Teoria dei Numeri

- Aritmetica modulare
  - Usata per definire un campo finito
  - *a* = *b* mod *n* significa che se *a* e *b* sono divisi per *n* producono lo stesso resto
  - *a***b* mod *n* può ritornare 0 anche se *a* e *b* non sono 0
  - a/b mod n è calcolato come $a*b^{-1}$ mod n, dove b^{-1} è l'inverso di b, $b*b^{-1} = 1$ mod n

Teoria dei Numeri

- Interi modulo n con le operazioni di + e * formano un anello commutativo con le seguenti proprietà:
  - **Associatività**
    - (a+b)+c = a+(b+c) mod n (vale anche per *)
  - **Commutatività**
    - a+b = b+a mod n (vale anche per *)
  - **Distributività**
    - (a+b)*c = (a*c)+(b*c) mod n
  - **Identità additiva e moltiplicativa**
    - a+0 = a mod n and a*1= a mod n
  - **Inverso additivo**
    - a+(-a) = 0 mod n

Teoria dei Numeri

- Un ***CAMPO*** è un insieme che le stesse proprietà dell'anello commutativo più l'inverso moltiplicativo per ogni elemento, escluso lo 0
  - $a*a^{-1} = 1 \bmod n$, per ogni $a \neq 0$
- Si può scegliere se fare l'operazione e poi estrarne il modulo, oppure estrarne il modulo e poi effetutare l'operazione. (l'estrazione del modulo è un omomorfismo dall'insieme degli interi agli interi modulo *n*)
  - a op b mod n = [a mod n op b mod n] mod n
  - Dove op può essere +, -, oppure *

Teoria dei Numeri

- Se *n* è un numero primo *p* allora otteniamo un Campo di **Galois modulo p -GF(p)-** e valgono tutte le normali regole aritmetiche sugli interi
- **Esponenziazione in GF(p)**
  - Molti algoritmi di cifratura usano l'esponenziazione modulare – elevare un numero *a* (base) ad una potenza *b* (esponente) mod p
  - $b = a^e \bmod p$
  - l'esponenziazione si può vedere come composta di moltiplicazioni ripetute,

Teoria dei Numeri

- **Esponenziazione in GF(p)**
  - Un metodo migliore è l'algoritmo eleva al quadrato e moltiplica (square and multiply)

$$\text{Power}(x, n) = \begin{cases} x, & \text{if } n = 1 \\ \text{Power}(x^2, n/2), & \text{if } n \text{ is even} \\ x \times \text{Power}(x^2, (n-1)/2), & \text{if } n > 2 \text{ is odd} \end{cases}$$

  - Richiede solo $O(\log_2 n)$ moltiplicazioni per un numero n

Teoria dei Numeri

- **Logaritmo Discreto in GF(p)**
  - Il problema inverso dell'esponenziazione è quello di trovare il **logaritmo discreto** di un numero modulo p
  - trovare x tale che $a^x = b \bmod p$
  - l'esponenziazione è relativamente facile, mentre il **logaritmo discreto** è generalmente un problema difficile
  - Se p è primo, esiste sempre un *a* tale che $a^x = b \bmod p$ per b diverso da 0
    - Le potenze di *a* "generano" il gruppo mod p
  - Questo *a* è chiamato **radice primitiva** e sono anch'esse difficili da trovare

Algoritmo di Euclide

- L'algoritmo di Euclide è un algoritmo per trovare il g.c.d.(a,b) senza dover fattorizzare *a* e *b*.
  - Sia a>b
  - Troviamo un q_1 ed un r_1 tali che $a=b*q_1+r_1$
  - Troviamo un q_2 ed un r_2 tali che $b = q_2*r_1+r_2$
  - Troviamo q_i e gli r_i usando $r_{i-2}=q_i*r_{i-1}+r_i$ per i>2 finché $r_i=0$.
  - g.c.d.(a,b)=r_{i-1}
- L'algoritmo ha complessità $O(\log^3(a))$

Algoritmo di Euclide

- Esempio:
  - Trovare g.c.d.(1547, 560)
  - 1547 = 2*560+427
  - 560= 1*427 + 133
  - 427 = 3*133 + 28
  - 133 = 4*28 + 21
  - 28 = 1*21 + 7
  - 21 = 3*7 + 0
  - e quindi g.c.d(1547,560) = 7

Algoritmo di Euclide esteso

- Spesso si vuole trovare l'inverso di *a* ovvero a^{-1} tale che $a*a^{-1}=1 \bmod n$
- Utilizziamo la conoscenza che:
  - Se d = g.c.d.(a,b), dove *a>b*, allora esistono interi u e v tali che d = u*a + v*b.
  - Trovare u e v richiede $O(\log^3 a)$ passi
- E quindi utilizziamo l'algoritmo di Euclide esteso per trovare a^{-1}, assumendo che g.c.d.(a, n) = 1

Algoritmo di Euclide esteso

- L'algoritmo:
  - Inverso(a,n) è dato da:
  - g_0=n u_0=1 v_0=0
  - g_1=a u_1=0 v_1=1
  - poniamo
    - $y = g_{i-1} \text{ div } g_i$
    - $g_{i+1} = g_{i-1} - y*g_i = g_{i-1} \bmod g_i$
    - $u_{i+1} = u_{i-1} - y*u_i$
    - $v_{i+1} = v_{i-1} - y*v_i$
  - quando g_i=0 allora Inverso(a,n) = v_{i-1}

Algoritmo di Euclide esteso

- Esempio:
  - Trovare l'inverso di 3 mod 460

i	y	g	u	v
0	-	460	1	0
1	-	3	0	1
2	153	1	1	-153
3	3	0	-3	460

→ 3^{-1} mod 460 = -153 mod 460 = 307 mod 460

Funzione phi di Eulero

- La funzione φ di Eulero
  - $\varphi(n) = |\ (0 \leq b < n) \mid g.c.d.(b,n) = 1)\ |$
  - se p è primo allora $\varphi(p) = p-1$
  - $\varphi(m*n) = \varphi(m)* \varphi(n)$, se $g.c.d.(m,n) = 1$
  - $\varphi(p^{\alpha}) = p^{\alpha} - p^{\alpha-1}$

Piccolo Teorema di Fermat

- Sia p un primo; *qualsiasi* intero che soddisfi $a^p = a \bmod p$ e *qualsiasi* intero *a* non divisibile da p soddisfano:
  - $a^{p-1} = 1 \bmod p$
- Generalizzazione del piccolo teorema di Fermat, dovuta ad Eulero:
  - se g.c.d.(a, m) = 1, then $a^{\varphi(m)} = 1 \bmod m$

Algoritmi per trovare l'inverso

- Algoritmi per trovare

  Inverso di a mod n

  1. Provare 1,…,n-1 finchè si trova un a^{-1} tale che $a*a^{-1} = 1 \bmod n$
  2. Se $\varphi(n)$ è noto, allora dalla generalizzazione di Eulero:
     - $a^{-1} = a^{\varphi(n)-1} \bmod n$
  3. Altrimenti, usare l'algoritmo di Euclide esteso

Teorema Cinese del resto

- Motivazione:
  - Trovare un numero x tale che abbia resto 1 quando diviso per 3, resto 3 quando diviso per 5 e resto 3 quando diviso per 7. Ovvero, x=1 mod 3, x = 3 mod 5 ed x = 3 mod 7
- Vogliamo cioè risolvere un sistema di congruenze con moduli differenti

Teorema Cinese del resto

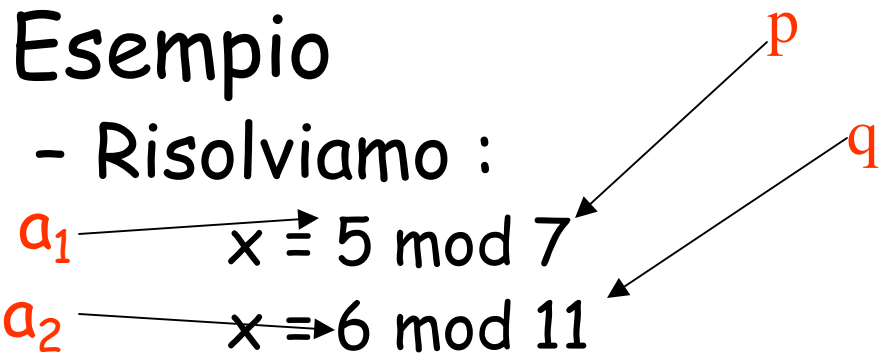
- Si consideri il sistema:
  - $x = a_1 \bmod m_1$
  - $x = a_2 \bmod m_2$
  - ….
  - $x = a_r \bmod m_r$
  - Assumiamo che g.c.d.$(m_i, m_j) = 1$ per $i \neq j$. il sistema ha un'unica soluzione modulo $M=m_1m_2…m_r$

Teorema Cinese del resto

- Il Toerema Cinese del resto fornisce un metodo per risolvere un'equazione modulo n, dove n=p*q e p e q sono primo, risolvendo separatamente equazioni modulo mod p e mod q
  - Consideriamo $b_1 = q^{-1} \bmod p$ e $b_2 = p^{-1} \bmod q$
  - se $a = a_1 b_1 q + a_2 b_2 p \bmod pq$ abbiamo che
    - $a = a_1 b_1 q + a_2 b_2 p = a_1 \bmod p$
    - $a = a_1 b_1 q + a_2 b_2 p = a_2 \bmod q$
  - Quindi, se conosciamo a_1 e b_1 conosciamo a

Teorema Cinese del resto

- Esempio
  - Risolviamo :



b_2 - $7^{-1} \bmod 11 = 8$

b_1 - $11^{-1} \bmod 7 = 2$

  - Così $a = 5 * 2 * 11 + 6 * 8 * 7 = 446$
  - $a = 61 \bmod 77$ è la soluzione per entrambe le equazioni

Sommario

- La **Teoria dei Numeri è Fondamentale** per il settore della sicurezza

- E' richiesta una conoscenza:
  - Analitica di tutte le sue branche per essere un crittografo (matematici i migliori crittografi);
  - In profondità, per lavorare a livello sistemico;
  - In profondità, per produrre soluzioni efficienti;