

# **Deljenje skrivnosti**

## **Shamirjeva Metoda**

**Predmet: Izbrani Algoritmi**

**Izdelal: Stefan Srnjakov**

Deljenje Skrivnosti in Analiza Rekonstrukcije .....	3
1. Uvod .....	3
2. Implementacija.....	3
3. Rezultati in Analiza .....	3
3.1 Natančnost rekonstrukcije .....	3
Robustna metoda: .....	4
Običajna metoda:.....	4
3.2 Časovna zahtevnost.....	5
BigInt metoda: .....	5
Deljenje po bajtih:.....	6
4. Dodatno .....	7
5. Zaključek .....	7

# Deljenje Skrivnosti in Analiza Rekonstrukcije

## 1. Uvod

V tem poročilu predstavimo implementacijo aplikacije za deljenje skrivnosti na podlagi Shamirjeve metode. Cilj je omogočiti razbitje skrivnosti na deleže in rekonstrukcijo originalne skrivnosti na osnovi kombinacij deležev. V nalogi smo razvili robustno rešitev z uporabo knjižnice BigInt in omogočili testiranje delovanja algoritma pri različnih nastavitvah ( $n$ ,  $k$ ) ter različnih velikostih datotek.

## 2. Implementacija

Implementacija je bila razdeljena na dva dela:

### 1. Deljenje skrivnosti:

- Skrivnost razbijemo na  $n$  deležev, pri čemer za rekonstrukcijo zadostuje vsaj  $k$  deležev.
- Algoritem generira deleže z uporabo polinoma stopnje  $k-1$ , kjer je prosti člen skrivnost.

### 2. Rekonstrukcija skrivnosti:

- Za rekonstrukcijo skrivnosti uporabimo metodo Lagrangeve interpolacije.
- Implementirana je bila robustna različica algoritma, ki deluje zgolj s celimi števili (BigInt).

Aplikacija omogoča:

- Izbiro metode deljenja (BigInt ali deljenje po bajtih).
- Rekonstrukcija skrivnosti
- Grafično analizo časovne zahtevnosti in natančnosti rekonstrukcije.

## 3. Rezultati in Analiza

### 3.1 Natančnost rekonstrukcije

Na spodnjih grafih so prikazane natančnosti rekonstrukcije za robusten algoritem (BigInt metoda) in običajen algoritem (BigInt metoda).

### Robustna metoda:

- **Graf:** Prikazuje uspešnost rekonstrukcije 1KB datotek za različne pare  $(n, k)$  pri 100 ponovitvah.
- **Rezultati:**
  - Natančnost rekonstrukcije je bila konstantno blizu 100 % pri vseh testnih primerih.
  - Robustna metoda se je izkazala za izjemno zanesljivo, tudi pri večjih vrednostih  $n$  in  $k$ .

## Secret Sharing App

Split files into secure shares, reconstruct them, and analyze performance.

SHARING SECRETS

RECONSTRUCTION

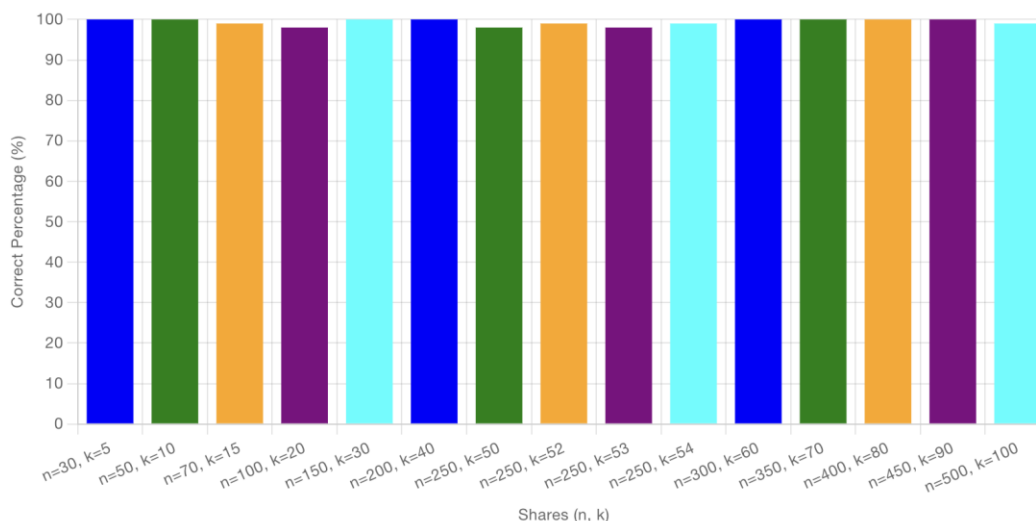
ANALYSIS

ANALYSIS RECONSTRUCTION

### Analyze Precision of Robust Reconstruction

START ANALYSIS

Analysis of reconstruction of 1kb file, for each pair of shares  $(n, k)$ . repeated 100 times.



### Običajna metoda:

- **Graf:** Prikazuje rezultate za natančnost rekonstrukcije brez robustnega ravnanja.
- **Rezultati:**
  - Pri večjih vrednostih  $n$  in  $k$  so bile opazne napake zaradi zaokroževanja in omejitev pri decimalnih številih.
  - Metoda je bila uspešna pri manjših vrednostih  $k$

# Secret Sharing App

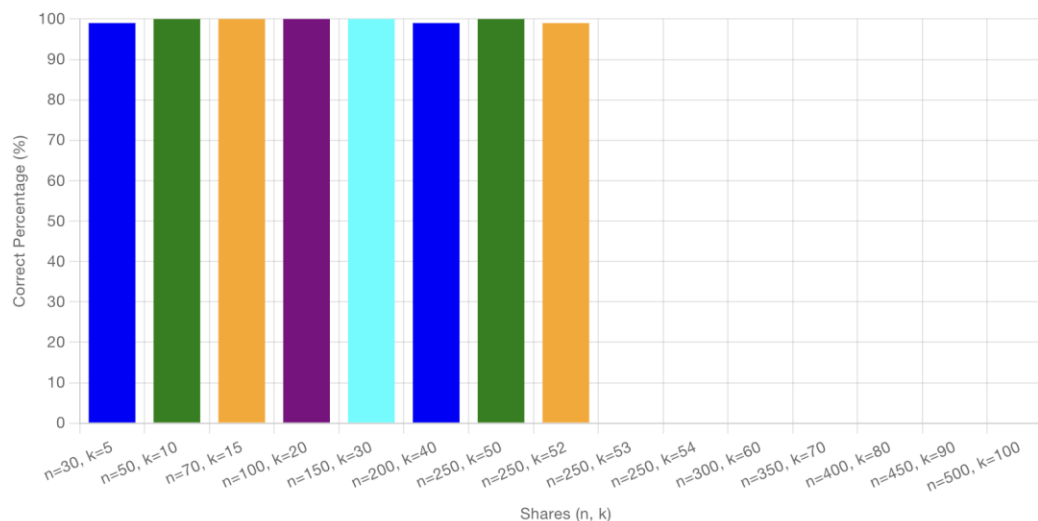
Split files into secure shares, reconstruct them, and analyze performance.

SHARING SECRETS   RECONSTRUCTION   ANALYSIS   **ANALYSIS RECONSTRUCTION**

## Analyze Precision of not precise reconstruction

START ANALYSIS

Analysis of reconstruction of 1kb file, for each pair of shares (n, k). repeated 100 times.



## 3.2 Časovna zahtevnost

Na spodnjih grafih je prikazana časovna zahtevnost deljenja in rekonstrukcije skrivnosti glede na velikost datotek in uporabljeno metodo.

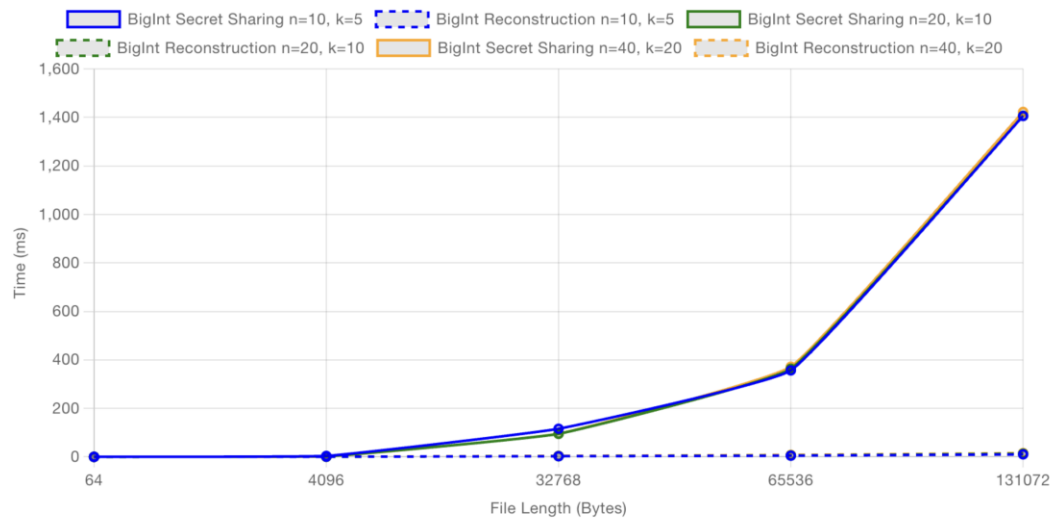
**BigInt metoda:**

- **Graf:** Časovna zahtevnost za deljenje in rekonstrukcijo z BigInt.
- **Rezultati:**
  - Časovna zahtevnost narašča linearno z velikostjo datotek.
  - N in k parametri nimajo veliki vpliv

## Analyze Performance

START ANALYSIS

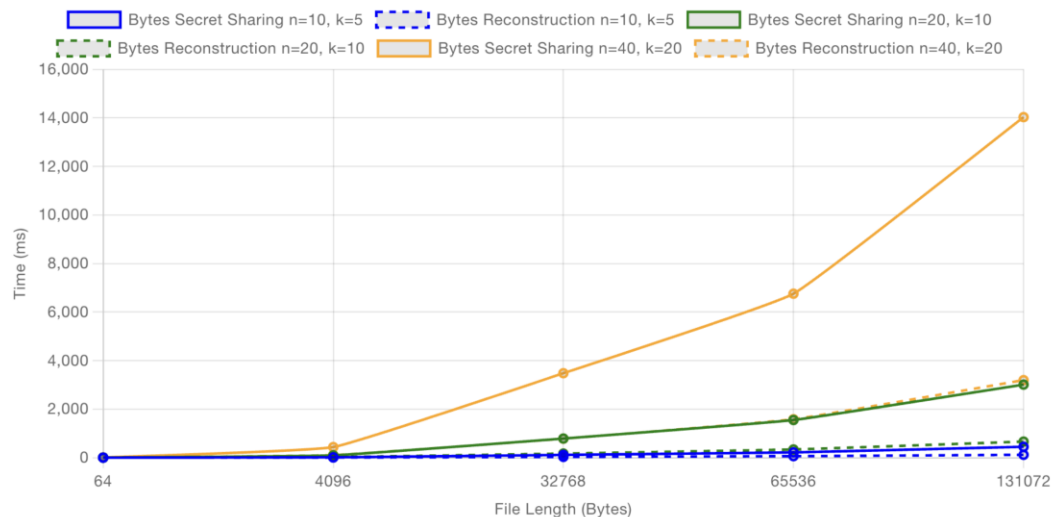
### BigInt Method Analysis



### Deljenje po bajtih:

- **Graf:** Časovna zahtevnost za deljenje in rekonstrukcijo po bajtih.
- **Rezultati:**
  - Deljenje in rekonstrukcija sta bila počasnejša v primerjavi z BigInt metodo. Najbolj velika razlika se opazi pri narascanje n in k.

### Byte Method Analysis



## 4. Dodatno

Posnetki zaslona:

### Secret Sharing App

Split files into secure shares, reconstruct them, and analyze performance.

SHARING SECRETS

RECONSTRUCTION

ANALYSIS

ANALYSIS RECONSTRUCTION

#### Share Secrets

Total Shares (n)  shares

Minimum Shares (k)  shares

[UPLOAD FILE](#)

Prefix for Share Files

Secret Sharing Method  
☒ Secret Sharing with BigIntegers ☐ Secret Sharing By Bytes

[GENERATE SHARES](#)

### Secret Sharing App

Split files into secure shares, reconstruct them, and analyze performance.

SHARING SECRETS

RECONSTRUCTION

ANALYSIS

ANALYSIS RECONSTRUCTION

#### Reconstruct Secrets

k (threshold)

Upload share files:  
[UPLOAD SHARE FILES](#)

☐ Reconstruct By Bytes

Output File Name

[SUBMIT](#)



## 5. Zaključek

- **Robustna metoda rekonstrukcije** je bistveno bolj zanesljiva in natančna ter popolnoma odpravlja napake, ki jih povzroča zaokroževanje v decimalnih operacijah.
- **Navadna metoda rekonstrukcije** je primerna zgolj za manjše vrednosti  $n$  in  $k$ , saj napake zaradi zaokroževanja postanejo preveč izrazite pri velikih vrednostih.

- **BigInt metoda** je hitrejša in natančnejša za deljenje in rekonstrukcijo skrivnosti, še posebej v primerjavi z metodo bajt za bajtom.
- **Metoda bajt za bajtom** je počasnejša in manj primerna za obdelavo velikih datotek ali visokih vrednosti  $n$  in  $k$ .