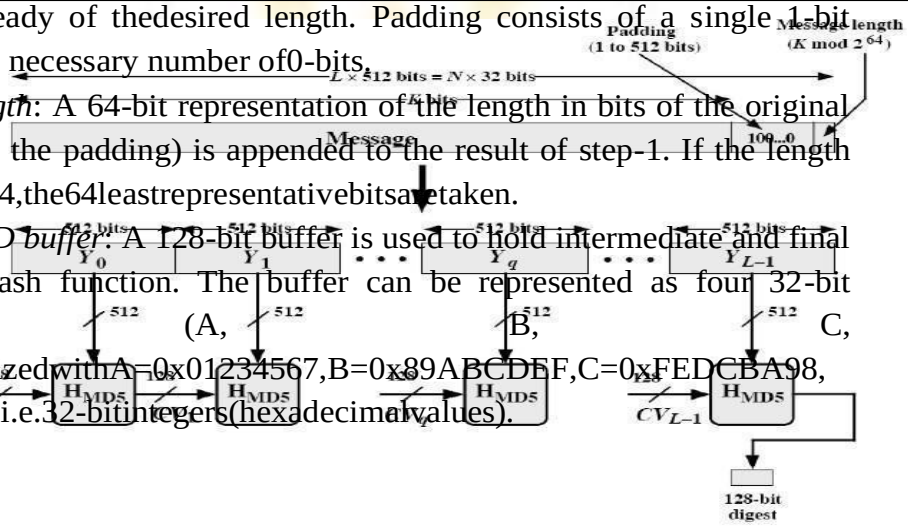
MD5MESSAGEDIGESTALGORITHM

The MD5 message-digest algorithm was developed by Ron Rivest at MIT and itremained as the most popular hash algorithm until recently. The algorithm takes asinput, a message of arbitrary length and produces as output, a 128-bit message digest.The input is processed in 512-bit blocks. The processing consists of the followingsteps:

1.) *Append Padding bits*: The message is padded so that its length in bits is congruentto 448 modulo 512 i.e. the length of the padded message is 64 bits less than an integermultiple of 512 bits. Padding is always added, even if the message is already of thedesired length. Padding consists of a single 1-bit followed by the necessary number of0-bits.

2.) *Append length*: A 64-bit representation of the length in bits of the original message(before the padding) is appended to the result of step-1. If the length is larger than264,the64leastrepresentativebitsaretaken.

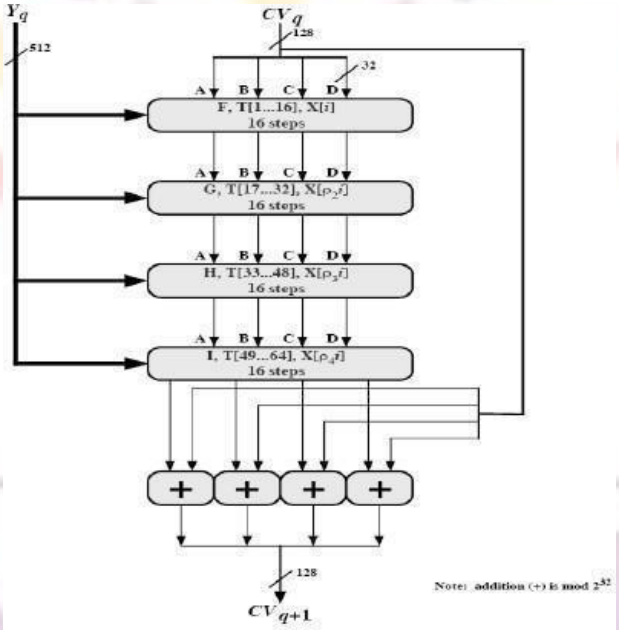
3.) *Initialize MD buffer*: A 128-bit buffer is used to hold intermediate and final resultsof the hash function. The buffer can be represented as four 32-bit registers (A, B, C, D)andareinitializedwithA=0x01234567,B=0x89ABCDEF,C=0xFEDCBA98, D=0x76543210i.e.32-bitintegers(hexadecimalvalues).



Message Digest Generation Using MD5

4.)*ProcessMessagein512-bit(16-word)blocks*:Theheartofalgorithmisthecompression function that consists of four rounds of processing and this module islabeled HMD5 in the above figure and logic is illustrated in the following figure. Thefourroundshaveasimilarstructure,buteachusesadifferentprimitivelogicalfun

ction, referred to as F, G, H and I in the specification. Each block takes as input thecurrent 512-bit block being processed Yq and the 128-bit buffer value ABCD andupdates the contents of the buffer. Each round also makes use of one-fourth of a 64-element table T*1….64+, constructed from the sine function. The ith element of T,denoted T[i], has the value equal to the integer part of 232 * abs(sin(i)), where i is inradians. As the value of abs(sin(i)) is a value between 0 and 1, each element of T is aninteger that can be represented in 32-bits and would eliminate any regularities in theinputdata.Theoutputoffourthroundisaddedtotheinputtothefirstround(CVq)t oproduce CVq+1. The addition is done independently for each of the four words in thebuffer with each of the corresponding words in CVq, using addition modulo 232. Thisoperation isshown inthefigurebelow:



5.)*Output*:AfterallL512-bitblockshavebeenprocessed,theoutputfromtheLthstageisthe 128-bit messagedigest.MD5canbesummarized as follows:

$CV0=IVCV_{q+1}=SUM_{32}(CV_q,RF_IY_qRF_H[Y_q,RF_G[Y_q,RF_F[Y_q,CV_q]]]])MD=CV_L$

Where,

IV=initialvalueofABCDbuffer,definedinstep3.Y_q=theq_{th}512-bitblockofthemessage

L=thenumberofblocksinthemessage

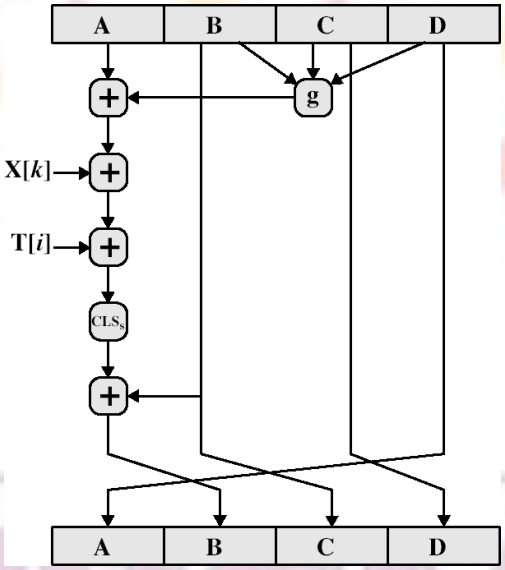
CV_q=chainingvariableprocessedwiththeq_{th}blockofthemessage.

RFx=roundfunctionusingprimitivelogicalfunctionx.

MD=finalmessagedigestvalue
SUM_{32} =Additionmodulo2_{32}performedseparately

MD5CompressionFunction:

EachroundconsistsofasequenceofstepsoperatingonthebufferABCD.Each step isoftheform, **a=b+((a+g(b,c,d)+X[k]+T[i])<<<s)**where a, b, c, d refer to the four words of the buffer but used in varying permutations.After 16 steps, each word is updated 4 times. g(b,c,d) is a different nonlinear functionineachround(F,G,H,I).ElementaryMD5operationofasinglestepisshownbelow.



Round	Primitive function g	g(b, c, d)
1	F(b, c, d)	(b ∧ c) ∨ (b' ∧ d)
2	G(b, c, d)	(b ∧ d) ∨ (c ∧ d')
3	H(b, c, d)	b ⊕ c ⊕ d
4	I(b c, d)	c ⊕ (b ∨ d')

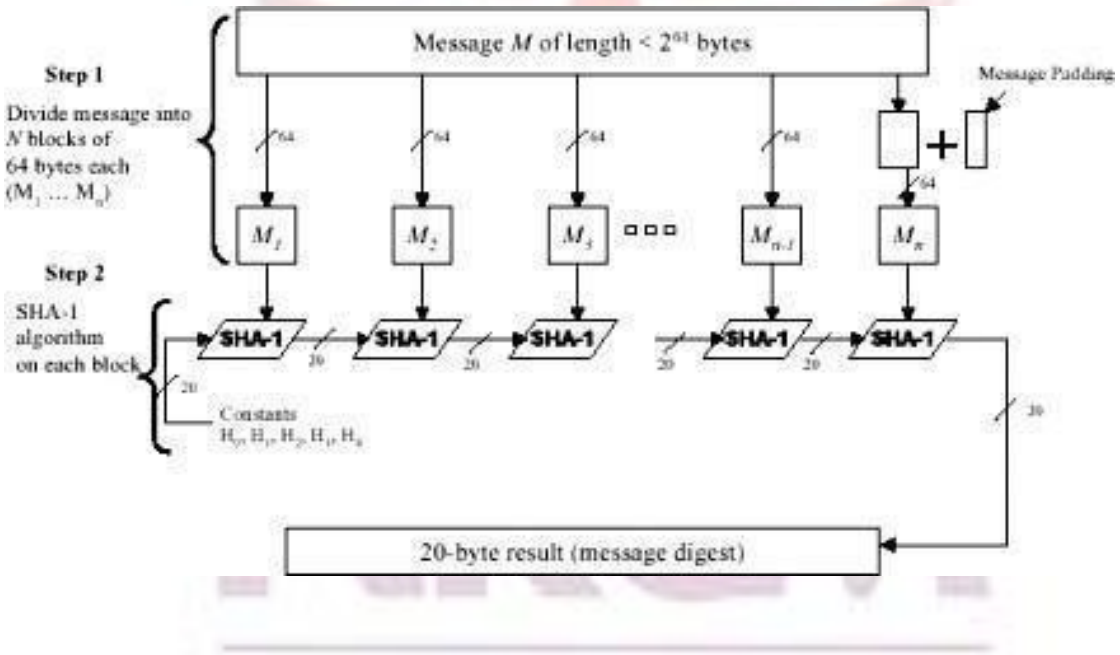
Truth table

b	c	d	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

Each round mixes the buffer input with the next "word" of the message in a

complex,non- linear manner. A different non-linear function is used in each of the 4 rounds(but the same function for all 16 steps in a round). The 4 buffer words (a,b,c,d) arerotated from steptostep soall areused andupdated.g is oneof theprimitivefunctions F,G,H,I for the 4 rounds respectively. X[k] is the kth 32-bit word in thecurrent message block. T[i] is the ith entry in the matrix of constants T. The additionof varying constants T and the use of different shifts helps ensure it is extremelydifficult to compute collisions. The array of 32-bit words X[0..15] holds the value ofcurrent 512-bit input block being processed. Within a round, each of the 16 words ofX[i] is used exactly once, during one step. The order in which these words is usedvariesfrom roundtoround.Inthefirstround,thewordsareusedintheiroriginalorder.Forrounds 2through4,thefollowingpermutationsareused

p2(i)=(1+5i)mod16p3(i)
=(5+3i)mod16p4(I)
= 7imod16



MD4

PrecursortoMD5DesigngoalsofMD4(whicharecarried over to MD5) SecuritySpeed
Compactness
Favorlittle-
Architecture

Eachstepnowhasauniqueadditiveconstant.

- Thefunctioninround2waschangedfrom(bcvbdvcd)to(bdvcd')tomakeglesssy mmetric.

Eachstepnowaddsintheresultofthepreviousstep.Thispromotesafaster"avala ncheeffect".

Theorderinwhichinputwordsareaccessedinrounds2and3ischanged,tomake thesepatternslesslikeeachother.

Theshiftamountsineachroundhavebeenapproximatelyoptimized,toyieldafaste r"avalanche effect."Theshiftsindifferentrounds aredistinct.

SECUREHASH ALGORITHM

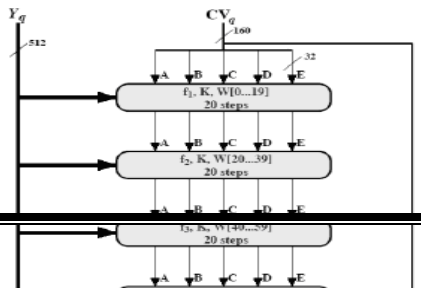
Thesecurehashalgorithm(SHA)wasdevelopedbytheNationalInstituteofStandar ds and Technology (NIST). SHA-1 is the best established of the existing SHAhash functions, and is employed in several widely used security applications andprotocols. The algorithm takes as input amessage with a maximum length of lessthan264bitsandproducesasoutputa160-bitmessagedigest The input is processed in 512-bit blocks. The overall processing of a message follows thestructure of MD5 with block length of 512 bits and a hash length and chaining variablelengthof160bits.Theprocessingconsistsoffollowingsteps:

1.) ***Append Padding Bits:*** The message is padded so that length is congruent to 448modulo 512; padding always added–one bit 1 followed by the necessary number of 0bits.

2.) ***Append Length:*** a block of 64 bits containing the length of the original message isadded. 3.) ***Initialize MD buffer*****:** A 160-bit buffer is used to hold intermediate and finalresults on the hash function. This is formed by 32-bit registers A,B,C,D,E. Initial values:A=0x67452301,B=0xEFCDAB89,C=0x98BADCFE,D=0x1032547 6,E=C3D2E1F0.Stores inbig-endianformat i.e.themostsignificantbitinlowaddress.

4.) ***Process message in blocks 512-bit (16-word) blocks*****:** The processing of a single 512-bit block is shown above. It consists of four rounds of processing of 20 steps each. Thesefour rounds have similar structure, but uses a different primitive logical function, whichwe refer to as f1, f2, f3 and f4. Each round takes as input the current 512-bit block beingprocessed and the 160-bit buffer value ABCDE and updates the contents of the buffer.Each round also makes use of four distinct additive constants Kt. The output of the fourthround i.e.eightiethstep isaddedtotheinputtothefirstround toproduceCV_{q+1}.

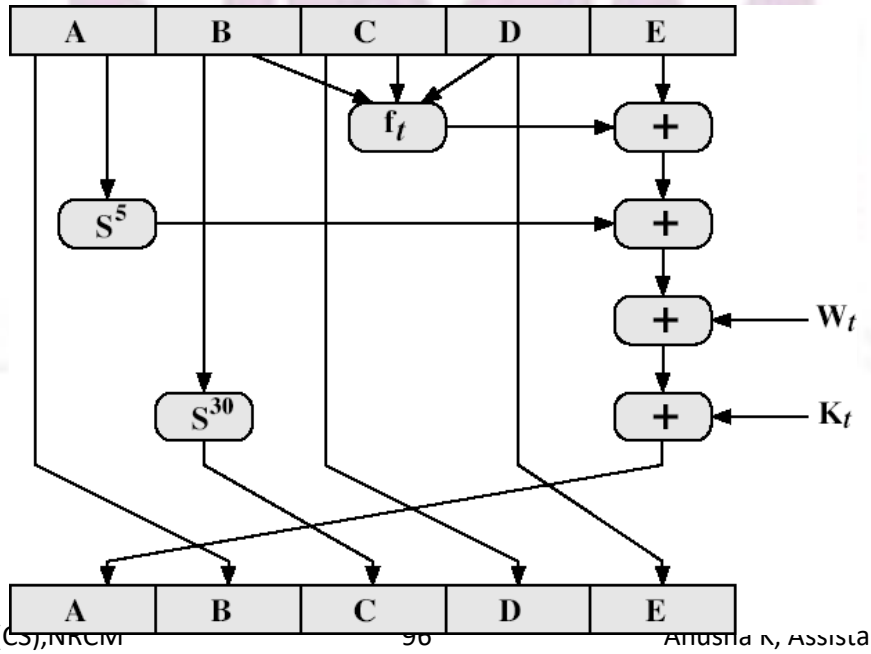
5.) *Output:* After all L 512-bit blocks have been processed, the output from the Lthstageisthe160-bitmessagedigest.



The behavior of SHA-1 is as follows: **CV0 = IV CVq+1 = SUM32(CVq, ABCDEq) MD =CVL** Where, IV = initial value of ABCDE buffer ABCDEq = output of last round ofprocessing of qthmessage block L =number ofblocks in the message SUM32 =Additionmodulo232MD=finalmessagedigestvalue.

SHA-1CompressionFunction:

Each round has 20 steps which replaces the 5 buffer words. The logic present in eachone of the 80 rounds present is given as **(A,B,C,D,E) <- (E + f(t,B,C,D) + S_5(A)+ Wt+Kt),A,S_{30}(B),C,D** Where, A, B, C, D, E = the five words of the buffer t = step number; 0<t< 79 f(t,B,C,D) = primitive logical function for step t S_k = circular left shift of the 32-bitargument by k bits Wt = a 32-bit word derived from current 512-bit input block. Kt =anadditive constant;fourdistinctvalues are used+=moduloaddition.

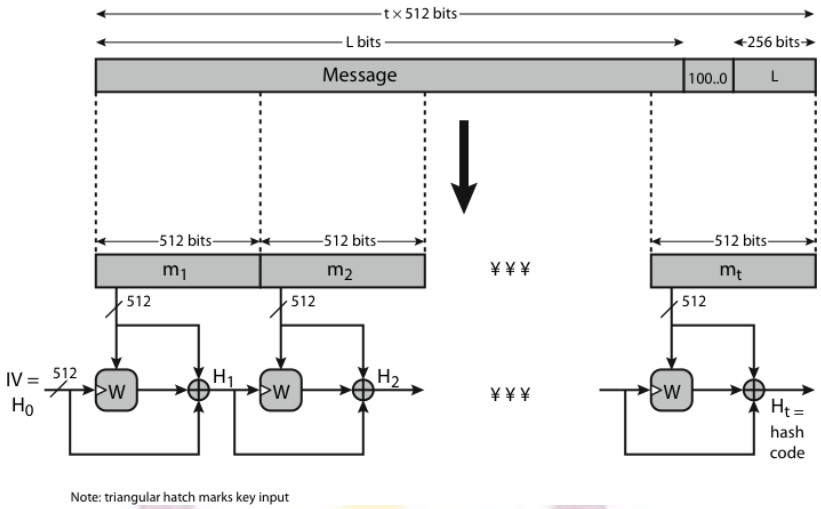


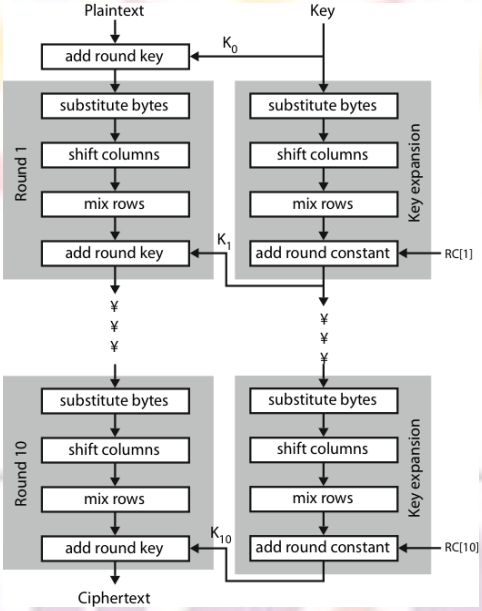
SHA shares much in common with MD4/5, but with 20 instead of 16stepsineachofthe4rounds.Notethe4constantsarebasedonsqrt(2, 3,5,10). Note also that instead of just splitting the inputblockinto 32-bit words and using them directly, SHA-1 shuffles and mixesthem using rotates & XOR's to form a more complex input, and greatlyincreasesthedifficultyoffindingcollisions.Asequenceoflogicalfunctions f0, f1,..., f79 is used in the SHA-1. Each f**t**, 0<=t<=79, operates onthree32-bitwordsB,C,Dandproducesa32-bitwordasoutput.ft(B,C,D) is defined as follows: for words B, C, D, **ft(B,C,D) = (B AND C) OR((NOTB)ANDD)(0<=t<=19)ft(B,C,D)=BXOR CXORD(20<=t <=39)ft(B,C,D)=(BANDC)OR(BANDD)OR(CANDD)(40<=t<=59)ft(B,C,D) =B XORCXORD(60 <=t<=79).**

WHIRLPOOLHASHFUNCTION

- Created byVincentRijmenandPauloS.L.M.Barreto
- Hashesmessagesofplaintextlength2^256
- Result isa512bitmessage
- Threeversionshavebeenreleased–WHIRLPOOL-0– WHIRLPOOL-T– WHIRLPOOL
  - designedspecificallyforhashfunctionuse
  - withsecurityandefficiencyofAES
  - butwith512-bitblocksizeandhencehash
  - similarstructure&functionsasAESbut
    - inputismappedrowwise
    - has10rounds
    - adifferentprimitivepolynomialforGF(2^8)
    - usesdifferentS-boxdesign&values
- "W"isa512-bitblockcipher
- "m"istheplaintext,splitinto512bitblocks
- "H"istheblocksformedfromthehashes

WHIRLPOOLOVERVIEW





TheblockcipherWisthecoreelementoftheWhirlpoolhashfunction

- Itiscomprisedof4steps.
  - AddRoundKey
  - ShiftColumns
  - MixRows
  - Substitutebytes

AddRoundKey

- DuringtheAddRoundKeystep,themessageisXOR'dwiththekey
- Ifthisisthefirstmessageblockbeingrunthrough,thekeyisablockofallzeros
- Ifthisisanyblockexceptthefirst,thekeyisthedigestofthepreviousblock

ShiftColumns

• Startingfromlefttoright,eachcolumngetsrotatedvertically anumberofbytesequaltowhichnumbercolumnitis,fromtoptobottom–

Ex: MixRows

• [0,0][0,1][0,2][0,0][2,1][1,2]

• [1,0][1,1][1,2]------>[1,0][0,1][2,2]

• [2,0][2,1][2,2][2,0][1,1][0,2]

• Eachrowgetsshiftedhorizontallybythenumberofrowitis.Similar to the shift column function, but rotated left to right –Ex:

- [0,0][0,1][0,2][0,0][0,1][0,2]
- [1,0][1,1][1,2]------>[1,2][1,0][1,2]
- [2,0][2,1][2,2][2,1][2,2][0,2]

Substitutebytes

• Eachbyteinthemessageispassedthroughasetofs-boxes

• Theoutputofthisisthensettobethekeyforthenextround

Authentication Requirements

In the context of communications across a network, the following attacks can be identified:

1. Disclosure: Release of message contents to any person or process not possessing the appropriate cryptographic key.
2. Traffic analysis: Discovery of the pattern of traffic between parties. In a connection-oriented application, the frequency and duration of connections could be determined. In either a connection-oriented or connectionless environment, the number and length of messages between parties could be determined.
3. Masquerade: Insertion of messages into the network from a fraudulent source. This includes the creation of messages by an opponent that are purported to come from an authorized entity. Also included are fraudulent acknowledgments of message receipt or nonreceipt by someone other than the message recipient.
4. Content modification: Changes to the contents of a message, including insertion, deletion, transposition, and modification.
5. Sequence modification: Any modification to a sequence of messages between parties, including insertion, deletion, and reordering.
6. Timing modification: Delay or replay of messages. In a connection-oriented application, an entire session or sequence of messages could be a replay of some previous valid session, or individual messages in the sequence could be delayed or replayed. In a connectionless application, an individual message (e.g., datagram) could be delayed or replayed.
7. Source repudiation: Denial of transmission of message by source.
8. Destination repudiation: Denial of receipt of message by destination.

Measures to deal with the first two attacks are in the realm of message confidentiality and are dealt with in Part One. Measures to deal with items 3 through 6 in the foregoing list are generally regarded as message authentication. Mechanisms for dealing specifically with item 7 come under the heading of digital signatures. Generally, a digital signature technique will also counter some or all of the attacks listed under items 3 through 6. Dealing with item 8 may require a combination of the use of digital signatures and a protocol designed to counter this attack.

In summary, message authentication is a procedure to verify that received messages come from the alleged source and have not been altered. Message authentication may also verify sequencing and timeliness. A digital signature is an authentication technique that also includes measures to counter repudiation by the source.

HMAC

Interest in developing a MAC, derived from a cryptographic hash codehas been increasing mainly because hash functions are generally fasterand are alsonot limited by export restrictions unlike block ciphers.Additional reason also would be that the library code for cryptographichashfunctionsiswidelyavailable.Theoriginalproposalisforincorporation of a secret key into an existing hash algorithm and theapproach that received most support is HMAC. HMAC is specified asInternetstandardRFC2104.Itmakes use of the hash function on the given message. Any of MD5,SHA-1,RIPEMD-160can beused.

HMACDesignObjectives

To use, without modifications, available hash functionsToallowforeasyreplaceabilityoftheembedded hash function To preserve theoriginal performance of the hash functionTouseandhandlekeysinasimplewayTohaveawellunderstoodcryptographicanalysisofthestrengthofthe MAC based on reasonable assumptions on the embedded hashfunctionThe first two objectives are very important for the acceptability of HMAC. HMACtreats the hash function as a “black box”, which has two benefits. First is that anexisting implementation of the hash function can be used for implementing HMACmaking the bulk of HMAC code readily available without modification. Second is thatifeveranexistinghashfunctionisto bereplaced,theexistinghashfunctionmoduleisremoved and new module is dropped in. The last design objective provides the mainadvantage of HMAC over other proposed hash-based schemes. HMAC can be provensecureprovidedthattheembeddedhashfunctionhassomereasonablecryptographic strengths.

StepsinvolvedinHMACalgorithm:

1. Appendzeroesto the left endof K to create ab-bitstring

K_+ (ex: If Kisoflength 160-
bitsandb=512,thenKwillbeappendedwith44zero
bytes).

2. XOR(bitwiseexclusive-OR)K_+withipadtoproducetheb-bitblockS_i.
3. Append M to S_i.
4. Now apply H to the stream generated in step-3
5. XOR K_+ with opad to produce the b-bit block S_0.
6. Append the hash result from step-4 to S_0.
7. Apply H to the stream generated in step-6 and output the result.

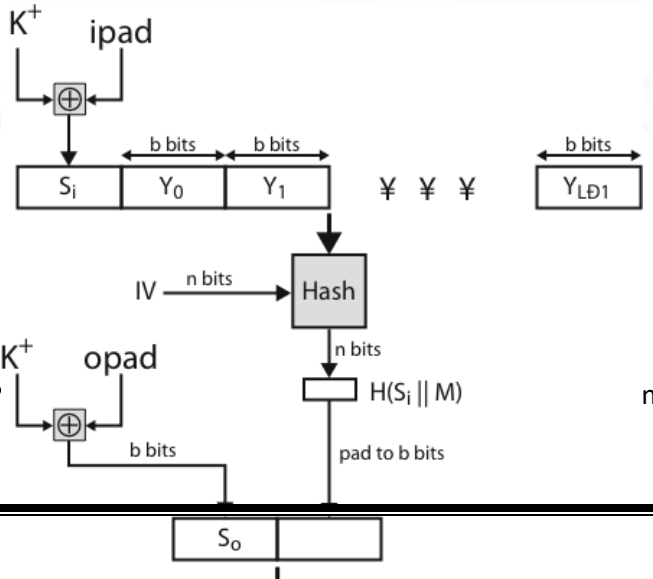
HMAC Algorithm

- **Define the following terms**

H	= embedded hash function
M	= message input to HMAC
Y_i	= i^{th} block of M, $0 \le i \le L - 1$
L	= number of blocks in M
b	= number of bits in a block
n	= length of hash code produced by embedded hash function
K	= secret key; if key length is greater than b, the key is input to the hash function to produce an n-bit key; recommended length $\ge$ n
K^+	= K padded with 0's on the left so that the result is b bits in length
ipad	= 00110110 repeated b/8 times
opad	= 01011100 repeated b/8 times

- **Then HMAC can be expressed as**

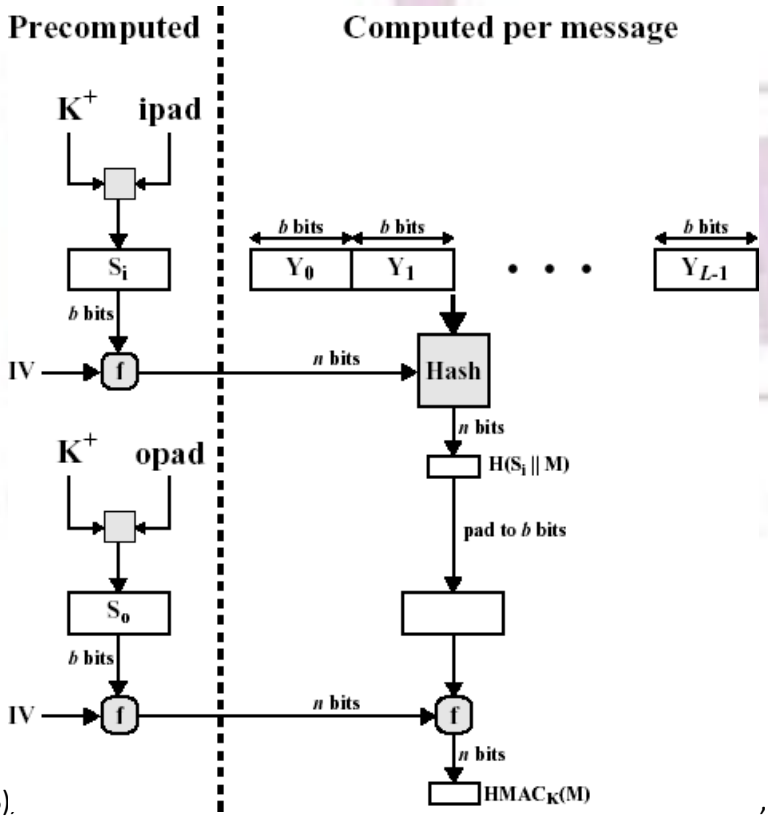
$$HMAC_K = H[\ (K^+ \oplus opad) \ ||\ H[K^+ \oplus ipad) \ ||\ M]\ ]$$



The XOR with ipad results in flipping one-half of the bits of K. Similarly, XOR withopadresultsinflippingone-halfofthebitsofK,butdifferentsetofbits.BypassingS$_i$ andS$_0$throughthecompressionfunctionofthehashalgorithm,wehavepseudorandomlygeneratedtwokeysfromK.

HMACshouldexecuteinapproximatelythesametimeastheembeddedhashfunction for longmessages. HMAC addsthreeexecutions ofthehashcompression function (forS$_0$,S$_i$,andtheblockproduced fromtheinnerhash)

Amoreefficientimplementationispossible.Twoquantitiesareprecomputed.f(IV,(K$_+$f(IV,(K$_+$where f is the compression function for the hash function which takes as arguments achaining variable of n bits and a block of b-bits and produces a chaining variable of nbits.



As shown in the above figure, the values are needed to be computed initially andeverytimeakeychanges.Theprecomputedquantitiessubstitutefortheinitialva lue(IV) in the hash function. With this implementation, only one additional instance of thecompression function is addedtotheprocessingnormally produced by thehashfunction. This implementation is worthwhile if most of the messages for which a MACiscomputedareshort.

TheappealofHMACisthatitsdesignershavebeenabletoproveanexactrelationship between the strength of the embedded hash function and the strength ofHMAC.Thesecurityof aMACfunctionisgenerallyexpressedintermsoftheprobabilityofsuccessful forgerywithagiven amount oftimespent bytheforgerand agivennumberofmessage-MACpairs createdwiththesamekey.Havetwo classes ofattacksontheembeddedhashfunction:

1. Theattackerisabletocomputeanoutputofthecompressionfunction
2. evenwithanIVthatisrandom,secretand unknown totheattacker.
3. TheattackerfindscollisionsinthehashfunctionevenwhentheIVisrandomandsecr et.

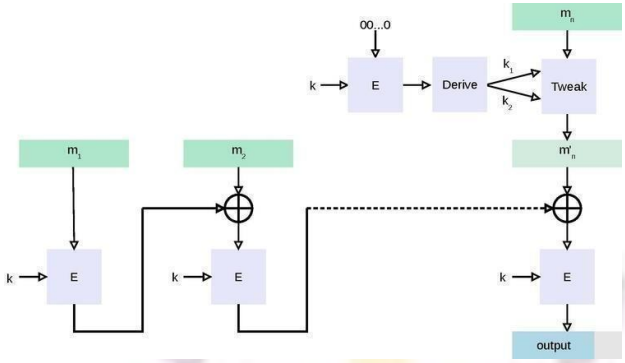
These attacks are likelyto be causedby brute force attack on keyusedwhich has workof order 2_n; or abirthdayattack which requires work of order $2_{(n/2)}$ - but whichrequires the attacker to observe 2_n blocks of messages using the same key- veryunlikely.SoevenMD5isstillsecureforuse inHMACgiventheseconstraints.

CMAC

Incryptography,**CMAC**(Cipher-basedMessageAuthentication Code)[1]isablockcipher-basedmessageauthentication codealgorithm. Itmay beused toprovideassurance of the authenticity and, hence, the integrity of binary data. This mode ofoperation fixes security deficiencies of CBC-MAC (CBC-MAC is secure only for fixed-lengthmessages).

The core of the CMAC algorithm is a variation of CBC-MAC that Black and Rogaway proposed and analyzed under the name XCBC[2]andsubmittedtoNIST.[3]TheXCBCalgorithmefficientlyaddressesthe securitydeficiencies of CBC-MAC, but requires three keys. Iwata and Kurosawa proposed animprovement of XCBC and named the resulting algorithm One-Key CBC-MAC

(OMAC)intheirpapers.[4][5]TheylatersubmittedOMAC1[6],arefinementofO MAC,andadditionalsecurityanalysis.[7]TheOMACalgorithmreducestheamoun tofkeymaterialrequired forXCBC.CMACisequivalenttoOMAC1.



Togenerateanℓ-bitCMACtag(*t*)of amessage(*m*)usinga*b*-bitblockcipher(*E*)anda secret key (*k*), one first generates two *b*-bit sub-keys (*k*1 and *k*2) using the followingalgorithm(thisisequivalenttomultiplicationby*x*andx^2 inafinitefieldGF(2^b)).Letdenotethestandardleft-shiftoperatorand⊕denoteexclusiveor:

1. Calculateatemporaryvalue$k_0=E_k(0)$.
2. If msb(*k*0) = 0, then *k*1 = *k*0 ≪ 1, else *k*1 = (*k*0 ≪ 1) ⊕*C*; where *C* is a certainconstant that depends only on *b*. (Specifically, *C* is the non-leading coefficientsof the lexicographically first irreducible degree-*b* binary polynomial with theminimalnumberof ones.)

3.Ifmsb(*k*1)=0, then*k*2=*k*1≪1,else*k*2=(*k*1 ≪1)⊕*C*.

4.Returnkeys(*k*1,*k*2)fortheMACgenerationprocess.

Asasmallexample,suppose*b*=4,*C*=0011_2,and$k_0=E_k(0)=0101_2$.Then$k_1=1010_2$a nd$k_2=0100⊕0011=0111_2$.

TheCMACtaggenerationprocessisasfollows:

1. Dividemessageinto*b*-bitblocks$m=m_1\|...\|m_{n-1}\|m_n$where$m_1,...,m_{n-1}$are completeblocks.(Theemptymessageistreatedas1incompleteblock.)
2. If$m_n$isacompleteblockthen$m_n'=k_1⊕m_n$else$m_n'=k_2⊕(m_n\|10...0_2)$.3.Let *c*0=00…02.
4. For*i*=1,...,*n-1*,calculate$c_i=E_k(c_{i-1}⊕m_i)$.
5. $c_n=E_k(c_{n-1}⊕m_n')$
6. Output*t*=$msb_ℓ(c_n)$.

Theverificationprocessisasfollows:

1. Usetheabovealgorithmtogeneratethetag.
2. Checkthatthegeneratedtagisequaltothereceivedtag.

DIGITALSIGNATURES

The most important development from the work on public-key cryptographyis the digital signature. Message authentication protects two parties who exchangemessages from any third party. However, it does not protect the two parties againsteach other. A digitalsignature is analogousto thehandwrittensignature,and provides a setof security capabilities that would be difficult to implement in any other way. It musthave the following properties:

• It must verify the author and the date and time of thesignature
It must to authenticate the contents at the time of the signature • It must beverifiable by third parties, to resolve disputes Thus, the digital signature functionincludestheauthenticationfunction.Avarietyofapproacheshasbeenproposedforthe digital signature function. These approaches fall into two categories: direct andarbitrated.

DirectDigitalSignature

DirectDigitalSignaturesinvolvethedirectapplicationofpublic-keyalgorithmsinvolvingonlythecommunicatingparties.Adigitalsignaturemaybeformedbyencrypting the entire message with the sender's private key, or by encrypting a hashcode of the message with the sender's private key. Confidentiality can be provided byfurther encrypting the entire message plus signature using either public or privatekey schemes. It is important to perform the signature function first and then an outerconfidentiality function, since in case of dispute, some third party must view themessage and its signature. But these approaches are dependent on the security of thesender's private-key. Will have problems if it is lost/stolen and signatures forged.Needtime-stampsandtimelykeyrevocation.**ArbitratedDigitalSignature**

The problems associated with direct digital signatures can be addressed by using
anarbiter,inavarietyofpossiblearrangements.Thearbiterplaysasensitiveandcrucialrole in this sort of scheme, and all parties must have a great deal of trust that thearbitration mechanism is working properly. These schemes can be implemented witheither private or public-key algorithms, and the arbiter may or may not see the actualmessagecontents.

UsingConventionalencryption

XA :M||E(Kxa,[IDx||H(M)])

A Y:E(Kay,[IDx||M||E(Kxa,[IDx||H(M))])||T])

It is assumed that the sender X and the arbiter A share a secret key Kxa and that A andYsharesecretkeyKay.XconstructsamessageMandcomputesitshashvalueH(m) ThenXtransmitsthemessageplusasignaturetoA.thesignatureconsistsofanid entifierIDxofXplusthe hashvalue,allencryptedusingKxa.

A decryptsthesignatureandchecksthehashvaluetovalidatethemessage.ThenA transmitsamessagetoY,encryptedwithKay.

ThemessageincludesIDx,theoriginalmessagefromX,thesignature,andatimestam p.

Arbiterseesmessage

Problem:thearbitercouldformanalliancewithsendertodenyasignedmessage,orwith thereceivertoforgethesender'ssignature.

UsingPublicKeyEncryption

X :IDx||E(PRx,[IDx||E(PUy,E(PRx,M))])A

A Y:E(PRa,[IDx ||E(PUy,E(PRx,M))||T])

X double encrypts a message M first with X's private key,PRx, and then with Y'spublic key, PUy. This is a signed, secret version of the message. This signed message,together with X's identifier , is encrypted again with PRx and, together with IDx, issenttoA. The inner,doubleencryptedmessageissecurefromthe arbiter(andeveryoneelseexceptY)

A can decrypt the outer encryption to assure that the message must have comefromX(becauseonlyXhasPRx).ThenAtransmitsamessagetoY,encrypted withPRa.ThemessageincludesIDx,thedoubleencryptedmessage,andatimestam p.Arbiterdoesnotseemessage

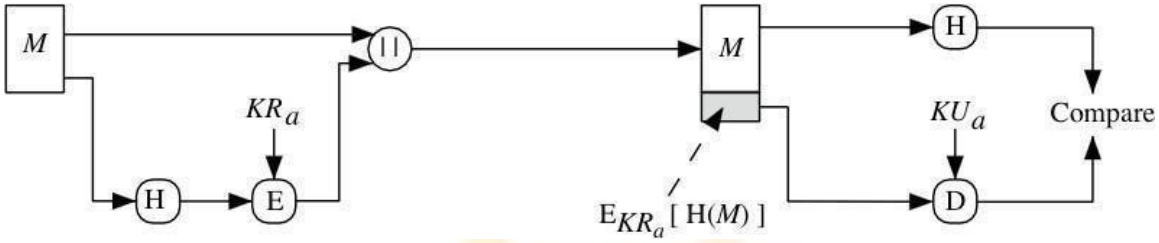
DigitalSignatureStandard(DSS)

TheNationalInstituteofStandardsandTechnology(NIST)haspublishedFederal Information Processing Standard FIPS 186, known as the Digital SignatureStandard (DSS). The DSS makes use of the Secure Hash Algorithm (SHA) and presentsa new digital signature technique, the Digital Signature Algorithm (DSA). The DSSuses an algorithm that is designed to provide only the digital signature function andcannotbeusedforencryptionorkeyexchange,unlikeRSA.

TheRSAapproachisshownbelow.Themessagetobesignedisinputtoahashfunctio n that produces a secure hash code of fixed length. This hash code is

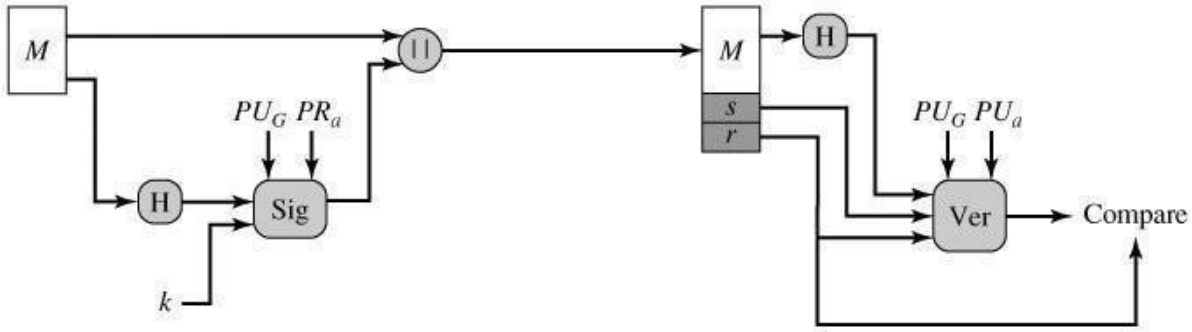
thenencrypted using the sender's private key to form the signature. Both the message andthesignaturearethen transmitted.

Therecipienttakesthemessageandproducesahashcode.Therecipientalso



decryptsthesignatureusingthesender's publickey. If the calculated hash codematches the decrypted signature, the signature is accepted as valid. Because only thesenderknowstheprivatekey,onlythesendercouldhaveproducedavalidsignatu re.

TheDSSapproachalso makesuseofahashfunction.Thehashcodeisprovidedas input to a signature function along with a random number k generated for thisparticular signature. The signature function also depends on the sender's private key(PRa) and a set of parameters known to a group of communicating principals. We canconsider this set to constitute a globalpublic key (PUG).The result is a signatureconsisting oftwocomponents,labeled sandr. At the receiving end, the hash code of the incoming message is generated. Thisplusthesignatureisinputtoaverificationfunction.Theverificationfunctionals o



(b) DSS approach

depends on the global public key as well as the sender's public key (PUa), which ispaired with the sender's private key. The output of the verification function is a valuethat is equal to the signature component r if the signature is valid. The signaturefunction is such that only the sender, with knowledge of the private key, could haveproducedthevalid signature.

ELGAMAL DIGITAL SIGNATURE SCHEME

Before examining the NIST Digital Signature standard, it will be helpful to understand the ElGamal and Schnorr signature schemes. Recall from Chapter 10, that the ElGamal encryption scheme is designed to enable encryption by a user's public key with decryption by the user's private key. The ElGamal signature scheme involves the use of the private key for encryption and the public key for decryption [ELGA84, ELGA85].

Before proceeding, we need a result from number theory. Recall from Chapter 8 that for a prime number q, if a is a primitive root of q, then

$$\alpha, \alpha^2, \ldots, \alpha^{q-1}$$

are distinct (mod q). It can be shown that, if a is a primitive root of q, then

1. For any integer m, $\alpha^m \equiv 1 \pmod{q}$ if and only if $m \equiv 0 \pmod{q-1}$.
2. For any integers, i, j, $\alpha^i \equiv \alpha^j \pmod{q}$ if and only if $i \equiv j \pmod{q-1}$.

As with ElGamal encryption, the global elements of **ElGamal digital signature** are a prime number q and a, which is a primitive root of q. User A generates a private/public key pair as follows.

1. Generate a random integer XA, such that 1 6 XA<q - 1.
2. Compute $YA = aXA \bmod q$.
3. A's private key is XA; A's pubic key is {q, a, YA}.

To sign a message M, user A first computes the hash m = H(M), such that m is an integer in the range 0 <= m <= q - 1. A then forms a digital signature as follows.

1. Choose a random integer K such that 1 <= K <= q - 1 and gcd(K, q - 1) = 1. That is, K is relatively prime to q - 1.
2. Compute $S1 = aK \bmod q$. Note that this is the same as the computation of $C1$

for ElGamal encryption.

3. Compute K-1mod(q-1). That is, compute the inverse of K modulo q-1.
4. Compute $S2 = K$-1(m - $XAS1$) mod (q - 1).
5. The signature consists of the pair ($S1$, $S2$).

Any user B can verify the signature as follows.

1. Compute $V1 = am \bmod q$.
2.

Compute $V2$ = (YA) 1($S1$) mod q.

The signature is valid if $V1 = V2$. Let us demonstrate that this is so. Assume that the equality is true. Then we have

$$\alpha^m \bmod q = (Y_A)^{S_1}(S_1)^{S_2} \bmod q \qquad \text{assume } V_1 = V_2$$
$$\alpha^m \bmod q = \alpha^{X_A S_1}\alpha^{KS_2} \bmod q \qquad \text{substituting for } Y_A \text{ and } S_1$$
$$\alpha^{m-X_A S_1} \bmod q = \alpha^{KS_2} \bmod q \qquad \text{rearranging terms}$$
$$m - X_A S_1 \equiv KS_2 \bmod (q-1) \qquad \text{property of primitive roots}$$
$$m - X_A S_1 \equiv KK^{-1}(m - X_A S_1) \bmod (q-1) \qquad \text{substituting for } S_2$$

For example, let us start with the prime field GF(19); that is, $q = 19$. It has primitive roots {2, 3, 10, 13, 14, 15}, as shown in Table 8.3. We choose $a = 10$.

Alice generates a key pair as follows:

1. Alice chooses $XA = 16$.
2. Then $YA = aXA \bmod q = a16 \bmod 19 = 4$.
3. Alice's private key is 16; Alice's pubic key is $\{q, a, YA\} = \{19, 1$

04. Suppose Alice wants to sign a message with hash value $m = 14$.

1. Alice chooses $K = 5$, which is relatively prime to $q - 1 = 18$.
2. $S1 = aK \bmod q = 105 \bmod 19 = 3$ (see Table 8.3).
3. $K^{-1} \bmod (q-1) = 5^{-1} \bmod 18 = 11$.
4. $S_2 = K^{-1}(m - X_A S_1) \bmod (q-1) = 11(14 - (16)(3)) \bmod 18 = -374 \bmod 18 = 4$.

Bob can verify the signature as follows.

1. $V_1 = \alpha^m \bmod q = 10^{14} \bmod 19 = 16$.
2. $V_2 = (Y_A)^{S_1}(S_1)^{S_2} \bmod q = (4^3)(3^4) \bmod 19 = 5184 \bmod 19 = 16$.

Thus, the signature is valid.

SYMMETRIC KEY DISTRIBUTION USING SYMMETRIC ENCRYPTION

For symmetric encryption to work, the two parties to an exchange must share the same key, and that key must be protected from access by others. Furthermore, frequent key changes are usually desirable to limit the amount of data compromised if an attackerlearns the key. Therefore, the strength of any cryptographic system rests with the *key distribution technique*, a term that refers to th e means of deliver- ing a key to two parties who wish to exchange data without allowing

others to see the key. For two parties A and B, key distribution can be achieved in a number of ways, as follows:

1 A can select a key and physically deliver it to B.
2. A third party can select the key and physically deliver it to A and B.
3. If A and B have previously and recently used a key, one party can transmit the new key to the other, encrypted using the old key.
4. If A and B each has an encrypted connection to a third party C, C can deliver a key on the encrypted links to A and B.

Options 1 and 2 call for manual delivery of a key. For link encryption, this is a reasonable requirement, because each link encryption device is going to be exchanging data only with its partner on the other end of the link. However, for **end-to-end encryption** over a network, manual delivery is awkward. In a distributed system, any given host or terminal may need to engage in exchanges with many other hosts and terminals over time. Thus, each device needs a number of keys supplied dynamically. The problem is especially difficult in a wide-area distributed system.

The scale of the problem depends on the number of communicating pairs that must be supported. If end-to-end encryption is done at a network or IP level, then a key is needed for each pair of hosts on the network that wish to communicate. Thus, if there are N hosts, the number of required keys is $[N(N - 1)]/2$. If encryption is done at the application level, then a key is needed for every pair of users or processes that require communication. Thus, a network may have hundreds of hosts but thousands of users and processes. Figure 14.1 illustrates the magnitude of thekey distribution task for end-to-end encryption.1 A network using node-levelencryption with 1000 nodes would conceivably need to distribute as many as half ami llion keys. If that same network supported 10,000 applications, then as many as 50 million keys may be required for application-level encryption.Returning to our list, option 3 is a possibility for either link encryption or end-toend encryption, but if an attacker ever succeeds in gaining access to one key, then all subsequent keys will be revealed. Furthermore, the initial distribution of potentially millions of keys still must be made.

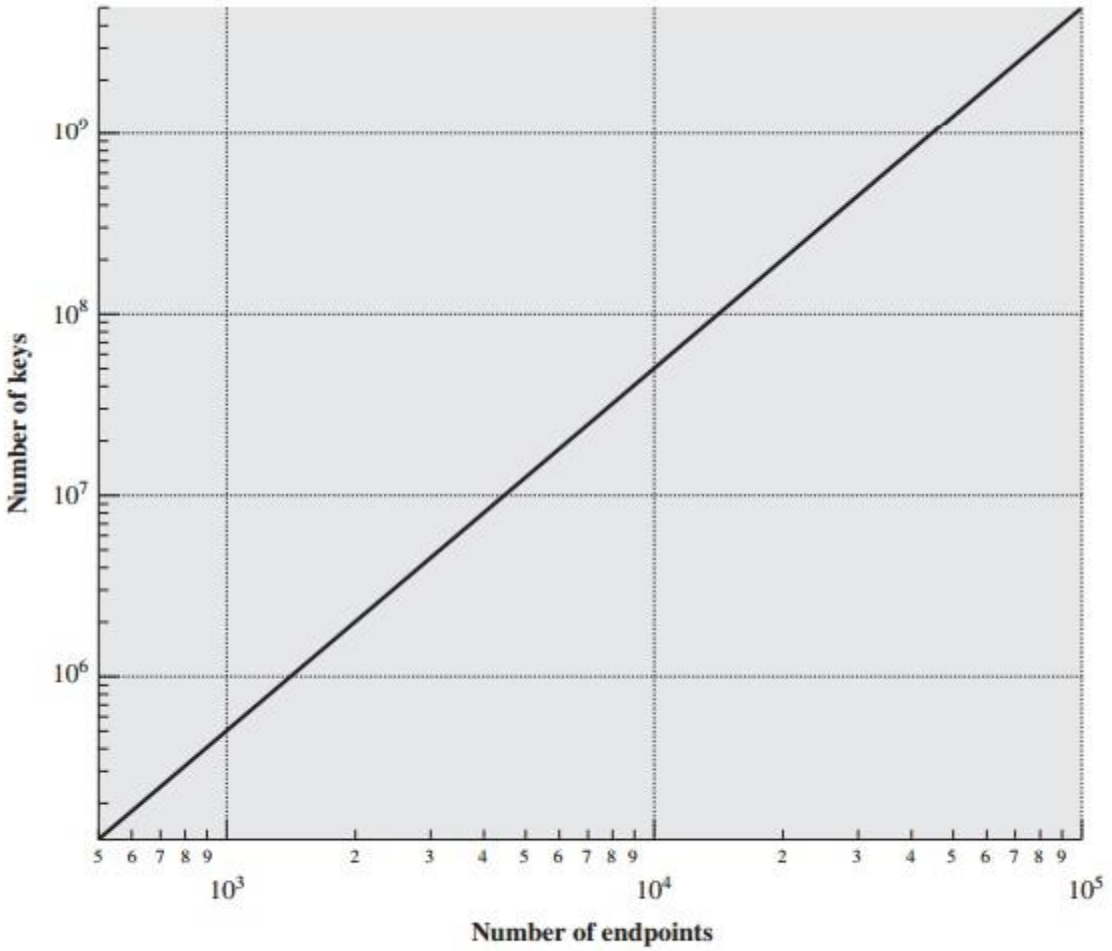


Figure 14.1 Number of Keys Required to Support Arbitrary Connections between Endpoints

For end-to-endencryption, some variation on option 4 has been widely adopted. In this scheme, a key distribution center is responsible for distributing keys to pairs of users (hosts, processes, applications) as needed. Each user must share a unique key with the key distribution center for purposes of key distribution.

The use of a key distribution center is based on the use of a hierarchy of keys. At a minimu m, two levels of keys are used (Figure 14.2). Communication between end systems is encrypted using a temporary key, often referred to as a **session key**. Typical ly, the session key is used for the duration of a logical connection, such as a frame relay connection or transport connection, and then discarded. Each session key is obtained from the key distribution center over the same networking facilities used f or end-user communication. Accordingly, session keys are transmitted in encrypted form, using a **master key** that is shared by the key distri

bution center and an end system or user.

For each end system or user, there is a unique master key that it shares with the key distribution center. Of course, these master keys must be distributed in some fashion. However, the scale of the problem is vastly reduced. If there are N entities that wish to communicate in pairs, then, as was mentioned, as many as $[N(N - 1)]/2$ session keys are needed at any one time. However, only N master keys are required, one for each entity. Thus, master keys can be distributed in some noncryptographic way, such as physical delivery.

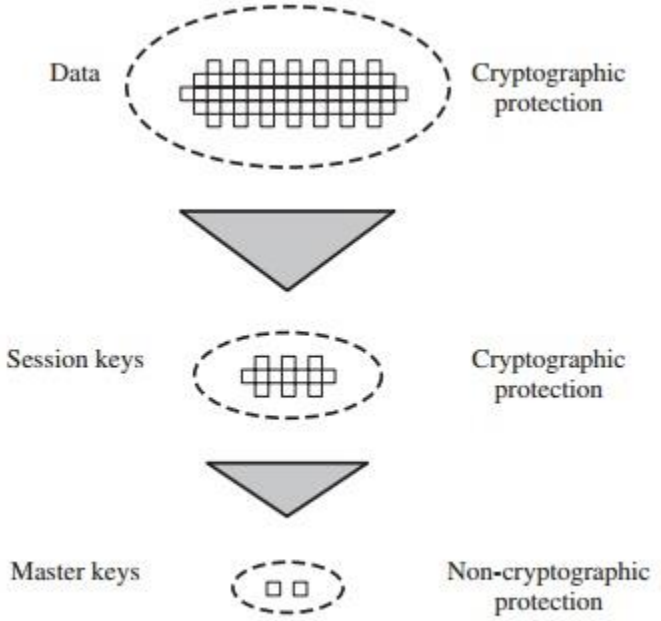


Figure 14.2 The Use of a Key Hierarchy

A Key Distribution Scenario

The key distribution concept can be deployed in a number of ways. A typical scenario is illustrated in Figure 14.3, which is based on a figure in [POPE79]. The scenario assumes that each user shares a unique master key with the key distribution center (KDC).

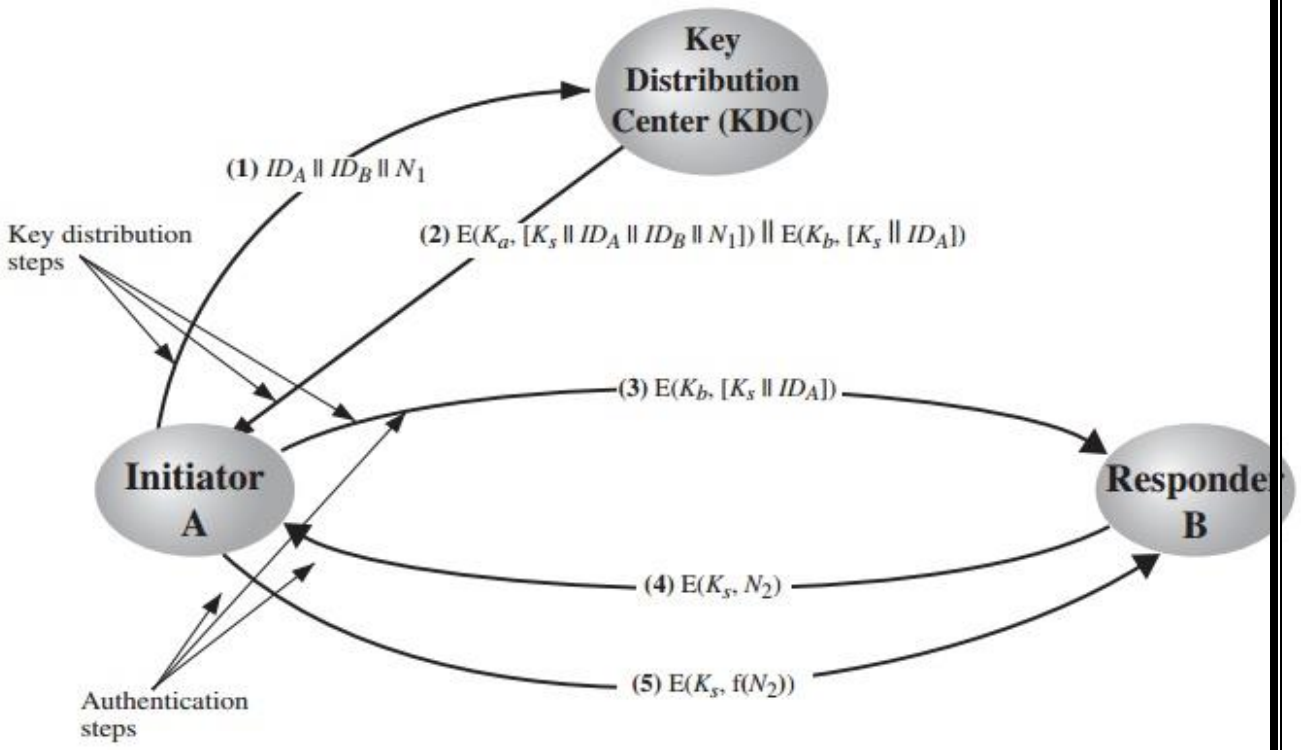


Figure 14.3 Key Distribution Scenario

Let us assume that user A wishes to establish a logical connection with B andrequires a on e-time session key to protect the data transmitted over the connection. A has a master key, *Ka*, known only to itself and the KDC; similarly, B shares the master key *Kb* with the KDC. The following steps occur.

1. A issues a request to the KDC for a session key to protect a logical connection to B. The message includes the identity of A and B and a unique identifier, *N*1, for this transaction, which we refer to as a **nonce**. The nonce may be a time-stamp, a counter, or a random number; the minimum requirement is that it differs with each request. Also, to prevent masquerade, it should be difficult for an opponent to guess the nonce. Thus, a random number is a good choice for a nonce.
2. The KDC responds with a message encrypted using *Ka*. Thus, A is the only one who can successfully read the message, and A knows that it originated at the KDC. The message includes two items intended for A:
   - The one-time session key, *Ks*, to be used for the session
   - The original request message, including the nonce, to enable A to match this response with the appropriate request

Thus, A can verify that its original request was not altered before reception by the KDC and, because of the nonce, that this is not a replay of some previous request.

In addition, the message includes two items intended for B:

- The one-time session key, *Ks*, to be used for the session
- An identifier of A (e.g., its network address), *IDA*

These last two items are encrypted with *Kb* (the master key that the KDC shares with B).They are to be sent to B to establish the connection and prove A's identity.

A stores the session key for use in the upcoming session and forwards to B the information that originated at the KDC for B, namely, $E(Kb,[Ks \,||\, IDA])$. Because this information is encrypted with *Kb*, it is protected from eavesdropping. B now knows the session key (*Ks*), knows that the other party is A (from *IDA*), and knows that the information originated at the KDC (because it is encrypted using *Kb*).

At this point, a session key has been securely delivered to A and B, and they may begin their protected exchange. However, two additional steps are desirable:

4. Using the newly minted session key for encryption, B sends a nonce, *N*2, to A.
5. Also, using *Ks*, A responds with f(*N*2), where f is a function that performs some transformation on *N*2 (e.g., adding one).

These steps assure B that the original message it received (step 3) was not a replay.

Note that the actual key distribution involves only steps 1 through 3, but that steps 4 and 5, as well as step 3, perform an authentication function.

Hierarchical Key Control

It is not necessary to limit the key distribution function to a single KDC. Indeed, for very large networks, it may not be practical to do so. As an alternative, a hierarchy of KDCs can be established. For example, there can be local KDCs, each responsible for a small domain of the overall internetwork, such as a single LAN or a single building. For communication among entities within the same local domain, the local KDC is responsible for key distribution. If two entities in different domains desire a shared key, then the corresponding local KDCs can communicate through a globalKDC. In this case, any one of the three KDCs involved can actually select the key. The hierarchical concept can be extended to three or even more layers, depending on the size of the user population and the geographic scope of the internetwork.

A hierarchical scheme minimizes the effort involved in master key distribution, because most master keys are those shared by a local KDC with its local entities. Furthermore, such a scheme limits the damage of a faulty or subverted KDC to its local area only.

Session Key Lifetime

The more frequently session keys are exchanged, the more secure they are, because the opponent has less ciphertext to work with for any given session key. On the other hand, the distribution of session keys delays the start of any exchange and places a burden on network capacity. A security manager must try to balance these competing considerations in determining the lifetime of a particular session key.

For connection-oriented protocols, one obvious choice is to use the same session key for the length of time that the connection is open, using a new session key for each new session. If a logical connection has a very long lifetime, then it would be prudent to change the session key periodically, perhaps every time the PDU (protocol data unit) sequence number cycles.

For a connectionless protocol, such as a transaction-oriented protocol, there is no explicit connection initiation or termination. Thus, it is not obvious how often one needs to change the session key. The most secure approach is to use a new session key for each exchange. However, this negates one of the principal benefits of connectionless protocols, which is minimum overhead and delay for each transaction. A better strategy is to use a given session key for a certain fixed period only or for a certain number of transactions.

A Transparent Key Control Scheme

The approachsuggested in Figure 14.3 has many variations, one of which is described in this subsection. The scheme (Figure 14.4) is useful for providing end-to-end encryption at a network or transport level in a way that is transparent to the end users. The approach assumes that communication makes use of a connection-oriented end-to-end protocol, such as TCP. The noteworthy element of this approach is a session security module (SSM), which may consist of functionality at one protocol layer, that performs end-to-end encryption and obtains session keys on behalf of its host or terminal.

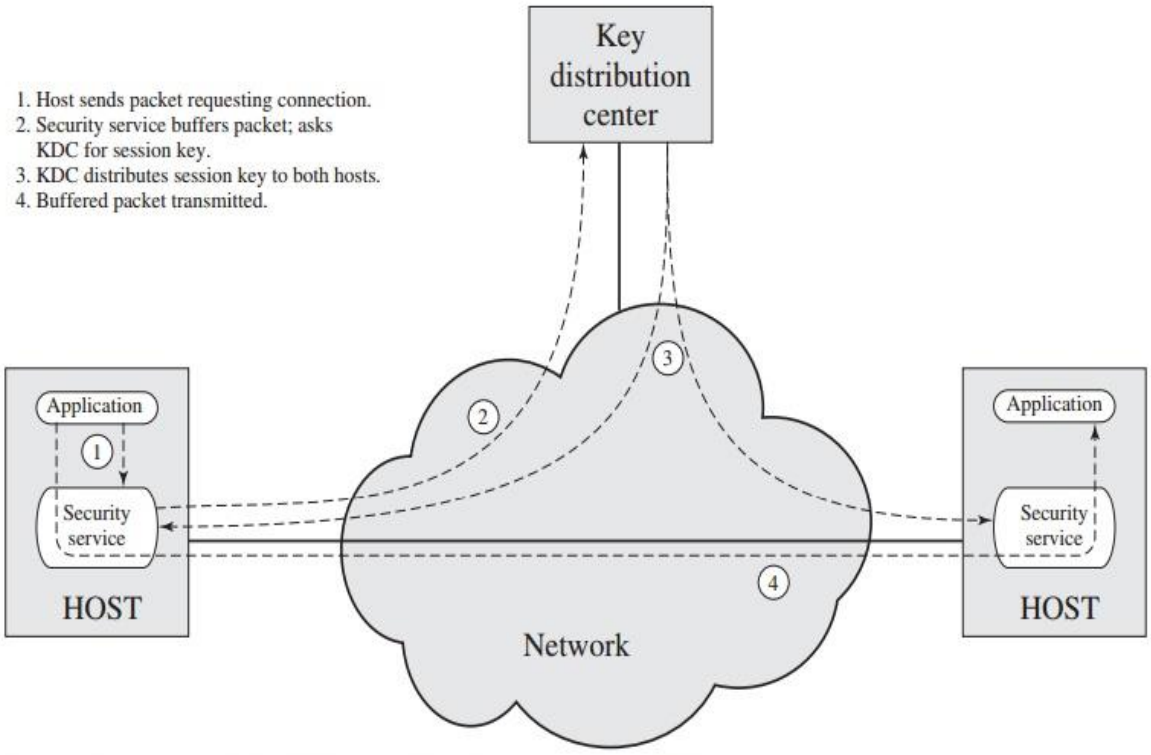


Figure 14.4 Automatic Key Distribution for Connection-Oriented Protocol

The stepsinvolved in establishing a connection are shown in Figure 14.4. When one hos t wishes to set up a connection to another host, it transmits a connec-tion request packet (step 1). The SSM saves that packet and applies to the KDC for permission to establish the connection (step 2). The communication between the SSM and the KDC is encrypted using a master key shared only by this SSM and the KDC. If the KDC approves the connection request, it generates the session key and delivers it to the two appropriate SSMs, using a unique permanent key for each SSM (step 3). The requesting SSM can now release the connection request packet, and a connection is set up between the two end systems (step 4). All user data exchanged between the two end systems are encrypted by their respective SSMs using the one-time session key.

The automated key distribution approach provides the flexibility and dynamic characteristics needed to allow a number of terminal users to access a number of hosts and for the hosts to exchange data with each other.

Decentralized Key Control

The use of a key distribution center imposes the requirement that the KDC be trusted and be protected from subversion. This requirement can be avoided if key distribution is fully decentralized. Although full decentralization is not practical for larger

networks using symmetric encryption only, it may be useful within a local context.

A decentralized approach requires that each end system be able to communi- cate in a secure manner with all potential partner end systems for purposes of ses- sion key distribution. Thus, there may need to be as many as $[n(n - 1)]/2$ master keys for a configuration with n end systems.

A session key may be established with the following sequence of steps (Figure 14.5).

1. A issues a request to B for a session key and includes a nonce, $N1$.
2. B responds with a message that is encrypted using the shared master key. The response includes the session key selected by B, an identifier of B, the value f($N1$), and a nother nonce, $N2$.
3. Using the new session key, A returns f($N2$) to B.

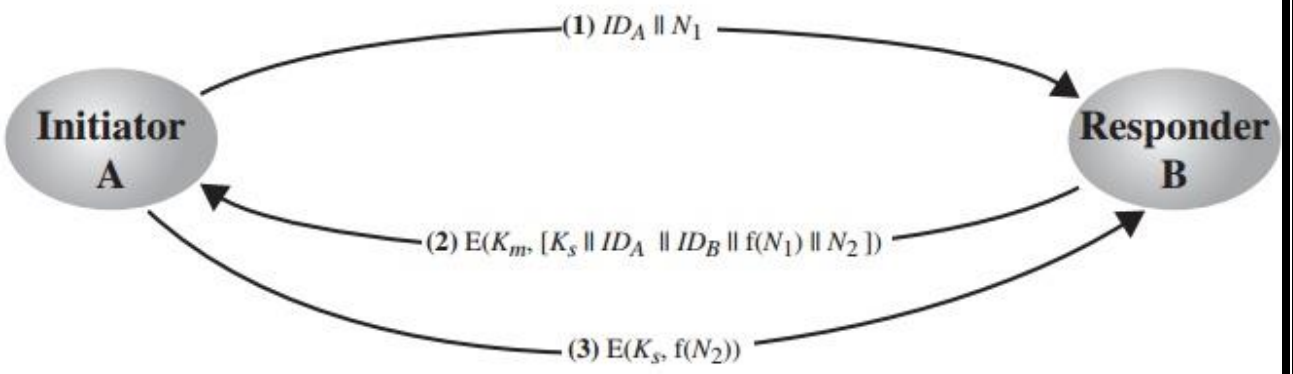


Figure 14.5 Decentralized Key Distribution

Thus, although each node must maintain at most (n - 1) master keys, as many session keys as required may be generated and used. Because the messages trans- ferred using the master key are short, cryptanalysis is difficult. As before, session keys are used for only a limited time to protect them.

Controlling Key Usage

The concept ofa key hierarchy and the use of automated key distribution techniques greatly reduce the number of keys that must be manually managed and distributed. It also may be desirable to impose some control on the way in which automatically distributed keys are used. For example, in addition to separating mas-
ter keys from session keys, we may wish to define different types of session ke ys on the basis of use, such as

- Data-encrypting key, for general communication across a network
- PIN-encrypting key, for personal identification numbers (PINs) used in elec- tronic funds transfer and point-of-sale applications
- File-encrypting key, for encrypting files stored in publicly accessible locations

To illustrate the value of separating keys by type, consider the risk that a master key is imported as a data-encrypting key into a device. Normally, the mas- ter key is physically secured within the cryptographic hardware of the key distrib- ution center and of the end systems. Session keys encrypted with this master key are available to application programs, as are the data encrypted with such session keys. However, if a master key is treated as a session key, it may be possible f or an unauthorized application to obtain plaintext of session keys encrypted with that master key.

Thus, it may be desirable to institute controls in systems that limit the ways in which keys are used, based on characteristics associated with those keys. One s imple plan is to associate a tag with each key ([JONE82]; see also [DAVI89]). The pro- posed technique is for use with DES and makes use of the extra 8 bits in each 64-bit DES key. That is, the eight non-
key bits ordinarily reserved for parity checking form
the key tag. The bits have the following interpretation:

- One bit indicates whether the key is a session key or a master key.
- One bit indicates whether the key can be used for encryption.
- One bit indicates whether the key can be used for decryption.
- The remaining bits are spares for future use.

Because the tag is embedded in the key, it is encrypted along with the key whe n that`key is distributed, thus providing protection. The drawbacks of this sche me are

1. The tag length is limited to 8 bits, limiting its flexibility and functionality.
2. Because the tag is not transmitted in clear form, it can be used only at the point of decryption, limiting the ways in which key use can be co ntrolled.

A moreflexible scheme, referred to
as the control vector, is described in [MATY91a and b]. In this scheme, each session key has an associated control vector

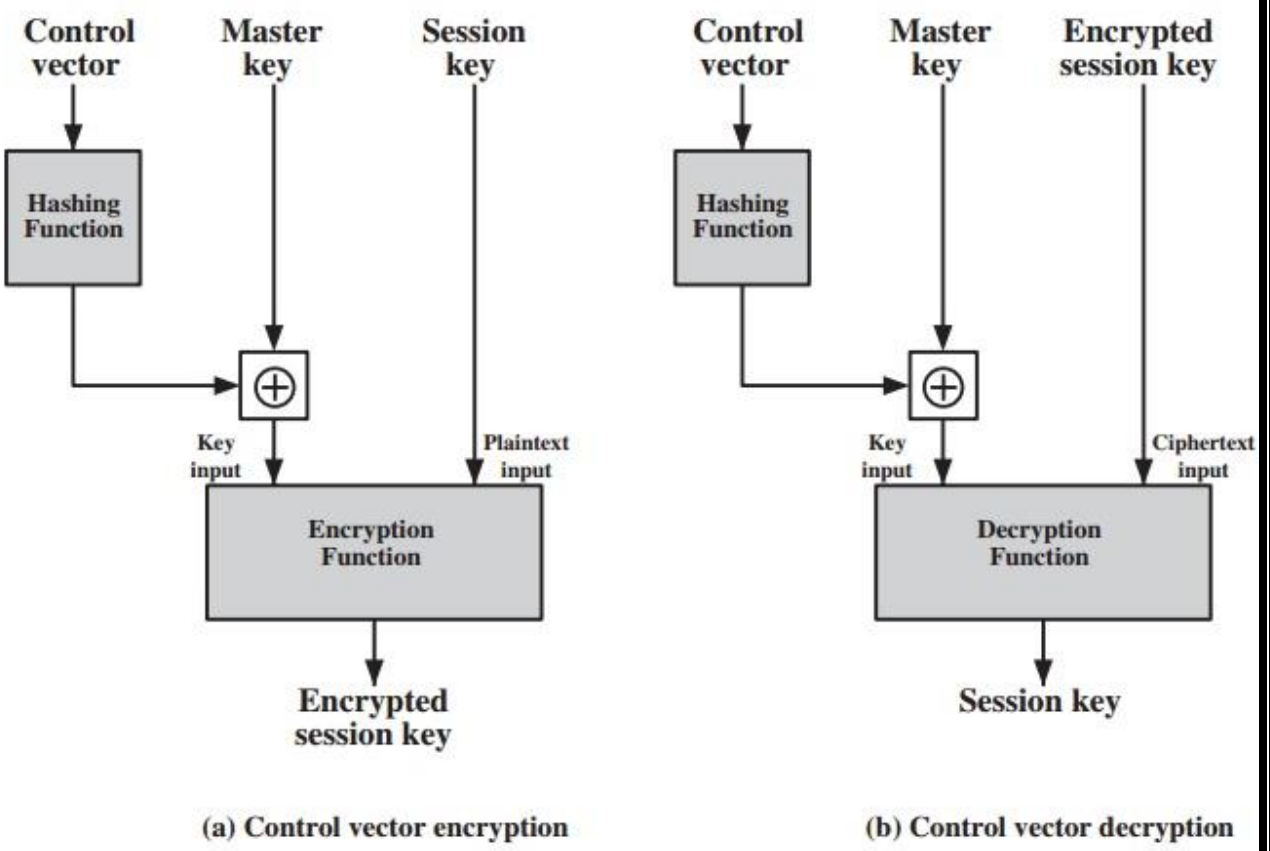


Figure 14.6 Control Vector Encryption and Decryption

consisting of a number of fields that specify the uses and restrictions for that s ession key. The length of the control vector may vary.

The control vector is cryptographically coupled with the key at the time of key generation at the KDC. The coupling and decoupling processes are illustrated in Figure 14.6. As a first step, the control vector is passed through a hash function that produces a value whose length is equal to the encryption key length. Hash functions are discussed in detail in Chapter 11. In essence, a hash function maps values from a larger range into a smaller range with a reasonably uniform spread. Thus, for example, if numbers in the range 1 to 100 are hashed into n umbers in the range 1 to 10, approximately 10% of the source values should map into each of the target values.

The hash value is then XORed with the master key to produce an output that is used as the key input for encrypting the session key. Thus,

Hash value = H = h(CV)

Key input = $K_m \oplus H$

Ciphertext = $E([K_m \oplus H], K_s)$

where K_m is the master key and K_s is the session key. The session key is recove red in plaintext by the reverse operation:

$D([K_m \oplus H], E([K_m \oplus H], K_s))$

When a session key is delivered to a user from the KDC, it is accompanied by the control vector in clear form. The session key can be recovered only by using both the master key that the user shares with the KDC and the control vector. Thus, the linkage between the session key and its control vector is maintained.

Use of the control vector has two advantages over use of an 8-bit tag. First, there is no restriction on length of the control vector, which enables arbitrarily com- plex controls to be imposed on key use. Second, the control vector is available in clear form at all stages of operation. Thus, control of key use can be exercised in multiple locations.

SYMMETRIC KEY DISTRIBUTION USING ASYMMETRIC ENCRYPTION

Because of the inefficiency of public key cryptosystems, they are almost never used for the direct encryption of sizable block of data, but are limited to relatively small blocks. One of the most important uses of a public-key cryptosystem is to encrypt secret keys for distribution. We see many specific examples of this in Part Five. Here, we discuss general principles and typical approaches.

Simple Secret Key Distribution

An extremely simple scheme was put forward by Merkle [MERK79], as illust rated in Figure 14.7. If A wishes to communicate with B, the following procedure is employed:

1. A generates a public/private key pair $\{PU_a, PR_a\}$ and transmits a mes sage to B consisting of PU_a and an identifier of A, ID_A.
2. B generates a secret key, K_s, and transmits it to A, which is encrypted with A's public key.

3. A computes D(PR_a, E(PU_a, K_s)) to recover the secret key. Because only A can decrypt the message, only A and B will know the identity of K_s.
4. A discards PU_a and PR_a and B discards PU_a.

A and B can now securely communicate using conventional encryption and the session key K_s. At the completion of the exchange, both A and B discard K_s.

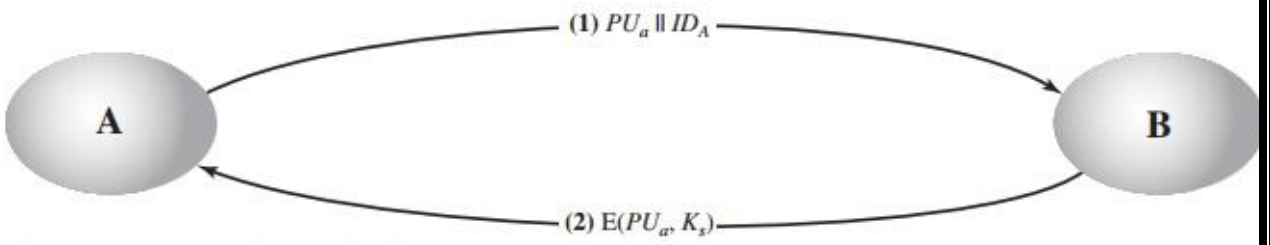


Figure 14.7 Simple Use of Public-Key Encryption to Establish a Session Key

Despite its simplicity, this is an attractive protocol. No keys exist before the start of the communication and none exist after the completion of communication. Thus, the risk of compromise of the keys is minimal. At the same time, the communication is secure from eavesdropping.

The protocol depicted in Figure 14.7 is insecure against an adversary who can intercept messages and then either relay the intercepted message or substitute another message (see Figure 1.3c). Such an attack is known as a **man-in-the-middle attack** [RIVE84]. In this case, if an adversary, E, has control of the intervening communication channel, then E can compromise the communication in the following fashion without being detected.

1. A generates a public/private key pair {PU_a, PR_a} and transmits a message intended for B consisting of PU_a and an identifier of A, ID_A.
2. E intercepts the message, creates its own public/private key pair {PU_e, PR_e} and transmits $PU_e \| ID_A$ to B.
3. B generates a secret key, K_s, and transmits E(PU_e, K_s).
4. E intercepts the message and learns K_s by computing D(PR_e, E(PU_e, K_s)).
5. E transmits E(PU_a, K_s) to A.

The result is that both A and B know K_s and are unaware that K_s has also been revealed to E. A and B can now exchange messages using K_s. E no longer activelyinterferes with the communications channel but simply eavesdrops. Knowing K_s, E can decrypt all messages, and both A and B are unaware of the problem. Thus, this simple protocol is only useful in an en vironment where the only threat is eavesdropping.

Secret Key Distribution with Confidentiality and Authentication

Figure 14.8, based on an approach suggested in [NEED78], provides protection against both active and passive attacks. We begin at a poi nt when it is assumed that
A and B have exchanged public keys by one of the schemes described subseq uently in this chapter. Then the following steps occur.

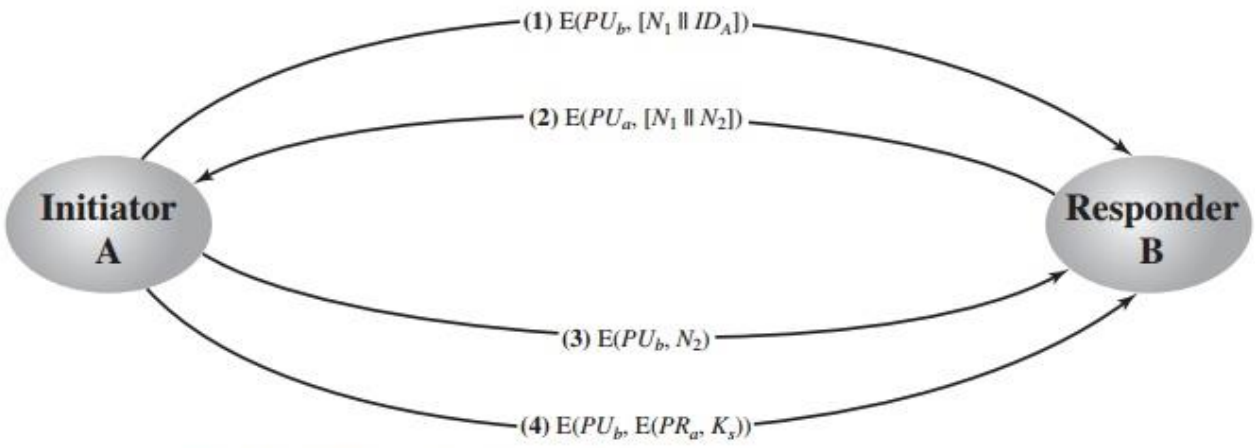


Figure 14.8 Public-Key Distribution of Secret Keys

1. A uses B's public key to encrypt a message to B containing an identifier of A(ID_A) and a nonce (N_1), which is used to identify this transaction uniquely.

B sends a message to A encrypted with PU_a and containing A's nonce (N_1) as ell as a new nonce generated by B (N_2). Because only B could have (N_2). Because only B could have decrypted message (1), the presence of N_1 in message (2) assures A that the correspondent is B.

2. A returns N_2, encrypted using B's public key, to assure B that its correspondent is A.

A selects a secret key K_s and sends $M = E(PU_b, E(PR_a, K_s))$ to B. Encryption of this message with B's public key ensures that only B can read it; encryption with A's private key ensures that only A could have sent it.

3. B computes $D(PU_a, D(PR_b, M))$ to recover the secret key.
4. The result is that this scheme ensures both confidentiality and authentication in the exchange of a secret key.

A HYBRID SCHEME

Yet another way to use public-key encryption to distribute secret keys is a hybrid approach in use on IBM mainframes [LE93]. This scheme retains the use of a key distribution center (KDC) that shares a secret master key with each user and distributes secret session keys encrypted with the master key. A public key scheme is used to distribute the master keys. The following rationale is provided for using this three-level approach:

Performance: There are many applications, especially transaction-oriented applications, in which the session keys change frequently. Distribution of session keys by public-key encryption could degrade overall system performance because of the relatively high computational load of public-key encryption and decryption. With a three-level hierarchy, public-key encryption is used only occasionally to update the master key between a user and the KDC.

Backward compatibility: The hybrid scheme is easily overlaid on an existing KDC scheme with minimal disruption or software changes.

The addition of a public-key layer provides a secure, efficient means of distributing master keys. This is an advantage in a configuration in which a single KDC serves a widely distributed set of users.

DISTRIBUTION OF PUBLIC KEYS

Several techniques have been proposed for the distribution of public keys. Virtuallyall these proposals can be grouped into the following general schemes:

- Public announcement
- Publicly available directory

- Public-key authority
- Public-key certificates

Figure 14.9 Uncontrolled Public-Key Distribution

Public Announcement of Public Keys

On the face of it, the point of publickey encryption is that the public key is public. Thus, if there is some broadly accepted publickey algorithm, such as RSA, any participant can send his or her public key to any other participant or broadcast the keyto the community at large (Figure 14.9). For example, because of the growing popularity of PGP (pretty good privac y, discussed in Chapter 18), which makes use of RSA, many PGP users have adopted the practice of appending their public key to messages that they send to public forums, such as USENET newsgroups and Internet mailing lists.

Although this approach is convenient, it has a major weakness. Anyone can forge such a public announcement. That is, some user could pretend to be user A and send a public key to another participant or broadcast such a public key. Until such time as user A discovers the forgery and alerts other participants, the forger is able to read all encrypted messages intended for A and can use the forged keys for authentication (see Figure 9.3).

Publicly Available Directory

A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public keys. Maintenance and distribution of the public direc- tory would have to be the responsibility of some trusted entity or organization (Figure 14.10). Such a scheme would include the following elements :

1. The authority maintains a directory with a {name, public key} entry for each participant.
2. Each participant registers a public key with the directory authority. Registration would have to be in person or by some form of secure authenticated communication.
3. A participant may replace the existing key with a new one at any time, either because of the desire to replace a public key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.

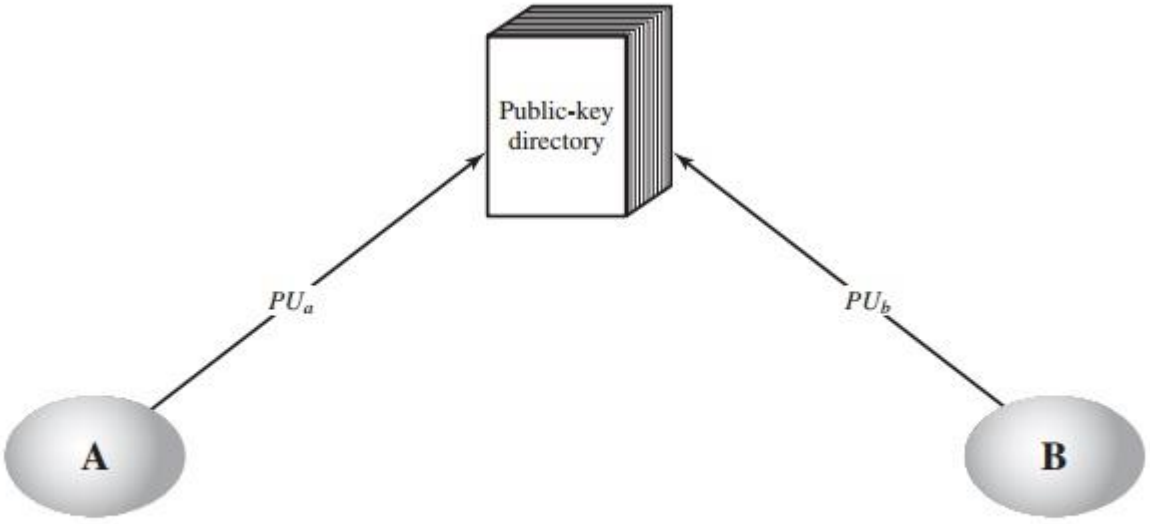


Figure 14.10 Public-Key Publication

4. Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory.

This scheme is clearly more secure than individual public announcements but still has vulnerabilities. If an adversary succeeds in obtaining or computing the private key of the directory authority, the adversary could authoritatively pass out counterfeit public keys and subsequently impersonate any participant and eaves- drop on messages sent to any participant. Another way to achieve the same end is for the adversary to tamper with the records kept by the authority.

Public-Key Authority

Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public keys from the directory. A typical scenario is

illustrated in Figure 14.11, which is based on a figure in [POPE79]. As before, the scenario assumes that a central authority maintains a dynamic directory of public keys of all participants. In addition, each participant reliably knows a public key for the authority, with only the authority knowing the corresponding private key. The following steps (matched by number to Figure 14.11) occur.

1. A sends a timestamped message to the public-key authority containing a request for the current public key of B.
2. The authority responds with a message that is encrypted using the authority's private key, *PR*auth.Thus,A is able to decrypt the message using the authority's public ke y.Therefore,A is assured that the message originated with the authority.The message includes the following:
   - B's public key, *PUb*, which A can use to encrypt messages destined for B
   - The original request used to enable A to match this response with the corresponding earlier request and to verify that the original request was not altered before reception by the authority

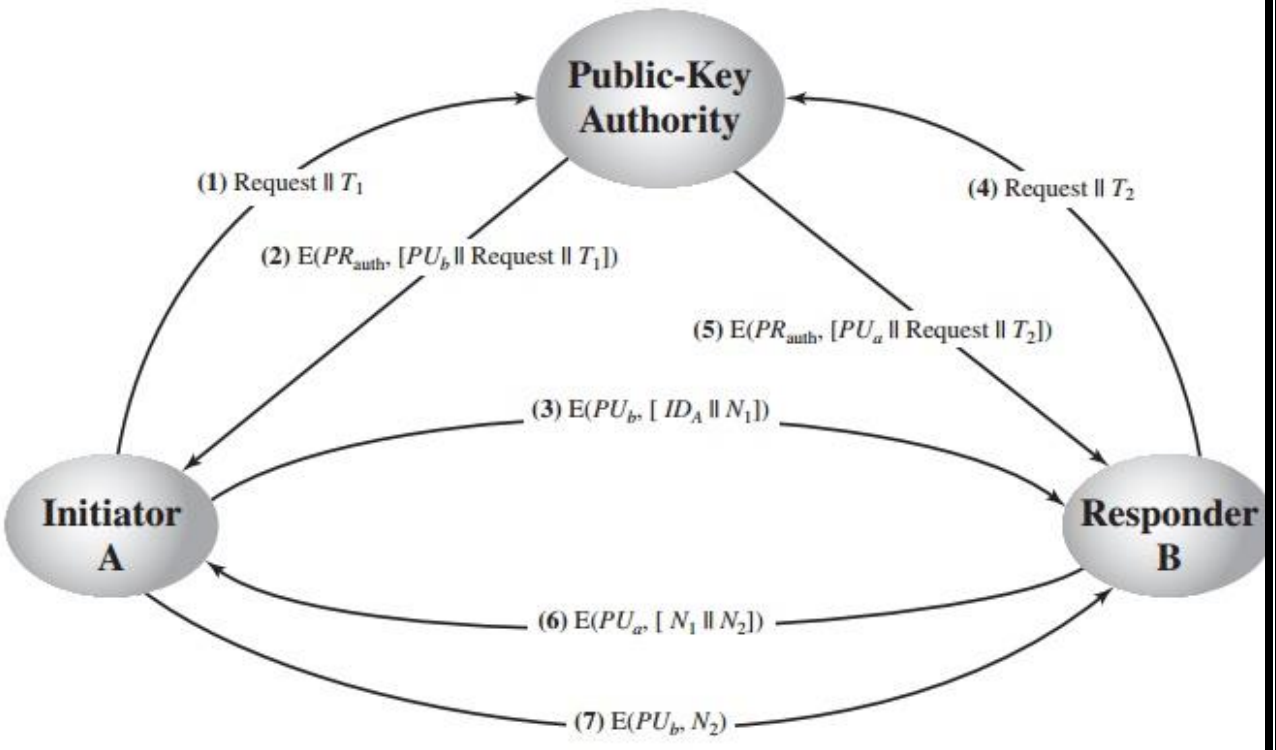


Figure 14.11 Public-Key Distribution Scenario

- The original timestamp given so A can determine that this is not an old message from the authority containing a key other than B's current public key

3. A stores B's public key and also uses it to encrypt a message to B containing an identifier of A (*IDA*) and a nonce (*N*1), which is used to identify this transaction uniq uely.

4, 5. B retrieves A's public key from the authority in the same manner as A retrieved B's public key.

At this point, public keys have been securely delivered to A and B, and they may begin their protected exchange. However, two additional steps are desirable:

6.B sends a message to A encrypted with PUa and containing A's nonce (N_1) as well as a new nonce generated by B (N_2). Because only B could have decrypted message (3), the presence of N_1 in message (6) assures A that the correspondent is B.

6.A returns *N*2, which is encrypted using B's public key, to assure B that its correspondent is A.

Thus, a total of seven messages are required. However, the initial four mes- sages need be used only infrequently because both A and B can save the other's public key for future use—a technique known as caching. Periodically, a user should request fresh copies of the public keys of its correspondents to ensure currency.

Public-Key Certificates

The scenario of Figure 14.11 is attractive, yet it has some drawbacks. The public-key authority could be somewhat of a bottleneck in the system, for a user must appeal to the authority for a public key for every other user that it wishes to contact. As before, the directory of names and public keys maintained by the authority is vul- nerable to tampering.

An alternative approach, first suggested by Kohnfelder [KOHN78], is to use **certificates** that can be used by participants to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority. In essence, a certificate consists of a public key, an identifier of the key owner, and the whole block signed by a trusted third party. Typically, the third party is a certificate authority, such as a government agency or a financial institution, that is trusted by the user community. A user can present his or her public key to the authority in a secure manner and obtain a cer- tificate. The user can then publish the certificate. Anyone needing this user's pub- lic key can obtain the certificate and verify that it is valid by way of the attached trusted signature. A participant can also convey its key information to another by transmitting its certificate. Other participants can verify that the certificate was created by the authority. We can place the following requirements on this scheme:

1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
3. Only the certificate authority can create and update certificates.

These requirements are satisfied by the original proposal in [KOHN78]. Denning [DENN83] added the following additional requirement:

4. Any participant can verify the currency of the certificate.

A certificate scheme is illustrated in Figure 14.12. Each participant applies to the certifica te authority, supplying a public

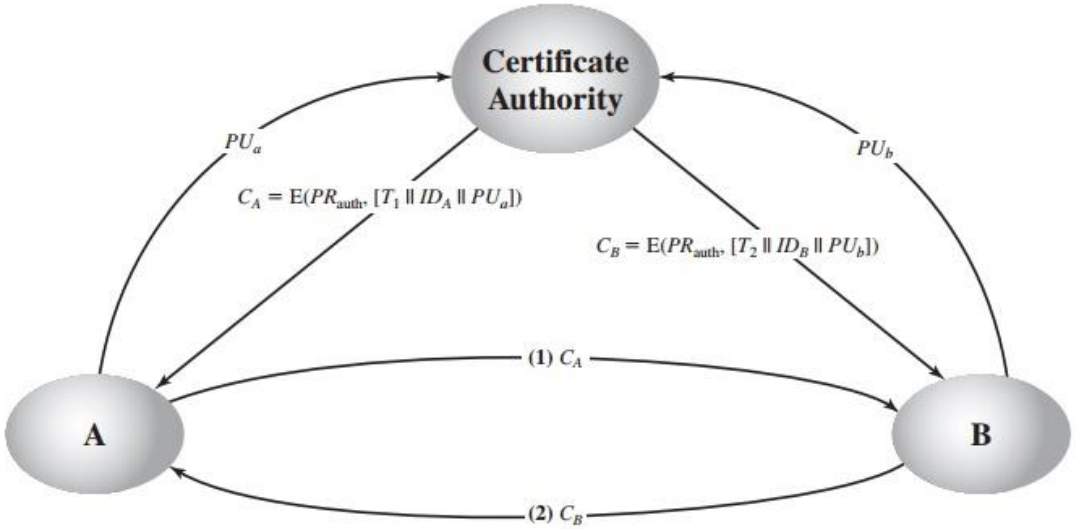


Figure 14.12 Exchange of Public-Key Certificates

Application must be in person or by some form of secure authenticated communi- cation. For participant A, the authority provides a certificate of the form

$CA = E(PRauth, [T \| IDA \| PUa])$

where *PR*auth is the private key used by the authority and *T* is a timestamp. A may then pass this certificate on to any other participant, who reads and verifies the cer- tificate as follows:

$D(PUauth, CA) = D(PUauth, E(PRauth, [T \| IDA \| PUa])) = (T \| IDA \| PUa)$

The recipient uses the authority's public key, *PU*auth, to decrypt the certifi- cate. Because the certificate is readable only using the authority's public key, this verifies that the certificate came from the certificate authority. The elements *IDA* and *PUa* provide the recipient with the name and public key of the certificate's holder. The

timestamp *T* validates the currency of the certificate. The timestamp counters the following scenario. A's private key is learned by an adversary. A gen- erates a new private/public key pair and applies to the certificate authority for a new certificate. Meanwhile, the adversary replays the old certificate to B. If B then encrypts messages using the compromised old public key, the adversary can read those messages.

In this context, the compromise of a private key is comparable to the loss of a credit card. The owner cancels the credit card number but is at risk until all possible communicants are aware that the old credit card is obsolete. Thus, the timestamp serves as something like an expiration date. If a certificate is sufficiently old, it is assumed to be expired.

One scheme has become universally accepted for formatting public-key cer- tificates: the X.509 standard. X.509 certificates are used in most network security applications, including IP security, transport layer security (TLS), and S/MIME.

KERBEROS

Kerberos is an authentication service developed as part of Project Athena at MIT.It addresses the threats posed in an open distributed environment in which users atworkstations wish to access services on servers distributed throughout the network.Someofthesethreatsare:

- Ausermaygainaccesstoaparticularworkstationandpretendtobeano theruseroperatingfrom thatworkstation.
- Ausermayalterthenetworkaddressofaworkstationsothatthereque stssentfromthealtered workstation appeartocomefrom theimpersonatedworkstation.
- Ausermayeavesdroponexchangesandusereplayattacktogainen trancetoaserverortodisrupt operations.

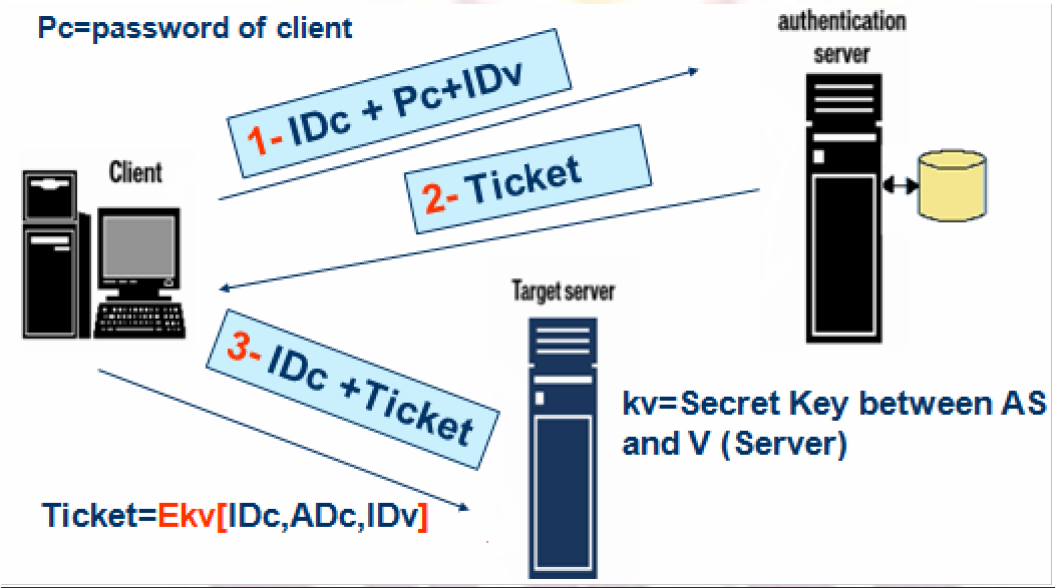
Two versions of Kerberos are in current use: Version-4 and Version-5.
The firstpublishedreportonKerberoslistedthe followingrequirements:
Secure: Anetworkeavesdroppershouldnotbeabletoobtainthenecessaryinformati on to impersonate a user. More generally, Kerberos should be strong enoughthatapotentialopponentdoesnotfindittobetheweak link.
Reliable: For all services that rely on Kerberos for access control, lack of availabilityof the Kerberos service means lack of availability of the supported services.
Hence,Kerberosshouldbehighlyreliableandshouldemployadistributedserverarc hitecture,withonesystem abletobackupanother.
Transparent: Ideally,theusershouldnotbeawarethatauthentication

istakingplace,beyond therequirementtoenterapassword.

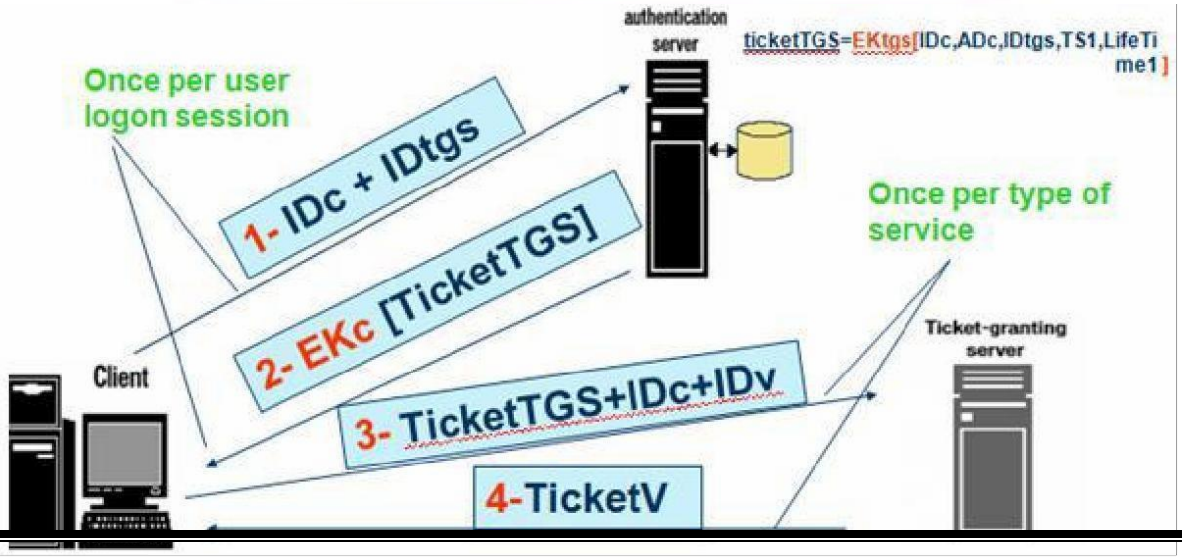
Scalable: The system should be capable of supporting large numbers of clients andserv Two versions of Kerberos are in common use: Version 4 is most widely used version.Version 5 corrects some of the security deficiencies of Version 4. Version 5 has beenissued asadraftInternetStandard(RFC1510) ers.Thissuggestsamodular,distributed architecture

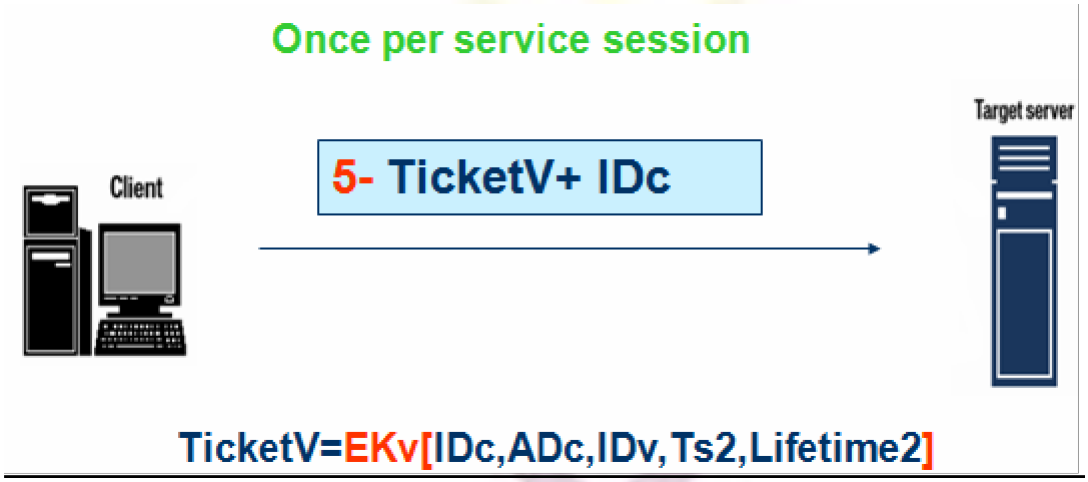
KERBEROSVERSION4

1.)SIMPLEDIALOGUE:



MORESECUREDIALOGUE





(1) C → AS $ID_c \| ID_{tgs} \| TS_1$

(2) AS → C $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

(a) Authentication Service Exchange to obtain ticket-granting ticket

(3) C → TGS $ID_v \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C $E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$

$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

(5) C → V $Ticket_v \| Authenticator_c$

(6) V → C $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

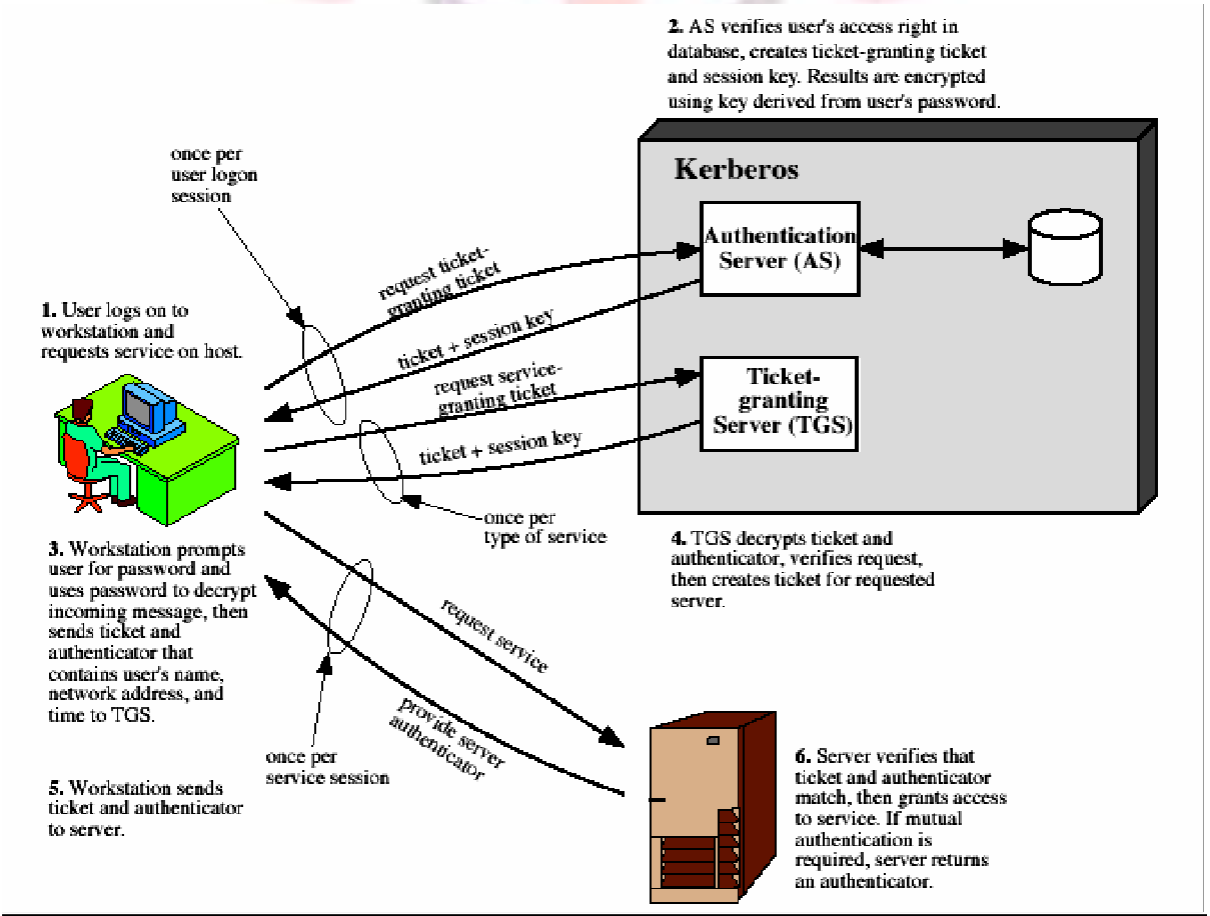
$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$

$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$

(c) Client/Server Authentication Exchange to obtain service

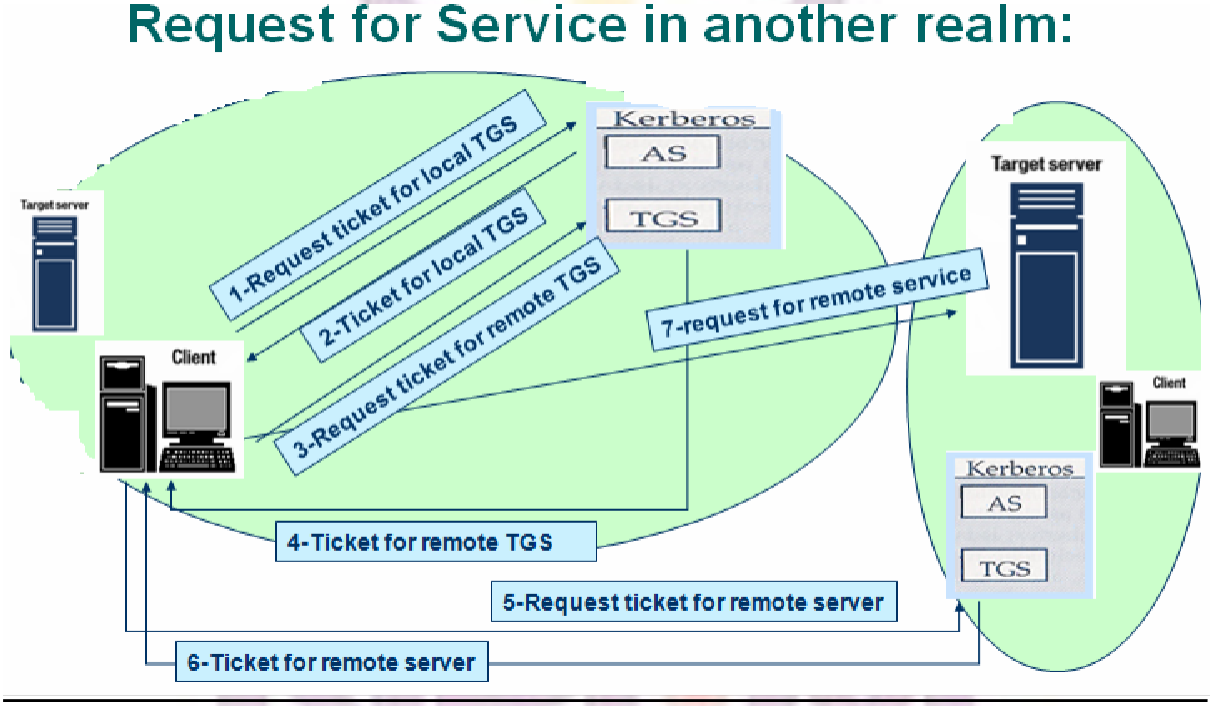
Thereisaproblemofcapturedticket-grantingticketsandtheneedtodetermine that the ticket presenter is the same as the client for whom the ticket wasissued. An efficient way of doing this is to use a session encryption key to secureinformation.

Message (1) includes a timestamp, so that the AS knows that the message is timely.Message (2) includes several elements of the ticket in a form accessible to C. Thisenables C to confirm that this ticket is for the TGS and to learn its expiration time.Note that the ticket does not prove anyone's identity but is a way to distribute keyssecurely.Itistheauthenticatorthatprovestheclient'sidentity.Becausetheauthenticatorcanbeusedonlyonceandhasashortlifetime,thethreatofanopponentstealingboththeticketandtheauthenticatorforpresentationlateriscountered. C then sends the TGS a message that includes the ticket plus the ID of therequested service (message 3). The reply from the TGS, in message (4), follows theformofmessage(2).Cnowhasareusableservice-grantingticketforV.WhenCpresentsthisticket,asshowninmessage(5),italsosendsanauthenticator



KerberosRealmsAfull-serviceKerberosenvironmentconsistingofaKerberosserver, a number of clients, and a number of application servers is referred to as aKerberos realm. A Kerberos realm is a set of managed nodes that share the sameKerberos database, and are part of the same administrative domain. If have multiplerealms,theirKerberos serversmust sharekeysand trusteachother.

Thefollowingfigureshowstheauthenticationmessageswhereserviceisbeingrequested from another domain. The ticket presented to the remote server indicatestherealminwhichtheuserwasoriginallyauthenticated.Theserverchooseswhether to honor the remote request. One problem presented by the foregoing approach isthatitdoesnotscalewellto manyrealms,aseachpair ofrealmsneedtoshareakey.



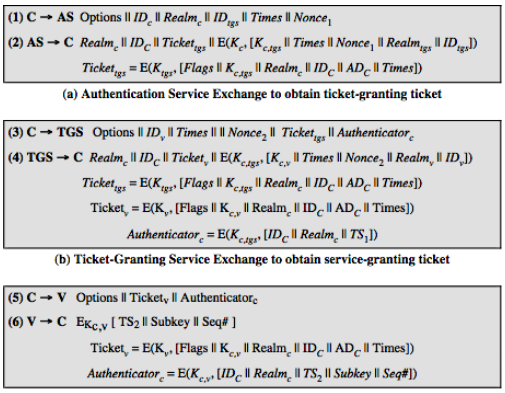
KERBEROSVERSION5

Kerberos Version 5 is specified in RFC 1510 and provides number ofimprovementsoverversion4intheareasofenvironmentalshortcomings and technical deficiencies. It includes some new elementssuchas:

- From:thedesiredstarttimefortheticket
- Till:therequestedexpirationtime
- Rtime:requestedrenew-tilltime

Nonce:Arandomvalueto assuretheresponseisfresh

ThebasicKerberosversion5authenticationdialogueisshownhereFirst,considerthe



Message (1) is a client request for a ticket-granting ticket. Message (2) returns aticket- granting ticket, identifying information for the client, and a block
encryptedusingtheencryptionkeybasedontheuser'spassword.Thisblockinclu desthesession key to be used between the client and the TGS. Now compare the **ticket-grantingservice**exchangeforversions4and5.Seethatmessage(3)forbothversi ons includes an authenticator, a ticket, and the name of the requested service.
Inaddition,version5includesrequestedtimesandoptionsfortheticketandanon ce,allwith functions similar to those of message (1). The authenticator itself is essentiallythe same as the one used in version 4. Message (4) has the same structure as message(2), returning a ticket plus information needed by the client, the latter encrypted withthe session key now shared by the client and the TGS. Finally, for the client/serverauthentication exchange, several new features appear in version 5, such as a requestfor mutual authentication. If required, the server responds with message (6) thatincludesthetimestampfromthe
authenticator. The flags field included in tickets in version 5 supportsexpandedfunctionalitycomparedtothatavailableinv ersion4.

AdvantagesofKerberos:

User's passwords are never sent across the network,
encryptedorinplaintextSecretkeysare*only*passedacrossthe networkin encrypted form
Clientandserversystems
mutuallyauthenticatelimitsthe It durationoftheirusers' authentication.

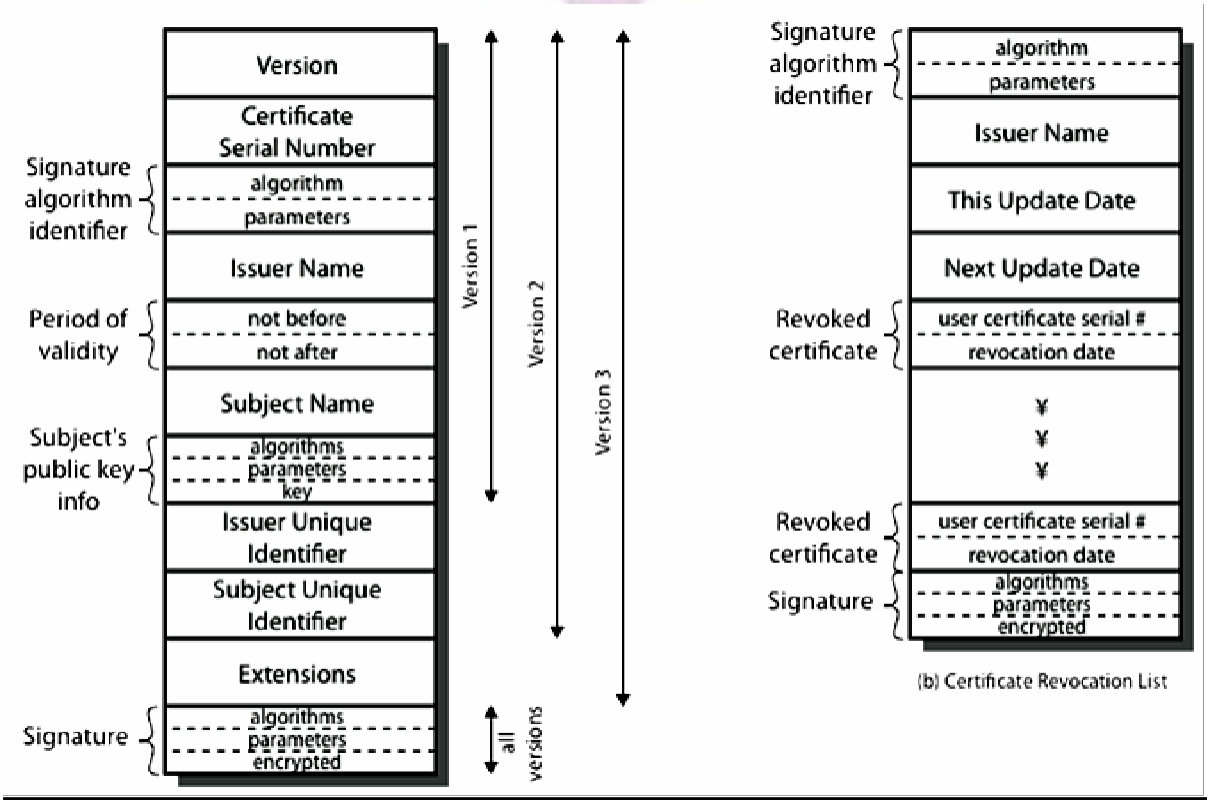
Authenticationsare**reusable**and**durable**

- Kerberoshasbeenscrutinizedbymanyofthetopprogrammers,cryptologists and securityexpertsintheindustry

X.509AUTHENTICATIONSERVICES

ITU-T recommendation X.509 is part of the X.500 series of recommendationsthat define a directory service. The directory is, in effect, a server or distributed set ofserversthatmaintainsadatabaseofinformationaboutusers.Theinformationincludes a mapping from user name to network address, as well as other attributesandinformationabouttheusers.X.509isbasedontheuseofpublic-keycryptography and digital signatures. The heart of the X.509 scheme is the public-keycertificateassociated witheach user. These user certificates are assumed to be createdby some trusted certification authority (CA) and placed in the directory by the CA orby the user. The directory server itself is not responsible for the creation of publickeys or for the certification function; it merely provides an easily accessible locationforuserstoobtaincertificates.

Thegeneralformatofacertificateisshownabove,whichincludesthefollowing elements :version1, 2,or3serialnumber(uniquewithinCA)identifying certificate signaturealgorithmidentifier issuerX.500name(CA)period of validity(from - to dates)subject X.500 name(nameofowner)subjectpublic-keyinfo(algorithm,parameters, key) issuer uniqueidentifier(v2+)



Thestandardusesthefollowingnotationtodefineacertificate:

CA<<A>>=CA{V,SN,AI,CA,TA,A,Ap}WhereY<<X>>=thecertificateofuserXissuedbycertificationauthorityYY{I}== thesigningofIbyY.ItconsistsofIwann encrypted

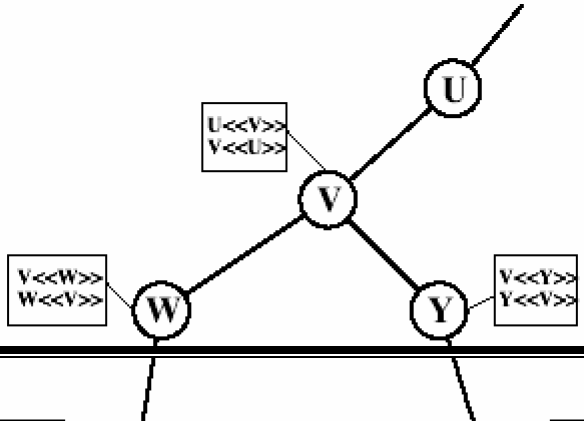
hashcodeappendedUsercertificatesgeneratedbyaCAhavethefollowingcharacteristics:

Any user with CA's public key can verify the user public keythatwascertifiedNopartyotherthantheCAcanmodifythecertificate without being detected because they cannot beforged,certificatescanbeplacedinapublicdirectory

Scenario: Obtaining a User Certificate If both users share a common CA then theyare assumed to know its public key. Otherwise CA's must form a hierarchy and usecertificateslinkingmembersofhierarchytovalidateotherCA's.EachCAhascertificates for clients (forward) and parent (backward). Each client trusts parentscertificates.Itenables verification of anycertificate fromone CAbyusers ofall other CAsinhierarchy.A has obtained a certificate from the CA X1.B has obtained a certificatefrom the CA X2.A can read the B's certificate but cannot verify it. In order to solve theproblem,theSolution:**X1<<X2>X2<<B>>.**AobtainthecertificateofX2signedby X1fromdirectory.ain X2's public key. A goes back to directory and obtain the certificate of B signedbyX2.obtainB'spublickey securely.ThedirectoryentryforeachCAincludestwotypesofcertificates:Forwardcertificates:CertificatesofXgeneratedbyotherCas Reversecertificates:CertificatesgeneratedbyXthatarethecertificatesofotherCas

X.509CAHierarchy

AacquiresBcertificate usingchain:X<<W>>W<<V>>V<<Y>>Y<<Z>>Z<<B>>
BacquiresAcertificate usingchain:Z<<Y>>Y<<V>>V<<W>>W<<X>>X<<A>>

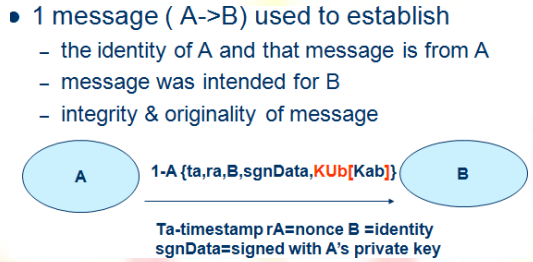


just before the expiration of the old one. In addition, it may be desirableon occasiontorevokeacertificate before itexpires,foroneofthefollowingreasons:Theuser'sprivatekeyisassumedtobe compromised. The user is nolongercertifiedbythisCA.TheCA'scertificateisassumedtobecompromised.Each CA must maintain a list consisting of all revoked but not expired certificatesissued by that CA, including both those issued to users and to other CAs. These listsshouldalsobepostedonthedirectory.Each**certificaterevocationlist(CRL)posted** to the directory is signed by the issuer and includes the issuer's name, thedatethelistwascreated,thedatethenextCRLisscheduledtobeissued,andanentryfor each revoked certificate. Each entry consists of the serial number of a certificateand revocation date for that certificate. Because serial numbers are unique within aCA,theserialnumberissufficienttoidentifythecertificate.

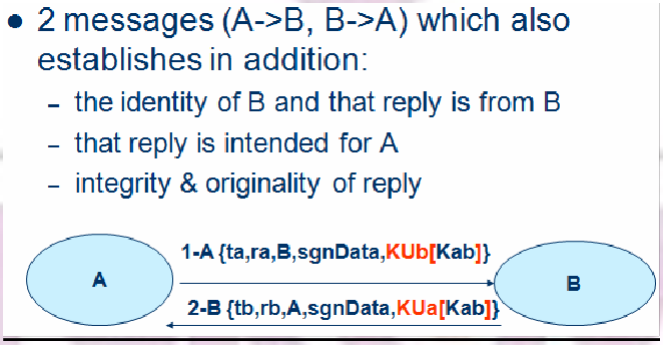
AUTHENTICATIONPROCEDURES

X.509 also includes three alternative authentication procedures that are intended for useacross a variety of applications. All these procedures make use of public-key signatures. Itis assumed that the two parties know each other's public key, either by obtaining eachother's certificates from the directory or because the

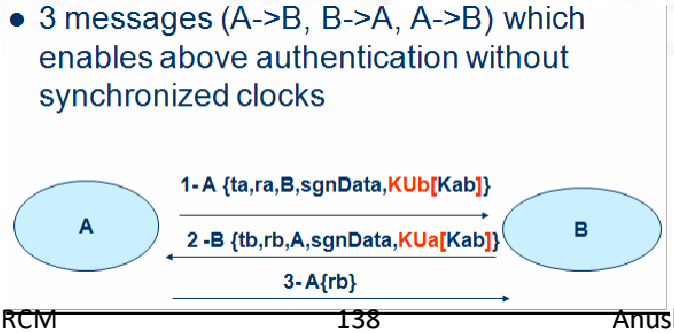
certificate is included in the initialmessage from each side. 1. One-Way Authentication: One way authentication involves asingletransferof information from one user (A) to another (B),andestablishesthedetailsshownabove. Notethatonlytheidentityoftheinitiatingentityis verifiedinthisprocess,notthatofthe responding entity. At a minimum, the message includes a timestamp ,a nonce, and theidentityofBandissignedwithA'sprivatekey.Themessagemayalsoincludeinformation tobeconveyed,suchasasessionkeyforB.



Two-Way Authentication: Two-way authenticationthuspermitsboth parties inacommunication to verify the identity of the other, thus additionally establishing theabove details. The reply message includes the nonce from A, to validate the reply. It alsoincludes a timestamp and nonce generated by B, and possible additional informationforA.



Three-Way Authentication:Three-Way Authenticationincludes a finalmessage from Ato B, which contains a signed copy of the nonce, so that timestamps need not bechecked,forusewhensynchronizedclocksarenotavailable.



Public-Key Infrastructure

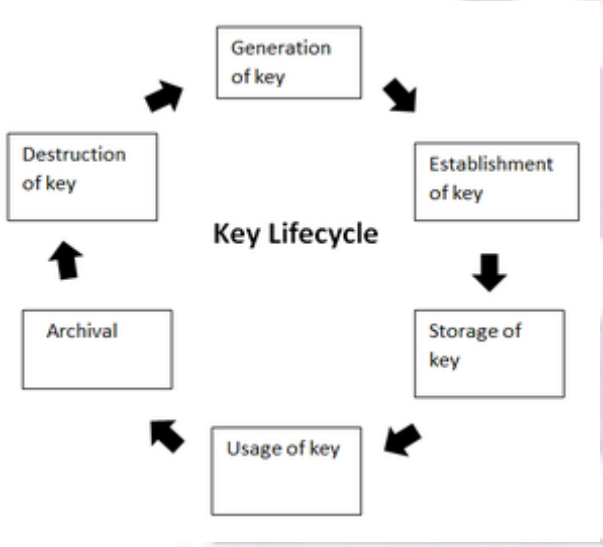
Public key infrastructure or PKI is the governing body behind issuing digital certificates. It helps to protect confidential data and gives unique identities to users and systems. Thus, it ensures security in communications.

The public key infrastructure uses a pair of keys: the public key and the private key to achieve security. The public keys are prone to attacks and thus an intact infrastructure is needed to maintain them

Managing Keys in the Cryptosystem:

The security of a cryptosystem relies on its keys. Thus, it is important that we have a solid key management system in place. The 3 main areas of key management are as follows:

- A cryptographic key is a piece of data that must be managed by secure administration.
- It involves managing the key life cycle which is as follows:



- Public key management further requires:
  - **Keeping the private key secret:** Only the owner of a private key is authorized to use a private key. It should thus remain out of reach of any other person.
  - **Assuring the public key:** Public keys are in the open domain and can be publicly accessed. When this extent of public accessibility, it becomes hard to know if a key is correct and what it will be used for. The purpose of a public key must be explicitly defined.

PKI or public key infrastructure aims at achieving the assurance of public key.

Public Key Infrastructure:

Public key infrastructure affirms the usage of a public key. PKI identifies a public key along with its purpose. It usually consists of the following components:

- A digital certificate also called a public key certificate
- Private Key tokens
- Registration authority
- Certification authority
- CMS or Certification management system

Working on a PKI:

Let us understand the working of PKI in steps.

- **PKI and Encryption:** The root of PKI involves the use of cryptography and encryption techniques. Both symmetric and asymmetric encryption uses a public key. The challenge here is – "how do you know that the public key belongs to the right person or to the person you think it belongs to?". There is always a risk of MITM(Man in the middle). This issue is resolved by a PKI using digital certificates. It gives identities to keys in order to make the verification of owners easy and accurate.
- **Public Key Certificate or Digital Certificate:** Digital certificates are issued to people and electronic systems to uniquely identify them in the digital world. Here are a few noteworthy things about a digital certificate. Digital certificates are also called X.509 certificates. This is because they are based on the ITU standard X.509.
    - The Certification Authority (CA) stores the public key of a user along with other information about the client in the digital certificate. The information is signed and a digital signature is also included in the certificate.
    - The affirmation for the public key then thus be retrieved by validating the signature using the public key of the Certification Authority.
- **Certifying Authorities:** A CA issues and verifies certificates. This authority makes sure that the information in a certificate is real and correct and it also digitally signs the certificate. A CA or ***Certifying Authority performs these basic roles***:
    - Generates the key pairs – This key pair generated by the CA can be either independent or in collaboration with the client.
    - Issuing of the digital certificates – When the client successfully provides the right details about his identity, the CA issues a certificate to the client. Then CA further signs this certificate digitally so that no changes can be made to the information.
    - Publishing of certificates – The CA publishes the certificates so that the users can find them. They can do this by either publishing them in an electronic telephone directory or by sending them out to other people.
    - Verification of certificate – CA gives a public key that helps in verifying if the access attempt is authorized or not.
    - Revocation – In case of suspicious behavior of a client or loss of trust in them, the CA has the power to revoke the digital certificate.

Classes of a Digital Certificate:

A digital certificate can be divided into four broad categories. These are :

- Class 1: These can be obtained by only providing the email address.
- Class 2: These need more personal information.
- Class 3: This first checks the identity of the person making a request.
- Class 4: They are used by organizations and governments.

Process of creation of certificate:

The creation of a certificate takes place as follows:

- Private and public keys are created.
- CA requests identifying attributes of the owner of a private key.
- Public key and attributes are encoded into a CSR or Certificate Signing Request.
- Key owner signs that CSR to prove the possession of a private key.
- CA signs the certificate after validation.

Creation of Trust layers among CA Hierarchies:

Each CA has its own certificate. Thus, trust is built hierarchically where one CA issues certificates to other CAs. Moreover, there is a root certificate that is self-signed. For a root CA, the issuer and the subject are not two separate parties but a single party.

Security of Root CA:

As you saw above, the ultimate authority is the root CA. Hence, the security of root CA is of huge importance. If the private key of a root CA is not taken care of, then it might turn into a catastrophe. This is because anyone disguised as the root CA can then issue certificates. To meet security standards, a root CA should be offline 99.9% of the time. However, it does need to come online to create public and private keys and to issue new certificates. Ideally, these activities should be performed 2-4 times a year.

Use of PKI in Today's Digital Age:

Today, there are an enormous number of applications that need require authentication. Certifications are needed at millions of places. This can not be done without a Public key infrastructure. The importance of PKI, depending on the use case and needs, has evolved over time. Here is a part of that track.

- For the very first time during the period of 1995 to 2002, the use of PKI was limited to the most important and high-value certificates. This included the certificates of eCommerce websites that enabled them to display the lock icon in the search bar. The goal was to make consumers confident about the security and authenticity of various websites.
- The second episode of PKI emerged around 2003 to 2010 when enterprises came into the picture. It was at this time that employees received laptops and the use of mobile phones was rising. Thus, employees needed access to the organization's assets even outside the office. That is when the use of PKI looked like the best way for authentication.
- The third phase started in 2011 and is continuing to date. With the advent of new technologies like IoT(Internet of Things) and need the to scale PKI, the use, as well as the challenges in using PKI, have increased tremendously. Today, millions of certificates are issued to authenticate mobile workforces. However, managing this huge number of certificates is quite challenging.
- S/MIME, Document Signing, code or app signing also uses PKI.

Challenges that a PKI Solves:

PKI owes its popularity to the various problems its solves. Some use cases of PKI are:

- Securing web browsers and communicating networks by SSL/TLS certifications.
- Maintaining Access Rights over Intranets and VPNs.
- Data Encryption
- Digitally Signed Software
- Wi-fi Access Without Passwords

Other than these, one of the most important use cases of PKI is based around IoT(Internet of Things). Here are two industries that are using PKI for IoT devices:

- **Auto Manufacturers:** Cars these days have features like GPS, call for services, assistants, etc. These require communication paths where a lot of data is passed. Making these connections secure is very important to avoid malicious parties hacking into the cars. This is where PKI comes in.
- **Medical device Manufacturers:** Devices like surgical robots require high security. Also, FDA mandates that any next-generation medical device must be updatable so that bugs can be removed and security issues can be dealt with. PKI is used to issues certificates to such devices.

Disadvantages of PKI:

- **Speed:** Since PKI uses super complex algorithms to create a secure key pair. So it eventually slows down the process and data transfer.

- **Private Key Compromise:** Even though PKI can't be hacked very easily but a private key can be hacked by a professional hacker, since PKI uses Public and Private key to encrypt and decrypt data so with user's private key in hand and public key which is easily available the information can be decrypted easily.

BIOMETRICAUTHENTICATION

BiometricauthenticationisatypeofsystemthatreliesontheuniquebiologicalcharacteristicsofindividualstoverifyidentityforsecureaccesstoelectronicsystemsBiometric verification is considered asubsetofbiometricauthentication.The biometric technologies involved are based on the ways in which individuals canbe uniquely identified through one or more distinguishing biological traits, such asfingerprints, hand geometry, earlobe geometry, retina and iris patterns, voice waves,keystroke dynamics, DNA and signatures. Biometric authentication is the applicationofthat proofofidentityas part ofa processvalidatingauserforaccesstoa system.Biometrictechnologiesareusedtosecureawiderangeofelectroniccommunications, including enterprise security, online commerce and banking -- evenjustloggingin toacomputerorsmartphone.

Biometricauthentication systems comparethecurrentbiometricdatacapturetostored, confirmed authentic data in a database. If both samples of the biometric datamatch,authentication is confirmed and access is granted. The process is sometimes part ofamultifactor authentication system. For example, a smartphone user

might log onwithhispersonalidentificationnumber(PIN)andthenprovideanirisscantoco mpletetheauthenticationprocess.

Types of biometric authentication technologies:

Retinascan

Iris recognitionisused toidentify individualsbased on uniquepatternswithin thering-shapedregionsurroundingthepupiloftheeye.

Fingerscanning,thedigital version of theink-and-paperfingerprintingprocess,workswithdetailsinthepatternofraisedareasandbran chesinahumanfingerimage.

FingerveinIDisbasedontheuniquevascularpatterninanindividual'sfinger.

Facialrecognitionsystemsworkwithnumericcodescalledfaceprints,whichidentify8 0 nodalpointson ahuman face.

Voiceidentificationsystemsrelyoncharacteristicscreatedbytheshapeofthes peaker'smouth andthroat,ratherthanmorevariableconditions.

Once seen mostly in spy movies (where it might be used to protect access to a top-secretmilitarylab,forexample),biometricauthenticationisbecoming relativelycommonplace.Inadditiontothesecurityprovidedbyhard-to-fakeindividualbiological traits, the acceptance of biometricverification hasalso been driven byconvenience:Onecan’teasilyforgetorloseonesbiometrics.

Thehistoryofbiometricverification:

Theoldestknownuseofbiometricverificationisfingerprinting.Thumbprintsmadeon clay seals were used as a means of unique identification as far back as ancientChina.Modernbiometricverificationhasbecomealmostinstantaneous,an disincreasinglyaccurate with the advent of computerized databases and the digitization of analogdata.

The market for biometrics products is still too fractured to name specific topproviders. The physical characteristics of the biometrics products available todayvaryfromthemundane,suchasfingerprinting,totheesoteric,liketypingspeedsandele ctrophysiologicalsignals

Untilrecently,biometricswastypicallyusedataphysicalsecuritylevel–protectingfacilities at military bases or impenetrable bank vaults, for example. But, becausesingle- factor authentication methods are easy to break, companies have startedlookingto two-factorsolutions,likebiometrics.

However,thefollowingfivefundamentalbarriersmaylimitthegrowthofbiometricauthenticati on:

1. Biometrics can becomplicated andcostlytodeploy.All biometric

deploymentsrequireinstallationoftheirownhardwareandapplications ervers.

2. The market is still fractured. Should you buy a fingerprint reader, a voicerecognition system or an iris scanner? Since each product differs greatly in itsapproachand installation,it isdifficult to comparethemduring atypical companybidprocess.
3. Biometricdataislikeany otherdata.Itsitsonservers,whicharebaitforhackersifnot properly hardened and secured. Therefore, when reviewing any biometricproduct, make sure it transmits data securely, meaningencrypted, from thebiometric reader back to the authenticating server. And, make sure theauthenticatingserverhasbeenhardened,patchedandprotected.
4. Biometricreadersarepronetoerrors.Fingerprintscansmudge,facesan dvoicescan be changed and all of them can be misread, blocking a legitimate user, orpermitting accesstoan unauthorized ormalicioususer.
5. Difficulties with user acceptance. Properly trained employees may be willing tousebiometricsdevices,butcustomers,likethoseloggingontoyourWebsit e,maybe more reluctant to use – or worse, forced to purchase – a device that's difficulttouseormakesdoingbusiness,suchasbanking,onyoursite,ahassle insteadofconvenience. And both your employees and customers may be squeamish aboutexposingtheireyestodeviceslikeirisscanners,eveniftheyappearhar mless.

Despite these issues, biometrics is slowly gaining acceptance for two-factorauthentication purposes. The products are getting better, lighter and easier to use.Error rates are going down, and fingerprint readers installed on tokens and laptopsare getting smaller and less intrusive. And, like the rest of the security productindustry,vendorswilleventuallymergeandconsolidate,unitingafractur edmarket,whichwillmakeiteasiertochooseaproductthatsuitsyourbusinessnee ds.

DescriptiveQuestions:

(a) 2MarksQuestions

1. Listouttheapplicationsofthepublickeycryptosystems.

- Encryption/decryption
- Digitalsignature
- Keyexchange

2. Listoutthedifferenttechniquesofdistributingthepublickey.

- Publicannouncement
- Publiclyavailabledirectory
- Publickeyauthority
- Publickeycertificate

3. WhatismeantbyMessageAuthentication?

Message Authentication is a mechanism or service used to verify the integrity of a message.

Messageauthenticationassuresthatdatareceivedareexactlyassentby(i.e.,contain nomodification,insertion,deletion,
orreplay)andthatthepurportedidentityofthesenderisvalid.

4. Definetheclassesofmessageauthenticationfunction.

- Messageencryption
- MessageAuthenticationCode(MAC)
- Hashfunction

5. DefineHashfunction.
A function that maps a message of any length into a fixed length hash value, which serves as theauthenticator

6. DifferentiateMessageAuthenticationCodeandHashfunction.

InMAC,apublicfunctionofthemessageandasecretkeyareusedtoproduceafix edlengthauthenticator.

Ahashfunctionacceptsavariablesizemessageasinputandproduces afixedsizeoutput(hashcode)whichissimilartoMAC.Buthashcodedoesnotuseakey.

7. Defineonewayproperty,weakcollisionresistanceandstrongcollisionresistance ofhashfunction.

Foranygivenvalueh,itiscomputationallyinfeasibletofindxsuchthatH(x)=h– onewayproperty.

Foranygivenblockx,itiscomputationallyinfeasibletofindy≠xwithH(y)=H(x)–weakcollision resistance.

Itiscomputationallyinfeasibletofindanypair(x,y)suchthatH(x)=H(y)– strongcollisionproperty.

8. WhatyoumeantbyMAC?

MACisMessageAuthenticationCode.Itisafunctionofmessageandsecretk eywhichproduceafixedlength valuecalledasMAC.

T=MAC(K,M)

whereMisavariable-
lengthmessage,Kisasecretkeysharedonlybysenderandreceiver,andMAC(K,M)isthefi xed-lengthauthenticator.

9. ListouttheattackonMAC.

- Brute-forceattacks
- Cryptanalysis.

10. DefineDigitalsignature.

A digital signature is an authentication mechanism that enables the creator of a message toattach a code that acts as a signature. Typically the signature is formed by

taking the hash of themessage and encrypting the message with the creator's private key. The signature guarantees thesourceand integrity ofthemessage

11. WhatarethepropertiesofDigitalSignature?

Thedigitalsignaturemusthavethefollowingproperties:
Itmustverifytheauthorandthedateandtimeofthesignature.Itmustauthenticatethecontent satthetimeofthesignature.Itmustbeverifiablebythirdparties,toresolvedisputes

12. ListouttheattacksrelatedtoDigitalSignature.

Key-onlyattack:

- Knownmessageattack
- Genericchosenmessageattack
- Directedchosenmessageattack
- Adaptivechosenmessageattack

13. MentionthesignaturefunctioninDSS?

Thehash**function**usedinthe**DSS**standardisspecifiedintheSecureHashStandard(SHS), whicharethespecificationsfortheSecureHashAlgorithm(SHA).

14. DefineUniversalforgery

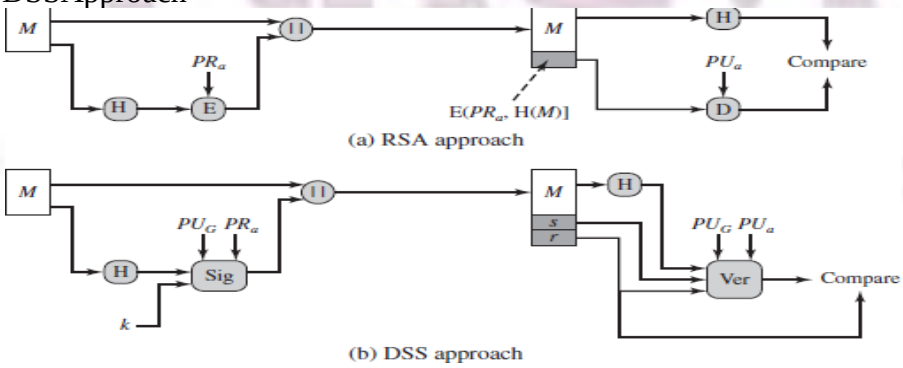
If A is the sender and C is the attacker. Then Cfinds an efficientsigning algorithmthatprovides anequivalentwayofconstructingsignaturesonarbitrarymessages.

15. DefineExistentialforgery

IfAisthesenderandCistheattacker.ThenCforgesasignatureforatleastonemessage.Chas nocontroloverthemessage.Consequently,thisforgerymayonlybeaminornuisancetoA.

16. WhatarethetwoapproachesofDigitalSignature?

- RSAApproach
- DSSApproach



(b))10MarksQuestions

1. Withtheexample,explainindetailaboutSecureHash Algorithm
2. ExplainindetailaboutHMACandDigitalSignatureStandard
3. A)Explainmessageauthentication requirements.Whatare the attacks related tomessagecommunication?
   B)Giveabriefnoteonbasicusesofmessageauthenticationcode.
4. A)Explaintheprocessinvolvedinmessagedigestgenerationandprocessingofsingleblock inSHA-512.
   B)Whataretheapproachesofmessageauthentication?Explainthem.
5. A)Explainaboutcharacteristicsofhashfunctions.
   B)Whatisthepurposeofdigitalsignature? Explainitspropertiesandrequirements.
6. A)Writeshortnotesonauthenticationprotocols.

   B)Explainthevarioustypesofcryptographicfunctionswithanexample.
7. Explain the requirements of digital signatures and also discuss how problems related to digitalsignatureaaretakencarebyanarbiter?
8. Statetheneedforauthenticationprotocolsandexplainany threeofthemDescribeMD5.CompareitwithMD4
   DescribeSHA-1
   DescribeRIPEMD/HMACalgorithms
9. Stateandexplainthedifferentapproachestomessageauthentication
10. Explainthevariousmethodsofproducinganauthenticaton

UNIT-4
WEB SECURITY CONSIDERATIONS

The World Wide Web is fundamentally a client/server application running over the Internet and TCP/IP intranets. As such, the security tools and approaches discussed so far in this book are relevant to the issue of Web security. But, as pointed out in [GARF02], the Web presents new challenges not generally appreciated in the context of computer and network security.

The Internet is two-way. Unlike traditional publishing environments—even electronic publishing systems involving teletext, voice response, or fax-back— the Web is vulnerable to attacks on the Web servers over the Internet.

- The Web is increasingly serving as a highly visible outlet for corporate and product information and as the platform for business transactions. Reputations can be damaged and money can be lost if the Web servers are subverted.

- Although Web browsers are very easy to use, Web servers are relatively easy to configure and manage, and Web content is increasingly easy to develop, the underlying software is extraordinarily complex. This complex software may hide many potential security flaws. The short history of the Web is filled with examples of new and upgraded systems, properly installed, that are vulnerable to a variety of security attacks.

- A Web server can be exploited as a launching pad into the corporation's or agency's entire computer complex. Once the Web server is subverted, an attacker may be able to gain access to data and systems not part of the Web itself but connected to the server at the local site.

- Casual and untrained (in security matters) users are common clients for Web-based services. Such users are not necessarily aware of the security risks that exist and do not have the tools or knowledge to take effective countermeasures.

WEB SECURITY THREATS

Table 16.1 provides a summary of the types of security threats faced when using the Web. One way to group these threats is in terms of passive and active attacks. Passive attacks include eavesdropping on network traffic between browser and server and gaining access to information on a Web site that is supposed to be restricted. Active attacks include impersonating another user, altering messages in transit between client and server, and altering information on a Web site.

Another way to classify Web security threats is in terms of the location of the threat: Web server, Web browser, and network traffic between browser and server. Issues of server and browser security fall into the category of computer system security; Part Four of this book addresses the issue of system security in general but is also applicable to Web system security. Issues of traffic security fall into the category of network security and are addressed in this chapter.

WEB TRAFFIC SECURITY APPROACHES

A number of approaches to providing Web security are possible. The various approaches that have been considered are similar in the services they provide and, to some extent, in the mechanisms that they use, but they differ with respect to their scope of applicability and their relative location within the TCP/IP protocol stack.

Figure 16.1 illustrates this difference. One way to provide Web security is to use IP security (IPsec) (Figure 16.1a). The advantage of using IPsec is that it is trans- parent to end users and applications and provides a general-purpose solution. Furthermore, IPsec

includes a filtering capability so that only selected traffic need incur the overhead of IPsec processing.

Another relatively general-purpose solution is to implement security just above TCP (Figure 16.1b). The foremost example of this approach is the Secure

Table 16.1 A Comparison of Threats on the Web

	Threats	Consequences	Countermeasures
Integrity	• Modification of user data • Trojan horse browser • Modification of memory • Modification of message traffic in transit	• Loss of information • Compromise of machine • Vulnerabilty to all other threats	Cryptographic checksums
Confidentiality	• Eavesdropping on the net • Theft of info from server • Theft of data from client • Info about network configuration • Info about which client talks to server	• Loss of information • Loss of privacy	Encryption, Web proxies
Denial of Service	• Killing of user threads • Flooding machine with bogus requests • Filling up disk or memory • Isolating machine by DNS attacks	• Disruptive • Annoying • Prevent user from getting work done	Difficult to prevent
Authentication	• Impersonation of legitimate users • Data forgery	• Misrepresentation of user • Belief that false information is valid	Cryptographic techniques

Sockets Layer (SSL) and the follow-on Internet standard known as Transport Layer Security (TLS). At this level, there are two implementation choices. For full generality, SSL (or TLS) could be provided as part of the underlying protocol suite and therefore be transparent to applications. Alternatively, SSL can be embedded in specific packages. For example, Netscape and Microsoft Explorer browsers come equipped with SSL, and most Web servers have implemented the protocol.

Application-specific security services are embedded within the particular application. Figure 16.1c shows examples of this architecture. The advantage of this approach is that the service can be tailored to the specific needs of a given application.

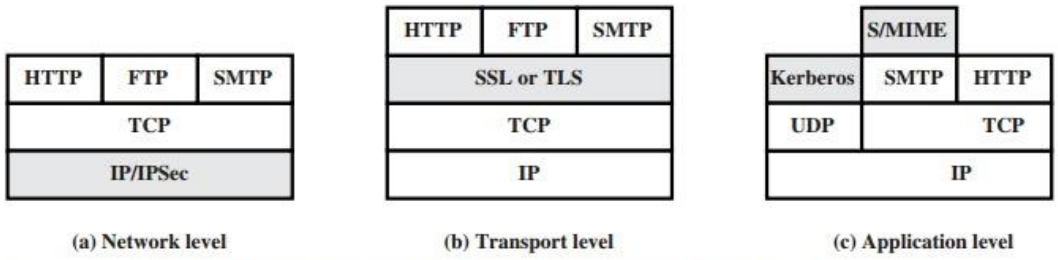


Figure 16.1 Relative Location of Security Facilities in the TCP/IP Protocol Stack

SECURE SOCKET LAYER (SSL)

Secure Socket Layer provides security to the data that is transferred between web browser and server. SSL encrypts the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.

Secure Socket Layer Protocols:

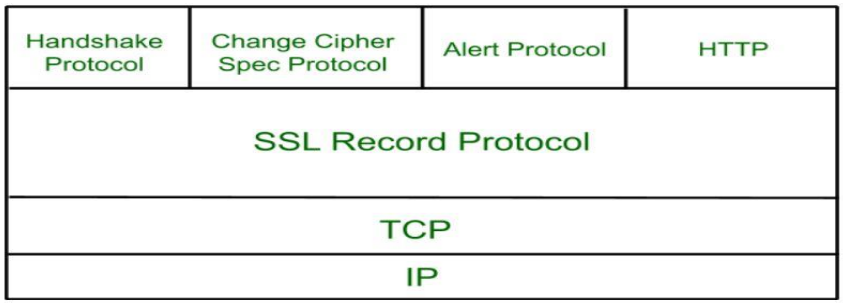
•SSL record protocol

•Handshake protocol

•Change-cipher spec protocol

•Alert protocol

SSL PROTOCOL STACK:



SSL RECORD PROTOCOL:

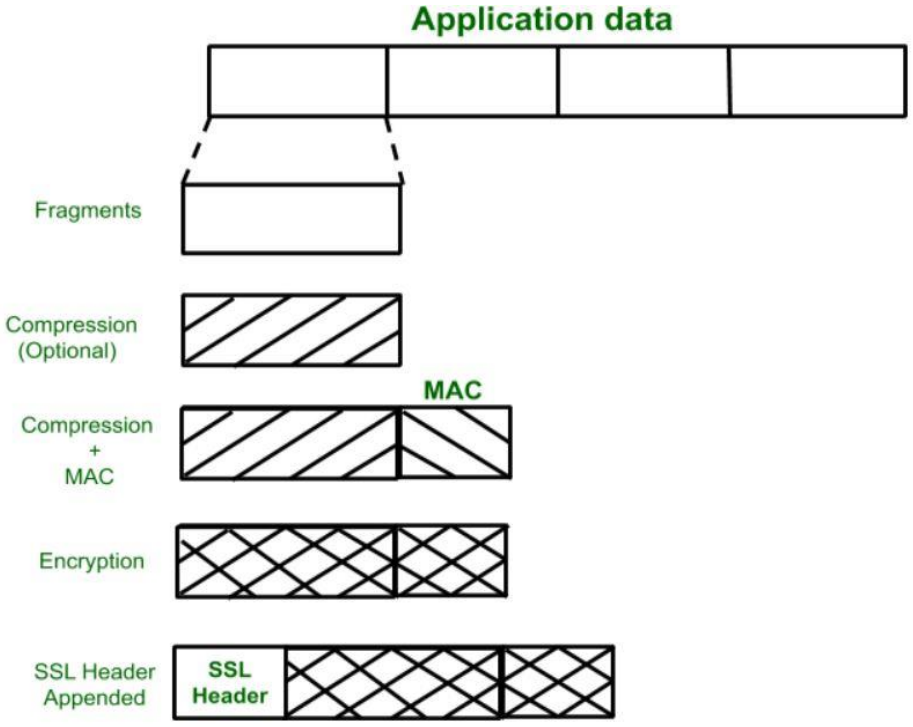
SSL Record provides two services to SSL connection.

•Confidentiality

•Message Integrity

In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by

algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.



HANDSHAKE PROTOCOL:

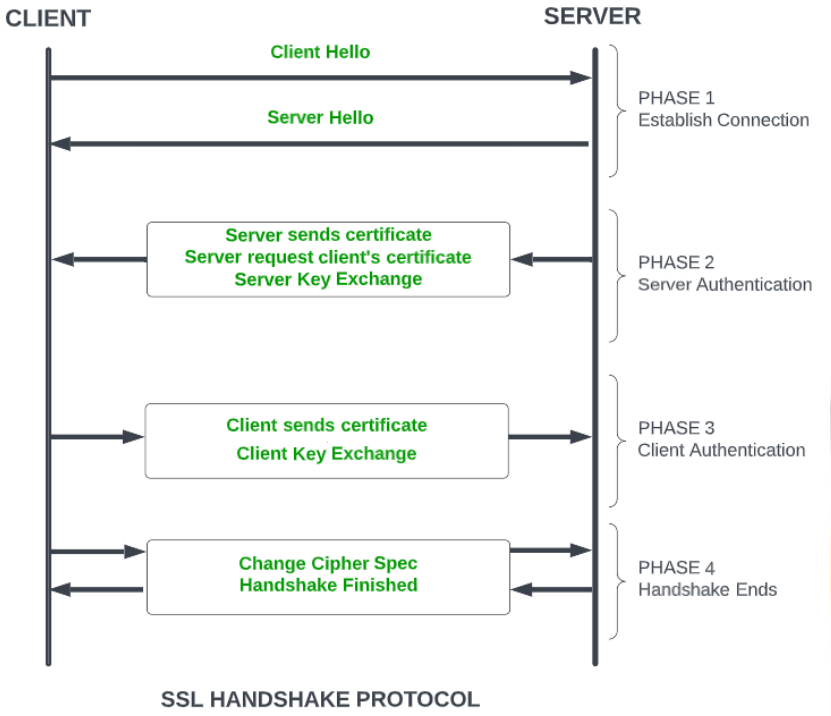
Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

PHASE-1: In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.

PHASE-2: Server sends his certificate and Server-key-exchange. The server end phase-2 by sending the Server-hello-end packet.

PHASE-3: In this phase, Client replies to the server by sending his certificate and Client-exchange-key.

PHASE-4: In Phase-4 Change-cipher suite occurred and after this Handshake Protocol ends.



SSL HANDSHAKE PROTOCOL

CHANGE-CIPHERPROTOCOL:

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After the handshake protocol, the Pending state is converted into the current state.

Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol's purpose is to cause the pending state to be copied into the current state.

ALERT PROTOCOL:

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contains 2 bytes.

The level is further classified into two parts:

WARNING (LEVEL= 1):

This Alert has no impact on the connection between sender and receiver. Some of them are:

BAD CERTIFICATE: When the received certificate is corrupt.

NO CERTIFICATE: When an appropriate certificate is not available.

CERTIFICATE EXPIRED: When a certificate has expired.

CERTIFICATE UNKNOWN: When some other unspecified issue arose in processing the certificate, rendering it unacceptable.

CLOSE NOTIFY: It notifies that the sender will no longer send any messages in the connection.

FATAL ERROR (LEVEL = 2):

This Alert breaks the connection between sender and receiver. The connection will be stopped, cannot be resumed but can be restarted. Some of them are:

HANDSHAKEFAILURE: When the sender is unable to negotiate an acceptable set of security parameters given the options available.

DECOMPRESSION FAILURE: When the decompression function receives improper input.

ILLEGAL PARAMETERS: When a field is out of range or inconsistent with other fields.

BADRECORD MAC: When an incorrect MAC was received.

UNEXPECTED MESSAGE: When an inappropriate message is received.

The second byte in the Alert protocol describes the error.

SILENT FEATURES OF SECURE SOCKET LAYER:

•The advantage of this approach is that the service can be tailored to the specific needs of the given application.

•Secure Socket Layer was originated by Netscape.

•SSL is designed to make use of TCP to provide reliable end-to-end secure service.

•This is a two-layered protocol.

VERSIONSOF SSL:

SSL 1 – Never released due to high insecurity.

SSL 2 – Released in 1995.

SSL 3 – Released in 1996.

TLS 1.0 – Released in 1999.

TLS 1.1 – Released in 2006.

TLS 1.2 – Released in 2008.

TLS 1.3 – Released in 2018.

TRANSPORT LAYER SECURITY

TLS was released in response to the Internet community's demands for a standardized protocol. TLS (Transport Layer Security), defined in RFC 2246, is a protocol for establishing a secure connection between a client and a server. TLS (Transport Layer Security) is capable of authenticating both the client and the server and creating a encrypted connection between the two. Many protocols use TLS (Transport Layer Security) to establish secure connections, including HTTP, IMAP, POP3, and SMTP. The TLS Handshake Protocol first negotiates key exchange using an asymmetric algorithm such as RSA or Diffie-Hellman. The TLS Record Protocol then begins opens an encrypted channel using a symmetric algorithm such as RC4, IDEA, DES, or 3DES. The TLS Record Protocol is also responsible for ensuring that the communications are not altered in transit. Hashing algorithms such as MD5 and SHA are used for this purpose. RFC 2246 is very similar to SSLv3. There are some minor differences ranging from protocol version numbers to generation of key material.

Version Number: The TLS Record Format is the same as that of the SSL Record Format and the fields in the header have the same meanings. The one difference is in version values. For the current version of TLS, the Major Version is 3 and the Minor Version is 1.

Message Authentication Code: Two differences arise one being the actual algorithm and the other being scope of MAC calculation. TLS makes use of the HMAC algorithm defined in RFC 2104. SSLv3 uses the same algorithm, except that the padding bytes are concatenated with the secret key rather than being XORed with the secret key padded to the block length. For TLS, the MAC calculation encompasses the fields indicated in the following expression:

HMAChash (MACwrite_secret, seq_num || TLSCompressed.type || TLSCompressed.version || TLSCompressed.length || TLSCompressed.fragment)

The MAC calculation covers all of the fields covered by the SSLv3 calculation, plus the field TLSCompressed.version, which is the version of the protocol being employed.

Pseudorandom Function: TLS makes use of a pseudorandom function referred to as PRF to expand secrets into blocks of data for purposes of key generation or validation. The PRF is based on the following data expansion function:

P_hash(secret, seed) = HMAC_hash(secret, A(1) || seed) || HMAC_hash(secret, A(2) || seed) || HMAC_hash(secret, A(3) || seed) || ...

where A () is defined as A(0) =seed

A(i) = HMAC_hash (secret, A(i - 1))

The data expansion function makes use of the HMAC algorithm, with either MD5 or SHA- 1 as the underlying hash function. As can be seen, P_hash can be iterated as many times as necessary to produce the required quantity of data. each iteration involves two executions of HMAC, each of which in turn involves two executions of the underlying hash algorithm.

HTTPS

WHATIS HTTPS?

HTTPS (Hypertext Transfer Protocol Secure) is a secure version of the HTTP protocol that uses the SSL/TLS protocol for encryption and authentication. HTTPS is specified by RFC 2818 (May 2000) and uses port 443 by default instead of HTTP's port 80.

The HTTPS protocol makes it possible for website users to transmit sensitive data such as credit card numbers, banking information, and login credentials securely over the internet. For this reason, HTTPS is especially important for securing online activities such as shopping, banking, and remote work. However, HTTPS is quickly becoming the standard protocol for all websites, whether or not they exchange sensitive data with users.

HOWIS HTTPS DIFFERENTFROM HTTP?

HTTPS adds encryption, authentication, and integrity to the HTTP protocol:

ENCRYPTION: Because HTTP was originally designed as a clear text protocol, it is vulnerable to eavesdropping and man in the middle attacks. By including SSL/TLS encryption, HTTPS prevents data sent over the internet from being intercepted and read by a third party. Through public-key cryptography and the SSL/TLS handshake, an encrypted communication session can be securely set up between two parties who have

never met in person (e.g. a web server and browser) via the creation of a shared secret key.

AUTHENTICATION: Unlike HTTP, HTTPS includes robust authentication via the SSL/TLS protocol. A website's SSL/TLS certificate includes a public key that a web

browser can use to confirm that documents sent by the server (such as HTML pages) have been digitally signed by someone in possession of the corresponding private key. If the server's certificate has been signed by a publicly trusted certificate authority (CA), such as SSL.com, the browser will accept that any identifying information included in the certificate has been validated by a trusted third party.

HTTPS websites can also be configured for mutual authentication, in which a web browser presents a client certificate identifying the user. Mutual authentication is useful for situations such as remote work, where it is desirable to include multi-factor authentication, reducing the risk of phishing or other attacks involving credential theft. For more information on configuring client certificates in web browsers, please read this how-to.

INTEGRITY: Each document (such as a web page, image, or JavaScript file) sent to a browser by an HTTPS web server includes a digital signature that a web browser can use to determine that the document has not been altered by a third party or otherwise corrupted while in transit. The server calculates a cryptographic hash of the document's contents, included with its digital certificate, which the browser can independently calculate to prove that the document's integrity is intact.

Taken together, these guarantees of encryption, authentication, and integrity make HTTPS a much safer protocol for browsing and conducting business on the web than HTTP.

What information does HTTPS provide users about website owners?

CAs use three basic validation methods when issuing digital certificates. The validation method used determines the information that will be included in a website's SSL/TLS certificate:

• **DOMAIN VALIDATION (DV)** simply confirms that the domain name covered by the certificate is under the control of the entity that requested the certificate.

• **ORGANIZATION / INDIVIDUAL VALIDATION (OV/IV)** certificates include the validated name of a business or other organization (OV), or an individual person (IV).

• **EXTENDED VALIDATION (EV)** certificates represent the highest standard in internet trust, and require the most effort by the CA to validate. EV certificates are only issued to businesses and other registered organizations, not to individuals, and include the validated name of that organization.

WHY USE HTTPS?

There are multiple good reasons to use HTTPS on your website, and to insist on HTTPS when browsing, shopping, and working on the web as a user:

INTEGRITYAND AUTHENTICATION: Through encryption and authentication, HTTPS protects the integrity of communication between a website and a user's browsers. Your users will know that the data sent from your web server has not been intercepted and/or altered by a third party in transit. And, if you've made the extra investment in EV or OV certificates, they will also be able to tell that the information really came from your business or organization.

PRIVACY: Of course, no one wants intruders scooping up their credit card numbers and passwords while they shop or bank online, and HTTPS is great for preventing that. But would you really want everything else you see and do on the web to be an open book for anyone who feels like snooping (including governments, employers, or someone building a profile to de-anonymize your online activities)? HTTPS plays an important role here too.

USER EXPERIENCE: Recent changes to browser UI have resulted in HTTP sites being flagged as insecure. Do you want your customers' browsers to tell them that your website is "Not Secure" or show them a crossed-out lock when they visit it? Of course not!

COMPATIBILITY: Current browser changes are pushing HTTP ever closer to incompatibility. Mozilla Firefox recently announced an optional HTTPS-only mode, while Google Chrome is steadily moving to block mixed content (HTTP resources linked to HTTPS pages). When viewed together with browser warnings of "insecurity" for HTTP websites, it's easy to see that the writing is on the wall for HTTP. In 2020, all current major browsers and mobile devices support HTTPS, so you won't lose users by switching from HTTP.

SEO: Search engines (including Google) use HTTPS as a ranking signal when generating search results. Therefore, website owners can get an easy SEO boost just by configuring their web servers to use HTTPS rather than HTTP.

In short, there are no longer any good reasons for public websites to continue to support HTTP. Even the United States government is on board!

HOWDOES HTTPS WORK?

HTTPS adds encryption to the HTTP protocol by wrapping HTTP inside the SSL/TLS protocol (which is why SSL is called a tunneling protocol), so that all messages are encrypted in both directions between two networked computers (e.g. a client and web server). Although an eavesdropper can still potentially access IP addresses, port numbers, domain names, the amount of information exchanged, and the duration of a session, all of the actual data exchanged are securely encrypted by SSL/TLS, including:

• Request URL (which web page was requested by the client)

• Website content

• Query parameters

• Headers

• Cookies

HTTPS also uses the SSL/TLS protocol for authentication. SSL/TLS uses digital documents known as X.509 certificates to bind cryptographic key pairs to the identities of entities such as websites, individuals, and companies. Each key pair includes a private key, which is kept secure, and a public key, which can be widely distributed. Anyone with the public key can use it to:

• Send a message that only the possessor of the private key can decrypt.

• Confirm that a message has been digitally signed by its corresponding private key.

If the certificate presented by an HTTPS website has been signed by a publicly trusted certificate authority (CA), such as SSL.com, users can be assured that the identity of the website has been validated by a trusted and rigorously-audited third party.

WHAT HAPPENS IF MY WEBSITE DOESN'T USE HTTPS?

In 2020, websites that do not use HTTPS or serve mixed content (serving resources like images via HTTP from HTTPS pages) are subject to browser security warnings and errors. Furthermore, these websites unnecessarily compromise their users' privacy and security, and are not preferred by search engine algorithms. Therefore, HTTP and mixed-content websites can expect more browser warnings and errors, lower user trust and poorer SEO than if they had enabled HTTPS.

HOW DO I KNOW IF A WEBSITE USES HTTPS?

An HTTPS URL begins with https:// instead of http://. Modern web browsers also indicate that a user is visiting a secure HTTPS website by displaying a closed padlock symbol to the left of the URL:

In modern browsers like Chrome, Firefox, and Safari, users can click the lock to see if an HTTPS website's digital certificate includes identifying information about its owner.

SECURE SHELL (SSH)

WHAT IS SSH?

SSH, also known as Secure Shell or Secure Socket Shell, is a network protocol that gives users, particularly system administrators, a secure way to access a computer over an unsecured network.

SSH also refers to the suite of utilities that implement the SSH protocol. Secure Shell provides strong password authentication and public key authentication, as well as encrypted data communications between two computers connecting over an open network, such as the internet.

In addition to providing strong encryption, SSH is widely used by network administrators to manage systems and applications remotely, enabling them to log in to another computer over a network, execute commands and move files from one computer to another.

SSH refers both to the cryptographic network protocol and to the suite of utilities that implement that protocol. SSH uses the client-server model, connecting a Secure Shell client application, which is the end where the session is displayed, with an SSH server, which is the end where the session runs. SSH implementations often include support for application protocols used for terminal emulation or file transfers.

SSH can also be used to create secure tunnels for other application protocols, for example, to securely run X Window System graphical sessions remotely. An SSH server, by default, listens on the standard Transmission Control Protocol (TCP) port 22.

HOW DOES SSH WORK?

Secure Shell was created to replace insecure terminal emulation or login programs, such as Telnet, rlogin (remote login) and rsh (remote shell). SSH enables the same functions -- logging in to and running terminal sessions on remote systems. SSH also replaces file transfer programs, such as File Transfer Protocol (FTP) and rcp (remote copy).

The most basic use of SSH is to connect to a remote host for a terminal session. The form of that command is the following:

```
ssh UserName@SSHserver.example.com
```

This command will cause the client to attempt to connect to the server named server.example.com, using the user ID UserName. If this is the first time negotiating a connection between the local host and the server, the user will be prompted with the remote host's public key fingerprint and prompted to connect, despite there having been no prior connection:

```
The authenticity of host 'sample.ssh.com' cannot be
established.
 DSA key fingerprint is
01:23:45:67:89:ab:cd:ef:ff:fe:dc:ba:98:76:54:32:10.
 Are you sure you want to continue connecting (yes/no)?
```

Answering yes to the prompt will cause the session to continue, and the host key is stored in the local system's known_hosts file. This is a hidden file, stored by default in a hidden directory, called /.ssh/known_hosts, in the user's home directory. Once the host key has been stored in the known_hosts file, the client system can connect directly to that server again without need for any approvals; the host key authenticates the connection.

WHAT IS SSH USED FOR?

Present in all data centers, SSH ships by default with every Unix, Linux and Mac server. SSH connections have been used to secure many different types of communications between a local machine and a remote host, including secure remote access to resources, remote execution of commands, delivery of software patches, and updates and other administrative or management tasks.

In addition to creating a secure channel between local and remote computers, SSH is used to manage routers, server hardware, virtualization platforms, operating systems (OSes), and inside systems management and file transfer applications.

Secure Shell is used to connect to servers, make changes, perform uploads and exit, either using tools or directly through the terminal. SSH keys can be employed to automate access to servers and often are used in scripts, backup systems and configuration management tools.

Designed to be convenient and work across organizational boundaries, SSH keys provide single sign-on (SSO) so that users can move between their accounts without typing a password each time.

While playing pivotal roles in identity management and access management, SSH does more than authenticate over an encrypted connection. All SSH traffic is encrypted. Whether users are transferring a file, browsing the web or running a command, their actions are private.

While it is possible to use SSH with an ordinary user ID and password as credentials, SSH relies more often on public key pairs to authenticate hosts to each other. Individual users must still employ their user ID and password -- or other authentication methods -- to connect to the remote host itself, but the local machine and the remote machine authenticate separately to each other. This is accomplished by generating a unique public key pair for each host in the communication. A single session requires two public key pairs: one public key pair to authenticate the remote machine to the local machine and a second public key pair to authenticate the local machine to the remote machine.

SECURE SHELL CAPABILITIES:

Functions that SSH enables include the following:

•secure remote access to SSH-enabled network systems or devices for users, as well as automated processes;

•secure and interactive file transfer sessions;

•automated and secured file transfers;

•secure issuance of commands on remote devices or systems; and

•secure management of network infrastructure components.

SSH can be used interactively to enable terminal sessions and should be used instead of the less secure Telnet program. SSH is also commonly used in scripts and other software to enable programs and systems to remotely and securely access data and other resources.

THE HISTORY OF SSH

The first version of SSH appeared in 1995 and was designed by Tatu Ylönen, who was, at the time, a researcher at Helsinki University of Technology and later started SSH Communications Security, a cybersecurity vendor based in Finland.

Over time, various flaws were found in SSH-1. That version is now considered to be deprecated and not safe to use.

SSH-2, the current version of Secure Shell protocols, was adopted as a Standards Track specification by the Internet Engineering Task Force (IETF) in 2006. SSH-2 is not compatible with SSH-1 and uses a Diffie-Hellman key exchange and a stronger integrity check that uses message authentication codes to improve security.

SSH clients and servers can use a number of encryption methods, the mostly widely used being Advanced Encryption Standard (AES) and Blowfish.

There are no known exploitable vulnerabilities in SSH-2, though information leaked by Edward Snowden in 2013 suggested the National Security Agency (NSA) may be able to decrypt some SSH traffic.

SECURE SHELL SECURITY ISSUES

Enterprises using SSH should consider finding ways to manage host keys stored on client systems. These keys can accumulate over time, especially for information technology (IT) staff that needs to be able to access remote hosts for management purposes.

Because the data stored in an SSH known_hosts file can be used to gain authenticated access to remote systems, organizations should be aware of the existence of these files and should have a standard process for retaining control over the files, even after a

system is taken out of commission, as the hard drives may have this data stored in plaintext.

Developers should be careful when incorporating SSH commands or functions in a script or other type of program. While it is possible to issue an SSH command that includes a user ID and password to authenticate the user of the local machine to an account on the remote host, doing so may expose the credentials to an attacker with access to the source code.

Shellshock, a security hole in the Bash command processor, can be executed over SSH but is a vulnerability in Bash, not in SSH.

The biggest threat to SSH is poor key management. Without the proper centralized creation, rotation and removal of SSH keys, organizations can lose control over who has access to which resources and when, particularly when SSH is used in automated application-to-application processes.

SSH VS. TELNET

Telnet was one of the first internet application protocols -- the other is FTP. It is used to initiate and maintain a terminal emulation session on a remote host.

SSH and Telnet are functionally similar, with the primary difference being that the SSH protocol uses public key cryptography to authenticate endpoints when setting up a terminal session, as well as for encrypting session commands and output.

While Telnet is primarily used for terminal emulation, SSH can be used to do terminal emulation -- similar to the rlogin command -- as well as for issuing commands remotely as with rsh, transferring files using SSH File Transfer Protocol (SFTP) and tunneling other applications.

SSH VS. SSL/TLS

The Transport Layer Security (TLS) protocol, which updates the Secure Sockets Layer (SSL) protocol, was designed to provide security for network transmissions at the transport layer. The SSH protocol also operates at or just above the transport layer, but there are important differences between the two protocols.

While both rely on public/private key pairs to authenticate hosts, only the server is authenticated with a key pair under TLS. SSH uses a separate key pair to authenticate each connection: one key pair for a connection from a local machine to a remote machine and a second key pair to authenticate the connection from the remote machine to the local machine.

Another difference between SSH and TLS is that TLS enables connections to be encrypted without authentication or authenticated without encryption. SSH encrypts and authenticates all connections.

SSH provides IT and information security (infosec) professionals with a secure mechanism to manage SSH clients remotely. Rather than requiring password authentication to initialize a connection between an SSH client and server, SSH authenticates the devices themselves. This enables IT staff to connect with remote systems and modify SSH configurations, including adding or removing host key pairs in the known_hosts file.

SSH IMPLEMENTATIONS

SSH is an open protocol. It has been implemented for most computing platforms. The open source OpenSSH implementation is the one most commonly found on Linux, Unix and other OSes based on Berkeley Software Distribution (BSD), including Apple's macOS.

OpenSSH was ported to run in Windows PowerShell starting in 2015. In 2018, optional OpenSSH support was added to Windows 10. While SSH is directly accessible by default in most Unix-like OSes, Microsoft's ported version of OpenSSH must be explicitly enabled in the Windows Settings app.

PuTTY is another open source implementation of SSH. While it currently is available for Windows, macOS and Unix/BSD, PuTTY was originally written to run on Windows. It has long been one of the top options for using SSH on a Windows system.

Most implementations of the SSH suite comprise three utilities:

1.slogin (secure login)

2.ssh

3.scp (secure copy)

These are secure versions of the earlier insecure Unix utilities: rlogin, rsh and rcp.

SSH uses public key cryptography to authenticate the remote computer and enables the remote computer to authenticate the user, if necessary.

There are currently dozens of SSH implementations available for various platforms and under a variety of open source and proprietary licenses.

SSH COMMANDS

While there are graphical implementations of SSH, the program is usually invoked at the command line or executed as part of a script. Running the ssh command on its own, with no arguments such as a destination host or user ID, returns a list of SSH command parameters and options.

The most basic form of SSH command is to invoke the program and the destination host name or Internet Protocol (IP) address:

```
ssh server.example.org
```

This will connect to the destination, server.example.org. The destination host will respond by prompting for a password for the user ID of the account under which the client is running. In other words, if the user ID in use is jsmith, then the remote host will ask for a password associated with the account jsmith on the remote host.

In many cases, the user ID for the remote host will be different, in which case the command should be issued with the remote host user ID, like this:

```
ssh remote_host_userID@server.example.org
```

SSH can also be used from the command line to issue a single command on the remote host and then exit -- for example:

```
ssh example.org ls
```

This command executes the Unix ls command, which lists all contents of the current directory on the remote host. While this example is trivial, it demonstrates that SSH can be used to execute more interesting commands on a remote host. For example, a command can be crafted that initializes a server instance that will give a remote machine access to a single file -- or other resource -- and then terminate the server after the file is accessed by the specified remote host.

In addition to the ssh executable, SSH has other executable commands used at the command line for additional functions, including the following:

•sshd initiates the SSH server, which waits for incoming SSH connection requests and enables authorized systems to connect to the local host.

•ssh-keygen is a program to create a new authentication key pair for SSH, which can be used to automate logins, to implement SSO and to authenticate hosts.

•ssh-copy-id is a program used to copy, install and configure an SSH key on a server to automate passwordless logins and SSO.

•ssh-agent is a helper program that tracks identity keys and their passphrases -- from which SSH derives an encryption key -- and enables the user to use the identity keys to log in to different servers without the need to reenter passwords or passphrases.

•ssh-add is used to add a key to the SSH authentication agent and is used with ssh-agent to implement SSO using SSH.

•scp is a program used for copying files from one computer to another and is an SSH-secured version of rcp.

•sftp is a program used to copy files from one computer to another and is an SSH-secured version of ftp, the original File Transfer Protocol. SFTP has become the preferred mechanism for file sharing over the internet, replacing both FTP and FTP/S (FTP Secure), which is a protocol for using FTP over an SSL/TLS tunnel.

WHAT IS SSH TUNNELING?

SSH tunneling, also known as SSH port forwarding, is a technique that enables a user to open a secure tunnel between a local host and a remote host.

SSH port forwarding redirects network traffic to a particular port/IP address so that a remote host is made directly accessible by applications on the local host. The destination may be on the remote SSH server, or that server may be configured to forward to yet another remote host.

SSH tunnels are powerful tools for IT administrators, as well as malicious actors, because they can transit an enterprise firewall undetected. As a result, there are tools available to prevent unauthorized use of SSH tunnels through a corporate firewall.

WIRELESS SECURITY

Like the system's security and data security, keeping a sound knowledge about different wireless security measures is also essential for security professionals. It is because different wireless security mechanisms have a different level of strength and capabilities.

There are automated wireless hacking tools available that have made cybercriminals more powerful. List of some of these tools are:

•AirCrack.

•AirSnort.

•Cain & Able.

•Wireshark.

•NetStumbler etc.

Different hacking techniques include remote accessing, shoulder surfing, wireless router's dashboard accessing, and brute-forcing attack that are used to penetrate wireless security. In this chapter, you will learn about the different security postures that exist in the wireless domain.

WHAT IS WIRELESS SECURITY?

Wireless security revolves around the concept of securing the wireless network from malicious attempts and unauthorized access.

The wireless security can be delivered through different ways such as:

1.Hardware-based: where routers and switches are fabricated with encryption measures protects all wireless communication. So, in this case, even if the data gets compromised

by the cybercriminal, they will not be able to decrypt the data or view the traffic's content.

2.Wireless setup of IDS and IPS: helps in detecting, alerting, and preventing wireless networks and sends an alarm to the network administrator in case of any security breach.

3.Wireless security algorithms: such as WEP, WPA, WPA2, and WPA3. These are discussed in the subsequent paragraphs.

WIREDEQUIVALENT PRIVACY (WEP)

Wired Equivalent Privacy (WEP) is the oldest security algorithm of 1999. It uses the initialization vector (IV) method. The first versions of the WEP algorithm were not predominantly strong enough, even when it got released. But the reason for this weak release was because of U.S. limits on exporting different cryptographic technologies, which led the manufacturing companies to restrict their devices to 64-bit encryption only. As the limitation was withdrawn, the 128 bit and 256 bit WEP encryption were developed and came into the wireless security market, though 128 became standard.

WI-FI PROTECTED ACCESS (WPA)

Wi-Fi Protected Access (WPA) was the next Wi-Fi Alliance's project that replaced the WEP standard's increasingly noticeable vulnerabilities. WPA was officially adopted in the year 2003, one year before the retirement of WEP. WPA's most common configuration is with WPA-PSK, which is abbreviated as Pre-Shared Key. WPA uses 256-bit, which was a considerable enhancement above the 64-bit as well as 128-bit keys.

WI-FIPROTECTED ACCESS II (WPA2)

Wi-Fi Protected Access II (WPA2) became official in the year 2006 after WPA got outdated. It uses the AES algorithms as a necessary encryption component as well as uses CCMP (Counter Cipher Mode - Block Chaining Message Authentication Protocol) by replacing TKIP.

WI-FIPROTECTED ACCESS 3 (WPA3)

Wi-Fi Protected Access 3 (WPA3) is the latest and the third iteration of this family developed under Wi-Fi Alliance. It has personal and enterprise security-support features and uses 384-bit Hashed Message Authentication Mode, 256-bit Galois / Counter Mode Protocol (GCMP-256) well as Broadcast/Multicast Integrity Protocol of 256-bit. WPA3 also provides perfect forward secrecy mechanism support.

MOBILEDEVICE SECURITY

WHAT IS MOBILE DEVICE SECURITY?

Mobile Device Security refers to the measures designed to protect sensitive information stored on and transmitted by laptops, smartphones, tablets, wearables, and other portable devices. At the root of mobile device security is the goal of keeping unauthorized users from accessing the enterprise network. It is one aspect of a complete enterprise security plan.

WHY IS MOBILE DEVICE SECURITY IMPORTANT?

With more than half of business PCs now mobile, portable devices present distinct challenges to network security, which must account for all of the locations and uses that employees require of the company network. Potential threats to devices include malicious mobile apps, phishing scams, data leakage, spyware, and unsecure Wi-Fi networks. On top of that, enterprises have to account for the possibility of an employee losing a mobile device or the device being stolen. To avoid a security breach, companies should take clear, preventative steps to reduce the risk.

WHAT ARE THE BENEFITS OF MOBILE DEVICE SECURITY?

Mobile device security, or mobile device management, provides the following:

•Regulatory compliance

•Security policy enforcement

•Support of "bring your own device" (BYOD)

•Remote control of device updates

•Application control

•Automated device registration

•Data backup

Above all, mobile device security protects an enterprise from unknown or malicious outsiders being able to access sensitive company data.

HOW DOES MOBILE DEVICE SECURITY WORK?

Securing mobile devices requires a multi-layered approach and investment in enterprise solutions. While there are key elements to mobile device security, each organization needs to find what best fits its network.

To get started, here are some mobile security best practices:

•ESTABILSH, SHARE, AND ENFORCE CLEAR POLICIES AND PROCESSES

Mobile device rules are only as effective as a company's ability to properly communicate those policies to employees. Mobile device security should include clear rules about:

1.What devices can be used

2.Allowed OS levels

3.What the company can and cannot access on a personal phone

4.Whether IT can remote wipe a device

5.Password requirements and frequency for updating passwords

•PASSWORD PROTECTION

One of the most basic ways to prevent unauthorized access to a mobile device is to create a strong password, and yet weak passwords are still a persistent problem that contributes to the majority of data hacks. Another common security problem is workers using the same password for their mobile device, email, and every work-related account. It is critical that employees create strong, unique passwords (of at least eight characters) and create different passwords for different accounts.

•LEVERAGE BIOMETRICS

Instead of relying on traditional methods of mobile access security, such as passwords, some companies are looking to biometrics as a safer alternative. Biometric authentication is when a computer uses measurable biological characteristics, such as face, fingerprint, voice, or iris recognition for identification and access. Multiple biometric authentication methods are now available on smartphones and are easy for workers to set up and use.

•AVOID PUBLIC WI-FI

A mobile device is only as secure as the network through which it transmits data. Companies need to educate employees about the dangers of using public Wi-Fi networks, which are vulnerable to attacks from hackers who can easily breach a device, access the network, and steal data. The best defense is to encourage smart user behavior and prohibit the use of open Wi-Fi networks, no matter the convenience.

•BEWARE OF APPS

Malicious apps are some of the fastest growing threats to mobile devices. When an employee unknowingly downloads one, either for work or personal reasons, it provides unauthorized access to the company's network and data. To combat this rising threat, companies have two options: instruct employees about the dangers of downloading unapproved apps, or ban employees from downloading certain apps on their phones altogether.

•MOBILE DEVICE ENCRYPTION:

Most mobile devices are bundled with a built-in encryption feature. Users need to locate this feature on their device and enter a password to encrypt their device. With this method, data is converted into a code that can only be accessed by authorized users. This is important in case of theft, and it prevents unauthorized access.

WHAT ARE THE DIFFERENT TYPES OF MOBILE DEVICE SECURITY?

There are many aspects to a complete security plan. Common elements of a mobile security solution include the following:

•ENTERPRISE MOBILE MANAGEMENT PLATFORM: In addition to setting up internal device policies that protect against unauthorized access, it's equally important to have an Enterprise Mobile Management (EMM) platform that enables IT to gather real-time insights to catch potential threats.

•EMAIL SECURITY: Email is the most popular way for hackers to spread ransomware and other malware. To combat such attacks, it's critical for businesses to be armed with advanced email security that can detect, block, and address threats faster; prevent any data loss; and protect important information in transit with end-to-end encryption.

•ENDPOINT PROTECTION: This approach protects enterprise networks that are remotely accessed by mobile devices. Endpoint security protects companies by ensuring that portable devices follow security standards and by quickly alerting security teams of detected threats before they can do damage. Endpoint protection also allows IT administrators to monitor operation functions and data backup strategies.

•VPN: A virtual private network, or VPN, extends a private network across a public network. This enables users to send and receive data across shared or public networks as if their computing devices were directly connected to the private network. VPNs' encryption technology allows remote users and branch offices to securely access corporate applications and resources.

•SECURE WEB GATEWAY: A secure web gateway protects against online security threats by enforcing company security policies and defending against phishing and malware in real-time. This is especially important for cloud security as this type of protection can identify an attack on one location and immediately stop it at other branches.

•CLOUD ACCESS SECURITY BROKER: A cloud access security broker (CASB) is a tool that sits between cloud service consumers and cloud service providers to enforce security, compliance, and governance policies for cloud applications. CASBs help organizations extend the security controls of their on-premises infrastructure to the cloud.

HOW DOES MOBILE DEVICE SECURITY COMPLEMENT EXISTING APPLICATION SECURITY AND NETWORK SECURITY EFFORTS?

In addition to monitoring and protecting against malicious threats to a company's data, mobile device security—when paired with an EMM platform and other network and application security solutions—enables an IT department to remotely manage users and their devices. This capability provides security for all mobile devices connected to a network, while giving IT the option to remotely disable unauthorized users and applications. An EMM also allows IT to remotely wipe company data from a lost or stolen device and to control device updates. All of these measures enhance security significantly.

Making mobile devices secure is not a simple task, but it should be a high priority for any enterprise. To combat the growing threat of cyber-attacks, companies must continually audit their mobile security solutions and consider new security measures as they become available.

IEEE 802.11 WIRELESS LAN

Wireless LAN is one of the fastest-growing technologies. IEEE 802.11 refers to the set of standards that define communication for wireless LANs (wireless local area networks, or WLANs). The technology behind 802.11 is branded to consumers as Wi-Fi.

Wireless LAN can be found on college campuses, in office buildings, in hospitals, stock exchanges and in many public areas. It has become popular due to the ease of installation and location freedom with the gaining popularity of laptops.

Wi-Fi is now one of the major forms of communication for many devices, and with home automation increasing, even more, devices are using it. Home Wi-Fi is a big area of usage of technology, with most homes that use broadband connections to the Internet using WiFi access as a key means of communication.

The core of any Wi-Fi system is known as the Access Point, AP. The Wi-Fi access point is essentially the base station that communicates with the Wi-Fi enabled devices - data can then be routed onto a local area network, normally via Ethernet and typically links onto the Internet.

ADVANTAGES

There are various advantages of WLAN, which are as follows −

FAST INSTALLATION AND SIMPLICITY

Installing a wireless LAN system can be fast and easy and can eliminate the need to pull cables through walls, floor, and ceilings.

INCREASED PRODUCTIVITY FOR THE MOBILE EMPLOYEE

The mobile user whose primary computer is a portable computer can change location and always remain connected to the network. This enables the mobile user to travel to various places, let it be meeting rooms, hallways, lobbies, cafeterias, classrooms, and so forth.

REDUCED COST

The initial investment required for wireless LAN hardware is higher than the cost of wired LAN hardware. However, the overall installation expenses and life cycle costs are significantly lower. Long-term cost benefits are greatest in dynamic environments, requiring frequent moves and changes.

MOBILITY AND COLLABORATION

It can stay connected while moving throughout your worksite. Access up-to-the-minute communications and all documents and apps on the network, anywhere, anytime.

ACCESSIBILITY

It can provide network access across your organization, even in areas that have been challenging to reach with the wired network, so your entire team can stay in touch.

EXPANDABILITY

It is used to grow your network efficiently, adding new users and locations without needing to run cables and wires.

GUEST ACCESS

It can offer secure network access to guest users, including customers and business partners while keeping your network resources protected.

IEEE 802.11I WIRELESS LAN SECURITY

There are two characteristics of a wired LAN that are not inherent in a wireless LAN.

1.In order to transmit over a wired LAN, a station must be physically connected to the LAN. On the other hand, with a wireless LAN, any station within radio range of the other devices on the LAN can transmit. In a sense, there is a form of authentication with a wired LAN in that it requires some positive and presumably observable action to connect a station to a wired LAN.

2.Similarly, in order to receive a transmission from a station that is part of a wired LAN, the receiving station also must be attached to the wired LAN. On the other hand, with a wireless LAN, any station within radio range can receive. Thus, a wired LAN provides a degree of privacy, limiting reception of data to stations connected to the LAN.

These differences between wired and wireless LANs suggest the increased need for robust security services and mechanisms for wireless LANs. The originalspecification included a set of security features for privacy and authentication that were quite weak. For privacy, 802.11 defined the Wired Equivalent Privacy (WEP) algorithm. The privacy portion of the 802.11 standard contained major weak- nesses. Subsequent to the development of WEP, the 802.11i task group has developed a set of capabilities to address the WLAN security issues. In order to accelerate the introduction of strong security into WLANs, the Wi-Fi Alliance promulgated Wi-Fi Protected Access (WPA) as a Wi-Fi standard. WPA is a set of security mechanisms that eliminates most 802.11 security issues and was based on the current state of the 802.11i 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 WPA2 program.

IEEE 802.11I SERVICES

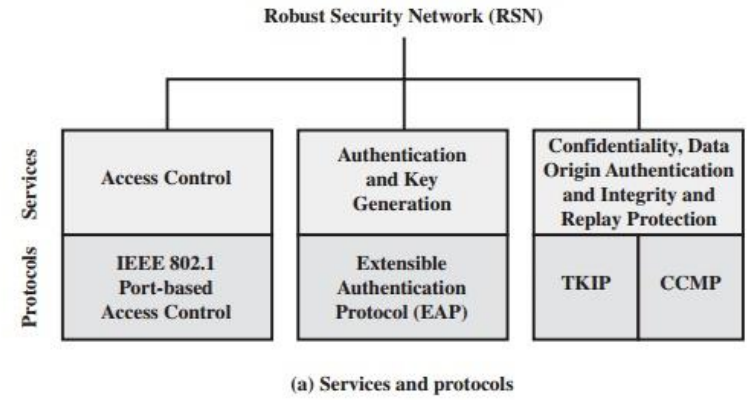
The 802.11i RSN security specification defines the following services.

• **AUTHENTICATION:** A protocol is used to define an exchange between a user and an AS that provides mutual authentication and generates temporary keys to be used between the client and the AP over the wireless link.

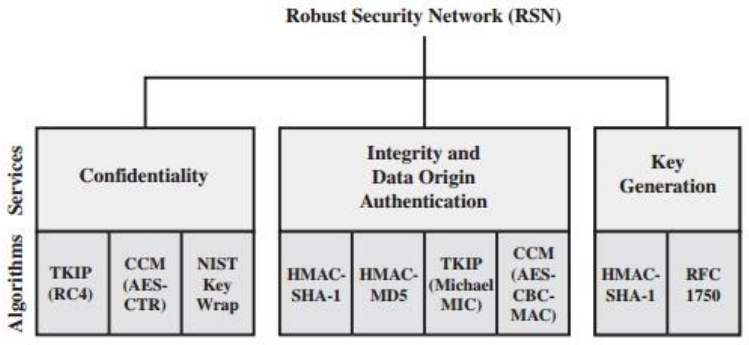
•**ACCESS CONTROL:** This function enforces the use of the authentication function, routes the messages properly, and facilitates key exchange. It can work with a variety of authentication protocols.

•**PRIVACY WITH MESSAGE INTEGRITY:** MAC-level data (e.g., an LLC PDU) are encrypted along with a message integrity code that ensures that the data have not been altered.

Figure 17.4a indicates the security protocols used to support these services, while Figure 17.4b lists the cryptographic algorithms used for these services.



(a) Services and protocols



(b) Cryptographic algorithms

CBC-MAC = Cipher Block Block Chaining Message Authentication Code (MAC)
CCM = Counter Mode with Cipher Block Chaining Message Authentication Code
CCMP = Counter Mode with Cipher Block Chaining MAC Protocol
TKIP = Temporal Key Integrity Protocol

Figure 17.4 Elements of IEEE 802.11i

IEEE 802.11I PHASEES OF OPERATION

The operation of an IEEE 802.11i RSN can be broken down into five distinct phases of operation. The exact nature of the phases will depend on the configuration and the end points of the communication. Possibilities include (see Figure 17.3):

1. Two wireless stations in the same BSS communicating via the access point (AP) for that BSS.

2. Two wireless stations (STAs) in the same ad hoc IBSS communicating directly with each other.

3. Two wireless stations in different BSSs communicating via their respective APs across a distribution system.

4. A wireless station communicating with an end station on a wired network via its AP and the distribution system.

IEEE 802.11i security is concerned only with secure communication between the STA and its AP. In case 1 in the preceding list, secure communication is assured if each STA establishes secure communications with the AP. Case 2 is similar, with the AP functionality residing in the STA. For case 3, security is not provided across the distribution system at the level of IEEE 802.11, but only within each BSS. End- to-end security (if required) must be provided at a higher layer. Similarly, in case 4, security is only provided between the STA and its AP.

With these considerations in mind, Figure 17.5 depicts the five phases of oper- ation for an RSN and maps them to the network components involved. One new component is the authentication server (AS). The rectangles indicate the exchange of sequences of MPDUs. The five phases are defined as follows.

• **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 mecha- nism 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.

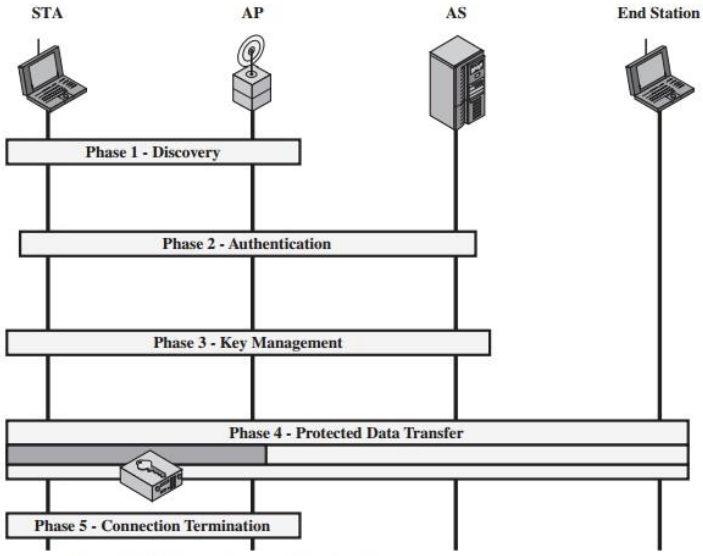


Figure 17.5 IEEE 802.11i Phases of Operation

• **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.

DISCOVERY PHASE

We now look in more detail at the RSN phases of operation, beginning with the discovery phase, which is illustrated in the upper portion of Figure 17.6. The purpose of this phase is for an STA and an AP to recognize each other, agree on a set of secu- rity capabilities, and establish an association for future communication using those security capabilities.

SECURITY CAPABILITIES During this phase, the STA and AP decide on specific techniques in the following areas:

•Confidentiality and MPDU integrity protocols for protecting unicast traffic (traffic only between this STA and AP)

• Authentication method

• Cryptography key management approach

Confidentiality and integrity protocols for protecting multicast/broadcast traf- fic are dictated by the AP, since all STAs in a multicast group must use the same pro- tocols and ciphers. The specification of a protocol, along with the chosen key length (if variable) is known as a cipher suite. The options for the confidentiality and integrity cipher suite are

• WEP, with either a 40-bit or 104-bit key, which allows backward compatibility with older IEEE 802.11 implementations

• TKIP

• CCMP

• Vendor-specific methods

The other negotiable suite is the authentication and key management (AKM) suite, which defines (1) the means by which the AP and STA perform mutual authentication and (2) the means for deriving a root key from which other keys may be generated. The possible AKM suites are

• IEEE 802.1X

• Pre-shared key (no explicit authentication takes place and mutual authentica- tion is implied if the STA and AP share a unique secret key)

• Vendor-specific methods

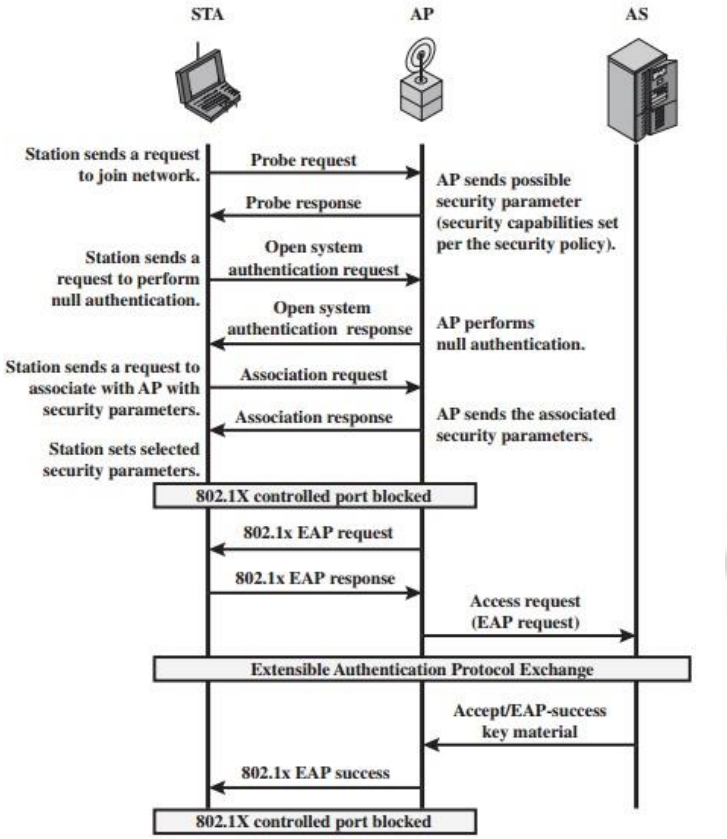


Figure 17.6 IEEE 802.11i Phases of Operation: Capability Discovery, Authentication, and Association

MPDU EXCHANGE The discovery phase consists of three exchanges.

• **NETWORK AND SECURITY CAPABILITY DISCOVERY:** During this exchange, STAs dis- cover the existence of a network with which to communicate. The AP either periodically broadcasts its security capabilities (not shown in figure), indicated by RSN IE (Robust Security Network Information Element), in a specific channel through the Beacon frame; or responds to a station's Probe Request through a Probe Response frame. A wireless station may discover available access points and corresponding security capabilities by either passively monitoring the Beacon frames or actively probing every channel.

OPEN SECURITY AUTHENTICATION: The purpose of this frame sequence, which provides no security, is simply to maintain backward compatibility with the IEEE 802.11

state machine, as implemented in existing IEEE 802.11 hard- ware. In essence, the two devices (STA and AP) simply exchange identifiers.

ASSOCIATION: The purpose of this stage is to agree on a set of security capabilities to be used. The STA then sends an Association Request frame to the AP. In this frame, the STA specifies one set of matching capabilities (one authentication and key management suite, one pairwise cipher suite, and one group- key cipher suite) from among those advertised by the AP. If there is no match in capabilities between the AP and the STA, the AP refuses the Association Request. The STA blocks it too, in case it has associated with a rogue AP or someone is inserting frames illicitly on its channel. As shown in Figure 17.6, the IEEE 802.1X controlled ports are blocked, and no user traffic goes beyond the AP. The concept of blocked ports is explained subsequently.

AUTHENTICATION PHASE

As was mentioned, the authentication phase enables mutual authentication between an STA and an authentication server (AS) located in the DS. Authentication is designed to allow only authorized stations to use the network and to provide the STA with assurance that it is communicating with a legitimate network.

IEEE 802.1X ACCESS CONTROL APPROACH IEEE 802.11i makes use of another standard that was designed to provide access control functions for LANs. The standard is IEEE 802.1X, Port-Based Network Access Control. The authentication protocol that is used, the Extensible Authentication Protocol (EAP), is defined in the IEEE 802.1X standard. IEEE 802.1X uses the terms supplicant, authenticator, and authentication server (AS). In the context of an 802.11 WLAN, the first two terms correspond to the wireless station and the AP. The AS is typically a separate device on the wired side of the network (i.e., accessible over the DS) but could also reside directly on the authenticator.

Before a supplicant is authenticated by the AS using an authentication proto- col, the authenticator only passes control or authentication messages between the supplicant and the AS; the 802.1X control channel is unblocked, but the 802.11 data channel is blocked. Once a supplicant is authenticated and keys are provided, the authenticator can forward data from the supplicant, subject to predefined access control limitations for the supplicant to the network. Under these circumstances, the data channel is unblocked.

As indicated in Figure 17.7, 802.1X uses the concepts of controlled and uncontrolled ports. Ports are logical entities defined within the authenticator and refer to physical network connections. For a WLAN, the authenticator (the AP) may have only two physical ports: one connecting to the DS and one for wireless communication within its BSS. Each logical port is mapped to one of these two physical ports. An uncontrolled port allows the exchange of PDUs between the supplicant and the other AS, regardless of

the authentication state of the supplicant. A controlled port allows the exchange of PDUs between a supplicant and other systems on the LAN only if the current state of the supplicant authorizes such an exchange.

•The 802.1X framework, with an upper-layer authentication protocol, fits nicely with a BSS architecture that includes a number of wireless stations and an AP.

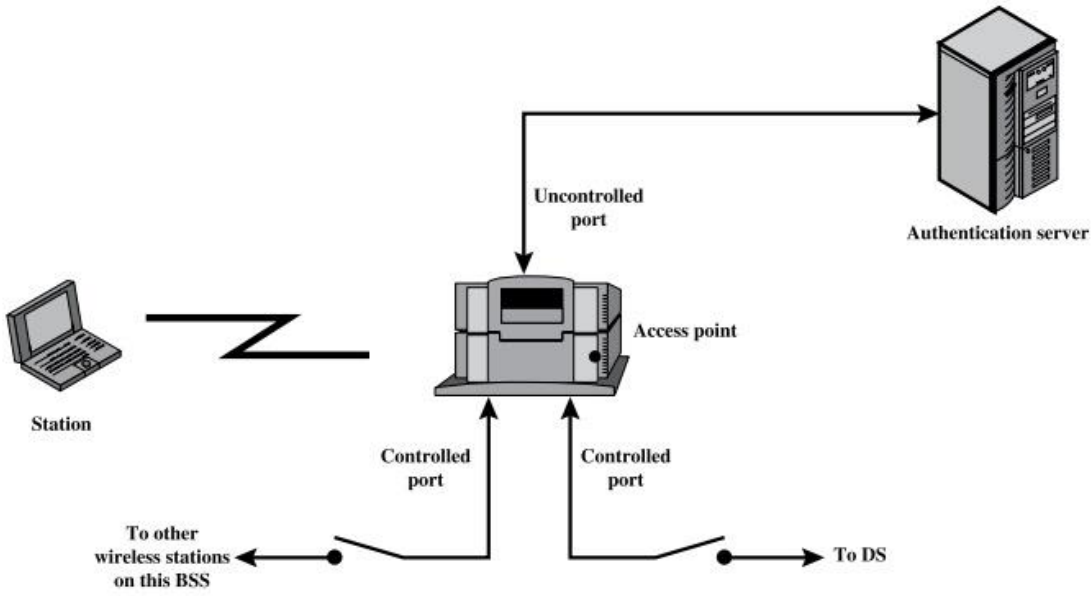


Figure 17.7 802.1X Access Control

However, for an IBSS, there is no AP. For an IBSS, 802.11i provides a more complex solution that, in essence, involves pairwise authentication between stations on the IBSS.

MPDU EXCHANGE The lower part of Figure 17.6 shows the MPDU exchange dictated by IEEE 802.11 for the authentication phase. We can think of authentication phase as consisting of the following three phases.

•**CONNECT TO AS:** The STA sends a request to its AP (the one with which it has an association) for connection to the AS. The AP acknowledges this request and sends an access request to the AS.

• **EAP EXCHANGE:** This exchange authenticates the STA and AS to each other. A number of alternative exchanges are possible, as explained subsequently.

• **SECURE KEY DELVERY:** Once authentication is established, the AS generates a master session key (MSK), also known as the Authentication, Authorization, and Accounting (AAA) key and sends it to the STA. As explained subsequently, all the cryptographic keys needed by the STA for secure communication with its AP are generated from this MSK. IEEE 802.11i does not prescribe a method for secure delivery

of the MSK but relies on EAP for this. Whatever method is used, it involves the transmis- sion of an MPDU containing an encrypted MSK from the AS, via the AP, to the AS.

EAP EXCHANGE As mentioned, there are a number of possible EAP exchanges that can be used during the authentication phase. Typically, the message flow between STA and AP employs the EAP over LAN (EAPOL) protocol, and the message flow between the AP and AS uses the Remote Authentication Dial In User Service (RADIUS) protocol, although other options are available for both STA-to- AP and AP-to-AS exchanges. [FRAN07] provides the following summary of the authentication exchange using EAPOL and RADIUS.

1. The EAP exchange begins with the AP issuing an EAP-Request/Identity frame to the STA.

2. The STA replies with an EAP-Response/Identity frame, which the AP receives over the uncontrolled port. The packet is then encapsulated in RADIUS over EAP and passed on to the RADIUS server as a RADIUS-Access-Request packet.

3. The AAA server replies with a RADIUS-Access-Challenge packet, which is passed on to the STA as an EAP-Request. This request is of the appropriate authentication type and contains relevant challenge information.

4. The STA formulates an EAP-Response message and sends it to the AS. The response is translated by the AP into a Radius-Access-Request with the response to the challenge as a data field. Steps 3 and 4 may be repeated multi- ple times, depending on the EAP method in use. For TLS tunneling methods, it is common for authentication to require 10 to 20 round trips.

5. The AAA server grants access with a Radius-Access-Accept packet. The AP issues an EAP-Success frame. (Some protocols require confirmation of the EAP success inside the TLS tunnel for authenticity validation.) The controlled port is authorized, and the user may begin to access the network.

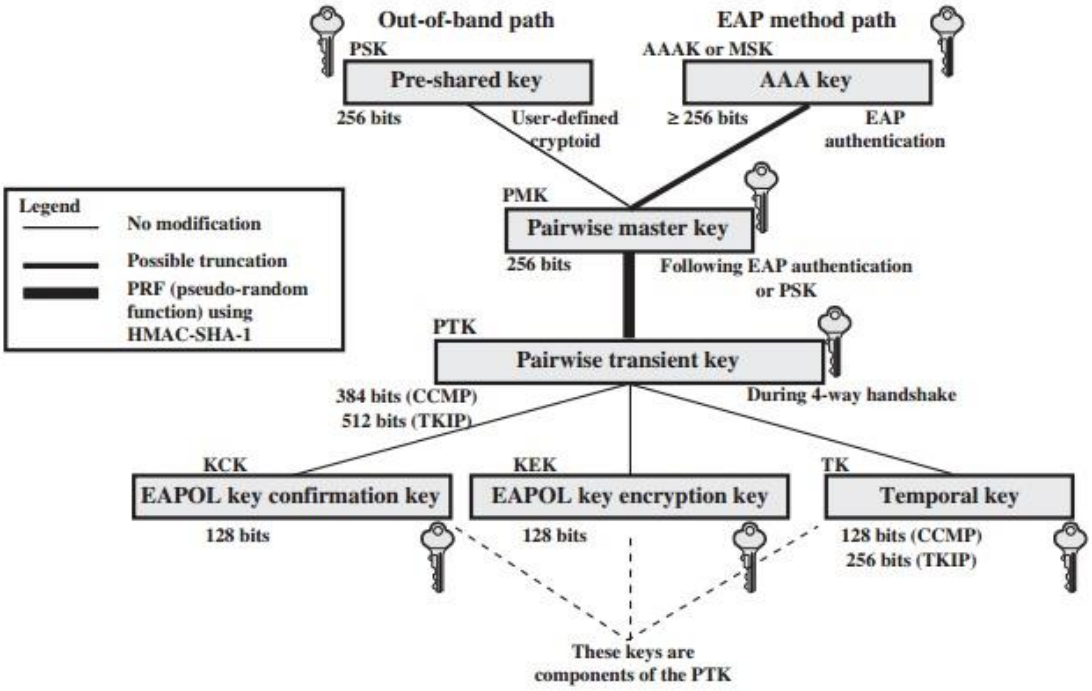
Note from Figure 17.6 that the AP controlled port is still blocked to general user traffic. Although the authentication is successful, the ports remain blocked until the temporal keys are installed in the STA and AP, which occurs during the 4-Way Handshake.

KEY MANAGEMENT PHASE

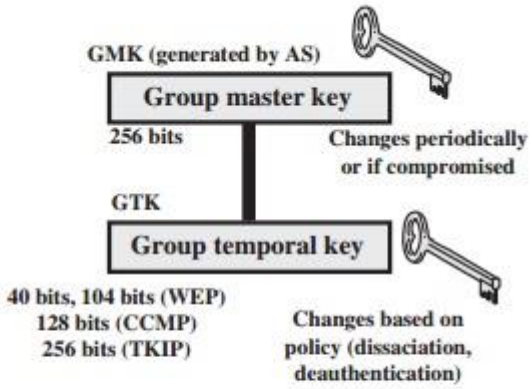
During the key management phase, a variety of cryptographic keys are generated and distributed to STAs. There are two types of keys: pairwise keys used for communication between an STA and an AP and group keys used for multicast communication. Figure 17.8, based on [FRAN07], shows the two key hierarchies, and Table 17.3 defines the individual keys.

PAIRWISE KEYS Pairwise keys are used for communication between a pair of devices, typically between an STA and an AP. These keys form a hierarchy beginning with a master key from which other keys are derived dynamically and used for a limited period of time.

At the top level of the hierarchy are two possibilities. A pre-shared key (PSK) is a secret key shared by the AP and a STA and installed in some fashion outside the scope of IEEE 802.11i. The other alternative is the master session key (MSK), also known as the AAAK, which is generated using the IEEE 802.1X protocol dur- ing the authentication phase, as described previously. The actual method of key generation depends on the details of the authentication protocol used. In either



(a) Pairwise key hierarchy



(b) Group key hierarchy

Figure 17.8 IEEE 802.11i Key Hierarchies

case (PSK or MSK), there is a unique key shared by the AP with each STA with which it communicates. All the other keys derived from this master key are also unique between an AP and an STA. Thus, each STA, at any time, has one set of keys, as depicted in the hierarchy of Figure 17.8a, while the AP has one set of such keys for each of its STAs.

The pairwise master key (PMK) is derived from the master key. If a PSK is used, then the PSK is used as the PMK; if a MSK is used, then the PMK is derived from the MSK by truncation (if necessary). By the end of the authentication phase, marked by the 802.1x EAP Success message (Figure 17.6), both the AP and the STA have a copy of their shared PMK.

Table 17.3 IEEE 802.11i Keys for Data Confidentiality and Integrity Protocols

Abbrev-iation	Name	Description / Purpose	Size (bits)	Type
AAA Key	Authentication, Accounting, and Authorization Key	Used to derive the PMK. Used with the IEEE 802.1X authentication and key management approach. Same as MMSK.	≥ 256	Key generation key, root key
PSK	Pre-shared Key	Becomes the PMK in pre-shared key environments.	256	Key generation key, root key
PMK	Pairwise Master Key	Used with other inputs to derive the PTK.	256	Key generation key
GMK	Group Master Key	Used with other inputs to derive the GTK.	128	Key generation key
PTK	Pair-wise Transient Key	Derived from the PMK. Comprises the EAPOL-KCK, EAPOL-KEK, and TK and (for TKIP) the MIC key.	512 (TKIP) 384 (CCMP)	Composite key
TK	Temporal Key	Used with TKIP or CCMP to provide confidentiality and integrity protection for unicast user traffic.	256 (TKIP) 128 (CCMP)	Traffic key
GTK	Group Temporal Key	Derived from the GMK. Used to provide confidentiality and integrity protection for multicast/broadcast user traffic.	256 (TKIP) 128 (CCMP) 40, 104 (WEP)	Traffic key
MIC Key	Message Integrity Code Key	Used by TKIP's Michael MIC to provide integrity protection of messages.	64	Message integrity key
EAPOL-KCK	EAPOL-Key Confirmation Key	Used to provide integrity protection for key material distributed during the 4-Way Handshake.	128	Message integrity key
EAPOL-KEK	EAPOL-Key Encryption Key	Used to ensure the confidentiality of the GTK and other key material in the 4-Way Handshake.	128	Traffic key / key encryption key
WEP Key	Wired Equivalent Privacy Key	Used with WEP.	40, 104	Traffic key

The PMK is used to generate the pairwise transient key (PTK), which in fact consists of three keys to be used for communication between an STA and AP after they have been mutually authenticated. To derive the PTK, the HMAC-SHA-1 function is applied to the PMK, the MAC addresses of the STA and AP, and nonces generated when needed. Using the STA and AP addresses in the generation of the PTK provides protection against session hijacking and impersonation; using nonces provides additional random keying material.

The three parts of the PTK are as follows.

• **EAP OVER LAN (EAPOL) KEY CONFIRAMATION KEY (EAPOL-KCK):** Supports the integrity and data origin authenticity of STA-to-AP control frames during operational setup of an RSN. It also performs an access control function: proof-of-possession of the PMK. An entity that possesses the PMK is autho- rized to use the link.

• **EAPOL KEY ENCRYPTION KEY (EAPOL-KEK):** Protects the confidentiality of keys and other data during some RSN association procedures.

• **TEMPORAL KEY (TK):** Provides the actual protection for user traffic.

GROUP KEYS Group keys are used for multicast communication in which one STA sends MPDU's to multiple STAs. At the top level of the group key hierarchy is the group master key (GMK). The GMK is a key-generating key used with other inputs to derive the group temporal key (GTK). Unlike the PTK, which is generated using material from both AP and STA, the GTK is generated by the AP and transmitted to its associated STAs. Exactly how this GTK is generated is undefined. IEEE 802.11i, however, requires that its value is computationally indistinguishable from random. The GTK is distributed securely using the pairwise keys that are already established. The GTK is changed every time a device leaves the network.

PAIRWISE KEY DISTRIBUTION The upper part of Figure 17.9 shows the MPDU exchange for distributing pairwise keys. This exchange is known as the 4-way handshake. The STA and SP 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. The four parts of the exchange are as follows.

• **AP: STA:** Message includes the MAC address of the AP and a nonce (Anonce)

• **STA: AP:** The STA generates its own nonce (Snonce) and uses both nonces and both MAC addresses, plus the PMK, to generate a PTK. The STA then sends a message containing its MAC address and Snonce, enabling the AP to generate the same PTK. This message includes a message integrity code (MIC)2 using HMAC-MD5 or HMAC-SHA-1-128. The key used with the MIC is KCK.

• **AP: STA:** The AP is now able to generate the PTK. The AP then sends a message to the STA, containing the same information as in the first message, but this time including a MIC.

• **STA: AP:** This is merely an acknowledgment message, again protected by a MIC.

GROUP KEY DISTRIBUTION For group key distribution, the AP generates a GTK and distributes it to each STA in a multicast group. The two-message exchange with each STA consists of the following:

AP: STA: This message includes the GTK, encrypted either with RC4 or with AES. The key used for encryption is KEK. A MIC value is appended.

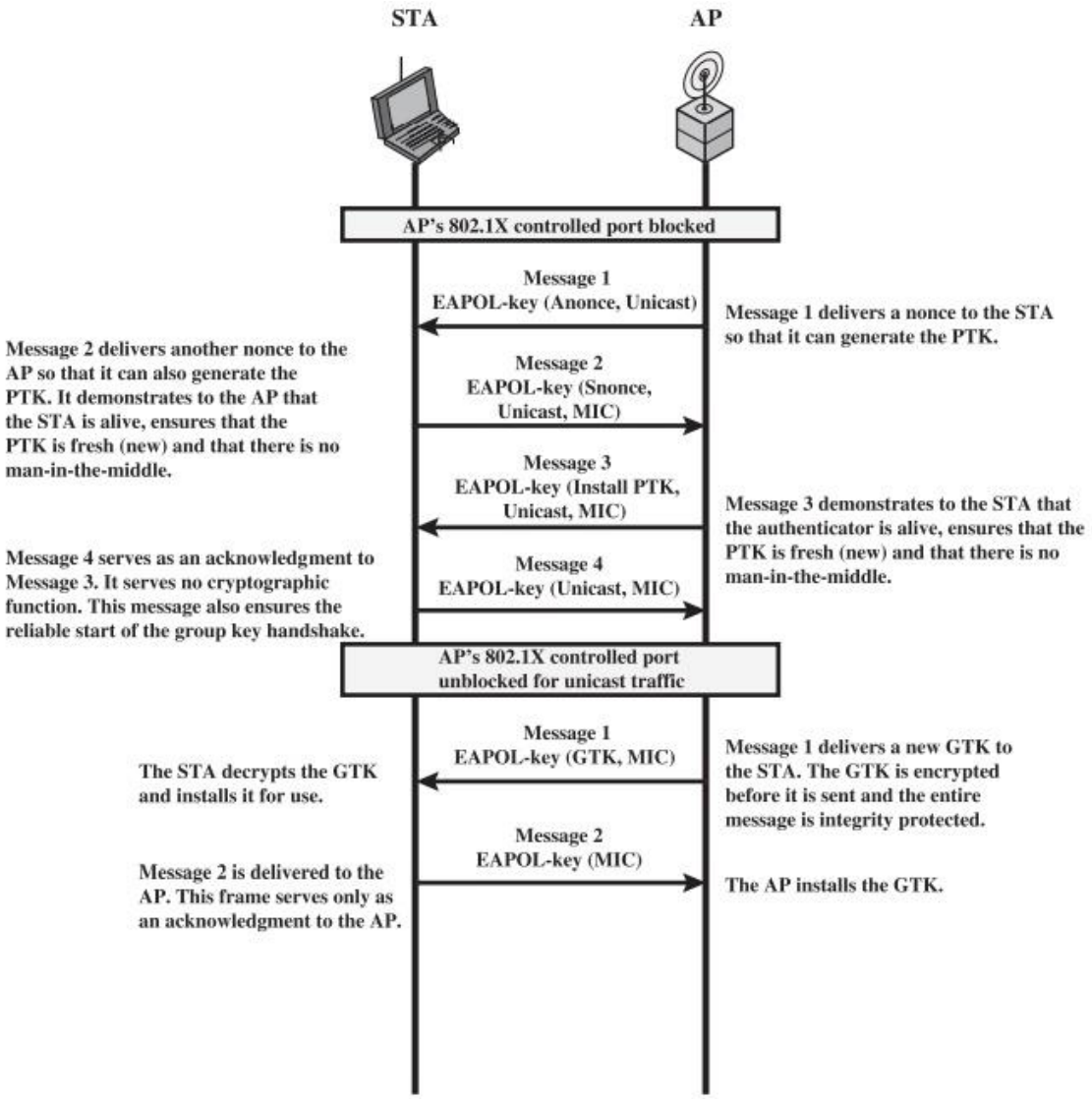


Figure 17.9 IEEE 802.11i Phases of Operation: Four-Way Handshake and Group Key Handshake

• **STA: AP:** The STA acknowledges receipt of the GTK. This message includes a MIC value.

PROTECTED DATA TRANSFER PHASE

IEEE 802.11i defines two schemes for protecting data transmitted in 802.11 MPDUs: the Temporal Key Integrity Protocol (TKIP), and the Counter Mode-CBC MAC Protocol (CCMP).

TKIP TKIP is designed to require only software changes to devices that are implemented with the older wireless LAN security approach called Wired Equivalent Privacy (WEP). TKIP provides two services:

• **MESSAGEINTEGRITY:** TKIP adds a message integrity code (MIC) to the 802.11 MAC frame after the data field. The MIC is generated by an algorithm, called Michael, that computes a 64-bit value using as input the source and destination MAC address values and the Data field, plus key material.

• **DATA CONFIDENTIALITY:** Data confidentiality is provided by encrypting the MPDU plus MIC value using RC4.

The 256-bit TK (Figure 17.8) is employed as follows. Two 64-bit keys are used with the Michael message digest algorithm to produce a message integrity code. One key is used to protect STA-to-AP messages, and the other key is used to protect AP-to-STA messages. The remaining 128 bits are truncated to generate the RC4 key used to encrypt the transmitted data.

For additional protection, a monotonically increasing TKIP sequence counter (TSC) is assigned to each frame. The TSC serves two purposes. First, the TSC is included with each MPDU and is protected by the MIC to protect against replay attacks. Second, the TSC is combined with the session TK to produce a dynamic encryption key that changes with each transmitted MPDU, thus making cryptanalysis more difficult.

CCMP CCMP is intended for newer IEEE 802.11 devices that are equipped with the hardware to support this scheme. As with TKIP, CCMP provides two services:

•**MESSAGE INTEGRITY:** CCMP uses the cipher-block-chaining message authentication code (CBC-MAC), described in Chapter 12.

• **DATA CONFIDENTIALITY:** CCMP uses the CTR block cipher mode of operation with AES for encryption. CTR is described in Chapter 6.

The same 128-bit AES key is used for both integrity and confidentiality. The scheme uses a 48-bit packet number to construct a nonce to prevent replay attacks.

THE IEEE 802.11I PSEUDORANDOM FUNCTION

At a number of places in the IEEE 802.11i scheme, a pseudorandom function (PRF) is used. For example, it is used to generate nonces, to expand pairwise keys, and to generate the GTK. Best security practice dictates that different pseudorandom number streams be used for these different purposes. However, for implementation efficiency, we would like to rely on a single pseudorandom number generator function.

The PRF is built on the use of HMAC-SHA-1 to generate a pseudorandom bit stream. Recall that HMAC-SHA-1 takes a message (block of data) and a key of length at least

160 bits and produces a 160-bit hash value. SHA-1 has the property that the change of a single bit of the input produces a new hash value with no appar- ent connection to the preceding hash value. This property is the basis for pseudorandom number generation.

The IEEE 802.11i PRF takes four parameters as input and produces the desired number of random bits. The function is of the form PRF(K, A, B, Len), where

K = a secret key

A= a text string specific to the application (e.g., nonce generation or pair- wise key expansion)

B = some data specific to each case

Len = desired number of pseudorandom bits

For example, for the pairwise transient key for CCMP:

PTK = PRF(PMK, "Pairwise key expansion", min(AP– Addr, STA–Addr)|| max(AP–Addr, STA–Addr) | min (Anonce, Snonce) | max(Anonce, Snonce), 384)

So, in this case, the parameters are

K = PMK

A= the text string "Pairwise key expansion"

B = a sequence of bytes formed by concatenating the two MAC addresses and the two nonces

Len = 384 bits

Similarly, a nonce is generated by

Nonce = PRF (Random Number, "Init Counter", MAC | Time, 256)

where Time is a measure of the network time known to the nonce generator.

The group temporal key is generated by

GTK = PRF (GMK, "Group key expansion", MAC | Gnonce, 256

Figure 17.10 illustrates the function PRF(K, A, B, Len). The parameter K serves as the key input to HMAC. The message input consists of four items concatenated together: the parameter A, a byte with value 0, the parameter B, and a counter i. The counter is initialized to 0. The HMAC algorithm is run once, producing a 160-bit hash value. If more bits are required, HMAC is run again with the same inputs, except that i is

incremented each time until the necessary number of bits is generated. We can express the logic as

PRF(K, A, B, Len)

R ‹- null string

for i ‹- 0 to ((Len + 159)/160 – 1) do

R ‹- R | HMAC–SHA–1(K, A ||0 |B |i)

Return Truncate–to–Len(R, Len)

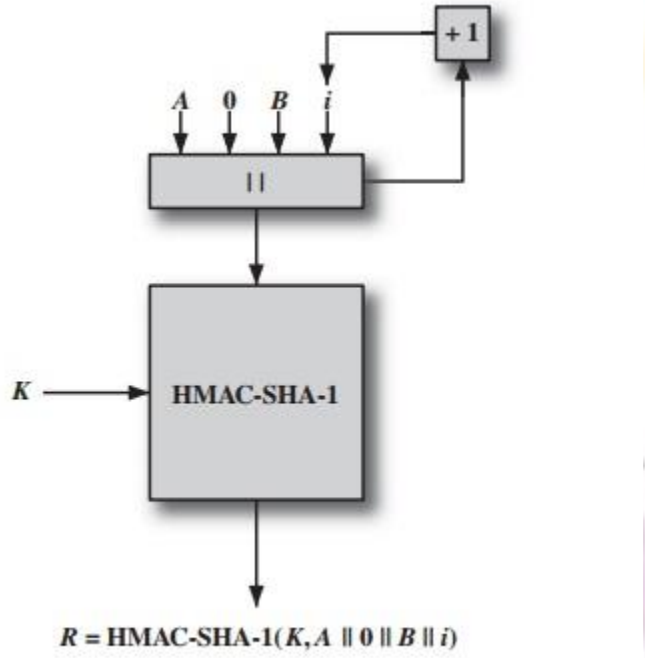


Figure 17.10 IEEE 802.11i Pseudorandom Function

UNIT-5

PRETTY GOOD PRIVACY

In virtually all distributed environments, electronic mail is the most heavilyusednetwork-

basedapplication.Butcurrentemailservicesareroughlylike"postcards",anyonew howantscouldpickitupandhavealookasit'sintransitorsitting in the recipients mailbox. PGP provides a confidentiality and authenticationservice that can be used for electronic mail and file storage applications. With theexplosively growing reliance on electronic mail for every conceivable purpose, theregrows a demand for authenticationand confidentiality services. The Pretty GoodPrivacy(PGP)secureemailprogram,isaremarkablephenomenon,hasgrown explosivelyandisnowwidelyused.Largelytheeffortofasingleperson,PhilZimme rmann, who selected the best available crypto algorithms to use & integratedthem into a single program, PGP provides a confidentiality and authentication servicethat can be used for electronic mail and file storage applications. It is independent ofgovernmentorganizationsandrunsonawiderangeofsystems,inbothfree&com mercialversions.Therare**five**importantservicesinPGP

Authentication(Sign/Verify)

Confidentiality(Encryption/Decryption)

Compression

Email Compatibility

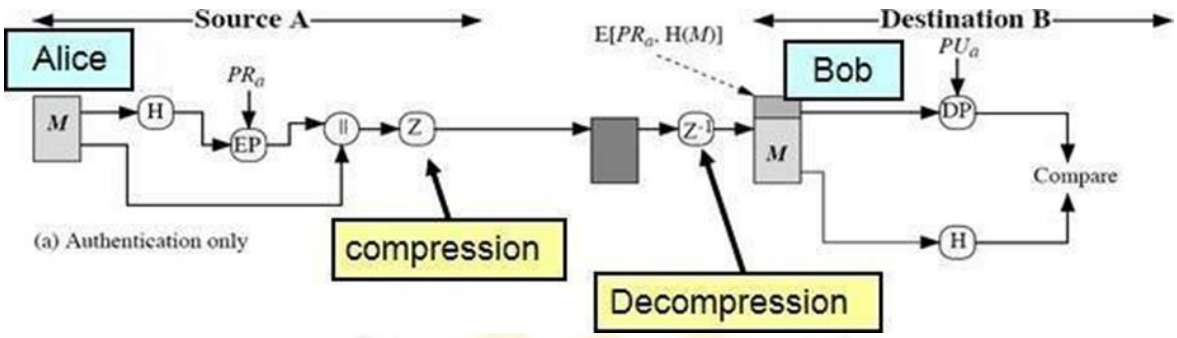
Segmentation and Reassembly

The last three are **transparent to** the user

PGP Notations:

Ks	=sessionkeyusedinsymmetric encryptionscheme
PRa	=privatekeyofuserA,usedinpublic-keyencryptionscheme
PUa	=publickeyofuserA,usedinpublic-keyencryptionscheme
EP	=public-keyencryption
DP	=public-keydecryption
EC	=symmetricencryption
DC	=symmetricdecryption
H	=hashfunction
||	=concatenation
Z	=compressionusingZIPalgorithm
R64	=conversiontoradix64ASCIIformat

PGP Operation-Authentication



1.sendercreatesmessage

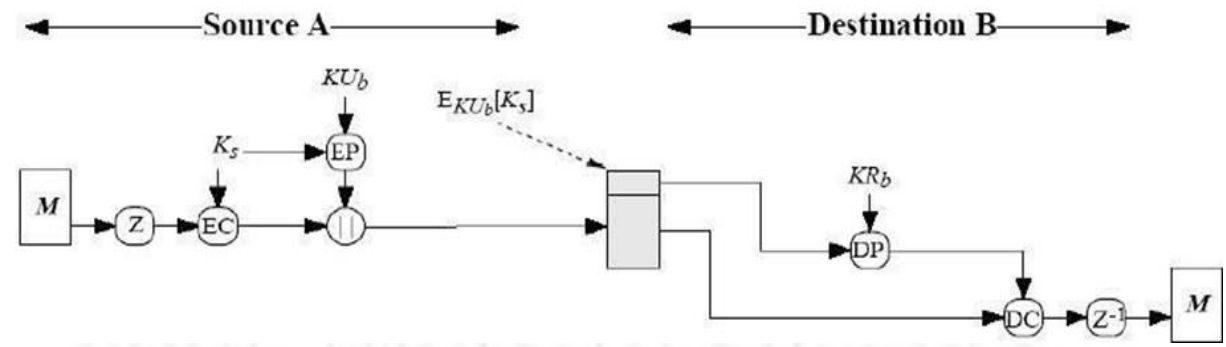
2.useSHA-1togenerate160-bithashofmessage

3.signedhashwithRSAusingsender'sprivatekey,andisattachedtomessage

4.receiverusesRSAwithsender'spublickeytodecryptandrecoverhashcode

5.receiververifiesreceivedmessage usinghash of it andcompares with decryptedhashcode

PGP Operation-Confidentiality



Sender:

1.Generatesmessageandarandomnumber(sessionkey)onlyforthismessage

2.EncryptsmessagewiththesessionkeyusingAES,3DES,IDEAorCAST-128

3.Encryptssessionkeyitselfwithrecipient'spublickeyusingRSA

4.Attachesittomessage

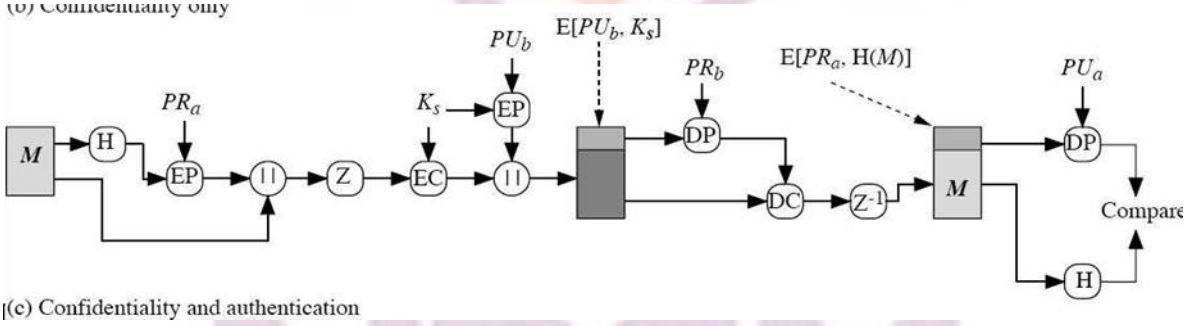
Receiver:

1.Recoverssessionkeybydecryptingusinghisprivatekey

2.Decryptsmessageusingthesessionkey

Confidentiality service provides no assurance to the receiver as to the identity ofsender(i.e.noauthentication).Onlyprovidesconfidentialityforsenderthatonly therecipient can read the message (and no one else) can use both services on samemessageocreatesignature&attachtomessageoencryptbothmessage&signatureoattachRSA/ElGamalencryptedsession key o*iscalled**authenticatedconfidentiality*.

PGPOperation–Confidentiality&Authentication



(c) Confidentiality and authentication

PGP Operation-Compression

As a default, PGP compresses the message after applying the signature but beforeencryption. This has the benefit of saving space both for e-mail transmission and forfilestorage.Theplacementofthecompressionalgorithm,indicatedbyZforcompressionand Z_{-1} fordecompression is critical. The compression algorithm used isZIP.

Thesignatureisgeneratedbeforecompressionfortworeasons:

1.sothatonecanstoreonlytheuncompressedmessagetogetherwithsignatureforlaterverification

2.Applying the hash function and signature after compression would constrain allPGP implementations to the same version of the compression algorithm as the PGPcompressionalgorithm isnotdeterministic

Messageencryptionisappliedaftercompressiontostrengthencryptographicsecurity. Because the compressedmessage has less redundancy than the originalplaintext,cryptanalysisismoredifficult.

PGP Operation-Email Compatibility

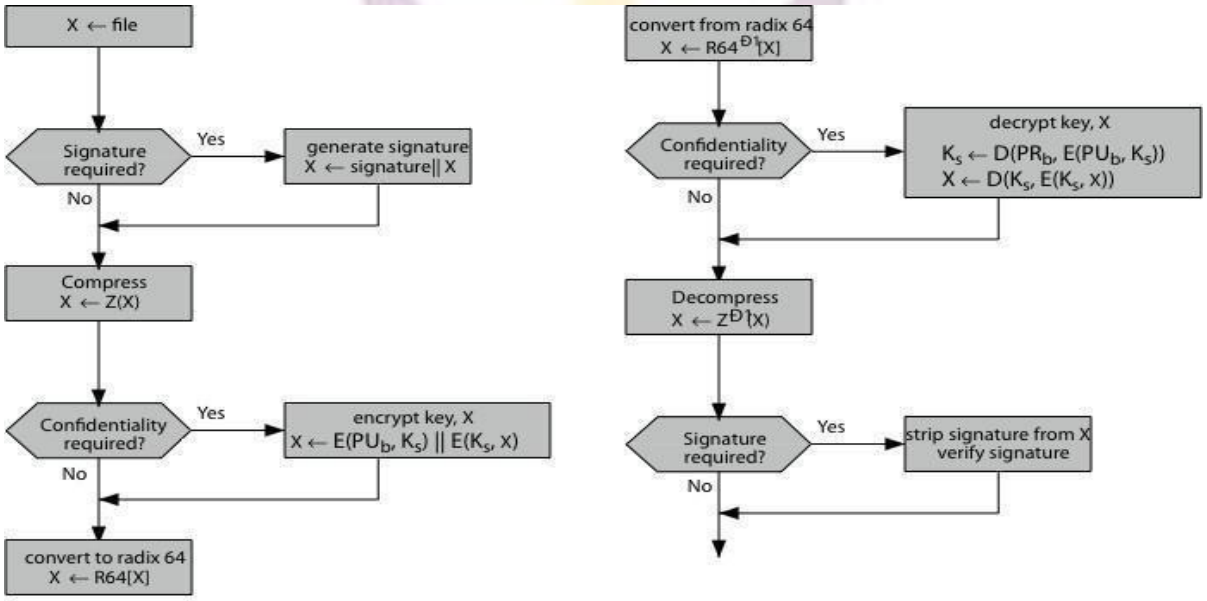
When PGP is used, at least part of the block to be transmitted is encrypted, and thusconsists of a stream of arbitrary 8-bit octets. However many electronic mail systemsonly permit the use of ASCII text. To accommodate this restriction, PGP provides theserviceof converting the raw 8-bit binary stream to a stream of printable ASCII characters. Itusesradix-64conversion,inwhicheachgroupofthreeoctetsofbinarydataismappedintofourASCIIcharacters.ThisformatalsoappendsaCRCtodetecttransmission errors. The use of radix 64 expands a message by 33%, but still anoverallcompressionofaboutone-thirdcanbeachieved.

PGP Operation-Segmentation/Reassembly

E-mailfacilitiesoftenarerestrictedtoa maximummessagelength.Forexample,manyof the facilities accessible through the Internet impose a maximum length of 50,000octets. Any message longer than that must be broken up into smaller segments, eachof which is mailed separately. To accommodate this restriction, PGP automaticallysubdividesamessagethatistoolargeintosegmentsthataresmallenoughto sendviae-mail. The segmentation is done after all of the other processing, including the radix-64 conversion. Thus, the session key component and signature component appearonly once, at the beginning of the first segment. Reassembly at the receiving end isrequired beforeverifyingsignatureordecryption

PGP Operations-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
		To provide transparency for email



(a) Generic Transmission Diagram (from A)

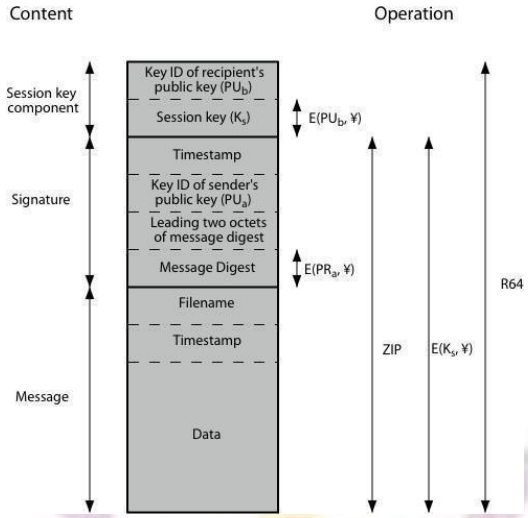
(b) Generic Reception Diagram (to B)

PGP Message Format

Amessageconsistsofthreecomponents:themessagecomponent,asignature(optional), and a session key component (optional). The *message component* includesthe actual data to be stored or transmitted, as well as a filename and a timestamp thatspecifiesthetimeofcreation.The*signaturecomponent*includesthefollowing:

1.Timestamp:Thetimeatwhichthesignaturewasmade.
2.Messagedigest: The160-bitSHA-1 digest, encrypted with the sender's privatesignaturekey.
3.Leading two octets of message digest:To enable the recipient to determine if thecorrect public key was used to decrypt the message digest for authentication, bycomparing this plaintext copy of the first two octets with

the first two octets of digest. These octets a 16-bit frame check themessage.



4.Key ID thedecrypted also serve as sequence for ofsender'spublickey:Identifies thepublickeythat shouldbe used todecrypt the message digest and, hence, identifies the private key that was used toencryptthemessagedigest.

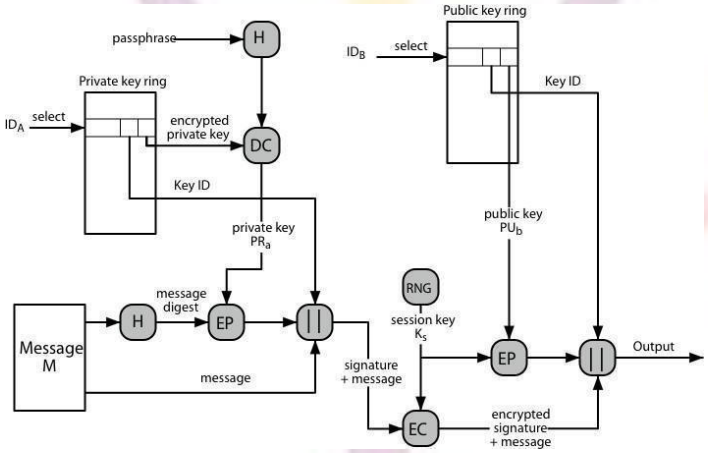
Thesessionkeycomponentincludesthesessionkeyandtheidentifieroftherecipient's public key that was used by the sender to encrypt the session key. Theentireblock isusuallyencoded withradix-64encoding.

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

PGPMessageTransmissionandReception

Messagetransmission

The following figure shows the steps during message transmission assuming that themessageistobebothsignedandencrypted.



ThesendingPGPentityperformsthefollowingsteps:

Signingthemessage

a. PGP retrieves the sender's private key from the private-key ring using your_useridas an index. If your_userid was not provided in the command, the first private key ontheringisretrieved.

b. PGPpromptstheuserforthepassphrasetorecovertheunencryptedprivate key.

c. Thesignaturecomponentofthemessageisconstructed

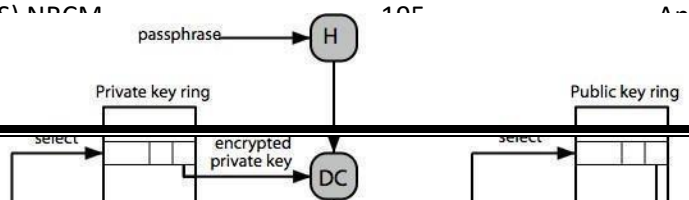
Encryptingthemessage

a. PGPgeneratesasessionkeyandencryptsthemessage.

b. PGPretrievestherecipient'spublickeyfromthepublic-keyringusingher_useridasanindex.

c. Thesessionkeycomponentof themessageis constructed.

MessageReception



ThereceivingPGPentityperformsthefollowingsteps:

Decryptingthemessage

a. PGPretrievesthereceiver'sprivatekeyfromtheprivate-keyring,usingtheKeyID fieldinthesessionkeycomponentofthemessageasanindex.

b. PGPpromptstheuserforthepassphrasetorecovertheunencryptedprivate key.

c. PGPthenrecoversthesessionkeyanddecryptsthemessage.

Authenticatingthemessage

a. PGPretrievesthesender'spublickeyfromthepublic-keyring,usingtheKeyIDfieldinthesignaturekeycomponentofthemessageasanindex.

b. PGPrecoversthetransmittedmessagedigest.

c. PGPcomputesthemessagedigestforthereceivedmessageandcomparesittothetransmitted messagedigesttoauthenticate.

S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extension) is a security enhancement tothe MIME Internet e-mail format standard, which in turn provided support for varyingcontenttypesandmulti-partmessages

overthetextonlysupportinthe originalInternet RFC822 email standard. MIME allows encoding of binary data to textual formfor transport over traditional RFC822 email systems.S/MIMEis defined in a numberofdocuments,mostimportantlyRFCs3369,3370,3850and3851andS/MIMEs upportisnowincluded inmanymodernmailagents.

RFC822

RFC 822 defines a format for text messages that are sent using electronic mail and ithas been the standard for Internet-based text mail message. The overall structure of amessage that conforms to RFC 822 is very simple. A message consists of some numberof header lines (the header) followed by unrestricted text (the body). The header isseparated 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 longline to 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 ofthe problems and limitations of the use of SMTP (Simple Mail Transfer Protocol) orsome other mail transfer protocol and RFC 822 for electronic mail.**Problems withRFC822andSMTP**

ExecutablefilesorotherbinaryobjectsmustbeconvertedintoASCII.Vario usschemes exist(e.g.,UnixUUencode),butastandardisneeded

Textdatathatincludesspecialcharacters(e.g.,Hungariantext)cannotbetr ansmittedasSMTPislimitedto7-bitASCII

Someserversrejectmailmessagesoveracertainsize

SomecommonproblemsexistwiththeSMTPimplementationswhichdonotadhereco mpletelytotheSMTPstandardsdefined inRFC821.Theyare:

delete,add,orreorderCRandLFcharacterstruncateorwraplines longer than 76 characters remove trailing white space(tabs and spaces) pad lines in a message to the same lengthconverttabcharactersintomultiplespaces

MIME is intended to resolve these problems in a manner that is compatible withexistingRFC822implementationsandthespecificationisprovidedinRFC's2045through 2049.

TheMIMEspecificationincludesthefollowingelements:

1.Five newmessage headerfields are defined,which provide information about thebodyofthemessage.

2.Anumberofcontentformatsaredefined,thusstandardizingrepresentations thatsupportmultimediaelectronicmail.

3.Transfer encodings aredefinedthat protectthe contentfromalterationby the mailsystem.

MIME-Newheaderfields ThefiveheaderfieldsdefinedinMIMEareasfollows:

MIME-Version:Musthavetheparametervalue1.0.Thisfieldindicatesthatthemessageconformstos RFCs2045and2046.

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 thedatatotheuserorotherwisedealwiththedatainanappropriatemanner.

Content-Transfer-Encoding:Indicates thetypeoftransformationthathas been usedtorepresentthebodyofthemessageinawaythatisacceptableformailtransport.

Content-ID:UsedtoidentifyMIMEentitiesuniquelyinmultiplecontexts.

Content-Description:Atextdescriptionoftheobjectwiththebody;thisisusefulwhenthe objectisnotreadable(e.g.,audiodata).

MIMEContentTypesThebulkoftheMIMEspecificationisconcernedwiththedefinitionofavarietyofcontenttypes.Therearesevendifferentmajortypesofcontent and a total of 15 subtypes. In general, a content type declares the general typeof data, and the subtype specifies a particular format for that type of data. For the texttype of body, theprimary subtype isplain text, which issimply astring of ASCIIcharacters or ISO 8859 characters. The enriched subtype allows greater formattingflexibility. The multipart type indicates that the body contains multiple, independentparts. The Content-Type headerfield includes a parameter called boundary that definesthe delimiter between body parts. This boundary should not appear in any parts of themessage. Each boundary starts on a new line and consists of two hyphens followed bytheboundaryvalue.Thefinalboundary,whichindicatestheendofthelastpart,alsohasa suffix of two hyphens. Within each part, there may be an optional ordinary MIMEheader. There are four subtypes of the multipart type, all of which have the sameoverallsyntax.

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.

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.

The message type provides a number of important capabilities in MIME. Themessage/rfc822subtypeindicatesthatthebodyisanentiremessage,including header and body. Despite the name of this subtype, the encapsulated message may be notonly a simple RFC 822 message, but also any MIME message. The message/partialsubtype enables fragmentation of a large message into a number of parts, which mustbe reassembled at the destination. For this subtype, three parameters are specified inthe Content-Type: Message/Partial field: an id common to all fragments of the samemessage,asequencenumberuniquetoeachfragment,andthetotalnumberoffrag ments. The message/external-body subtype indicates that the actual data to beconveyed in this message are not contained in the body. Instead, the body contains theinformation needed to access the data. The application type refers to other kinds ofdata, typically either uninterpreted binary data or information to be processed by amail-basedapplication.

MIME Transfer Encodings The other major component of the MIME specification, inadditiontocontenttypespecification,isadefinitionoftransferencodingsformessage bodies. The objective is to provide reliable delivery across the largest rangeofenvironments.

MIME Transfer Encodings

De
pr

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.

TheMIMEstandarddefinestwomethodsofencodingdata.TheContent-Transfer-Encoding field can actually take on six values. Three of these values (7bit, 8bit, and binary)indicate that no encoding has been done but provide some information about the nature ofthe data. Another Content-Transfer-Encoding value is x-token, which indicates that someotherencodingschemeisused,forwhichanameistobesupplied.Thetwoactualencodingschemes defined are quoted- printable and base64. Two schemes are defined to provide achoicebetween atransfertechniquethatisessentiallyhumanreadable and one that is safe for all types of data in a way that is reasonably compact. Thequoted-printable transfer encoding is useful when the data consists largely of octets thatcorrespond to printable ASCII characters. In essence, it represents nonsafe characters bythe hexadecimal representation of their code and introduces reversible (soft) line breaks tolimit message lines to 76 characters. The base64 transfer encoding, also known as radix-64encoding, is a common one for encoding arbitrary binary data in such a way as to beinvulnerabletotheprocessing bymailtransportprograms.

CanonicalForm

An important concept in MIME and S/MIME is that of canonical form. Canonical formis a format, appropriate to the content type, that is standardized for use betweensystems. This is in contrast to native form, which is a format that may be peculiar to aparticularsystem.

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/or encryptmessages.

Functions

S/MIMEprovidesthefollowingfunctions:

Enveloped data: This consists of encrypted content of any type and encrypted-contentencryption keysforoneormorerecipients.

Signed data: A digitalsignature is formed by takingthemessagedigest of thecontent to be signed and then encrypting that with the private key of the signer.

Thecontentplussignaturearethenencodedusingbase64encoding.Asigneddatame ssagecan onlybeviewedbyarecipientwithS/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,recipients without S/MIME capability can view the message content, although theycannotverifythesignature.

Signedandenvelopeddata:Signed-onlyandencrypted-onlyentitiesmaybenested, so that encrypted data may be signed and signed data or clear-signed datamaybeencrypted.

IP SECURITY OVERVIEW

Definition: Internet Protocol security (IPSec) is a framework of open standards forprotectingcommunicationsoverInternetProtocol(IP)networksthroughtheuse 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**, inwhich intruders create packets with false IP addresses and exploit applications thatuse authentication based on IP and various forms of**eavesdropping and packetsniffing**, in which attackers read transmitted information, including logon informationanddatabasecontents. In response to these issues, the IAB included authentication and encryptionas necessarysecurityfeaturesinthenext-generationIPi.e.IPv6.

ApplicationsofIPSec

IPSec provides the capability to secure communications across a LAN, across privateandpublicwidearea networks(WAN's),andacrossthe Internet.

Secure branch office connectivity over the Internet: A company can build a securevirtualprivatenetworkovertheInternetoroverapublicWAN.Thisenablesab usiness to rely heavily on the Internet and reduce its need for private networks,savingcostsandnetworkmanagementoverhead.

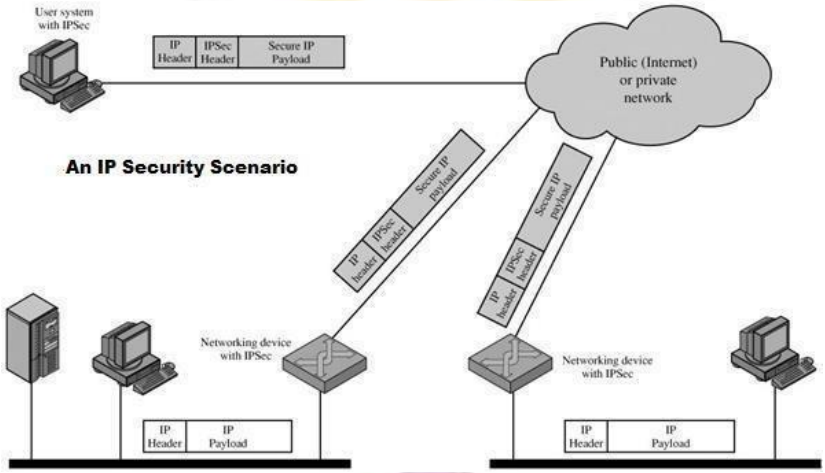
Secure remote access over the Internet: An end user whose system is equippedwith IP security protocols can make a local call to an Internet service provider (ISP)and gain secure access to a company network. This reduces the cost of toll charges fortravellingemployeesandtelecommuters.

Establishing extranet and intranet connectivity with partners: IPSec can be usedtosecurecommunicationwithotherorganizations,ensuringauthenticationan dconfidentialityandprovidingakeyexchangemechanism.

Enhancing electronic commerce security: Even though some Web and electroniccommerceapplicationshavebuilt-insecurityprotocols,theuseofIPSecenhancesthatsecurity.

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, canbesecured.

The following figure shows a typical scenario of IPSec usage. An organization maintainsLANsatdispersed locations.NonsecureIPtrafficisconducted oneachLAN.



BenefitsofIPSec

ThebenefitsofIPSecarelistedbelow:

- IPSecinafirewall/routerprovidesstrongsecuritytoalltrafficcrossin gtheperimeter
- IPSecinafirewallisresistanttobypass
- IPSecisbelowtransportlayer(TCP,UDP),hencetransparenttoapplications
- IPSeccanbetransparenttoendusers
- IPSec can provide security for individual users if needed (useful foroffsite workersandsettingupa secure virtualsubnetwork forsensitiveapplications)

RoutingApplications

IPSecalsoplaysavitalroleintheroutingarchitecturerequiredforinternetwo rking.Itassuresthat:

- routeadvertisementscomefromauthorizedrouters
- neighboradvertisementscomefromauthorizedrouters

- redirectmessagescomefromtheroutertowhichinitialpacketwassent
- Aroutingupdateisnotforged

IPSECURITYARCHITECTURE

To understand IP Security architecture,we examine IPSecdocumentsfirstandthenmoveontoIPSecservicesandSecurityAssociations.

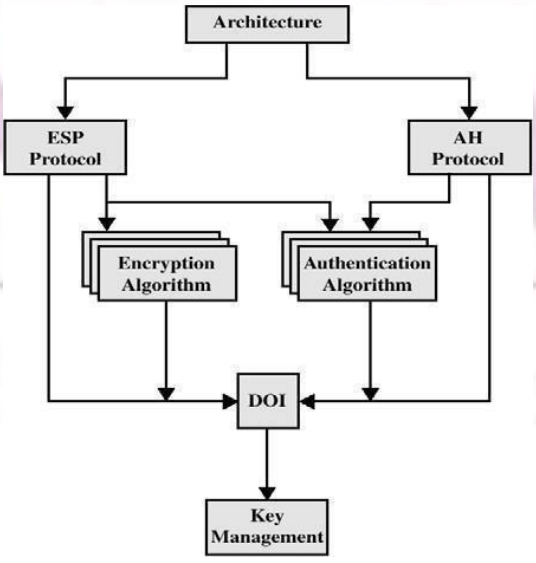
IPSecDocuments

The IPSec specification consists of numerous documents. The mostimportantofthese,issuedinNovemberof1998,areRFCs2401,2402, 2406,and2408:

- RFC2401:Anoverviewofasecurityarchitecture
- RFC2402:DescriptionofapacketauthenticationextensiontoIPv4andIPv6
- RFC2406:DescriptionofapacketencryptionextensiontoIPv4andIPv6
- RFC2408:Specificationofkeymanagementcapabilities

Support forthesefeaturesismandatoryforIPv6andoptionalforIPv4.Inbothcases,thesecurityfeaturesareimplementedasextensionheadersthatfollowthemainIPheader.

The extension header for authentication is known as the Authentication header; thatforencryptionisknownastheEncapsulatingSecurityPayload(ESP)header.Inaddition to these four RFCs, a number of additional drafts have been published by theIP Security Protocol Working Group set up by the IETF. The documents are dividedintoseven groups,asdepictedinfollowingfigure



- **Architecture**: Covers the general concepts, security requirements, definitions, andmechanismsdefiningIPSectechnology
- **EncapsulatingSecurityPayload(ESP):**Coversthepacketformatandgeneralissuesrelatedtotheuseoftheespforpacketencryptionand,optionally,authentication.
- **Authentication Header (AH):** Covers the packet format and general issues relatedtotheuseofAHforpacket authentication.

- **EncryptionAlgorithm**:AsetofdocumentsthatdescribehowvariousencryptionalgorithmsareusedforESP.
- **AuthenticationAlgorithm:**AsetofdocumentsthatdescribehowvariousauthenticationalgorithmsareusedforAHandfortheauthenticationoptionofESP.
- **KeyManagement**:Documentsthatdescribekeymanagementschemes.
- **Domain of Interpretation (DOI):** Contains values needed for the other documentstorelatetoeachother.Theseincludeidentifiersforapprovedencryptionandauthenticationalgorithms,aswellasoperationalparameterssuchaskeylifetime.

IPSecServices

IPSec architecture makes use of two major protocols (i.e., Authentication Header andESPprotocols)forprovidingsecurityatIPlevel.Thisfacilitatesthesystemtobeforehand choose an algorithm to be implemented, security protocols needed and anycryptographic keys required to provide requested services. The IPSec services are asfollows:

Connectionless Integrity:-Data integrity service is provided by IPSec via AHwhichpreventsthe datafrombeingaltered duringtransmission.

DataOriginAuthentication:-ThisIPSecservicepreventstheoccurrenceof replayattacks,addressspoofingetc.,whichcanbefatal.

Access Control:- The cryptographic keys are distributed and the traffic flow iscontrolledinbothAHandESPprotocols,whichisdonetoaccomplishaccesscontroloverthedatatransmission.

Confidentiality:-Confidentialityonthedatapacketisobtainedbyusingan encryptiontechniqueinwhichallthedatapacketsaretransformedintociphertext packetswhichareunreadableand difficulttounderstand.

Limited Traffic Flow Confidentiality:- This facility or service provided by IPSecensuresthat the confidentialityismaintainedon thenumber of packetstransferred orreceived.ThiscanbedoneusingpaddinginESP.

ReplaypacketsRejection:-
Theduplicateorreplaypacketsareidentifiedanddiscardedusingthesequencenumberfield inbothAHandESP.

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

AUTHENTICATIONHEADER

The Authentication Header provides support for data integrity and authentication ofIPpackets.Thedataintegrityfeatureensuresthatundetectedmodificationtoapacket's content in transit is not possible. The authentication feature enables an endsystem or network device to authenticate the user or application and filter trafficaccordingly; it also prevents the address spoofing attacks observed in today's Internet.The AH also guards against the replay attack. Authentication is based on the use of amessage authentication code (MAC), hence the two parties must share a secret key.TheAuthenticationHeaderconsistsofthefollowingfields:

IPSecAuthenticationHeader

NextHeader(8bits): Identifiesthetypeofheaderimmediatelyfollowingthisheader.

PayloadLength(8bits): LengthofAuthenticationHeaderin32-bitwords,minus2.For example, the default length of the authentication data field is 96 bits, or three 32-bit words. With a three-word fixed header, there are a total of six words in the header,and thePayloadLengthfieldhasavalueof4.

Reserved(16bits): Forfutureuse.

SecurityParametersIndex(32bits): Identifiesasecurityassociation.

Sequence Number (32 bits): A monotonically increasing counter value, discussedlater.

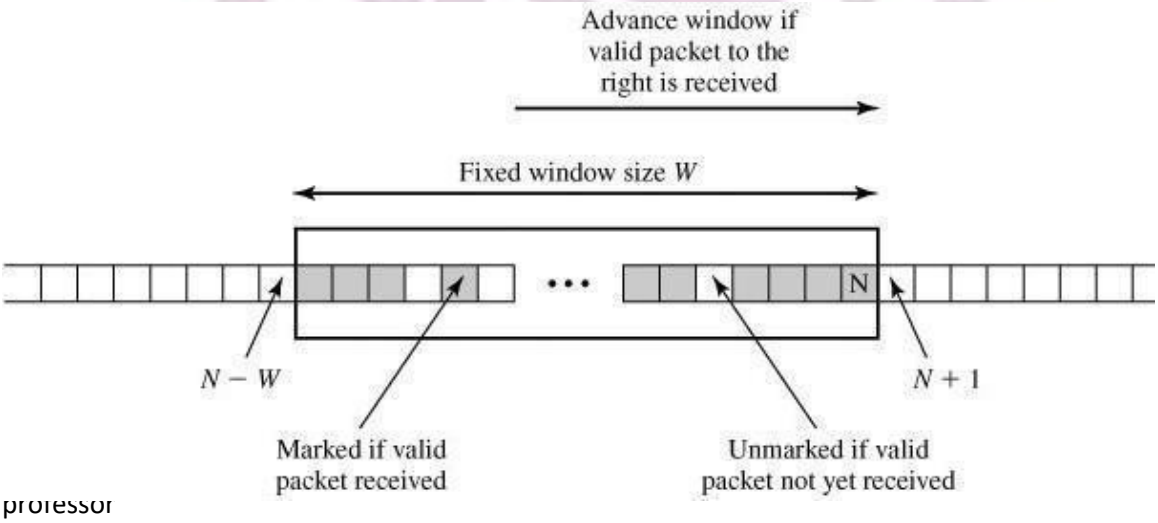
AuthenticationData(variable): Avariable-lengthfield(mustbeanintegralnumber of 32-bit words) that contains the Integrity Check Value (ICV), or MAC, forthispacket.

Anti-ReplayService

Anti-replay service is designed to overcome the problems faced due to replay attacksin which an intruder intervenes the packet being transferred, makeone or moreduplicate copies of that authenticated packet and then sends the packets to the desireddestination, thereby causing inconvenient processing at thedestination node. TheSequenceNumberfield isdesignedtothwartsuchattacks.

When a new SA is established, the sender initializes a sequence number counter to 0.Each time that a packet is sent on this SA, the sender increments the counter andplaces the value in the Sequence Number field. Thus, the first value to be used is 1.Thisvaluegoesonincreasingwithrespecttothenumberofpacketsbeingtransmitted. The sequence number field in each packet represents the value of thiscounter. The maximum value of the sequence number field can go up to 2_{32}-1. If thelimit of 2_{32}-1 is reached, the sender should terminate this SA and negotiate a new SAwithanewkey.

TheIPSecauthenticationdocumentdictatesthatthereceivershouldimplementa window of size W, with a default of W = 64. The right edge of the window representsthe highest sequence number, N, so far received for a valid packet. For any packetwitha sequencenumberintherangefromN-W+1toNthathasbeencorrectlyreceived(i.e.,properlyauthenticated),thecorrespondingslotinthewindow ismarkedasshown.Inboundprocessingproceedsasfollowswhenapacketisreceived:



professor

1.Ifthereceivedpacketfallswithinthewindow andisnew,theMACischecked.Ifthepacketis authenticated,thecorrespondingslotinthe windowismarked.

2.If the received packet is to the right of the window and is new, the MAC is checked.If the packet is authenticated, the window is advanced so that this sequence numberistherightedgeofthewindow,andthecorrespondingslotinthewindowismarked.

3.If the received packet is to the left of the window, or if authentication fails, thepacketisdiscarded;thisisan 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 canalso be referred as MAC or part of MAC algorithm. MD5 hash code and SHA-1 hashcodeareimplementedalongwithHMACalgorithmsi.e.,

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

In both cases, the full HMAC value is calculated but then truncated by using the first96 bits, which is the default length for the Authentication Data field. The MAC iscalculatedover IPheaderfieldsthateitherdonotchangeintransit(immutable)orthatarepredictableinvalueuponarrivalattheendpointfortheAHSA.Fieldsthatmaychangein transit and whose value on arrival is unpredictable are set to zero for purposes ofcalculationatbothsourceanddestination.

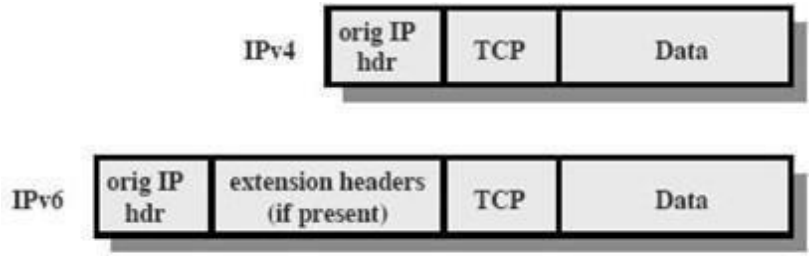
The AH header other than the Authentication Data field. The Authentication Datafieldis set tozeroforpurposesofcalculationat bothsource anddestination.

The entire upper-level protocol data, which is assumed to be immutable in transit(e.g.,aTCPsegmentoraninnerIPpacketintunnelmode).

TransportandTunnelModes

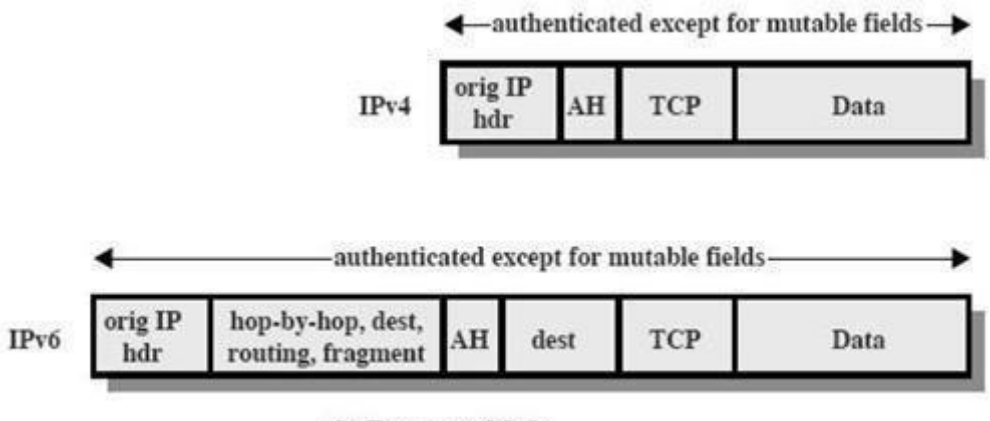
The following figure shows typical IPv4 and IPv6 packets. In this case, the IP payloadis a TCP segment; it could also be a data unit for any other protocol

that uses IP, suchasUDPorICMP.

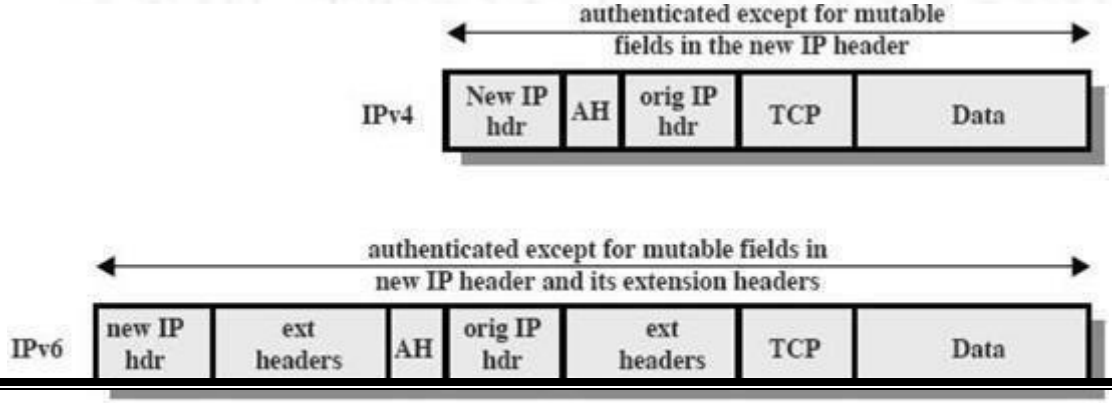


(a) Before Applying AH

For transport mode AH using IPv4, the AH is inserted after the original IP header andbefore the IP payload (e.g., a TCP segment) shown below. Authentication covers theentire packet, excluding mutable fields in the IPv4 header that are set to zero for MACcalculation. In the context of IPv6, AH is viewed as an end-to-end payload; that is, it isnot examined or processed by intermediate routers. Therefore, the AH appears afterthe IPv6 base header and the hop-by-hop, routing, and fragment extension headers.The destination options extension header could appear before or after the AH header,depending on the semantics desired. Again, authentication covers the entire packet,excludingmutablefieldsthataresettozeroforMACcalculation.



(b) Transport Mode



For tunnel mode AH, theentire original IP packet is authenticated, and the AH isinserted between the original IP header and a new outer IP header. Theinner IPheadercarriestheultimatesourceanddestinationaddresses,whileanouterIPheadermaycontaindifferentIPaddresses(e.g.,addressesoffirewallsorothersecuritygateways).Withtunnelmode,theentireinnerIPpacket,includingtheentire inner IP header is protected by AH. The outer IP header (and in the case of IPv6, theouterIPextensionheaders)isprotectedexceptformutableandunpredictablefields.

ENCAPSULATINGSECURITYPAYLOAD

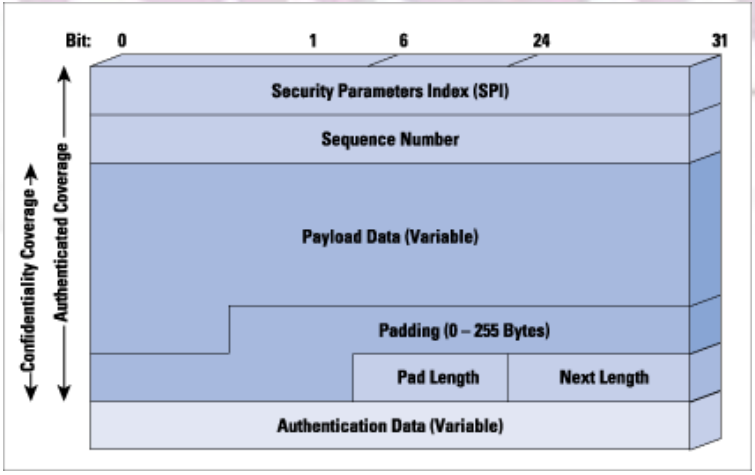
TheEncapsulatingSecurityPayloadprovidesconfidentialityservices,includingconfidentialityofmessagecontentsandlimitedtrafficflowconfidentiality.Asanoptionalfeature,ESPcanalsoprovideanauthenticationservice.

ESPFormat

The following figure shows the format of an ESP packet. It contains the followingfields:

SecurityParametersIndex(32bits):Identifiesasecurityassociation.

SequenceNumber(32bits):Amonotonicallyincreasingcountervalue;thisprovidesan anti-replayfunction,asdiscussed forAH.



Payload Data (variable): This is a transport-level segment (transport mode) or IPpacket(tunnelmode)thatisprotected byencryption.

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

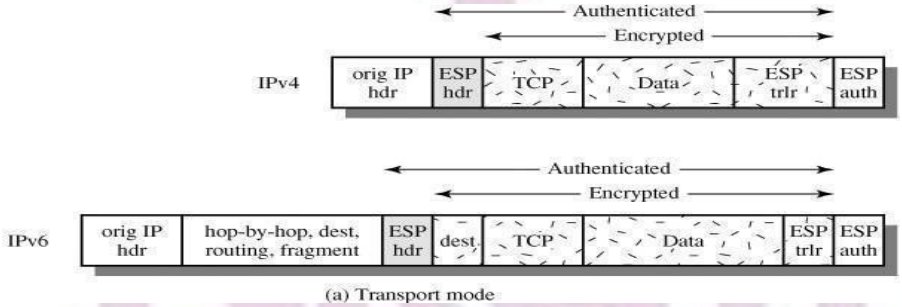
Pad Length (8 bits): Indicates the number of pad bytes immediately preceding thisfield.

Next Header (8 bits): Identifies the type of data contained in the payload data fieldby identifying the first header in that payload (for example, an extension header inIPv6,oran upper-layerprotocolsuchasTCP).

AuthenticationData(variable): Avariable-lengthfield(mustbeanintegralnumber of 32-bit words) that contains the Integrity Check Value computed over theESPpacketminustheAuthenticationDatafield.

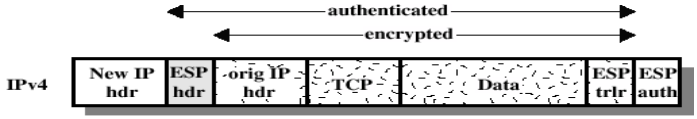
AddingencryptionmakesESPabitmore complicatedbecausetheencapsulation*surrounds* the payload rather than *precedes* it as with AH: ESP includes header andtrailer

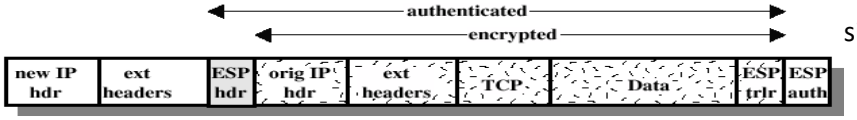
Transport ModeESP



(a) Transport mode

TunnelModeESP



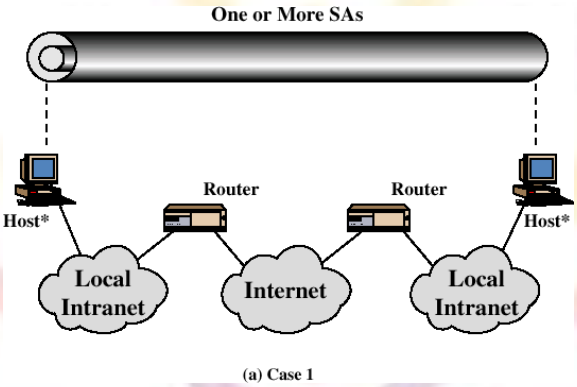


(b) Tunnel Mode

BasicCombinationsofSecurity Associations

TheIPSecArchitecturedocumentlistsfourexamplesofcombinationso fSAs that must be supported by compliant IPSec hosts (e.g., workstation,server)orsecuritygateways(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 anSA, they must share the appropriate secret keys.Amongthepossiblecombinations:

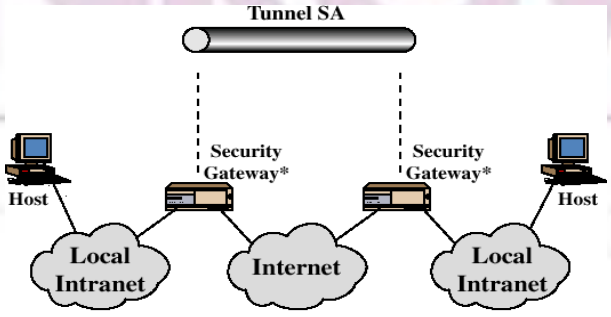
a)AHintransportmode

b)ESPintransportmode

c)ESPfollowedbyAHintransportmode(anESPSAinsideanAHSA)

d)Anyoneofa,b,orcinsideanAHorESPintunnelmode

Case:-2

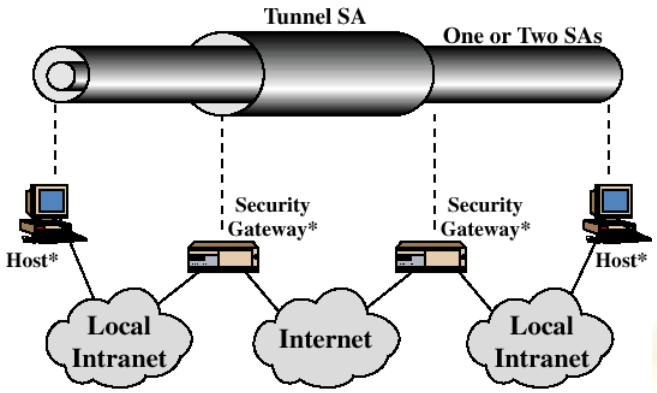


(b) Case 2

Security is provided only between gateways (routers, firewalls, etc.) and no hostsimplement IPSec. This case illustrates simple virtual private network support. Thesecurity architecture document specifies that only a single

tunnelSAis needed forthis case. The tunnel could support AH, ESP, or ESP with the authentication option.Nested tunnels are not required because the IPSec services apply to the entire innerpacket.

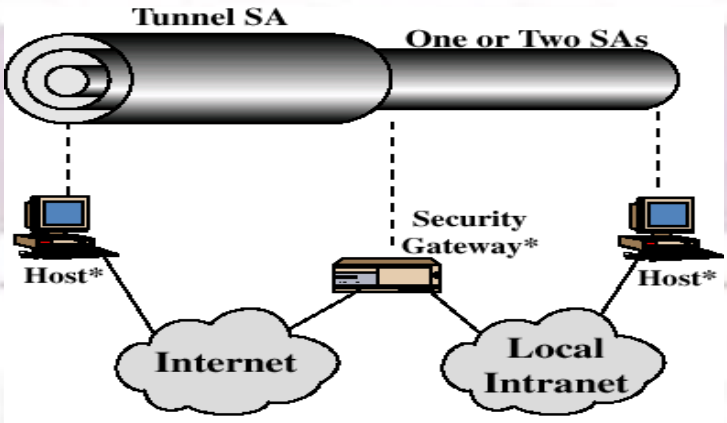
Case-3:-



(c) Case 3

The third combination is similar to the second, but in addition provides security eventonodes.Thiscombination makesuseoftwotunnelsfirstforgatewaytogatewayandsecond fornode tonode. Either authentication orthe encryption or both can beprovided by using gateway to gateway tunnel. An additional IPSec service is providedtotheindividualnodesbyusing nodetonodetunnel.

Case:-4



(d) Case 4

Thiscombinationissuitableforservingremoteusersi.e.,theendusersittinganywhere in the world can use the internet to access the organizational workstationsviathefirewall.Thiscombinationstatesthatonlyonetunnelisneededforcom municationbetweenaremote userandanorganizationalfirewall.

COMBINING SECURITY ASSOCIATIONS

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 whichspecifies security parameters controlling security operations. They are referenced bythe sending host and established by the receiving host. An index parameter called theSecurity 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 isuniquelyidentifiedbythreeparameters:

Security Parameters Index (SPI): A bit string assigned to this SA and having localsignificance only. The SPI is carried in AH and ESP headers to enable the receivingsystem toselecttheSAunderwhichareceivedpacketwillbeprocessed.

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 anetwork system suchasafirewallorrouter.

Security Protocol Identifier: This indicates whether the association is an AH orESPsecurityassociation.

SAParameters

In each IPSec implementation, there is a nominal Security Association Database thatdefines the parameters associated with each SA. A security association is normallydefined bythefollowingparameters:

Sequence Number Counter: A 32-bit value used to generate the Sequence Numberfieldin AHorESPheaders

Sequence Counter Overflow: A flag indicating whether overflow of the SequenceNumberCountershould generatean auditableeventand preventfurther transmissionofpacketsonthisSA(required forallimplementations).

Anti-ReplayWindow: UsedtodeterminewhetheraninboundAHorESPpacketisarepl ay

AHInformation: Authenticationalgorithm,keys,keylifetimes,andrela tedparametersbeingusedwithAH(requiredforAHimplementations).

ESPInformation: Encryption andauthentication algorithm, keys, initialization

values,keylifetimes,andrelatedparametersbeingusedwithESP(requiredforESPimplementations).

Lifetime of This Security Association: A time interval or byte count after which anSA must be replaced with a new SA (and new SPI) or terminated, plus an indication ofwhichoftheseactionsshouldoccur(requiredforallimplementations).

IPSecProtocolMode: Tunnel,transport,orwildcard(requiredforallimplementations).Thesemodesare discussed laterinthissection.

Path MTU: Any observed path maximum transmission unit (maximum size of apacket that can be transmitted without fragmentation) and aging variables (requiredforallimplementations).

TransportandTunnelModes

Both AHandESPsupport twomodes of use:transport and tunnelmode.

	TransportModeSA	**TunnelModeSA**
AH	**Authenticates** IP payloadand selected portions of IPheaderandIPv6extension headers	**Authenticates** entire innerIP packet plus selectedportions ofouterIPheader
ESP	**Encrypts**IPpayloadand anyIPv6extesionheader	**Encrypts** innerIPpacket
ESPwithauthentication	**Encrypts** IP payload andany IPv6 extesion header.**Authenticates**IPpayload but noIPheader	**Encrypts** inner IP packet.**Authenticates** inner IPpacket

IPseccanbeused(bothAHpacketsandESPpackets)intwo modes

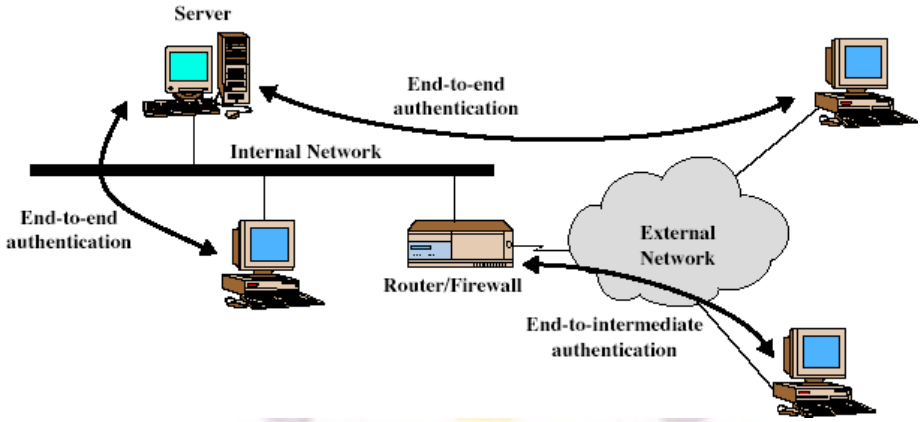
a.Transport mode: the IP sec header is inserted just after the IP header – thiscontainsthesecurityinformation,suchasSAidentifier,encryption,authentication

Typicallyusedinend-to-endcommunicationIPheadernotprotected

b.Tunnelmode:theentireIPpacket,headerandall,isencapsulatedinthebodyofa newIPpacketwithacompletelynewIPheader

Typicallyusedinfirewall-to-firewallcommunication Providesprotection

forthewholeIPpacket
Noroutersalongthewaywillbeable(andwillnotneed)tocheckthecontentofthepackets.



Internet Key Exchange

What Does Internet Key Exchange (IKE) Mean?
Internet Key Exchange (IKE) is a key management protocol standard used in conjunction with the Internet Protocol Security (IPSec) standard protocol. It provides security for virtual private networks' (VPNs) negotiations and network access to random hosts. It can also be described as a method for exchanging keys for encryption and authentication over an unsecured medium, such as the Internet.

IKE is a hybrid protocol based on:

A.ISAKMP (RFC2408): Internet Security Association and Key Management Protocols are used for negotiation and establishment of security associations. This protocol establishes a secure connection between two IPSec peers.

B.Oakley (RFC2412): This protocol is used for key agreement or key exchange. Oakley defines the mechanism that is used for key exchange over an IKE session. The default algorithm for key exchange used by this protocol is the Diffie-Hellman algorithm.

C.SKEME: This protocol is another version for key exchange.

D.IKE enhances IPsec by providing additional features along with flexibility. IPsec, however, can be configured without IKE.

IKE has many benefits. It eliminates the need to manually specify all the IPSec security parameters at both peers. It allows the user to specify a particular lifetime for the IPsec security association. Furthermore, encryption can be changed during IPsec sessions. Moreover, it permits certification authority. Finally, it allows dynamic authentication of peers.

Techopedia Explains Internet Key Exchange (IKE)

The IKE works in two steps. The first step establishes an authenticated communication channel between the peers, by using algorithms like the Diffie-Hellman key exchange, which generates a shared key to further encrypt IKE communications. The communication channel formed as a result of the algorithm is a bi-directional channel. The authentication of the channel is achieved by using a shared key, signatures, or public key encryption.

There are two modes of operation for the first step: main mode, which is utilized to protect the identity of the peers, and aggressive mode, which is used when the security of the identity of the peers is not an important issue. During the second step, the peers use the secure communication channel to set up security negotiations on behalf of other services like IPSec. These negotiation procedures give rise to two unidirectional channels of which one is inbound and the other outbound. The mode of operation for the second step is the Quick mode.

IKE provides three different methods for peer authentication: authentication using a pre-shared secret, authentication using RSA encrypted nonces, and authentication using RSA signatures. IKE uses the HMAC functions to guarantee the integrity of an IKE session. When an IKE session lifetime expires, a new Diffie-Hellman exchange is performed and the IKE SA is re-established.

Case Studies on cryptography and Security

Secure Multiparty Calculation

In the last few decades, data privacy and security has become the primary concern to everyone. Due to the rise in technological advancements and the internet, it has been a challenging task to provide data security and data privacy of the data, when data is distributed over large distributed networks. As everyone is now concerned with their data, a lot of research is going on how to provide data security and privacy to the participants in the network. One of the techniques which provide the solution to the problems of data security and data privacy is **Secure Multiparty Computation.** The secure multiparty computation may be defined as the problem of 'n' players to compute jointly on an agreed function securely on the inputs without revealing them.

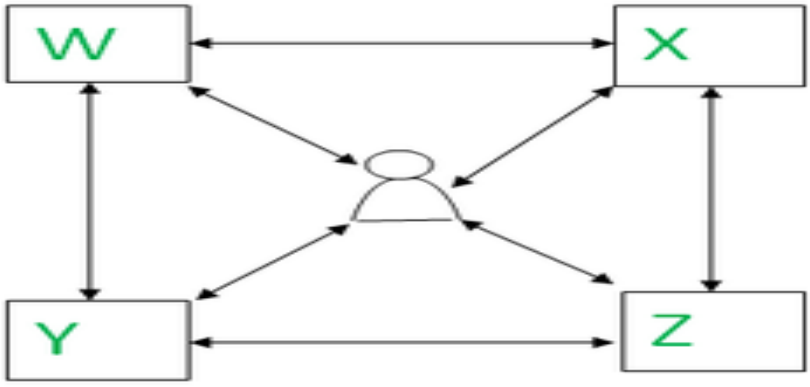
History

Secure multiparty computation started early in the 1970s. It was known as multiparty computation at that time. It does not gain popularity at that time as it was not implemented practically. In the 1982's it was introduced as secure two-party multiparty computation. It is used to solve a lot of problems of computation without revealing the inputs to other parties. Finally, it came with a name as secure multiparty computation in which the functions of different types are computed, that is the reason it is sometimes called **SFE- Secure Function Evaluation**.

1.The secure multiparty computation is used for the utilization of data without compromising privacy.
2.It is the cryptographic subfield that helps in preserving the privacy of the data.
3.Emerging technologies like blockchain, mobile computing, IoT, cloud computing has resulted in the rebirth of secure multiparty computation.
4.Secure multiparty computation has become the hot area of research in the last decade due to the rise of blockchain technology.
5.The researchers are now more interested to implement secure multiparty computation in distributed systems.
6.Unlike in centralized systems, secure multiparty computation may have better performance in distributed systems.

Architecture

The secure multiparty computation provides a protocol where no individual can see the other parties data while distributing the data across multi parties. It enables the data scientists and analysts to compute privately on the distributed data without exposing it.



Multiparty sharing data among each other with any third party using a specified protocol.

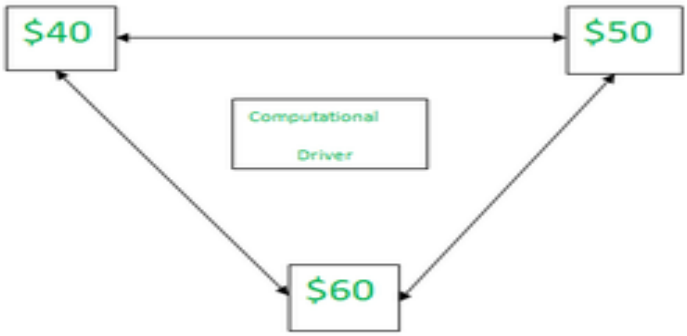
The co-workers want to compute the maximum salary without revealing their individual salary to others. To perform such a computation, secure multiparty computation is implemented to calculate the maximum salary. The parties in a distributed manner jointly perform a function to calculate it without revealing the salary. Data in use is kept in encrypted form, broken up, and distributed across parties, there are no chances of quantum attacks. It is impossible to have a trusted party in the real world, as all parties communicate with each other in one or the other way In such a scenario, the parties may get corrupted. The corrupted parties have behavior like semi-honest and malicious.

1.A semi-honest opponent is one who follows the specified protocol but makes the parties corrupted. The protocol is run honestly, but they try to extract information from the messages exchanged between parties.

2.A malicious adversary makes an attempt to breach security and does not follow the specified protocol. The adversary can make the changes during the execution process of the protocol. While using multiparty computation, we assume the party is honest which follows all the protocols.

Example

Suppose we want to compute the average salary among three employees without revealing the actual salary, for such problems one can use secure multiparty computation. Let's take an example-



Example of computing average salary of multiparty using additive sharing.

Mathematical representation of the problem can be given as:

F(A, B, C) = Average (A, B, C)

Sam, Bob, and Cassy want to calculate their average salary.

1. Say Sam's salary is $40k. Using additive sharing, $40k is split into randomly generated three pieces $44k, $-11k, and $7k.
2. Sam keeps one of these secret pieces with herself and distributes the other two to each.
3. The same procedure is followed by all three.
4. Secret sharing keeps the data in encrypted form when in use. The procedure is given below-

Sam	Bob	Cassy	
44	-11	7	**$40**

-6	32	24	**$50**
20	0	40	**$60**
$58	**$21**	**$71**	

Total salary = $150

Average Salary = 150/3

= $50

From the above data shared there is no clue about the actual salary, but the average salary is being calculated.

Techniques

There are a number of techniques developed for secure multiparty computation protocol construction having different features. Some techniques used in secure Multiparty computation are listed below:

1. **Shamir Secret Sharing:** Secret sharing is utilized as the basic tool when there is an honest majority in secure multiparty computation. A secret sharing scheme is that a secret s is shared among n parties, such that t+1 or more parties come together to reconstruct the secret. The parties lesser than t cannot get any information or reconstruct the secret. The scheme which fulfills the requirements of t+1 out of n is called the threshold secret sharing scheme.
2. **Honest Majority MPC:** The function can either be represented by Boolean or arithmetic circuit in an honest majority. For MPC-based secret sharing having the honest majority, there is finite field Zp with p>n for arithmetic circuit and the circuit is Turing complete.
3. **Input sharing:** Every party shares the input using the Shamir secret sharing. The circuit is being provided the input for computation. Every party keeps his input private by adding some random number to the input and finally, after getting the output the random number is known to the party is removed, and we get the output.
4. **Circuit evaluation:** The circuit is evaluated by parties one gate at a time. The gates are evaluated serially from input to output. The evaluation consists of the computation of addition and multiplication gates. For inputs a(x) and b(x), the output of addition for the ith party is calculated as c(i) = a(i) + b(i). Similarly, the output of multiplication for the ith party is calculated as c(i) = a(i) . b(i).

5. **Private set intersection:** The private set intersection protocol is very efficient for the two parties' problems. Two parties who wish to find the elements of intersection with private set of inputs without revealing the input, the private set intersection is better approach for both honest and dishonest adversaries.
6. **Threshold cryptography:** Threshold cryptography aims to carry out the cryptographic operations for a set of parties without holding the secret by any of the single party. RSA algorithm is used for the scheme where the basic function is y=xe mod n. RSA is used for encrypting secrets or messages.
7. **Dishonest majority MPC:** In the secure multiparty computation, there can be both honest and dishonest parties. The secure Multiparty computation is secure as long as there is an honest majority. If the adversaries are corrupt more than the majority, new approaches are required for security. For the dishonest majority, there are protocols like GMW oblivious transfer, garbled circuit, Tiny oz and many more protocols.

Benefits Of Secure Multiparty Computation

Let's discuss some benefits of secure multiparty computation:

1. **Trusted third party:** In Secure Multiparty Computation, we can share data in a distributed manner with different organizations without any third party and even the privacy of data will be preserved while sharing data.
2. **Data Privacy:** The private data of organizations can be shared for computation purposes. The concern of data privacy is provided by using secure multiparty computation, which keeps the data in use in encrypted form. Thus, the data is not revealed or compromised.
3. **High accuracy:** Secure Multiparty Computation provides highly accurate results for different computations using cryptography.
4. **Quantum safe:** The data shared between parties is safe against quantum attacks, as the data is broken up and encrypted when distributed among parties for computation.

Limitations Of Secure Multiparty Computation

Secure multiparty computation being used for solving different problems, but there are few limitations. The main limitations are the computational overhead and high communication costs.

1. **Computational overhead:** To provide the security we need to generate the random numbers, the random number generation requires more computation overhead which slows down runtime.

2. **High communication costs:** Distributing the data to multiple parties for computation over the networks leads to higher costs of communication.

Virtual Elections

1.INTRODUCTION

Elections are held everywhere. However, voters have to go to the polling station to cast their vote. The political membership process is exceptionally complex and many things are needed to advance voting. Large arrangements have been made to finish. It involves manual work. Government elections are held by area. To vote, the voter must be available to vote at the polling place. This may reduce voter support; Web-based voting simplifies this undertaking. Voting in Cryptography involves security and a secure system. It is important to implement such a system. This will reduce labor, make ballet easier to use and more productive. Individuals must be available at the location for selection. Cryptography is a system of encoding voter details. In this system, the client will be contacted to upload a security message and voter details during registration. The customer will receive the security part of the security picture via email. This share will be in an encrypted format. The customer can log in to the system to change the details at any time. Only when voting, the customer must upload a security share. If the share is incorrect, the poll cannot be voted on, because the security share is generated using random pixels, so the real picture cannot be predicted. Additionally, the share cannot be retrieved by some other client or disapproved person as it will be securely sent via email. Ballet casting will only be fruitful if the correct share relating to that client is uploaded.

Fraud sends fake messages or sets up fake sites that copy. Phishing is a form of identity online identity theft in which fraudsters manipulate Internet users to submit personal information to illegal websites. Phishing tricks are usually displayed as spam or pop-up and are always difficult to identify. When fraudsters obtain your data, they can use it for all kinds of identity fraud, risking your great reputation and a great name. Fishers will become more sophisticated in the design of their fake sites. Phishing is the data of the types of fraud, so be comfortable with a variety of phishing tricks for you as well as figure out how to prepare for it. The most valid and direct way to secure a system asset is to assign it a unique name and a corresponding password.

Cryptography is the study of protecting data. It has been used as a means of safe communication between people and governmental organizations. Today, cryptography is the foundation of advanced security technologies used to secure data and assets on both open and closed networks. Belief is the process of examining the personality of a person or thing. When you confirm something, the purpose is to check that you have a real deal. It is necessary to implement their methods to determine the level of authorization of the user of the application. Applications often do this by keeping private records that include the names of customers to whom who has access. Databases applications, for example, regularly maintain private approval tables to control the fields in records that a particular

client can view or modify. Few people advocate the benefits it brings, for example, mobility, openness, improved speed and accuracy in the delivery of ballots from home and the same number that it represents are concerned about the crisis, for example, inconsistent entry, breach of mystery, and ambiguity. And a change in the effect of a political race. The project focuses on the prevention of phishing attacks and secure authentication of Internet voting systems using cryptography. Cryptography is an encryption strategy to hide data so that it can be decrypted by human vision if the right key picture is used. Cryptography is the study of protecting data. It has been used as a means of safe communication between people and governmental organizations. Today, cryptography is the foundation of advanced security technologies used to secure data and assets on both open and closed networks. Belief is the process of examining the personality of a person or thing. When you confirm something, the purpose is to check that you have a real deal. It is necessary to implement their methods to determine the level of authorization of the user of the application. Applications often do this by keeping private records that include the names of customers to whom who has access. Databases applications, for example, regularly maintain private approval tables to control the fields in records that a particular client can view or modify. Few people advocate the benefits it brings, for example, mobility, openness, improved speed and accuracy in the delivery of ballots from home and the same number that it represents are concerned about the crisis, for example, inconsistent entry, breach of mystery, and ambiguity. And a change in the effect of the political race. The project focuses on the prevention of phishing attacks and secure authentication of Internet voting systems using cryptography. Cryptography is an encryption strategy to hide data so that it can be decrypted by human vision if the right key picture is used.

2.LITERATURE REVIEW

From the time it takes to the current technological development, there are online voting systems. That was clarified in this document. Develop voting plans to make more efficient voting services avalable with ICT resources than traditional paper-based voting methods. Voters regard themselves as consumers and it is expected that the government will make the voting business more convenient. In the past decade, various forms of electronic voting, especially as additional methods of voting for remote voting, political parties, candidates, the electoral administration, and most importantly to improve the efficiency and promise of the democratic process to the electorate have attracted considerable attention.

It allows voters to access the public algorithm and parameters to confirm their turnout.

Three types of voting systems exist:

1.System of paper voting

The paper voting system is the most common system for voting. Before the electronic voting system is implemented, it will be used. The system of paper ballet includes paper

and sealed ballet. Each voter uses and does not share one ballot. This system's disadvantages are

i) the time it takes;

ii) the speed is low.[16]

2.Electronic voting system

Electronic voting systems are electronic voting devices. A voting machine that uses an electronic voting machine to allow voters to pass on their secret ballots. The inconvenience is I poor computer science individuals cannot vote correctly, (ii) safety threats sensitive, (iii) electricity consumption at polling stations; and (iv) costs.

3.Online voting system

A new platform for secure votes and voting is the online voting system. Online voting systems are a web-based voting system, which transmits votes via a web browser over the internet. Voters from all over the world are eligible to vote online.

Security issues arising from online voting are as follows: In general applications, password protection is high and phishing attacks are not the focus of the application. Website users are not protected efficiently from phishing.

The key proposal for ensuring a secure online polling protocol to meet privacy, anonymity, eligibility, equity, verification, and unique online voting safety requirements

To achieve reliability, eligibility, transparency, accuracy, and uniqueness of the e-vote system, two milliardaires couples have created secure online voting for identities based on cryptographic algorithms.

A secure, end-to-end verifiable, Identity-based blind signature Internet voting system: IEEE, newspapers, 2020; This document has been amended Early vote, elliptical curve cryptography, verifiable end-to-end digital signature, Internet vote system. Batch venerability. Functional digital signature used by the BLS short signature system to protect voting against any changes anonymously to issue a blank ballot to voters. Future of voting: Specifications and feasibility study of verifiable Internet vote from end to end.

Phish-haven-An Efficiency Real-Time AI Phishing URLs Detection System: IEEE, newspapers, 2020; This article changed phishing URLs generated by AI, machine learning, phishing URLs created by people, lexical features, multi- threads, HTML URL encoding. Extracts web pagecontent which is therefore ineffective in computation. Non-proactive method Needs source codes or the website's entire website content. The use of multiple threading technologies on an input unit and output unit may be further enhanced by the incorporation of unattended learning.

SeVEP: Electronic polling system secure and verifiable: 2019 IEEE, journals, Authentication modified, efficiency, electronic polling, malware, security, compliance. Authentication, electronic polling process has resource allocation polling system. Developing a working SeVEP prototype and assessing its scalability and usability for real-world use.[13]

Towards Developing a Secure and Robust Solution for E-Voting using Block-chain: 2019 IEEE, Spring, This paper modified coercion resistance problem, Blockchain, Online Voting process, Developing a Secure Solution for online Election process information and To solve coercion resistance problem to solve using cryptographic algorithms.[18]

End to End Verifiable Electronic Voting System for Shareholders: IEEE 2019, newspaper, this article amended Electronic vote, voting by shareholders, verification end-to- end, zero evidence of knowledge, Decision Diffuse the assumption by Hellman, safety evidence and verifiable electoral process. More generally, voters can leave and leave dynamically within calculation periods if using a smartphone.

Secure Online Voting System Using VC: 2018 IEEE, Spring, this paper modified and using Visual cryptography, security share, voting system. Secure a voting process for using Cryptography task scenario and Improvement in an existing algorithm.

A Scheme for Three-Way Secure and Verifiable E-Voting: 2019 IEEE, journal, This paper modified and using Electronic Voting, Anonymity, Verifiability, and Paillier Cryptosystem, Homomorphic Encryption process on the distributed implementation of Three way Secure and Verifiable Election process.[21]

The Security Issues of The Online Voting System: While inheritance of such items in the source code is not acceptable, the root of the security problems which have occurred have not only been attributed to outsiders (for example voters and attackers) but also to insiders (for example program developers and administrators). These mistakes caused a vote system crash.

The solutions suggested for stopping these attacks have therefore been outlined. To prevent hackers from getting into the voting system over a network we can, for example, develop our system to transmit data without a network. Another example is to limit voting to unique input data to prevent command injection.

4.METHODOLOGY

The rapid development of technologies and Internet popularity lead to the digitization of diverse types of technology, such as electronic commerce, e-democracy, e- government, etc. To minimize costs and red tape in public departments, the contemporary states are seeking to provide people who can participate and benefit from online services by increasing the number of activities associated with this new medium. Electronic voting is one of the most important Internet-related activities. The modern recently We consider

the same methodology as the one we discussed for estimating the operating machine cycles (for example, private and public operations based on Salsa20 algorithm, operations on elliptic curve and pairing).

For example, (1) use of electronic voting can reduce or eliminate undesirable human errors, (2) in addition to its reliability, the online voting system does not need geographical proximity of voters which increase the number of participating voters, (3) e-voting saves a lot of time for voters and reduce a cost when counting the voted ballots.

a.What approach is taken by the author

Once all the nodes of the network are running, a new user can connect to the server. The user registers a non-anonymous user (using Adhar Card, phone, password, etc), and performs the login. The user produces an RSA key pair locally (private key & public key). With the Public-Key server, the user blinds his public key. The public key of the user is blinded and forwarded to the server.[16]

The server Blind Signs the Public-Key blinded from the user and returns it to the user. The user unbinds the Public-Key signed by the server, and now has the Public-Key Blind Signed by the server. The user sends the Public-Key blind signed to the p2p network. The peers verify that the Public- Key Blind Signed is correctly signed by the server, if it is, they add the Public-Key to the Ethereum Blockchain, inside a new block.

b.Our approach

As per recent research RSA method to secure data with blind signature has some flaws and can be cracked using high-end computational devices. So we will be using a more secure Salsa20 security algorithm which is found more to be more secure than an existing algorithm like RSA and AES. Also, Salsa20 is more FAST and lightweight than RSA and AES. Salsa20 is FAST in terms of encrypting and decrypting. This means it can encrypt more messages per cycle compare to RSA and AES. Also, it is lightweight means it requires less computational resources compared to others. Despite such benefits, Salsa20 provides better security.

5.REQUIREMENT ANALYSIS

Before designing a voting system, a complete and detailed set of requirements must be developed. The design requirements for the online voting system are divided into 2 groups during this work: the general one and the system one. The general requirements of any voting system are complied with. The requirements of a system are, on the other hand, essential for the development of a developed system. System-specific requirements, on the other hand, are system-specific demands. Allow system requirements specific to the system:

i.Multi-user: Many voters can vote simultaneously;

ii.Accessibility: System access can be accessed by voters in any location using secure internet and/or mobile devices.

iii.Design of the system framework:

iv.The framework was designed to define the frameworks for the application. The structure for the defined objective is the emerging framework of this design process. The infrastructural model architecture in which models are developed is an integral component of the model design.

v.Based on the earlier (the study was not published), in comparison to cryptography, it can be seen that the cryptographic algorithms of voice data packets using serpent damage or loss of some packages during shipping. No Voice Data Packet Loss occurs when you push to talk to the algorithm salsa20.

vi.And the salsa20 algorithm in another previous study Implementing the security and SMS is found to be relatively short in Salsa20 encryption and decryption.

vii.This experimental test enables analysis of salsa20 Stream cipher algorithm as a cryptographic sound data packet algorithm. From Table 1. we can see that the first packet encryption process is Salsa 20 Faster than the decryption process.

viii.When the Salsa20 algorithms are being implemented to speak, there is a delay of 1.9 seconds, but the push to talk application doesn't change performance.

ix.Encryption & decryption of voice data Packs is successful because the encrypted voice data packets on the Android Smartphone can be heard using Salsa20 algorithms.

x.The bits Modified from the bits of the normal audio data packet with bits of the encrypted audio data packets can be seen from an avalanche effect test. We know that salsa20 has good performance to secure voice data packets based on the Avalanche effect's value.[19]

6.SOFTWARE REQUIREMENT

To test this framework, the software has been developed and deployed. The program is based on Java, Spring Tool, XAMPP server, HTTP SMS gateway. Windows XP, Windows10, and others.

Evaluation and performance checks:

User understanding of the system is developed following experimental use to determine if the core values required in the voting system have been developed in accordance with the online voting system. The following research questions arose in connection with guided

questions whether the developed online voting system meets the desired general safety requirements of voting systems:

a.Can a vote be unreserved? "Integrity" requirement,

b.Is it possible to verify who electors claim to be? "Authenticity" requirement.

c.Is it possible to vote only once by eligible voters through the developedonline voting system? "Democracy Requirement."

d.Can no polling be ensured by the developed online voting system.

liked to the electorate or any other voter? Requirement for Privacy.

7.SYSTEM DESIGN

The system has three modules such as:

1.admin module,

2.client module and

3.server module.

The application requires the user to register and then register with the same username and password. The user must then choose the candidate to vote for. After the user clicks on the

'Vote' buttons, his e-mail id will be transmitted to him, whereas share2 will be downloaded automatically on the server.

Certified users will be shown a captcha that users must use properly. Properly entered into the voter details, the users vote will be successfully registered.

To approximate the computer cycles consumed by operations using a highly verifiable safe online voting system in which each elector is authenticated using a unique identifier provided by the relevant authority and his biometric details (for example, private and public operations based on RSA, operations on elliptic curve and pairing). The appropriate cryptographic operations' notations and the number of computer cycles they absorb. Our system's success in comparison to other systems. The suggested method and system are based on ECDL and GDH problems and use elliptic curve cryptography. The security of the systems is focused on the discrete logarithm problem (DLP) and integer factorization problems, and they are based on the standard RSA public-key cryptosystem

(IFP). The ECC-based operations (scalar multiplication and addition) are more efficient than the RSA-based operations.

4.Requirements for an election system

Researchers also defined a series of specifications for a reliable electronic voting protocol in this framework.

5.Security Requirements

Since the internet seems to be an unstable place, protection plays an essential role in every voting method, particularly e- voting. For the electronic voting framework to function without bugs, it must be applied according to safe design. Despite the system's difficulty of architecture and execution, it seems that certain principles are universally agreed upon as the minimum security specifications for electronic voting.

6.System-Wide requirements

In this section, the system-wide requirements for implementing voting protocols are discussed-Voter conveniently: Voters should be able to vote without consulting the voting authorities and complete the voting procedures with the bare minimum of skills and equipment- Voter mobility: Voters should be able to vote from either location without restriction. The scheme is successful if the number of electors and the authorities involvement in the protocols is equal to the computing and communications resources.

Salsa20 is Daniel J. Bernstein's software-oriented stream cipher. The algorithm can support 128-bit and 256-bit keys. The updated state is used as a 512-bit keystream output following r iterations of the Salsa20/r round function. Each output block is an independent key, nonce, and counter combination and, since there is no link between blocks, the

Salsa20 operation is similar to a block cipher operation in counter mode.

Analysis: Salsa20 underwent significant cryptographic analysis over the years following its publication. While several attacks on smaller versions of the cipher have been found, there is no better attack than an exhaustive key search on either of the Salsa20.

STREAM CIPHER WITH SYMMETRIC SECRET KEY

Key length = 32 bytes

8.THE SYSTEM ARCHITECTURE

For found detection and prevention, we are proposing a new method for detecting phishing websites. Our method uses cryptography and it is based on an Anti-phishing providing authentication scheme. The proposed system can be divided into two Flowcharts one is Registration Flowchart and the second is Login Flowchart.

1.Registration Flowchart

The text of these images is the password for the user when the flowchart is registered. The image is divided between the user and the server. The image is shared. The user will share the user with the login Flowchart for further verification. The detals of voters are also stored as confidential data for the current website database.

2.Login Flowchart

The first user to request a user name (user id) in the login phase is a user name. Then the users requested to enter their share with him. This share is forwarded to the server where every user stays the share and share stored in the website dataset for the security of each user.

Authentication is the process by which the person claims to be. The user id is sent to the server for this purpose and the appropriate password will be retrieved from the database. Now you compare the password of the user and the password of the database. Thus you can check whether the website is a real/secure website or a phishing website using the username or password generated by stacking two shares and also check whether the user is authenticated or not.

3.Datasets

The online voting system uses an online voting database consisting of two datasets:

Login details The table contains registered user/voter logs and passwords with appropriate user names. It has voters/user contacts, telephone, and e-mail addresses as well.

Voting details – The candidate record and the voters who voted for the candidate shall be included. Its main key is the ID field that is also necessary for counting votes.

8.PERFORMANCE ANALYSIS

This compares our system and its performance with the related online voting systems. Assume that Weil is defined by the PBC library's Type-F (BN curve), 256-bit-group pairing, and 512-bit embedded pairing with RSA-identical security level.

Implementing and comparing the performance of our proposed online voting system to the corresponding machine cycle schemes obtained through simulation. We consider

estimation of the operational machine cycles (for example, private and public operations based on curve and pairing, RSA operations on elliptic).

These systems are based on the traditional public-key system Salsa20 based on the discrete logarithm (DLP) system and are safe (IFP). Compared to Salsa20 it can be seen that the ECC (scalar multiplication and supplement) operation is efficient.

9.CONCLUSION AND FUTURE WORK

The cryptographic online voting system overcomes restrictions on the traditional voting system. This system offers more safety and takes a while. There is also no chance of voting fraud. There is a significant reduction in the money spent on security. This method aims primarily to provide full privacy to voters and to ensure that the online voting system is coordinated optimally. The fundamental idea of this system is to use a strong voting authentication security mechanism. Visual encryption encrypts information and can decrypt it without mathematical calculations. People with an internet

connection at home can vote at the polls without any problems. Visual cryptography is used to conduct elections fairly easily and efficiently using these internet-based voting systems since voters can vote from the point of view in which they operate using the online voting system. Various advantages include low costs and increased voting attendance online voting. Online voting offers This online voting system takes careful account of safety and human factors, and in particular, ensures that the electorate has reliable and intuitive indications on the validity of the vote. The system we proposed to provide voters with mutual authentication and choosing with visual encryption.

Single Sign On

What is single sign-on?

Single sign-on (SSO) is an authentication scheme that enables users to log in to a session once, using a single set of login credentials, and gain secure access to multiple related applications and services during that session without logging in again.

SSO is used commonly to manage authentication in company intranets or extranets, student portals, public cloud services, and other environments where users need to move between multiple applications to get their work done. It's also used increasingly in customer-facing web sites and apps – such as banking and e-commerce sites – to combine applications from third-party providers into seamless, uninterrupted user experiences.

How SSO works

Single sign-on is based on a digital trust relationship between a group of related, trusted applications, web sites and services, called *service providers*, and an SSO solution, called

an *identity provider*. The SSO solution is often part of a larger IAM (identity and access management) solution.

In general, SSO authentication works as follows:

1. A user logs into one of the trusted applications - or into a central portal connecting all the trusted applications (such as an employee portal or college student web site) - using SSO log in credentials.
2. When the user is successfully authenticated, the SSO solution generates a session authentication token containing specific information about the user's identity - a username, email address, etc. This token is stored with the user's web browser, or on the SSO or IAM server.
3. When the user attempts to access another of the trusted applications, the application checks with the SSO or IAM server to determine if user is already authenticated for the session. If so, the SSO solution validates the user by signing the authentication token with a digital certificate, and the user is granted access to the application. If not, the user is prompted to reenter log in credentials.

The process can vary depending on several factors. For example, a user who has been idle for a specified period may need to log in when they attempt to access another app. Or, if an authenticated user attempts an app or service that deals with particularly sensitive information, the user may be prompted for an additional authentication factor, such as a code sent to the user's mobile phone or email (see 'Adaptive SSO' below).

Benefits of SSO

Obviously, SSO saves users time and trouble. Take corporate users, for example: Instead of logging into multiple applications multiple times per day, with SSO they are often able be able to log into the corporate intranet or extranet just once for all-day access to every application they need.

But by dramatically reducing the number of passwords users need to remember and the number of user accounts administrators need to manage, SSO strengthens an organizations security posture. Specifically, SSO can

Replace password fatigue with one strong password. Users with lots of passwords to manage often lapse into using the same short, weak passwords - or slight variations thereof - for every application. A hacker who cracks one of these passwords can easily gain access to multiple applications. SSO can often reduce scores of short weak passwords to a single long, complex, strong password that's easier for users to remember - and much more difficult for hackers to break.

Help prevent unsafe password storage habits. SSO can reduce or eliminate the need for password managers, passwords stored in spreadsheets, passwords written on sticky

notes and other memory aids - all of which make passwords easier for the wrong people to steal or stumble upon.

Reduce help desk calls - by a lot. According to industry analyst Gartner, 20 to 50 percent of IT help desk calls are related to forgotten passwords or password resets. Most SSO solutions make it easy for users to rest passwords themselves, with help desk assistance.

Give hackers a smaller target. According to IBM's Cost of a Data Breach 2021 report, compromised credentials were the most frequent initial attack vector for a data breach, accounting for 20% of all data breaches - and breaches that began with compromised credentials cost their victims $4.31 million on average. Fewer passwords mean fewer potential attack vectors.

Simplify management, provisioning and decommissioning of user accounts. With SSO, administrators have more centralized control over authentication requirements and access permissions. And when a user leaves the organization, administrators can remove permissions and decommission the user account in fewer steps.

Help simplify regulatory compliance. SSO meets or makes it easier to meet regulatory requirements around protection of personal identity information (PII) and data access control, as well as specific requirements in some regulations - such as HIPAA - around session time-outs.

SS0 risks

The chief risk of SSO is that if a user's credentials are compromised, they can grant an attacker access to all or most of the applications and resources on the network.

Requiring users to create long and complex passwords - and carefully encrypting and protecting them wherever they're stored - goes a long way toward preventing this worst-case scenario. But most security experts recommend implementing SSO with multi-factor authentication, or MFA. MFA requires users to provide at least one authentication factor in addition to a password - e.g., a code sent to a mobile phone, a fingerprint, or an ID card. Because these additional credentials are ones that hackers can't easily steal or spoof, MFA can dramatically reduce risks related to compromised credentials in SSO.

SSO variations

The SSO scheme describe above - a single log-in and set of user credentials providing session access to multiple related applications - is sometimes called simple or pure SSO. Other types of SSO - or authentication methods similar to SSO - include:

- **Adaptive SSO** initially requires a username and password at log-in, but subsequently requires additional authentication factors or a new log-in when additional risks emerge - such as when a user logs in from a new device or attempts to access particularly sensitive data or functionality.
- **Federated SSO -** more correctly called federated identity management (FIM) - is a superset of SSO. While SSO is based on a digital trust relationship among

applications within a single organization's domain, FIM extends that relationship to trusted third parties, vendors, and other service providers outside the organization. For example, FIM might enable a logged-in employees to access third-party web applications, such as Slack or WebEx, without an additional log-in, or with a simple username-only log-in.

- **Social log-in** lets users use the same credentials they use to access popular social media sites to access third-party applications. Social log-in simplifies life for users. For third-party application providers, it can discourage undesirable behaviors (e.g., false logins, shopping cart abandonment) and provide valuable information for improving their apps.

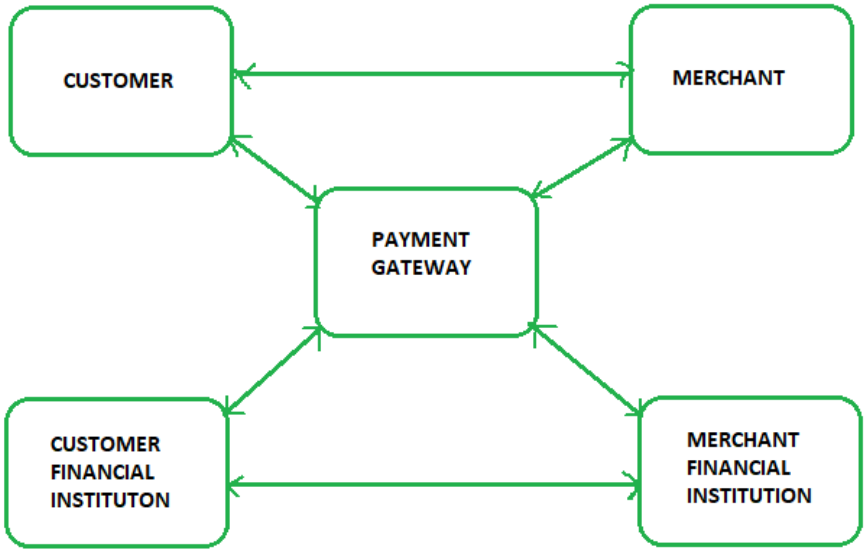
Secure Inter-branch Payment Transactions

Secure Electronic Transaction or SET is a system that ensures the security and integrity of electronic transactions done using credit cards in a scenario. SET is not some system that enables payment but it is a security protocol applied to those payments. It uses different encryption and hashing techniques to secure payments over the internet done through credit cards. The SET protocol was supported in development by major organizations like Visa, Mastercard, Microsoft which provided its Secure Transaction Technology (STT), and Netscape which provided the technology of Secure Socket Layer (SSL).

SET protocol restricts the revealing of credit card details to merchants thus keeping hackers and thieves at bay. The SET protocol includes Certification Authorities for making use of standard Digital Certificates like X.509 Certificate.

Before discussing SET further, let’s see a general scenario of electronic transactions, which includes client, payment gateway, client financial institution, merchant, and

merchant financial institution.



Requirements in SET :

The SET protocol has some requirements to meet, some of the important requirements are :

- It has to provide mutual authentication i.e., customer (or cardholder) authentication by confirming if the customer is an intended user or not, and merchant authentication.
- It has to keep the PI (Payment Information) and OI (Order Information) confidential by appropriate encryptions.
- It has to be resistive against message modifications i.e., no changes should be allowed in the content being transmitted.
- SET also needs to provide interoperability and make use of the best security mechanisms.

Participants in SET :

In the general scenario of online transactions, SET includes similar participants:

1. Cardholder – customer
2. Issuer – customer financial institution
3. Merchant

4. Acquirer – Merchant financial
5. Certificate authority – Authority that follows certain standards and issues certificates(like X.509V3) to all other participants.

SET functionalities :

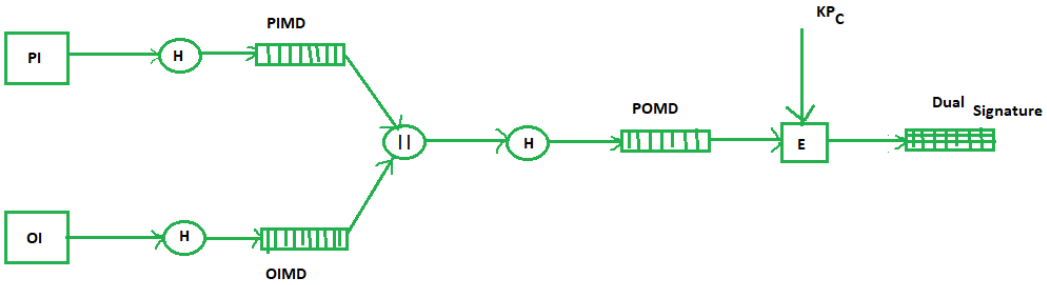
Provide Authentication

- Merchant Authentication – To prevent theft, SET allows customers to check previous relationships between merchants and financial institutions. Standard X.509V3 certificates are used for this verification.
- Customer / Cardholder Authentication – SET checks if the use of a credit card is done by an authorized user or not using X.509V3 certificates.

- Provide Message Confidentiality: Confidentiality refers to preventing unintended people from reading the message being transferred. SET implements confidentiality by using encryption techniques. Traditionally DES is used for encryption purposes.
- Provide Message Integrity: SET doesn't allow message modification with the help of signatures. Messages are protected against unauthorized modification using RSA digital signatures with SHA-1 and some using HMAC with SHA-1,

Dual Signature :
The dual signature is a concept introduced with SET, which aims at connecting two information pieces meant for two different receivers :
Order Information (OI) for merchant
Payment Information (PI) for bank

You might think sending them separately is an easy and more secure way, but sending them in a connected form resolves any future dispute possible. Here is the generation of dual signature:



Where,

PI stands for payment information

OI stands for order information

PIMD stands for Payment Information Message Digest

OIMD stands for Order Information Message Digest

POMD stands for Payment Order Message Digest

H stands for Hashing

E stands for public key encryption

KPc is customer's private key

|| stands for append operation

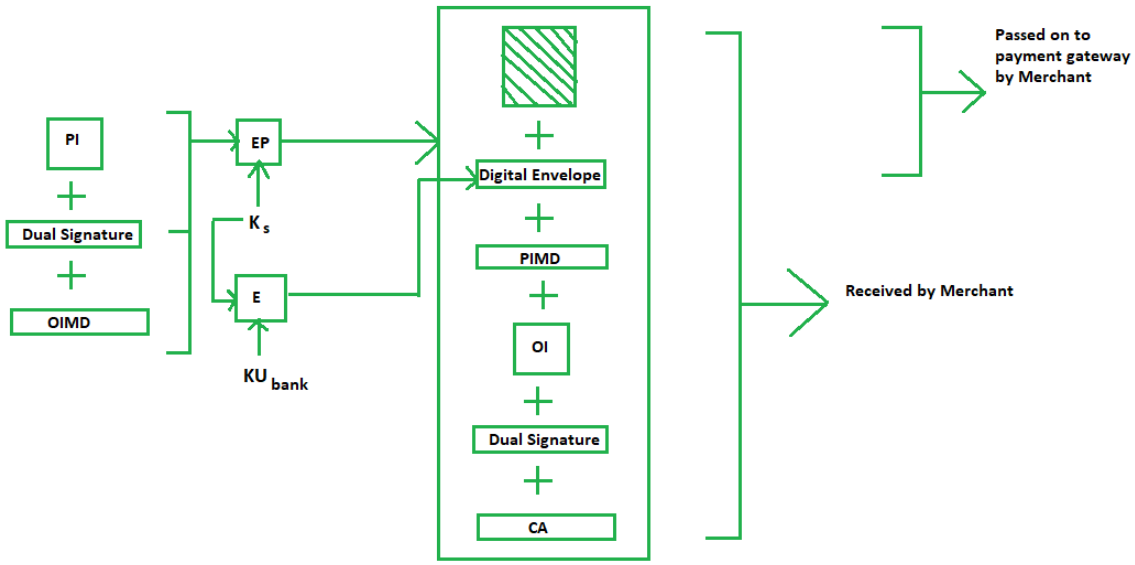
Dual signature, DS= E(KPc, [H(H(PI)||H(OI))])

Purchase Request Generation :

The process of purchase request generation requires three inputs:

- Payment Information (PI)
- Dual Signature
- Order Information Message Digest (OIMD)

The purchase request is generated as follows:



Here,

PI, OIMD, OI all have the same meanings as before.

The new things are :

EP which is symmetric key encryption

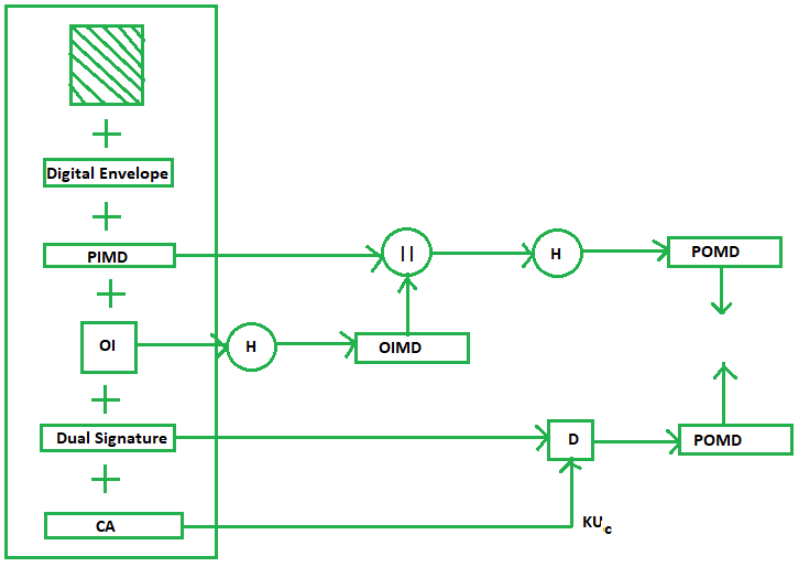
Ks is a temporary symmetric key

KUbank is public key of bank

CA is Cardholder or customer Certificate

Digital Envelope = E(KUbank, Ks)

Purchase Request Validation on Merchant Side : The Merchant verifies by comparing POMD generated through PIMD hashing with POMD generated through decryption of Dual Signature as follows:

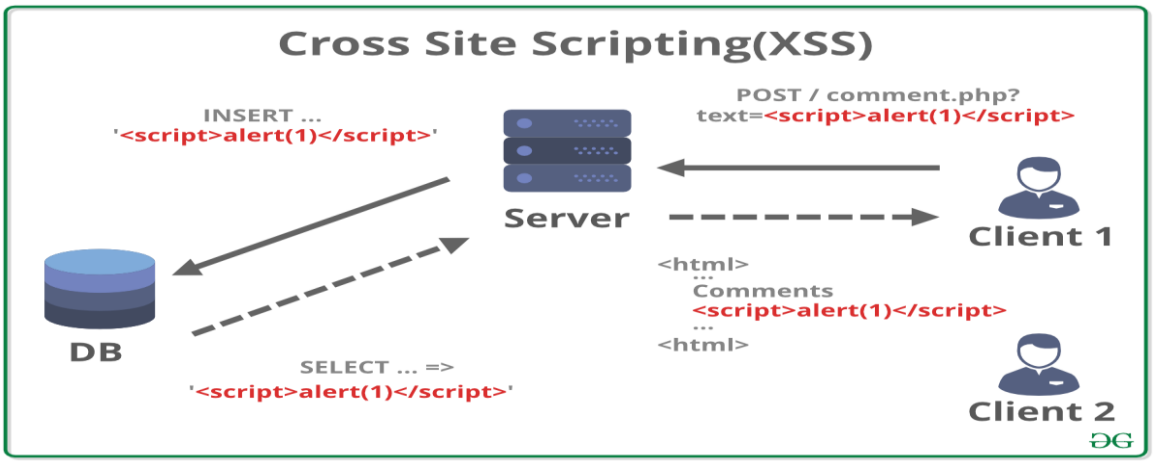


Since we used Customer's private key in encryption here we use KUC which is the public key of the customer or cardholder for decryption 'D'.

Payment Authorization and Payment Capture : Payment authorization as the name suggests is the authorization of payment information by the merchant which ensures payment will be received by the merchant. Payment capture is the process by which a merchant receives payment which includes again generating some request blocks to gateway and payment gateway in turn issuespayment to the merchant.

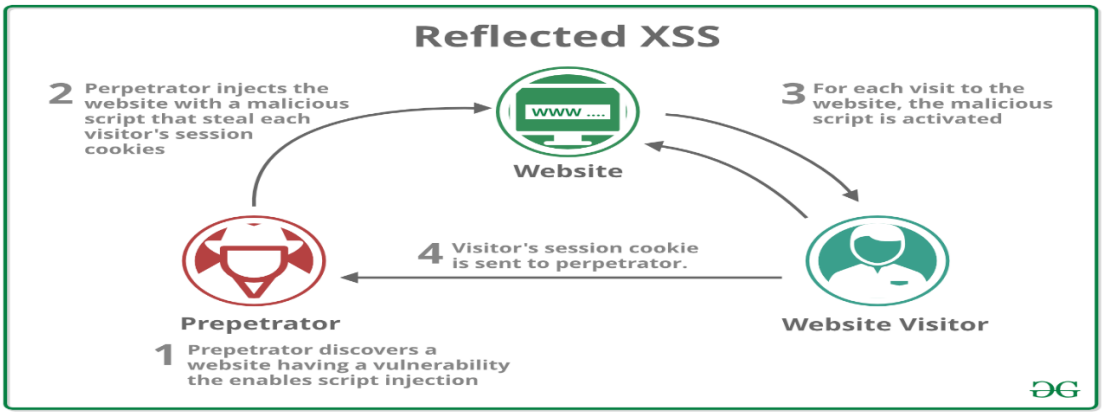
Cross site Scripting Vulnerability

Cross Site Scripting (XSS) is a vulnerability in a web application that allows a third party to execute a script in the user's browser on behalf of the web application. Cross-site Scripting is one of the most prevalent vulnerabilities present on the web today. The exploitation of XSS against a user can lead to various consequences such as account compromise, account deletion, privilege escalation, malware infection and many more.

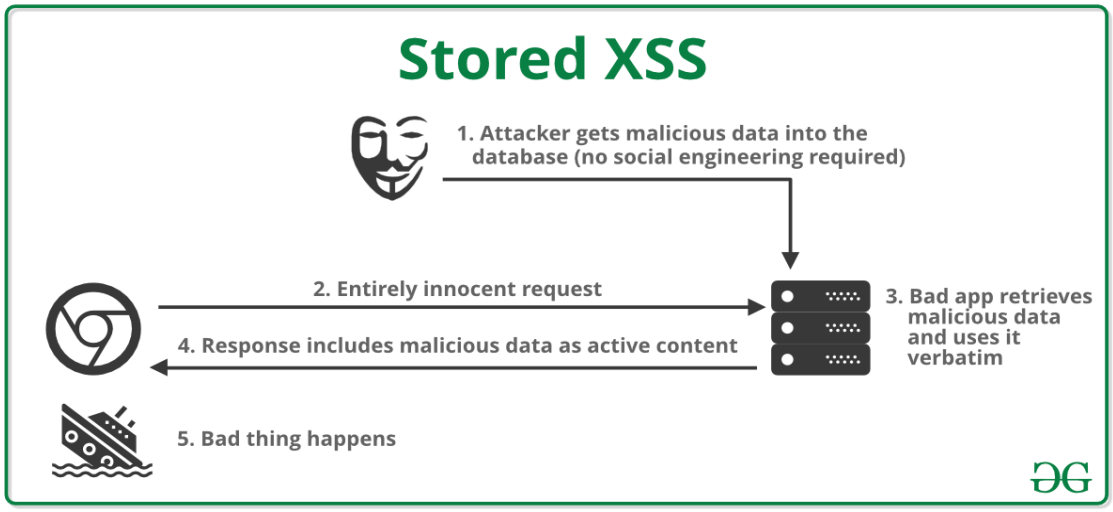


In its initial days, it was called CSS and it was not exactly what it is today. Initially, it was discovered that a malicious website could utilize JavaScript to read data from other website's responses by embedding them in an iframe, run scripts and modify page contents. It was called CSS (Cross Site Scripting) then. The definition changed when Netscape introduced the Same Origin Policy and cross-site scripting was restricted from enabling cross-origin response reading. Soon it was recommended to call this vulnerability as XSS to avoid confusion with Cascading Style Sheets(CSS). The possibility of getting XSSed arises when a website does not properly handle the input provided to it from a user before inserting it into the response. In such a case, a crafted input can be given that when embedded in the response acts as a JS code block and is executed by the browser. Depending on the context, there are *two types* of XSS –

Reflected XSS: If the input has to be provided each time to execute, such XSS is called reflected. These attacks are mostly carried out by delivering a payload directly to the victim. Victim requests a page with a request containing the payload and the payload comes embedded in the response as a script. An example of reflected XSS is XSS in the search field.



Stored XSS: When the response containing the payload is stored on the server in such a way that the script gets executed on every visit without submission of payload, then it is identified as stored XSS. An example of stored XSS is XSS in the comment thread.



There is another type of XSS called *DOM based XSS* and its instances are either reflected or stored. DOM-based XSS arises when user-supplied data is provided to the DOM objects without proper sanitizing. An example of code vulnerable to XSS is below, notice the variables *firstname* and *lastname* :

php

```
<?php

  if(isset($_GET["firstname"]) && isset($_GET["lastname"]))
  {

     $firstname = $_GET["firstname"];
     $lastname = $_GET["lastname"];

     if($firstname == "" or $lastname == "")
     {

        echo "<font color=\"red\">Please enter both fields...</font>";
```

```
    }

    else
    {

       echo "Welcome " . $firstname. " " . $lastname;

    }
}
?>
```

User-supplied input is directly added in the response without any sanity check. Attacker an input something like –

Html and it will be rendered as JavaScript. There are two aspects of XSS (and any security issue) –

<script> alert(1) </script>

1.Developer: If you are a developer, the focus would be secure development to avoid having any security holes in the product. You do not need to dive very deep into the exploitation aspect, just have to use tools and libraries while applying the best practices for secure code development as prescribed by security researchers. Some resources for developers are – a). OWASP Encoding Project : It is a library written in Java that is developed by the Open Web Application Security Project(OWASP). It is free, open source and easy to use. b). The "X-XSS-Protection" Header : This header instructs the browser to activate the inbuilt XSS auditor to identify and block any XSS attempts against the user. c). The XSS Protection Cheat Sheet by OWASP : This resource enlists rules to be followed during development with proper examples. The rules cover a large variety of cases where a developer can miss something that can lead to the website being vulnerable to XSS. d). Content Security Policy : It is a stand-alone solution for XSS like problems, it instructs the browser about "safe" sources apart from which no script should be executed from any origin.

2.Security researchers: Security researchers, on the other hand, would like similar resources to help them hunt down instances where the developer became lousy and left an entry point. Researchers can make use of – a). CheatSheets – 1. XSS filter evasion cheat sheet by OWASP. 2. XSS cheat sheet by Rodolfo Assis. 3. XSS cheat sheet by Veracode. b). Practice Labs – 1. bWAPP 2. DVWA(Damn vulnerable Web Application)

3. prompt.ml 4. CTFs c). Reports – 1. Hackerone Hactivity 2. Personal blogs of eminent security researchers like Jason Haddix, Geekboy, Prakhar Prasad, (Portswigger) etc.

DescriptiveQuestions:

a.2MarksQuestions

1.Whatistheneedofpublickeyringandprivatekeyring?

Publickeyringisoneofthedatastructureswhichisusedtostorethepublickeysoftheotherparticipants

PrivateKeyringisadatastructurewhichisusedtostorethepublicandtheprivatekeysoftheowneralone.

2.MentionthebenefitsofIPSec.

a.Itprovidesstrongsecuritythatcanbeappliedtoalltrafficcrossingtheperimeter.

b.IPSecinafirewallisresistanttobypass.

c.IPSecisbelowthetransportlayerandsoistransparenttoapplications.

d.IPSecistransparenttousers.

3.ListouttheservicesprovidedbytheIPSec.

a.Accesscontrol

b.Connectionlessintegrity

c.Dataoriginauthentication

d.Rejectionofreplayedpackets

e.Confidentiality

f.LimitedtrafficflowconfidentiaIity

4.NametheprotocolsthatprovidesecurityinIPSec.

i.Authenticationheader

ii.Encapsulatingsecuritypayload

5.Whatissecurityassociation?

It is a one way relationship between a sender and a receiver that affords securityservicestothetrafficcarried onit.

6.Definetransportandtunnelmode.

i.Transport mode provides protection primarily for upper layer protocols. Transportmode protection extends to the payload of an IP packet. Transport mode is used forend-to-endcommunicationbetweentwohosts.

ii.Tunnel mode provides protection to the entire packet. The entire packet (originalpacket) plus security fields is treated as the payload of new outer IP packet

with anew outer IP header. Here the packet travels through a tunnel from one point of anIP network to another.

7.Whydo weneedanantireplayservice?

Anti replay service is required in order to avoid the duplicate packets (created bytheopponent)whichmaycausedisruptionintheservice.

8.WhatistheneedpfpaddinginEncapsulatingSecurityPayload(ESP)?

i.Ifanencryptionalgorithmrequires theplaintexttobeamultipleofsomenumberofbytes,thepaddingfieldisusedtoexpandthep laintexttotherequiredlength.

ii.ESPformatrequiresthatthepadlengthandthenextheaderfieldsberightalignedwithina32-bitword.Thepaddingfieldisusedtoassurethisalignment.

iii.Additionalpaddingmaybeaddedtoprovidepartialtrafficflowconfidentialitybyconcealingtheactuallength ofthepayload.

9.Howthesecurityassociationsbecombined?

Itcanbedoneintwoways:

i.transportadjacency

ii.iteratedtunneling

10.Definetheterms:connectionandsession

i.Aconnectionisatransportthatprovidesasuitabletypeofservice.Everyconnectionisassociated with onesession.

ii.Asessionisanassociationbetweenaclientandaserver.Onesessionmayhavemultipleconnections.

11.Whatisthe useofthechangecipherspecprotocol?

This protocol consists of a single message which consists of a single byte with a value '1'.The sole purpose of this message is to cause the pending state to be copied into the currentstate,whichupdatestheciphersuite(cryptographicalgorithm)tobeusedontheconnection.

12.MentionthephasesoftheHandshakeprotocol.

a. Phase1–establishsecuritycapabilities
b. Phase2–serverauthenticationandkeyexchange
c. Phase3– clientauthenticationandkeyexchange
d. Phase4–finish

13.Whatisthe needofDualsignature?

Thepurposeof thedualsignatureistolinktwomessagesthatareintendedfortwodifferentparticipants.

e.g.,DS=EKR_C[H(H(PI)||H(OI))]where

DS–Dualsignature

EKR_C– Encryptionusingtheprivatekeyofthecardholder(sender)

H–Hashfunction

PI–paymentinformation

OI–orderinformation

14.Classifytheintruders.

i.Masquerader

ii.Misfeasor

iii.Clandestineuser

15.Howthepasswordfilesbeprotected?

Thepasswordfilescanbeprotectedinoneofthetwoways:

i.oneway encryption ii.accesscontrol

16.Definefirewall.

Firewallistheinwhichprotectsthepremisesnetworkfrominternetbasedattacksandtoprovidea singlechokepointwheresecurityandauditcanbeimposed.

17.Whatarethedesigngoalsofthefirewall.

- Alltraffic frominsidetooutside,andviceversa,mustpassthroughthefirewall.
- Onlyauthorizedtraffic,asdefinedbythelocalsecuritypolicy,willbeallowedto pass.
- Itisimmunetopenetration.

18.Listoutthelimitationsofthefirewall.

- Itcannotprotectagainstattacksthatbypassthefirewall.
- Thefirewalldoesnotprotectagainstinternalthreats.
- Itcannotprotectagainstthetransferofvirusinfectedprogramsorfiles.

19.Whatarethetypesoffirewall?

Packetfilteringfirewall

Applicationlevelgateway

Circuitlevelgateway

20.DefineBastionhost.

ABastionhostisasystemidentifiedbythefirewalladministratorasacriticalstrongpointinthenetwor ksecurity

21.Listoutthefirewallconfigurations.

- Screenedhostfirewall,singlehomedbastion
- Screenedhostfirewall,dualhomedbastion
- Screenedsubnetfirewall

22.Definethetworulesformulti-levelsecurity.

- Noreadup– asubjectcanonlyreadonobjectoflessorequalsecuritylevel.Thisisreferredto assimplesecurity property.
- Nowritedown– asubjectcanonlywriteintoanobjectofgreaterorequalsecuritylevel.Thisisre ferredto as‘*’property.

23.DefineTrojanhorseattack.

The Torjan horse attack begins with a hostile user, named X, gain legitimate access into the systemandinstallsboththetorjanhorseprogramandaprivatefiletobeusedintheattackasa‘ba ckpacket’.

Xgivesread/writepermissiontoitselfandgivesY

(authorized user) write-only permission. X now indicates Y to invoketorjan horse program, byadvertising it as a useful utility. When the program detects that it is being executed by Y, it reads thesensitivecharacterstringfromY’sfileandcopiesit intoX’sbackpocketfile.

Definevirus.Specifythetypesofviruses?

Avirusisa programthatcaninfectotherprogrambymodifyingthemthemodificationincludesacopy ofthevirusprogram,whichcanthengoontoinfectotherprogram,

Types:

1) Parasiticvirus

2) Memory-residentvirus

3) Bootsectorvirus

4) Polymorphic virus

5) Stealth virus

b.10MarksQuestions

1. A)BrieflyexplainabouttransportlayersecurityandPadding.
   B)WithaneatdiagramexplaintheoperationofSSLandSSHRecordProtocol.
2. A)ListthefiveimportantfeaturesofIKEkeydeterminationalgorithm

B)Whatarethedesigngoalsforafirewall?AlsomentionitsLimitations.

3. A)whataredifferentTypesofViruses?Explain.
   B)ExplainindetailaboutIPSecurityPolicy.
4. A)Identifyanddescribedifferentapproachesofintrusiondetection.
   B)Whatisthestructureofvirus?
5. A)Writeshortnotesonnonmaliciousprogramerrors.
6. Writeashortnoteson: A)Intrusiondetection b)TrustedSystem
7. A)Explainthevarioustypesoffirewall.

   B)Explainthedifferentfirewallconfiguration.

8. A)Describethevariouswaysofcombiningthesecurityassociations.

   B)Whatarethecapabilities,limitationsanddesigngoalsoffirewalls?

9. A)DifferentiateSSL&TLS

   B)Describeindetail,thepaymentprocessingofSET.

10. A)Explainindetail,theHandshakeprotocolinsecuresocketlayer

    B)Explaintheconceptofpasswordprotectionsystem