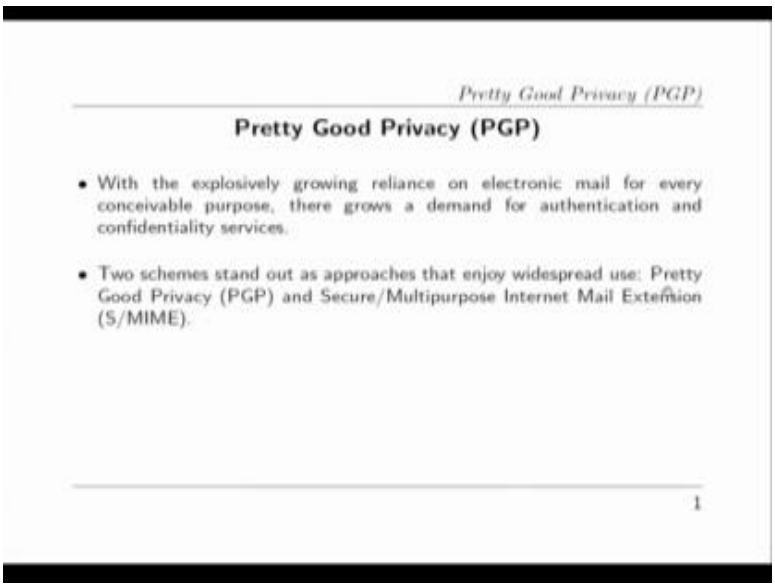
Internetwork Security
Prof. Sourav Mukhopadhyay
Department of Mathematics
Indian Institute of Technology, Kharagpur

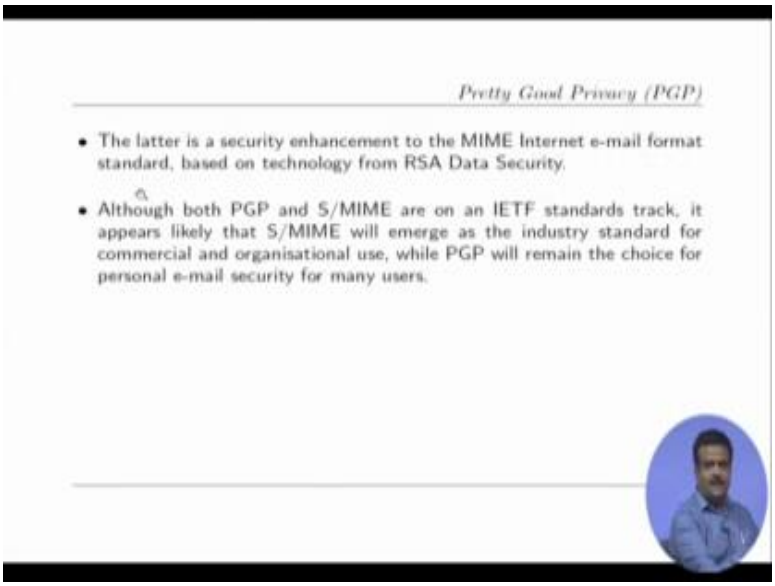
Lecture – 60
Pretty Good Privacy (PGP)

(Refer Slide Time: 00:28)



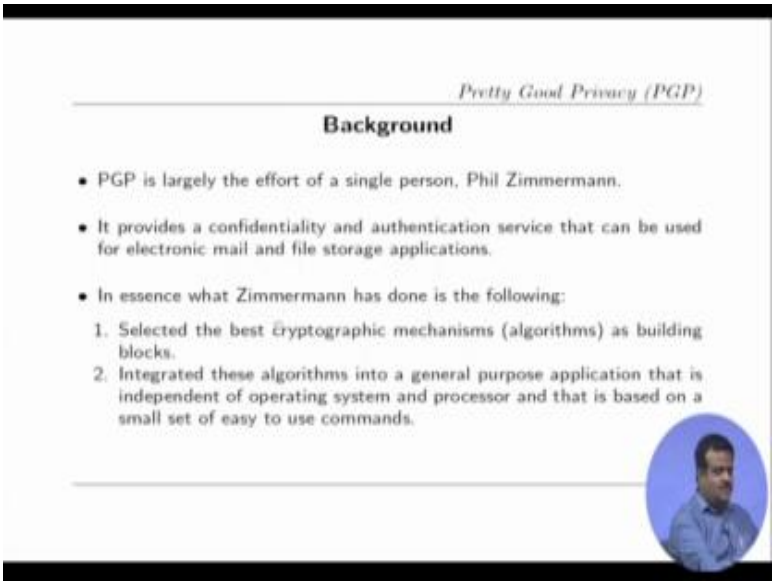
We talk about Pretty Good Privacy which is basically PGP. So, this is a scheme to provide the authenticity and privacy in the e-mail; electronics mail. So, with the explosively growing reliance on electronic mail for every conceivable purpose there grows a demand of authentication and confidentiality services. So, there are basically 2 schemes available, one is the pretty good privacy; PGP and another one is secure multipurpose internet mail extension which is basically S slash MIME, this is basically extension version of MIME.

(Refer Slide Time: 01:08)



The latter is a security enhance of the MIME that S slash MIME internet email for this is the enhance of the MIME internet email format standard based on the technology form RSA data security. Although both PGP and S MIME are if EITF standards track, it appears likely that MIME will emerge as the industry standard for commercial and organizational use, while PGP will remain the choice for personal email security for many users.

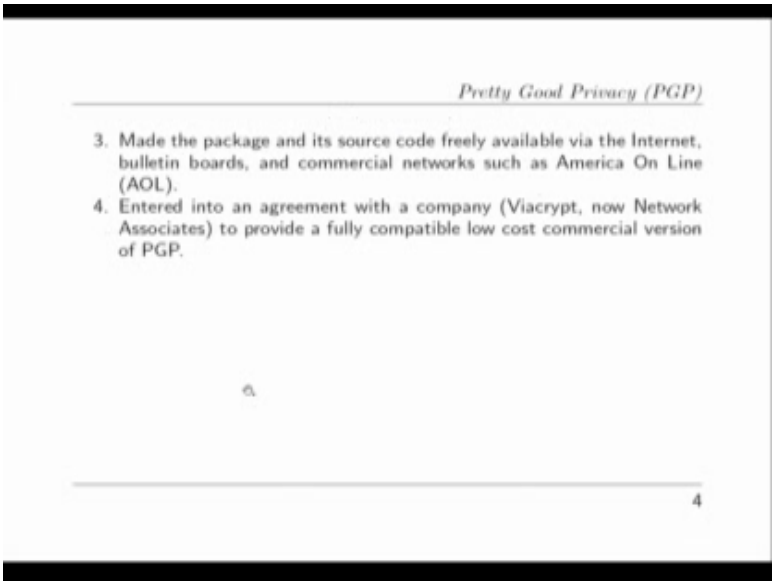
(Refer Slide Time: 01:54)



And usually the PGP is available freely. So, this is one of the points to become PGP more popular. So, let us talk about background of this PGP. So, PGP was basically designed by Zimmermann and it provides a confidentiality and authentication service that can be used for electronic mail and also the file storage applications. The essence of this PGP has done in the though Zimmermann done this in a following way. So, Zimmermann selected the best cryptographic algorithm as a building block and used those I mean integrated those algorithms into the general purpose application that is independent of operating system and the processor and that is based on a small set of easy use command.

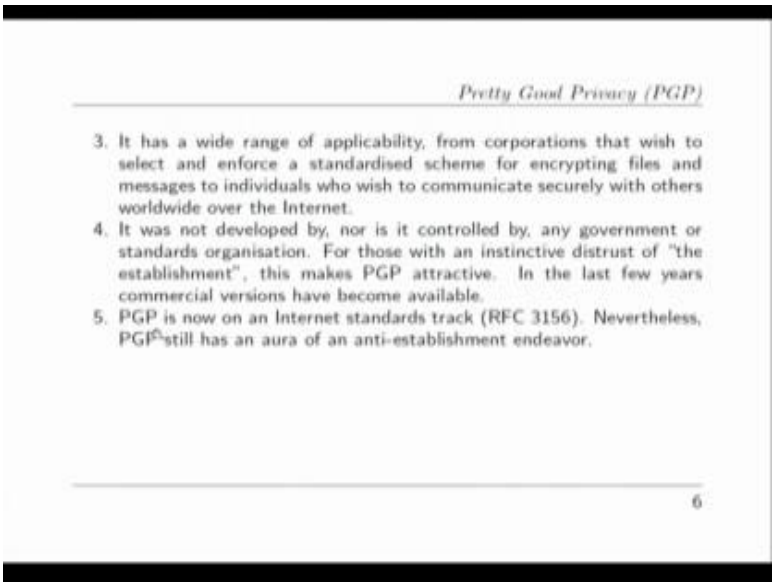
So, basically the idea is to choose the best cryptographic algorithm like AES, DEAES, RSA, hash function, AE7. So, then we use those and make a integrate those algorithm into a general purpose application that is independent of the operating system and processor and that is based on a small set of easy to use command.

(Refer Slide Time: 03:14)



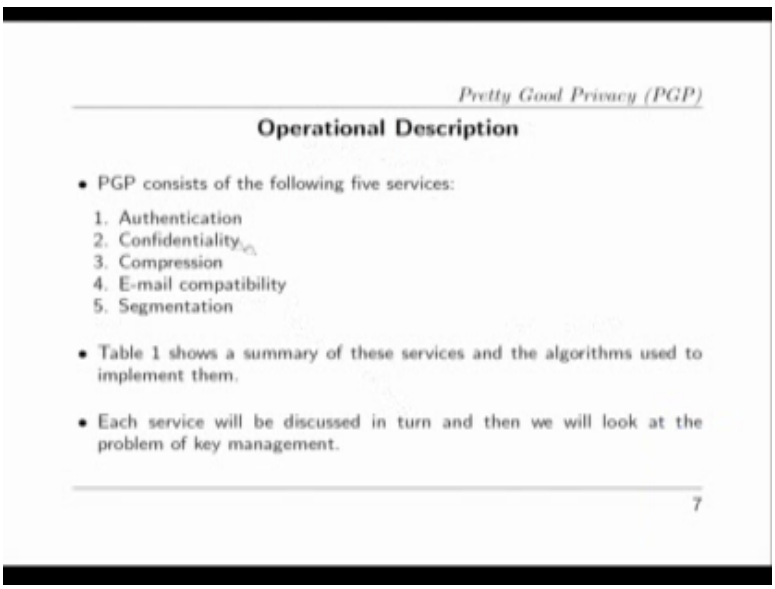
Made the package and its source code freely available via the internet, so what Zimmermann did? So, he made this package freely available over the internet and entered into the agreement with the company via script and network associates to provide the fully compatible, low cost commercial version of the PGP. So, there is commercial version of PGP which is low cost, but it is the version then there is free version available over the internet. So, that is why it became more popular.

(Refer Slide Time: 03:54)



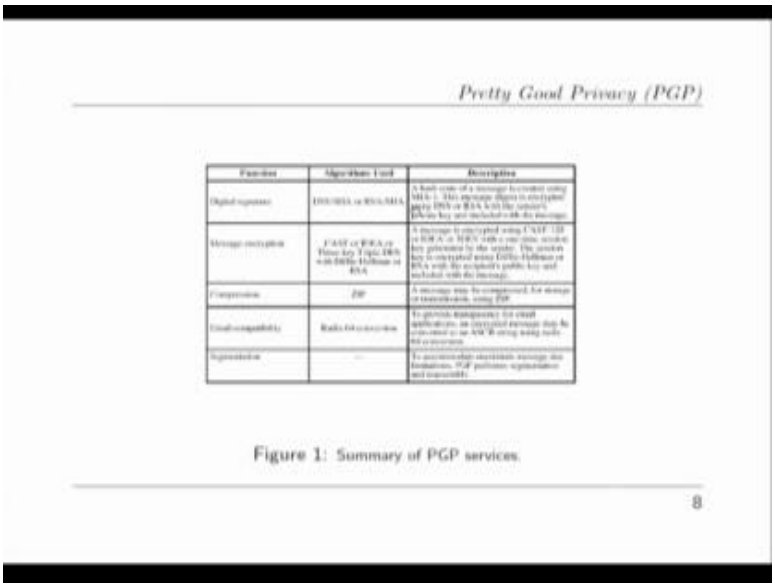
It has wide range of applicability from the corporations that wish to select the enforce a standard scheme for encrypting files and messages to individuals who wish to communicate securely with other worldwide over the internet, it was not developed by nor it is controlled by any government or standards organization. So, it is free. So, anybody can be put a comment on it, anybody can like a Linux operating system, for this with an instinctive distrust on the establishment, this make PGP attractive. In the last few years commercial version has become available. So, now, PGP is on an internet standard track RFC 3156.

(Refer Slide Time: 04:50)



This is the operation description of the PGP, PGP consists of basically the 5 services, 1 is authentication and then next one is confidentiality, another one is compression, then the email compatibility and then segmentation.

(Refer Slide Time: 05:19)

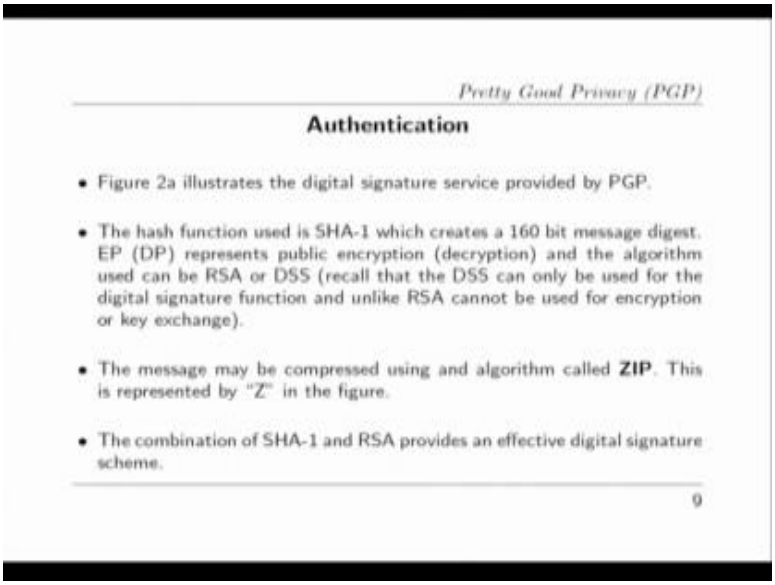


The table 1 shows; this table shows the summary of the services and the algorithm used to implement them. For example, for digital signature, we use the digital signature standard or SHA and also we use RSA and description hash code or the message is created using the SHA 1, this message digest is encrypted using the DAs or RSA with the sender private key and this the digital signature on it and after applying the hash code, we will compress the message into smaller size and then we encrypt this using the sender; sender's secret key. So, this is digital signature on the message then the message encryption; message encryption in message encryption we use basically cast IDEA or 3 key triple DES with Diffie-Hellman or RSA.

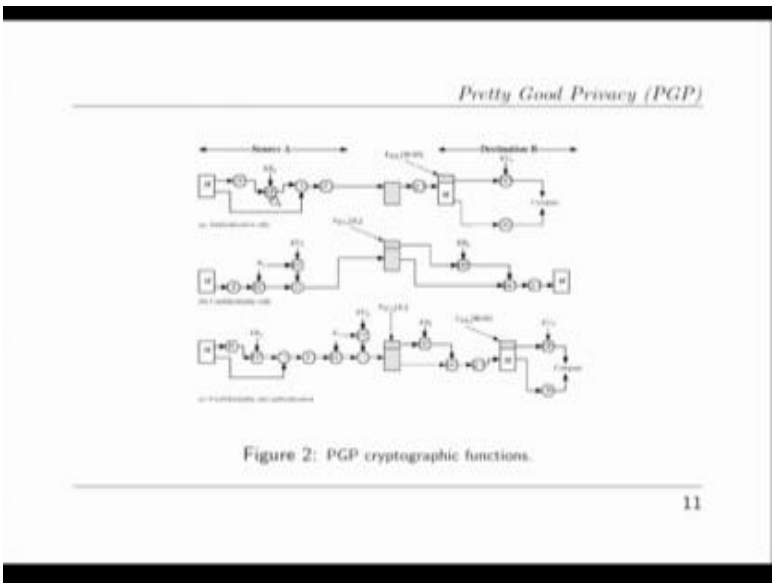
This message encrypted using cost 128 or IDEA or 3 DA, triple DAs with 1 time session key is generated by the sender, the session key is encrypted using Diffie-Hellman or RSA with the receipt public key and the, include these with the message and then also another functional key is the compression, it is referred zip, the message may be compressed or for storage or transmission using zip and email compatibility. So, this is the algorithm we use that radix 64 conversion and then the segmentation we do this is the

one of the service by PGP. So, each service is discussed in. So, we will discuss the service in more details.

(Refer Slide Time: 06:59)



(Refer Slide Time: 07:06)



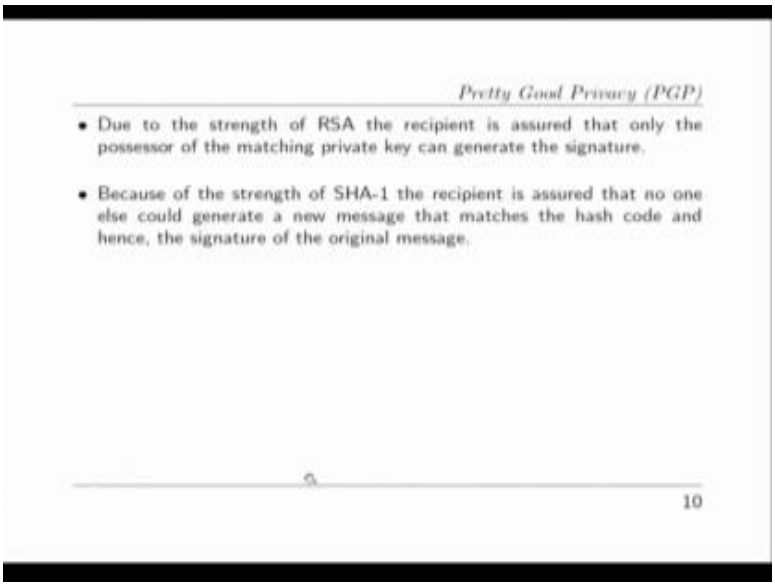
Let us talk about the authentication service by the PGP. So, let us go to this figure, there is figure. So, this is telling us the authentication service of the PGP. So, we have a message. So, M so, this is the authentication only, there is no confidentiality. So, we have the message M. So, we apply the hash function on it to compress this message then we apply the encryption algorithm, this is the public key encryption EP.

So, we apply the encryption algorithm using the secret; using the public key and then we send it to the then we compress then we concatenate with this with M and then you zip it and send it then after receiving this, this is source and this is destination, after we receiving this we unzip it and we get this whole thing back and this is basically encrypted version of the; this is basically the digital signature. So, we are signing, we are signing the message, we are signing on the hash function hash code of the message using the secret key of the secret key of a. So, this is the k r a is the secret key of a and k r b is the private key k u b is the private key of b. So, k r a is the secret key of, k r a is the secret key of a and k u a is the private key of a.

So, after doing the unzip, so what we do? We have then we have this each of encrypted the digital signature on this. So, we just do the description algorithm on this, public key description algorithm using the public key of a. So, it will give back us the M of h of m then we have M, we compute h again. So, we compare these 2, this is matching then the authentication is done. So, this is the way we apply the authentication.

Let us come back. So, that figure illustrates the digital signature the service provided by PGP the hash function is used here is SHA 1 which is which creates a 64, 160 bit message digest EP and EP DP represents the public key encryption or decryption and the algorithm used here is RSA or digital signature standard which is basically use the Elgamal crypto system and this zip is denoted by now this is the compression zipping is denoted we zip the function the folder.

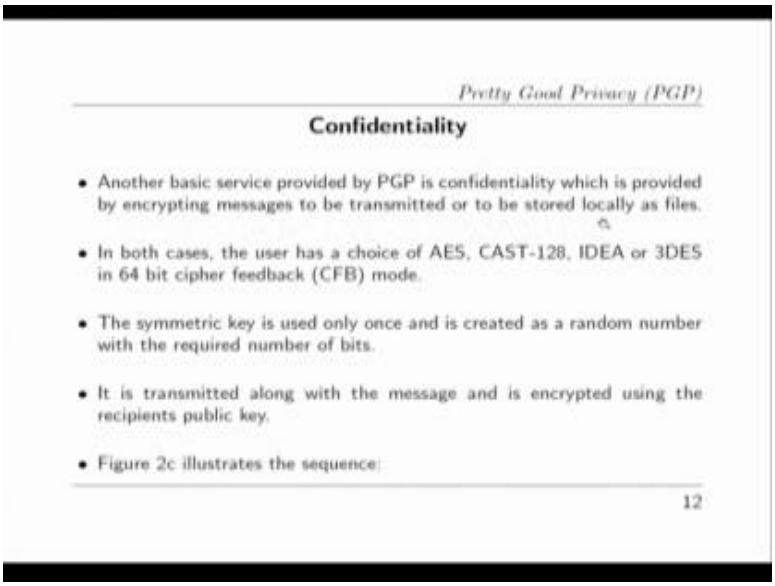
(Refer Slide Time: 09:57)



This is the message along with the hash code and this denoted by Z in the figure and this is the due to strength of RSA, the recipient is assured that only the possess only the possessor of the matching private key can generate the signature and because the strength of the SHA 1, the recipient is assured that no one else could generate the new message that match the hash code and hence the signature of the original message. So, this is the (Refer Time: 10:24), this is the collision; collision resistance.

So, it is very difficult to generate another function is which is having the same hash function. So, that is collision resistance. So, this is the first applications authentication and this is here is the confidentiality. So, here we have the aim. So, we apply this Z on it to compress then we apply then we then we sign it on sigh on it then we apply again this case we use the public key encryption using the public key of b. So, that b can decrypt it and then this combination we are sending to the b and then up on receiving this. So, this has to be decrypt first and then we get this and this we again using the decryption algorithm; symmetric description algorithm, we will get back this and then we unzip it and get the message. So, this is the confidentiality.

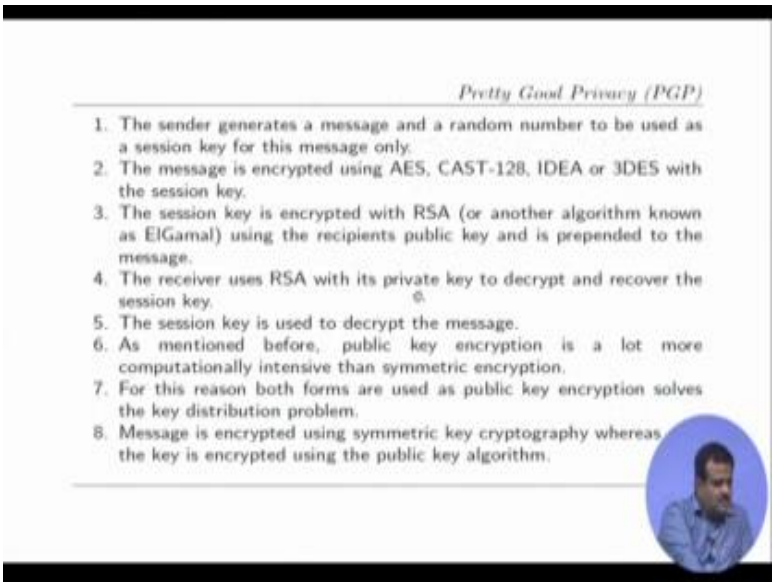
(Refer Slide Time: 11:35)



This is another basic service provided by PGP is confidentiality which is provided by encrypting the message to be transmitted or to be stored locally in the files. So, this PGP is applied to the as a file storage mechanism. So, in both the cases, the user has a choice, choose the AES or this CAST-128 bit, IDEA or triple DES or in the 64 bit cipher CFB modes of operation, the symmetric key is used only once and is created as random number with the required number of bits it is transmitted along with the message and it is encrypted using the recipient public key.

So, this is symmetric key encryption which is encrypted using the k s and this k s; transmitted k s is encrypted using the receiver public key and transmitted along with this. So, that receiver can able to get this case then only receiver after getting this case receiver will decrypt this and get the zip message and then unzip it and get the message M, so the confidentiality so this is only the confidentiality.

(Refer Slide Time: 13:01)



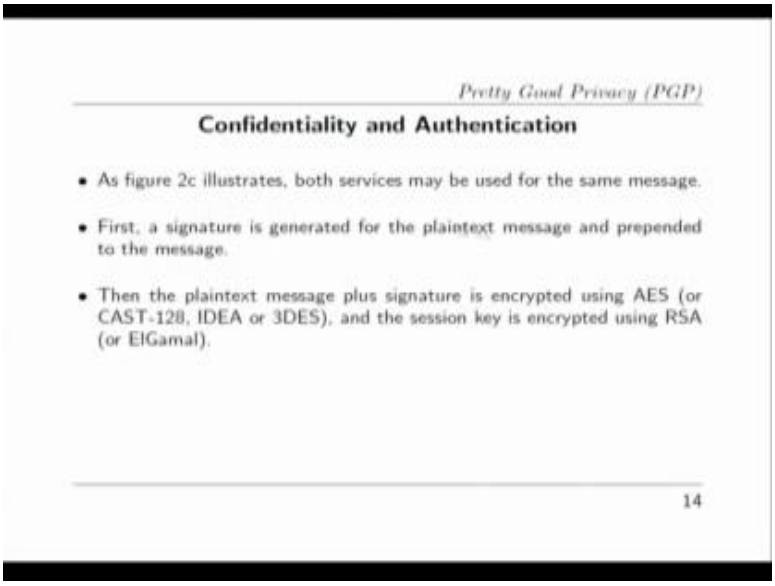
Now, suppose we want both the confidentiality and the authenticity. So, for that what we have to do? So, the sender generates a message and the random number to be used for the as a session key for this message only. So, let us go back to the slide picture.

So, this is this figure is both confidentiality and authentication what we are doing here we have the message we first apply the hash function on this message then we are using the public key encryption using this is the digital signature, we are signing the; a is signing on this message on the hash code of this message. And then we are appending this M with the signed on the hash code of the message and then we compress this zip and then we do the symmetric key encryption using a randomly generated key which is k s and k s need to send Bob, k s need to send b in order to decrypt it and the k s is send to b by encrypting this k s using the public key of b. So, that only b can decrypt it. So, these we are just combining and send it to b.

So, after receiving this b will first decrypt this and get the k s. So, after getting k s b we will just apply the symmetric key decryption and get this and then apply the zip unzip then we got this then we do the; we check this authentication this is the authentication checking. So, this is the way we both use the public key and private key setup in order to get the both confidentiality and the authenticity. So, this is both confidentiality and authenticity.

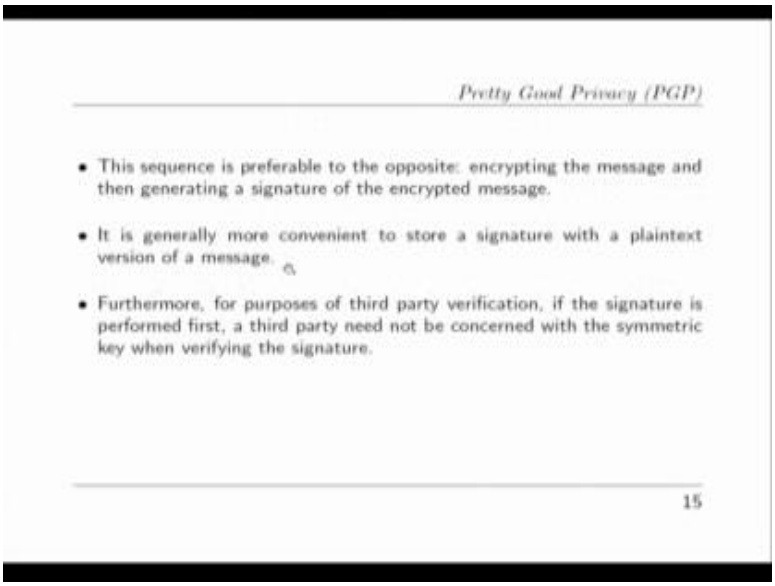
Here we use AES or triple DES for the symmetric key encryption and for the public key encryption we can use RSA Elgamal. So, any standard cryptographic based on session key is used for decryption the message and which is randomly chosen by the sender a as mentioned before public key encryption is a lot is a lot more computationally intensive than the symmetric encryption. So, public key encryption is more costly than the symmetric key encryption because public key encryption involves if you use the RSA it involves a m to the power e or if it is Elgamal it involve the decrypt power into the power some index. So, it is very much expensive on the other hand the symmetric key encryption we use the s box we use permutation we use the XOR. So, those are the very fast operation so, but this is not much secured than the public key encryption, but public key encryption is more expensive.

(Refer Slide Time: 16:18)

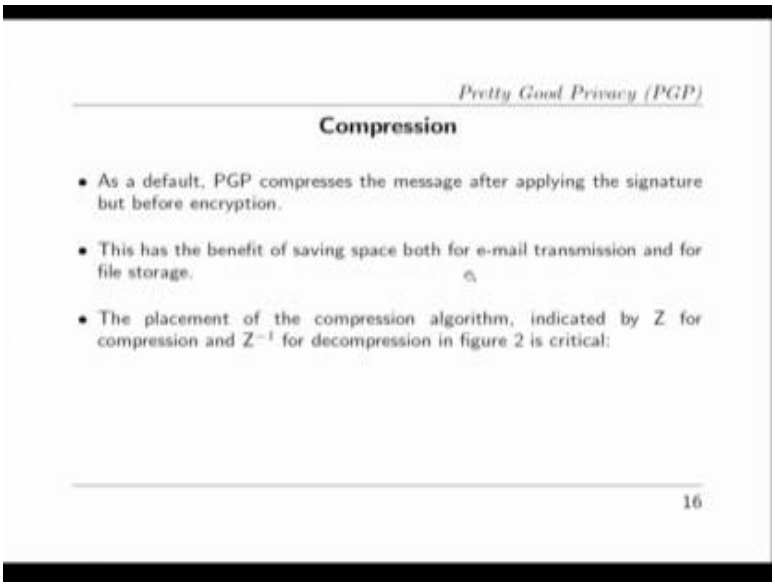


For this reason both forms of used public key encryption solves the key distribution problem. So, this is the confidentiality and authentication which we describe just now in figure 2 c. So, this we discuss this sequence is preferable the opposite encryption the message we can have some other variant of it.

(Refer Slide Time: 16:26)



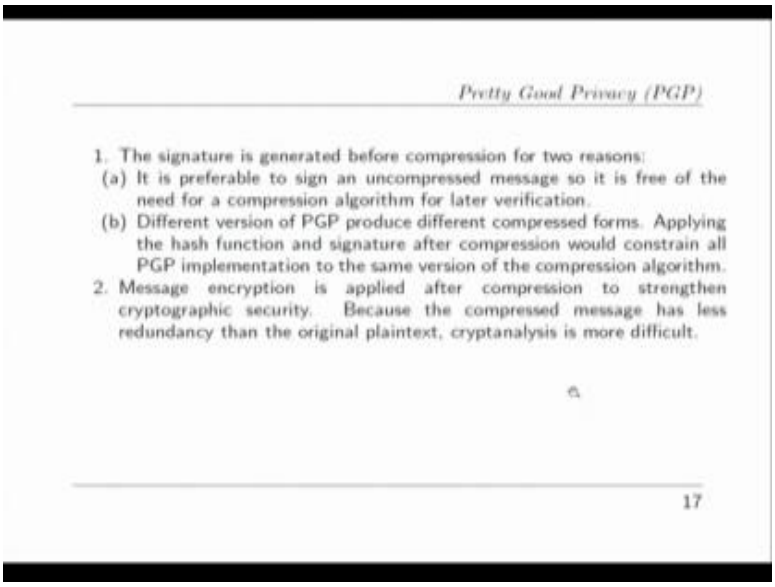
(Refer Slide Time: 16:35)



Now let us talk about of compression service, this is also another service of PGP as a default PGP compresses the message after applying the signature, but before the encryption. So, the zip; this has a benefit of saving space both email transmission and for file storage because if we can compress before the encryption because encryption if we use the public key encryption, public key encryption is expensive. So, if we have a very long message, say 120 bit message. So, if you just use the public key encryption on the 120 bit message.

So, if we use RSA we have to do M to the power of e. So, it is very expensive. So, instead of that if you can compress this message to a fixed length say just 128 bit. So, 124 bit, 128 bit then if we can then we can apply the RSA this expensive operation exponentiation. So, that is why we just apply the compression function and it also has the benefit of saving the storage for file storing the placement of the compression algorithm indicated by Z for compression and Z inverse for decompression as we have discussed.

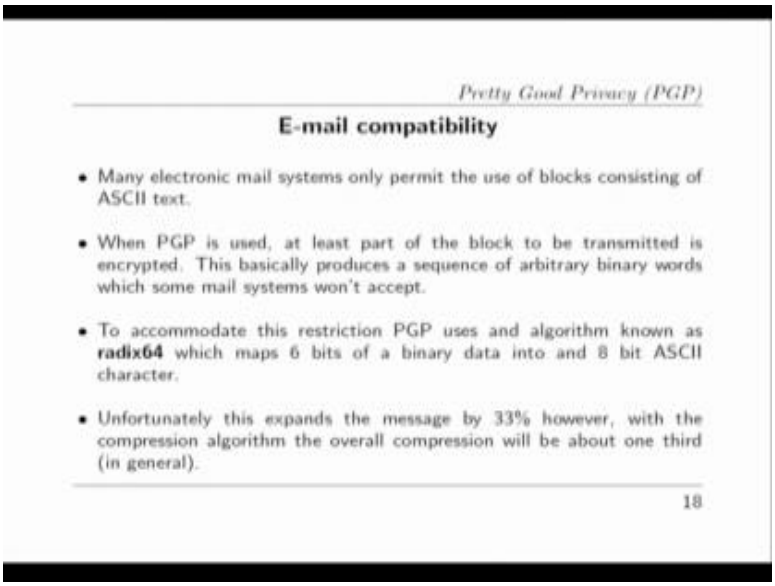
(Refer Slide Time: 17:57)



The signature is generated before compression for 2 reasons because it is preferable to sign an uncompressed message. So, it is free of the need of compression algorithm for later verification and different version of PGP produce different compression. So, forms applying the hash function and signature after the compression would constrain all PGP implementation to the same version of the compressed algorithm.

So, that is why we apply the compression we generate the signature before the compression and the message encryption is applied after the compression to strengthen cryptographic security because the compressed message has less redundancy than the original plaintext cryptanalysis is more difficult and as well as the computationally it is easy faster because if we have less size complex message then exponentiation is much more faster.

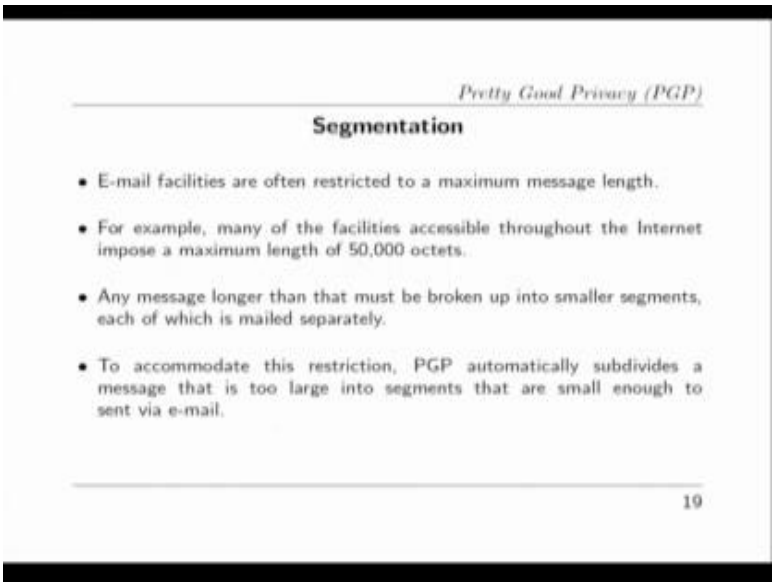
(Refer Slide Time: 18:56)



Then next service of PGP is called email compatibility. So, many many electronic mail systems only permit the use of blocks consistence of ASCII text. So, when PGP is used at least part of block to be transmitted is encrypted this basically produces a sequence of arbitrary binary words which some mail systems would not accept.

To accommodate this restriction, PGP uses an algorithm known as radix 64 which maps 64 bits of binary data into 8 bit ASCII character because in many many email systems they are comfortable with the ASCII text; they are not comfortable with the binary data. So, for that we have algorithm which called radix 64 because we are doing the encryption decryption. So, after that we are getting the binary bits 0 ones. So, this radix 64 will convert this take this binary bits to the ASCII characters, but unfortunately this expands the message by 33 percent; however, with the compression algorithm the overall compression will about one-third of in general.

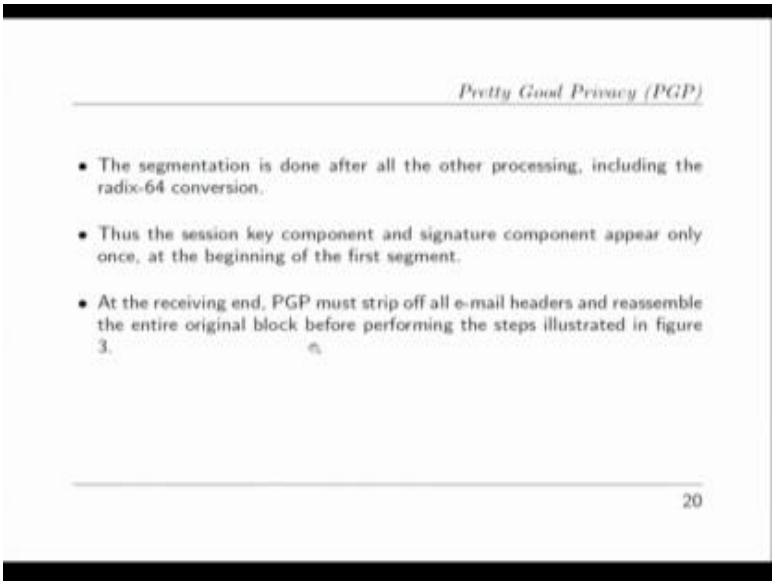
(Refer Slide Time: 20:20)



Another functionality of this PGP is segmentation. So, email facilities are often restricted to a maximum message length for example, many of the facilities accessible throughout the internet imposed a maximum length of octets 550000 octets any message longer than that must be broken off into a smaller segments.

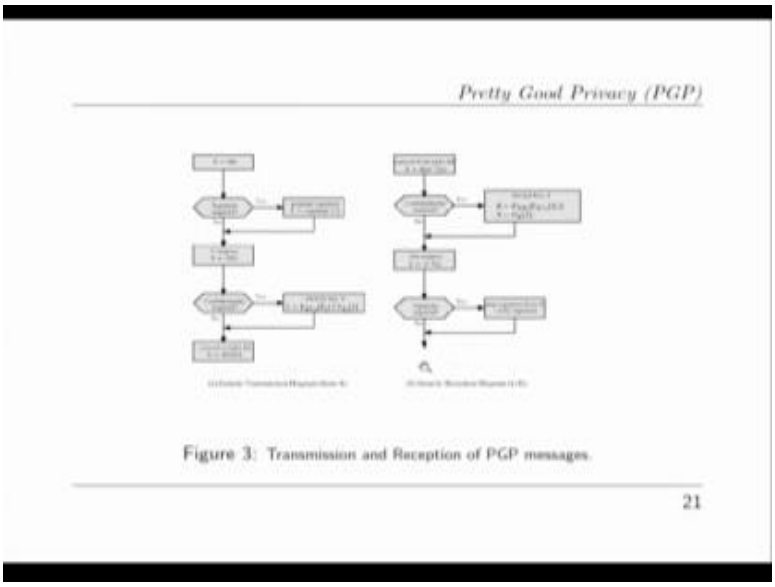
Qe need do the segmentation each of which is mailed separately to accommodate this restriction PGP automatically subdivides a message that is too large into segments and that are small enough to sent via email.

(Refer Slide Time: 21:13)



This is segmentation is required because of size restriction of the message to be sent. So, this is one service the segmentation is done after all processing including the radix 64 conversion that the session key component and the digital component appear only once at the beginning of the first segment. So, at the receiving end, PGP must strip off all email headers and reassemble the entire original block before performing the steps illustrated in figure 3.

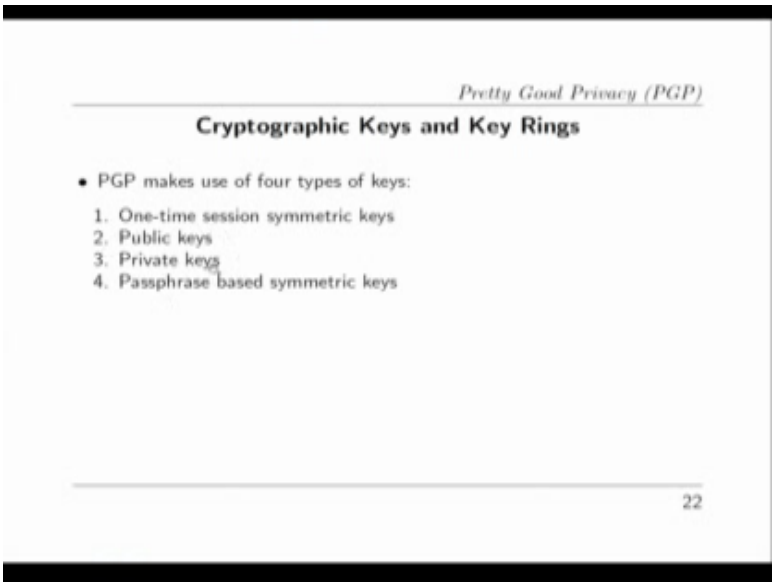
(Refer Slide Time: 21:40)



This is the step, it is performing suppose, this is the x is the file which need to be sent. So, is the signature is required if a if no, then we go for the compression if yes we generate the signature and we put the signature and the original message and then we go for the compression signature is done before compression and then after compression we ask for the confidentiality if we need the confidentiality of the message then we go for this encryption and then otherwise we go for the direct this and we convert this into ASCII because email is email system is comfortable with ASCII.

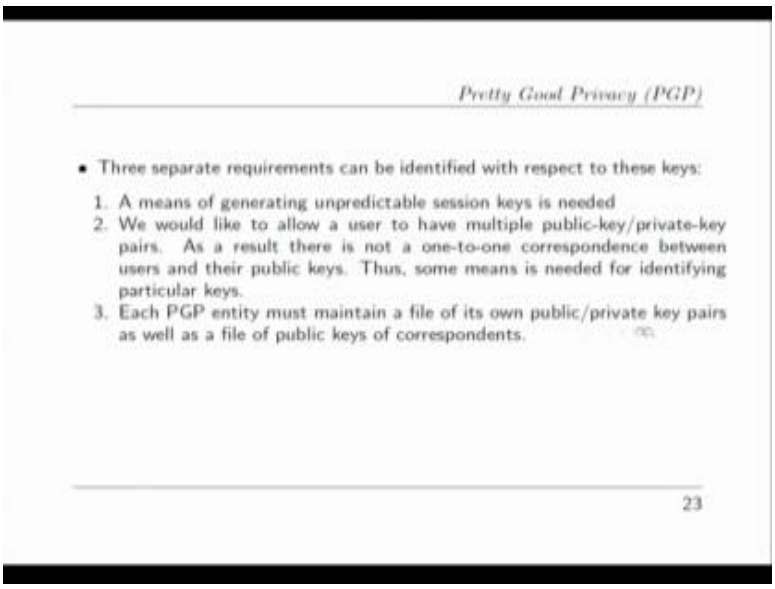
So, this is the generic transmission diagram form a and from b. So, recipient what recipient is doing. So, the recipient is getting the ASCII and recipient is converting from ASCII to binary by this radix 64 the reverse I mean ASCII to binary and then the confidentiality is required if it was used then decrypt it and get back here and then again apply the decompose Z inverse and get it and whether it was signed if it is signed then the signature is checked otherwise this is the file x which is sent.

(Refer Slide Time: 23:13)



Now, talk about cryptographic key and key rings. So, PGP makes use of four type of keys - one is one-time session symmetric key that is the case which is randomly chosen and which needs to be sent to the receiver and second one is public key that public key and private key pair and the private key and the passphrase based symmetric key.

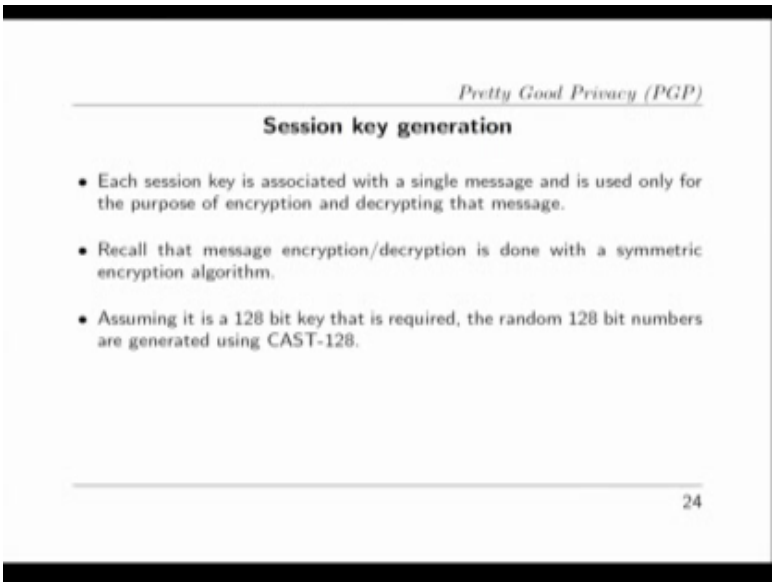
(Refer Slide Time: 23:38)



This 3 separate requirement can be identified with respect to these 3 keys a means to means of generating the unpredictable session keys is needed because the session key is chosen by a that case. So, the case is chosen randomly by a and we would like to allow

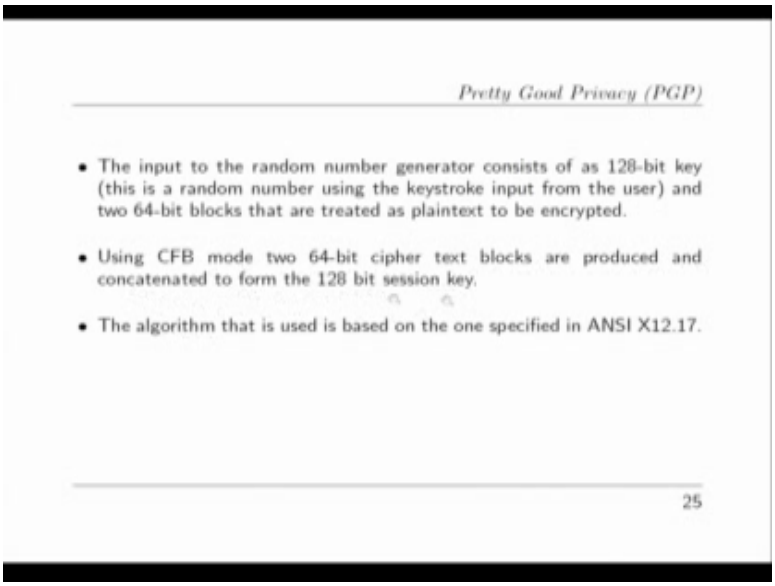
the user to have the multiple private key public key pairs as a result there is no there is not a one to one corresponding between users and his public keys the some means is needed for identifying the particular key. So, each PGP entity must maintain a file for each own public and private key pairs as well as a file of public keys of correspondents.

(Refer Slide Time: 24:26)



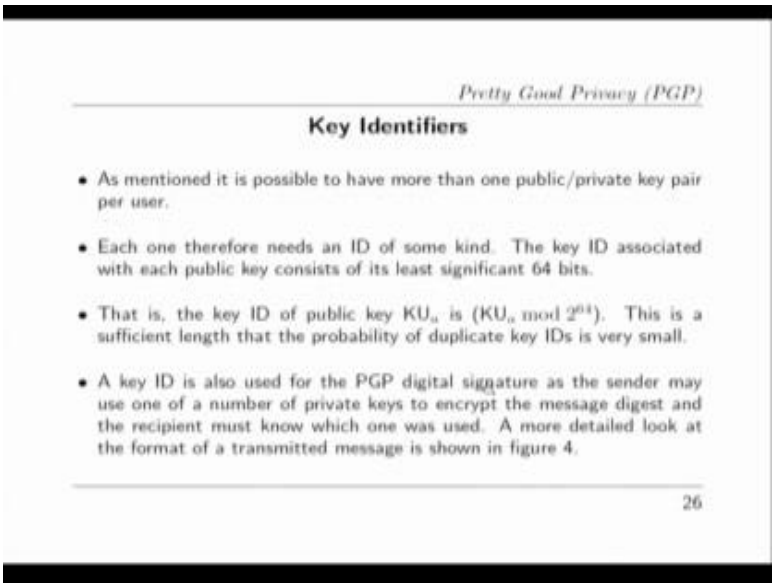
Now the question is how one can generate the session key each session key is associated with the single message. So, once that session is gone we destroy the key. So, this is this session key is basically for a specific message for the specific session. So, once that session is gone, we will no more use that key. So, that is called of session key. So, each session key associated with a single message and it is used only for the purpose of encryption and decryption for that particular message recall the message encryption is done with the symmetric key the; so, we use the case and this is maybe 128 bit random key if you are using AES.

(Refer Slide Time: 25:11)



Now the input of random number generator consists of 128 bit key because if you are using the AES, AES needs 128 bit key using CFB modes 2, 64 bit cipher text blocks are produced and concatenated to the form of 64 bit session key.

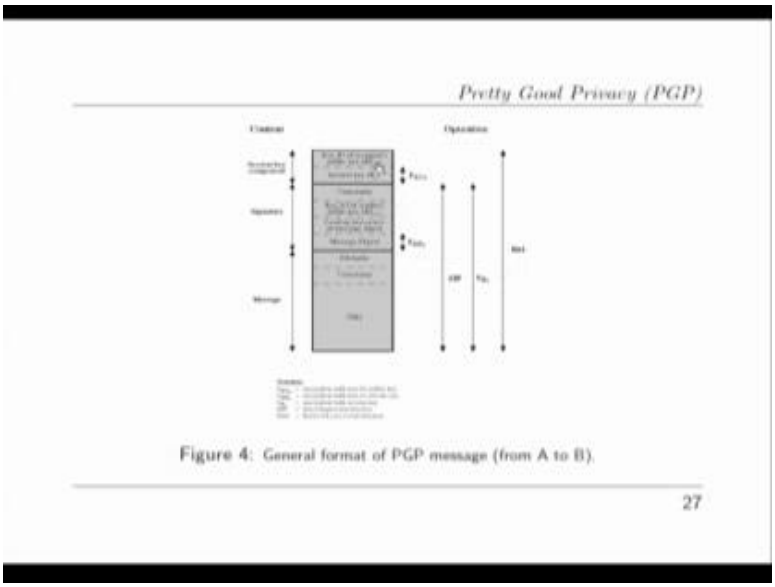
(Refer Slide Time: 25:31)



If it is the session key generation then the key identifier how we can identify the key as mentioned it is possible to have more than one public key private key pair each one therefore, need to have an identity of some kind the ID is associated with the public key consists of at least signification 64 bits. So, this is maybe we can use the ID, ID is

associated with the public key and that is the ID; ID of the public key u a. So, UU KU, this is the public key of a. So, mod could be the 64. So, we want to make this each of this key is 64 bit.

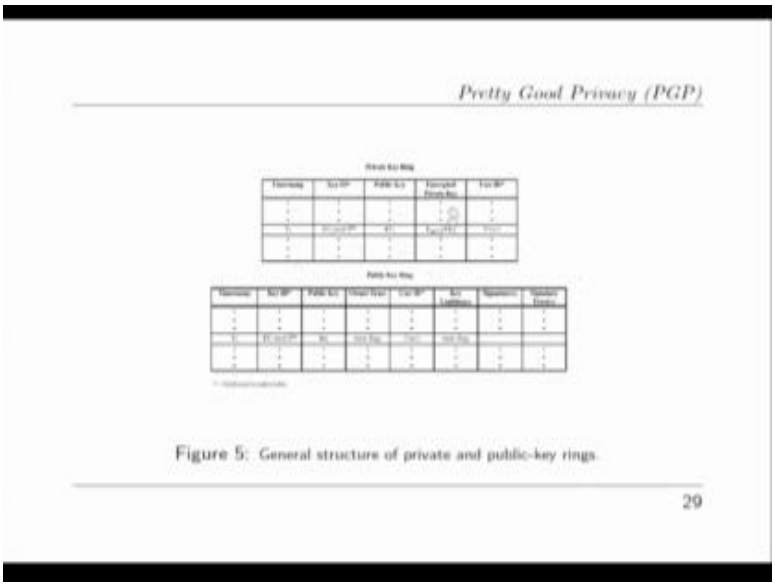
(Refer Slide Time: 26:19)



This is the sufficient length because this is public key encryption. So, this we use here. So, this is the general format of PGP message. So, this is the contain. So, this is the session key is kept here and this is the ID of the recipient and this is the timestamp and key ID of the sender and this is the digital this is the signature part and this is the message part where which contains the file name timestamp and the data and this is the encryption of e e k k u b k u b means the public key of b. So, like this, so this part we usually zip and this part is e k s, we encrypt this part; we encrypt this part using the session key, this is the symmetric key encryption and then we apply this r x 64 to convert this into the into the ASCII.

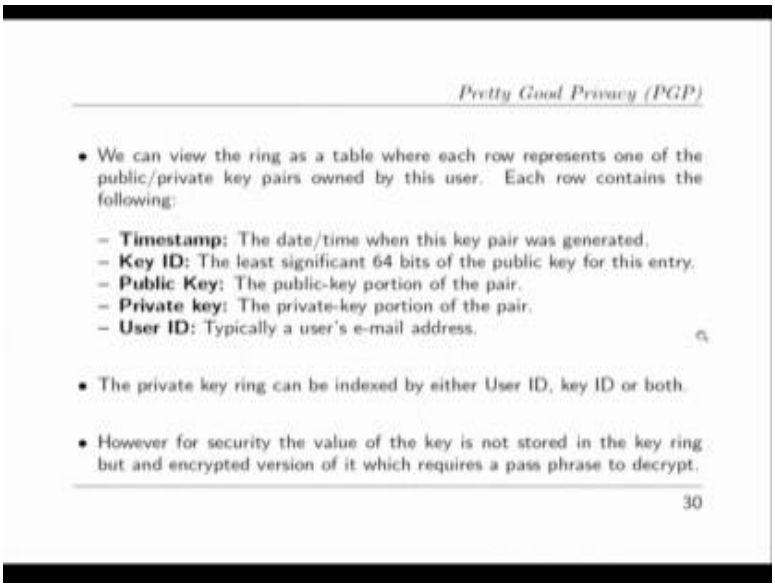
Then we have the key rings. So, the key IDs are critical to the operation of PGP. So, this is the general structure of private and private and the public rings. So, here we have the timestamp we have this key IDs.

(Refer Slide Time: 27:26)



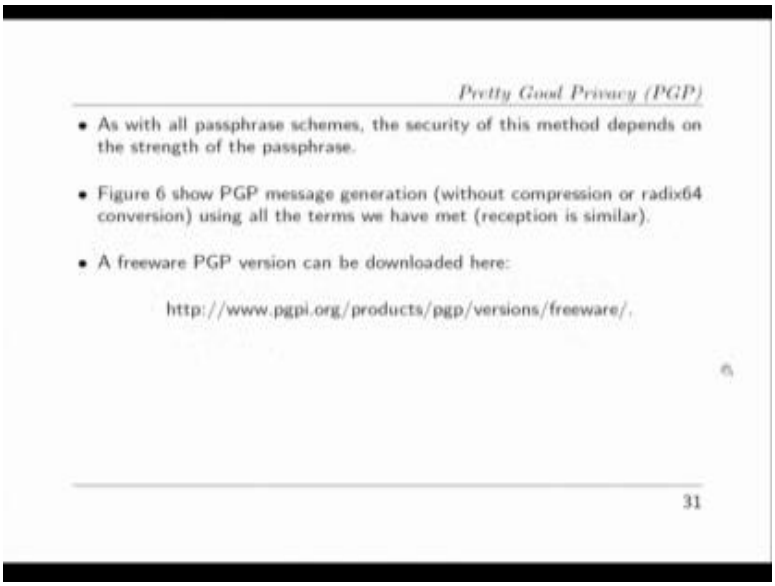
This is the public key and the encrypted public; encrypted private key and this is the user ID and this is basically private key rings and this is this is the public key rings. So, this is basically key rings use in PGP.

(Refer Slide Time: 27:50)

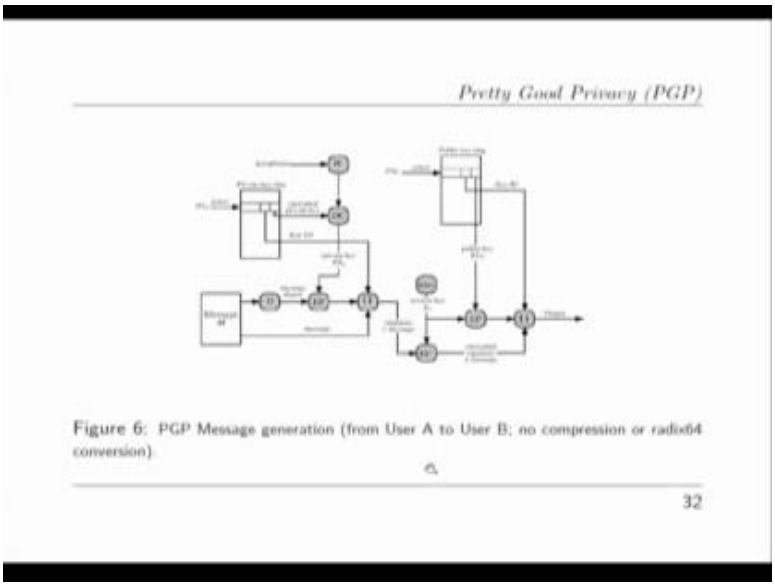


This we have discussed the timestamp key ID public key private key and the user ID is used.

(Refer Slide Time: 28:00)



(Refer Slide Time: 28:09)



A free version of PGP is available here. So, one can download these for their own use and this is the PGP message generation structure which we have discussed. So, this is basically ID is giving us the encrypted, the key ring is here so we get the ID then the message is hashed then we apply the encryption, this is the public key encryption and this public key we are getting from here and then after encrypting we are sending to the bob as to the receiver b and b is doing the same thing, b has a public key ring from there upon this is the ID of b getting the public key and doing the same process. So, this is the

encrypted signature plus the message and then we are giving the output. So, this is the general message generation of the PGP.

Thank you.