PGP authentication operation

What we are doing PGP authentication operation is sender creates a message , SHA1 is used to generate 160 bit hash code of message, assuming that we are using RSA SHA1 algorithm we can also use DSS, but in DSS we are using SHA1, I mean we are using SHA1 to generate hash. Hash code is encrypted with RSA using the sender's private key and result Is attached to message, if you are using DSS this thing is different, but some other parameters ,then receiver uses DSS or RSA, with sender public key to decrypt and recover the hash, this is the verification part, and then receiver generates new hash code message and compares with decrypted hash code ,if match then message is accepted as authentic, if they are same, they are not then we are not going to accept

PGP confidentiality

Sender creates a message and creates a random number what is that number ?..that number is going to be the session key and the message is going to be encrypted by CASH, IDEA with that session key and session key is going to be encrypted using RSA with the recipient public key ,then receiver uses his private key to decrypt and recover session key. And recovered session key is used to decrypt message that is the confidentiality part.

PGP authentication and confidentiality

if you both then what you are going to do is you are going to attach the message and encrypt both message and attach both RSA attached session key to the message so this way you have both authentication and confidentiality.

PGP Compression

By default compression is signing after signing and before encryption , so before encrypting we are going to zip or before signing we are going to message the i amportant thing over here is signing is signing is going to be done on clear plaintext not on the compressed message , compression is not deterministic they are using zip compression algorithm that explained.

PGP email compatibility

For compatibility they are using Radix 64 algorithm it maps 3 bytes to ,it means 24 bits to 4 print base characters, and pgp is going to use this one to convert encrypt files into text files, the result of encryption is the binary file. Like extra file ,and also pgp will segment the message if the message is too big, crc is the code checking mechanism that we use in ASCII codes, fore example ASCII as a crc is not used code checking or error checking

PGP operation

There is the file signature required ,do you want to sign the message , because whenever you use PGP or SMI AME there will be two option on the tools, that is user interface, one part is going it say sign and the other is going it say encrypt, it is your option ,either you are going to sign the message , but if you

say sign and encrypt its going to do both , so that's why here the application is required to generate both , if signature required then generate signature and go to compress, if compression is required ,then compress it, then if confidentiality required if yes encrypt the message with the session key and encrypt the session key with public key of the message and then convert into radix 64 and send and when I receive that email i am going to convert form radix64 to that message ,confidentiality required then it is yes , decryption the session with the private key and use it decrypt the message so then compress the message so I know that decrypted message is compress the message so now I have to decompress the message , unzip it so that signature required ,yes, if the verification is successful then you are done with the process, you see this is overall structure of PGP and now we are going toad more things and idea with key now, as it will cause many problems

PGP session keys

First of PGP session keys are of varying sizes, 56bit DES, 128 bit CAST or IDEA ,168 bit triple DES. Key generation is using this algorithm ANSIX12.17 they are using 3 triple des algorithm for this, if you check the appendix you can check out the algorithm as i am not going to go into details, this is actually a random number generator and it is using a standard, random number generator , uses random inputs taken from previous and from keystroke ti aming of user. there are different things here, someti ames they are going to say move the mouse around and you a re going to move the mouse around the table so they are going to take the pixels and it is random and they are going to use as seed in the algorithm or they are going to say write something and you are going to write ti amings that's going to be different and random and for seed or previous ones you generated something some of the output of this guy is the seed of the next generation you are going to use the output of this algorithm as seed, point is pgp is using this algorithm to generate this session keys.

PGP public and private keys

We are going to have problems with this guys, private and public key ,we know that public and private key may be used ,need to identify which is actually used to encrypt session key of the message you may have multiple public and private key so that's why we have to know which key is used for signing and form encryption ,we could send a full public key of message as you suggest but this is inefficient ,because key lengths are quite , we are going to use key identifier as key lengths are quite bit, so what we are doing is ,this significant 64 bits will be key identifier ,most likely it is going to be unique.{question from a student}. We dint has key identifier we had actual key in the certificate, that was real to usage of the key it was identification of the key whether this is s used for signature or identification but we dint use this kind of identity in certification but over here we are going to use 64 bits has key id, what we are going to do is this is actual message structure of PGP keys and i am going to send a message and tell you that I sent you the message on this day and ti ame and there is file name this part is you message ad the second part is the signature part, message digest is calculated using hash algorithm and then this message digest is encrypted with the private key of the sender and this is the signature part and there is ti amestamp saying that this signature is created at this ti ame and the key id of the sender public key , now i am saying the recipient to receive the public key you have to have the private key of the recipient , so then recipient when it trying to verify it does know from this key only corresponding from the public

key , so I m going to use this public key of that user, i am telling recipient uses that public key as well, and then reading 2 octets of message digest we calculated message digest and it is encrypted and then released the 2 octets of the message digest ,octets are sized 8 bits which means 16 bits what is the digest size is 160 bits and 8 bits living 2 octets Is going to be put over here, why is that?..this key id Is over here and I am going decipher this and i am going to get leading 8 octet and i am going to compare with the leading 8 octets and i am going to use the corresponding public key , because if something happens then I may change the key or these guys may something wrong instead of trying to verify the whole signature i am making sure that i am using the correct key by deciphering and by using the correct key of the octets , if these 2 octets are same you are absolutely sure that you are actually using the correct public key, clear?.so this part is the signature part appended and then this message and signature part is going to be zipped, compressed , and the result is going to be encrypted with the ANSI x12.17 ,from that random number , so i am going to encrypt the signature part and message part and it is compressed and i am encrypting and then session key component and what i am doing is i am going put this session key and i am going to encrypt this one with the recipient public key but recipient doesn't know which public key I used so what i am going to do is i am going to put the key id of the public key so then the recipient can understand the sender used this particular public key and result will be converted using radix 64 as its binary and they are going to make it txt and they can send it, you see how secure they are, they are making protocol secure and when you look at the overall structure the logic is very sound and the technique is all the technique we have discussed so far security is always like this ,DS is always like this and confidentiality for encryption and then try to solve the session key distribution ,you have security product and this guy does everything in neat way, you received as email what you are going to do , now from here you are going to go to key id but now how you are going to use is different thing , what we are going to have is two key rings on machine, pgp client will have two key rings , public key rings and the private key rings, public key will contain the public key info of the users that we know, and private key is going to keep the keys that you are using and the corresponding private keys and these keys are going to be listed in that directory based on key Id and the user name, so then when I get he key id ,i am going to go to public key ring ,so that when I go to key id, it is for my public key ring ,and that i am going to go to private key ring and i am going to go to private key ring ,but private key must be kept secret, if you are keeping the private key in the directory like that then you are going to have problems, one more thing that we have to solve we are going to provide security top private key in the private key ring ,solution is yo u are going to encrypt the key and that key will be encrypted actually but this time the problem is going to be this , which key you are going to use for that private key so we have generate another key and that key Is the function for the password that you are going to be selected, there will be password and you are going to choose one password and encrypt that password and result is going to be the key to encrypt the private key ,everytime you are receiving this message to recover the message from the corresponding private key PGP will ask that password, that password is not stored in any appln ,any time everytime when you want to access the private key I nthe private key ring PGP will ask the password and that password is going to be very weak, all the key that will be generated will be used and it is that important.