Е.Ц. АЛАВЕРДЯН, Х.В. КЕРАМАТ

УКРЕПЛЕНИЕ ШИФРОВАНИЯ ОТКРЫТЫМ КЛЮЧОМ С ПРИМЕНЕНИЕМ МНОГОУРОВНЕВЫХ ДАННЫХ

Представлены три метода защищенного обмена данными, а именно: лишение хакеров их целостного знания цифровых сетевых систем; постулирование численной базы, отличной от двоичной, и достижение системы шифрования посредством матрицы перестановки битов размерности семь, при этом превращая ее в достаточно конфузную для злоумышленников; обеспечение очевидной высокой скорости, чем в среднем передача данных по сети. Приведена алгоритмическая реализация для предлагаемых резолюций.

Ключевые слова: блочный шифр, шифрование открытым ключом, односторонняя функция, многоуровневые данные.

ՀՏԴ 004.312.26:621.394.147.3

Ա.Ա. ԽԵՄՉՅԱՆ

ԳԱՂՏՆԻՔԻ ԲԱՇԽՄԱՆ ՇԵՄԱՅԻՆ ՍԽԵՄԱ՝ ՀԱՄԱՊԱՏԱՍԽԱՆ ՍԽԱԼՆԵՐ ՈՒՂՂՈՂ ՀԵՄՄԻՆԳԻ ԿՈԴԻՆ

Մշակված է գաղտնիքի բաշխման շեմային սխեմա, որի հիմքում ընկած է սխալներ ուղղող Հեմմինգի կոդը: Նկարագրված է Հեմմինգի կոդի այն մոդիֆիկացիան, որն օգտագործվել է բաշխման սխեմայում: Ներկայացված շեմային սխեման հարմար միջոց է մեծ ծավալի գաղտնի ինֆորմացիա բաշխելու համար, երբ կարևոր է նաև արագագործությունը: Ներկայացված են նոր և Շամիրի մեթոդների համեմատական գրաֆիկները:

Առանցքային բառեր. գաղտնիքի բաշխում, շեմային սխեմա, սխալներ ուղղող կոդեր, սխալներ հայտնաբերող կոդեր, Հեմմինգի կոդ:

Ներածություն: Ժամանակակից պայմաններում տվյալների փոխանակման զգալի մասն իրականացվում է գլոբալ բաց ցանցերի միջոցով, որոնցով փոխանցվում է մեծ ծավալիկարևոր տեղեկատվություն: Այս հարցի առանձնակի արդիականությունը պայմանավորված է էլեկտրոնային բիզնեսի զարգացմամբ և էլեկտրոնային փաստաթղթերի շրջանառությամբ: Այս ամենին զուգահեռ միշտ անհրաժեշտություն է առաջանում ապահովել բազմակողմանի տեղեկատվական անվտանգություն՝ գաղտնագրում, ցանցային պաշտպանություն, գաղտնի տեղեկատվության (գաղտնի բանալու) պաշտպանություն և այլն [1]: Ժամանակակից քոմփյութերները, հաշվարկների տեխնոլոգիաները հնարավոր դարձրին դեռ վերջերս գործնականորեն ապահով համարվող գաղտնագրային համակարգերի մեծամասնության կոտրումը: Տվյալների պաշտպանության ժամանակակից եղանակները հատկորոշում են անվտանգության “դիմացկունության կանոնները”, օրինակ՝ սպառողների սարքերում պաշտպանված տվյալներ ապահովելու համար: Ընդհանրապես այս կանոնները ենթադ-

րում են, որ ծրագրային ապահովման միջոցով իրականացված անվտանգությունը դիմացկունակ չէ:

Գաղտնիքի բաշխում: Վերը նկարագրված դժվարությունների հաղթահարման իրական միջոցներ են գաղտնագրային մեթոդները կիրառող կանոնակարգերի մշակումը և օգտագործումը, որոնք կապահովեն գաղտնիությունը: Համակարգի անվտանգությունն ապահովելու համար գաղտնագրման ալգորիթմների մեծ մասը հիմնվում է որոշակի գաղտնի բանալու պաշտպանության վրա: Գաղտնի բանալիները չպետք է անմիջական տեսքով հայտնվեն ինչ-որ տեղ համակարգում, առանց համապատասխան ֆիզիկական պաշտպանվածության ապահովման, սակայն այդ պայմանը անիրագործելի է: Այս խնդրի լուծման համար բանալիները պետք է մասնատվեն առնվազն երկու մասի և պահվեն տարբեր տեղերում: Այստեղից առաջացել է բաշխված գաղտնիքի գաղափարը: «Բաշխված գաղտնիքի» գաղափարի օգտագործումը շատ արդյունավետ է, որի դեպքում օգտագործողների միջև կան բաշխված արժեքներ, որոնց (կամ դրանցից մի քանիսի) համակցությունից ստացվում է բանալին: Այսպիսով, համակարգի իրական անվտանգությունը կարելի է ապահովել գաղտնի բանալիները բաշխելու եղանակով: Օրինակ, կարելի է ունենալ այնպես ստեղծված յոթ բաղադրամասեր, որ դրանցից ցանկացած չորսը ստեղծեն եզակի բանալի, իսկ ցանկացած երեք բաղադրամասերի իմացությունը չտա ոչ մի տեղեկություն բանալու մասին: Այս ալգորիթմներն անվանվում են շեմային սխեմաներ: Շեմային սխեմաներից ամենատարածվածը Շամիրի շեմային սխեման է: Շամիրի սխեմայի դեպքում բաղադրամասերը հաշվարկվում են ստորև ներկայացված բանաձևերով:

$$k_i = F(i) = \left(a_{k-1} \times i^{k-1} + a_{k-2} \times i^{k-2} + \ \ldots + \ a_1 \times i + M\right) mod\ p, \qquad (1)$$

որտեղ k-ն շեմի արժեքն է, M-ը՝ գաղտնի թիվը, p-ն՝ գաղտնիքից մեծ պարզ թիվ: Ինչպես տեսնում ենք, k-ն ցուցիչում է, և շեմի մեծ արժեքի դեպքում այդ հաշվարկները էականորեն բարդանում են: Շամիրի մեթոդի դեպքում վերականգնումը կատարվում է Լագրանժի միջարկման բազմանդամով, որը նույպես շեմի մեծ արժեքների դեպքում բարդ է: Երկու դեպքում էլ համակարգիչը այդ գործողությունները կատարում է բավականին դանդաղ: Եթե հաշվի առնենք, որ գաղտնիքի բաշխումը կիրառվում է ոչ միայն գաղտնի բանալիներ բաշխելու, այլև մեծ ծավալի գաղտնի ֆայլեր բաշխելու համար նույնպես, ապա խնդիր է առաջանում ունենալ ավելի արագագործ գաղտնիքի բաշխման շեմային սխեմա: Ինչպես հայտնի է, բարձր անվտանգության ապահովումը՝ տվյալների բաշխման եղանակով, հիմնված է որոշակի ավելցուկայնության մակարդակի ապահովման վրա: Տվյալների ավելցուկայնություն ապահովում է նաև կոդավորումը, որը կարելի է դիտել որպես Շամիրի սխեմայի այլընտրանք՝ տվյալների բաշխումն իրականացնելու համար: Եվ քանի որ կոդավորումը հիմնա-

կանում իրականացվում է տրամաբանական գործողություններով, հետևաբար՝ նոր սկզբունքով կառուցված տվյալների բաշխման համակարգը պետք է ունենա ավելի բարձր արագագործություն: Առաջարկվում է կառուցել գաղտնիքի բաշխման շեմային սխեմա՝ հիմնված սխալներ ուղղող Հեմմինգի կոդի վրա [2,3]: Քանի որ գոյություն ունեն Հեմմինգի կոդի տարբեր մոդիֆիկացիաներ, այդ պատճառով դիտարկենք կոդի այն տարբերակը, որը օգտագործվել է այս բաշխման ալգորիթմում:

Հեմմինգի կոդ: Հեմմինգի կոդի աշխատանքը հիմնված է բիթային հաջորդականությունում մեկերի քանակի զույգության հայտանիշի վրա: Կոդավորվող հաջորդականությանն ավելացվում է այնպիսի բիթային շարք, որ ստացված հաջորդականությունում մեկերի քանակը լինի զույգ:

$$r_1 = i_1 \oplus i_2 \oplus \dots \oplus i_k, \tag{2}$$

որտեղ $\oplus$ նշանակում է գումարում ըստ մոդուլ 2-ի: Դիտարկենք Հեմմինգի (7,4) [3]: Այս կոդի դեպքում հսկիչ բիթերը ստացվում են (3)-ում ներկայացված բանաձևերով.

$$\begin{aligned} r_1 &= i_1 \oplus i_2 \oplus i_3, \\ r_2 &= i_2 \oplus i_3 \oplus i_4, \\ r_3 &= i_1 \oplus i_2 \oplus i_4: \end{aligned} \tag{3}$$

Կոդավորելու համար մուտքային հաջորդականությունը բազմապատկվում է գեներատոր մատրիցով (4).

$$\begin{pmatrix} i_1 & i_2 & i_3 & i_4 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} i_1 & i_2 & i_3 & i_4 & r_1 & r_2 & r_3 \end{pmatrix} : \tag{4}$$

Հնարավոր սխալի հայտնաբերման և ուղղման համար կառուցվում է հայտանիշերի հաջորդականությունը՝ $S = (S_1, S_2, S_3)$, որտեղ S_1, S_2, S_3 արժեքները ստացվում են (5) արտահայտությամբ.

$$\begin{aligned} S_1 &= r_1 \oplus i_1 \oplus i_2 \oplus i_3, \\ S_2 &= r_2 \oplus i_2 \oplus i_3 \oplus i_4, \\ S_3 &= r_3 \oplus i_1 \oplus i_2 \oplus i_4: \end{aligned} \tag{5}$$

Հայտանիշերի հաշվարկը ավելի արագ հնարավոր է կազմակերպել գեներատոր մատրիցի միջոցով (6):

$$(i_1 \quad i_2 \quad i_3 \quad i_4 \quad r_1 \quad r_2 \quad r_3)\begin{pmatrix}1 & 0 & 1\\ 1 & 1 & 1\\ 1 & 1 & 0\\ 0 & 1 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{pmatrix}=(S_1 \quad S_2 \quad S_3)\,: \tag{6}$$

Աղ. 1-ում ներկայացված են Հեմմինգի (7,4) կոդի բոլոր կոդաբառերը: (0,0,0) հայտանիշը նշանակում է, որ հաջորդականությունը վնասված չէ: Յուրաքանչյուր ոչ զրոյական հայտանիշի համապատասխանում է որևէ սխալ, որը ուղղվում է ապակոդավորման ընթացքում:

Աղյուսակ 1

Հեմմինգ կոդի բոլոր հնարավոր կոմբինացիաները

i_1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
i_2	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
i_3	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
i_4	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
r_1	0	0	1	1	1	1	0	0	1	1	0	0	0	0	1	1
r_2	0	1	1	0	1	0	0	1	0	1	1	0	1	0	0	1
r_3	0	1	0	1	1	0	1	0	1	0	1	0	0	1	0	1

Աղ. 2-ում ներկայացված են բոլոր հնարավոր սխալներին համապատասխանող հայտանիշերը: Վերը նկարագրված Հեմմինգի կոդը կարող է ուղղել 1 սխալ: Այսուհետ այդ կոդը կնշանակենք Հեմմինգ (7,4,1):

Աղյուսակ 2

Սխալների հայտանիշեր

Հայտանիշ	001	010	011	100	101	110	111
Սխալի դիրքը	0000001	0000010	0001000	0000100	1000000	0010000	0100000
Վնասված բիթը	r_3	r_2	i_4	r_1	i_1	i_3	i_2

Ալգորիթմի նկարագրությունը: Բաշխվող գաղտնի ֆայլի յուրաքանչյուր 4 բիթը կոդավորվում է, և ստացվում համապատասխան կոդաբառը (7 բիթ): Ստաց-

ված կոդաբառերից ստացվում է նկ.1-ում պատկերված երկչափ զանգվածը (դա բաշխվող գաղտնի ֆայլի կառուցվածքն է Հեմմինգ (7,4,1) կոդով կոդավորելուց հետո): Այստեղ k-ն կոդաբառի երկարությունն է (այսդեպքում k=7), իսկ q-նկախված է բաշխվող ֆայլի ծավալից: Այս զանգվածի յուրաքանչյուր տող ներկայացնում է Հեմմինգ (7,4,1) կոդի կոդաբառ, իսկ դա նշանակում է, որ այդ 7 բիթում կամայական 1 սխալ բիթկարելի է ուղղել:

Հաշվի առնելով այս հատկությունը՝ կատարենք բաշխում ըստ սյուների: Եթե յուրաքանչյուր սյուն դիտարկենք որպես առանձին բաղադրամաս, ապա ակնհայտ է, որ կկարողանանք վերականգնել նախնական ֆայլը կամայական 1 բաղադրամասի վնասվելու կամ կորստի դեպքում: Ստացվում է 7/4 շեմային սխեմա: Սակայն ակնհայտ է,որ այս դեպքում խախտվում է շեմային սխեմաներն ներկայացվող պայմաններից մեկը, այն է՝շեմից պակաս բաղադրամասերի միավորումից հնարավոր չլինի վերականգնել գաղտնի ինֆորմացիան: Այս դեպքում ակնհայտ է, որ առաջին 4 սյուների միավորումը բավարար է գաղտնիքի վերականգնման համար: Այս պատճառով կատարվում է խմբավորում, և կողմերին տրվում են սյուների խումբ:

1	2	3	:::	k
V1,1	V1,2	V1,3	:::	V1,k
V2,1	V2,2	V2,3	:::	V2,k
V3,1	V3,2	V3,3	:::	V3,k
V4,1	V4,2	V4,3	:::	V4,k
V5,1	V5,2	V5,3	:::	V5,k
:::	:::	:::	:::	:::
Vq,1	Vq,2	Vq,3	:::	Vq,k

Նկ. 1. Կոդավորված ֆայլի կառուցվածքը

Խմբավորում: Որպեսզի բարձրացնենք գաղտնակայունությունը, և բաղադրամասերի ծավալները հավասար լինեն բաշխվող ֆայլի ծավալին՝ կատարենք խմբավորում: Որպեսզի ստացված բաղադրամասերն ունենան նույն ծավալը, ինչ սկզբնական ֆայլը, անհրաժեշտ է յուրաքանչյուր մասում վերցնել 4 սյուն (որոշակի մեթոդով): Օրինակ՝ խմբավորելով ըստ աղ. 3-ի, կստանանք 4/2 շեմային սխեմա: Յուրաքանչյուր տողը ներկայացնում է առանձին բաղադրամաս: Աղյուսակներում լրացված թվերը 1-7 բիթերի (սյուների) համարներն են: Աղ. 3-ից երևում է, որ կամայա-

կան 3 սյուն միավորելուց հետո պակասում է 1 բիթ: Օգտագործելով Հեմմինգ (7,4,1) կոդի դեկոդավորման ալգորիթմը՝ կարելի է վերականգնել այդ 1 բիթը: Ստացվում է, որ ըստ աղ. 3-ի բաշխման դեպքում ունենում ենք 4/2 շեմային սխեման: Գաղտնի ֆայլը բաշխվում է 4 մասի և կամայական 2-ով վերականգնվում է: Աղ. 4-ում ներկայացված է 3/2 շեմայինսխեման: Նույն մեթոդով բաշխումը կատարվում է այլ բաղադրամասի և շեմի արժեքների դեպքում:

Աղյուսակ 3

4/2 շեմային սխեմա

Բաղադրամաս 1	3	6	2	7
Բաղադրամաս 2	1	6	7	5
Բաղադրամաս 3	5	4	2	6
Բաղադրամաս 4	7	5	3	4

Աղյուսակ 4

3/2 շեմային սխեմա

Բաղադրամաս 1	5	2	6	4
Բաղադրամաս 2	4	7	3	5
Բաղադրամաս 3	6	2	7	3

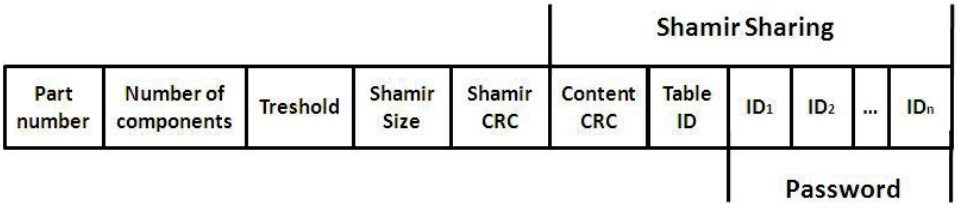
Վերադասավորում: Ենթադրելով, որ պոտենցիալ հակառակորդին կարող է հայտնի լինել, թե որ աղյուսակով է կատարվել բաշխումը՝ հանգում ենք նրան, որ նա, ունենալով ավելի քիչ բաղադրամասեր, կկարողանա որոշակի տեղեկություն ստանալ նախնական գաղտնի տեղեկության մասին: Սա հակասում է այն պայմաններին, որոնք ներկայացվում են շեմային սխեմաներին, այն է՝ շեմից պակաս բաղադրամասերի միավորումից հնարավոր չլինի որևէ տեղեկություն ստանալ նախնական գաղտնի տեղեկատվության վերաբերյալ: Չնայած այդ ճանապարհով գաղտնիքի վերականգնումը ամբողջությամբ հնարավոր չէ, այնուամենայնիվ, այդ հակասությունը լուծելու և բաշխման այս մեթոդը ավելի կատարյալ դարձնելու համար առաջարկվում է նոր մոտեցում՝ վերադասավորում: Մինչև բաշխման գործընթացը սկսելը՝ հարկավոր է բաշխման համար ընտրված աղյուսակում կատարել վերադասավորումներ: Այդ գործընթացը կատարվում է հետևյալ քայլերի հերթականությամբ.

1.Գեներացվում են 2*(m/3) քանակությամբ ID համարներ (դրանք աղյուսակում համապատասխան սյուների համարներն են): Այդ ID-ների համախումբը կան-

վանենք բաշխման գաղտնաբառ: Այն կունենա հետևյալ տեսքը՝ ID1|ID2|ID3…IDn: Այստեղ m-ը կոդաբառի երկարությունն է:

2. Արդեն ընտրված աղյուսակում կատարվում է սյուների տեղափոխություն (ID1-ը ID2-ի հետ և այլն): Քանի որ գեներացվել են զույգ քանակով ID համարներ, ապա այս գործընթացը միշտ հնարավոր է:

Այս վերադասավորումից հետո արդեն հակառակորդը չի կարող իմանալ, թե որ ֆայլի որ բիթում կոդավորված կոդաբառի որ բիթն է՝ չնայած նրան, որ նրան հայտնի կլինի աղյուսակը (մինչև վերադասավորումը): Այս ամենից հետո անհրաժեշտություն է առաջանում, որ վերականգնող կողմը մինչև վերականգնելը նույնպես իմանա այդ վերադասավորման գաղտնաբառը: Բնականաբար, այդ գաղտնաբառը հնարավոր չէ բաղադրամասերում բաց տեսքով ունենալ: Առաջարկվում է այդ բաղադրամասը բաշխել կողմերի մեջ և բաշխված տարբերակով տեղադրել ստեղծվող ֆայլերում: Քանի որ այդ գաղտնաբառի ծավալը մեծ չէ, հետևաբար՝ այն կարող ենք բաշխել Շամիրի մեթոդով, և դա 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 ֆայլի բաշխումը (նոր մեթոդով): Այս մեթոդը հնարավորություն է տալիս նաև բաշխված պահել օգտագործվող աղյուսակի համարը: Ասվածը ավելի պատկերավոր ներկայացված է նկ.2-ում, որը բաշխվող բաղադրամաս-ֆայլերի գլխամասի (Header) կառուցվածքն է:



Նկ. 2. Բաղադրամաս-ֆայլերի գլխամասի կառուցվածքը

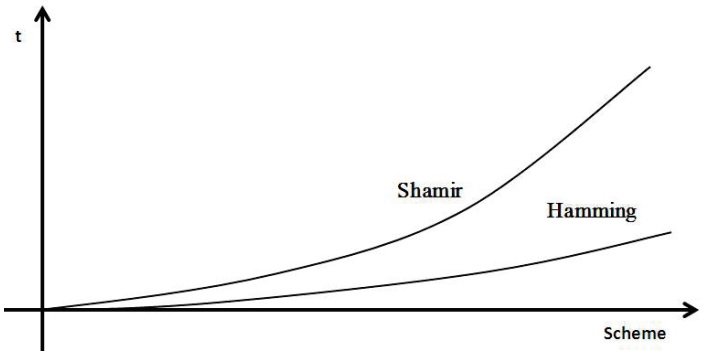
Գլխամասը բաղկացած է հետևյալ հատվածներից.

- Content CRC – այս դաշտում հաշվվում և պահվում է բունբաշխված մասի (առանց գլխամասի) CRC32 արժեքը: Նախատեսված է այդ հատվածում հնարավոր փոփոխությունն արձանագրելու համար:
- Table ID – ընտրված կոդի աղյուսակի համարն է, որով կատարվել է բաշխումը: Յուրաքանչյուր կոդ ունի բազում աղյուսակներ տարբեր շեմային բաշխումների համար:
- ID1ID2ID3…IDn – գեներացված ID համարներն են, որոնք միասին կազմում են բաշխման գաղտնաբառը:

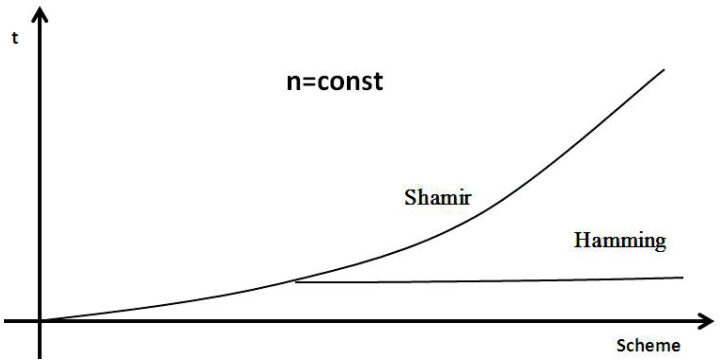
- Header CRC - Shamir Sharing հատվածի CRC32 արժեքն է և նախատեսված է այդ հատվածում հնարավոր սխալները հայտնաբերելու համար:
- Header Size – Shamir Sharing հատվածի բայթերի քանակը: Այս տվյալը անհրաժեշտ է ծրագրին՝ այդ հատվածը ճիշտ կարդալու համար:
- Part Number – բաղադրամասի հերթական համարը:
- Number of components – ընդհանուր բաղադրամասերի քանակը:
- Treshold - բաշխման շեմի արժեքը: Այս տվյալը անհրաժեշտ է Shamir Sharing հատվածը վերականգնելու համար:
- Content - գաղտնի տեղեկատվության բաշխված հատվածը (նոր մեթոդով):

Այս տարբերակով բաշխելուց հետո հակառակորդը, ունենալով շեմից պակաս քանակով բաղադրամասեր, բացարձակ ոչինչ չի կարող իմանալ նախնական գաղտնի ֆայլի մասին: Նա կարող է իմանալ միայն գաղտնի ֆայլի ծավալը, որը գաղտնիք չէ (դա հայտնի է բոլոր մասնակիցներին):

Նկ. 3 և 4-ում պատկերված են Շամիրի և Հեմմինգի կոդով բաշխման արագագործությունների համեմատական գրաֆիկները:



Նկ. 3. Արագագործությունների համեմատական գրաֆիկը



Նկ. 4. Արագագործությունների համեմատական գրաֆիկը՝ հաստատուն քանակով բաղադրամասերի դեպքում

Եզրակացություն: Այսպիսով, մշակվել է գաղտնիքի բաշխման շեմային սխեմա, որը հիմնված է սխալներ ուղղող Հեմմինգի կոդի վրա: Մշակված սխեմայի առավելությունը, Շամիրի սխեմայի համեմատ, արագագործությունն է: Դա պայմանավորված է Հեմմինգի կոդի արագագործությամբ: Այս մեթոդը ավելի շատ նախատեսված է մեծ ծավալի գաղտնի ինֆորմացիա բաշխելու համար, քանի որ բանալիներ բաշխելու համար Շամիրի մեթոդը ամբողջովին բավարարում է: Մշակված ալգորիթմը կիրառելի է այլ սխալներ ուղղող կոդերի դեպքում նույնպես: Անհրաժեշտ է շարունակել հետազոտությունը և ընդգրկել այլ կոդեր ևս:

ԳՐԱԿԱՆՈՒԹՅԱՆ ՑԱՆԿ

1. **Bruce Schneier** Applied Cryptography. -John Wiley & Sons, 1996. - Second Edition ISBN 0-471-12845-7.-784p.
2. **Peterson W.W.**, **Weldon E.J.** Error-Correcting Codes. -The Massachusetts Institute of Technology,1972. - Second Edition. ISBN 0-2621-6039-0.-572p.
3. **Assmus E.F.**, **Key J.D.** Designs and Their codes. -Cambridge University Press,1992.- ISBN 0-5214-5839-0.-364p.

А.А. ХЕМЧЯН

ПОРОГОВАЯ СХЕМА РАСПРЕДЕЛЕНИЯ СЕКРЕТА НА ОСНОВЕ КОДА ХЕММИНГА, ИСПРАВЛЯЮЩЕГО ОШИБКИ

Разработанновый метод распределения секрета на основе кода Хемминга, исправляющего ошибки. Описана модификация кода Хемминга, которая была использована в алгоритме. Приведенная пороговая схема является удобным средством для распределения больших объемов секретной информации, где важно быстродействие. Представлены сравнительные графики быстродействия нового метода и пороговой схемы Шамира.

Ключевые слова: распределение секрета, пороговая схема, коды, исправляющие ошибки, код Хемминга.

A.A. KHEMCHYAN

THE TRESHOLD SCHEME OF SECRET SHARING BASED ON THE HAMMING ERROR-CORRECTING CODE

A new method based on Hamming's code for secret sharing threshold scheme has been developed. The modification of Hamming's code used in the algorithm is described. In case of sharing large amounts of secret information, the performance speed is the key aspect. The described method is a relevant solution for that problem. The performance comparison of Shamir's scheme and the new developed method is shown here.

Keywords: secret sharing, threshold scheme, error correction codes, error detection codes, Hamming code.