Computer Networks (CS425)

Instructor: Dr. Dheeraj Sanghi

Prev | Next | Index

Network Security

Data on the network is analogous to possessions of a person. It has to be kept secure from others with malicious intent. This intent ranges from bringing down servers on the network to using people's private information like credit card numbers to sabotage of major organizations with a presence on a network. To secure data, one has to ensure that it makes sense only to those for whom it is meant. This is the case for data transactions where we want to prevent eavesdroppers from listening to and stealing data. Other aspects of security involve protecting user data on a computer by providing password restricted access to the data and maybe some resources so that only authorized people get to use these, and identifying miscreants and thwarting their attempts to cause damage to the network among other things.

The various issues in Network security are as follows :

1. **Authentication:** We have to check that the person who has requested for something or has sent an e-mail is indeed allowed to do so. In this process we will also look at how the person authenticates his identity to a remote machine.
2. **Integrity:** We have to check that the message which we have received is indeed the message which was sent. Here CRC will not be enough because somebody may deliberately change the data. Nobody along the route should be able to change the data.
3. **Confidentiality:** Nobody should be able to read the data on the way so we need Encryption
4. **Non-repudiation:** Once we sent a message, there should be no way that we can deny sending it and we have to accept that we had sent it.
5. **Authorization:** This refers to the kind of service which is allowed for a particular client. Even though a user is authenticated we may decide not to authorize him to use a particular service.

For authentication, if two persons know a secret then we just need to prove that no third person could have generated the message. But for Non-repudiation we need to prove that even the sender could not have generated the message. So authentication is easier than Non-repudiation. To ensure all this, we take the help of cryptography. We can have two kinds of encryption :

1. **Symmetric Key Encryption:** There is a single key which is shared between the two users and the same key is used for encrypting and decrypting the message.

2. **Public Key Encryption:** There are two keys with each user : a public key and a private key. The public key of a user is known to all but the private key is not known to anyone except the owner of the key. If a user encrypts a message in his private key then it can be decrypted by anyone by using the sender's public key. To send a message securely, we encrypt the message in the public key of the receiver which can only be decrypted by the user with his private key.

Symmetric key encryption is much faster and efficient in terms of performance. But it does not give us Non-repudiation. And there is a problem of how do the two sides agree on the key to be used assuming that the channel is insecure (others may snoop on our packet). In symmetric key exchange, we need some amount of public key encryption for authentication. However, in public key encryption, we can send the public key in plain text and so key exchange is trivial. But this does not authenticate anybody. So along with the public key, there needs to be a certificate. Hence we would need a public key infrastructure to distribute such certificates in the world.

Key Exchange in Symmetric Key Schemes

We will first look at the case where we can use public key encryption for this key exchange. . The sender first encrypts the message using the symmetric key. Then the sender encrypts the symmetric key first using it's private key and then using the receiver's public key. So we are doing the encryption twice. If we send the certificate also along with this then we have authentication also. So what we finally send looks like this :

$$Z : \quad \text{Certificate}_{\text{sender}} + \text{Public}_{\text{reciever}} (\text{Private}_{\text{sender}} (E_k)) + E_k (M)$$

Here E_k stands for the symmetric key and E_k (M) for the message which has been encrypted in this symmetric key.

However this still does not ensure integrity. The reason is that if there is some change in the middle element, then we will not get the correct key and hence the message which we decrypt will be junk. So we need something similar to CRC but slightly more complicated. This is because somebody might change the CRC and the message consistently. This function is called Digital Signature.

Digital Signatures

Suppose A has to send a message to B. A computes a hash function of the message and then sends this after encrypting it using its own private key. This constitutes the signature produced by A. B can now decrypt it, recompute the hash function of the message it has received and compare the two. Obviously, we would need the hash functions to be such that the probability of two messages hashing to the same value is extremely low. Also, it should be difficult to compute a message with the same hash function as another given message. Otherwise any intruder could replace the message with another that has the same hash value and leave the signatures intact leading to loss of integrity. So the message along with the digital signature looks like this :

$$Z + \text{Private}_{\text{sender}} (\text{Hash} (M))$$

Digital Certificates

In addition to using the public key we would like to have a guarantee of talking to a known person. We assume that there is an entity who is entrusted by everyone and whose public key is known to everybody. This entity gives a certificate to the sender having the sender's name, some other information and the sender's public key. This whole information is encrypted in the private key of this trusted entity. A person can decrypt this message using the public key of the trusted authority. But how can we be sure that the public key of the authority is correct ?   In this respect Digital signatures are like I-Cards. Let us ask ourselves the question : How safe are we with I-Cards? Consider a situation where you go to the bank and need to prove your identity. I-Card is used as a proof of your identity. It contains your signature. How does the bank know you did not make the I-Card yourselves? It needs some proof of that and in the case of I-Cards they contain a counter signature by the director for the purpose. Now how does the bank know the signature I claim to be of the director indeed belongs to him? Probably the director will also have an I-Card with a counter signature of a higher authority. Thus we will get a chain of signing authorities. Thus in addition to signing we need to prove that the signatures are genuine and for that purpose we would probably use multiple I-Cards each carrying a higher level of signature-counter signature pair.
So in order to distribute the public key of this authority we use certificates of higher authority and so on. Thus we get a tree structure where the each node needs the certificates of all nodes above it on the path to the root in order to be trusted. But at some level in the tree the public key needs to be known to everybody and should be trusted by everybody too.

---

back to top
Prev| Next | Index

sses messages across a link from one.machine to another. The mail is enclosed in what is called an **envelope** . The enveilope contains the To and From fields and these are followed by the mail . The mail consists of two parts namely the Header and the Data.