ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ ԵՎ ԳԻՏՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

Խեմչյան Արտակ Արարատի

ՍԽԱԼՆԵՐ ՈՒՂՂՈՂ ԿՈԴԵՐՈՎ ԳԱՂՏՆԻՔԻ ԲԱՇԽՄԱՆ ՄԻՋՈՑՆԵՐԻ ՄՇԱԿՈՒՄԸ

Տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման ատենախոսություն

Ե.13.05 «Մաթեմատիկական մոդելավորում, թվային մեթոդներ և ծրագրերի համալիրներ»

Գիտական ղեկավար՝ տ.գ.թ., պրոֆ. Գ. Ի. Մարգարով

Երևան-2017

Բովանդակություն

Ներածություն

<u>Աշխատանքի արդիականությունը:</u> Տեղեկատվական տեխնոլոգիաների արագ զարգացումը բերեց նրան, որ այսօր մեծ ծավալի գաղտնի ինֆորմացիա պահպանվում է էլեկտրոնային ձևաչափերով: Դրան զուգահեռ ավելանում է տեղեկատվական համակարգերի վրա հնարավոր հարձակումների հավանականությունը, որոնց նպատակը կարող է լինել ինչպես գաղտնի ինֆորմացիայի բացահայտումը, այնպես էլ նրա վնասումը կամ փոփոխումը: Մյուս կողմից հնարավոր է ինֆորմացիայի կորուստը կապված բնական երևույթների հետ: Բոլոր դեպքերում էլ առաջ է գալիս գաղտնի ինֆորմացիայի անվտանգ պահպանման խնդիրը: Մյուս կողմից կան դեպքեր, երբ գաղտնի ինֆորմացիան ի պահ է տրվում որոշակի խմբի, իսկ խմբի անդամները առանձին վստահություն չեն ներշնչում: Միայն մասնակիցների որոշակի խումբ, որը վստահելի է, պետք է մատչելիություն ունենա գաղտնի ինֆորմացիային: Ավելին, երբ պահվում է շատ կարևոր ինֆորմացիա, ապա ավելի նախընտրելի է բաշխել այն որոշակի մասերի և պահել տարբեր վայրերում: Առանձին բաղադրամասը պետք է որքան հնարավոր է քիչ տեղեկություն պարունակի գաղտնի ինֆորմացիայի վերաբերյալ, և միայն որոշակի բաղադրամասերի միավորումից հետո հնարավոր լինի վերականգնել նախնական գաղտնի ինֆորմացիան: Նկարագրված երկու դեպքում էլ գաղտնիքի բաշխման մեթոդները կարող են օգտակար լինել: Առավել հայտնի են Շամիրի և Բլեկլիի գաղտնիքի բաշխման մեթոդները: Երկու մեթոդներն էլ իրագործում են այսպես կոչված շեմային մուտքի կառուցվածք: Այս մեթոդները ի սկզբանե օգտագործվել են գաղտնագրման բանալիների բաշխման համար: Այսինքն, գաղտնի ինֆորմացիան գաղտնագրվում է ժամանակակից որևէ գաղտնագրման ալգորիթմով և գաղտնագրման բանալին բաշխվում է մասնակիցների միջև: Այս դեպքում լուծվում է գաղտնիության խնդիրը, սակայն չեն լուծվում ամբողջականության և հասանելիության խնդիրները: Եթե գաղտնագրված ինֆորմացիան վնասվի կամ տվյալ պահին հասանելի չլինի, ապա բանալու վերականգնմամբ հնարավոր չէ վերծանել գաղտնի

ինֆորմացիան: Ներկայումս լայն տարածում է գտնում ինֆորմացիայի ամբողջական բաշխման տարբերակը, որն արդեն լուծում է հասանելիության և ամբողջականության խնդիրները: Հաշվի առնելով այն, որ գոյություն ունեցող մեթոդները դանդաղ են և ինֆորմացիայի ծավալը օր-օրի աճում է, խնդիր է առաջանում՝ հետազոտել և մշակել նոր, ավելի արագագործ գաղտնիքի բաշխման շեմային մեթոդ, որը կկարողանա արագ բաշխել և վերականգնել մեծ ծավալի ինֆորմացիա: Հետազոտությունները ցույց են տալիս, որ արագագործ գաղտնիքի բաշխման շեմային մեթոդ հնարավոր է ստանալ՝ օգտագործելով սխալներ ուղղող կոդերը:

Գաղտնիքի բաշխման մեթոդներում կարևոր դեր ունի ոչ միայն բաշխման և վերականգնման արագությունը, այլ նաև բաղադրամասերի անվտանգությունը: Բաղադրամասը պահպանման ընթացքում կարող է վնասվել կամ առհասարակ կորել: Այդ իսկ պատճառով գաղտնիքի բաշխման մեթոդը պետք է օժտված լինի բաղադրամասի ստուգման և անհրաժեշտության դեպքում կորած կամ վնասված բաղադրամասի վերականգնման հնարավորությամբ: Այդ հնարավորությունները մեծացնում են ինֆորմացիայի անվտանգ պահպանման ժամանակը:

<u>Աշխատանքի նպատակը</u> կայանում է սխալներ ուղղող կոդերով գաղտնիքի բաշխման արագ միջոցների մշակումը: Այդ նպատակին հասնելու համար դրվել և լուծվել են հետևյալ խնդիրները՝

- մշակել գաղտնիքի բաշխման արագագործ շեմային մեթոդ, որը հնարավորություն կտա բաշխել և վերականգնել մեծ ծավալի ինֆորմացիա:
- մշակել սխալներ ուղղող կոդերով բաշխման դեպքում բաղադրամասի հավաստիության ստուգման մեթոդ:
- մշակել սխալներ ուղղող կոդերով բաշխման դեպքում վնասված կամ կորած բաղադրամասի արագ վերականգնման մեթոդ:

Գիտական նորույթ:

- Մշակվել է գաղտնիքի բաշխման շեմային մեթոդ՝ հիմնված սխալներ ուղղող կոդերի վրա, որն ի տարբերություն գոյություն ունեցողների ավելի արագ է և հնարավորություն է տալիս կատարել մեծ ծավալի ինֆորմացիայի բաշխում՝ ապահովելով ինֆորմացիայի ինչպես գաղտնիությունը, այնպես էլ ամբողջականությունն ու հասանելիությունը:
- Առաջարկվել է բաղադրամասի հավաստիության ստուգման արագ մեթոդ, որն ի տարբերություն գոյություն ունեցողների, հնարավորություն է տալիս ստուգել բաղադրամասը սխալներ ուղղող կոդերով բաշխման դեպքում և հայտնաբերել կեղծված կամ վնասված բաղադրամասերը :
- Առաջարկվել է վնասված կամ կորած բաղադրամասի արագ վերականգնման մեթոդ, որն ի տարբերություն գոյություն ունեցողների, հնարավորություն է տալիս օգտագործել այն սխալներ ուղղող կոդերով բաշխման դեպքում և մեծացնում է գաղտնիքի ապահով պահպանման ժամանակը:

Աշխատանքի գործնական նշանակությունը:

- Ստացված գիտական արդյունքների հիման վրա մշակվել է գաղտնիքի բաշխման ECC Sharing համակարգը, որը հնարավորություն է տալիս արագ բաշխել և վերականգնել մեծ ծավալի ինֆորմացիա, կատարել բաղադրամասի ստուգում, վերականգնել կորած կամ վնասված բաղադրամասերը, որի շնորհիվ բաշխման գործընթացը արագացել է 42.3%-ով, իսկ վերականգնմանը՝ 11.95%-ով:
- Ուսումնական նպատակների համար մշակված գաղտնիքի բաշխման մեթոդների հետազոտման ECC Sharing Explore համակարգը հնարավություն է տալիս հետազոտել ինչպես կոդերով, այնպես էլ գոյություն ունեցող այլ բաշխման

մեթոդները: Համակարգը հնարավորություն է տալիս համեմատել բաշխման մեթոդները ըստ արագագործության:

Ներդրումներ:

Ստենախոսության շրջանակներում մշակված ECC Sharing ծրագրային միջոցը ներդրվել է «ԱՅ ԷՅ ԷՄ Քլաուդ լիմիթեդ» ընկերության Հայաստանյան մասնաճյուղում գաղտնի ինֆորմացիայի ապահով պահպանման նպատակով: Ծրագրային միջոցի կիրառմամբ ընկերությունում կատարվում է գաղտնի ինֆորմացիայի անվտանգ պահուստավորումը և անհրաժեշտության դեպքում արագ վերականգնումը:

Ատենախոսության շրջանակներում մշակված ECC Sharing Explore ծրագրային միջոցը օգտագործվում է ՀԱՊՀ ՏԱԾԱ ամբիոնում լաբորատոր աշխատանքների անցկացման համար:

Ներդրման տեղեկանքները ներկայացված են հավելված 1-ում:

Աշխատանքի արդյունքները զեկուցվել են. "Հաջորդ սերնդի կիբեռ անվտանգություն" ուսանողական կոնֆերանսում (ԱՊՀ և Ռուսաստան փուլ, 2014թ., ք. Մոսկվա, Ռուսաստան), "Հաջորդ սերնդի կիբեռ անվտանգություն" ուսանողական կոնֆերանսում (եզրափակիչ փուլ, 2014թ., ք. Ստոկհոլմ, Շվեդիա), ՀԱՊՀ տարեկան գիտաժողովում (2015թ., ք. Երևան), "ՆԱՏՕ առաջադեմ հետազոտական աշխատաժողով"-ում (NATO ARW, 2015թ., գ. Աղվերան, ՀՀ), "Քոմփյութերային գիտությունների և տեղեկատվական տեխնոլոգիաների" միջազգային գիտաժողովում (CSIT 2015թ., ք. Երևան), "Գիտության և տեխնոլոգիաների մերձեցում" գիտաժողովում (STC, 2016թ. ք. Երևան), "Միջազգային գիտափորձնական ուսանողների և երիտասարդ գիտնականների գիտաժողով"-ում (Ազգային ավիացիոն համալսարան, 2016թ. ք. Կիև, Ուկրաինա), "XIII միջազգային գիտական և տեխնիկական կոնֆերանսում – նոր ինֆորմացիոն տեխնոլոգիաներ և համակարգեր" (NITaS, 2016թ.

ք. Պենզա, Ռուսաստան), “Ինտերնետ անվտանգության համաշխարհային կոնգրես”-ում (WorldCIS-2016, 2016թ. ք. Լոնդոն, Մեծ Բրիտանիա), ՀԱՊՀ ՏԱԾԱ ամբիոնի գիտատեխնիկական սեմինարներում (2014-2017թ., ք. Երևան):

<u>Հրապարակումներ:</u> Ատենախոսության հիմնական արդյունքները տպագրված են 10 գիտական աշխատություններում [35-44]:

<u>Աշխատանքի կառուցվածքը և ծավալը:</u> Ատենախոսությունը բաղկացած է ներածությունից, չորս գլուխներից, եզրակացությունից և 56 անուն գրականության ցուցակից [35-44]: Աշխատանքի ընդհանուր ծավալն է 107 էջ՝ ներառյալ 29 նկար:

Գլուխ 1:

Գաղտնիքի բաշխման մեթոդների և սխալներ ուղղող կոդերի հետազոտումը

Այս գլխում դիտարկվում են գաղտնիքի բաշխման գաղափարը, նրա առանձնահատկությունները և առավել հայտնի գաղտնիքի բաշխման շեմային մեթոդները: Տրվում են գաղտնիքի բաշխման հիմնական հասկացությունները և սահմանումները: Դիտարկվում են բաշխման մեթոդների կիրառման հիմնական երկու տարբերակները, նրանց առավելություններն ու թերությունները:

Ոսումնասիրվում են սխալներ ուղղող կոդերը: Տրվում են հիմնական սահմանումները, կոդերի առանձնահատկությունները և ատենախոսության ընթացքում օգտագործվող հիմնական սխալներ ուղղող կոդերի կառուցվածքը:

Այս գլխում նաև դիտարկվում են ներկայում հայտնի գաղտնիքի բաշխման մեթոդները, որոնք հիմնված են սխալներ ուղղող կոդերի վրա: Ներկայացվում են նրանց հիմնական թերությունները:

1.1. Գաղտնիքի բաշխման մեթոդների հետազոտումը

Այսօր ինֆորմացիան շատ կարևոր դեր ունի մարդու կենսագործունեության մեջ: Ինֆորմացիայի ծավալն ու շրջանառությունը գլոբալ բաց ցանցերով մեծացնում է նրա խոցելության հավանականությունը: Այդ իսկ պատճառով, գնալով ավելի կարևոր է դառնում ինֆորմացիայի պահպանման կամ փոխանցման ընթացքում նրա անվտանգության ապահովումը: Այդ ամենի համար լայնորեն կիրառվում են գաղտնագրային [1,2] և թաքնագրային [3,4,5,6] պաշտպանության միջոցներ:

Պատկերացնենք, որ մենք ստացել ենք նոր, շատ արդյունավետ մետաղական միացություն, որի արդյունքում ստացվում է գերպինդ մետաղ, որը որակական առումով մի քանի անգամ գերազանցում է գոյություն ունեցող միացություններին: Դա շատ կարևոր է և մենք ցանկանում ենք դրա պատրաստման բաղադրատոմսը (եղանակը) պահել գաղտնի: Միայն վստահելի աշխատողներին կարող ենք հայտնել մետաղի հստակ բաղադրությունը և նրա պատրաստման եղանակը: Սակայն կա մտավախություն, որ նրանցից մեկը կարող է այն հայտնել հակառակորդ կազմակերպությանը, որը տիրանալով գաղտնիքին, կարող է ստանալ նույն արդյունքը:

Այս տեսակ խնդիրների լուծումը ապահովում են այսպես կոչվող գաղտնիքի բաշխման մեթոդները [8,9,11]: Բաշխման արդյունքում ստացված յուրաքանչյուր բաղադրամաս առանձին վերցված ոչինչ չարժե և միայն նրանց միավորումից կարելի է ստանալ գաղտնի ինֆորմացիան ամբողջությամբ: Եթե յուրաքանչյուր աշխատողի մոտ գտնվում է գաղտնիքի մի բաղադրամասը, ապա նրանք միայն հավաքվելով և միավորելով իրենց բաղադրամասերը, կարող են վերականգնել պատրաստման եղանակը և ստանալ այդ մետաղը: Եթե աշխատողներից որևէ մեկը աշխատանքից հեռանա և վերցնի բաղադրամասը իր հետ, ապա նա չի կարող վերականգնել այդ մետաղի պատրաստման բաղադրատոմսը: Ասվածը ավելի առարկայական դարձնելու համար դիտարկենք երկու օրինակ, որոնք վերցված են [2]-ից և [10]-ից: Գաղտնիքի

բաշխման ամենապարզագույն տարբերակը դա գաղտնիքի բաշխումն է երկու կողմերի միջև: Պատկերացնենք մի արձանագրություն, որի օգնությամբ, Տրենտը կարող է բաշխել գաղտնի ինֆորմացիան Ալիսայի և Բոբի միջև [2]:

- Տրենտը գեներացնում է R պատահական բիթային շարք, որի երկարությունը հավասար է M գաղտնի ինֆորմացիայի երկարությանը (ծավալին):
- Տրենտը M-ի և R-ի հետ կատարում է «բացառող կամ» (XOR) գործողությունը և ստանում S-ը՝

$$\mathrm{R} \oplus \mathrm{M} = \mathrm{S}$$

- Տրենտը R-ը փոխանցում է Ալիսային, իսկ S-ը Բոբին: Գաղտնիքի վերականգնման համար Ալիսային և Բոբին մնում է կատարել ընդամենը մեկ գործողություն:
- Ալիսան և Բոբը օգտագործելով «բացառող կամ» գործողությունը, ստանում են գաղտնի ինֆորմացիան՝

$$\mathrm{R} \oplus \mathrm{S} = \mathrm{M}$$

Այս մեթոդի ճիշտ կիրառումն ապահովում է կատարյալ անվտանգություն [2]: Ինչպես տեսանք, բաղադրամասը առանձին վերցրած իր մեջ իմաստ չի պարունակում: Եթե փորձենք այլ կերպ ասել, ապա Տրենտը ինֆորմացիան գաղտնագրում է միանգամյա նոթատետրով [2] և մի կողմին տրամադրում է գաղտնագրված տեքստը, իսկ մյուսին՝ նոթատետրը: Անգամ հատարկման եղանակով հնարավոր չէ վերականգնել գաղտնիքը, եթե նույնիսկ հայտնի է բաղադրամասերից մեկը:

Նկարագրված մեթոդը կարելի է օգտագործել գաղտնիքը երկուսից ավել կողմերի միջև բաշխելու համար: Այդ նպատակի համար անհրաժեշտ է XOR գործողությունը կատարել ավելի շատ պատահական բիթային հաջորդականությունների հետ: Օրինակ, դիտարկենք ինչպես է Տրենտը բաշխում գաղտնի ինֆորմացիան չորս կողմի միջև.

- Տրենտը գեներացնում է պատահական բիթային շարք՝ R, S և T: Այդ բիթային շարքերի և M-ի երկարությունները նույնն են:
- Տրենտը կատարում է «բացառող կամ» գործողություն գաղտնի ինֆորմացիայի և գեներացված երեք բիթային շարքերի հետ.

$$\mathrm{M} \oplus \mathrm{R} \oplus \mathrm{S} \oplus \mathrm{T} = \mathrm{U}$$

- Տրենտը Ալիսային տալիս է R-ը, Բոբին՝ S-ը, Կերոլին՝ T-ն և Դեյվին՝ U-ն:
- Ալիսան, Բոբը, Կերոլը և Դեյվը միավորվում են և հաշվարկում գաղտնի ինֆորմացիան՝

$$\mathrm{R} \oplus \mathrm{S} \oplus \mathrm{T} \oplus \mathrm{U} = \mathrm{M}$$

Այսպիսի բաշխման սխեմայում Տրենտը կարող է անել ամեն ինչ: Նա կարող է բաշխել կամայական ինֆորմացիա: Այդ ինֆորմացիան կարող է և ոչ մի արժեք չունենալ և կողմերը չեն կարող այն ստուգել, քանի դեռ չեն միավորվել և փորձել վերականգնել այն: Սակայն այս դեպքում դա խնդիր չէ, քանզի այդ գաղտնիքը պատկանում է հենց Տրենտին:

Այնուամենայնիվ այս մեթոդն ունի մեկ թերություն: Եթե գաղտնիքի բաղադրամասերից մեկը կորի և Տրենտը ներկա չլինի այդ ժամանակ, ապա մասնակիցները չեն կարող վերականգնել գաղտնիքը և այն կկորի: Օրինակ, եթե Կերոլը տեղափոխվի այլ աշխատանքի (օրինակ՝ մրցակից կազմակերպություն), ապա մյուս մասնակիցները կկորցնեն գաղտնիքը: Կերոլը չի կարող վերականգնել գաղտնիքը, սակայն դա չեն կարող անել նաև Ալիսան, Բոբը և Դեյվը՝ միավորելով իրենց մոտ եղած բաղադրամասերը: Կերոլի բաղադրամասը նույնքան կարևոր է վերականգնման համար, որքան մյուս բաղադրամասերը: Ալիսային, Բոբին և Դեյվին հայտնի է ընդամենը մեկ բան, դա ինֆորմացիայի երկարությունն է (ծավալը): Դա բնական է, քանզի R, S, T, U բաղադրամասերի և M ինֆորմացիայի ծավալները հավասար են և հետևաբար մասնակիցները գիտեն այն: Կարևոր է նշել, որ M

հաղորդագրությունը չի բաժանվում մասերի, այլ ենթարկվում է «բացառող կամ» (XOR) գործողության պատահական բիթային հաջորդականության հետ [2]:

Գաղտնիքի բաշխման մեթոդների նշանակությունն ավելի ակնառու դարձնելու համար նկարագրենք մեկ այլ օրինակ, որտեղ հարկավոր է գաղտնի ինֆորմացիայի ընդհանուր օգտագործում (տիրապետում):

Ենթադրենք մենք նախագծում ենք միջուկային հրթիռի գործարկման համակարգ և ցանկանում ենք վստահ լինել, որ ոչ մի սպա միայնակ չի կարող գործարկել այն: Մենք նաև ցանկանում ենք վստահ լինել, որ երկուսով նույպես չեն կարող գործարկել այն: Ցանկանում ենք, որ գործարկումը հնարավոր լինի միայն այն դեպքում, երբ նվազագույնը երեք սպա (հինգ սպաներից) ներկա լինեն, նոր հնարավոր լինի գործարկել այդ հրթիռը:

Այս տեսակի խնդիրները հնարավոր է լուծել հետևյալ եղանակով: Հարկավոր է ստեղծել հրթիռի գործարկման կառավարման մեխանիկական սարք և հինգ սպաներից յուրաքանչյուրին տրամադրել մեկական բանալի: Գործարկման ժամանակ ստուգել, որ նվազագույնը երեք սպա տեղադրեն իրենց բանալիները համապատասխան տեղերում և նոր թույլատրել գործարկել հրթիռը: Բանալիների համար նախատեսված տեղերը հարկավոր է տեղակայել միմյանցից հնարավորինս հեռու, որպեսզի նրանք բանալիները տեղադրեն միաժամանակ և հնարավոր չլինի գողացված բանալին միաժամանակ տեղադրել և գործարկել հրթիռը:

Անվտանգության բարձրացման համար կարելի է ավելի բարդացնել թողարկման մեխանիզմը: Պայման դնենք, որ միայն գեներալին և որոշ սպաների է թույլատրվում գործարկել հրթիռը: Իսկ եթե գեներալը ներկա չէ գործարկման ժամանակ, ապա այն կարող է իրականացվել միայն այն դեպքում, երբ ներկա են հինգ սպաները: Այս խնդրի լուծման համար կարելի է ստեղծել հինգ բանալիով թողարկվող համակարգ, գեներալին տալ երեք բանալի, իսկ սպաներին մեկական բանալի: Գեներալը և

ցանկացած երկու սպա ունենում են հինգ բանալի և կարող են թողարկել հրթիռը: Հինգ բանալի կլինի նաև այն դեպքում, երբ միավորվեն բոլոր հինգ սպաները:

Գաղտնիքի բաշխման խնդիրների հետազոտման հաջորդ փուլը նվիրված է եղել այնպիսի բաշխմանը, որի դեպքում M մասնակիցներից կամայական N-ը կարող են վերականգնել գաղտնիքը: Այդ տեսակ բաշխման մեթոդները կոչվում են շեմային (*threshold schemes*) և ընդունակ են լուծել ինչպես նկարագրված խնդիրները, այնպես էլ ավելի բարդ մաթեմատիկական խնդիրներ: Այդ մեթոդների օգնությամբ կարելի է գաղտնի ինֆորմացիան (գաղտնի բաղադրատոմս, թողարկման գաղտնաբառ և այլն) բաշխել n մասի այնպես, որ կամայական $m\,(m \leq n)$ մասով հնարավոր լինի վերականգնել այն: Գրականության մեջ ընդունված է այդ մեթոդն անվանել (m,n) շեմային սխեմա կամ (m,n) շեմային մեթոդ:

Օրինակ նախորդ խնդրում Տրենտը կարող է օգտագործել (3,4) շեմային մեթոդ և բաշխել գաղտնիքը Ալիսայի, Բոբի, Կերոլի և Դեյվի միջև այնպես, որ նրանցից ցանկացած երեքը կարող են վերականգնել այն: Եթե Բոբը աշխատանքից հեռանա և իր հետ տանի իր բաղադրամասը, ապա գաղտնի ինֆորմացիան հնարավոր է վերականգնել Ալիսայի, Կերոլի և Դեյվի բաղադրամասերից: Միևնույն ժամանակ, եթե Բոբը աշխատանքից հեռացել է, իսկ Կերոլը արձակուրդում է, ապա Ալիսան և Դեյվը չեն կարող վերականգնել գաղտնիքը:

Առհասարակ, շեմային մեթոդները շատ ավելի ճկուն են և կարող են լուծել ավելի բարդ խնդիրներ: Օրինակ, հնարավոր է գաղտնիքը բաշխել շենքի բնակիչների միջև այնպես, որ եթե նրա վերականգնման համար ոչ ոք երրորդ հարկից ներկա չէ, ապա գաղտնիքի վերականգնման համար կպահանջվի յոթ հոքի առաջին հարկից և հինգ հոքի երկրորդ հարկից, հակառակ դեպքում բավարար է մեկ ներկայացուցիչ երրորդ հարկից, երեք մասնակից առաջին հարկից և երկու մասնակից երկրորդ հարկից: Եթե ներկա է որևէ մեկը չորրորդ հարկից, ապա վերականգնման համար բավարար է ևս

մեկ մասնակից երրորդ հարկից կամ այդ մասնակիցը և երկու մասնակից առաջին հարկից և ևս մեկը երկրորդից: Առաջին հայացքից այս բարդ և «խճճված» խնդիրը կարող է ունենալ լուրջ տրամաբանություն և ապահովել անվտանգության քաղաքականության իրագործումը: Կարևորն այն է, որ նույնիսկ այս տեսակ բարդ խնդիրները հնարավոր է մոդելավորել և լուծել շեմային մեթոդների օգնությամբ:

Շեմային մեթոդներ առաջին անգամ իրարից անկախ առաջարկել են Ադի Շամիրը (*Adi Shamir*) [8] և Ջորջ Բլեկլին (*George Blakley*) [9]: Նրանց հետազոտությունները ակտիվորեն շարունակել է Գուս Սիմմոնսը (*Gus Simmons*) [11].

Առաջին հայացքից այս մեթոդները ապահովում են բարձր մակարդակի անվտանգություն և լիովին լուծում են իրենց առաջ դրված խնդիրները: Սակայն դա միայն առաջին հայացքից: Պատկերացնենք, որ սպաներ Ալիսան, Բոբը և Կերոլը գտնվում են մեկուսացված տարացքում և նախագահի կողմից ստանում են գաղտնագրված հաղորդագրություն այն մասին, որ անհրաժեշտ է գործարկել հրթիռը: Ալիսան և Բոբը համակարգ են մուտքագրում իրենց բաղադրամասերը, իսկ Կերոլը միտումնավոր ներմուծում է սխալ տվյալներ (ինչ-ինչ պատճառներով նա չի ցանկանում, որ հրթիռը գործարկվի): Քանզի Կերոլը չի ներմուծել ճիշտ բաղադրամասը, ապա այն գաղտնի ինֆորմացիան, որը պետք է վերականգնվեր այդ բաղադրամասերից, ճիշտ չի վերականգնվում և հրթիռի գործարկումը տեղի չի ունենում: Հրթիռները մնում են իրենց տեղում: Այստեղ ամենավատն այն է, որ ոչ ոք չի իմանում պատճառը: Ալիսան և Բոբը չեն կարող ցույց տալ, որ այդ ամենի մեղավորը Կերոլն է:

Այս տեսակ բարդությունների հաղթահարման համար հարկավոր են լրացուցիչ միջոցներ, որով հնարավոր կլինի ստուգել բաղադրամասի իսկությունը: Հարկավոր է որև մեթոդ, որը հնարավություն կտա ստուգել արդյո՞ք այդ բաղադրամասը իրենից ներկայացնում է այդ գաղտնիքի բաղադրամաս, թե ոչ: Ունենալով այսպիսի մեթոդ Ալիսան ու Բոբը կարող էին ապացուցել, որ մեղավորը Կերոլն է:

Բաշխված գաղտնիքի համակարգերում միայն բաղադրամասի ստուգման հնարավորությունը բավարար չէ: Պատկերացնենք իրավիճակ, երբ գաղտնիքի պահպանման ընթացքում որևէ բաղադրամաս վնասվել է կամ կորել: Այդպիսի դեպքերի համար նպատակահարմար է ունենալ բաղադրամասի ստուգման և անհրաժեշտության դեպքում նրա վերականգնման մեթոդ, որը հնարավորություն կտա վերականգնել այդ բաղադրամասը: Այդպիսի մեթոդի բացակայության դեպքում հարկավոր է լինելու գաղտնիքը վերականգնել և կրկին բաշխել կողմերի միջև: Հակառակ դեպքում, որոշ ժամանակ անց կարող է պարզվել, որ գաղտնիքի վերականգնման համար անհրաժետ նվազագույն քանակով չվնասված բաղադրամասեր առկա չեն, իսկ դա իր հերթին նշանակում է, որ բաշխված գաղտնիքը կորել է և վերականգնել այլևս հնարավոր չէ:

1.2. Գաղտնիքի բաշխման շեմային մեթոդների աշխատանքի սկզբունքները

Գաղտնագրության ոլորտում գաղտնիքի բաշխում ասելով հասկանում են գաղտնիքի ցանկացած ձևով բաշխումը մասնակիցների միջև այնպես, որ նրանց որոշակի կոալիցիան կարող է վերականգնել այն: Առանձին բաղադրամասը ինքն իրենով ոչինչ չի բացահայտում գաղտնիքի վերաբերյալ: Այս տեսակ բաշխման ալգորիթմները նաև անվանում են գաղտնիքի բաշխման սխեմաներ կամ գաղտնիքի բաշխման մեթոդներ:

Գաղտնիքի բաշխման մեթոդում մասնակցում են գաղտնիքը բաշխողը (գրականության մեջ նաև օգտագործվում է դիլեր անվանումը) և n մասնակիցներ: Բաշխողը սովորաբար մասնակիցների շարքից չէ (հաճախ որպես բաշխող հանդես է գալիս ծրագիրը կամ սարքը): Նրա հիմնական խնդիրն է ինֆորմացիայի ընդունումը (կամ գեներացումը) և նրա բաշխումը բաղադրամասերի: Սկզբում գաղտնիքը գտնվում է բաշխողի մոտ, որից հետո նա հաշվարկում է բաղադրամասերը և տրամադրում մասնակիցներին: Պարզագույն դեպքում յուրաքանչյուր մասնակից ստանում է մեկական բաղադրամաս: Ի սկզբանե բաշխողը սահմանում է վերականգնման t շեմը: Եթե մասնակիցների թիվը t կամ ավել է, ապա նրանք կարող են ճշգրիտ վերականգնել գաղտնիքը, իսկ t-ից պակաս մասնակիցները ոչինչ չեն կարող իմանալ գաղտնի ինֆորմացիայի վերաբերյալ:

Գոյություն ունեն գաղտնիքի բաշխման մեթոդների կիրառման մի քանի հիմնական ուղղություններ.

1 Գաղտնագրման մեջ՝ գաղտնիքի բաշխման ալգորիթմների կիրառում:
2 Բանալիների վավերականացում (*key verification*) - գաղտնիքը պահվում է համակարգում: Եթե բաղադրամասերից վերականգնվածը

համապատասխանում է համակարգում պահվածի հետ, ապա թույլատրվում է դիմել պաշտպանված ինֆորմացիային:

3 Ինֆորմացիայի հուսալի թաքնագրում:

Ընդհանրացնելով հետազոտությունները կարող ենք ասել, որ գաղտնիքի բաշխման մեթոդների ստեղծման հիմնական նպատակներն են՝

1. ինֆորմացիայի (գաղտնի բանալու և այլն) ապահովագրում կորստից,
2. որոշումներ կայացնելու պարագայում պատասխանատվության բաշխում,
3. մարդկային գործոնի (նենգադուլ, գաղտնի ինֆորմացիան տիրապետող մարդկանց առևանգում, կաշառում) հետ կապված գրոհների կանխարգելում:

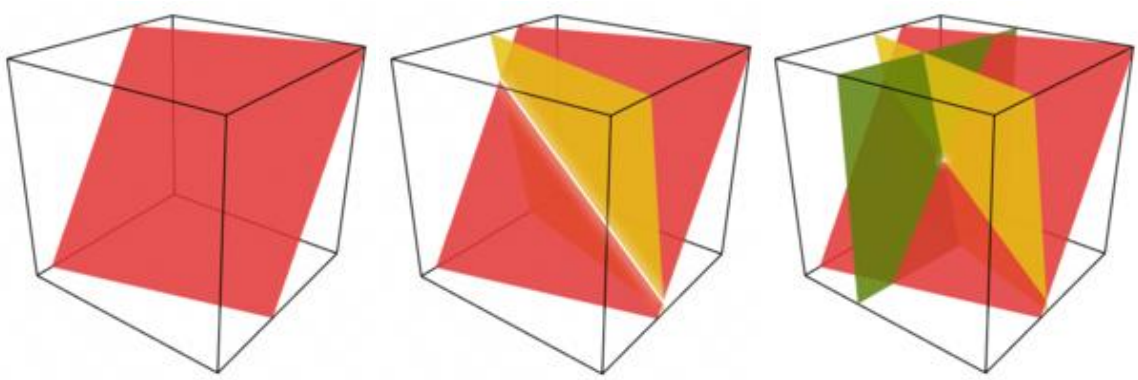
Գաղտնիքի բաշխման կիրառման օրինակ ներկայացված է [7]-ում:

Հաջորդիվ կնկարագրենք ներկայում ամենատարածված գաղտննիքի բաշխման երկու մեթոդները: Դրանք են Շամիրի [8] և Բլեկլիի [9] շեմային մեթոդները:

Բլեկլիի մեթոդը հիմված է այն գաղափարի վրա, որ հարթության մեջ երկու ոչ զուգահեռ գծեր հատվում են մի կետում, իսկ երկու ոչ զուգահեռ հարթություններ հատվում են մեկ ուղիղով, իսկ երեք ոչ զուգահեռ հարթություններ հատվում են մեկ կետում [9]: Ընդհանրացնելով այս գաղափարը կարող ենք ասել, որ n չափանի տարածությունում n հատ ոչ զուգահեռ գերհարթություններ միշտ հատվում են մեկ կետում: Եթե որպես գաղտնիք վերցնենք այդ հարթությունների հատման կետի կոորդինատները, իսկ որպես բաղադրամաս հանդես գա այդ կետով անցնող գերհարթությունը, ապա ակնհայտ է, որ ունենալով որոշակի քանակի գերհարթություններ կարելի է ստանալ նրանց հատման կետը և դրանով վերականգնել գաղտնիքը: Նկ. 1-ում պատկերված է Բլեկլիի մեթոդը երեք չափանի տարածության համար: Նկարից երևում է, որ ունենալով մեկ հարթություն հնարավոր չէ ստանալ գաղտնիքը: Երկու հարթության դեպքում հնարավոր է ստանալ այն ուղիղը, որի վրա գտնվում է այդ կետը, սակայն հստակ իմանալ կետը հնարավոր չէ (նրանց քանակը

անթիվ է): Միայն երեք հարթության միավորումից է միարժեքորեն որոշվում գաղտնի կետի կոորդինատները:

Բլեկլիի մեթոդի օգնությամբ հնարավոր է ստանալ (t,n) շեմային մեթոդ t-ի և n-ի ցանկացած արժեքի դեպքում: Դրա համար անհրաժեշտ է ունենալ t չափանի տարածություն և n մասնակիցներից յուրաքանչյուրին տրամադրել գաղտնի կետով անցնող մեկ գերհարթություն: Այսպիսի բաշխման դեպքում ցանկացած t հատ գերհարթություն միանշանակ կհատվեն այն կետում, որի կոորդինատները գաղտնիքն է:



Նկ. 1. Բլեկլիի բաշխման մեթոդը եռաչափ տարածության համար

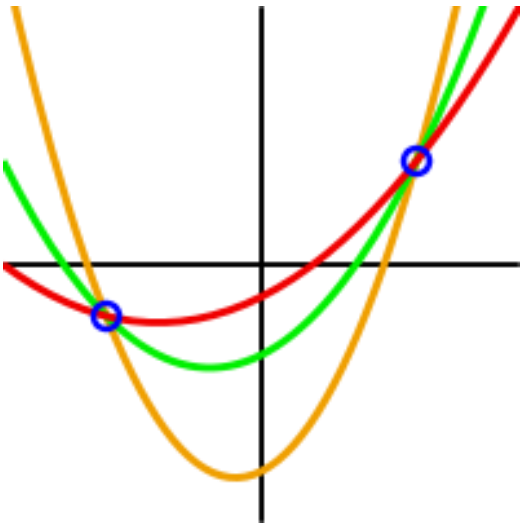
Բլեկլիի մեթոդի հիմնական թերությունը բաղադրամասերի ծավալն է: Բաղադրամասի ծավալը t անգամ գերազանցում է գաղտնիքի ծավալը: Այդ թերությունից զուրկ է Շամիրի շեմային մեթոդը [8]:

Այժմ դիտարկենք գաղտնիքի բաշխման Շամիրի շեմային մեթոդը:

Դիցուք տրված են t և w ամբողջ թվերը և $t \leq w$: (t,w) շեմային մեթոդ է կոչվում K ինֆորմացիայի բաշխումը w մասնակիցների միջև այնպես, որ ցանկացած t մասնակից կարող է վերականգնել K-ն, իսկ $t-1$ մասնակիցների ցանկացած խումբ չկարողանա կատարել դա:

Շամիրի առաջարկած մեթոդի հիմքում ընկած է այն գաղափարը, որ երկու կետը բավարար է ուղիղը որոշելու համար, երեք կետը բավարար է պարաբոլը որոշելու համար, չորս կետը քառակուսային պարաբոլը և այդպես շարունակ: Համապատասխանաբար n չափանի բազմանդամի կորը միարժեքորեն որոշվում է $n+1$ հատ կետով: Դիցուք մենք ցանկանում ենք ստանալ (k,n) $(k \le n)$ շեմային կառուցվածք S գաղտնիքի բաշխման համար: Դրա համար հարկավոր է վերցնել $k-1$ հատ կամայական $a_1, a_2 \dots, a_{k-1}$ գործակիցներ և $a_0 = S$: Դրանից հետո ձևավորում ենք $f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$ բազմանդամը: Որպես գաղտնիքի բաղադրամաս հանդես կգան n հատ կետի կոորդինատներ՝ $(i, f(i))$, որտեղ $i = 1, \dots, n$: Ունենալով k հատ բաղադրամաս (k հատ կետ որոնք գտնվում են այդ կորի վրա) կարող ենք հաշվարկել բազմանդամի գործակիցները, որոնց թվում է նաև գաղտնիքը՝ $S = a_0$: Հաշվարկի համար կարելի է օգտագործել Լագրանժի միջարկող բազմանդամը:

Այս բաշխման անվտանգության հիմքում ընկած է այն գաղափարը, որ օրինակ երկու կետով կարելի է անցկացնել երկրորդ աստիճանի անթիվ կորեր: Նրանցից մեկը միանշանակորեն կարելի է ստանալ միայն այն դեպքում, երբ կհաստատենք ևս մեկ կետ: Նկ. 2-ը ցույց է տալիս, որ երկու կետով անցնում է մեկից ավել պարաբոլներ:



Նկ. 2. Երկրորդ աստիճանի բազմանդամի օրինակ

Շամիրի մեթոդում նաև օգտագործվում է $GF(q)$ վերջավոր դաշտ, որի դեպքում բազմանդամի գրաֆիկը դժվար է պատկերել [8] :

Շամիրի օրինակի վրա դիտարկենք շեմային մեթոդների որոշ հատկություններ և սահմանումներ.

- Կատարյալ անվտանգություն (*Perfect Security*) – ինֆորմացիայի տեսական անվտանգություն: Գաղտնի բանալու t բաղադրամասերի առկայության դեպքում եզակիորեն հնարավոր է վերականգնել գաղնտիքը: Գիտենալով գաղտնի բանալու *t-1* կամ քիչ քանակությամբ բաղադրամասեր և գաղտնի բանալին վերականգնելու միջակայքը, այն դեռևս մնում է բարդ խնդիր: Գաղտնի բանալին կարող է լինել m բազմությունից ցանկացածը ($0 \leq S \leq m-1$):
- Իդեալական (*Ideal*) – գաղտնի բանալու բիթային չափը համընկնում է բանալու յուրաքնչյուր բաղադրամասի չափի հետ (կամ տարբերությունը շատ չնչին է):
- Ընդլայնելիությունը (*Extendable*) – չօգտագործելով արդեն ստեղծված բաղադրամասերը, գաղտնի բանալու նոր բաղադրամասերը (նոր ի հայտ եկած մասնակիցների համար) կարող են հաշվարկվել և բաշխվել, բազմանդամի համար լրացուցիչ կետեր հաշվարկելու մեթոդով:
- Ճկունություն (*Flexible*) – հնարավոր է գաղտնիքը բաշխել ըստ կողմերի կարևորության աստիճանի, այսինքն տարբեր թույլատրելի ենթաբազմությունների բաշխել տարբեր «քաշեր»:
- Հոմոմորֆիզմի հատկությունը (*Homomorphic property*) – Շամիրի մեթոդի համար հոմոմորֆիզմն ունի (+,+) տեղ: Օրինակ, ենթադրենք ունենք S և R գաղտնի բանալիներ: Դրանք բաշխված են Շամիրի գաղտնիքի բաշխման մեթոդով՝ $(f(1),\dots,f(n))$, $f(X)$ $(g(1),\dots,g(n))$ բազմանդամից որոշիչներ $g(X)$ համապատասխանաբար S և R համար: Ենթադրենք յուրաքանչյուր i-րդ մասնակից կգումարի՝ $h(i)=f(i)+g(i)$ $(i=[1..n])$: Ստացված յուրաքանչյուր

գումար իր հերթին հանդիսանում է $S+R$ գաղտնի բանալու բաղադրամաս, որոշված *h(X)=f(X)+g(X)*, և $h(0)=S+R$ բազմանդամից:

- Թվաբանական հաշվարկների համար արդյունավետ բաշխման մեխանիզմ (*Efficient Distributed Mechanism For Arithmetic Calculations*) – Օրինակ, հաստատունով բազմապատկում՝ յուրաքանչյուր մասնակից կարող է գաղտնի բանալու իր բաղադրամասը բազմապատկել հաստատունով:
- Անկախություն (*Independent*) – ի տարբերություն բազմաթիվ գաղտնագրային համակարգերի, գաղտնի բանալու բաշխման համակարգի անվտանգությունը ուղղակիորեն կախված չէ բանալու բարդությունից:
- Դինամիկություն (*Dynamic*) - չփոփոխելով գաղտնիքը՝ հեշտությամբ կարելի է բարձրացնել անվտանգությունը, ժամանակ առ ժամանակ փոփոխելով բազմանդամը (նույնը պահելով ազատ անդամը) և մասնակիցների համար ստանալով նոր բաղադրամասեր:

Այս մեթոդն ունի հետևյալ թերությունները.

- Բաշխող (*Trusted dealer*) – Բաշխողը համարվում է վստահված կողմ և համակարգի անվտանգությունը հիմնվում է նաև այդ փաստի վրա: Սակայն միշտ չէ, որ բաշխողին կարելի է վստահել:
- Բացակայում է բաղադրամասի իսկության ստուգման հնարավորությունը (*Verify correctness*): Մինչ գաղտնիքի վերականգնման գործողությունը մասնակիցները չեն կարող վստահաբար պնդել, որ իրենց բաղադրամասը ճիշտ է:

Այդ իսկ պատճառով, գոյություն ունեցող մեթոդներում, ինչպես նաև նոր մշակվողներում կարևոր է մեծ ուշադրություն դարձնել այդ հարցին: Հարկավոր է նախատեսել բաղադրամասերի իսկության ստուգման մեթոդներ:

Ինչպես և բազում ալգորիթմների համար, այնպես էլ գաղտնիքի բաշխման շեմային մեթոդների համար կարևոր է նրա ալգորիթմական բարդության գնահատականը:

- Լագրանժի միջարկման հիման վրա ստացված հավասարումներից, *f(x)* ֆունկցիայի համար բոլոր գործակիցների հաշվարկման համար միջին հաշվով պահանջվում է $O(k^2)$ քայլ (ժամանակային բարդույթ): Սակայն մատրիցների հետ աշխատելու դեպքում թվաբանական հաշվարկների օպտիմալացումը թույլ է տալիս քչացնել քայլերի թիվը մինչև՝ $k \cdot log^2 k$:
- Որոշ դեպքերում հարմար է S "երկար գաղտնի ինֆորմացիան" ոչ թե բաշխել բաղադրամասերի, այլ սկզբում այն բաժանել j ավելի փոքր մասերի և անմիջականորեն աշխատել այդ փոքր մասերի հետ առանձին: Դա կնվազեցնի ժամանակային բարդույթը $O(k^2)$ –ից մինչև՝ $O(j(k/j)^2) = O(k^2/j)$:
- Գաղտնի բանալու բաղադրամասերի բիթային երկարությունը պետք է հավասարեցվի գաղտնի բանալու երկարությանը:

Այժմ դիտարկենք շեմային մեթոդների վրա հնարավոր գրոհների տեսակները: Եթե հակառակորդին հայտնի է մասնակիցների քանակը, ապա նա կարող է փորձել կատարել հետևյալ գրոհները.

- Հակառակորդը կարող է հատուկ օգտագործել սխալ բաղադրամաս (կամայական թիվ): Խումբը կփորձի վերականգնել գաղտնիքը, չի ստանա այն և չի կարող հասկանալ թե որ բաղադրամասն է սխալ:
- Հակառակորդը կարող է հրահրել գաղտնիքի վերականգնման գործընթաց և այդ ընթացքում փորձել ստանալ մյուսների բաղադրամասերը: Օրինակ (3,4) շեմային բաշխման դեպքում, հակառակորդը կարող է ներկայանալ որպես չորրորդ մասնակից և քանի որ երեք բաղադրամասը արդեն իսկ բավարար է

գաղտնիքի վերականգնման համար, փորձի ստանալ մյուս երեք բաղադրամասերը և ինքնուրույն վերականգնել գաղտնիքը:

Գաղտնիքի բաշխման մեթոդներում հնարավոր է նաև բաղադրամասի կորուստ կամ վնասում (պատահական կամ նպատակաուղված): Այդպիսի իրավիճակներում հարկավոր է ձեռնարկել համապատասխան միջոցներ գաղտնիքի բաղադրամասը վերականգնելու համար: Բաղադրամասը չվերականգնելու դեպքում խախտվում է բաշխման համակարգի վրա դրված տրամաբանությունը և այն լիարժեք չի կատարում այն խնդիրը, որի համար ներդրվել է: Բացի այդ, հնարավոր է որ ժամանակի ընթացքում վնասվեն կամ կորեն այնքան բաղադրամասեր, որ գաղտնիքի վերականգնումը դառնա անհնար: Այդ իսկ պատճառով հարկավոր է նախատեսել վերականգնման հնարավորությունը: Պետք է պարբերաբար ստուգել բաղադրամասերի իսկությունը և հասանելիությունը: Կորած կամ վնասված բաղադրամաս հայտնաբերելու դեպքում անհրաժեշտ է վերականգնել այն:

Բաղադրամասի վերականգնման համար գոյություն ունի երեք մոտեցում

- Գաղտնիքի ամբողջական վերականգնում և կրկին բաշխում:
- Բաղադրամասի վերականգնում՝ առանց գաղտնիքի բացահայտման:
- Գաղտնիքի բացահայտմամբ, բայց ավելի արագ (հեշտ) վերականգնման մեթոդ, որն ավելի արդյունավետ է, քան գաղտնիքի ամբողջական վերականգնումը և կրկին բաշխումը:

Նոր մշակվող մեթոդների համար նպատակահարմար է մշակել երկրորդ կամ երրորդ կետին համապատասխան բաղադրամասի վերականգնման մեթոդ:

1.3. Գաղտնիքի բաշխման մեթոդների կիրառման տարբերակները

Նախորդ բաժնում նշվեց, որ առավել տարածված է գաղտնիքի բաշխման Շամիրի շեմային մեթոդը, որի հիմքում ընկած է միջարկման գաղափարը [8]: Մասնավորապես, այդ մեթոդում բաղադրամասերը հաշվարկվում են ստորև տրված բանաձևերով:

$$k_1 = F(1) = (a_{k-1} \cdot 1^{k-1} + a_{k-2} \cdot 1^{k-2} + \ldots + a_1 \cdot 1 + M) \bmod p$$

$$k_2 = F(2) = (a_{k-1} \cdot 2^{k-1} + a_{k-2} \cdot 2^{k-2} + \ldots + a_1 \cdot 2 + M) \bmod p$$

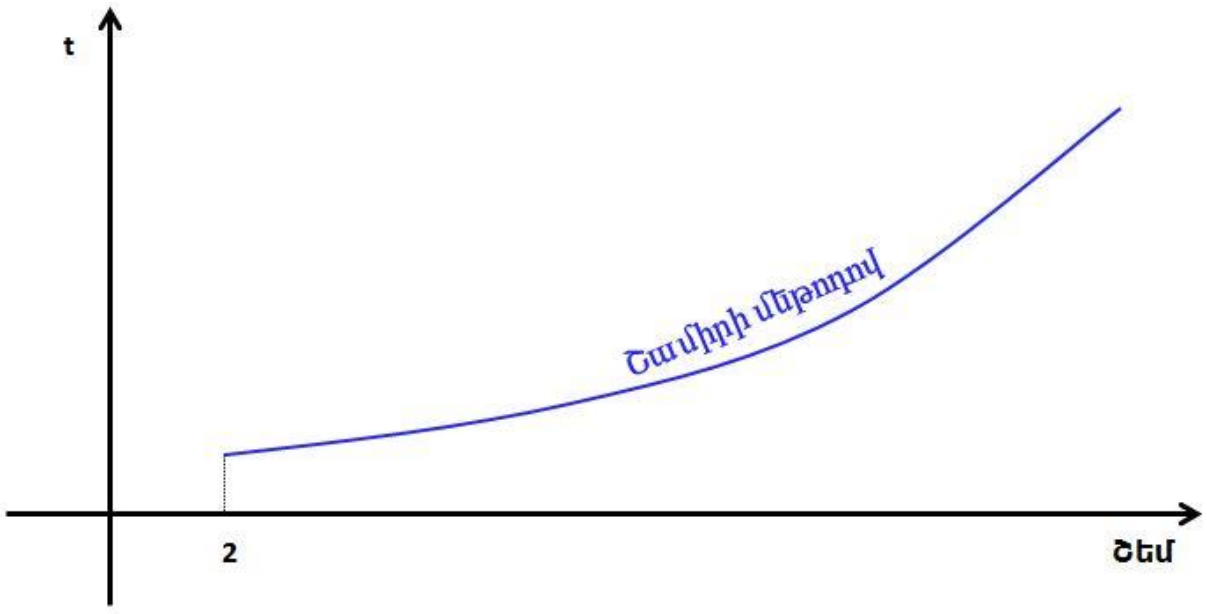
$$\ldots$$

$$k_i = F(i) = (a_{k-1} \cdot i^{k-1} + a_{k-2} \cdot i^{k-2} + \ldots + a_1 \cdot 1 + M)\ mod\ p$$

$$\ldots$$

$$k_n = F(n) = (a_{k-1} \cdot n^{k-1} + a_{k-2} \cdot n^{k-2} + \ldots + a_1 \cdot 1 + M)\ mod\ p$$

Ակնհայտ է, որ շեմի մեծացմանը զուգահեռ Շամիրի մեթոդի արագագործությունը նվազում է: Դա վկայում է նաև Շամիրի մեթոդի արագագործությունը արտահայտող գրաֆիկը, որը պատկերված է նկ. 3-ում:

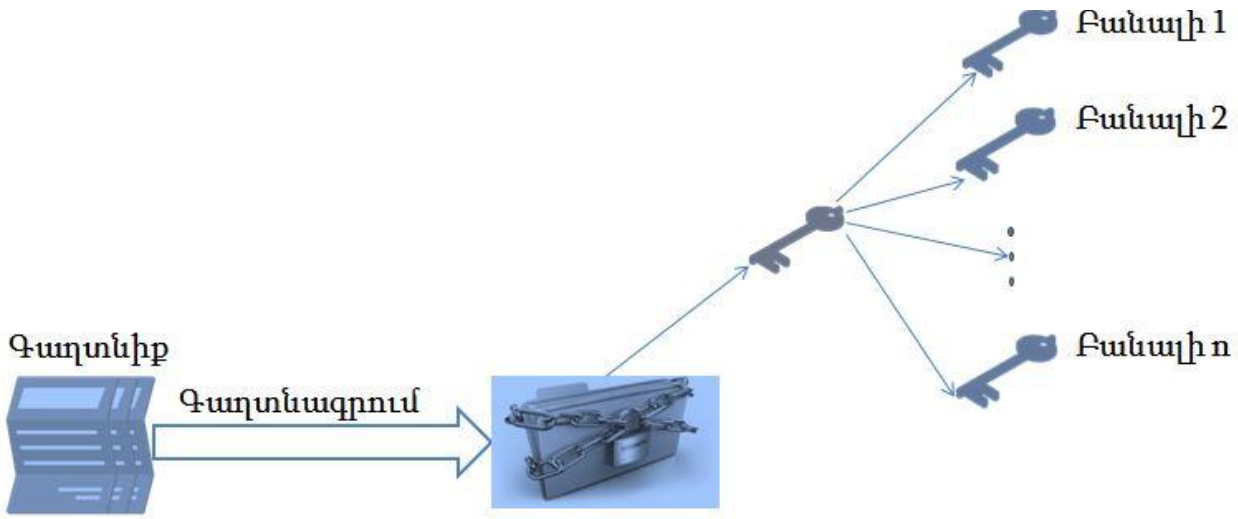


Նկ. 3. Շամիրի մեթոդում բաշխման ժամանակի՝ շեմի արժեքից կախվածության գրաֆիկը

Գրաֆիկում պատկերված է Շամիրի մեթոդով ինֆորմացիայի բաշխման ժամանակի կախվածությունը շեմի արժեքից: Փոքր ծավալի գաղտնի ինֆորմացիայի բաշխման ժամանակ արագագործության խնդիր չի առաջանում: Օրինակ, գաղտնագրման բանալիների և գաղտնաբառերի բաշխման համար Շամիրի մեթոդը լիովին բավարարում է և արագագործության և անվտանգության տեսանկյունից: Մյուս կողմից հասկանալի է, որ Շամիրի մեթոդով մեծ ծավալի գաղտնի ինֆորմացիայի բաշխումը երկար ժամանակ կպահանջի: Այս ամենի հետևանքով մեծ ծավալի ինֆորմացիայի բաշխման համար Շամիրի մեթոդը կիրառվում է հետևյալ տարբերակով.

1. գաղտնի ինֆորմացիան գաղտնագրվում է որևէ ժամանակակից համաչափ գաղտնագրային ալգորիթմով,
2. գաղտնագրման բանալին բաշխվում է մասնակիցների միջև:

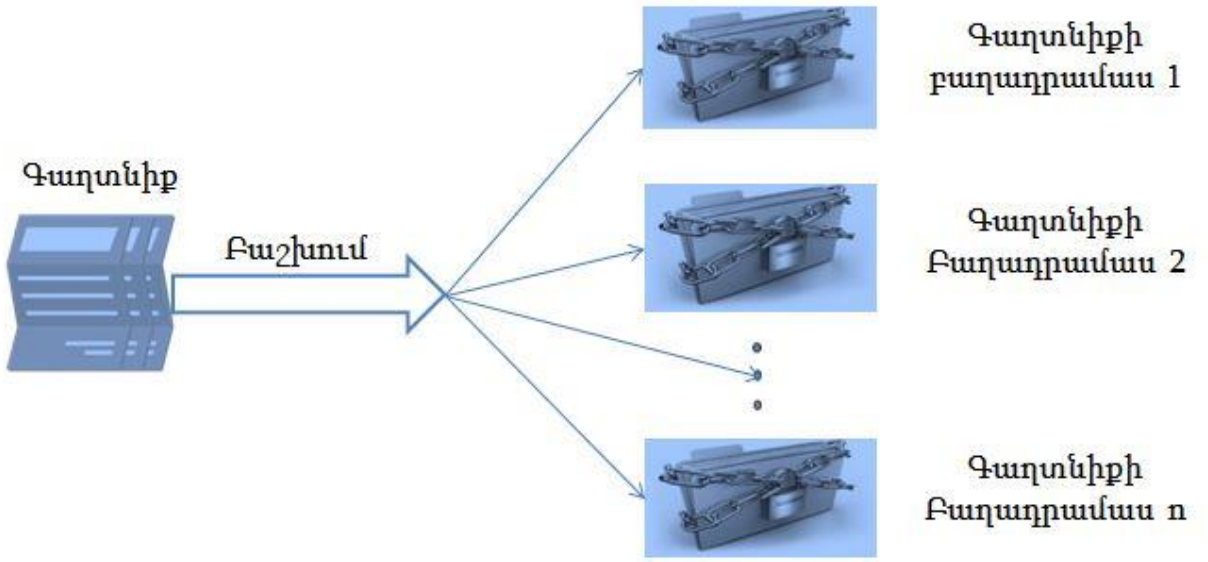
Վերը նշվածը սխեմատիկ պատկերված է նկ. 4-ում: Այս տարբերակով բաշխման դեպքում ունենում ենք մեկ գաղտնագրված ֆայլ և n հատ գաղտնագրման բանալու բաղադրամաս:



Նկ. 4. Գաղտնի ինֆորմացիայի գաղտնագրման և գաղտնագրման բանալու բաշխման սխեմա

Այս տարբերակով բաշխումը ապահովում է ալգորիթմով նախատեսված տվյալների հնարավոր գաղտնիությունը, սակայն չի ապահովում նրանց ամբողջականությունն ու հասանելիությունը: Եթե գաղտնագրված ինֆորմացիան վնասվի, կորի կամ անհրաժեշտ պահին հասանելի չլինի, ապա ակնհայտ է, որ գաղտնագրման բանալու վերականգնման դեպքում հնարավոր չէ կատարել գաղտնի ինֆորմացիայի վերծանում: Այսինքն չի ապահովվում և հասանելիությունը և ամբողջականությունը:

Այդ խնդիրների լուծման համար գոյություն ունի գաղտնիքի բաշխման մեկ այլ մոտեցում, այն է՝ ինֆորմացիայի ամբողջական բաշխումը: Այս դեպքում գաղտնագրում չի կատարվում, այլ ինֆորմացիան ամբողջությամբ բաշխվում է և ստացվում է n հատ գաղտնիքի ծավալին հավասար բաղադրամաս: Ամբողջական բաշխումը սխեմատիկ պատկերված է նկ. 5-ում:



Նկ. 5. Գաղտնի ինֆորմացիայի ամբողջական բաշխման սխեմա

Այս տարբերակով բաշխման դեպքում ապահովվում է և գաղտնիությունը և ամբողջականությունը և հասանելիությունը: Գաղտնիության ապահովումը ակնհայտ է, քանի որ պակաս քանակի բաղադրամասերով պարզապես հնարավոր չէ

վերականգնել գաղտնիքը: Ամբողջականությունը և հասանելիությունը ապահովվում է նրա շնորհիվ, որ օրինակ մեկ բաղադրամասի վնասման կամ հասանելի չլինելու դեպքում հնարավոր է մնացած բաղադրամասերով կատարել գաղտնի ինֆորմացիայի վերականգնում (այն դեպքում երբ $t < n$): Վերը նշվածը չի վերաբերվում այն դեպքին, երբ $t = n$: Այս դեպքում նույնիսկ մեկ բաղադրամասի վնասումը կամ տվյալ պահին հասանելի չլինելը բերում է գաղտնիքի վերականգնման անհնարինության:

Նկարագրված երկու մոտեցումների համեմատական աղյուսակը ներկայացված է աղյուսակ 1-ում:

Աղյուսակ 1: Գաղտնիքի բաշխման երկու տարբեր մոտեցումների համեմատման աղյուսակ

		Անվտանգություն		
		Գաղտնիություն	**Ամբողջականություն**	**Հասանելիություն**
Մեթոդ	**Գաղտնագրում + բանալու բաշխում**	այո	ոչ	ոչ
	Ամբողջական բաշխում	այո	այո	այո

Այսպիսով, գաղտնի ինֆորմացիայի ամբողջական բաշխումը անվտանգության տեսանկյունից ավելի նախընտրելի է, սակայն ավելի տարածված է գաղտնանգրման և գաղտնագրման բանալու բաշխման տարբերակը: Դա պայմանավորված է նրանով, որ ներկայում գոյություն ունեցող գաղտնիքի բաշխման մեթոդները դանդաղ են և հնարավարություն չեն ընձեռում կատարել մեծ ծավալի ինֆորմացիայի բաշխում (երկար ժամանակ է պահանջվում): Անհրաժեշտություն է առաջանում ունենալ ավելի արագագործ գաղտնիքի բաշխման մեթոդ, որը հնարավորություն կընձեռի հնարավորինս արագ կատարել մեծ ծավալի ինֆորմացիայի բաշխում և վերականգնում:

Գաղտնիքի բաշխման մեթոդների ուսումնասիրությունը ցույց տվեց, որ հնարավոր է կառուցել նոր, արագագործ գաղտնիքի բաշխման մեթոդներ՝ հիմնված սխալներ ուղղող կոդերի վրա [12- 18]:

1.4. Սխալներ ուղղող կոդերի հետազոտումը

Այս մասում նկարագրված են կոդավորման տեսության հիմնարար հասկացությունները, որոնք կարևոր են այս աշխատանքի շրջանակում [19, 22]:

Կոդավորման տեսության գլխավոր նպատակն է ձևափոխել ինֆորմացիան այնպես, որ փոխանցման կամ պահպանման ընթացքում առաջացող պատահական բնույթի սխալները հնարավոր լինի հայտնաբերել և նույնիսկ ուղղել: Այդ նպատակի համար նախնական ինֆորմացիային կցվում է ավելցուկային մաս, որը արհեստականորեն մեծ տարբերություն է ստեղծում տվյալների միջև: Այսինք, մի ինֆորմացիան մյուսից տարբերվում է հնարավորին շատ դիրքերում: Դա հնարավորություն է ընձեռում հայտնաբերել արդյոք սխալ առաջացել է թե ոչ և եթե սխալ է առաջացել, ապա գտնել դրա դիրքը ու ուղղել:

Կոդավորման տեսության մեջ ինֆորմացիան նախապես սահմանված այբուբենի բառ է: Ձևափոխված ինֆորմացիան անվանում են կոդաբառ: Բոլոր այդ կոդաբառերի համախումբը կոչվում է կոդ, իսկ այդ ձևափոխման գործընթացը՝ կոդավորում [20]: Այս աշխատանքում օգտագործվել են միայն հատվածային կոդեր: Հատվածային կոդերը այն կոդերն է, որոնց բոլոր կոդաբառերը ունեն նույն երկարությունը:

Սահմանում 1: Դիցուք A-ն վերջավոր բազմություն է (այբուբեն) և $n \in N$: Յուրաքանչյուր C ենթաբազմություն, որն ունի հետևյալ տեսքը՝ $A_n = \underbrace{A \times \ldots \times A}_{n}$ հանդիսանում է n երկարության հատվածային կոդ : C-ի տարրերը կոչվում են կոդաբառեր [19, 20]: Այն դեպքում, երբ $A = Z_2$, ապա C-ն երկուական հատվածային կոդ է:

Օրինակ, ենթադրենք ունենք $A = Z_2$ այբուբենը և հետևյալ երկուական հատվածային կոդը՝

$$C = \begin{Bmatrix} (0,0,0,0), (0,0,1,1), (0,1,0,1), (0,1,1,0), \\ (1,0,0,1), (1,0,1,0), (1,1,0,0), (1,1,1,1) \end{Bmatrix} \subseteq \mathrm{A}^4$$

Այս կոդը ստացվում է ութ նախնական բառերից՝ $\{(0,0,0), (0,0,1), (0,1,0), (0,1,1), (1,0,0), (1,0,1), (1,1,0), (1,1,1)\} \subseteq \mathrm{A}^3$, որոնց կոդավորելուց հետո ավելացվում են հետևյալ ավելցուկային բիթերը՝

$$(0,0,0) \to (0,0,0,0) \qquad (0,0,1) \to (0,0,1,1)$$

$$(0,1,0) \to (0,1,0,1) \qquad (0,1,1) \to (0,1,1,0)$$

$$(1,0,0) \to (1,0,0,1) \qquad (1,0,1) \to (1,0,1,0)$$

$$(1,1,0) \to (1,1,0,0) \qquad (1,1,1) \to (1,1,1,1)$$

Այսպիսի կոդավորումը կոչվում է զույգության ստուգման կոդ [19]: Նախնական բառը կոդավորվում է ՝ ավելացնելով այնպիսի բիթ, որ ստացված կոդաբառում մեկերի քանակը լինի զույգ:

Սահմանում 2: Դիցուք A-ն վերջավոր այբուբեն է (բազմություն), իսկ $n \in N$: Դիտարկենք երկու բառեր՝$a = (a_1, \dots, a_n) \in A^n$ և $b = (b_1, \dots, b_n) \in A^n$: Հեմմինգի հեռավորություն է կոչվում այն դիրքերի քանակը, որոնցում այս երկու կոդաբառերը ունեն տարբեր արժեքներ ($a_i \neq b_i$) [20]: Հեմմինգի հեռավորությունը սահմանվում է նույն երկարության կոդաբառերի համար [52]: Այդ հեռավորությունը պայմանական նշանակենք այսպես.

$$d(a,b) := \{i : 1 \leq i \leq n, a_i \neq b_i\}$$

Դիցուք $C \in A^n$ կամայական կոդ է, ապա

$$d(c) = \min \{d(a,b) : a, b \in C, a \neq b\}$$

d(c)-ն կոչվում C կոդի նվազագույն հեռավորություն [19]:

Օրինակ, դիտարկենք հետևյալ երկուական կոդը՝

$$C = \{\underbrace{(0,0,0,0)}_{a}, \underbrace{(0,0,1,1)}_{b}, \underbrace{(0,1,0,1)}_{c}, \underbrace{(0,1,1,0)}_{d}, \underbrace{(1,0,0,1)}_{e}, \underbrace{(1,0,1,0)}_{f}, \underbrace{(1,1,0,0)}_{g}, \underbrace{(1,1,1,1)}_{h},\}$$

Ըստ վերը տրված սահմանման, այս կոդը ունի հետևյալ Հեմմինգի հեռավորությունները՝

dmi	a	b	c	d	e	f	g	h
a	0	2	2	2	2	2	2	4
b	2	0	2	2	2	2	4	2
c	2	2	0	2	2	4	2	2
d	2	2	2	0	4	2	2	2
e	2	2	2	4	0	2	2	2
f	2	2	4	2	2	0	2	2
g	2	4	2	2	2	2	0	2
h	4	2	2	2	2	2	2	0

Աղյուսակի ուսումնասիրությունը ցույց է տալիս, որ նվազագույն հեռավորությունը՝ $d(c) = 2$: Եթե համեմատենք նախնական բառերը, ապա նվազագույն կոդային հեռավորությունը կլինի մեկ (օրինակ (0,0,1) և (0.1.1)), իսկ կոդավորումից հետո այն դառնում է երկու: Այսինքն կոդավորումը արհեստականորեն մեծացնում է նվազագույն կոդային հեռավորությունը, որը հետագայում հնարավորություն է ընձեռում հայտնաբերել և նույնիսկ ուղղել սխալները [54, 55]:

Այժմ պատկերացնենք, որ c կոդաբառը փոխանցվել է կապուղիով և կապուղու մյուս մասում ստացվել է x բառը: Բնականաբար ստացողը գիտի, որ փոխանցվել է կոդաբառ: Հնարավոր է երկու տարբերակ.

- ստացված բառը հանդիսանում է այդ կոդի կոդաբառ,
- ստացված բառը չի հանդիսանում է այդ կոդի կոդաբառ:

Առաջին դեպքում ստացողը համարում է, որ ուղարկվել է x կոդաբառը և այն չի վնասվել: Երկրորդ դեպքում ստացողը կամ պահանջում է նորից ուղարկել այդ

կոդաբառը կամ փորձում է x-ից ստանալ c կոդաբառը: Այդ ստացման գործընթացը կոչվում է ապակոդավորում (*decoding*):

Ստորև նկարագրվող ալգորիթմը կոչվում է Հեմմինգի ապակոդավորման ալգորիթմ:

Սահմանում 3: Դիցուկ ունենք $C \subseteq A^n$ կոդը: Պատկերացնենք, որ փոխանցվել է այդ կոդի կոդաբառ և ստացվել $x \subseteq A^n$ բառը: Հեմմինգի ապակոդավորման ալգորիթմը դուրս է բերում $c \subseteq C$ կոդաբառը, որը բավարարում է հետևյալ պայմանին [19].

$$d(c, x) = \min_{c' \in c} d(c', x)$$

Այլ կերպ ասած, Հեմմինգի ապակոդավորման ալգորիթմը գտնում է այն կոդաբառը, որն ամենաքիչն է տարբերվում ստացված x բառից: Եթե այդպիսի կոդաբառերի քանակը մեկից ավել է, ապա ալգորիթմը պատահականորեն դուրս է բերում նրանցից մեկը:

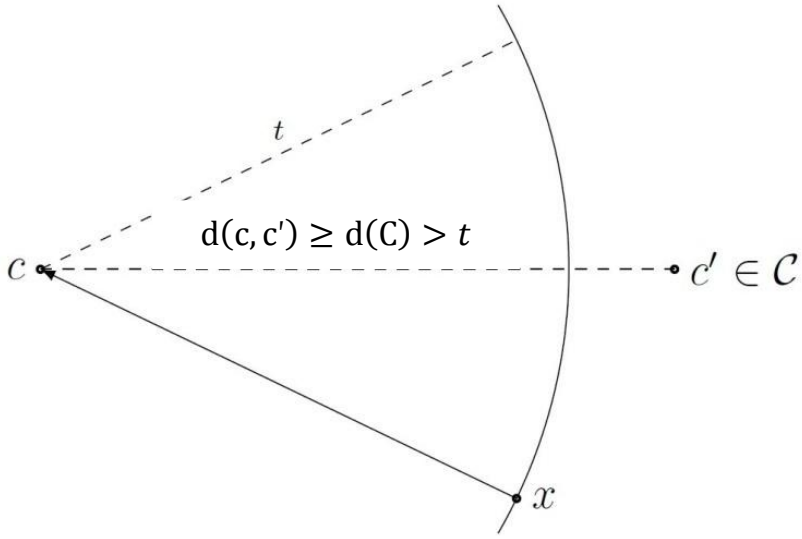
Վերադառնալով վերը նկարագրված կոդին, պատկերացնենք, որ ստացվել է $x = (1,0,0,0) \in Z_2^4$: Այս դեպքում Հեմմինգի ապակոդավորման ալգորիթմը կարող է դուրս բերել հետևյալ բառերը.

(0,0,0), (1,0,0), (1,0,1), (1,1,0):

Բոլոր այս բառերի դեպքում էլ Հեմմինգի հեռավորությունը x-ից հավասար է մեկի:

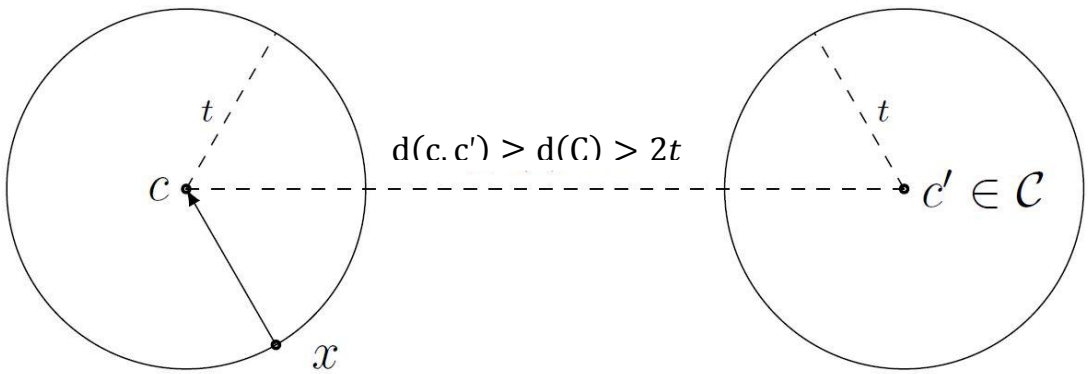
Այժմ դիտարկենք նվազագույն կոդային հեռավորությունը՝ կոդի սխալներ հայտնաբերելու և ուղղելու համատեքստում [50]: Այն դեպքում, երբ փոխանցվել է $c \in C \subseteq A^n$ կոդաբառը և ստացվել է $x \in A^n$, որը ունի $d(c, x) = t$ Հեմմինգի հեռավորություն, ապա կարող ենք ասել, որ ստացված բառը ունի t քանակի սխալ [47]: Այլ խնդիր է, հնարավոր կլինի հայտնաբերել և ուղղել այդ սխալները թե ոչ: Այդ ամենը կախված է կոդի նվազագույն հեռավորությունից:

- Դիցուք $d(C) \geq t+1$, ապա x-ը չի կարող լինել C կոդի կոդաբառ և կարելի է վստահ ասել, որ սխալ է առաջացել: C-ն անվանում են t սխալ հայտնաբերող կոդ:



Նկ. 6. Հեմմինգի հեռավորության և սխալներ հայտնաբերելու քանակի սխեմատիկ պատկերումը

- Դիցուք $d(C) \geq 2t+1$, ապա c-ն կոդաբառ է, որը ընկած է x-ի մոտակայքում և Հեմմինգի ապակոդավորումը դուրս է բերում կոդաբառը: C-ն կոչվում է t սխալ ուղղող կոդ:



Նկ. 7. Հեմմինգի հեռավորության և սխալներ ուղղելու քանակի սխեմատիկ պատկերումը

Սա նշանակում է, որ C կոդը կարող է ուղղել մինչև $\left[\frac{d(C)-1}{2}\right]$ սխալ, անկախ նրանից թե որ կոդաբառն է ուղարկվել և որ դիրքերում են սխալներն առաջացել: Այդ իսկ պատճառով $\left[\frac{d(C)-1}{2}\right]$-ն անվանում են C կոդի սխալներ ուղղելու հնարավորություն (*error-correction capability*): Այնուամենայնիվ, կարող են լինել կոդաբառեր և սխալների դիրքեր, որոնց դեպքում Հեմմինգի ապակոդավորումը կարող է դուրս բերել ճիշտ կոդաբառը, այն դեպքում, երբ սխալների քանակը ավելին է քան կոդի սխալներ ուղղելու կարողությունը՝ $\left[\frac{d(C)-1}{2}\right]$:

Դիտարկենք երկու օրինակ՝

1. Ինչպես տեսանք վերը նկարագրված օրինակում, երկուական կոդի նվազագույն կոդային հեռավորությունը $d = 1$, որից հետևում է, որ այն մեկ սխալ հայտնաբորող և զրո սխալ ուղղող կոդ է:
2. Դիտարկենք հետևյալ երկուական կոդը.

$$C = \{(0,0,0,0,0,0), (0,1,0,1,0,1), (1,0,1,0,1,0), (1,1,1,1,1,1)\}$$

Այս կոդն ունի $d(C) = 3$ նվազագույն կոդային հեռավորություն, որից հետևում է, որ այն երկու սխալ հայտանբերող և մեկ սխալ ուղղող երկուական կոդ է: Սակայն, երբ փոխանցվում է $c = (0,0,0,0,0,0)$ կոդաբառը և ստացվում $x = (1,1,0,0,0,0)$ բառը, ապա առաջացել է երկու սխալ և Հեմմինգի ապակոդավորումը այնուամենայնիվ դուրս է բերում ուղարկված c կոդաբառը:

Այժմ սահմանենք կոդի մեկ այլ կարևոր հատկանիշ՝ ծածկման շառավիղ (*covering radius*) [51, 56]: Ծածկման շառավիղը դա Հեմմինգյան հեռավորությունն է կոդից մինչև ամենահեռու բառը (կոդից դուրս):

Սահմանում 4: Դիցուք K-ն վերջավոր դաշտ է, $n \in N$ և $C \subseteq K^n$ կոդ է: Ամենափոքր $r \in \mathrm{N}_0$, այնպես, որ բոլոր $x \in K^n$ բառերի համար, գոյություն ունի $c \in C$ կոդաբառ, որ $d(x,c) \leq r$, կոչվում է C կոդի ծածկման շառավիղ՝ $p(C)$ [23]:

Այս սահմանումից պարզ է դառնում, որ r-ը այն ամենափոքր շառավիղն է, որ եթե բոլոր կոդաբառերի շուրջը գծենք r շառավիղով շրջաններ, ապա այդ շրջանները կծածկեն ամբող A^{n} տիրույթը: Սահմանումից ուղիղ հետևում է, որ՝

$$p(C) \geq \left[\frac{d(C)-1}{2}\right]$$

Զույգուցյան ստուգման կոդը ունի $p(C) = 1$ ծածկման շառավիղ: Դիցուք $x = (x_1, x_2, x_3, x_4) \in Z_2^4$ կամայական բառ է, ապա x -ը կամ $x' = (x_1, x_2, x_3, x_4 + 1)$ ունի զույգ քանակի մեկեր, որից հետևում է, որ այն C կոդի կոդաբառ է: Այսպիսով Հեմմինգի հեռավորությունը x-ից մինչև կոդ կամ «զրո» է կամ «մեկ»:

Առավել հաճախ օգտագործվում են այսպես կոչված գծային կոդերը:

Սահմանում 5: Դիցուք K-ն վերջավոր դաշտ է, $n \in N$ և $C \subseteq K^n$ կոդ է: Ենթադրենք C-ն K^n-ի գծային ենթատարածությունն է: Այս դեպքում C –ն կոչվում է գծային կոդ: Դիցուք k-ն C կոդի չափն է, ապա այդ կոդը կոչվում է $[n,k]$ կոդ կամ $[n,k,d(C)]$ կոդ: Եթե K-ն ունի q տարր, ապա մենք նաև ասում են, որ C-ն $[n,k,d(C)]_q$ կոդ է և $q = 2$ դեպքի համար անվանում ենք երկուական $[n,k,d(C)]$ կոդ:

Գծային կոդերի օգտագործմամբ շատ է հեշտանում կոդավորման և կոդաբառի ստուգման գործողությունները:

Այսպիսով, սխալներ ուղղող կոդերը, օգտագործելով ավելցուկայնության գաղափարը, կարողանում են ուղղել վնասված կամ պակասող բիթերը: Այս առումով կոդերը նման են գաղտնիքի բաշխման մեթոդներին [8-11], քանզի նրանցում նույնպես ներդրված է ավելցուկայնության գաղափարը, որը հնարավորություն է տալիս

վերականգնել գաղտնիքը որոշակի քանակով բաղադրամասերի միավորումով: Գաղտնիքի բաշխման մեթոդների ուսումնասիրությունը ցույց է տալիս, որ բոլոր բաշխման մեթոդների հիմքում էլ ընկած է ավելցուկայնության գաղափարը: Ուստի տրամաբանական է, որ նոր բաշխման մեթոդ նախագծելիս հարկավոր է ուսումնասիրել այն ուղղությունները, որոնցով հնարավոր է ստանալ ավելցուկայնություն (իմաստային առումով կապակցված): Այդ հետազոտությունները ցույց են տվել, որ սխալներ ուղղող կոդերը կարող են հանդիսանալ այդ ավելցուկայնության աղբյուրը [12,16,29-34]:

1.5. Սխալներ ուղղող կոդերի և գաղտնիքի բաշխման մեթոդների կապը

Առաջին անգամ սխալներ ուղղող կոդերի և գաղտնիքի բաշխման մեթոդների կապի մասին խոսվել է դեռևս 1981թ.-ին [15]: Մքիլլիսը (*McEliece*) և Սարվաթը (*Sarwate*) իրենց հոդվածում [15] խոսեցին Շամիրի շեմային մեթոդի և Ռիդ-Սոլոմոնի սխալներ ուղղող կոդի [22] նմանության մասին: Ստորև հանգամանալից ներկայացնենք այդ կապը:

Դիցուք $(\alpha_1, \alpha_2, \dots, \alpha_{r-1})$, F վերջավոր դաշտի ֆիքսված, ոչ զրոյական տարրեր են: Ռիդ-Սոլոմոնի կոդով կոդավորման դեպքում $a = (a_0, a_1, \dots, a_{k-1}), a_i \in F$ բառը կոդավորվում է $D = (D_1, D_2, \dots, D_{r-1})$ կոդաբառի, որտեղ $D_i = \sum_{j=0}^{k-1} a_j \alpha_i^j$: Այստեղ կարող ենք որպես «գաղտնիք» վերցնել $a_0 = -\sum_{i=1}^{r-1} D_i$, իսկ գաղտնի բաղադրամասերը՝ D_i-ները: Ենթադրենք s հատ բաղադրամաս փոխանցվել է և t հատը ստացվել են սխալներով: Այստեղ կիրառելով սխալներ ուղղող ալգորիթմը [22], կարող ենք վերականգնել D-ն և հետևաբար նաև a_0-ն (միայն այն դեպքում, երբ $s - 2t \geq k$): Շամիրի մեթոդն իրենից ներկայացնում է վերը նկարագրվածի մասնավոր դեպքը, երբ r-ը պարզ թիվ է, $\alpha_i = i$ և $t = 0$:

Եթե հակառակորդը փորձում է այնպես անել, որ արտոնված մասնակիցները չկարողանան վերականգնել գաղտնիքը, ապա դրա համար նա կարող է կեղծել բաղադրամասերը (գաղտնի կամ կաշառման եղանակով): Դժվար չէ տեսնել, որ եթե t հատ բաղադրամաս կեղծվել է, ապա գաղտնիքը կարող է վերականգնվել իրական բաղադրամասերի միջոցով (եթե նվազագույնը $k + t$ չվնասված բաղադրամաս մնացել է): Այլ կերպ ասած, (k,n) շեմային սխեմայում հակառակորդը պետք է կեղծի $\left[\frac{n-k}{2}\right]$-ից ավել բաղադրամաս, որպեսզի գաղտնիքը դառնա անհասանելի (վերականգնելը հնարավոր չլինի):

Վերը նկարագրված նմանությունից հետևում է, որ կոդերի կիրառությունը գաղտնիքի բաշխման մեթոդներ ստողծելու առումով կարող է բավական հետաքրքիր արդյունքների բերել: Օրինակ հաշվի առնելով սխալներ ուղղող կոդերի այն հատկությունը, որ նրանք կարողանում են գտնել սխալի դիրքը, հնարավոր է ստանալ բաշխման մեթոդ, որտեղ վերականգնման ժամանակ կարելի է գտնել կեղծված բաղադրամասը: Օրինակ հակառակորդի կողմից ներկայացված կեղծված բաղադրամասը և հետևաբար նաև հակառակորդին:

1.6. Խնդրի դրվածքը

Հետազոտություններից հետևում է, որ անհրաժեշտություն կա հետազոտել և մշակել սխալներ ուղղող կոդերի վրա հիմնված գաղտնիքի բաշխման արագ մեթոդ, որը հնարավորություն կընձեռի արագ բաշխել և վերականգնել մեծ ծավալի ինֆորմացիա, որով հնարավոր կլինի ապահովել այդ ինֆորմացիայի ինչպես գաղտնիությունը, այնպես էլ ամբողջականությունն ու հասանելիությունը: Այդ մեթոդի համար անհրաժեշտ է նաև մշակել բաղադրամասի ստուգման և կորած կամ վնասված բաղադրամասի վերականգնման հնարավորություններ: Այսպիսով ընդհանրացնելով ասվածը, որպես աշխատանքի երեք հիմնական նպատակ կարող ենք նշել.

- մշակել գաղտնիքի բաշխման արագագործ շեմային մեթոդ, որը հնարավորություն կտա բաշխել և վերականգնել մեծ ծավալի ինֆորմացիա:
- մշակել սխալներ ուղղող կոդերով բաշխման դեպքում բաղադրամասի հավաստիության ստուգման մեթոդ:
- մշակել սխալներ ուղղող կոդերով բաշխման դեպքում վնասված կամ կորած բաղադրամասի արագ վերականգնման մեթոդ:

1.7. Գլուխ 1-ի ամփոփում

Այս գլխում ներկայացվեց գաղտնիքի բաշխման հիմնական գաղափարը, կարևոր սահմանումները և հասկացությունները: Նկարագրվեց խնդիրների այն շրջանակը, որոնց լուծման համար անհրաժեշտ են այդ մեթոդները: Հակիրճ նկարագրվեցին առավել հայտնի Շամիրի և Բլեկլիի շեմային մեթոդները: Այս գլխում նաև ցույց տրվեց գաղտնիքի բաշխման շեմային մեթոդների կիրառման երկու տարբերակները, նրանց առավելություններն ու թերությունները:

Բացի այդ, այս գլխում տրվեց սխալներ ուղղող կոդերի ընդհանուր գաղափարը, կարևոր սահմանումներն ու հասկացողությունները: Խոսվեց նաև սխալներ ուղղող կոդերի և գաղտնիքի բաշխման մեթոդների կապի մասին:

Հետազոտությունների արդյունքում պարզ դարձավ, որ գոյություն ունեցող գաղտնիքի բաշխման շեմային մեթոդները դանդաղ են և մեծ ծավալի ինֆորմացիայի բաշխման համար անհրաժեշտություն է առաջանում ունենալ ավելի արագագործ գաղտնիքի բաշխման շեմային մեթոդ: Որպես այդ մեթոդի մշակման ուղություն ընտրվեց սխալներ ուղղող կոդերի վրա մեթոդի կառուցումը:

Սխալներ ուղղող կոդերով գաղտնիքի բաշխման որոշակի լուծումներ նկարագրված են [12-18,29-34] հոդվածներում: Սակայն բոլոր այդ լուծումները մասնավոր դեպքեր են և դրանցից ոչ մեկում ներկայացված չէ ամբողջական գաղտնիքի բաշխման մեթոդ հիմնված սխալներ ուղղող կոդերի վրա: Բացակայում է նրանց արագագործության և անվտանգության գնահատականները: Ինչպես նաև մշակված չեն գաղտնիքի բաղադրամասի ստուգման, կորած կամ վնասված բաղադրամասի վերականգնման և այլ հարակից հնարավորություններ, որոնք սովորաբար բնորոշ են գաղտնիքի բաշխման մեթոդներին (օրինակ Շամիրի մեթոդը [8] օժտված է բոլոր այդ հնարավորություններով):

Գլուխ 2:

Սխալներ ուղղող կոդերով գաղտնիքի բաշխման մեթոդի մշակումը

Գոյություն ունեն գաղտնիքի բաշխման մեթոդների կառուցման տարբեր ճանապարհներ, որոնք հիմնվում են որոշակի մաթեմատիկական ապարատների և գաղափարների վրա: Օրինակ Շամիրը օգտագործել է վերջավոր դաշտում բազմանդամի միջարկումը [8]: Բլեկլին և Սիմմոնսը ներկայացրել են գաղտնիքի բաշխման երկրաչափական լուծում, որտեղ որպես գաղտնիք հանդես է գալիս գերհարթությունների հատման կետը [9,11]: Ասմութն ու Բլումը տվել են հանրահաշվական լուծում, որը հիմնված է մնացորդների մասին Չինական թեորեմի վրա [27]: Ստորև նկարագրվում է սխալներ ուղղող կոդերով գաղտնիքի բաշխման լիովին նոր մեթոդ: Մեթոդի հիմքում ընկած են հատվածային սխալներ ուղղող կոդերը: Մշակված մեթոդը համեմատվում է առավել տարածված Շամիրի գաղտնիքի բաշխման մեթոդի հետ:

2.1. Բաշխման նոր մեթոդի նկարագրությունը

Բաշխման նոր մեթոդը հիմնված է սխալներ ուղղող կոդերի վրա և կիրառելի է ցանկացած հատվածային երկուական կոդի համար [48]: Ներկայումս գոյություն ունեցող սխալներ ուղղող կոդերով գաղտնիքի բաշխման մեթոդների գաղափարը հետևյալն է [12-18, 29-34].

1. որպես գաղտնիք հանդես է գալիս տվյալ կոդի կոդաբառը,
2. բաղադրամասերը իրենցից ներկայացնում են նույն երկարության բառեր, այնպիսի հատկությամբ, որ արտոնված խմբի բաղադրամասերի միավորումից հետո ստացված բառը շատ «մոտ» է գաղտնիքին և տվյալ կոդի ուղղող հատկությունը կարող է ուղղել այդ տարբերությունը և ստանալ գաղտնիքը:

Եթե գաղտնիքը նշանակենք s-ով (դա որևէ կոդի կոդաբառ է), իսկ վերականգնման համար ընտրված բաղադրամասերը՝ $k_{j_1}, \dots, k_{j_e}$, ապա վերը նշվածը մաթեմատիկորեն կարելի է ներկայացնել այսպես՝

$$d(k_1 + \cdots + k_j, s) \leq \frac{d-1}{2}, \text{ եթե խումբը արտոնված է}$$

$$d(k_1 + \cdots + k_j, s) > \frac{d-1}{2}, \text{ եթե խումբը արտոնված չէ}$$

Այստեղ d-ն կոդաբառերի Հեմմինգյան հեռավորությունն է, իսկ գումարում գործողության տակ հարկավոր է հասկանալ որևէ միավորման գործողություն (որոշակի ալգորիթմով): Այս մոտեցումը ունի էական թերություն, որ գաղտնիքը անպայման պետք է լինի կոդաբառ, որը նվազեցնում է հնարավոր տարբեր գաղտնիքների քանակը և հեշտացնում հատարկման եղանակով գրոհի արդյունավետությունը: Սխալներ ուղղող կոդերով ներկայում հայտնի բաշխման մեթոդները այս կամ այն կերպ կրկնում են այս տրամաբանությունը [12-18, 29-34]:

Առաջարկվում է վերը նշված մոտեցումից տարբերվող այլ մոտեցում, որտեղ գաղտնիքը կարող է լինել կամայական բիթերի շարք: Ինչպես արդեն նշվեց բաշխման մեթոդը կիրառելի է ցանկացած հատվածային երկուական սխալներ ուղղող կոդի համար:

Պարզության համար բաշխման ալգորիթմը դիտարկենք (7,4,1) Հեմմինգի կոդի համար [20, 49]: Այս կոդը աշխատում է 4 բիթ մուտքային ինֆորմացիայի հետ և կոդավորումից հետո ստացվում է 7 բիթ երկարությամբ կոդաբառ, և այս կոդը կարող է ուղղել կամայական 1 սխալ: Ալգորիթմի աշխատանքը սկսվում է նրանից , որ գաղտնի ինֆորմացիան (գաղտնի ֆայլը) կոդավորվում է (7,4,1) Հեմմինգի կոդով (նկ. 8), որից հետո ստացվում է կոդավորված ինֆորմացիան:

Նկ. 8. Գաղտնի ինֆորմացիայի կոդավորումը

Այդ կոդավորված ինֆորմացիան կարելի է ներկայացնել նկ. 9-ում պատկերված երկչափ զանգվածի տեսքով:

1	2	3	…	k
$V_{1,1}$	$V_{1,2}$	$V_{1,3}$	⋯	$V_{1,k}$
$V_{2,1}$	$V_{2,2}$	$V_{2,3}$	⋯	$V_{2,k}$
$V_{3,1}$	$V_{3,2}$	$V_{3,3}$	⋯	$V_{3,k}$
$V_{4,1}$	$V_{4,2}$	$V_{4,3}$	⋯	$V_{4,k}$
$V_{5,1}$	$V_{5,2}$	$V_{5,3}$	⋯	$V_{5,k}$
…	…	…	…	…
$V_{q,1}$	$V_{q,2}$	$V_{q,3}$	⋯	$V_{q,k}$

Նկ. 9. Կոդավորված գաղտնի ինֆորմացիայի կառուցվածքը

Այստեղ k-ն կոդաբառի երկարությունն է (այս դեպքում k=7), իսկ q-ն կախված է բաշխվող ինֆորմացիայի ծավալից: Այս զանգվածի յուրաքանչյուր տող իրենից ներկայացնում է (7,4,1) Հեմմինգի կոդի կոդաբառ, ինչը նշանակում է, որ այդ 7 բիթում կամայական 1 սխալ կամ բացակայող բիթ կարելի է ուղղել: Հիմնվելով այս հատկության վրա կատարենք բաշխում ըստ սյուների: Եթե յուրաքանչյուր սյուն դիտարկենք որպես առանձին բաղադրամաս, ապա ակնհայտ է, որ կկարողանանք վերականգնել նախնական գաղտնի ինֆորմացիան (գաղտնի ֆայլը) կամայական 1 բաղադրամասի վնասվելու կամ կորստի դեպքում: Ստացվում է (4,7) շեմային կառուցվածք: Սակայն ակնհայտ է, որ այս դեպքում խախտվում է շեմային մեթոդների վրա դրվող պայմաններից մեկը՝ շեմից պակաս բաղադրամասերի միավորումից պետք է հնարավոր չլինի վերականգնել գաղտնի ինֆորմացիան: Այս դեպքում առաջին 4 սյուների միավորումը բավարար է գաղտնի ինֆորմացիայի վերականգնման համար: Այդ պատճառով առաջարկվում է կատարել խմբավորում և կողմերին տրամադրել սյուների խմբերը:

Որպեսզի բարձրացնենք գաղտնակայունությունը, և բաղադրամասերի ծավալները հավասար լինեն բաշխվող ինֆորմացիայի (ֆայլի) ծավալին՝ կատարվում է խմբավորում [35-38]: Որպեսզի ստացված բաղադրամասերը ունենան նույն ծավալը, ինչ սկզբնական ինֆորմացիան (ֆայլը), անհրաժեշտ է յուրաքանչյուր մասում վերցնել 4 սյուն (որոշակի մեթոդով ընտրված): Օրինակ, խմբավորելով ըստ աղյուսակ 2-ի կստանանք (2,4) շեմային կառուցվածք:

Աղյուսակ 2: Հեմմինգի կոդով (2,4) շեմային կառուցվածք

Մաս 1	**3**	**6**	**2**	**7**
Մաս 2	**1**	**6**	**7**	**5**
Մաս 3	**5**	**4**	**2**	**6**
Մաս 4	**7**	**5**	**3**	**4**

Յուրաքանչյուր տող իրենից ներկայացնում է առանձին բաղադրամաս: Աղյուսակում լրացված թվերը 1-7 բիթերի (սյուների) համարներն են: Աղյուսակ 2-ից երևում է, որ կամայական 2 բաղադրամաս միավորելուց հետո պակասում է 1 սյուն: Օգտագործելով (7,4,1) Հեմմինգի կոդի ապակոդավորման ալգորիթմը՝ կարելի է վերականգնել այդ 1 սյունը: Ստացվում է, որ, ըստ աղյուսակ 2-ի բաշխման դեպքում, ունենում ենք (2,4) շեմային կառուցվածք: Գաղտնի ինֆորմացիան բաշխվում է 4 մասի և կամայական 2-ով վերականգնվում: Աղյուսակ 3-ում ներկայացված է (2,3) շեմային կառուցվածքի աղյուսակը, իսկ աղյուսակ 4-ն ապահովում է (3,4) շեմային բաշխումը:

Աղյուսակ 3: (7,4,1) Հեմմինգի կոդով (2,3) շեմային կառուցվածք

Մաս 1	**5**	**2**	**5**	**4**
Մաս 2	**4**	**7**	**5**	**3**
Մաս 3	**6**	**2**	**7**	**3**

Աղյուսակ 4: (7,4,1) Հեմմինգի կոդով (3,4) շեմային կառուցվածք

Մաս 1	**7**	**6**	**2**	**7**
Մաս 2	**4**	**6**	**5**	**6**
Մաս 3	**5**	**1**	**5**	**7**
Մաս 4	**7**	**6**	**3**	**6**

Նույն տրամաբանությամբ խմբավորումը կատարվում է այլ բաղադրամասի և շեմի արժեքների համար [38-39]:

Վերը նկարագրված գաղտնիքի բաշխման մեթոդը բավարարում է իդեալական գաղտնիքի բաշխման մեթոդի սահմանմանը, սակայն չի բավարարում կատարյալության սահմանմանը: Այս բաշխման մեթոդը չի կարելի համարել

կատարյալ, քանի որ բաղադրամասերում մասնակցում են նաև նախնական գաղտնի ինֆորմացիայի բիթերը: Այս թերությունը բնորոշ է նաև այլ հեղինակների կողմից առաջարկված սխալներ ուղղող կոդերով բաշխման մեթոդներին [29-34]: Եթե հավանական հակառակորդը տեղյակ է, թե որ աղյուսակն է օգտագործվել բաշխման ժամանակ, ապա նա ունենալով շեմից պակաս բաղադրամասեր, կարող է որոշակի տեղեկություն ստանալ գաղտնի ինֆորմացիայի վերաբերյալ: Ստացվում է գաղտնիքի բաշխման մեթոդներին ներկայացվող պահանջներից մեկի խախտում, այն է. շեմից պակաս բաղադրամասերի միավորումը ոչինչով չպետք է նպաստի գաղտնիքի ամբողջական կամ մասնակի բացահայտմանը: Չնայած այն հանգամանքին, որ այդ ճանապարհով գաղտնիքի ամբողջական բացահայտում հնարավոր չէ, այնուամենայնիվ այդ խնդրի լուծման համար առաջարկվում է մի մոտեցում, որը պայմանականորեն կանվանենք վերադասավորում [40, 41]: Վերադասավորում գործողության իմաստը կայանում է նրանում, որ մինչ բաշխման գործընթացը, խմբավորման աղյուսակում կատարվեն որոշակի տեղափոխումներ, որի կատարման ձևի մասին հակառակորդը չպետք է իմանա: Առաջարկվում է խմբավորման աղյուսակում սյուները միմյանց հետ տեղափոխել, որից հետո ձևափոխված աղյուսակով կատարել բաշխում: Այլ կերպ ասած՝ խմբավորման աղյուսակը գաղտնագրվում է վերադասավորման գաղտնագրով: Ակնհայտ է, որ այդ տեղափոխությունը չի խախտում աղյուսակով ստացվող շեմային կառուցվածքը, բայց միևնույն ժամանակ հակառակորդին զրկում է աղյուսակին տիրապետելու հնարավորությունից: Վերադասավորման համար առաջարկվում է գեներացնել որոշակի քանակությամբ սյուների համարներ, որոնք անհրաժեշտ է զույգ առ զույգ տեղափոխել դիրքերով: Գեներացված հաջորդականությունը պայմանական անվանենք վերադասավորման գաղտնաբառ և որն ունի հետևյալ տեսքը՝ $ID_1,ID_2,ID_3,\ldots,ID_n$: Գեներացված սյուների քանակը պետք է լինի զույգ:

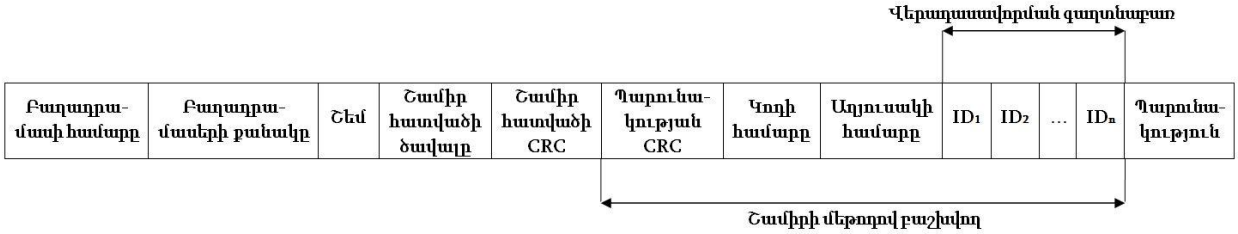
Վերադասավորման գաղտնաբառի ձևավորման գործընթացը բաղկացած է հետևյալ երկու քայլից.

1. Գեներացվում են n ($n = m + 2$ կամ $n = m + 3$ (այնպես, որ $n - ը$ լինի զույգ թիվ)) քանակությամբ ID համարներ (դրանք աղյուսակում համապատասխան սյուների համարներն են): Այդ ID-ների համախումբը կանվանենք բաշխման գաղտնաբառ: Այն կունենա հետևյալ տեսքը՝ ID_1,ID_2,ID_3,…,ID_n: Այստեղ m-ը կոդաբառի երկարությունն է:
2. Արդեն ընտրված աղյուսակում կատարվում է սյուների տեղափոխություն (ID_1-ը ID_2-ի հետ և այլն): Քանի որ գեներացվել են զույգ քանակով ID համարներ, ապա այս գործընթացը միշտ հնարավոր է:

Հետազոտությունները և փորձնական արդյունքները ցույց են տվել, որ այդ քանակով պատահականորեն գեներացված սյուների համարները ապահովում են աղյուսակի նախնական տեսքի համեմատ 75-100% տարբերություն: Հաշվի առնելով, որ այս գործողություններից հետո հակառակորդը կարող է միայն ենթադրություններ անել և կատարել հատարկման գործողություն, ապա այդ քանակով ID համարների գեներացումը կհամարենք բավարար, որպեսզի նախնական աղյուսակը համարվի «թաքցված»:

Այս ամենից հետո անհրաժեշտություն է առաջանում, որ վերականգնող կողմը նույնպես իմանա այդ վերադասավորման գաղտնաբառը, այլապես չի կարող վերականգնել գաղտնիքը: Բնականաբար այդ գաղտնաբառը հնարավոր չէ բաղադրամասերում բաց տեսքով ունենալ: Առաջարկվում է այդ բաղադրամասը բաշխել կողմերի միջև և բաշխված տարբերակով տեղադրել ստեղծվող բաղադրամասերում (ֆայլերում) [42-44]: Քանի որ այդ գաղտնաբառի ծավալը մեծ չէ, հետևաբար այն կարող ենք բաշխել Շամիրի մեթոդով և դա բաշխման մեթոդի արագագործության վրա էական ազդեցություն չի ունենա: Այդ բաշխումը կատարվում է

միայն մեկ անգամ՝ բաշխման սկզբում: Բնական է, որ բաշխումը պետք է կատարվի նույն պարամետրերով, որով կատարվելու է ամբողջ ինֆորմացիայի բաշխումը (նոր մեթոդով): Այս մոտեցումը հնարավորություն է տալիս նաև բաշխված պահել օգտագործվող կոդի և աղյուսակի համարը: Ասվածը ավելի պատկերավոր ներկայացված է նկ. 10-ում, որը բաղադրամաս-ֆայլերի գլխամասի (*Header*) կառուցվածքն է:



Նկ. 10. Բաղադրամաս-ֆայլի կառուցվածքը

Այժմ ներկայացնենք այդ կառուցվածքն ավելի մանրամասն.

- **Բաղադրամասի համարը** – բաղադրամասի հերթական համարը:
- **Բաղադրամասերի քանակը** –բաղադրամասերի ընդհանուր քանակը:
- **Շեմ** - բաշխման շեմի արժեքը: Այս տվյալը անհրաժեշտ է **«Շամիրի մեթոդով բաշխվող»** հատվածը վերականգնելու համար:
- **Շամիր հատվածի ծավալը** – **«Շամիրի մեթոդով բաշխվող»** հատվածի բայթերի քանակը: Այս տվյալը հարկավոր է ծրագրին այդ հատվածը ճիշտ ընթերցելու համար:
- **Շամիր հատվածի CRC** - **«Շամիրի մեթոդով բաշխվող»** հատվածի CRC32 արժեքը:
- **Պարունակության CRC** – այս դաշտում հաշվարկվում և պահվում է **«Պարունակություն»** հատվածի CRC32 արժեքը: Նախատեսված է այդ հատվածում հնարավոր փոփոխությունը արձանագրելու համար:
- **Կոդի համարը** – օգտագործվող սխալներ ուղղող կոդի հերթական համարը:

- **Աղյուսակի համարը** – ընտրված աղյուսակի համարն է, որով կատարվել է բաշխումը: Յուրաքանչյուր կոդ ունի բազում աղյուսակներ տարբեր շեմային բաշխումների համար:
- **ID_1,ID_2,ID_3,…,ID_n** – գեներացված ID համարները, որոնք միասին կազմում են վերադասավորման գաղտնաբառը:
- **Պարունակություն** - գաղտնի ինֆորմացիայի բաշխված հատվածը (նոր մեթոդով):

Այս տարբերակով բաշխելուց հետո հակառակորդը ունենալով շեմից պակաս քանակով բաղադրամասեր՝ բացարձակ ոչինչ չի կարող իմանալ նախնական գաղտնի ինֆորմացիայի վերաբերյալ: Նա կարող է իմանալ միայն գաղտնի ինֆորմացիայի (ֆայլի) ծավալը, որը գաղտնիք չէ (դա հայտնի է բոլոր մասնակիցներին): Մշակված մեթոդի արագագործությունը ներկայացված է այս գլխի վերջում:

Հարկավոր է նաև նշել, որ խմբավորման աղյուսակները կարելի է ձևափոխել նաև հետևյալ եղանակով. աղյուսակի ցանկացած տողում նշված սյուների համարները կարելի է անկանոն տեղափոխել միմյանց հետ և ստանալ նոր աղյուսակ: Այս փոխատեղումները չեն խախտում աղյուսակի շեմային կառուցվածքը: Այս գործողությունը հնարավորություն է ընձեռում ալգորիթմը կիրառելուց առաջ աղյուսակները ենթարկել նախնական փոփոխության և դրանով հակառակորդի խնդիրը էլ ավելի բարդացնել: Օրինակ, ծրագրային ապահովումը մշակելիս կարելի է այնտեղ ներառել արդեն փոփոխված աղյուսակները: Այդ դեպքում հակառակորդը պետք է նախ և առաջ փորձի վերծանել ծրագրի կոդը, որտեղ կարող է տեսնել փոփոխված աղյուսակները: Յուրաքանչյուր տողում այդպիսի փոխատեղումների քանակը որոշվում է $P = a!$ բանաձևով, որտեղ a-ն կոդի հատվածի երկարությունն է (Օրինակ վերը նշված Հեմմինգի կոդի դեպքում այն հավասար էր $a = 4$): Իսկ ընդհանուր փոխատեղված աղյուսակների քանակը $P = P_1 * P_2 * \dots * P_n$, որտեղ P_i-ն i-րդ

տողի փոխատեղումների քանակն է: Պարզ է, որ $P_1 = P_2 = \cdots = P_n = a!$: Օրինակ, աղյուսակ 4-ի փոխատեղումների ընդհանուր քանակը հավասար է 333776:

2.2. Խմբավորման աղյուսակների ձևավորումը

Վերը նշվեց, որ բաշխման ընթացքում մեծ դերակատարություն ունեն խմբավորման աղյուսակները: Այդ աղյուսակներն են ապահովում նոր մշակված մեթոդի բաշխման պարամետրերը (բաղադրամասերի քանակը և շեմի արժեքը): Աղյուսակի սխալ լինելը կնշանակի, որ կամ շեմից պակաս բաղադրամասերի միավորումով հնարավոր կլինի վերականգնել գաղտնի ինֆորմացիան կամ էլ շեմի արժեքին հավասար քանակի բաղադրամասերի միավորումով չեն ստացվի անհրաժեշտ քանակի այուներ, որից հետո կարող է սխալներ ուղղող կոդը վերականգնել պակասողները: Մեթոդի անխափան աշխատանքը ապահովելու համար անհրաժեշտ է նախապես ձևավորել և ստուգել այդ աղյուսակները: Նկարագրված մեթոդից պարզ է, որ աղյուսակի ձևավորումը կախված է մի քանի պարամետրերից.

- սխալներ ուղղող կոդից, նրա երկարությունից և ուղղող հատկություններից – (m, k, d_{min})
- բաշխման բաղադրամասերի ընդհանուր քանակից - n
- վերականգնման շեմից - t

Ձևավորված աղյուսակները պետք է հաշվի առնեն այս երեք պարամետրերը կամ այլ կերպ ասված բավարարեն դրանց:

Օրինակ դիտարկենք (31, 21, 2) ԲՉՀ սխալներ ուղղող կոդի [46] համար կառուցված խմբավորման աղյուսակները (*Bose–Chaudhuri–Hocquenghem code*): Աղյուսակ 5-ը նախատեսված է այդ կոդով (3, 4) շեմային բաշխում իրականացնելու համար:

Աղյուսակ 5: (31, 21, 2) ԲՉՀ կոդով (3, 4) շեմային կառուցվածք

Մաս 1	23	12	19	13	28	29	31	17	15	9	7	10	14	6	4	5	2	25	8	3	1
Մաս 2	17	22	26	10	20	24	30	4	6	9	1	16	2	18	8	7	3	25	5	28	13
Մաս 3	27	29	13	23	8	3	1	17	6	25	5	14	4	19	20	10	9	26	2	7	31
Մաս 4	2	7	5	10	30	18	16	4	12	9	6	14	15	1	17	3	22	24	26	27	8

Աղյուսակ 6-ով և 7-ով կարելի է կատարել համապատասխանաբար (3,5) և (4,5) շեմային բաշխումներ:

Աղյուսակ 6: (31, 21, 2) ԲՉՀ կոդով (3,5) շեմային կառուցվածք

Մաս 1	28	20	5	10	8	25	1	14	6	9	12	13	4	15	17	7	23	3	2	29	31
Մաս 2	21	7	17	26	19	3	24	4	6	9	12	16	5	18	8	2	1	25	10	29	30
Մաս 3	19	25	5	10	8	13	1	27	30	31	3	14	17	2	20	23	7	26	4	6	9
Մաս 4	16	15	24	28	31	3	18	4	6	27	12	7	2	1	22	23	5	9	10	30	8
Մաս 5	14	22	5	10	8	19	1	24	30	9	12	2	15	16	18	3	21	7	2	25	6

Աղյուսակ 7: (31, 21, 2) ԲՉՀ կոդով (4,5) շեմային կառուցվածք

Մաս 1	17	23	14	18	13	3	19	26	16	9	12	8	5	15	6	2	10	1	7	4	29
Մաս 2	25	15	5	10	23	17	1	4	22	9	12	13	7	16	3	18	6	8	2	28	30
Մաս 3	15	7	21	17	8	3	27	19	6	23	13	2	16	10	18	4	5	9	24	25	1
Մաս 4	18	14	16	10	26	21	1	4	15	9	11	13	7	6	5	17	2	3	23	8	28
Մաս 5	27	29	30	10	8	3	1	4	6	9	11	13	14	15	16	17	22	24	2	7	5

Աղյուսակ 8-11-ում ներկայացված են (12,8,2) Ռիդ-Սոլոմոնի սխալներ ուղղող կոդի համար ձևավորված որոշ շեմային կառուցվածքների խմբավորման աղյուսակները:

Աղյուսակ 8: (12,8,2) Ռիդ-Սոլոմոնի կոդով (2,4) շեմային կառուցվածք

Մաս 1	**8**	**1**	**10**	**1**	**9**	**4**	**7**	**6**
Մաս 2	**4**	**8**	**11**	**1**	**12**	**2**	**6**	**7**
Մաս 3	**6**	**11**	**1**	**9**	**2**	**10**	**4**	**12**
Մաս 4	**7**	**4**	**11**	**6**	**12**	**8**	**10**	**9**

Աղյուսակ 9: (12,8,2) Ռիդ-Սոլոմոնի կոդով (2,5) շեմային կառուցվածք

Մաս 1	**7**	**1**	**10**	**4**	**9**	**2**	**6**	**8**
Մաս 2	**8**	**2**	**11**	**1**	**12**	**7**	**6**	**4**
Մաս 3	**12**	**1**	**9**	**4**	**2**	**11**	**10**	**8**
Մաս 4	**11**	**6**	**8**	**1**	**12**	**7**	**9**	**10**
Մաս 5	**9**	**2**	**12**	**4**	**11**	**7**	**6**	**10**

Աղյուսակ 10: (12,8,2) Ռիդ-Սոլոմոնի կոդով (3,4) շեմային կառուցվածք

Մաս 1	**8**	**1**	**10**	**1**	**9**	**4**	**7**	**6**
Մաս 2	**4**	**8**	**11**	**1**	**9**	**2**	**6**	**7**
Մաս 3	**6**	**8**	**1**	**7**	**2**	**10**	**4**	**12**
Մաս 4	**7**	**4**	**11**	**6**	**12**	**8**	**2**	**1**

Աղյուսակ 11: (12,8,2) Ռիդ-Սոլոմոնի կոդով (3,5) շեմային կառուցվածք

Մաս 1	**7**	**1**	**10**	**4**	**9**	**2**	**6**	**8**
Մաս 2	**9**	**2**	**11**	**1**	**10**	**7**	**6**	**4**
Մաս 3	**12**	**1**	**9**	**4**	**22**	**7**	**10**	**6**
Մաս 4	**11**	**2**	**8**	**1**	**12**	**4**	**6**	**10**
Մաս 5	**4**	**2**	**12**	**1**	**11**	**7**	**8**	**10**

Աղյուսակ 2-11-ը ուսումնասիրելով տեսնում ենք, որ նույն շեմային կառուցվածքների համար ունենք մի քանի աղյուսակ (տարբեր կոդերի համար) և տրամաբանական է, որ անհրաժեշտ է այդ աղյուսակները համեմատել ըստ արագագործության և կատարել դասակարգում: Արագագործության տեսանկյունից ավելի նախընտրելի է ունենալ ամենարագ խմբավորման աղյուսակները, իսկ մյուս կրկնությունները (որոնք համեմատած ավելի դանդաղ են) պարզապես անտեսել: Սակայն մյուս կողմից, եթե արագագործությունը մի պահ դիտարկենք երկրորդ դերում, ապա հակառակորդի աշխատանքը կբարդանա, եթե միևնույն շեմային կառուցվածքի համար ունենանք տարբեր կոդեր և տարբեր աղյուսակներ: Այն դեպքում, երբ արագագործության խնդիրը շատ սուր չէ դրված, բաշխման ժամանակ կարելի է պատահականորեն ընտրել և օգտագործել այդ շեմային կառուցվածքը ապահովող աղյուսակներից մեկը: Քանզի կոդի և աղյուսակի համարները գաղտնի են պահվում, ապա հակառակորդի խնդիրը էլ ավելի է բարդանում: Նա առաջին հերթին պետք է հասկանա, թե որ աղյուսակն է ընտրվել բաշխման համար: Նույնիսկ այն դեպքերում երբ արագագործության նկատառումից ելնելով ընտրվել է ամենարագ բաշխում ապահովող աղյուսակը, այնուամենայնիվ հակառակորդի մոտ չկա վստահություն, որ կիրառվել է հենց այդ աղյուսակը և նա ստիպված է դիտարկել նաև մյուս աղյուսակները: Այս մոտեցման շնորհիվ ստանում ենք հետևյալ առավելությունները.

- տվյալ շեմային կառուցվածքի համար ամենարագ բաշխումը,
- հակառակորդի խնդրի բարդացում:

Այդ ամենի համար, արդեն իսկ ձևավորված աղյուսակները համեմատվում են ըստ արագագործության, որի արդյունքում ստեղծվում է Նկ. 11-ում պատկերված զանգվածի նման զանգված: Բաշխման ծրագիրը ստանալով բաշխման պարամետրերը, առաջին հերթին դիտարկում է այս զանգվածը, որից հետո կախված առաջադրված խնդրից (արդյոք հարկավոր է ամենարագ բաշխումը, թե ոչ) ընտրում է, թե որ աղյուսակով է կատարվելու բաշխումը:

Բաղադրամաս	Շեմ	Կոդ	Աղյուսակ	Արագություն
3	4	Hamming	3	1
3	4	BCH	1	2
3	5	Reed-Solomon	6	1
3	5	BCH	2	2
4	6	Reed-Muller	2	1
4	6	Hamming	5	2
…	…	…	…	…

Նկ. 11. Խմբավորման աղյուսակների արագագործությունների զանգված

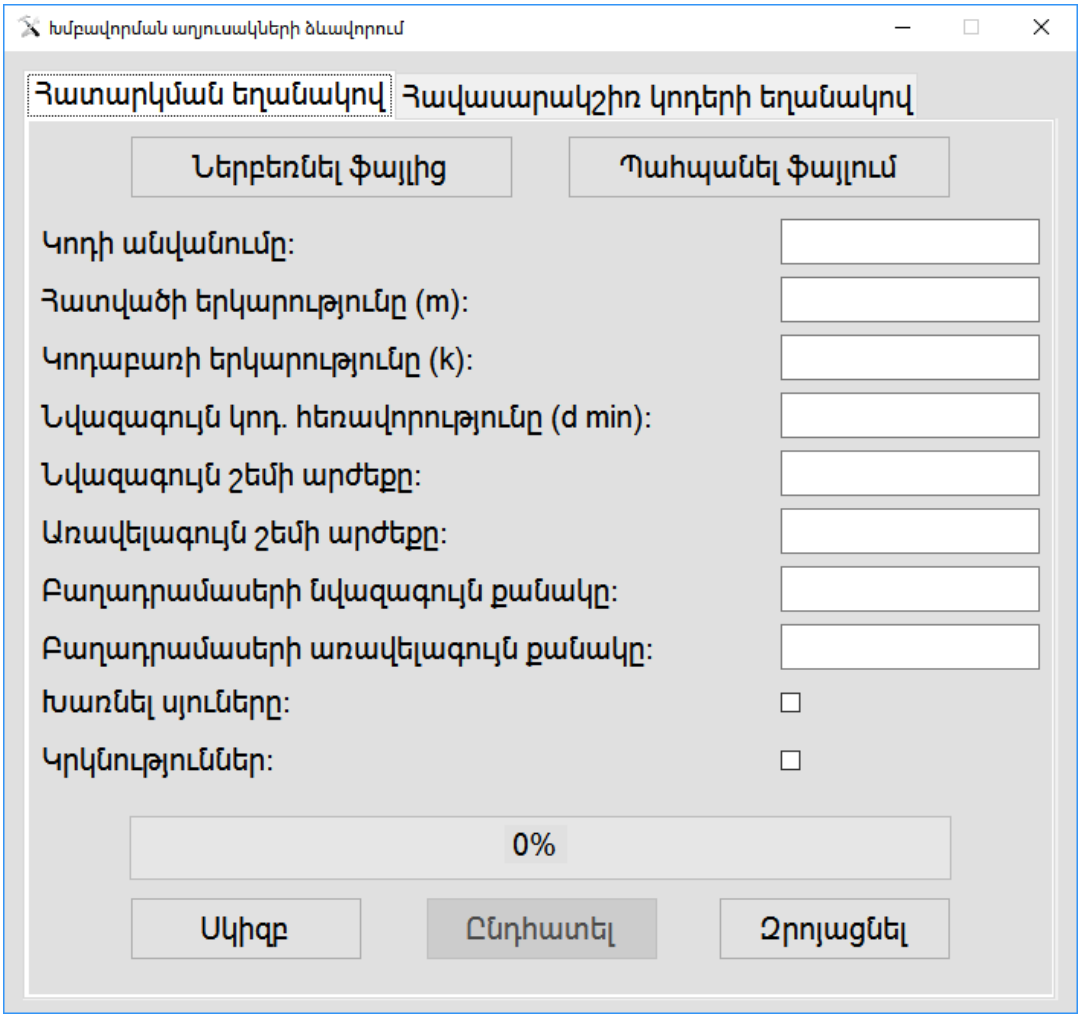
Այժմ դիտարկենք, թե ինչ եղանակով են ձևավորվում այդ բաշխման աղյուսակները: Պարզ է, որ այդ աղյուսակները կարելի է կառուցել պարզապես տրամաբանությունից ելնելով, որոշակի հատարկումներ անելով և ստուգելով աղյուսակի համապատասխանությունը տվյալ շեմային կառուցվածքին: Մյուս կողմից հաշվի առնելով, որ մշակված ալգորիթմը ճկուն է և հնարավորություն է տալիս ցանկացած պահի ավելացնել այլ սխալներ ուղղող կոդեր նույնպես, ապա ստացվում է, որ ամեն անգամ հարկավոր է երկար հատարկումներ կատարել և դուրս բերել տվյալ կոդի համար բաշխման աղյուսակներ: Երկար կոդաբառ ունեցող կոդերի դեպքում այդ գործընթացը շատ ժամանակատար կլինի:

Ընդհանուր դեպքում, եթե ունենք (m, k, d_{min}) սխալներ ուղղող կոդ, ապա հնարավոր բաղադրամասերի քանակը կլինի՝

$$C_m^k = \frac{m!}{k! \cdot (m-k)!}$$

Հատարկումը անհրաժեշտ է կատարել այդքան բաղադրամասերից՝ ստանալով այնպիսի բաղադրամասերի խումբ, որը կբավարարի տվյալ շեմային կառուցվածքին:

Մյուս կողմից շատ ավելի տրամաբանական կլինի այդ աղյուսակների ձևավորման համար ունենալ ծրագրային լուծում, որը կլինի համընդհանուր և կկարողանա ավտոմատացված կարգով հաշվարկել և դուրս բերել խմբավորման աղյուսակներ տրված պարամետրերով կոդի համար՝ հաշվի առնելով անհրաժեշտ շեմային կառուցվածքի տվյալները (բաղադրամասերի քանակը և շեմի արժեքը): Այդ նպատակի համար մշակվել է **«Խմբավորման աղյուսակների ձևավորում»** ծրագրային ապահովումը: Դիտարկենք այդ ծրագրի աշխատանքը: Նկ. 12-ում պատկերված է այդ ծրագրի ինտերֆեյսի տեսքը գործարկումից անմիջապես հետո:



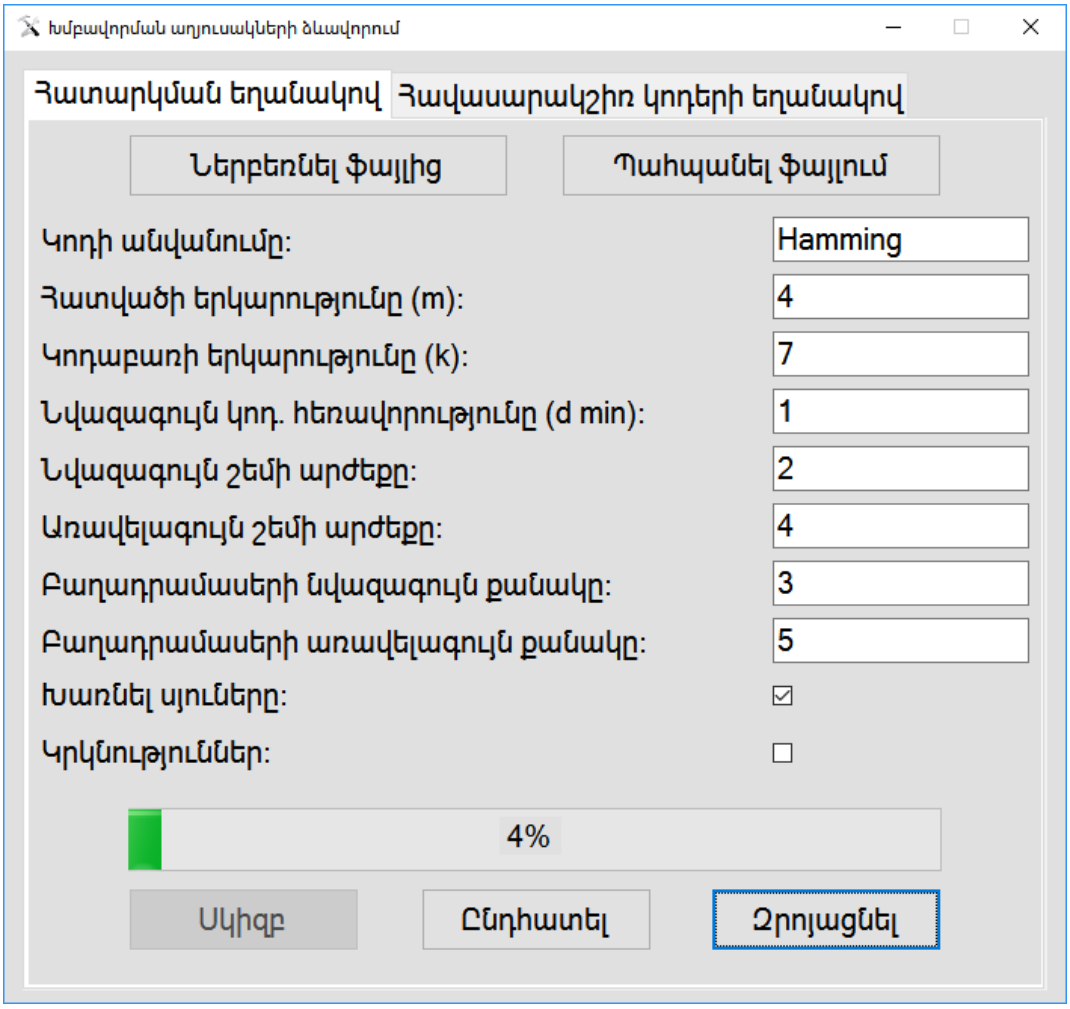
Նկ. 12. «Խմբավորման աղյուսակների ձևավորում» ծրագրի տեսքը գործարկումից հետո

Ստորև ներկայացված է յուրաքանչյուր դաշտի նկարագրությունը և գործառույթները.

- **Կոդի անվանումը** – ներմուծվում է այն կոդի անվանումը, որի համար գեներացվում է աղյուսակը: Այս տվյալը կցվում է գեներացված աղյուսակներին և ունի ինֆորմացիոն բնույթ:
- **Հատվածի երկարությունը (m)** – կոդավորվող ինֆորմացիայի երկարությունը (մինչև կոդավորելը):
- **Կոդաբառի երկարությունը (k)** – կոդաբառի երկարությունը:
- **Նվազագույն կոդ. հեռավորությունը (d_{min})** – կոդի նվազագույն հեռավորությունը (Հեմմինգի հեռավորություն):
- **Նվազագույն շեմի արժեքը** – ձևավորվող աղյուսակը նվազագույնը ինչ շեմային բաշխման համար պետք է լինի:
- **Առավելագույն շեմի արժեքը** – առավելագույնը ի՞նչ շեմային բաշխման համար պետք է լինի
- **Բաղադրամասերի նվազագույն քանակը** – բաղադրամասերի նվազագույն քանակը:
- **Բաղադրամասերի առավելագույն քանակը** – բաղադրամասերի առավելագույն քանակը:
- **Խառնել սյուները** – ձևավորվող աղյուսակում կատարել սյուների տեղափոխումներ («խառնել»), թե ոչ:
- **Կրկնություններ** – թույլատրվում է միևնույն սյունը մեկից ավել անգամ օգտագործել տվյալ բաղադրամասում, թե ոչ:
- **Սկիզբ** – գործարկել ծրագիրը:
- **Ընդհատել** – ժամանակավոր ընդհատել ծրագրի աշխատանքը:
- **Ներբեռնել ֆայլից** – ներբերնել ծրագրի կարգավորումները ֆայլից:

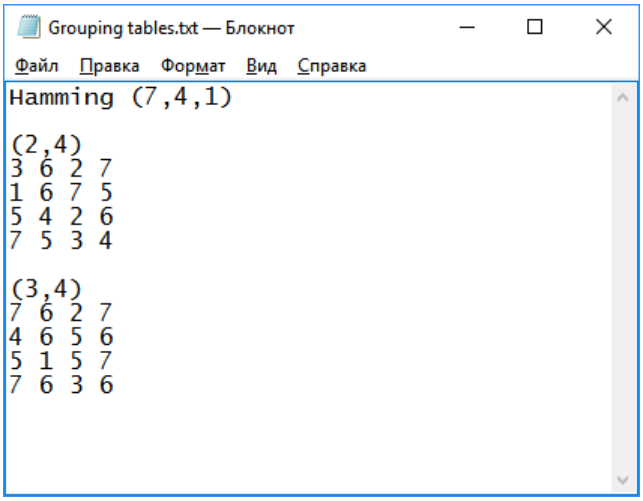
- **Պահպանել ֆայլում** – պահպանել ծրագրի կարգավորումները ֆայլում (հնարավորություն է ընձեռում պահպանել և հետո շարունակել աշխատանքը ընդհատված տեղից):
- **Զրոյացնել** – նախատեսված է բոլոր դաշտերը նախնական տեսքի բերելու համար:

Նկ. 13-ում պատկերված է ծրագրիի աշխատանքի օրինակ: Ինչպես երևում է, գեներացվում են աղյուսակներ (7, 4, 1) Հեմմինգի կոդի համար: Շեմային սահմանը 2-4, իսկ բաղադրամասերի քանակը 3-5 սահմանում: Թույլատրված է նաև սյուների տեղափոխությունը:



Նկ. 13. «Խմբավորման աղյուսակների ձևավորում» ծրագրի աշխատանքի օրինակ

Նկ.14-ում պատկերված է ծրագրի որոշ ժամանակ աշխատանքի արդյունքում ստացված աղյուսակների նախնական տեսքը:



Նկ. 14. Հեմմինգի կոդի համար գեներացված աղյուսակներ

Մշակված ծրագրի աշխատանքը հիմնված է որոշակի կոմբինատոր հնարքների և հատարկման վրա: Այլ կերպ ասած ծրագիրը, որպես մուտքային տվյալներ ընդունում է սխալներ ուղղող կոդի և բաշխման պարամետրերը և հատարկման եղանակով փորձում է ստանալ այդ պայմաններին բավարարող խմբավորման աղյուսակներ: Ինչպես և մյուս բաշխման ալգորիթմներում, այս ալգորիթմի համար անհրաժեշտ աղյուսակների գեներացումը կատարվում է հետևյալ սկզբունքով. որոշ քանակի բաղադրամասեր վերցվում են պատահականորեն, իսկ մյուսները հաշվարկվում դրանցից կախված, այնպես, որ բավարարվեն շեմային պայմանները: Հատարկման ընթացքում ստացված հերթական աղյուսակը հարկավոր է ստուգել որոշակի պայմանների բավարարում է, թե ոչ: Նրանից կարևորները թվարկված են ստորև.

- ստացված աղյուսակում շեմից պակաս քանակի բաղադրամասերի ոչ մի համակցություն հնարավորություն չնձեռի ստանալ այնքան սյուն, որոնցով հնարավոր կլինի գաղտնիքի վերականգնում կատարել:
- շեմի քանակով բաղադրամասերի բոլոր համակցություններով հնարավոր լինի ստանալ այնքան բաղադրամաս, որ սխալներ ուղղող կոդը ճշգրիտ վերականգնի պակասող սյուները:

Այս երկու գլխավոր պայմաններին պետք է բավարարի ձևավորված աղյուսակը: Եթե այն չի բավարարում այս պայմաններից առնվազն մեկին, ապա այդ աղյուսակը անտեսվում է և հատարկումը շարունակվում է:

Այս գործընթացը բավականին ժամանակատար է և պարզ է, որ կոդաբառի երկարության, բաղադրամասերի և շեմի արժեքների մեծացմանը զուգահեռ այն ավելի է բարդանում: Սակայն այս գործողությունը չի մասնակցում բաշխման և վերականգնման ընթացքին: Այն կատարվում է մեկ անգամ, որից հետո ստացված աղյուսակները ներմուծվում են ալգորիթմի մեջ և կիրառվում բազմակի անգամներ: Գործընթացի հեշտացման համար կարելի է ստացված աղյուսակի հիման վրա ստանալ այլ աղյուսակ, դրանում ավելացնելով կամ պակասեցնելով բաղադրամաս: Օրինակ (3,5)-ից հնարավոր է (3,6)-ի ստացում: Դրա համար անհրաժեշտ է հատարկման եղանակով ավելացնել վեցերորդ բաղադրամասը և ստուգել աղյուսակը ըստ վերը նշված պայմանների: Նույն եղանակով հնարավոր է (3,5)-ից (3,4)-ի արագ ստացում:

Հատկանշական է, որ այս տարբերակով կարելի է գեներացնել ոչ միայն շեմային մուտքի կառուցվածք ապահովող աղյուսակներ, այլ նաև շեմայինից տարբեր մուտքի կառուցվածքներ: Օրինակ հնարավոր է ստանալ բաղադրամասեր, որոնք կունենան ավելի մեծ կշիռ և վերականգնման համար կպահանջվի ավելի քիչ բաղադրամասեր: Օրինակ տնօրենի համար նախատեսված բաղադրամասը կարող է ունենալ երկու կշիռ, իսկ մյուս բաղադրամասերը մեկ: Այդպիսի աղյուսակներ կարելի է ձևավորել **«Խմբավորման աղյուսակների ձևավորում»** ծրագրի օգնությամբ՝ նախապես փոփոխության ենթարկվելով ծրագրում աղյուսակների ստուգման պայմանները:

Աղյուսակների ձևավորման վերը նկարագրված մեթոդը հնարավորություն է տալիս ստանալ աղյուսակը կոնկրետ կոդի համար: Աղյուսակների ձևավորման մյուս տարբերակը կայանում է հետևյալում: Եթե առաջին տարբերակում կախված կոդի պարամետրերից ստացվում էր որոշակի շեմային կառուցվածք ապահովող աղյուսակ, ապա այս մեթոդի դեպքում գործողությունների տրամաբանությունը ճիշտ հակառակն է:

Այս մեթոդի ժամանակ, շեմային կառուցվածքից կախված ձևավորվում է աղյուսակ, որից հետո գտնվում է այն կոդի պարամետրերը, որին կբավարարի այդ աղյուսակը: Այս դեպքում աղյուսակի հիմքում ընկած են հավասարակշիռ կոդերը [20]: Ինչպես գիտենք, հավասարակշիռ են կոչվում այն կոդերը, որոնց բոլոր կոդաբառերի Հեմմինգի կշիռը նույնն է: Օրինակ $R(5,3)$ հավասարակշիռ կոդը, դա 5 երկարության այն երկուական բառերի համախումբն է, որոնցում մեկերի քանակը երեք է, իսկ զրոների քանակը երկու:

Ընդհանուր դեպքում $R(n,q)$ հավասարակշիռ կոդում կոդաբառերի քանակը որոշվում է (1)-ում տրված բանաձևով:

$$C_n^q = \frac{n!}{(n-q)!\cdot q!} \quad (1)$$

Օրինակ, $R(5,3)$-ի դեպքում ստացվում է $\frac{5!}{2!\cdot 3!} = 10$ կոդաբառ:

Այժմ դիտարկենք մի օրինակ, որը հնարավորություն է տալիս հավասարակշիռ կոդի օգտագործմամբ ստանալ շեմային կառուցվածք: Դիցուք անհրաժեշտ է $(3,5)$ շեմային կառուցվածք $(n=5, h=3)$: Կառուցենք երեք կշռով հավասարակշիռ հինգ կարգանի կոդաբառերի աղյուսակը: Ըստ (1)-ի այդ կոդաբառերի քանակը $C_n^q = 10$: Աղյուսակ 12-ում պատկերված է այդ կոդաբառերի համախումբը:

Աղյուսակ 12: $R(5,3)$ հավասարակշիռ կոդի կոդաբառերի աղյուսակ

	1	2	3	4	5
1)	1	1	1	0	0
2)	1	1	0	1	0
3)	1	1	0	0	1
4)	1	0	1	1	0
5)	1	0	1	0	1
6)	1	0	0	1	1
7)	0	1	1	1	0
8)	0	1	1	0	1
9)	0	1	0	1	1
10)	0	0	1	1	1

Այժմ, եթե բաշխման ալգորիթմում որպես բաղադրամասերի համարներ դիտարկենք աղյուսակ 12-ում տողերի համարները (1), 2), 3) և այլն), իսկ սյուների համարները որպես ձևավորվող բաղադրամասերի համարներ (որոնք տրվելու են կողմերին), ապա դժվար չէ նկատել, որ կամայական երեք սյան միավորումից ստացվում են տաս հատ սյուն, իսկ կամայական երկուսի միավորումից տասից պակաս: Եթե մի պահ պատկերացնենք, որ տաս սյունը բավարար է գաղտնիքի վերականգնման համար, ապա կարող ենք պնդել, որ նկարագրված աղյուսակը ապահովում է (3,5) շեմային կառուցվածք: Վերջնական ձևավորված աղյուսակը պատկերված է աղյուսակ 13-ում: Ինչպես տեսնում ենք, ըստ աղյուսակ 13-ի յուրաքանչյուր կողմ ստանում է ճիշտ վեց բաղադրամաս:

Աղյուսակ 13: (3,4) շեմային բաշխման աղյուսակ

Մաս 1	**1**	**2**	**3**	**4**	**5**	**6**
Մաս 2	**1**	**2**	**3**	**7**	**8**	**9**
Մաս 3	**1**	**4**	**5**	**7**	**8**	**10**
Մաս 4	**2**	**4**	**6**	**7**	**9**	**10**
Մաս 5	**3**	**5**	**6**	**8**	**9**	**10**

Վերը նկարագրվածից կարող ենք եզրակացություն կատարել, որ այդ աղյուսակը կարող է բավարարել $(6,10+t,t)$ տեսքի երկուական սխալներ ուղղող կոդի, որտեղ t-ն կոդի սխալներ ուղղելու քանակն է: Այսինքն, որպես մուտքային ինֆորմացիա կլինի վեց բիթանոց հատված, որը կոդավորելուց հետո կդառնա $10+t$ բիթ երկարությամբ կոդաբառ և կոդը կարող է ուղղել t քանակի սխալ: Աղյուսակ 13-ն էլ կլինի այդ կոդի համար (3,5) շեմային բաշխման աղյուսակը: Այսինքն կամայական երեք բաղադրամասի միավորումից կհավաքվի տաս բաղադրամաս, իսկ պակասող t հատ սյունը կուղղենք կոդի ապակոդավորման ալգորիթմի օգնությամբ:

Վերը նկարագրված մեթոդը ընդհանուր տեսքի բերելու համար ձևակերպենք հետևյալ թեորեմը:

Թեորեմ 1: Եթե բաղադրամասերը (սյուները) բաշխված են n թվով կողմերի միջև և անհրաժեշտ է լուծել խնդիրը h շեմի դեպքում, ապա բաղադրամասերի քանակը հավասար կլինի $R(n,n-h+1)$ հավասարակշիռ կոդի կոդային համակցությունների թվին: Այսինքն՝

$$K=P(h-1,n-h+1)=\frac{n!}{(h-1)!\cdot(n-h+1)} \quad (2)$$

Բերենք մեկ օրինակ, որից հետո կապացուցենք թեորեմը:

Օրինակ $n=6,h=2$: Կառուցենք հավասարակշիռ երկուական կոդ $R(6,5)$: Աղյուսակ 14:

Աղյուսակ 14: $R(6,5)$ հավասարակշիռ կոդի կոդաբառերի աղյուսակ

	1	2	3	4	5	6
1)	1	1	1	1	1	0
2)	1	1	1	1	0	1
3)	1	1	1	0	1	1
4)	1	1	0	1	1	1
5)	1	0	1	1	1	1
6)	0	1	1	1	1	1

Ցանկացած երկու սյուների փոխհաջորդմամբ (միավորմամբ) ստացվում է գաղտնիքը: Այսինքն այս աղյուսակը ապահովում է (2,6) շեմային կառուցվածք և դժվար չէ նաև գտնել այն կոդը, որի դեպքում այդ աղյուսակը կլինի (2,6) շեմային բաշխում ապահովողը:

Այժմ ներկայացնենք թեորեմի ապացույցը: $R(n,q)$-ից դիտարկենք i-րդ սյունը և այն նշանակենք՝ r(i): Նրա կշիռը որոշվում է կոդում մեկերի քանակով և նշանակենք այն $|r(i)|$: Որոշենք r(i) և r(j) սյուների փոխհաջորդումների արդյունքը վեկտորի տեսքով: Վեկտորի չափը որոշվում է նրա բաղադրիչներով, որոնք ստացվել են r(i) և r(j) վեկտորների բաղադրիչների փոխհաջորդումների դիզունկցիայի ճանապարհով: Հետևաբար s վեկտորների նկատմամբ կիրառենք փոխհաջորդում (դիզունկցիա-«v»)

$$r(i_1)\ v\ r(i_2)\ v\ r(i_3)\ v \dots v\ r(i_s)$$

Դիտարկենք n երկարությամբ, q կշռով կոդերի ամբողջ աղյուսակը, որը նշանակել ենք $R(n,q)$: Այս աղյուսակից ընտրենք կամայական i-րդ սյուն, այսինքն՝ $r(i_1)$: Որոշենք դրա կշիռը: Հեշտ է նկատել, որ i -ի ցանկացած արժեքի դեպքում՝

$$|r(i)| = P(n-q, q-1) = C_{n-q}^{q-1} \quad (3)$$

Կամայական երկու սյուների (i-րդ և j-րդ) Փոխհաջորդումների կողի կշիռը՝

$$|r(i)\ \mathrm{v}\ \mathrm{r(j)}| = \mathrm{C}_{\mathrm{n}-1}^{\mathrm{q}-1} + \mathrm{C}_{\mathrm{n}-2}^{\mathrm{q}-1} \quad (4)$$

Կամայական s հատ սյուների (i_1-րդ, i_2-րդ, i_3-րդ…i_s-րդ) փոխհաջորդումների կողի կշիռը, $1 < s < n - q + 1$ դեպքում հետևյալն է՝

$$\mathrm{r}(i_1)\ \mathrm{v}\ \mathrm{r}(i_2)\ \mathrm{v}\ \mathrm{r}(i_3)\ \mathrm{v} \ldots \mathrm{v}\ \mathrm{r}(i_\mathrm{s}) = \mathrm{C}_{\mathrm{n}-1}^{\mathrm{q}-1} + \mathrm{C}_{\mathrm{n}-2}^{\mathrm{q}-1} + \mathrm{C}_{\mathrm{n}-3}^{\mathrm{q}-1} + \cdots + \mathrm{C}_{\mathrm{q}-1}^{\mathrm{q}-1} \quad (5)$$

Եթե $\mathrm{s} = \mathrm{n} - \mathrm{q} + 1$, ապա բանաձևում գումարը հավասար է $\mathrm{R(n, q)}$–ում կողերի քանակին, այսինքն՝

$$C_n^q = \frac{n!}{q! \cdot (n-q)!} \quad (6)$$

Իսկապես, (5) բանաձևում վերջին անդամը հավասար է մեկի: Եթե (5)-ում S անդամների քանակը փոքր է $\mathrm{n} - \mathrm{q} + 1$, ապա (5)-ի գումարը փոքր է $P(n - q, q)$ –ի արժեքից, որն էլ ապացուցում է թեորեմը, որովհետև եթե մասնակիցների քանակը փոքր է, քան $\mathrm{n} - \mathrm{q} + 1$, ապա նրանք գաղտնի հաղորդագրությունը վերծանելու համար չեն ունենա բաղադրամասերի ամբողջական հավաքածուն:

Ընդհանրացնելով վերը նկարագրված մեթոդը, կարող ենք ասել, որ այս տարբերակով (h, n) աղյուսակի ձևավորման համար անհրաժեշտ է կառուցել $R(n, n - h + 1)$ հավասարակշիռ կողի կողաբառերի աղյուսակը և այդ աղյուսակի հիման վրա ստանալ բաշխման աղյուսակը: Ստացված աղյուսակին բավարարող կողը կունենա հետևյալ տեսքը՝ $(\mathrm{m}, \mathrm{k}, \mathrm{t})$, որտեղ՝

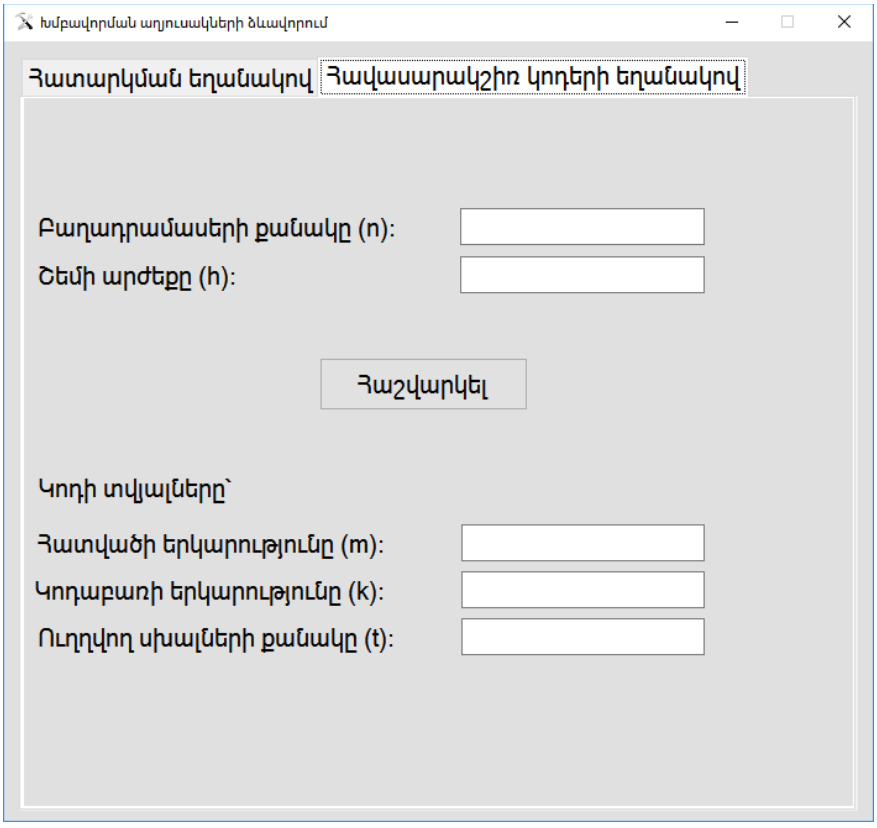
$$m = \frac{n!}{(h-1)! \cdot (n-h+1)!} + h - n + 1$$

$$k = \frac{n!}{(h-1)! \cdot (n-h+1)!} + t$$

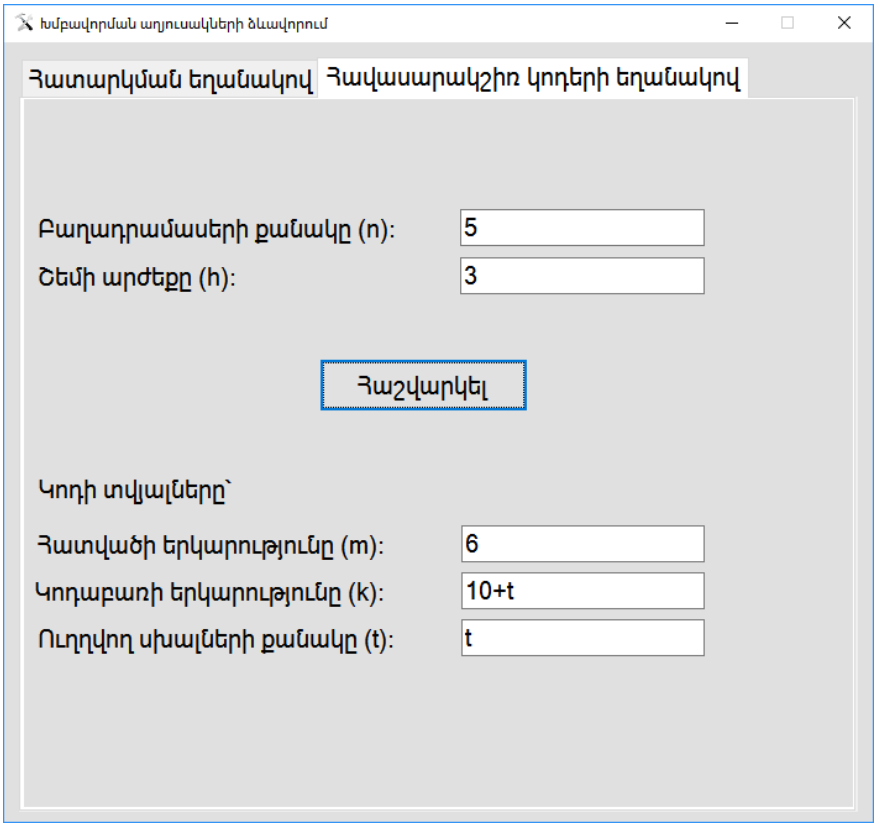
$$t = 1,2,3 \ldots$$

Այս տարբերակով աղյուսակների ձևավորումը շատ արագ է կատարվում: Այն ավելի նպատակահարմար է երկար կոդերի համար, քանզի կարճ կոդաբառերի դեպքում հատարկման եղանակով հնարավոր է արագ ստանալ շեմային կառուցվածքներին բավարարող աղյուսակները:

Այս մեթոդի հիման վրա մշակվել է ծրագիր, որը հնարավորություն է տալիս տրված (h,n) շեմային կառուցվածքի համար ստանալ բաշխման աղյուսակը և այն կոդի պարամետրերը, որին բավարարում է այդ աղյուսակը: «Խմբավորման աղյուսակների ձևավորում» ծրագրի երկրորդ հատվածը նախատեսված է այդ մեթոդով աղյուսակների ձևավորման համար: Նկ.15-ում պատկերված է այդ ծրագրի տեսքը գործարկումից հետո: Նկ.16-ում պատկերված է այդ ծրագրի աշխատանքի օրինակ ((3,5) շեմային կառուցվածքի համար):



Նկ. 15. Աղյուսակի և կոդի ձևավորման ծրագրի տեսքը



Նկ. 16. Աղյուսակի և կոդի ձևավորման ծրագրի աշխատանքի օրինակ

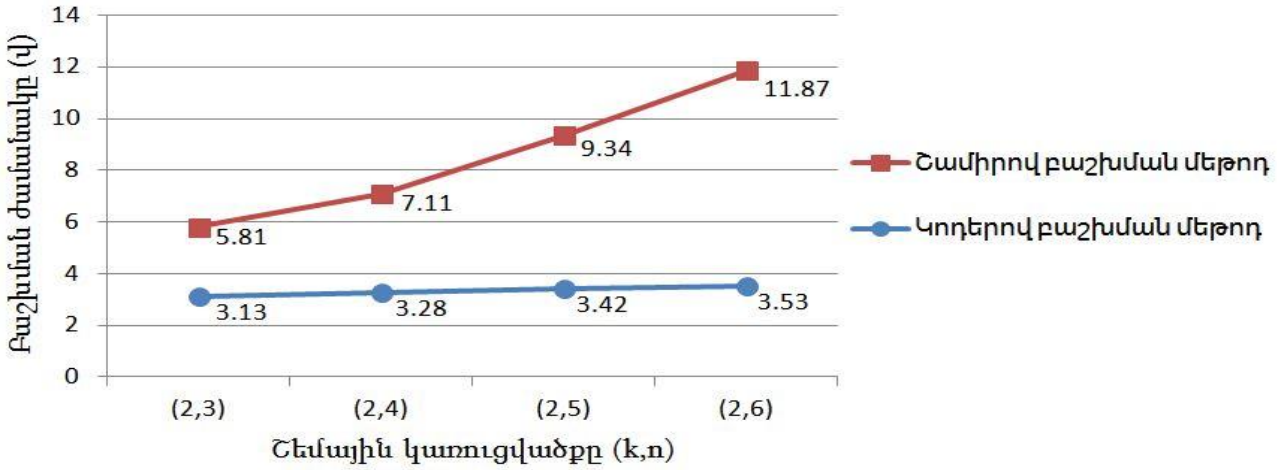
2.3. Մշակված մեթոդի արագագործության գնահատականը

Խնդրի արդիականությունը նկարագրելիս նշվեց, որ անհրաժեշտություն է առաջացել ունենալ ավելի արագագործ գաղտնիքի բաշխման մեթոդ: Ատենախոսության հիմնական նպատակներից մեկը հենց արագագործ գաղտնիքի բաշխման մեթոդի մշակումն էր: Այդ իսկ պատճառով անհրաժեշտ է կատարել մշակված մեթոդի արագագործության գնահատական: Հասկանալ, թե որ դեպքերում է այն ավելի նպատահարմար կիրառել: Արագագործության գնահատականներ անհրաժեշտ են ոչ միայն բաշխման և վերականգնման գործողությունների, այլ նաև բաղադրամասի իսկության ստուգման և կորած կամ վնասված բաղադրամասի վերականգնման գործողությունների համար: Անհրաժեշտ է նաև համեմատել մշակված մեթոդը գոյություն ունեցող մեթոդների հետ: Այստեղ նպատահարմար է համեմատությունը կատարել Շամիրի շեմային մեթոդի հետ, քանզի այդ մեթոդը ամենատարածվածն է և գործնականում բոլոր նոր ստեղծվող բաշխման մեթոդները համեմատվում են այդ մեթոդի հետ: Ունենալով համեմատությունը Շամիրի մեթոդի հետ, կարելի է համարել, որ այն համեմատվել է նաև մյուսների հետ:

Նկ. 17-ում պատկերված են մշակված մեթոդի և Շամիրի մեթոդի բաշխման արագագործությունների գրաֆիկները, այն դեպքում, երբ շեմի արժեքը հաստատուն է, իսկ բաղադրամասերի քանակը փոփոխվում է: Տրված է բաշխման ժամանակի կախվածությունը շեմային կառուցվածքից: Կոդերով բաշխման բոլոր շեմային կառուցվածքներում օգտագործված է Հեմմինգի նույն կոդը: Բաշխման համար ընտրված է 10Mb ծավալով ֆայլ: Փորձերը կատարվել են հետևյալ տվյալներով համակարգչով՝

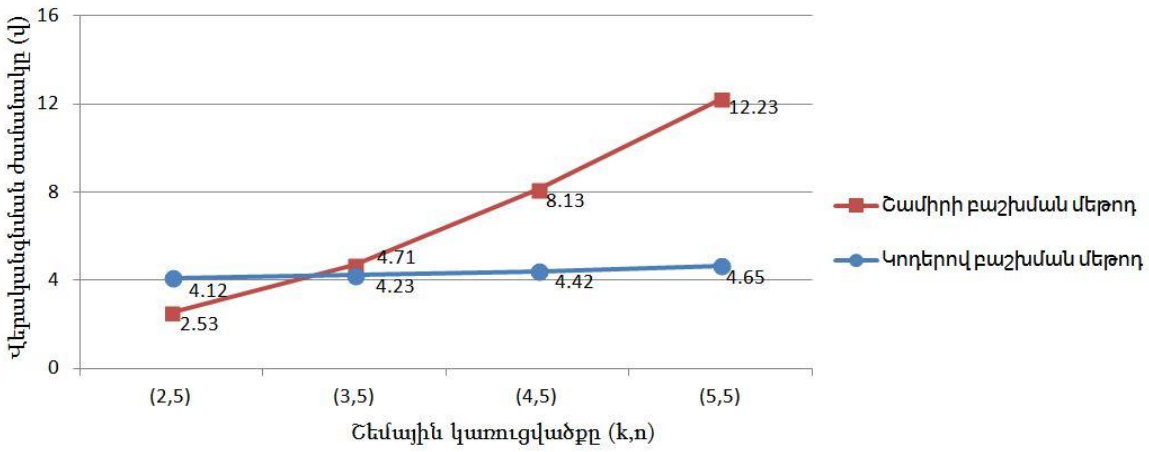
- CPU - Intel Core i3 2.00Ghz,
- RAM - 2GB,
- HDD – 7200prm:

Գրաֆիկից երևում է, որ շեմի հաստատուն արժեքի դեպքում, բաղադրամասերի քանակի աճը քիչ ազդեցություն է թողնում մշակված մեթոդի արագագործության վրա: Դա պայմանավորված է նրանով, որ հիմնական ժամանակատար գործողությունը (կոդավորումը) կատարվում է նույն ժամանակում (այդ գործողության ժամանակը կախված է միայն գաղտնի ինֆորմացիայի ծավալից): Տարբերությունը հիմնականում պայմանավորված է ստեղծվող ֆայլերի քանակի աճով: Գրաֆիկների համեմատումից երևում է, որ մշակված մեթոդն ավելի արագագործ է և բաղադրամասերի քանակի աճին զուգահեռ ժամանակի աճը ավելի փոքր է, քան Շամիրի մեթոդում:



Նկ. 17. Մշակված մեթոդի և Շամիրի մեթոդի գաղտնիքի բաշխման արագործությունների գրաֆիկները՝ շեմի հաստատուն արժեքի դեպքում (t=2,n=3,4,5,6)

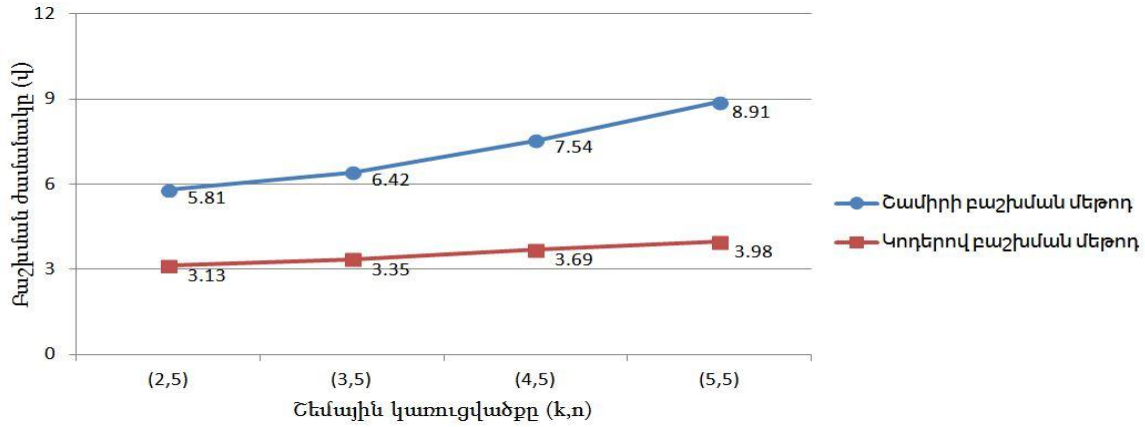
Նույն տրամաբանությամբ հարկավոր է տալ մշակված մեթոդի և Շամիրի մեթոդի գաղտնիքի վերականգնման գործողությունների ժամանակային գրաֆիկները: Նկ.18-ում պատկերված է նոր մեթոդի և Շամիրի մեթոդի գաղտնիքի վերականգնման ժամանակային գրաֆիկները: Ինչպես երևում է գրաֆիկների համեմատումից, բաղադրամասերի քանակի աճն ավելի քիչ ազդեցություն է թողնում վերականգման ժամանակի վրա, քան Շամիրի մեթոդով վերականգնման դեպքում:



Նկ. 18. Մշակված մեթոդի և Շամիրի մեթոդի գաղտնիքի վերականգնման արագործությունների գրաֆիկները՝ շեմի հաստատուն արժեքի դեպքում (t=2,n=3,4,5,6)

Շատ ավելի հետաքրքիր արդյունք է ստացվում այն դեպքում, երբ հաստատուն է պահվում բաղադրամասերի քանակը և ավելացվում շեմի արժեքը : Պարզվում է, որ մշակված մեթոդն այս դեպքում գրեթե արագագործության անկում չի ունենում (աննկատ աճ գոյություն ունի), իսկ Շամիրի մեթոդի դեպքում աճը ունի էքսպոնենցիալ տեսք [44]: Դա ավելի պատկերավոր երևում է գրաֆիկներում: Նկ.19-ում պատկերված է մշակված մեթոդի և Շամիրի մեթոդի բաշխման ժամանակային գրաֆիկները՝ բաղադրամասերի հաստատուն քանակի դեպքում: Այս արդյունքը հետևանք է այն բանի, որ շեմային տարբեր կառուցվածքները ստացվում են աղյուսակների շնորհիվ, իսկ հիմնական ժամանակատար գործողությունը (կոդավորումը) բոլոր դեպքերում կատարվում է նույն ժամանակում: Համակարգչի համար արագագործության տեսանկյունից “տարբերություն չկա”, թե որ սյունը որ բաղադրամասում (ֆայլում) կտեղադրի: Իսկ ժամանակային չնչին տարբերությունը պայմանավորված է նրանով, որ մշակված մեթոդում գլխամասային կառուցվածում օգտագործվում է Շամիրի մեթոդով բաշխում: Բաշխվող ֆայլի ծավալի մեծացմանը զուգահեռ այդ տարբերությունն ավելի կնվազի, քանզի ընդհանուր բաշխման ժամանակի մեջ ավելի

քիչ տոկոս կկազմի Շամիրի մեթոդով բաշխման հատվածը (այդ հատվածի բաշխման ժամանակը կախված չէ բաշխվող ֆայլի ծավալից):



Նկ. 19. Մշակված մեթոդի և Շամիրի մեթոդի գաղտնիքի բաշխման արագործությունների համեմատական գրաֆիկները՝ բաղադրամասերի հաստատուն արժեքի դեպքում (n=5,t=2,3,4,5)

Խնդրի արդիականության մեջ նշվեց, որ մշակվող մեթոդի հիմնական նպատակը մեծ ծավալի ինֆորմացիայի արագ բաշխումը և վերականգնումն է: Այդ նպատակի համար ցանկալի է աղյուսակի տեսքով ներկայացնել տարբեր ծավալի ֆայլերի բաշխման և վերականգնման ժամանակները: Աղյուսակ 15-ում պատկերված է տարբեր ծավալի ֆայլերի բաշխման և վերականգնման ժամանակները: Մեթոդի առավելությունը ընդգծելու համար աղյուսակ 16-ում ներկայացված են նույն ֆայլերի բաշխման և վերականգնման ժամանակները Շամիրի մեթոդի համար: Ընտրված է (3,5) շեմային կառուցվածքը:

Աղյուսակ 15-ի և 16-ի համեմատումից երևում է, որ օրինակ 2Gb ֆայլի բաշխման դեպքում բաշխման գործողությունը մշակված մեթոդի դեպքում 42.3%-ով ավելի արագ է Շամիրի մեթոդից, իսկ վերականգնամ ժամանակը ավելի արագ է 11.95%-ով:

Աղյուսակ 15: Մշակված մեթոդով բաշխման և վերականգնման ժամանակները (3,5) շեմային կառուցվածքի համար

Ֆայլի ծավալը	Բաշխման ժամանակը (վ)	Վերականգնման ժամանակը (վ)
10mb	3.75	4.56
50mb	18.11	21.83
100mb	37.23	42.51
200mb	73.12	83.98
500mb	185.14	209.87
1gb	370.33	416.33
2gb	740.21	829.38

Աղյուսակ 16: Շամիրի մեթոդով բաշխման և վերականգնման ժամանակները (3,5) շեմային կառուցվածքի համար

Ֆայլի ծավալը	Բաշխման ժամանակը (վ)	Վերականգնման ժամանակը (վ)
10mb	6.42	4.71
50mb	32.1	23.55
100mb	64.2	47.1
200mb	128.4	94.2
500mb	321	235.5
1gb	642	471
2gb	1284	942

2.4. Գլուխ 2-ի ամփոփում

Այս գլխում մանրամասն նկարագրվեց սխալներ ուղղող կոդերի հիման վրա աշխատող գաղտնիքի բաշխման շեմային մեթոդ, որը հնարավորություն է տալիս բաշխել և վերականգնել մեծ ծավալի ինֆորմացիա: Նկարագրված մեթոդը ճկուն է և կարող է աշխատել ցանկացած երկուական հատվածային կոդի դեպքում: Նկարագրվեց խմբավորման աղյուսակների ստացման երկու մոտեցում, որոնցից առաջինը հիմնված է հատարկման վրա, իսկ երկրորդ տարբերակում օգտագործվում է հավասարակշիռ կոդի կոդաբառերի աղյուսակը, որպես խմբավորման աղյուսակ: Նկարագրվեց աղյուսակների ձևավորման համար մշակված ծրագրային ապահովումը: Այս գլխում նաև համեմատվեց նոր ստեղծված մեթոդը Շամիրիի մեթոդի հետ, որից պարզ դարձավ, որ մշակված մեթոդը մեծ ծավալի ինֆորմացիան ավելի արագ է բաշխում և վերականգնում: Մշակված մեթոդի կիրառմամբ գաղտնիքի բաշխման գործողությունը արագացել է 42.3%-ով, իսկ վերականգման ժամանակը ավելի արագ է 11.95%-ով:

Գլուխ 3:

Բաղադրամասի հավաստիության ստուգումը և վնասված կամ կորած բաղադրամասի վերականգնումը

Այս գլխում ներկայացվում է սխալներ ուղղող կոդերով բաշխման դեպքում բաղադրամասի հավաստիության ստուգման մեթոդ, որը մասամբ հիմնված է Շամիրի մեթոդում բաղադրամասի հավաստիության ստուգման մեթոդի վրա: Մշակված ստուգման մեթոդը համեմատվում է Շամիրի բաշխման համար բաղադրամասի հավաստիության ստուգման մեթոդի հետ ըստ արագագործության: Նաև նկարագրվում է սխալներ ուղղող կոդերով բաշխման դեպքում կորած կամ վնասված բաղադրամսի վերականգնման հնարավորությունը: Այս դեպքում նույնպես համեմատություն է կատարվում Շամիրի մեթոդի հետ:

3.1. Բաղադրամասի հավաստիության ստուգումը

Վերը նշվել է, որ գաղտնիքի բաշխման համակարգերում մասնակիցները չեն վստահում մեկը մյուսին և նրանցից յուրաքանչյուրը, ինչպես նաև դիլերը կարող են հակառակորդի դերում լինել: Գաղտնիքի բաշխման համակարգերի կայունության ապահովման խնդիրը նաև կապված է բաղադրամասերի հուսալի փոխանակման հետ: Դրա կայունության ապահովման համար կարելի է ստեղծել այնպիսի համակարգեր, որոնցում կողմերից յուրաքանչյուրը կարող է ստուգել իրեն հատկացված բաղադրամասի ճիշտ կամ սխալ լինելը:

Շամիրի գաղտնիքի բաշխման շեմային մեթոդն ապահովում է նաև բաղադրամասերի ստուգման, թարմացման, կորած բաղադրամասերի վերականգնման և լրացուցիչ այլ հնարավորություններ: Առաջարկվող բաղադրամասերի ստուգման մեթոդը հիմնված է Շամիրի բաղադրամասերի ստուգման մեթոդի վրա, այդ իսկ պատճառով դիտարկենք այդ մեթոդը մանրամասն:

Բաղադրամասի ստուգման հնարավորությամբ գաղտնիքի բաշխման Շամիրի մեթոդը բաղկացած է վեց փուլից.

1. Ըստ գաղտնիքի բաշխման Շամիրի (միջարկման հիման վրա) մեթոդի դիլերն ընտրում է կամայական t աստիճանի բազմանդամ, որտեղ $S = a_0$:

$$\boldsymbol{Q(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_t x^t}$$

2. Ընտրվում են $\boldsymbol{p}$ և $\boldsymbol{q}$ պարզ թվերն այնպես, որ $\boldsymbol{p = 2 \cdot q + 1}$: Օրինակ՝$\mathbf{p = 59, q = 29}$ կամ $\mathbf{p = 23}$ և $\boldsymbol{q = 11}$, ընդ որում այս թվերը գաղտնի չեն:
3. Ընտրվում են $\boldsymbol{g}$ թիվն այնպես, որ՝ $\boldsymbol{g^q \ mod \ p = 1}$: Օրինակ՝$\boldsymbol{p = 11}$ և $\boldsymbol{q = 5, g = 3, 3^5 \ mod \ 11 \equiv 1}$:
4. Դիլերը հաշվարկում է՝ $\mathbf{r_i = g^{a_i} \ (mod \ p), i = 0, 1, 2 \ldots t}$ և հրապարակում $\boldsymbol{r_0, r_1, \ldots, r_t}$ թվերը:

5. Դիլերը կամայական $\boldsymbol{j = 1, 2 \ldots n}$ թվերի համար հաշվարկում և փակ կապուղիով կողմերին է հաղորդում կորդինատները՝ $\boldsymbol{S_j = Q(j)}$:
6. Կողմերից յուրաքանչյուրը ստուգում է հետևյալ հավասարումը՝

$$\boldsymbol{g^{S_j} = r_0 \cdot (r_1)^j \cdot \ldots \cdot (r_t)^{j^t} (mod\ p)}$$

Որպեսզի համոզվի, որ իր բաղադրամասը իրոք հանդիսանում է S գաղտնիքի բաղադրամաս: Քանի որ իրոք, տեղի ունի հետևյալ հավասարումը՝

$$\boldsymbol{r_0 \cdot (r_1)^j \cdot \ldots \cdot (r_t)^{j^t} = g^{a_0} \cdot g^{a_1 \cdot j} \cdot \ldots \cdot g^{a_t \cdot j^t} = g^{a_0 + a_1 \cdot j + \cdots + a_t \cdot j^t} = g^{Q(j)} (mod\ p)}$$

Դիտարկվող բաշխման համակարգի բաղադրամասերի թարմացման մեթոդի համար հարկավոր է ուսումնասիրել այդ բաղադրամասերի կառուցվածքը [38, 39], որը ներկայացված է նկ.10-ում:

Այս մեթոդով բաշխման դեպքում բաղադրամասերի ստուգումը հիմնված է **«Շամիրի մեթոդով բաշխվող»** և **«Պարունակություն»** հատվածների իսկության ստուգման վրա: Դիլերը բաշխումից առաջ որոշում է g, q և p թվերը, որից հետո հաշվարկում և հրապարակում է $r_0, r_1, \ldots, r_t$ թվերը: Այնուհետև ընտրում է բաշխման համար սխալներ ուղղող կոդը և այն աղյուսակը, որով պետք է կատարվի գաղտնի տվյալների բաշխումը: Հաջորդ քայլում դիլերը գեներացնում է $\mathbf{ID_1, ID_2, \ldots, ID_n}$ գաղտնաբառը, դրան կցում սխալներ ուղղող կոդի համարը, օգտագործվող աղյուսակի համարը և համարելով դա որպես S գաղտնիք, կատարում է այդ հատվածի բաշխումը Շամիրի մեթոդով: Որպես բաշխման տվյալներ ընդունվում են այն բաղադրամասերի և շեմի արժեքը, որը օգտագործվելու է գաղտնի ինֆորմացիայի բաշխման ժամանակ: Այս ամենից հետո կատարվում է գաղտնի ֆայլի բաշխումը (2.1)-ում նկարագրված մեթոդով, որից հետո ոչնչացվում է գաղտնիք հանդիսացող ինֆորմացիան: Վերջին քայլում դիլերի կողմից հաշվարկվում և հրապարակվում են ստացված բաղադրամաս ֆայլերի CRC32 արժեքները [45] (CRC1,CRC2,…,CRCt): Հաշվի առնելով, որ CRC32-ը միակողմանի ֆունկցիա է, ապա կարող ենք պնդել, որ նրա հրապարակումը ոչինչ չի

բացահայտում բուն բաղադրամասի վերաբերյալ և հետևաբար անվտանգության տեսանկյունից ոչ մի խնդիր չի առաջացնում: Այսքանով դիլերի կողմից բաշխման գործընթացը ավարտվում է:

Այժմ դիտարկենք այն գործողությունների հաջորդականությունը, որը պետք է կատարի բաղադրամասի ստացողը, որպեսզի համոզվի, որ իր ստացած բաղադրամասը իրական է և փոփոխության ենթարկված չէ:

1. Ստանալով բաղադրամասը, նախ և առաջ հարկավոր է առանձնացնել **«Շամիրի մեթոդով բաշխվող»** հատվածը և վերը նկարագրված ալգորիթմով ստուգել այդ հատվածի իսկությունը (օգտագործելով դիլերի կողմից հրապարակված տվյալները): Եթե այդ հատվածում սխալ չի հայտնաբերվում , ապա հարկավոր է շարունակել ստուգումը և ստուգել **«Պարունակություն»** հատվածի իսկությունը: Հակառակ դեպքում ստուգումը ավարտվում է և բաղադրամասը համարվում նենգափոխված (կեղծ կամ վնասված):
2. Առաջին կետի ստուգման դրական արդյունքի դեպքում, բաղադրամասի սեփականատերը հաշվարկում է **«Պարունակություն»** հատվածի CRC32 արժեքը և համեմատում այն դիլերի կողմից հրապարակված համապատասխան CRC արժեքի հետ: Համընկնելու դեպքում համարվում է, որ այդ հատվածում փոփոխություն չի կատարվել և սրանով ստուգման գործընթացը ավարտվում է:

Պետք է արձանագրել, որ ստուգման գործընթացից դուրս մնացին **«Բաղադրամասի համարը»**, **«Բաղադրամասերի քանակը»**, **«Շեմը», «Շամիր հատվածի ծավալը»** և **«Շամիր հատվածի CRC»** հատվածները: Այս հատվածները չստուգելու պատճառը հետևյալն է:

1. **«Բաղադրամասի համարը»**, **«Բաղադրամասերի քանակը»** և **«Շեմը»** հատվածները համարվում են հայտնի տվյալներ և դրանց ստուգման համար, որևէ լրացուցիչ ալգորիթմական լուծումներ անհրաժեշտ չեն: Հարկավոր է

ընթերցել դրանք և համեմատել նախապես հայտնի տվյալների հետ և անհամապատասխանություն հայտնաբերելու դեպքում համարել, որ բաղադրամասը փոփոխված է ու ոչ պիտանի:

2. **«Շամիր հատվածի ծավալը»** և **«Շամիր հատվածի CRC»** հատվածները ստուգելու կարիք չկա, քանի որ նրանցից առնվազն մեկի փոփոխությունը արդեն կառաջացներ սխալ **«Շամիր հատվածի ծավալը»** հատվածի ստուգման ժամանակ:

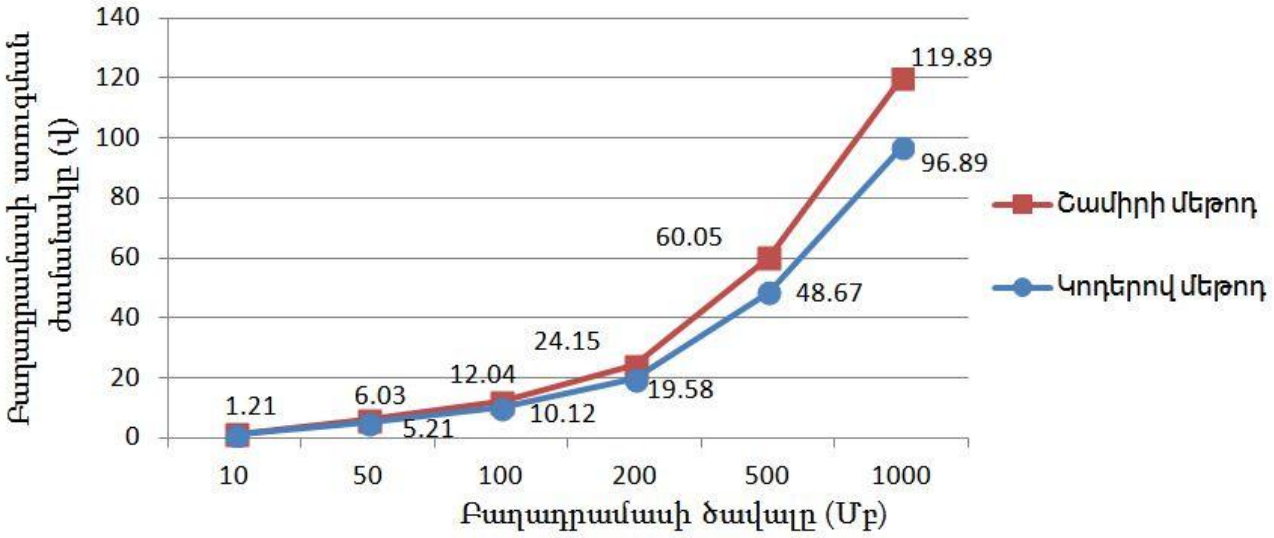
Այսքանով բաղադրամասերի ստուգման գործընթացը կարելի է համարել ավարտված:

Այժմ ուսումնասիրենք այս եղանակով բաղադրամասերի ստուգման հավաստիությունը: Հնարավոր է արդյոք “խափել” այս ստուգման մեթոդին: Քանի որ ստուգման գործընթացը բաժանված է երկու մասի, ապա անվտանգության ստուգման համար դիտարկենք նրանից յուրաքանչյուրն առանձին:

1. **«Շամիր հատվածի ծավալը»** հատվածի անվտանգությունը հիմնված է Շամիրի մեթոդով բաղադրամասերի ստուգման վրա, իսկ այդ մեթոդը համարվում է, որ ապահովում է կատարյալ անվտանգությունը: Հետևաբար այս հատվածի ստուգման իսկությունը կասկած չի հարուցում:
2. **«Պարունակություն»** հատվածի ստուգման ժամանակ օգտագործվում է CRC32 արժեքը, որը կարելի է համարել հեշ ֆունկցիա: Իհարկե հեշ ֆունկցիաների դեպքում գոյություն ունի կոլիզիայի հավանականություն և տեսականորեն հնարավոր է, որ լինի այլ **«Պարունակություն»** հատված, որի դեպքում ստացվի միևնույն CRC32 արժեքը: Սակայն հաշվի առնելով, որ **«Պարունակություն** հատվածը մեծ ծավալի ինֆորմացիա է և այդ հատվածի ծավալը գաղտնի չէ, ապա հակառակորդի խնդիրը էականորեն բարդանում է: Նա պետք է փորձի գտնել այդ նույն երկարության այլ բիթային շարք, որի դեպքում կստանա նույն

CRC32 արժեք: Չնայած որ այս գործողությունը բավականին բարդ է և ժամանակատար, այնուամենայնիվ անհրաժեշտության դեպքում կարելի CRC32 հատվածները փոխարինել այլ, ավելի երկար հեշ արժեքներով, որոնց դեպքում շատ ավելի դժվար կլինի գտնել միևնույն հեշ արժեք ունեցող և նույն երկարության բիթային շարք: CRC32-ի ընտրությունը պայմանավորված է նրա արագագործությամբ և միաժամանակ փոքր ծավալով:

Նկ.20-ում պատկերված են Շամիրի և մշակված մեթոդներում բաղադրամասի իսկության ստուգման մեթոդների արագործությունների գրաֆիկները: Գրաֆիկները ցույց են տալիս, որ մշակված մեթոդն ավելի արագ է, քան Շամիրիր մեթոդի դեպքում:



Նկ. 20. Շամիրի և մշակված մեթոդներում բաղադրամասի իսկության ստուգման մեթոդների արագործությունների գրաֆիկները

3.2. Կորած կամ վնասված բաղադրամասի վերականգնումը

Գաղտնիքի բաշխման համակարգերում, ինչպես և մնացած համակարգերում, հնարավոր են չնախատեսված իրավիճակներ: Միայն վերականգնման, բաշխման և բաղադրամասի ստուգման հնարավորությունները բավարար չեն: Այդպիսի համակարգերում նախատեսվում են նաև կորած կամ վնասված բաղադրամասի վերականգնման, նոր բաղադրամասի ստացման, բաղադրամասերի թարմացման և այլ հնարավորություններ: Օրինակ առավել հայտնի Շամիրի մեթոդում բոլոր այդ հնարավորությունները առկա են:

Գաղտնիքի բաշխված պահպանման ժամանակ հնարավոր է, որ մասնակիցներից որևէ մեկի մոտ բաղադրամասը վնասվի կամ պարզապես կորի: Այս իրավիճակի համար անհրաժեշտ է նախապես ունենալ այդ բաղադրամասի վերականգնման մեթոդ: Քանզի գաղտնիքը բաշխելուց հետո ոչնչացվում է, հետևաբար բնական է, որ վնասված կամ կորած բաղադրամասի վերականգնման ժամանակ պետք է մասնակցեն որոշ քանակությամբ այլ բաղադրամասեր: Գոյություն ունի կորած բաղադրամասի վերականգնման երկու տարբերակ.

1. առանց գաղտնիքի բացահայտման
2. գաղտնիքի բացահայտմամբ

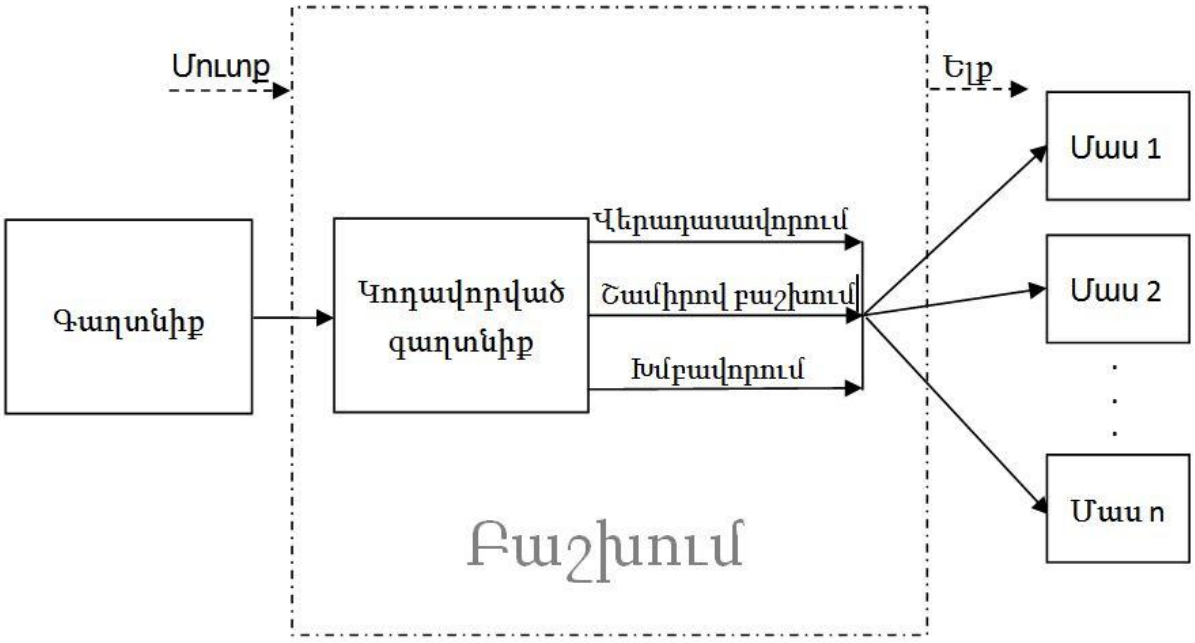
Երկրորդ տարբերակը իր հերթին ունի երկու տարբերակ: Առաջին տարբերակում կատարվում է գաղտնիքի ամբողջական վերականգնում, որից հետո վերականգնված գաղտնիքը կրկին բաշխվում է: Ակնհայտ է, որ այս տարբերակը դանդաղ է և բաղադրամասի վերականգնման ժամանակը հավասար է բաշխման և վերականգնման ժամանակների գումարին: Մյուս տարբերակում հնարավոր է բաղադրամասի ավելի արագ վերականգնում՝ գաղտնիքի միջանկյալ բացահայտմամբ: Առավել նախնտրելի է առանց գաղտնիքի բացահայտման տարբերակը, սակայն ոչ

բոլոր մեթոդներն են ընձեռում այդ հնարավորությունը: Այդպիսի մեթոդներում ցանկալի է ունենալ գաղտնիքի միջանկյալ բացահայտմամբ, սակայն ավելի արագ բաղադրամասի վերականգնման հնարավորություն: (2.1) բաժնում ներկայացված մեթոդը ևս չի ընձեռում առանց գաղտնիքի բացահայտման բաղադրամասի վերականգնման հնարավորություն: Սակայն այդ մեթոդի համար հնարավոր գաղտնիքի միջանկյալ բացահայտմամբ կորած բաղադրամասի արագ վերականգնման հնարավորություն, որը նկարագրված է ստորև:

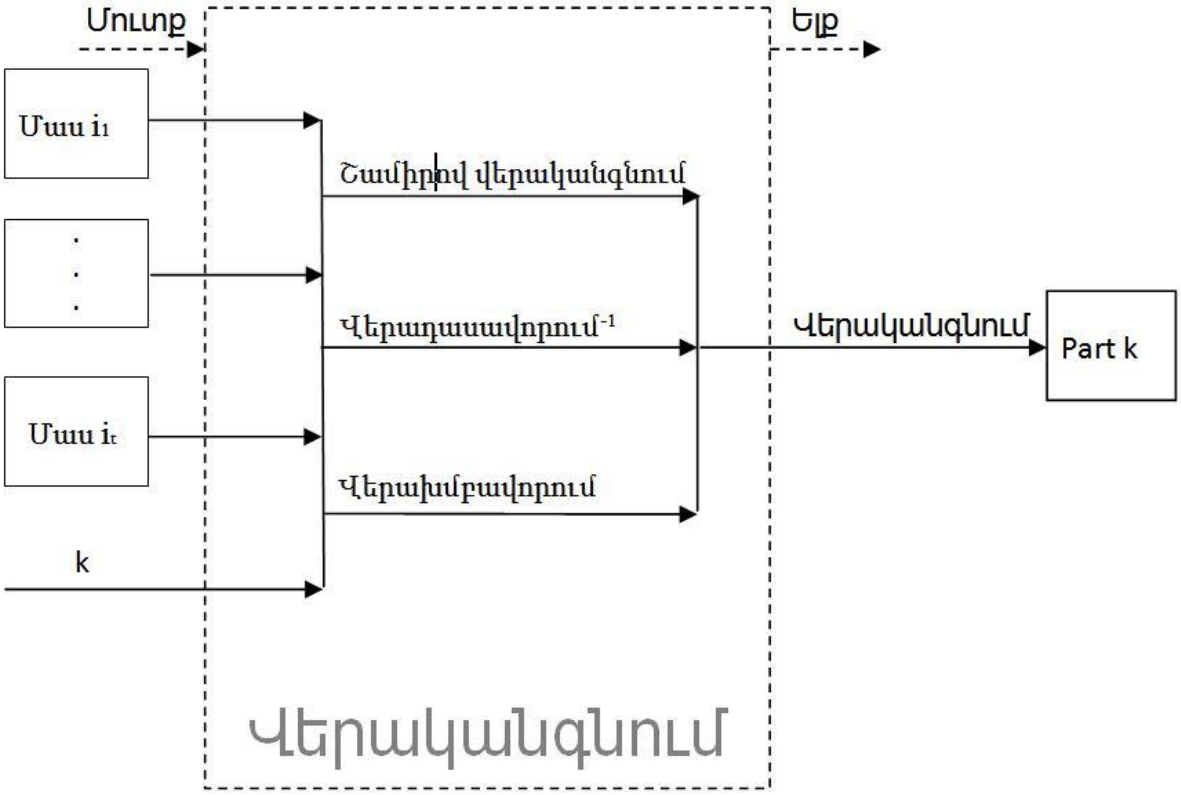
Դիտարկենք (2.1) բաժնում նկարագրված գաղտնիքի բաշխման մեթոդը: Նկ. 21-ում պատկերված է այդ մեթոդի բաշխման գործընթացը ներկայացնող գծագիրը: Ինչպես տեսնում ենք սխեմայի մուտքին տրվում է գաղտնի ֆայլը, որից հետո այն կոդավորվում է ընտրված սխալներ ուղղող կոդով, այնուհետև կատարվում են խմբավորման, վերադասավորման և գլխամասի բաշխման գործողությունները: Այդ ամենից հետո ստացվում են բաղադրամաս ֆայլերը:

Այժմ պատկերացնենք, որ որևէ բաղադրամաս կորել է: Վերականգնման գործընթացը հիմնված է գաղտնիքի միջանկյալ բացահայտման վրա, սակայն այդ գործընթացում բացակայում է վերջնական ապակոդավորման գործողությունը: Վերականգնման գործընթացը պատկերված է նկ.22-ում: Որպես մուտքային ինֆորմացիա տրվում է t հատ բաղադրամաս և այն բաղադրամասի համարը, որը անհրաժեշտ է վերականգնել: Վերականգնման համար նախ անհրաժեշտ է վերականգնել **«Շամիր հատվածի ծավալը»** հատվածը և ստանալ վերադասավորման գաղտնաբառը, սխալներ ուղղող կոդի և աղյուսակի համարները: Ունենալով այդ տվյալները հնարավոր է կատարել վերադասավորման և խմբավորման հակադած գործողությունները: Այս ամենից հետո գաղտնիքի վերականգնման համար անհրաժեշտ է կատարել դեկոդավորման գործողությունը, սակայն բաղադրամասի վերականգնման համար այդ գործընթացը անհրաժեշտ չէ: Վերականգնման համար

ընդամենը անհրաժեշտ է ստացված տվյալներից առանձնացնել այն սյուների տվյալները, որոնք օգտագործվել են k-րդ բաղադամասի ստացման համար:



Նկ. 21. Գաղտնիքի բաշխման մեթոդի աշխատանքը



Նկ. 22. Կորած բաղադրամասի վերականգնումը

Դիտարկենք օրինակ: Ենթադրենք բաշխման համար օգտագործվել է աղյուսակ 17-ը, որն ապահովում է (4,5) շեմային կառուցվածք: Որպես կորած բաղադրամաս դիտարկենք առաջին բաղադրամասը: Այս դեպքում վերականգնման համար որպես մուտքային ինֆորմացիա հարկավոր է տալ 2-5 բաղադրամասերը և 1 թիվը: **«Շամիրով վերականգնում»**, **«Վերադասավորում**$^{-1}$**»** և **«Վերախմբավորում»** գործողություններից հետո կստանանք այն սյուների տվյալները, որոնք նշված են 2-5 բաղադրամասերի դիմաց: Դժվար չէ տեսնել, որ առաջին բաղադրամասի համար անհրաժեշտ սյուները այդտեղ կան: Անհրաժեշտ է վերցնել այդ սյուները, ձևավորել առաջին բաղադրամասը և ելքում տալ դա: Ինպես տեսանք այս գործընթացից դուրս մնաց ամենաժամանակատար գործողությունը, տվյալների դեկոդավորումը և բացակայող սյուների վերականգնումը:

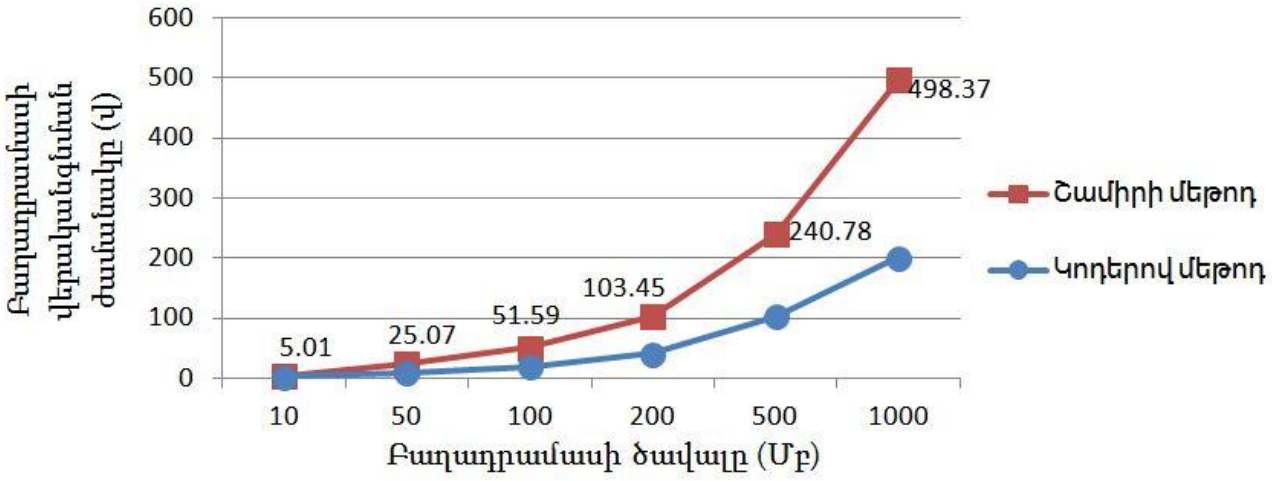
Աղյուսակ 17: (31,21,2) ԲՉՀ կոդով (4,5) շեմային կառուցվածք

Մաս 1	**17**	**23**	**14**	**18**	**13**	**3**	**19**	**26**	**16**	**9**	**12**	**8**	**5**	**15**	**6**	**2**	**10**	**1**	**7**	**4**	**29**
Մաս 2	**25**	**15**	**5**	**10**	**23**	**17**	**1**	**4**	**22**	**9**	**12**	**13**	**7**	**16**	**3**	**18**	**6**	**8**	**2**	**28**	**30**
Մաս 3	**15**	**7**	**21**	**17**	**8**	**3**	**27**	**19**	**6**	**23**	**13**	**2**	**16**	**10**	**18**	**4**	**5**	**9**	**24**	**25**	**1**
Մաս 4	**18**	**14**	**16**	**10**	**26**	**21**	**1**	**4**	**15**	**9**	**11**	**13**	**7**	**6**	**5**	**17**	**2**	**3**	**23**	**8**	**28**
Մաս 5	**27**	**29**	**30**	**10**	**8**	**3**	**1**	**4**	**6**	**9**	**11**	**13**	**14**	**15**	**16**	**17**	**22**	**24**	**2**	**7**	**5**

Այսպիսով ստացվեց գաղտնիքի միջանկյալ բացահայտմամբ, սակայն էականորեն ավելի արագ վնասված կամ կորած բաղադրամասի վերականգնման մեթոդ: Նկարագրված մեթոդը չի կարող վերականգնել կորած բաղադրամասը այն դեպքում, երբ $t = n$: Այդ դեպքում հնարավոր չի լինի վերականգնել **«Շամիրով վերականգնում»** հատվածը և հետևաբար մնացած գործողությունները արդեն անհնար կլինի կատարել: Այսինքն կարող ենք փաստել, որ $t = n$ դեպքում բաշխման սխեման շատ ռիսկային է և նույնիսկ մեկ բաղադրամասի վնասումը կամ կորուստը կբերի գաղտնիքի կորստին:

Գաղտնիքի բաշխման համակարգի անխափան աշխատանքի համար նախընտրելի է ժամանակ առ ժամանակ կատարել բոլոր բաղադրամասերի ստուգում և վնասված կամ կորած բաղադրամասերի հայտնաբերման դեպքում կատարել այդ բաղադրամասերի վերականգնում: Հակառակ դեպքում երկար ժամանակ անց կարող է պարզվել, որ վնասվել կամ կորել են այնքան բաղադրամասեր, որ գաղտնիքի վերականգնումը այլևս հնարավոր չէ, որն իր հերթին կբերի գաղտնիքի կորստին: Կորած բաղադրամասի վերականգնման մեթոդի արագագործությունը և նրա համեմատումը Շամիրի մեթոդում վերականգնման հետ ներկայացված է հաջորդ գլխում:

Նկ.23-ում պատկերված են Շամիրի և մշակված մեթոդներում կորած կամ վնասված բաղադրամասի վերականգնման արագործությունների գրաֆիկները: Գրաֆիկները ցույց են տալիս, որ մշակված մեթոդն ավելի արագ է, քան Շամիրի մեթոդի դեպքում:



Նկ. 23. Շամիրի և մշակված մեթոդներում բաղադրամասերի վերականգնման մեթոդների արագործությունների գրաֆիկները

3.3. Գլուխ 3-ի ամփոփում

Այսպիսով, նկարագրվեց գաղտնիքի բաղադրամասերի հավաստիության ստուգման և կորած կամ վնասված բաղադրամասի վերականգնման մեթոդներ սխալներ ուղղող կոդերով բաշխման համար: Նկարագրված մեթոդները հնարավորություն են տալիս ստուգել բաղադրամասի պատկանելությունը տվյալ գաղտնիքին, նրա վնասված կամ կեղծ լինելը, անհրաժեշտության դեպքում կատարել վնասված կամ կորած բաղադրամասի վերականգնման գործընթաց: Նկարագրված երկու մեթոդները միաժամանակ կիրառելու դեպքում հնարավոր է պարբերաբար ստուգել բոլոր բաղադրամասերը և թարմացնել վնասվածները: Այս ամենը մեծացնում է բաշխված գաղտնիքի բաղադարամասերի անվտանգությունը և տվյալների անվտանգ պահպանման ժամանակը:

Գլուխ 4:

Գաղտնիքի բաշխման մշակված շեմային մեթոդի ծրագրային իրականացումը

Այս գլխում ներկայացված է սխալներ ուղղող կոդերի հիման վրա մշակված գաղտնիքի բաշխման շեմային մեթոդի ծրագրային իրականացումը [42]: Ծրագիրը նախատեսված է Windows օպերացիոն համակարգի համար և իրականացվել է C# ծրագրավորման լեզվի միջոցով: Ծրագիրը նախատեսված է ինչպես կիրառական նշանակության, այնպես էլ հետազոտական նպատակների համար: Ատենախոսության ընթացքում գրաֆիկները ստացվել են մշակված ծրագրի օգնությամբ:

4.1. Մշակված ծրագրերը

Ատենախոսության ընթացքում ստացված գիտական արդյունքների հիման վրա մշակվել են մի քանի ծրագրային ապահովումներ.

1. Խմբավորման աղյուսակների ձևավորման ծրագրային մոդուլ:
2. Բաշխման աղյուսակների՝ հավասարակշիռ կոդերի հիման վրա աղյուսակի ձևավորման և սխալներ ուղղող կոդի որոշման ծրագրային մոդուլ:
3. Մշակված մեթոդով ինֆորմացիայի բաշխման, վերականգնման, բաղադրամասի ստուգման և վնասված կամ կորած բաղադրամասի վերականգնման կիրառական նշանակության ծրագրային ապահովում (ECC Sharing):
4. Մշակված մեթոդի և այլ շեմային մեթոդների ուսումնասիրման, համեմատման ու հետազոտման ծրագրային ապահովում (ECC Sharing Explore):

Հերթով նկարագրենք յուրաքանչյուրի նշանակությունը և ֆունկցիոնալությունը: Առաջին ծրագրային ապահովման մասին արդեն խոսվել է (2.2) բաժնում և ինչպես նշվեց, նախատեսված է հատարկման եղանակով տրված պահանջներին համապատասխան բաշխման աղյուսակի ավտոմատ ձևավորման համար: Երկրորդ ծրագրային մոդուլը նույնպես նախատեսված է բաշխման աղյուսակների ձևավորման համար, սակայն այդ ծրագրի հիմքում ընկած է (2.2) բաժնում նկարագրված հավասարակշիռ կոդերի հիման վրա աղյուսակների ձևավորումը և դրան համապատասխան սխալներ ուղղող կոդի որոշման ալգորիթմը: Այս երկու ծրագրերի աշխատանքը նկարագրվել է (2.2)-ում: Այս երկու ծրագրերը նախատեսված են ավելի շատ հետազոտական և ալգորիթմի ընդլայնման համար և գաղտնիքի բաշխման և վերականգման համար անհրաժեշտ չեն:

Երրորդ ծրագիրը նախատեսված է նոր մեթոդով գաղտնիքի բաշխում կատարելու համար: Այն իր մեջ ներառում է ինչպես գաղտնիքի բաշխման և վերականգնման, այնպես էլ (3.1), (3.2) բաժիններում նկարագրված բաղադրամասի

թարմացման և վերականգնման գործողություններ կատարելու համար: Այս ծրագիրը նպատակահարմար է օգտագործել այն դեպքերում, երբ անհրաժեշտ է բաշխել մեծ ծավալի ինֆորմացիա:

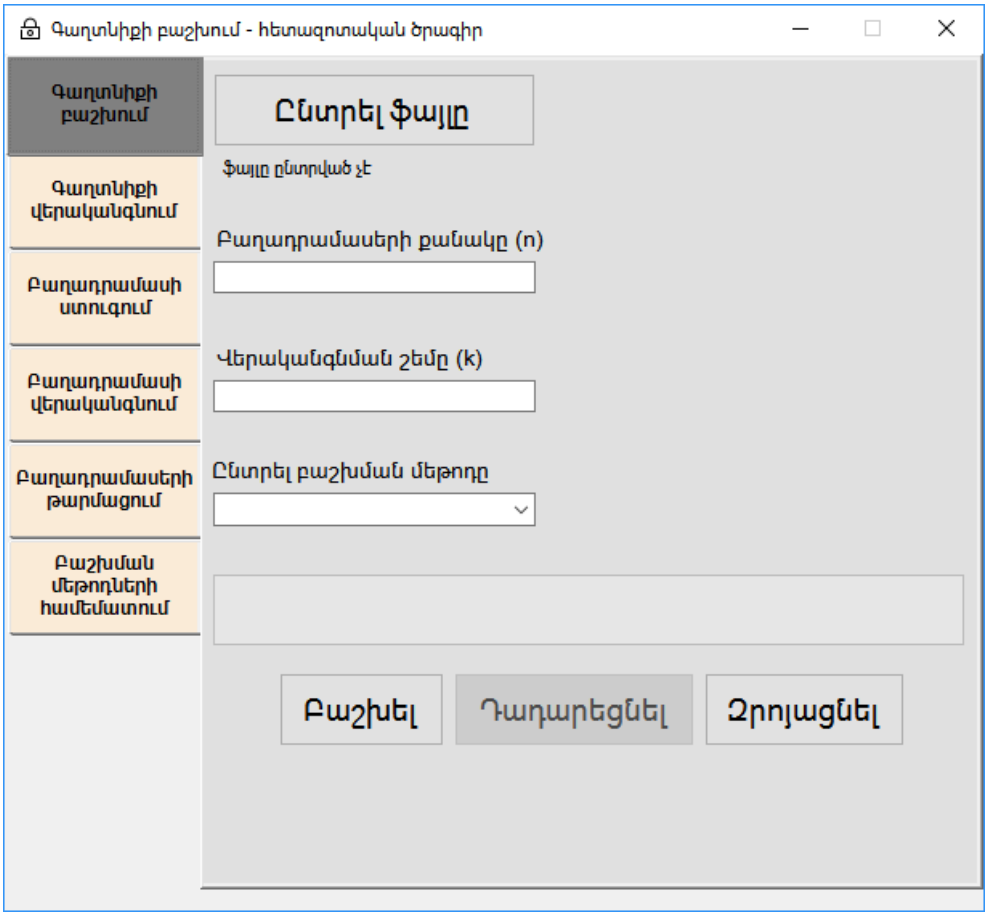
Չորրորդ ծրագրային ապահովումը նախատեսված է ուսումնական և հետազոտական նպատակների համար: Այն իր մեջ ընդգրկում է երրոր ծրագրի բոլոր հատկությունները և բացի այդ նախատեսված են այլ շեմային մեթոդների հետ համեմատման հնարավորություններ: Հնարավորություն է ընձեռվում կատարել համեմատություններ և ստանալ ժամանակային գրաֆիկներ: Այս ծրագրային ապահովումը նախատեսված է ուսումնական և հետազոտական նպատակների համար:

Հաջորդ բաժնում մանրամասն կդիտարկենք չորրորդ ծրագրի աշխատանքը:

4.2. ECC Sharing Explore ծրագրային ապահովման աշխատանքի օրինակ

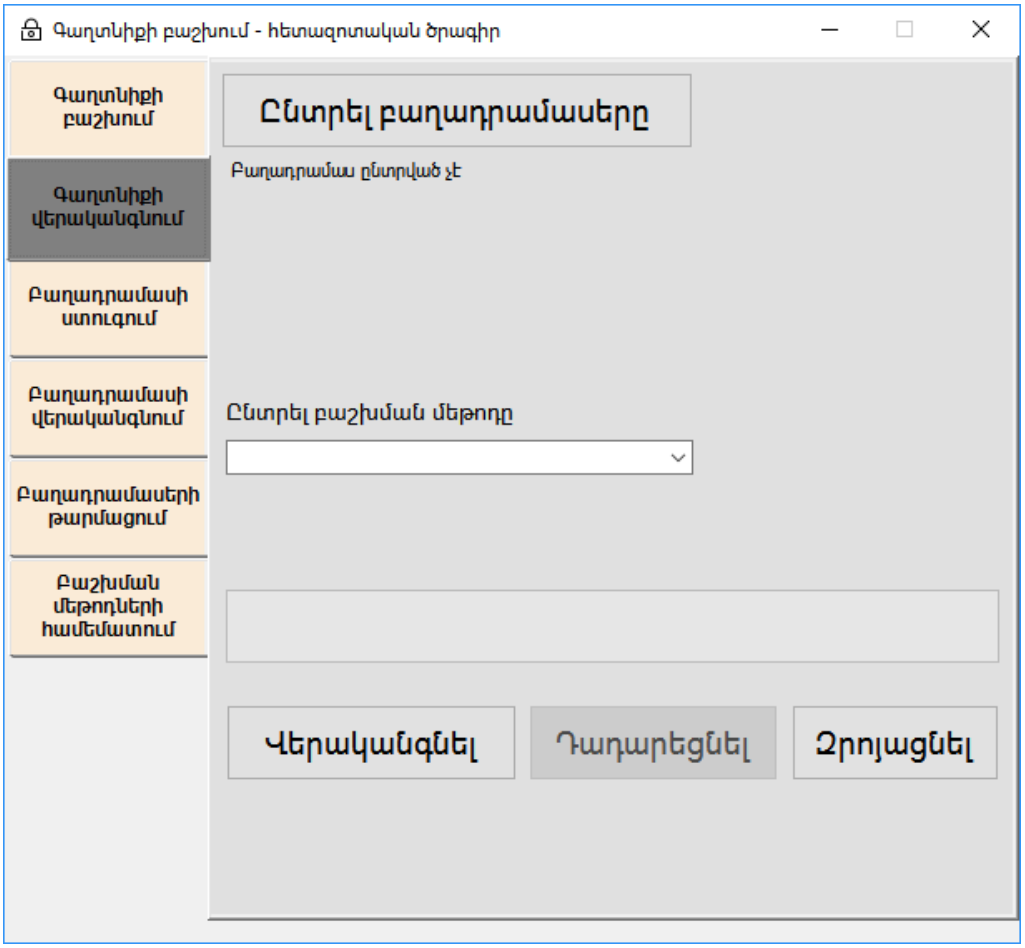
Ուսումնական նպատակների համար նախատեսված **ECC Sharing Explore** ծրագիրը վեց բաժիններից:

- Գաղտնիքի բաշխում (նկ. 24)
- Գաղտնիքի վերականգնում (նկ. 25)
- Բաղադրամասի ստուգում (նկ. 26)
- Բաղադրամասի վերականգնում (նկ. 27)
- Բաղադրամասի թարմացում (նկ. 28)
- Բաշխման մեթոդների համեմատում (նկ. 29)



Նկ. 24. ECC Sharing Explore ծրագրի տեսքը թողարկումից անմիջապես հետո (գաղտնիքի բաշխման պատուհան)

Գաղտնիքի բաշխում պատուհանը նախատեսված է ֆայլեր բաշխելու համար: Այդ նպատակի համար անհրաժեշտ է ընտրել ֆայլը, բաղադրամասերի քանակը, շեմը և բաշխման մեթոդը, որից հետ կատարել բաշխումը (նկ. 24): Բաշխման գործընթացը կարելի է ժամանակավորապես դադարեցնել և հետո շարունակել նույն կետից:

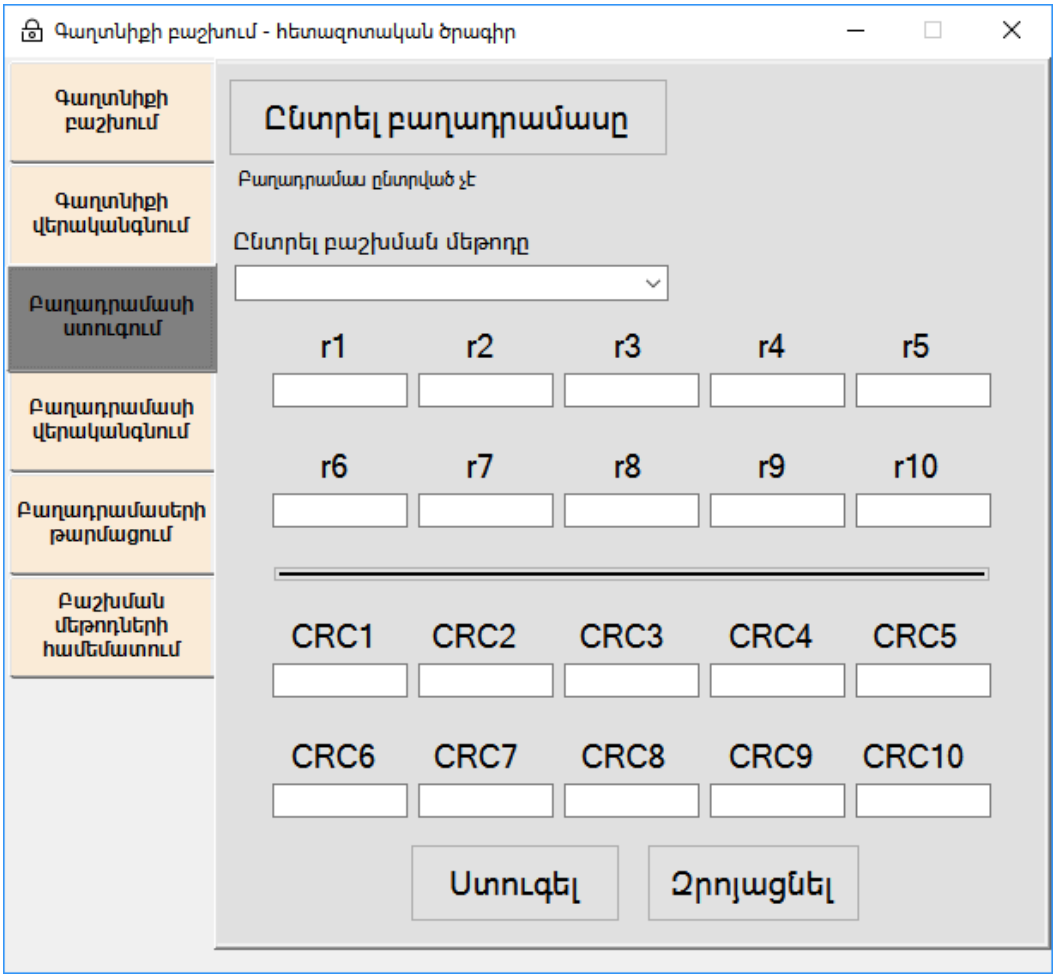


Նկ. 25. ECC Sharing Explore ծրագրի տեսքը (գաղտնիքի վերականգնում պատուհան)

Գաղտնիքի վերականգնում պատուհանում կարելի է վերականգնել բաշխված մեթոդը: Այդ նպատակի համար հարկավոր է ընտրել անհրաժեշտ քանակով բաղադրամասեր և բաշխման մեթոդը, որից հետո վերականգնել այն (նկ. 25): Գործողությունը կարելի է ժամանակավորապես դադարեցնել և նորից շարունակել նույն կետից:

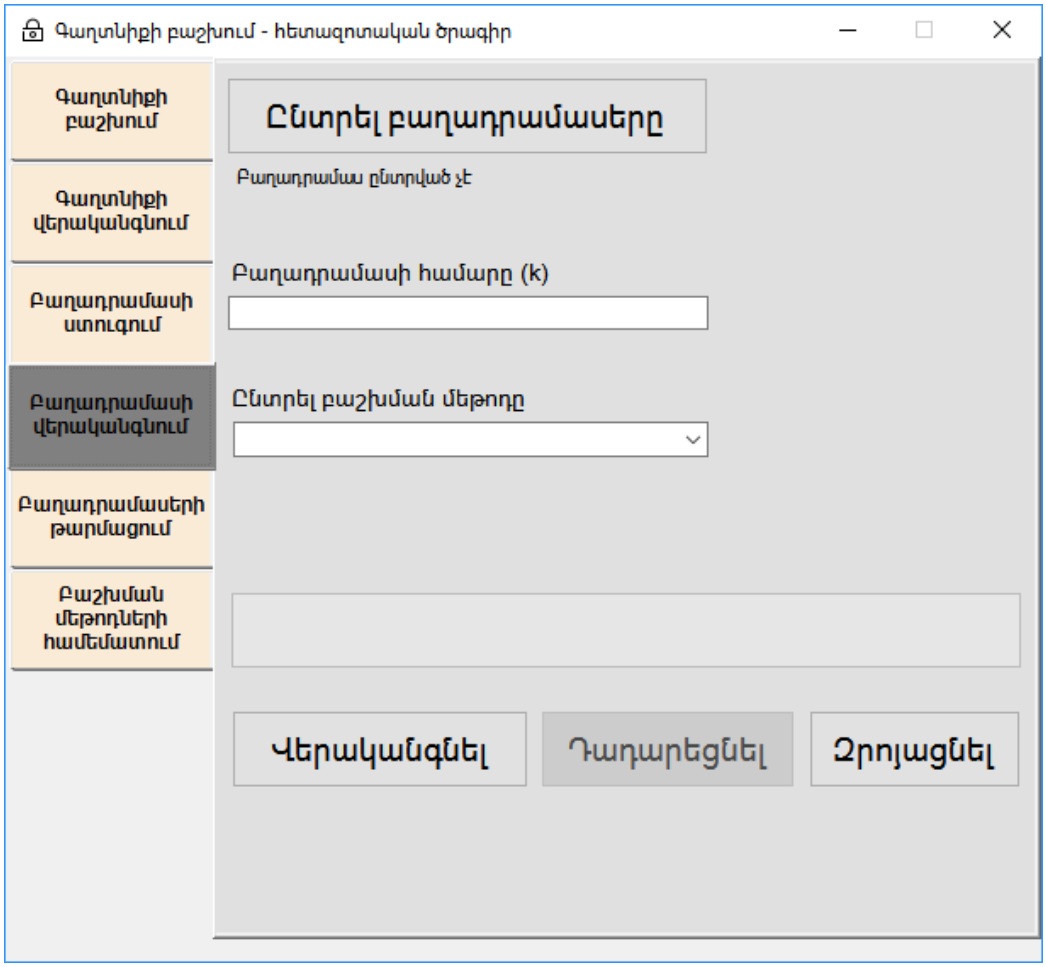
Բաղադրամասի ստուգում էջը նախատեսված է բաղադրամաս ֆայլերի իսկության ստուգման համար: Այն նախատեսված է կոդերով և Շամիրի մեթոդներով

բաշխման համար: Անհրաժեշտ է ընտրել բաղադրամաս ֆայլը, բաշխման մեթոդը և ներմուծել դիլերի կողմից հրապարակված տվյալները, որից հետո ծրագիրը կստուգի ընտրված բաղադրամասի իսկությունը (նկ. 26):



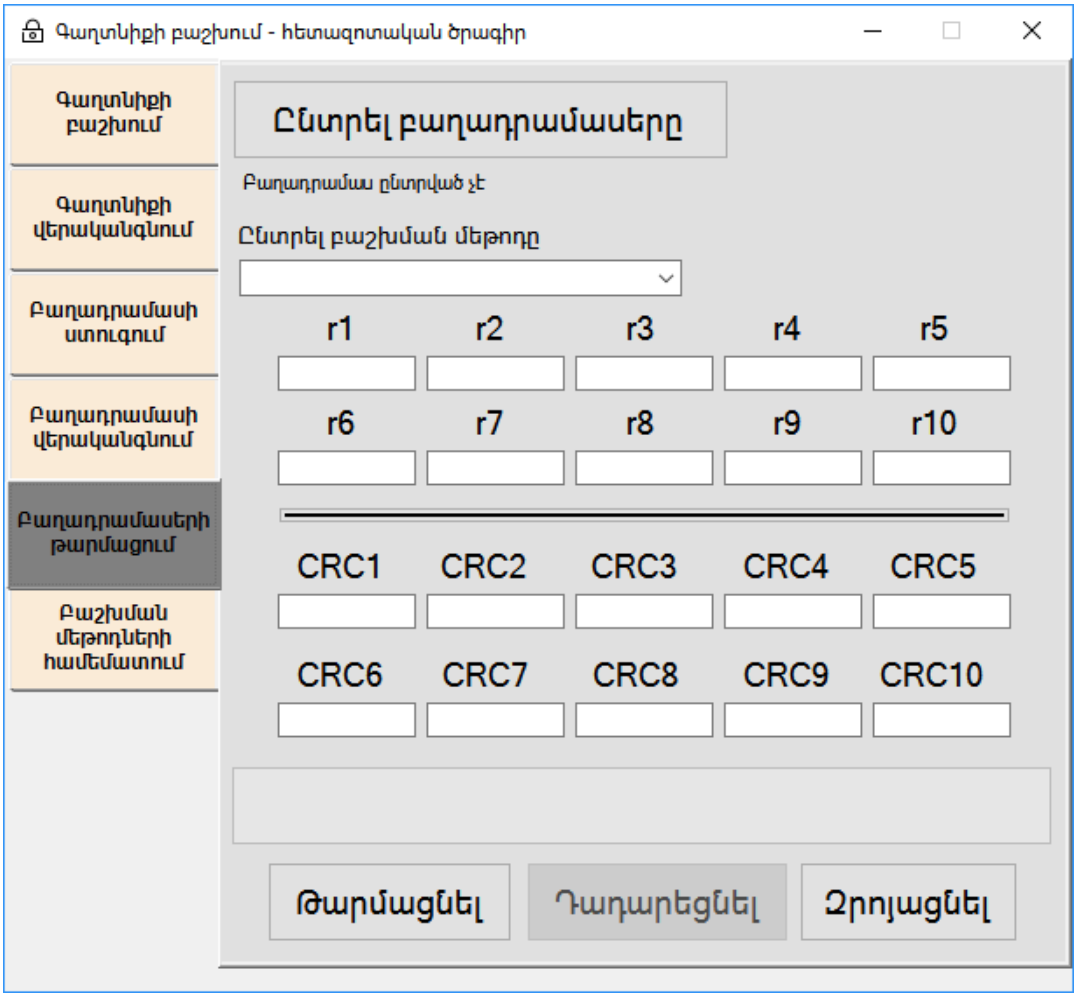
Նկ. 26. ECC Sharing Explore ծրագրի տեսքը (բաղադրամասի ստուգման պատուհան)

Բաղադրամասի վերականգնում պատուհանում կարելի է վերականգնել կորած կամ վնասված բաղադրամասը: Հարկավոր է ընտրել անհրաժեշտ քանակով բաղադրամասեր, վերականգնվող բաղադրամասի համարը և բաշխման մեթոդը, որից հետո վերականգնել այն (նկ. 27): Այս հնարավորությունը նախատեսված է միայն կոդերով և Շամիրի մեթոդների համար:



Նկ. 27. ECC Sharing Explore ծրագրի տեսքը (բաղադրամասի վերականգնման պատուհան)

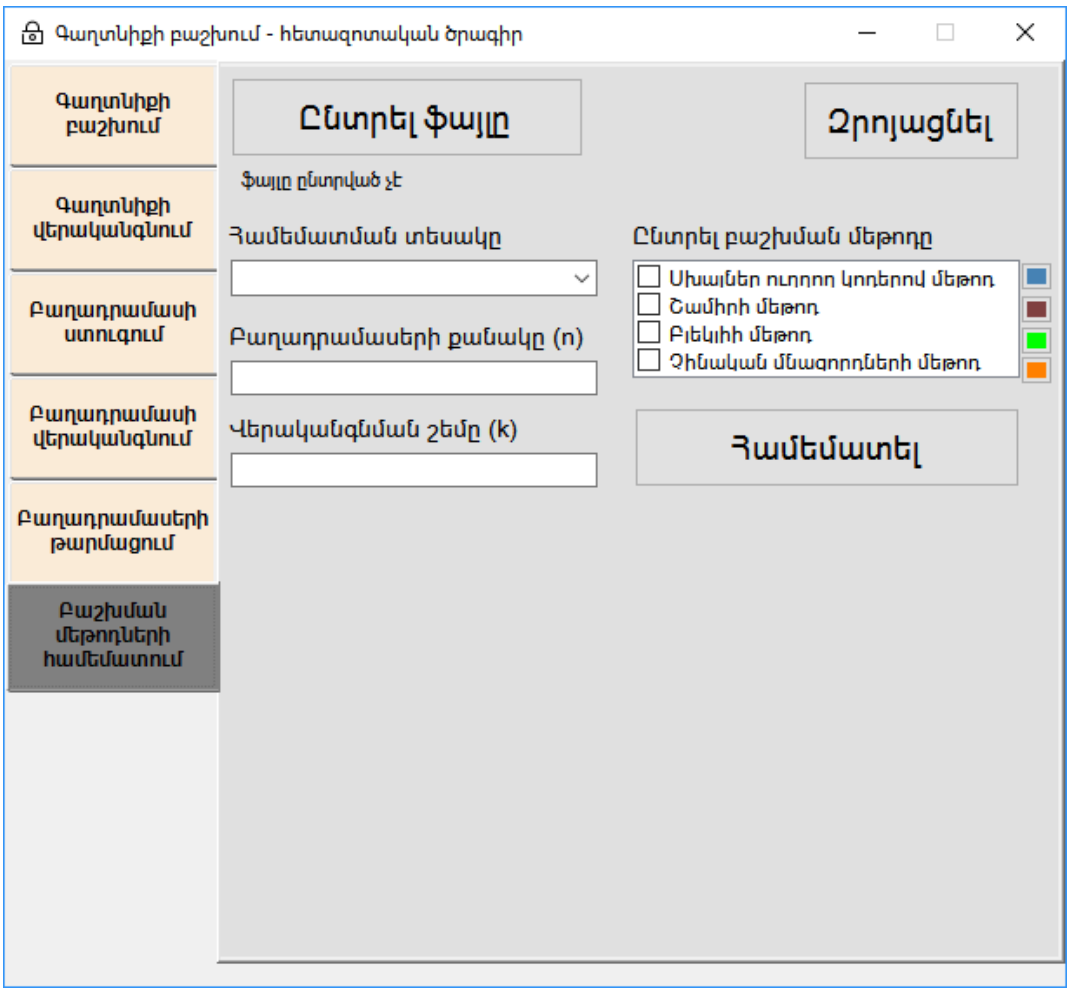
Բաղադրամասի թարմացում պատուհանը նախատեսված է բոլոր բաղադրամասերի թարմացման համար: Այդ նպատակի համար հարկավոր է ընտրել բաղադրամասերը, բաշխման մեթոդը և ներմուծել դիլերի կողմից հրապարակված տվյալները, որից հետո կատարել բաղադրամասերի թարմացում (նկ. 28): Դիլերի կողմից հրապարակված տվյալների բացակայության դեպքում ծրագիրը թարմացնում է բոլոր բաղադրամասերը, իսկ այդ տվյալների առկայության դեպքում, կատարվում է բաղադրամասերի ստուգում և թարմացվում են միայն վնասված կամ կորած բաղադրամասերը:



Նկ. 28. ECC Sharing Explore ծրագրի տեսքը (բաղադրամասերի թարմացման պատուհան)

Բաշխման մեթոդների համեմատում պատուհանը նախատեսված է բաշխման մեթոդների միմյանց հետ համեմատման համար: Այն հնարավորություն է տալիս ստանալ համեմատման ժամանակային գրաֆիկներ (նկ. 29): Հնարավոր է ընտրել համեմատման հետևյալ տարբերակները.

- Հաստատուն բաղադրամասերի քանակ
- Հաստատուն շեմի արժեք



Նկ. 29. ECC Sharing Explore ծրագրի տեսքը (բաշխման մեթոդների համեմատման պատուհան)

4.3. Գլուխ 4-ի ամփոփում

Այսպիսով, ստացված գիտական արդյունքների հիման վրա ստեղծվել են կիրառական և ուսումնահետազոտական նպատակների համար նախատեսված ծրագրային ապահովումներ: Ուսումնական նպատակների համար նախատեսված ծրագիրը հնարավորություն է տալիս կատարել փորձեր, հասկանալ բաշխման մեթոդների աշխատանքի սկզբունքները: Այն հնարավորություն է տալիս համեմատել բաշխման մեթոդները ըստ արագագործության: Մշակված ծրագրի օգնությամբ ստացվել են ատենախոսության ընթացքում ներկայացվող գրաֆիկները:

Եզրակացություն

- Մշակվել է գաղտնիքի բաշխման շեմային մեթոդ՝ հիմնված սխալներ ուղղող կոդերի վրա, որն ի տարբերություն գոյություն ունեցողների ավելի արագ է և հնարավորություն է տալիս կատարել մեծ ծավալի ինֆորմացիայի բաշխում՝ ապահովելով ինֆորմացիայի ինչպես գաղտնիությունը, այնպես էլ ամբողջականությունն ու հասանելիությունը:
- Առաջարկվել է բաղադրամասի հավաստիության ստուգման արագ մեթոդ, որն ի տարբերություն գոյություն ունեցողների, հնարավորություն է տալիս ստուգել բաղադրամասը սխալներ ուղղող կոդերով բաշխման դեպքում և հայտնաբերել կեղծված կամ վնասված բաղադրամասերը :
- Առաջարկվել է վնասված կամ կորած բաղադրամասի արագ վերականգնման մեթոդ, որն ի տարբերություն գոյություն ունեցողների, հնարավորություն է տալիս օգտագործել այն սխալներ ուղղող կոդերով բաշխման դեպքում և մեծացնում է գաղտնիքի ապահով պահպանման ժամանակը:
- Ստացված գիտական արդյունքների հիման վրա մշակվել է գաղտնիքի բաշխման ECC Sharing համակարգը, որը հնարավորություն է տալիս արագ բաշխել և վերականգնել մեծ ծավալի ինֆորմացիա, կատարել բաղադրամասի ստուգում, վերականգնել կորած կամ վնասված բաղադրամասերը, որի շնորհիվ բաշխման գործընթացը արագացել է 42.3%-ով, իսկ վերականգնմանը՝ 11.95%-ով:
- Ուսումնական նպատակների համար մշակված գաղտնիքի բաշխման մեթոդների հետազոտման ECC Sharing Explore համակարգը հնարավություն է տալիս հետազոտել ինչպես կոդերով, այնպես էլ գոյություն ունեցող այլ բաշխման մեթոդները: Համակարգը հնարավորություն է տալիս համեմատել բաշխման մեթոդները ըստ արագագործության:

Գրականություն

1. Аграновский А.В., Балакин А.В., Ьади Р.А. Классические шифры и методы их криптоанализа // Информационные технологии. 2001. N 10.
2. Шнайер Б. Прикладная криптография. – М.: Издательство ТРИУМФ, 2003 – 816 с.: ил.
3. Fridrich J. Steganography in digital media: principles, algorithms, and applications. Cambridge University Press, 2009. P 437.
4. Аграновский А.В., Балакин А.В. Стеганография в тексте // Труды конференции «Безопасность информационных технологий». Т.2. – Пенза, 2001. С. 15-16.
5. Генне О.В., Основные положения стеганографии // Защита информации. Конфицент. 2000. N 3. С. 20.
6. Аграновский А.В., Жижелев А.В., Хади Р.А., Балакин А.В. Оценка уровня скрытности встраивания данных в стеганографических системах первого поколения // Шестая международная конференция «Комплексная защита информации», ВНИИПВТИ, - Москва, 2002.
7. Маркаров В., Хачатуров А., “Обеспечение высокой стойкости стеганографической системы за счет реконфигурируемости структуры” // Сборник материалов годичной конференции ГИУА. – Ереван, 2008. – Т. 2. – С. 438-441.
8. Shamir A., How to share a secret. Communications of the ACM, P 612–613, 1979.

9. Blakley G. R., One-Time Pads are Key Safeguarding Schemes, Not Cryptosystems Fast Key Safeguarding Scheme (Threshold Scheme Exists), Proceeding of the 1980 Symposium on Security and Privacy, IEEE Computer Society. Apr 1980, pp 108-113.
10. Berkovits S., How to Broadcast a Secret // Advances in Cryptology EUROCRYPT '91 Proceedings. Springer-Verlag. 1991. P. 535-541.
11. Simmons G.J. Contemporary Cryptology. The Science of Information Integrity. – IEEE Press, 1991, P. 393.
12. Li Z., Xue T., Lai H., Secret sharing schemes from binary linear codes, Information Sciences, 180 (2010), pp. 4412 – 4419.
13. Massey J.L, Minimal Codewords and Secret Sharing, in Proceedings of the 6th Joint Swedish-Russian International Workshop on Information Theory, 1993, pp. 276–279.
14. Massey J. L., Some applications of coding theory in cryptography, Codes and Ciphers: Cryptography and Coding IV, pp. 33–47, 1995.
15. McEliece R.J., Sarwate D.V, On Sharing Secrets and Reed-Solomon Codes, Communications of the ACM, 24 (1981), pp. 583–584.
16. Tan X., Wang Z., New secret sharing scheme based on linear code, Applied Mathematics-A Journal of Chinese Universities, 19 (2004), pp. 160–166.
17. Tentu A.N, Paul P., Venkaiah V.C., Ideal and Perfect Hierarchical Secret Sharing Schemes based on MDS codes, IACR Cryptology ePrint Archive, 2013 (2013), p. 189.
18. Yuan J., Ding C., Secret Sharing Schemes from Three Classes of Linear Codes, IEEE Transactions on Information Theory, 52 (2006), pp. 206–212.

19. Berlekamp E. R., Algebraic Coding Theory, Aegean Park Press, 1984.
20. Lint J. V, Introduction to Coding Theory, Springer, 1999.
21. Wei V. K., Generalized Hamming Weights for Linear Codes, IEEE Transactions on Information Theory, 37(5), pp. 1412–1418, 1991.
22. Reed I.S, Solomon G., Polynomial Codes Over Certain Finite Fields, Journal of the Society for Industrial and Applied Mathematics, 8 (1960), pp. 300– 304.
23. Cohen G., Honkala I., Litsyn S., Lobstein A., Covering Codes, North Holland, Amsterdam, 1997.
24. Beimel A., Chor B., Universally ideal secret sharing schemes, IEEE Trans. on Information Theory, 40(3), pp. 786-794, 1994.
25. Simmons G.J, An Introduction to Shared Secret and/or Shared Control Schemes and Their Application, Wiley-IEEE Press, 1992, pp. 441–497.
26. Beiter M., Secret Sharing Schemes on General Access Structures, PhD thesis, University at Tubingen, 2008.
27. Asmuth C., Bloom J., A modular approach to key safeguarding, IEEE Transactions on Information Theory, 29 (1983), pp. 208–210.
28. Brickell E. F., Some ideal secret sharing schemes, Journal of Combin. Math. and Combin. Comput. 6, pp. 105-113, 1989.
29. Ding C., Kohel D., Ling S., Secret sharing with a class of ternary codes, Theor. Comp. Sci., vol. 246, pp. 285–298, 2000.

30. Ding C., Yuan J., Covering and secret sharing with linear codes, in Discrete Mathematics and Theoretical Computer Science (Lecture Notes in Computer Science). Berlin, Germany: Springer-Verlag, 2003, vol. 2731, pp. 11–25.
31. C. Ding, Yuan J., Secret sharing schemes from two-weight codes, in Proc. R. C. Bose Centenary Symp. Discrete Mathematics and Applications, Kolkata, India, Dec. 2002.
32. Karnin E.D., Greene J. W., Hellman M. E., On secret sharing systems, IEEE Trans. Inf. Theory, vol. IT-29, no. 1, pp. 35–41, Jan. 1983.
33. Pieprzyk J., Zhang X. M., Ideal threshold schemes from MDS codes, in Information Security and Cryptology - Proc. of ICISC 2002 (Lecture Notes in Computer Sceince). Berlin, Germany: Springer-Verlag, 2003, vol. 2587, pp. 269–279.
34. Cramer R., Damg I. , Maurer U., General secure multi-party computation from any linear secret-sharing scheme, In Advances in Cryptology–EUROCRYPT ’00 Springer, pp. 316–334, 2000.
35. Хемчян А., Арутюнян С. “Система распределения секрета на основе кодов, исправляющих ошибки” // Системный администратор – Москва, Россия, 2014, N4(137), С. 81-82
36. Margarov G., Khemchyan A. “Secret sharing based on error-correcting codes” //Proceedings of national polytechnic university of Armenia, information technologies, electronics, radio engineering – Երևան, Հայաստան, 2015, Հ.1, N1, Է. 62-67

37. Khemchyan A. “Secret sharing based on BCH error correction code” //Proceedings of the Conference Computer Science and Information Technologies (CSIT-2015) – Yerevan, Armenia, 2015, P. 280-282
38. Խեմչյան Ա. “Գաղտնիքի բաշխման շեմային սխեմա՝ սխալներ ուղղող Հեմմինգի կոդի հիման վրա” // ՀԱՊՀ Լրաբեր-82, գիտական և մեթոդական հոդվածների ժողովածու – Երևան, Հայաստան, 2016, Հ.1, N1,.Է. xxx-yyy
39. Хемчян А. “Пороговые схемы разделения секрета и коды исправляющие ошибки” // Тезисы докладов международной научно-практической конференции молодых ученых и студентов – Киев, Украиа, 2016, С. 216-217
40. Хемчян А. “Распределение данных на основе кодов, исправляющих ошибки” // Науковий журнал Безпека інформації – Киев, Украина, 2016, С. 261-264
41. Khemchyan A. “Distributed Data Storage in Cloud Systems Based on Error Correcting Codes” // Meeting Security Challenges Through Data Analytics and Decision Support – 2016, P. 287-292
42. Khemchyan A., Harutyunyan S. “A New (k,n)-Threshold Secret Sharing Scheme Based on Error-Correcting Codes” // Proceedings of the Conference World Congress on Internet security (WorldCIS2016) – London, United Kingdom, 2016, P. 92-96.
43. Hovsepyan V., Khemchyan A., Atayan B. “Data Security and Backup in Cloud Envirement” // Proceedings of the Conference World Congress on Internet security (WorldCIS2016) – London, United Kingdom, 2016, P. 101-105.

44. Хемчян А. “Новая (k,n) пороговая схема распределения секрета на основе кодов, испровляющих ошибки” // Сборник научних статей XIII международной научно-техноческой конференции Новые информационные технологии и системы ("НИТиС-2016") – Пенза, Россия, 2016, С. 260-262.
45. Koopman P. "32-Bit Cyclic Redundancy Codes for Internet Applications" // The International Conference on Dependable Systems and Networks, 2002, P. 459 468.
46. Bose R., Ray-Chaudhuri K., "On A Class of Error Correcting Binary Group Codes" // Information and Control vol. 3, 1960, P. 68-79.
47. Вернер М., “Основы кодирования” // Техносфера, 2004, С. 152-154.
48. Морелос-Сарагоса Р., “Исскуство помехоустойчивого кодирования. Методы, алгоритмы, применение” //Техносфера, 2005, С. 23-26.
49. Орлов В. А., Филиппов Л. И., “Теория информации в упражнениях и задачах”, Высшая школа, 1976, С. 57-64.
50. Потапов В. Н., “Теория информации. Кодирование дискретных вероятностных источников” // Новосибирск, 1999, С. 110-117.
51. Питерсон У., Уэлдон Э., “Коды, Исправляющие ошибки” // Мир, 1976, С. 167-175.
52. Мак-Вильямс Ф., Дж., Слоэн Н., Дж. А., “Теория кодов, исправляющих ошибки” // Связь, 1979, С. 87-92.
53. Берлекэмп Е. Р., “Алгебраическая теория кодирования” // Мир, 1971, С 275-281.
54. Галлагер Р. Г., “Теория информации и надежная связь” // Сов. радио, 1974, С. 56-58.

55. Касами Т., Токура Н., Ивадари Е., Инагаки Я., “Теория кодирования” // Мир, 1978, С 375-379.
56. Cohen G., Honkala I., Litsyn S., Lobstein A., “Covering codes” // Netherlands: Elsevier, 1997, P. 540-544.

Նկարների ցանկ

Աղյուսակների ցուցակ