Part II

Cryptography Demystified

Find out how the issues of security and trust are addressed by the public key infrastructure (PKI) technology

■ PROF. G.R. KULKARNI, PROF. A.C. SUTHAR, ASHISH N. JANI

A public key infrastructure (PKI) not only contains the certificate storage facilities of a certificate server but also provides certificate management facilities (the ability to issue, revoke, store, retrieve and trust certificates). The main feature of a PKI is the introduction of what is known as a certification authority (CA), which is a human entity (a person, group, department, company or other association) that an organisation has authorised to issue certificates to its computer users.

The role of a CA is analogous to a government's passport office. It creates certificates and digitally signs them using its private key. Because of its role in creating certificates, the CA is the central component of a PKI. Using the CA's public key, anyone wanting to verify a certificate's authenticity verifies the issuing CA's digital signature, and hence the integrity of the contents of the certificate (most importantly, the public key and the identity of the certificate holder).

Certificate formats

A digital certificate is basically a collection of identifying information bound together with a public key and signed by a trusted third party to prove its authenticity. A digital certificate can be one of a number of different formats.

Pretty good privacy (PGP) recognises two different certificate formats: PGP certificates and X.509 certificates.

PGP certificate format. A PGP certificate includes (but is not limited to) the following information:

1. *The PGP version number.* This identifies the version of PGP used to create the key associated with the certificate.

2. *The certificate holder's public key.* The public portion of your key pair, together with the algorithm of the key; RSA, Diffie-Hellman (DH) or digital signature algorithm (DSA).

3. *The certificate holder's information.* This consists of 'identity' information about the user, such as his or her name, user ID, photograph and so on.

4. *The digital signature of the certificate owner.* Also called a self-signature, this is the signature using the corresponding private key of the public key associated with the certificate.

5. *The certificate's validity period.* The certificate's start date/time and expiration date/time, indicating when the certificate will expire.

6. *The preferred symmetric encryption algorithm for the key.* The encryption algorithm to which the certificate owner prefers to have information encrypted. The supported algorithms are CAST, IDEA or Triple-DES.

PGP certificate is a public key with one or more labels tied to it. On these 'labels' you'll find

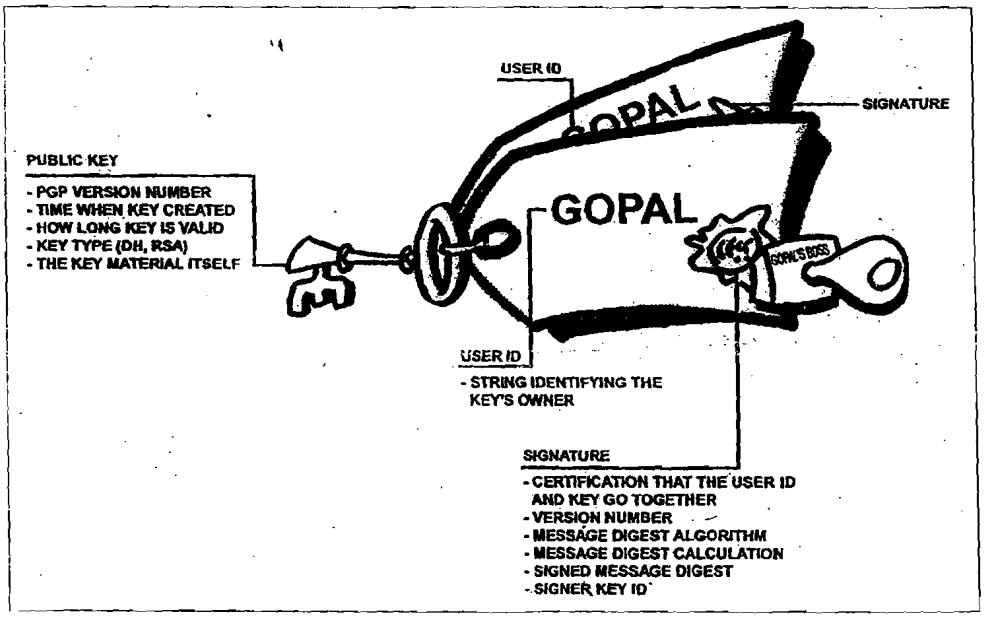

Fig. 9: A PGP certificate

the key and a signature of the key's owner, which states that the key and the identification go together. (This particular signature is called a self-signature; every PGP certificate contains a self-signature.)

One unique aspect of the PGP certificate format is that a single certificate can contain multiple signatures. Several or many people may sign the key/identification pair to attest to their own assurance that the public key definitely belongs to the specified owner. If you look on a public certificate server, you may notice that certain certificates contain many signatures.

Some PGP certificates consist of a public key with several labels, each of which contains a different means of identifying the key's owner; for example, the owner's name and corporate e-mail account, the owner's nickname and home email account, a photograph of the owner—all in one certificate. The list of signatures of each of those identities may differ; signatures attest to the authenticity that one of the labels belongs to the public key, not that all the labels on the key are authentic. Note that 'authentic' is in the eye of its beholder—signatures are opinions, and different people devote different levels of due diligence in checking authenticity before signing a key.

X.509 certificate format. *X.509* is another very common certificate format. All X.509 certificates comply with the ITU-T X.509 international standard; thus (theoretically) X.509 certificates created for one application can be used by any application complying with X.509. In practice, however, different companies have created their own extensions to X.509 certificates, not all of which work together.

A certificate requires someone to validate that a public key and the name of the key's owner go together. With PGP certificates, anyone can play the role of validator. With X.509 certificates, the validator is always a certification authority or someone designated by a CA. (Bear in mind that PGP certificates also fully support a hierarchical structure using a CA to validate certificates.)

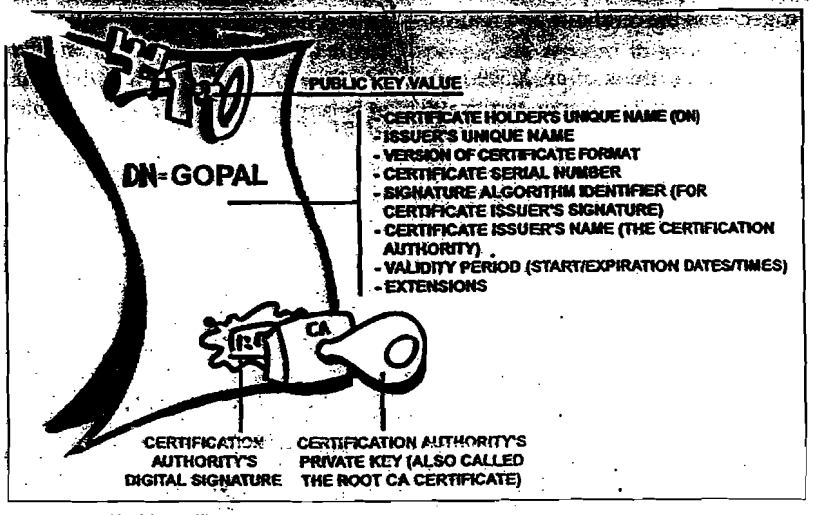

Fig. 10: An X.509 certificate

An X.509 certificate is a collection of a standard set of fields containing information about a user or device and their corresponding public key. The X.509 standard defines the information going into the certificate, and describes how to encode it (the data format). All X.509 certificates have the following data:

1. The X.509 version number. This identifies the version of the X.509 standard applying to this certificate, which affects the information that can be specified in it. The most current is version 3.

2. The certificate holder's public key. The public key of the certificate holder, together with an algorithm identifier that specifies the cryptosystem that the key belongs to and any associated key parameter.

3. The serial number of the certificate. The entity (application or person) that created the certificate is responsible for assigning it a unique serial number to distinguish it from other certificates it issues. This information is used in numerous ways; for example, when a certificate is revoked, its serial number is placed in a certificate revocation list or CRL.

4. The certificate's validity period. The certificate's start date/time and expiration date/time, indicating when the certificate will expire.

5. The unique name of the certificate issuer. The unique name of the entity that signed the certificate. This is normally a CA. Using the certificate implies trusting the entity that signed this certificate. Note that in some cases, such as root or top-level CA certificates, the issuer signs its own certificate.

6. The digital signature of the issuer. The signature using the private key of the entity that issued the certificate.

7. The signature algorithm identifier. Identifies the algorithm used by the CA to sign the certificate.

There are many differences between an X.509 certificate and a PGP certificate, but the most salient are:

1. You can create your own PGP certificate; you must request and be issued an X.509 certificate from a certification authority.

2. X.509 certificates natively support only a single name for the key's owner.

3. X.509 certificates support only a single digital signature to attest to the key's validity.

To obtain an X.509 certificate, you must ask a CA to issue you a certificate. You provide your public key, proof that you possess the corresponding private key and some specific information about yourself. You then digitally sign the information and send the whole package—the certificate request—to the CA.

The CA then performs some due diligence in verifying that the information you provided is correct, and if

so, generates the certificate and returns it.

You might think of an X.509 certificate as a standard paper certificate (similar to one you might have received for completing a class in basic first aid) with a public key taped to it. It has your name and some information about you on it, plus the signature of the person who issued it to you.

X.509 certificates are widely used in Web browsers.

Validity and trust

Every user in a public key system is vulnerable to mistaking a phony key (certificate) for a real one. Validity is confidence that a public key certificate belongs to its purported owner. It is essential in a public key environment where you must constantly establish whether or not a particular certificate is authentic.

When you've assured yourself that a certificate belonging to someone else is valid, you can sign the copy on your keyring to attest to the fact that you've checked the certificate and that it's an authentic one. If you want others to know that you gave the certificate your stamp of approval, you can export the signature to a certificate server so that others can see it.

As described in the public key infrastructure section, some companies designate one or more CAs to indicate certificate validity. In an organisation using a PKI with X.509 certificates, it is the job of the CA to issue certificates to the users—a process that generally entails responding to a user's request for a certificate. In an organisation using PGP certificates without a PKI, the CA checks the authenticity of all the PGP certificates and then signs the good ones. Basically, the main purpose of a CA is to bind a public key to the identification information contained in the certificate and thus assure third parties that some measure of care was taken to ensure that this binding of the identification information and key is valid.

The CA is the Grand Pooh-bah of validation in an organisation: someone whom everyone trusts, and in some organisations, like those using a PKI, no certificate is considered valid unless it has been signed by a trusted CA.

Checking the validity

One way to establish validity is to go through some manual process. There are several ways to accomplish this. You could require your intended recipient to physically hand you a copy of his public key. But this is often inconvenient and inefficient.

Another way is to manually check the certificate's fingerprint. Just as every human's fingerprints are unique, every PGP certificate's fingerprint is unique. The fingerprint is a hash of the user's certificate and appears as one of the certificate's properties. In PGP, the fingerprint can appear as a hexadecimal number or a series of biometric words, which are phonetically distinct and used to make the fingerprint identification process a little easier.

One can check that a certificate is valid by calling the key's owner (so as

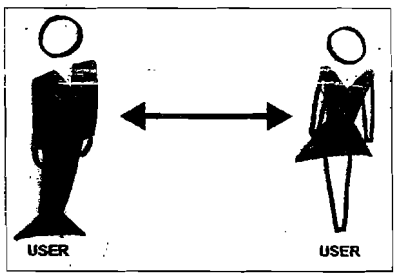

Fig. 11: Direct trust

to originate the transaction) and asking him to read his key's fingerprint and verifying that fingerprint against the one believed to be the real one. But how to manually verify the identity of some stranger? Some people put the fingerprint of their key on their business cards for this very reason.

Another way to establish validity of someone's certificate is to trust that a third individual has gone through the process of validating it. A CA, for example, is responsible for ensuring that prior to issuing a certificate, he carefully checks it to be sure that the public key portion really belongs to the purported owner. Anyone who trusts the CA will automatically consider any certificate signed by the CA to be valid.

Establishing trust

You validate certificates. You trust people. More specifically, you trust people to validate other people's certificates. Typically, unless the owner hands you the certificate, you have to go by someone else's word that it is valid.

Meta and trusted introducers

In most situations, people completely trust the CA to establish the certificates' validity. This means that everyone else relies upon the CA to go through the whole manual validation process for them. This is fine up to a certain number of users or number of work sites, and then it is not possible for the CA to maintain the same level of quality validation. In that case, adding other validators to the system is necessary.

A CA can also be a meta-introducer. A meta-introducer bestows not only validity on keys but also the ability to trust keys upon others. Similar to the king who hands his seal to his trusted advisors so that they can act on his authority, the meta-introducer enables others to act as trusted introducers. These trusted introducers can validate keys to the same effect as that of the meta-introducer. They cannot, however, create new trusted introducers.

Meta-introducer and trusted introducer are PGP terms. In an X.509 environment, the meta-introducer is called the root certification authority (root CA) and trusted introducers subordinate certification authorities.

The root CA uses the private key associated with a special certificate type called a root CA certificate to sign certificates. Any certificate signed by the root CA certificate is viewed as valid by any other certificate signed by the root. This validation process works even for certificates signed by other CAs in the system—as long as the root CA certificate signed the subordinate CA's certificate, any certificate signed by the CA is

considered valid to others within the hierarchy. This process of checking back up through the system to see who signed whose certificate is called tracing a certification path or certification chain.

Trust models

In relatively closed systems, such as within a small company, it is easy to trace a certification path back to the root CA. However, users must often communicate with people outside their corporate environment, including those they have never met, such as vendors, customers, clients, associates and so on. Establishing a line of trust to those who have not been explicitly trusted by your CA is difficult.

Companies follow one or another trust model, which dictates how users will go about establishing certificate validity. There are three different models: direct trust, hierarchical trust and a web of trust.

Direct trust. Direct trust is the simplest trust model. In this model, a user trusts that a key is valid because he knows where it came from. All cryptosystems use this form of trust in some way. For example, in Web browsers, the root certification authority keys are directly trusted because they were shipped by the manufacturer. If there is any form of hierarchy, it extends from these directly trusted certificates.

In PGP, a user who validates keys himself and never sets another certificate to be a trusted introducer is using direct trust.

Hierarchical trust. In a hierarchical system, there are a number of 'root' certificates from which trust extends. These certificates may certify certificates themselves, or they may certify certificates that certify still other certificates down some chain. Consider it as a big trust 'tree.' The 'leaf' certificate's validity is verified by tracing backward from its certifier, to other certifiers, until a directly trusted root certificate is found.

Web of trust. A web of trust encompasses both of the other models, but also adds the notion that trust is in the eyes of the beholder (which is the real-world view) and the idea that more information is better. It is thus a cumulative trust model. A certificate might be trusted directly, or trusted in some chain going back to a directly trusted root certificate (the meta-introducer), or by some group of introducers.

Perhaps like the term 'six degrees of separation,' which suggests that any person in the world can determine some link to any other person in the world using six or fewer other people as intermediaries, this is a web of introducers.

It is also the PGP view of trust. PGP uses digital signatures as its form of introduction. When any user signs another's key, he becomes an introducer of that key. As this process goes on, it establishes a web of trust.

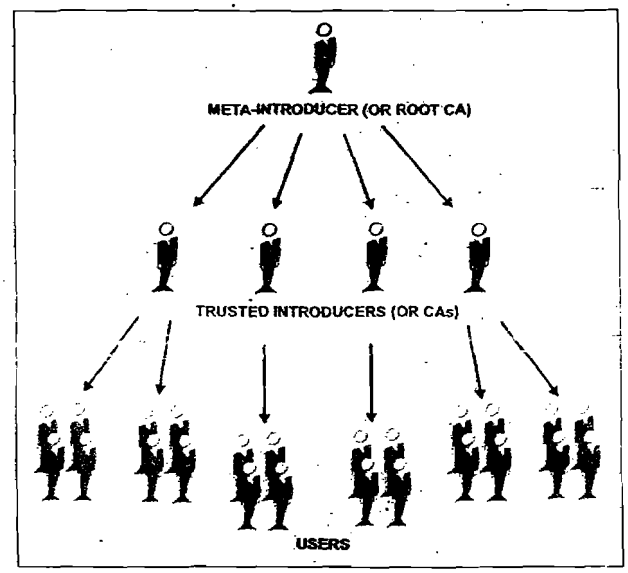

Fig. 12: Hierarchical trust

In a PGP environment, any user can act as a certifying authority. Any PGP user can validate another PGP user's public key certificate. However, such a certificate is valid to the other user only if the relying party recognises the validator as a trusted introducer. Stored on each user's public keyring are indicators of:

1. Whether or not the user considers a particular key to be valid.

2. The level of trust the user places on the key that the key's owner can serve as the certifier of others' keys.

It's a reputation system: certain people are reputed to give good signatures, and people trust them to attest to other keys' validity.

Levels of trust in PGP

The highest level of trust in a key, implicit trust, is trust in your own key pair. PGP assumes that if you own the private key, you must trust the actions of its related public key. Any key signed by your implicitly trusted key is valid.

There are three levels of trust you can assign to someone else's public key: complete trust, marginal trust and no trust (or untrusted). There are also three levels of validity: valid, marginally valid and invalid.

To define another's key as a trusted introducer, you start with a valid key, one that is either signed by you or signed by another trusted introducer, and then set the level of trust you feel the key's owner is entitled.

PGP requires one completely trusted signature or two marginally trusted signatures to establish a key as valid. PGP's method of considering "two marginals equal to one complete" is similar to a merchant asking for two forms of ID.

Certificate revocation

Certificates are useful only while they are valid. It is unsafe to simply assume that a certificate is valid forever. In most organisations and in all PKIs, certificates have a restricted lifetime. This constrains the period in which a system is vulnerable should a certificate compromise occur.

Certificates are thus created with a scheduled validity period: a start date/time and an expiration date/time. The

certificate is expected to be usable for its entire validity period (its lifetime). When the certificate expires, it will no longer be valid, as the authenticity of its key/identification pair is no longer assured. (The certificate can still be safely used to reconfirm information that was encrypted or signed within the validity period—it should not be trusted for cryptographic tasks moving forward, however.)

There are also situations where it is necessary to invalidate a certificate prior to its expiration date, such as when the certificate holder terminates employment with the company or suspects that the certificate's corresponding private key has been compromised. This is called revocation. A revoked certificate is much more suspect than an expired certificate. Expired certificates are unusable, but do not carry the same threat of compromise as a revoked certificate.

Anyone who has signed a certificate can revoke his signature on the certificate (provided he uses the private key that created the signature). A revoked signature indicates that the signer no longer believes that the public key and identification information belong together, or that the certificate's public key (or corresponding private key) has been compromised. A revoked signature should carry nearly as much weight as a revoked certificate.

With X.509 certificates, a revoked signature is practically the same as a revoked certificate given that the only signature on the certificate is the one that made it valid in the first place—the signature of the CA. PGP certificates provide the added feature that a person can revoke your entire certificate (not just the signatures on it) if he feels that the certificate has been compromised.

Only the certificate's owner (the holder of its corresponding private key) or someone whom the certificate's owner has designated as a revoker can revoke a PGP certificate. (Designating a revoker is a useful practice, as it's often the loss of the passphrase for the certificate's corresponding private key that leads a PGP user to revoke his certificate—a task that is possible only if one has access to the private key.) Only the certificate's issuer can revoke an X.509 certificate.

Communicating that a certificate has been revoked

When a certificate is revoked, it is important to make potential users of the certificate aware that it is no longer valid. With PGP certificates, the most common way to communicate that a certificate has been revoked is to post it on a certificate server so that others who may wish to communicate with you are warned not to use that public key.

In a PKI environment, communication of revoked certificates is most commonly achieved via a data structure called a certificate revocation list, or CRL, which is published by the CA. The CRL contains a time-stamped, validated list of all revoked, unexpired certificates in the system. Revoked certificates remain on the list only until they expire, then they are removed from the list—this keeps the list from getting too long.

The CA distributes the CRL to users at some regularly scheduled interval (and potentially off-cycle, whenever a certificate is revoked). Theoretically, this prevents users from unwittingly using a compromised certificate. It is possible, though, that there may be a time period between CRLs in which a newly compromised certificate is used.

Typically composed of multiple words, a passphrase is more secure against standard dictionary attacks, wherein the attacker tries all the words in the dictionary in an attempt to determine your password.

Passphrase

Most people are familiar with restricting access to computer systems via a password, which is a unique string of characters that a user types in as an identification code. A passphrase is a longer version of a password, and in theory, a more secure one.

Typically composed of multiple words, a passphrase is more secure against standard dictionary attacks, wherein the attacker tries all the words in the dictionary in an attempt to determine your password. The best passphrases are relatively long and complex and contain a combination of upper- and lower-case letters, numeric and punctuation characters.

PGP uses a passphrase to encrypt your private key on your machine. Your private key is encrypted on your disk using a hash of your passphrase as the secret key. You use the passphrase to decrypt and use your private key.

A passphrase should be hard for you to forget and difficult for others to guess. It should be something already firmly embedded in your long-term memory, rather than something you make up from scratch. Why? Because if you forget your passphrase, you are out of luck. Your private key is totally and absolutely useless without your passphrase and nothing can be done about it.

PGP is a cryptography that will keep major governments out of your files. It will certainly keep you out of your files, too.

Key splitting

A secret is not a secret if more than one person know it. Sharing a private key pair poses such a problem. While it is not a recommended practice, sharing a private key pair is necessary at times.

Corporate signing keys, for example, are private keys used by a company to sign, for example, legal documents, sensitive personnel information or press releases to authenticate their origin. In such a case, it is worthwhile for multiple members of the company

to have access to the private key. However, this means that any single individual can act fully on behalf of the company.

In such a case it is wise to split the key among multiple people in such a way that more than one or two people must present a piece of the key in order to reconstitute it to a usable condition. If very few pieces of the key are available, the key is unusable.

Some examples are to split a key into three pieces and require two of them to reconstitute the key, or split it into two pieces and require both pieces. If a secure network connection is used during the reconstitution process, the key's shareholders need not be physically present in order to rejoin the key.

Making secure e-business a reality

The Internet has revolutionised the way the world communicates and become the primary means for companies to disseminate information. Companies use their Websites, e-mail and other Internet technologies as a means for communication with their customers, business partners and employees. Many companies have existing business processes in place or are developing business processes to conduct their e-business. Integrating security within these business processes is vital to the success of a company's e-business strategy.

The most formidable obstacles to the wide acceptance of e-business and e-commerce are the issues of security and trust.

Any communication sent through an open medium like the Internet is inherently public. As it moves through the Internet via multiple servers, anyone can intercept it and read the contents with impunity. Unlike traditional modes of communication and transaction which were paper based, electronic transactions are inherently anonymous in nature. The user typically interacts with a computer, which, in turn, acts as a surrogate in interacting with other computers and people. It is therefore easy to impersonate someone and assume his identity.

People who do not have the requisite authorisation regularly break into computer systems through hacking or stealing of passwords. It is necessary to set in place secure access controls to prevent leakage of sensitive or important documents. Security is about ensuring that their contents are not tampered with and controlling access to such documents.

Trust is fundamental to any kind of business transaction such as a contract whereby the parties involved in a transaction agree to certain terms and conditions. In a traditional paper-based society, this trust is achieved through the act of physically signing a contract in the presence of the parties concerned. Unlike security, trust in e-business is about giving electronic transactions legal acceptance equivalent to paper-based transactions. It is not about hacker risks but actual business risks that are involved in normal business transactions.

If you forget your passphrase, you are out of luck. Your private key is totally and absolutely useless without your passphrase and nothing can be done about it.

There are seven key attributes that are essential for creation and maintenance of trust in an e-business environment. These attributes, though important in normal commercial transactions and communications, are even more important when dealing with a medium of which the layman is naturally distrustful.

The attributes are:

1. Authentication. Authentication refers to the various means that are employed to validate the originator of an electronically transmitted message for the recipient. This means that the recipient of an e-message or e-document can verify that the originator is actually who he claims to be.

2. Data integrity. This means that there should exist means to protect the contents of electronic transactions and communications from being altered or corrupted and to provide acknowledgement of delivery by the recipient.

3. Data confidentiality. This refers to the need to protect the contents of electronically transmitted messages, as well as to protect against unauthorised use of the user's system. You have to prevent access to the information during transmission, and also protect against unauthorised access of the recipient's system.

4. Chain of custody. Documents may pass through a predefined hierarchy as in the case of an accounts statement. This hierarchy has to be maintained for electronically transmitted messages and documents.

5. Time stamping. This attribute is necessary for e-documents that are 'time sensitive.' For example, a document or paper created at a particular time is checked by the time or date put on the document along with the signature. Similarly, in the case of e-documents, one should be able to show when it was created.

6. Data archiving. In normal paper transactions, one maintains files or folders in which all pertinent documents are stored as long as necessary. In the case of e-documents also, it is imperative to keep a long-term store of all electronic transmissions and to facilitate the retrieval of this electronically transmitted data.

7. Audit capabilities. It is necessary for a company to keep track of all its correspondence. A similar need is felt in the case of e-documents as well, where tracking needs to be done throughout its life cycle.

All the parties involved should assure the person who is conducting a transaction over the Internet that this transaction is non-repairable.

These issues of security and trust are addressed by the PKI and digital signatures, which use PKI.

G.R. Kulkarni and A.C. Suthar are professors in Electronics & Communication Department, C.U. Shah College of Engineering & Technology, Wadhwan, Distt Surendranagar (Gujarat), and Ashish N. Jani, an M.Sc in IT, is a consultant