

## CHAPTER 18

# Security Issues in Mobile Computing

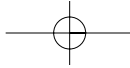
### 18.1 INTRODUCTION

Mobile computing is pervading our society and our lifestyles with a high momentum. Mobile computing with networked information systems help increase productivity and operational efficiency. This however, comes at a price. Mobile computing with networked information systems increase the risks for sensitive information supporting critical functions in the organization which are open to attacks.

The fundamental premise of mobile computing is that the information will be accessed from outside of the organization. As long as the information is within the four walls, the environment will be better known. It may be easier to control this environment and make it secure. When the information or computing environment is outside the controlled environment we do not have much control either from its users or usage patterns. Today, all the computers of the world are interconnected through extranet. Moreover, in a majority of cases, mobile computing uses wireless networks. Wireless media works on the principle of broadcast; information is radiated to everyone within the radio wave range thus increasing the security threats. Unlike a physical attack, cyber attacks can be replicated quite easily. Therefore, unless special care is taken, all systems are open to attack. This chapter discusses different techniques to secure information over mobile computing environment.

### 18.2 INFORMATION SECURITY

In any defense system, we need to know our enemy. We also need to determine possible areas—weak points, vulnerabilities—where the enemy may attack. We need to build a defense system around these vulnerabilities. To build an information security system, we



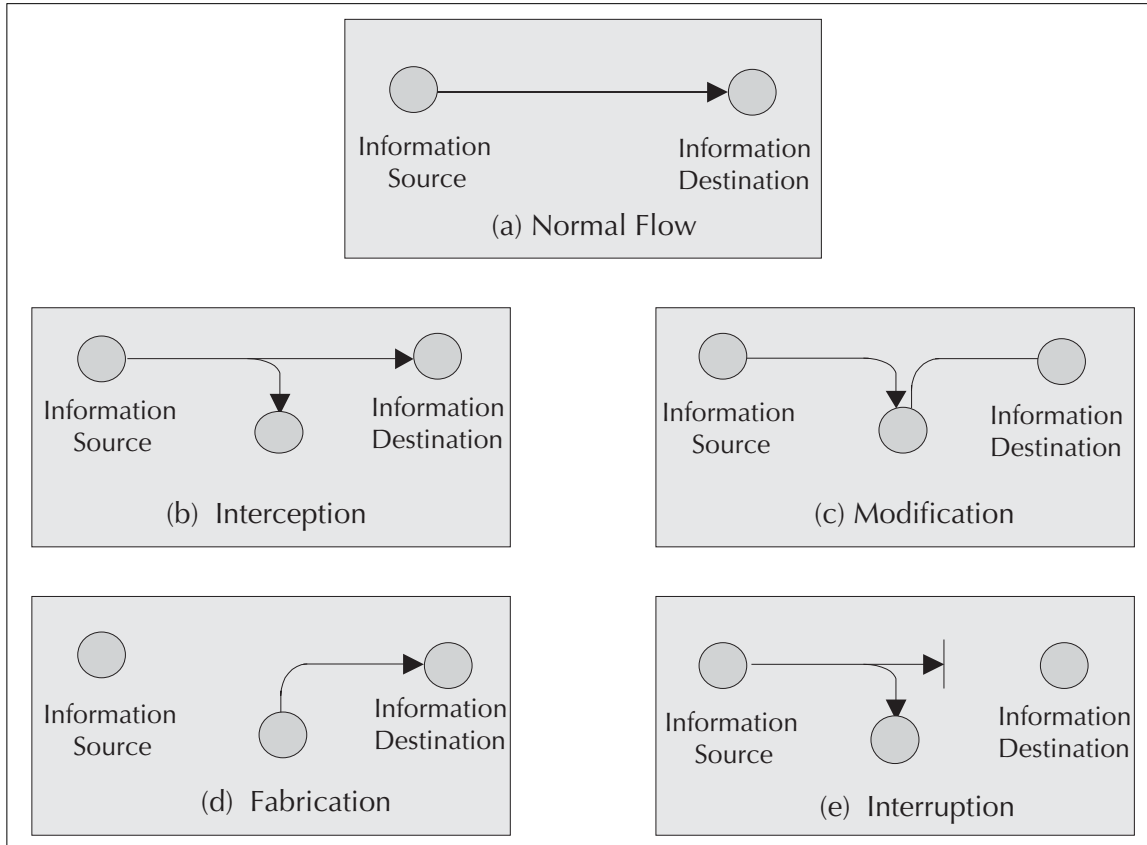
need to answer the following questions:

- Who is the enemy?
- What are the vulnerabilities? What are the weak links in the system?
- What could be the possible exploitation of these vulnerabilities by the resulting attacks?
- What needs special protection?
- To protect our assets from attack, we need to build a security system. How much does the security system cost in terms of money, resource and time?
- When the security system is deployed, to what extent will it affect the openness and add to inconvenience?
- Is prevention better than cure? If prevention is expensive or impractical, what is the strategy to recover from the loss following an attack?

There is no absolute security. What may appear to be absolute security in one context may not be absolute security in another context. Therefore, while building a security system, we need to arrive at a proper balance amongst the answers emerging from the above questions. In a mobile environment, the user roams through different networks with heterogeneous security infrastructure. In such an environment where device mobility and network mobility is a necessity, offering homogenous service over heterogeneous devices and networks is the key. In such an environment weak security link from a wireless network could become a point of vulnerability for the entire system. Therefore, in a mobile computing environment, it is necessary to have a robust security and trust infrastructure.

## 18.2.1 Attacks

A security system is a system to defend our assets from attacks. In the physical world, these attacks are carried out at the weak points in the defense system. Likewise in the electronic world, attacks are carried out at the point of vulnerability. When the vulnerability is exploited for some interest or selfish motive, it is an attack on the system. Of course there could be occasions where the vulnerability is exposed by accident as well. Where the vulnerability is exploited, there is a loss. This loss can be either of static information asset (static asset) or an information asset in transit (dynamic asset). If we look at an information system, static assets cover a large portion of the asset base. All the databases, files, documents, etc. in the computers fall in this category. Examples of attacks on static asset are virus deleting files in a computer or jamming a network. An example of



**Figure 18.1** Types of attacks

an attack on a dynamic asset is the theft of a credit card number while a user is doing a credit card transaction on the web.

Attack on dynamic assets can be of the following types (Figure 18.1):

- **Interception:** An unauthorized party gaining access to an asset will be part of this attack. This is an attack on **confidentiality** like unauthorized copying of files or tapping a conversation between parties. Some of the sniffing attacks fall in this category.
- **Modification:** An unauthorized party gaining control of an asset and tampering with it is part of this attack. This is an attack on **integrity** like changing the content of a message being transmitted through the network. Different types of man-in-the-middle attacks are part of this type of attack.

- **Fabrication:** An unauthorized party inserts counterfeited objects into the system; for example, impersonating someone and inserting a spurious message in a network.
- **Interruption:** An asset is destroyed or made unusable. This is an attack on **availability**. This attack can be on a static asset or a dynamic asset. An example could be cutting a communication line or making the router so busy that a user cannot use a server in a network. These are all Denial of service attack.

Attacks on static assets can be of the following types:

- **Virus and Worms:** These are a type of program that replicates and propagates from one system to another. Most of the virus do malicious destructive functions in the system.
- **Denial of Service:** These are attacks on the system to prevent legitimate users from using the service.
- **Intrusion:** These are people or software, which enter into computer systems and perform function without the knowledge of the owner of the asset. These are also called hackers.
- **Replay Attack:** In a replay attack the opponent passively captures the data without trying to analyze the content. At a later time, the same is used in the same sequence to impersonate an event and gain unauthorized access to resource.
- **Buffer overflow attacks:** In a buffer overflow attack, the vulnerability of an executable program is exploited to force a stack overflow condition, inducing the program counter of the process to change. The program counter is then manipulated to do the work for the attacker.
- **Trapdoor attacks:** These are exploitations of some undocumented features of a system. Undocumented functionality are designed to debug, service, support or take control of the system.

A security system needs to be so designed that the system is able to counter and recover from attacks.

## 18.2.2 Components of Information Security

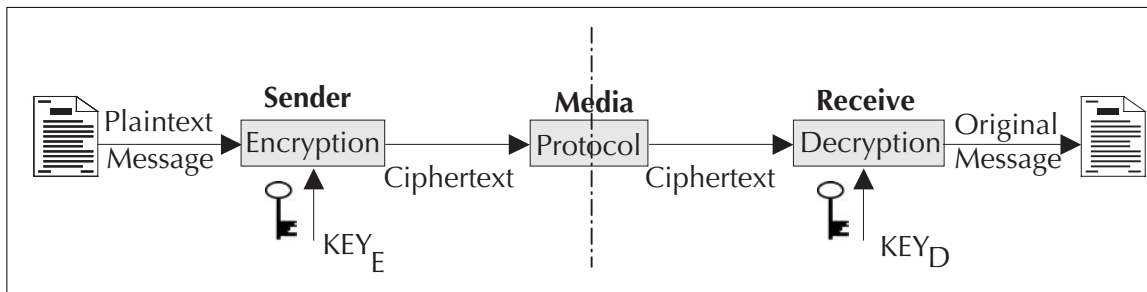
For centuries, information security was synonymous with secrecy. The art of keeping a message secret was to encrypt the message and thus hide it from others getting to know of it. However, in today's netcentric electronic world, the taxonomy of information

security is much beyond encryption. Information security needs to cater to all the possible attacks related to confidentiality, integrity, availability, non-repudiation, authorization, trust and accounting (CIANATA). **Confidentiality** is the property where the information is kept secret so that unauthorized persons cannot get at the information. **Integrity** is the property of keeping the information intact. **Availability** is the property of a system by which the system will be available to its legitimate users. **Non-repudiation** is the property by which the identity of both sender and receiver of the message can be identified and verified. **Authorization** is the property by which the user's properties can be associated to the information access. **Trust** is the property of expectation, confidence, and belief over time. **Accounting** is the property of calculating the fee for a service rendered.

### Confidentiality

Confidentiality is ensured through encryption of the data. To a person a comprehensible message is written in a particular language. The language can be English, Hindi, French or any other language. These messages are called plaintext or cleartext messages. Through encryption (or encipher) we disguise this message in such a fashion that it is no longer understandable by either a person or a machine. An encrypted message is called ciphertext. The process of converting a ciphertext back into plaintext is called decryption (or deciphering). Plaintext need not be a written text. It can even be an audio or video message as well. When leaders of two countries talk, the message is encrypted so that a man eaves dropping cannot make any sense of the conversation. The plaintext message can also be a data file in the computer disk. Figure 18.2 depicts the process of encryption and decryption.

In cryptography there are two components, viz., **algorithms** and **protocols**. A cryptographic algorithm is a mathematical function used for encryption and decryption, and



**Figure 18.2** Encryption and decryption with a key

protocol relates to the process and procedure of using algorithms. A protocol is the way algorithms are used to ensure that the security is ensured and the system is less prone to attack. In a security system the plaintext message is encrypted by using a key  $KEY_E$ . The encrypted message is then sent from the sender to the receiver through a media (wired, wireless, or even postal) using some protocol. The encrypted message is then decrypted using a key  $KEY_D$  to extract the original message. A cryptographic key is generally a large number. The range of possible values of a key is called **keyspace**. The larger the keyspace is, the more difficult it is for an attacker to guess the key and restore the original message. Therefore a larger keyspace makes a ciphertext more secure. This is similar to a lock. A conventional lock of 11 levers is more secure compared to a 7-lever lock.

The art of keeping message secure using the science of encryption and decryption is called cryptography. People who practise cryptography are called **cryptographers**. There are people who try to break the secrecy of encryptions. These are for many purposes; some are for research purposes to measure the strength of the security and some, for stealing the information. Some are hackers who try to break the security for fun or for a price. These people who try to break the secrecy of the cryptography are called **cryptanalysts**. The practice of cryptanalyst is called **cryptanalysis**. There is another science in security engineering. This is called **steganography**. Steganography is the science of hiding secret message in other messages so that the existence of the secret message is concealed; for example, sending some secret message by changing some bits in a large picture message. By looking at the picture, others will not be able to guess that in reality the picture is carrying a secret message.

## Integrity

**Integrity** is to ensure the integrity of the message. Integrity is achieved by adding additional information in to the message. This is done through checksums, message digests or digital signature. In a crypto system, the receiver of the message checks this extra information to verify whether the message has been tampered with. This is similar to a bank cheque. A cheque issued to a customer is honored only when the customer signs it. The cheque number and the signature are verified to ensure integrity. Integrity check is advised for both static asset and asset on transit.

## Authorization

**Authorization** deals with privileges. In any transaction, there is a subject (a person) and an object (data items or file). The subject wants some function to be performed on

the object. The privilege to an object is defined through ACL or Access Control List. ACL is used while allowing access to the object. The privilege on an object can be read, write, or execute. Besides objects there need to be privilege-based type of subjects. This is done through authorization.

Authorization is implemented through policy-based resource accessibility. In an organization (or society) where there is a hierarchy, there will be certain functions allowed to certain levels in the hierarchy. A clerk in a corporation may have authorization to approve an expense claim less than a specified threshold, supervisors might have a higher limit, and vice-presidents might have a still higher limit. Similarly, role-based security will be used when an application requires multiple layers of authorization and approvals to complete an action. Privilege management infrastructure together with the role-based authorization allows the administration and enforcement of user privileges and transaction entitlements. In the authorization process, users are checked to see if they have the required rights to access the resource. If they have been granted the required rights, they can access the resource, otherwise they are denied access.

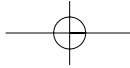
## Non-repudiation

Authentication and **Non-repudiation** have some overlapping properties. Authentication is a process by which we validate the identity of the parties involved in a transaction. In non-repudiation we identify the identity of these parties beyond any point of doubt. Non-repudiation can be considered as authentication with formal record. These records will have legal bindings. Like a signature in a cheque, using digital signature we achieve non-repudiation.

## Availability

Media management is not within the scope of security protocols and algorithms. However, media management is part of the larger security framework. Media management is needed to ensure **availability** of service. For a message a confidentiality may be maintained; also, the integrity is intact but an attacker can manipulate the media to make sure that the message does not reach the destination. This is like there is no theft of power, power quality is good, but someone blows the transmission line of the power grid.

Attack on availability happens for industrial espionage or from political motivation. During a festive season, one company may target to block the e-commerce site of a competition. In a social framework, someone may try to stop people's voice by using threats



or other means of intimidation to compel the author to remove the web page. If these methods prove unsuccessful, various denials of service attacks can be launched against the site to make it impossible to access. In less high-profile cases, people often enjoy far less support for exposing corruption or criticizing employers and particularly litigious organizations. Also, there need to be some way where terrorist organizations or dictators cannot block the mass opinion. This field of research area is called **Censorship-resistant Publishing**. Censorship-resistant publishing is achieved through document entanglement.

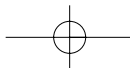
## Trust

Computers rely on user authentication and access control to provide security. Within a network, it may be safe to assume that the keyholder is authentic, and the keyholder is using the key assigned to him or her. However, these strategies are inadequate for mobile computing environments with high level of flexibility. Mobile computing lacks centralized control and its users are not all predetermined. Mobile users expect to access resources and services anywhere and anytime. This leads to serious security risks and access control problems. To handle such dynamic everchanging context, **trust**-based security management is necessary. Trust involves developing a security policy, assigning credentials to entities, verifying that the credentials fulfill the policy. Also, we need delegation of trust to third parties, and reasoning about users' access rights.

## Accounting

For any service, the service provider needs to be paid. The service can be either a content service or a network service. Accounting and billing is a very critical aspect in mobile computing environment. **Accounting** is the process by which the usage of the service is metered. Based upon the usage, the service provider collects the fee either directly from the customer or through the home network. This will be true even if the user is roaming in a foreign network, and using the services in the foreign network.

RADIUS (Remote Authentication Dial In User Service) protocol (RFC 2865) has been in use for a long time for the AAA (Authentication, Authorization, and Accounting) functions in Internet. With the demanding service requirement of mobile computing, it is now apparent that RADIUS is incapable of supporting all these complexities. A new protocol called Diameter (RFC 3588) has been released to address the AAA needs for data roaming and mobile computing. Diameter can work in both local and roaming AAA situations.



## 18.3 SECURITY TECHNIQUES AND ALGORITHMS

Generally the encryption algorithms are divided into two main groups. These are symmetric key encryption and public key encryption. In a symmetric key encryption, the key used for decryption is the same as the key for encryption. In some cases of symmetric encryption, even the algorithm used for encryption and decryption is the same. In the case of public key algorithms, the key used for decryption is different from the key used for encryption.

### 18.3.1 Stream Ciphering and Block Ciphering

In stream cipher, a bit or a byte is taken at a time and encrypted. The algorithm looks at the input plaintext as a stream of bits and encrypts them one bit (or byte) at a time as the stream progresses. In this technique, the length of the plaintext and the key size will be same. Wireless LAN (WiFi) uses stream cipher. In this methodology, the key has to be unique for every encryption. If the same key is used for multiple packets, and these packets can be captured, there is vulnerability. The other technique is block cipher. In a block cipher, one block of plaintext is taken as a whole and used to produce a ciphertext block of equal length. Typically a block of 64 bits (8 octets) or 128 bits (16 octets) is used for block cipher. Majority of cryptosystems use block cipher.

### 18.3.2 Symmetric Key Cryptography

In a symmetric key cryptography, the same key is used for both encryption and decryption. This is like a lock where the same key is used to lock and unlock. In cryptography, symmetric key algorithms are in use for centuries; that is why symmetric key algorithms are called conventional or classical algorithms as well. In this type of encryption, the key is secret and known only to the encrypting (sender) and decrypting (receiver) parties. Therefore, it is also known as a secret key algorithm. Some authors refer to symmetric key cryptography as shared key cryptography as well. This is because the same key is shared between the sender and the receiver of the message. The unique key chosen for use in a particular transaction makes the results of encryption unique. Selection of a different key causes the cipher that is produced for any given set of inputs to be different. The cryptographic security of the data depends on the security of the algorithm used and the key used to encipher the data. The strength of the security depends on the size of the key. Unauthorized recipients of the cipher who know the algorithm but do not have the correct key cannot derive the original data algorithmically. However, anyone who does have the key and the algorithm can easily decipher the cipher and obtain the original

data. Symmetric key algorithms are much faster compared to their asymmetric (public key) counterparts.

In a symmetric key cryptography, there are four components. These are **plaintext**, **encryption/decryption algorithm**, **secret key** (key for encryption and decryption), and the **ciphertext**. In Figure 18.2, if we make  $\text{Key}_E = \text{Key}_D$ , this becomes a symmetric key algorithm. There are many symmetric key algorithms. The most popular symmetric key algorithms are:

**DES:** Data Encryption Standard, this algorithm is the most widely used, researched and has had the longest life so far.

**3DES:** This is a modification of DES. In this algorithm, DES is used 3 times in succession.

**AES:** Advances Encryption Standards, this is the current accepted standard for encryption by FIPS (Federal Information Processing Standards) of USA.

**Skipjack/FORTEZZA:** This is a token-based symmetric algorithm used by defense personnel in the US.

## DES (Data Encryption Standard)

In the late 1960s, IBM set up a research project in computer cryptography led by Horst Feistel. In 1971, the project concluded with an outcome of an algorithm named Lucifer. The original algorithm used 64-bits block and 128-bits key. IBM reduced the length of the key to fit the algorithm into a single chip. This algorithm was adopted in 1977 by NIST (National Institute of Standards and Technology) as the data encryption standard (DES). A DES key consists of 64 bits of which 56 bits are randomly generated and used directly by the algorithm. The other 8 bits are used for error detection and not for encryption.

DES employs the principle of scrambling and substitution. These processes are repeated a number of times with keys to ensure that the plaintext is completely transformed into a thoroughly scrambled bit stream. The DES can be divided into the following major functions. These are:

- Permutations of bits in a block. This is the first and last step in DES. In this step the 64-bit plaintext block is rearranged through Initial Permutation **IP**. This is

done through a 64-bit register where the bits of the input block are scrambled in a particular fashion. As the last step, the reverse permutation is done through  $IP^{-1}$ .

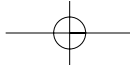
- A key dependent computation. This includes multiple rounds (iteration) of transformation through combination of permutation and substitution. This is in the core of the encryption function.
- Swapping of half blocks of data in each round.
- Key schedule; this breaks the 56-bit key into two 28-bit subkeys and use them to compute the bits in data blocks. In each iteration, the bits within the subkey are shifted to generate a new subkey.
- The key-dependent computation is run through 16 rounds. Each round uses the data from the previous round as input.

The beauty of DES algorithm is that the same algorithm is used for both encryption and decryption. DES demonstrates a very high avalanche effect. In an avalanche effect one bit of change in either the input data or the key changes many bits in the output. For example, in DES one bit of change in the input data changes 34 bits, whereas one bit change in the key affects 35 bits.

**3DES (Triple DES):** With the increase of processing power available in PC, 56 bits of key became vulnerable for attack. Therefore, to protect the investment and increase security 3DES (commonly known as Triple DES) was proposed. 3DES uses the same DES algorithm three times in succession with different keys. This increases the keysize resulting in higher security. Also, as the fundamental algorithm in 3DES is practically the DES, it is easily adaptable without additional investment. There are two different flavors of 3DES. One uses two 56-bit key and the other uses three 56-bit key. By using three 56-bit key, the effective security can be increased to the key size, to 168 bits. Till today 3DES is the most widely used algorithm for symmetric cryptography.

## AES (Advanced Encryption Standard)

We have discussed that the strength of security of a cryptographic algorithms depends on the size of the key. The larger the size of the key, the longer it takes to decipher the encrypted data through brute force. With GHz of computing power easily available, 56-bit key size is found to be unsafe today. To overcome these challenges, 3DES became popular. However, 3DES was quite slow. Also, scientists found that the 64-bit block which both DES and 3DES use, may not be the best. A higher block size is desirable from efficiency and security point of view.



## 602 *Mobile Computing*

---

To overcome these drawbacks, in 1997 NIST (National Institute of Standards and Technology) in US issued a call for algorithms for advanced encryption standard or AES. According to the call for proposal, the AES standard was to have equal or better security compared to 3DES and more efficient than the 3DES. NIST also specified that AES had to be a symmetric cipher with block size of 128 bits. Also, it has to support keys of size 128-bits, 192-bits, and 256-bits. Many algorithms competed for the AES standard. Following a rigorous evaluation process in November 2001, NIST selected the Rijndael as the AES algorithm. Rijndael is named after two researchers from Belgium who developed the algorithm. They were Joan Daemen and Vincent Rijmen. Rijndael was designed to have the following characteristics:

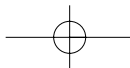
- Resistance against all known attacks
- Design simplicity
- Speed and code compactness on a wide range of platforms.

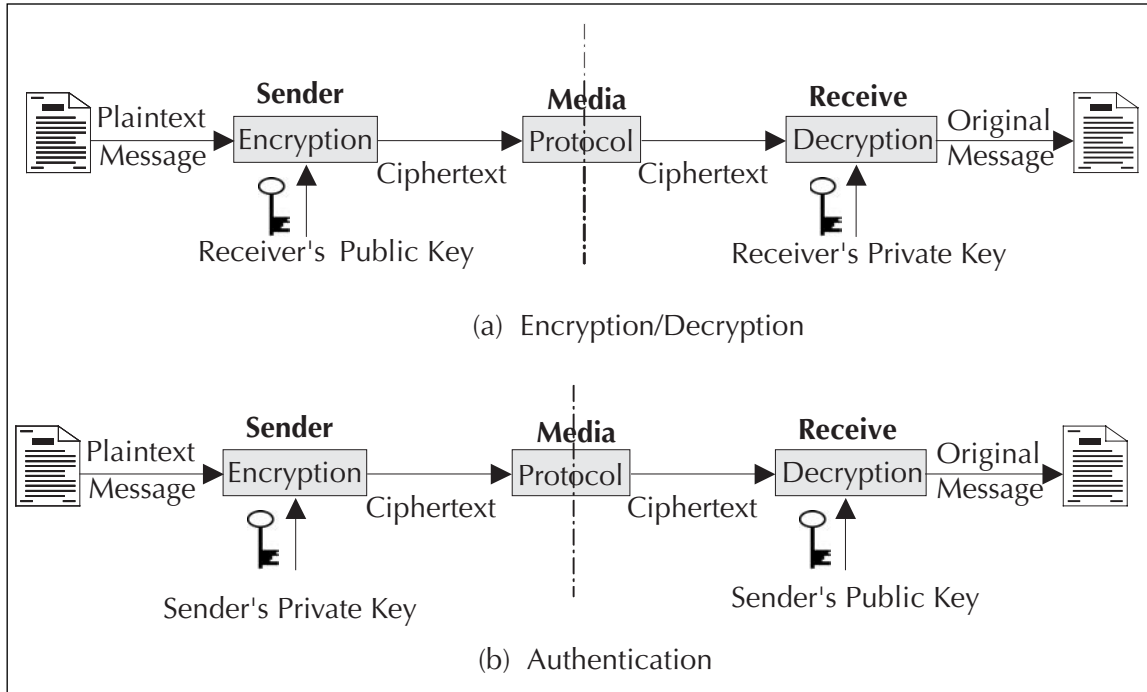
Like DES, AES also uses permutation and substitution. However, AES does not use Feistel structure. In a Feistel structure, one half of the data block is used to modify the other half of the data and then swapped.

### 18.3.3 Public Key Cryptography

In symmetric key encryption we use the same key for both encryption and decryption. In public key cryptography we use two different keys, one key for encryption and a different key for decryption. As there are two different keys used, this is also called asymmetric key cryptography. The development of public key cryptography can be considered as the greatest advance in the history of cryptography. Public key cryptosystem is based on mathematical functions rather than permutation and substitution. However, it is not true that the public key cryptosystem is more secure for general purpose. There is nothing in principle, which makes one algorithm superior to another from the point of view of resisting cryptanalysis. It is computationally infeasible to derive the decryption key given only the encryption key and knowledge of the cryptographic algorithm. The encryption key and the decryption key together form a key pair. One of these keys from the key pair is made public and the other one kept private or secret. That is why this algorithm is called public key cryptosystem.

Whitfield Diffie and Martin Hellman in 1976 came up with the principle of asymmetric key or public key cryptography. Public key cryptography proposed by Diffie and Hellman solved two difficult problems of **Key distribution** and **digital signature** in





**Figure 18.3** Public key cryptography

cryptography. In public key cryptography, there are six components (Figure 18.3). These are:

- **Plaintext:** This is the human readable message or data given to the public key algorithm as input for encryption.
- **Ciphertext:** This is the scrambled data produced as output of the encryption algorithm. This is a unique data and depends only on the unique key used for encryption.
- **Encryption algorithm:** This is the algorithm that does computation and various transformations on the input plaintext. The output of the transformation is too garbled to be decipherable for an intruder.
- **Decryption algorithm:** This algorithm does the reverse function of the encryption algorithm. This function accepts the ciphertext as input and does some transformation on the data so that the original data is recovered.
- **Public key:** This is one of the keys from the key pair. This key is made public for anybody to access. This key can be used either for encryption or decryption.

- **Private key:** This is the other key from the key pair. This key is called the private key, because this is kept secret. This can be used either for encryption or decryption.

There are three public key cryptosystems most widely used today. These are **Diffie Hellman**, **RSA**, and **Elliptic curve**.

The methodology used for encryption of data and the digital signature is different. During the encryption, the sender uses the public key of the receiver. This is because only the receiver should be able to decrypt the message using his or her own secret private key. If there is a surrogate who is able to intercept the encrypted message, he will not be able to decrypt the message, as the key required to do so is the private key. Receiver's private key is kept secret with the receiver. The methodology used for authentication or digital signature is just reverse. In case of signing the transaction, the private key of the sender is used by the sender. The receiver uses the public key of the sender to read the signature. This authenticates that the transaction was indeed done by the sender.

## Diffie Hellman

Whitfield Diffie and Martin Hellman first introduced the notion of public key cryptography in 1976. In Diffie Hellman technique, secret keys are never exchanged. However, the technique allows two parties to arrive at a secret key through the usage of public keys. Communicating parties select a pair of private and public keys. Public keys are exchanged. The shared secret key is generated from the private key and the public key of the other party.

Let us assume that there are two parties A and B. A and B choose some prime number  $p$  and another number  $g$  less than  $p$ . These numbers are selected and made available to both A and B in advance. The steps followed in Diffie Hellman algorithms for key generation are as follows:

1. Let these  $p$  and  $g$  be:  $p = 13$  and  $g = 3$ ;
2. A chooses a random number  $S_A$ . This number is kept secret as a private key with A. Let this number be 5.
3. B chooses a random number  $S_B$ . This number is kept secret as a private key with B. Let this number be 7.

4. A takes  $g$  and raises it with his secret key  $S_A$  modulo  $p$ . This will be  $T_A = (g \wedge S_A) \bmod p \Rightarrow (3 \wedge 5) \bmod 13 = (243) \bmod 13 = 9$ . This number 9 is A's public key. A already has chosen 5 as his private key.
5. B takes  $g$  and raises it with his secret key  $S_B$  modulo  $p$ . This will be  $T_B = (g \wedge S_B) \bmod p \Rightarrow (3 \wedge 7) \bmod 13 = (2187) \bmod 13 = 3$ . This number 3 is B's public key. B has already chosen 7 as his private key.
6. Public keys of A and B are exchanged. This means A send the public key 9 to B and B send his public key 3 to A over a public channel like Internet.
7. A takes B's public key and raises it with his own private key mod  $p$ . Therefore, we now have  $K_A = (T_B \wedge S_A) \bmod p \Rightarrow (3 \wedge 5) \bmod 13 = (243) \bmod 13 = 9$ .
8. B now takes A's public key and raises it with his own private key mod  $p$  in a similar fashion as A. The result will be  $K_B = (T_A \wedge S_B) \bmod p \Rightarrow (9 \wedge 7) \bmod 13 = (4782969) \bmod 13 = 9$ .
9. The value of  $(T_A \wedge S_B) \bmod p = (T_B \wedge S_A) \bmod p = 9$ . Though  $K_A$  and  $K_B$  have been calculated by A and B independently; it will always be equal. Therefore, these keys  $K_A$  and  $K_B$  can now be used by A and B as the shared key for payload encryption.

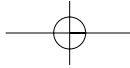
Neither A nor B shared their secret key for use in symmetric encryption, but arrived at that using some properties of modulo arithmetic with prime numbers. The example above may look trivial. However, when these numbers are large, nobody can calculate the key just by knowing  $p$ ,  $g$  and  $S_x$  in a reasonable period of time. An eavesdropper could not compute discrete logarithm, i.e., figure out  $K_A$  based on seeing  $S_B$ .

## RSA

RSA is named after its inventors R.L. Rivest, A. Shamir and L. Adleman. It is a public key algorithm that does encryption/decryption, authentication, and digital signature. The key length is variable and the most commonly used key size is 512 bits. The key length used In India by CCA (Controller of Certifying Authorities) is 2048 bits. Key length can be large for higher security; the key length can be smaller for better efficiency. The plaintext data block is always smaller than the key length. However, the ciphertext block is the same as the key length. RSA is much slower than symmetric key encryption. That is why RSA is generally not used for payload encryption. RSA is used primarily for encrypting a secret key for key exchange.

The RSA algorithm works as follows:

1. Choose two prime numbers  $p$  and  $q$ .



2. Multiply  $p$  and  $q$  to generate  $n$ .  $n$  will be used as the modulus.
3. Calculate  $\Phi(n) = (p - 1) * (q - 1)$ .  $\Phi(n)$  is the Euler's totient function.  $\Phi(p)$  is the number of positive integers less than  $p$  and relatively prime to  $p$ .
4. Choose a number  $e$  such that it is relatively prime to  $\Phi(n)$ .
5. Find  $d$  such that it is multiplicative inverse of  $e$ ;  $d = e^{-1} \text{ mod } \Phi(n)$ .
6.  $(e, n)$  is the public key and  $(d, n)$  is the private key
7. To encrypt we use the formula (Ciphertext block) = (Plaintext block)<sup>e</sup> mod  $n$
8. To decrypt we use the formula (Plaintext block) = (Ciphertext block)<sup>d</sup> mod  $n$

Let us take an example where we choose two prime numbers  $p = 7$  and  $q = 17$ .

Calculate  $n = p * q = 7 * 17 = 119$

Find the value of  $\Phi(n)$  using the formula  $\Phi(n) = (p - 1) * (q - 1) = (7 - 1) * (17 - 1) = 6 * 16 = 96$ .

Now we need to select an  $e$ .  $e$  will be relatively prime to  $\Phi(n)$  and less than  $\Phi(n)$ . We can see that 2, 3, 4 have factors with 96, therefore, are not relatively prime. Whereas, 5 is relatively prime to 96. Therefore, we can choose  $e$  to be 5.

We know that  $d * e = 1 \text{ mod } \Phi(n)$ , which in other words  $d * e = ((Y * \Phi(n) + 1) \text{ mod } \Phi(n))$ . To find the value of  $d$ , we use the formula  $((Y * \Phi(n) + 1) / e)$ . Replace  $Y$  with 1 then 2 then 3 and so on until we get an Integer. When we set  $Y = 4$ , the equation evaluates:

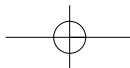
$$d = (4 * 96 + 1) / 5 = (384 + 1) / 5 = 385 / 5 = 77$$

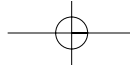
Therefore, we get  $d = 77$ . We have just generated our key pair. The public key is  $(5, 119)$  and private key is  $(77, 119)$ . We can now use this to encrypt and decrypt values.

To encrypt we use the formula

(Ciphertext block) = (Plaintext block)<sup>e</sup> mod  $n$ . Assuming that the plaintext block is 8 bits long and the value is 65. Therefore, the ciphertext will be  $(65 \wedge 5) \text{ mod } 119 \Rightarrow (1160290625) \text{ mod } 119 = 46$ . To decrypt, we use the formula (Plaintext block) = (Ciphertext block)<sup>d</sup> mod  $n \Rightarrow (46 \wedge 77) \text{ mod } 119 = (1.077340631679169568093835458385e+128) \text{ mod } 119 = 65$

The example above may look trivial and someone may think that by knowing  $(5, 119)$  one can easily find out  $d$ . This is almost impossible if the numbers are large,





for example 128 bits long. Also, to know the private key, the eavesdropper needs to evaluate  $p$  and  $q$  from  $n$ . The eavesdropper has to factorize the number  $n$  to get the two large prime numbers, which is extremely hard even in a huge timeframe. RSA uses the complexity in prime factorization.

## Elliptic Curve

A majority of the products and standards that use the public key cryptography use RSA for encryption, authentication, and digital signature. Due to extensive research in cryptanalysis in RSA and increase in availability of computing power, some vulnerabilities of RSA have been discovered. There are subexponential algorithms available today for breaking RSA and Diffie-Hellman algorithms. To overcome these threats, the size of the RSA key has been increasing over time. This puts a tremendous demand on computing power. Elliptic Curve Cryptography (ECC) has shown a lot of promise for higher security with lesser resource. Elliptic curve cryptography was proposed by Victor Miller and Neal Koblitz in the mid 1980s. Till date there is no subexponential algorithms available to break ECC. An elliptic curve is the set of solutions  $(x,y)$  to an equation of the form  $y^2 = x^3 + ax + b$ , together with an extra point  $O$  which is called the point at infinity.

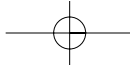
ECC is believed to offer a similar level of security with a much smaller size of key. For example, it is claimed that the level of security that 1024 bits of RSA provide can be achieved by 160 bits of ECC. A 210-bit key of ECC is equivalent to 2048 bits of RSA. This makes ECC very attractive for small footprint devices like cell phones or PDAs.

### 18.3.4 Hashing Algorithms

Hashing functions are one-way functions used for message digests. Hash function takes an input data of any size and produces an output stream of some fixed size. The outputs are collision free. This means that two different inputs will not produce the same output. It is also not possible to derive the input from a known output. This means that if we have a message digest, it is impossible to derive the original message. The most commonly used hash functions are MD5 and SHA-1.

#### MD5

MD5 (Message Digest version 5) hashing algorithm is described in RFC 1321. The MD5 algorithm is an extension of the MD4 message-digest algorithm and is slightly



slower than MD4. The MD5 algorithm takes a message of arbitrary length as input and produces a 128-bit 'message digest' as output. The algorithm processes 512 bits of the input message in blocks. The digest produced by the algorithm can also be considered as a 'fingerprint' of the message. It is conjectured that it is computationally infeasible to produce two messages having the same message digest. It is also conjectured that it is computationally infeasible to produce any message having a given message digest. The MD5 algorithm is intended for digital signature applications in a public key cryptosystem.

## SHA

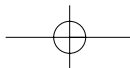
The Secure Hash Algorithm (SHA) was developed by the NIST (National Institute of Standards and Technology). SHA was first published in 1993. Later in 1995, a revised version of the algorithm was published as SHA-1. SHA processes input in 512 bits block and produces 160 bits of output. Like MD5, SHA-1 is also based on MD4 algorithm. As both MD5 and SHA-1 are based on MD4, they are quite similar in nature. However, as SHA-1 generates a longer digest of 160 bits compared to 128 bits by MD5, it is considered to be more secure.

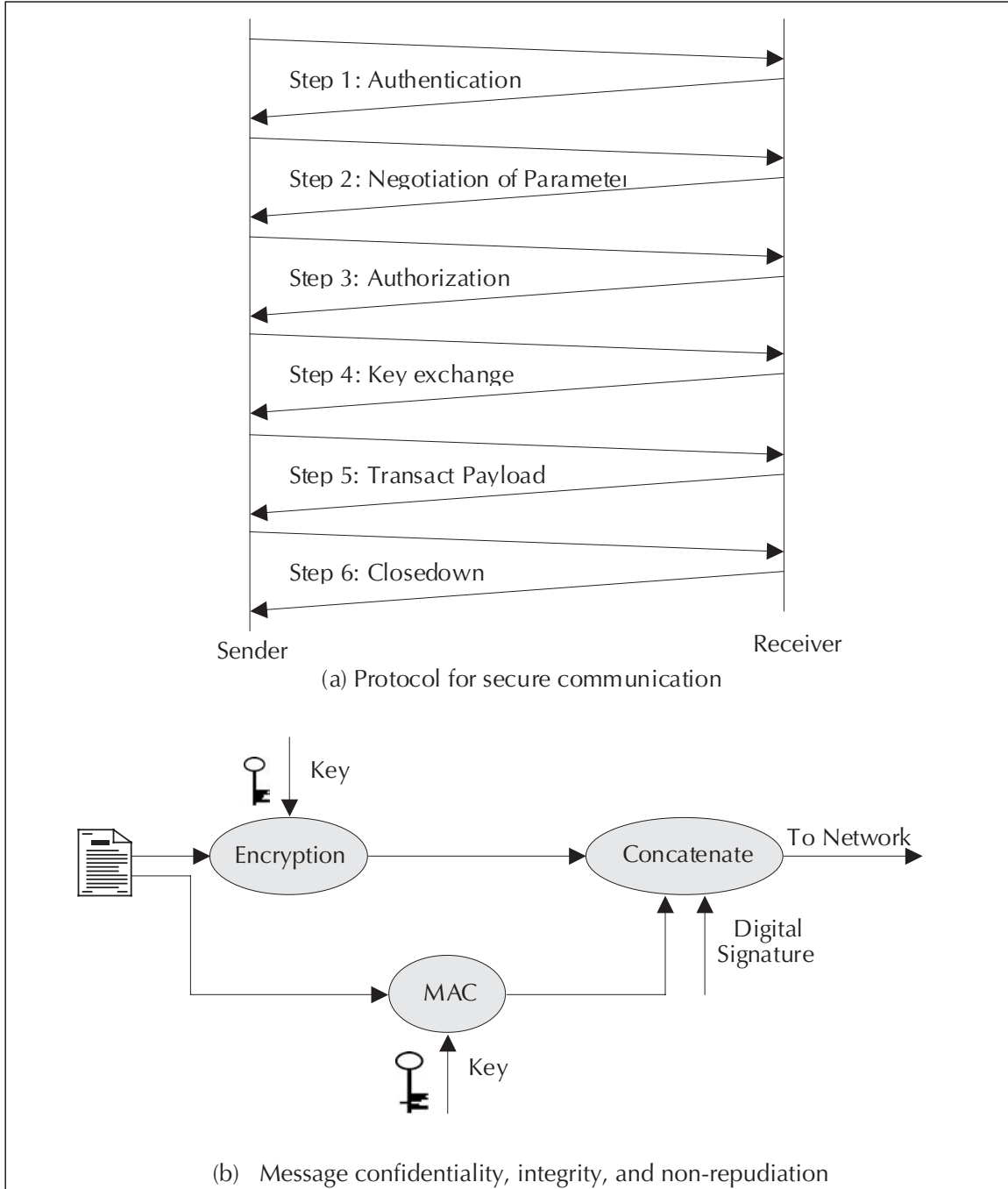
## MAC

MAC stands for Message Authentication Code. MAC is used to do the integrity check on the message. A secret key is used to generate a small fixed size data block from the message. This is similar to a checksum of the message. Both the sender and the receiver share the same secret key for MAC. When the sender has a message to be sent to the receiver, the message is sent along with the MAC. The receiver receives the message; and calculates the MAC from the message and the shared key. The receiver checks the MAC received from the sender. If they are the same, the message is considered to be in perfect state. HMAC is another mechanism for message authentication using cryptographic hash functions like MD5, or SHA-1, in combination with a secret shared key. HMAC has been defined in RFC 2104. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

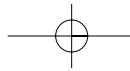
## 18.4 SECURITY PROTOCOLS

To provide confidentiality, integrity etc. we need to use different algorithms. However, we need to device protocols that will use these algorithms in such a fashion that vulner-





**Figure 18.4** Security protocols

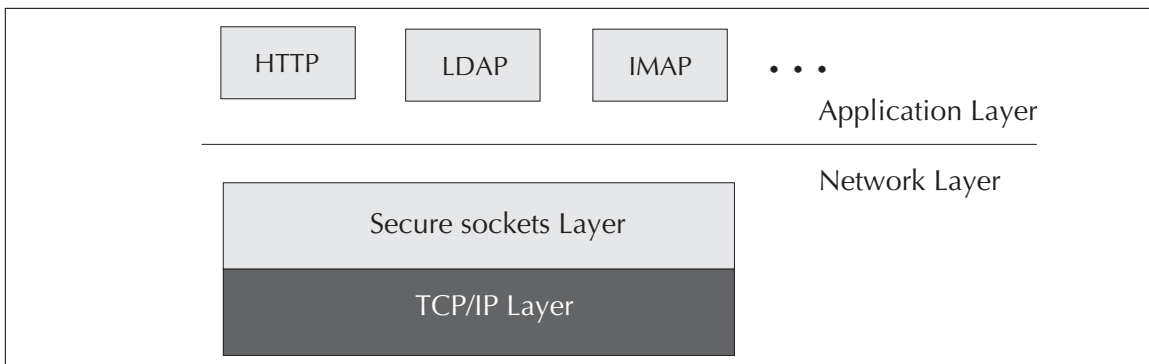


abilities are eliminated and security is ensured. The protocol needs to be so robust that a masquerader is unable to get the message being sent. The protocols need to ensure that if the masquerader is able to modify the message, we can detect it. There are many protocols for secured communication. One such protocol is depicted in Figure 18.4. However, the most popular protocol is SSL (Secured Socket layer—section 18.4.1). SSL was originally developed by Netscape. The Internet standards for TLS (Transport Layer Security—section 18.4.2) and WTLS (Wireless Transport Layer Security—section 18.4.3) have been derived from the SSL protocol.

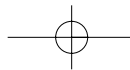
### 18.4.1 Secured Socket Layer (SSL)

The Secured Socket Layer or SSL protocol is used to provide security of data over public networks like Internet. It runs above the TCP/IP protocol layer and below higher level protocols such as HTTP or IMAP (Figure 18.5). SSL allows both machines (server and the client) to establish a secured encrypted channel so that all the data transacted between them are confidential and tamper-resistant.

Public-key encryption provides better authentication techniques. On the other hand, symmetric key encryption is much faster than the public key encryption. The SSL protocol uses a combination of both public key and symmetric key encryption. An SSL session begins with **SSL handshake**. SSL handshake allows the server to authenticate itself to the client using public-key techniques. Optionally, the handshake also allows the client to authenticate itself to the server. It then allows the client and the server to cooperate in the creation of symmetric key. It then uses this shared key for payload encryption, decryption, and tamper detection during the session that follows.



**Figure 18.5** SSL layer



## 18.4.2 TLS

Transport Layer Security or TLS in short is a security protocol to offer secured communication at the transport layer. TLS protocol is the Internet standard and based on the SSL 3.0 protocol specification. According to RFC 2246 (TLS Protocol Version 1.0), the primary goal of the TLS protocol is to provide privacy and data integrity between two communicating applications. At the lower levels, TLS uses TCP transport protocol. The TLS protocol is composed of two layers: the TLS Handshake Protocol and the TLS Record Protocol.

The TLS Handshake Protocol provides connection security that has three basic properties:

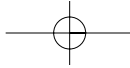
1. Peer's identity can be authenticated using asymmetric or public key cryptography (e.g., Diffie-Hellman, RSA, etc.).
2. The negotiation is reliable: no attacker can modify the negotiation communication without being detected by the parties in the communication.
3. The negotiation of a shared secret is secured: the negotiated secret is unavailable to anybody eavesdropping in the middle of the connection.

TLS Record Protocol provides connection security that has two basic properties:

1. Privacy: The confidentiality of the data is maintained through encryption. Symmetric cryptography is used for data encryption (e.g., AES, DES, RC4, etc.). Keys for symmetric encryption are generated uniquely for each connection. These encryption algorithms are negotiated by the TLS Handshake Protocol.
2. Integrity: The connection is reliable. Message transport includes a message integrity check using a keyed MAC. Secure hash functions (e.g., SHA, MD5, etc.) are used for MAC computations.

## 18.4.3 WTLS

The transport layer security protocol in the WAP architecture is called the Wireless Transport Layer Security or WTLS in short. WTLS provides functionality similar to TLS 1.0 (see Chapter 8) and incorporates new features such as datagram support, optimized handshake and dynamic key refreshing. The WTLS layer operates above the transport protocol layer similar to TLS. WTLS provides the upper-level layer of WAP with a secure transport service interface that preserves the transport service interface below it. In addition, WTLS provides an interface for creating and terminating secure connections. The



primary goal of the WTLS layer is to provide privacy, data integrity and authentication between two communicating applications. The WTLS protocol is optimized for low-bandwidth bearer networks with relatively long latency.

## 18.4.4 Multifactor Security

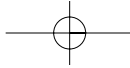
In a security system larger key implies higher security. This is simply because, larger key means larger lock. However, it may not be always possible to keep on increasing the size of the key. Therefore, we keep on looking for alternate methods of increasing security. One such method is splitting the key and distributing it. For example, in a bank, a locker cannot be opened with one key. It requires multiple keys. One key belongs to the customer; the other key is with the bank employee. Both the keys need to be used to open the locker. Take the example of ATM, where an ATM card and the PIN are required to withdraw cash. This technique is called multifactor security. These factors are generally a combination of 'what you have', 'what you know', and 'what you are'. Multifactor security can be a combination of any of the following factors.

### What You Have

- Magnetic stripe card
- Private key protected by password
- Smart card
- Hardware token
- RF badge
- Physical key.

### What You Know

- Password
- Pass Phrase
- PIN (Personal Identification Number)
- Answer to some personal questions
- Sequence of numbers
- Predetermined events.



### Who You Are

- Fingerprint
- Voice Recognition
- Retinal Scan
- Hand Geometry
- Visual Recognition
- Face (picture in passport)
- Other biometric identities.

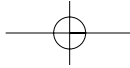
Most of the multifactor security systems in use today are two-factor ones. However, for defense systems and high security establishment three-factor securities are used. In a two-factor security any two of the above factors are used. In a three-factor security, one each from the above factors are used.

### 18.4.5 Digital Watermark

Watermarks are being used for a long time as a security measure. If we take a 100-rupee currency note of Reserve Bank of India and hold it in front of light, we can see Gandhi's face on the white circle. This is called the watermark in the currency note. If we photocopy a currency note using a color photocopier, we will not be able to copy the watermark. The term 'digital watermark' refers to a pattern of information inserted into a file. The file can be a digital audio file, digital video file, or a data file that identifies the file's copyright information (author, rights, etc.). The purpose of digital watermark is to provide copyright protection for intellectual property that is in digital format. Unlike printed watermark which are intended to be visible, digital watermarks are designed to be completely invisible. In the case of audio clips, the watermarks are inaudible. The information representing the watermark is scattered throughout the file in such a way that it cannot be identified, manipulated or reproduced.

### 18.4.6 Key Recovery

Encryption is an important tool for protecting the confidentiality of data. This data can be either data on transit over a network or a static data in a file. When suitably strong encryption algorithms are employed and implemented with appropriate assurance,



encryption can prevent the disclosure of data to unauthorized parties. However, the unavailability, loss or corruption of the keys may prevent legitimate parties from accessing the data. For law enforcement agencies it may be sometime necessary to decrypt encrypted data. To facilitate authorized access to encrypted data in the face of such situations, there are needs for key recovery procedures and standards. This type of systems is called Key Recovery System (KRS). KRS will enable authorized persons to recover plaintext from encrypted data when the decryption key is not otherwise available. Key recovery is achieved through different key recovery techniques and key recovery information (KRI). Key recovery information refers to the aggregate of information needed by a key recovery technique to recover a target key. In third party systems like a Certification Authority (CA), the KRI is securely stored with the CA.

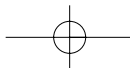
## 18.5 PUBLIC KEY INFRASTRUCTURE

Public Key Infrastructure or PKI in short consists of mechanism to securely distribute public keys. PKI is an infrastructure consisting of certificates, a method of revoking certificates, and a method of evaluating a chain of certificates from a trusted root public key. The framework for PKI is defined in the ITU-T X.509 Recommendation. PKI is also defined through RFC3280. In RFC3280 the goal of PKI is defined as ‘to meet the needs of deterministic, automated identification, authentication, access control, and authorization functions. Support for these services determines the attributes contained in the certificate as well as the ancillary control information in the certificate such as policy data and certification path constraints.’

**PKIX** is the Internet adaptation for PKI and X.509 recommendation suitable for deploying a certificate-based architecture on the Internet. PKIX also specifies which X.509 options should be supported. RFC2510, RFC2527 and RFC3280 define the PKIX specifications.

### 18.5.1 Public Key Cryptography Standards

Public-key Cryptography Standards or PKCS in short comprises of standards proposed and maintained by RSA lab. These standards are accepted as de-facto standards for public key cryptography helping interoperability between applications using cryptography for security. Most of the crypto libraries available today support PKCS standards. PKCS standards consist of a number of components, which are defined through PKCS #1, #3, #5, #6, #7, #8, #9 #10, #11, #12, #13 and #15.



- **PKCS #1, RSA Encryption Standard:** PKCS #1 describes a method for encrypting data using the RSA public-key cryptosystem. Its intended use is in the construction of digital signatures and digital envelopes as described in PKCS #7. Digital enveloping is a process in which someone ‘seals’ a plaintext message in such a way that no one other than the intended recipient can open the sealed message. PKCS #1 also describes syntax for RSA public keys and private keys.
- **PKCS #2:** Incorporated as part of PKCS #1.
- **PKCS #3, Diffie-Hellman Key Agreement Standard:** PKCS #3 describes a method for implementing Diffie-Hellman key agreement whereby two parties, without any prior arrangements, can agree upon a secret key that is known only to them.
- **PKCS #4:** Incorporated as part of PKCS #1.
- **PKCS #5, Password-Based Encryption Standard:** PKCS #5 describes a method for encrypting an octet string with a secret key derived from a password. PKCS #5 is generally used for encrypting private keys when transferring them from one computer system to another, as described in PKCS #8.
- **PKCS #6, Extended-Certificate Syntax Standard:** PKCS #6 describes syntax for extended certificates. An extended certificate consists of an X.509 public-key certificate and a set of attributes, collectively signed by the issuer of the X.509 public-key certificate.
- **PKCS #7, Cryptographic Message Syntax Standard:** PKCS #7 describes a general syntax for data that may have cryptography applied to it, such as digital signatures and digital envelopes.
- **PKCS #8, Private-Key Information Syntax Standard:** PKCS #8 describes a syntax for private-key information. PKCS #8 also describes syntax for encrypted private keys.
- **PKCS #9, Selected Attribute Types:** PKCS #9 defines selected attribute types for use in PKCS #6 extended certificates, PKCS #7 digitally signed messages and PKCS #8 private-key information.
- **PKCS #10, Certification Request Syntax Standard:** PKCS #10 describes a syntax for certification requests. A certification request consists of a distinguished name, a public key and optionally a set of attributes, collectively signed by the entity requesting certification. Certification authorities may also require non-electronic forms of request and may return non-electronic replies.

- **PKCS #11, Cryptographic Token Interface Standard:** This standard specifies an API, called Cryptoki to devices which hold cryptographic information and perform cryptographic functions.
- **PKCS #12, Personal Information Exchange Syntax Standard:** This standard specifies a portable format for storing or transporting a user's private keys, certificates, miscellaneous secrets, etc.
- **PKCS #13, Elliptic Curve Cryptography Standard:** It will address many aspects of elliptic curve cryptography, including parameter and key generation and validation, digital signatures, public-key encryption, and key agreement.
- **PKCS #15, Cryptographic Token Information Format Standard:** PKCS #15 is intended at establishing a standard which ensures that users, in fact, will be able to use cryptographic tokens to identify themselves to multiple, standards-aware applications, regardless of the application's cryptoki provider.

## 18.5.2 Storing Private Keys

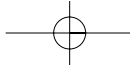
For optimum security, some security information needs to be stored in tamper-resistant storage. This will help in protecting some of the sensitive data like private keys. Also, to provide application level security, part of the security functionality needs to be performed through this tamper-resistant device. The WAP Identity Module (WIM) is designed to address all these needs. WIM will be used to perform WTLS and application level security functions. In a GSM, GPRS or 3G phones, it can be the SIM (Subscriber Identity Module) or USIM (Universal SIM) card containing additional functionality of the WIM or an external physically separate smart card. Use of generic cryptographic features with standard interfaces like PKCS#15 makes it possible to use the WIM for non-WAP applications like SSL, TLS, S/MIME etc.

## 18.6 TRUST

A portable computer never connected to a network, a standalone computer, never exposed to any unknown environment, can be assumed to be safe and secure. What happens to the security if we connect the same computer to a small private network? What happens if we connect the same computer to the Internet? What happens if we take this computer out in a football stadium and connect to the Internet over WiFi? The question is, can we trust these environments?

In early days, business was always face-to-face. In those days business used to be carried out amongst people who knew each other and in close physical proximity. In those days, one handshake literally closed the deal. The problem posed by mobile computing today is very much like that faced by business in the second half of the nineteenth century. During that time, the growth of transportation and communication networks in the form of railroads and telegraphs formed national markets and people were forced to do business with people whom they had never met. Let us take some examples. When a person searches the web for some authentic information on earthquake, what are the options? The obvious answer is to use an Internet search engine like Google. There are shops, forums, music groups with name earthquake. How do we know out of a few million hits, which are authentic information on earthquake? It may be relatively easy for a human being to determine whether or not to trust a particular web page. But is it that easy for software agents in our computers? Like in a database, can we form a SQL like query to extract an authentic technical research paper on earthquake from the Internet? In another example, let us assume for the moment that you are 55 years old and having a chest pain with sweating and vomiting; will you go to Google and give a keyword 'chest pain doctor' to look for medical help? The question, therefore, is 'Which information sources should my software agent believe?' This is equally important like the question "Which agent software should I believe and allow to access my information source?" If we look into these questions carefully we will find that first question is about trust and the second question is about security. In mobile computing, we need to address both.

We said the question, 'Which agent software should I believe and allow to access my information source?' relates to security. However, there is a catch. Suppose a person by name Anita tries to access my information source. My agent denies access to her. She then produces a certificate that she is a student in my mobile computing course, what action is expected from my agent? Of course, the agent should allow her to access my information source. This is an example of trust. The person who was not trustworthy becomes trustworthy when she produces a certificate. It is interesting to note that this certificate is not the conventional certificate as issued by a CA. Trust is explained in terms of a relationship between a trustor and a trustee. Trustor is a person who trusts a certain entity, whereas, trustee is the trusted entity. Based on the trust in the trustee, a trustor can decide whether the trustee should be allowed to access her resources and what rights should be granted. Therefore, trust plays an important role in deciding both the access rights as well as provenance of information. Trust management involves using the trust information, including recommendations from other trustees. There are different models of trust. These are direct trust, hierarchical trust and web of trust.



**Direct Trust:** In a direct trust model, parties know each other. This is like early days where everyone personally knew others in the line of business. A user trusts that a key or certificate is valid because he or she knows where it came from. Every organization today uses this form of trust in some way. Many companies today do business through Internet. However, before they start doing business over Internet, a due diligence and audit is done. Following this they do business over Internet with proper trust using trusted certificates and known key source.

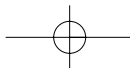
**Hierarchical Trust:** In a hierarchical system, there are a number of 'root' certificates from which trust extends. This is like the holding company establishing a trust and then member companies use this trust and key (certificate). These root certificates may certify certificates themselves or they may certify certificates that certify still other certificates down the chain. This model of trust is used by conventional CA.

**Web of Trust:** A web of trust encompasses both of the above models. A certificate might be trusted directly or trusted in some chain going back to a directly trusted root certificate or by some group of introducers. The web of trust uses digital signatures as its form of introduction. When any user signs another's key, he or she becomes an introducer of that key. As this process goes on, it establishes a web of trust. PGP (Pretty Good Privacy) uses this model of trust. PGP does not use the CA in its conventional sense. Any PGP user can validate another PGP user's public key certificate. However, such a certificate is only valid to another user if the relying party recognizes the validator as a trusted introducer.

## 18.6.1 Certificate

Digital certificate plays a significant role in establishing trust. Through a digital certificate, we can associate a name with a public key. Certificate is a signed instrument vouching that a particular name is associated with a particular public key. It is a mapping between a domain name (like mybank.co.in for example) and a public key. The structure of certificates is hierarchical originating from a trusted root certificate. For example, the root certification authority in India is called Controller of Certification Authority (CCA—<http://cca.gov.in>). CCA is responsible for generating the key pair using SHA-1 and 2048 bit RSA algorithm. CCA issues these certificates to users through different RAs (registration authority). An RA is an organization to which a CA delegates administrative functions of creation, distribution, and bookkeeping of the public-private key pair.

Here are the data and signature sections of a certificate in human-readable format taken from an example sited in the Netscape site:



Certificate:

Data:

Version: v3 (0x2)  
Serial Number: 3 (0x3)  
Signature Algorithm: PKCS #1 MD5 With RSA Encryption  
Issuer: OU=Ace Certificate Authority, O=Ace Industry, C=US  
Validity:

Not Before: Fri Oct 17 18:36:25 1997

Not After: Sun Oct 17 18:36:25 1999

Subject: CN=Jane Doe, OU=Finance, O=Ace Industry, C=US

Subject Public Key Info:

Algorithm: PKCS #1 RSA Encryption

Public Key:

Modulus:

00:ca:fa:79:98:8f:19:f8:d7:de:e4:49:80:48:e6:2a:2a:86:  
ed:27:40:4d:86:b3:05:c0:01:bb:50:15:c9:de:dc:85:19:22:  
43:7d:45:6d:71:4e:17:3d:f0:36:4b:5b:7f:a8:51:a3:a1:00:  
98:ce:7f:47:50:2c:93:36:7c:01:6e:cb:89:06:41:72:b5:e9:  
73:49:38:76:ef:b6:8f:ac:49:bb:63:0f:9b:ff:16:2a:e3:0e:  
9d:3b:af:ce:9a:3e:48:65:de:96:61:d5:0a:11:2a:a2:80:b0:  
7d:d8:99:cb:0c:99:34:c9:ab:25:06:a8:31:ad:8c:4b:aa:54:  
91:f4:15

Public Exponent: 65537 (0x10001)

Extensions:

Identifier: Certificate Type

Critical: no

Certified Usage:

SSL Client

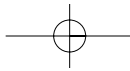
Identifier: Authority Key Identifier

Critical: no

Key Identifier:

f2:f2:06:59:90:18:47:51:f5:89:33:5a:31:7a:e6:5c:fb:36:  
26:c9

Signature:



**620** *Mobile Computing*

Algorithm: PKCS #1 MD5 With RSA Encryption

Signature:

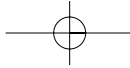
```
6d:23:af:f3:d3:b6:7a:df:90:df:cd:7e:18:6c:01:69:8e:54:
65:fc:06:
30:43:34:d1:63:1f:06:7d:c3:40:a8:2a:82:c1:a4:83:2a:fb:
2e:8f:fb:
f0:6d:ff:75:a3:78:f7:52:47:46:62:97:1d:d9:c6:11:0a:02:
a2:e0:cc:
2a:75:6c:8b:b6:9b:87:00:7d:7c:84:76:79:ba:f8:b4:d2:62:
58:c3:c5:
b6:c1:43:ac:63:44:42:fd:af:c8:0f:2f:38:85:6d:d6:59:e8:
41:42:a5:
4a:e5:26:38:ff:32:78:a1:38:f1:ed:dc:0d:31:d1:b0:6d:67:
e9:46:a8:
dd:c4
```

Here is the same certificate displayed in the 64-byte-encoded form interpreted by software:

```
-----BEGIN CERTIFICATE-----
MIICKzCCAZSgAwIBAgIBAzANBgkqhkiG9w0BAQQFADA3MQswCQYDVQQGEwJVUzERMA8G
A1UEChMITmV0c2NhcgUxFTATBgNVBAsTDFN1cHJpeWEncyBDQTAeFw05NzEwMTgwMTM2
MjVaFw05OTEwMTgwMTM2MjVaMEgxCzAJBgNVBAYTA1VTMREwDwYDVQQKEwhOZXRzY2Fw
ZTENMAsgA1UECXM EUHViczEXMBUGA1UEAxMOU3Vwcm15YSBTAeG90dHkwZ8wDQYJKoZI
hvcNAQEFBQADgY0AMIGJAoGBAMr6eZiPGfjX3uRjgEjmKiqG7SdATYazBcABu1AVyd7c
hRkiQ31FbXFOGD3wNktbf6hRo6EAmM5/R1AskzZ8AW7LiQZBcrXpc0k4du+2Q6xJu2MP
m/8WKuMOnTuvzpo+SGXelmHVChEqooCwfdiZywyZNMmrJgaoMa2MS6pUkfQVAgMBAAGj
NjA0MBEGCWCsAGG+EIBAQQEAwIAgDAfBgNVHSMEGDAWgBTy8gZzkBhHUfWJM1oxeuZc
+zYmyTANBgkqhkiG9w0BAQQFAAOBgQBTI6/z07Z635DfzX4XbAFpj1Rl/AYwQzTSYx8G
fcNAqCqCwaSDKvsuj/vwbf91o3j3UkdGYpcd2cYRCgKi4MwqdWyLtpuHAH18hHZ5uvi0
0mJYw8W2wUOsY0RC/a/IDy84hW3WWehBUqVK5SY4/zJ4oTjx7dwNmdGwbWfprqjd1A==
-----END CERTIFICATE-----
```

**18.6.2 Simple PKI**

It was thought that digital certificate would address issues related to trust. However, finally, certificates emerged as an instrument for authentication. PKCS also made some attempt to address the need of trust through PKCS#6 and PKCS#9. IETF developed yet



another standard called Simple PKI or SPKI (RFC 2692, RFC 2693) in short. SPKI defined a different form of digital certificates whose main purpose is authorization in addition to authentication. Purpose of SPKI is to define a certificate structure and operating procedure for trust management in the Internet.

## 18.7 SECURITY MODELS

We have discussed different types of security algorithms. We have also discussed security protocols. These algorithms and protocols are used to protect our assets. These assets can be either static assets in the form of priceless data in a database, files, or documents within a computer or assets in transit. The security and trust model we choose should be able to secure our assets and protect our interests. To protect ourselves from different threats, we need to look at security and trust at system level and application level.

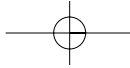
### 18.7.1 Infrastructure Level Security

Infrastructure level security offers security at the perimeter of the system. This will primarily include networks and the infrastructure. We can call this **Network security** as well. Infrastructure level security will include protecting the infrastructure or the network so that attacks from worms, viruses, Trojan horses can be prevented. Prevention from other forms of attacks like intrusion etc. different firewalls in the network are all part of the infrastructure-level security. Virtual private network (VPN) is part of infrastructure security as well.

Infrastructure level security for mobile-computing environment need to handle some additional threats compared to a wired network. For mobile-computing network, the last mile access network will be wireless in most of the situations. Therefore, at the access level, additional infrastructure security is necessary. An example is encryption in GSM using A5 algorithm. WiFi/wireless LAN networks use WEP. Some vulnerabilities have been identified in infrastructure security for WiFi; therefore, new security protocols like 802.1x and 802.11i have been proposed to take care of the over the air interface in the access network.

### 18.7.2 System Level Security

In system level security we secure our systems to protect our assets. In the security framework provided by the operating system, shells will be part of system level



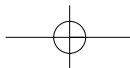
security. In authentication challenges during Unix login, or login into a mainframe computer through username, password is the system level security. Access control list (ACL), file system security, memory security etc will also be part of the system level security. It protects the system from worms, viruses and Trojan horses. Prevention from other forms of attacks like buffer overflow attacks, intrusion etc can also be part of system level security as also security protocols like SSL and TLS. There is a concept of capability-based system, where security is policy-driven and managed through capability. Even if a virus enters into such a system, or an intrusion happens, it will not be able to damage any asset in the system. One of such operating system is EROS.

Database security is part of system level security. In database security, data in the database is secured by the database software. This can be encrypting a column in a row or some special check based on ACL and capability. Most of the database software today offer security at this layer. This will be over and above the security offered by the operating system.

### **18.7.3 Policy Based Security**

Security systems implemented for wired networks in organization are primarily policy based. Effective security policies make frequent references to standards and guidelines that exist within an organization. Policy is a written down rules about what is allowed or what is not allowed in the network. Policies are usually area-specific, covering a single area. According to the RFC 2196, security policy is defined as “A security policy is a formal statement of the rules by which people who are given access to an organization's technology and information assets must abide.” For example, there could be a rule in the organization that nobody will be allowed to have a global IP address. To stop spam and mail bound viruses, there may be another rule that prevents access to external email systems (like hotmail) from the corporate network. To stop the possibility of espionage, there could be a rule that FTP from the Internet is not allowed to any machine within the intranet. A standard is typically collection of system-specific requirements that must be met by everyone. Standards are necessary when we need interoperability. A guide line is typically collections of system specific procedural specific “suggestions” for best practice. Guidelines are not requirements to be met, but are strongly recommended.

In a wired network where systems are stationary, and the network structure is static, it is possible to define security policies. In such networks, it is possible to enforce these policies or rules. However, things are different when we move to a mobile computing



environment. In a mobile computing environment, user will move from one device to another device, from one network to another network. These devices or networks may be of similar type or different types. For example user moves from a WiFi network to a CDMA2000 network or from a PalmOS to a WindowsCE. In a network with static nodes, it is possible to define a security policy. It may be possible to enforce such policies as well. However, in case of mobile computing where nodes are roaming from one network to another, it may not be practical to define a security policy and implement it. Therefore, over and above policy based security, for mobile nodes, we need object security. In object security, objects will carry their security signatures and capabilities. This is achieved through the concept of principal. Therefore, when a device moves from network to network, the device carries the security requirement and security signature with it. Principal based security system is in the process of maturing. OMA DRM (discussed in section 8.4.1) is an example of security principal.

## 18.7.4 Application Level Security

Infrastructure and system level security take care of security at the infrastructure and system level respectively. The parameters for these securities are not very flexible; most of the time vendors define them. In a mobile-computing environment we need security at the application level. Application security looks at the security at the content level. This can also be termed as **Peer to peer security**. The application at the client device will talk to the application at the server and handle security requirements end-to-end as the content may demand.

In a mobile computing environment, we cannot make any assumption related to the client context or the network context. Therefore, the security needs to be addressed at the content level, using the context awareness, J2ME, .NET, WIM or MExE (Mobile Execution Environment) environment. Using cryptographic libraries, we can build security at the application level. This security will be custom-built and can use standard algorithms or new algorithms as agreed by the peer nodes. Of course the system/infrastructure level security, if any, will be over and above the application level security.

## 18.7.5 Java Security

Security model provided by Java covers both system level security and application level security. Java system level security is provided through the 'sandbox' model. Sandbox provides a restricted environment for code execution through Java virtual machine. In

the sandbox model, local code is trusted to have full access to system resources like file system, memory etc. However, downloaded code from a remote site as an applet is not trusted. Therefore, applet can access only the limited resources provided inside the sandbox. Java supports digitally signed trusted applet. A digitally signed applet is treated like local code, with full access to resources. Digitally signed applets use public key infrastructure. Prior to transmission, the applet server signs an applet JAR file using its digital certificate. Upon receipt, the client side Java security manager verifies the signature and decides whether the origin and integrity of the application is trusted. Once the authentication is successful, the application code is delivered to the client for execution.

Java offers tools to facilitate various security-related operations. These are:

- **Keytool:** This is a command line tool. Keytool is used to manage keystore, which includes the following functions.
  - o Create public/private key pairs
  - o Issue certificate requests (which will be sent to the appropriate Certification Authority)
  - o Import certificate replies (obtained from the Certification Authority)
  - o Designate public keys belonging to other parties as trusted keys and certificates are used to digitally sign applications and applets. A **keystore** is a protected database that holds keys and certificates for an enterprise. Access to a keystore is guarded by a password. In addition, each private key in a keystore can be guarded by its own password.
- **Jar:** This is a command line tool to create JAR (Java Archive) files. The JAR file format enables users to bundle multiple files into a single archive file. Typically a JAR file will contain the class files and auxiliary resources associated with applets and applications. After importing appropriate keys into the keystore, **jarsigner** tool is used to digitally sign the JAR file.
- **Jarsigner:** This is a command line tool to sign JAR files. This is also used to verify signatures on signed JAR files. The jarsigner tool accesses a keystore that is created and managed by **keytool**, when it needs to find the private key and its associated certificate chain. The jarsigner tool prompts for needed passwords.
- **Policytool:** Unlike the other tools, this tool has a graphical user interface. Policytool is used to create and modify the external policy configuration files that define installation's security policy.

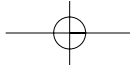
As a part of application level security, Java framework supports cryptographic library through Java cryptography architecture (JCA). JCA refers to a framework for accessing and developing cryptographic functionality for the Java platform. These cryptographic services are:

- Symmetric key encryption algorithms
- Public key encryption algorithms
- Digital signature algorithms
- Message digest algorithms
- Message authentication code generation
- Key generation algorithms
- Key exchange algorithms
- Keystore creation and management
- Algorithm parameter management
- Algorithm parameter generation
- Key factory support to convert between different key representations
- Certificate factory support to generate certificates and certificate revocation lists (CRLs) from their encodings
- Random-number generation (RNG) algorithm
- Support of SSL and TLS through http support.

Cryptographic library is available for the entire Java framework. This includes J2EE (Java 2 Enterprise Edition), J2SE (Java 2 Standard Edition), J2ME (Java 2 Micro Edition), and JC (Java Card). However, due to security reasons and resource constraints J2ME and JC functionalities are restrictive. Some of the APIs, which are available in J2EE and J2SE are not supported in Java card.

## **18.8 SECURITY FRAMEWORKS FOR MOBILE ENVIRONMENT**

Mobile applications usually span over several networks. One of these networks will be a wireless radio network. Others will be wired networks. At the boundary of any of these networks, there is a need for protocol conversion gateways. These gateways run either at



the transport layer or at the application layers. Moreover, while the user is roaming in foreign networks, there will be multiple wired networks (PLMNs) managed and controlled by different network operators. Multiple gateways and multiple networks make security challenges in mobile environments complex.

In a security system, authentication, and non-repudiation are meaningful only when these are implemented end-to-end between parties that need to authenticate each other. Authorization is a direct function of authentication; therefore, it is also an end-to-end function. Authentication, authorization, and non-repudiation must therefore be implemented at the application layer. Confidentiality and integrity on the other hand can be implemented at any layer or through multiple layers. Confidentiality can be realized by encrypting isolated legs between gateways or even end-to-end. When confidentiality is realized in isolated legs, the gateways or nodes between the legs need to be secured and trusted.

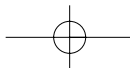
Therefore, to offer secured environment in a mobile environment, security procedures will be a combination of many procedures and functions. Following sections cover some of the vulnerabilities and techniques to offer security in Mobile environment.

### **18.8.1 3GPP Security**

In a mobile computing environment inside a campus the access network is likely to be WiFi. However, outside of the campus, access network will be one of the cellular wide area wireless networks like GPRS, CDMA, or GSM. It could also be WiMax. We have discussed WiFi security in section 10.8. We have discussed GSM security in section 5.9. We have discussed the GPRS security in section 7.3.4. We have also discussed the security issues of CDMA networks in section 9.3.5.

WiFi security is an extension of the LAN security and primarily designed for data and applications. However, security procedures for wireless wide area networks GSM, GPRS, CDMA, are designed primarily keeping the operator in mind. All these security principles mainly try to protect an operator from fraud and network misuse. None of these procedures address the security concerns of user information or the application. Current security procedures in all these wide area wireless networks failed to provide trusted environment where mobile users felt confident enough to place sensitive information over these networks.

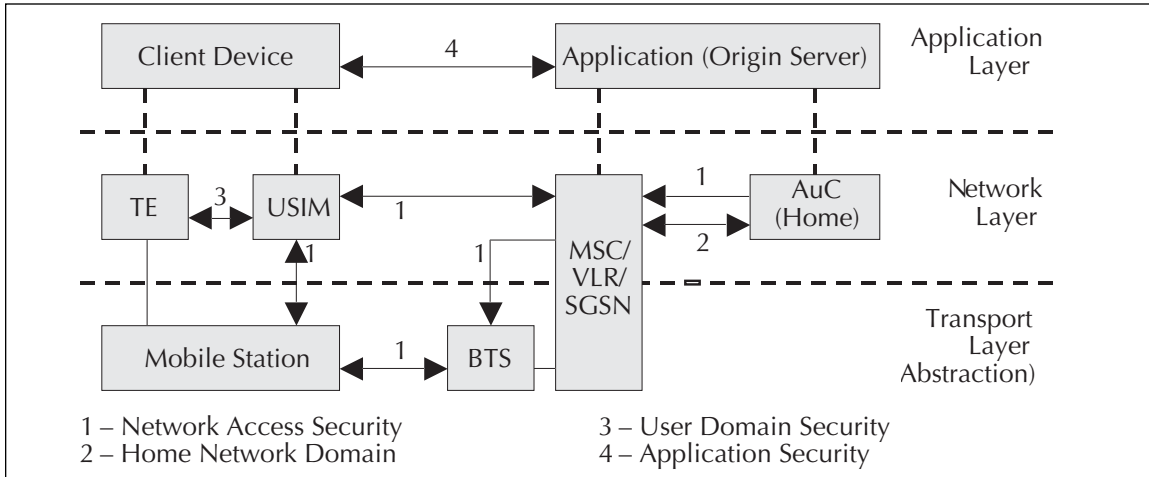
In order to perform an attack in a wireless wide area network, the adversary has to possess one or more of the following capabilities:



- **Eavesdropping.** This is the capability through which the adversary eavesdrops signalling and data traffic associated with a user. Equipment required for such attack is a modified mobile station or a radio receiver.
- **Impersonation of a user.** This is the capability whereby the adversary sends signalling and user data to the network, in an attempt to make the network believe that they originate from a genuine (target of the impersonation) user. Equipment required for such attack is a modified mobile station or a radio transmitter/receiver.
- **Compromising authentication vectors in the network.** The adversary possesses a compromised authentication vector, which may include challenge/response pairs, cipher keys and integrity keys. This data may have been obtained by compromising network nodes or by intercepting signalling messages on network links and then through brute force attack.
- **Impersonation of the network.** This is the capability whereby the adversary sends signalling and user data to the target user, in an attempt to make the target user believe that the data originate from a genuine network. Equipment required for such attack is a modified base station or a radio transmitter/receiver.
- **Man-in-the-middle.** This is the capability whereby the adversary puts itself in between the target user and a genuine network and has the ability to eavesdrop, modify, delete, re-order, replay, and spoof signalling and user data messages exchanged between the sender and the receiver. The required equipments in such attack are modified base station in conjunction with a modified mobile station.

3GPP looked into these concerns and come up with changes in the security architecture of the current wireless wide area networks. 3GPP proposed a new architecture (Figure 18.6) through following important changes:

- Changes were made to defeat the false base station attack. The extended security mechanism is now capable of identifying the network.
- Key lengths are increased to allow stronger algorithms for encryption and integrity.
- Mechanisms are included to support security within and between networks.
- Security is based within the switch rather than the base station to ensure that links are protected between the base station and switch.
- The authentication algorithm has not been defined, but guidance on choice will be given.



**Figure 18.6** 3GPP Security Architecture

- Integrity mechanisms for the terminal identity (IMEI) have been included.

## 18.8.2 MOBILE VIRTUAL PRIVATE NETWORK

Virtual Private Network (VPN) provides an end-to-end security infrastructure. This generally deals with authentication, non-repudiation, integrity, and confidentiality at a layer between the transport layer and the application layer. In section 10.8.8 we have discussed Wireless VPN with respect to WiFi. Like wireless VPN, mobile VPN is a private network over a public network (usually the Internet) to connect two endpoints. Instead of using a dedicated physical connection such as leased line, a VPN uses "virtual" connections routed through the Internet from the enterprise's private network to the remote mobile device. VPN implements this through an encrypted private connection between nodes. It generally uses IPSec and other PKI frameworks to offer confidentiality, authentication, non-repudiation (through digital signature), and integrity. With mobile VPN, mobile workers have the freedom to safely use wireless applications on their PDAs, smart phones and other handheld devices in the field as if they are in a private network.

## 18.8.3 Multifactor Security

In section 18.4.4, we have discussed multifactor security where factors could be "what you have", "what you know", "what you are". In mobile network multifactor security can

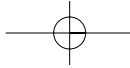
be extended to multiple networks. For example the security key (session key) in a GPRS network can be split into multiple parts and sent through Internet (TCP/IP) and AMS network. By now we know that SMS uses SS#7 network for its traffic. SS#7 network is closed and protected. SS#7 is physically more secured than the IP network.

## 18.8.4 Smartcard Security

Smart cards offer data encryption and the ability to store secret information for the purpose of authenticating the cardholder. There are various types of smartcards used in different application scenarios. One of such examples is the SIM card on a mobile phone. ETSI standard 03.48 specifies procedures for SIM card to be used as a security engine. SIM cards used in mobile phones are processor cards. Processor cards are smart cards with an inbuilt processor and memory. This local processor protects the content in the SIM and makes it tamper resistant. The 03.48 standard specifies the interoperability standards for cryptographic functions. Also, many of the SIM cards have RSA, DES, 3DES, AES algorithms implemented within the card. There are different file systems within a SIM card. These files have very stringent security controls. Files can be protected through passwords known to the user or operator. Private key and many other secret keys can be stored in these files. As these files are protected through password, even if the card is lost, these information are protected. To counter brute force attack, a smartcard processor does not allow more than 10 attempts to read a file data with wrong password. Therefore, a user can make use of the card as a security factory. Using this security factory is quite easy through Java card interfaces. In Java card technology, a Java interface is provided on the SIM card. Java cryptographic architecture (JCA) and Java programs running on the smartcard can do all these things quite easy.

## 18.8.5 Mutual and Spatial Authentication

In section 18.8.4 we have discussed how a SIM card can be used as a security factory. The SIM card can store secured information like private keys, wireless identity module, and many other private secured information. It also has various algorithms implemented. This can facilitate mutual authentication. In SSL or TLS over Internet generally client authentication is not done. However, using SIM card, we can do client authentication over wireless wide area networks. This is called mutual authentication; because, using GSM 03.08 procedures, a client can authenticate the server, also the server can authenticate the client.



SIM cards in a GSM/GPRS network store location information. This includes country, network, and base station information. This information can be obtained and sent from the mobile phone using GSM 03.48 standards specification. Location information then can be used to implement spatial authentication. For example, if the user is in a neighborhood which is insecure, access to some critical applications can be prevented.

### **18.8.6 RFID Security**

In section 4.3 we have discussed about RFID. Application areas for RFID is increasing. However, it has certain vulnerabilities. For example using the RFDump tool (<http://www.rf-dump.org/>), an adversary can detect an RFID-Tag and extract its meta information like Tag ID, Tag Type, manufacturer etc. It can even be used to rewrite the data stored in some RFID tags using either a Hex or an ASCII editor. All these vulnerabilities pose a serious threat toward RFID based systems starting from merchandise in a store to the e-passports. The US governments sometime ago decided to issue passports with RFIDs. It is nicknamed as "e-passports". However, the concerns over RFID security delayed this plan. According to the specification of e-passport, there will be 64-bit RFID tags attached in the passport that will contain name, date of birth, place of birth, a digital photograph and a digital face recognition template of the passport holder. This RFID is supposed to work only in a very close proximity. A RFID reader placed beyond the distance of more than 10 centimeters should not be able to read the content of the e-passport. However, in reality it was found that the radio tags' readable distance is as large as 30 feet. This makes the security information in the e-passport available over radio for an adversary to grab.

### **18.8.7 Mobile Agent Security**

Mobile agents are processes that can autonomously migrate from one networked computer to another. Mobile agents can be useful for many applications, especially these in Internet. For example I give my weekly shopping list to my mobile agent, which visits the web sites of all the stores in 3 kilometer radius and tells me which shop is having which fish at what price, where tomato is cheapest, where can I get my favorite pickle etc. The mobile visits all these store's web site does a shopping plan for me.

Despite its many practical benefits, mobile agent technology results in significant security threats from both malicious agents and malicious hosts. For examples, as the mobile agent traverses multiple hosts that are trusted to different degrees, its state may be changed in a way that an adversely can impact the decision making process of the agent.

### 18.8.8 Mobile Virus

Viruses are common in the PC and desktop environment. However, they were not common in mobile environment. However, things are changing; as the mobile device become more intelligent with more flexibility and higher capabilities, viruses are surfacing. Already some of them are in the wild. In June 2004, as a proof concept a virus called Cabir was developed to exploit Bluetooth vulnerability. In early November 2004 a mobile virus called "Skull.A" was reported for some models of Nokia phones. A new version named as "Skull.B" emerged in the wild, which combines Skull.A and Cabir. One more virus identified as Commwarrior.A surfaced in March 2005. This virus uses a combination of Bluetooth and MMS (Multimedia Messaging Service) to propagate. The principles these viruses use are similar in concept as the desktop viruses do.

### 18.8.9 Mobile Worm

A worm needs to propagate, execute, and reproduce in an automated fashion. To reproduce and then propagate, the worm needs to execute a piece of code (designed by the worm writer) on the target system. Therefore, it is necessary to have an execution environment available to the worm code on the target mobile device. On a mobile equipment today we have various execution environments like:

1. WAP/WML Script (MExE Classmark I)
2. JavaPhone/Personal Java (MExE Classmark II)
3. J2ME (MExE Classmark III)
4. Symbion
5. WindowsCE
6. PalmOS
7. Linux

These can access both the TCP/IP and SMS interfaces. Therefore, worms can replicate and propagate through both TCP/IP and SMS interfaces of JavaPhone, PersonalJava or J2ME framework.

Along with JavaPhone, technology on the mobile equipment, JavaCard facility is also available on the SIM cards. Using all these technologies, it will be possible to develop viruses, worms, and Trojan horses for mobile phones. These viruses and worms will be able to replicate, access the address book, use the network facility and propagate.

**REFERENCES/FURTHER READING**

1. 3G TR 33.900 V1.2.0 (2000-01) 3rd Generation Partnership Project; Technical Specification Group SA WG3; A Guide to 3rd Generation Security (3G TR 33.900 version 1.2.0), January 2000.
2. S Gindraux, From 2G to 3G: A Guide to Mobile Security, Proceedings of Third International Conference on 3G Mobile Communications Technologies, 2002.
3. Yang Kun, Guo Xin, Liu Dayou, Security in Mobile Agent System: Problems and Approaches, ACM SIGOPS Operating Systems Review, 2000.
4. Katherine M. Shelfer and J. Drew Procaccino, Smart Card Evolution, Communications Of The ACM July 2002, Vol. 45, No. 7, p83.
5. Thomas S. Messerges, Ezzat A. Dabbish, Digital Rights Management in a 3G Mobile Phone and Beyond, DRM'03, October 27, 2003.
6. Asoke K Talukder, Debabrata Das, Artificial Hygiene: Non-Proliferation of Virus in Cellular Network, Journal of Systems and Information Technology, Volume 8, 1st December 2004, pp 10–22.
7. 3GPP TS 03.48, Digital cellular telecommunications system (Phase 2+); Security mechanisms for SIM application toolkit; Stage 2, 1999.
8. Zhiqun Chen, Java Card Technology for Smart Cards, Addison Wesley, 2000.
9. Wireless Transport Layer Security, Version 06-Apr-2001: WAP-261-WTLS-20010406-a; <http://www.wapforum.com>.
10. Transport Layer Security Protocol: RFC 2246.
11. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile: RFC3280.
12. Controller for Certification Authority; Government of India: Digital Certificate, <http://cca.gov.in>.
13. Shashi Kiran, Patricia Lareau, Steve Lloyd: PKI Basics-A Technical Perspective, November 2002, <http://www.pkiforum.org>.

14. Data Encryption Standard (DES); Federal Information Processing Standard Publication, 1999 October 25, U.S. Department Of Commerce/National Institute of Standards and Technology.
15. Elliptic Curve Cryptosystem: <http://www.certicom.com>.
16. Charlie Kaufman, Radia Perlman, Mike Speciner: Network Security Private Communication in a Public World 2nd Edition; Prentice Hall of India, 2002.
17. William Stallings, Network Security Essentials: Applications and Standards; Pearson Education. 2000.
18. William Stallings: Cryptography and Network Security Principles and Practices; Pearson Education, Third edition, 2003.
19. R. Rivest, The MD5 Message-Digest Algorithm, RFC 1321, April 1992.
20. H. Krawczyk, M. Bellare and R. Canetti: HMAC: Keyed-Hashing for Message Authentication, February 1997, RFC 2104.
21. Secured Socket Layer (SSL): <http://developer.netscape.com/docs/manuals/security/pkin/contents.html>.
22. Report of the Technical Advisory Committee to Develop a Federal Information Processing Standard for the Federal Key Management Infrastructure, <http://csrc.nist.gov/keyrecovery/>.
23. Marc Waldman and David Mazieres Tangler: A Censorship-Resistant Publishing System Based On Document Entanglements, December 8, 2001.
24. WPKI, WAP-217-WPKI, Wireless Application Protocol Public Key Infrastructure Definition, Apr-2001.
25. Wireless Application Protocol Identity Module Specification, Version 05-Nov-1999, WAP Forum.
26. Burton S. Kaliski Jr.; An Overview of the PKCS Standards, published by RSA Lab <http://www.rsasecurity.com/rsalabs/pkcs/>.
27. C. Rigney, S. Willens, A. Rubens and W. Simpson, RADIUS (Remote Authentication Dial In User Service), RFC 2865, June 2000.

28. P. Calhoun, J. Loughney, E. Guttman, G. Zorn and J. Arkko: Diameter Base Protocol, RFC 3588, September 2003.
29. Whitfield Diffie and Martin Hellman: *New Directions in Cryptography*, 1976.
30. Francis Fukuyama, *The Virtual Handshake: E-Commerce and the Challenge of Trust*, <http://www.ml.com/woml/forum/ecommerce1.htm>.
31. Tim Finin and Anupam Joshi, *Agents, Trust and Information Access on the Semantic Web*.
32. Henry M Levy, *Capability-Based Computer Systems*, Digital Press, 1984.
33. Jonathan S. Shapiro and Norm Hardy: *EROS: A Principle-Driven Operating System from the Ground Up*, *IEEE Software Magazine*, January 2002.
34. Java Security:  
<http://java.sun.com/products/jdk/1.2/docs/guide/security/CryptoSpec.html>.
35. <http://www.sans.org/resources/policies/>.
36. [http://www.tbs-sct.gc.ca/pubs\\_pol/gospubs/TBM\\_12A/gsp-psg1\\_e.asp#poli](http://www.tbs-sct.gc.ca/pubs_pol/gospubs/TBM_12A/gsp-psg1_e.asp#poli).

## REVIEW QUESTIONS

- Q1: What are the different components of information security?
- Q2: What do you understand by security algorithms and security protocols? What are the differences between them? How are they related?
- Q3: Describe symmetric key and Public key encryption. If you are required to design a security system when will you use which algorithms?
- Q4: Explain the security framework for mobile computing. How do we ensure security in a mobile environment through Mobile VPN?
- Q5: Give examples of RFID security vulnerability?