CSE 5171 Advanced Cryptography

Dr. Renuka A
Professor
Department of Computer Science and Engineering
Manipal Institute of Technology Manipal

Syllabus

Groups, rings, Fields, Characteristic of a field, prime fields, Arithmetic of polynomials over fields. Field extensions, Galois group of field extensions, Fixed field and Galois extensions. Minimum polynomial, Splitting field of a polynomial, Separable polynomial and Separable extensions. Construction of finite fields and their structure. Enumeration of irreducible polynomials over finite fields. The fundamental theorem of Galois Theory. ElGamal Cryptosystem, Elliptic Curve Architecture, and Cryptography: Elliptic Curve over real numbers, Elliptic Curve Cryptography, ECDH, ECDSA. RSA variants. Authentication functions, Message Authentication Codes and systems, Advanced Digital signature systems. Entity Authentication, One-time password, Challenge – Response: using a symmetric- key cipher, using keyed-hash functions, using as an asymmetric-key cipher, using a digital signature, Zero-Knowledge proof, Fiat. -Shamir protocol, Feige-Fiat-Shamir protocol, Guillou-Quisquater protocol, Biometric, Key Management: Symmetric key distribution, servers. the symmetric key agreement, Deffie-Hellman key agreement, Station to station key agreement. public key distribution, public announcements, certification authority, public key infrastructure, trust model, hijacking.

Detailed Syllabus

1. INTRODUCTION

A quick introduction to groups, rings, integral domain, and fields.

2. BACKGROUND THEORY

Fields, Characteristic of a field, prime fields, Arithmetic of polynomials over fields. Field extensions, Galois group of field extensions, Fixed field and Galois extensions. Minimum polynomial, Construction of fields with the help of an irreducible polynomial. Splitting field of a polynomial, Separable polynomial and Separable extensions. Construction of finite fields and their structure. Enumeration of irreducible polynomials over finite fields. The fundamental theorem of Galois Theory. Overview of Fermat’s Little Theorem, Euler’s Theorem, Chinese remainder theorem, Primality testing algorithm, Euclid’s algorithm for integers. Cauchy 's theorem quadratic residues, Legendre symbol, Jacobi symbol.

3. PUBLIC KEY CRYPTOSYSTEMS

ElGamal Cryptosystem, Elliptic Curve Architecture, and Cryptography: Elliptic Curve over real numbers, Elliptic Curve over GF(p), Elliptic Curve GF(2^n), Elliptic Curve Cryptography simulating ElGamal, Elliptic Curve Cryptography, ECDH, ECDSA. RSA variants

4. HASHING

Cryptographic hash functions, Properties of hashing, Serial and parallel hashing, Hashing based on Cryptosystems, MD5, Keyed hashing. Authentication requirements, Authentication functions, Message Authentication Codes, Hash Functions, MD5 message Digest algorithm, Secure Hash Algorithm, HMAC, CMAC.

5. DIGITAL SIGNATURES

RSA signatures, Blind signatures, Authentication Protocols, Digital Signature Standard (DSS), ElGamal DSS, Schnorr Digital Signature Scheme, ECDSA, Variations, the stamped signatures, Blind Signatures, Undeniable Digital Signatures

6. ENTITY AUTHENTICATION

Data-origin versus Entity Authentication, One-time password, Challenge – Response, using a symmetric- key cipher, using keyed-hash functions, using as an asymmetric-key cipher, using a digital signature, Zero-Knowledge, Fiat. -Shamir protocol, Feige-Fiat-Shamir protocol, Guillou-Quisquater protocol, Biometric.

7. KEY MANAGEMENT

Symmetric key distribution, KDC, session keys, servers. the symmetric key agreement, Deffie-Hellman key agreement, Station to station key agreement. public key distribution, public announcements, trusted center, controlled trusted center, certification authority, X.509, public key infrastructure, trust model, hijacking

References

- Behrouz A. Forouzan and Debdeep Mukhopadhyay – “Cryptography and Network Security”, McGraw Hill, 2nd Edition, 2008.
- S. Vaudenay, “A Classical Introduction to Cryptography: Applications for Communications Security”, Springer International Edition, 2006.
- Lawrence C. Washington, “Elliptic curves: number theory and cryptography”, Chapman & Hall/ CRC Second Edition, 2008.
- William Stallings,“Cryptography And Network Security Principles And Practice”, Fifth Edition, Pearson Education, 2013

COURSE OUTCOMES

CO1: Describe the principles of number theory for cryptography
CO2: Apply number theory concepts in cryptographic algorithms
CO3: Analyse the various hashing algorithms
CO4: Compare the various digital signature schemes
CO5:Demonstrate the concepts of entity authentication and key management

Introduction

The art of war teaches us to rely not on the likelihood of the enemy's not coming, but on our own readiness to receive him; not on the chance of his not attacking, but rather on the fact that we have made our position unassailable.

—*The Art of War,* Sun Tzu

- Information security-provided by physical and administrative means
- Computer security-collection of tools designed to protect data
- Network security (internet security)-means to protect data during transmission

Examples of security violations

- A→B, C captures a copy of the file
- D→E, F intercepts, alters and forwards
- F→E own message
- Intercepts, delays and then transmits
- Denies that a message was sent

Internetwork security -Complex

- Mechanisms are complex
- Potential attacks need to be considered
- Decide where to use them(physical /logical)
- Secret key
  - Creation, distribution, protection

  Reliance on communication protocols

  Time limits on transit time

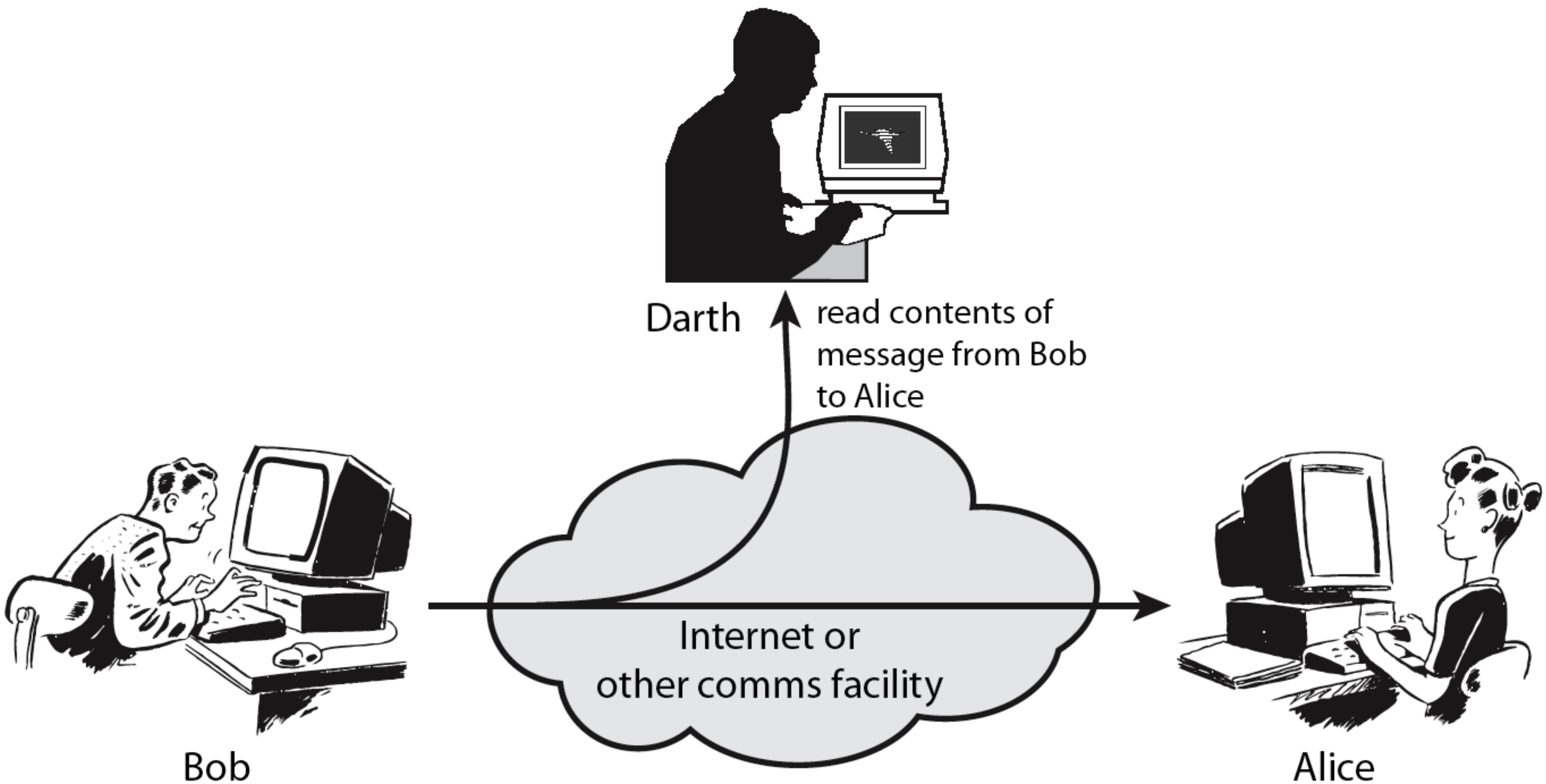
ITU/IETF X.800: Security Threats, Attacks, Services, and Mechanisms

- *Security Threat*: A potential for violation of security ; Possible danger that might exploit a vulnerability
- *Security Attack*: An attempt to compromise the security of systems or information
  - Example: Eavesdropping on communication
- *Security Service*: Use of one or more mechanisms to enhance the security of a system or application
  - Example: Confidentiality of communications
- *Security Mechanism*: A specific method to detect, prevent, or recover from an attack, and to provide the required service
  - Example: Encryption software

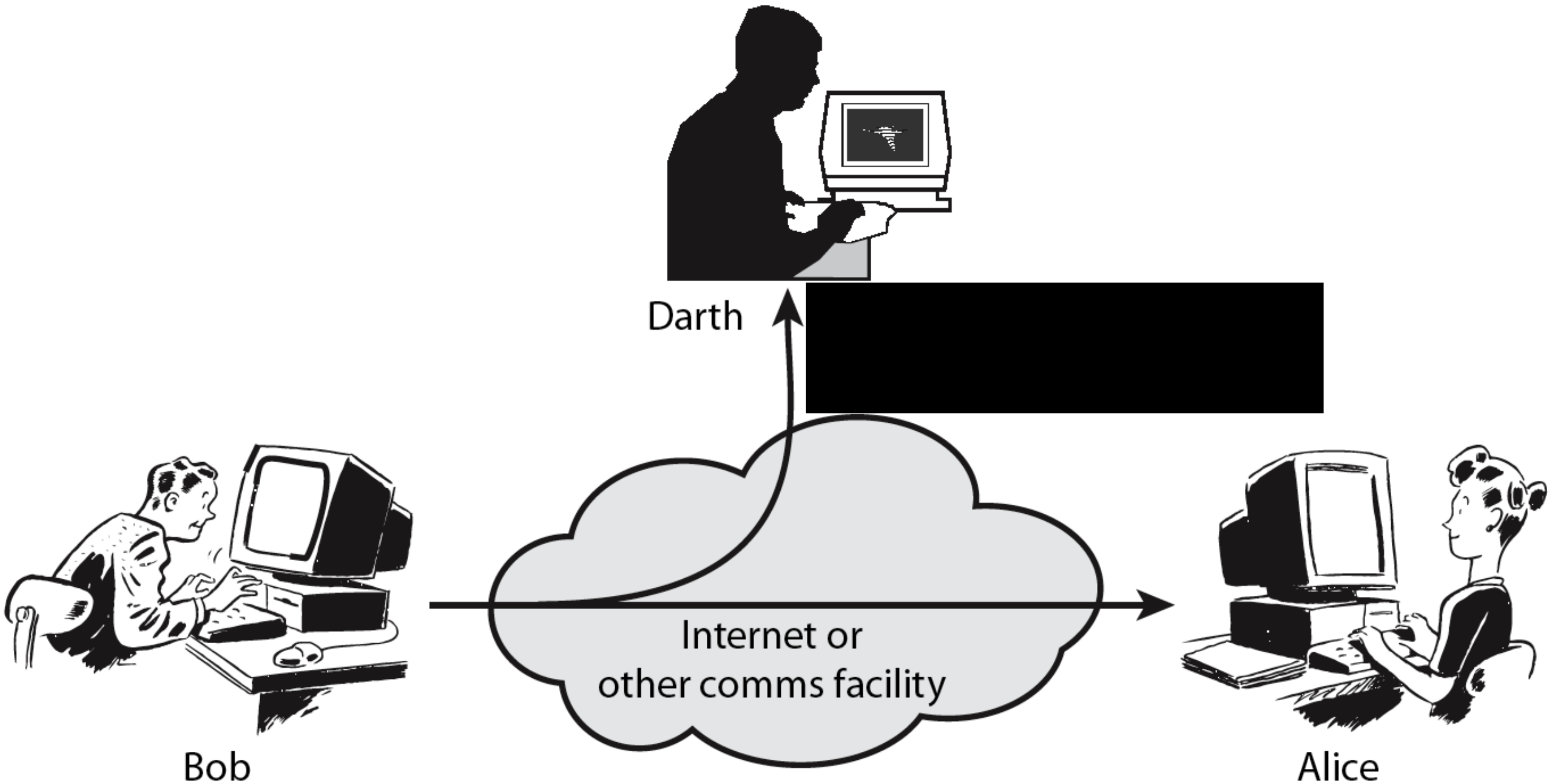
Classify Security Attacks

- **passive attacks** - eavesdropping on, or monitoring of, transmissions to:
  - obtain message contents, or
  - monitor traffic flows (traffic analysis)
    - Location of communicating hosts and observe frequency and length of messages being exchanged.
- **active attacks** – modification of data stream to:
  - masquerade of one entity as some other
  - replay previous messages
  - modify messages in transit
  - denial of service

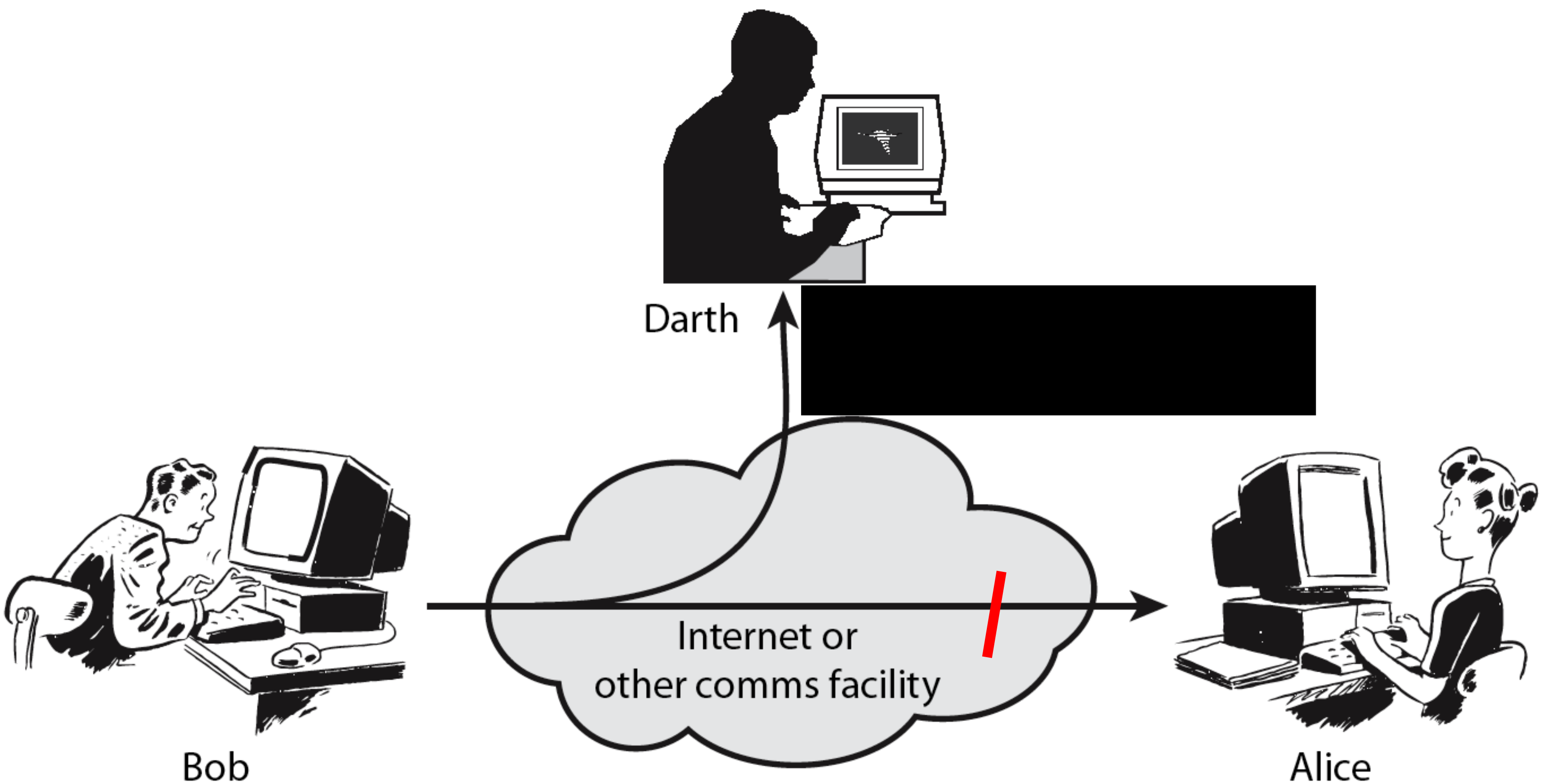
Passive Attack - Interception



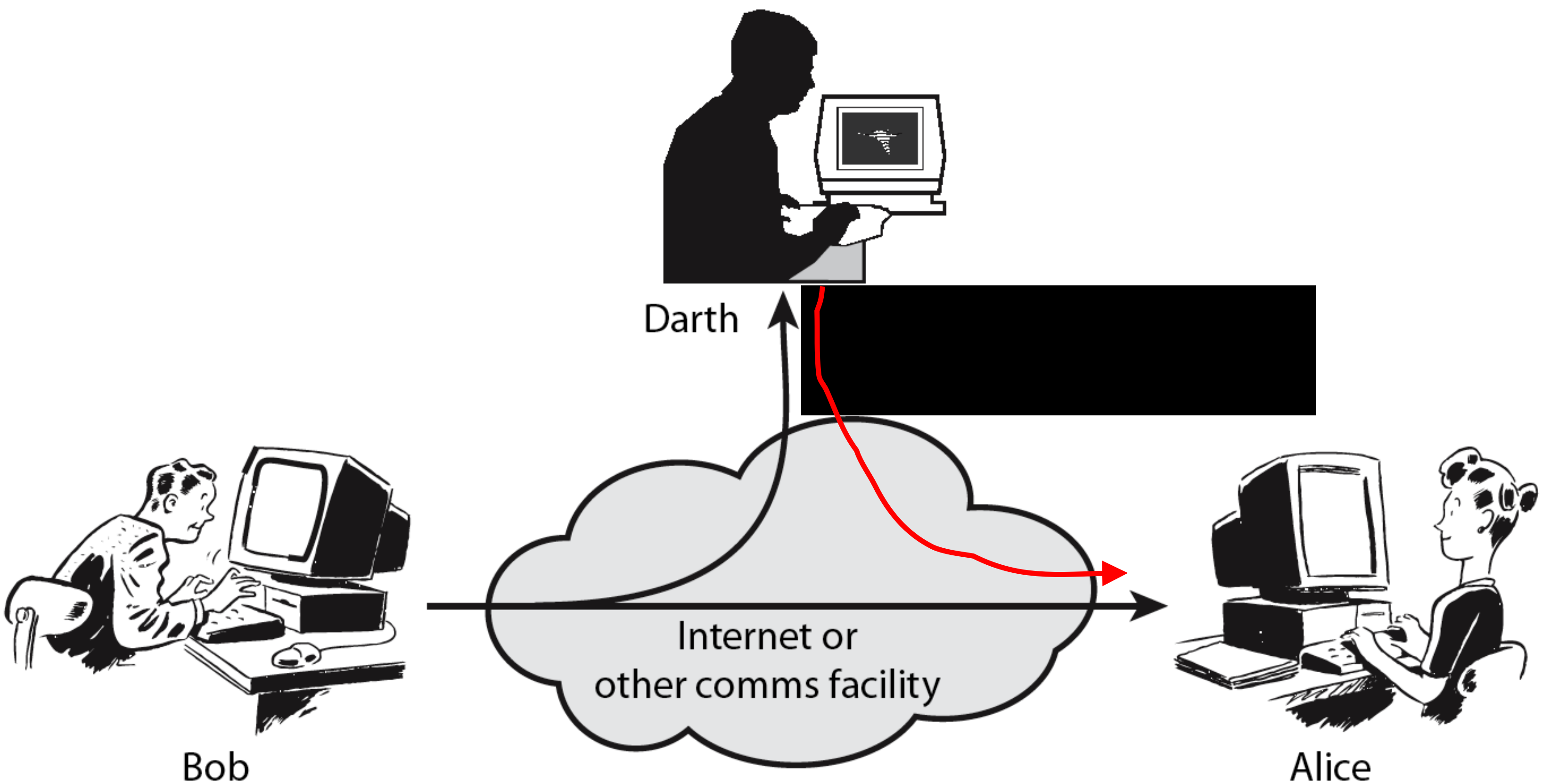
Passive Attack: Traffic Analysis



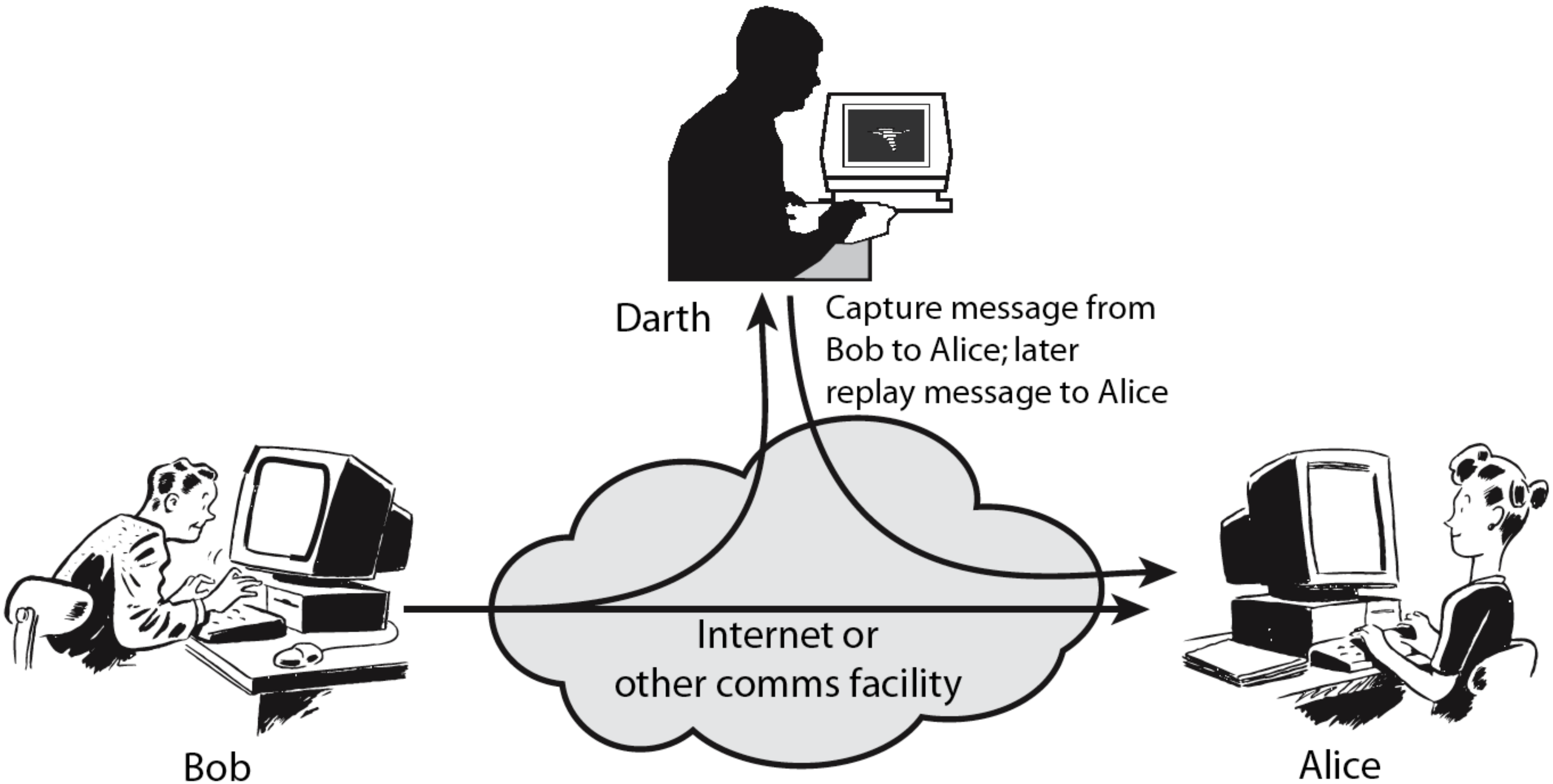
Active Attack: Interruption



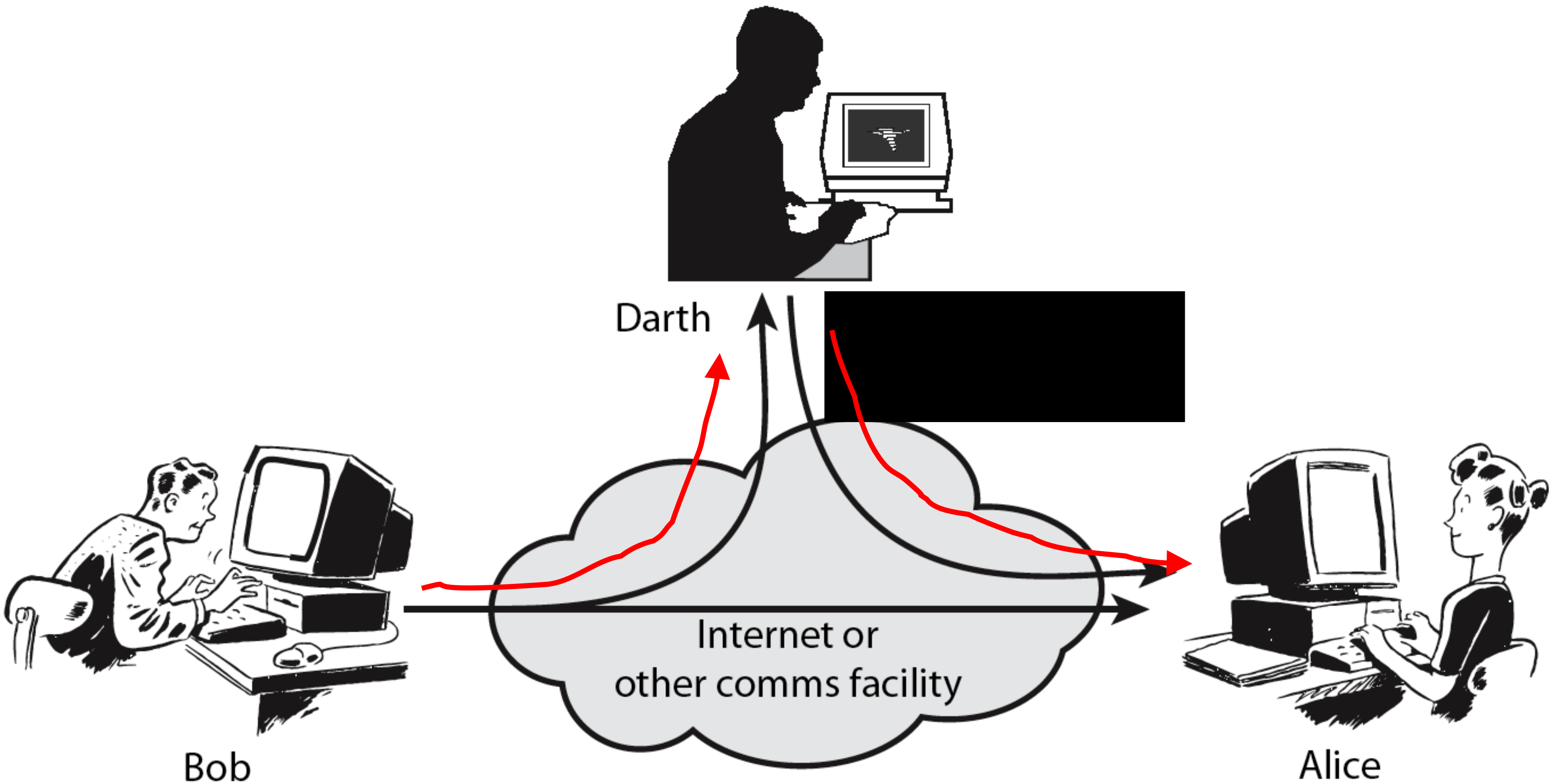
Active Attack: Fabrication



Active Attack: Replay



Active Attack: Modification



Security Services

- X.800 defines it as: a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers
- RFC 2828 defines it as: a processing or communication service provided by a system to give a specific kind of protection to system resources
- X.800 defines it in 5 major categories

X.800 Security Services

- Authentication
  – Identify peers, Source authentication for data
- Access Control
  – Who can access to what
- Data Confidentiality
  – Connection, Connectionless (system), Traffic, Privacy
- Data Integrity
  – With or without recovery
- Non-repudiation
  – Origin, Destination, Both
- Availability
  – A service on its own, or a property of other services

Security Services (X.800)

- **Authentication** - assurance that the communicating entity is the one claimed
  - Peer entity Authentication-provide confidence in the entities connected logical connection)
  - Data Origin Authentication-provides assurance that the source of received data is as claimed,
- **Access Control** - prevention of the unauthorized use of a resource
- **Data Confidentiality** –protection of data from unauthorized disclosure
  - Connection Cofidentiality- Protection of all user data on a connection
  - Connectionless Confidentiality-Protection of all user data in a single block
  - Selective Field Confidentiality-confidentiality of selected fields within the user data on a connection or in a single block
  - Traffic Flow Confidentiality-protection of information that might be derived from observation of traffic flows

- **Data Integrity** - assurance that data received is as sent by an authorized entity
  - Connection Integrity with recovery-Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion etc. with recovery attempted
  - Connection Integrity with recovery-Provides only detection without recovery
  - Selective field Connection Integrity-provides integrity for selected field
  - Connectionless integrity-integrity of single connectionless data block-detection of data modification, limited form of replay detection
  - Selective field Connectionless Integrity-
- **Non-Repudiation** - protection against denial by one of the entities involved in a communication
  - Nonrepudiation, Origin- proof that the message was sent by the specified party
  - Nonrepudiation, Destination-proof that the message was received by the specified party

Security Mechanisms (X.800)

- specific security mechanisms:
  - encipherment, digital signatures, access controls, data integrity, authentication exchange, traffic padding, routing control, notarization
- pervasive security mechanisms:
  - trusted functionality, security labels, event detection, security audit trails, security recovery

Figure 1.3 Security services

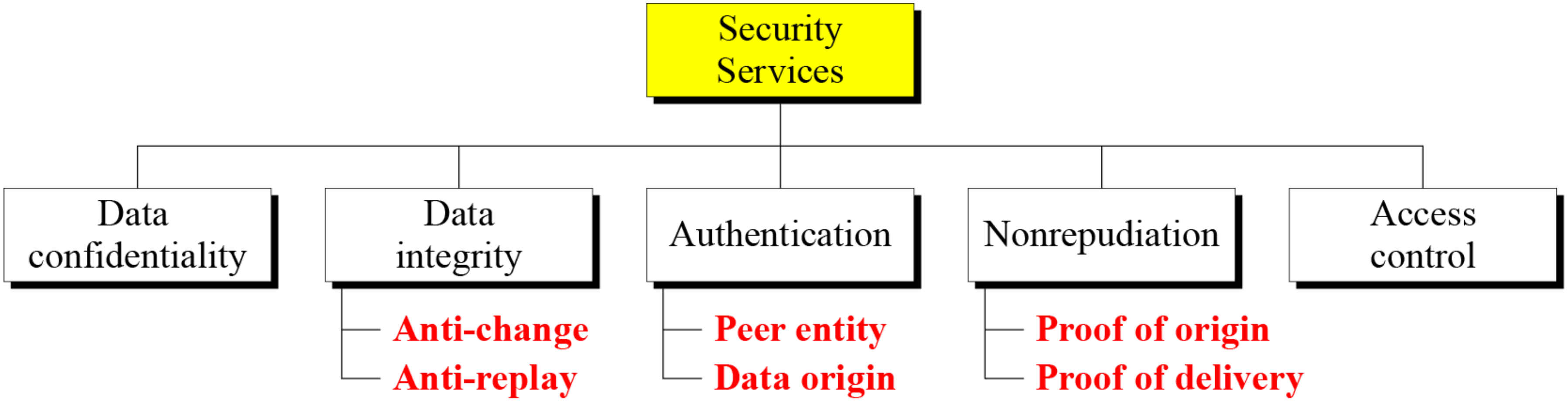
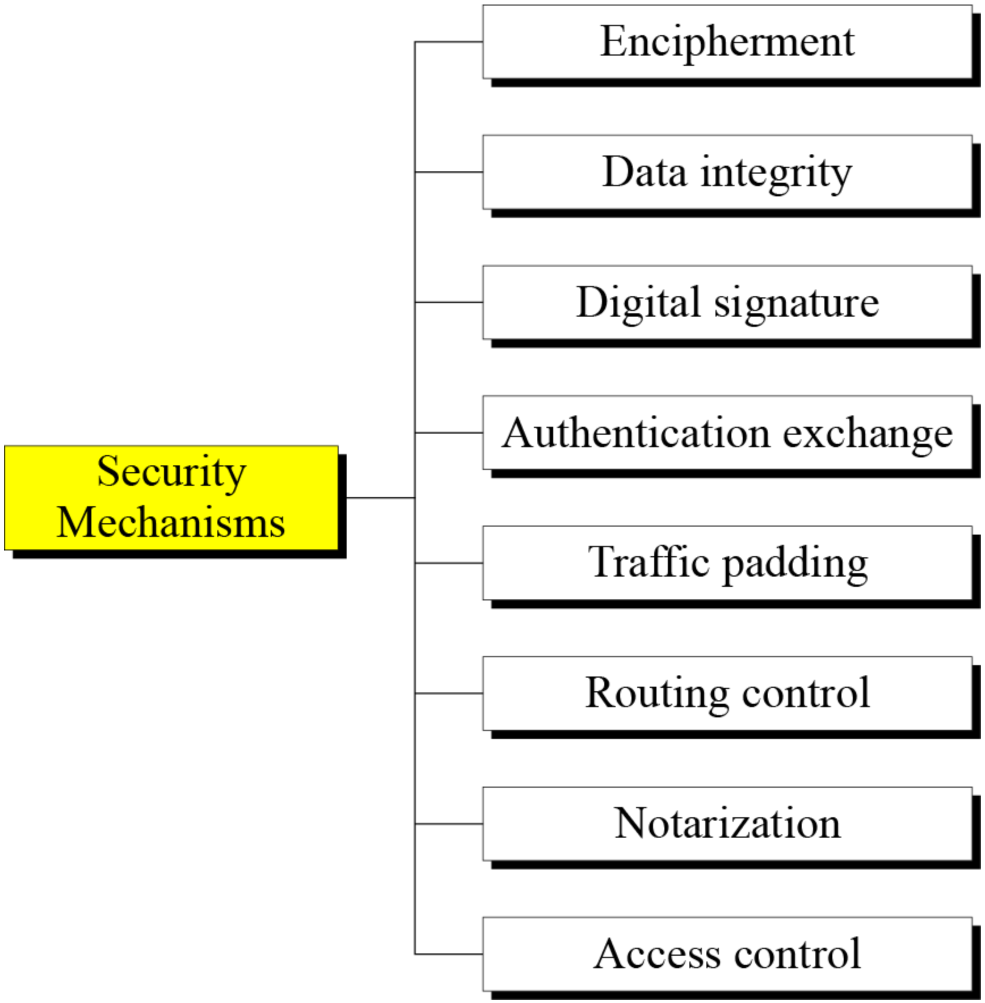


Figure 1.4 Security mechanisms



1.3.3 Relation between Services and Mechanisms

Table 1.2 Relation between security services and mechanisms

Security Service	*Security Mechanism*
Data confidentiality	Encipherment and routing control
Data integrity	Encipherment, digital signature, data integrity
Authentication	Encipherment, digital signature, authentication exchanges
Nonrepudiation	Digital signature, data integrity, and notarization
Access control	Access control mechanism

Secured Access Model

- Identify and filter requests for information

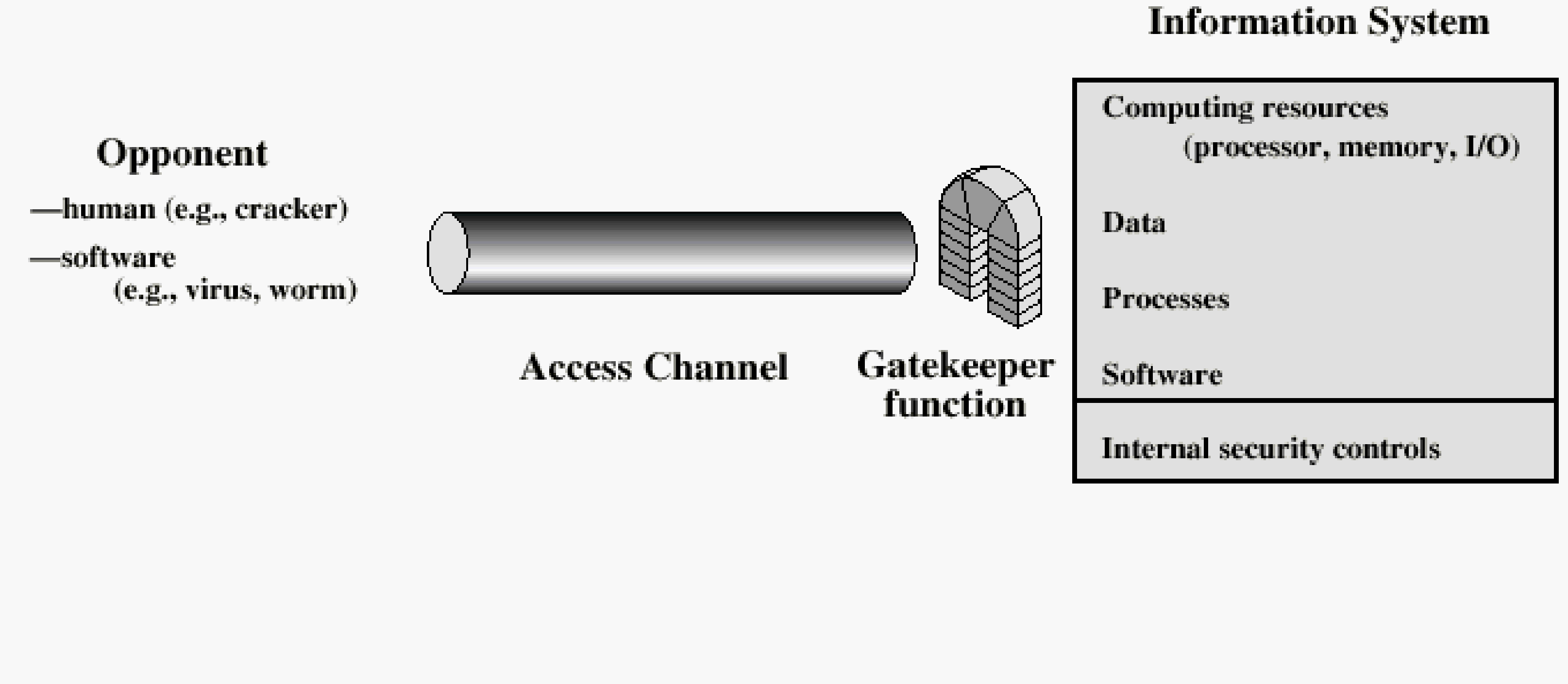


Figure 1.4 Network Access Security Model

Modular Arithmetic

- define **modulo operator** `a mod n` to be remainder when a is divided by n
- use the term **congruence** for: $a \equiv b \bmod n$
  - when divided by *n,* a & b have same remainder
  - eg. 100 = 34 mod 11
- b is called the **residue** of a mod n
  - since with integers can always write: $a = qn + b$
- usually have $0 <= b <= n-1$

$$-12 \bmod 7 \equiv -5 \bmod 7 \equiv 2 \bmod 7 \equiv 9 \bmod 7$$

Modulo 7 Example

```
...
-21 -20 -19 -18 -17 -16 -15
-14 -13 -12 -11 -10  -9  -8
 -7  -6  -5  -4  -3  -2  -1
  0   1   2   3   4   5   6
  7   8   9  10  11  12  13
 14  15  16  17  18  19  20
 21  22  23  24  25  26  27
 28  29  30  31  32  33  34
...
```

Divisors

- say a non-zero number `b` **divides** `a` if for some `m` have `a=mb` (`a,b,m` all integers)
- that is `b` divides into `a` with no remainder
- denote this `b|a`
- and say that `b` is a **divisor** of `a`
- eg. all of 1,2,3,4,6,8,12,24 divide 24

Modular Arithmetic Operations

- is 'clock arithmetic'
- uses a finite number of values, and loops back from either end
- modular arithmetic is when do addition & multiplication and modulo reduce answer
- can do reduction at any point, ie
  - `a+b mod n = [a mod n + b mod n] mod n`

Modular Arithmetic

- can do modular arithmetic with any group of integers: $Z_n = \{0, 1, \ldots, n-1\}$
- form a commutative ring for addition
- with a multiplicative identity
- note some peculiarities
  - if $(a+b) \equiv (a+c) \bmod n$ then $b \equiv c \bmod n$
  - but $(ab) \equiv (ac) \bmod n$ then $b \equiv c \bmod n$ only if a is relatively prime to n

Modulo 8 Example

+	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	1	2	3	4	5	6	7	0
2	2	3	4	5	6	7	0	1
3	3	4	5	6	7	0	1	2
4	4	5	6	7	0	1	2	3
5	5	6	7	0	1	2	3	4
6	6	7	0	1	2	3	4	5
7	7	0	1	2	3	4	5	6

(a) Addition modulo 8

Greatest Common Divisor (GCD)

- a common problem in number theory
- GCD (a,b) of a and b is the largest number that divides evenly into both a and b
  - eg GCD(60,24) = 12
- often want **no common factors** (except 1) and hence numbers are **relatively prime**
  - eg GCD(8,15) = 1
  - hence 8 & 15 are relatively prime

Euclid's GCD Algorithm

- an efficient way to find the GCD(a,b)
- uses theorem that:
  - `GCD(a,b) = GCD(b, a mod b)`
- **Euclid's Algorithm** to compute GCD(a,b):
  - `A=a, B=b`
  - `while B>0`
    - `R = A mod B`
    - `A = B, B = R`
  - `return A`

Example GCD(1970,1066)

1970 = 1 x 1066 + 904	gcd(1066, 904)
1066 = 1 x 904 + 162	gcd(904, 162)
904 = 5 x 162 + 94	gcd(162, 94)
162 = 1 x 94 + 68	gcd(94, 68)
94 = 1 x 68 + 26	gcd(68, 26)
68 = 2 x 26 + 16	gcd(26, 16)
26 = 1 x 16 + 10	gcd(16, 10)
16 = 1 x 10 + 6	gcd(10, 6)
10 = 1 x 6 + 4	gcd(6, 4)
6 = 1 x 4 + 2	gcd(4, 2)
4 = 2 x 2 + 0	gcd(2, 0)

Prime Numbers

- prime numbers only have divisors of 1 and self
  - they cannot be written as a product of other numbers
  - note: 1 is prime, but is generally not of interest
- eg. 2,3,5,7 are prime, 4,6,8,9,10 are not
- prime numbers are central to number theory
- list of prime number less than 200 is:

```
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59
61 67 71 73 79 83 89 97 101 103 107 109 113 127
131 137 139 149 151 157 163 167 173 179 181 191
193 197 199
```

Prime Factorisation

- to **factor** a number `n` is to write it as a product of other numbers: `n=a x b x c`
- note that factoring a number is relatively hard compared to multiplying the factors together to generate the number
- the **prime factorisation** of a number `n` is when its written as a product of primes
  - eg. $91=7 \times 13$; $3600=2^4 \times 3^2 \times 5^2$

$$a = \prod_{p \in \mathrm{P}} p^{a_p}$$

Relatively Prime Numbers & GCD

- two numbers a, b are **relatively prime** if have **no common divisors** apart from 1
  - eg. 8 & 15 are relatively prime since factors of 8 are 1,2,4,8 and of 15 are 1,3,5,15 and 1 is the only common factor
- conversely can determine the greatest common divisor by comparing their prime factorizations and using least powers
  - eg. $300=2^1 x 3^1 x 5^2$ $18=2^1 x 3^2$ hence $GCD(18,300)=2^1 x 3^1 x 5^0=6$

Fermat's Theorem

- $a^{p-1} = 1 \pmod{p}$
  - where p is prime and $\gcd(a,p)=1$
- also known as Fermat’s Little Theorem
- also have: $a^p = a \pmod{p}$
- useful in public key and primality testing

Euler Totient Function $\varnothing(\mathtt{n})$

- when doing arithmetic modulo n
- **complete set of residues** is: `0..n-1`
- **reduced set of residues** is those numbers (residues) which are relatively prime to n
  - eg for n=10,
  - complete set of residues is {0,1,2,3,4,5,6,7,8,9}
  - reduced set of residues is {1,3,7,9}
- number of elements in reduced set of residues is called the **Euler Totient Function ø(n)**

Euler Totient Function $\varnothing(n)$

- to compute ø(n) need to count number of residues to be excluded
- in general need prime factorization, but
  - for p (p prime) $\varnothing(p) = p-1$
  - for p.q (p,q prime) $\varnothing(p.q) = (p-1)x(q-1)$
- eg.

$\varnothing(37) = 36$

$\varnothing(21) = (3-1)x(7-1) = 2x6 = 12$

Euler's Theorem

- a generalisation of Fermat's Theorem
- $a^{\varnothing(n)} = 1 \pmod{n}$
  - for any a,n where $\gcd(a,n)=1$
- eg.

  $a=3;n=10;\ \varnothing(10)=4;$

  hence $3^4 = 81 = 1 \bmod 10$

  $a=2;n=11;\ \varnothing(11)=10;$

  hence $2^{10} = 1024 = 1 \bmod 11$
- also have: $a^{\varnothing(n)+1} = a \pmod{n}$

Primality Testing

- often need to find large prime numbers
- traditionally **sieve** using **trial division**
  - ie. divide by all numbers (primes) in turn less than the square root of the number
  - only works for small numbers
- alternatively can use statistical primality tests based on properties of primes
  - for which all primes numbers satisfy property
  - but some composite numbers, called pseudo-primes, also satisfy the property
- can use a slower deterministic primality test

Miller Rabin Algorithm

- a test based on prime properties that result from Fermat’s Theorem
- algorithm is:

TEST (n) is:

1. Find integers k, q, $k > 0$, q odd, so that $(n-1)=2^k q$
2. Select a random integer a, $1<a<n-1$
3. **if** $a^q \bmod n = 1$ **then** return (“inconclusive");
4. **for** $j = 0$ **to** $k - 1$ **do**
   5. **if** ($a^{2^j q} \bmod n = n-1$)
      then return(“inconclusive")
6. return (“composite")

Probabilistic Considerations

- if Miller-Rabin returns “composite” the number is definitely not prime
- otherwise is a prime or a pseudo-prime
- chance it detects a pseudo-prime is < $^1/_4$
- hence if repeat test with different random a then chance n is prime after t tests is:
  - Pr(n prime after t tests) = $1-4^{-t}$
  - eg. for t=10 this probability is > 0.99999
- could then use the deterministic AKS test

Prime Distribution

- prime number theorem states that primes occur roughly every (`ln n`) integers
- but can immediately ignore evens
- so in practice need only test `0.5 ln(n)` numbers of size `n` to locate a prime
  - note this is only the "average"
  - sometimes primes are close together
  - other times are quite far apart

Chinese Remainder Theorem

- used to speed up modulo computations
- if working modulo a product of numbers
  - eg. $\text{mod } M = m_1 m_2 .. m_k$
- Chinese Remainder theorem lets us work in each moduli m_i separately
- since computational cost is proportional to size, this is faster than working in the full modulus M

Chinese Remainder Theorem

- can implement CRT in several ways
- to compute $A \pmod{M}$
  - first compute all $a_i = A \bmod m_i$ separately
  - determine constants c_i below, where $M_i = M/m_i$
  - then combine results to get answer using:

$$A \equiv \left(\sum_{i=1}^{k} a_i c_i \right) \pmod{M}$$

$$c_i = M_i \times (M_i^{-1} \bmod m_i) \quad \text{for } 1 \le i \le k$$

TEST (*n*)

```
1. Find integers k, q, with k > 0, q odd, so that
   (n − 1 = 2^k q);
2. Select a random integer a, 1 < a < n − 1;
3. if a^q mod n = 1 then return("inconclusive");
4. for j = 0 to k − 1 do
5. if a^(2^j q) mod n = n − 1 then return("inconclusive");
6. return("composite");
```

Let us apply the test to the prime number $n = 29$. We have $(n - 1) = 28 = 2^2(7) = 2^k q$. First, let us try $a = 10$. We compute $10^7 \bmod 29 = 17$, which is neither 1 nor 28, so we continue the test. The next calculation finds that $(10^7)^2 \bmod 29 = 28$, and the test returns **inconclusive** (i.e., 29 may be prime). Let's try again with $a = 2$. We have the following calculations: $2^7 \bmod 29 = 12$; $2^{14} \bmod 29 = 28$; and the test again returns **inconclusive**. If we perform the test for all integers a in the range 1 through 28, we get the same **inconclusive** result, which is compatible with n being a prime number.

Now let us apply the test to the composite number $n = 13 \times 17 = 221$. Then $(n-1) = 220 = 2^2(55) = 2^k q$. Let us try $a = 5$. Then we have $5^{55} \bmod 221 = 112$, which is neither 1 nor 220 $(5^{55})^2 \bmod 221 = 168$. Because we have used all values of j (i.e., $j = 0$ and $j = 1$) in line 4 of the TEST algorithm, the test returns **composite**, indicating that 221 is definitely a composite number. But suppose we had selected $a = 21$. Then we have $21^{55} \bmod 221 = 200$; $(21^{55})^2 \bmod 221 = 220$; and the test returns **inconclusive**, indicating that 221 may be prime. In fact, of the 218 integers from 2 through 219, four of these will return an inconclusive result, namely 21, 47, 174, and 200.

To represent 973 mod 1813 as a pair of numbers mod 37 and 49, define[9]

$$m_1 = 37$$

$$m_2 = 49$$

$$M = 1813$$

$$A = 973$$

We also have $M_1 = 49$ and $M_2 = 37$. Using the extended Euclidean algorithm, we compute $M_1^{-1} = 34 \bmod m_1$ and $M_2^{-1} = 4 \bmod m_2$. (Note that we only need to compute each M_i and each M_i^{-1} once.) Taking residues modulo 37 and 49, our representation of 973 is (11, 42), because 973 mod 37 = 11 and 973 mod 49 = 42.

Now suppose we want to add 678 to 973. What do we do to (11, 42)? First we compute $(678) \leftrightarrow (678 \bmod 37, 678 \bmod 49) = (12, 41)$. Then we add the tuples element-wise and reduce $(11 + 12 \bmod 37, 42 + 41 \bmod 49) = (23, 34)$. To verify that this has the correct effect, we compute

$$\begin{aligned}(23, 34) &\leftrightarrow a_1 M_1 M_1^{-1} + a_2 M_2 M_2^{-1} \bmod M \\ &= [(23)(49)(34) + (34)(37)(4)] \bmod 1813 \\ &= 43350 \bmod 1813 \\ &= 1651\end{aligned}$$

and check that it is equal to $(973 + 678) \bmod 1813 = 1651$. Remember that in the above derivation, M_i^{-1} is the multiplicative inverse of M_1 modulo m_1 modulo M_2^{-1} is the multiplicative inverse of M_2 modulo m_2.

Extended Euclidean Method to find Multiplicative Inverse

Find inverse of b in Zn when n and b are given and gcd(n, b) =1

$r_1 \leftarrow n;\ r_2 \leftarrow b;$

$t_1 \leftarrow 0;\ t_2 \leftarrow 1;$

while $(r_2 > 0)$

{

$q \leftarrow r_1 / r_2;$

$r \leftarrow r_1 - q \times r_2;$

$r_1 \leftarrow r_2;\quad r_2 \leftarrow r;$

$t \leftarrow t_1 - q \times t_2;$

$t_1 \leftarrow t_2;\quad t_2 \leftarrow t;$

}

if $(r_1 = 1)$ then $b^{-1} \leftarrow t_1$

Find inverse of 11 in Z_{26}

q	r_1	r_2	r	t_1	t_2	t
2	26	11	4	0	1	−2
2	11	4	3	1	−2	5
1	4	3	1	−2	5	−7
3	3	1	0	5	−7	26
	1	0		−7	26	

- Multiplicative inverse is (-7) mod 26 = 19

- Find the multiplicative inverse of 23 in Z_{100}.

- Find the multiplicative inverse of 23 in Z_{100}.

q	r_1	r_2	r	t_1	t_2	t
4	100	23	8	0	1	−4
2	23	8	7	1	−4	19
1	8	7	1	−4	9	−13
7	7	1	0	9	−13	100
	1	0		−13	100	

- Find inverse of 8 in Z_{17}

Chinese Remainder Theorem

The **Chinese remainder theorem** (CRT) is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime, as shown below:

$$
\begin{gathered}
x \equiv a_1 \pmod{m_1} \\
x \equiv a_2 \pmod{m_2} \\
\cdots \\
x \equiv a_k \pmod{m_k}
\end{gathered}
$$

The solution to the set of equations follows these steps:

1. Find $M = m_1 \times m_2 \times \cdots \times m_k$. This is the common modulus.
2. Find $M_1 = M/m_1$, $M_2 = M/m_2$, …, $M_k = M/m_k$.
3. Find the multiplicative inverse of M_1, M_2, …, M_k using the corresponding moduli (m_1, m_2, …, m_k). Call the inverses M_1^{-1}, M_2^{-1}, …, M_k^{-1}.
4. The solution to the simultaneous equations is

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \cdots + a_k \times M_k \times M_k^{-1}) \bmod M$$

$$x \equiv 2 \pmod{3}$$
$$x \equiv 3 \pmod{5}$$
$$x \equiv 2 \pmod{7}$$

1. $M = 3 \times 5 \times 7 = 105$
2. $M_1 = 105 / 3 = 35, \quad M_2 = 105 / 5 = 21, \quad M_3 = 105 / 7 = 15$
3. The inverses are $M_1^{-1} = 2, M_2^{-1} = 1, M_3^{-1} = 1$
4. $x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \bmod 105 = 23 \bmod 105$

Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.

Solution

This is a CRT problem. We can form three equations and solve them to find the value of x.

$$x = 3 \bmod 7$$
$$x = 3 \bmod 13$$
$$x = 0 \bmod 12$$

Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.

Solution

This is a CRT problem. We can form three equations and solve them to find the value of x.

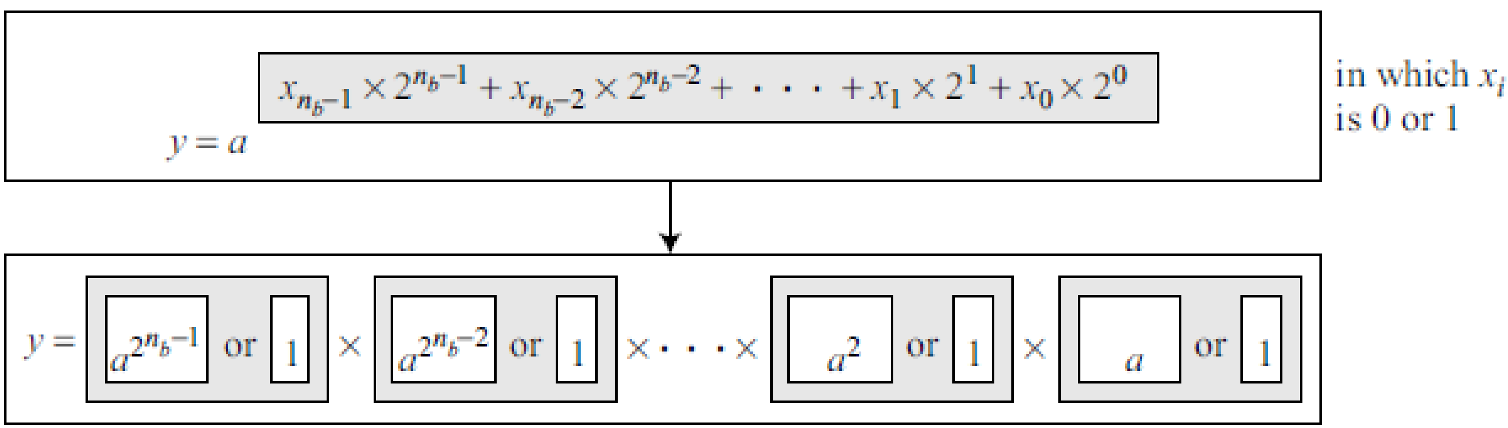
$$x = 3 \bmod 7$$
$$x = 3 \bmod 13$$
$$x = 0 \bmod 12$$

If we follow the four steps, we find $x = 276$. We can check that $276 = 3 \bmod 7$, $276 = 3 \bmod 13$ and 276 is divisible by 12 (the quotient is 23 and the remainder is zero).

Fast Exponentiation

- treat the exponent as a binary number of n_b bits (x_0 to $x_{nb} - 1$)
- x = 22 = (10110)

Figure 9.6 *The idea behind the square-and-multiply method*



Example:

$$\boxed{y = a^9 = a^{1001_2} = a^8 \times 1 \times 1 \times a}$$

```
Square_and_Multiply (a, x, n)
{
    y ← 1
    for (i ← 0 to n_b − 1)                    // n_b is the number of bits in x
    {
        if (x_i = 1)   y ← a × y  mod n        // multiply only if the bit is 1

        a ← a^2 mod n                          // squaring is not needed in the last iteration
    }
    return y
}
```

Calculation of 17^{22} mod 21

i	x_i	*Multiplication (Initialization: $y = 1$)*		*Squaring (Initialization: $a = 17$)*
0	0		$\rightarrow$	$a = 17^2 \bmod 21 = 16$
1	1	$y = 1 \times 16 \bmod 21 = 16$	$\rightarrow$	$a = 16^2 \bmod 21 = 4$
2	1	$y = 16 \times 4 \bmod 21 = 1$	$\rightarrow$	$a = 4^2 \bmod 21 = 16$
3	0		$\rightarrow$	$a = 16^2 \bmod 21 = 4$
4	1	$y = 1 \times 4 \bmod 21 = 4$	$\rightarrow$	

References

- Behrouz A. Forouzan and Debdeep Mukhopadhyay – “Cryptography and Network Security”, McGraw Hill, 2nd Edition, 2008.
- William Stallings,“Cryptography And Network Security Principles And Practice”, Fifth Edition, Pearson Education, 2013