DESCrack

Programmazione di Sistemi Multicore – a. a. 2019-2020

Stefano Scannavini

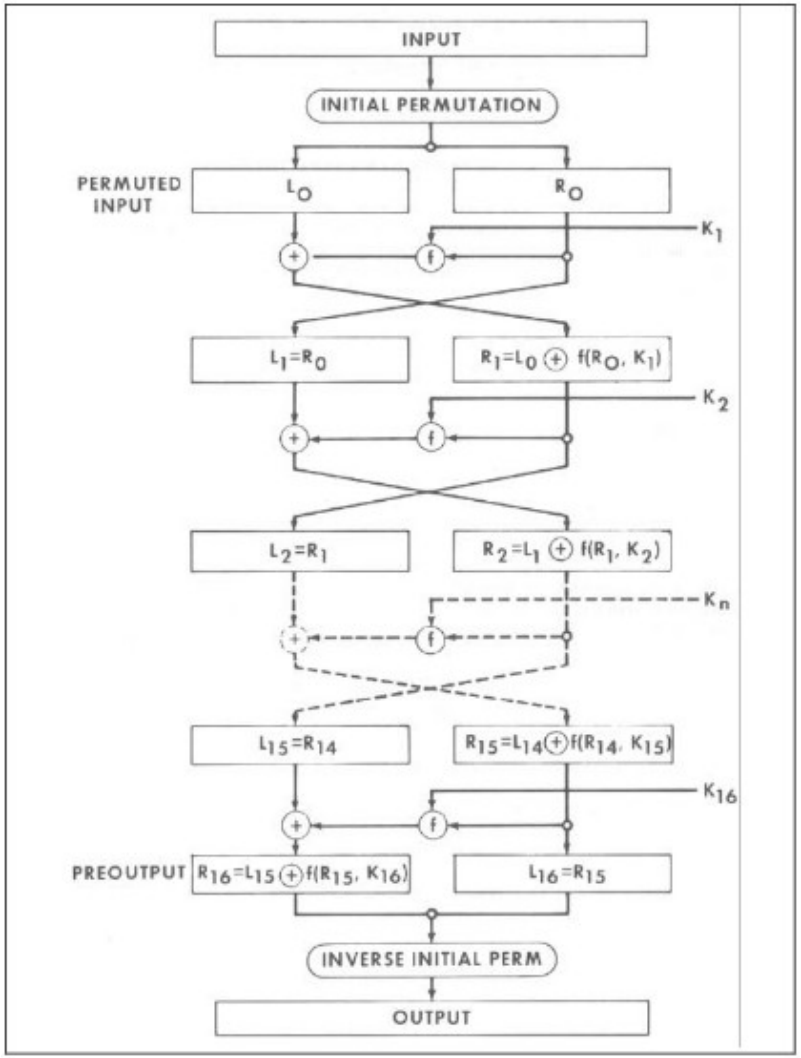
DES: breve storia

- Sviluppato da IBM nel 1973
- Diventò standard nel 1976
- Pubblicato nel 1977 come FIPS 46
- Riconfermato nel 1983, 1988, 1993, 1999
- Attacco bruteforce nel 1998
- Sostituito da AES nel 2002
- Ritirato nel 2005

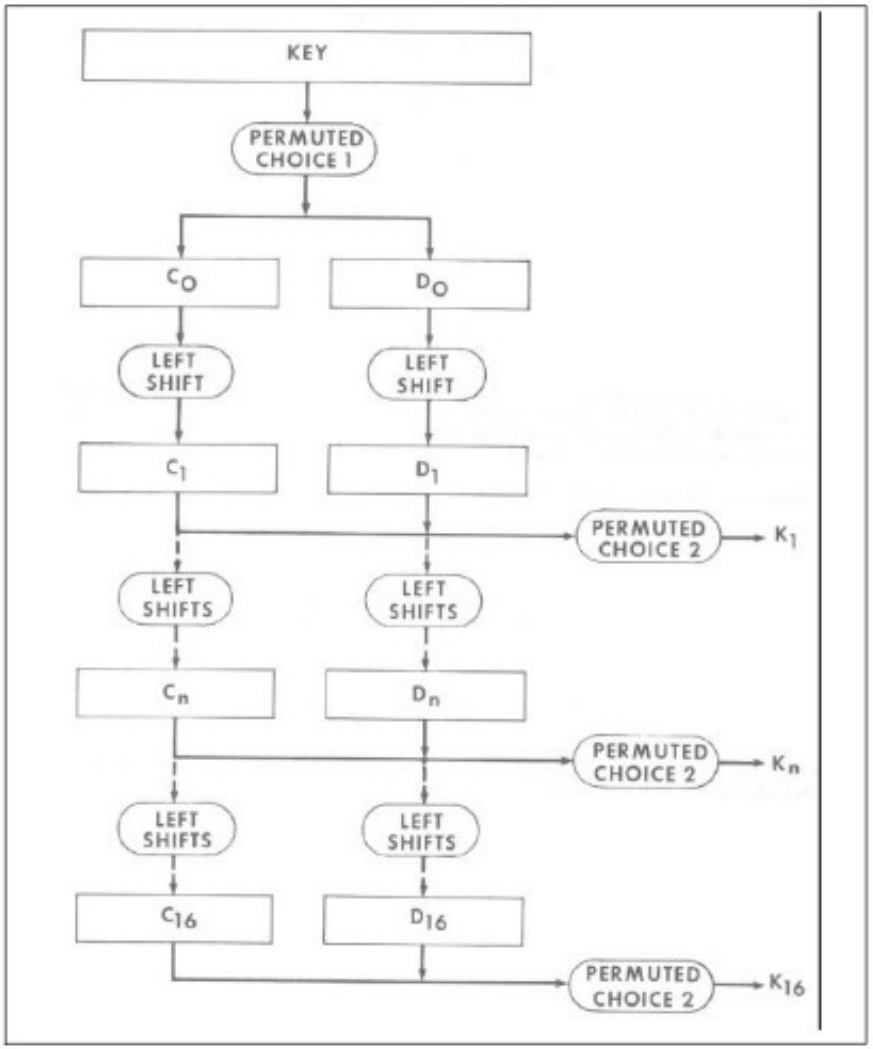
DES: caratteristiche principali

- **Algoritmo a chiave simmetrica**
  - stesso algoritmo per cifratura e decifratura
- **Cifrario a blocchi**
  - messaggio diviso in blocchi (di 64 bit)
  - diverse modalità: ECB, CBC, …
- **Pensato per realizzazione hardware**

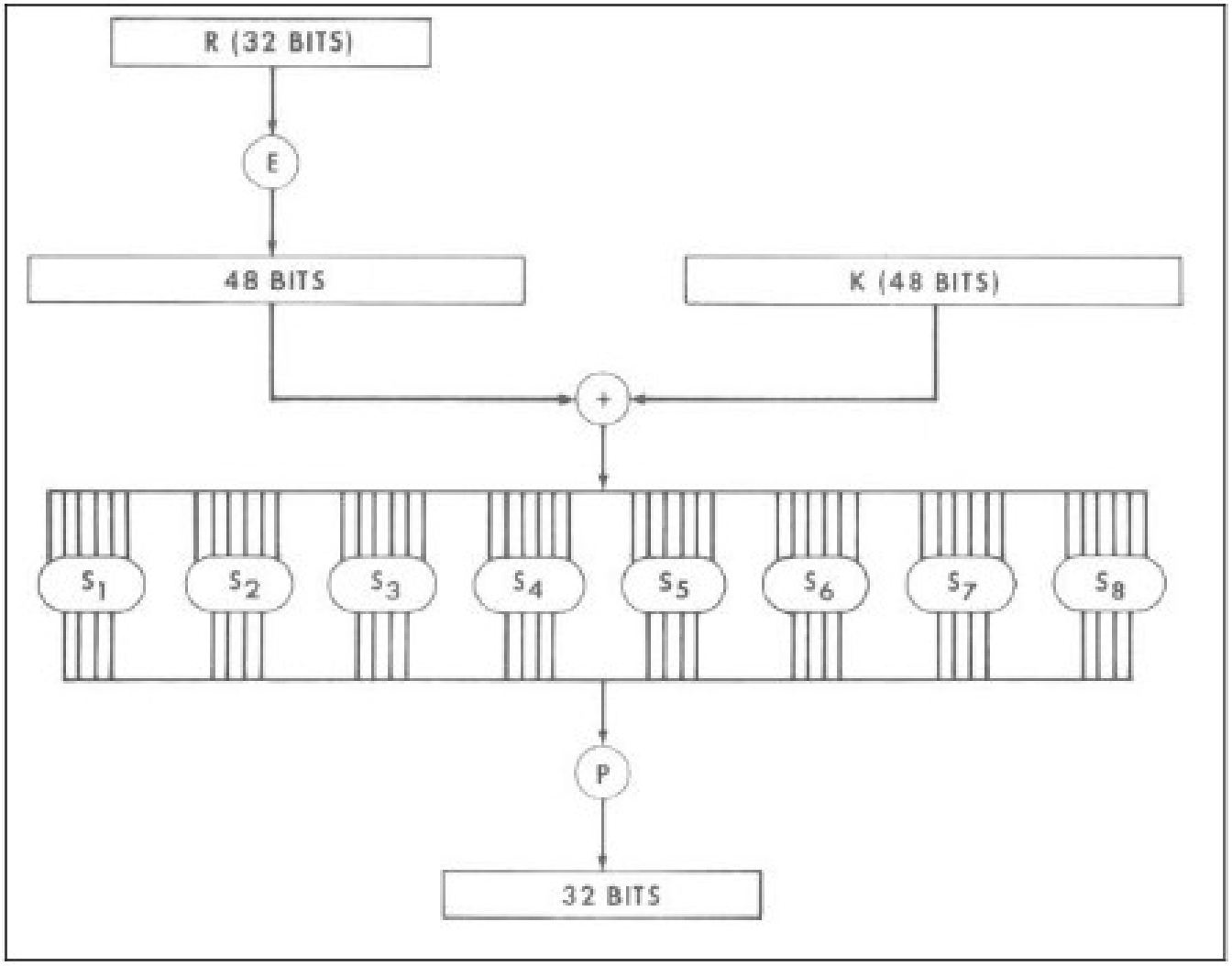
DES: flusso principale



DES: generazione sottochiavi



DES: funzione di cifratura



DES: chiave

- **Lunghezza chiave limitata**
  - 64 bit
- **Bit di parità riducono lunghezza effettiva**
  - da 64 bit a 56 bit
- **Chiave ASCII-only è ancora più corta**
  - da 56 bit a 48 bit

DESCrack

- **Ipotesi:**
  - modalità ECB, blocco singolo
  - padding di zeri a destra con chiavi “corte”
  - N = |alfabeto|
- **Generazione chiave:**

  $chiave[i] = alfabeto[(v \bmod N^{i+1}) / N^{i}] \; \forall \; i = 0..7$

Diamo un’occhiata al codice!

DESCrack con MPI

- **Ipotesi aggiuntiva:**
  - n° processi < N
- **Generazione chiave:**
  - stessa funzione
  - ogni processo ha un proprio intervallo per v

Diamo un’occhiata al codice!

DESCrack con CUDA

- **Generazione chiave:**

  chiave[0] = alfabeto[v] con v = 0..N

  chiave[1] = alfabeto[threadId.x]

  chiave[2..4] = alfabeto[blockId.* mod N]

  chiave[5..7] = alfabeto[blockId.* / N]

Diamo un’occhiata al codice!

Bonus: DESCrack con OpenMP

- **Generazione chiave:**
  - come MPI, ma…
  - …la libreria si occupa di suddividere il task
- **Non è possibile interrompere l’iterazione**
- **Basta l’aggiunta di una direttiva!**

Diamo un’occhiata al codice!

Bonus: DESCrack con Pthreads

- **Generazione chiave:**
  - come OpenMP, ma…
  - …bisogna esplicitare tutti i passaggi

Diamo un’occhiata al codice!

Confronto prestazioni

	8 bit	16 bit	24 bit	32 bit	48 bit
Base	0,001 s	0,013 s	1,154 s	151,931 s	19858,328 s
Threads	0,001 s (+0,00)	0,003 s (+4,33)	0,303 s (+3,81)	39,819 s (+3,82)	5254,081 s (+3,78)
OpenMP	0,001 s (+0,00)	0,004 s (+3,25)	0,303 s (+3,81)	40,004 s (+3,80)	5258,905 s (+3,78)
MPI	0,102 s (-102,00)	0,114 s (-8,77)	0,425 s (+2,72)	40,219 s (+3,78)	5274,099 s (+3,77)
CUDA	0,286 s (-286,00)	0,288 s (-22,15)	0,323 s (+3,57)	3,963 s (+38,34)	470,640 s (+42,19)
CPU: INTEL i5-3570k			**GPU: NVIDIA Geforce GTX 770**		

Possibili miglioramenti

- **Cambiare implementazione DES**
  - libdes, inclusa in OpenSSL
- **Togliere limitazione su versione MPI**
  - modifica al calcolo dell'intervallo di valori
- **Codice DEVICE su file unico**
  - leggero aumento di prestazioni
- **Implementazione su FPGA!**

Fonti

- https://it.wikipedia.org/wiki/Data_Encryption_Standard#Cronologia
- https://csrc.nist.gov/CSRC/media/Publications/fips/46/3/archive/1999-10-25/documents/fips46-3.pdf
- http://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm
- https://www.open-mpi.org/doc/current/
- https://docs.nvidia.com/cuda/archive/10.2/
- https://developer.nvidia.com/blog/separate-compilation-linking-cuda-device-code/
- https://www.openmp.org/wp-content/uploads/openmp-4.5.pdf
- https://crack.sh
- https://cmake.org/cmake/help/latest/
- https://developer.nvidia.com/blog/building-cuda-applications-cmake/
- https://colab.research.google.com