FAKULTA
APLIKOVANÝCH VĚD
ZÁPADOČESKÉ
UNIVERZITY
V PLZNI

Semestrální práce z předmětu KIV/BIT

Implementace Data Encryption Standard

Patrik Harag

harag@students.zcu.cz

(A15B0034P)

Celková doba řešení: 24 h

19. dubna 2018

1 Zadání

Předmětem této práce je implementace blokové symetrické šifry *Data Encryption Standard* (*DES*) s vybranými operačními módy.

2 Analýza

Data Encryption Standard Je symetrická bloková šifra. V USA byla od sedmdesátých let standardem pro šifrování dat v civilních státních organizacích, až do roku 2002, kdy byla nahrazena šifrou *Advanced Encryption Standard*. Archivovaná verze původního standardu [1], byla použita jako hlavní zdroj informací o DES.

Šifra pracuje s 64bitovým blokem dat a 16 klíči o 48 bitech. Dešifrování se liší pouze opačným pořadím klíčů. Vstupní blok je nejprve permutován podle tabulky definované v [1] a je rozdělen na levou a pravou část. Následuje 16 iterací:

$$R_{n-1} = L_n$$

$$L_{n-1} = R_n \oplus f(L_n, K_n)$$

kde L_n je levá strana iterace n, P_n je pravá strana iterace n, K_n je klíč iterace n, operátor $\oplus$ je XOR a funkce f je Feistelova funkce. Po poslední iteraci jsou strany naposledy prohozeny a sjednoceny. Vzniklý blok je permutován podle tabulky definované v [1]. Celé schéma je vidět na obr. 1.

Feistelova funkce se skládá z expanze vstupního půl-bloku na 48 bitů podle tabulky definované v [1], po kterém následuje XOR s klíčem. Vzniklý mezivýsledek je rozdělen na 8 částí po 6 bitech. Tyto části jsou podle tabulek z [1] nahrazeny 4bitovými čísly (první + poslední bit = řádek, prostřední 4 bity = sloupec) a opět sloučeny do půl-bloku. Vzniklý půl-blok je permutován podle tabulky definované v [1]. Celé schéma je vidět na obr. 2.

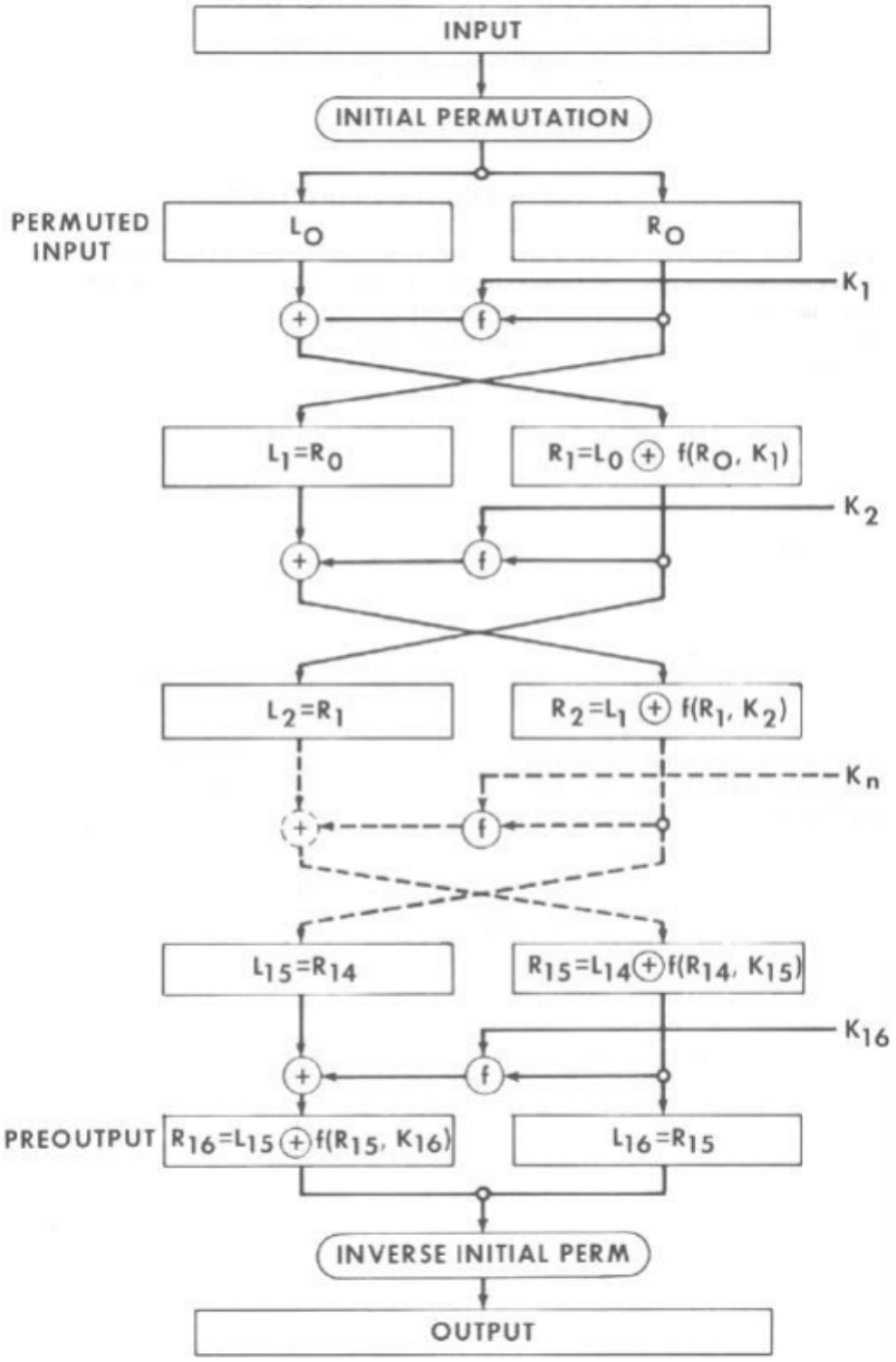
Oněch 16 klíčů o 48 bitech, které se používají v jednotlivých iteracích algoritmu je možné vygenerovat z jednoho 64bitového klíče. Schéma ukazuje obr. 3.

Operační módy DES pracuje pouze s 64bitovými bloky dat. Existuje více způsobů, jak přistoupit k šifrování/dešifrování většího počtu boků. Souhrnně se jim říká operační módy a nejedná se pouze o záležitost DES, ale blokových šifer obecně.

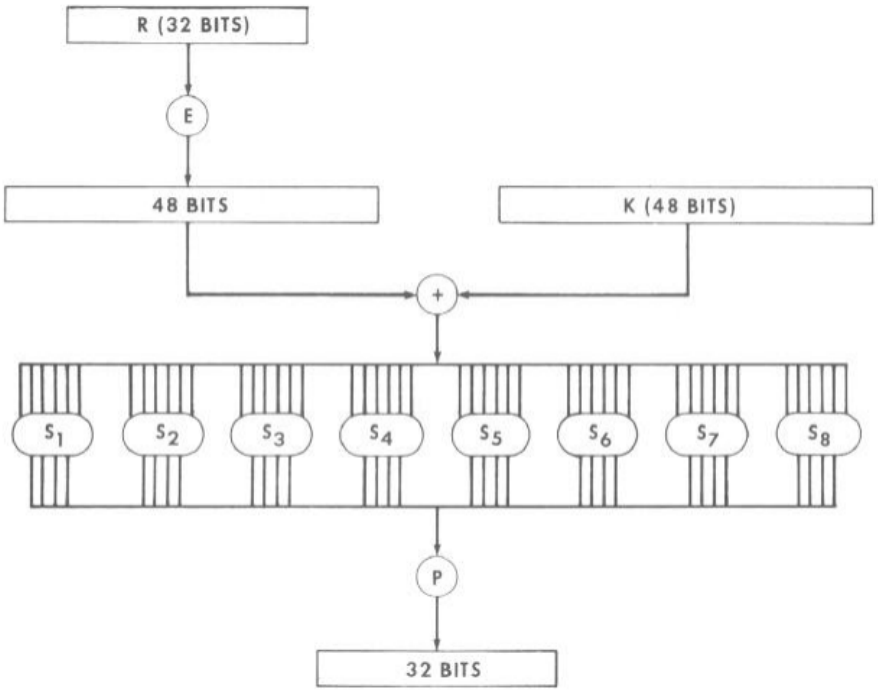
Nejjednodušší mód se nazývá *Electronic Codebook* (ECB) a spočívá v pouhém rozdělení vstupních dat na bloky, které šifra zvládne a jejich postupném zašifrování/dešifrování.

Další významný mód se nazývá *Cipher Block Chaining* (CBC) a řeší jeden z problémů ECB, kterém je skutečnost, že stejné vstupní bloky jsou po zašifrování reprezentovány stejně. Docílí toho tak, že při šifrování provádí XOR bloku s předešlým blokem po zašifrování a teprve tento výsledek zašifruje. Při dešifrování naopak provádí XOR právě dešifrovaného bloku s předešlým blokem před dešifrováním. Vyžaduje tzv. inicializační vektor pro první blok.

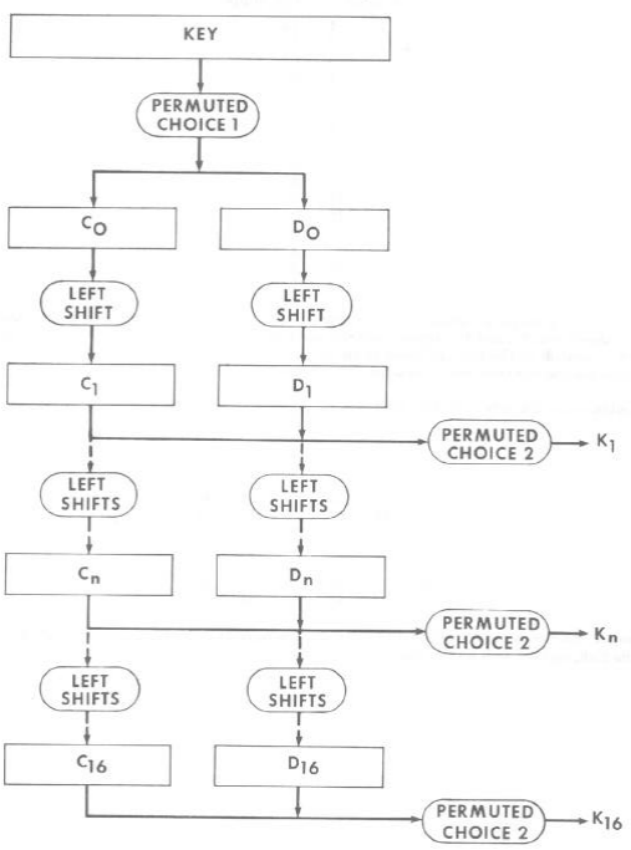
Možnosti paralelizace Iterativní charakter DES paralelizaci dosti komplikuje, ale efektivní paralelizaci by mělo být možné snadno zavést na úrovni operačního módu ECB.



Obrázek 1: Schéma algoritmu DES [1]



Obrázek 2: Schéma Feistelovy funkce [1]



Obrázek 3: Schéma generování klíčů [1]

3 Návrh

Bude implementována šifra DES s módy *Electronic Codebook* (ECB), paralelní ECB a *Cipher Block Chaining* (CBC). Dále bude vytvořen program s CLI, který umožní zašifrování a dešifrování souborů. Jádro bude naprogramováno v jazyce Java, testy a nekritická logika v dynamickém jazyce Groovy.

4 Popis implementace

Byl vytvořen program umožňující šifrování a dešifrování pomocí DES.

4.1 Seznam tříd

- `Main` – Vstupní třída zpracovávající parametry příkazové řádky.
- `DES` – Poskytuje metody pro šifrování a dešifrování vstupních proudů (třída `InputStream`) pomocí DES.
- `DESCore` – Stará se o šifrování a dešifrování jednoho bloku.
- `DESKeyGenerator` – Třída pro generování klíčů. Vygeneruje 16 klíčů o 48 bitech z jednoho 64 bitového klíče.
- `BitUtils` – Pomocné metody pro práci s bloky.
- `LongInputStream` – Vlastní implementace vstupního proudu pro načítání čísel typu `long` pracující s třídou `InputStream`.
- `LongInputStreamPlainText` – Speciální typ `LongInputStream` pro plain text.
- `LongInputStreamCipherText` – Speciální typ `LongInputStream` pro zašifrovaný text.
- `LongOutputStream` – Vlastní implementace výstupního proudu pro zápis čísel typu `long` pracující s třídou `OutputStream`.

4.2 Řešené problémy

Zarovnání do bloku DES je navržen tak, že pracuje pouze s 64bitovými bloky. Pokud nastane situace, kdy potřebujeme zašifrovat kratší blok, musíme ho prodloužit na 64 bitů. Po dešifrování však nastane situace, kdy je dešifrovaný blok jiný než blok na vstupu a není možné jednoznačně určit, zda byl prodloužen.

Tento problém jsem vyřešil tak, že na začátek zašifrovaného souboru ukládám délku původního souboru. Třídy `LongInputStream`, `LongInputStreamPlainText`, `LongInputStreamCipherText` a `LongOutputStream` vznikly pouze proto, aby řešily tento problém.

5 Uživatelská dokumentace

5.1 Sestavení

Pro sestavení je vyžadován JDK 8 a Gradle. Sestavení spustíme příkazem:

```
gradle build
```

5.2 Spuštění

Pro spuštění je vyžadován JRE 8. Program spustíme příkazem:

```
java -jar <program> <parametry>
```

Seznam parametrů

- `-in <soubor>` vstupní soubor,
- `-out <soubor>` výstupní soubor,
- `-e` nebo `-encrypt` provede šifrování vstupního souboru – výchozí,
- `-d` nebo `-decrypt` provede dešifrování výstupního souboru,
- `-k <číslo>` nebo `-key <klíč>` 64bitový klíč v hexadecimální soustavě,
- `-ecb` použití módu Electronic Codebook (ECB) – výchozí mód,
- `-pecb` nebo `-parallel-pecb` použití módu paralelní ECB,
- `-cbc` použití módu Cipher Block Chaining (CBC),
- `-iv <číslo>` inicializační vektor pro CBC v hexadecimální soustavě,

U přepínačů na velikosti písmen nezáleží. Pokud se budu parametry opakovat, dojde k jejich přepsání. Projekt obsahuje ukázkový dávkový soubor s názvem `example.bat`, který obsahuje několik příkazů pro inspiraci.

6 Závěr

Práce obsahuje implementaci šifry *Data Encryption Standard* (*DES*) s programem pro šifrování a dešifrování souborů.

Implementace šifry je přímočará a bez optimalizací, které by zastíraly podstatu věci, je proto vhodná spíše pro vzdělávací účely než běžné použití. Paralelizovaný mód ECB zašifruje/dešifruje soubor na čtyřjádrovém CPU Intel Core i5-4570 zhruba za 1/3 času oproti klasickému ECB (platí pro větší soubory).

Správnost je ověřena pomocí jednotkových testů s použitím knihovny JUnit 4.

Reference

[1] U.S. DEPARTMENT OF COMMERCE/National Institute of Standards and Technology. Data encryption standard. In *FIPS PUB 46-3, Federal Information Processing Standards Publication*, 1999.