Shamir's secret sharing

Milica Todorović, RA17-2015

Problem

- Slike se koriste za prenos i čuvanje različitih tipova informacija.

Postoje različiti scenariji kada vlasnici slika žele da ih zaštite od zlobupotrebe i redistribucije.

- Primeri:
- Medicinski snimci
- Stalitski snimci
- Skice i nacrti u automobilskoj industriji, građevini, itd.
- Formule u farmaceutskoj industriji
- Jedno rešenje:

Sačuvati sliku tako da je nečitljiva dok ne dođe do destinacije. Na destinaciji je moguće izvršiti rekonstrukciju i pročitati sliku.

- Sumirano:

Slika se deli na n delova od kojih je k potrebno da bi se rekonstruisala.

Osnovna ideja

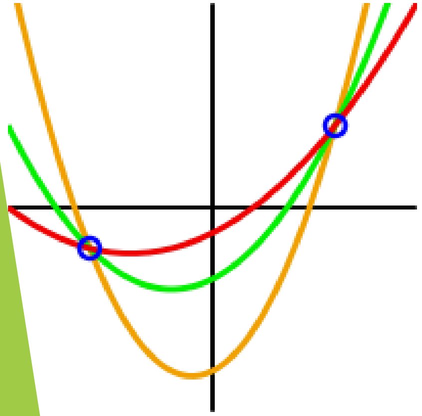
Ideja:

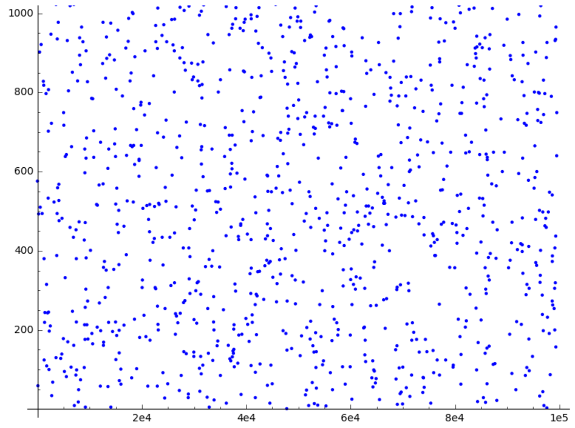
- Cilj je podeliti tajnu S na n delova takvih da:

1. Tajna S može se rekonstruisati ako je poznato k ili više delova.
2. Ako je poznato manje od k delova nije moguće rekonstruisati tajnu S.

- Ideja:

Polinom stepena k-1 određen je sa k tačaka.





Primer:

- Tajni broj je 658, k = 3, n = 6.
- Za a_0 se uzima 658. Preostala dva koeficijenta se slučajno generišu: a_1 = 166, a_2 = 17.
- Polinom ima oblik:

$$f(x) = 658 + 166x + 17x^2$$

- Konstruiše se 6 tačaka od kojih se svaka dodeljuje jednom učesniku:

D_0(33,24649) D_1(56,63266)

D_2(78,117034) D_3(107,213053)

D_4(199,706909) D_5(204,741994)

- Interpolacijom bilo koje 3 (ili više) datih tačaka dobija se početni polinom i otkriva tajna.
- Interpolacija manje od 3 tačke daje polinom sa drugim koeficijentima.

Zašto konačna polja?

- Primer: Pretpostavimo da se koeficijenti polinoma pripadaju N.

Osobi su poznate dve tačke $D_0(1,1494)$ i $D_1(2,1942)$, kao i **javne informacije** k=3, n=6.

- Polinom tada ima oblik: $f(x) = S + a_1x + a_2x^2$
- Ako se uvrste poznate informacije:

$$1494 = S + a_1 1 + a_2 1^2 \Rightarrow 1494 = S + a_1 + a_2$$
$$1942 = S + a_1 2 + a_2 2^2 \Rightarrow 1942 = S + 2a_1 + 4a_2$$

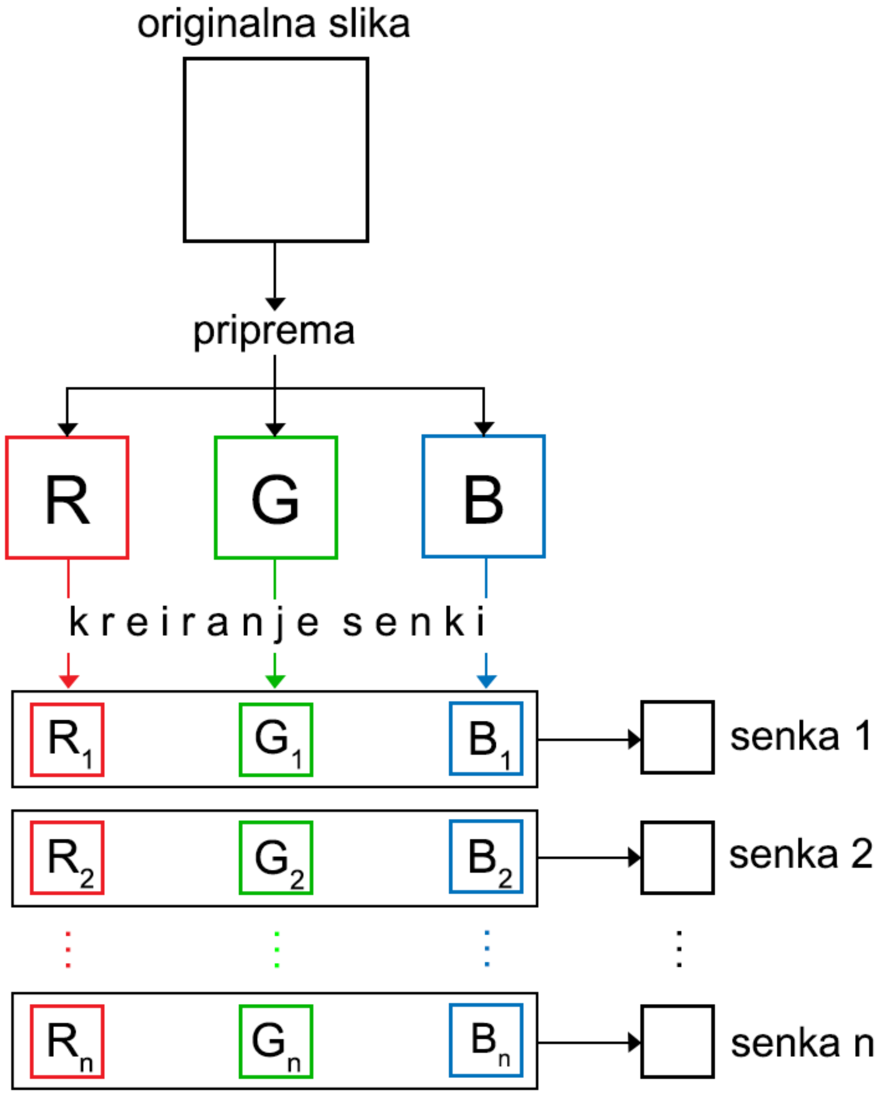
- Oduzimanjem se dobije: $a_1 = 448 - 3a_2$
- Uvrštavanje u početni polinom: $1494 = S + (448 - 3a_2) + a_2 \Rightarrow S = 1046 + 2a_2$
- Da bi se pogodila tajna S, potrebno je izračunati a_2. Osoba može da pogađa a_2.

$$a_2 = 0 \rightarrow a_1 = 448 - 3 \times 0 = 448$$
$$a_2 = 1 \rightarrow a_1 = 448 - 3 \times 1 = 445$$
$$a_2 = 2 \rightarrow a_1 = 448 - 3 \times 2 = 442$$
$$\ldots$$
$$a_2 = 148 \rightarrow a_1 = 448 - 3 \times 148 = 4$$
$$a_2 = 149 \rightarrow a_1 = 448 - 3 \times 149 = 1$$

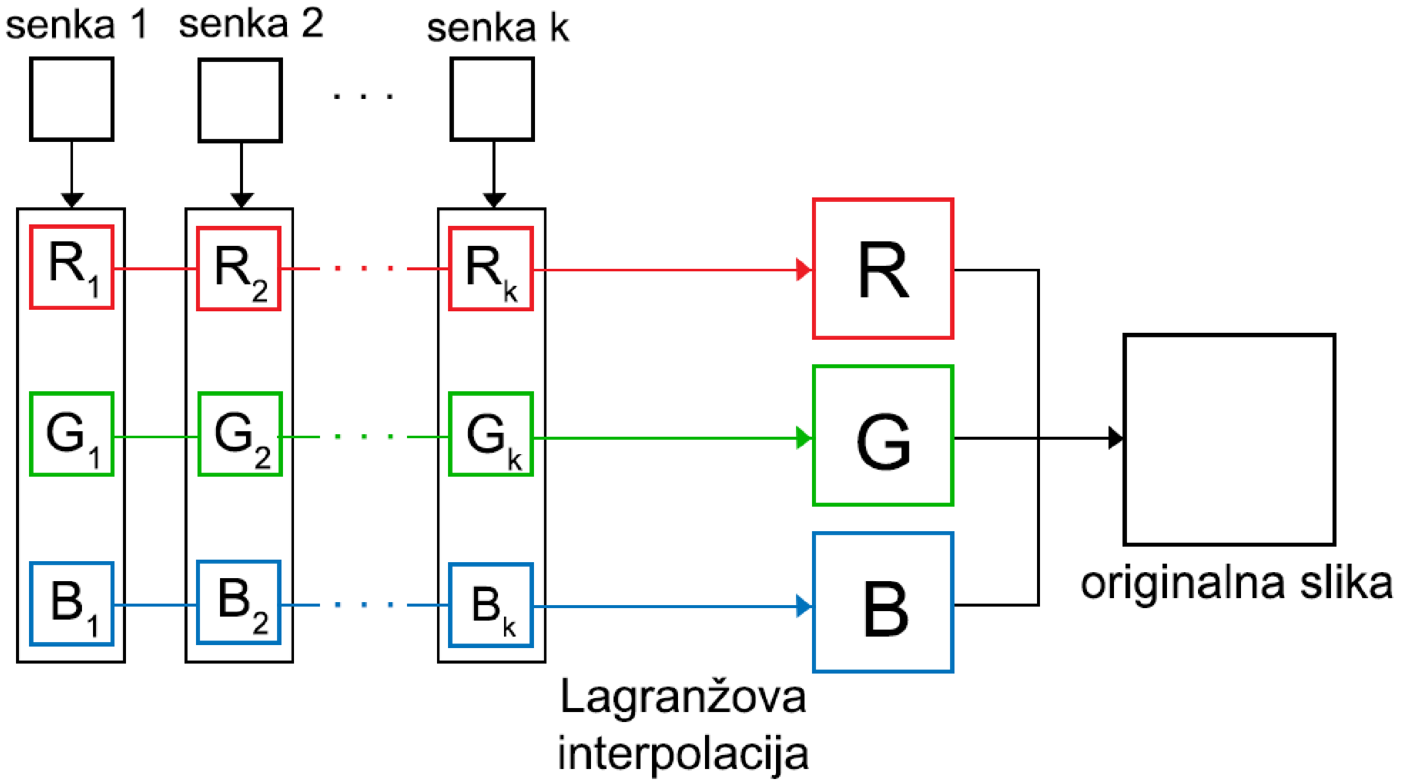
- Dakle, $a_2 \in [0, 1, \ldots, 148, 149]$, a $S \in [1046, 1048, \ldots, 1342, 1344]$.
- To je 150 mogućnosti, umesto beskonačno.
- Konačna polja otklanjaju ovaj nedostatak.

Algoritam primenjen na slike:

Generisanje senki (enkripcija):



Rekonstrukcija početne slike (dekripcija):



Izbor k i n:

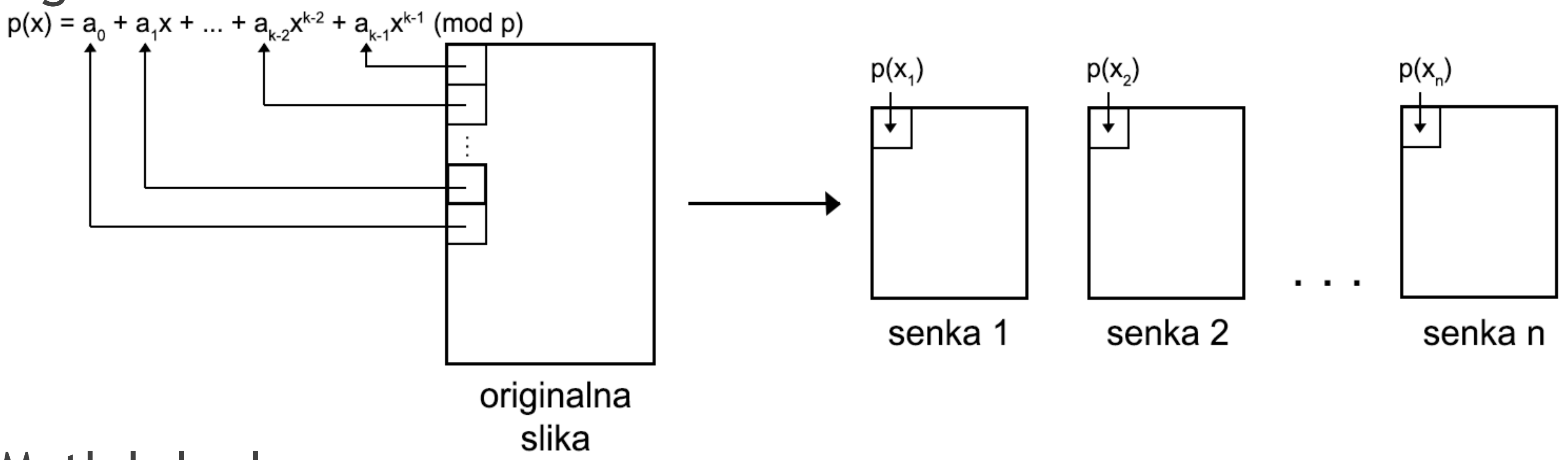
- k mora biti izabrano tako da proizvod visine i širine slike bude deljiv sa k.
- **$1 < k \leq n$**

Pretprocesiranje slike:

- Potrebno je izabrati prost broj p, koji određuje konačno polje u kome se vrši računanje.
- Prost broj p se bira tako da je veći od svih koeficijenata u polinomu.
- Vrednosti dodeljene pikselima u slici su iz opsega [0,255]. Date vrednosti piksela će predstavljati koeficijente u polinomima.
- Izabrano je p = 251, jer je 251 najveći prost broj manji od 255.
- Pre dalje primene algoritma **potrebno je sve vrednosti piksela svesti tako da budu manje od 251**.
- **Ovaj korak rezultuje gubicima informacija iz originalne slike.**

Generisanje senki (enkripcija):

Algoritam:



Matlab kod:

```
%racunanje senki
for i = 1:brojPikselaSenke %racuna se piksel i iz svake senke
    %uzima se k piksela iz originalne slike i oni predstavljaju koeficijente polinoma
    pocetak = (i-1)*k + 1;
    kraj = pocetak + k - 1;
    polinom = A(pocetak:kraj);

    %iskoristi se generisani broj x da bi se dobila vrednost tog piksela u senci
    shadows( i:brojPikselaSenke:i+(n-1)*brojPikselaSenke) = mod(polyval(polinom,x),p);
end
```

Zašto ne posmatrati svaki piksel kao zasebnu tajnu?

Memorijska neefikasnost

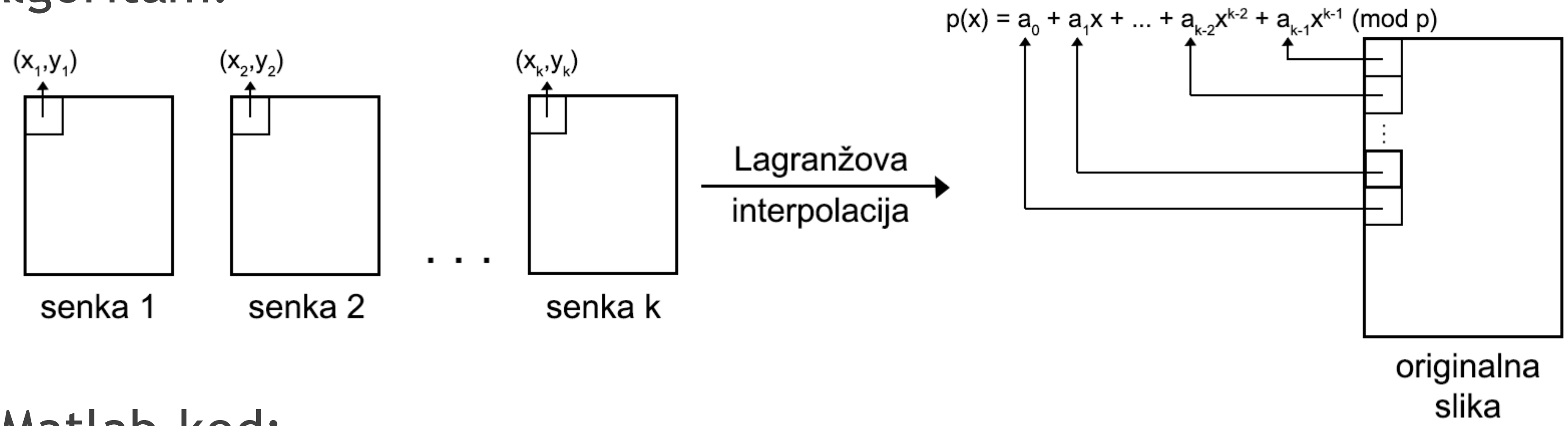
- Ukoliko bi se svaki piksel posmatrao kao zasebna tajna, svaka generisana senka imala bi jednako piksela kao originalna i zauzimala bi jednako memorijskog prostora.
- U toku rekonstrukcije originalne slike, bilo bi potrebno koristiti k*veličina početne slike memorijskog prostora.
- Uzimanjem grupa od po k piksela dobijaju se senke veličine 1/k početne slike.
- Na taj način se u toku rekonstrukcije koristi onoliko memorijskog prostora koliko zauzima originalna slika.

Računska neefikasnost

- U slučaju da se svaki zasebni piksel posmatra kao tajna, bilo bi potrebno izvršiti Lagranževu interpolaciju onoliko puta koliko originalna slika ima piksela.
- Uzimanjem grupa od po k piksela, Lagranževa interpolacija se izvršava broj piksela originale slike/k puta.

Rekonstrukcija slike (dekripcija)

Algoritam:



Matlab kod:

```
for i = 1:visina*sirina
    %uzima se piksel sa indeksom i iz svake senke
    y = shadows(i:brojPikselaSenke:i + (brojSenki-1)*brojPikselaSenke);
    %Lagranzevom interpolacijom nalaze se koeficijenti polinoma
    koeficijenti = lagrangeInterpolation(x,y,p);
    %upisuju se u rekonstrukciju slike
    pocetak = (i-1)*brojSenki + 1;
    kraj = pocetak + brojSenki - 1;
    original(pocetak:kraj) = koeficijenti;
end
```

Lagranževa interpolacija:

Matlab kod:

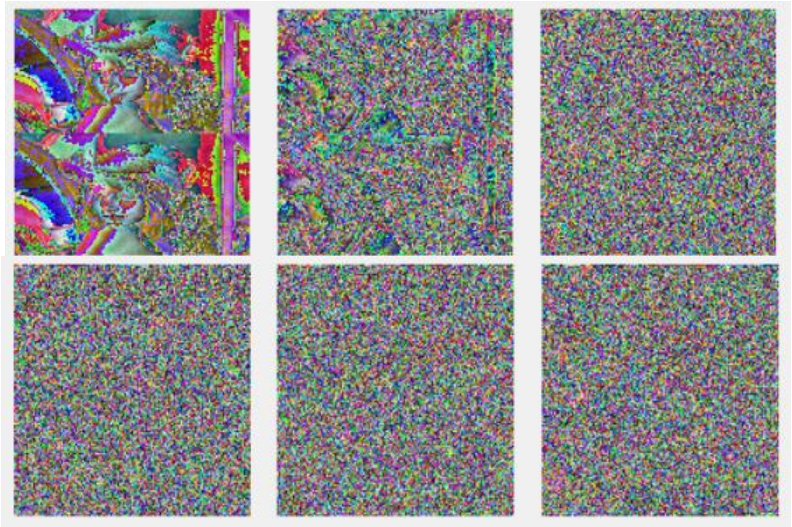
```
n = length(x);
polinom = zeros(1, n); %inicijalizacija povratne vrednosti

for i = 1:n
    brojilac = 1;
    imenilac = 1;
    for j = 1:n
        if j~=i
            brojilac = gfconv( brojilac, [1, mod(-x(j),p)], p);
            imenilac = imenilac *  mod( x(i) - x(j), p);
        end
    end
    imenilac = mod(imenilac,p);
    L = gfdeconv(brojilac,imenilac,p);
    proizvod = gfconv(L,y(i),p);
    polinom = polinom + proizvod;
end
    polinom = mod(polinom,p);
end
```

$$l_k(x) = \prod_{\substack{i=1 \\ i \neq k}}^{t} \frac{x - x_i}{x_k - x_i} \pmod{p}$$

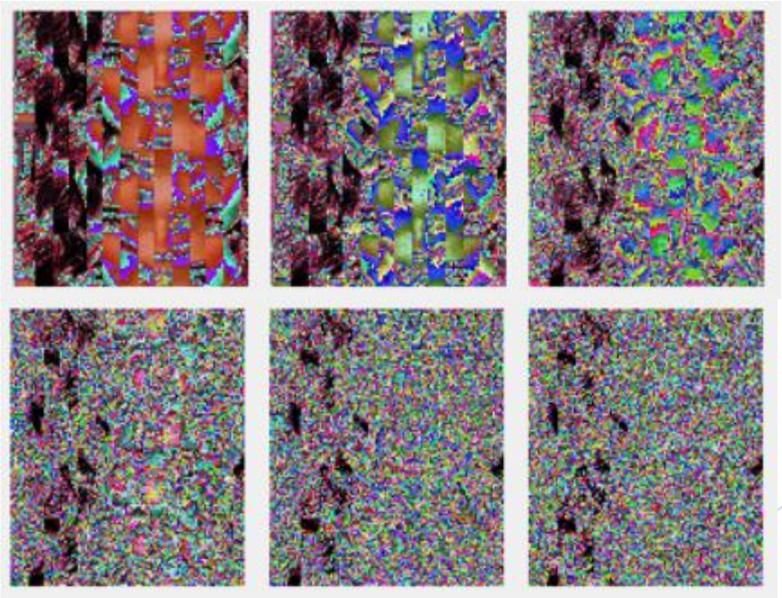
$$p(x) = \sum_{k=1}^{t} y_k l_k(x)$$

Primeri upotrebe:

- Veličina slike: 512x512, k = 4, proteklo vreme = 3s



- Veličina slike: 610x500, k = 3, proteklo vreme = 4s



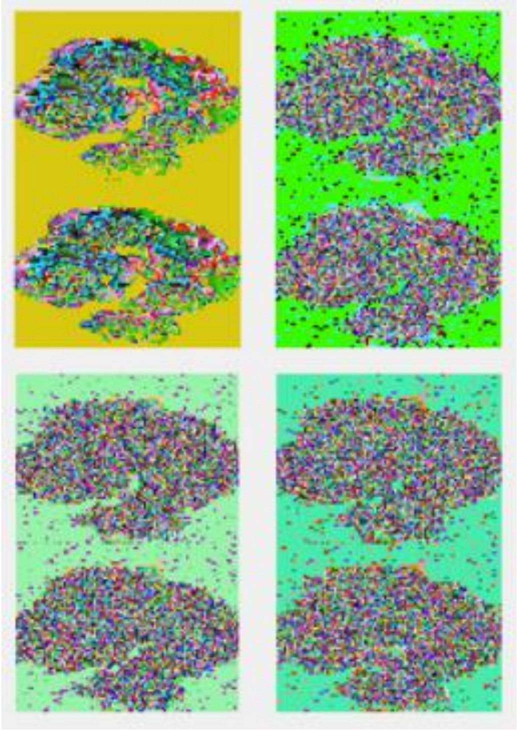
Problem susedinih piksela:

Homogene regije:

- Ukoliko originalna slika poseduje homogene regije, dolazi do pojave da senke liče na original.

Primer:

Veličina slike 564x376, k = 4, proteklo vreme = 2s

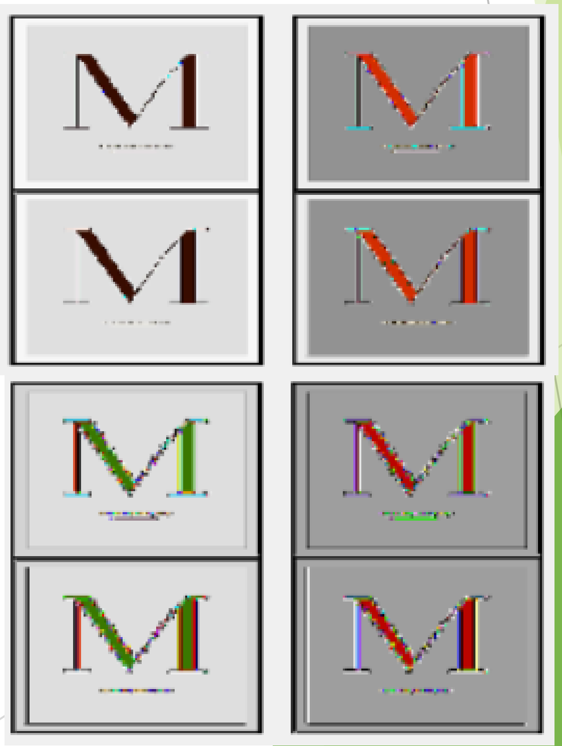


Sličnost susednih piksela:

- Zbog sličnosti susednih piksela, ako se kao koeficijenti koriste uzastopni pikseli, rekonstrukciju je moguće izvršiti i sa manje od k senki.

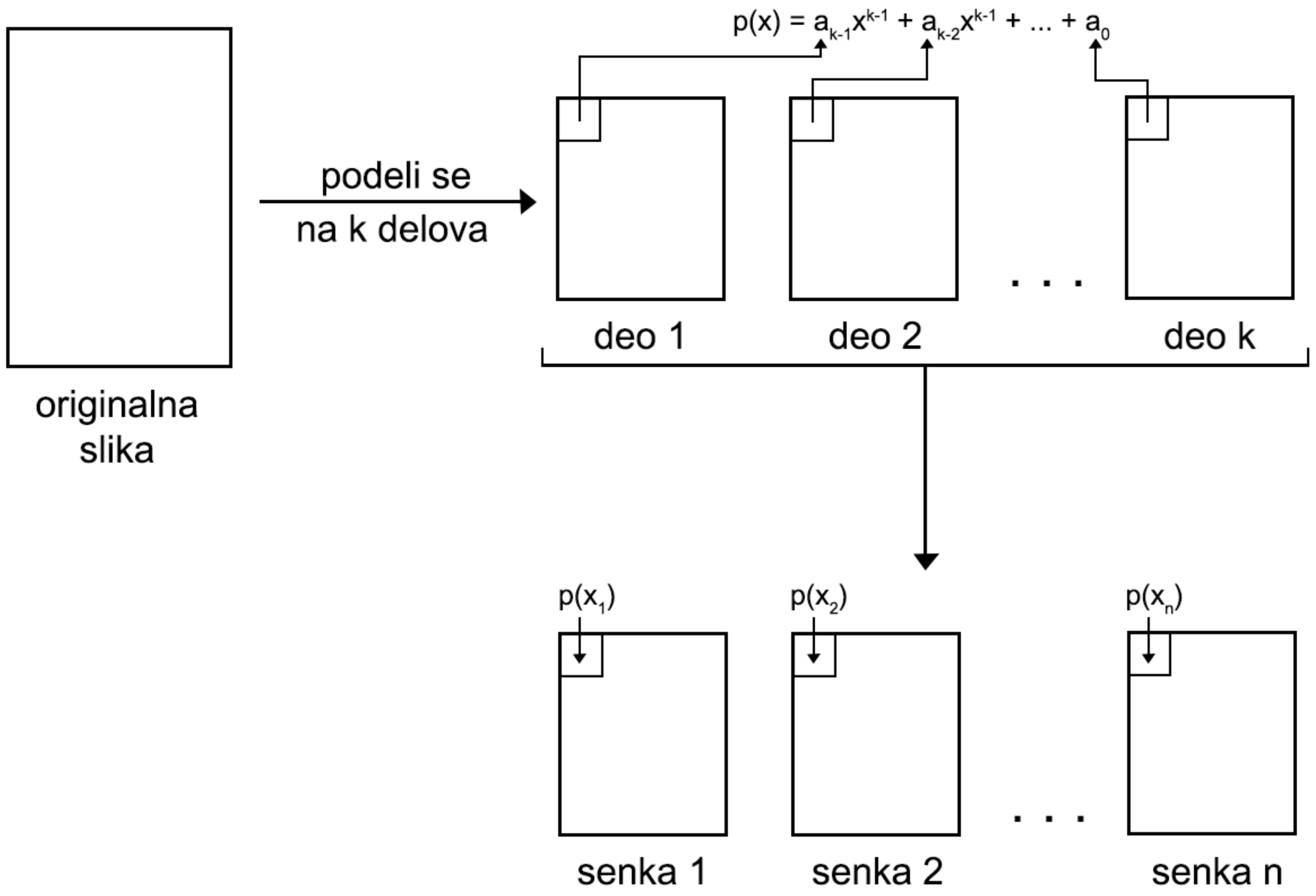
Primer:

Veličina slike 750x540, k = 4, proteklo vreme = 4s



Jedno rešenje:

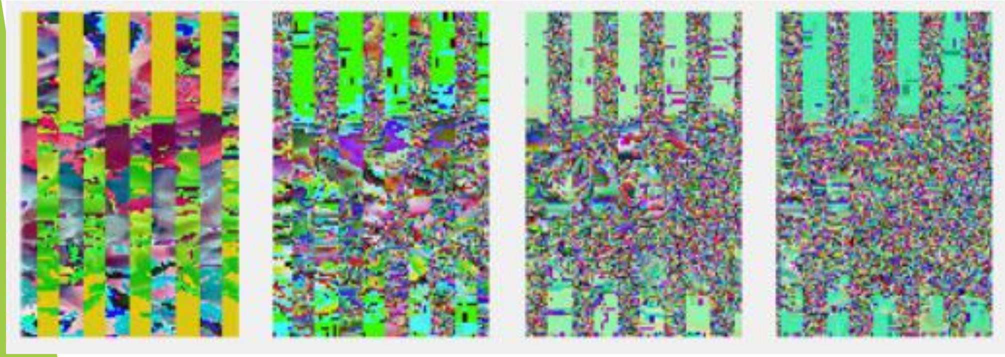
- Umesto da se koriste susedni pikseli kao koeficijenti polinoma, slika se prvo podeli na k delova.



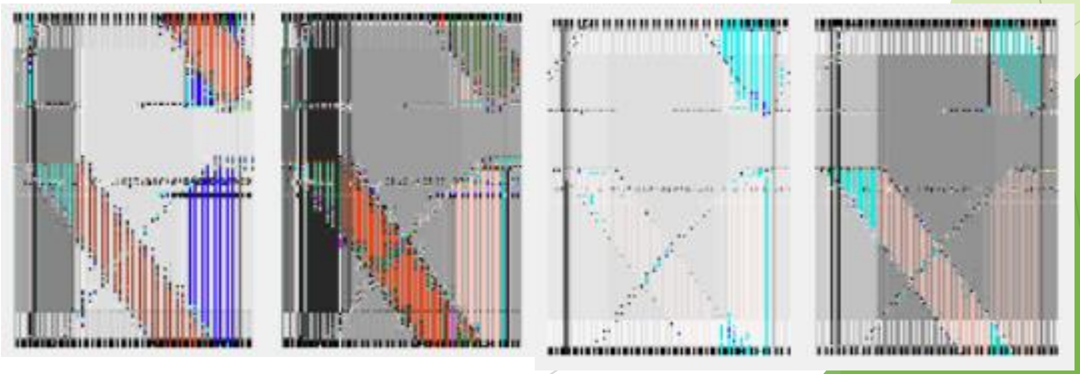
- Ovako korišteni pikseli nemaju jasnu međusobnu vezu.

Rezultati primene:

Veličina slike 564x376, k = 4, proteklo vreme = 2s



Veličina slike 750x540, k = 4, proteklo vreme = 4s



Zaključak:

- Gubici: **navedeni algoritmi nisu bez gubitaka**, zbog izabranog p
- Verovatnoća rekonstrukcije bez poznavanje k delova je 251^{xy} , gde su x i y dimenzije slike, pa je praktično **nemoguće izvršiti rekonstrukciju bez k delova**.
- Vreme izvršavanje algoritma zavisi od dimenzija slike, k i n.

Korištena literatura:

- Alharthi, Saeed & K. Atrey, Pradeep. (2010). An improved scheme for secret image sharing. 1661-1666. 10.1109/ICME.2010.5583180. Link
- Lukac R., Plataniotis K.N., Venetsanopoulos A.N. (2004) A {*k*, *n*}-Secret Sharing Scheme for Color Images. In: Bubak M., van Albada G.D., Sloot P.M.A., Dongarra J. (eds) Computational Science - ICCS 2004. ICCS 2004. Lecture Notes in Computer Science, vol 3039. Springer, Berlin, Heidelberg Link
- Wikipedia Link