Cifrario di ElGamal su curve ellittiche

Necessità di incapsulare un messaggio $m < p$ in un punto di una curva ellittica. Cerchiamo una funzione

$$\text{MSG} \to E_p(a, b)$$

- se sostituisco m a x, il valore di $x^3 + ax + b$ corrisponde a un residuo quadratico con probabilità $\simeq 1/2$, che non è abbastanza;
- con l'algoritmo di Koblitz la probabilità di fallimento scende a $1/2^h$, e se un messaggio non è convertibile in punto si può modificare e riprovare.

$E_p(a, b)$ sicura (su alcune log discreto più facile) punto base B di ordine n elevato

Chiavi

L'utente u genera la coppia:

- $1 < k_{\text{priv}} = n_u < n$ scelto casualmente;
- $k_{\text{pub}} = n_u B = P_u$.

Ricavare k_{priv} da k_{pub} richiede di calcolare un logaritmo discreto su curva ellittica, che ha costo esponenziale.

Cifratura

L'utente A, per cifrare un messaggio per B:

- trova un punto P_m corrispondete a m con l'algoritmo di Koblitz;
- sceglie casualmente $1 < r < n$;
- calcola $V = rB$ (polinomiale con raddoppi ripetuti);
- calcola $W = P_m + rP_B$.

Il crittogramma da inviare è la coppia (V, W).

Decifrazione

$$P_m = (x, y) = W - n_B V \qquad m = \left\lfloor \frac{x}{h} \right\rfloor,$$

infatti:

$$\begin{aligned} W - n_B V &= P_m + rP_B - n_B V \\ &= P_m + r(n_B B) - n_B(rB) \\ &= P_m. \end{aligned}$$

Sicurezza

Attacchi:

- trovare le chiavi private da quelle pubbliche – logaritmo discreto;
- scoprire r: $P_m = W - rP_B$, ma se è generato correttamente l'unico modo per ricavarlo è un logaritmo discreto ($V = rB$);
- man-in-the-middle.

Il miglior algoritmo noto per il logaritmo discreto su macchine tradizionali ha complessità $\mathrm{O}(2^{\frac{n}{2}})$, perciò i bit di sicurezza sono la metà dei bit dell'ordine $(\lceil \log_2 n \rceil)$.