

VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
MATEMATINĖS INFORMATIKOS KATEDRA

Rasa Ivanauskaitė

Informatikos studijų programa
Matematinės informatikos šaka

Vaizdo įrašų vizualioji kriptografija

Bakalauro baigiamasis darbas

Vadovas: dr. **Vilius Stakėnas**

Vilnius 2016

Turinys

Įvadas	3
1 Vizualioji kriptografija	5
1.1 Vizualiosios kriptografijos pradmenys	5
1.2 Tolimesnė vizualiosios kriptografijos raida	10
1.2.1 Vizualioji kriptografija spalvotiems vaizdams	11
1.2.2 Dinaminė vizualioji kriptografija	14
2 Vizualiosios kriptografijos pritaikymas	17
2.1 Vizualiosios kriptografijos pritaikymo pavyzdžiai	17
2.2 Vaizdo įrašų ir vizualiosios kriptografijos suderinimas	18
2.2.1 Vaizdo įrašų ypatybės	19
2.2.2 Vizualiosios kriptografijos pritaikymas vaizdo įrašams	20
3 Sukurtos programos analizė ir pavyzdžiai	21
3.1 Programos specifiška	21
3.2 Programos veikimo principas	22
3.3 Algoritmai programoje	25
Literatūra	29
Santrauka	30
Summary	31

Įvadas

Kriptografijos arba kitaip slaptojo kodo metodai sudaro esminę grandį informacijos saugumui užtikrinti. Žodynuose kriptografija įvardijama su tokiais sąvokomis kaip slaptaraštis ar duomenų pateikimo būdas, kuriuo įmanoma įslaptinti duomenų turinį. Kriptografija paslapčių apsaugojimui naudojama jau tūkstantmečius ir turi pakankamai ilgą istoriją [3]. Pirmasis žinomas kriptografijos panaudojamas siekia maždaug 1500 metus prieš Kristų Egipte. Taip pat dar žinomas hebrajų ATBASH šifras iš 500 - 600 metų prieš Kristų arba graikų skytalė kriptografijos metodas iš 486 metų prieš Kristų, kuris buvo naudojamas dar Spartoje. Plačiau aprašytus šiuos ir dar daugiau senųjų kriptografijos metodų galima rasti [1]. Pavyzdžiui, skytalės kaip pirmojo istorijoje žinomo kriptografijos įrenginio metodas aprašomas [1]. Skytalė - graikiškos kilmės žodis turintis kelias reikšmes - lazda, lazdelė ar siaura odinė juostelė su užrašytu tekstu taip, kuris užrašytas uždėjus šią juostelę ant lazdelės taip, kad juostelės sluoksniai nepersidengtų, o būtų šalia vienas kito, galėtume perskaityti tekstą. Būtent paskutinė reikšmė ir atskleidžia šio kriptografijos metodo specifiką. Esant tokiai situacijai, kai paslapties siuntėjas ir gavėjas turi tokią pat lazdelę, siuntėjas taisyklingai užvynioja odinę juostelę ant lazdelės ir užrašo paslaptį eilutėmis, kas kartą vis pasukant lazdelę. Pabaigus rašyti paslaptį juostelė nuimama nuo lazdelės siekiant, kad niekas nesužinotų užrašytos paslapties, išskyrus paslapties gavėją. Paslapties gavėjas išsiaiškina paslaptį uždėjęs odinę juostelę ant savo turimos lazdelės.

Pagrindinis pirmųjų klasikinės kriptografijos metodų duomenų užšifravimo tikslas yra tas, kad dalis žmonių nori dalintis savo informacija taip, kad niekas kitas apie tai nesužinotų. Tarkim Jonas turi paslaptį ir nori ją atskleisti tik Petrui taip, kad Onutė nieko nesužinotų. Taigi Jonas užšifruos savo žinutę pasinaudodamas algoritmu, kuris transformuos jo žinutę, tarkime, kad tai šiuo atveju bus tekstas, į užšifruotą žinutę, kuri tegu šiuo atveju irgi bus tekstas. Užšifruota žinutė atrodys kaip kliedėjimas, neaiškus raidžių išsidėstymas, kuris Onutei bus visiškai neaiškus. Taigi, jeigu Onutė ir gautų perduodamą žinutę nuo Jono Petrui, jai būtų visiškai neaiški pirminė žinia. Petrui tai taip pat galėtų atrodyti kaip kliedesys, bet Petras žino algoritmą, kuriuo gali iššifruoti Jono žinutę ir perskaityti pradinę originalią žinutę. Šiuo atveju iškyla tik viena problema - reikia, kad algoritmas būtų pakankamai sudėtingas ir kad Onutei jis nebūtų lengvai suprantamas ir išsiaiškinamas. Gali būti, kad Onutė jau yra susidūrusi su Jono ir Petro naudojamais algoritmais, tai ji ne tik gali lengvai išsiaiškinti jų pradinę žinutę, bet taip pat ir apgauti juos sukurdamą klaidingai užšifruotą žinutę.

Keičiantis amžiams keitėsi ir kriptografija ir štai 1466 metais italas Leon Battista Alberti pradėjo naudoti kriptografijos šifrus su keičiamais raktais. Keičiamų raktų panaudojimas naudingas tuo, kad tą patį algoritmą gali žinoti visi ir juo naudotis gali visi, tačiau kiekvienas turi atskirą raktą su kuriuo gali šifruoti žinutes. Jei Jonas su Petru nori dalintis paslaptimi, jie turi jau ne tik algoritmą, bet dar ir raktą, kuris yra kaip papildoma apsauga. Taigi, Onutė žinodama tik algoritmą, o rakto ne, liks vėl žinutę supratęs tik kaip neaiškų kliedėjimą. kriptografas Bruce Schneier [4] kriptografijos raktą paaikšina pasinaudojęs durų

spynos modeliui. Tarkim, kad yra bendras spynų gamintojas, kuris visiems Vilniaus gyventojams į duris įmontuoja tą pačią spyną, skiriasi tik jos raktas. Jonas turėdamas raktą nuo savo durų galės atsirakinti duris ir užėti namo, o jei savo raktą perduos draugui Petriui, tai ir draugas galės atsirakinti duris ir užėti į Jono namus. Onutė neturėdama rakto nuo Jono namų negalės atsirakinti Jono namų durų ir užėti į vidų. Net jei Onutė gautų raktą, Jonas galėtų pasikeisti spynos raktą į kitą ir Onutė vėl negalėtų atsirakinti Jono namų durų. Lygiai taip pat viskas vyksta ir kriptografijoje algoritams su raktais. Būtent taip kriptografija veikia ir iki šių dienų, skirtumas tik tas, kad algoritmai vis sudėtingėja. Sudėtingi algoritmai ne tik padidina tikimybę, kad užšifruota žinutė nebus iššifruota, bet taipogi ir apsunkina žinutės užšifravimą.

Kriptografijos atšaka vizualioji kriptografija, kurią toliau nagrinėsime, nereikalauja sudėtingų algoritmų. Šios šakos schemas yra kur kas paprasčiau suprasti ir perteikti žmonėms, kurie neturi daug žinių apie kriptografiją. Remiantis [5] vizualiosios kriptografijos metodai leidžia saugiai dalintis saugumo reikalaujančia vizualine medžiaga viešuosiuose kanaluose, kadangi metodų sugeneruoti vaizdai neatskleidžia jokios informacijos apie originalų vaizdą iki kol vaizdai nebus atitinkamai sukombinuoti. Šiuo metodu sugeneruotiems vaizdams skirtiems saugiam dalijimuisi negresia net ir sudėtingiausios algoritminės atakos skirtos informacijos vagystei įgyvendinti. Nepaisant to, kad vizualiosios kriptografijos metodų naudojami būdai duomenims užšifruoti skiriasi nuo klasikinės kriptografijos metodų, pagrindinė esmė lieka ta pati – įslaptinti perduodamus duomenis.

1 Vizualioji kriptografija

Kiekvienais metais nuo 1982 metų pasaulyje vyksta tarptautinė kriptografijos tyrinėjimų konferencija „EUROCRYPT“. Būtent šioje „EUROCRYPT’94“ konferencijoje 1994 metais mokslininkai Moni Naor ir Adi Shamir pristatė savo mokslinį tyrimą [5] apie vizualiąją kriptografiją. Jų pateikiamo pranešimo įvade yra išskirta, kad jie pristato naują kriptografijos schemą, kuri iššifruodama paveikslėlius nereikalauja jokių skaičiavimų yra paprasta ją pateikti ir yra visiškai saugi. Vien toks pareiškimas sudomino daugelį kriptografų, kadangi taip atsivėrė nauja vizualiosios kriptografijos sfera.

Vizualioji kriptografija naudinga saugiai vaizdinei informacijai perduoti. Šios technikos naudojimas yra plačiai paplitęs skaitmeninėje erdvėje, be to, ši kriptografijos technika gali būtų pritaikoma ir realiems vaizdams esantiems popieriniame ar kitokiame apčiuopiamame pavidale. Vizualioji kriptografija – tai kitokių kriptografijos schemų erdvė, kurias galima iššifruoti naudojantis paprasčiausiai regos sistemos gebėjimais, iššifravimui nereikia visiškai jokių kriptografijos žinių. Schema yra saugi ir labai paprasta ją pritaikyti. Vizualiosios kriptografijos schemą galima būtų paaiškinti, taip lyg mes turime k iš n paslapčių dalinimosi problemą. Problema grindžiama, kad paslapties turėtojas perduoda skaidres kiekvienam n dalyvių, kur kiekvienas k iš jų sudėję savo skaidres vieną ant kitos gali pamatyti paslaptį. Išskirtiniu atveju, kai $k = 1$ dalyviai sudės savo turimas skaidres vieną ant kitos, negaus nė menkiausios informacijos apie paslaptį.

1.1 Vizualiosios kriptografijos pradmenys

Vizualiosios kriptografijos technika, kaip jau prieš tai minėta, pristatė – Moni Naor ir Adi Shamir. Šie mokslininkai pateikė šį būdą tinkamą šifruoti spausdintui tekstu, ranka rašomiems rašteliams, paveikslėliams ir panašioms vaizdinėms medžiagoms saugiai dalintis. Pačio paprasčiausio modelio reikalavimai ir kitos detalės:

1. Žinutė turi būti sudaryta tik iš juodų ir baltų pikselių.
2. Kiekvienas pikselis transformuojamas atskirai nuo visų kitų pikselių.
3. Kiekvienas pikselis iš originalaus paveikslėlio persikels į n modifikuotų versijų, po kiekvieną kiekvienai skaidrei.
4. Kiekviena modifikacija yra m juodų ir baltų vidinių pikselių kolekcija, kurie yra taip arti, kad žmogaus rega mato tik tarpinį variantą tarp juodos ir baltos spalvos.
5. Gauta rezultato struktūra yra $n \times m$ matrica $S = [s_{ij}]$.
6. Sudėjus skaidres $i_1, i_2, \dots, i_j$ ir kartu, sulygiuojant visas vidinių pikselių kolekcijas, turime gauti sukombinuotas visas paslapties dalis taip, kad paslapyje buvę juodi pikseliai ir išryškėję kaip juodi pikseliai, o buvę balti pikseliai tampa chaotišku fonu iš juodų ir baltų pikselių, kurie žmogaus akiai atrodo kaip pilkas fonas.

Aukščiau išvardintų reikalavimų reikia šifravimo [7] specifikai įgyvendinti:

1. Tarkime turime binarinį paveikslėlį, kuriame yra slaptoji žinutė ir norime išskaidyti žinutę į dvi paslapties skaidres. Paimame kiekvieną pikselį paeiliui nuo pirmojo ir tikriname:

- Jei turime baltą pikselį - renkamės vieną iš dviejų būdu užšifruoti baltą pikselį pirmajam ir antrajam išskaidytui paveiksliukui s_1 , s_2 .
 - (a) Padalijame pikselį į du klasterius, kur s_1 -ojo paveiksliuko pirmas klasteris bus juodas, antras klasteris- baltas o s_2 -ojo paveiksliuko lygiai toks pats, pirmasis - juodas, antrasis - baltas.
 - (b) Padalijame pikselį į du klasterius, kur s_1 -ojo paveiksliuko pirmas klasteris bus baltas, antras klasteris- juodas o s_2 -ojo paveiksliuko vėl lygiai toks pats, pirmasis - baltas, antrasis - juodas.
- Jei turime juodą pikselį - renkamės vieną iš dviejų būdu užšifruoti juodą pikselį pirmajam ir antrajam išskaidytui paveiksliukui s_1 , s_2 .
 - (a) Padalijame pikselį į du klasterius, kur s_1 -ojo paveiksliuko pirmas klasteris bus juodas, antras klasteris- baltas o s_2 -ojo paveiksliuko atvirkščiai, pirmasis - baltas, antrasis - juodas.
 - (b) Padalijame pikselį į du klasterius, kur s_1 -ojo paveiksliuko pirmas klasteris bus baltas, antras klasteris- juodas o s_2 -ojo paveiksliuko atvirkščiai, pirmasis - juodas, antrasis - baltas.

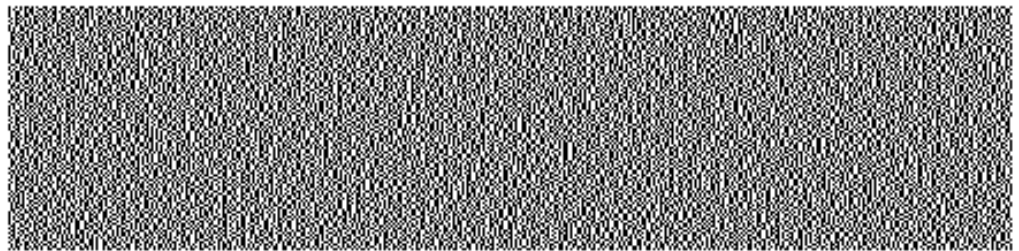
Po tokio darbo su kiekvienu pikseliu, galiausiai turėsime jau tris paveiksliukus - pirmąjį paslapties paveikslėlį, kurį jau turėjome ir du naujus viešus paveiksliukus S_1 ir S_2 skirtus dalinimuisi. Šio pikselio skaidymui į klasterius aiškesnės vizualizacijos tikslais žemiau pateikiu paveiksliuką.

		s1	s2	s1 xor s2
	a) būdas :			
	b) būdas :			
	a) būdas :			
	b) būdas :			

2. Turint viešuosius paveikslėlius s1 ir s2 galime juos saugiai perduoti gavėjams. Gavėjas gavęs du paveikslėlius galės juos iššifruoti pasinaudojęs xor operacija. Tekstas ar paveikslėlis, kuris prieš šifravimą buvo juodos spalvos ir liks juodos spalvos, o kur buvo baltas pikselis, matysime baltai juodą foną, kuris mūsų akiai atrodys, kaip pilkas.

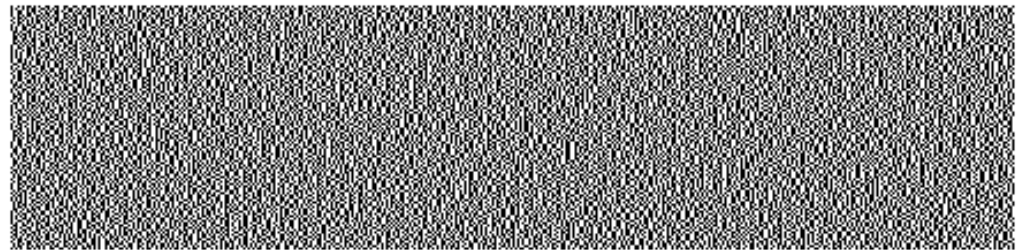
Žemiau pateikiamas iššifravimo pavyzdys, kur pradinė originali žinutė, baltame lape užrašytas žodis "kriptografija", buvo išskaidyta į dvi skaidres - pirmąją ir antrąją, o vėliau šios skaidrės buvo sudėtos tiksliai viena ant kitos taip iššifruojant ir gaunant rezultatą:

1 skaidrė



+

2 skaidrė



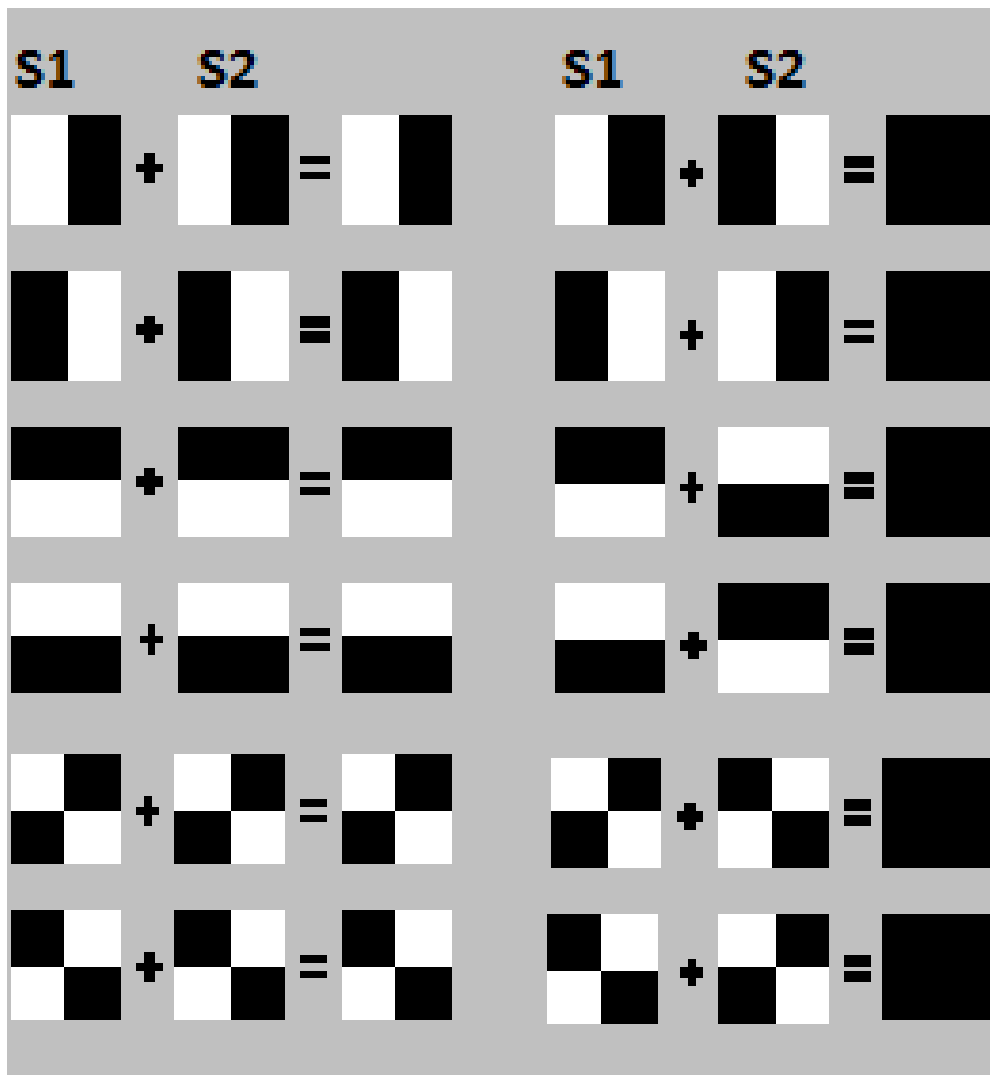
=

Rezultatas



Vienas svarbus skirtumas tarp šios schemos taikymo skaitmeniniam vaizdui ir rašytiniam vaizdui yra iššifruoto paveikslėlio dydis. Kadangi skaitmeniniame paveikslėlyje pikselį mes interpretuojame, kaip mažiausią ir nebeskaidomą paveikslėlio detalę, skaitmeninėje aplinkoje šiai schemai naudojamas pikselių išplėtimas, dėl šios priežasties iššifruoto paveikslėlio dydis skiriasi nuo pradinio originalaus paveikslėlio dydžio.

Klasikinė vizualiosios kriptografijos schema gali būti ir ne būtent tokia, kaip aprašyta prieš tai. Yra įvairių pikselių variacijų, ne tik išplėtimas į šoną, lygiai taip pat galima šifruoti vaizdą plečiant jį į paveikslėlio ilgį, arba šifruoti įstrižais kampais, jei norima turėti dar saugesnį šifrą galime dar labiau praplėsti vaizdą. Visos šios variacijos nekeičia prieš tai aprašytos schemos logikos, tik siūlo platesnį variantų spektrą. Kitų variantų schemos pavyzdžius galime išvysti paveikslėlyje apačioje:

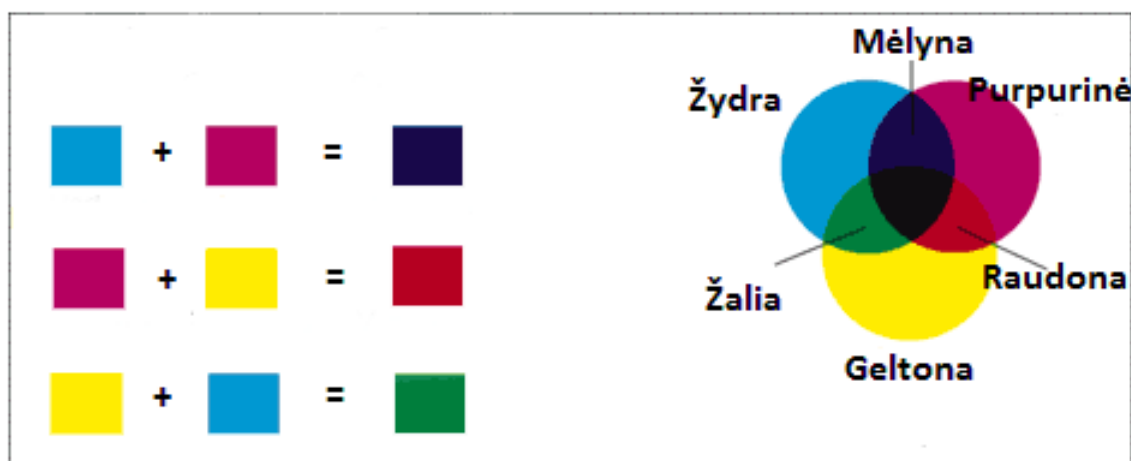


Dviejuose viršutinėse eilutėse palyginimui pateikta jau aptarta schema, trečioje ir ketvirtoje eilutėse nuo viršaus pateiktas schemas pavyzdys su vaizdo praplėtimu į ilgį. Paskutinėse eilutėse pateikiama dar didesnė vaizdo išplėtimo schema. Užšifruojant vaizdą bet kuriuo iš šių būdų kiekvienos eilutės s_1, s_2 yra 50 % tikimybė būti pasirinktiems šifruojant atitinkamai baltos spalvos pikselį (kairysis stulpelis paveikslėlyje) ir juodos spalvos pikselį (dešinysis stulpelis paveikslėlyje).

Siekiant išvengti pikselių plėtimo Ching-Nung Yang [5] pasiūlė naudoti tikimybinę vizualiąją kriptografiją. Šios schemas specifiška yra kiek kitokia, formuojant pikselius iššifravimui, šie nėra praplečiami. Vietoj praplėtimo atsitiktinai pasirenkamas tik vienas dalijimosi pikselis iš $s_1, \dots, s_n$ dalių. Formuojant užšifruotą vaizdą tikimybinės kriptografijos būdu nėra garantijos, kad užkoduoti pikseliai bus atkurti tiksliai taip pat. Apibūdžiant šią schemą taip pat reiktų pastebėti, kad vietoj vieno užkoduojamo pikselio dalinimuisi skirtose dalyse $s_1, \dots, s_n$ bus sukuriamas tik vienas pikselis, o ne pikselių pora, kaip kad buvo pateikta prieš tai pavyzdžiuose.

1.2 Tolimesnė vizualiosios kriptografijos raida

Pirminis vizualiosios kriptografijos variantas buvo pristatytas nespalvotiems vaizdams, tolimesnėje šios srities raidoje aprašomi metodai ir spalvotiems vaizdams. Vizualiosios kriptografijos pritaikymas vaizdams tik iš baltų ir juodų pikselių yra pakankamai suprantamas ir mūsų rega lengvai apdoroja informaciją, pavyzdžiui, jei ant baltos permatomos spalvos uždėsime juodą, tai ir matysime juodą, o su spalvotais pikseliais yra jau šiek tiek kitaip. Apžvelkime sunkumus, kurie gali kilti su spalvotais pikseliais. Visu pirma, reikia suprasti, kokią spalvą žmogaus akys matys sudėjus dvi permatomų spalvų skaidres vieną ant kitos. Persidengiančius spalvotus pikselius mūsų regos sistema supranta jau kaip kitas spalvas. Iliustruotame pavyzdyje galima matyti, kokias spalvas gausime maišydami pikselius. Sumaišius geltoną pikselį su purpuriniu gausime raudoną pikselį, su kitomis spalvomis atitinkamai, kaip pavaizduota pavyzdyje:

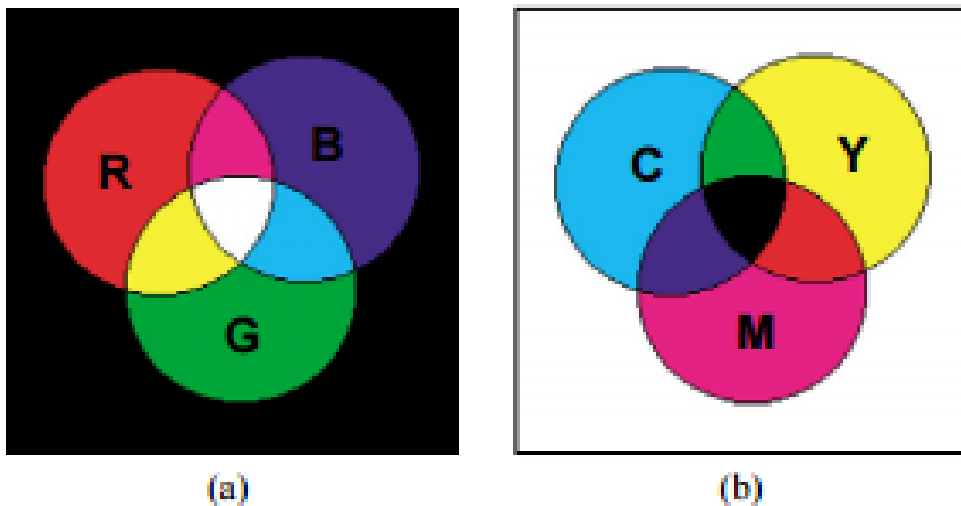


Modernus šviesos ir spalvų principų supratimas yra paremtas Niutono ir kitų garsių mokslininkų darbais. Spalvos ir šviesos dėsnių paaiškinimas neįeina į aptarinėjamos temos medžiagą, tačiau reikia žinoti bent bazines žinias norint suprasti spalvotų pikselių vizualiąją kriptografiją. Šviesa susideda iš elektromagnetinės energijos, kurių bangų ilgis yra maždaug nuo 350 iki 750 *nm*. Šis matomas laukas yra tik maža dalis viso elektromagnetinio spektro. Tiesa, matomumo laukas kiekvienam žmogui individualiai skiriasi, jis gali būti šiek tiek mažesnis ar didesnis. Kai konkretus bangos ilgis (iš prieš tai parašyto matomumo intervalo) pasiekia mūsų akies tinklainę, tinklainė perteikia šios bangos ilgio informaciją kaip spalvą. Mūsų matomame spalvų spektre trumpesni bangų ilgiai atpažįstami kaip melsvos spalvos, viduriniai kaip žalsvi, o ilgiausi bangų ilgiai - rausvai. Mūsų regai susidūrus su keliais skirtingais bangų ilgiais gauta informacija apdorojama kaip jų suma. Akiai susidūrus su visais matomais bangų ilgiais, regai gauta suprantama spalva - balta. Visi bangų ilgiai atitinka kažkurią spalvą, taigi spalvų spektras yra begalinis. Tam tikri objektai mums atrodo esantys konkrečios spalvos tik todėl, kad objektas absorbuoja jį pasiekusią šviesą, tai yra absorbuoja bangos ilgį, o likusį bangos ilgį mes suprantame kaip išskirtinai to objekto spalvą. Pavyzdžiui, jei mūsų akiai matomas objektas yra raudonos spalvos, vadinasi objektas geriausiai atspindi raudoną šviesą, o visus kitus gautus bangos ilgius sugeria labiau nei

atspindi. Grįžtant prie mūsų temos, jeigu turime permatomas skaidres ir šias pasiekia šviesa, tai šviesos banga, kuri nėra sugerama perduodama per visas permatomas skaidres.

Taip pat reikėtų paminėti naudojama spalvų modelį RGB, tai toks modelis, kuris spalvas gauna sudėtinu būdu iš trijų pirminių spalvų - raudonos, žalios ir mėlynos. Tai taip pat vienas standartinių modelių naudojamas kompiuteriuose, televizoriuose ir panašiai. Taip pat, spalvas dažnai ir aprašo kaip trijų kintamųjų rinkinį (x, y, z) sugeneruojantį $(N + 1)^3$ skirtingų spalvų, kur kintamieji atitinka praleidžiamą spalvos kiekį (x - raudonos, y - žalios, z- mėlynos). Spalvos kiekis čia yra sveikasis skaičius režiuose $[0, N]$, įprastai kompiuteriuose N - numatytasis skaičius 255, toliau visi susiję skaičiavimai ir bus pateikti pagal skaitmeninius skaičiavimus. Baltą spalvą atitinkantis rinkinys yra $(255, 255, 255)$, o juodą rinkinys $(0, 0, 0)$. Turint dvi spalvotas permatomas skaidres S1, S2 ir sudėjus jas vieną ant kitos galime išsiaiškinti kokia spalva bus, pagal formulę $(x1 * x2/255, y1 * y2/255, z1 * z2/255)$ pritaikytą kiekvienam pikselių rinkiniui $(x1, y1, z1)$ iš S1 ir $(x2, y2, z2)$ iš S2.

Spalvotų vaizdų kriptografijos schemas be RGB spalvų rinkinio naudos ir CMY spalvų rinkinį, kuo jie panašūs ir kuo skiriasi galima spręsti iš pateikto paveikslėlio iš [13] šaltinio:



Paveikslėlyje (a) pateikti RGB spalvų rinkinio persidengimai, o paveikslėlyje (b) - CMY rinkinio.

1.2.1 Vizualioji kriptografija spalvotiems vaizdams

Vizualiosios kriptografijos schemų spalvotiems vaizdams yra pakankamai nemažai, todėl visų jų neapstarinėsime.

Vienas iš tokių algoritmų [14] išsiskiria vaizdo kaukės naudojimu. Remiantis algoritmu, turint spalvotą paveikslėlį jo užkodavimą panagrinėkime pažingsniui:

1. Išskaidome turimą vaizdą pagal (C, M, Y) spalvų rinkinio komponentes.
2. Pagal prieš tai išskaidytus komponentus gausime tris tuos pačius vaizdus, tik jau su kita vyraujančia komponente.

3. Formuojame kaukę atitinkamai pagal visų trijų gautų vaizdų pikselių spalvos reikšmes. Visos galimos reikšmės: $(0, 0, 0)$, $(0, 0, 1)$, $(0, 1, 0)$, $(0, 1, 1)$, $(1, 0, 0)$, $(1, 1, 0)$, $(1, 0, 1)$, $(1, 1, 1)$, kur 0 - balta pikselio spalva iš prieš tai suformuoto vaizdo, 1 - pikselio spalva atitinkamai pagal kuri vieną iš trijų vaizdų rašome, jei pagal pirmąjį žydrai baltą paveikslėlį pikselis yra baltas tai rinkinio pirmoji reikšmė - 0, o jei žydrai tai atitinkamai - 1. Reikia išskirti du atvejus, kad $(0, 0, 0)$ čia reiškia baltą pikselį, o $(1, 1, 1)$ - juodą.
4. Kiekvienam pikseliui uždedame sugeneruotą kaukę pagal žemiau pateiktą schemą [13].

	$(0, 0, 0)$					$(1/2, 1/2, 1/2)$
	$(1, 0, 0)$					$(1, 1/2, 1/2)$
	$(0, 1, 0)$					$(1/2, 1, 1/2)$
	$(0, 0, 1)$					$(1/2, 1/2, 1)$
	$(1, 1, 0)$					$(1, 1, 1/2)$
	$(0, 1, 1)$					$(1/2, 1, 1)$
	$(1, 0, 1)$					$(1, 1/2, 1)$
	$(1, 1, 1)$					$(1, 1, 1)$

Paveikslėlio pirmajame stulpelyje galime matyti naudojamą vaizdo kaukę. Antrame stulpelyje surašytos galimos pikselių reikšmės pagal tris sugeneruotus vaizdus remiantis (C, M, Y) spalvų rinkiniu. Sekančiuose trijuose stulpeliuose pateikiami pikselių išsidėstymai pagal antrąjį stulpelį. Šeštajame stulpelyje jau galime matyti užkoduotą pikselį. Paskutiniame stulpelyje aprašomas atskleidžiamos spalvos kiekis. Iš schemos taip pat matosi, kad buvę balti pikseliai rezultate tokie nebebus. Visa metodo duomenų paslėpimo specifika atlieka naudojama kaukė. tik žinant kaukę bus galima iššifruoti pradinį spalvotą originalų paveikslėlį.

Šios schemos pagalba galima taip pat įsitikinti spalvų rinkinio (C, M, Y) naudojimo pagrįstumu, kadangi visus spalvų atvaizdžius spalvotame vaizde performuojame į tris pagrindines rinkinio spalvas - žydrą, purpurinę ir geltoną, taip pat dar pasinaudojame balta spalva.

Norint užšifruoti spalvotą paveikslėlį galime pasinaudoti vizualiosios kriptografijos sukeitimo modeliu: vėl taip pat transformuosime spalvotą pradinį vaizdą į tris išskaidytus pus-tonius vaizdus pagal žydrą, purpurinę ir geltoną spalvas. Kiekvienam pradinio paveikslėlio pikseliui reikia atlikti sekančius veiksmus:

1. Išskleisti pikselį į pikselių rinkinį 2×2 į S1 paveikslėlį ir užpildyti pikselių rinkinį žydra, purpurine, geltona ir permatoma (kuri mūsų akiai atitinka baltą spalvą) spalvomis.

2. Sugeneruoti pikselių rinkinį 2×2 į S2 paveikslėlį pagal punkte prieš tai atitinkamų pikselių išskaidymus S1 paveikslėliui, nustatyti spalvų pasiskirstymą pikselių rinkinyje pagal iliustruotą pavyzdį iš [13] šaltinio.

Spalva pagal rinkinį CMY	S1	S2	S1 ir S2 persidengimas	Metodas	Gautas rezultatas
(0, 0, 0)				S1 ir S2 su ta pačia spalvų kombinacija	
(1, 0, 0)				Žydros ir permatomos spalvos pozicijos sukeitimas	
(0, 1, 0)				Purpurinės ir permatomos spalvos pozicijos sukeitimas	
(0, 0, 1)				Geltonos ir permatomos spalvos pozicijos sukeitimas	
(1, 1, 0)				Žydros ir purpurinės spalvos pozicijos sukeitimas	
(0, 1, 1)				Geltonos ir purpurinės spalvos pozicijos sukeitimas	
(1, 0, 1)				Žydros ir geltonos spalvos pozicijos sukeitimas	
(1, 1, 1)				Pikselių poros sukeitimas	

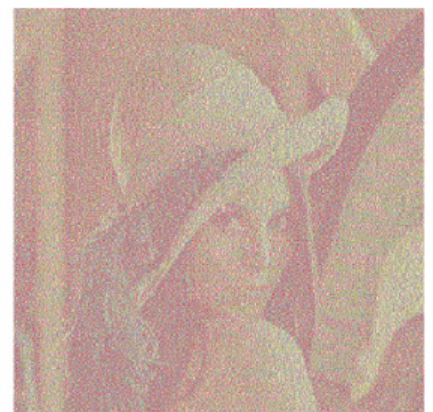
3. Kartoti antrąjį žingsnį, kol visi pradinio vaizdo pikseliai bus transformuoti, gausim dvi vizualiosios kriptografijos skaidres. Sudėjus šias dvi skaidres, slaptasis paveikslukas bus dekodotas regos pagalba. Dekoduoto paveikslėlio pavyzdys iš teksto [13] :



(a)



(b)



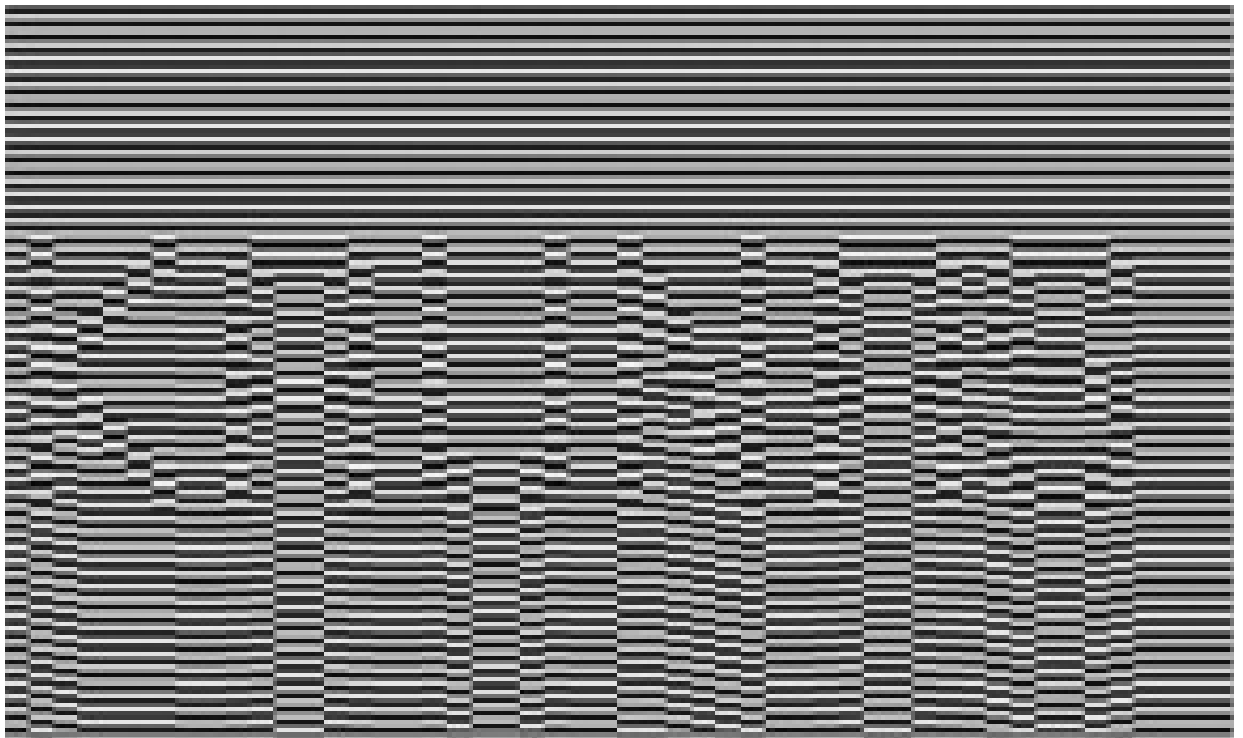
(c)

1.2.2 Dinaminė vizualioji kriptografija

Laikui bėgant, nuo 1994 metų vizualiosios kriptografijos pristatymo, vizualioji kriptografija gerokai išsiplėtė ir be jau aprašytų metodų 2009 m. išsiskyrė kita atšaka - dinaminė vizualioji kriptografija. Ši atšaka jau gerokai skiriasi nuo pradinės vien dėl to, kad skiriasi saugumo atžvilgiu, bet šis rodiklis taip pat priklauso nuo naudojamos schemos. Dar vienas skirtumas yra, kad užkoduojant vaizdą dinaminės vizualiosios kriptografijos būdu, vaizdas nėra skaidomas į atskiras skaidres, tai yra iš vieno vaizdo gauname tik vieną užkoduotą vaizdą. Kiekviena slapta žinutė naudojantis šiuo metodu išskaidoma į dvi dalis - žinutės informaciją ir jos foną.

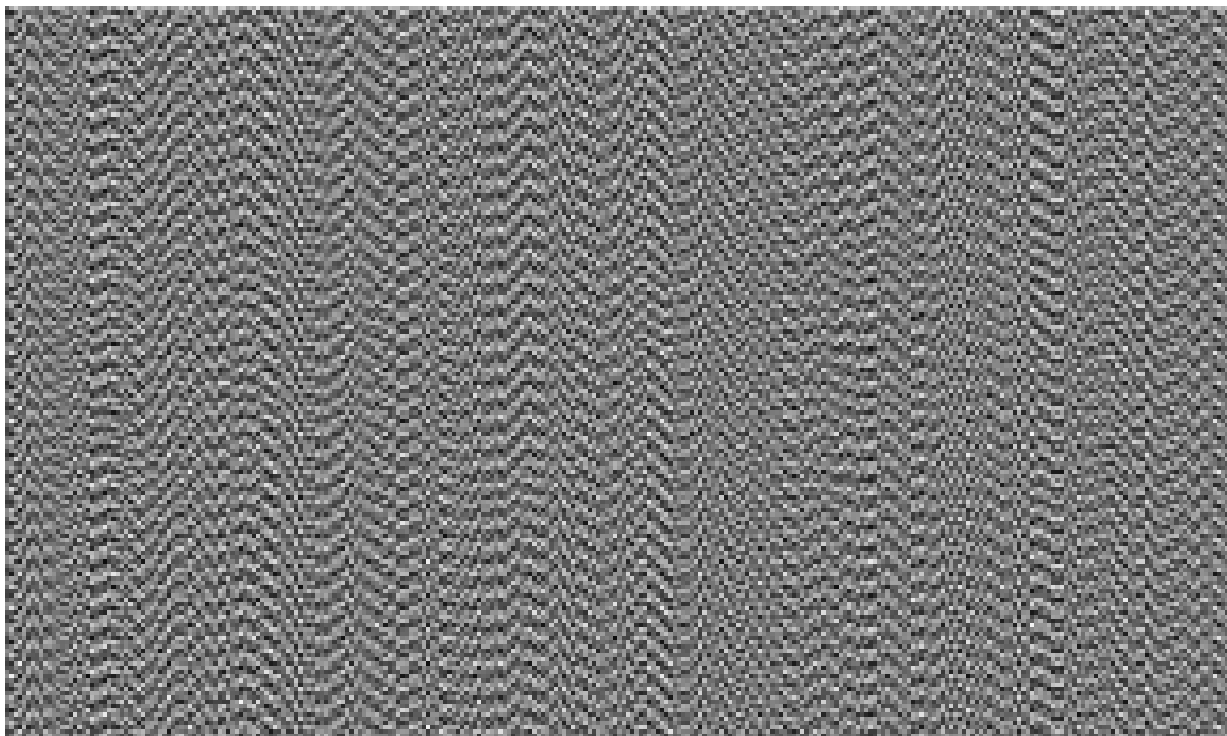
Panagrinėkime tokio metodo pavyzdžius su iliustracijomis iš [2] šaltinio. Slapto vaizdo informacija perkonstruojama į tikimybinę gardelę. Vaizdo užkodavimas naudojantis šia schema yra pakankamai sudėtingas, reikalaujantis fizikinių optikos žinių ir remiasi sudėtingais skaičiavimais, nepaisant to, iššifruojamas vaizdas remiasi vizualiosios kriptografijos metodais. Vaizdo iššifravimui, kaip ir įprastai vizualiajai kriptografijai, užtenka pasinaudoti įprasta žmogaus rega ir keliais mechaniniais judesiais. Iššifruojamas vaizdas išryškėja papilkėjusių zonų pavidalu.

Panagrinėkime plačiau šios schemos pavyzdžius. Pasinaudokime tos pačios autorės [2] pavyzdžius demonstruojant slapto vaizdo šifravimą ir iššifravimą. Originali mūsų turima paslaptis - žodis „KAUNAS“. Paslaptis paslepama paveikslėlio pavidalu. Šios paslapties pritaikymas metodui, įdedant paslapties pikselius taip lyg iš vaizdo sudarinėtume grotelės struktūros pavidalą, atrodys taip, kaip pateikta paveikslėlyje žemiau iš [2] šaltinio:

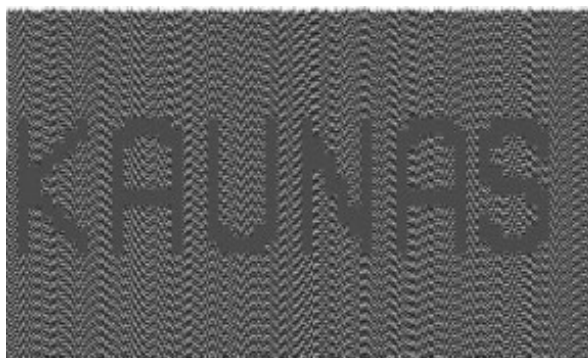


Tokiame pavidale slaptąjį užrašą nesunku įžiūrėti plika akimi be didesnių pastangų. Šiam žodžio „KAUNAS“ šifravimui į gardelės foną panaudotas grotelių nuolydis $\lambda_0 = 0.19$, o

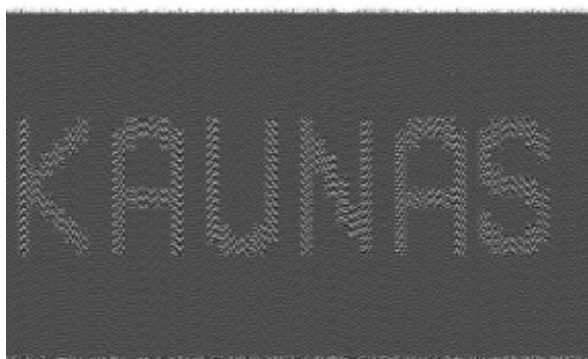
nuolydis tuose vietose, kuriuose yra šifruojamas tekstas parinktas $\lambda_1 = 0.15$. Pritaikius atsitiktinių nukrypimų procedūrą vaizdas bus matomas kitaip [2]:



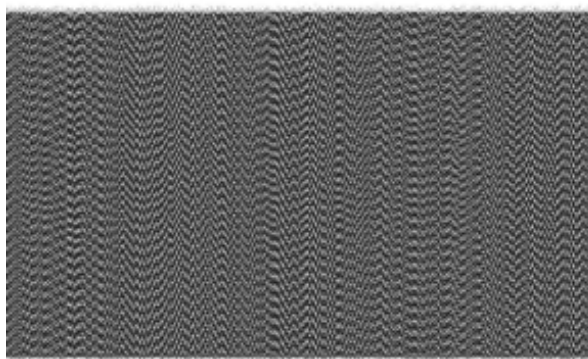
Kompiuteriu iššifruotas vaizdas atrodys kaip pateikta paveikslėliuose žemiau. Gautus vaizdu pavyko gauti atitinkamais harmoniniais virpesiais, kur virpesiai $A = 0.0574$; $B = 0.0747$; $C = 0.1334$. Pavyzdys pasinaudotas iš [2] šaltinio.



A



B



C

Galima pateikti ir kiek kitokį dinaminės kriptografijos metodą pasiūlytą Wu ir Chen [15]. Šis algoritmas yra kiek paprastesnis ir artimesnis jau prieš tai apibūdintai standartinei vizualiajai kriptografijai. Šis metodas ypatingas tuo, kad gali šifruoti kelis vaizdus su tomis pačiomis dvejomis užšifruotomis originalių vaizdų dalimis. Pavyzdžiui, turime du užšifruotus vaizdus s_1 ir s_2 skirtus dalijimuisi. Uždėjus s_2 vaizdą tiksliai ant s_1 vaizdo matysime paslėptą originalų vaizdą, viskas taip pat kaip ir klasikinėje vizualiosios kriptografijos schemoje. Šiame metode skirtumas atsiranda tik kai s_2 vaizdą pasukame 90 laipsnių kampu taip, kad jis dar būtų tiksliai uždėtas ant s_1 vaizdo. Po posūkio matysime antrąją paslėptą žinutę.

2 Vizualiosios kriptografijos pritaikymas

Vizualioji kriptografija pasiūlyta Naor ir Shamir yra pavyzdys kriptografijos schemoms, kurių ypatybė - dekoduoti užslėptus paveikslėlius be jokių kriptografinių skaičiavimų. Vizualioji kriptografija gali būti taikoma priėjimo kontrolėse, autorinių teisių apsaugai, vandens antspaudams, vizualiajam autentikavimui ir identifikacijai [6].

2.1 Vizualiosios kriptografijos pritaikymo pavyzdžiai

Pagrindinis vizualiosios kriptografijos pavyzdys realizuoja paslapties dalijimo kriptografijos protokolą. Sutartoje paslapčių dalijimosi schemeje paveikslėlio paslaptis dalijama n dalyvių tokiu būdu, kad atkurti paslaptį gali tik tie dalyviai, kuriems yra leidžiama, kiti dalyviai, kuriems uždrausta išsiaiškinti paslaptį negaus jokios informacijos. Naudojantis šiuo modeliu galima įvesti prieigos kontrolę. Įsivaizduokime, kad reikia pergabenti kokį nors ypatingai svarbų meno kūrinį, kaip garsioji Mona Liza nutapyta Leonardo da Vinčio, į kitą saugią vietą. Šiai operacijai yra paskirti 5 asmenys, tačiau pagal saugumo nurodymus tik trys iš jų gali gauti saugos kodą. Mona Liza paveikslo apsaugos kodo negali gauti du asmenys iš penkių. Šią problemą įmanoma išspręsti pagal vizualiosios kriptografijos k iš n schemą, kur k bus lygu 3, o n lygu 5. Tokios schemas tik su atitinkamomis reikšmėmis gali būti taikomos elektroninių pinigų, skaitmeninių apklausų ar rinkimų kriptografijos protokoluose.

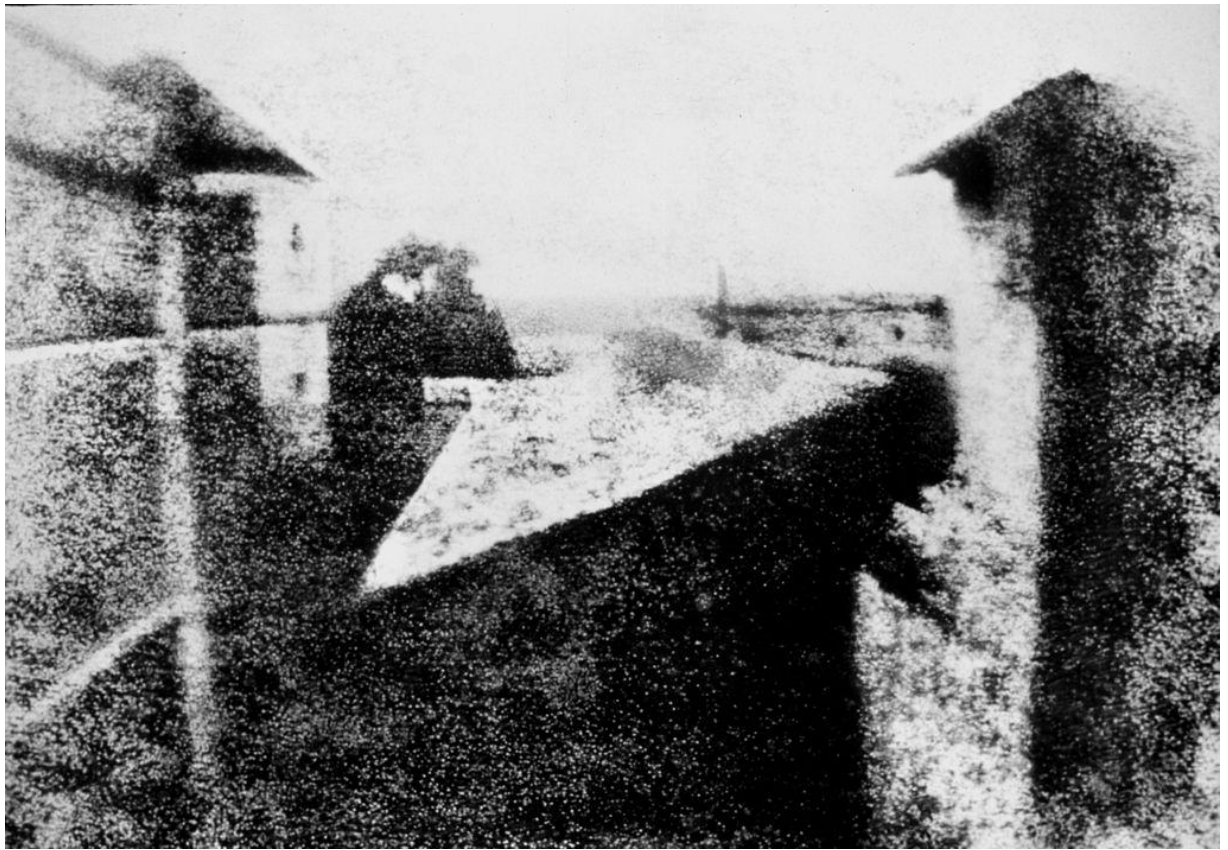
Kitas galimas vizualiosios kriptografijos pritaikymas yra apsimestinių vaizdų metodas, kuris papildomai naudojami kita kriptografijos atšaka - steganografija. Pagal įprastą steganografijos schemą siuntėjas išsiunčia paveikslėlį norimam gavėjui, gavėjas iš anksto susitaręs su siuntėju žino, kad atsiųstas vaizdas iš tikrųjų slepia kitą vaizdą ir pagal susitartą schemą gauna tikrąjį vaizdą paslėptą pradiniam siųstame vaizde. Pašaliniams neleistinai gavusiems siųstą vaizdą gali net nekilti įtarimo, kad po juo kažkas paslėpta. Šiam steganografijos modeliui galimas vizualiosios kriptografijos pritaikymas, kai siunčiamas vaizdas $B_1, \dots, B_m$ gavėjams, o norima, kad tik B_1 gavėjas pamatytų slepiamą vaizdą. Šis būdas gali neatrodyti nei kiek įtartinais, priešingai nei klasikinis kriptografijos metodas. Sudarant kelias juodai baltų pikselių dalijimosi dalis, jas nėra patogu kaupiti ir laikyti, kitiems asmenims gali įtartinais pasirodyti vien tai, kad tarp asmeninių failų yra pilna juodai baltai mirgančių vaizdų.

Galvojant apie dažnesnius atvejus mūsų kasdieniniame gyvenime ir šiek tiek paprasčiau žvelgiant galima rasti socialinių sferų, kur tiktų vizualiosios kriptografijos metodai. Pavyzdžiui, įsivaizduokime, kad reikia atlikti apklausą tema, kuri nėra itin maloni. Tai galėtų būti apklausa į kurios klausimas dalyviai būtų labiau linkę meluoti, nei sakyti tiesą. Viena iš priežasčių kodėl žmonės būtų linkę meluoti - informacijos saugumas. Žmonės pildydami anketą gali jaustis saugiau žinodami, kad jų padėtis visuomenėje nepakis dėl atliktos apklausos, tai ypač, būdinga žinomiems asmenims. Vienas iš būdų galėtų būti anketoms pritaikytas vizualiosios kriptografijos algoritmas. Tarkime, apklausą vykdomas asmuo, apklausos dalyviui pateikia apklausos formą, kur prie klausimų atitinkamai yra vizualiosios kriptografi-

jos schema, naudojantis kompiuteriu sugeneruojami tai apklausai išskirtini klausimų reikiamumo atsakyti lygiai. Vienas lygis "T" - atsakymams bus formuojamas ir siunčiamas į duomenų bazę, jį reikia atsakyti teisingai, kitas lygis "N" - atitinkamo klausimo atsakymas sistemai nėra būtinas ir jo neįtrauks į bendrą statistiką, į šį klausimą galima pateikti klaidingą atsakymą. Šiuos reikiamus klausimus konkrečiai anketai tuo momentu žinos tik apklausos dalyvis ir sistema, sistema šios informacijos nesaugotų dėl saugumo sumetimų.

2.2 Vaizdo įrašų ir vizualiosios kriptografijos suderinimas

Norint suprasti vaizdo įrašų pritaikymo galimybes vieno vaizdo šifravimo technikai reikėtų prisiminti vaizdo įrašų istoriją. Remiantis [12] vaizdus žmonija naudoja jau ne mažiau kaip trisdešimt tūkstančių metų. Tai liudija dar urvų tapyba, kuri prisidėjo prie evoliucijos ir civilizacijos vystymosi. Urvų tapyba tuo laikotarpiu išsivystė į kalbas. Vienas didesnių informacijos perdavimo šuolių kilo, kai menininkai savo darbuose pradėjo taikyti perspektyvą ir realistinį stilių, kuris atvaizdavo realų pasaulį. Toks šuolis į tikslumą ir detales padėjo meną pritaikyti mokslui ir mokslinei informacijai perduoti. Norint perteikti kuo tikslesnį vaizdą taip pat buvo išrasta "juodojo kambario" technologija ("Pinhole" angl. k., camera obscura - lot. k.). Tokia technika šiomis dienomis įvardijama kaip pirmosios fotokameros pradininkė ir vis dar yra naudojama. "Juodojo kambario" technikos veikimas apibūdinamas kaip fotoaparato be objektyvo veikimas. Tokiame "juodajame kambaryje" suprojektuojamas šviesos srautas einantis per mažą ertmę ant jautraus paviršiaus toks kaip foto popierius ar juostelė, įvairūs garsūs menininkai naudojami šia technika, kad išgautų tikslesnius vaizdus savo kūriniams perpiešti, be to, mokslininkai naudojami ja įvairiems matavimams atlikti. Toks projektavimas paskatino ir pirminių projektorių atsiradimą, kurie ant stiklo nupieštus piešinius projektavo ant sienos. 1826 metais prancūzų mokslininkas Joseph Nicéphore Niepce sukūrė pirmąją fotografiją, naudojantis jau minėta "Camera Obscura" technika. Šios fotografijos [10] vaizdas pateikiamas žemiau:



Fotografijoje užfiksuotas vaizdas pro Le Gras langą.

XIX amžiaus pabaigoje fotografijų sferoje įvyko progresas išrandant prietaisus judesiams nuotraukoje fiksuoti [11]. Tuo metu buvo kuriami begarsiai, trumpi iki minutės trukmės filmai. Galiausiai 1927 metais buvo sukurtas pirmasis filmas su garsu [9]. Visa tai tobulėjo ir galiausiai vaizdai, vaizdo įrašai perėjo į skaitmeninius pavidalus. Taigi, vaizdo įrašams tikrai galime pritaikyti vizualiosios kriptografijos metodus, kadangi patys vaizdo įrašai yra kilę iš paveikslėlių.

2.2.1 Vaizdo įrašų ypatybės

Siekiant išvengti nesusipratimų perteikiant informaciją, informacijos formatai vis keitėsi link aiškesnių formų, nuo piešinių urvuose link vaizdo įrašų. Šiuo metu technologijos mums leidžia turėti dar daugiau informacijos perdavimo formų. Žmonės verbaliniu bendravimo būdu dažniausiai perduoda tik dalį informacijos, kita dalis informacijos būna pateikiama neverbaliniu kalbos būdu – pavyzdžiui, gestais. Tekstu yra pakankamai sudėtinga apibūdinti neverbalinę informaciją, tačiau naudojantis technologijomis – vaizdo įrašymo, garso įrašymo prietaisais galime užfiksuoti ir perduoti visą pradinę žinutę nuo pranešėjo, neiškerpant nei dalies teksto ir neatskiriant vaizdo nuo garso. Taigi, didžiausias vaizdo įrašų privalumas palyginti su nuotraukomis ar tekstine informacija – tikslesnis informacijos perdavimas.

Dirbant su vaizdo įrašais taip pat reikėtų žinoti jų technines detales, ne tik bendrą vaizdo įrašo veikimo principą. Vienas svarbiausių vaizdo įrašų parametrų - kadro dydis per sekundę,

tai nusako kaip sklandžiai matysime vaizdo įrašą [8]. Judančio vaizdo standartinis kadro skaičius per sekundę yra 24. Tiesa, dokumentinėse kamerose dažnai naudojamas 30 kadro per sekundę standartas. Šie standartai nusako kaip greitai keisis kadrai per vieną sekundę. Norint imituoti judėjimą užtenka ir 10 kadro per sekundę. Kitas svarbus parametras - raiška. Skaitmeninio vaizdo įrašo raiška kaip ir bet kokio kito skaitmeninio vaizdo raiška matuojama pikseliais. Nuo šių parametrų priklauso vaizdo įrašo kokybė.

Kiekvienas skaitmeninis vaizdo įrašas turi savo raišką. Vaizdo įrašo rezoliucija arba bet kokio skaitmeninio vaizdo rezoliucija - parametras nusakantis vaizdo plotį ir ilgį. Dažniausiai šis dydis aprašomas tokiu pavidalu : plotis pikseliais x ilgis pikseliais. Rezoliucija nusako kiek pikselių yra atvaizduojama. Vaizdo įrašo formatas tai pat vienas iš vaizdo įrašų parametrų, nusakantis kokio formato tipu vaizdo įrašo failas bus saugomas skaitmeninėje erdvėje. Formatas yra svarbus vaizdo įrašo parametras, kadangi gali lemti vaizdo įrašo dydį. Dažniausiai vaizdo įrašų formatai pasitarnauja kaip vaizdo įrašo informacijos suspaudikliai. Vaizdo įrašų standartiniai konteineriai AVI, Quicktime, Windows Media Video dažnai maišomi su vaizdo įrašo formatu. Vieni žymesnių formatų - MPEG-4 ar MPEG-2 ar bet kurių kitų formatų vaizdo įrašai gali būti panaudoti vaizdo įrašė su .avi plėtiniu.

2.2.2 Vizualiosios kriptografijos pritaikymas vaizdo įrašams

Vizualiąją kriptografiją galime nesunkiai pritaikyti vaizdo įrašė esančiam vaizdui. Visi vaizdo įrašai susideda iš kadro, skiriasi tik kadro skaičius per sekundę ir kadro dydis. Susitarus dėl standartinio kadro skaičiaus n per sekundę, kur n turi būti toks, kad įprastai žmogaus regai vaizdo įrašo kokybė per daug nenukentėtų ir vaizdas neatrodytų lyg skaidrių peržiūra, o ne vientisas filmas. Reiktų pastebėti, kad jei pasirinktume skaičių n mažesnį, nei apdorojamo vaizdo įrašo kadro skaičių per sekundę, tai vaizdo įrašo kokybė galimai sumažės, tačiau įprastai žmogaus regai, tai didelio skirtumo nesudarys.

3 Sukurtos programos analizė ir pavyzdžiai

Parašytoji programa skirta pademonstruoti užšifruotus vaizdo įrašus naudojantis vizualiosios kriptografijos schema. Kiekvienam vaizdo įrašo kadrai pritaikomas klasikinės vizualiosios kriptografijos algoritmas, kuris paslaptį dalina į dvi užkoduotas dalis. Kai vizualiosios kriptografijos schemos pritaikomos vaizdams, rezultate būtų gaunami du atskiri užkoduoti vaizdai, tai lygiai taip pat ir su kadrais, iš vieno originalaus kadro gausime du užkoduotus kadrus. Iš užšifruotų kadru gausime du užšifruotus vaizdo įrašus. Programa pasirinkta šifruoti tik su nespaltvotų vaizdų vaizdo įrašus, todėl spalvotus vaizdo įrašus programa konvertuos į juodai baltus be jokių pilkų atspalvių.

3.1 Programos specifiika

Programa parašyta C# programavimo kalba naudojantis Windows Form aplikacijų šablonu. Programos pilnam veikimui įgyvendinti reikalingi plėtiniai:

1. AForge Video FFMPEG
2. NReco Video Converter

Pirmasis plėtinys yra iš atviro kodo šablono AForge.NET FFmpeg skirto C# programuotojams dirbantiems su kompiuterine grafika, paveikslėliais ir kitais vaizdais. AForge Video FFMPEG vardų klasė savyje turinti klases, kuriose įgyvendintas duomenų skaitymas ir rašymas iš / į vaizdo įrašo failus naudojantis FFmpeg biblioteka. Nors šis plėtinys ir leidžia vaizdo failus įrašyti į vaizdo įrašus, tačiau negalime įrašyti garso failų. Programoje naudojama tik VideoFileWriter klasė vaizdo informacijai įrašyti į naujai sukuriamą vaizdo įrašo failą.

Antrasis plėtinys NReco Video Converter, kuris naudingas tuo, kad gali apdoroti pakankamai nemažai vaizdo įrašų formatų, kaip kad mp3, mp4, avi, wav, wmv, flv, mov. Šioje programoje naudojamas kadru iš vaizdo įrašo nuskaitymo procesui. Pasirinktas būtent dėl suderinamumo su įvairiais vaizdo įrašų formatais.

Programa į vaizdo šifravimo procesą neįtraukia vaizdo įrašo garso, vaizdo įrašo garsas ne tik nėra šifruojamas, bet taip pat ir nėra perduodamas originalusis garsas tiek užkoduotoms dalims, tiek vidiniams procesams su apdirbamais kadrais ar iššifruojamam vaizdai.

Atsižvelgiant į šifravimo spartumą ir apdorojamos informacijos dydį, nuspręsta vaizdo įrašo šifravimui pritaikyti schemos modelį vaizdai susidedančiam tik iš juodų ir baltų pikselių.

Pasirinktas norimas šifruoti vaizdo įrašas bus apdorojamas kaip 24 kadru per sekundę vaizdo įrašas, taigi, jei originalus vaizdo įrašas turėjo daugiau ar mažiau kadru per sekundę, iššifruotasis vaizdo įrašas turės lygiai 24 kadru per sekundę. Be to, visi programos veikimo metu sudaryti vaizdo įrašai sukuriama avi vaizdo įrašų formatu.

3.2 Programos veikimo principas

Programos pagrindinė idėja - užkoduoti pasirinktą vaizdo įrašą klasikinės vizualiosios kriptografijos schemos principu. Programos veikimą paprasčiausia paaiškinti pažingsniui:

1. Pirmiausiai pasirenkame vaizdo įrašą, kurį norėsime užšifruoti.
2. Pasirinkus vaizdo įrašą atsiranda galimybė jį užšifruoti. Pasirinkus užšifravimo mygtuką programa sureaguoja į šį veiksmą tolimesniais žingsniais:
 - Perskaitomas gautas vaizdo įrašo failas ir pradedami traukti kadrai iš vaizdo įrašo.
 - Kiekvienas kadras performuojamas taip, kad susidėtų tik iš juodų ir baltų pikselių.
 - Kiekvienam juodai baltam kadrai vykdomė transformacijas pagal vizualiosios kriptografijos schemas ir užšifruojam turimą vaizdą. Rezultate iš kiekvieno kadro gauname po du užkoduotus kadro vaizdus skirtus dalinimuisi.
 - Užkoduotus kadrus atitinkamai eilės tvarka sudedame į du sąrašus, kurių dėka bus suformuojami du užkoduoti vaizdo įrašai dalinimuisi.
 - Vaizdo įrašai įrašomi į kompiuterio atmintį.

Kitas galimas programos veikimas, kai norime iššifruoti, kas paslėpta turimuose užkoduotuose vaizdo įrašuose:

1. Pasirenkame du užkoduotus vaizdo įrašus, kurie slepia originalų vaizdo įrašą.
2. Spaudžiam mygtuką skirtą iššifravimui. Paspaudus šį mygtuką vyksta vidiniai procesai:
 - Vaizdo įrašai išskaidomi į kadrų rinkinius.
 - Kadrų rinkiniai vienas po kito iššifruojami naudojantis klasikine vizualiosios kriptografijos schema. Iš atskirų vaizdo įrašo užkoduotų kadrų rinkinių gauname vieną iššifruotą kadrų rinkinį.
 - Iššifruotus kadrus eilės tvarka perkeliame į naujai sukurtą vaizdo įrašą ir išsaugome.

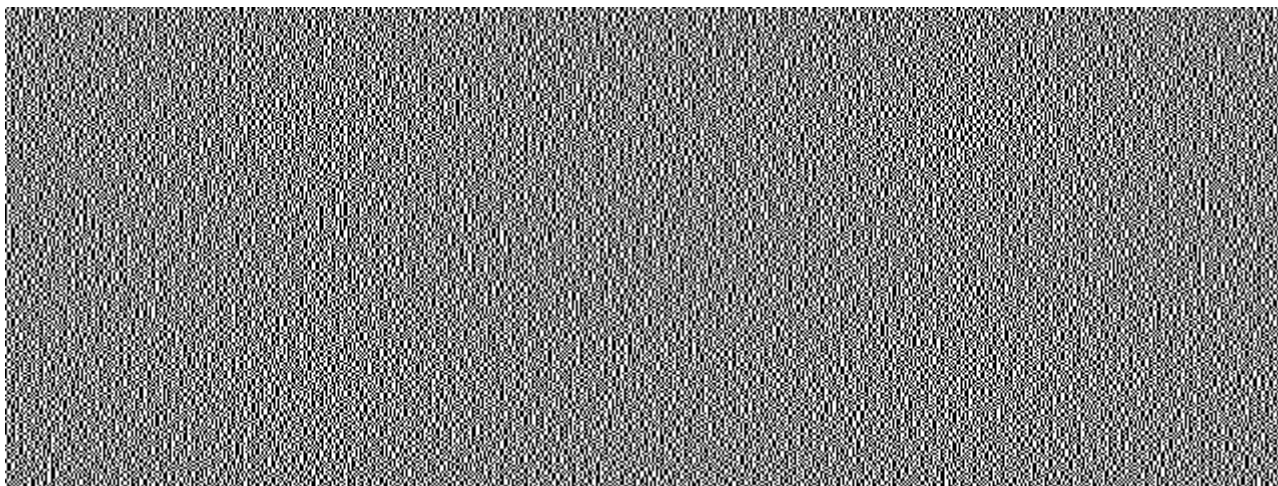
Nagrinėjant žingsnius detaliau, apžvelgsime kaip keičiasi vaizdo įrašo kadrų parametrai programos veikimo metu. Tarkime, paėmus pirmą pavyzdinį kadrą programai šifruojant pavyzdinį vaizdo įrašą.

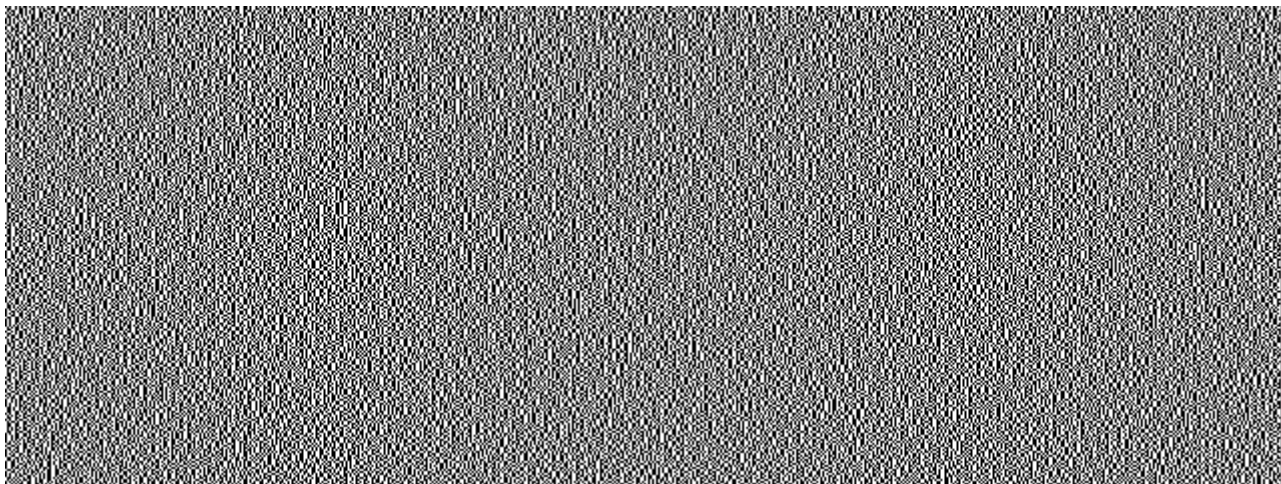


Šio vaizdo dimensija 320×240 , dydis 13,8 KB, vaizdo kaip paveikslėlio plėtinys - jpg. Šiam vaizdai pritaikius pikselių spalvos keitimo algoritmą, kurį aprašysime sekančiame poskyryje, vaizdas pasikeičia į nespaltvotą:



Pasikeitusio vaizdo dimensija išlieka tokia pati 320×240 , dydis sumažėja iki 6,17 KB, o vaizdo kaip paveiksliuko plėtinys nekeičiamas, paliktas tas pats .jpg . Po sekančio žingsnio - šifravimo pritaikymo vietoj prieš tai matyto kadro turėsime du vaizdus, dvi užkoduotas vaizdo skaidres:





Akivaizdžiai matosi, kad užkoduoti vaizdai yra išsiplėtę. Abiejų užkoduotų vaizdų parametrai yra tokie patys. Dimensija - 640×240 , dydis padidėjo taip pat, dabar jis siekia 40,5 KB, o plėtinys vėl nesikeičia.

3.3 Algoritmai programoje

Programoje naudojami trys algoritmai - pikselių spalvos keitimo kiekvienam kadrai, kiekvieno kadro užšifravimo, kadro iššifravimo algoritmas. Kiekvieną algoritmą išskirsime atskirai ir apibūdinsime jo veikimą vieno kadro apdorojimo situacijoje.

Pirmasis naudojamas - pikselių spalvos keitimas kiekvienam kadrai. Reikėtų pastebėti, kad šis algoritmas yra taikomas net ir nespaltotiems vaizdo įrašo kadrams, siekiant apsisaugoti nuo kadruose galimai likusių spalvotų pikselių. Algoritmo veikimas ilgės nuo didesnio kadro skaičiaus, kuo trumpesnis vaizdo įrašas, tuo mažiau kadro ir tuo greičiau veikia algoritmas. Šis algoritmas gauna keičiamą paveikslėlį *K1* Bitmap klasės pavidalu. Šios klasės dėka galime operuoti kiekvienu paveikslėlio pikseliu. Prieš aprašant algoritmą pateikiama programinio kodo C# kalba ištrauka padėsianti lengviau suprasti algoritmą:

```

public Bitmap makeImageBlackAndWhite(Bitmap vaizdas)
{
    Bitmap resultImage = new Bitmap(vaizdas.Width, vaizdas.Height);

    for (int i = 0; i < vaizdas.Width; i++)
    {
        for (int j = 0; j < vaizdas.Height; j++)
        {
            Color c = vaizdas.GetPixel(i, j);

            int vidurkis = ((c.R + c.B + c.G) / 3);

            if (vidurkis < 128)
                resultImage.SetPixel(i, j, Color.Black);
            else
                resultImage.SetPixel(i, j, Color.Empty);
        }
    }

    return resultImage;
}

```

Pradedame for ciklą, kuris tęsis nuo parametro i , kuris $i = 0$ iki $K1$ pločio dydžio skaičiaus w kiekvienos iteracijos metu prie parametro i pridedant skaičių 1. Viduje šio ciklo yra kitas For ciklas pasidedantis nuo parametro $j = 0$ ir kylantis lyg parametro h , kuris yra $K1$ paveikslėlio ilgio dydis, kiekvienos šio ciklo iteracijos metu prie parametro j pridedamas skaičius 1. Antrojo ciklo viduje atliekame tolimesnius veiksmus:

1. Gauname pikselį p , kurio koordinatės atitinka parametrų (i, j) porą.
2. Gauname pikselio p spalvos RGB rinkinį.
3. Sudedam visus pikselio p RGB rinkinio spalvų kiekius ir padalijame gautą skaičių iš 3, gaudami pikselio spalvos vidurkio parametą v .
4. Pagal gautą v nusprendžiamas rezultato pikselio spalva - juoda ar balta. Tikriname ar $v < 128$, jei mažiau tai rezultato paveikslėlyje pikselis koordinatėmis (i, j) bus juodas, priešingu atveju - baltas.
5. Nustatome rezultato paveikslėliui pikselį (i, j) .

Antrasis algoritmas - šifravimo. Šis algoritmas yra analogiškas pirmuosiuose puslapiuose aprašytui klasikinės kriptografijos schemos algoritmui, todėl šioje vietoje nėra tikslo jo vėl aprašinėti.

Trečiasis algoritmas skirtas iššifravimui, kadangi prieš tai skyriuose nebuvo pilnai aprašyta, kaip jis veikia su skaitmeniniais vaizdais, trumpai jį apžvelgsime. Algoritmui pateikiami du užšifruoti paveikslėliai $U1$ ir $U2$. Pradedame pirmąjį for ciklą su parametrais $i = 0, h$, kuris lygus gautų paveikslėlių $U1, U2$ ilgiui, ciklas prasideda nuo parametro i ir tęsiasi iki parametro h , kiekvienos iteracijos metu pridodant prie i skaičių 1. Viduje yra kitas ciklas for su parametrais $j = 0$ ir w , kuris lygus $U1$ ar $U2$ pločiui padalintam iš dviejų. Antrojo ciklo viduje atliekami veiksmai:

1. Gauname pirmuosius (i, j) ir antruosius $(i + 1, j + 1)$ pikselius iš $U1, U2$.
2. Pagal aprašytas vizualiosios kriptografijos schemos pikselių keitimąsi vykdom pikselio (i, j) tikrinimus pagal tai kokios spalvos šie yra. Tarkim, jei pikselis (i, j) tiek iš $U1$, tiek ir iš $U2$ - juodas, o pikselis $(i + 1, j + 1)$ tiek iš $U1$, tiek iš $U2$ - baltas, vadinasi iššifruoto paveikslėlio pikselis (i, j) bus baltas.
3. Nustatome rezultato paveikslėlio pikselio (i, j) spalvą.

Pateikiamas algoritmo programinis kodas:

```
for (int y = 0; y < height; y++)
{
    for (int x = 0; x < pusePlocio; x++) //int pusePlocio = width / 2;
    {
        Color image1a = image[0].GetPixel(x*2, y);
        Color image2a = image[1].GetPixel(x*2, y);

        if (image1a.ToArgb() == Color.Empty.ToArgb() && image2a.ToArgb() == Color.Black.ToArgb())
        {
            issifruotastat.SetPixel(x, y, Color.Black);
        }

        if (image1a.ToArgb() == Color.Empty.ToArgb() && image2a.ToArgb() == Color.Empty.ToArgb())
        {
            issifruotas.SetPixel(x, y, Color.Empty);
        }

        if (image1a.ToArgb() == Color.Black.ToArgb() && image2a.ToArgb() == Color.Empty.ToArgb())
        {
            issifruotas.SetPixel(x, y, Color.Black);
        }

        if (image1a.ToArgb() == Color.Black.ToArgb() && image2a.ToArgb() == Color.Black.ToArgb())
        {
            issifruotas.SetPixel(x, y, Color.Empty);
        }
    }
}
```

Reikėtų pastebėti, kad iššifravimo algoritmas gali veikti šiek tiek gudriau nei žmogaus akis, pakeisdamas iššifruotame paveikslėlyje turėsiantį gautis pilkšvą foną į baltą. Programos algoritmai buvo optimizuojami, kiek įmanoma labiau siekiant sutrumpinti programos veikimo laiką, pasitelkiant ir prieš tai minėtus plėtinius, nepaisant to reiktų suprasti, kad vaizdo įrašai nėra mažų duomenų failai ir kad šifravimo ir dešifravimo veiksmai gali užtrukti atitinkamai pagal vaizdo įrašo dydį bei trukmę. Ši programa yra įrodymas, kaip nesudėtingai

galime pasinaudoti paprasčiausiu klasikiniu vizualiosios kriptografijos metodu sudėtingesniems vaizdų duomenims apsaugoti. Šis būdas įgalina perduoti tikslesnius šifruotus duomenų pranešimus, jei kartu su duomenimis norima perduoti vaizdų atlikimo specifiką, gestus ar emocijas tiesiogiai nuo pranešimo autoriaus.

Programos veikimas ir galimybės gali būti keičiamos ateityje modifikuojant programinį kodą. Modifikacijos gali būti atliktos siekiant sudaryti daugiau nei du vaizdo įrašus skirtus dalinimuisi. Taip pat gali būti atlikta šifravimo modifikacija pritaikant šifravimo metodus ir spalvotiems vaizdams. Siekiant išpildyti pilną vaizdo įrašo užšifravimą galima būtų pridėti algoritmą vaizdo įrašo garsinei medžiagai užšifruoti.

Literatūros sąrašas

- [1] V. Stakėnas, *Kodai ir šifrai*, Vilnius, 2007.
- [2] R. Palivonaitė, Doktoro disertacijos santrauka: *Chaotinė vizualinė kriptografija*, http://ktu.edu/sites/default/files/santrauka_52.pdf, Kaunas, 2015.
- [3] M. Stamp, *Information Security Principles and Practice*
- [4] B. Schneier, *Secrets & Lies Digital Security in a Networked World*
- [5] S. Cimato, Ching-Nung Yang, *Visual Cryptography and Secret Image Sharing*, CRC Press.
- [6] Ratheesh V.R, Jogesh J., Jayamohan M., *A Visual Cryptographic Scheme for Owner Authentication Using Embedded Shares*, <http://www.ijcse.com/docs/INDJCSE14-05-05-062.pdf>
- [7] J. Weir, W. Yan, *Visual Cryptography and Its Applications*, Ventus Publishing ApS, 2012
- [8] F. Romano, *The history of frame rates*, 2012, <https://vanillavideo.com/blog/2012/history-frame-rates-why-speeds-vary>
- [9] D. A. Cook, *History of the motion picture*, <http://www.britannica.com/art/history-of-the-motion-picture>
- [10] *World's First Photograph*, National Geographic, <http://photography.nationalgeographic.com/wallpaper/photography/photos/milestones-photography/niepce-first-photo/>
- [11] *Motion pictures*, http://www.encyclopedia.com/topic/motion_pictures.aspx, The Columbia Encyclopedia, 2016
- [12] B. McKernan, *Digital Cinema*, The McGraw-Hill Companies Inc, 2005, https://archive.org/stream/The_Revolution_In_Cinematography_Post_Production_And_Distribution_Brian_Mckernan#page/n15/mode/2up
- [13] N. Gupta, M. Gupta, A. Mishra, *Journey of VCS from Black and White Images to Colored Images with their Performance Analysis*, 2013, <http://research.ijcaonline.org/volume79/number9/pxc3891621.pdf>
- [14] Young-Chang Hou, *Visual Cryptography For Color Images*, Pattern Recognition 36, (2003) 1619-1629, 2003
- [15] C.C. Wu, L.H. Chen, *A Study on Visual Cryptography*, Master's thesis, Institute of Computer And Information Science, National Chiao Tung University, Taiwan, R.O.C., 1998

Santrauka

Kriptografija duomenų saugumo srityje yra viena svarbiausių mokslo šakų. Šiame darbe apžvelgiamos klasikinės kriptografijos schemų naudojimo alternatyvos vaizdiniais duomenimis. Naudojantis vizualiosios kriptografijos technika galime įgyvendinti saugų vaizdinių duomenų perdavimą. Vizualiosios kriptografijos schemas garantuoja duomenų saugumą ir paprastumą naudojantis jomis, dėl nesudėtingo iššifravimo metodo. Šio darbo tekste pateikiama informacija apie klasikinės vizualiosios kriptografijos schemas ir jų naudojimosi pavyzdžius. Be to, apžvelgiama galimybė naudoti vizualiosios kriptografijos schemas vaizdo įrašams bei pristatytas suprogramuotas įrankis pritaikytas vaizdo įrašų vizualiajai kriptografijai.

Summary

Cryptography is one of the most important subject in data security sphere. This work is an overview of possible alternatives of ordinary classic cryptography schemes and algorithms, which will secure visual data. Using visual cryptography techniques is possible to secure visual data sharing without any difficult algorithm. Visual cryptography techniques is perfectly secure and easy to use because of decrypt simplicity. A brief information about classic visual cryptography schemes and examples showing how to use schemes can be found in this work. Furthermore, this work suggests to use visual cryptography schemes for video files and represents implemented tool which was developed for visual cryptography for video.