Cubica è definita da 9 parametri a meno di proporzionalità:

$$a_1 X_1^3 + a_2 X_1^2 X_2 + a_3 X_1 X_2^2 + a_4 X_1 X_2 X_3 + a_5 X_1^2 X_3 +$$
$$a_6 X_1 X_3^2 + a_7 X_2^3 + a_8 X_2^2 X_3 + a_9 X_2 X_3^2 + a_{10} X_3^3 = 0$$

DATI 9 punti in pos. generale
$\exists!$ cubica che passa per essi

$P_1 = (\bar{x}_{11}\ \bar{x}_{12}\ \bar{x}_{13})$
$P_2 = (\bar{x}_{21}\ \bar{x}_{22}\ \bar{x}_{23})$
$\vdots$
$P_9 = (\bar{x}_{91}\ \ldots\ \bar{x}_{93})$

→ sistema lineare omogeneo in 10 incognite → ∞^2 soluzioni (salvo sorprese).

Due cubiche distinte si intersecano in 9 punti → se ci sono 9 reali in $\mathbb{R}\mathbb{P}^2$ ci sono sicuramente 9.

Oss: Date 7 cubiche che si intersecano in 8 punti $\Rightarrow \exists \infty^2$ cubiche che passano tutte per 8 punti di cui 8 sono quelli dati ed il nono si trova...

DLOG $\rightarrow$ DLOG su curve ellittiche.

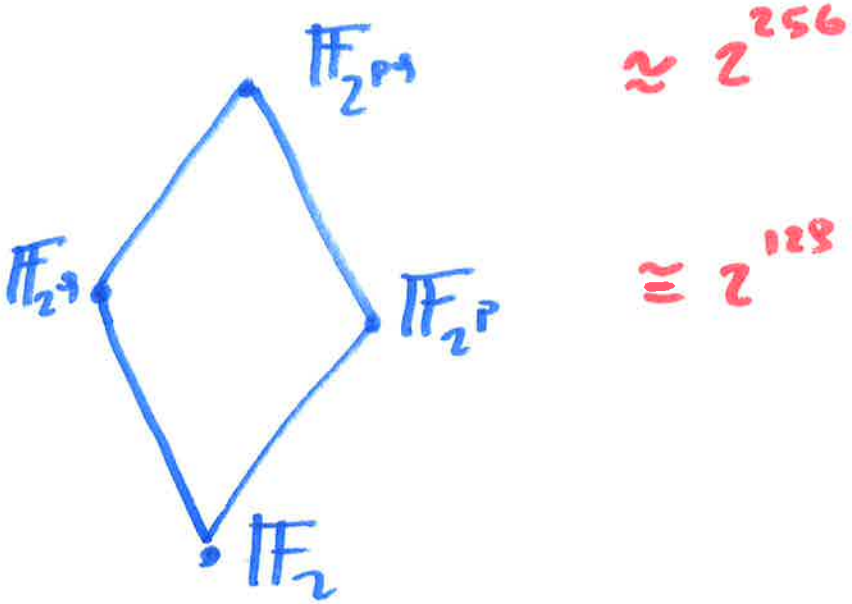
Su $\mathbb{F}_{2^n}$ perché n primo?

$\mathbb{F}_{2^{pq}}$ campo di ordine $\mathbb{Z}_2^{pq}$.

$\Rightarrow$ si può vedere convenientemente come

$$\frac{\mathbb{F}_2[x]}{(p(x))} \qquad \frac{\mathbb{F}_{2^p}[x]}{(p'(x))} \qquad \frac{\mathbb{F}_{2^q}[x]}{(q'(x))}$$

$\deg p(x) = pq$ $\quad \deg p'(x) = q$ $\quad \deg q'(x) = p$



Invece che studiare la curva in $\mathbb{F}_{2^{256}}$ lo studio (ridotto) in $\mathbb{F}_{2^{128}}$ e $\mathbb{F}_{2^{128}}$ e poi cerco di "incollare" l'informazione.

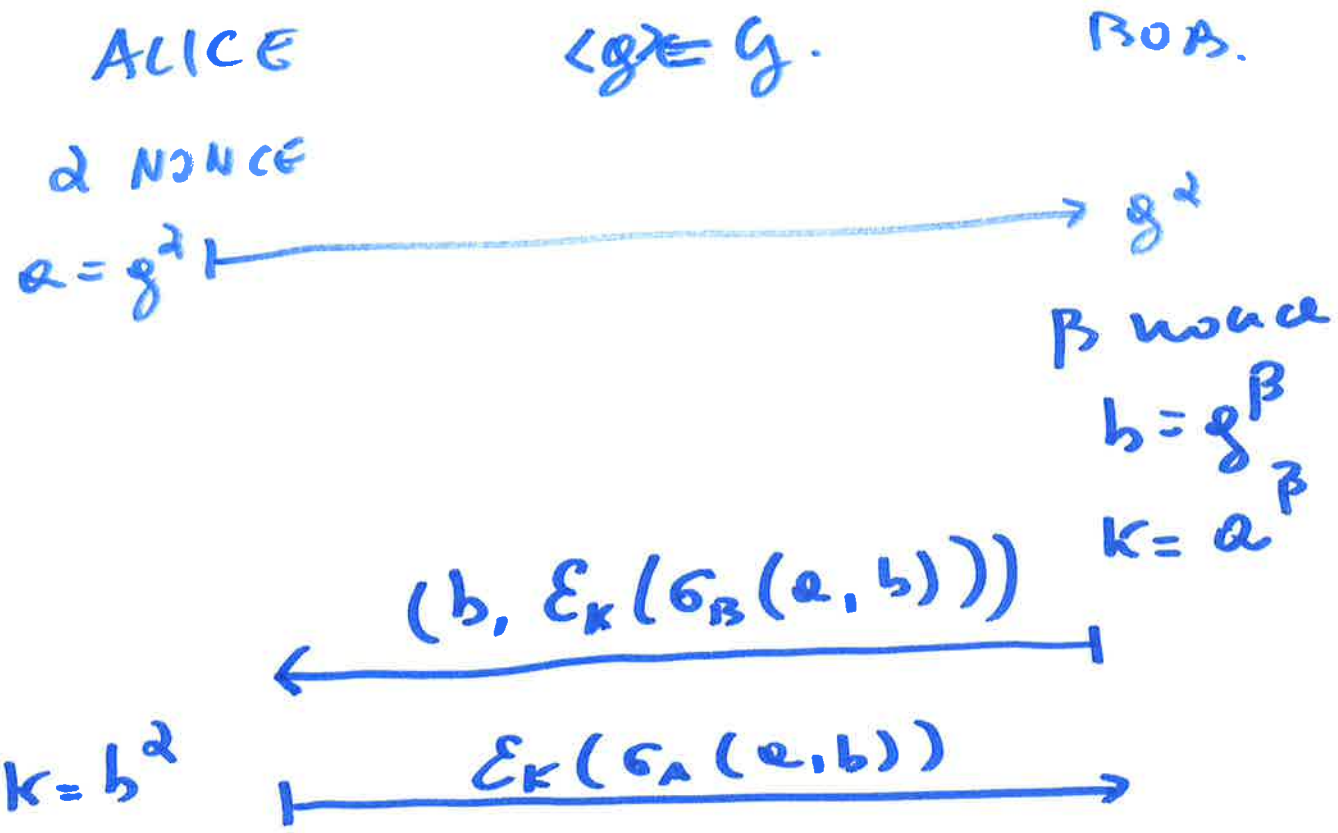
DH: 2 messaggi → NON AUTENTICATO (MITM).

STS → 3 messaggi

MQV → è possibile avere autenticazione in DH con 2 soli messaggi?

(data una infrastruttura di chiave pubblica).

STS



MAV → far sì che il calcolo di k dipenda dalle identità pubbliche e dalle chiavi private di ALICE e BOB.

$K_{ALICE} = f(\alpha, b, B_{pub}, A_{priv})$ ← identità di ALICE privata

$K_{BOB} = f(\beta, a, A_{pub}, B_{priv})$ identità di Bob pubblica.

HMQV.

Chiavi pubbliche.

$\langle g \rangle = G.$

ALICE

$a \rightarrow$ esponente segreto

BOB.

$b \rightarrow$ esponente segreto.

$A = g^a$

$B = g^b.$

A, B identità pubbliche.

α : NONCE

$X = g^\alpha$

β NONCE.

$y = g^\beta$

$X \longmapsto X$

$y \longmapsto y$

$e = H(X, B) \qquad d = H(y, A)$ entrambi.

$\sigma_A = (y \cdot B^e)^{d+da}$

$\sigma_B = (XA^d)^{\beta+eb}$

ASSERIAMO CHE

$G_A = G_B$

$$G_A = (y B^e)^{\alpha + d e} = \text{(chiave privata di ALICE)}$$

$$= (g^{\beta} \cdot g^{be})^{\alpha + de} =$$

$$= g^{(\beta + be)(\alpha + de)}$$

$$G_B = (X \cdot A^d)^{\beta + eb} = \text{(chiave privata di BOB)}$$

$$= (g^{\alpha} \cdot g^{ad})^{\beta + eb} = g^{(\alpha + ad)(\beta + eb)}$$

□

in particolare $H(G_A) = H(G_B)$

∀ funzione hash che vogliamo usare. → ricaviamo una chiave k comune fra ALICE e BOB.

N.B: la chiave dipende dalle identità delle persone.

ECDSA

$e = H(M)$

k NONCE.

$(x_1, y_1) = kG.$

$x_1 = f(kG)$ con G gen del gruppo della cubica ellittica

$x_1 \longmapsto \bar{x}_1$ inverso

$r \neq 0.$

$r = \bar{x}_1 \bmod n.$

$s = k^{-1}(e + dr) \bmod n$

(r, s) firma.

2 inversi!

verifica.

$c = (s^{-1}) \bmod n$

$u_1 = ec \bmod n$

$u_2 = rc \bmod n$

$(x_1, y_2) = u_1 G \oplus u_2 Q.$ → in E.

si verifica $\bar{x}_1 = r \bmod n$

DSA

x privata
y pubblica
k NONCE

M messaggio.

$y = g^x$

$r = (g^k \bmod p) \bmod q$

$z = Hash(M)$.

$s = k^{-1}(z + x \cdot r) \bmod q$.

$k^{-1} \bmod q$.

$\mathbb{F}_p \quad \mathbb{F}_q$.

(r, s) DUE INTERI. VERIFICA

$w = (s)^{-1} \bmod q$.

$z = Hash(M)$

$u_1 = zw \bmod q$

$u_2 = (rw) \bmod q$.

$v = (g^{u_1} \cdot y^{u_2} \bmod p) \bmod q$.

$v = r$?

in $\mathbb{F}_p$

DRBG

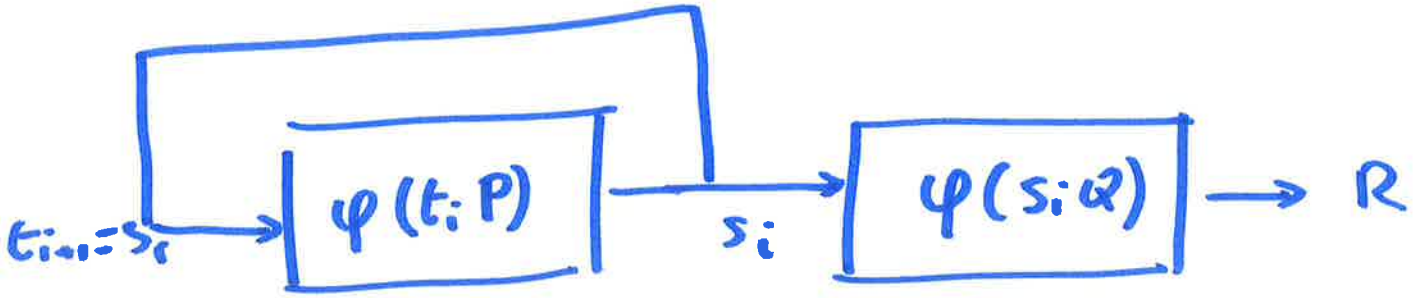
deterministic random bit generator.

s → $f(s)$ è univocamente determinata da s.

random bit generator → deve essere "difficile" prevedere $f(s)$ dato s.

più in generale noi abbiamo un algoritmo che produce uno stream di bit e noto l'output nelle posizioni $1 \dots n$ non deve essere possibile prevedere con $p \neq \frac{1}{2}$ il bit $(n+1)$ senza conoscere lo stato iniziale.

Avversario: dati i bit in output.
in pos. $1 \dots n$ prevedere l'output in pos $(n+1)$. → i bit devono essere possibili anche per più stati iniziali.



supponiamo $\boxed{P = a\,Q.}$

$\Rightarrow \quad s_i Q = t_{i+1} P$

ma $\quad s_i P = s_i a Q = t_{i+1} P$

Generare P e Q in modo tale che $P = a\,Q$.

Supp che l'output del DRBG sia proprio $s_i Q$ al tempo i.

$\Rightarrow a(s_i Q) = s_i P$ ma al tempo $(i+1)$ lo stato interno del sistema è $t_{i+1} P = s_i P$.

⇒ da $s_i Q$ sapete ricavare lo stato interno ⇒ predire tutti i valori futuri.

pb: ricavare $s_i Q$ da $\varphi(s_i Q)$.

per EC_DRBG

$$\varphi(s_i Q)$$

rappresenta la x di $s_i Q$ come un intero. (eventualmente con troncamento).

DA $\varphi(s_i Q)$ si può risalire a

$Q_i = s_i Q \Rightarrow P_{i+1} = a Q_i$

[EC_DRBG NON È DA USARE!]

Funzioni Hash.

$$f: M \longrightarrow A^n$$

f associa a messaggi arbitrari stringhe di lunghezza fissata

$$h_a : \begin{cases} M \longrightarrow A \\ a_0 a_1 \dots a_n \longrightarrow a_1 \end{cases}$$

$h_a \rightarrow$ induce l'ordine lessicografico su M.

Definiamo "$\leq$" sull'insieme A dei caratteri. $w_1, w_2 \in M$

$$w_1 \leq w_2 \iff h_a(w_{11}) < h_a(w_{21})$$

oppure $h_a(w_{11}) = h_a(w_{21})$ &

$$w_{12} \dots w_{1n} < w_{21} \dots w_{2n}.$$

Cioè w_1 viene prima di w_2

$\Leftrightarrow$ il primo carattere di w_1 precede il primo carattere di w_2 oppure è uguale e la parola ottenuta cancellando il primo carattere da w_1 viene prima della parola ottenuta cancellando il primo carattere da w_2

albero < casa perché a < c

casa < cosa perché c = c ma asa < osa perché a < o

<u>oss</u>: le funzioni hash di questo tipo non sono "bilanciate".

↓

per un db si vorrebbe che i dati fossero ripartiti in classi

aggiungendo tutte della stessa dimensione.

MDC = MODIFICATION DETECTION CODE
(hash crittografico).

$h: M \to A^n$ hash tale che

1) h è resistente alle preimmagini
i.e. dato $a \in A^n$ è difficile
trovare $m \in M$ con $f(m) = a$

2) h è resistente alle II preimmagini
i.e. dato $m \in M$ è difficile
trovare m' con $h(m) = h(m')$.

3) h è resistente alle collisioni
cioè è difficile trovare
m, m' con $h(m) = h(m')$.

2) principio delle piccionaia → $|A^n|$ tentativi

3) paradosso dei compleanni $\geq \sqrt{|A^n|}$ tentativi