[03/01] Algoritmi vari

01 March 2018 10:41

$\exists x, y \mid : ax + yb = MCD(a, b)$

MCD(0,n)=n
MCD(m,n)=MCD(n mod(m), m)
Con $m < n$

Proprietà dell'aritmetica modulare:

- $(a \pm b) mod\ m \equiv [(a\ mod\ m) \pm (b\ mod\ m)](mod\ m)$
- $(a \cdot b) mod\ m \equiv [(a\ mod\ m) \cdot (b\ mod\ m)](mod\ m)$
- $[a \cdot (b + c)] mod\ m \equiv \{[(a \cdot b) mod\ m] + [(a \cdot c) mod\ m]\}(mod\ m)$

Potenze di numeri mod m:

$a^x mod\ m$

- $(a_1 \cdot a_2 \cdot \cdots \cdot a_x)\ mod\ m$
- $(a^{x-y} mod\ m)^y mod\ m; \quad y < x$

Algoritmo di Euclide esteso:

Questo algoritmo permette di generare i due interi $x, y \mid : ax + by = MCD(a, b)$
Assumendo $\boldsymbol{a \leq b}$

	$x_0 = 0; x_1 = 1$	$y_0 = 1; y_1 = 0$
$b = q_1 a + r$	$x_2 = -q_1 x_1 + x_0$	$y_2 = -q_1 y_1 + y_0$
$a = q_2 r_1 + r_2$	$x_3 = -q_1 x_2 + x_1$	…
$r_1 = q_3 r_2 + r_3$	…	…
…	…	…
$r_{k-2} = q_k r_{k-1} + r_k$	$\boldsymbol{x_{k+1} = -q_k x_k + x_{k-1}}$	$\boldsymbol{y_{k+1} = -q_k y_k + y_{k-1}}$
$r_{k-1} = q_{k-1} r_k + 0$	-	-

Una volta definita la tabella i due numerix_{k+1} e y_{k+1} sono i due coefficienti cercati e r_k = MCD(a,b).
Inoltre $\boldsymbol{x_{k+1} = a^{-1} (mod\ b)}$

Cosa faccio, mi arrampico?

Proprietà dell'operatore di congruenza

- $a \equiv 0\ (mod\ n) \Leftrightarrow n \backslash a\ (n\ divide\ a)$
- $a \equiv b\ (mod\ n) \equiv b + kn\ (mod\ n), k \in \mathbb{Z}$
- $a \equiv b \Leftrightarrow b \equiv a$
- $a \equiv b, b \equiv c \rightarrow a \equiv c$
- $a \equiv b, c \equiv d \rightarrow a \pm c \equiv b \pm d \ \wedge \ ac \equiv bd$
- $ab \equiv ac\ (mod\ n)\ se\ a \perp n \Rightarrow b \equiv c$

$2x + 7 \equiv 3(mod\ 17)$

Sommo -7 a sx e dx

$2x \equiv -4\ mod(17)$

Divido sx e dx per 2

$x \equiv -2 mod(17)$

Tutto ciò funziona ⇒ $ab \equiv ac\ (mod\ n)$; posso dividere per a solo se a è *primo relativo* rispetto a n:
Se $a \perp n \Rightarrow b \equiv c$

Es: $5x + 6 \equiv 13\ mod(17)$
$5x \equiv 7\ mod(17)$

Posso dividere per 5 perché $5 \perp 17$, ma non posso usare le frazioni
$5x = 7, 18, 29, 40, 51, \ldots$
Usando la terza proprietà
$5x \equiv 40\ mod\ (11)$, ora posso dividere per 5
$x \equiv 8\ mod(11)$, il valore 8 in $\mathbb{Z}_{11}$ si comporta come $\frac{7}{5}$

Numeri inversi:
Dato $a\ mod\ m$; $\quad a \neq 0$
l'inverso di tale numero è $x \in \mathbb{Z} \mid: (x \cdot a)\ mod\ m = 1$
Si può scrivere anche: $a^{-1} \equiv x\ (mod\ m)$, questa equazione ha una sola soluzione $\Leftrightarrow a \perp m$

$5^{-1} \equiv 9\ mod(11)$

$5^{-1}5x \equiv 7 * 5^{-1} mod(11)$
$x \equiv 7 * 9 \equiv 8$

Funzione toziente $\phi(n)$
L'insieme di tutti i residui $\mathcal{Z}_m$; $\quad (0, m-1)$ comprende tutti i numeri da 0 a m-1
quindi $|\mathcal{Z}_m| = m$
L'insieme ridotto
$\mathcal{Z}_m^* \in \mathcal{Z}_m$
Comprende i residui che sono primi rispetto ad m, ovvero
$p_i, 1 < i < \phi(m), \quad p_i \perp m$
La cardinalità dell'insieme ridotto è uguale alla funzione toziente:
$|\mathcal{Z}_m^*| = \phi(m)$

$*\ \phi(p) = p - 1 \Leftrightarrow p$ è $primo$
Comprende tutti i residui tranne lo 0.

Esempio:	Elementi di $Z_{10}^* = \{1, 3, 7, 9\} \rightarrow \phi(10) = 4$

(primi relativi di pedice, $MCD = 1$)

Se $MCD(a, n) = 1$, *ovvero* $a \perp n$; ricorda che $a < n$
$\exists\ s, t \mid: as + nt = 1$
E si possono trovare con Euclide esteso.
$as \equiv 1\ mod(n)$
In questo caso il coefficiente x_{k+1} è l'inverso di a.

- Eq. Congruenziali
- Inverso di un elemento di Z_p

Se $m = (p \cdot q)$; con p e q primi:
$\phi(m) = \phi(p \cdot q) = (p-1)(q-1)$

Esempio di prima facile

Soluzione di equazioni frazionarie:
Caso generale: $a \equiv b\ (mod\ n)$; $\quad MCD(a, b) = d > 1$:
- Se d non divide b ⇒ non esiste soluzione
- Se d divide b ⇒$\frac{a}{d}x \equiv \frac{b}{d}(mod\frac{n}{d})$

$*\ 2x \equiv -4\ mod(17)$ divido per 2
$12x \equiv 21 mod(39)$
Non posso dividere per 3 perché $3\ not \perp 39$
Se MCD(a,n) > 1
$ax \equiv b\ mod(n)$

1. Se $d\ X\ b\ (d\ non\ divide\ b) \Rightarrow \nexists\ soluzioni$
2. Se d/b → $\frac{a}{d}x \equiv \frac{b}{d} mod(\frac{n}{d})$; se questa eq. Ha una singola soluzione $x = x_0$, quella originale avrà soluzione $x_0, x_0 +$

$\frac{n}{d}, x_0 + 2\frac{n}{d}, \ldots + x_0 + \frac{(d-1)\,n}{d}$

$4x \equiv 7\ mod(13);\ \ x_0 = 5$
$x = 5, 18, 31$

Teorema cinese del resto

Di Sun Tzu (non quella dell'Arte della guerra)
In alcune condizioni da una coppia di congruenze se ne può combinare una singola.
Avendo: $x \perp n \Rightarrow \begin{cases} x \equiv a\ mod(n) \\ x \equiv b\ mod(m) \end{cases} \Leftrightarrow x \equiv c\ mod(n \cdot m)$

$\{x \equiv 25\ mod(42)$
42 è composto: $6(7k); 7(6k)$

$x \equiv 1\ \ mod(6)$
$x \equiv 4\ \ mod(7)\}$

Esempio:

$\begin{cases} x \equiv 3\ mod\ (7) \\ x \equiv 5\ mod(15) \end{cases} \Leftrightarrow x \equiv 80\ mod\ (106)$

Metodo più generale:

La seconda equazione dice che i valori accettabili sono $a + mk \equiv b\ mod(n)$; va risolta in k
$a - b \equiv mk\ mod(n)$;
$k = (a - b)m^{-1}\ mod(n)$

Esempio:

$\begin{cases} x \equiv 3\ mod(17) \\ x \equiv 5\ mod(15) \end{cases}\ 15^{-1} mod\ 7 = 1$
$k \equiv (3 - 5) mod\ 7 \equiv -2 \equiv -5$
$x \equiv 5 + 5 \cdot 15 = 80\ mod\ (7 \cdot 15)$

Altra versione dei fatti:
$b + nk \equiv a\ mod\ n$
$k \equiv (a - b)n^{-1} mod\ m$

Estensione a n congruenze del teorema cineso

$m_1, m_2, \ldots, m_k, m_N\ |: m_i \perp m_j; \forall i, j$

$\begin{cases} x \equiv a_1 mod\ m_1 \\ x \equiv a_2 mod\ m_2 \\ x \equiv a_N mod\ m_N \end{cases} \Leftrightarrow x \equiv A(mod \prod_{i=1}^{N} m_i$

Es:
$x^a mod\ n$

Algoritmo square&multiply:

Esempio pratico:
$7^{11} mod\ 26$
S&M divide 11 in binario: $11_{10} \rightarrow 1011_{bin}$ ovvero $2^3 + 2^1 + 2^0$
Quindi: $7^{2^3+2^1+2^0} = \left(7^{2^3}\right)\left(7^{2^1}\right)\left(7^{2^0}\right)$
1 $7^1 \equiv 7\ mod\ 26$
1 $7^2 \equiv 23\ mod\ 26$
0 $7^4 \equiv 23^2 \equiv 9\ mod\ 26$
1 $7^8 \equiv 9^2 \equiv 3\ mod\ 26$

$7^{11} \equiv 7 \cdot 23 \cdot 3\ mod\ 26$
$\equiv 15$

Piccolo teorema di Fermat

Prendo un primo p un numero a |: $a \perp p$, ovvero $p \not\chi a$ (non divide)

$a^{p-1} \equiv 1 \; mod \; p$

Non vale il viceversa.

Questo teorema si usa come test di primalità facile, dimostra che p non è primo. Se p è primo esce resto 1, se non è primo *potrebbe* uscire 1 lo stesso.

Ciò significa anche che:

$a^p \equiv a \; (mod \; p)$

Esempio:

$2^{1002} \; mod \; 11 =$

So che $2^{10} \equiv 1 \; mod \; 11$

Quindi: $2^{1002} \equiv 1^{100} \cdot 2^2 mod \; 11 \equiv 4 \; mod \; 11$ perché $2^{1002} = \left(2^{10}\right)^{100} \cdot 2^2$

Espressione di Eulero

Se $a \perp n \Rightarrow a^{\phi(n)} \equiv 1 \; mod \; n$

Come calcolo la funzione $\phi(n)$?

$$Dato \; n = \prod_{i=1}^{m} {p_i}^{r_i} \rightarrow \phi(n) = n \prod_{p_i \backslash n} \left(1 - \frac{1}{p_i}\right)$$

Es: $\phi(1000) = 1000(1 - \frac{1}{2})(1 - \frac{1}{5}) = 400$

Ci sono 400 numeri primi relativi rispetto a 1000

$$\phi(n) = \prod_{i=1}^{m} (p_i^{r_i} - p_i^{r_{i-1}})$$

Es: $\phi(1000) = \left(2^3 - 2^2\right)\left(5^3 - 5^2\right) = 4 \cdot 100$

Calcolo semplificato per esponenti elevati:

$x^y (mod \; n) = x^{y \; (mod \; \phi(n))} (mod \; n)$

ES: $7^{803} (mod \; 1000) =$

$7^{803 \; mod \; \phi(1000)} (mod \; 1000) =$

$7^{803 \; mod \; 400} (mod \; 1000) =$

$7^3 mod \; 1000 = 343$

[02/28] Introduzione

- Matematica dei numeri interi
- Concetto di divisibilità

Dati $a \neq 0, b$

$a \backslash b \Rightarrow \exists\, k \in \mathbb{Z} \mid : b = ka$

Ovvero a è un divisore di b se b è multiplo intero di a.

Numeri primi

$[\prod(x)]$è l'insieme dei numeri primi fino a x

$$\prod(x) \sim \frac{x}{\log x}, x \to \infty$$

- I numeri primi sono tutti dispari tranne 2

Funzione modulo:

$p \equiv 1\ mod\ 4$, ovvero tutti i numeri primi sono congruenti alla funzione $1 \pmod 4$

Dati $n > 0, a \in \mathbb{Z}$

$a\ (mod\ n) = r$, r è il resto della divisione $\frac{a}{n}$

quindi: $a = kn + r$

Scrivendo $a \equiv b\ (mod\ n)$

Ovvero $\begin{cases} a\ (mod\ n) = b\ (mod\ n) \\ a - b = kn \end{cases}$

Congruenza:

$b \equiv a\ (mod\ m)$

b ed a sono congruenti se hanno lo stesso resto quando vengono divisi per m

$Quindi\ se\ \begin{cases} a = k_1 m + r_1 \\ b = k_{2m} + r_2 \end{cases} con\ m > 0;\ \ b, a < 0\ o > 0$

Si ha congruenza se:

$r_1 = r_2 = r = b\ (mod\ m) = a\ (mod\ m)$

$(b - a)(mod\ m) = 0$

Tutti (o quasi) i numeri primi possono essere espressi come

$p = \pm 1\ (mod\ 6) \Rightarrow p = k \cdot 6 \pm 1$

Es: 5,7,11,13,17,23,~~25,~~29,31, …

Massimo Comun Divisore

MCD(a,b) = d

Se $MCD(a, b) = 1 \Rightarrow a \perp b$, ovvero a e b sono primi relativi

Algoritmo di Euclide

Per la fattorizzazione dei numeri

Dato $MCD(m, n), con\ m < n$

$MCD(m, n) = MCD(n\ (mod\ m), m)$

Applico tale formula iterativamente fino ad arrivare a:

$MCD(0, k) = k$

Esempio:

1.

MCD(482, 1180);	$2 \cdot 482 + 216$
MCD(216, 482)	$2 \cdot 216 + 50$
MCD(50, 216)	$4 \cdot 50 + 16$
MCD(16, 50)	$3 \cdot 16 + 2$
MCD(2, 16) = 2	*Resto 0*

L'ultimo resto diverso da 0 (2) è l'MCD.

Si può esprimere l'MCD(a,b) come combinazione lineare di a e b:

$$MCD(a, b) = x \cdot a + y \cdot b; \quad x, y \in \mathbb{Z}$$

[03/07] Esercizi aritmod

07 March 2018 10:24

Ay lmao

Riprendo il teorema di Fermat
Test di primalità semplice, funziona *molto spesso*, se il risultato è un residuo diverso da uno il numero testato è composto.
561 è il primo numero a passare il test senza essere primo.
$2^{560}(mod\ 561) = 1$
561 è uno *pseudoprimo di fermat.*

Se moltiplico la formula a sx e dx per a ottengo:
$a^p \equiv a\ (mod\ p)$
Quindi elevando un numero a per p mod p ottengo a
$(a \cdot a^{p-2})(mod\ p) = 1$
Perciò $a^{p-2} \equiv a^{-1}(mod\ p)$

Teorema di Lagrange (o Eulero)
$a^{\phi(n)} \equiv 1(mod\ n);\quad se\ a \perp n$

Calcolo l'inverso di a mod n
$a^{-1} = a^{\phi(n)-1}(mod\ n)$

ES: inverso di 50 mod 100
Solo che 50 non è primo relativo di 100
Altro es: 51 mod 100
$51^{-1}(mod\ 100) = 51^{\phi(100)-1}$
$\phi(100) = (2^2 - 2)(5^2 - 5) = 40$
$51^{39}(mod\ 100)$
Altro es: $27^{32768}(mod\ 65536)$
$32368 = 2^{15}$
La $\phi(2^{16}) = 2^{15}$
$27^{32768}(mod\ 65536) = 1$

Al posto di lavorare (mod n), lavoro mod $\phi(n)$ sugli esponenti
Considero 4 interi $a, n, x, y;\quad a \perp n$
$x \equiv y(mod\ \phi(n)) \Rightarrow a^x \equiv a^y(mod\ n)$
Perché $x = y + k\phi(n)$

Non è possibile risolvere x in $51^x = 2\ (mod\ 101)$

Esercizi:

1) Algoritmo di Euclide esteso
   $a^{-1}(mod\ n)$
   $MCD(a, n)$
   $E.E. \sim O(\log n)$ la complessità di questo algoritmo è nettamente inferiore
   Numero medio di divisioni: $0.843 \cdot \ln(n) + 1.47$
   l'EE è nel caso peggiore quando come argomenti si hanno due numeri consecutivi della sequenza di Fibonacci

   $28x \equiv 16(mod\ 412)$

MCD(28,412)=4

$7x \equiv 4 (\text{mod } 103)$
Devo trovare $7^{-1}(mod\ 103)$, uso Euclide esteso
$7^1 \equiv 7^{103-2}(mod\ 103)$ perché 103 è primo quindi p-2

i. MCD(7,103)
ii. $103 = 14 \cdot 7 + 5$
$b = q_1 a + r_1$
iii. $7 = 1 \cdot 5 + 2$
iv. $5 = 2 \cdot 2 + 1$
v. $2 = 2 \cdot 1 + 0$

Devo costruire la sequenza degli x
i. $x_1 = 1; x_0 = 0$
ii. $x_2 = -q_1 x_1 + x_0 = -14$
iii. $x_3 = -q_2 x_2 - x_1 = -1(-14) + 1 = 15$
iv. $x_4 = -q_3 x_3 + x_2 = -2 \cdot 15 - 14 = -44$ (corrisponde con la riga a resto 0)

Quindi: $7^{-1} \equiv -44(mod\ 103) \equiv 59$
$x \equiv 4 \cdot 7^{-1}(mod\ 59) \equiv 4 \cdot 59\ (mod\ 59)$
$\equiv 30$

$x_i = 30,133,236,339$
$(mod\ 412)$ queste sono le soluzioni dell'equazione congruenziale originale

2) Esercizio con teorema cinese del resto
Trovare la congruenza equivalente a
$x \equiv 5(mod\ 11)$
E
$x \equiv 2(mod\ 20)$
Condizione $11 \perp 20$ soddisfatta
La prima eq. Dice che $x = 5 + k11 \equiv 2(mod\ 20)$
$k11 \equiv -3(mod\ 20)$
Devo trovare $11^{-1} \equiv 11^7(mod\ 20) \equiv 11$
Quindi 11 è inverso di se stesso in $\mathbb{Z}_{20}$ si dice elemento pivot
$k \equiv -33(mod\ 20) \equiv (-33 + 20 + 20)(mod\ 20) \equiv 7(mod\ 20)$
$x \equiv 5 + 77 \equiv 82\ (mod\ 220)$

3) Tema d'esame 2015
Soldati in fila su 10 colonne, ne avanza uno, li mette su 11 e ne avanzano due
$x \equiv 1(mod\ 10) \rightarrow x = 1 + k10 \equiv 2(mod\ 11)$
$x \equiv 2(mod\ 11)$
$x \equiv 0(mod\ 3)$
$k10 \equiv 1(mod\ 11)$
$k = 10$
$x \equiv 101(mod\ 110)$ (fusione delle prime due)
$x = 101 + k110 \equiv 0(mod\ 3)$
$k2 \equiv -2(mod\ 3)$
$2k \equiv 1(mod\ 3)$
$k \equiv 2$
$x \equiv 101 + 220\ (mod\ 330)$
$x \equiv 321(mod\ 330)$

[03/14] Cifrari primitivi

14 March 2018 10:19

Sistemi di crittografia a chiave simmetrica

Alice scrive messaggio in chiaro, viene criptato con chiave di cifratura k, si ha E_k che andrà decifrato D_k

Manca disegno

Il canale di trasferimento della chiave deve essere un canale sicuro poiché ci può essere *Eva o Oscar* ad ascoltare i messaggi o a modificarli. Ci sono anche algoritmi di Key Agreement (le due parti costruiscono una chiave insieme nota solo ad esse).
Si ha un insieme dei messaggi in chiaro $\mathbb{P} \in \mathbb{Z}_{26}$ (26 lettere dell'alfabeto), insieme dei testi cifrati $\mathbb{C}$, insieme di funzioni di cifratura e decifratura K. Decifrando con K quello che esce dall'alg di cifratura di $\mathbb{P}$ si dovrebbe riottenere P. La funzione E_k deve *essere iniettiva, ovver* $E_k(P_1) \neq E_k(P_2); P_1 \neq P_2$

Cifrario a scorrimento (di Cesare, shift cypher)
Si può formalizzare come:
$\mathbb{P} = \mathbb{C} = K = \mathbb{Z}_{26}$
Il cifrario è definito come:
1. $c = E_k(P) = (P + K)(mod\ 26)$
2. $p = (c - k)(mod\ 26)$

Con la brute force ci vogliono 26 prove per decifrare il codice.

Cifrario affine:

$c = (aP + 6) mod\ 26$	$k \in \mathbb{Z}_{26} \times$

$|K| = 26 \cdot \phi(26)$
Non rispettando la condizione $a \perp n$, non si ha una funzione di cifrature iniettiva (non permette di tornare indietro ad un'unica parola).
L'attacco testo in chiaro noto è basato sulla conoscenza di una coppia testoInChiaroNoto-testoCifrato

$P_1, C_1; P_2, C_2$

$$\begin{cases} C_1 \equiv aP_1 + b(mod\ n) \\ C_2 \equiv aP_2 + b(mod\ n) \end{cases}$$

i. Faccio la sottrazione membro a membro per togliere b (dopo aver messo a sistema)
ii. $(C_1 - C_2) \equiv a(P_1 - P_2) mod\ n$
iii. Devo trovare l'inverso di $(P_1 - P_2) mod\ n$ (esiste? Non è detto, se non c'è inverso sottraggo la prima alla seconda, potrei non avere fortuna neanche in questo caso.)

Cifrario a sostituzione
Generalizzazione del cifrario di Cesare
Si ha una tabella che associa ad ogni carattere in chiaro un carattere cifrato.
$\mathbb{P} = \mathbb{C} = \mathbb{Z}_{26}$
La chiave è una permutazione dell'alfabeto
$K = \pi$

a	F
b	C
c	Z
d	D

La cardinalità dell'insieme è $|K| = 26!$

Approssimazione del fattoriale
$m! \cong \sqrt{2\pi m}\left(\frac{m}{e}\right)^m$
Sapendo che il testo è in una lingua nota si sa che non tutti i caratteri non equiprobabili.
Es: in inglese il 12% di caratteri nei testi in inglese sono *e (ciò permette di analizzare il testo dal punto di vista della frequenza di apparizione dei caratteri)*
Questo sistema lavora a blocchi di 8 bit. È un cifrario monoalfabetico perché ogni carattere viene cifrato in un solo modo.

Blocchi vettori di n caratteri: Codifica di Blaise de Vigènere

$\mathbb{P} = \mathbb{C} = K = (\mathbb{Z}_{26})^n$
La chiave è un vettore:
$\vec{k} = (k_1, k_2, \dots, k_n)$
È costituita da un vettore di n scorrimenti.
$\vec{P} = (P_1, \dots, p_n)$
$\vec{C} = (c_1, \dots, c_n)$

$\vec{P} + \vec{K}$
È un cifrario polialfabetico.
26^n chiavi, per l'analisi delle frequenze (n non troppo elevato) bisogna separare gli alfabeti (rispetto alla posizione nel blocco). Problema: non conosco n né l'allineamento dei blocchi. LPT: faccio l'autocorrelazione per vedere quanto frequente è ciascun punto. Non è un vero e proprio cifrario a blocchi perché si lavora carattere per carattere.

ADFGX
In morse i caratteri ADFGX sono molto diversi per tutelarsi dagli errori di trasmissione.

Cifrario a permutazione
Prende un blocco di n caratteri e li permuta (es.
$P = C = (\mathbb{Z}_{26})^n$
La chiave è una permutazione degli n caratteri
$K\ permutazione\ di\ \{1,2, \dots, n\}$

Es: n=6

1	2	3	4	5	6
3	5	1	6	4	2

Si tratta di permutare le posizioni dei caratteri nel blocco.

SHESEL	LSSEAS	HELLSx
EESLSH		

$|\mathcal{K}| = n!$
Ogni carattere può diventare uno qualsiasi degli n caratteri del blocco.

La cosa più importante della giornata (11.51):

Partiamo da LVI (Shannon), 1949, introdusse due concetti fondamentali per la crittografia:

i. Confusione: ha a che fare con la chiave, tutti i bit della chiave vanno ad influenzare tutti i bit del blocco cifrato.
Cambiando un bit della chiave cambiano tutti i bit del blocco cifrato con $p = \frac{1}{2}$. Relazione $uno \Leftrightarrow tutti$

ii. Diffusione: (modo inurbano: cambiando un bit nel blocco in chiaro cambia una l'intero blocco cifrato)
Cambiando un bit nel blocco in chiaro, tutto il blocco cifrato cambia, ogni carattere con $p = \frac{1}{2}$, ciò vale anche al contrario.

Tutti i cifrari moderni più usati godono di entrambe le proprietà.

[03/08] Primitivi, sqrt, residui quadratici

Thursday, March 8, 2018 10:33

GRUPPO

Il gruppo è una struttura algebrica formata da un insieme ed un'operazione binaria che soddisfa gli assiom di:

- associatività
- elemento neutro
- elemento inverso
- composizione

ELEMENTO PRIMITIVO GENERATIVO

L'ordine di un elemento di un insieme $\mathbb{Z}_p^*$ è il minor numero intero n tale che $\alpha^n \equiv 1 \pmod{p}$.

$$\alpha \in \mathbb{Z}_p^*, \qquad \mathrm{Ord}(\alpha) = \text{minor } n > 0 : \alpha^n \equiv 1 \pmod{p}$$

Nel caso in cui $\mathrm{Ord}(\alpha) = p - 1$, α è un **elemento primitivo generativo** di $\mathbb{Z}_p^*$. Elevando un elemento primo generativo di $\mathbb{Z}_p^*$ a potenza si ottengono tutti i numeri di $\mathbb{Z}_p^*$:

$$\boldsymbol{\alpha}\textbf{ elemento primitivo generativo} \Rightarrow \mathbb{Z}_{\boldsymbol{p}}^* = \{\boldsymbol{\alpha^i} : \boldsymbol{i} = \mathbf{1}, \dots, \boldsymbol{p} - \mathbf{1}\}$$

Il numero di elementi primi di $\mathbb{Z}_p^*$ è $\phi(p-1)$.

ES:

$p = 7, \qquad \alpha = 3$
$\beta \equiv \alpha^i, i = 1, \dots, p - 1$

$3^0 \equiv 1 \pmod{7}$
$3^1 \equiv 3 \pmod{7}$
$3^2 \equiv 2$
$3^3 \equiv 6$
$3^4 \equiv 4$
$3^5 \equiv 5$
$3^6 \equiv 1$ (ricomincia)
$3^7 \equiv 3$

$\mathrm{Ord}(\alpha) = 6 = p - 1 \Rightarrow \alpha$ è un elemento primitivo generativo di $\mathbb{Z}_p^*$.

Una **radice primitiva** di $\mathbb{Z}_n^i$ è un numero β tale che:

$$\beta \in \mathbb{Z}_n^*, \beta = \alpha^i : \mathrm{MCD}\left(i,\ \phi(p)\right) = 1 \Rightarrow \beta \text{ è radice primitiva di } \mathbb{Z}_n^*$$

Test per elemento primitivo

$$p-1=\prod_{i=1}^{n} q_i^{r_i}, (q_i \text{ primo})$$

$\alpha \in \mathbb{Z}_p^*$ è primitivo per $\mathbb{Z}_p^*$ $\Leftrightarrow \alpha^{\frac{p-1}{q_i}} \not\equiv 1 \pmod p$

La relazione degli esponenti:

$$\begin{cases} \text{a} \perp \text{n} \\ x \equiv y \pmod{\phi(n)} \end{cases} \Rightarrow a^x \equiv a^y \pmod n$$

diventa una doppia implicazione se a è un elemento numero primitivo di $\mathbb{Z}_n^*$.

ES: Trovare ed elencare in ordine crescente gli elementi primitivi di $\mathbb{Z}_{19}^*$.

$\phi(18) = 6$
$18 = 2 \cdot 3^2$

2:
$2^{\frac{18}{2}} = 2^9 \equiv 18 \pmod{19}$
$2^{\frac{18}{3}} = 2^6 \equiv 7 \pmod{19}$
$\Rightarrow$ 2 è un elemento primitivo

3:
$3^9 \equiv 18 \pmod{19}$
$3^6 \equiv 7 \pmod{19}$
$\Rightarrow$ 3 è un elemento primitivo

4:
$4^9 \equiv 1 \pmod{19}$
$\Rightarrow$ 4 **non** è un elemento primitivo

...eccetera...

L'equazione generale $a^x \equiv b \pmod p$ ha sicuramente soluzione se a è un elemento primitivo di $\mathbb{Z}_p^*$.

Questa equazione è infatti usata in alcuni algoritmi di crittografia a chiave pubblica e privata dove a, b, p sono pubblici ed x è privato. Risolvere tale equazione vorrebbe dire calcolare $x = \log_a^D(b) \pmod{\phi(p)}$, ma il logaritmo discreto ($\log_a^D b$) non è calcolabile se non utilizzando la forza bruta, e ciò rende la chiave privata impossibile da scoprire se p è abbastanza grande.

MODULO DELLA RADICE QUADRATA

Tutti i numeri primi come sappiamo sono congruenti o a 1 o a 3 modulo 4. Nel secondo caso, ovvero se $p \equiv 3 \pmod 4$:

$$x^2 \equiv a \pmod p \Rightarrow x \equiv \pm a^{\frac{p+1}{4}} \pmod p$$

$\pm a^{\frac{p+1}{4}}$ potrebbe però essere $\sqrt{a}$ oppure $\sqrt{-a}$, quindi non siamo totalmente sicuri che sia giusto. Per verificare ciò sfruttiamo questa relazione:

$$a^{\frac{p-1}{2}} \equiv +1 \pmod p \Rightarrow \exists \sqrt{a}$$
$$a^{\frac{p-1}{2}} \equiv -1 \pmod p \Rightarrow \exists \sqrt{-a}$$

Nel caso di primi $p \equiv 3$ **(mod 4) esiste sempre una sola radice tra** $\sqrt{a}$ **e** $\sqrt{-a}$.
Nel caso di primi $p \equiv 1$ **(mod 4) o esistono entrambe** $\sqrt{a}$ **e** $\sqrt{-a}$ **o nessuna delle due**.

Nel caso di numeri composti è possibile scomporli in fattori primi e fare il calcolo delle radici per le "sotto"-equazioni derivate da ognuno di essi, cercando poi le soluzioni all'equazione originale usando il teorema cinese del resto.

ES: $x^2 \equiv 71 \pmod{77}$ $\quad 77 = 7 \cdot 11$

$x^2 \equiv 1 \pmod 7 \rightarrow x \equiv \pm 1 \pmod 7$
$x^2 \equiv 5 \pmod{11} \rightarrow x \equiv \pm 5^3 \pmod{11} \equiv \pm 4 \pmod{11}$ → soluzioni

Teo cinese:

1) $\begin{cases} x \equiv 1 \ (\%7) \\ x \equiv 4 \ (\%11) \end{cases}$ 2) $\begin{cases} x \equiv -1 \ (\%7) \\ x \equiv 4 \ (\%11) \end{cases}$ 3) $\begin{cases} x \equiv 1 \ (\%7) \\ x \equiv -4 \ (\%11) \end{cases}$ 4) $\begin{cases} x \equiv -1 \ (\%7) \\ x \equiv -4 \ (\%11) \end{cases}$

1) $x = 1 + 7k \equiv 4 \ (\%11)$ $\quad = 8 \ (\%11)$
$k \equiv 3 \cdot 7^{-1} \equiv 3 \cdot 7^9 \equiv 2 \ (\%11)$
$\Rightarrow x \equiv 15 \ (\%77)$

2) $x = -1 + 7k \equiv 4 \ (\%11)$
$k \equiv 5 \cdot 7^{-1} \equiv 5 \cdot 7^9 \equiv 40 \equiv 7 \ (\%11)$
$\Rightarrow x \equiv 48 \ (\%77)$
$\equiv -29 \ (\%77)$

3) $x = 1 + 7k \equiv -4 \ (\%11)$
$k \equiv -5 \cdot 7^{-1} \equiv -5 \cdot 7^9 \equiv -5 \cdot 8 \equiv -40 \equiv -7 \equiv +4 \ (\%11)$
$\Rightarrow x \equiv 1 + 7 \cdot 4 \equiv 29 \ (\%77)$ opposta della precedente!

4) Sarà l'opposta della prima: $x \equiv -15 \ (\%77) \equiv 62 \ (\%77)$

$x \equiv \begin{cases} \pm 15 \ (\%77) & x_0 = 15 \quad x_2 = 48 \\ \pm 29 \ (\%77) & x_1 = 29 \quad x_3 = 62 \end{cases}$

RESIDUI QUADRATICI

$a \in \mathbb{Z}_p^*: a \equiv b^2 \pmod p, b \in \mathbb{Z}_p^*$

$a_q \equiv (\pm b)^2 \pmod p, b = 1, \dots, \frac{p-1}{2}$

Trovare i residui quadratici

Primo metodo: provo tutte le b da 1 a $\frac{p-1}{2}$.

ES:

$p = 11$

$\mathbb{Z}_{11}^* = \{1, 2, 3, \dots, 10\}$

b	$b^2 \pmod{11}$
1	1
2	4
3	9
4	5
$5 \left(= \frac{p-1}{2}\right)$	3

$a_q = \{1, 3, 4, 5, 9\}$

Secondo metodo (B O H):

ES:

$p = 13$ (WARNING $13 \equiv 1 \pmod 4$... just don't)

$\mathbb{Z}_{13}^* = \{1, 2, 3, \dots, 12\}$

$\alpha = 2 \ (\text{or})$

$\alpha^1 \equiv 2$
$\alpha^2 \equiv 4$
$\alpha^3 \equiv 8$
$\alpha^4 \equiv 3$
$\alpha^5 \equiv 6$
$\alpha^6 \equiv 12$

$\alpha^7 \equiv 11$
$\alpha^8 \equiv 9$
$\alpha^9 \equiv 5$
$\alpha^{10} \equiv 10$
$\alpha^{11} \equiv 7$
$\alpha^{12} \equiv 1$

Residui quadratici:

$a_q = \{1, 3, 4, 9, 10, 12\}$

Terzo metodo: provo tutti gli elementi di $\mathbb{Z}_p^*$ vedendo se verificano $a^{\frac{p-1}{2}} \equiv +1 \pmod p$.

ES:

$p = 11 \qquad \mathbb{Z}_{11}^* = \{1 \dots 11\}$

$\frac{p-1}{2} = 5 \qquad a^5 \pmod{11}$

$1^5 \quad 1 \qquad 6^5 \equiv$

$2^5 \equiv -1$ $\quad -1$

$3^5 \equiv 1$ $\quad -1$

$4^5 \equiv 1$ $\quad 1$

$5^5 \equiv 1$ $\quad -1$

$a_9 = \{1, 3, 4, 5, 9\}$

[03/15] Cifrari pr0

15 March 2018 10:30

Do you even cypher

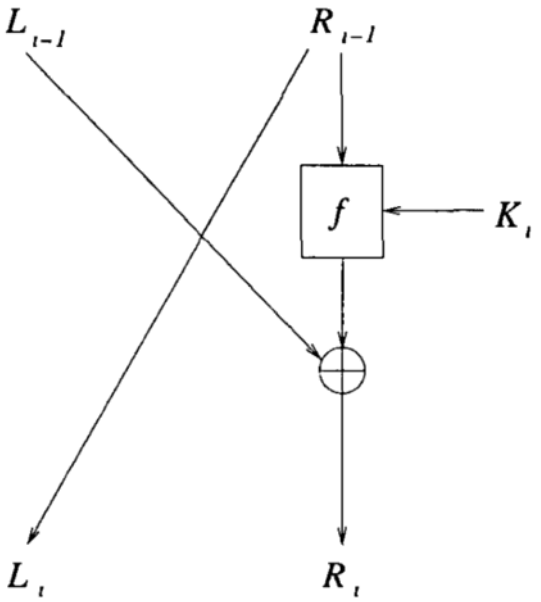
DES (Data Encryption System)

Con blocchi grandi (64bit) diventa difficile/impossibile crearsi un dizionario di parole chiaro/cifrato, capita però che ci siano messaggi che si ripetono (negativo).
Chiave DES originale 56 $bits \rightarrow 2^{56} \approx 7.2 \cdot 10^{16}$
(Considerato sicuro nei 70s)
Es: $2^{746} \approx 10^{0.3 \cdot 746} = 10^{224}$

Criptoanalisi differenziale (1990, metodo matematico per trovare la chiave, basata sul cifrare due blocchi diversi e vedere la differenza tra i blocchi criptati e quella in chiaro).
DES è il primo cifrario a blocco (64bit), opera con ECB ($64bit\ P \rightarrow 64bit\ C$). La chiave è apparentemente di 64bit, ma 8 sono inutili (=56 bits). È basato sull'applicazione di 16 rounds (stadi di elaborazione a cascata). Ovviamente gode delle proprietà di confusione e diffusione. La permutazione è *pseudocasuale*, i numeri vengono generati in maniera apparentemente casuale.

Feistel Cipher

l'uscita del primo stadio non gode di diffusione (cambia solo una delle due metà della parola). Il DES utilizza in cascata 16 round di questo cifrario, ciascun round prende una parte di chiave per cifrare metà parola per creare confusione.



La decifrazione parte da R_n e L_n ovvero l'uscita dell'n-esimo stadio.

1a uscita:

R_n	L_n
L_n	$R_n \oplus f(L_n; K_n)$
R_{n-1}	$L_{n-1} \oplus f(R_{n-1}; K_n) \oplus f(L_n; K_n)$

Ogni stadio usa una chiave di 48 bits presi dai 56 della chiave originale in modo pseudocasuale (confusione).

Modi di operare:
Electronic Code Book (ECB), blocchi da 64bits
Encrypt:

- $C_i = E_k(P_i)$

Decrypt:

- $P_i = D_k(C_i)$

I blocchi ripetuti sono evidenti
Si possono inserire blocchi a caso per disturbare la comunicazione

Cipher Block Chaining (CBC)
En: $C_i = E_k(P_i \oplus C_{i-1}); IV = C_0$
DE: $P_i = C_{i-1} \oplus D_k(C_i); IV = C_0$
IV è casuale e inviato in chiaro (vettore di inizializzazione)
Vantaggio: mancanza di ripetizione

Cipher Feedback Mode (CFB)
EN: $C_i = P_i \oplus E_k(C_{i-1}); IV = C_0$
DE: $P_i = C_i \oplus E_k(C_{i-1}); IV = C_0$
La decifrazione non chiama la funzione di decriptazione (generalmente più lenta di quella di cifratura)
Un eventuale errore di trasmissione si propaga solo in P_i, P_{i+1}
La sequenza è senza memoria (Autocorrelazione impulso in 0)

CFB: 8-bit mode

Output feedback mode (OFB)
Sommando mod 2 una sequenza pseudocasuale permette di avere un livello di sicurezza maggiore, posto che anche il decifratore abbia la stessa sequenza. Gli errori in C_i non si propagano.

CounTeR Mode (CTR)
Metodo alternativo di generare la sequenza pseudocasuale da sommare ai blocchi in chiaro (prima veniva creata sommando più volte in sequenza il vettore di inizializzazione), cripto l'IV prima di generare la sequenza PR.

[03/21] Doppio e triplo DES

21 March 2018 10:10

Il DES.
Doppio DES, cifro due volte a cascata, chiave totale 112bit. Può esistere una chiave singola tale che possa decifrare sia la prima che la seconda volta (non col DES). Devo essere sicuro che fare double DES mi raddoppi la lunghezza della chiave, esiste però l'attacco Meet-in-the-Middle che riduce lo spazio delle chiavi a 2^{57}. Soluzione: Triplo DES

$$C = E_{K1}\left(E_{K2}\left(E_{K3}(P)\right)\right)$$

Con il MiiM lo spazio delle chiavi si riduce a ben 2^{112}
Posso usare il 3DES con solo due chiavi

$$C = E_{K1}\left(D_{K2}\left(E_{K1}(P)\right)\right)$$

Se $K_1 = K_2$ il 3DES diventa DES
Il MiiM non ne riduce lo spazio delle chiavi $\mathcal{K} = 2^{112}$

MiiM Attack (2DES)

È un attacco testo-in-chiaro-noto (chi attacca conosce una coppia $P \leftrightarrow C$)
Conosco $P\ e\ C = E_{K2}\left(E_{K1}(P)\right)$
Devo trovare $K_1\ e\ K_2$
Possibili valori per ciascuna chiave 2^{56}
Calcolo $N = 2^{56}$ valori di $E_{K1}(P)$ (quindi per ogni chiave K_1 possibile)
Calcolo " per ogni K_2 $D_{K2}(C)$
Confronto ogni $D_{K2}(C)$ con tutti gli $E_{K1}(P)$
Ci deve essere almeno una corrispondenza
Ci sono 2N encrypt/decrypt ($+N^2$ confronti, $N \cdot 8$ bytes)
Lo spazio delle chiavi è ridotto da $2^{112} \rightarrow 2^{57}$

Sicurezza delle password

Le password ovviamente non si memorizzano in chiaro.
Introduzione concetti di digest e funzione unidirezionale.

One-Way Function

f(x) deve essere one-way e *lenta,* in modo da ostacolare eventuale bot/script.

- **Dictionary attack**
  Creo un dizionario di parole usuali aggiungendo leggere variazioni alfanumeriche e si provano con f(x) e confronto l'hash
  - Prevenzione: aggiungere il **salt,** ovvero un numerino (12bit in unix) aggiunto alla password, si memorizza in chiaro di fianco all'user nel file delle password. È in chiaro perché non migliora la difesa nei confronti degli attacchi user specific, ma contro gli attacchi (generici). Il salt estende i valori del vocabolario di 4096 volte.
  - È rivolto al file delle password, non sapendo chi la vittima sia (solo che usa una parola banale come password), con una vittima specifica bisogna provare 2^{56} tentativi.

Advanced Encryption System (AES)

È una famiglia di cifrari, lavora su blocchi multipli di 32 bit 128-256, sono stati selezionati quelli che lavorano su 128. si basa sull'applicazione ripetuta di *rounds*, di solito 10, 12, 14.

Campi finiti di Galois (sez 3.11 libro): $\mathbb{Z}_p^*$ è un campo chiuso, esistono gli elementi inversi rispetto all'addizione e alla moltiplicazione (perché p è primo e i numeri sono primi relativi).
Generalizzazione: per ogni potenza p^n $\exists$ $GF(p^n)$ (esiste un campo di Galois di p^n elementi). c'è una maniera di rappresentare gli elementi del campo.

$\mathbb{Z}_p[x]$, $\sum a_i x$ di grado qualsiasi, $a_i \in \mathbb{Z}_p$
$P(x)$, $grado\ n$ irriducibile
$GF(p^n) = \mathbb{Z}_p(x)[mod\ P(x)]$

Noi usiamo il campo $GF(2^8)$

$P(x) grado\ 8$

Polinomio specifico: $P(x) = x^8 + x^4 + x^3 + x + 1$

l'elemento del campo è rappresentabile con un byte. La somma si può fare sommando i polinomi mod 2 e il risultato mod P(x)

[03/22] Campi di Galois, AES

22 March 2018 10:32

Il prof ha studiato da solo sul libro

Ci servono per il resto del corso? No.

Si prende l'insieme dei polinomi in x

$\mathbb{Z}_p[x]; a_i \in \mathbb{Z}_p$

$P(x)\ irriducibile, grado\ n$

Il campo di Galois viene definito come insieme dei resti possibili della divisione tra polinomi P(x)

$GF(p^n) = \mathbb{Z}_p[x](mod\ P(x))$

Numerosità del GF

$|GF(p^n)| = p^n$

In questo GF si possono svolgere tutte e quattro le operazioni:

- Somma: si sommano i polinomi, ma i coefficienti sono (mod p)
  - Es con p=2: $E4 = x^2$
    $5 = x^2 + 1$
- Divisione: $a(x) \neq 0; \frac{b(x)}{a(x)}; se\ MCD(a(x), P(x)) = 1$, voglio P(x) irriducibile così tutti gli elementi ammettono inverso.

$GF(2^8)$, Rijndael

$P(x) = x^8 + x^4 + x^3 + x + 1$; irriducibile.

Elemento generico del campo:

$a(x) = b_7x^7 + b_6x^6 + \cdots + b_0$; ci sono 8 b; $b_i \in \mathbb{Z}_2$

Essendo modulo 2 i coefficienti possono assumere $0/1$.

Insieme dei numeri interi

$\mathbb{Z} \leftrightarrow \mathbb{Z}_p(x)$

Un primo q è analogo:

$q \leftrightarrow P(x)$ irriducibile di grado n.

$\mathbb{Z}_q \leftrightarrow \mathbb{Z}_p(x)[mod\ P(x)]$

$GF(q) \leftrightarrow GF(p^n)$

La radice primitiva e polinomio generatore

$\alpha \leftrightarrow g(x)|: g(x)^n \equiv 1\ [mod\ P(x)]; grado\ massimo = [p^n - 1]$

Radici primitive vs Polinomi generatori

$\phi(q-1) \leftrightarrow \phi(p^n - 1)$

Se $(p^n - 1)$ è primo (capita solo con p=2)→qualunque elemento del campo è generatore (tranne 1, caso degenere).

Prendendo i polinomi coi coefficienti mod p

$a_i \in \mathbb{Z}_p$

Grado n

Quanti

$N_{IRR}(p.n) = \frac{1}{n}\sum u\left(\frac{n}{i}\right)p^i$; n divisibile per i

Funzione di Moebius

$\mu(m)\begin{cases} 0;\ se\ m \text{ è } composto, & primi\ con\ ripetizioni \\ 1;\ se\ m = 1, (-1)^k se\ n \text{ è } prod\ di\ k\ primi\ senza\ ripetizioni \end{cases}$

Es: p=2; n=6

$= \frac{1}{6}(\mu(6)2^1 + \mu(3)2^2 + \mu(2)2^3 + \mu(1)2^6)$

$$= \frac{1}{6}\left(1 * 2^1 + (-1)2^2 + (-1)2^3 + 1 * 2^6\right)$$
$$= \frac{1}{6}(2 - 4 - 8 + 64) = 9$$

AES

Agisce su blocchi di 128 bit, produce blocchi cifrati di 128 bit con una chiave di 128 bit. I blocchi si esprimono con matrici 4x4 (Byte).
Il campo di Galois usato è $GF(2^8)$

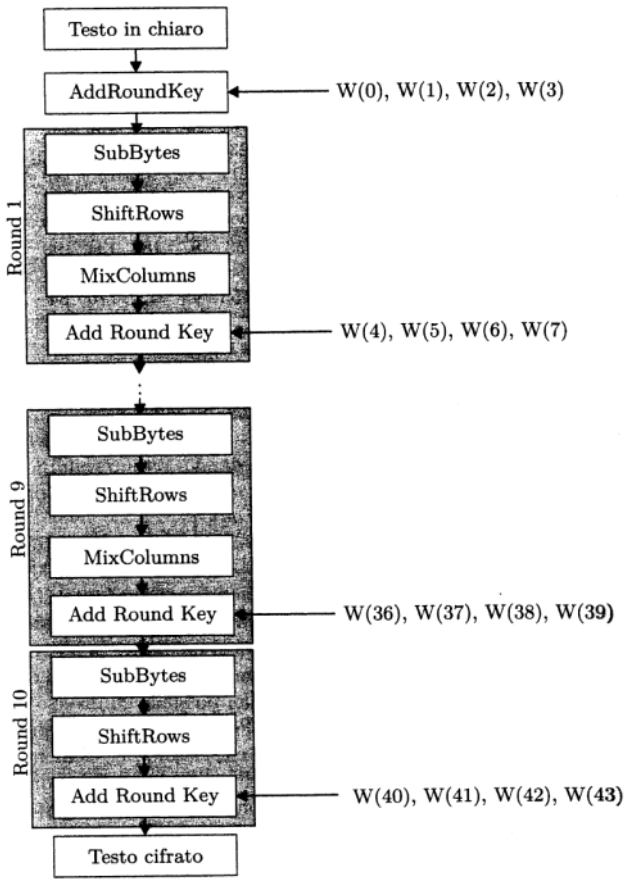


Figura 5.1 L'algoritmo AES-Rijndael.

Ogni round è costituito da 4 layers (passi).

1. Sostituzione dei Byte (c'è una tabella come in DES), dato un Byte in ingresso ce n'è un altro in uscita. Relazione non lineare, la S-Box è generata con un algoritmo pubblico.
2. Shift rows: si shiftano le righe di un offset variabile (diffusione)
3. Mix columns: si mischiano le colonne (diffusione)
4. Add round key: la chiave del round viene aggiunta K_i

Entra un blocco che si chiama matrice A ($a_{00}, a_{01}, \dots, \dots \backslash\backslash a_{10}$ *ecc*)
Es sulla tabella:
I primi 4 bit mi danno la riga, gli ultimi 4 la colonna (i bit di $a_{i,j}$) che mi restituiscono $b_{i,j}$
Le quattro righe vengono shiftate
Mix column: operazione matriciale, prendo la matrice C, moltiplico per M ottenendo D (tutto nel campo 2^8)
La non linearità dell'algoritmo sta nella S-Box, non nelle operazioni.
Add round key è semplicemente la somma bit a bit della matrice K(i), i è il round
$\boldsymbol{E = D \oplus K(i)}$
Mi serve produrre 44 colonne di byte ($w(i)$
$K = K(0) = [w(0), w(1), w(2), w(3)]$

$w(i) = w(i-4) \oplus w(i-1)$, se i non è multiplo di 4
$w(i) = w(i-4) \oplus T[w(i-1)]$, se i è multiplo di 4
La chiave di round iesima è
$K(i) = [w(4i), w(4i+1), w(4i+2), w(4i+3)]$

La S-Box è una lookup table 16x16, prende $a_{i,j}$ e lo sostituisce con $b_{i,j}$

È possibile invertire i passi di questo algoritmo, esiste una tabella per le sostituzioni inverse (ISB), un'operazione inversa per lo shift delle righe e del mix colonne (ISR, IMC), la (ARK) è l'inverso di sé stessa.
Per decifrare bisogna applicare i passi inversi in ordine inverso.

Sequenze di bit pseudocasuali

Cifratura:	$R_i + P_i = C_i$

R_i è una sequenza generata in maniera casuale, chiamata PRBS (Pseudo Random Binary Sequence)

Decifratura	$R_i + C_i = P_i$

$x_n \equiv ax_{n-1} + b \ (mod\ m)$
L'obiettivo è quello di creare una sequenza pseudocasuale che sia impossibile da distinguere da una veramente casuale.
Usare una funzione unidirezionale: dato x posso calcolare y, ma non il contrario (dato $y = f(x)$). Con la funzione unidirezionale genero la sequenza casuale utilizzando il **seed**.
$x_i = f(S + i)$
$b_i = LSB(x_i)$
Usando il DES l'algoritmo diventa pesante in fatto di potenza computazionale.
Si usa quindi l'algoritmo **Blum-Blum-Shab**
È un generatore quadratico, usa due primi p e q
$p, q \equiv 3\ [mod\ 4]$
$n = p \cdot q$
Prendo un x casuale (seme)
$x \in \mathbb{Z}_n$
$X_0 \equiv x^2 [mod\ n]$

$$\begin{cases} x_i \equiv x_{i-1}^2 [mod\ n] \\ b_i \equiv LSB(x_i) \end{cases}$$

[03/26] BBS, LSFR (2.11)

26 March 2018 15:25

Cose per niente banali, richiedono un po' di attenzione

Blum-Blum-Shab

Si prendono due primi p e q $\equiv 3(mod\ 4)$
Si prende un seme iniziale s
$x_0 \equiv s^2 mod\ n$
$n = p \cdot q$
$x_i \equiv x_{i-1}^2 (mod\ n)$
$b_i = LSB(x_i)$
Prove di Bregni: p=43 q=31
$n = 1331$
$x = 50$
Quello che salta fuori è molto deludente:
$x_0 = 1167\ (LSB = 1)$
$896\ (LSB = 0)$
$350\ (LSB = 0)$
$1197\ (LSB = 1)$
1167
Il periodo può essere anche molto piccolo, qual è quello di BBS?
$\pi(x) \backslash \lambda(\lambda(n))$
Cos'è $\lambda(n)$
Funzione di Carmichael (matematico inglese)
$\lambda(n)\ divide\ \phi(n)$
Definizione di $\lambda(n)$:
$\lambda(n) =$ il più piccolo intero positivo m |: $a^m \equiv 1\ (mod\ m) \forall a \perp n$
$\lambda(n) =$ MCM degli ordini (periodi) degli elementi a del gruppo moltiplicativo $\mathbb{Z}_n^*$
Estensione di Carmichael del teorema di Eulero
$a^{\lambda(n)} \equiv 1\ (mod\ n)$
Prendo un $n \in \mathbb{N} \neq 0$
$n = P_1^{a_1} \cdot P_2^{a_2} \dots \cdot P_r^{a_r}$

$\lambda(n) = mcm\{\phi(p_i^{q_i})\}$

Esempio:

p=23	n=437
q=19	$x \equiv 7$

$x \perp n$
Calcolo $\lambda(\lambda(n))$ che deve essere diviso da $\pi(x)$
$\lambda(n) = mcm(22,18) = 198$
$\lambda(198) = mcm(\phi(2), \phi(9), \phi(11)) = mcm(1,6,10) = 30$
$\pi(x) = \{1,2,3,5,6,10,15,30\}$
Questi sono tutti i valori possibili del periodo di n

LFSR (sul libro è sbagliato)

Sommo $R_i \oplus p_i = c_i$
Cifrario ideale se R_i è casuale
Si cerca di generare una sequenza con pochi parametri che abbia una parvenza di casualità.
Voglio generare una sequenza casuale con periodicità lineare:
$x_{n+M} = c_M \cdot x_n + c_{M-1} x_{n+1} + \dots + c_1 x_{n+M-1}$

Sottraggo M a tutti i pedici

$x_n = c_M x_{n-M} + \cdots + c_1 x_{n-1}$

Ricorrenza lineare di ordine M, si definisce data la sequenza dei coefficienti (M coeff)

Polinomio caratteristico

$P(T) = T^M + C_{M-1} T^{M-1} + \cdots + C_1 T + C_0$

$C_M = 1$

$C_0 = 1$

[03/28] Scrambling

28 March 2018 10:32

Esempio

$x_{n+3} = x_{n+1} + n_n$

$= \text{significa} \equiv [mod\ 2]$

Ogni campione è funzione di due precedenti, il più "antico" viene due passi prima. Possiamo traslare di tre passi ottenendo:

$x_n = x_{n-2} + x_{n-3}$

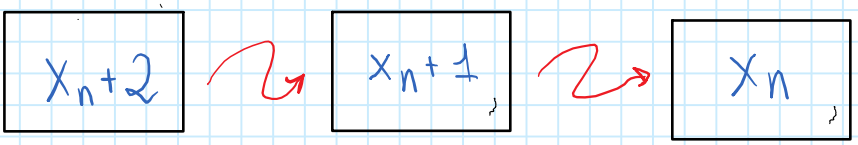
Il polinomio caratteristico di questa ricorrenza (grado 3, la memoria della sequenza):

$P(X) = X^3 + X^2 + 1$

Il coeff di X^3 è il più antico.

$x_{n+M} = c_M x_n + c_{M-1} x_{n+1} + \cdots + c_1 x_{n+M-1}$

$P(X) = X^M + c_{M-1} X^{M-1} + \cdots + c_1 X + c_0$



Mancano disegni

Da questo sistema esce una sequenza PR che andrà a sommarsi XOR con la parola in chiaro.

Ho M+1 coefficienti, di cui 2 predeterminati ($C_0, C_1 = 1$)

Le incognite del problema PTK sono $2 \cdot M$

Sistema non robusto nei confronti degli attacchi PTK. Si usa per cifrare il segnale televisivo.

Ho $2^M - 1$ modi per inizializzare la frequenza (perché tutti 0 non è possibile perché elemento degenere).

Il periodo deve essere un sottomultiplo di $(2^M - 1)/\pi$

Se $2^M - 1$ è primo, il periodo è $2^M - 1$

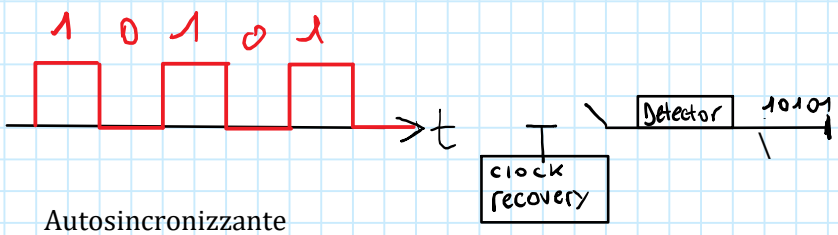
M dovrà essere tale da avere un numero primo $(2^M - 1)$ in modo da conoscere il periodo.

Scrambling

Dare un'apparenza pseudocasuale ad una sequenza.

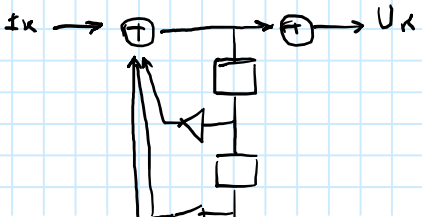
(Rilevatore di soglia per leggere uni e zeri sulla fibra ottica, trasmettendo una sequenza di tutti uni il clock recovery si "perde").

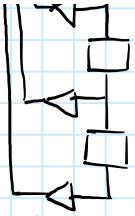
Rendere equiprobabili uni e zeri dando apparenza PR.



Autosincronizzante

Additive Scrambler (usa lo scrambler autosincronizzante senza sequenza in ingresso, per invertire il processo devo sommare la sequenza di inizializzazione allineata





Esempio: scrambler autosincronizzante

Polinomio caratteristico: $1 + x + x^3$

Periodo al più 7

Dividiamo per x+1=$x^2 + x; R = 1$

K	I_k	D_{1k}	D_{2k}	D_{3k}	U_k
0	1	0	0	0	1
1	1	1	0	0	0
2	1	0	1	0	1
3	1	1	0	1	1
4	1	1	1	0	0
5	1	0	1	1	0
6	1	0	0	1	0
7	1	0	0	0	1
8	1				

Le righe azzurre segnano inizio e fine del primo periodo

Descrambling:

K	I_K	D_{1k}	D_{2k}	D_{3k}	U_K
0	1	1	0	0	0
1	0	1	1	0	1
2	1	0	1	1	0
3	1	1	0	1	1
4	0	1	1	0	1
5	0				
6	0				
7	1				
8					

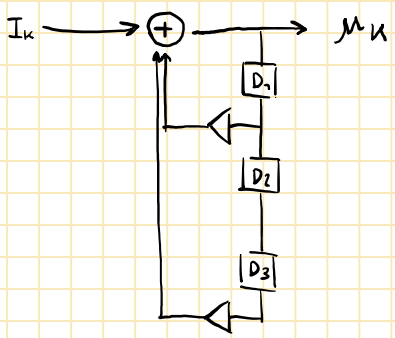
Sulla riga gialla inizia la sequenza di tutti 1 (allineamento)

Ogni errore con questo scrambler moltiplica per 3 il BER

[Esercizio] Scrambler

03 April 2018 Es 1 16:16

a) Pol. car: $P(x) = x^3 + x + 1$



b)

Sommo solo questi

i	I_k	D_{1k}	D_{2k}	D_{3k}	U_k
0	1	0	0	0	1
1	1	1	0	0	0
2	1	0	1	0	1
3	1	1	0	1	1
4	1	1	1	0	0
5	1	0	1	1	0
6	1	0	0	1	0
7	1	0	0	0	1
8	1	1	0	0	0

Periodo $2^3 - 1 = 7$

c) Descrambler

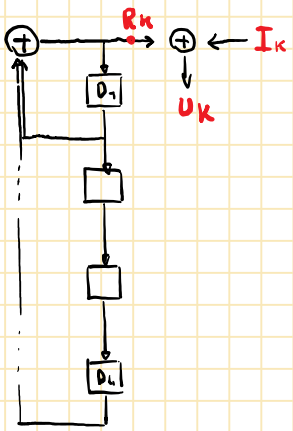
i	I_k	D_{1k}	D_{2k}	D_{3k}	U_k
0	1	1	0	0	0
1	0	1	1	0	1
2	1	0	1	1	0
3	1	1	0	1	1
4	0	1	1	0	1
5	0	0	1	1	1
6	0	0	0	1	1
7	1	0	0	0	1
8	0	1			1

3 bit di sincronizzazione

Esercizio 2

a) $P(x) = 1 + x + x^4$

a) $P(x) = 1 + x + x^4$



b)

i	I_K	D_{1K}	D_{2K}	D_{3K}	D_{4K}	R_K	U_K
0	0	1	0	0	0	1	
1	0	1	1	0	0	1	
2	0	1	1	1	0	1	
3	0	1	1	1	1	0	
4	0	0	1	1	1	1	
5	0	1	0	1	1	0	
6	0	0	1	0	1	1	
7	0	1	0	1	0	1	
8	0	1	1	0	1	0	
9	0	0	1	1	0	0	
10	0	0	0	1	1	1	
11	0	1	0	0	1	0	
12	0	0	1	0	0	0	
13	0	0	0	1	0	0	
14	0	0	0	0	1	1	
15	0	1	0	0	0	1	
16							
17							
18							

$T = 2^4 - 1$
$= 15$

[04/04] RSA; Test Primalità

04 April 2018 10:31

Sistema di cifratura a chiave pubblica (il primo e il più utilizzato anche oggi). "*è una cosa geniale*"
Non è necessario mantenere il segreto della chiave.

RSA: Rivest, **S**hamir, **A**dleman

Come funziona:

i. Si scelgono due primi p e q
ii. Si calcola $p \cdot q$
iii. $P, C \in \mathbb{Z}_n$
iv. Scelgo $e|: e \perp \phi(n) = (p-1)(q-1)$
v. La chiave pubblica è la coppia di valori PUB: (n, e) pubblica

Come si cifra:

i. $C \equiv p^e (mod\ n)$

Decifro:

i. $P \equiv C^d (mod\ n)$

Come si ricava d:

i. $e \cdot d \equiv 1\ [mod\ \phi(n)] \rightarrow d \equiv e^{-1} [mod\ \phi(n)]$ segreta
ii. $ed = 1 + k\phi(n)$
iii. $p^{ed} \equiv p^{1+k\phi(n)} \equiv p \cdot p^{k\phi(n)} (mod\ n)$

Alice

PUB	$n = p \cdot q, \quad e\|: e \perp \phi(n)$
SEGR	$d \equiv e^{-1} [mod\ \phi(n)]$

ES:

p=3
q=11
n=33
$\phi(n) = 20$
e=7 (nota: non potrà essere **mai pari**)

$d = 7^{-1} mod\ \phi(n) = 7^7 mod\ 20 = 3$
p=19
$C = p^e mod\ n = 19^7 mod\ 33 = 13$

È possibile calcolare d, ma è computazionalmente complicato (idealmente lo si vorrebbe rendere impossibile da ricavare, con centinaia di cifre si rende esageratamente lungo il processo di forzatura *brute force*).

- PUB: n,e
- SEGR: d (p, q, $\phi(n)$)

Esponente usato spesso: $e = 65537 = 2^{16} + 1$

La conoscenza di p e q è equivalente alla conoscenza di n e $\phi(n)$.

$n = p \cdot q$
$n - \phi(n) + 1 = pq - (p-1)(q-1) + 1 = p + q$

$X^2 - AX + B = 0$ (le cui radici sono p e q)

Esercizio:

Dati:

- n=11413
- e=7
- $\phi(n) = 11200$

Risoluzione:
i. voglio trovare p e q
Uso l'equazione di secondo grado:

$$X^2 - 214X + 11413 = 0$$

$$X = p, q = \frac{214 \pm \sqrt{214^2 - 4 \cdot 11413}}{2} = \frac{214 \pm 12}{2}$$

$$p = 113$$
$$q = 101$$

Conoscendo e,d si può fattorizzare n, se n ha m cifre e conosco le prime $\frac{m}{4}$ cifre$\Rightarrow p, q$
Conoscendo almeno le ultime $\frac{m}{4}$ cifre di d, risulta più semplice trovare le altre.

RSA serve a trasmettere la chiave per poi utilizzare un sistema di cifratura a chiave simmetrica (usarlo per trasmettere tutto sarebbe troppo pesante a livello computazionale).

Test di primalità di Fermat

Per verificare se n sia primo o composto.

Dal teorema di Fermat: p primo, $a \perp p$, allora $a^{p-1} \equiv 1 \ (mod\ p)$
Se $a^{n-1} \equiv 1 \ (mod\ n) \Rightarrow n\ primo$? Molto probabilmente sì.

Test di Miller-Rabin

Dato $n\ intero$
$se\ \exists x, y | : x^2 \equiv y^2 (mod\ n)$
$\equiv \backslash \equiv \pm y (mod\ n)$

n è composto?
Se $b^{n-1} \equiv \backslash 1 (mod\ n) \Rightarrow n\ composto$
Se $b^{n-1} \equiv 1 (mod\ n) \Rightarrow n\ probabilmente\ primo$, se invece n è composto, si chiama **pseudoprimo di fermat** per la base b (il più piccolo è 341).

I numeri di Carmichael sono pseudo primi per qualunque base. (Sono detti anche **pseudoprimi assoluti**),
il più piccolo è $561 = 3 \cdot 11 \cdot 17$
41041=7*11*13*41
825265=5*7*17*19*73

I numeri di Carmichael fino a X sono:

$$C(X) < xe^{\frac{-k \cdot \log x \cdot \log\log\log X}{\log\log X}}$$

Esempio con base 2:
561 è pseudoprimo di Fermat per la base 2 (oltre a tutte le altre), ma **non è pseudoprimo forte.**
"Miller Rabin lo sgama"

[04/05] Fattorizzazione, Log Discreto

04 April 2018 12:09

"Ma vi piace sta roba?"

Vai a cercare: Factor.exe

Algoritmo di fattorizzazione di Fermat (proprio lui)

Per composti esprimibili come differenza di quadrati:
$n = x^2 - y^2 = (x - y)(x + y)$
$n + 1 =^? x^2$ (n+1 è il quadrato di qualcosa?)

Se $n + 3^2 = x^2$, allora si può scomporre in $n = x^2 - 3^3 = (x + 3)(x - 3)$

ES: 295927+1=x^2 no
$+2^2 = x^2$ no
$+3^2 = x^2$ sì→ 544^2
Quindi: $295927 = 541 \cdot 547$

Algoritmo di Pollard
Condizione: p\n
$p - 1$ scomponibile nel prodotto di fattori primi piccoli (funziona anche se sono grandi, ma l'algoritmo impiega più tempo a convergere).

$p - 1 = \prod p_i^{2_i}$

a>1; a=2
$b_1 \equiv a\ (mod\ n)$
$b_j \equiv b_{j-1}^{j} (mod\ n)$
…
$b_B \equiv a^{B!}$
$d = MCD\left(b_j - 1, n\right) > 1 \Rightarrow STOP$ d\n (d fattore non banale di n)

Esempio: Fattorizziamo n=11413, base a=2

Tutte le operazioni saranno (mod 11413)
$b_1 \equiv 2$
$b_2 \equiv 2^2 \equiv 4$
MCD(3,11413)=1
$b_3 \equiv 4^3 \equiv 64$
MCD(63, 11413)=1
$b_4 \equiv 64^4 \equiv 16777216$
MCD(105, 11413)=1
$b_5 \equiv 106^5 \equiv 11104 \equiv -309$
MCD(11103, 11413)
$b_6 \equiv (-309)^6 \equiv 9993$
MCD=1
…
$b_9 \equiv (-1016)^9 \equiv 5538$
MCD(5537, 11413)=113 STOP

$\Rightarrow 11413 = 113 \cdot 101$

Pollard è rapido con cifre piccole, per proteggersi conviene utilizzare numeri non scomponibili in fattori "grandi". p=2q+1, con p e q primi.

Cap.6 [fine]

Logaritmo discreto

$\beta \equiv \alpha^x (mod\ p)$
$\alpha, \beta \in \mathbb{Z}_p^*$
$x = \log_\alpha^D(\beta)[mod\ (p-1)]$

>Esempio
$2^x \equiv 9(mod\ 11)$
Se 2 è una radice primitiva di $\mathbb{Z}_{11}$ esiste un intero che soddisfa l'equazione sopra.

Proprietà:
- $\log_a^D(\beta_1 \cdot \beta_2) \equiv somma\ dei$ logar mod $(p-1)$

Funzione unidirezionali: *l'elevamento a potenza è invertibile se la base è una radice primitiva di p*

Se so che:
$a^{m_1} \equiv \alpha^{m_2}(mod\ p) \Rightarrow m_1 \equiv m_2[mod\ (p-1)]$

Pohlig-Hellman

$p - 1 = \prod\prod p_i^{r_i}$

Baby-step Giant-step

$N = \left\lfloor \sqrt{p-1} \right\rfloor + 1$

Prendo gli indici i e k

i	α^i	β_d^{-NK}	k
0			0
1			1
2			2
3			3
4			4
…			…
N-1			N-1

$\alpha^j \equiv \beta\alpha^{-NK}(mod\ p)$
$\alpha^x \equiv \alpha^j\alpha^{NK}(mod\ p)$
$x \equiv j + NK(mod\ p-1)$

Il numero di passi (ciascuno richiedente un elevamento a potenza) è N a sx e N-1 a dx, proporzionale a N, il quale è proporzionale a $\sqrt{p}$.

Algoritmo di scambio della chiave di Diffie-Hellmann
Serve a costruire insieme una chiave in modo sicuro.

Alice e Bob scelgono un primo p **sicuro:** $p|: DLP\ in\ \mathbb{Z}_p^*\ difficile$ e un α elemento primitivo. I numeri sono pubblici.
Alice sceglie un x segreto: $1 < x \leq (p-2)$
Bob sceglie un y segreto: $1 < x \leq (p-2)$
Alice spedisce a Bob un messaggio con scritto:
$\alpha^x (mod\ p)$
Bob spedisce ad Alice:
$\alpha^y (mod\ p)$
Si forma così K
$K = (\alpha^x)^y = (\alpha^y)^x = \alpha^{xy} (mod\ p)$
Una chiave da150 bit (se ne usano 128 o quello che è).

Esempio numerico:
p=105
$\alpha = 5$
$\begin{cases} 5^{53} \equiv 106 (mod\ 105 \\ 5^2 \equiv /1 (mod\ 107) \end{cases}$
Ok α è una radice primitiva

Alice SEC: x=23
Bob SEC: y=3

Alice spedisce: $5^{23} \equiv 59 (mod\ 107) \rightarrow$ Bob
Bob spedisce: $5^3 \equiv 18\ (mod\ 107) \rightarrow Alice$
$K_{BA} = 59^3 \equiv 46 (mod\ 107)$
$K_{AB} = 18^{23} \equiv 46 (mod\ 107)$

EVA

Problema computazionale di Diffie-Hellmann

$$Dati \begin{cases} p, \alpha \\ a^x mod\ p\ ; a^{xy} (mod\ p) \\ a^y mod\ p \end{cases}$$

Problema decisionale di D-H

Noti $p, \alpha, \alpha^x mod\ p, \alpha^y mod\ p, c\ mod\ p, decidere\ se\ p \equiv \alpha^{xy} (mod\ p)$

Crittosistema a chiave pubblica di El-Gamal

Chiave pubblica, DLP

- Si sceglie p (grande) e α, PUB (Bob)
- P (msg) viene spedito
- Bob si tiene un intero a SEGR $(1 < a \leq p-2)$
- β è pubblico, $\beta \equiv \alpha^a (mod\ p)$

Alice:

- $P \in \mathbb{Z}_p^*\ e$
- k segreto NONCE (utilizzabile una volta sola)
- $1 < k < p-2$
- $C = (r, t)$
  - $r \equiv a^k (mod\ p)$
  - $r \equiv \beta^k P (mod\ p =$

Bob per decifrare:

- $tr^{-a} \equiv \beta^k P (\alpha^k)^{-a} \equiv \alpha^{akP\alpha^{-ak}} \equiv M (mod\ p)$

Attacco del NONCE RIPETUTO

Alice riutilizza k, Eva se ne rende conto perché verrebbero generate due r,t uguali, in questo modo alice è esposta ad un attacco PTK.

[Esercizio] Cifrario Affine in CFB

07 April 2018 12:01

Alfabeto da 32 caratteri: A-Z + a-m

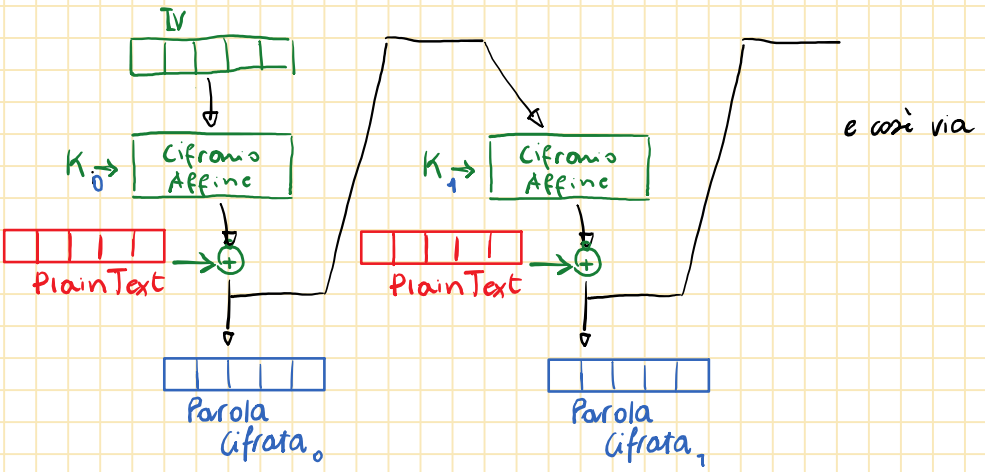
$P, C \in \mathbb{Z}_{32}$

$C_i = P_i + E_k(C_{i-1})$

$C_0 = IV$ (ignota)

} CFB

$E_k(X) = (aX+b) \pmod{32}$



$P = TCP \Rightarrow C = UDP$ (3)

P: T C P

T = 17 =	1 0 0 0 1	U = 18 =	1 0 0 1 0
C = 2 =	0 0 0 1 0	D = 3 =	0 0 0 1 1
P = 13 =	0 1 1 0 1	P = 13 =	0 1 1 0 1

Decifro

Passo 3:
0 1 1 0 1
0 1 1 0 1
0 0 0 0 0 → 0 = $E_k(3)$

Passo 2:
0 0 0 1 1
0 0 0 1 0
0 0 0 0 1 → 1 = $E_k(18)$

$18\alpha + \beta \equiv 1 \pmod{32}$

$3\alpha + \beta \equiv 0 \pmod{32}$

$3 \cdot 13 + \beta \equiv 0 \mod 32$

$39 + \beta = 0 \mod 32$

$5\alpha \equiv 1 \mod (32)$

$5\alpha = m+1$

$$\frac{}{32} = n+1$$

$$5a = 32n + 1$$

$$33$$

$$5a = 65$$

$$a = 13$$

$$\beta = 25$$

[04/09] Hash functions

09 April 2018 15:14

Una funzione di hash viene applicata ad una stringa di lunghezza arbitraria per ottenerne una di lunghezza fissa.

$y = h(m)$

$|y| = 160bit$

$|m| < 10kB$

Hash=tritare

Non è una funzione invertibile.

Proprietà:

- Veloce da calcolare
- **Unidirezionalità:** (one way func) dato $y, m^?|: h(m) = y \rightarrow$ *non dovrebbe essere possibile*.
- Deve essere computazionalmente impossibile trovare due messaggi (m_1, m_2) tali che $h(m_1) \neq h(m_2)$ (gli hash devono essere diversi). In questo caso si dice che la funzione è *fortemente resistente alle collisioni.*
- *Debolmente resistente alle collisioni:* dato $m_1 \Rightarrow \nexists m_2 |: m_1 \neq m_2 \wedge h(m_1) = h(m_2)$

Una funzione fortemente resistente lo è anche debolmente, non vale il viceversa.

Esempi:

1) $h(m) = m \ mod \ n$
   $|n| \simeq 10^{48}$
   Funzione non unidirezionale e non resistente alle collisioni (se $m_1 \equiv m_2 \ (mod \ n)$)
2) $h(x) = \alpha^x (mod \ p)$
   È unidirezionale scegliendo p molto grande
   Esistono due messaggi $x_1 e \ x_2$ che mi diano lo stesso risultato? Sì $x_1 \equiv x_2 mod \ (p-1)$
3) $h(x) = x^2 (mod \ n)$
4) **BIP**:

Famiglie di messaggi di hash:

- **SHA:**

Famiglia	**SHA-1**	**SHA-256**	**SHA-384**	**SHA-512**
N bit (|h|)	160	512	1024	1024

- **MD**

[04/11] Firme

11 April 2018 10:31

- Paradosso del compleanno in Excel

$P(\exists CORR) \approx 1 - e^{-\frac{r^2}{2N}}$

Supponiamo di avere un hash di
|h|=60 bit
Non sono abbastanza bit perché non è difficile scrivere 2^{30} variazioni di un documento. (C'è probabilità che due di queste abbiano lo stesso hash). Il paradosso del compleanno ci interessa per considerare la probabilità che due gruppi di circa $\sqrt{N}$ scelgano lo stesso "compleanno" all'interno di uno spazio di N possibilità.
Problema col logaritmo discreto:

$\alpha^x \equiv \beta(mod\ p)$

α^k	$\beta\alpha^{-l}$
$\sqrt{p^k}$ valori	$\sqrt{p^l}$ valori

Quando trovo valori tali che $\alpha^{k_0} \equiv \beta\alpha^{-l_0}$ ho la soluzione

$\alpha^{k_0-l_0} \equiv \beta$

Usare le funzioni di hash come algoritmo crittografico

$P_i \oplus R_i = C_i$
$C_i = X_i \oplus P_i$
X_i seq. pseudocasuale
$X_i = L_8[h(k_{AB} \vee x_{i-1})$
Come inizializzware? $X_0^?$, X_0 può essere trasmesso in chiaro, il segreto deve essere K_{AB} che viene usata per cifrare. Sistema molto robusto, a condizione di avere una funzione di hash unidirezionale buona (SHA ecc)

Firme

Solo il firmatario legittimo dovrebbe essere capace di produrla, dovrebbe essere improbabile riprodurla generandola in modo aleatorio. Si firma l'hash di un documento perché è costante indipendente da ciò da cui viene generato.

L'algoritmo di firma deve essere tale che solo Bob può generarla, ma tutti possono comprovarla.

RSA

PUB	$\boldsymbol{n = p \cdot q}; e \perp \phi(n); 1 \ll e < \phi(n)$
SEG	**p,q;** $d \perp \phi(n); ed \equiv 1[mod\ \phi(n)\ ]$

$C = P^e(mod\ n)$
$P = C^d(mod\ n)$

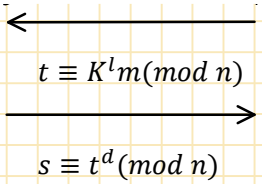
$A = SIG_{alice}(m) = m^d(mod\ n)$	Firma di Alice
$VER(A)$ vera se $A^e(mod\ n) = m$	Verifica

Per spacciarmi per alice devo avere d (SEG), per calcolare d bisogna invertire $[e\ mod\ \phi(n)]$
Prendo una firma A, lo elevo a e: $\tilde{A}^e mod\ n = \tilde{m}$

RSA Blind Signature

Bob vuole che Alice firmi un messaggio m, senza che lei lo veda. Bob sceglie K (mod n) **NONCE SEGRETO.** Il vincolo è $K \perp n$

Bob	m,	$k(mod\ n)$: $k \perp n$



Ora Bob può calcolare $\frac{s}{k} \equiv m^d \equiv Km^d \equiv k^{ed} (mod\ n)$

[04/12] Firma El-Gamal [+esercizi]

12 April 2018 10:34

Si basa anch'esso sull'irrisolvibilità del problema del log discreto.

Alice

PUB	$p\ (grande, prob\log difficile, [2q+1]), \alpha\ (rad\ prim), \beta = \alpha^a (mod\ p)$
SEGR	$1 \ll a \leq (p-2), K \perp (p-1)$

Da attaccante proverei a calcolare a (log discr impossibile per come p è scelto).

Anche qui si usa un **nonce <u>segreto</u>** (da usare una volta sola) **K**, primo relativo con (p-1) verificato con Euclide. Ogni volta che si firma un messaggio (anche lo stesso) la firma cambia, quindi ci sono più firme valide per un documento.

Algoritmo:

$A(m) = (r, s)$

$r \equiv \alpha^k\ (mod\ p)\ \in \mathbb{Z}_p^*$

$s \equiv k^{-1}(m - ar)[mod\ (p-1)] \in \mathbb{Z}_{p-1}$

Verifica: (chiunque deve essere in grado di compiere questo passaggio)

VER(r,s,m)

$vera\ if\ \beta^r r^s \equiv \alpha^m (mod\ p)$

$\alpha^{ar}\alpha^{Ks} \equiv \alpha^m (mod\ p)$

$m \equiv ar + Ks[mod\ (p-1)]$

Attacco (Eva):

Manca a, bisogna ricavare r, trovare simultameamente (r,s) non è un problema di log discreto, ma è egualmente difficile (non è stato trovato un algoritmo per risolverlo).

Attacco Nonce Ripetuto

Alice usa due volte lo stesso K (scema). Cosa succede:

$A(m_1) = (r, s_1)$

$A(m_2) = (r, s_2)$

Vedendo che r rimane lo stesso si nota che Alice non ha cambiato K

Dalla def di s ricavo che:

$sK \equiv m - ar\ [mod\ (p-1)]$

$-ar \equiv s_1K - m_1 \equiv s_2K - m_2\ [mod\ (p-1)]$

$(s_1 - s_2)K \equiv m_1 - m_2\ [mod\ (p-1)]$

Ci sarà almeno una soluzione di K valida, potrebbe averne anche più di una perché la differenza tra le due s non è necessariamente prima relativa rispetto a (p-1). Dopo aver ricavato il K giusto posso ricavare a (inserendolo dove è evidenziato di ciano), ovvero il segreto effettivo di Alice (persistente), permettendo di firmare qualsiasi messaggio a nome di Alice.

Esercizio (*esempio numerico con numeri piccoli*):

Alice

PUB	$p = 43, \alpha = 3$
SEGR	$a = 10, K = 11, \beta = \alpha^a mod\ p = 3^{10} mod\ 43 \equiv 10\ mod\ 43$

Per verificare α che sia primitivo: $\alpha^{\frac{p-1}{q_i}} not \equiv 1\, mod\ p$

$p-1=42=2\cdot 3\cdot 7$

$3^{\frac{42}{2}}=3^{21}(mod\ 43)=42$

$3^{\frac{42}{3}}=3^{14}(mod\ 43)=36$

$3^{6}(mod\ 43)=41$

3 va bene perché nessuno di questi numeri è congruente con 1 (mod 43)

$K\perp(p-1)=11\perp 42$ sì

Il messaggio m è limitato da p, $0<m\leq 42, m\in\mathbb{Z}_{42}$

$A(15)=(r_1,s_1)=$

$r_1=a^K=3^{11}\equiv 30\ (mod\ 43)$

$s_1=k^{-1}(m-ar)(mod\ 42)=11^{-1}\equiv 11^{11}\equiv 23$ (ho elevato a $\phi(p)-1$)

$s_1\equiv 23(15-10\cdot 30)(mod\ 42)\equiv 39$

Verifico

$\mathrm{VER}(m_1,r_1,s_1)=\begin{cases}10^{30}30^{39}\equiv 22(mod\ 43)\\ 3^{15}\equiv 22(mod\ 43)\end{cases}$

15,30,39

Risultato: $m_1=15\rightarrow r_1 30; s_1=39$

Riprovo con $m_2=20$

$r_2=30, s_2=28$

Noto che 30 è lo stesso, quindi posso calcolare il K

$s\cdot K\equiv m-ar\left(mod\ (p-1)\right)$

$\begin{cases}39\cdot K\equiv 15-a\cdot 30(mod\ 42)\\ 28\cdot K\equiv 20-a\cdot 30(mod\ 42)\end{cases}$

$11\cdot K\equiv -5\ (mod\ 42)$

Inverso di 11 mod 42 (condizione: $11\perp 42$): $11^{-1}\equiv 23\ (mod\ 42)$

$K\equiv -5\cdot 23(mod\ 42)\equiv 11$

Verifico che 11 sia corretto

$\alpha^K\equiv\beta\rightarrow 3^{11}\equiv 30\ (mod\ 4x)$

$39\cdot 11\equiv 15-q\cdot 30(mod\ 42)$

$930\equiv -414\equiv 6\ (mod\ 42)$

$30\cdot a\equiv 6\ (mod\ 42)$

Non posso trovare l'inverso

MCD(30,42)=6

$5\cdot a\equiv 1\ (mod\ 7)$

$a\equiv 5^{-1}\equiv 3\ (mod\ 7)$

a=3,10,17,24,31,38

$\beta=10\equiv\alpha^a(mod\ 43)$

Altro esercizio: [RSA] T.E. 11/07/2013

Bob

PUB	$n=323, e=17$

Devo verificare che $e\perp\phi(n)$

Alice trasmette $C=55, P=?^?$

Devo calcolare d

$d\equiv e^{-1}(mod\ \phi(n))$

Occorre fattorizzare $n=17\cdot 19$

$\phi(n)=16\cdot 18=288=2^5 3^2$

Come trovo d

$d\equiv 17^{-1}(mod\ 288)=\cdots$

$\phi(\phi(n)) = (2^5 - 2^4)(3^2 - 3) = 96$

$\ldots \equiv 17^{95} (mod\ 288) = 17$ (ho il dubbio, verifico)

$17 \cdot 17 = 1\ (mod\ 288)$

Decifrazione:

$P \equiv C^d (mod\ n)$

$= 55^{17} (mod\ 323) = 123$

Messaggio: 55, C=123 (perché d=e)

Esercizio da TE 27/02/13

Bob

PUB	$n = 437, e = 17$

Calcolare la firma A(P=77)

$A(P) = P^d (mod\ n)$

Calcolo d:

$d \equiv e^{-1} (mod\ n) = \cdots$

Fattorizzo n:

$n = 19 \cdot 23$

$\phi(n) = 18 \cdot 22 = 396 = 2^2 \cdot 3^2 \cdot 11$

$\phi[\phi(n)] = (2^2 - 2) \cdot (3^2 - 3) \cdot 10 = 120$

$\ldots d \equiv 17^{119} (mod\ 396)$

Euclide esteso

MCD(17, 396)=

$396 = 23 \cdot 17 + 5$	$x_0 = 0; x_1 = 1$
$17 = 3 \cdot 5 + 2$	$x_2 = -x_1 \cdot q_1 + x_0 = -23 = 373$
$5 = 2 \cdot 2 + 1$	$x_3 = -x_2 \cdot q_2 + x_1 = 70$
$2 = 2 \cdot 1 + 0$	$x_4 = (-2)70 - 23 = -163 = +233$

Verifico: $233 \cdot 17 \equiv 1 (mod\ 396) ok \Rightarrow d = 233$

Firmo:

$77^{233} (mod\ 437)$

Devo usare S&M

Domanda c) $A \equiv 2\ (mod\ n)$ è uguale alla firma valida di quale messaggio?

Firma cieca 25/09/2013

Alice

PUB	$n = 391, e = 13$

Bob estrae K = 12 e vuole far firmare ciecamente (P=32) ad Alice.

$n = 17 \cdot 23 \rightarrow \phi(n) 2^5 \cdot 11 \rightarrow \phi[\phi(n)] = 160$

$e \perp \phi(n); K \perp n$ (condizioni)

$d \equiv e^{-1} (mod\ \phi(n)) \equiv 13^{159} (mod\ 352) \equiv 325$

Bob→Alice	$t = K^C \cdot P (mod\ n) = 12^{13} \cdot 32\ (mod\ 391) = 261$

Alice→Bob $s \equiv t^d \ (mod\ n) = 261^{325} (mdo\ 391) = 381$

Bob calcola la firma A

$$A = \frac{s}{k} (mod\ n)$$

Serve l'inverso di K

$$K^{-1} \equiv 163 (mod\ n)$$

$$A = 381 \cdot 163 = 325$$

325 è la firma del messaggio P=32 calcolata da Alice (che non conosce P):

Verifico:

$$32^{325} (mod\ 391) = 325$$

[04/18]Esercizi

18 April 2018 10:35

EL-Gamalleelel (27/2/13)

Chiave pubblica E-G

Bob

PUB	$p=127, \alpha = 3;\ \beta = \alpha^a (mod\ p)$
SEGR	a=98

Test

$p - 1 = 126$

Scompongo in fattori primi

$126 = 2 \cdot 3^2 + 7$

Alpha radice primitiva?

$\alpha^{\frac{p-1}{q_i}} ! \equiv 1 (mod\ p)$

$\begin{cases} 3^{63} \equiv 126 \\ 3^{42} \equiv 107 \Rightarrow \alpha = 3 \\ 3^{18} \equiv 4 \end{cases}$

(per 3^{63} uso S&M)

Alice estrae nonce K=110 e spedisce il messaggio P=33

b) C(r,t)

$r \equiv \alpha^k (mod\ p)$

$t \equiv \beta^k P (mod\ p)$

$r = 3^{110} (mod\ 127) = 34$

$t \equiv 37^{110} \cdot (mod\ 127) = 92$

c) Decifrare: C=(r',t')=(37,102)

$P' \equiv t' \cdot (r')^{-a} (mod\ p)$

$\equiv 102 \cdot 37^{-98} (mod\ 127) \equiv 102 \cdot 37^{28} (mod\ 127) = 91$

Altro esercizio

Bob

PUB	$p = 59;\ \alpha = 2; \beta = \alpha^a (mod\ p) = 30$
SEG	a=57

Alice

K=23

$P_1 = 10$

$\rightarrow r_1 = 2^{23} mod\ 59 \equiv 47$

$\rightarrow t_1 = 30^{23} \cdot 10 (mod\ 59) \equiv 9$

P_2

$K = 23$

$r_2 = 47$

$t_2 = 4$

Eve attacc (nonce ripetuto)
l'attacco a E-G è di tipo plain text known
Elementi noti: p, α, β
$P_1, C_1 \rightarrow P_2, C_2 (stesso\ K)$
Cerca di ricavare P_2
$\frac{P_1}{t_1} = \frac{P_2}{t_2} (mod\ p)$

$P_2 \equiv \frac{P_1}{t_1} \cdot t_2 (mod\ p)$

$t_1^{-1} \equiv 9^{-1} (mod\ 59)$
$\equiv 9^{57} (mod\ 59)$
$\equiv 46$

$P_2 \equiv 10 \cdot 46 \cdot 4 \equiv 11 (mod\ 59)$

Esercizio firma E-G

P=113 $\alpha = 7$ $\beta = \alpha^{\alpha} (mod\ p)$	PUB		SEG	a=29

$p - 1 = 112 = 2^4 \cdot 7$
$\begin{cases} 7^{56} \equiv 1 \\ 7^{16} \equiv 49 \end{cases} (mod\ 113) \rightarrow \alpha = 7\ NO$

$\begin{cases} 6^{56} \equiv 112 \\ 6^{16} \equiv 30 \end{cases} (mod\ 113) \rightarrow \alpha = 6\ YOS$

b) Bob estrae il nonce **k=51**
A(P=101)=(r,s)

$r = \alpha^k (mod\ p) = 6^{51} (mod\ 113) \equiv 70$

$s \equiv k^{-1} (P - a \cdot r) \left(mod\ (p-1)\right)$
$\equiv 11 (101 - 29 \cdot 70) (mod\ 112) = 61$
Devo trovare k: $k^{-1} = 51^{-1} (mod\ 112)$
$\phi(112) = 48$
$51^{-1} \equiv 51^{47} (mod\ 112) \equiv 11$

c) Verificare che A'(03,46) è valida per P'=101
d) $\begin{cases} 23^{103} \cdot 103^{46} \equiv 101 \\ 6^{101} \equiv 101 \end{cases} (mod\ 113)$

A=(103, 46) è la firma valida per P=101 per k=3

[04/19] Altri esercizi El Gamal

19 April 2018 11:39

Firma E_G (13/09/13)
Alice

PUB	$P=113, \alpha = 6, \beta = \alpha^a (mod\ p) = 31$
SEGR	a

Bob

Estrae	K

Bob sbaglia e usa sempre lo stesso K
Firme di Bob:

$A_1 = (r_1, s_1) = (92,4)$ $P_1 = 12$
$A_2 = (r_2, s_2) = (92,44)$ $P_2 = 100$
$A_3 = (\dots) = (92,12)$ $P_3 = 35$

Verifico le firme:

$\beta^r r^s \equiv \alpha^P (mod\ p)$

$A_1 \rightarrow 31^{92} \cdot 92^4 \equiv 56 (mod\ 113)$
$6^{12} \equiv 56$

Firma OK

A_2	$31^{92} \cdot 92^{44} \equiv 111$
	$6^{100} \equiv 111$
Quicc maffs	

A_3	Non va bene

Ricavo r ed s

$s \equiv K^{-1}(P - ar)[mod\ (p-1)]$

$\begin{cases} 4K \equiv 12 - a \cdot 92 (mod\ 112) \rightarrow 40K \equiv 88 (mod\ 112) \\ 44K \equiv 100 - a \cdot 92\ (mod\ 112) \end{cases}$

MCD(40,112)=8

$5K \equiv 11 (mod\ 14)$
Calcolo l'inverso di 5 mod 14
$5^{-1} \equiv 5^5 \equiv 3\ (mod\ 14)$
$K_0 = 11 \cdot 5^{-1} \equiv 33 \equiv 5\ (mod\ 14)$
$K_i \equiv 5,19,33,47,61,75,89,103\ (mod\ 112)$
Devo scegliere uno di questi valori (guardo la def di r)
$r \equiv \alpha^K (mod\ p)$
Provo i valori e scopre che è il primo (5)
$r = 6^5\ (mod\ 112)$

Devo ricavare a

$4 \cdot 5 = 20 \equiv 12 - a \cdot 92\ (mod\ 112)$
$a \cdot 92 \equiv -8\ (mod\ 112) \equiv 104$

$MCD(92,112) = 4$
$23a \equiv 26\ (mod\ 28)$
Inverso di 23
$23^{-1} \equiv 23^{\phi(28)-1} \equiv 23^{11} (mod\ 28) \equiv 11$
$a \equiv 11 \cdot 26 (mod\ 28) \equiv 6$

$a_i = 6,34,62,90\ (mod\ 112)$

Provo questi valori
$a = 90$

Baby-step Giant-Step
$\log_\alpha \beta$
$\alpha^x \equiv \beta(mod\ p)$
$5^x \equiv 21\ (mod\ 23)$

Test per verificare che 5 sia radice primitiva
$5^{11} \equiv 22\ (mod\ 23)$
$55^2 \equiv 2(mod\ 23)$
$\alpha = 5$ è una radice primitiva

$21 \cdot 5^{-5K+22}$

$N = \left\lceil \sqrt{p-1} \right\rceil = 5$

J	a^j	k	$\beta\alpha^{-NK}$
0	$5^0 \equiv 1$	0	21
1	$5^1 \equiv 5$	1	$21 \cdot 5^{-5+22} \equiv 21 \cdot 5^{17} \equiv 16$
2	2	2	$21 \cdot 5^{12} \equiv 10$ *(match for j=3)*
3	10	3	
4	$50\ (mod\ 23) \equiv 4$	4	
5		5	

$\alpha^j \equiv \beta\alpha^{-5K}(mod\ 23)$
j=3
k=2
$\alpha^{3+5\cdot 2} \equiv \beta(mod\ 23)$

$x \equiv 13\ (mod\ 22)$

[04/19] Key Agreement and Distribution

19 April 2018 12:58

(Trappe-Washington cap 10)

Distribuzione delle chiavi
La sicurezza risiede sulla segretezza delle chiavi.
Sistemi di cifratura:

- Chiave simmetrica, segreto condiviso tra le due parti. Problema: come fanno gli interlocutori a condividere il segreto?
- Chiave pubblica, le chiavi private non possono essere ricavate da quelle pubbliche a meno di conoscere la trapdoor

Come fidarsi di una chiave pubblica?

- Certificati
- Autenticazioni

Chiavi simmetriche

Sono da costruire/stabilire, procedura per arrivare a stabilire un segreto condiviso, può essere di due tipi:

- Key-agreement (calcolata insieme dalle due parti (attraverso canale pubblico), es: Diffie-Hellmann)
- Key distribution, c'è un'autorità centrale che decide le chiavi e le distribuisce agli utenti. Sistema non molto scalabile (si usa il sistema a chiave pubblica per distribuire le chiavi)

Diffie-Hellman

Scelta degli esponenti

Alice SEG	$1 \leq x \leq p-2$
Bob SEG	$1 \leq y \leq p-2$

Trasmissione

$A \rightarrow B$	$a^x mod\ p = C_A$
$B \rightarrow A$	$a^y mod\ p = C_B$

Calcolo delle chiavi

Alice	$C_B^x (mod\ p) = K$
Bob	$C_A^y (mod\ p) = K$

Attacco *Man-in-the-Middle:*

Eva finge di essere A con B e B con A intercettando i valori pubblici $a^x e\ a^y$ e inviando a ciascuno il proprio valore con esponente z

Intercettazione → Invio esponenti → Filtraggio/modifica comunicazioni

$A \nrightarrow B$ $A \rightarrow E$	$\alpha^x (mod\ p)$
$E \rightarrow B$	$\alpha^z (mod\ p); 1 \leq z \leq p-2$
K_{EB}	$(\alpha^z)^y (mod\ p)$
$B \nrightarrow A$ $B \rightarrow E$	$\alpha^y (mod\ p)$
$E \rightarrow A$	$\alpha^z (mod\ p)$
K_{EA}	$(\alpha^z)^x (mod\ p)$

Ora Eva riesce a decifrare, leggere e modificare i messaggi che Alice e Bob si scambiano.

Alice chiede a Trent di verificare la firma di Bob per assicurarsi dell'identità del suo interlocutore

[Esercizi] Eserciziario di Verticale

21 April 2018 14:33

1.1 GCD e algoritmo di Euclide

Esercizio 1.1 *Calcolare* $d = \gcd(360, 294)$ *in due modi:*

1. *fattorizzando ciascuno dei numeri e poi fattorizzando d;*
2. *usando l'algoritmo di Euclide.*

$d = GCD(360, 294)$

1. $360 = 6^2 \cdot 10 = (2\cdot 3)^2 \cdot 2\cdot 5 = 5\cdot 3^2 \cdot 2^3$

$294 = 2\cdot 7\cdot 7\cdot 3 = 7^2 \cdot 3\cdot 2$

$GCD = 7^2 \cdot 5 \cdot 3 \cdot 2 = 6$ (non comuni)

2. Algoritmo di Euclide

$r_0 = \max(a,b) = 360$

$r_1 = \min(a,b) = 294$

$r_0 = q_1 r_1 + r_2$

$360 = 1\cdot 294 + 66$

$r_{j-2} = q_{j-1} r_{j-1} + r_j$

j		
0	360	–
1	294	–
2	66	$360 = 1\cdot 294 + 66$
3	30	$294 = 4\cdot 66 + 30$
4	6	$66 = 2\cdot 30 + 6$
5	∅	$30 = 5\cdot 6$

Esercizio 1.2 *Trovare* $d = \gcd(841, 294)$ *ed esprimere d come combinazione lineare dei due numeri.*

$GCD(841, 294) = r_0 s + r_1 t$

$$s_j = \begin{cases} 1 & \text{è uguale una iniziale al contrario} \\ 0 & \\ s_{j-2} - q_{j-1} s_{j-1} & \end{cases} \quad \text{if } j = \begin{matrix} 0 \\ 1 \\ \geq 2 \end{matrix}$$

	r_j	q_j		s_j	t_j	$r_0 s_j + r_1 t_j$
0	841	–	–	1	0	841
1	294	2	–	0	1	294
2	253	1	$841 = 2\cdot 294 + 253$	1	-2	253
3	41	6	$294 = 1\cdot 253 + 41$	-1	3	41
4	7	5	$253 = 6\cdot 41 + 7$	7	-20	7

2	253	1	$841 = 2 \cdot 294 + 253$	1	-2	253
3	41	6	$294 = 1 \cdot 253 + 41$	-1	3	41
4	7	5	$253 = 6 \cdot 41 + 7$	7	-20	7
5	6	1	$41 = 5 \cdot 7 + 6$	-36	103	6
ultimo resto non nullo 6	1	6	$7 = 1 \cdot 6 + 1$	43	-123	1
7	/		$6 = 6 \cdot 1 + 0$	-294	841	0

$$d = 6 = 841 \cdot 43 - 123 \cdot 294$$

1.3 Teorema cinese del resto

Esercizio 1.6 *Trovare le soluzioni della seguente congruenza:*

$$3x \equiv 4 \pmod 7$$

Ritagio schermata acquisito: 21/04/2018 15:15

$3x \equiv 4 \pmod 7$

$3^{-1} \equiv 5 \mod 7$

$5 \cdot 3x \equiv 5 \cdot 4 \pmod 7$

$x \equiv 20 \pmod 7 \equiv 6$

$x \equiv 7k + 6$

0	7	1	$3 + 1$
1	3	3	$1 + 0$
2			

$x_0 = 0 \quad x_1 = 1$

$x_2 = -2 \cdot 1 + 0 = -2$

$-3 \cdot -2 + 1 = 5$

$$\begin{cases} x \equiv 2 \pmod 3 \\ x \equiv 3 \pmod 5 \\ x \equiv 4 \pmod{11} \\ x \equiv 5 \pmod{16} \end{cases}$$

$a_i = \{2, 3, 4, 5\}$

$m_i = \{3, 5, 11, 16\}$

$M = \prod_{1}^{4} m_i = 3 \cdot 5 \cdot 11 \cdot 16 = 2640$

$z_i = \frac{M}{m_i}$

$y_i = z_i^{-1} \mod (m_i)$

$x = \sum_{1}^{4} a_i y_i z_i \mod M$

i	a_i	m_i	z_i	y_i
1	2	3	880	$880^{-1} \mod 3 = 880$
2	3	5	528	$528^{-1} \mod 5 = 2$
3	4	11	240	$240^{-1} \mod 11 = 5$
4	5	16	165	$165^{-1} \mod$

$240^{-1} \bmod 11$

[05/09] Key Distribution Protocols

09 May 2018 10:32

Symmetric-key agreement

"Diffie-Hellman è una delle cose più importanti che ho spiegato"

C'è qualcuno che decide una chiave ed è necessario un modo per distribuirla.
Cose che mi sono perso: Shamir-Massey-Omura 3-Pass protocol, sicuro dal punto di vista computazionale (sempre col rischio del MiiM).

Esistono algoritmi che servono a ridurre il numero di messaggi usati per distribuire le chiavi:

Blom Scheme for Key Pre-Distribution

A ciascun utente U è assegnato un numero distinto pubblico identificativo	$r_U (mod\ p); r_U \in \mathbb{Z}_p$
Trent (trusted authority) sceglie 3 numeri segreti	$a, b, c\ (mod\ p)$
Per ogni utente Trent calcola i numeri: E li invia agli utenti tramite canale sicuro	$a_U \equiv a + br_U (mod\ p)$ $b_U \equiv b + cr_U (mod\ p)$
Ciascun utente calcola	$g_U(x) = a_U + b_U x$
Se A vuole comunicare con B calcola	$K_{AB} = g_A(r_B)$
Se B vuole comunicare con A calcola	$K_{BA} = g_B(r_A)$

Es numerico:

Utenti:	$r_A = 11; r_B = 3; r_C = 2; p = 23$
Numeri segreti:	$a = 8, b = 3, c = 1$
Trent calcola	$a_A \equiv 8 + 3 \cdot 11 \equiv 18\ \ b_A \equiv 3 + 1 \cdot 11 \equiv 14$ $a_B \equiv 17\ \ \ b_B \equiv 6$ $a_c \equiv 14\ \ b_C \equiv 5$
Gli utenti ricevono i numeri e calcolano il polinomio	$g_A(x) = a_A + b_A x =$ $g_B(x) = a_B + b_B x =$
Alice per comunicare con Bob	$K_{AB} \equiv 18 + 14 \cdot 3 \equiv 14$
Bob per comunicare con Alice	$K_{BA} \equiv 17 + 6 \cdot 11 \equiv 14$

Il vantaggio è l'utilizzo del canale sicuro esclusivamente per spedire una coppia di numeri per utente, lo svantaggio è la necessità di un canale sicuro.

Problema: la minaccia principale è il **Replay attack**, Oscar vede passare dei messaggi (non è necessario che li capisca/decifri), può rispedirlo al mittente o a un altro destinatario fingendo di essere Trent (se manca l'autenticazione).
Per contrastare i replay attack:

1. Autenticazione dei mittenti, però ho bisogno di altre chiavi (key encryption keys).
2. Fare in modo di poter tracciare l'età del messaggio cifrato:
   i. Sequence number, un numero di sequenza inserito nel messaggio cifrato (se il MiiM prova a rispedirlo si nota che il SN è inferiore all'ultimo arrivato)
   ii. Timestamp: si imposta un tempo di time out oltre al quale un timestamp non è più valido (è necessario un clock centrale (gli host devono usare NTP/PTP per sincronizzarsi), problema di sincronizzazione)
   iii. Nonce: numero casuale usato una volta in uno schema sfida-risposta

Wide-Mouthed Frog Protocol

Alice sceglie una chiave di sessione	K_{AB}

Alice invia a **Trent** chiedendo di inviare a **Bob**	$E_{K_{AT}}[t_A \Big\|\|ID_B\Big\| K_{AB}]$
Trent invia a **Bob**	$E_{K_{TB}}[t_T \Big\|\|ID_B\Big\| K_{AB}]$

MiiM Attack to WMFP
Todo

Needham-Schroeder Protocol
Basato sull'utilizzo del nonce

$A \rightarrow T$	$ID_A \Big\|\|ID_B\Big\| r_1$
$T \rightarrow A$	$[K_S \Big\|\|ID_B\Big\| r_1 \Big\|\|E_{K_{BT}}\Big\| K_S\|\|ID_A]$
$A \rightarrow B$	$E_{K_{BT}}[K_S\|\|ID_A]$
$B \rightarrow A$	$E_{K_S}[r_2]$
$A \rightarrow B$	$E_{K_S}[R_2 - 1]$

"il jitter è importante"

[05/09] Esercizi da temi d'esame

09 May 2018 11:43

TE. 11/7/13

$p = 227$; $r_A = 100$, $r_B = 101$, $r_C = 102$ | $a = 10$, $b = 15$, $c = 50$

Calcolo le coppie di numeri

$a_A = a + b r_A = 10 + 15 \cdot 100 \equiv 140$

$b_A = b + c r_A = 15 + 50 \cdot 100 \equiv 21$

$a_B \equiv 163$ $\quad a_C \equiv 178$

$b_B \equiv 71$ $\quad b_C \equiv 121$

B -> C

$K_{BC} \equiv g_B(r_C) \equiv 163 + 71 \cdot 102 \equiv 141$

$K_{CB} \equiv g_C(r_B) \equiv 178 + 121 \cdot 101 \equiv 141$

Ci sarebbe anche K_{AC} da calcolare

A e B si accordano e scambiano

$a_A \equiv 22$ $\quad a_B \equiv 88$

$b_A \equiv 205$ $\quad b_B \equiv 77$

$$\begin{cases} a_A \equiv a + b \cdot 100 \equiv 22 \rightarrow b \equiv 66 \\ a_B \equiv a + b \cdot 101 \equiv 88 \nearrow \quad a \equiv 88 - 66 \cdot 101 \equiv 5 \\ b_A \equiv b + c \cdot 100 \equiv 205 \rightarrow 100 \cdot c \equiv 205 - 66 \equiv 139 \end{cases}$$

$c \equiv 100^{-1} \cdot 139 \bmod 227$

$100^{-1} \bmod 227 \equiv 84$

Conviene Euclide esteso

$c \equiv 84 \cdot 139 \equiv 99$

Ora è possibile calcolare le chiavi

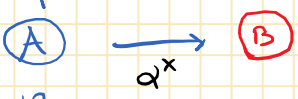
[05/10] Esercizio Diffie-Hellman da TE

10 May 2018 10:35

Protocollo Diffie-Hellman (*"Guardate che è bello questo D-H"*)

T.E 21/0716

Alice pubblica



$p = 47$
$\alpha = 4$

x segreto y segreto

Verifico se α è una radice prima di p

$\alpha \in \{3,4,5,6,7,8,9\}$ se nessuna di queste è valida
A rinuncia

$$\alpha^{\frac{p-1}{q_i}} \not\equiv 1 \bmod p$$

$p - 1 = 46 = 2 \cdot 23$

$\alpha = 4$ $\begin{cases} 4^{23} \bmod 47 = 1 \\ 4^2 \bmod 47 = 16 \end{cases}$ x 4 non va bene

$\begin{cases} 3^{23} \equiv 1 \bmod 47 \\ 3^2 \equiv 9 \bmod \end{cases}$ 3 non va bene

$\begin{cases} 5^{23} \equiv 46 \\ 5^2 \equiv 25 \end{cases}$ 5 va bene

A $\longrightarrow$ B
$\alpha^x \equiv 22$

A $\longleftarrow$ B
$\alpha^y \equiv 23$

Calcolare $K_{AB} = \alpha^{xy}$

Dato che α è radice primit. il log è valido

$5^5 \bmod 47 \equiv 23$

i	α^i	β^{-Nk}	k
0	1	22	0
1	5		1
2			2
3			3
4			4

$N = \lceil \sqrt{p-2} \rceil = 7$

$22^{-2} \bmod 47$

$x = 25 \bmod 46$
$y = 5 \bmod 46$

$K_{AB} = 22^5 \equiv 35 \pmod{47}$

[Problema computaz. di Diffie-Hellman]

[05/10] Autenticazione

10 May 2018 11:03

Si tratta di garantire l'identità di una persona/entità. l'autenticazione implica la registrazione degli utenti.
Registrazione: Claimant presenta un identificativo al verifier (il sistema di autenticazione).
Tipi di auth:

- Locale: accesso a un sistema locale
- Direct (remote) auth: l'utente accede direttamente a un sistema remoto che offre un servizio (FTP server).
- Indirect auth: ci si rivolge a un *third party* per l'autenticazione (Kerberos)
- Offline auth

Come ci si autentica:

Il client deve fornire un'informazione che conosce solo lui (PIN, password), un oggetto, un aspetto fisico (retina, impronte digitali, facial recognition), voice pattern e handwriting.
Oppure un sistema combinato (PIN + impronta e robe così).

La sicurezza dei sistemi di autenticazione sta nella trasmissione e conservazione della credenziale.
Principio di Kerchoff: la sicurezza è basata sulla segretezza della chiave.
La password non dovrebbe essere salvata e/o trasmessa in chiaro (digest hash), che comunque è sensibile all'attacco del vocabolario (per questo c'è il salting).

Entropia

X è una sorgente di informazione che genera messaggi senza memoria

$x \in \{x_1, \dots, x_N\}$

I messaggi non sono equiprobabili

$p(x_i) = P(X = x_i)$

l'informazione di un messaggio è data da:

Messaggio {A p=0.5,B p=0.1,C p=0.3,D p=0.1}

$I(x = A) = -\log_2 P(x = A) \geq 0$

Un messaggio con probabilità 1 non porta informazione (perché sai già che il messaggio sarà quello).
Evento raro = più informazione.
l'entropia è la quantità di informazione media della sorgente pesata sul singolo carattere

$H(x) = \Sigma\, P(x = X) \cdot [-\log_2 P(X = x_i)]$

Il massimo contenuto di informazione si ha per distribuzioni uniformi di probabilità.

$P(A) = P(B) = P(C) = P(D) = 0.25$

$H(X) = \frac{1}{4} \cdot (4 \cdot 2) = 2$ bit (4 valori si possono esprimere con 2 bit)

Entropia della password

Password X, combinazione di caratteri ASCII di 7 bit (2^7 combinazioni), -32 caratteri di controllo = 95 caratteri stampabili.

$H(X) = \log_2 95 = 6.57$ bit/char

Usando una parola del linguaggio reale (Alfabeto latino):

27 lettere $\Rightarrow H(X) = \log_2 27 = 4.75$ bit/char

Ma i caratteri hanno probabilità diverse, ad esempio i promessi sposi hanno entropia $H_1(X) = 3.97$ bit/char.

Entropia congiunta

Nel testo l'entropia è molto bassa (spesso sotto i 2bit/char), il contenuto informativo della password è quindi ridotto.

Come si fa a trovare una password?

- Traffic sniffing: con protocolli poco sicuri la password viene trasmessa in chiaro
- Online guessing: si tenta l'attacco a un server tirando a indovinare password
- Offline guessing: attacco del vocabolario sui password files (contrastato dal salting)

Hashcat, cain and abel, john the ripper

[05/16] cont. Autenticazione

16 May 2018 10:28

Challenge & Response

Challenge: il server può invia
re una sequenza PR
Response: qualcosa che solo l'interlocutore può calcolare, e.g.

- Chiave simmetrica (la sfida quindi può essere cifrata con la chiave condivisa)
- Funzione hash: si concatena la sfida con una chiave simmetrica condivisa e si hasha tutto
- Cifratura a chiave pubblica: il client prende la sfida e la firma con la sua chiave privata verificabile da chiunque

MiiM non funziona perché si presuppone che la C&R sia diversa in qualcosa ogni volta.
La debolezza di questo sistema sta nella memorizzazione delle password (a meno di usare il sistema di cifratura a chiave pubblica).
Rimedi:

- **One-Time Password**: (password monouso) valida solo per una sessione di login o transazione. Occorre assicurarsi che le password sia generabile solo dall'utente desiderato. c'è il token fisico che ne genera una da aggiungere eventualmente al PIN (*what you have + what you know*).
  - **Lamport's Hash Chain Scheme:** un sistema C&R per generare OTP. L'utente Alice conosce una password p, il server Bob conosce per ciascun utente (un *seed* intero n e il risultato di una funzione hash $h^n(p)$)
    - Processo di autenticazione:
      i. Il server tiene memorizzato $h^n(p)$ e si aspetta che il client invii $h^{n-1}(p)$
      ii. Il client invia $h^{n-1}(p)$
      iii. Il server calcola $h(h^{n-1}(p)) = h^n(p)$ e la confronta con quella memorizzata
      iv. Se c'è un match il server memorizza $h^{n-1}(p)$ e si aspetterà $h^{n-2}(p)$
    - Il replay attack qui non funziona perché la funzione hash è unidirezionale, quindi non ci sarebbe modo di ricevere il risultato di una hash computata più n volte una volta che h^n è già stata ricevuta.
    - Debolezze:
      - se ci sono due server e non sono ben sincronizzati possono inviare contemporaneamente lo stesso numero
      - Small-n attaccccc: dopo n autenticazioni il sistema deve essere reinizializzato
    - Rimedio: un salt pubblico s diverso per ogni server da aggiungere alla password prima di hasharla, in modo che si abbiano risultati differenti e inutilizzabili con server diversi da quello destinato
  - **Security Token**: device fisici che generano OTP che diventano invalide dopo tot secondi (necessita di sincronizzazione con il server) o si basano su un contatore che viene incrementato ad ogni utilizzo.
- **Biometric patterns**: tentativo di ovviare alle debolezze dell'utente.
  - Applicazioni:
    - Autenticazione dell'utente (matching uno a uno, eg impronte, fotografie)
    - Identificazione dell'utente (uno a molti, si ha un database dell'utenza e si tratta di trovare un utente tra gli altri)
    - Uniqueness: capire se l'user è registrato o meno
  - What you are or what you do:
    - Static biometrics: caratteristiche fisiche
      - Impronta digitale
      - Geometria della mano
      - Retina
      - Faccia
    - Dynamic biometrics: patterns comportamentali
      - Riconoscimento vocale

        - Firma
    - Precisione dell'autenticazione biometrica
        - FAR: false acceptance rate (A viene erroneamente identificata come Bob, falso positivo)
        - FFR: false rejection rate, falso negativo (Alice non viene riconosciuta come Alice)
        - Occorre stabilire una soglia di accettazione, se è troppo piccola aumenta la probabilità di FFR e contemporaneamente i FAR tendono a 0. il punto ottimo è dove le probabilità di FAR e FFR si equivalgono.
        - Le prestazioni di questi sistemi sono abbastanza deboli. L'impronta digitale è facilmente bypassabile.
- **Kerberos**: sviluppato nell'MIT. Servizio centralizzato di key-distribution e autenticazione indiretta
    - Kerberos v4: TGT. TGT+chiave, richiesta biglietto, biglietto + chiave, richiesta, autenticazione

[05/17] Certificati (PKI), TLS

17 May 2018 10:31

La chiave privata di un sistema a chiave pubblica è teoricamente calcolabile, ma la sicurezza sta nella difficoltà computazionale del calcolo effettivo.
Problemino:
Chi garantisce che la chiave pubblica sia veramente dell'interlocutore desiderato. La PKI (public key infrastructure) è l'insieme di regole e algoritmi che permettono di certificare le chiavi delle varie entità. Il certificato è firmato da una Certification Authority che dovrebbe essere affidabile (le CA si basano sulla credibilità). Per la veridicità di un certificato si controlla che sia firmato da una CA. La chiave pubblica della CA si trovano nel sistema operativo.
Ci sono due tipi di certificati:

1- Identità: certifica che il signor x ha la chiave pubblica y
2- Boh

Certificato	$A, K_A, [h(A, K_A)]_{K_{TA}^{-1}}$

La CA certifica che Alice sia Alice, ma non che sia affidabile.
Una CA può certificare un'altra CA di livello inferiore (in modo da poter certificare entità inferiori a loro volta). Se BregniSign viene verificato da VeriSign e ha qualcuno che si fida di lui potrà pubblicare la sua chiave pubblica e verificare identità con un livello inferiore.

Certificati X.500

Il certificato X.509 comprende le specifiche per la crittografia a chiave pubblica e la firma digitale.

È possibile anche revocare i certificati, c'è una lista di certificati revocati ma non ancora scaduti.

"Gli esami non si provano"

Security protocols in the TCP/IP stack

Si usa la pila TCP/IP.
(UDP oltre alla destinazione conosce la porta in cui c'è l'applicazione in ascolto).
Occorre rendere sicura la comunicazione attraverso la rete, come?

1- IPsec: usa i pacchetti IP,
   vantaggi: la comunicazione degli utenti dell'applicazione è sicura (traffico cifrato), include possibilità di filtraggio permettendo l'unione al traffico non cifrato.
2- Secure Socket Layer o Transport Level Security
   Protocolli programmati non nella rete, ma nell'applicazione (browser) e usa le primitive del TCP
3- Servizi di sicurezza inclusi nell'applicazione (sicurezza end2end)
   Il gestore della rete non ha modo di vedere cosa sta passando.

TLS (Transport Level Security)

TLS si basa su TCP, ma il protocollo TLS è formato da un record protocol e dei protocolli di servizio (eg handshake protocol, che servono a stabilire le chiavi). Il record protocol senza quelli per creare le chiavi non funziona.
La connessione TCP è definita da due end point (4 numeri, IP/portax2) i dati vengono trasmessi in maniera sequenziale e integra, ma non sicura perché è sniffabile. Sopra a TCP si può costruire TLS, che ha anch'esso il concetto di connessione come TCP e fornisce un modo sicuro di trasmettere dati tra due end point e di solito si appoggia su una TCP.
Viene creato il concetto di sessione TLS che ha un set di parametri di crittografia/algoritmi che vengono condivisi all'interno di tutte le connessioni instaurate all'interno della sessione. Un client e un server stabiliscono una sessione TLS all'interno della quale si instaurano varie connessioni TCP.
TLS Session state:

- Robe

- Robe
- Roba

TLS connection state:

- Server and client random
- robe
- Vettore inizializzazione

TLS fornisce due servizi:

- Confidenzialità: connessione cifrata
- Integrità: nella fase di handshake si usa una chiave condivisa che viene usata per formare un Message Authentication Code (MAC) , solo il destinatario può sapere se il messaggio è intero. Questo codice può essere creato solo da chi conosce il segreto (è molto di più di un CRC)

Dati applicazione > Frammentazione > Compress > Add MAC > Encrypt > Append TLS Record Header

Robe

HTTPS

È HTTP che si basa su SSL e TLS. La porta è la 443 (al posto della solita 80). Dal punto di vista dell'utente è tutto cifrato (anche la url precisa del documento in download ad esempio). Il client HTTP comincia una connessione TLS, dopodiché tutta la comunicazione avviene qui sopra. Ci sono 3 livelli di connessioni (HTTP manda una richiesta al livello inferiore (TCP o TLS se è HTTPS). HTTP chiude la connessione con close, HTTPS deve chiudere anche TLS che comporta una chiusura di una connessione TCP

[05/17] Esercizietti

17 May 2018 11:06

Certificato

TE 8/2/2013

Certificato di Alice $C_A = \left\{A, K_A, \left[h(A, K_A)\right]_{K_{TA}^{-1}}\right\}$

RSA: n=221	n=221	A=200	$K_{TA} = 35$	$K_A = 25$

Funzione hash:	$h = h(x,y) = (x \oplus SL_3(y) \oplus SL_4(y))(mod\ n)$

Tutti questi numeri sono rappresentati come parole di 8 bit

a) Verificare la correttezza dei dati forniti secondo RSA

$n = 221 = 13 \cdot 17$

$\phi(n) = 12 \cdot 16 = 192 = 2^6 \cdot 3$

$\phi(\phi(n)) = 64$

$K_{TA} \perp \phi(n) \Rightarrow \exists K_{TA}^{-1}$

$K_A \perp \phi(n) \Rightarrow \exists K_A^{-1}$

Condizione verificata

b) Calcolare d

$K_{TA}^{-1} \equiv 35^{-1} \equiv 35^{63}(mod\ 192) \equiv 11$

Ora occorre calcolare l'hash

$A_{bin} = 11001000$

$K_{TA_{bin}} = 00100011$ *non serviva*

$K_{A_{bin}} = 00011001$

$SL_3(K_A) = 11001000$

$SL_4(K_A) = 10010001$

$11001000 \oplus$

$11001000 \oplus$

$10010001 =$

$--------$

$h = 10010001$

$h_{dec} = 145$

$\{h\}_{K_{TA}^{-1}} = 145^{11}(mod\ 221)$

$C_A = \{200, 25, 202\}$

Per verificare che sia valido calcolo l'hash di 200 e 25, elevo 202 a 35 e deve essere uguale all'hash

ESERCIZI da T.E.

Ritaglio schermata acquisito: 01/06/2018
15:01

Bob adotta il *sistema di firma elettronica di El Gamal* e pubblica $p = 199$, $\alpha = 7$, $\beta = \alpha^a \bmod p$, tenendo segreto l'esponente $a = 56$.

a) Verificare la correttezza dei dati forniti, in base alle ipotesi del metodo di El Gamal. Se $\alpha = 7$ non risultasse una scelta valida, Bob userà invece $\alpha = 6$ (anch'esso da verificare). Se anche questa scelta non risultasse valida, Bob rinuncerà a proseguire (e l'esercizio termina qui). Calcolare β.

b) Bob estrae il numero casuale segreto (*nonce*) $k = 151$. Per questo valore di k, calcolare la firma $A = (r, s)$ del messaggio $P = 200$.

c) Verificare se anche la firma $A' = (r', s') = (44, 100)$ è valida per lo stesso messaggio $P = 200$.

01 June 2018 15:00

a) Verificare la correttezza dei dati

$\alpha = 7$

$\alpha^{\frac{p-1}{9}} \not\equiv 1 \bmod p$

$p - 1 = 198 = 2 \cdot 3^2 \cdot 11$

$\alpha^{99} \bmod 199$

$\alpha^{66} \equiv 7^{66} \equiv 7^{54} \cdot 7^{12} \equiv 121^3 \cdot 188 \cdot 7^3 \equiv 117 \cdot 188 \equiv 106 \bmod 199$ (63; 144)

$\alpha^{18} \equiv 7^{18} \equiv 188^2 \equiv 121 \bmod 199$

$7^{99} \equiv 106 \cdot 7^{33} \equiv 106 \cdot 121 \cdot 7^{15} \equiv 106 \cdot 121 \cdot 188 \cdot 40 \equiv 61 \cdot 121 \cdot 188 \equiv 18 \cdot 188 \equiv 1$ NO

($7^{15} = 7^9 \cdot 7^3$; 188; $144^2 = 40$)

$\alpha = 6$ Verifica

6^{18} 6^{66} 6^{99}

$6^{18} \equiv (6^9)^2 \equiv 137^2 \equiv 63 \bmod 199$

$6^{66} \equiv ((6^3)^2)^{11} \equiv (17^2)^{11} \equiv 90^{11} \equiv 90 \cdot 63^3 \equiv 90 \cdot 103 \equiv 116$

$6^{99} \equiv 6^{66} \, 6^{33} \equiv 116 \cdot 6^{33} \equiv 116 \cdot 36 \cdot 17^3 \equiv 137 \cdot 116 \cdot 36 \equiv 196 \cdot 137 \equiv 186 \bmod 199$

($6^{33} = (6^3)^3 \cdot 6^2$; $17^3 \cdot 6^2$)

$6^{56} \equiv (6^{18})^3 \cdot 6^2 \equiv 63^3 \cdot 36 \equiv 103 \cdot 36 \equiv 126 \bmod 199$

---

b) NONCE $k = 151$, calcolare firma di $P = 200$

$[A(r, s)]$

$r = \alpha^{k} \bmod p = 6^{151} \bmod 199$

$6^{151} \equiv (6^{56})^2 \cdot 6^{39} \equiv 126^{2} \cdot 63^{2} \cdot 6^{3} \equiv 155 \cdot 155 \cdot 17 \equiv 139 \cdot 48 \equiv 69 \bmod 199$

$(6^{18})^2 \cdot 6^3$ — $126 \equiv 155$, $63 \equiv 189$, $6^3 \equiv 17$

$s = k^{-1}(P - ar) \bmod (p-1)$

$151^{-1} \bmod 198$

$\phi(198) = 1 \cdot (3^2 - 3)(11 - 1) = 60$

$151^{-1} \equiv 151^{59} \bmod 198$

$\equiv (151^2)^{29} \cdot 151 \equiv 31^{29} \cdot 151 \equiv 91^{9} \cdot 169 \cdot 151 \equiv 181^{3} \cdot 169 \cdot 151$

$(31^3)^9 \, 31^2$ — $181 \equiv 37$

$\equiv 169 \cdot 151 \cdot 37 \equiv 169 \cdot 43 \equiv 139$

$139 \cdot (200 - 56 \cdot 69) \bmod 199$

$-509296 \equiv 158 \bmod 199$

$A(69, 158)$

---

c) Verif. validità $A(44, 100)$ di $P = 200$

$\beta^{r} r^{s} \equiv \alpha^{P} \bmod p$

$126^{44} \cdot 44^{100} \bmod 199 \equiv 162 \cdot 44^{100} \equiv 162 \cdot 155 \equiv 36$

145^{11}

$130^{5} \cdot 145$

$184^{2} \cdot 130 \cdot 145$

$26 \cdot 130 \cdot 145 \equiv 196 \cdot 145 \equiv 162$

130^{25}

$(26 \cdot 130)^{5}$

196^{5}

$9^{2} \cdot 196 \equiv 155$

$6^{200} \equiv (6^{99})^2 \cdot 6^2 \equiv 169 \cdot 36 \equiv$

[06/06] Protocolli (Secure Mail)

06 June 2018 10:28

Secure mail

RFC (Request for comments), standard, 5598 (numerazione cronologica)

i. Invio messaggio email (Message User Agent)
ii. Message Handling System
    a. SMTP è il protocollo usato per trasportare il messaggio da un MTA all'altro

SMTP

Protocollo che serve a portare i messaggi mail. Basato su messaggi di testo scambiati tra server e client (es: 220 foo.com ecc, il 220 indica che il server è pronto a offrire il servizio).
Per aggiungere sicurezza alla posta elettronica è necessario cifrare il messaggio e richiedere l'autenticazione del mittente (STARTTLS, aggiunge cifratura e autenticazione, una volta connesso al server SMTP il client lancia una connessione TLS, negoziazione e scambio delle chiavi, una volta terminata c'è EHLO: extended helo).

POP3 (leggere e scaricare mail dal server)

Post Office Protocol, "*io ad esempio sono un estimatore del pop3"*
Scarica le mail localmente, tutta la comunicazione pop3 è in chiaro (anche user e pw), i comandi APOP e AUTH servono ad autenticare l'utente in modo pseudo-sicuro (hashing della password)

IMAP

Lascia sul server le mail, porta 143, anche questo ha la versione col comando STARTTLS nel mezzo

RFC 5322

È il formato standard per le email.
l'header è separato dal corpo della mail da una riga vuota.

MIME (Multipurpose Internet Mail Extension)

SMTP trasferisce solo testo dei primi 128 caratteri ASCII, per inviare un file jpeg occorreva codificarlo in caratteri con Uuencode. Il MIME fa la stessa cosa in maniera automatica

Sender Policy Framework: associa domain name ad uno specifico IP per designare un tot di mittenti autorizzati e verificati (se devo inviare mail dall'estero tocca usare una VPN)

PGP

Versione spontanea di mail sicura, fa le stesse cose dell'S/MIME, freeware, freeware.
È decentralizzato (non si appoggia a CA), è possibile certificare le chiavi di altri costruendo una Web of Trust.

- Autenticazione:

Alice hasha il suo messaggio	SHA-1
Alice firma il suo hash	RSA
Alice trasmette l'hash firmato a bob all'inizio del messaggio	

- Cifratura

Alice genera un numero casuale che faccia da chiave di sessione	128-bit, 3-DES
A cifra il messaggio	
A cripta la chiave	RSA
A manda la chiave cifrata, messaggio e firma cifrati a Bob	
Bob decifra la chiave di sessione	

B usa la chiave per decifrare firma e messaggio	
B verifica la firma RSA	

IPsec

IP security, rende sicuro in questo caso il protocollo di rete.
Tutti gli utenti e le applicazioni verranno rese sicure. I pacchetti sono tutti autenticati (uno per uno).

- Autenticazione: ogni pacchetto viene firmato e si è sicuri che non è stato alterato
- Confidenzialità: robe

IPsec è una rete privata virtuale (VPN, tunnel mode). l'applicazione genera messaggi, pacchetti TCP/IP ed entra nella rete, il pacchetto IP viene "imbustato" in un pacchetto IP sicuro. ESP aggiunge un header e un trailer e si cifra ciò che contiene il pacchetto IP (non si sa a chi si scrive né cosa si scrive). Il tutto (ESP hdr, IP hdr, IP payload e ESP trl) viene firmato. Il nuovo pacchetto IP fa tutto il suo viaggio come un normale pacchetto.
Sicurezza forte senza che gli utenti se ne preoccupino della end2end security.
IPsec serve anche a mettere in sicurezza i messaggi dei protocolli di routing (OSPF, RIP), per evitare che le tabelle di routing siano inquinate da malintenzionati/vandali.

Transport mode ESP: cifra ed eventualmente autentica il payload del pacchetto IP (non l'header).
Tunnel mode: cifra l'intero pacchetto IP aggiungendo i campi addizionali, diventando il payload di un nuovo pacchetto IP. Security Association (coppia di security gateways o router IPsec)
IP security policy, in base a una serie di parametri applica a ciascun pacchetto una policy di sicurezza.

Domande di teoria

25 June 2018 09:48

4) Definire la proprietà *debolmente resistente alle collisioni* di una funzione di hash. Perché questa proprietà è più facile da soddisfare della proprietà *fortemente resistente alle collisioni*? *(2 punti)*

5) Descrivere lo *Schema di Lamport* per l'autenticazione di un host A da parte di un server B, precisando quali informazioni sono pubbliche o trasferite in chiaro, e quali altre sono private e memorizzate in A o B. *(3 punti)*

Ritaglio schermata acquisito: 25/06/2018 09:52

1) Cos'è un *elemento primitivo* $\alpha \in \mathbb{Z}_p^*$? Quanti sono gli elementi primitivi di $\mathbb{Z}_{1109}^*$? Qual è l'*ordine* dell'elemento $\alpha = 4$ in $\mathbb{Z}_{1109}^*$? *(3 punti)*

Ritaglio schermata acquisito: 25/06/2018 15:18

2) Si consideri la sequenza binaria pseudo-casuale $\{x_i\}$ generata dall'algoritmo *Blum-Blum-Shab* per $p = 83$, $q = 139$. In base alla teoria, qual è il valore massimo che può assumere il suo periodo $P = \pi(x_0)$ per valori arbitrari del seme $x_0 = x^2 \in \mathbb{Z}_n$? Si ricorda che $\pi(x_0)$ divide $\lambda(\lambda(n))$, dove $\lambda(n) := \text{mcm}\left(\left\{\varphi\left(p_i^{a_i}\right)\right\}\right)$ è la Funzione di Charmichael. *(2 punti)*

Ritaglio schermata acquisito: 25/06/2018 16:15

3) Descrivere la procedura di autenticazione e cifratura di un messaggio PGP inviato da Alice a Bob basato sugli algoritmi RSA, 3DES, SHA, DSA (firma El Gamal). *(3 punti)*

Ritaglio schermata acquisito: 25/06/2018 16:16

4) A cosa serve il Protocollo di Needham-Schroeder? Chi sono gli interlocutori del protocollo? Qual è la sua caratteristica principale e come evita i *replay attack*? *(3 punti)*

Ritaglio schermata acquisito: 25/06/2018 16:17