Analisi su vulnerabilità del generatore di numeri pseudo-casuali su curve ellittiche Dual EC-DRBG

Tesi di Laurea in

Ingegneria Informatica

Candidato
Alex Parri

Relatori
Prof. Giuseppe Lettieri
Dott. Gaspare Ferraro

UNIVERSITÀ DI PISA

Introduzione e curve ellittiche

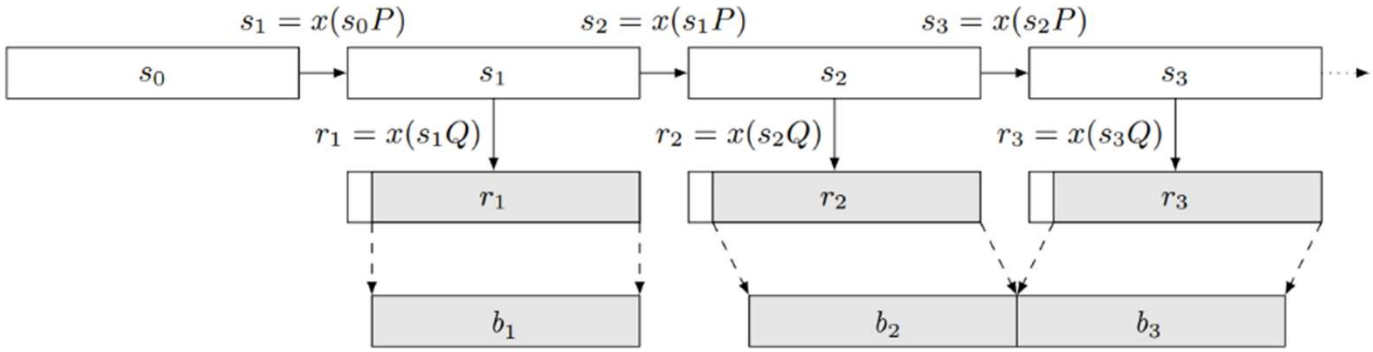
- La Crittografia ad oggi è fondamentale per tutti
  - A chiave pubblica — due coppie di chiavi per ogni utente
  - A chiave private — unica chiave condivisa tra tutti gli utenti
- Sistema alternativo alla crittografia pubblica è basato su curve ellittiche, in particolare quelle definite su campi finiti modulo p, cioè coppie di punti $(x, y) \in \mathbb{Z}_p^2$ verificanti l'equazione:

$$y^2 \equiv x^3 + ax + b \pmod{p} \quad a, b \in \mathbb{Z}_p \ (p \in \mathbb{N} > 3)$$

- Sono interessanti per via dell'operazione di moltiplicazione scalare con punti della curva, funzione $f(x)$ detta one-way, ovvero
  - $f(x) = y$ — richiede tempo polinomiale conoscendo x
  - $x = f^{-1}(y)$ — richiede tempo esponenziale anche conoscendo y
- Praticamente, con x scalare e P punto della curva
  - $A = P + \cdots + P = xP$ — facile
  - $x = \log_P A$ — difficile (problema del logaritmo discreto su CE)

Generatore EC-DRBG e vulnerabilità

- NIST (SP 800-90A) divenuto standard quasi immediatamente
- Ha lo scopo di generare stringhe casuali di bit
- È un PRNG (crittograficamente sicuro) cioè genera stringhe "sufficientemente casuali" di bit per essere usato in ambito crittografico, verificano test statistici molto stringenti
- Setup: punti P, Q generati dall'NSA, seme s_0 generato "bene"



- Problema: relazione (backdoor) $P = eQ$ inserita dal NIST in fase di design, permette di risalire allo stato interno dell'iterazione successiva, s_{i+1} partendo da un output b_i (pubblico)

Proof of Concept e risultati

- Attacco a ritroso, partendo da una singola stringa di bit in output
  1) Generazione di tutte le 2^{16} combinazioni dei 16 bit scartati e concatenazione di esse con i bit in output, ottenendo un array di $r_{x\,i}$
  2) Sostituzione nell'equazione della curva, trovando circa 2^{15} soluzioni r_{y_i} (residui quadratici), una delle quali sarà la $R = (\bar{r}_{x_i}, \bar{r}_{y_i})$ di partenza
  3) Quest'ultima sarà tale che $eR = e(s_i Q) = s_i(eQ) = s_i P$, notare $P = eQ$
  4) La coordinata x di quest'ultima sarà s_{i+1}.
- Programma scritto in Python, con backdoor inserita di proposito e script batch per l'esecuzione di 20 test per 5 curve differenti
- L'attacco impiega in media 3min (aumentare bit non cambia!)
- Soluzioni:
  - Aumentare i bit scartati da ogni iterazione (rallenterebbe il generatore)
  - Generarsi i propri P, Q (rischioso e nega la validazione FIPS 140)
  - Usare un altro generatore (il DRBG è sconsigliato da molti anni ormai)
- Conclusione: EC-DRBG è un generatore insicuro e da non impiegare