CHAPTER 8

Cryptography

Introduction

It is easy for someone to read data if it is in plain text, and confidential and sensitive messages in plain text can be easily compromised. Information meant for a specific set of eyes must be carefully guarded. Spies use secret codes to communicate with their secret agents. Julius Caesar never trusted his messengers carrying message to his generals. He encrypted his messages by replacing every A with a D, every B with E, and so on, so only the intended recipient could decipher the message.

Information security is the protection of organizational/personal data from unauthorized users. The basic components of Information security are: Confidentiality, Integrity and Authenticity, and Availability. Confidentiality is secrecy. No one else should read the data apart from the one who is sending the data and the authorized receiver. With the increasing use of the Internet as an e-commerce tool, it is important for users, banks, and commercial institutions to make sure that their information is secured and no one is able to read change the data during its transmission.

When computer systems can code plain text and the recipient understands and interprets this coded message, users feel more secure transmitting data over the Internet, or any other media. This method of coding a plain text message into a secret coded message is called cryptography. The method of disguising plain text to hide the actual data is called **encryption**. The new encrypted text is called ciphertext. The encrypted data is not readable by others and hence it is secur. Once it reaches its destination, the receiver can reverse the process to read the ciphertext. This process is called ***decryption***. The typical process of encryption and decryption is illustrated in Figure 8-1.

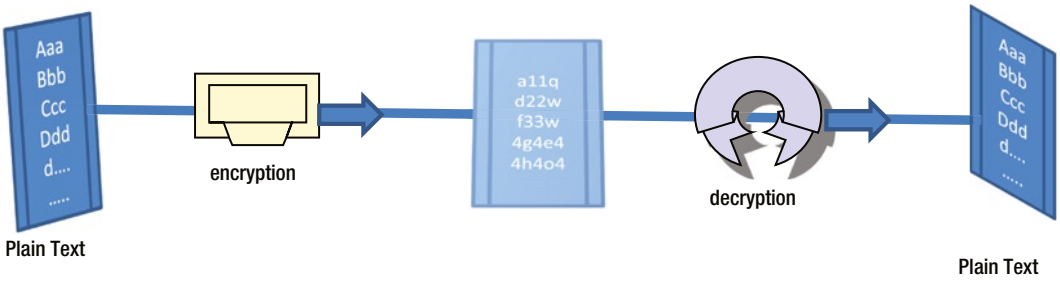

Figure 8-1. *Encryption and Decryption*

Cryptography is the process of converting simple plain text into secret text called ciphertext, and converting ciphertext back to its original simple text, as shown in the Figure 8-1. The process uses algorithms known as crypto-algorithms to perform the encryption and decryption process.

Encryption and decryption are done using a "key" or "code." Sometimes, only one key is used to perform both encryption and decryption; sometimes two separate keys are used, one for encryption and the other key for decryption.

In today's Internet world, cryptography applications are used to enable digital signatures, money transfers, online shopping, online booking, and credit card payments, where authentication and privacy are crucial. Cryptography makes transactions on the web more secure through digital certificates, 3-D secure, and other encryption technology.

With the rise in government surveillance of Internet data, which is making headlines every day, people are even more concerned about their privacy and personal data. E-mails sent in plain text can become a serious problem, as someone could tap the network and read your personal e-mail–something that has become quite common. Hence, companies prefer to use encrypted e-mail for employees–certainly, for senior executives at the very least. Though encrypting e-mail may or may not protect an individual or business completely from government surveillance, it can certainly keep your data safe from intruders who are looking to find useful information from your e-mail.

Cryptography is mainly used to protect confidentiality of the data. However, it is not restricted to the confidentiality. It is used for checking integrity and authentication processes as well. For example, in many governance processes, a signature is an essential part of the process for authentication and maintaining integrity. If we make this process computerized, where approval, and other governance is done via network or Internet, then we need a mechanism to authenticate the user's signature digitally (digital signatures), and provide a digital timestamp. Cryptography provides such a mechanism.

Cryptography is also used to regulate access to your cable or satellite television. It is controlled centrally and only the channels you are subscribed to can be accessed and all other signals are "scrambled" using cryptographic technology. For example, pay-per-view, annual, or monthly subscriptions are all controlled centrally by scrambling and unscrambling signals based on the payment. Setup boxes installed at houses, hotels, and other places will have the ability to decode the channels only upon receipt of payments.

Although cryptography is widely used, its application on the Internet is increasingly demanding and growing as hackers are cracking cryptographic algorithms. Researchers are working on providing better algorithms and keys so that users data and authentication is protected. Cryptography is still fundamentally based on problems that are difficult to solve because of the complexity of the keys for decrypting and encrypting messages or signing documents digitally.

Cryptography, cryptanalysis, and cryptology are interrelated. In general cryptography refers to the technique of encrypting and decrypting plain text. Cryptanalysis refers to analyzing and breaking the keys used for encryption and decryption (generally used by hackers). Cryptology refers to both: study of cryptography and cryptanalysis.

In this chapter, we will focus on the basics of cryptography and its application. We will not be covering in-depth analysis of cryptography itself. There are several textbooks and papers that exclusively discuss different cryptographic algorithms and techniques. Bruce Schneier is known as an authority on cryptography. He has contributed to the community with more than 10 useful books related to the concept of cryptography and has several blogs on the topic.

Cryptographic Algorithms

In cryptography, encryption and decryption are performed using a mathematical function, often known as cryptographic algorithm. The mathematical function consists of keys: a word, number, or phrase. The cryptographic algorithm makes use of one or more of these keys to encrypt the data. The same plaintext can be encrypted using different keys to get different ciphertext. The strength of the encryption depends on the keys and cryptographic algorithm which makes use of these keys to encrypt.

There are three types of cryptoalgorithms (based on key), which are discussed in detail in this chapter:

- **Symmetric Key (Secret Key Cryptography)**: Uses a single key to encrypt and decrypt the messages
- **Asymmetric Key (Public Key Cryptography)**: Uses one key to encrypt and another key to decrypt the messages
- **Hash Functions**: Uses a mathematical transformation that transforms the message into a fixed length data that is unique to the corresponding source. These transformations are carried out using hashing functions/algorithms and are not normally reversible or are one way hashes.

Figure 8-2 illustrates the above three types of cryptography.

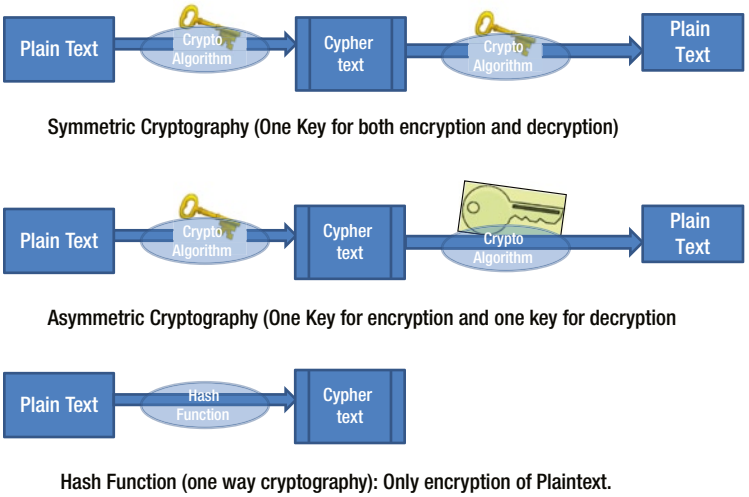

Figure 8-2. *Three types of Cryptography*

For any encryption approach, there are two major challenges: key distribution and key management. Key distribution is a mechanism to convey the keys to those who need them to establish secure communication. Key management is to manage large number of keys and provide the right key to the right user as needed.

Symmetric Key Cryptography

In this technique a single key is used to encrypt and decrypt the data. Both, the sender and receiver must share the same key in order to share confidential information. Because a single key is used for both encryption and decryption, this method is called symmetric cryptography. In this method, a single key, which is secret, must be known to both the sender and receiver.

Symmetric key cryptography operates in two modes, stream or block. In stream mode, each bit is considered for encryption whereas in block mode, blocks of data are considered for encryption. In case of block mode, one block of data is encrypted using the same key but in case of stream mode, the same block will have multiple key to encrypt the data. Since the messages are normally more than one block, block mode method needs a mechanism to arrange different blocks together.

Figure 8-3 illustrates how the symmetric key cryptography is used to ensure confidentiality of the message that is sent.

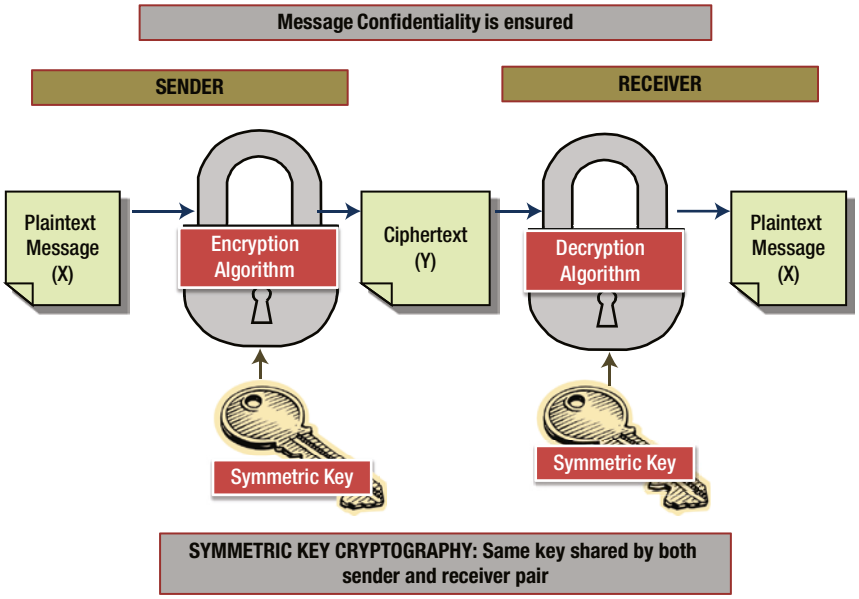

Figure 8-3. *Symmetric Key Cryptography*

There are several algorithms developed for both the modes. The Data Encryption Standard (DES) and the Advanced Encryption Standard (AES) are two block cipher algorithms recognized by US government. DES was developed by IBM as a standard for applications in 1977 and since then it has been used in many applications. DES was the most popular algorithm used across a wide range of applications from ATM encryption to e-mail privacy. [1] However, the known and exploitable weaknesses of DES have caused the community to discourage its use.

In stream mode, encryption is performed one byte at a time. Instead of blocks of data, each byte is encrypted using a stream of keys. RC4 is a variable-key-size stream cipher developed in 1987 by Rivest. RC4 is a stream cipher licensed by RSA which is a widely used stream cypher method.

Some of the most popular cryptoalgorithms are:

- Data Encryption Standard (DES)
- Advanced Encryption Standard (AES)
- Rivest Cipher (RC)
- International Data Encryption Algorithm (IDEA)
- Blowfish

DES is one of the first widely used algorithms but it has been cracked and no longer considered secured. AES is used by the US government and IDEA is used by European nations. Blowfish is an open-source symmetric algorithm created in 1993.

Key Distribution

Distribution of the key and managing the key between different set of users is the most challenging task. Symmetric key cryptography is more useful for encryption of files or file systems in the local machines and less useful for communication between the two systems in the network because of “key distribution” challenges.

There are two ways of solving key distribution problems. One approach is to physically exchange the keys in advance. The secret keys are personally handed over to the parties, which is manual. The second approach is to use a "Trusted Key Distribution Center" to distribute the keys, a trusted network entity with whom one has shared the secret key. This process can be automated.

Suppose Anna and Barry want to communicate using a symmetric key. But, they have never met before and thus they do not have the shared keys to exchange information. Now, there are two problems, one is sharing the key but more important is sharing the key with a person who is a stranger but still wants to communicate. A solution that is often adopted is to use a trusted party known as Key Distribution Center (KDC).

The KDC is a server that manages different symmetric keys with each of the registered user. Each user who wants to communicate with the other user must register with KDC. KDC will check the credentials of each user to ascertain the authenticity. A user who wishes to communicate with the other user, let's say, Anna wants to communicate with Barry, Anna and Barry both have to first register with KDC. Anna takes the first step to send a request for a key as well as the user it wants to communicate. Once the request is processed with proper authentication, KDC sends shared key to both Anna and Barry. Henceforth, both can communicate with each other with the secret key that was given to them by KDC. KDC can also set expiration and other parameters of the key.

Figure 8-4 illustrates the entire process of symmetric key distribution through the KDC.

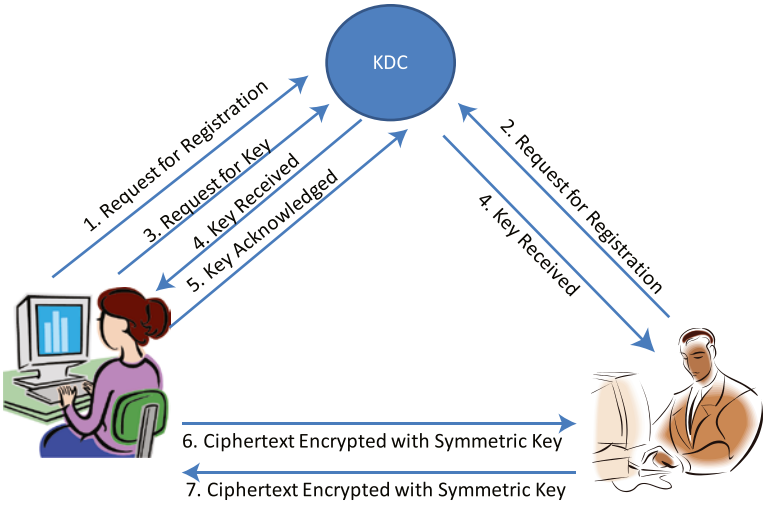

Figure 8-4. *Symmetric Key Distribution Center*

Asymmetric Key Cryptography

There are two problems with symmetric key cryptography:

- Distribution of key – Secret key sharing among senders and receivers. If there are n parties involved in the communication then n(n-1) keys to be distributed. Managing this many keys is another problem.
- Authenticity – Trust and Authenticity of two parties.

In 1976, Diffie and Hellman at Stanford University came up with a new method to solve both the problems of symmetric cryptography that changed the world of cryptography and digital communication radically. This new method is called **Public Key Cryptography** also known as **Asymmetric Key Cryptography**.

Public Key Cryptography

Public key cryptography uses a pair of keys for encryption and decryption. A ***public key*** is used to encrypt the data and a ***private key*** is used to decrypt the data. Using the public key, anyone can encrypt the data, but they cannot decrypt the data. In this approach, both sender and receiver have the ability to generate both keys (using a computer system) together. However, only the public key is made known to the other party, who can download this key even from a web server; the private key is not known to anyone. It is not sent to the other party, hence the problem of distribution of the key never arises. In case of intrusion or any other problems, the system can generate a private key, and a corresponding public key that can be published again. The algorithms that generate keys are related to each other mathematically in such a way that knowledge of one key does not permit anyone to determine the other key easily.

Figure 8-5 illustrates how the confidentiality of a message is ensured through asymmetric key cryptography (alternatively known as public key cryptography).

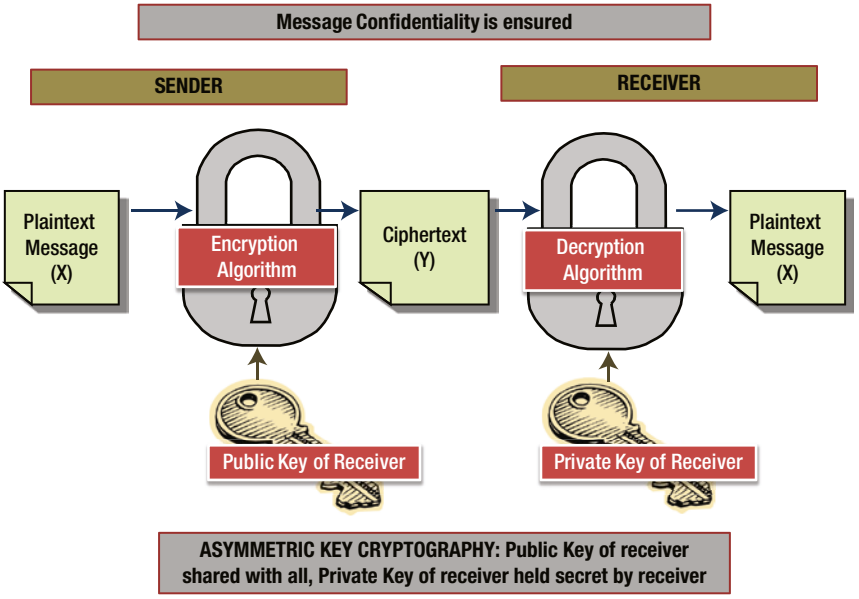

Figure 8-5. *Public Key Cryptography – How Confidentiality is ensured*

Figure 8-6 illustrates how the authenticity of the message is ensured through asymmetric key cryptography (i.e., public key cryptography).

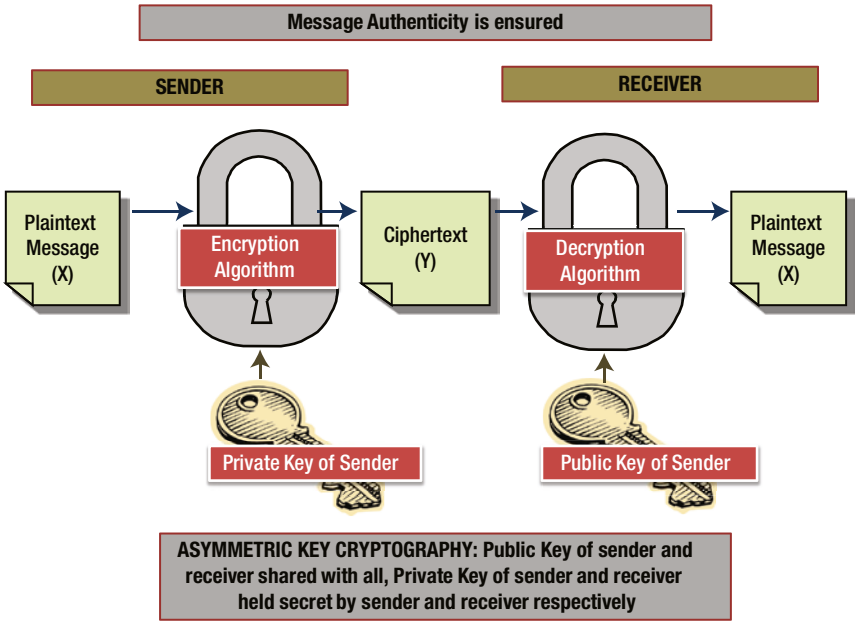

Figure 8-6. *Public Key Cryptography – How Authenticity is ensured*

Figure 8-7 illustrates how both the message confidentiality and authenticity are ensured through asymmetric key cryptography (i.e., public key cryptography).

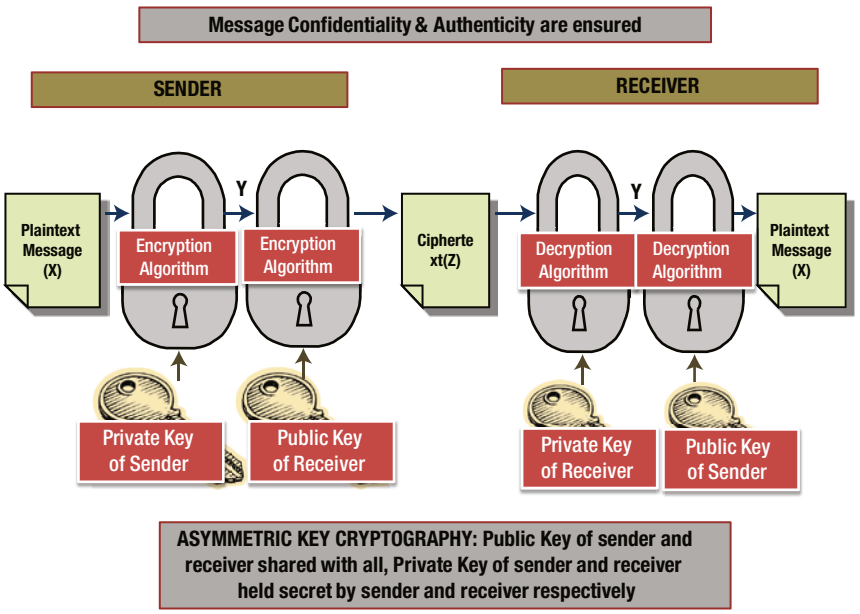

Figure 8-7. *Public Key Cryptography – How both Confidentiality and Authenticity are ensured*

The Public Key Cryptography (PKC) concept was invented by Whitefield Diffie and Martin Hellman in 1976 paper. The primary benefit of the PKC is that only the public key is shared, the need to share private key via some secure channel is eliminated, and private keys are not transmitted or shared. A public key system is constructed using a mathematically infeasible solution where one key cannot be generated using the other key and both the keys are required for a secured communication. The historian David Kahn[2] described public key cryptography as "the most revolutionary new concept in the field since polyalphabetic substitution emerged in the Renaissance."

There are many algorithms based on PKC, but the most popular ones are:

- Diffie Hellman
- RSA (Rivest, Shamir, Adleman)
- Digital Signature Algorithm (David Kravitz)

RSA Algorithm

RSA is an encryption and authentication algorithm developed by Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman. It is used in many applications including browsers. The algorithm is owned and licensed by RSA Security which is part of EMC[2]. It uses one key for encryption and another key for decryption. The mathematical function for generating keys itself can be found in specifications and standards as well as in the RSA web page. Using the mathematical functions, two sets of numbers (keys) are generated: public key and private key. Both the public key and private keys are required for encryption and decryption but private keys are kept private by the owner and are never sent across the Internet. The public key is used to encrypt the data and private key is used to decrypt when the message confidentiality has to be maintained.

Authentication can be provided by using the PKC system and RSA algorithm (RFC 3447). The message is encrypted using the private key of the sender to authenticate the sender. As the sender's private key is only held by the sender, encryption by using the private key by the sender authenticates that the message was in fact originated by the sender himself. During the authentication process, a private key is used by the sender and the receiver decrypts using the public key. This does not guarantee confidentiality, but does assure the authenticity of the message. When the original message is transformed using the message digest function and encrypted by the private key, it is known as a digital signature. It is also possible to encrypt a portion of the message being sent using the private key of the sender to demonstrate the authenticity rather than encrypting the entire message. Such a system should have the capability that the unencrypted portion is not possible to be modified without the modification of the encrypted portion.

Table 8-1 summarizes the entire authentication process.

Table 8-1. *Steps in the Authentication Process*

Process	Key	Owner
Send encrypted message digest (i.e., digital signature)	Private Key of sender	Sender
Decrypt signature	Public Key of sender	Receiver
Send encrypted message	Public Key of receiver	Sender
Decrypt message	Private Key of receiver	Receiver

The RSA keys are derived from a variable size encryption block and a variable size key. The key-pair (public and private) is derived from a very large prime number, chosen according to special rules. The strength of RSA depends on the key length but choosing a long key can slow down the system. For bulk data encryption, it is recommended to use DES for better performance.

If you want both the confidentiality and the authenticity of the message, the following encryption mechanism has to be used:

- The message has to be first encrypted using the private key of the sender.
- The encrypted message is then encrypted using the public key of the receiver.
- The encrypted message is sent to the receiver.
- The receiver on receiving the encrypted message decrypts it using his (receiver's) private key.
- The semi-decrypted message is then decrypted using the public key of the sender.
- The receiver obtains the plain text message.

Advantages of Public Key Cryptography

The advantages of public key cryptography are:

- No need to exchange the keys
- Another key cannot be derived from one key
- The confidentiality of the message can be ensured by using the public key cryptography
- It is possible to establish authentication of the sender by using public key cryptography (digital signature)
- It is possible to ensure the confidentiality and authentication of the message at the same time
- It is possible to use public key cryptography for session key exchange

Applications of PKC

Public Key Cryptography is used in a number of applications and systems software. Some examples of application of cryptography are:

- Digitally signed document
- E-mail encryption software such as PGP and MIME
- RFC 3161 authenticated timestamps
- Digital signatures in the Operating System software such as Ubuntu, Red Hat Linux packages distribution
- SSL protocol
- SSH protocol

Public Key Infrastructure (PKI)

A Public Key Infrastructure (PKI) enables users to securely transact through the use of public key cryptography. Key pairs are obtained from a third-party trusted authority called Certificate Authority (CA). The PKI provides an infrastructure to issue a "digital certificate" that identifies an individual or organization. Based on the identity of the digital certificate, transactions are made securely over public networks such as the Internet. The PKI is based on the use of public key cryptography, which is commonly used.

A public key infrastructure consists of:

- A Certificate Authority (CA) that issues and verifies digital certificates. A certificate includes the public key or information about public key
- A registration Authority (RA) which verifies the user's authenticity for CA before CA issues a digital certificate
- A secured storage place to hold the certificates and public keys
- A certificate management system
- Hardware, software, policies, procedures, and people used to create, manage, and revoke digital certificates along with the distribution and storage of the digital certificates

A certificate contains information referring to a public key, issued by a Certification Authority (CA). The information in the certificate should conform to the ITU (IETF) standard X.509 v3. Certificates conforming to that standard include information about the published identity of the owner of the corresponding public key, the key length, the algorithm used, associated hashing algorithm, dates of validity of the certificate, and the actions the key can be used for.

Certificate Authority (CA)

A CA is responsible for issuing certificates. CA issues the digital certificate based on the recommendation of RA. This digital certificate is signed by the CA using its own private key. The CA issues the certificate which contains the public key of the party who owns the certificate. Certificates have to be purchased from the CA. CA can issue a certificate only after it confirms all the credentials to prove your identity. Once identity is proved, it stamps the certificate to prevent modifications of the details contained in the certificate. CA is analogous to a passport agency. An individual or organization may have any number of certificates issued by different CAs. Different web applications may insist to use a particular certificate. For example, a particular bank may insist to use a certificate issued by that bank for a secured transaction, whereas some other web site may accept any certificate issued by any CA.

Registration Authority (RA) is a third-party verification agency for a Certificate Authority (CA), to perform the verification of the organization or individuals who have applied for the certificate. Final component of the PKI is the Certificate Management System (CMS) through which certificates are published, renewed, or revoked. Examples of Certificate Authority (CA) include Verisign, Thawte, SSL.com, RapidSSL, Network Solutions, GlobalSign, Digicert, Enutrust.net, PinkRoccade, and PKI.CAcert. [24]

Digital Certificate

Digital Certificate provides an electronic identity to conduct secure transactions by providing your identity (authentication). It is similar to a passport or driver's license. With a digital certificate, an organization or an individual can provide authentication for all the transactions with friends, business partners, and other online services. Digital certificate assures identity among all the parties involved in the transactions. The most widely used format of a digital certificate is as defined by the CCITT X.509 standards. [25] Digital certificate uses public key cryptography to verify the integrity of the certificate itself.

Hash Function Cryptography

Hash functions, also called message digests, use a fixed length hash value to transform the data that makes it difficult for someone to decrypt or change the data without affecting the hash value, thus securing the data from intruders. Hashing functions are one-way mathematical functions that are easy to compute but hard to reverse. A hash function

H( ), applied on input (x), and returns a fixed string, h_s. Mathematically it is written as $h_s = H(x)$. A cryptographic hash function in general should have the following properties:

- Flexible input length (x)
- H(x) should be relatively easy to compute
- H(x) is one way function and cannot be reversible
- The output is of fixed length and does not depend on input length

Hashing is generally used in the following situations:

- Password management in case of PPP, CHAP, and Microsoft EAP. This method of cryptography is normally used in operating systems to protect passwords.
- Digital signatures and file integrity checkers to check the integrity of data.

Hashing functions are used to vouch for the integrity of the message by appending the message with the hash value. If the message is changed, the hash value when recomputed will not match the precomputed hash value. In order to avoid man-in-the middle attacks, it is ideal to send the hash value in a secure way to the intended party. Such secure transfer is possible using public key cryptography.

Further, hash value is used to store passwords of the operating systems like Microsoft Windows. Here, the original passwords are not stored; instead the SAM corresponding hash values are stored. These provide high security to the passwords, as hash value is not reversible to find out the original password. Only when the passwords are entered in the system will it compute the hash value and check with the hash value stored in the SAM.

"Salting" the password before hashing by either suffixing or prefixing it with a random string decreases the possibility of cracking the password.

Hashing is also used in some of the implementation of digital signatures which vouches for the integrity of the message sent. Hashing functions are also used in virus detection as well as intrusion detection.

Figure 10-8 illustrates how hashing ensures the integrity of the message that is sent.

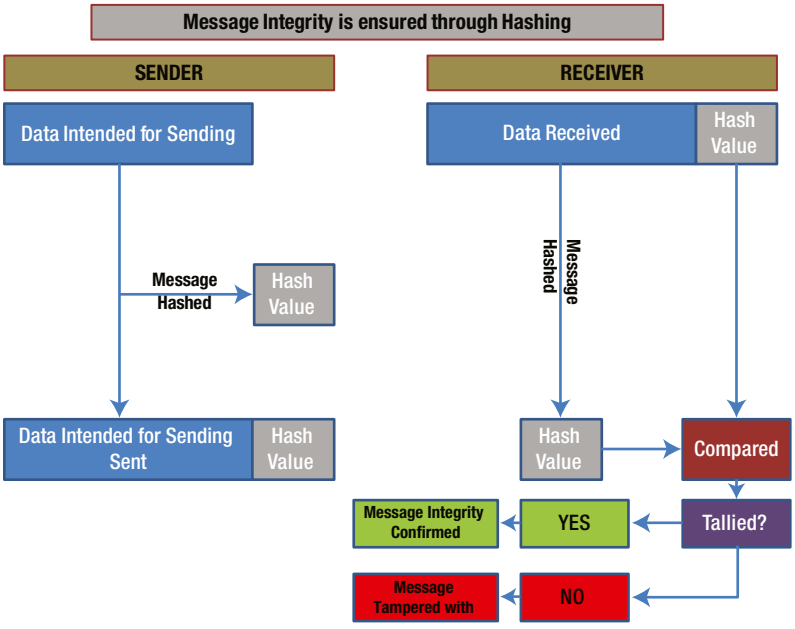

Figure 8-8. *Message Integrity Check through Hashing*

Popular Hashes

MD5 (Message Digest Function 5), SHA1 (Secure Hash Algorithm 1), SHA2 (Secure Hash Algorithm 2), and SHA3 (Secure Hash Algorithm 3) are the popular hashing functions /algorithms. MD5 outputs are of 128 bits and are popularly used for storing of the passwords as well as to ensure file integrity. MD5 is prone for collision.

SHA algorithms again provide for one way hash. SHA1 provides for 160 bit output. SHA-224, SHA-256, SHA-384, and SHA-512 are known as SHA-2. SHA3 is the most advanced hashing function which was announced by NIST in 2012. SHA-3 has a unique structure known as sponge construction.

MAC (Message Authentication Code) is another popular hash function which is also known as a Keyed Hash Function.

Digital Signatures

A digital signature is like a handwritten signature but it is in the digital form for an electronic document. The document containing the digital signature is verified by the recipient using a hash function to check whether the message has been altered either intentionally or accidentally during the transmission. If the message is altered, the hash function returns a different result. Digital signature ensures authenticity and non-repudiation.

Here, usually the hash value is encrypted with the sender's private key. This provides for the authenticity. When the receiver decrypts the private key using the sender's public key, he gets the hash value. He can check this hash value with the hash value generated using the hash algorithm from the message received. Alternatively, both the message and the appended hash value both can be encrypted with the sender's private key in a similar way as above. If both the hash value received and the hash value generated from the message received tally that means the integrity of the message is maintained. Because it has been signed by the sender's private key, the message sender is also authenticated. Another alternative is to encrypt the message and the hash value using the symmetric key shared between both the parties.

Summary of Cryptography Standard Algorithms

Table 8-2 summarizes some of the **Symmetric cryptographic** algorithms that are used today.

Table 8-2. *Summary of Symmetric Cryptographic Algorithms*

Symmetric Key	Description
Data Encryption Standard (DES)	Developed by IBM in 1970 adopted by National Institute of Standards and technology (NIST)
Triple-DES	A variant of DES that employs up to three 56-bit keys and is recommended replacement of DES.
DESX	Devised by Ron Rivest with a 120-bit key length
Advanced Encryption Standard (AES)	Officially replaced DES in 2001. Uses a key length of 128, 192, or 256 bits and blocks of length 128, 192, or 256 bits. Federal Information Processing Standard (FIPS) PUB 197[3] describes a 128-bit block cipher employing 128, 196, or 256 bit key.
CAST-128/256	DES-like substitution permutation algorithm employing 128 bit key length of 64-bit block. It is defined in RFC 2144[4] and RFC2162[5]. CAST is named after its inventors, Carlisle Adams and Stafford Tavares.

(*continued*)

Table 8-2. (*continued*)

Symmetric Key	Description
Rivest Ciphers (Ron's Code) (named after Ron Rivest)	RC1 – not implemented RC2[6] – 64-bit cipher RC4 – variable length key RC5[7] – A block cipher supporting 32, 64,or 128 bit key length RC6 – 128 bit key improvement over RC5
Blowfish[8]	A symmetric 64-bit block cipher invented by Bruce Schneier. It is a substitute for DES and is in use by large number of commercial products.
Twofish[9]	Designed by Bruce Schneier and team. A 128 bit block cipher with 128,192, or 256 key length. Used in hardware encryption.
Camellia[10]	Developed in 2000 by Nippon Telegraph and Telephone (NTT) Corp and Mitsubishi Electric Corporation (MEC). Suitable for both hardware and software implementation. Is a 128-bit block size, supports 128,192, or 256 key length. RFC 4312 describes the application of Camellia in IPSec. RFC 5581 describes the application in OpenPGP.
MISTY1[11]	A block cipher using a 128-bit key length and 64-bit blocks. It is used in both hardware and software applications. Described in RFC 2994
SEED[12]	128-bit key length and 128-bit blocks. Developed by Korean Information Security Agency (KISA) and adopted as a national standard encryption algorithm in South Korea. Described in RFC 4269
ARIA[13]	A 128-bit block cipher employing 128,192, or 256-bit key length. Described in RFC 5794
CLEFIA[14]	128-bit block cipher with a key length of 192, 256 bits developed in 2007 by SONY corporation. Is one of the latest algorithms to support high performance software and hardware applications. Described in RFC 6114
KCipher-2[15]	K-Cipher-2 has been used for industrial applications especially for mobile health monitoring and diagnostic services in Japan. Described in RFC 7008
GSM (Global System for Mobile) Encryption	All mobile communications are over the air and vulnerable to security threats as it is open to eavesdroppers with an appropriate receivers. Several security functions are built into the GSM to safeguard subscribers privacy[16]: Authentication of the registered subscribers Secure data transfer Subscriber identity protection For authentication process, A3 authentication algorithms are used. For encryption and decryption of data A8 algorithms are used.
GPRS Encryption	The A5/4, A5/3, and GEA4, GEA3 algorithms are based on the 3GPP ciphering algorithm (F8). Mitsubishi Electric Corporation holds essential patents on the Algorithms[17] ETSI is Custodian of the 3GPP™ confidentiality and integrity algorithms UEA2 & UIA2, UEA1 & UIA1, and EEA3 & EIA3 which have been developed through the collaborative efforts of the European Telecommunications Standards Institute (ETSI), the Association of Radio Industries and Businesses (ARIB), the Telecommunications Technology Association (TTA) and ATIS

Table 8-3 summarizes the Public-Key Cryptography Algorithms that are commonly used today.

Table 8-3. *Summary of Public-Key Cryptography Algorithms*

Public Key Cryptography Algorithms	Description
RSA[18]	RSA is an encryption and authentication algorithm developed by Ron **R**ivest, Adi **S**hamir, and Leonard **A**dleman. It is used in many applications including browsers. The algorithm is owned and licensed by RSA Security which is part of EMC[2]
	NIST currently supports three different RSA algorithm implementations. ANSI X9.31-1998 and is called RSA. The other algorithms are specified in the PKCS #1 v2.1: RSA Cryptography Standard dated June 2002. They are defined as signature schemes with appendix and are called RSASSA-PSS and RSASSA-PKCS1-v1_5. FIPS 186-4 imposes additional constraints on these RSA algorithm implementations[19]
Diffie-Hellman	Diffie and Hellman came up with their own algorithm but does not support authentication. Details are described in RFC 2631
Digital Signature Algorithm (DSA)	Specified by NIST's Digital Signature Standard for digital signature authentication process
Elliptical Curve Cryptography (ECC)[20]	A PKC algorithm based on elliptic curves with small keys.
Public Key Cryptography Standards (PKCS)	A set of interoperable standards and guidelines developed by RSA Security (now EMC[2]):
	RFC 3447: RSA Cryptography Standard RFC 2898: Password based Cryptography Standards RFC 2986: Certification Request Syntax Standard version 1.7 RFC 2315: Cryptographic Message Syntax Version 1.5 RFC 2985: Selected Object Classes and Attribute Types version 2.0 RFC 5208: Public-Key Cryptography Standards (PKCS) #8: Private Key Information Syntax Specification Version 2

Table 8-4 summarizes some of the Hash function algorithms that are commonly used.

Table 8-4. *Summary of Hash Function Algorithms*

Hash function Algorithm	Description
MD2	Produces an output of 128-bit "message digest".[21] It is conjured that it is computationally infeasible to produce two similar message digest. An algorithm intended for digital signature application where a file must be compressed. Designed for systems with limited memory, such as smart cards. Described in RFC 1319[21] and RFC 6149
MD4	Designed specifically for fast processing software applications. Described in RFC 1320 and RFC 6150
MD5	Improvement of MD4 algorithm. Described in RFC 1321
Secure Hash Algorithm (SHA) 1	NIST's Secure Hash Standard algorithm. Produces 160 bit hash value. Published in NIS's FIPS PUB 180-1 and RFC 3174
Secure Hash Algorithm (SHA) 2	The United States has adopted a suite of Secure Hash Algorithms (SHAs), including four beyond SHA-1, as part of a Federal Information Processing Standard (FIPS), specifically SHA-224 (RFC 3874), SHA-256, SHA-384, and SHA-512.[22]. This can produce hash values that are 224, 256, 384, or 512 bits in length
SHA-3	SHA-3 is a new algorithm as an alternative to SHA-2. In 2007, **SHA-3 competition**[23] was a launched and it received 64 submissions. NIST announced Keccak as the wined of the SHA-3 cryptography hash algorithm competition and the new SHA-3 algorithm is in press release. Keccak was designed by a team of cryptographers from Belgium and Italy. They are: Guido Bertoni Joan Daemon Michael Peeters Gilles Van Assche
Others	**RIPEMD** – optimized for 32-bit processors to replace 128-bit hash function **HAVAL** (HAsh of VAriable Length – can create hash values of 128, 160, 192, 224, or 256 bits length **Tiger:** Replacement for SHA and MD5. Run efficiently on 64-bit processor.

Each of the algorithms in Table 8-4 is used in different applications and for different purposes. For example, Hash function algorithms are well suited for data integrity. Any change made to the content during the transmission will result in a hash value different from the original value sent by the sender. Since it is highly unlikely that the same hash value is generated for two different messages, data integrity is ensured.

Symmetric key cryptography is suited for encrypting and decrypting messages, thus providing privacy and confidentiality. The sender can generate a key for each data session to encrypt the message and the receiver can decrypt the message but needs to have the same key for the same session. Symmetric key cryptography also may be used for file encryptions.

Public Key cryptography technique uses a pair of keys called private and public. This is used for not only confidentiality of message but also for non-repudiation and user authentication.

Table 8-4 provided an overview of different algorithms used for different types of cryptography techniques. Table 8-5 provides an overview of some of the common cryptographic algorithms that are used in various applications, particularly in e-commerce applications.

Table 8-5. *Overview of common cryptographic algorithms used in various applications*

Algorithm	Description
Capstone[26]	CAPSTONE is an NSA developed, hardware oriented, cryptographic Device. It implements cryptographic algorithm that is implemented in CLIPPER chip. In addition, the CAPSTONE chip includes the following functions: 1. The Digital Signature Algorithm (DSA) proposed by NIST as a Federal Information Processing Standard (FIPS); 2. The Secure Hashing Algorithm (SHA) recently approved as FIPS 180; 3. A Key Exchange Algorithm based on a public key exchange; 4. A general purpose, random number generator which uses a pure noise source.
Clipper[27]	CLIPPER is an NSA developed, hardware oriented, cryptographic device that implements a symmetric encryption/decryption algorithm. The cryptographic algorithm (SKIPJACK) is completely specified (and classified SECRET). The cryptographic algorithm (called CA) has the following characteristics: 1. Symmetric, 80-bit key encryption/decryption algorithm; 2. Similar in function to DES (i.e., basically a 64-bit code book transformation that can be used in the same four modes of operation as specified for DES in FIPS 81); 3. 32 rounds of processing per single encrypt/decrypt operation;
Federal Information Processing Standards (FIPS) [28]	Federal Information Processing Standards Publications (FIPS PUBS) are issued by NIST after approval by the Secretary of Commerce pursuant to the Federal Information Security Management Act (FISMA) of 2002. The computer security and crypto-algorithms used by U.S Government
GOST	GOST is a family of algorithms used by Russian Federal Standards used by Russian Government. RFC 4357: Additional Cryptographic Algorithms for Use with GOST 28147-89, GOST R 34.10-94, GOST R 34.10-2001, and GOST R 34.11-94 Algorithms RFC 5830: GOST 28147-89: Encryption, Decryption, and Message Authentication Code (MAC) Algorithms RFC 6986: GOST R 34.11-2012: Hash Function Algorithm RFC 7091: GOST R 34.10-2012: Digital Signature Algorithm (Updates RFC 5832: GOST R 34.10-2001)
Identity-Based Cryptography Standard (IBCS) (described in RFC 5091)	IBE is a public-key technology, but one which varies from other public-key technologies in a slight, yet significant way. In particular, IBE keys are calculated instead of being generated randomly, which leads to a different architecture for a system using IBE than for a system using other public-key technologies.

(*continued*)

Table 8-5. (*continued*)

Algorithm	Description
IP Security Protocol (IP sec)	The IPSec is a set of protocol suites which provide confidentiality and authentication services at the IP layer. RFC 2411 describes the overview of IPSec protocol. IPSec protocol suites include: RFC 4301: IP security architecture. RFC 4302: IP Authentication Header (AH), RFC 4303: IP Encapsulating Security Payload (ESP) RFC 4304: Extended Sequence Number (ESN) Addendum, RFC 4305: Cryptographic algorithm implementation requirements for ESP and AH. RFC 4307: Cryptographic algorithms used with IKEv2. RFC 4308: Crypto suites for IPsec, IKE, and IKEv2. RFC 4309: The use of AES in CBC-MAC mode with IPsec ESP. RFC 4312: The use of the Camellia cipher algorithm in IPsec. RFC 4359: The Use of RSA/SHA-1 Signatures within Encapsulating Security Payload (ESP) and Authentication Header (AH). RFC 4434: Describes AES-XCBC-PRF-128, a pseudo-random function derived from the AES for use with IKE. RFC 5996: The Internet Key Exchange (IKE) protocol, version 2 RFC 2403: Describes use of the HMAC with MD5 algorithm RFC 2405: Describes use of DES-CBC RFC 2407 (application of ISAKMP to IPsec), RFC 2408 (ISAKMP, a framework for key management and security associations), and RFC 2409 RFC 2412: Describes OAKLEY, a key determination and distribution protocol. RFC 2451: Describes use of Cipher Block Chaining (CBC) mode cipher algorithms with ESP. RFCs 2522 and 2523: Description of Photuris, a session-key management protocol for IPsec.
Internet Security Association and Key Management Protocol (ISAKMP) (Described in RFC 2408)	The Internet Security Association and Key Management Protocol (ISAKMP) defines the procedures for authenticating a communicating peer, creation and management of Security Associations, key generation techniques, and threat mitigation (e.g., denial of service and replay attacks).
Message Digest Cipher (MDC)	Invented by Peter Gutman, new Zealand. More details can be found in https://www.cs.auckland.ac.nz/~pgut001/
HMAC: Keyed-Hashing for Message Authentication (RFC 2104)	HMAC is a mechanism used for message authentication using cryptographic hash functions such as MD5, SHA-1, etc.
The Keyed-Hash Message Authentication Code (HMAC) Described in FIPS-198[29]	HMAC is used with any iterative approved cryptographic hash function, in combination with a shared secret key. The cryptographic strength of HMAC depends on the properties of the underlying hash function.

(*continued*)

Table 8-5. **(*continued*)**

Algorithm	Description
NSA - Advanced Encryption Standard (AES)	Advanced Encryption Standard (AES) with key sizes of 128 and 256 bits, per FIPS PUB 197 for encryption
Elliptic Curve Digital Signature Algorithm (ECDSA) Described in FIPS 186-3	Using the curves with 256 and 384-bit prime moduli
Secure Hash Algorithm (SHA)	Using 256 and 384 bits. Described in FIPS180-3
Cryptographic Suites for Secure Shell (SSH).	Described in RFC 6239. Secure Shell Transport Layer Protocol
Pretty Good Privacy (PGP)	Philip Zimmermann developed this algorithm for email and file storage applications. It uses RSA for key management and digital signatures, IDEA for message encryption, and MD5 for computing hash value. More information can be found in RFC 1991.
Secure Hypertext Transfer Protocol (S-HTTP)	An extension to HTTP to provide secure exchange of documents over the World Wide Web. Supported algorithms include RSA and Kerberos for key exchange, DES, IDEA, RC2, and Triple-DES for encryption.
Secure Sockets Layer (SSL) Described in RFC 6101	SSL is a security protocol that provides communications privacy over the Internet. This is mainly designed for secure HTTP and FTP connections. This protocol allows applications to communicate securely to prevent any attack on confidentiality and data integrity. SSL also uses MD5 for message digests and X.509 public-key certificates. For more details, refer to RFC 6101.
Transport Layer Security (TLS)	TLS uses 3DES, SHA, DSS and Diffie-Hellman. TLS also provides data privacy and data integrity. TLS was developed to replace SSH. For more information, please refer to RFC 5246.
TrueScript[30]	Open source, multi-platform cryptography software that can be used to encrypt a file, partition, or entire disk.
X.509	ITU-T recommendation for the PKI infrastructure is mainly used in the Telecommunication industry.

Disk / Drive Encryption

With the increased use of the mobile devices like laptops and the storage of confidential data on their hard disk and data carried on USB and other drives, we need to protect the data from theft and misuse. Hence, the disk encryption utilities have emerged. The entire hard disk, USB drive, and other drives can be encrypted so that the data on them cannot be read and misused by unauthorized persons. Now, we have the possibility to burn the data on to portable disks but encrypt them so that unauthorized persons will not be able to misuse them.

The utilities and tools such as TrueCrypt and Gilisoft are widely used for disk encryption, and are found to be very effective in disk encryption. There are many other tools available such as DriveCrypt, DiskCryptor, Rohos Disk Encryption, and Symantec Drive Encryption.

Attacks on Cryptography

There are various attacks possible on cryptography. Some of the common attacks are:

- Rubber Hose Attack: Obtaining by force the secret key like password to the file from those who have them
- Ciphertext-only Attack: Here the attacker has the ciphertext and tries to get the encryption key using the ciphertext
- Known-plaintext Attack: Here the attacker has some plaintext. Using this he tries to get the encryption key
- Chosen-plaintext Attack: Here the attacker uses his own plaintext. He then encrypts them and analyses the resulting output, i.e., ciphertext.
- Adaptive Chosen-plaintext Attack: Here the attacker uses various plaintexts. The subsequent plaintext will be used by him based on the result of earlier output.

Brute force and Frequency Analysis methods are popularly used by the attackers to break the encryption key.

Chapter Summary

- We discussed encoding of messages to ensure the secrecy of the message being sent. We also explored encryption, decryption, cryptography, cryptanalysis, and cryptology. We discussed how cryptography helps in maintaining the confidentiality of the message as well as assures the authentication of the message.
- We explored various cryptographic algorithms such as symmetric key cryptography and asymmetric key cryptography. We also explored the problem of key distribution in the case of symmetric key cryptography and how this can be resolved using the third party. We then explored the asymmetric key cryptography and under that looked into the public key cryptography and how it resolves the issue of key distribution. We looked into the concepts of private key and public key. We also elaborated further as to how this helps out in ensuring the confidentiality of the messages being sent as well as the authentication of the messages. We briefly touched upon the RSA algorithm. We then briefly touched upon the applications and advantages of the public key cryptography. We also briefly touched upon Public Key Infrastructure, the role of Certificate Authority, and Digital Certificates.
- We briefly discussed hashing algorithms, some of the popular hashing functions/algorithms, and the uses of hashing functions/algorithms. We also briefly described how Digital Signatures are implemented using hashing algorithms.
- We listed various symmetric, asymmetric, and hashing algorithms along with further references to them.
- We looked into the disk encryption mechanism to protect the drives / disks from misuse and then concluded the section with the possible attacks on cryptography.