Network Security Using Cryptography

P.Pavan Kumar
IInd year B.E.
Computer Science & Engineering Department,
Saveetha University

Abstract— Network security is a hardest subject, In past it is only handled by well-trained exports. People should understand the basics of security in network world. Now we are going to see the what are the common attacks take place security against over network , what is the solution for those attacks, introduction to cryptography, algorithms of cryptography and some features on secure network. Any way we are going to present the best ciphers currently in use.

Key word: Cryptography, Network security, Encryption, decryption.

I. INTRODUCTION

An understanding of computer network is first of all every one the principles of network security

A. What is a Network?

A "network" has been defined as "any set of interlinking lines resembling a net, *a* network of roads, an interconnected system, *a* network of alliances". This definition suits well for network:

A Computer Network is simply an interconnected collection of autonomous systems.

B. What is network security?

Network Security is one which making sure that nosy people could not either access or alter the information intended for the recipient.

II. COMMON ATTACKS AGAINST SECURITY OVER NETWORK

- Tapping the wire: To get access to clear text data and passwords
- Impersonation: To get unauthorized access to data or to create unauthorized e-mails, orders, etc.
- Denial-of-service: To render network resources non-functional.
- Replay of messages: To get access to information and change it in transit.
- Guessing of Keys Passwords: To get access to encrypted data and passwords •Virus: To destroy data. *(Viral Information Resource Under Seized).*

Solutions For Attacks Against Security Over Networks

- Encryption: To protect data and passwords.
- Authentication: By using digital signatures and certificates this will do verify who is sending data over the network.
- Authorization: To prevent improper access of data over the network.
- Integrity checking: To protect against improper alteration of messages.
- Non-repudiation: To make sure that an action cannot be denied by the person who performed it.

A. Concept Of Cryptography

To keeping your data and communications secure, techniques such as encryption, decryption and authentication are used. The key factor to strong cryptography is the difficulty of reverse engineering.

Strong cryptography means that the computational effort needed to retrieve your clear text messages without knowing the proper keys makes the retrieval infeasible.

ENCRYPTION AND DECRYPTION - CRYPTOGRAPHIC ALGORITHMS

Encryption is the transformation of a clear text message into an unreadable
form in order to hide its meaning. The opposite transformation, which retrieves the original clear text, is the decryption. The mathematical function used for encryption and decryption is the cryptographic algorithm or cipher.

There are many drawbacks to restricted ciphers. It is very difficult to keep an algorithm a secret when many people use it.

For these reasons, the currently used algorithms are keyed, that is, the encryption and decryption makes use of a parameter, known as the key.

The key can be chosen from a set of possible values, called the keyspace. The keyspace usually is huge, the bigger the better.

III. IMPORTANCE OF CRYPTOGRAPHY

Encryption provides confidentiality to messages. When communicating over an un-trusted medium, such as the Internet, you may also need, in addition to

- Confidentiality- Protection of information disclosure by means of data encryption to those who are not intended to receive it.
- Authentication- A method for verifying that the sender of a message is really who he or she claims to be. Any intruder masquerading as someone else is detected by authentication.
- Integrity checking - A method for verifying that a message has not been altered along the communication path. Any tampered message sent by an intruder is detected by an integrity check.
- Non-repudiation–The possibility to prove that the sender has really sent the message.

IV. SYMMETRIC OR SECRET-KEY ALGORITHMS

Symmetric algorithms are keyed algorithms where the decryption key is the same as the encryption key. These are conventional cryptographic algorithms where the sender and the receiver must agree on the key before any secured communication can take place between them.

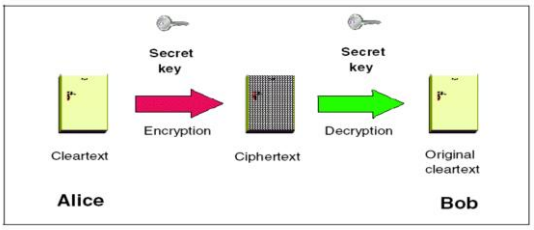

Figure 258. Keyed encryption and decryption

Fig. 1: Encryption and Decryption

There are two types of symmetric algorithms:

1. Block ciphers: A cryptosystem in which encryption/decryption is done on blocks of data. The full message is divided into fixed length blocks, then each block is encrypted/decrypted and the blocks are grouped to get the plaintext/ciphertext.
2. Stream ciphers: An encryption method that uses continuous input, as opposed to fixed length blocks of data.

The algorithms used in Block Ciphers:

Secret key block ciphers	Data Block Size (bits)	Crypto key size bits
Data Encryption Standards (DES)	64	56
International data Encryption algorithm (IDEA)	64	128
Modular multiplication Block cipher (MMB)	128	128
Cellular automata cipher	384	1088
SKIPJACK	64	80

The most significant use of IDEA is in the freeware secure e-mail package Pretty Good Privacy (PGP).

An example of a stream algorithm is A5, The advantage of the symmetric algorithms is their efficiency.

They can be easily implemented in hardware. A major disadvantage is the difficulty of key management. A secure way of exchanging the keys must exist, which is often very hard to implement.

V. Asymmetric Or Public-Key Algorithms

These algorithms address the major drawback of symmetric ciphers, the requirement of the secure key-exchange channel. The idea is that two different keys should be used:

Public key which, as the name implies, is known to everyone, and

Private key, which is to be kept in tight security by the owner.

The private key cannot be determined from the public key.

A clear text encrypted with the public key can only be decrypted with the corresponding private key.

A clear text encrypted with the private key can only be decrypted with the corresponding public key.

Thus, if someone sends a message encrypted with the recipient's public key, it can be read by the intended recipient only. The process is shown in figure where Alice sends an encrypted message to Bob.

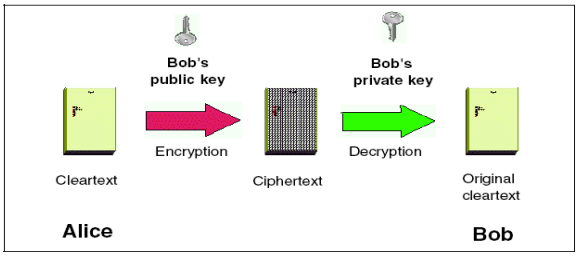

Figure 259. Encryption using the recipient's public key

Fig. 2:

As the public key is available to anyone, privacy is assured without the need for a secure key-exchange channel. Parties who wish to communicate retrieve each other's public key.

A. Authentication and non-repudiation

1) using Digital Signatures

An interesting property of the public-key algorithms is that they can provide authentication. The private key is used for encryption. Since anyone has access to the corresponding public key and can decrypt the message, ***This provides no privacy***. However, it authenticates the message. If one can successfully decrypt it with the claimed sender's public key, then the message has been encrypted with the corresponding private key, which is known by the real sender only. Thus, the sender's identity is verified. Encryption with the private key is used in **Digital Signatures.** The principle is shown in figure. Alice encrypts her message with her private key ("signs" it), in order to enable Bob to verify the authenticity of the message.

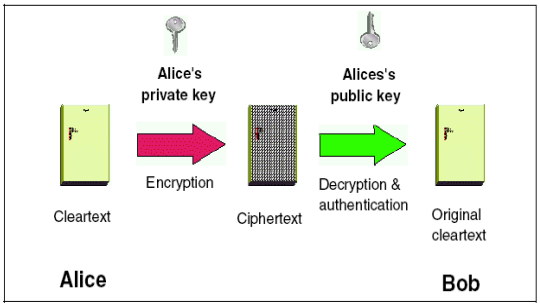

Figure 260. Authentication by encrypting with a private key

Fig. 3:

Going a step further, encrypting with the private key gives non-repudiation too. Additionally, if a timestamp is included, then the exact date and time can also be proven. There are protocols involving trusted third parties that prevent the sender from using phony timestamps.

B. Hash Functions

Hash functions (also called message digests) are fundamental to cryptography.

A hash function is a function that takes variable-length input data and produces fixed length output data (the hash value), which can be regarded as the "fingerprint" of the input. That is, if the hashes of two messages match, it is highly probable that the messages are the same.

Cryptographically useful hash functions must be *one-way*, which means that they should be easy to compute, but infeasible to reverse. An everyday example of a one-way function is mashing a potato; it is easy to do, but once mashed, reconstructing the original potato is rather difficult.

A good hash function should also be *collision-resistant*. It should be hard to find two different inputs that

hash to the same value. As any hash function maps an input set to a smaller output set, theoretically it is possible to find collisions. The point is to provide a unique digital "fingerprint" of the message, that identifies it with high confidence, much like a real fingerprint identifying a person.

A hash function that takes a key as a second input parameter and its output depends on both the message and the key is called a Message Authentication Code *(MAC)*, as shown in figure

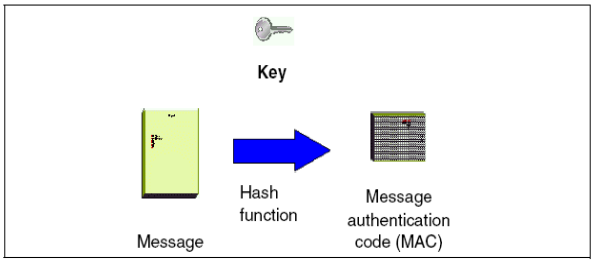

Figure 261. Generating a message authentication code (MAC)

Fig. 4:

Put simply, if you encrypt a hash, it becomes a MAC. If you add a secret key to a message, then hash the concatenation, the result is a MAC. Both symmetric and asymmetric algorithms can be used to generate MACs.

1) Hash functions are primarily used to assure integrity and authentication :

- The sender calculates the hash of the message and appends it to the message.
- The recipient calculates the hash of the received message and then compares the result with the transmitted hash.
- If the hashes match, the message was not tampered with.
- If the encryption key (symmetric or asymmetric) is only known by a trusted sender, a successful MAC decryption indicates that the claimed and actual senders are identical. The Message* and MAC* notations reflect the fact that the message might have been altered while crossing the untrusted channel.

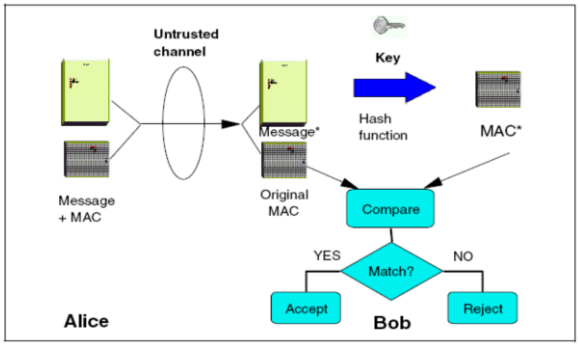

Figure 262. Checking integrity and authenticity with MAC

Fig. 5:

One could argue that the same result can be obtained with any kind of encryption, because if an intruder modifies an encrypted message, the decryption will result in nonsense, thus tampering can be detected. The answer is that many times only integrity and/or authentication is needed, maybe with encryption on some of the fields of the message. Also encryption

is very processor-intensive. Examples include the personal banking machine networks, where only the PIN's are encrypted, however MAC's are widely

used. Encrypting all the messages in their entirety would not yield noticeable benefits and performance would dramatically decrease. The encryption of a hash with the private key is called a Digital Signature. The encryption of a secret key with a public key is called a digital envelope. This is a common technique used to distribute secret keys for symmetric algorithms.

2) Random-Number Generators

An important component of a cryptosystem is the random-number generator. Many times random session keys and random initialization variables (often referred to as initialization vectors) are generated.

The quality, that is the randomness of these generators, is more important than you might think. The ordinary random function provided with most programming language libraries is good enough for games, but not for cryptography. Those random-number generators are rather predictable; if you rely on them, be prepared for happy cryptanalysts finding interesting correlations in your encrypted output. The fundamental problem faced by the random-number generators is that the computers are ultimately deterministic machines, so real random sequences cannot be produced.

Cryptographically strong pseudorandom generators must be unpredictable. It must be computationally infeasible to determine the next random bit, even with total knowledge of the generator. A common practical solution for pseudorandom generators is to use hash functions. This approach provides sufficient randomness and it can be efficiently implemented. Military-grade generators use specialized devices that exploit the inherent randomness in physical phenomena. An interesting solution can be found in the PGP software. The initial seed of the pseudorandom generator is derived from measuring the time elapsed between the keystrokes of the user.

VI. Secure Network Devices Secure Modems:

A. Walk-up Network Connections

By "walk-up" connections, we mean network connection points located to provide a convenient way for users to connect a portable host to your network.

Consider whether you need to provide this service, bearing in mind that it allows any user to attach an unauthorized host to your network. This increases the risk of attacks via techniques such as IP address spoofing, packet sniffing, etc. Users and site management must appreciate the risks involved. If you decide to provide walk-up connections, plan the service carefully and define precisely where you will provide it so that you can ensure the necessary physical access security.

A walk-up host should be authenticated before its user is permitted to access resources on your network. As an alternative, it may be possible to control physical access. For example, if the service is to be used by students, you might only provide walk-up connection sockets in student laboratories. If you are providing walk-up access for visitors to connect back to their home networks (e.g., to read e-mail,

etc.) in your facility, consider using a separate subnet that has no connectivity to the internal network.
Keep an eye on any area that contains unmonitored access to the network, such as vacant offices. It may be sensible to disconnect such areas at the wiring closet, and consider using secure hubs and monitoring attempts to connect unauthorized hosts.

B. Modems:

If modem access is to be provided, this should be guarded carefully. The *terminal server* , or network device that provides dial-up access to your network needs to be actively administered, and its logs need to be examined for strange behavior. Its password need to be strong -- not ones that can be guessed. Accounts that aren't actively used should be disabled. In short, it's the easiest way to get into your network from remote: guard it carefully.

1. Modem Lines Must Be Managed
2. Dial-in Users Must Be Authenticated
3. Call-back Capability
4. All Logins Should Be Logged
5. Choose Your Opening Banner Carefully
6. Dial-out Authentication
7. Make Your Modem Programming as "Bullet-proof" as Possible

C. Dial-back systems

There are some remote access systems that have the feature of a two-part procedure to establish a connection. The first part is the remote user dialing into the system, and providing the correct userid and password. The system will then drop the connection, and call the authenticated user back at a known telephone number. Once the remote user's system answers that call, the connection is established, and the user is on the network. This works well for folks working at home, but can be problematic for users wishing to dial in from hotel rooms and such when on business trips.

Other possibilities include one-time password schemes, where the user enters his userid, and is presented with a ``challenge," a string of between six and eight numbers. He types this challenge into a small device that he carries with him that looks like a calculator. He then presses enter, and a ``response" is displayed on the LCD screen. The user types the response, and if all is correct, he login will proceed. These are useful devices for solving the problem of good passwords, without requiring dial-back access. However, these have their own problems, as they require the user to carry them, and they must be tracked, much like building and office keys.

D. Crypto-Capable Routers:

A feature that is being built into some routers is the ability to session encryption between specified routers. Because traffic traveling across the Internet can be seen by people in the middle who have the resources (and time) to snoop around, these are advantageous for providing connectivity between two sites, such that there can be secure routes.
Other Network Technologies

Technologies considered here include X.25, ISDN, SMDS, DDS and Frame Relay. All are provided via physical links which go through telephone exchanges, providing the potential for them to be diverted. Crackers are certainly interested in telephone switches as well as in data networks!

With switched technologies, use Permanent Virtual Circuits or Closed User Groups whenever this is possible. Technologies which provide authentication and/or encryption (such as IPv6) are evolving rapidly; consider using them on links where security is important.

VII. CONCLUSION

Cryptography has emerged as an alternative to protect Internet data and it does the job well. New cryptographic products and technologies have been developed particularly for Internet applications. Thus these Crypto techniques provide sophisticated, protected and reliable networks for secure Data Interchange over the networks.

REFERENCE

[1] www.crypto.com
[2] www.cryptography.com
[3] www.infosyssec.net
[4] www.uow.edu.au
[5] www.amazon.com
[6] www.phptr.com
[7] www.csrc.nist.gov

