Indira Gandhi National Open University
School of Vocational Education and Training

MSEI-022
Network Security

Cryptography Techniques

3

''शिक्षा मानव को बन्धनों से मुक्त करती है और आज के युग में तो यह लोकतंत्र की भावना का आधार भी है। जन्म तथा अन्य कारणों से उत्पन्न जाति एवं वर्गगत विषमताओं को दूर करते हुए मनुष्य को इन सबसे ऊपर उठाती है।''

— इन्दिरा गांधी

"Education is a liberating force, and in our age it is also a democratising force, cutting across the barriers of caste and class, smoothing out inequalities imposed by birth and other circumstances."

– Indira Gandhi

ignou
THE PEOPLE'S
UNIVERSITY

Indira Gandhi National Open University
School of Vocational Education and Training

MSEI-022
Network Security

Block

3

CRYPTOGRAPHY TECHNIQUES

UNIT 1

Introduction to Cryptography 5

UNIT 2

Symmetric Key Cryptography 15

UNIT 3

Asymmetric Key Cryptography 25

UNIT 4

Applications of Cryptography 32

Programme Expert/Design Committee of Post Graduate Diploma in Information Security (PGDIS)

Block Preparation

Unit Writer	Block Editor	Proof Reading
Ms. Urshla Kant Assistant Professor, School of Vocational Education & Training, IGNOU (Unit 1, 2, 3 & 4)	Mr. Anup Girdhar, CEO Sedulity Solutions & Technologies, New Delhi	Ms. Urshla Kant Assistant Professor School of Vocational Education & Training IGNOU

Production

Mr. B. Natrajan Dy. Registrar (Pub.) MPDD, IGNOU, New Delhi	Mr. Jitender Sethi Asstt. Registrar (Pub.) MPDD, IGNOU, New Delhi	Mr. Hemant Parida Proof Reader MPDD, IGNOU, New Delhi

July, 2011

ISBN: 978–81–266–5614–1

Further information about the School of Vocational Education and Training and the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110068. or the website of IGNOU www.ignou.ac.in

Printed and published on behalf of the Indira Gandhi National Open University, New Delhi, by the Registrar, MPDD

Laser typeset by Mctronics Printographics, 27/3 Ward No. 1, Opp. Mother Dairy, Mehrauli, New Delhi-30

Printed at: Berry Art Press A-9, Mayapuri Phase-I New Delhi-64

BLOCK INTRODUCTION

Once more and more information is kept in digital form, the protection of data in computer systems begins to pose challenges to designers, researchers and system managers. One of the most powerful and important methods for security in computer systems is to encrypt sensitive records and messages in transit and in storage. Cryptography is the translation of information (known as plaintext) into a coded form (known as cypertext) using a key. Cryptography is mostly used to protect the privacy of information. In a strong cryptosystem, the original information (plaintext) can only be recovered by the use of the decryption key. So the plaintext information is protected from "prying eyes". A strong encryption algorithm is one who cannot be easily inverted on a Supercomputer today (i.e. the PC in 10 years time). There are two principal methods of cryptography, Shared Key and Public Key cryptography. This block comprises of four units and is designed in the following way;

The **Unit One** introduces cryptography. Cryptography is a branch of applied mathematics which concerns with transforming messages into seemingly unintelligible forms and back again. Cryptography does for electronic information what locks and lockers do for printed information. Information is protected by scrambling it in such a manner that it can be unscrambled only with a secret key. The scrambled message called ciphertext is totally unintelligible to anyone who does not know the key. Anyone getting hold of the cipher text while it is on the unsafe channel would need to have the appropriate key to be able to get to the original information. The authorized receiver is assumed to have that key. The process of producing ciphertext is called encryption and the reverse process of restoring the original message called plaintext is called decryption.

The **Unit two** covers the detailed descriptions of the Symmetric key cryptography. Symmetric key cryptography is often much faster than asymmetric or public-key cryptography so it's preferred for encrypting large amounts of data. But the key length and complexity in current crypto systems don't make it feasible to transfer the shared secret in a telephone call. So public-key technology is often used to encrypt only the shared secret. First the shared secret is decrypted and then symmetric key cryptography is used to efficiently decrypt the large blocks of data.

The **Unit three** explains Asymmetric key cryptography. Asymmetric encryption have two related keys - a key pair. A public key is made freely available to anyone who might want to send you a message. A second, private key is kept secret, so that only you know it. Any message (text, binary files or documents) that are encrypted by using the public key can only be decrypted by applying the same algorithm, but by using the matching private key. Any message that is encrypted by using the private key can only be decrypted by using the matching public key.

Unit four describes applications of cryptography. Encryption is often portrayed as the silver bullet for information security. Unfortunately, it is a far cry from that. It does not protect against the majority of attacks, including insider abuse, social engineering, break-ins that exploit system vulnerabilities such as buffer overflows, data tampering, Trojan horses, viruses, web scams and so on. It has a great role in working of electronic signatures. It is but one element of a defensive information warfare program.

Hope you benefit from this block.

ACKNOWLEDGEMENT

The material we have used is purely for educational purposes. Every effort has been made to trace the copyright holders of material reproduced in this book. Should any infringement have occurred, the publishers and editors apologize and will be pleased to make the necessary corrections in future editions of this book.

UNIT 1 INTRODUCTION TO CRYPTOGRAPHY

Structure

1.0 Introduction

1.1 Objectives

1.2 Cryptography

1.3 History of Cryptography

1.4 Encryption

1.5 Stages of Cryptology

1.6 Let Us Sum Up

1.7 Check Your Progress: The Key

1.8 Suggested Readings

1.0 INTRODUCTION

Once more and more information is kept in digital form, the protection of data in computer systems begins to pose challenges to designers, researchers and system managers. One of the most powerful and important methods for security in computer systems is to encrypt sensitive records and messages in transit and in storage. Cryptography is the process used to encode (encrypt) an electronic information. Cryptography dates back to the period of the ancient Greeks. The sparatans would a belt in a spiral around a stick, wrote a message along the length of the stick and unwound the belt. This created the first even transposition cipher. Only a person who had a stick exactly the right size could read the message. This encryption process encodes information with a view to make it secure from unauthorized access. The reverse of this process is known as decryption.

Cryptography is the translation of information (known as plaintext) into a coded form (known as cypertext) using a key. Cryptography is mostly used to protect the privacy of information. In a strong cryptosystem, the original information (plaintext) can only be recovered by the use of the decryption key. So the plaintext information is protected from "prying eyes".

1.1 OBJECTIVES

After studying this unit, you should be able to:

- define cryptography;
- know history of cryptography;
- define encryption; and
- elaborate stages of cryptology;

1.2 CRYPTOGRAPHY

The word "Cryptography" has been taken from greek words "kryptos" which means "hidden" and "Graphein" which means "to write". It simply means "the design and analysis of codes and ciphers".

Cryptography is the process of transforming plain text or original information into an unintelligible form (cipher text) so that it may be sent over unsafe channels of communication.

Cryptography is a branch of applied mathematics which concerns with transforming messages into seemingly unintelligible forms and back again. Cryptography does for electronic information what locks and lockers do for printed information. Information is protected by scrambling it in such a manner that it can be unscrambled only with a secret key. The transformation process is controlled by a data string (key). The scrambled message called ciphertext is totally unintelligible to anyone who does not know the key. Anyone getting hold of the cipher text while it is on the unsafe channel would need to have the appropriate key to be able to get to the original information. The authorized receiver is assumed to have that key. The process of producing ciphertext is called encryption and the reverse process of restoring the original message called plaintext is called decryption.

The digital signatures are created and verified by means of a technique called cryptography. Such digital signature is a block of data at the end of an electronic message that attests to the authenticity of the said message. The expression "digital signature" has been defined in the sec 2(1)(p) of IT Act as "Authentication of any electronic record by a subscriber by means of an electronic method or procedure in accordance with the provisions of sec 3."

Public Key Cryptography

Both parties have a private key and a public key. The private keys are known only to their owners, but the public keys are available to anyone (like telephone numbers). The sending party encrypts the message with the receivers public key and the receiver decrypts with his own private key. This is possible due to the discovery by Diffie and Hellman (at Stanford University, autumn 1975) that algorithms can be developed which use one key for encryption and a different key for decryption. The public and private key constitute a key pair.

The following public key crypto-systems are well known:

- the RSA (named after it's inventors Rivest, Shamir and Adleman) algorithm was developed at MIT in 1977 and is the most common public key system is use today. A key minimum key length of 768 bits is recommended by RSA Inc. RSAREF is a library from RSA Inc. which is integrated into many commercial products and public domain products (such as the US version of PGP). International, public domain RSA compatible libraries (with large key sizes) do exist and are used in products such as SSH (SSH uses 1024 bit keys by default). RSA key generation is slower than verification. RSA is patented in the U.S. until 20.9.2000.

- The Diffie-Hellman (named also after it's inventors) key exchange protocol, published in 1976, produces shared secret keys from publicly known information over unsecured networks. These shared keys can be used o produce session keys. It's strength is based on the "discrete logarithm" problem. Since parties are not authenticated, it is vulnerable to "man in the middle" attacks, which can be prevented by use additional protocols or digital. Sun make extensive use of this algorithm in Secure RPC and SKIP.

  This algorithm has the added advantage that it's patent expired in 1997.

- The ElGamal Public Key system (invented by Taher ElGamal) consists of both an encryption and signature algorithm. It is similar to the Diffie-Hellman key exchange and it's strength is based on the "discrete logarithm" problem. Key length strengths are similar to RSA. It is quite slow and requires very good random number generation. DSA is based on the signature algorithm.

- DSS is the Digital Signature Standard, that uses the DSA (Digital Signature Algorithm) approved in May 1994 by the U.S. government (NIST & NSA) as the standard for digital authentication. DSA is based on crypto algorithms from ElGamal and Schnorr. Signature generation is faster than verification (which is unusual, there are likely to be more verifications than generations). DSA lacks a key exchange mechanism, is very new and has been criticised because the NSA were heavily involved in it's selection and it was not subject to open peer review.

Advantages of PK

Only the private key need be kept secret. No secret channels need exist for key exchange, since only public keys need be exchanged. However the public key must be transferred to the sender in such a way that he is absolutely sure that it is the correct public key! Public key cryptography also provides a method for digital signatures.

Disadvantages

Slow, due to the mathematical complexity of the algorithms.

Typical Applications

Ensuring proof or origin, ensuring that only the receiver can decrypt the information, transmission of symmetric session keys.

Hashing /Message Digest

A hash function creates a fixed length string from a block of data. If the function is one way, it is also called a message digest function. These (fast) functions analyse a message and produce a fixed length digest which is practically unique i.e. finding a message with an identical hash very unlikely with very fast computers. There is no known feasible way of producing another message with the same digest. Such algorithms are normally used to create a signature for a message which can be used to verify it's integrity.

- MD2, MD4 and MD5 are hash functions developed by Ron Rivest of RSA Inc. They all produce 128-bit digests. MD2 is the slowest, MD4 the fastest. MD5 has a more conservative design than MD4. Both of these are publicly available.

- The MD5 algorithm is the de-facto hashing standard for digests. Public domain versions are available for most platforms on the Internet and it is widely used in integrity checking systems. SHA-1 (Secure hashing algorithm) is a NIST sponsored hashing function has been adopted by the U.S. government as a standard. It produces a 160-bit hash (i.e. larger than MDx) and is roughly 25% slower than MD5. SHA-1 is recommended over MD5 .

- Ripe-MD-160 is an algorithm from the European Community.

Advantages

It is much faster than encryption and output is fixed length (so even a very large file produces the same digest, which is much more efficient for data transmission).

Disadvantages

It guarantees integrity only.

Interesting variations of hashes are Message Authentication Codes (MAC), which are hash functions with a key. To create or verify the MAC, one must have the key. This is useful for verifying that hashes have not been tampered with during transmission. Two examples are HMAC (RFC 2104) and NMAC, based on SHA-1.

1.3 HISTORY OF CRYPTOGRAPHY

Before the modern era, cryptography was concerned solely with message confidentiality (i.e. encryption) conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge (namely the key needed for decryption of that message). Encryption was used to (attempt to) ensure secrecy in communications, such as those of spies, military leaders and diplomats. In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, interactive proofs and secure computation, among others.

Classic Cryptography

The earliest forms of secret writing required little more than local pen and paper analogs, as most people could not read. More literacy or literate opponents, required actual cryptography. The main classical cipher types are transposition ciphers, which rearrange the order of letters in a message (e.g. 'hello world' becomes 'ehlol owrdl' in a trivially simple rearrangement scheme) and substitution ciphers, which systematically replace letters or groups of letters with other letters or groups of letters (e.g. 'fly at once' becomes 'gmz bu podf' by replacing each letter with the one following it in the Latin alphabet). Simple versions of either have never offered much confidentiality from enterprising opponents. An early substitution cipher was the Caesar cipher, in which each letter in the plaintext was replaced by a letter some fixed number of positions further down the alphabet. Suetonius reports that Julius Caesar used it with a shift of three to communicate with his generals. Atbash is an example of an early Hebrew cipher. The earliest known use of cryptography is some carved ciphertext on stone in Egypt (ca 1900 BC), but this may have been done for the amusement of literate observers rather than as a way of concealing information. Cryptography is recommended in the Kama Sutra as a way for lovers to communicate without inconvenient discovery.

The Greeks of Classical times are said to have known of ciphers (e.g. the scytale transposition cipher claimed to have been used by the Spartan military). Steganography (i.e. hiding even the existence of a message so as to keep it confidential) was also first developed in ancient times. An early example, from Herodotus, concealed a message-a tattoo on a slave's shaved head-under the regrown hair. Another Greek method was developed by Polybius (now called the "Polybius Square"). More modern examples of steganography include the use of invisible ink, microdots and digital watermarks to conceal information.

Ciphertexts produced by a classical cipher (and some modern ciphers) always reveal statistical information about the plaintext, which can often be used to break them. After the discovery of frequency analysis perhaps by the Arab mathematician and polymath, Al-Kindi (also known as *Alkindus*), in the 9th century, nearly all such ciphers became more or less readily breakable by any informed attacker. Such classical ciphers still enjoy popularity today, though mostly as puzzles. Al-Kindi wrote a book on cryptography entitled *Risalah fi Istikhraj al-Mu'amma (Manuscript for the Deciphering Cryptographic Messages)*, in which described the first cryptanalysis techniques.

Essentially all ciphers remained vulnerable to cryptanalysis using the frequency analysis technique until the development of the polyalphabetic cipher, most clearly by Leon Battista Alberti around the year 1467, though there is some indication that it was already known to Al-Kindi. Alberti's innovation was to use different ciphers (i.e. substitution alphabets) for various parts of a message (perhaps for each successive plaintext letter at the limit). He also invented what was probably the first automatic cipher device, a wheel which implemented a partial realization of his invention. In the polyalphabetic Vigenère cipher, encryption uses a *key word*, which controls letter substitution depending on which letter of the key word is used. In the mid-19th century Charles Babbage showed that the Vigenère cipher was vulnerable to Kasiski examination, but this was first published about ten years later by Friedrich Kasiski.

Although frequency analysis is a powerful and general technique against many ciphers, encryption has still been often effective in practice; many a would-be cryptanalyst was unaware of the technique. Breaking a message without using frequency analysis essentially required knowledge of the cipher used and perhaps of the key involved, thus making espionage, bribery, burglary, defection, etc. more attractive approaches to the cryptanalytically uninformed. It was finally explicitly recognized in the 19th century that secrecy of a cipher's algorithm is not a sensible nor practical safeguard of message security; in fact, it was further realized that any adequate cryptographic scheme (including ciphers) should remain secure even if the adversary fully understands the cipher algorithm itself. Security of the key used should alone be sufficient for a good cipher to maintain confidentiality under an attack. This fundamental principle was first explicitly stated in 1883 by Auguste Kerckhoffs and is generally called Kerckhoffs' principle; alternatively and more bluntly, it was restated by Claude Shannon, the inventor of information theory and the fundamentals of theoretical cryptography, as *Shannon's Maxim*-'the enemy knows the system'.

Different physical devices and aids have been used to assist with ciphers. One of the earliest may have been the scytale of ancient Greece, a rod supposedly used by the Spartans as an aid for a transposition cipher (see image above). In medieval times, other aids were invented such as the cipher grille, which was also used for a kind of steganography. With the invention of polyalphabetic ciphers came more sophisticated aids such as Alberti's own cipher disk, Johannes Trithemius' tabula recta scheme and Thomas Jefferson's multi-cylinder (not publicly known and reinvented independently by Bazeries around 1900). Many mechanical encryption/ decryption devices were invented early in the 20th century and several patented, among them rotor machines-famously including the Enigma machine used by the German government and military from the late '20s and during World War II. The ciphers implemented by better quality examples of these machine designs brought about a substantial increase in cryptanalytic difficulty after WWI.

The Computer Era

The development of digital computers and electronics after WWII made possible much more complex ciphers. Furthermore, computers allowed for the encryption of any kind of data representable in any binary format, unlike classical ciphers which only encrypted written language texts; this was new and significant. Computer use has thus supplanted linguistic cryptography, both for cipher design and cryptanalysis. Many computer ciphers can be characterized by their operation on binary bit sequences (sometimes in groups or blocks), unlike classical and mechanical schemes, which generally manipulate traditional characters (i.e., letters and digits) directly. However, computers have also assisted cryptanalysis, which has compensated to some extent for increased cipher complexity. Nonetheless, good modern ciphers have stayed ahead of cryptanalysis; it is typically the case that use of a quality cipher is very efficient (i.e., fast and requiring few resources, such as memory or CPU capability), while breaking it requires an effort many orders of magnitude larger and vastly larger than that required for any classical cipher, making cryptanalysis so inefficient and impractical as to be effectively impossible. Alternate methods of attack (bribery, burglary, threat, torture) have become more attractive in consequence.

The 3-by-5-mm chip embedded in the card is shown, enlarged. Smart cards combine low cost and portability with the power to compute cryptographic algorithms.

Extensive open academic research into cryptography is relatively recent; it began only in the mid-1970s. In recent times, IBM personnel designed the algorithm that became the Federal (i.e., US) Data Encryption Standard; Whitfield Diffie and Martin Hellman published their key agreement algorithm and the RSA algorithm was published in Martin Gardner's Scientific American column. Since then, cryptography has become a widely used tool in communications, computer networks and computer security generally. Some modern cryptographic techniques can only keep their keys secret if certain mathematical problems are intractable, such as the integer factorization or the discrete logarithm problems, so there are deep connections with abstract mathematics. There are no absolute proofs that a cryptographic technique is secure (but see one-time pad); at best, there are proofs that some techniques are secure if some computational problem is difficult to solve or this or that assumption about implementation or practical use is met.

As well as being aware of cryptographic history, cryptographic algorithm and system designers must also sensibly consider probable future developments while working on their designs. For instance, continuous improvements in computer processing power have increased the scope of brute-force attacks, thus when specifying key lengths, the required key lengths are similarly advancing. The potential effects of quantum computing are already being considered by some cryptographic system designers; the announced imminence of small implementations of these machines may be making the need for this preemptive caution rather more than merely speculative.

Essentially, prior to the early 20th century, cryptography was chiefly concerned with linguistic and lexicographic patterns. Since then the emphasis has shifted and cryptography now makes extensive use of mathematics, including aspects of information theory, computational complexity, statistics, combinatorics, abstract algebra, number theory and finite mathematics generally. Cryptography is, also, a branch of engineering, but an unusual one as it deals with active, intelligent and malevolent opposition (see cryptographic engineering and security engineering); other kinds of engineering (e.g. civil or chemical engineering) need deal only with neutral natural forces. There is also active research examining the relationship between cryptographic problems and quantum physics.

Check Your Progress 1

Notes: a) Write your answer in the space given below.

b) Compare your answer with the one given at the end of this unit.

1) Define cryptography.

....

....

....

....

2) Explain historical perspective of cryptography.

....

....

....

....

....

3) Explain public key cryptography.

....

....

....

....

4) Define hash function.

....

....

....

....

1.4 ENCRYPTION

Encryption is the automated process of hiding data so that no unauthorized people can access it. This is done by means of a procedure (algorithm) and a key. Decryption is the reverse process.

A crypto system is an implementation of an encryption scheme or algorithm. The making of crypto systems is called cryptography. Cryptography also includes making authentication or digital signature schemes that use an algorithm and a key. A clear message is called a plain text message, which is transformed by cryptography into a ciphertext message. In symmetric crypto systems, both sender and receiver use the same key. In asymmetric or public key cryptography, they use different keys. Symmetric keys are called secret keys, whereas public key encryption uses pairs consisting of one private and one public key.

Encryption is mainly used to ensure secrecy of data. It can also be used to secure authentication, certainly about the identity of the sender (digital signature) and certainly about the unimpairedness of data (integrity). This is especially important if electronic data flow is to have legal consequences.

Encryption is one common method of protecting information transmitted over unreliable links. In practice, the following is the mechanism of encryption:

1) The information (plain text) is encrypted (encoded) from its initial readable form to an internal form (cipher text). This internal form, although readable, does not make any sense.

2) The cipher text can be stored in a readable file or transmitted over unprotected channels.

3) The receiver must decrypt (decode) it back into clear text to understand the meaning of cipher text.

Algorithms used in Encryption

1) **Secret Key Althorithm** – A system where one secret key shared is called symmetric or secret key cryptography.

2) **Data Encryption Standard (DES)** – It is a symmetric cryptosystem. Here, the cipher text is decrypted using the same key. DES specifies a method for encrypting 64 bit blocks of clear data plaintext into corresponding 64 bit blocks of cipher text employing a user specified 56 bit key. DES is commonly used in the design, generation and verification of PINS.

3) **Public Key Algorithm** – A cryptosystem where two different keys are used for encryption and decryption is called Asymmetric or public key system.

4) **RSA Algorithm** – It is a asymmetric cryptographic algorithm and uses two different keys for encoding and decoding. It is a complex algorithm so far no breaking of RSA has been reported.

1.5 STAGES OF CRYPTOLOGY

The first stage for cryptology is encryption (cryptography). The information (plain text) is encoded from its initial readable form to the cipher text. It comprises of following terminologies:

1) **Crypto system** – It is a system of a secure key pair consisting of a private key, for creating a digital signature and a public key to verify the digital signature.

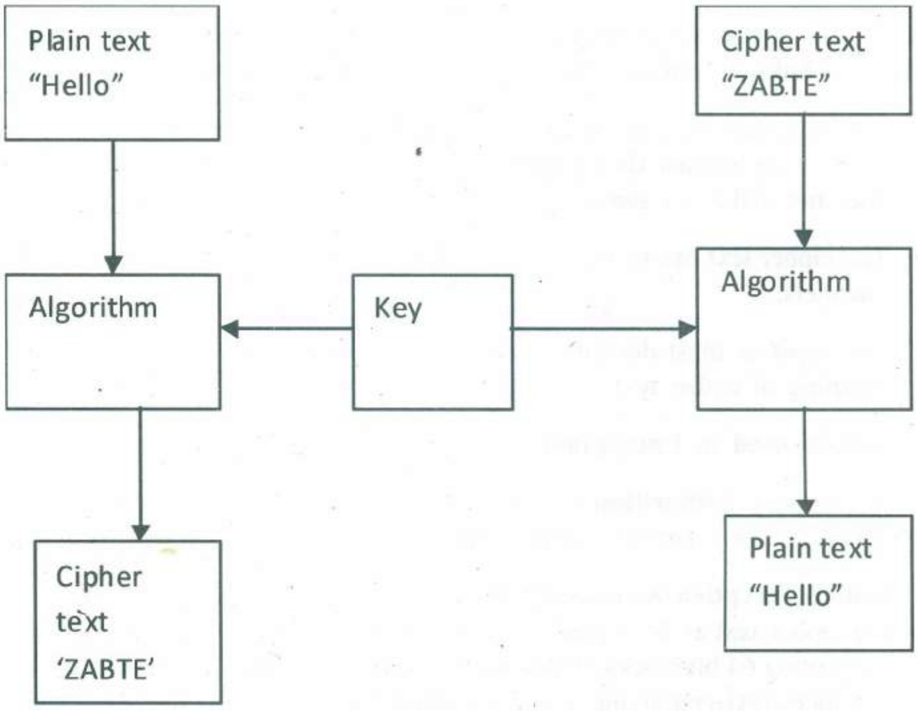
2) **Key pair** – It means a private key and its mathematically related public key, which are so related that the public key can verify a digital signature created by the private key.

3) **Private key** – It means the key of a key pair used to create a digital signature.

4) **Public key** – It means the key of a key pair used to verify a digital signature and listed in the Digital Signature certificate.

5) **Hash function** – It means an algorithm mapping or translation of one sequence of bits into another, generally smaller, set known as "hash result" such that an electronic record yields the same hash result every time the algorithm is executed with the same electronic record as its input making it computationally infeasible (a) to derive or reconstruct the original electronic record from the hash result produced by the algorithm. (b) That two electronic records can produce the same hash result using the algorithm.

The second stage for cryptology is decryption (cryptoanalysis). Here, the cipher text back into the original form, when done by the authorized person.



Encryption **Decryption**

Modern cryptographic systems are implemented with computer programs that have two inputs– the plaintext message and key, both of which are represented as sequences of 0s and 1s.

Check Your Progress 2

Notes: a) Write your answer in the space given below.

b) Compare your answer with the one given at the end of this unit.

1) Explain the encryption process.

....

....

....

....

....

2) What is the full form of DES?

....

....

....

....

....

3) What are the various stages of cryptology?

....

....

....

....

....

1.6 LET US SUM UP

Cryptography is the translation of information into a coded form using a key. Cryptography is the process of transforming plain text or original information into an unintelligible form (cipher text) so that it may be sent over unsafe channels of communication. Cryptography is a branch of applied mathematics which concerns with transforming messages into seemingly unintelligible forms and back again. Cryptography does for electronic information what locks and lockers do for printed information. Information is protected by scrambling it in such a manner that it can be unscrambled only with a secret key. The transformation process is controlled by a data string (key). The scrambled message called ciphertext is totally unintelligible to anyone who does not know the key. Anyone getting hold of the cipher text while it is on the unsafe channel would need to have the appropriate key to be able to get to the original information. The authorized receiver is assumed to have that key. The process of producing ciphertext is called encryption and the reverse process of restoring the original message called plaintext is called decryption.

1.7 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) Cryptography is the process of transforming plain text or original information into an unintelligible form (cipher text) so that it may be sent over unsafe channels of communication.

2) Before the modern era, cryptography was concerned solely with message confidentiality (i.e., encryption)-conversion of messages from a comprehensible form into an incomprehensible one and back again at the other end, rendering it unreadable by interceptors or eavesdroppers without secret knowledge (namely the key needed for decryption of that message). Encryption was used to (attempt to) ensure secrecy in communications, such as those of spies, military leaders and diplomats. In recent decades, the field has expanded beyond confidentiality concerns to include techniques for message integrity checking, sender/receiver identity authentication, digital signatures, interactive proofs and secure computation, among others.

3) Both parties have a private key and a public key. The private keys are known only to their owners, but the public keys are available to anyone (like telephone numbers). The sending party encrypts the message with the receivers public key and the receiver decrypts with his own private key. The public and private key constitute a key pair.

4) A hash function creates a fixed length string from a block of data. If the function is one way, it is also called a message digest function. These (fast) functions analyse a message and produce a fixed length digest which is practically unique i.e. finding a message with an identical hash very unlikely with very fast computers. There is no known feasible way of producing another message with the same digest. Such algorithms are normally used to create a signature for a message which can be used to verify it's integrity.

Check Your Progress 2

1) Encryption is one common method of protecting information transmitted over unreliable links. In practice, the following is the mechanism of encryption:

   a) The information (plain text) is encrypted (encoded) from its initial readable form to an internal form (cipher text). This internal form, although readable, does not make any sense.

   b) The cipher text can be stored in a readable file or transmitted over unprotected channels.

   c) The receiver must decrypt (decode) it back into clear text to understand the meaning of cipher text.

2) Data Encryption Standard.

3) The stages for Cryptology are encryption and decryption.

1.8 SUGGESTED READINGS

- Cryptology and Network Security by William Stallings
- Information Technology – Law and Practice by Vakul Sharma
- Law relating to Computers, Internet and E-commerce – A guide to cyber laws by Nandan Kamath
- www.cryptographworld.com

UNIT 2 SYMMETRIC KEY CRYPTOGRAPHY

Structure

2.0 Introduction

2.1 Objectives

2.2 Symmetric versus Asymmetric Cryptography

2.3 Symmetric Cryptography

2.4 Symmetric Key Cryptography

2.5 Types of Symmetric Ciphers

2.6 Examples of Symmetric Ciphers

2.7 Advantages and Disadvantages of Symmetric Cryptography

2.8 Let Us Sum Up

2.9 Check Your Progress: The Key

2.10 Suggested Readings

2.0 INTRODUCTION

A system where one secret key shared is called symmetric or secret key Cryptography. An encryption system in which the sender and receiver of a message share a single, common key that is used to encrypt and decrypt the message. Contrast this with public-key cryptology, which utilizes two keys – a public key to encrypt messages and a private key to decrypt them.

Symmetric-key systems are simpler and faster, but their main drawback is that the two parties must somehow exchange the key in a secure way. Public-key encryption avoids this problem because the public key can be distributed in a non-secure way and the private key is never transmitted.

Symmetric-key cryptography is sometimes called *secret-key cryptography*. The most popular symmetric-key system is the *Data Encryption Standard (DES)*.

2.1 OBJECTIVES

After studying this unit, you should be able to:

- difference between symmetric and asymmetric cryptography;
- explain symmetric cryptography;
- explain types of symmetric ciphers; and
- explain advantages and disadvantages of symmetric cryptography.

2.2 SYMMETRIC VERSUS ASYMMETRIC CRYPTOGRAPHY

Symmetric cryptography uses the same secret (private) key to encrypt and decrypt its data whereas asymmetric uses both a public and private key. Symmetric requires that the secret key be known by the party encrypting the data and the party decrypting the data. Asymmetric allows for distribution of your public key to anyone with which they can encrypt the data they want to send securely and then it can only be decoded by the person having the private key. This eliminates the need of having to give someone the secret key (as with symmetric encryption) and risk having it compromised.

Symmetric encryption algorithms encrypt and decrypt with the same key. Main advantages of symmetric algorithms are its security and high speed. Asymmetric encryption algorithms encrypt and decrypt with different keys. Data is encrypted with a public key, and decrypted with a private key. Asymmetric encryption algorithms are incredibly slow and it is impractical to use them to encrypt large amounts of data. Generally, symmetric encryption algorithms are much faster to execute on a computer than asymmetric ones. In practice they are often used together, so that a public-key algorithm is used to encrypt a randomly generated encryption key and the random key is used to encrypt the actual message using a symmetric algorithm.

The issue with asymmetric is that it is about 1000 times slower than symmetric encryption which makes it impractical when trying to encrypt large amounts of data. Also to get the same security strength as symmetric, asymmetric must use a stronger key than symmetric.

2.3 SYMMETRIC CRYPTOGRAPHY

Symmetric cryptography uses a single private key to both encrypt and decrypt data. Any party that has the key can use it to encrypt and decrypt data. They are also referred to as block ciphers. Symmetric cryptography algorithms are typically fast and are suitable for processing large streams of data. The disadvantage of symmetric cryptography is that it presumes two parties have agreed on a key and been able to exchange that key in a secure manner prior to communication. This is a significant challenge. Symmetric algorithms are usually mixed with public key algorithms to obtain a blend of security and speed.

With symmetric cryptography or symmetric-key encryption, the encryption key can be calculated from the decryption key and vice versa. With symmetric algorithms, the same key is used for both encryption and decryption. Symmetric key ciphers use the same key to both encrypt and decrypt data. This type of cipher is valuable because:

- It is relatively inexpensive to produce a strong key for these ciphers.
- The keys tend to be much smaller for the level of protection they afford.
- The algorithms are relatively inexpensive to process.

Therefore, implementing symmetric cryptography can be highly effective because you do not experience any significant time delay as a result of the encryption and decryption. Symmetric cryptography also provides a degree of authentication because data encrypted with one symmetric key cipher cannot be decrypted with any other symmetric key cipher. Therefore, as long as the symmetric key cipher is kept secret by the two parties using it to encrypt communications, each party can be sure that is communicating with the other as long as the decrypted messages continue to make sense.

Typically, with a symmetric key cipher, you can exchange the key with another trusted participant; usually you produce a unique key for each pair of participants. You can be assured that any messages that you exchange, which are encrypted in a specific key, between the participants can only be deciphered by the other participant that has that key. In this way, the key must be kept secret to each participant. Consequently, these keys are also referred to as secret-key ciphers.

Therefore, symmetric cryptography is effective only if the symmetric key cipher is kept secret by the two parties involved. If anyone else finds the key, it affects both confidentiality and authentication. A person with an unauthorized symmetric key cipher not only can decrypt messages sent with that key, but can encrypt new messages and send them as if they came from one of the two parties who were originally using the key.

The major drawback to secret-key ciphers is in exchanging the secret key because any exchange must retain the privacy of the key. This usually implies that any key is also encrypted, but in a different key, because the recipient must already have the key that will be needed to decrypt the key-exchange message. This can lead to a never-ending dependency on another key. Symmetric cryptography plays an important role in the SSL protocol and encryption over TCP/IP networks.

2.4 SYMMETRIC KEY CRYPTOGRAPHY

Both parties exchanging data have a key; this key is used to encrypt the data before transmission on one side and to decrypt on receipt on the other side. There are two kinds of symmetric ciphers: Block (which encrypt blocks of data at a time) and stream ciphers (which encrypt each bit/byte or word sequentially). Sample algorithms:

- The well known DES (Data Encryption Standard) is a shared key block cipher. DES was developed under contract to NIST (National Institute of Standards and Technology) by IBM. In basic mode it encrypts 64-bit blocks of plaintext, with a 56 bit key, using 16 iterations of an elaborate combination of table lookups and bit rearrangements.
  - The US government certified DES in 1977 (it became an ANSI standard in 1981) and continues to re-certify DES every 5 years. With the advances in computing hardware, DES is now breakable by large organisations with significant resources (by brute force: trying all possible combinations of the key). This is mainly due to the relatively small key size of 56 bits used by DES.
  - One solution is the so called Triple-DES or 3DES system, in which the data block is encrypted (two or) three times using three different keys (in a time slightly faster than 3 times a normal encryption). 3DES has a key strength roughly equivalent to 112bits. 3DES has not yet been certified by the US government, but it is unlikely that the original DES will be re-certified again.
  - There are several "modes of operation" that DES can use, Electronic Codebook (ECB), Cipher Feedback (CFB) and Cipher Block Chaining (CBC). The U.S. government recommends not using the weakest mode ECB. Unfortunately, many commercial encryption packages use ECB mode.
  - DESX is a modified version of DES which apparently strengthens it significantly (see the RSA Faq).
- RC2 is a block cipher from RSA Inc., that was a trade secret until anonymously published on the Internet in 1996. It seems quite strong and allows key sizes between 40 and 255 bits (or 2048 bits??).
- RC4, RC5 are (proprietary) variable key size stream ciphers from RSA Inc., developed in 1994. Since key size is variable, they can be more or less secure

than DES. USA export approved versions have approx. 40-bit key sizes. Domestic versions can have keys between 40 and 1024 bits. RC4 is the fastest (it was also published as an Internet draft without RSA's approval, called "ARCFOUR" in 1994) and RC5 is considered the "safest"..

- **IDEA:** Developed by the Swiss ETH University in Zurich and Ascom (patented). Published in 1990 and finalised in 1992 by Lai & Massey, it uses a 128 bit key. No weaknesses are currently known in this algorithm and a brute-force attack will not be feasible in the foreseeable future. It is patented by Ascom Tech AG. The licensing terms are basically: personal use is free and integrating IDEA into a sellable product costs money. PGP uses IDEA as it's symmetric algorithm.
- **Blowfish:** is a public domain algorithm, that is new (1993) but hasn't shown any major weaknesses. It is fast and compact, with variable key sizes (32-448 bits, typically 128 or 256 bit), uses 8 byte blocks and is optimised for 32 and 64bit processors.
- **AES:** The Advanced Encryption Standard is designed to replace DES. NIST is accepting proposals until June 1998. It should have keys of 128, 192, 256 bits and use 128 bit blocks. The final algorithm probably won't be available until the year 2000. Schneier has proposed Twofish for AES, which is a 128bit block, 16 round block cipher that is in the public domain, faster than Blowfish and requires few resources (can run on smart cards).
- **CAST:** It is a block cipher from Carlisle Adams and Stafford Tavares of Northern Telecom (Nortel). It is fast. Nortel has applied for a patent for CAST, but they have made a commitment in writing to make CAST available to anyone on a royalty-free basis. CAST has no weak or semiweak keys. There are strong arguments that CAST is completely immune to both linear and differential cryptanalysis, the two most powerful forms of cryptanalysis in the published literature, both of which have been effective in cracking DES. CAST is too new to have developed a long track record, but its formal design and the good reputations of its designers will undoubtedly attract the attentions and attempted cryptanalytic attacks of the rest of the academic cryptographic community.

Advantages

Shared key algorithms are much faster than their public key counterparts.

Disadvantages

Both side must know the same key and they must find a secure way of exchanging it (via a separate secure channel).

Typical applications

Encryption of information to protect privacy. i.e. local encryption of data files (where no transmission is required), data session encryption, banking systems (PIN encryption).

2.5 TYPES OF SYMMETRIC CIPHERS

Block Ciphers

Block ciphers convert a fixed-length block of plain text into cipher text of the same length, which is under the control of the secret key. Decryption is effected using the reverse transformation and the same key. For many current block ciphers the block size is 64 bits, but this is likely to increase.

Plain text messages are typically much longer than the particular block size and different techniques or modes of operation, that are used. Examples of such modes are electronic codebook (ECB), cipher block chaining (CBC) or cipher feedback (CFB). ECB simply encrypts each block of plain text, one after another, using the same key; in CBC mode, each plain text block is XORed with the previous cipher text block before being encrypted, thus adding a level of complexity that can make certain attacks harder to mount. Output FeedBack mode (OFB) resembles CBC mode although the quantity that's XORed is generated independently. CBC is widely used, for example in DES (qv) implementations and these various modes are discussed in depth in appropriate books on technical aspects of cryptography. Note that a common vulnerability of roll-your-own cryptosystems is to use some published algorithm in a simple form rather than in a particular mode that gives additional protection.

Iterated block ciphers are those where the process of encryption has several rounds, thus improving security. In each round, an appropriate transformation may be applied using a subkey derived from the original secret key that uses a special function. Inevitably, this additional computing requirement has an impact on the speed at which encryption can be managed, therefore there is a balance between security needs and speed of execution. Nothing is free and in cryptography; as elsewhere, part of the skill in applying appropriate methods is derived from understanding the tradeoffs that need to be made and how these relate to the balance of requirements.

Block ciphers include DES, IDEA, SAFER, Blowfish and Skipjack -- this last being the algorithm used in the US National Security Agency (NSA) Clipper chip.

Stream Ciphers

Stream ciphers can be extremely fast compared with block ciphers although some block ciphers working in certain modes (such as DES in CFB or OFB) effectively operate as stream ciphers. Stream ciphers operate on small groups of bits, typically applying bitwise XOR operations to them using as a key a sequence of bits, known as a keystream. Some stream ciphers are based on what is termed a Linear Feedback Shift Register (LFSR), a mechanism for generating a sequence of binary bits.

Stream ciphers are developed out of a specialist cipher, the Vernam cipher, also known as the one-time pad. Examples of stream ciphers include RC4 and the Software Optimized Encryption Algorithm (SEAL), as well as the special case of the Vernam cipher or one-time pad.

Message Authentication Codes

A message authentication code (MAC) is not a cipher but a particular form of checksum, typically 32 bits, generated using a secret key in combination with a particular authentication scheme and appended to a message. In contrast to message digests, generated using a one-way hash function and the closely-connected digital signature, generated and validated using asymmetric key pairs, the intended recipient requires access to the secret key in order to validate the code.

2.6 EXAMPLES OF SYMMETRIC CIPHERS

DES

Data Encryption Algorithm (DEA), of which the Data Encryption Standard (DES) is the formal description, derives from work done by IBM and adopted officially by the US government in 1977. It is probably the most widely used secret key system, particularly in securing financial data and was originally developed to be embedded in hardware. Automated Teller Machines (ATMs) typically use DES.

DES uses a 56-bit key with an additional eight parity bits to bring the block size up to 64 bits. It's an iterated block cipher using what's known as Feistel techniques where the text block being encrypted is split into two halves. The round function is applied to one half using a subkey and that output is then XORed with the other half; the two halves are then swapped and the process continues except that the last round is not swapped. DES uses 16 rounds.

The main form of attack on DES is what's known as brute force or exhaustive key search, a repeated trying of keys until one fits. Given that DES uses a 56-bit key, the number of possible keys is 2 56. With the growth in power of computer systems, this makes DES far less secure than it was when first implemented, although for practical purposes of a non-critical nature, it can still be considered adequate. However, DES is now certified only for legacy systems and a new encryption standard – Advanced Encryption Standard (AES) – has been selected.

A common variant on DES is triple-DES, a mechanism that encrypts the material three times using a key of 168; this generally (but not always) provides considerably more security. If the three-key 56-bit sub-elements are identical, then triple-DES is backwards compatible with DES.

For years, IBM held a patent on DES, but this expired in 1983 and was placed in the public domain, allowing royalty-free use under certain conditions.

IDEA

The International Data Encryption Algorithm (IDEA) was developed at ETH in Zurich by two researchers, Xuejia Lai and James L. Massey, with the patent rights held by a Swiss company, Ascom Systec. IDEA is implemented as an iterative block cipher and uses 128-bit keys and eight rounds. This gives much more security than DES does, but when choosing keys for IDEA it's important to exclude what are known as "weak keys." Whereas DES has only four weak keys and 12 semi-weak keys, the number of weak keys in IDEA is considerable at 2 51. However, given that the total number of keys is substantially greater at 2 128 this still leaves 2 77 keys to choose from.

IDEA is widely available throughout the world with royalty charges, typically of around $6.00 a copy (these charges apply in some areas but not in others. IDEA is considered extremely secure. With a 128-bit key, the number of tests made in a brute force attacks needs to be increased significantly compared with DES, even allowing for weak keys. Further, it's shown itself particularly resistant to specialist forms of analytical attack.

CAST

CAST is named for its designers, Carlisle Adams and Stafford Tavares of Nortel. It's a 64-bit Feistel cipher using 16 rounds and allowing key sizes up to 128 bits. A variant, CAST-256, uses a 128-bit block size and allows the use of keys of up to 256 bits.

Although CAST is fairly new, it appears to be extremely secure against attacks, both brute force and analytical. Although reasonably fast, its main benefit is security rather than speed. It is used in recent versions of PGP as well as in products from IBM, Microsoft and elsewhere.

Entrust Technologies holds a patent on CAST but says that it can be used without royalty payments in both commercial and non-commercial applications.

The One-time Pad

The one-time pad or Vernam cipher, has the merit of being considered completely secure and so has great value in certain specialized situations, typically in war time. It uses a randomly-generated key exactly as long as the message. This is applied to the plain text, typically using bitwise XOR, to produce the encrypted text. Applying the same key and appropriate algorithm easily decrypts the message:

Simple illustration of one-time pad encryption/decryption

00101100010....11011100101011 Original plain text message

01110111010....10001011101011 Randomly generated key equal to message in length

01011011000....01010111000000 Encrypted message

01110111010....10001011101011 Key re-used to decrypt

00101100010....11011100101011 Original message restored

Although the one-time pad is completely and absolutely secure, it is often not very practical, since the key of the same length as the message needs to be transmitted in some secure way to the receiver to allow decryption. Further, the key is used only once and is then discarded and although this clearly benefits security, it adds to the key management problems. One area where the one-time pad might currently be used is in MACs.

AES

The Advanced Encryption Standard (AES) is intended to replace DES as a new, secure standard, given that DES has reached the end of its useful life. In 1997, a competition was announced by the US National Institute of Standards and Technology (NIST) and the 15 original entries were reduced to a short list of five. The eventual winner was a product submitted by Joan Daemen and Vincent Rijmen of Belgium, named Rijndael, which is currently undergoing extensive trials and evaluation.

Rijndael is technically complex and somewhat unconventional in its construction but appears to be extremely secure and versatile in that it is fast in execution, well-suited to modern requirements (such as in smart cards) and capable of being used with a range of key sizes.

2.7 ADVANTAGES AND DISADVANTAGES OF SYMMETRIC CRYPTOGRAPHY

Because symmetric key cryptography uses the same key for both decryption and encryption, it is much faster than public key cryptography, is easier to implement and generally requires less processing power. A disadvantage of symmetric key cryptography is that the parties sending messages to each other must agree to use the same private key before they start transmitting secure information. This may be impossible depending on the circumstances – because the parties who want to communicate with each other through a secure means may be on different sides of the world. And this means that they will need a secure way to tell each other what the private key will be – if there were a secure way to do this, then the cryptography would not have been necessary in the first place in order to create that secure channel.

The advantage of using public key cryptography is that the public key used for encryption does not need to remain secure (that is why it's called "public" – because it does not matter if other people know about it). What often happens is that people use public key cryptography to create a shared session key and then they communicate through symmetric key cryptography using the shared session key. This way they can get the best of both worlds – the performance/speed of shared key cryptography along with the convenience of public key cryptography.

Check Your Progress 1

Notes: a) Write your answer in the space given below.

b) Compare your answer with the one given at the end of this unit.

1) Explain symmetric cryptography and its working.

....

....

....

....

....

2) Explain the difference between symmetric and asymmetric cryptography.

....

....

....

3) Explain DES.

....

....

....

....

4) What is the full form of AES?

....

....

....

5) Explain types of symmetric cipher.

....

....

....

6) Write certain examples of symmetric ciphers.

....

....

....

7) What are the advantages and disadvantages of symmetric cryptography?

....

....

....

....

2.8 LET US SUM UP

Symmetric-key algorithms are a class of algorithms for cryptography that use trivially related, often identical, cryptographic keys for both decryption and encryption. In symmetric key cryptography the encryption key and decryption keys are either same are easily derivable from each other. Other terms for symmetric-key encryption are secret-key, single-key, shared-key, one-key encryption.

Symmetric key cryptography is often much faster than asymmetric or public-key cryptography so it's preferred for encrypting large amounts of data. But the key length and complexity in current crypto systems don't make it feasible to transfer the shared secret in a telephone call. So public-key technology is often used to encrypt only the shared secret. First the shared secret is decrypted and then symmetric key cryptography is used to efficiently decrypt the large blocks of data.

2.9 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) Symmetric cryptography uses the same secret (private) key to encrypt and decrypt its data whereas asymmetric uses both a public and private key. Symmetric requires that the secret key be known by the party encrypting the data and the party decrypting the data. Asymmetric allows for distribution of your public key to anyone with which they can encrypt the data they want to send securely and then it can only be decoded by the person having the private key. This eliminates the need of having to give someone the secret key (as with symmetric encryption) and risk having it compromised.

2) Symmetric cryptography uses the same secret (private) key to encrypt and decrypt its data whereas asymmetric uses both a public and private key. Symmetric requires that the secret key be known by the party encrypting the data and the party decrypting the data. Asymmetric allows for distribution of your public key to anyone with which they can encrypt the data they want to send securely and then it can only be decoded by the person having the private key. This eliminates the need of having to give someone the secret key (as with symmetric encryption) and risk having it compromised.

3) Data Encryption Algorithm (DEA), of which the Data Encryption Standard (DES) is the formal description, derives from work done by IBM and adopted officially by the US government in 1977. It is probably the most widely used secret key system, particularly in securing financial data and was originally developed to be embedded in hardware. Automated Teller Machines (ATMs) typically use DES. DES uses a 56-bit key with an additional eight parity bits to bring the block size up to 64 bits. It's an iterated block cipher using what's known as Feistel techniques where the text block being encrypted is split into two halves. The round function is applied to one half using a subkey and that output is then XORed with the other half; the two halves are then swapped and the process continues except that the last round is not swapped. DES uses 16 rounds.

4) Advanced Encryption Standard

5) Block and stream ciphers

6) DES, IDEA, CAST

7) Because symmetric key cryptography uses the same key for both decryption and encryption, it is much faster than public key cryptography, is easier to implement and generally requires less processing power. A disadvantage of symmetric key cryptography is that the parties sending messages to each other must agree to use the same private key before they start transmitting secure information. This may be impossible depending on the circumstances – because the parties who want to communicate with each other through a secure means may be on different sides of the world. And this means that they will need a secure way to tell each other what the private key will be – if there were a secure way to do this, then the cryptography would not have been necessary in the first place in order to create that secure channel.

   The advantage of using public key cryptography is that the public key used for encryption does not need to remain secure (that is why it's called "public" – because it does not matter if other people know about it). What often happens is that people use public key cryptography to create a shared session key and then they communicate through symmetric key cryptography using the shared session key. This way they can get the best of both worlds – the performance/ speed of shared key cryptography along with the convenience of public key cryptography.

2.10 SUGGESTED READINGS

- Information Technology – Law and Practice by Vakul Sharma
- Law relating to Computers, Internet and E-commerce – A guide to cyberlaws by Nandan Kamath
- www.aspencrypt.com
- www.securitycerts.org
- www.symmetriccryptography.com
- www.wikipedia.org

UNIT 3 ASYMMETRIC KEY CRYPTOGRAPHY

Structure

3.0 Introduction
3.1 Objectives
3.2 Asymmetric Cryptography
3.3 How Asymmetric Encryption Works?
3.4 Hash Function
3.5 Types of Asymmetric Cryptology
3.6 Let Us Sum Up
3.7 Check Your Progress: The Key
3.8 Suggested Readings

3.0 INTRODUCTION

Asymmetric cryptography or public-key cryptography is cryptography in which a pair of keys is used to encrypt and decrypt a message so that it arrives securely. Initially, a network user receives a public and private key pair from a certificate authority. Any other user who wants to send an encrypted message can get the intended recipient's public key from a public directory. They use this key to encrypt the message and they send it to the recipient. When the recipient gets the message, they decrypt it with their private key, which no one else should have access to.

3.1 OBJECTIVES

After studying this unit, you should be able to:

- explain asymmetric cryptography;
- explain hash function; and
- recognize types of asymmetric cryptography.

3.2 ASYMMETRIC CRYPTOGRAPHY

In asymmetric cryptography (also known as "public key cryptography"), a public key and private key are used as a two-key system which allows for free distribution of the encryption key while making the decryption key public. This gives the benefit of allowing a person to give away the encryption key to anyone while still keeping the data enciphered and sent to them secure. Even if someone intercepts the message and has a copy of the public key, they can't access the data. This is the major benefit of using the two-key system rather than one-key. A graphical explanation of this is shown to the right.

Asymmetric key algorithms can also be used in the reverse- proving someone's identity to a large group of people. Suppose a big political leader wants to make announcements via the internet, but wants to ensure against imposters. He would give away the public key to everyone, encrypt data with the private key and send out the ciphertext via the internet. Since he has the only private key, only messages that were actually from him would be readable with his corresponding public key.

Mathematically, asymmetric encryption is considerably more complicated than single-key ciphers. The basic idea is to make computations which are incredibly hard without the key, but simple when you have the key available. With RSA (and several other 2-key ciphers) this is done with a pair of very large prime numbers. With the public key and these two large primes, the calculation of the private key (and therefore the decryption of the ciphertext) is simple. However, without the two primes, the calculation is incredibly complicated, with millions or billions of possibilities. This renders the public key useless in decryption, but accessible to encryption functionality on its own.

This use of two key encryption is the basis for digital signatures and certificates as well. *Digital signatures* are essentially an assurance of authentication and non-repudiation by tying a public key to a private key. Just like it's physical counterpart (theoretically), a digital signature is an undeniable proof of identity– it can't be forged and therefore cannot be denied at a later time by it's owner.

Digital signatures would be done by using both sets (sender and receiver) of public/private keys. The sender would encrypt a message using the receivers public key so only the receiver can read it. He would then encrypt the ciphertext "message" with his *private key*, so anyone who decrypts the message knows it comes from him. The message is sent and decrypted twice, once for authentication/non-repudiation and again for confidentiality.

Certificates are similar, but use a certificate authority (CA) which acts as a trusted third party or voucher, to verify identity. Certificates have more details, however, giving the user's name, the CA, expiration date (if applicable) and any approved operations. This is a major aspect in modern e-commerce systems, where companies such as Verisign act as a reputable 3rd party, issuing certificates to companies so customers can ensure they are communicating with the company they expect.

3.3 HOW ASYMMETRIC ENCRYPTION WORKS?

The process of asymmetric encryption involves two keys: one key for encryption and a second key for decryption. An asymmetric key encryption scheme involves six main parts:

Plaintext

This is the text message to which an algorithm is applied.

Encryption Algorithm

It performs mathematical operations to conduct substitutions and transformations to the plaintext.

Public and Private Keys

These are a pair of keys where one is used for encryption and the other for decryption.

Ciphertext

This is the encrypted or scrambled message produced by applying the algorithm to the plaintext message using key.

Decryption Algorithm

This algorithm generates the ciphertext and the matching key to produce the plaintext.

The sender and the recipient must have the same software. The recipient makes a pair of keys – public key and private key (both keys can be unlocked with a single password). Public key can be used by anyone with the same software to encrypt a message. Public keys can be freely distributed without worrying since it is only used to scramble (encrypt) the data. The sender does not need the recipient's password to use his or her public key to encrypt data. The recipient's other key is a private key that only he or she can use when decrypting the message. Private key should never be distributed since the private key assures that only the intended recipient can unscramble (decrypt) data intended for him or her. The recipient can freely distribute the public key without worrying since it is only used to scramble the data.

To understand asymmetric encryption better please read an example:

Jack makes public key A and private key A and Jill makes public key B and private key B. Jack and Jill exchange their public keys. Once they have exchanged keys, Jack can send an encrypted message to Jill by using Jill's public key B to scramble the message. Jill uses her private key B to unscramble it. If Jill wants to send an encrypted message to Jack, she uses Jack's public key A to scramble her message, which Jack can then unscramble with his private key A. Asymmetric cryptography is typically slower to execute electronically than symmetric cryptography.

Some Asymmetric Algorithms (public key algorithms) such as RSA allow the process to work in the opposite direction as well: a message can be encrypted with a private key and decrypted with the corresponding public key. If the recipient wants to decrypt a message with Bob's public key he/she must know that the message has come from Bob because no one else has sender's private key.

3.4 HASH FUNCTION

A "Hash function" is a complex encryption algorithm used primarily in cryptography and is like a shortened version of full-scale encryption. A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, h = H(m)). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.

The basic requirements for a cryptographic hash function are:

- the input can be of any length,
- the output has a fixed length,
- H(x) is relatively easy to compute for any given x ,
- H(x) is one-way,
- H(x) is collision-free.

A hash function H is said to be one-way if it is hard to invert, where "hard to invert" means that given a hash value h, it is computationally infeasible to find some input x such that H(x) = h.

If, given a message x, it is computationally infeasible to find a message y not equal to x such that H(x) = H(y) then H is said to be a weakly collision-free hash function.

A strongly collision-free hash function H is one for which it is computationally infeasible to find any two messages x and y such that H(x) = H(y).

The hash value represents concisely the longer message or document from which it was computed; one can think of a message digest as a "digital fingerprint" of the larger document. Examples of well-known hash functions are MD2 and MD5 and SHA.

Perhaps the main role of a cryptographic hash function is in the provision of digital signatures. Since hash functions are generally faster than digital signature algorithms, it is typical to compute the digital signature to some document by computing the signature on the document's hash value, which is small compared to the document itself. Additionally, a digest can be made public without revealing the contents of the document from which it is derived. This is important in digital timestamping where, using hash functions, one can get a document timestamped without revealing its contents to the timestamping service.

Cryptographic hash functions have many information security applications, notably in digital signatures, message authentication codes (MACs) and other forms of authentication. They can also be used as ordinary hash functions, to index data in hash tables, for fingerprinting, to detect duplicate data or uniquely identify files and as checksums to detect accidental data corruption. Indeed, in information security contexts, cryptographic hash values are sometimes called (digital) fingerprints, checksums or just hash values, even though all these terms stand for functions with rather different properties and purposes.

3.5 TYPES OF ASYMMETRIC CRYPTOLOGY

RSA asymmetric encryption

RSA is the best known asymmetric (public key) algorithm, named after its inventors: Rivest, Shamir and Adleman. RSA uses public and private keys that are functions of a pair of large prime numbers. Its security is based on the difficulty of factoring large integers. The RSA algorithm can be used for both public key encryption and digital signatures. The keys used for encryption and decryption in RSA algorithm, are generated using random data. The key used for encryption is a public key and the key used for decryption is a private key. Public keys are stored anywhere publicly accessible. The sender of message encrypts the data using public key and the receiver decrypts it using his/her own private key. That way, no one else can intercept the data except receiver.

DSA

The Digital Signature Algorithm (DSA) is a United States Federal Government standard or FIPS for digital signatures. It was proposed by the National Institute of Standards and Technology (NIST) in August 1991 for use in their Digital Signature Standard (DSS).

PGP

PGP (Pretty Good Privacy) is a public-private key cryptography system which allows for users to more easily integrate the use of encryption in their daily tasks, such as electronic mail protection and authentication and protecting files stored on a computer. PGP was originally designed by Phil Zimmerman. It uses IDEA, CAST or Triple DES for actual data encryption and RSA (with up to 2048-bit key) or DH/DSS (with 1024-bit signature key and 4096-bit encryption key) for key management and digital signatures. The RSA or DH public key is used to encrypt the IDEA secret key as part of the message.

Check Your Progress 1

Notes: a) Write your answer in the space given below.

b) Compare your answer with the one given at the end of this unit.

1) Define asymmetric cryptography.

....

....

....

....

2) Explain the working of asymmetric encryption.

....

....

....

....

3) Explain Hash function.

....

....

....

....

4) What are the types of asymmetric cryptology?

....

....

....

....

3.6 LET US SUM UP

The problem with secret keys is exchanging them over the Internet or a large network while preventing them from falling into the wrong hands. Anyone who knows the secret key can decrypt the message. One answer is asymmetric encryption, in which there are two related keys – a key pair. A public key is made freely available to anyone who might want to send you a message. A second, private key is kept secret, so that only you know it.

Any message (text, binary files or documents) that are encrypted by using the public key can only be decrypted by applying the same algorithm, but by using the matching private key. Any message that is encrypted by using the private key can only be decrypted by using the matching public key.

This means that you do not have to worry about passing public keys over the Internet (the keys are supposed to be public). A problem with asymmetric encryption, however, is that it is slower than symmetric encryption. It requires far more processing power to both encrypt and decrypt the content of the message.

3.7 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) Asymmetric cryptography or public-key cryptography is cryptography in which a pair of keys is used to encrypt and decrypt a message so that it arrives securely. In asymmetric cryptography (also known as "public key cryptography"), a public key and private key are used as a two-key system which allows for free distribution of the encryption key while making the decryption key public. This gives the benefit of allowing a person to give away the encryption key to anyone while still keeping the data enciphered and sent to them secure. Even if someone intercepts the message and has a copy of the public key, they can't access the data. This is the major benefit of using the two-key system rather than one-key. A graphical explanation of this is shown to the right.

2) The process of asymmetric encryption involves two keys: one key for encryption and a second key for decryption. An asymmetric key encryption scheme involves six main parts:

   Plaintext – this is the text message to which an algorithm is applied.

   Encryption Algorithm – it performs mathematical operations to conduct substitutions and transformations to the plaintext.

   Public and Private Keys – these are a pair of keys where one is used for encryption and the other for decryption.

   Ciphertext – this is the encrypted or scrambled message produced by applying the algorithm to the plaintext message using key.

   Decryption Algorithm – this algorithm generates the ciphertext and the matching key to produce the plaintext.

   The sender and the recipient must have the same software. The recipient makes a pair of keys – public key and private key (both keys can be unlocked with a single password). Public key can be used by anyone with the same software to encrypt a message. Public keys can be freely distributed without worrying since it is only used to scramble (encrypt) the data.The sender does not need the recipient's password to use his or her public key to encrypt data. The recipient's other key is a private key that only he or she can use when decrypting the message. Private key should never be distributed since the private key assures that only the intended recipient can unscramble (decrypt) data intended for him or her. The recipient can freely distribute the public key without worrying since it is only used to scramble the data.

3) A "Hash function" is a complex encryption algorithm used primarily in cryptography and is like a shortened version of full-scale encryption. A hash function H is a transformation that takes a variable-size input m and returns a fixed-size string, which is called the hash value h (that is, h = H(m)). Hash functions with just this property have a variety of general computational uses, but when employed in cryptography the hash functions are usually chosen to have some additional properties.

4) RSA, DSA, PGP

3.8 SUGGESTED READINGS

- Information Technology – Law and Practice by Vakul Sharma
- Law relating to Computers, Internet and E-commerce – A guide to cyber laws by Nandan Kamath
- searchsecurity.techtarget.com
- www.asymmetriccryptography.com
- www.**encryption**anddecryption.com/**encryption/asymmetric_encryption**.html
- www.digitalcomputersecurity.com

UNIT 4 APPLICATIONS OF CRYPTOGRAPHY

Structure

4.0 Introduction

4.1 Objectives

4.2 Applications of Cryptography

- 4.2.1 Applying Cryptography
  - 4.2.1.1 Encryption Strength
  - 4.2.1.2 Legal Issues / Export Restrictions
  - 4.2.1.3 Digital Time-Stamping Service (DTS)
  - 4.2.1.4 Certificates, Certification Authorities (CA), PKI and Trusted Third Parties (TTP)
  - 4.2.1.5 Emergency File Access
- 4.2.2 Secure Data Transmission using Cryptography
- 4.2.3 Authentication
  - 4.2.3.1 Summary of Authentication Mechanisms
  - 4.2.3.2 SSH
  - 4.2.3.3 Logon
  - 4.2.3.4 Firewalls & Authentication
  - 4.2.3.5 HTTP Basic Authentication
  - 4.2.3.6 NT Domains/Lan Manager/SMB & NetBIOS/NetBeui/CIFS
  - 4.2.3.7 Authentication Products
  - 4.2.3.8 Remote Access Control Protocols
  - 4.2.3.9 Availability Mechanisms

4.3 Limits of Encryption

4.4 Cryptographic Signatures (PKI)

4.5 Electronic Signatures

4.6 Electronic Signature Technology

- 4.6.1 Single Key Encryption
- 4.6.2 Process of Creating and Verifying a Digital Signature
- 4.6.3 Need/use of Verification

4.7 How Electronic Signature Technology Works?

4.8 Use of Electronic Signatures

4.9 Encryption and Steganography

4.10 Let Us Sum Up

4.11 Check Your Progress: The Key

4.12 Suggested Readings

4.0 INTRODUCTION

Cryptography can be used to protect stored data, including complete files and objects within files or communications, including phone calls, fax transmissions, email, web- transactions, banking transactions, corporate extranets and other types of network applications. Some encryption systems will encrypt everything on a hard disk so that the computer is effectively unusable without knowing the key. Encrypting files and complete disks is particularly useful with laptop computers. If the computer is stolen, sensitive data with not be exposed.

4.1 OBJECTIVES

After studying this unit, you should be able to:

- identify the application of cryptography;
- explain electronic signatures and its working;
- define steganography; and
- know PKI (Public key infrastructure).

4.2 APPLICATIONS OF CRYPTOGRAPHY

Encryption is available as hardware and software implementations both. They can be obtained as stand alone encryption devices and software packages or as a feature of other products. Many software applications and utility programs support encryption including spreadsheets, software for word processing, file management, databases, web browsing, e-mail and internet telephony. Increasingly, computer software comes with built-in encryption.

Communications can be either end-to-end encrypted or link encrypted. End-to-end encryption provides a secure channel between the end points of a message regardless of how many computers or links the message traverses. E-mail is usually end-to-end encrypted. Link encryption protects a message across a single link or sub-network but not across the entire path. Its advantage is that both endpoints need not support encryption or have compatible encryption. Global System for Mobile (GSM), which is used worldwide for digital cellular communications, uses link encryption to protect the over the air link between a mobile phone and a base station, which is the segment most prone to interception. The wireless link is encrypted regardless of whether the person at the other end of the communication is using a cell pone or GSM. Between base stations, the communications travel through the public telephone network, where segments may or may not be independently encrypted.

One increasingly popular application of encryption is Virtual Private Networks (VPNs). A VPN connects the geographically dispersed facilities of an enterprise over a public network like the Internet. It essentially provides secure global communications across the enterprise without the need for private leased lines. The VPN can be implemented with dedicated hardware or with software or it can be integrated into a firewall. A VPN is a cheaper alternative to leased lines. VPN over the internet can be implemented using IP– Layer encryption.

The Secure Socket Layer (SSL) protocol is used extensively on the web to protect credit card numbers and other sensitive data transmitted between a user's web browser and an Internet web server through the HTTP protocol. SSL supports different encryption systems and key lengths. The protocol is bundled into the web browsers, so it does not depend on the host computer to supply encryption. With SSL, a credit card number is encrypted by the customer's computer and decrypted by the merchant's. With the number in hand, the merchant then charges the purchase against the account. This process has a weakness. Insiders and intruders with access to the merchant's customer records can potentially compromise the card number.

The Secure Electronic Transaction (SET) protocol addresses this vulnerability in SSL by providing an encrypted channel between the customer and the bank. Upon receipt of an order, the merchant forwards the encrypted payment information to the bank. The bank decrypts the message, validates the payment information and informs the merchant whether to go ahead with the sale. With this approach, a customer's credit card number is never made available to the merchant and never exposed on the merchant's website or available to the merchant.

4.2.1 Applying Cryptography

Applications such as PGP, S/MIME, Secure RPC (and hence secure NFS & NIS+) and SKIP use a combination public key cryptography and symmetric cryptography to ensure non repudiation and privacy. Hashing algorithms are used for (fast) generation of signatures.

- The principal problem with most encryption systems is how to distribute and manage keys. Many systems require manual key-ring management. See Certification Authorities below.

4.2.1.1 Encryption Strength

There are several possible weaknesses in a crypto system and the strength of the system is the strength of the weakest link.

- The secrecy of the symmetric or private key.
- The difficulty of guessing the key or trying all possible keys. The key length determines the encryption strength of an algorithm. All cryptographic algorithms are vulnerable to "brute force" attacks (trying all possible key combinations).
- Bad implementation
  - "Pseudo" random number generators used in encryption engines may be (too) predictable. They must be at least as difficult to predict as it is difficult to guess the encryption key.
  - Algorithms can be incorrectly implemented.
  - Backdoors may exist.
- Bad design
  - Certain algorithms are easily inverted (easy to analyse and break), such as WinWord, Pkzip, WordPerfect etc.
  - Algorithms which are not published and subjected to peer review should not be considered as strong, "security through obscurity" is not a defence against the determined, financially powerful attacker.
  - Known plaintext attack: by encrypting many known texts and analysing the output, it may be possible to guess how the algorithm works.
- Mathematics advances each year, so new mathematical ideas can weaken existing cryptosystems (examples are the discovery of differential and linear cryptanlysis in recent years). The strength of current Public key (PK) systems is based on the difficulty of the mathematical factoring and discrete-logarithm problem. It is not impossible that faster mathematical methods to solving these problems be found, making PK guessing easier.

The following discussion concentrates on the issue of key lengths, but strong keys are useless if the above issues are not addressed.

Computers are getting faster (computing power doubles about every 2 years), cheaper and better networked each year. All cryptographic algorithms are vulnerable to "brute force" attacks (trying all possible key combinations).

Symmetric Algorithms

In general, the key length determines the encryption strength of an algorithm with the approximate formula of 2 to the power of the key length, so 56 bit keys take 65,536 times longer to crack than 40 bit keys.

Most products come from the U.S. and are subject to U.S. export restrictions, currently either a 40bit limit or escrowing of keys.

- 30 bits can be "brute force" guessed on a powerful PC.
- 40 bits:
    - In 1995, a French student Damien Doligez succeeded in breaking a 40bit Netscape shared encryption key in 8 days using a network of 120 UNIX machines (by brute force: trying all possible combinations of the key).
    - In 1996, an improved algorithm brought this down to 4 hours.
- 56 bits:
    - In 1997, 56 bit DES keys can be broken by dedicated chips (programmable gate arrays) within 3 weeks and by intelligence organisations such as the NSA within seconds.
    - Rumour has it that the NSA has a machine for several years that cracks DES in about 1-2 seconds.
- 64 bits: are probably breakable by governments and very powerful organisations today.
- 80 bits: probably not breakable today?
- 128 bits: probably not breakable in 50 years?

Public (asymmetric) Key Algorithms

- Key strength is more important for public keys since they are often use for digital signatures and non repudiation and are rarely changed.
- Public keys are longer than symmetric keys since the problem is guessing the private key, not the public. For the RSA algorithms this equates to factoring a large integer that has two large prime factors.
    - a 256 bit modulus is easily factored by ordinary people
    - 384 bit keys can be broken by university research groups or companies
    - 512 bits is within reach of major governments
    - Keys with 768 bits are probably not secure in the long term.
    - Keys with 1024 bits and more should be safe for now unless major algorithmic advances are made in factoring
    - keys of 2048 bits are considered by many to be secure for decades

The encryption key size should be chosen, based on:

- Who you wish to protect your information from (resources available to the attacker).
- How easy it is for an attacker to get hold of the encrypted information, e.g. how insecure the transport network is (sending information over the Internet certainly requires more protection that on your local subnet).

- how long the information must be protected. It is better to use weak encryption than not to protect data at all, however the danger of a weak encryption system is that it can give users a false sense of security.

Attacker	Time Span	Recommended key size
Curious hacker	Information must be protected for a few days.	Public Key 512 bits shared key 40 bits
Curious hacker	Information must be protected for minimum 2 years.	Public Key 1024 bits shared key 60 bits
Large organisation	Information must be protected for minimum 20 years.	Public Key 1568 bits shared key 90 bits
Government	Information must be protected for minimum 20 years.	Public Key 2048 bits shared key 128 bits

Here we define strong encryption as that which uses key sizes greater than or equal to:

Public Key 1568 bits (for RSA, DH and ElGamal)

Shared key 90 bits

"Strong" for new encryption system such as Elliptical curve or Quantum cryptography is not defined here, as yet.

4.2.1.2 Legal Issues/Export Restrictions

The U.S. and certain other countries consider encryption to be a weapon and strictly control exports. This is basically crippling the efforts to include encryption in Applications, Internet services such as Email and Operating systems.

In general the U.S. allows export of 40 bit shared key systems and 512 bit public key systems.

- Exceptions: There have been some exceptions to this rule, such as export to Canada & Australia and to large financial institutions world-wide.
- Lotus export Notes with a 64 bit key, of which 24bits are escrowed with the U.S. Govt., making more difficult for non U.S. agencies to look at your Notes communications!
- Certain products may be used by U.S. companies outside the U.S.
- Vendors have started building Interfaces into which strong encryption products can be plugged, assuming they're available internationally. E.g. Eudora Pro has a Plugin API which could allow seamless integration strong international encryption unit, without break U.S. law. Other examples are Sun (Solaris DES & Diffie Hellman libraries), Microsoft (NT Secure API), Qualcomm (Eudora Pro + PGP), various PGP Plugins and GUI's.

Some countries (e.g. France), forbid encryption except when a key has been deposit in an escrow (so the legal authorities can listen to all communications if they need).

Other countries allied to the U.S. (e.g. Germany, UK, Sweden, etc.) also enforce the U.S. restrictions by allowing strong encryption domestically, but restricting export of cryptographic devices.

Many countries have almost no restrictions, but some (especially European) countries are considering some kind of restriction of the use of cryptography in the future.

The only strong encryption software widely available internationally, known to the author of this document, are from Australia, Finland, Ireland and Russia.

4.2.1.3 Digital Time-Stamping Service (DTS)

A DTS issues a secure timestamp for a digital document.

- A message digest is produced of the document (by the sender) and sent to the DTS. The DTS sends back the timestamp, plus the date & time the timestamp was received with a secure signature form the DTS. This proves that the document existed on the said date. The document contents remain unknown to the DTS (only the digest is known).
- The DTS must use very long keys, since the timestamp may be required for many years.

4.2.1.4 Certificates, Certification Authorities (CA), PKI and Trusted Third Parties (TTP)

Certificates are digital documents attesting the identity of an individual to his public key. They allow verification that a particular public key does in fact belong to the presumed owner. The ISO certificate standard is X.509 v3 and is comprised of: Subject name, Subject attributes, Subject public key, Validity dates, Issuer name, Certificate serial number and Issuer signature. X.509 names are similar to X.400 mail addresses, but with a field for an Internet email address. The X.509 standard is used in S/MIME, SSL, S-HTTP, PEM, IPsec Key Management.

LDAP (Lighweight Directory Access Protocol) is an X.500 based directory service for certificate management. Certain secure email products such as PGP5 have inbuilt support for querying and updating LDAP servers.

Certificates are issued by the certification authority (CA). The CA is a trusted authority, who confirms the identity of users. The CA must have a trustworthy public key (i.e. very large) and it's private key must be kept in a highly secure location. CAs can also exist in a hierarchy, which lower level CAs trust high CAs.

Where sender and receiver must be absolutely sure of who their Peer is, a CA is a possible solution. Another name for a CA is a Trusted Third Party (TTP). If both sides trust a common authority, this authority can be used to validate credentials from each side. E.g. the sender sends his public key, name (and other validifying information) to the CA. The CA verifies this information as far as possible, add it's stamp to the packet and sends it to the receiver. The receiver can now be surer than the sender is who he says he is.

- The problem with CAs are that you have to trust them! Howëver, even Banks have overcome that problem with the implementation of SWIFT, a world wide financial transaction network.

4.2.1.5 Emergency File Access

A frequent requirement when protecting file confidentiality via encryption is Emergency File Access. If the file owner encrypts an important file and forgets the key, what happens? A second key is created, split into five parts such that any two of the five (partial) keys, when combined, could be used as a decryption key. The five (partial) keys could be kept by separate people, only to be used if the original owner was not able to decrypt the important file.

The Windows version of PGP supports these key splitting functions.

4.2.2 Secure Data Transmission using Cryptography

Secure data transmission is the exchange of data in a secure manner over (presumed) insecure networks.

Requirements

Secure data transmission is required for class systems or higher and can be divided into the following categories:

1) **Peer entity authentication:** Both sides (users & processes) must identify & authenticate themselves, prior to the exchange of data.

2) **Data integrity:** Data must remain complete during transmission. Unauthorised manipulation of user data, audit trail data and replay of transmissions shall be reliably identified as errors.

3) **Data confidentiality:** Only authorised persons should be able to access the data. (e.g. end-to-end data encryption).

4) **Data origin authentication:** Does the receiving process know who the data is coming from? For class systems, non repudiation of origin may be required: On receipt of data, it shall be possible to uniquely identify and authenticate the sender of the data. Has the receiver proof (e.g. digital signatures) of where information came from?

5) **Non repudiation of receipt:** Has the sender proof that the information sent was received by the intended receiver?

6) **Access control:** All information previously transmitted which can be used for unauthorised decryption shall be accessible only to authorised persons.

Secure data transmission is achieved by the use of cryptography. There are two principal cryptographic methods, public key and shared key. Normally a mixture of both is used for secure communication.

Using Cryptography for secure transmission, when choosing an authentication system, choose a signature function and encryption method and hash function that require comparable efforts to break. The encryption algorithms described in the previous section can be combined together to produce a system for secure data transmission (refer to the diagram below):

1) **Data integrity:** MD5 digests are created on the data part of message.

2) For performance reasons, normally it is sufficient to encrypt the MD5 digest noted above. The digest encrypted with the sender's private key is called a signature.

3) Non repudiation of receipt: not covered here.

4) **Data confidentiality:** Confidential parts of the message are encrypted. Shared key encryption is the most efficient method (performance). Normally the shared key is calculated from information known to both sides e.g. the sender uses his private key + receivers public key and the receiver uses his private key + senders public key. They can both generate the same unique key due to the mathematical properties of public key algorithms (i.e. multiplying numbers raised to powers). This data encryption key is often called the session key (it is valid only for a particular session).

5) **Peer entity authentication:** Where sender and receiver must be absolutely sure of who their peer is, a certification authority is a possible solution. If both sides trust a common authority, this Authority can be used to validate credentials from each side. E.g. the sender sends his public key, name (and other validifying information) to the Authority. The Authority verifies this information as far as possible, add it's stamp to the packet and sends it to the receiver. The receiver can now be surer than the sender is who he says he is. Similar encryption and hashing to that above would be applied this data.

6) Access control depends on implementation.

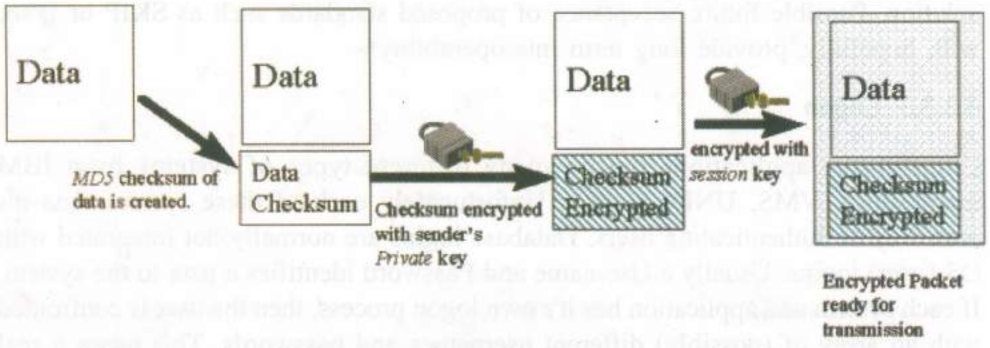

Data
Data
Data
Data
MD5 checksum of data is created.
Data
Checksum
Checksum encrypted with sender's Private key
Checksum
Encrypted
encrypted with session key
Checksum
Encrypted
Encrypted Packet ready for transmission

Fig. 1: Data is prepared for transmission

After receipt, the data is decrypted:

Example systems using this approach: Sun's Secure RPC (hence NIS+, NFS), SKIP, S/MIME isn't a million miles away either.

4.2.3 Authentication

Authentication is the process of verifying the identity of a subject. A subject (also called a principal) can be a user, a machine or a process i.e. a "network entity". Authentication uses something which is known to both sides, but not to others i.e. something the subject is, has or knows . Hence this can be biometrics (fingerprints, retina patterns, hand shape/size, DNA patterns, handwriting, etc.), passphrases, passwords, one-time password lists, identity cards, smart-tokens, challenge-response lists etc. Some systems consist of a combination of the above.

The most common methods of strong authentication today consist of one-time password lists (paper), automatic password generators (smart tokens) and intelligent identity cards.

4.2.3.1 Summary of Authentication Mechanisms

There is no industry standard today. Many different efforts are underway. In particular the Federated Services API, GSS API and RADIUS seem like a logical ways to interconnect the current incompatible systems, without requiring vendors to throw away their existing products. It is hard to imagine such an API offering more that basic functionality however (since advanced functionality is not common to all products). The IETF also have a number of active Authentication groups:

- Authenticated Firewall Traversal (aft)
- Common Authentication Technology (cat)
- One Time Password Authentication (otp)

For enterprise wide authentication and naming services DCE, NIS+ and NODS are the current main runners, with Microsoft's Active Directory service (planned for release with NT5) already generating interest for companies using NT Domains. Support for X.500 directory services will probably appear in most of these, allowing an interoperability gateway to be built. The fact that neither DCE nor NIS+ have been fully adopted in the PC client world is a pity, but perhaps reflects pricing and complexity problems.

4.2.3.2 SSH

SSH is a really impressive product for secure access to UNIX machines. It can use RSA, SecurID or UNIX user/password authentication.

For authentication across unsecured networks, proprietary (incompatible, expensive) encrypting firewalls using certificates or token based authentication are the current solution. Possible future acceptance of proposed standards such as SKIP or IPsec will, hopefully, provide long term interoperability.

4.2.3.3 Logon

Client/server applications run on many different types of systems from IBM mainframes, VMS, UNIX to PCs. Unfortunately each of these systems has it's own way of authenticating users. Database logins are normally not integrated with OS (user) logins. Usually a Username and Password identifies a user to the system. If each system and application has it's own logon process, then the user is confronted with an array of (possibly) different usernames and passwords. This poses a real security risk, as the user may be tempted to write down all the different passwords, change them rarely or use simple ones.

The ideal solution would be to provide a secure single signon. i.e. when a user logs on to a workstation on the network, his identity is established and can be shared with any system or application. Any user can sign from at any system anywhere and have the same name and password. The user needs to remember only one password. An even more secure signon can be achieved by using Personnel ID Cards to validate the user (via a card reader on each workstation) or via hand held Smartcards (with one time passwords).

Achieving single signon is not an easy task in today's heterogeneous environment, but it would seem that Kerberos is the main contender with Sun's NIS+ also an option.

4.2.3.4 Firewalls & Authentication

Strong authentication relies (normally) on something the user knows (e.g. a password) and something the user has (e.g. a list, smart card). Applications must support the authentication mechanism (or it must be transparent to the application). The following is a sample of strong firewall authentication methods/products.

Strong authentication mechanisms on Firewalls are very important, if protocols such as Telnet, Rlogin or ftp (writeable) are to be allowed. TCP/IP has inherent security weaknesses (confidentiality, IP spoofing) and these need to be addressed in a strong authentication product. If keys are used, key distribution must considered.

No standards exist, each product has it's own API and interoperability is often very difficult. Some Firewall authentication servers can act as glue, allowing a common database to be used for different authentication products (en example is the Gauntlet authentication server).

4.2.3.5 HTTP Basic Authentication

A basic authentication method is supported in HTTP.

Algorithm

A WWW client sends a request for a document which is protected by basic authentication. The server refuses access and sends code 401 together with header information indication that basic authentication is required. The client presents the user with a dialog to input username and password and passes this to the server. The server checks the user name and password and sent the document back if OK.

Encryption

Very weak. The user name and password are encoded with the base64 method. Documents are sent in clear text.

4.2.3.6 NT Domains/Lan Manager/SMB & NetBIOS/NetBeui /CIFS

NT's domains are an extension of (IBM/Microsoft) Lan Manager (LM) and are not hierarchical, but domain based - i.e. more suitable for separate LANs.

LM authentication has several dialects: PC NETWORK PROGRAM 1.0, MICROSOFT NETWORKS 3.0, DOS LM1.2X002, DOS LANMAN2.1, Windows for Workgroups 3.1a, NT LM 0.12, CIFS. The last two are the most interesting as they are used in NT4.

- The first few dialects are very old and if supported (and asked for by the client in the SMB protocol) , will send passwords in cleartext. The (autumn 1997) patches for NT4 & Win95 mandate the use of encryption by default. The late 1997 version of Samba also supports encryption and more interestingly "pass through" authentication.
- Several weakness were published in early 1997 and are partially fixed in NT4.0 SP3 and individual patches. Win95 also has several patches.
- For all dialects except for the last two (i.e. NT4), cracking the encrypted message that passed the network is not that hard: a dictionary attack, coupled with LM's uppercase passwords and division into two 7 byte words makes cracking of works less that 7 characters in an dictionary easy enough.

4.2.3.7 Authentication Products

Kerberos (+ DCE)

Kerberos is a secret-key network authentication service developed at MIT by Project Athena. It is used to authenticate requests for network resources in a distributed, real-time environment. DES (i.e. shared key) encryption and CRC/MD4/MD5 hashing algorithms are used. The source code is freely available (for non-commercial version) and Kerberos runs on many different systems.

Kerberos requires a "security server" or Kerberos server (KDC) which acts as a certification authority, managing keys and tickets. This server maintains a database of secret keys for each principal (user or host), authenticates the identity of a principal who wishes to access secure network resources and generate sessions keys when two users wish to communicate securely.

There are many versions of the Kerberos authentication system:V3 (MIT), V4 (commercial: Transarc, DEC) and V5 (in beta/RFC 1510, DCE, Sesame, NetCheque). BSDI is the only OS to bundle the Kerberos server. Solaris 2 bundles a Kerberos client, which among other things allows NFS to use Kerberos for authentication.

NIS+

NIS+ is a hierarchical enterprise wide naming system, based on Secure RPC. In the default configuration it provides user, group, services naming, automounter and key distribution. NIS+ can be easily extended to define customised tables.

NIS+ is an improved version of the UNIX defacto standard NIS (Network Information System or yellow pages). NIS & NIS+ were developed by Sun. NIS is available on most UNIX platforms, but has very weak security. NIS+ is much more secure but it only available on Sun's Solaris and recently HP-UX and AIX.

Security is based in the use of Secure RPC, which in turn uses the Diffie/Hellman public key cryptosystem.

- NIS+ is very flexible and can be easily extended to manage customised tables.
- It is stable (Solaris 2.3 or later with the correct patch) enough for production use.

- NIS+ is integrated into the Sun Federated Services (see below) with Solaris 2.5 and higher.

BoKS

BoKS is a full authentication/single signon package for PC and UNIX systems, made by DynaSoft in Sweden. DynaSoft is a 10 year old company employing about 50 people. The BoKS concept has been developed and improved by DynaSoft since 1987. It is a comprehensive security solution covering areas such as access control, strong authentication, encryption, system monitoring, alarms and audit trails. BoKS functions in UNIX and DOS/Windows environments, offers high reliability and is ported to most UNIX platforms. BoKS can also be integrated with enterprise management systems such as Tivoli and database applications such as Oracle and Sybase.

BoKS can use Secure Dynamics SecurID smart tokens. Although the author has little practical experience with BoKS, it seems to be in extensive use where high security is required. Runs on UNIX (SunOS, Solaris and HP-UX) and PCs (Win95 & NT versions should be introduced in late 1996). BoKS uses shared key encryption (40 bit DES outside the U.S., 56bit DES in the U.S.).

OPIE (One-time Passwords In Everything)

OPIE is a public domain release of the U.S. Naval Research Laboratory's. OPIE is an improved version of S/Key Version 1 which runs on POSIX compliant UNIX like systems and has the following additional features to S/Key:

- Simpler (one command installation)
- An OPIE compliant ftp daemon and su, login and passwd utilities are provided.
- MD4 & MD5 are simultaneously supported with MD5 being the default.
- Runs well on Solaris.
- OPIE calculators are available on PCs and MACs too.

ACE Server (SecurID)

The SecurID system from Secure Dynamics is one of the more established names on the market today. It works with most clients (UNIX, NT, VPN clients, terminal servers etc.) and many firewalls provide support for SecurID. The server which manages the user database and allows/refuse access is called ACE and delivered only by Secure Dynamics (whereas clients are delivered by several vendors). The author has used this system for providing secure remote access to hundreds of users on diverse clients.

The tokens are known are SecurID and are basically credit card sized microcomputer, which generate a unique password every minute. In addition each user is attributed a 4 character pin-code (to protect against stolen cards). When a user logs on, he enters his PIN, plus the current pass-code displayed by the SecurID token. The server contains the same algorithm and secret encryption key, allowing both sides to authenticate securely. Software tokens are available for Win95/NT as are SecurID modems from Motorola. The tokens last typically 3 years.This form of authentication is strong, but there is a risk of a session being hijacked (for example if the one time password doesn't change often).

Safeword

Safeword by Secure Computing is direct competition for ACE/SecurID. It's servers run on UNIX. It supports many authentication protocols such as TACACS, TACACS+ and RADIUS.

Many token types are supported: Watchword, Cryptocard, DES Gold & Silver, Safeword Multi-sync and SofToken, AssureNet Pathways SNK (SecureNet Keys).

Watchword

This one time password system from Racal Guardata that are well established competition to the SecurIDs. It works basically as follows:

The server generates a piece of text. The user (on the client) enters this text (called a challenge) into his Watchword calculator. The calculator displays another text, which the users types in. The server verifies that this text was generated by a permitted Watchword calculator and if so, grants access. Attacks could occur in the form of chosen plaintext guessing. Racal Guardata also produce the Access Gateway.

Defender Security System

This system from AssureNet Pathways may be of interest to those using NT servers, since the server runs on NT (not UNIX like most of the above). Features: Authentication via ARA, NT/RAS, TACACS+. Multiple servers are possible via database replication.

The token used are SecureNet Keys (SNK) hardware or software tokens. The challenge/response authentication uses DES, the PIN is never transmitted over the network and sensitive information is encrypted.

4.2.3.8 Remote Access Control Protocols

RADIUS (Remote Authentication Dial In User Service)

Merit Network and Livingston developed the RADIUS protocol for identification and authentication. There is an IETF working group defining a RADIUS standard.

RADIUS is a vendor independent protocol which should allow multiple dial-in access points to user a centralized user database for authentication. There are however man vendor extensions to the standard and the standard is evolving, meaning that not all implementations are compatible.

RADIUS encrypts password transmitted by a hashing technique using a shared "secret". This secret has to be introduced to both sides by an out of band communication.

XTACACS (Enhanced Terminal Access Controller Access System)

XTACACS is an enhancement on TACACS (Terminal Access Controller Access System), which is a UDP based system from BBN which supports multiple protocols. SLIP/PPP, ARA, Telnet and EXEC protocols are supported.

TACACS+

Also an enhancement on TACACS (from CISCO), but not compatible with XTACACS or TACACS. It allows authentication via S/key, CHPA, PAP in addition to SLIP/PPP and telnet. Authentication and authorisation are separated and may be individually enabled/configured.

TCP is used as opposed to UDP (enhance security). Information transmitted may be encrypted. ACLs and password ageing are supported. Enhanced auditing & billing functions.

PPP Authentication protocols: PAP, CHAP

PAP (password authentication protocol) involves the username and password being sent to a server in clear-text. The password database is stored in a weakly encrypted format. CHAP (Challenge Handshake Authentication Protocol) is a challenge/response exchange with a new key being used at each login. However, the password database is not encrypted. Some vendors offer variations of the PAP and CHAP protocols but with enhancements, for example storing passwords in encrypted form in CHAP.

Access Control Lists (ACLs)

An ACL defines who (or what) can access (e.g. use, read, write, execute, delete or create) an object. Access Control Lists (ACL) are the primary mechanism used to ensure data confidentiality and integrity. A system with discretionary access control can discern between users and manages an ACL for each object. If the ACL can be modified by a user (or data owner), it is considered to be discretionary access control. If the ACL must be specified by the system and cannot be changed by the user, mandatory access control is being used. There is no standardised ACLs for access to OS services and applications in UNIX.

- The AIX (optionally), DCE and Windows NT use ACLs to govern access to most objects.
- Solaris 2.5 and later provides ACLs for filesystems (UFS, NFS).
- Normal UNIX also uses ACLs (sort of), but by a different name. For example: protecting files (/etc/groups), protecting sharing of filesystems (/etc/netgroups, /etc/exports), mounting of filesystems (/etc/fstab), remote access (.rhosts, / etc/hosts.allow), X Windows (xauth, xhost), NIS networks (/etc/securenets), Printers (/etc/hosts.lpd).

4.2.3.9 Availability Mechanisms

Backup & Restore

- Things to watch out for:
- If possible use heterogeneous products which work on all of your servers.
- On-line indices allow quick retrieval. Backups to disk allow quick restoring.
- Some products allow users to backup and restores their own files without administrator intervention. Jukeboxes (also called tape stackers) reduce the physical work of changing cassettes and can make restore time quicker (more cassettes are available). Some systems automatically label tapes.
- Hardware and software compression can reduce backup times, reduce network load and reduce the number of cassettes needed.
- Network backups load the network significantly, it may not be possible, for example to backup 100 4GB file servers each night over the network. Planning is important.

Environment

The computing environment can be protected with Air Conditioning, locked server rooms and UPS (220V protection).

Redundancy

Redundancy increases availability and may be implemented in hardware (RAID), disk drivers or OS (RAID) or at the application/service level (e.g. Replication, transaction monitors, backup domain controllers).

Application/Service Redundancy

- This is often the cheapest and easiest to implement, where available. The principle problem is that few applications support this type of redundancy.

Clients connecting to these servers automatically look for a backup or duplicate server if the primary is not available.

- Naming servers (NIS+, DNS, NIS, WINS, Lan Manager...) often have this capability in-built and it 's use is highly recommended. RAID / mirroring is not necessary for these servers, unless the cost of RAID is cheaper.
- Filesystem servers can increase availability by replicating files to another system or to another local disk regularly. If a major crash of the primary file server occurs, users can mount their files from the second system, but changes made since the last replication/ synchronization will be lost.

RAID / Mirroring

- The classical method of increasing system availability is by duplicating one of the weakest part in a computer: the disk. RAID (Redundant Array of Inexpensive Disks) is a de-facto standard for defining how standard disks can be used to increase redundancy. The top RAID systems duplicate disks, disk controllers, power supplies and communication channels. The simplest RAID systems are software-only disk drivers which group together disparate disks into a redundant set.

There are several RAID levels:

- RAID 1: This is basically mirroring.
- RAID 1+: This is RAID 1 with the addition of parity checking.
- RAID 5: Striping
- RAID 5+: Striping with parity (most commonly used RAID level).
- Things to watch for in RAID systems:
- A black box which is just attached to the scsi port is easier to manage and easier to fix/repair that inbuilt disks & controllers.
- For high availability systems, never buy the latest & hottest. Buy a RAID proven to work for/by others over a period of 6 months/1 year!
- Use a RAID which allows standard disks to be used.
- RAIDs rarely work they way you expect with large databases. Run a trail before buying.
- Use the same RAID for all your servers if possible (learn to use one system well, rather than use many different raids).
- Special device drivers and kernel patches needed for RAID increase difficulty of maintenance and probably downtime.
- Software only RAID, if used, should be small, easy to install/reinstall and have a very good user interface.

System Redundancy

If applications do not provide built in redundancy, special software (and perhaps hardware) can be installed on two systems to provide Hot Standby functionality. The principle is as follows: Both systems can access shared (high availability, dual ported) disks and have duplicate network connections. The backup machine monitors the primary constantly and if it notices that the primary is no longer functioning, it takes control of the shared disks, reconfigures it self to have the same network address as the primary and starts up the applications that were running on the master. Of course this with only work with certain applications e.g. if the primary crashes and it's principal application thrashes it's configuration or data files in doing so, the backup server will not be able to start the application.

Full Hardware Redundancy

Specialised computer systems offer compete redundancy in one system i.e. CPU, memory, disks etc.. are fully duplicated. A single point of failure should not exist. These systems often require specially adapted Operating Systems, cost a fortune and are rarely compatible with mainstream systems. Rarely used in the commercial arena, they are most reserved for military or special financial use.

4.3 LIMITS OF ENCRYPTION

Encryption is a powerful method for protecting data in transit or stored on media that are vulnerable to snooping or seizure. Nevertheless, it has two fundamental limitations. First, it cannot protect data while they are being processed on a computer. This is because data must be in the clear in order to be manipulated. Although it is possible to design encryption systems that allow operations to be performed on ciphertext, such systems will either be weak or have extremely limited functionality. The consequence of processing data in the clear is that if an intruder can gain access to the computer, the intruder may be able to pick up sensitive data as it is being typed in or processed. One way this might be done is with a keyboard sniffer program. These programs record the key strokes that you make on the keyboard and could send them over the Internet to the people who wrote them.

A second limitation of encryption is that it can be no better than the weakest link. Even if the encryption algorithm is excellent, the implementation could be flawed or the key management system weak.

Check Your Progress 1

Notes: a) Write your answer in the space given below.

b) Compare your answer with the one given at the end of this unit.

1) What is the application of cryptography?

....

....

....

....

2) Explain the authentication mechanism.

....

....

....

....

3) What are the limitations of cryptography?

....

....

....

....

4.4 CRYPTOGRAPHIC SIGNATURES (PKI)

Cryptography is the science of securing information. It is most commonly associated with systems that scramble information and then unscramble it. Security experts currently favor the cryptographic signature method known as Public Key Infrastructure (PKI) as the most secure and reliable method of signing contracts online. It is used as a solution for the problem related to integrity, confidentiality and authentication of data. It is necessary to have an identification strategy to ascertain the integrity, confidentiality and authentication channels and processes. It is not merely the question of efficiency but also of reliability.

PKI is about the management and regulation of key pairs by allocating duties between contracting parties laying down the licensing and business norms for CAs and establishing business processes to construct contractual relationships in a digitized world. The idea is to develop a sound public key infrastructure for an efficient allocation and verification of digital signature certificates.

PKI uses an algorithm to encrypt online documents so that they will be accessible only to authorized parties. The parties have "keys" to read and sign the document, thus ensuring that no one else will be able to sign fraudulently. Since the passage of the e-signature law in 2000, the use of PKI technology has become more widely accepted. Many online services offer PKI encrypted digital signature systems that function much like we use PINs for our bank cards.

One of the best known public key encryption methods is RSA. The two keys are formed of pairs of integers: ks and n for the secret key and kp and n for the public key. The key pair kp and n is made public. A document is encryption by breaking its digital form into blocks, each of which is treated as a single number, raising each number to the power of ks or kp (deeping on whether the secret or public key is being used) and then calculating the result modulus n. The document is decrypted using the same alogorithm with the other key pair.

Thus:

$(\text{plaintext})^{ks} \mod n = \text{ciphertext}$

$(\text{ciphertext})^{kp} \mod n = \text{plaintext}$

Te effective security of the RSA algorithm depends on mathematical proof of the fact that, because of the way kp , ks and n are derived, it is computationally infeasible to calculate ks knowing only kp and n.

The RSA algorithm was orignally devised to allow encrypted message to be send to the holder to the secret key, which only he would be able to decipher. However, because the algorithm is symmetrical it is also possible encrypt a document using the sender's secret key ks and decrypt it with the public key kp . This is the method used to effect a digital signature.

In practice, encrypting an entire document using RSA is computationally expensive and so a single key encryption system such as DES or IDEA is used to ensure that transmission of the document remains private, while RSA s used to make a digital signature by encrypting a smaller file which derives from the original document. Many encryption products which can be used to create digital signatures are now available, the best known of which is probably PGP.

Biometric Recording

A method of signing an electronic document which, at first sight, appears very different from the encryption techniques described above, uses a pen attached to a digitizing pad to record the physical signature of the maker of the document. This signature is normally displayed in a window on the screen of the computer to which the digitizing pad is connected and looks just like a traditional hard copy signature. However, the way in which this signature is attached to the document is very different from a physical signature.

The data captured by the digitizing pad is not merely the appearance of the signature but, more importantly, its biometric characteristics. These are, primarily, the speed and acceleration rates of the strokes used to make the signature and the occasions on which the pen is lifted from the digitizing pad, together with the time taken to make each pen stroke. This biometric data is recorded, 63 and can be checked against the signatory's known biometric signature data, either in the possession of the recipient closely, the extremely low probability that some other person could have created the same signature data can be given in evidence to prove the identity of the signatory.

Additionally, it is necessary to attach the signature to the document. This is achieved by deriving a numerical identifier from the document, using a function such that the identifier is very unlikely to match any other document and encrypting that number together with the biometric signature data. The evidential value of this process can be assessed by calculating.

- The probability that some person other than the alleged signatory could have created the biometric signature data and some document other than that alleged to be signed could have produced the same numerical identifier; and
- The probability that the encryption algorithm could have been 'cracked', thus allowing a genuine set of biometric data from one document to be linked to a numerical identifier from another.

Signature metrics are not only form of biometric data which can be used to effect a signature. Other data such as fingerprints or retina prints can be collected in digital form and attached to the document in identical ways.

4.5 ELECTRONIC SIGNATURES

In India, the IT Act, 2000 has led to the development of digital signature regime. Digital signature establishes the principle that in an electronic environment, the basic legal functions of a signature are performed by way of a method that identifies the signer of an electronic message and also confirms that the said signer approved the content of that electronic message. Hence, any attempt to change the content of the record must be seen to be incompatible with the signature.

Digital signatures are the electronic equivalent of the handwritten signatures. In an electronic message or transaction affixing handwritten signature is not possible. Authentication of the record has to be achieved by some electronic or digital method. Affixing digital signature has been defined in sec 2(1)(d) of the Act to mean adoption of any methodology or procedure by a person for the purpose of authenticating an electronic record by means of digital signature.

The expression 'Digital signature' has been defined in sec 2(1)(p) of the Act as Authentication of any electronic record by a subscriber by means of an electronic method or procedure in accordance with the provisions of sec 3. Here subscriber is a person in whose name the Digital signature certificate is issued. It is not a digitized image of a handwritten signature. It is a block of data at the end of an electronic message that attests to the authenticity of the said message. Digital signature is recognized as a valid method of authentication or as an authentication standard.

Any subscriber may authenticate an electronic record by affixing his digital signature. The authentication of the electronic record shall be effected by the use of 'asymmetric crypto system' and 'hash function' which envelop and transform the initial electronic record into another electronic record. Any person by the use of a public key of the subscriber can verify the electronic record. The private key and the public key are unique to the subscriber and constitute a functioning key pair.

4.6 ELECTRONIC SIGNATURE TECHNOLOGY

An electronic signature is produced by performing a mathematical function on the document or part of it, which identifies the signatory and authenticates the contents of the document. To be an effective signature, the modified document must be producible only by the maker and any change to the content of the document must invalidate the signature. These modifications can be achieved through the use of encryption technology.

Because an electronic dɔcument is a string of 1sand 0s it can be treated as a series of numbers. Encryption is carried out by performing a series of mathematical functions (an encryption algorithm) which has two inputs; the series of numbers which represents the document (the plaintext) and a key, which is itself a number. The result is a series of different numbers, the ciphertext. There are two distinct types of encryption algorithm:

Single key encryption uses the same key to encrypt and decrypt and thus the key needs to be known to both the sender and the recipient of a document. Public key encryption uses tow different keys, each of which will decrypt document encrypted by the other key. This means that one key can be kept secret, while the other is made public. All effective electronic signature techniques require the use of a 'one-way function'. This means that if a document or its signature element, but must not be able to re-encrypt it with A's key.

All encryption can be broken given sufficient time and computing resources. The effectiveness of encryption as a method of signing electronic documents relies on the fact that it is computationally infeasible to break the encryption method and thus become able to forge the signature, within a reasonable period of time.

4.6.1. Single Key Encryption

The most commonly used single key encryption system is the Data Encryption Standard (DES). DES is a complicated form of encryption which is normally effected in hardware, but in essence it requires a key (56- bit in DES,I, but normally much longer for current uses) which is common to sender and recipient and kept secret from all others. This key is used to scramble the document to such a degree that is computationally infeasible to unscramble it without knowing the key. The fact that a document is DES encrypted is therefore extremely strong evidence that it could have emanated only from one or other of the key holders. This, however, does not authenticate it fully as both parties have the key. Either could alter the contents of the document and then re-encrypt it. The alteration would be undetectable and the court would still be left with two documents, each claimed to be authentic. However techniques have been invented which enable one-way functions, encryption which can only have been performed by on of the parties, to be performed using DES and thus to create a digital signature of the electronic documents.

4.6.2 Process of Creating and Verifying a Digital Signature

Basically, a digital signature is a two way process, involving two parties:

a) Signer - creator of the digital signature

b) Recipient - verifier of the digital signature

A digital signature is complete, if and only if, the recipient successfully verifies it.

Creating a digital signature includes the following steps:

1) Signer demarcates what is to be signed. This delimited information to be signed is termed as the message.

2) A hash function in the signer's software computes a hash result unique to a message.

3) The signer's software then transforms the hash result into a digital signature using the signer's private key.

4) The digital signature is attached to the message and stored or transmitted with its message. The signer sends both digital signature and the message to the recipient.

Verifying a digital signature includes the following steps:

1) Recipient receives digital signature and the message.

2) Recipient applies signer's public key on the digital signature.

3) Recipient recovers the hash result from the digital signature.

4) Computes the new hash result of the original message be means of the same hash function used by the signer to create the digital signature.

5) Compares the hash results so recovered.

If the hash result computed by the verifier is identical to the hash result extracted from the digital signature during the message remained unaltered. If they are not equal, it would mean that the message either originated elsewhere or was altered after it was signed and the recipient can reject the message.

4.6.3 Need/use of Verification

1) To verify whether the signer's private key was used to digitally sign the message.

2) Whether the newly computed hash result matches the original hash result which was recovered from the digital signature during the verification process.

The processes of creating and verifying a digital signature provide a high level of assurance that the digital signature is genuinely the signer's. These processes grant legal sanctity to digital signatures.

4.7 HOW ELECTRONIC SIGNATURE TECHNOLOGY WORKS?

Digital signatures are created and verified by cryptography, the branch of applied mathematics that concerns itself with transforming message into seemingly unintelligible forms and back again. Digital signature use what is known as 'public key cryptography', which employs an algorithm using two different but mathematically related 'key'; one for creating a digital signature or transforming data into a seemingly unintelligible form and another key for verify a digital signature or returning the message to its original form. Computer equipment and software utilizing two such keys are often collectively termed and 'asymmetric cryptosystem'.

The complementary keys of an asymmetric cryptosystem for digital signatures are arbitrarily termed the private key, which is known only to the signer and used to create the digital signature and the public key, which is ordinarily more widely know and is used by a relying party to verify the digital signature. If many people need to verify the signer's digital signatures, the public key must be available or distributed to all of them, perhaps by publication in an on-line repository or directory where it is easily accessible. Although the key of the pair are mathematically related, if the asymmetric cryptosystem has been designed and implemented securely it is 'computationally infeasible' to derive the private key from knowledge of the public key. Thus, although many people may know the public key of a given signer and use it to verify that signer's signature, they cannot discover that signer's private key and use it to forge digital signatures. This is sometimes referred to as the principle of 'irreversibility'.

Another fundamental process, termed a 'hash function', is used in both creating and verifying a digital signature. A hash function is an algorithm which creates a digital representation or 'fingerprint' in the form of a 'hash value' or 'hash result' of a standard length which is usually much smaller than the message but nevertheless substantially unique to it. Any change to the message invariably produces a different hash result when the same hash function value. Hash functions therefore enable the software for creating digital signatures to operate on smaller and predictable amounts of data, while still providing robust evidentiary correlation to the original message content, thereby efficiently providing assurance that there has been no modification of the message since it was digitally signed.

Thus, use of digital signatures usually involves two processes, one performed by the signer and the other by the receiver of the digital signature.

- Digital signature creation uses a hash result derived from and unique to both the signed message and a given private key. For the hash result to be secure, there must be only a negligible possibility that the same digital signature could be created by the combination of any other message or private key.
- Digital signature verification is the process of checking the digital signature by reference to the original message and a given public key, thereby determining whether the digital signature was created for that same message using the private key that corresponds to the referenced public key.

Verification of a digital is accomplished by computing new hash result of the original message by means of the same hash function used to create the digital signature. Then, using the public key and the new hash result, the verifier checks: (1) whether the digital signature was created using the corresponding private key; and (2) whether the newly computed hash result matches the original hash result which was transformed into the digital signature during the signing process. The verification software will confirm the digital signature as 'verified' if : (1) the signer's private key was used to digitally sign the message, which is known to be the case if the signer's public key was used to verify the signature because the signer's public key will verify only a digital signature created with the signer's private key: and (2) the message was unaltered, which is known to be the case if the hash result computed by the verifier is identical to the hash result extracted from the digital signatures during the verification process.

Various asymmetric cryptosystems create and verify digital signatures using different algorithms and procedures, but share this overall operational pattern.

The process of creating a digital signature and verifying it accomplish the essential effects desired of a signature for many legal purposes:

- **Signer authentication:** If a public and private key pair is associated with an identified signer, the digital signature attributes the message to the signer. The digital signature cannot be forged, unless the signer loses control of the private key (a 'compromise' of the private key'), such as by divulging it or losing the media or device in which it is contained.

- **Message authentication:** The digital signature also identifies the signed message, typically with far greater certainty and precision than paper signatures. Verification reveals any tampering, since the comparison of the hash results (one made at signing and the other made at certifying) shows whether the message is the same as when signed.

- **Affirmative act:** Creating a digital signature requires the signer to use the signer's private key. This act can perform the 'ceremonial' function of alerting the signer to the fact that the signer is consummating a transaction with legal consequences.

- **Efficiency:** The process of creating and verifying a digital signature provide a high level of assurance that the digital signature is genuinely the signer's. As with the case of modern electronic data interchange ('EDI') the creation and verification processes are capable of complete automation (sometimes referred to as 'machinable'), with human interaction required on an exception basis only. Compared to paper methods such as checking specimen signature cards– methods so tedious and labor - intensive that they are rarely actually used in practice – digital signature yield a high – degree of assurance without adding greatly to the resources required for processing.

4.8 USE OF ELECTRONIC SIGNATURES

Use of electronic signatures are the followings:

1) It helps in making electronic communications or transactions legally binding.

2) It shows authenticity of the sender to enable the recipient to determine who really sent the message.

3) It enables the recipient to determine whether or not the message received has been modified en route or is incomplete.

4) It has the ability to ensure that the sender cannot falsely deny sending the message nor falsely deny the contents of the message. It prevents a person from unilaterally modifying or terminating legal obligation arising out of a transaction effected by computer based means.

5) It keeps the message or information confidential.

6) It is capable of fulfilling the demand of burgeoning e-commerce by not only providing message authentication, integrity and non-repudiation function but also making it highly scalable.

7) It helps to facilitates electronic governance such as it helps in the following activities- A. the filing of any form, application or any other document with any office, authority, body or agency owned or controlled by the appropriate government in a particular manner. B. issue or grant of any license, permit, sanction or approval C. The receipt or payment of money in a particular manner.

The goal of using digital signatures is to protect the message from any unauthorised alteration, modification, deletion, interception or transmission. A digital signature helps in determining whether the initial electronic record is retained intact or has been altered since such electronic record was so affixed with the digital signatures.

4.9 ENCRYPTION AND STEGANOGRAPHY

Encryption can present a significant challenge for digital forensic practitioners, particularly full disk encryption. Even when full disk encryption is not used or can be circumvented, additional effort is required to salvage data from password protected or encrypted files. When dealing with individually protected files, it is sometimes possible to use a hexadecimal editor like WinHex to simply remove the password within a file. There are also specialized tools that can bypass or recover passwords of various files.

Currently, the most powerful and versatile tools for salvaging password protected and encrypted data are PRTK and DNA from AccessData. The Password Recovery Toolkit can recover passwords from many file types and is useful for dealing with encrypted data. Also, it is possible for a DNA network to try every key in less time by combining the power of several computers. Distributed Network Attack (DNA) can brute-force 40-bit encryption of certain file types including Adobe Acrobat and Microsoft Word and Excel. Using a cluster of approximately 100 off-the-shelf desktop computers and the necessary software, it is possible to try every possible 40-bit key in five days. Rainbow tables can be used to accelerate the password guessing process. Some vendors also have hardware decryption platforms based on implementation of field programmable gate arrays that can increase the speed of brute force attacks.

When strong encryption is used such as BestCrypt, PGP or Windows Encrypting File System, a brute-force approach to guessing the encryption key is generally infeasible. In such cases, it may be possible to locate unencrypted versions of data in unallocated space, swap files and other areas of the system. For instance, printer spool files on Windows and UNIX systems can contain data from files that have been deleted or encrypted. Alternatively, it may be possible to obtain an alternate decryption key. For instance some encryption programs advise users to create a recovery disk in case they forget their password. When EFS is used, Windows automatically assigns an encryption recovery agent that can decrypt messages when the original encryption key is unavailable. In Windows 2000, the built-in administrator account is the default recovery agent (an organization can override the default by assigning a domain-wide recovery agent provided the system is part of the organization's Windows 2000 domain).

Notably, prior to Windows XP, EFS private keys were weakly protected and it was possible to gain access to encrypted data by replacing the associated NT logon password with a known value using a tool like ntpasswd and logging into a bootable/ virtualized clone of the system with the new password.

When investigating a child exploitation case, it is advisable to be on the lookout for other forms of data concealment such as steganography. Forensic analysts can make educated guesses to identify files containing hidden data-the presence of steganography software and uncharacteristically large files should motivate examiners to treat these as special files that require additional processing. In such cases, it may be possible to salvage the hidden data by opening the files using the steganography software and providing a password that was obtained during the investigation. More sophisticated techniques are available for detecting hidden data. Even if encryption or steganography cannot be bypassed, documenting which files are concealing data can help an investigator determine the intent of the defendant.

Encryption often presents challenges for analysts. Users might encrypt individual files, folders, volumes or partitions so that others cannot access their contents without a decryption key or passphrase. The encryption might be performed by the OS or a third-party program. Although it is relatively easy to identify an encrypted file, it is usually not so easy to decrypt it. The analyst might be able to identify the encryption method by examining the file header, identifying encryption programs installed on the system or finding encryption keys (which are often stored on other media). Once the encryption method is known, the analyst can better determine the feasibility of decrypting the file. In many cases, it is not possible to decrypt files because the encryption method is strong and the authentication (e.g., passphrase) used to perform decryption is unavailable.

Although an analyst can detect the presence of encrypted data rather easily, the use of steganography is more difficult to detect. Steganography, also known as steg, is the embedding of data within other data. Digital watermarks and the hiding of words and information within images are examples of steganography. Some techniques an analyst can use to locate stegged data include looking for multiple versions of the same image, identifying the presence of grayscale images, searching metadata and registries, using histograms and using hash sets to search for known steganography software. Once certain that stegged data exists, analysts might be able to extract the embedded data by determining what software created the data and then finding the stego key or by using brute force and cryptographic attacks to determine a password. However, such efforts are often unsuccessful and can be extremely time-consuming, particularly if the analyst does not find the presence of known steganography software on the media being reviewed. In addition, some software programs can analyze files and estimate the probability that the files were altered with steganography.

Check Your Progress 2

Notes: a) Write your answer in the space given below.

b) Compare your answer with the one given at the end of this unit.

1) Explain public key infrastructure (PKI).

....

....

....

....

....

2) Explain the use and working of electronic signatures.

....

....

....

....

....

3) What is steganography?

....

....

....

....

....

4.10 LET US SUM UP

Because of the need to ensure that only those eyes intended to view sensitive information can ever see this information and to ensure that the information arrives unaltered, security systems have often been employed in computer systems for governments, corporations and even individuals. Encryption schemes can be broken, but making them as hard as possible to break is the job of a good cipher designer. All you can really do is make it very difficult for the code breaker to decipher your cipher. Still, as long as both secure and encrypted data are available, it will always be possible to break your code. It just won't necessarily be easy.

Encryption is often portrayed as the silver bullet for information security. Unfortunately, it is a far cry from that. It does not protect against the majority of attacks, including insider abuse, social engineering, break-ins that exploit system vulnerabilities such as buffer overflows, data tampering, Trojan horses, viruses, web scams and so on. It has a great role in working of electronic signatures. It is but one element of a defensive information warfare program.

4.11 CHECK YOUR PROGRESS: THE KEY

Check Your Progress 1

1) Cryptography can be used to protect stored data, including complete files and objects within files or communications, including phone calls, fax transmissions, email, web- transactions, banking transactions, corporate extranets and other types of network applications. Some encryption systems will encrypt everything on a hard disk so that the computer is effectively unusable without knowing the key. Encrypting files and complete disks is particularly useful with laptop computers. If the computer is stolen, sensitive data with not be exposed. Encryption is available in both hardware and software implementations. They can be obtained as stand alone encryption devices and software packages or as a feature of other products. Many software applications and utility programs support encryption including spreadsheets, software for word processing, file management, databases, web browsing, e-mail and internet telephony. Increasingly, computer software comes with built-in encryption.

2) Authentication is the process of verifying the identity of a subject. A subject (also called a principal) can be a user, a machine or a process i.e. a "network entity". Authentication uses something which is known to both sides, but not to others i.e. something the subject is, has or knows . Hence this can be biometrics (fingerprints, retina patterns, hand shape/size, DNA patterns, handwriting, etc.), passphrases, passwords, one-time password lists, identity cards, smart-tokens, challenge-response lists etc. Some systems consist of a combination of the above.

   The most common methods of strong authentication today consist of one-time password lists (paper), automatic password generators (smart tokens) and intelligent identity cards.

3) Encryption is a powerful method for protecting data in transit or stored on media that are vulnerable to snooping or seizure. Nevertheless, it has two fundamental limitations. First, it cannot protect data while they are being processed on a computer. This is because data must be in the clear in order to be manipulated. Although it is possible to design encryption systems that allow operations to be performed on ciphertext, such systems will either be weak or have extremely limited functionality. The consequence of processing data in

the clear is that if an intruder can gain access to the computer, the intruder may be able to pick up sensitive data as it is being typed in or processed. One way this might be done is with a keyboard sniffer program. These programs record the key strokes that you make on the keyboard and could send them over the Internet to the people who wrote them.

Check Your Progress 2

1) Cryptography is the science of securing information. It is most commonly associated with systems that scramble information and then unscramble it. Security experts currently favor the cryptographic signature method known as Public Key Infrastructure (PKI) as the most secure and reliable method of signing contracts online. It is used as a solution for the problem related to integrity, confidentiality and authentication of data. It is necessary to have an identification strategy to ascertain the integrity, confidentiality and authentication channels and processes. It is not merely the question of efficiency but also of reliability. PKI is about the management and regulation of key pairs by allocating duties between contracting parties laying down the licensing and business norms for CAs and establishing business processes to construct contractual relationships in a digitized world. The idea is to develop a sound public key infrastructure for an efficient allocation and verification of digital signature certificates.

2) An electronic signature is produced by performing a mathematical function on the document or part of it, which identifies the signatory and authenticates the contents of the document. To be an effective signature, the modified document must be producible only by the maker and any change to the content of the document must invalidate the signature. These modifications can be achieved through the use of encryption technology. Because an electronic document is a string of 1sand 0s it can be treated as a series of numbers. Encryption is carried out by performing a series of mathematical functions (an encryption algorithm) which has two inputs; the series of numbers which represents the document (the plaintext) and a key, which is itself a number. The result is a series of different numbers, the ciphertext. There are two distinct types of encryption algorithm:

   Single key encryption uses the same key to encrypt and decrypt and thus the key needs to be known to both the sender and the recipient of a document. Public key encryption uses tow different keys, each of which will decrypt document encrypted by the other key. This means that one key can be kept secret, while the other is made public. All effective electronic signature techniques require the use of a 'one-way function'. This means that if a document or its signature element, but must not be able to re-encrypt it with A's key. All encryption can be broken given sufficient time and computing resources. The effectiveness of encryption as a method of signing electronic documents relies on the fact that it is computationally infeasible to break the encryption method and thus become able to forge the signature, within a reasonable period of time.

3) Steganography, also known as steg, is the embedding of data within other data. Digital watermarks and the hiding of words and information within images are examples of steganography. Some techniques an analyst can use to locate stegged data include looking for multiple versions of the same image, identifying the presence of grayscale images, searching metadata and registries, using histograms and using hash sets to search for known steganography software. Once certain that stegged data exists, analysts might be able to extract the embedded data by determining what software created the data and then finding the stego key or by using brute force and cryptographic attacks to

determine a password. However, such efforts are often unsuccessful and can be extremely time-consuming, particularly if the analyst does not find the presence of known steganography software on the media being reviewed. In addition, some software programs can analyze files and estimate the probability that the files were altered with steganography.

4.12 SUGGESTED READINGS

- Information Technology – Law and Practice by Vakul Sharma
- Law relating to Computers, Internet and E-commerce – A guide to cyber laws by Nandan Kamath
- www.crypto.wpi.edu
- www.ncbi.nlm.nih.gov
- www.rsa.com

Student Satisfaction Survey

ignou THE PEOPLE'S UNIVERSITY

Student Satisfaction Survey of IGNOU Students

Enrollment No.	
Mobile No.	
Name	
Programme of Study	
Year of Enrolment	
Age Group	□ Below 30 □ 31-40 □ 41-50 □ 51 and above
Gender	□ Male □ Female
Regional Centre	
States	
Study Center Code	

Please indicate how much you are satisfied or dissatisfied with the following statements

Sl. No.	Questions	Very Satisfied	Satisfied	Average	Dissati-sfied	Very Dissati-sfied
1.	Concepts are clearly explained in the printed learning material	□	□	□	□	□
2.	The learning materials were received in time	□	□	□	□	□
3.	Supplementary study materials (like video/audio) available	□	□	□	□	□
4.	Academic counselors explain the concepts clearly	□	□	□	□	□
5.	The counseling sessions were interactive	□	□	□	□	□
6.	Changes in the counseling schedule were communicated to you on time	□	□	□	□	□
7.	Examination procedures were clearly given to you	□	□	□	□	□
8.	Personnel in the study centers are helpful	□	□	□	□	□
9.	Academic counseling sessions are well organized	□	□	□	□	□
10.	Studying the programme/course provide the knowledge of the subject	□	□	□	□	□
11.	Assignments are returned in time	□	□	□	□	□
12.	Feedbacks on the assignments helped in clarifying the concepts	□	□	□	□	□
13.	Project proposals are clearly marked and discussed	□	□	□	□	□
14.	Results and grade card of the examination were provided on time	□	□	□	□	□
15.	Overall, I am satisfied with the programme	□	□	□	□	□
16.	Guidance from the programme coordinator and teachers from the school	□	□	□	□	□

After filling this questionnaire send it to:
Programme Coordinator, School of Vocational Education and Training,
Room no. 19, Block no. 1, IGNOU, Maidangarhi, New Delhi- 110068

MPDD-IGNOU/P.O.2T/August,2011

ISBN-978-81-266-5514-1