# Week 1

# Services, Mechanisms & Attacks

# Services

* methods to counter attacks
* replicate functions normally associated with physical documents, e.g. signatures

### CIA Triad

* confidentiality: can only be **accessed** by authentic users
* integrity: can only be **modified** by authentic users
* authenticity: data **origin** can be identified

### Other services

* non-repudiation: transmission cannot be denied by sender or receiver
* availability: assets are available to user when needed
* anonymity: records cannot be associated with a specific individual

## Attacks

* passive attacks:
* active attacks

# Types of attack

* Modification: 3rd party **alters** the original message
* Interruption: **prevents** the message from reaching the receiver.
* Fabrication: 3rd party **imitates** a sender and sends a message.
* Interception: 3rd party receives the message **as well as** the intended recipient.
* Replay: send the same one-time message **more than once**.

# Steganography

* alternative to encryption
* hides existence of a message
  + for text, pertinent message within a much larger one
    - e.g. Hello everyone! Life, politeness and new daily rights every while!
  + for audio/video, Least Significant Bit

# Underlying theories

# Number Theory

# Congruences and Modular Arithmetic

Where *a, b* and *n* > 0, *a* is congruent to *b* modulo *n* iff  
 *a – b­ = kn*, for some integer *k*. Therefore *n* / (*a-b*).  
 e.g. 17 ≡ 7 mod 5, because (17-7) = 2\*5

* if *a* ≡*n* *b*, then *b* is called a residue of *a* modulo *n* (and *a* is a residue of *b* modulo *n*).
* A set of integers {r1, r2, … rn} is called a complete set of residues if there is exactly **ONE *ri*** for each integer *a*.
* *a op b* mod *n* = [(*a* mod n) *op* (*b* mod *n*)] mod *n*, where *op* = +, - or \*
  + (135273 + 261909 + 522044) mod 9 = 2  
    135273 mod 9 = 3; 261909 mod 9 = 0; 522044 mod 9 = 8  
    3+0+8 mod 9 = 11 mod 9 = 2

# Fast Exponentiation

For *ab* mod *c = x*:  
 while *b* > 0  
 while *b* is even  
 *b* = *b*/2  
 *a* = *a*\**a* mod *c  
 b* = *b*-1  
 x = *x*\**a* mod *c*  
 return x

*x* = 310 mod 5  
 = (32)5 mod 5  
 = 95 mod 5  
 = 45 mod 5  
 = 4 \* 44 mod 5  
 = 4 \* 163 mod 5  
 = 4 \* 13 mod 5  
 = 4

# Additive Inverse (-w)

For each w ϵ Zn, there exists a z such that w + z ≡ 0 mod n

# Information Theory

# Complexity Theory

# Groups

Group *G* is a set of elements with a binary operation ● that associates each ordered pair (a,b) of elements in *G* to an element *a* ● *b* in *G*, for the following:

* Closure – if *a* and *b* belong to *G*, so does *a ● b*
* Association – *(a ● b) ● c* = *a ● (b ● c)* for all *a, b, c* in *G*
* Identity Element – there is an element *e* where *a* ● *e* = *e* ● *a* for all *a* in *G.*
* Inverse element – there is an element *a-1* in *G* where *a-1* ● *a* = *a* ● *a-1*

*G* is abelian if it is also:

* Commutative – *a* ● *b* = *b* ● *a* for all *a*, *b* in *G*

Groups are finite (all elements are the order of *G*) or infinite.

# Cyclic Group

Group *G* is cyclic if every element *k* in *G* is an integer *ak* of a fixed element. *a* is the **generator** of *G*. Cyclic groups are always abelian.

# Rings

A ring *R* is an abelian group where we can do addition, multiplication and subtraction without leaving the set. It has the following additional axioms:

* Closure under multiplication – if *a* and *b* belong to *R*, so does *ab*.
* Associativity of multiplication – *(ab)c* = *a(bc)* for all *a, b, c* in *R.*
* Distributive laws:
  + *a*(*b* + *c*) = *ab* + *ac* for all *a, b, c* in *R.*
  + (*a + b*)*c* = *ac* + *bc* for all *a, b, c* in *R.*

A ring is **commutative** if it has:

* Commutativity of multiplication – *ab* = *ba* for all *a, b* in *R*.

An **integral domain** is a commutative ring that has:

* Multiplicative identity – There is an element 1 in R where *a*\*1 = 1\**a* = *a* for all *a* in *R*
* No zero divisors – Where *a, b* in *R* and *ab* = 0, either *a* = 0 or *b* = 0.

# Fields

A field *F* is an integral domain with the following additional axiom:

* Multiplicative inverse – For each *a* ≠ 0 in *F*, there is an element *a-1* where *aa-1* = 1

# Chinese Remainder Theorem

# Week 2

# Euclid’s algorithm

* if *a* and *b* & *a* > *b* & *a* and *b* are divisible by *c*, *a mod b* is also divisible by *c*.
* write *a* = *kb* + *r*, where *k* is an integer and *r* ϵ [0, *b*-1]
* if *a* and *b* are divisible by c:
  + *ac = bc* + *r*
  + *r* = (*a*-*b*)\**c*
* so *r* is also divisible by *c*.

# Algorithm for computing GCD

* For example, 100 and 22 are divisible by 2, and 100 mod 22 = 12.
* Instead of finding the GCD of 100 and 22, look for 22 and 12, then 12 and 10, and so forth.
* When we hit “0”, the previous value if the GCD.
* 100, 22 // 100 – 22\*4 = 12   
  22, 12 // 22 – 12\*1 = 10  
  12, 10 // 12 – 10\*1 = 2  
  10, 2 // 2 – 2\*1 = 0  
  2, 0

# Extended Euclid’s Algorithm

ax mod n = 1  
i y g u v  
0 - n 0 1  
1 - a 1 0  
j g[j-2]/g[j-1] g[j-2] – y[j]\*g[j-1] u[j-2] – y[j]\*u[j-1] v[j-2] – y[j]\*v[j-1]

# Euler’s Totient Function

The number of integers <= n  
where n is prime, phi(n) = n-1  
where n = pq where p & q are primes, phi(n) = (p-1)x(q-1)  
where n = piq where q is prime and p is a prime to the power of i

# General Equations

For equations of type ax mod n = b:

* If gcd(a,n) = 1
  + 1 solution
    - x = bxo mod n
* If gcd(a,n) = g
  + If b mod g = 0
    - g solutions of form ((b/g)\*x0 + t(n/g)) mod n for t = 0,1,…,g-1
    - x0 = (a/g)x mod (n/g) = 1
  + Else
    - No solutions

# Galois Fields

GF(2n) with polynomial p(x) find d = a\*a

d = (a\*a) / p(x)

# Information Theory

Uncertainty of the receiver of a message through a noisy channel. Objective is to make message M recoverable from M’.

Secrecy of a message passed through an encryption channel. Objective is to make recovery of M from M’ infeasible.

# Entropy

-(sum(X1, …, Xn) \* log2(p(Xi))) = H(X), where X is the message and p(Xi) is the probability of the message being sent.

log2(1/p(Xi­)) = bits needed to encode message Xi, H(X) is weighted average.

event prob: p(a) = 0.5, p(b) = 0.2, p(c) = 0.3

-((0.5 \* log2(0.5)) + (0.3 \* log2(0.3)) + (0.2 \* log2(0.2)))  
= -(-0.5 - 0.5211 – 0.4644)  
= 1.4855

# Week 3

# Rate of Language

Rate r of language L is the average entropy per character N.

r = H(X) / N

Absolute rate: R = log2L, e.g. English is log226 = 4.7 (bits/letter)

If all sequences of characters in a language have equal probability, r = R

# Redundancy

For any natural language, the absolute rate is greatly higher than the actual rate

Redundancy D = R – r  
For English, 3.2 = 4.7 – 1.5  
D/R = 0.68, so English is 68% redundant (assuming rate of 1.5 bit/letter; 79% if 1 b/l)

# Equivocation

# Week 4

## Unicity Distance

The amount of ciphertext needed to break a permutation cipher of period d,   
[ N = H(K) / D = log2(d!) / D ]

Sterling’s approximation:

d! ~= (d/e)d sqrt(2\*pi\*d   
log2(d!) ~= d\*log2(d/e)  
N = ( d\*log2(d/e) ) / 3.2 = 0.3\*d\*log2(d/e)

# Week 5

# Week 6

# Week 7

## AES Evaluation criteria

Security: actual security (more than other submitted algorithms), randomness (how much the output differs from a random permutation), soundness (of the maths), other ()

Cost: licensing requirements (AES should be global, royalty-free, non-exclusive), computational efficiency (hardware and software speed), memory (gate counts, code size and RAM)

Algorithm and Implementation: flexibility, hardware and software suitability, simplicity

|  |  |  |  |
| --- | --- | --- | --- |
| Key size (bytes) | 16 | 24 | 32 |
| PT block size (bytes) | 16 | 16 | 16 |
| Rounds | 10 | 12 | 14 |
| Round key size (bytes) | 16 | 16 | 16 |
| Expanded key size (bytes) | 176 | 208 | 240 |

## Rijndael

[simplicity: clean, security: adequate, speed: fast, SP network cipher]

* 9/11/13 rounds
  + byte substitution (1 s-box per byte)
  + shift rows (permute bytes between columns)
  + mix columns (matrix multiplication)
  + add round key
  + initial XOR and incomplete last round

# Week 8

## Public-Key Cryptography

Each user A has enc proc EA which is public and dec proc DA which is private. E & D are easy to compute. D is not easy to compute from E.

Confidentiality and authenticity. Public-key can implement digital signatures and does not need secret key exchange.

## Signatures

(e,n) = (13,33) M = 30 C = 25 p = 3 q = 11

phi(33) = (3-1\*11-1) = 20 Me mod n = C

1. 3013 mod 33 =  
   30\*3012 mod 33 =  
   30\*(302)6 mod 33 =   
   30\*(900)6 mod 33 =  
   30\*(9)6 mod 33 =  
   30\*(92)3 mod 33 =  
   30\*(15)3 mod 33 =  
   30\*15\*225 mod 33  
   450\*225 mod 33 =  
   21\*27 mod 33 =  
   567 mod 33 = **6**
2. e \* d mod phi(n) = 1  
   13\*d mod 20 = 1  
   20 = 22 \* 5  
   13\*x mod 4 = 1 13\*x mod 5 = 1  
   x mod 4 = 1 3\*x mod 5 = 1  
   x = 1 x = 2  
   x mod 4 = 1 x mod 5 = 2  
   x1 = 1 x2 = 2  
   d1 = 4 d2 = 5  
     
   (n/di)\*yi mod di = 1   
   (20/4)\*yi mod 4 = 1 (20/5)\*y2 mod 5 = 1  
   5y1 mod 4 = 1 4y2 mod 5 = 1  
   y1 mod 4 = 1 y2 = 4  
   y1 = 1  
   d = (1\*1\*5 + 2\*4\*4) = 37 mod 20 = **17**
3. M = Cd mod n

= 2517 mod 33  
= 25\*2516 mod 33  
= 25\*(252)8 mod 33  
= 25\*(625)8 mod 33  
= 25\*(31)8 mod 33  
= 25\*(312)4 mod 33  
= 25\*(4)4 mod 33  
= 25\*(25) mod 33  
= 625 mod 33  
= 31

# Week 9

# Week 10

# Week 11

# Week 12