جامـــــعة المـــــستـقبـل
AL MUSTAQBAL UNIVERSITY

College of science
Department of Cyber Security

Lecture 1:

Introduction

Stage :2

م.م. مصطفى امير صبري

1. Intoduction of Cryptography

Computers are now fond in every layer society, and information is being communicated and processed automatically on a large scale. Such as medical and financial files, automatic banking, videophones, pay-tv, facsimiles, tele-shopping, and global computer networks. In all these cases there is a growing need for the protection of information to safeguard economic interests, to prevent fraud and to ensure privacy.

The term Cryptography is originally derived from the two greek words “kryptos" and “graph", meaning hidden and writing. This is an accurate representation of the meaning of the word, as cryptography is the art of ensuring that messages (writing) are kept secure (hidden) from those recipients to whom the messages are not addressed .

Cryptography is the science and study of methods of protecting data in computer and communication systems from unauthorized disclosure and modification.

The Cryptographic systems are classified into two cryptosystems, private-key cryptosystem and public-key cryptosystem. Both are based on complex mathematical algorithms and are controlled by keys.

The advances in cryptography have boosted its use in recent decades, opening up an amazing array of applications. It can be used to authenticate computer users, ensure the integrity and confidentiality of electronic communications, and keep sensitive information safely stored. From a secretive military technology, cryptography has emerged a key technology for all participants in the information society concerned about information security.

2. History of Cryptography cryptography arose as a means to enable parties to maintain privacy of the information they send to each other, even in the presence of an adversary with access to the

communication channel. While providing privacy there remains a central goal, the field has expanded to encompass many others, including not just other goals of communication security, such as guaranteeing, integrity and authenticity of communications, but many more sophisticated and fascinating goals. Cryptography is a discipline of mathematics and computer science concerned with information security and related issues, particularly encryption, authentication, and such applications as access Control. Cryptography, as an interdisciplinary subject, draws on several fields. Prior to the early 20th century, cryptography was chiefly concerned with Linguistic patterns. Since then, the emphasis has shifted, and Cryptography now makes extensive use of mathematics, including topics from information theory, computational complexity, statistics, combinatory, and especially number theory.

3. Complexity Theory

Modern cryptography introduces a new dimension: the amount of computing power available to an attacker. It seeks to have security as long as attackers don't have "too much" computing time. Schemes are breakable "in principle," but not in practice. Attacks are infeasible, not impossible. This is a radical shift from many points of view. It takes cryptography from the realm of information theory into the realm of computer science, and complexity theory in particular, since that is where its can be studied how hard, problems are to solve as a function of the computational resources invested. In addition, it changes what can efficiently achieve. It will be wanted to make statements like this: Assuming the attacker uses no more than computing cycles .

Notice again the statement is probabilistic. Almost all of our statements will be. Notice another important thing. Nobody said anything about how the attacker operates. What algorithm, or technique, does he use? Anything about that is not known. The statement holds nonetheless. So it is a very strong statement. It should be clear that, in practice, a statement like the one above would be good enough. As the attacker works harder, his chance of breaking the scheme increases, and if the attacker had 2200 computing cycles at his disposal, no security would be had left at all. But nobody has that much of computing power. Now we must ask ourselves how can hope to get protocols with such properties? The legitimate parties must be able to efficiently execute the protocol instructions: their effort should be reasonable. Somehow, the task of the attacker must be harder. The basic ingredient in any crypto system

is a difficult computational problem. Now it can be introduced some relevant terminology used in cryptography .

- **Algorithm**. An algorithm is an explicit description of how a particular computation should be performed (or a problem solved). The efficiency of an algorithm can be measured as the number of elementary steps it takes to solve the problem. Usually one has to settle for the asymptotic running time,

which can be expressed using the big -*O* notation: The algorithm is considered as *O(f(n))*, if its worst-case running time divided by *f(n)* is bounded by a fixed (positive) constant as the input size *n* increases.

- **Computational Complexity**. A problem is *polynomial* time or *in* **P** if it can be solved by an algorithm which takes less than *O(nt)* steps, where *t* is some finite number and the variable *n* measures the size of the problem instance. If a guessed solution to a problem can be verified in polynomial time then the problem is said to be *in* **NP** (non-deterministic polynomial time). The set of problems that lie in **NP** is very large; it includes the problem of integer factorization. A problem is **NP**-*hard* if there is no other problem in **NP** that is easier to solve. There is no known polynomial time algorithm for any **NP**hard problem, and it is believed that such algorithms in fact do not exist. In public-key cryptography, the attacker is interested in solving particular instances of a problem (factoring some given number), rather than providing a general solution (an algorithm to factor any possible number efficiently). This causes some concern for cryptographers, as some instances of a problem that is **NP**-hard in general may be easily solvable.
- **Factoring**. Every integer can be represented uniquely as a product of prime numbers. For example, *10 = 2 * 5* (the notation * is common for multiplication in computer science) and it is unique (except for the order of the factors *2* and *5*). The art of factorization is almost as old as mathematics itself. However, the study of fast algorithms for factoring is only a few decades old. One possible algorithm for factoring an integer is to divide the input by all small prime

numbers iteratively until the remaining number is prime. This is efficient only for integers that are saying of size less than 10^{100}as this already require trying all primes up to 10^{50}. In public-key cryptosystems based on the problem of factoring, numbers are of size 10^{200} and this would require trying all primes up to 10^{100} and there are about 10^{100} such prime numbers according to the prime number theorem. This far exceeds the number of atoms in the universe, and is unlikely to be enumerated by any effort. The easy instance of factoring is the case where the given integer has only small prime factors. For example, *759375* is easy to factor as we can write it as $3^5 * 5^5$. In cryptography we want to use only those integers that have only large prime factors. Preferably, is selecting an integer with two large prime factors, as is done in the RSA cryptosystem. Currently one of the best factoring algorithms is the number field sieve algorithm (NFS) that consists of a sieving phase and a matrix step. The sieving phase can be distributed (and has been

several times) among a large number of participants, but the matrix step needs to be performed on large supercomputers. The effectiveness of the NFS algorithm becomes apparent for very large integers; it can factor any integer of size 10^{100} in a few months time. The NFS algorithm takes subexponential time (which is still not very efficient). There is no known proof that integer factorization is an **NP**-hard problem nor that it is not polynomial time solvable. If any **NP**-hard problem were polynomial time solvable, then also factoring would, but there is very little hope that this is the case. It is plausible under current knowledge that factoring is not polynomial time solvable **.**

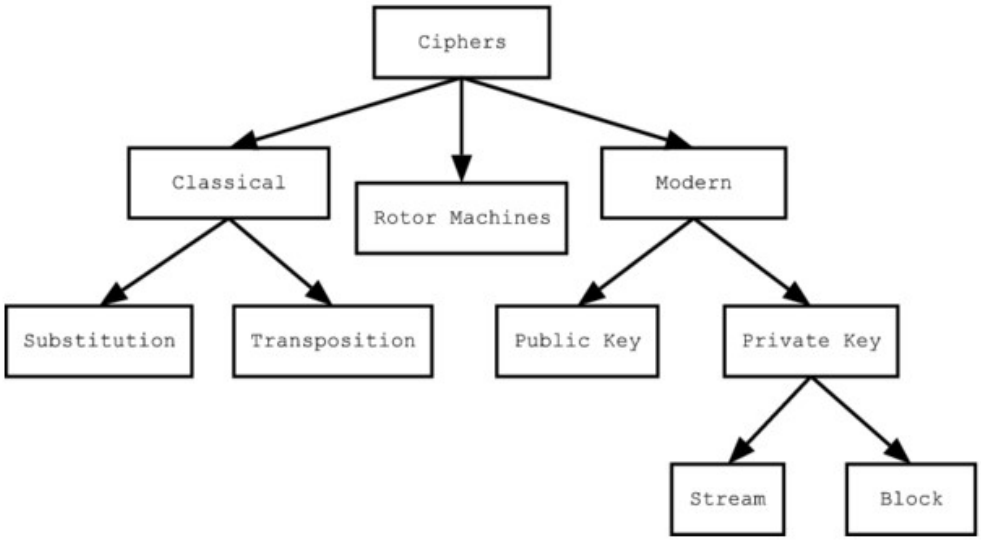
- **Discrete logarithms**. Another important class of problems is the problem of finding n given only some y such that $y = g^n$. The problem is easy for integers, but when the work is in a slightly different setting, it becomes very hard. To obscure the nature of n in g^n , divide the infinite set of integers into a finite set of *remainder classes*. Intuitively, we take the string of integers and wrap it on a circle (which has circumference of length m). The numbers *0, m, 2m, 3m, ...* all cover the same point on the circle, and therefore are said to be in the same equivalence class (we also write "$0 = m = 2m = ...$ *(mod m)*"). Each equivalence class has a least representative in $0 .. m-1$. So you can write any integer n as $t + km$ for any integer t, where

0 <= t < m. It is a convention to write *n = t (mod m)* in this case. Here *m* is said to be the *modulus*.

It can be shown that you can add, subtract and multiply with these Classes of integers (modulo some *m*). This structure, when *m = p* with *p* a prime number, is often called a prime field or even harder to solve the discrete logarithm problem over elliptic curves than over GF(*p*). This has also the effect that there are some key size benefits for using elliptic-curve-based public-key cryptosystems as opposed to factoring-based cryptosystems.

4. Types

There are a variety of different types of encryption. Algorithms used earlier in the history of cryptography are substantially different from modern methods, and modern ciphers can be classified according to how they operate and whether they use one or two keys.



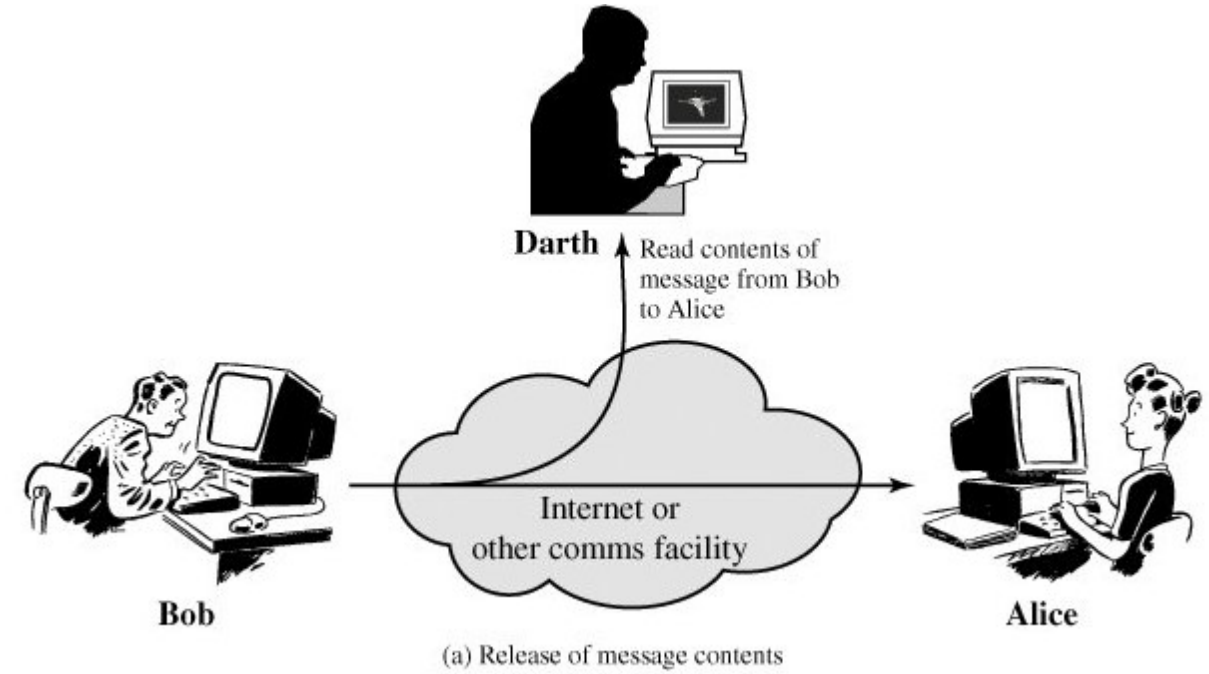
5. Security Attacks

A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of passive attacks and active attacks. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

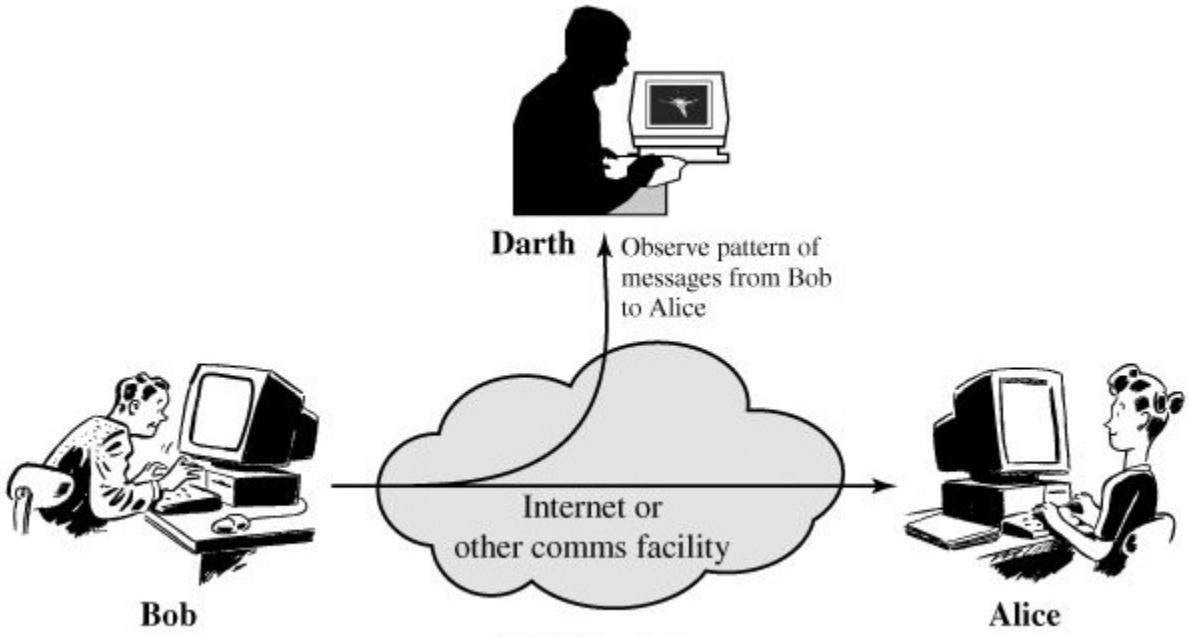
✓ ***Passive Attacks***

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are **release of message contents and traffic analysis.**

The release of message contents is easily understood (following Figure a). A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.



(a) Release of message contents



(b) Traffic analysis

A second type of passive attack, **traffic analysis**, is subtler (above b). Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to

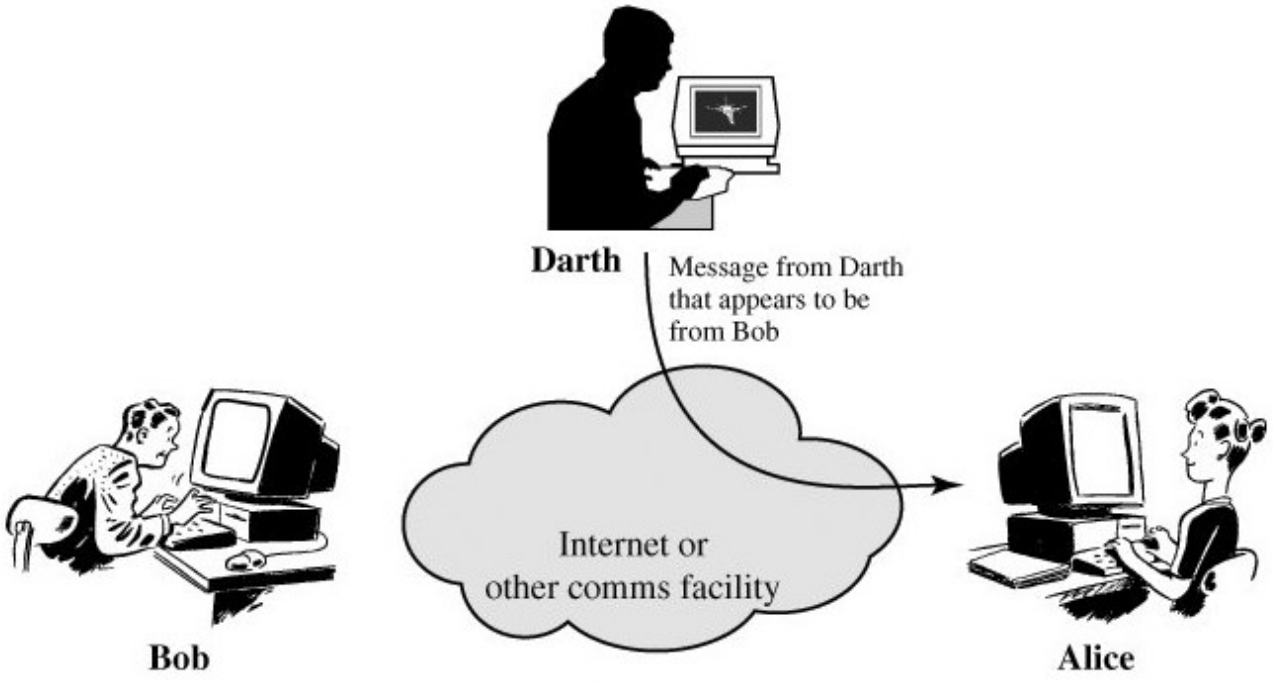
observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

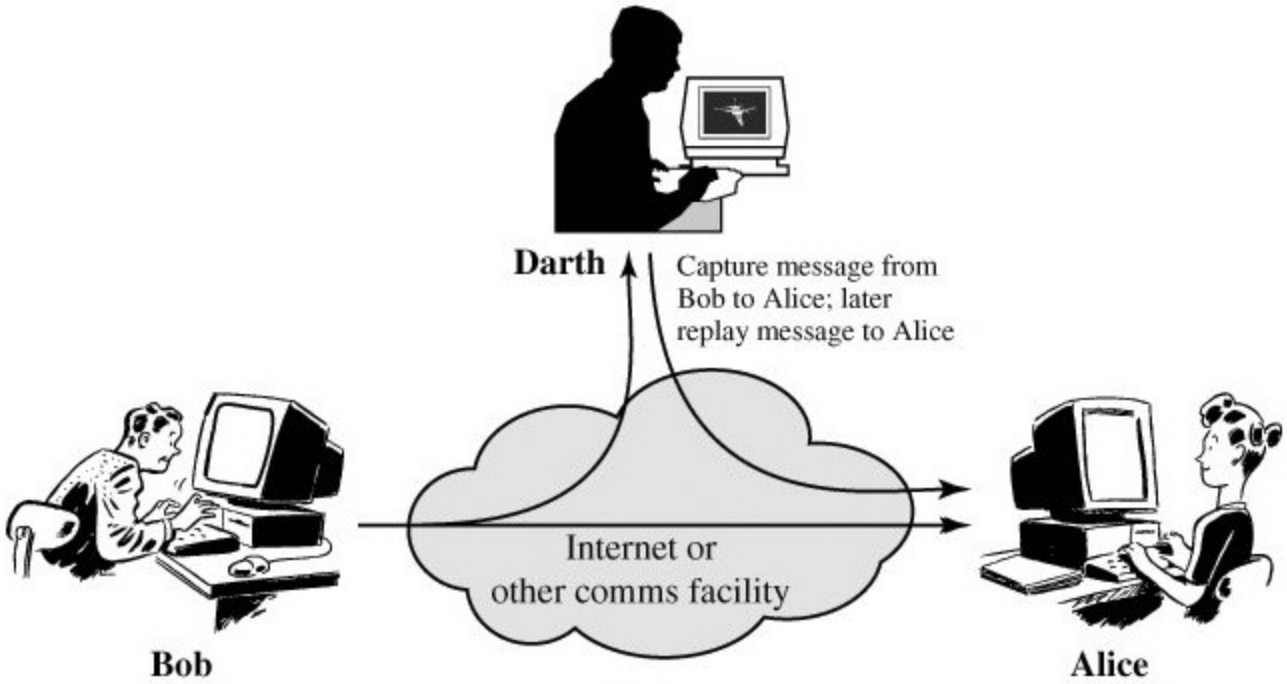
✓ *Active Attacks*

Active attacks involve some **modification of the data stream** or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

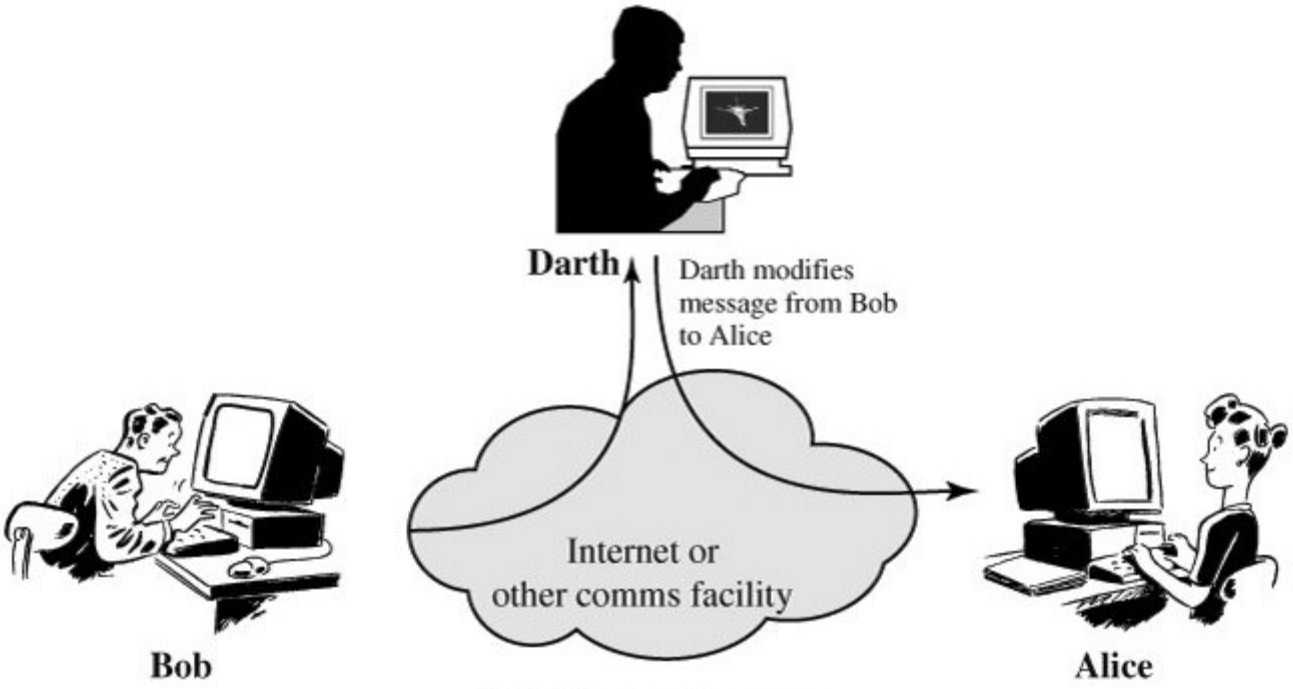
A **masquerade** takes place when one entity pretends to be a different entity (following Figure a). A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.



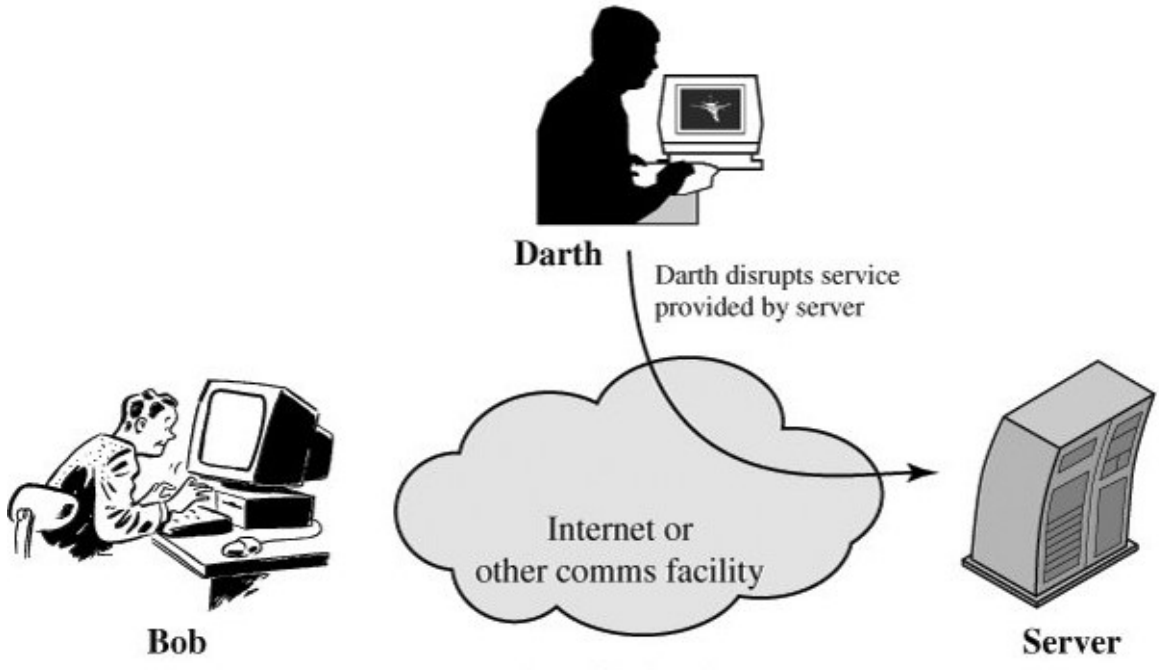
(a) Masquerade



(b) Replay



(c) Modification of messages



(d) Denial of service

Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect(Figure b)

Modification of messages: simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (Figure c). For example, a message meaning "Allow John Smith to read confidential file accounts" is modified to mean "**Allow Fred Brown to read confidential file accounts.**"

Denial of service: prevents or inhibits the normal use or management of communications facilities (Figure d). This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks absolutely, because of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it may also contribute to prevention.