ՀԱՅԱՍՏԱՆԻ ՀԱՆՐԱՊԵՏՈՒԹՅԱՆ ԿՐԹՈՒԹՅԱՆ ԵՎ ԳԻՏՈՒԹՅԱՆ ՆԱԽԱՐԱՐՈՒԹՅՈՒՆ

ՀԱՅԱՍՏԱՆԻ ԱԶԳԱՅԻՆ ՊՈԼԻՏԵԽՆԻԿԱԿԱՆ ՀԱՄԱԼՍԱՐԱՆ

Հովսեփյան Վլադիմիր Հովսեփի

ԱՄՊԱՅԻՆ ՏԵԽՆՈԼՈԳԻԱՆԵՐՈՎ ԱՆՎՏԱՆԳ ԱՇԽԱՏԵԼՈՒ ԼՐԱՑՈՒՑԻՉ ՄԻՋՈՑՆԵՐԻ ՄՇԱԿՈՒՄ

ԱՏԵՆԱԽՈՍՈՒԹՅՈՒՆ

05.13.04 «Հաշվողական մեքենաների, համալիրների, համակարգերի և ցանցերի մաթեմատիկական և ծրագրային ապահովում» մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման համար

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

Երևան-2017

ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆ

Ներածություն 5

Գլուխ 1: Ամպային տեխնոլոգիաների անվտանգության ապահովման հիմնախնդիրները

1.1. Ամպային տեխնոլոգիաների հիմնահարթակը և ենթակառուցավածքները 11

1.2. Ամպային տեխնոլոգիաների սպառնալիքների և առաջարկված լուծումների վերլուծություն 17

1.3. Իրերի ինտերնետի կիրառման առանձնահատկությունները 21

1.3.1. Բանալիների կառավարման սխեմաները 27

1.3.2. Գործառույթների ամբողջականության ապահովումը իրերի ինտերնետ միջավայրում 31

1.4. Խնդրի դրվածքը 33

1.5. Գլուխ 1-ի եզրակացություն 35

Գլուխ 2: Իրերի ինտերնետ միջավայրում բանալիների բաշխման և կառավարման սխեմայի մշակումը

2.1. Իրերի ինտերնետ միջավայրում բանալու բաշխման մեթոդի մշակումը տարատեսակ իրերի միջև պաշտպանված կապի հաստատման նպատակով 36

2.2. Բաշխված բանալիների ենթակառուցվածքի ստեղծումը 41

2.3. Ինտերնետ իրերի դասակարգումը համապատասխան խմբերում 43

2.4. Անվտանգ կապուղու հաստատումն իրերեի ինտերնետ միջավայրում 46

2.5. Բանալիների բաշխման առաջարկված սխեմայի կայունության գնահատականը 49

2.6. Պարզ ինտերնետ իրերի տվյալների անվտանգ փոխանակման գաղտնահամակարգի մշակում 52

2.7. Գլուխ 2-ի եզրակացություն 66

Գլուխ 3: Բաշխված ցանցերում գործառույթների ամբողջականության ու անվտանգության ապահովման մեթոդի մշակումը 67

3.1.Գործառույթների ամբողջականությունը ապահովող արձանագրության մշակումը 67

3.2. Հեշ և POW ֆունկցիաների հետազոտում 72

3.3. Ինտերնետ իրերի գործառույթների ամբողջականության վերահսկման հեշ ալգորիթմը 83

3.4. Գլուխ 3-ի եզրակացություն 92

Գլուխ 4: Իրերի ինտերնետ միջավայրում բանալիների բաշխման եվ գործառույթների ամբողջականության ապահովման ծրագրային համակարգի իրականացումը 93

4.1. Մշակված ծրագրային իրականացման ընդհանուր բնութագիրը 93

4.2. Իրերի ինտերնետ միջավայրի վիրտուալացման համակարգը 95

4.3. Գլուխ 4-ի եզրակացություն 100

Եզրակացություն 101

Օգտագործված գրականության ցանկ.....102

Նկարների ցուցակ 108

Աղյուսակների ցուցակ 110

Հավելված 1..... 111

Հավելված 2..... 112

ՆԵՐԱԾՈՒԹՅՈՒՆ

<u>Աշխատանքի արդիականությունը:</u> Ժամանակակից տեղեկատվական տեխնոլոգիաների և հեռահաղորդակցության համակարգերի զարգացումը, մասնավորապես՝ ամպային տեխնոլոգիաների լայնամասշտաբ ներգրավումը մարդկային կենսագործունեության բոլոր բնագավառներում, նոր պահանջներ են առաջադրում տեղեկատվական անվտանգության ապահովմանը, առանց որի հնարավոր չէ այդ բնագավառների հետագա զարգացումը [1]:

Ամպային տեխնոլոգիաների բուռն զարգացող ոլորտներից է իրերի ինտերնետը: Ամպային ինտերնետ իրերի միջև փոխադարձ կապը հնարավորություն է տալիս կառավարել և ավտոմատացնել ընթացող գործառույթները: Ինտերնետ իրերը ստեղծված են որոշակի գործառույթներ իրականացնելու համար և, որպես կանոն, ունեն համեմատաբար ցածր հաշվողական հզորություն: Մյուս կողմից՝ ինտերնետ իրեր արտադրող ընկերությունները, ստեղծելով պարզ կիրառության և քիչ ռեսուրսներ ունեցող ինտերնետ իրեր, իրենց առջև տեղեկատվական անվտանգության ապահովման խնդիրներ չեն դնում: Տեղեկատվական անվտանգության տեսանկյունից վերոհիշյալ խնդիրն ունեցող ինտերնետ իրերը դասակարգվում են երկու դասի՝ սարքեր, որոնք ունակ են իրականացնել գաղտնագրային գործառույթներ, և սարքեր, որոնցում անհնար է կատարել գաղտնագրային որևէ գործառույթ՝ սահմանափակ հաշվողական ռեսուրսների պատճառով: Վերոհիշյալ ցածր արտադրողականության սարքերը կարելի է անվանել նաև պարզ ինտերնետ իրեր [2]:

Գաղտնագրային համակարգերի սահմանափակումները տարատեսակ իրերի ինտերնետ միջավայրում պահանջում են ինտերնետ իրերի համաձայնեցում գաղտնագրային համակարգի հետ՝ նախքան ամպային ցանցին միանալը: Վերոհիշյալ

հանգամանքն բացասական ազդեցություն է թողնում ամպային միջավայրի արդյունավետության վրա, միևնույն ժամանակ ինտերնետ իրերի գերակշիռ զանգվածի համար բանալիների բաշխման հայտնի համակարգերը կիրառելի չեն, քանի որ դրանք նախատեսված են միատեսակ սարքերի համար, մինչդեռ իրերի ինտերնետ միջավայրը ներառում է տարատեսակ սարքեր [2]:

Ինչպես նշվել է, գոյություն ունեն ինտերնետ իրերի այնպիսի տեսակներ, որոնք թույլ չեն տալիս կիրառել գաղտնագրային գործառույթներ՝ տեխնիկական հնարավորությունների բացակայության կամ համապատասխան ծրագրային ապահովման անհամատեղելիության պատճառով: Այս խնդիրը կարելի է լուծել ինտերնետ իրերում տեխնիկական հնարավորությունների ուժեղացման իրականացմամբ, որը սակայն անխուսափելիորեն կբերի իրերի ինքնարժեքի բարձրացմանը: Ուստի անհրաժեշտություն է առաջանում տվյալ խնդրի համար գտնել այնպիսի լուծում, որը չի պահանջում սարքերի տեխնիկական կամ ծրագրային փոփոխություններ:

Ինտերնետ իրերին բնորոշ է նաև ֆիզիկական ազդեցությունը արտաքին միջավայրի վրա, ուստի կարևորվում է ինտերնետ իրերի կողմից իրականացվող գործառույթների ամբողջականության ապահովումը: Գործառույթների ամբողջականության ապահովման արգելք է հանդիսանում նաև ինտերնետ իրերի խոցելիությունը ցանցային գրոհների, մասնավորապես՝ ներխուժումների և DDoS գրոհների նկատմամբ: Իրերի ինտերնետ միջավայրում վերոհիշյալ սպառնալիքների չեզոքացումը հեռակառավարման հնարավորություն ունեցող սարքերի քանակի աճին զուգընթաց դառնում է ավելի ու ավելի կարևոր: Այս ուղղությամբ մինչ այժմ գոյություն ունեցող լուծումները հիմնականում ծրագրային են, իրականացված են կենտրոնացված սերվերային միջավայրի համար և պահանջում են զգալի հաշվողական ռեսուրսներ:

<u>Աշխատանքի նպատակն է</u> մշակել ամպային տեխնոլոգիաներով անվտանգ աշխատելու լրացուցիչ միջոցներ՝ տվյալների անվտանգ փոխանակման, գործառույթների ամբողջականության ապահովման և ցանցային գրոհներից պաշտպանվելու համար: Այդ նպատակին հասնելու համար դրվել և լուծվել են հետևյալ խնդիրները՝

- Մշակել գաղտնագրային բանալիների կառավարման մեթոդ, որն ապահովում է միջավայրին միացվող սարքերի ինքնակարգավորումը և դրանց միջև տվյալների անվտանգ փոխանակումը:
- Մշակել անվտանգ հաղորդակցման մեթոդ ամպային միջավայրում պարզ սարքերի համար, որոնք սահմանափակ ռեսուրսների պատճառով գաղտնագրային ընթացակարգեր չեն ապահովում:
- Մշակել ամպային բաշխված ցանցերում գործառույթների ամբողջականությունն ապահովող մեթոդ, որը նաև կիրականացնի ամպային միջավայրի վրա հատուկ գրոհների բացահայտումը և դրանց կանխարգելումը:

<u>Գիտական նորույթ:</u>

- Մշակվել է տարատեսակ իրերի ինտերնետ միջավայրում բանալիների կառավարման մեթոդ, որն, ի տարբերություն գոյություն ունեցող լուծումների, տվյալների անվտանգ փոխանակման համար սարքերի նախնական կարգաբերում չի ենթադրում:
- Մշակվել է գաղտնահամակարգ՝ ամպային միջավայրում պարզ ինտերնետ իրերի հետ անվտանգ հաղորդակցման ապահովման նպատակով, որը, ի տարբերություն ոլորտում առկա լուծումների, ապահովում է անհրաժեշտ անվտանգություն և սարքերի տեխնիկական բնութագրերի փոփոխություն չի պահանջում:

- Մշակվել է բաշխված ցանցերում գործառույթների ամբողջականության ու անվտանգության ապահովման մեթոդ, որը կենտրոնացված սերվերի օգտագործում չի պահանջում և ապահովում է ինտերնետ իրերի տեղեկատվական անվտանգությունը միջավայրին բնորոշ գրոհների նկատմամբ:

<u>Աշխատանքի գործնական նշանակությունը:</u>

- Մշակվել է IKS ծրագրային համակարգը, որը հիմնված է բանալիների բաշխման մեթոդի վրա և հնարավորություն է ընձեռում ապահովել տվյալների անվտանգ փոխանակում տարատեսակ ինտերնետ իրերի միջև:
- Մշակվել է SIT ծրագրային համակարգը, որի ներդրումը համապատասխան սարքում ապահովում է տվյալների անվտանգ փոխանակում այն ինտերնետ իրերի համար, որոնք գաղտնագրային ընթացակարգեր չեն ապահովում:
- Մշակվել է իրերի ինտերնետ միջավայրում գործառույթների ամբողջականությունն ապահովող ծրագրային գործիքամիջոց, որն օգտագործելով մշակված արձանագրությունը, ապահովում է ինտերնետ իրերի անվտանգությունը միջավայրին հատուկ գրոհների նկատմամբ:

<u>Պաշտպանության են ներկայացվում հետևյալ դրույթները.</u>

- Գաղտնագրային բանալիների կառավարման մեթոդ, որն աշխատում է առանց նախնական կարգաբերման անհրաժեշտության և ապահովում է ինտերնետ իրերի միջև տվյալների անվտանգ փոխանակումը:
- Ամպային միջավայրում ինտերնետ իրերի հետ անվտանգ հաղորդակցման մեթոդ, որը կիրառելի է գաղտնագրային ընթացակարգերից զուրկ ինտերնետ իրերի համար:

- Բաշխված ցանցերում գործառույթների ամբողջականության ու անվտանգության ապահովման մեթոդ, որը նաև ապահովում է ինտերնետ իրերի տեղեկատվական անվտանգությունը միջավայրին բնորոշ գրոհների նկատմամբ:

<u>Ներդրումները:</u>

Ատենախոսության արդյունքները կիրառվում են.

- Կանխարգելիչ սրտաբանության կենտրոնում առօրյա պայմաններում հիվանդի առողջական վիճակի հետազոտության տվյալների անվտանգ փոխանցման գործընթացներում:
- «Կորիզ» ՍՊԸ-ում՝ ինտերակտիվ խաղային հարթակում ինտերնետ իրերի հետ կապված գործառույթների անվտանգության ապահովման նպատակով:

<u>Աշխատանքի արդյունքները զեկուցվել են.</u> 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), «Միջազգային գիտափորձնական ուսանողների և երիտասարդ գիտնականների գիտաժողով»-ում (Ազգային ավիացիոն համալսարան, 2016թ. ք. Կիև, Ուկրաինա), «Ինտերնետ անվտանգության համաշխարհային կոնգրես»-ում (WorldCIS-2016, 2016թ. ք. Լոնդոն, Մեծ Բրիտանիա), ՀԱՊՀ ՏԱԾԱ ամբիոնի գիտատեխնիկական սեմինարներում (2014-2017թ., ք. Երևան):

<u>Հետազոտման օբյեկտներն են</u> բանալիների բաշխման մեթոդները, բանալիների ենթակառուցվածքի ձևավորման մոդելները, գաղտնագրային համակարգերը և քաոսային հեշավորման ֆունկցիաները:

Հրապարակումներ: Ատենախոսության հիմնական արդյունքները տպագրված են 6 գիտական աշխատություններում, որոնք թվարկված են սեղմագրի վերջում:

Աշխատանքի կառուցվածքը և ծավալը: Ատենախոսությունը բաղկացած է ներածությունից, չորս գլխից, եզրակացությունից և 67 անուն օգտագործված գրականության ցանկից: Աշխատանքի ընդհանուր ծավալն է 110 էջ՝ ներառյալ 20 նկար: Հավելվածները կազմում են 2 էջ:

ԳԼՈՒԽ 1.

ԱՄՊԱՅԻՆ ՏԵԽՆՈԼՈԳԻԱՆԵՐԻ ԱՆՎՏԱՆԳՈՒԹՅԱՆ ԱՊԱՀՈՎՄԱՆ ՀԻՄՆԱԽՆԴԻՐՆԵՐԸ

1.1 Ամպային տեխնոլոգիաների հիմնահարթակը և ենթակառուցվածքները

Ամպային տեխնոլոգիաները տվյալների և հաշվարկների հասանելիության ցանցային մոդել են, երբ տվյալների պահպանման և մշակման համար անհրաժեշտ ռեսուրսները և լուծումները տրամադրվում են վարձակալման սկզբունքով՝ հնարավորություն տալով օգտատերերին էապես նվազեցնել SS ենթակառուցվածքի կապիտալ և սպասարկման ծախսերը, մասնավորապես՝ թանկարժեք սերվերային սարքավորումների ներդրման և դրանց անխափան աշխատանքը կարգավորող միջավայրի ապահովման ծախսերը [3]:

Ամպային տեխնոլոգիաները ապահովում են տվյալների հասանելիությունը և մատչելիությունը, օգտատիրոջը տրամադրելով առցանց ծառայություններ ինչպես ամպային միջավայրում: Ամպային տեխնոլոգիաների առավելություններից են նաև՝ անհրաժեշտ ռեսուրսների միջոցով ծրագրային ծառայությունների պատշաճ մատուցումը, ինչպես նաև ամպային համակարգի հնարավորությունների ընդլայնումը օգտվողների աճին զուգընթաց [4]:

Վիրտուալացման տեխնոլոգիաների կիրառման շնորհիվ հնարավորություն է ընձեռվում տնտեսել օգտագործվող ռեսուրսները՝ ընդսմին խուսափելով լրացուցիչ սերվերային տեխնիկայի ներդրման ծախսերից՝ կիրառելով «վճարել միայն օգտագործման ժամանակ» սկզբունքը: Հարկ է նշել, որ ամպային համակարգերին առանձնահատուկ են

նաև ճկունությունը, չափազանց մեծ հաշվողական ռեսուրսներն ու բարձր հուսալիությունը [3]:

Ամպային միջավայրը կարելի է դիտարկել որպես մասշտաբային ենթակառուցվածք, որը մատակարարում է որոշակի ամպային հաշվողական ծառայություններ:

Ամպն ինքնին բաղկացած է փոխկապակցված և դինամիկորեն տրամադրվող հաշվողական ռեսուրսներից, որի հիմնական տեխնոլոգիաներից են՝ ճկուն և մասշտաբային հաշվողական հարթակների վիրտուալացումը, վեբ ծառայությունները և ամպային ծառայությունների վրա հիմնված կառուցվածքները [5]:

Արդի ամպային տեխնոլոգիաների կիրառումը գործարկվող ծրագրերը դարձնում են ավելի շարժունակ և համագործակցային՝ սպառողական ծրագրերին համանման, ինչպիսիք են՝ մասնավորապես, սոցիալական ցանցերը:

Ամպային տեխնոլոգիաներն ըստ կիրառման մոդելների դասակարգվում են հետևյալ կերպ [6].

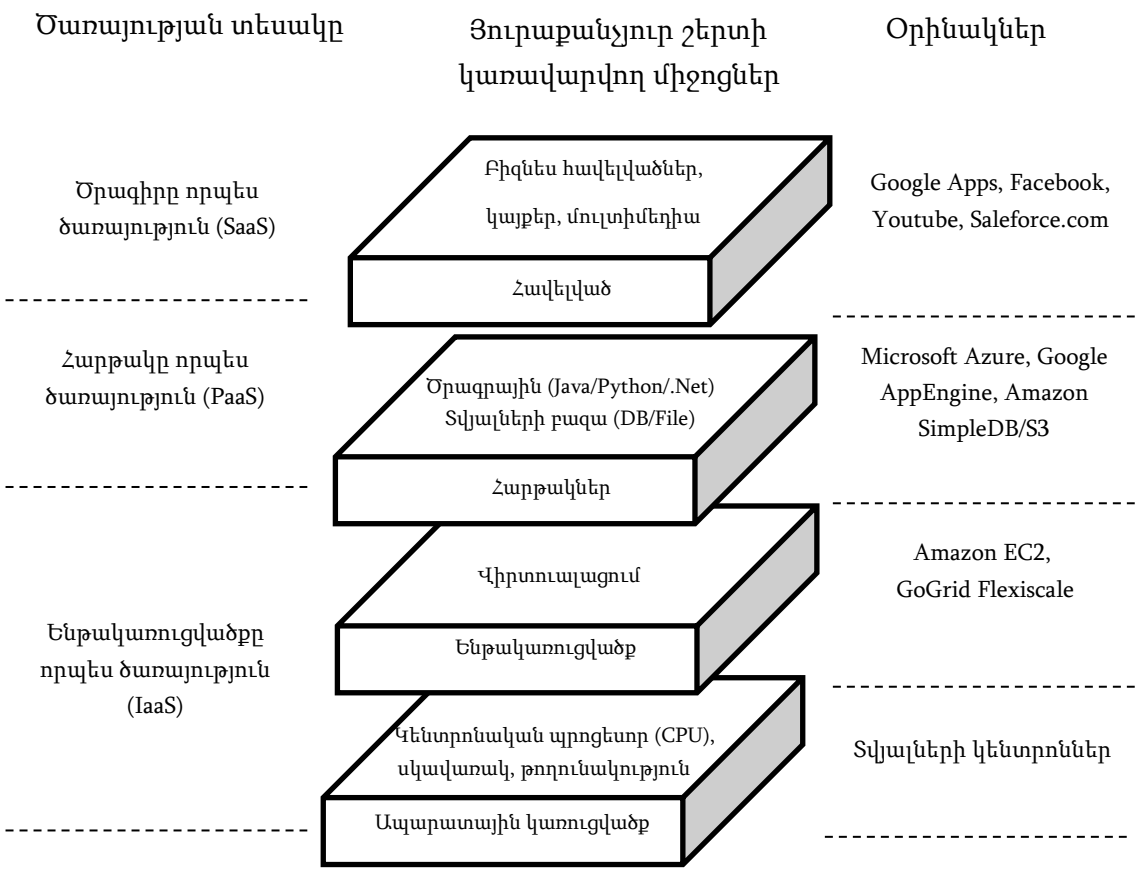
- Հանրային,
- Մասնավոր,
- Համայնքային,
- Հիբրիդային,

Հանրային ամպը, որպես կանոն, տրամադրում է ծառայություններ, որոնք հասանելի են բոլորին: Նրա հիմնական առավելություններն են՝ ծրագրային ապահովման համեմատաբար դյուրին տեղադրումը և ընդլայնման հնարավորությունը:

Մասնավոր ամպն արտոնագրված ցանց կամ տվյալների կենտրոն է, որը ծառայություններ է տրամադրում սահմանափակ թվով մարդկանց:

Համայնքային ամպը կառուցվում մի շարք կազմակերպությունների ընդհանուր պահանջների հիման վրա և ապահովում է այդ կազմակերպությունների կողմից համատեղ օգտագործման ենթակառուցվածքների հասանելիությունը:

Հիբրիդային ամպի միջավայրը բաղկացած է բազմաթիվ ներքին և/կամ արտաքին մատակարարներից և կիրառական է շատ կազմակերպությունների համար: Ամպային հիբրիդ պահոցները հաճախ օգտակար են տվյալների արխիվացման և կրկնօրինակման գործառույթների համար, որոնք իրենց հերթին հնարավորություն են ընձեռում տեղայնացված տվյալները փոխակերպել ամպային պահուստի:



Նկ. 1. Ամպային ծառայությունների հիմնական մակարդակները

Ամպային տեխնոլոգիաների ծառայությունները կարելի է տարակարգել հետևյալ երեք հիմնական մակարդակի (նկ.1) [7]:

- ամպային ծրագրային ապահովումը որպես ծառայություն, երբ ծառայության մատակարարը տեղադրում է ծրագրային միջավայրը ամպային տիրույթում՝ ապահովելով ծառայությունների հասանելիությունը ցանկացած հանգույցից:
- ամպային հարթակը (պլատֆորմը) որպես ծառայություն, որը նպաստում է ծրագրերի կիրարկմանը՝ առանց դրանց տեխնիկական միջոցների ձեռք բերման և կառավարման ունակությանը: Դրա փոխարեն, ամպային հարթակը հնարավոր է կառավարել ինտերֆեյսի, վեբ վահանակի կամ այլ ծրագրային միջոցներով:
- ամպային ենթակառուցվածքը որպես ծառայություն, որը տրամադրում է համակարգչային ենթակառուցվածքը (սովորաբար վիրտուալացման միջավայրը) որպես ծառայություն:

Օգտատիրոջ տեսանկյունից ամպային տեխնոլոգիաներն ընկալվում են որպես համակարգ, որի ներքին գործառույթների կազմակերպումը և իրականացումը օգտատիրոջ համար անհասանելի է: Որպես հետևանք, օգտատիրոջ կողմից առանձին վերցրած ամպային ենթակառուցվածքի փոփոխությունը և հարմարեցումը, ինչպես նաև անցումը նոր ծառայություններին՝ դժվար և երկարատև գործընթաց է: Չնայած վերոհիշյալ բարդությանը, ամպային տեխնոլոգիաների պահանջարկը բավականին բարձր է և շարունակում է աճել [8]:

Ամպային ծառայությունների մատուցման ընթացքում, երբ անհրաժեշտություն է առաջանում միաժամանակ սպասարկել բազմաթիվ օգտատերերի, բարդանում է տվյալների ու հաշվումների անվտանգության խնդիրների ապահովումը, և անվտանգության

լրացուցիչ միջոցների և դրանց կառավարման մեխանիզմների մշակման անհրաժեշտություն է առաջանում:

Ամպային ծառայությունները, որպես կանոն, ենթադրում են օգտագործողների վավերացման և դրանց տվյալների գաղտնիության մեխանիզմների ներգրավում: Մասնավորապես, գաղտնիությունն ապահովվում է ցանցային մակարդակում՝ պաշտպանելով հաշվողական ռեսուրսները գրոհներից:

Ամպային տեխնոլոգիաներում ներգրավված անվտանգության մեխանիզմները բաժանվում են.

- արտաքին անվտանգության ապահովում,
- ներքին անվտանգության ապահովում:

Թվարկած անվտանգության մեխանիզմներից առաջինը վերաբերում է ամպային տեխնոլոգիաների մեջ անվտանգ մուտքի ապահովմանը, իսկ 2-րդը՝ տարատեսակ վիրտուալ ծառայությունների առանձնացմանն ու պաշտպանությանը:

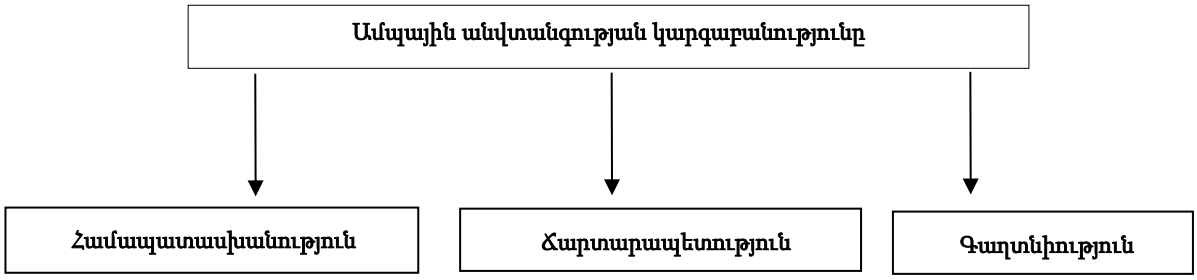
Ամպային ծառայությունների մեծամասնության մեջ կարելի է մուտք գործել Web դիտարկիչի միջոցով, որն օգտագործում է ստանդարտ HTTP (Hyper Text Transfer Protocol) արձանագրությունը: Ամպի և օգտատիրոջ միջև անվտանգ նույնականացման և գաղտնագրման համար օգտագործվում է SSL/TLS (Secure Socket Layer/Transport Layer Security) [9]:

Անվտանգության ապահովման մյուս մոտեցումները, որոնք օգտագործվում են տվյալների վավերացման համար, ներառում են բաց բանալինային ենթակառուցվածքը և X.509 SSL վկայագրերը [10]:

Ներկայումս ամպային տեխնոլոգիաների բնագավառում գոյություն ունեցող համապիտանի սահմանումները և ստանդարտները կիրառելի չեն ամբողջ ամպային միջավայրի համար, այլապես նման ստանդարտները հնարավորություն կընձեռեն դասակարգել ամպային ծառայություններն ըստ կարգաբանության (տաքսոնոմետրիա)՝ նպատակ ունենալով ապահովել տրամադրվող ծառայությունների անվտանգությունը: Նշված հանգամանքն անհրաժեշտություն է առաջացնում բացահայտել գոյություն ունեցող ամպային համակարգերի ուժեղ և թույլ կողմերը՝ ամպային հաշվողական ոլորտում արդի խնդիրների հետազոտման նպատակով:

1.2. Ամպային տեխնոլոգիաների սպառնալիքների և առաջարկված լուծումների վերլուծություն

Անվտանգության տեխնոլոգիաների բնագավառում տվյալների և ամպային ծառայությունների անվտանգության ապահովման նպատակով նախ անհրաժեշտություն է առաջանում դասակարգել ոլորտում առկա ռիսկերն ըստ տարակարգերի՝ նպատակ ունենալով ուրվագծել ամպային ծառայությունների անվտանգության հստակ պատկերը [11]: Առաջարկված դասակարգումը ներկայացված է նկ.2-ում:



Նկ. 2. Ամպային համակարգերի անվտանգության կարգաբանությունը

Ճարտարապետությունը ներկայացնում է ամպային տեխնիկական միջավայրի կառուցվածքը և դրա տարբեր հանգույցների փոխհարաբերությունները: ճարտարապետությունը վերաբերում է ցանցային անվտանգության, ինտերֆեյսերի և վիրտուալացման խնդիրներին, որոնք ներառում են ամպային ենթակառուցվածքը տնօրինողների և օգտատերերի ինտերֆեյսներ՝ ամպային միջավայր մուտք գործելու համար [12]:

Ճարտարապետությունը հնարավորություն է տալիս նաև հստակ տարանջատել պատասխանատվություններն ամպային ծառայությունների մատակարարների և հաճախորդների միջև: Կախված առաջարկված ծառայության տեսակից՝ կատարվում է ամպային ամբողջ համակարգի համապարփակ վերլուծություն՝ նպատակ ունենալով հստակ ձևակերպել օգտագործվող անվտանգության մեխանիզմները՝ նախքան

ծառայության պայմանագրի կնքումը: Ընդսմին նախասահմանվում են ծառայություն մատուցող կողմի գործառույթները՝ կոնկրետ պահանջների ապահովման համար:

Համապատասխանությունը ներկայացնում է տրամադրվող ամպային ծառայություններ մատուցողի կողմից կրվող պատասխանատվությունը՝ ներառյալ ծառայության ընթացքում առաջացած հնարավոր խափանումները:

Գաղտնիությունը ապահովվում է ամպային տեխնոլոգիաների կիրառման ողջ գործընթացներում՝ ներառյալ այդ տվյալների ստեղծումը, պահեստավորումը, օգտագործումը և հեռացումը, միաժամանակ ուղղորդվելով առկա իրավակարգերով և կանոնակարգերով:

Ամպային միջավայրում ռեսուրսների վիրտուալացման հիմնահարցերը պահանջում են հետազոտություններ այնպիսի խնդիրների լուծման համար, ինչպիսիք են՝

- ռեսուրսների մեկուսացումը,
- տվյալների արտահոսքը,
- վիրտուալ մեքենաների դեմ հարձակումները:

Թվարկված հիմնահարցերի հետազոտման նպատակով անհրաժեշտություն է առաջանում ուսումնասիրել ամպային տեխնոլոգիաների մյուս կարևոր դրույթները ևս, այդ թվում [13] ՝

1. **Վերահսկողությունը և ռիսկերի կառավարումը,** որը ներկայացնում է իրերի ինտերնետում ռիսկերի գնահատման մեթոդաբանությունը, կոնֆիդենցիալ տվյալների պաշտպանությունը և ոլորտի իրավական խնդիրները:
2. **Ինֆորմացիայի կառավարումը և տվյալների** անվտանգությունը**,** որն ապահովում է տվյալների նույնականացումը և պահպանված տվյալների վերահսկումը՝ ներառյալ

տվյալների ֆիզիկական վերահսկողությունը, ռիսկերը և դրանց հասցրած հնարավոր վնասների գնահատումը:

3. **Տեղափոխելիությունը**, որը վերաբերում է մատակարարի կամ ծառայությունների տեղափոխման հիմնահարցերին կամ ամպային ծառայությունները լոկալ տիրույթ հետ բերելու հնարավորությանը:
4. **Տեղեկատվական անվտանգությունը**, որը վերաբերում է գործընթացների շարունակականության անվտանգ իրականացմանը՝ այնպիսի լուծումների առաջարկում, որոնք ապահովում են անվտանգության պահանջները:
5. **Տվյալների մշակման կենտրոնի գործողությունները**, որը ներկայացնում է տվյալների մշակման կենտրոնների կառուցվածքի և գործողությունների վերլուծությունը և համակարգի վթարակայունության մակարդակը:
6. **Խափանումներին արձագանքը**, որն ապահովում է խափանման մասին առկա տեղեկատվության ծանուցումը հասցեատերերին, խափանումների շտկումը և ամպային ենթակառուցվածքի ներքին քաղաքականությունը:
7. **Ծրագրային հավելվածների անվտանգությունը,** որն ուղղված է անվտանգության հնարավոր խնդիրների բացահայտմանը՝ կապված ամպային միջավայրում որոշակի խնդրի լուծման ներդրման հետ:
8. **Գաղտնագրումը և բանալու կառավարումը,** որը կապված է գաղտնագրման և այլ մեխանիզմների վրա մասշտաբայնության ազդեցությամբ, որոնք օգտագործվում են ռեսուրսների և տվյալների պաշտպանության համար:
9. **Նույնականացման և մուտքի կառավարումը,** որպես վավերացման իրականացում ամպային լուծումների համար՝ անվտանգության և հասանելիության պահպանման նպատակով:
10. **Ռեսուրսների վիրտուալացումը,** որը կարգավորում է ռիսկերը՝ պայմանավորված բազմաօգտատիրության, մեկուսացման և վիրտուալ մեքենաների համատեղ տեղադրվածության տեսանկյունից:

11. **Անվտանգությունը որպես ծառայություն,** որն ենթադրում է երրորդ կողմի անվտանգության մեխանիզմների կիրառումը, ինչպես նաև անվտանգության պատասխանատվության փոխանցումը երրորդ կողմին:

Վերլուծությունն ամպային միջավայրում ցույց է տալիս, որ բազմաթիվ խնդիրներ դեռևս լուծված չեն և պահանջում են հետագա հետազոտություններ: Դրանք հիմնականում այն խնդիրներն են, որոնք վերաբերում են տվյալների գաղտնիության ապահովմանը[14]:

1.3. Իրերի ինտերնետի կիրառման առանձնահատկությունները

Ամպային տեխնոլոգիաների զարգացող ոլորտներից է իրերի ինտերնետը: Ֆիզիկական սարքերը, սենսորները, մեքենաները և այլ իրերը, որոնք կապված են համացանցին և կատարում են տվյալների փոխանակում կարելի է անվանել ինտերնետ իրեր: Իրերի ինտերնետը միջավայր է, որը բաղկացած է ինտերնետ իրերից: Իրերի ինտերնետը հիմնականում օգտագործում է անլար հաղորդակցման միջոցներ, մասնավորապես՝ wifi, bluethout, 3G և 4G ցանցերը՝ տվյալների փոխանցման և ծանուցումների ստացման համար: Իրերի ինտերնետն օգտագործում է հղումների (հասցեների) հատուկ գլոբալ սխեմա՝ հեռահաղորդակցման և պահանջված ծառայության տրամադրման նպատակով: Անլար սենսորները և ռադիոհաճախականային նույնականացման (RFID) տեխնոլոգիաները, համապատասխանաբար, իրերի ինտերնետի կարևոր բաղկացուցիչ մասերն են:

Ինտերնետ իրերին հատուկ է ցածր արտադրողականությունը և ֆիքսված ծրագրային ապահովումը, քանի որ, ինտերնետ իրերը հիմնականում ստեղծված են կատարելու հստակ գործառույթներ: Թվարկված հատկանիշները արգելք են տեղեկատվական անվտանգության գոյություն ունեցող լուծումների կիրառման համար:

RFID-ն օգտագործում է էլեկտրամագնիսական դաշտ՝ ավտոմատ նույնականացման և օբյեկտներին կցված թեգերի հետ հաղորդակցման համար: Ընդհանուր առմամբ RFID-ը կարող է ընդունել և պատասխանել ազդանշաններին՝ անկախ նրանից, թե ով է ուղարկողը: Հարկ է նշել, որ այստեղ առկա է առաջնային խնդիր՝ կապված չվավերացված մուտքի և թեգերի տվյալների մոդիֆիկացիաների հետ [15]:

Մյուս կողմից՝ RFID համակարգերը հաճախ հաղորդակցվում են կառավարող համակարգի հետ՝ օգտագործելով ստանդարտ TCP/IP փաթեթը, ընդհանուր GNU/Linux,

Windows սերվերները, VoIP (Voice over IP) և ցանցային սենսորները, որոնք իրենց հերթին մի շարք անվտանգության խնդիրներ են առաջացնում կարգավորողների և օգտատերերի համար [16]: Մասնավորապես՝ այդ համակարգերը մշակվում են առևտրային (կոմերցիոն) սարքավորումների ներգրավմամբ, որոնք հետապնդում են զուտ առևտրային նպատակներ և կիրառումից առաջ չեն ստուգվում ըստ անվտանգության պահանջների, մինչդեռ համակարգերը պահանջում են խիստ վերահսկողություն՝ ցանցային ներխուժումների բացահայտման նպատակով:

Իրերի ինտերնետի գործառույթը տվյալների անվտանգ հաղորդումն է համապատասխան հասցեատիրոջը՝ նշված ժամանակում և ճիշտ ձևաչափով: Այդ նպատակի իրականացման համար անհրաժեշտ է, որ առաջարկվող լուծումները ապահովեն համակարգի բաղադրիչների փոխհամագործակցությունը, և ինտեգրվեն մյուս բոլոր սարքերի հետ[17]:

Վերոհիշյալ խնդիրների լուծման մեթոդները և գործիքամիջոցները ծախսատար են և պահանջում են իրերի ինտերնետի տեղեկատվական անվտանգության նոր մոտեցումների հետազոտում և մշակում: Ինտերնետ իրերը ունեն տվյալների մշակման և պահպանման հսկայական հնարավորություններ, որոնց միջև կապն հիմնականում իրականացվում է համացանցի միջով, եթե նույնիսկ դրանք գտնվում են միմյանցից փոքր հեռավորության վրա:

Իրերի ինտերնետում տեղեկատվական անվտանգության գոյություն ունեցող լուծումները կատարելագործման կարիք ունեն այն հիմնավորմամբ, որ առ այսօր չկա որևէ մի հարթակ, որը միավորի բոլոր սարքերը և երաշխավորի տարբեր արտադրողների կողմից առաջարկվող ամպային ծառայությունների համատեղելիությունը և փոխգործունեությունը[18]:

Իրերի ինտերնետ միջավայրի ապակենտրոնացման մոտեցմամբ մասնակիորեն լուծվում են վերը նշված հարցերից շատերը: Մասնավորապես՝ ստանդարտացված «հավասարը հավասարին» կապի մոդելի ներմուծումը, որը նախատեսված է սարքերի միջև բարձր թվով գործառույթների իրականացման համար, զգալիորեն նվազեցնում է տվյալների կենտրոնացված խոշոր կենտրոնների տեղադրման և տեխնիկական սպասարկման ծախսերը, ինչպես նաև ապահովում է հաշվարկների բաշխումն ինտերնետ իրերի միջև: Այնուամենայնիվ, «հավասարը հավասարին» կապի հաստատումը ներկայացնում է խնդիրների համախումբ, որոնցից առաջնայինը՝ անվտանգության խնդիրն է[18]:

Ինտերնետ իրերի ցանցերը խոցելի են տարբեր հարձակումների նկատմամբ՝ ելնելով անլար կապերի հեռարձակման և տարբեր միջավայրերում ավտոմատացված աշխատանքի բնույթից: Անհրաժեշտ է գտնել այնպիսի լուծում, որը ապահովում է տվյալների ու հաշվարկների գաղտնիությունը և անվտանգությունն իրերի ինտերնետում և առաջարկում է վավերացման և փոխհամաձայնություն որոշակի մեթոդ՝ բոլոր գործառույթների համար, որպեսզի հնարավոր լինի կանխել տվյալների կեղծումը և կողոպտումը: Նշված պահանջների հետ մեկտեղ, առաջարկվող լուծումը պետք է հնարավոր լինի կիրառել տարատեսակ սարքերից բաղկացած իրերի ինտերնետ միջավայրում՝ ըստ այդմ ապահովելով համակարգի անխափան աշխատանքը իրերի թվաքանակի փոփոխության պարագայում:

Իրերի ինտերնետ միջավայրում տեղեկատվական անվտանգության խնդիրի արդյունավետ լուծման համար անհրաժեշտ է ինտերնետ իրերը դասակարգել հետևյալ երկու տարակարգի՝

- հաշվողական բավարար ռեսուրս ունեցող տարատեսակ ինտերնետ իրեր, որոնք ունակ են իրականացնել գաղտնագրային գործառույթներ,
- հաշվողական լրացուցիչ հնարավորություն չունեցող պարզ ինտերնետ իրեր, որոնց տեխնիկական ցածր հնարավորությունների պատճառով անհնար է ներդնել գաղտնագրման որևէ համակարգ:

Ինտերնետ իրերը, որոնք ունակ են կատարել գաղտնագրային գործառույթներ, հիմնականում օգտագործում են համաչափ գաղտնագրային ալգորիթմներ՝ անվտանգ տվյալների փոխանակում իրականացնելու նպատակաով, որտեղ առաջանում է բանալիների կառավարման խնդիր: Ներկայումս այս խնդիրը լուծվում է բանալիների նախօրոք ներդրմամբ, որն ազդում է ցանցի մասշտաբայնության և օգտագործվող հիշողության վրա: Հարկ է նշել, որ գոյություն ունեցող բանալիների բաշխման սխեմաները հիմնականում իրացվում են միատեսակ հաշվողական համակարգերի միջավայրում, որտեղ հանգույցների միջև պահանջվող ռեսուրսների հավասարաչափ բաշխման խնդիրը լուծելի է: Մինչդեռ իրերի ինտերնետում տարատեսակ սարքավորումների և հանգույցների առկայությունը վերոնշյալ խնդիրը լուծելու հնարավորություն չի տալիս:

Գոյություն ունեցող լուծումները հիմնված են ինտերնետ իրերում բանալիների նախօրոք ներդրման հետ, այսուհետ նախաբաշխման: Ինտերնետ իրերի միջև տեղակայումից առաջ բանալիները բաշխվում են՝ հարևան ինտերնետ իրերի հետ անվտանգ կապ ստեղծելու նպատակով: Տվյալ մեթոդի թերություններն են՝ մասշտաբայնության ապահովումը և տարատեսակ ինտերնետ իրերի միջավայրում իրացման սահմանափակումը: Բանալիների նախաբաշխման հիմնական տարատեսակներից է ընդհանուր բանալի նախաբաշխումը կամ եզակի բանալիների բաշխումը զույգ հանգույցների համար [19]:

Ընդհանուր բաշխված բանալու թերությունը կապված է ցանկացած հանգույցում տեղի ունեցած բանալու արտահոսքի հետ, որը խոցելի է դարձնում այդ բանալիով գաղտնագրված բոլոր այլ հանգույցների տվյալները: Հարկավոր է նշել, որ տվյալ մեթոդը ապահովում է հիշողության նվազագույն օգտագործում և բարձր մասշտաբայնություն:

Զույգ հանգույցների միջև եզակի բանալիների բաշխման հիմնական թերությունները կապված են հիշողության գերբեռման և կառավարման բարդության հետ, որը հանդիսանում է արգելք լայնամասշտաբ ցանցի ստեղծման համար: Առաջարկվող համակարգերը, որոնցից են բաց բանալիների ենթակառուցվածքը և բանալիների բաշխման կենտրոնը, ունեն բավականին սահմանափակ կիրառում՝ ինտերնետ իրերում մարտկոցի սնուցման, պրոցեսորի, հիշողության և այլ ռեսուրսների սահմանափակության պատճառով: Սակայն հարկ է նշել, որ տվյալ մեթոդը ունի բարձր հուսալիության հնարավորություն, քանի որ ցանկացած հանգույցում բանալու արտահոսքը չի ազդում այլ հանգույցների վրա:

Հաշվի առնելով իրերի ինտերնետ միջավայրի առանձնահատկությունները անհրաժեշտ մեթոդը պետք է ապահովի.

- բարձր մասշտաբայնություն,
- բարձր կապակցվածություն,
- տարատեսակ իրերի ինտերնետ միջավայրում իրացում,
- հիշողության ցածր օգտագործում:

Գոյություն ունեցող մեթոդներից ոչ մեկը չի բավարարում այս բոլոր պահանջներին: Մասշտաբայնության հիմնական խնդիր է նաև նոր սարքի ինտեգրումը, որը կարող է լինել անհնար որոշ իրացումներում, հատկապես երբ գաղտնագրումը կատարվում է բանալիների վաղօրոք ներդրված մատրիցների օգտագործմամբ:

Անհրաժեշտ է ստեղծել նոր մեթոդ, որը նախատեսված է լայնամասշտաբ իրերի ինտերնետ ցանցերում բանալիների կառավարման համար:

Կարևոր է նաև լրացուցիչ հաշվողական հնարավորություն չունեցող ինտերնետ իրերի խնդիրը, որոնց սահմանափակ ռեսուրսների պատճառով անհնար է ներդնել գաղտնագրման որևէ համակարգ: Այս դասի ինտերնետ իրերը կերելի է անվանել անվանել պարզ ինտերնետ իրեր: Պարզ ինտերնետ իրերը հիմնականում չեն փոխանցում դինամիկ տվյալներ և կառավարվում են հրամանների միջոցով ամպային միջավայրի կողմից: Այդ սարքերում տեխնիկական փոփոխություններ կատարելու փորձերը հանգեցնում են գնի բարձրացմանը, ուստի և անհրաժեշտություն է առաջանում ստեղծել այնպիսի լուծում, որն առանց որևէ տեխնիկական փոփոխության կատարմամբ կապահովի փոխանցվող տվյալների գաղտնիությունը :

1.3.1. Բանալիների կառավարման սխեմաները

Ինտերնետ իրերի ռեսուրսների սահմանափակության պատճառով անհնարին է դառնում ներկայումս կիրառվող բաց բանալու գաղտնահամակարգերի օգտագործումը՝ տեղեկատվական անվտանգության ապահովման նպատակով, մինչդեռ իրերի ինտերնետի համար բանալու կառավարման արդյունավետ սխեմաները անվտանգության ապահովման պարտադիր բաղադրիչներից են:

Գաղտնագրային բանալիների կառավարման ուղղությամբ կատարված հետազոտությունների ընթացքում դիտարկվել են ինչպես միատեսակ, այնպես էլ՝ տարատեսակ անլար սենսորային ցանցերը [20]:

Էշենաուերը և Գլիգորը առաջինն առաջարկել են նախաբաշխման վրա հիմնված բանալու կառավարման սխեման, որն պահանջում է զգալի հիշողություն՝ բանալիների պահպանման համար [21]:

Լյուն և մյուսները կատարելագործել են նախաբաշխման վրա հիմնված բանալու կառավարման սխեման՝ թույլ տալով կատարել բանալիների ստեղծումը տեղակայումից հետո, եթե ցանցի հանգույցներն գտնվում են միևնույն համակարգում [22]: Նրանց կողմից առաջարկվել է նաև ինքնակարգավորվող կառուցվածք՝ նախատեսված սենսորային ցանցերում գործարկվող բանալիների համար

Զհուն և մյուսները մշակել են բանալու կառավարման սխեմա՝ LEAP, որը ենթադրում է, որ ցանցն անվտանգ կլինի կարճ ժամանակահատվածում սենսորի տեղակայումից հետո, այնուհետև նախապես բեռնված գլոբալ բանալին կօգտագործվի բանալու կառավարման գործարկումը մեկնարկելու համար [23]:

Ընդունված է, որ բանալու նախաբաշխումը կատարվում է բացառապես հարևան ինտերնետ իրերի միջև ընդհանուր բանալիների հաստատման համար ցանցի տեղակայումից հետո: Այնուամենայնիվ, բանալու նախաբաշխման առաջարկված սխեմաներից ոչ մեկը միաժամանակ չի ապահովում բարձր կատարողականություն՝ ցանցի մասշտաբայնության առումով: Առկա են խնդիրներ, ինչպիսիք են՝ հարևան հանգույցների միջև բանալիների համատեղ օգտագործման հավանականությունը և կայունությունը գրոհների նկատմամբ:

Վերոհիշյալ խնդիրները մասամբ լուծվում են ինքնակարգավորվող կառուցվածքի կիրառմամբ, որը նախատեսված է լայնամասշտաբ սենսորային ցանցերում բանալիների օգտագործման համար: Այն սկզբունքորեն տարբերվում է բանալու նախաբաշխման մյուս սխեմաներից նրանով, որ այն չի պահանջում բանալու ստեղծման մասին նախնական տեղեկություն:

Ինքնակարգավորվող կառուցվածքը նախապես բեռնում է հանգույցների բանալիների կառավարման յուրաքանչյուր սխեման այնպիսի պարամետրերով, որոնք պետք է օգտագործվեն տեղակայումից հետո հարևան հանգույցների միջև բանալիների հաստատման համար:

Այնուամենայնիվ, գոյություն ունեցող սխեմաները նախագծված են միատեսակ սենսորային ցանցերի համար, որոնցում սենսորային հանգույցներն ունեն միատեսակ հնարավորություններ կամ աղբյուրներ:

Լոռեն և մյուսներն առաջարկել են տեղակայումից հետո բանալու ստեղծման արդյունավետ սխեմա՝ տարատեսակ սենսորային ցանցերի համար: Այդ սխեման օգտագործում սենսորային միջավայրում հաղորդման բարձր հնարավորությամբ մի քանի

հզոր հանգույցներ, տարատեսակ սենսորային ցանցերում բանալիների արդյունավետ գեներացման համար [24]:

Ազարդերախշը և Ռեյհանի-Մասոլեհը դիտարկել են անվտանգության խնդիրները տարատեսակ անլար սենսորային ցանցերում, ներկայացրել են անվտանգ քլասթերային սխեմա՝ դետերմինացված բանալիների կառավարման սխեմայի հետ միասին՝ հիմնված բաց բանալինային գաղտնագրման մեթոդի վրա [25]:

Առաջարկված անվտանգության մեխանիզմը երաշխավորում է, որ միևնույն քլասթերում գտնվող ցանկացած երկու սենսորային հանգույցներ կարող են գեներացնել բանալիներ հանգույցների զույգի համար՝ առանց ինֆորմացիան մյուս հանգույցներին հայտնելու: Սենսորային հանգույցներն իրենց էությամբ սահմանափակված են սարքի նախնական ոչ բավարար պարամետրերով, ինչպիսիք են՝ հիշողության հնարավորությունը և մարտկոցի ծառայության ժամկետը: Որպես կանոն, բանալու կառավարման որոշ ժամանակակից սխեմաներ համապատասխանում են անլար սենսորային ցանցերերին:

Հուանգը և մյուսները առաջարկել են բանալու կառավարման նոր մեթոդ, որն օգտագործում է բանալու կառավարման դինամիկ սխեմաներ՝ նախատեսված տարատեսակ սենսորային ցանցերի համար, որոնք բեռնում են h-ֆունկցիան բազային կայանի, քլասթերային գլխիկների և սենսորային հանգույցների մեջ [26]: Քլասթերային գլխիկները և սենսորային հանգույցները այնուհետև գեներացնում են իրենց սեփական բանալու շղթաները, որպեսզի հնարավոր լինի ապահովել բանալիների փոփոխության վավերացումը՝ նաև անվտանգության խափանման դեպքում: Քլասթերային գլխիկները և սենսորային հանգույցները ստեղծում են բանալիների զույգ՝ փոխանցման գաղտնիությունն ապահովելու համար:

Հաշվի առնելով այն հանգամանքը, որ ժամանակակից անլար սենսորային ցանցերը բաղկացած են տարատեսակ ինտերնետ իրերից, և հիմնվելով կատարված հետազոտության արդյունքների վրա կարելի է եզրահանգել. անվտանգության բանալու կառավարման նոր սխեմայի նախագծումը և կիրառումը իրերի ինտերնետում՝ իրերը իրերին (T2T) կապի համար սենսորներից և «խելացի» ներկառուցված սենսորային իրերից բաղկացած ցանցում, արդիական խնդիր է:

Ընդհանուր առմամբ, բանալու կառավարման համակարգի իրականացումը կախված է այն միջավայրից, որտեղ այն օգտագործվում է: Ինչպես նշվել է, միջավայրը կարող է լինել միատեսակ և տարատեսակ, իսկ տվյալների պաշտպանությունն իրերի ինտերնետում անհրաժեշտ է ապահովել կիրառման բոլոր փուլերում [27]:

1.3.2 Գործառույթների ամբողջականության ապահովումը իրերի ինտերնետ միջավայրում

Ինտերնետ իրերին հատուկ է նաև ֆիզիկական ազդեցությունը արտաքին միջավայրի վրա, որի պատճառով կարևորվում է ինտերնետ իրերի կողմից իրականացվող գործառույթների ամբողջականության ապահովումը: Այս տեսակի սարքերը կարող են օգտագործվել այլ ինտերնետ իրերի հետ, գործառույթների շղթա կազմելու նպատակով: Տվյալ դեպքում կարևոր է կատարվող գործառույթների ամբողջականության ապահովման խնդիրը, որը ենթադրում է գործառույթների հերթականության պահպանումը և խափանումների դեպքում կատարված գործառույթների չեղարկումը: Գործառույթների ամբողջականության ապահովման արգելք է հանդիսանում նաև ինտերնետ իրերի խոցելիությունը ցանցային գրոհների նկատմամբ, մասնավորապես՝ ներխուժումների և DDoS գրոհների նկատմամբ: Բաշխված ինտերնետ իրերի միջավայրում գործառույթների ամբողջականության ապահովման համար կարևոր են՝ գործառույթների կառավարումը, տվյալների գաղտնագրումը, սխալների հայտնաբերումը և գործառույթների չեղարկումը [28]:

Ներխուժման հայտնաբերման համակարգերը մշակված են այնպես, որ դրանք կարող են արդյունավետորեն հայտնաբերել և նույնականացնել սպառնալիքները: Ներխուժումները հայտնաբերվում են ըստ հետևյալ մոտեցումների.

- Ներխուժման սխալ գործածություն, որի հիմքում ընկած է ներխուժող կողմի ցանցային թրաֆիկի հետազոտումը և համեմատումը այլ սպառնալիքների հետ: Սխալ օգտագործման հայտնաբերման համակարգի առավելություններից մեկը համարվում է բոլոր հայտնի սպառնալիքների հայտնաբերման կարողությունը:

- Ներխուժման անկանոնություն, որը վարքագծի վրա հիմնված ներխուժումների հայտնաբերման համակարգ է: Այն նկատում է համակարգում նորմալ ընթացող գործընթացներում տեղի ունեցող փոփոխությունները՝ համակարգի պրոֆիլի ստեղծման միջոցով, որը վերահսկվում է:

Բազմաթիվ համակարգեր միավորում են ներխուժման այս երկու մոտեցումները դրանց փոխլրացնող բնույթի շնորհիվ: Կեղծ ներխուժումների պատճառով սխալ օգտագործման մոդելների վրա հիմնված համակարգերը սովորաբար կիրառվում են կոմերցիոն նպատակներով, մինչդեռ անկանոն ներխուժումը հայտնաբերվում է հետազոտական համակարգերում: Հաճախ ներխուժմից առաջ կատարվում է DDoS գրոհներ [29]:

Գոյություն ունեցող ներխուժումների հայտնաբերման մեթոդները ստեղծված են համակարգչային ցանցերում ներխուժումների հայտնաբերման համար, որոնք չեն կարող լինել արդյունավետ տարատեսակ իրերի ինտերնետ միջավայրում:

1. 4 Խնդրի դրվածքը

Տեղեկատվական անվտանգության անհրաժեշտ մակարդակը բավարարող իրերի ինտերնետի կառուցման նպատակով՝ միջավայրում գործառույթների ամբողջականության ապահովման, նույնականացման և գաղտնի բանալիների բաշխման նորագույն եղանակների, ինչպես նաև առավել նպատակահարմար գաղտնագրային ալգորիթմների հետազոտումը և մշակումը արդիական խնդիր է: Ընդ որում, առաջարկվող լուծումները՝ ապահովելով կայունություն, միևնույն ժամանակ պետք է կիրարկեն ոչ ծախսատար հաշվողական ռեսուրսներ՝ համակարգի արդյունավետության ապահովման նպատակով:

Մասնավորապես՝ գաղտնիքի բաշխման գոյություն ունեցող սխեմաների վերլուծությունն արդիական խնդիր է այն հիմնավորմամբ, որ հետազոտության արդյունքում մշակվել են իրերի ինտերնետ միջավայրին առավել համապատասխան սխեմաներ, որոնք հնարավորություն են ընձեռում ցանցային տարատեսակ սարքավորումներում տեղեկատվության գաղտնագրում/վերծանում իրականացնել՝ միևնույն ժամանակ ազատելով հաշվողական ռեսուրսները գերբեռնումից կամ տվյալ ռեսուրսների բացակայության դեպքում գործառույթը փոխանցել երրորդ կողմին:

Մյուս արդիական խնդրի՝ իրերի ինտերնետ միջավայրում գործառույթների ամբողջականության ապահովումը և ինտերնետ իրերին բնորոշ ցանցային գրոհները բացահայտող մեխանիզմների հետազոտումը նախադրյալներ կստեղծի նշված գրոհների կանխարգելման մեթոդի մշակման համար:

Այսպիսով, իրերի ինտերնետում տեղեկատվական անվտանգության ապահովման նպատակով անհրաժեշտություն է առաջանում լուծել հետևյալ խնդիրները.

- Մշակել գաղտնագրային բանալիների կառավարման մեթոդ, որն ապահովում է միջավայրին միացվող ինտերնետ իրերի ինքնակարգավորումը և դրանց միջև տվյալների անվտանգ փոխանակումը:
- Մշակել անվտանգ հաղորդակցման մեթոդ ամպային միջավայրում պարզ ինտերնետ իրերի համար, որոնք սահմանափակ ռեսուրսների պատճառով գաղտնագրային ընթացակարգեր չեն ապահովում:
- Մշակել ամպային բաշխված ցանցերում գործառույթների ամբողջականությունն ապահովող մեթոդ, որը նաև կիրականացնի ամպային միջավայրի վրա հատուկ գրոհների բացահայտումը և դրանց կանխարգելումը:

1. 5 Գլուխ 1-ի եզրակացություն

- Հետազոտվել են ամպային տեխնոլոգիաների տեղեկատվական անվտանգության հիմնախնդիրները:
- Հիմնավորվել է ամպային տեխնոլոգիաներում, մասնավորապես տարատեսակ իրերի ինտերնետում, առկա կարևորագույն խնդիրը, այն է՝ տեղեկատվական անվտանգությունը:
- Ցույց է տրվել, որ ամպային ենթակառուցվածքներում խնդրահարույց մասը ինտերնետ տարատեսակ իրերն են, որոնց հաղորդակցման պաշտպանվածությունը չլուծված խնդիր է:
- Հետազոտվել են բանալիների բաշխման գոյություն ունեցող սխեմաները՝ իրերի ինտերնետ միջավայրում դրանց կիրառման նպատակահարմարության տեսանկյունից և բացահայտվել են հասանելի սխեմաների թերությունները:
- Ուսումնասիրվել են իրերի ինտերնետ միջավայրում գործառույթների ամբողջականության ապահովման խնդիրը և ներխուժումների կանխարգելման գոյություն ունեցող մոդելները՝ իրերի ինտերնետ միջավայրում դրանց կիրառման նպատակահարմարության տեսանկյունից:
- Եզրահանգվել է, որ իրերի ինտերնետ միջավայրի տեղեկատվական անվտանգության պահանջվող մակարդակի ապահովման համար անհրաժեշտ է մշակել նույնականացման և գաղտնի բանալիների բաշխման նոր մեթոդ, իսկ գործառույթների ամբողջականության ապահովման համար՝ առանձնահատուկ մոդել:

ԳԼՈՒԽ 2

ԻՐԵՐԻ ԻՆՏԵՐՆԵՏ ՄԻՋԱՎԱՅՐՈՒՄ ԲԱՆԱԼԻՆԵՐԻ ԲԱՇԽՄԱՆ ԵՎ ԿԱՌԱՎԱՐՄԱՆ ՍԽԵՄԱՅԻ ՄՇԱԿՈՒՄԸ

2.1. Իրերի ինտերնետ միջավայրում բանալու բաշխման մեթոդի մշակումը տարատեսակ իրերի միջև պաշտպանված կապի հաստատման նպատակով

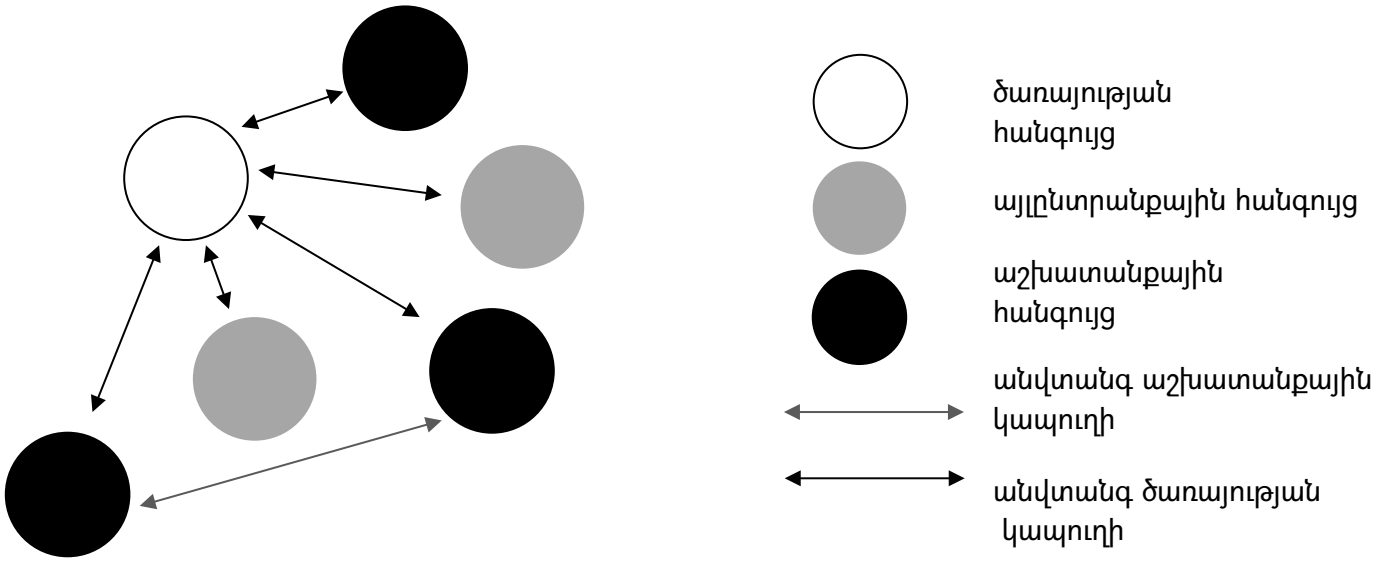
Իրերի ինտերնետ համակարգերն օգտագործում են գլոբալ հասցեավորման սխեմա՝ ցանցի հանգույցների միջև տվյալների անվտանգ փոխանակման և փոխադարձ հարցումների սպասարկման նպատակով: Տվյալների փոխանակումը, որպես կանոն, կատարվում է անլար կապուղիներով, որի հիմքում ընկած է գաղտնագրումը՝ բանալիների զույգի կիրառմամբ [30,31]: Բանալիների զույգի կիրառումը նշանակում է, որ ցանցում գտնվող ցանկացած երկու հանգույց օգտագործում են յուրօրինակ և անկախ բանալի:

Իրերի ինտերնետ միջավայրում հանգույցների մեծ քանակությունը և դրանց փոփոխականությունն անհամատեղելի են դարձնում գաղտնագրային բանալիների ենթակառուցվածքի ներդրումը, քանի որ կարող են առաջացնել հիշողության գերբեռնում [32,33]: Որպես հետևանք՝ անհրաժեշտություն է առաջանում հանգույցները խմբավորել և ստեղծել բաշխված բանալիների անհատական տիրույթ, որտեղ յուրաքանչյուր խումբ կունենա բանալիների անհատական տիրույթ: Քանի որ, ինչպես նշվել է, իրերի ինտերնետ միջավայրում հանգույցների քանակը կայուն չէ և ենթակա է հաճախակի փոփոխման, ուստի անխուսափելիորեն անհրաժեշտություն է առաջանում ապահովել բանալիների բաշխման սխեմայի անխափան աշխատանքը նոր հանգույցների ավելանալու դեպքում՝ ընդ որում ապահովելով հանգույցի պարզ ինտեգրումն արդեն ձևավորված խմբին:

Իրերի ինտերնետ միջավայրում հանգույցներն ունակ են սպասարկել մինչև որոշակի N թվով հարցումներ: Քանի որ նշված հանգույցները տարատեսակ են, ուստի ակնհայտ է, որ տվյալ N թիվը տարբեր է ամեն հանգույցի համար: Քանի որ բանալիների բաշխման սխեմայի օգտագործումը ենթադրում է լրացուցիչ հարցումների անհրաժեշտություն, որը կարող է բերել հարցումների սպասարկման խափանմանը, անհրաժեշտ է հնարավորինս խուսափել փոքր N թիվ ունեցող հանգույցները հարցումներով ծանրաբեռնելուց:

Այսպիսով, հաշվի առնելով հանգույցների տեղայնացված հիշողության գերբեռնման և վերոհիշյալ N պարամետրի սահմանափակ լինելու հետ կապված խնդիրները, անհրաժեշտություն է առաջանում մշակել բանալիների բաշխման և կառավարման նոր սխեմա, որով կարելի կլինի անվտանգ կերպով կառավարել ինտերնետի տարատեսակ իրերը:

Վերոհիշյալ սխեման ներգրավում է երեք տեսակի հանգույցներ, ինչպես բերված է նկար 3-ում:



Նկ. 3. Ցանցային հանգույցները և դրանց կապերը

Հանգույցների նման դասակարգումը թույլ է տալիս հնարավորինս խուսափել հանգույցների հիշողության գերբեռնումից և հարցումների սպասարկման ցածր արտադրողականությունից՝ ծառայության հանգույց ընտրելով այն հանգույցը, որը տվյալ պահին տեխնիկապես (կախված իրի տեխնիկական պարամետրերից) ավելի նպատակահարմար է:

Իրերի ինտերնետում բանալիների բաշխումն ու կառավարումը տեղի է ունենում հետևյալ հաջորդականությամբ.

Ալգորիթմ 1.

Քայլ 1. յուրաքանչյուր հանգույց ընտրվում է, որպես՝

- ծառայության հանգույց,
- այլընտրանքային հանգույց,
- աշխատանքային հանգույց:

Քայլ 2. ծառայության հանգույցի կողմից ստեղծվում է բանալիների ենթակառուցվածք՝ համապատասխան աշխատանքային հանգույցների համար:

Քայլ 3. հաստատվում է անվտանգ կապուղի՝ աշխատանքային հանգույցի և դրա հետ կապված ծառայության հանգույցի միջև, որի միջոցով աշխատանքային հանգույցները բանալու վերաբերյալ ամբողջական ինֆորմացիա են ստանում համապատասխան ծառայության հանգույցներից:

Քայլ 4. միևնույն ծառայության հանգույցի հետ կապված աշխատանքային հանգույցների զույգերը ստանում են ընդհանուր/բաշխված բանալի, որը պայմանավորում է կապի անվտանգությունն այդ հանգույցների միջև:

Իրերի ինտերնետ ցանցը դիտարկվում է որպես կամայականորեն բաշխված միատեսակ և տարատեսակ իրերի ընդլայնվող ցանց, որտեղ նշված հանգույցների կարևոր գործառույթներից են՝ սենսորներից տվյալների հավաքագրումը և գրանցումը [42, 43]: Ընդ որում, ենթադրվում է, որ հանգույցների որոշ պարամետրեր նախասահմանված և նախաբեռնված են՝ նախքան հանգույցի տեղակայվելը ցանցում, ինչպես նշված է աղյուսակ 1-ում:

Աղյուսակ 1. Հանգույցների նախասահմանված և նախաբեռնված պարամետրերը

Պարամետրեր	Պարզաբանումը/նկարագրումը
T_s	Ծառայության հանգույցի ընտրության մեկ փուլի ժամանակը
a, b	Երկու պարզ թիվ՝ բանալիների ենթակառուցվածքի գեներացման համար՝ Ռաբինի անհամաչափ գաղտնահամակարգի բանալու օգտագործմամբ
λ	Ծառայության հանգույցի կողմից սպասարկվող հանգույցների առավելագույն քանակը
H	Վերահասցեավորման սահմանային արժեքը
P_s	Որպես ծառայության հանգույց ընտրվելու հավանականությունը
UID	Հանգույցի եզակի նույնականացման նշիչը՝ 6LowPAN
T_{total} ,	Գործարկման առավելագույն ժամանակը

Ցանցում միատեսակ իրերի ռեսուրսները և հաշվողական հնարավորությունները կարող են կարող են զգալիորեն տարբերվել: Հանգույցների միջև տարբերությունները նկարագրելու համար, կիրառվել է Ps արժեքը, որի թվային արժեքը գտնվում է 0 – 1.2 միջակայքում և հաշվարկվում է՝ ելնելով տվյալ բանաձևից $P_p + (1 – 1 / λ)$, որտեղ P_p

պարամետրը ցույց է տալիս տեխնիկական ռեսուրսների գնահատականը և կարող է ընդունել 0 – 0.2 արժեքը:

H-ը վերահասցեավորման սահմանային արժեքն է, որը ցույց է տալիս այն առավելագույն ցատկերի քանակը, որի ընթացքում բանալին փոխանցվում է ցանցով դեպի նշանակետ հանգույց:

Բանալիների բաշխման գործընթացը մեկնարկում է յուրաքանչյուր հանգույցին՝ իր սպասելու T_{total} առավելագույն ժամանակի, ծառայության հանգույցի ընտրության համար մեկ T_s փուլի և λ առավելագույն քանակի նախաբեռնմամբ՝ հստակեցնելով սպասարկվող աշխատանքային այն հանգույցների քանակը, որոնք ծառայության հանգույց են դառնում:

Բացի այդ, յուրաքանչյուր հանգույց պատահականորեն ընտրում է երկու պարզ թիվ՝ a և b, որոնք տեղակայումից առաջ օգտագործվում են ներդրված անհամաչափ գաղտնահամակարգում որպես գաղտնի բանալի:

Հանգույցի նույնականացման համար ենթադրվում է, որ յուրաքանչյուր հանգույց ունի անհատական IPv6 հասցե: Այդ դեպքում, օգտվելով 6LoWPAN տեխնոլոգիայից, հնարավոր է ստանալ նույնականացման եզակի ցուցիչ (UID): Նշենք, որ 6LoWPAN-ն փոքր հզորություն օգտագործող անլար բաշխված ցանց է, որտեղ ամեն մի հանգույց ունի IPv6 հասցե՝ ինտերնետի հետ ուղղակի կապ հաստատելու համար[44] :

2.2. Բաշխված բանալիների ենթակառուցվածքի ստեղծումը

Ինչպես հայտնի է, անհամաչափ գաղտնահամակարգերն անվտանգության հիմնարար գործառույթ են ապահովում անլար ցանցերում և օգտագործվում են նաև իրերը իրերին կապի անվտանգությունն ապահովելու նպատակով [34,35]: Մյուս կողմից՝ իրերի ինտերնետում միատեսակ հանգույցների ռեսուրսների սահմանափակությունը լուրջ արգելք է բանալիների զույգերի կառավարման ավանդական եղանակների, բաց բանալիով գաղտնագրման և բանալու բաշխման կենտրոնի (Key Distribution Center)-ի գործածման համար:

Մինչ այժմ անհամաչափ գաղտնահամակարգերում կիրառվում են բաշխված բանալիների ենթակառուցվածքի ստեղծման դասական երկու մոդել՝

- բազմանդամային,
- մատրիցային:

Բազմանդամային հիմքով նախագծված բանալիների ենթակառուցվածքի ստեղծման համար օգտագործվում է համաչափ λ-աստիճանի բազմանդամը՝

$$f(x,y) = f(y,x) = \sum_{i,j=0}^{\lambda} a_{ij} x^i y^j \qquad \textbf{(2.1)}$$

կառուցված Fq վերջավոր դաշտի վրա, որտեղ q-ն պարզ թիվ է, որի արժեքը բավարար պետք է լինի՝ որպես բանալու արժեք օգտագործվելու և անհրաժեշտ գաղտնակայունություն ապահովելու համար: Իրերի ինտերնետ անլար ցանցում օգտագործելով հանգույցի նույնականացման եզակի նշիչը, կարելի է ստանալ բանալու վերաբերյալ ամբողջական ինֆորմացիա, որը հատկացվում է հանգույցին: Այսպիսով՝ I հանգույցը բանալին կարող է ստանալ տվյալ f (i, j) ֆունկցիայով [36]:

Հետևաբար, երկու հանգույց կարող են հաշվարկել ընդհանուր բանալին՝ իրենց բանալիների ինֆորմացիայից՝ որպես $f(x,y) = f(y,x)$: $f(x,y)$-ի բազմանդամային հիմք ունեցող բանալիների ենթակառուցվածքի ստեղծումը հիմնված է Բլունդոյի և մյուսների կողմից նկարագրված սխեմայի վրա [36]:

Մատրիցային հիմքով նախագծված բանալիների ենթակառուցվածքի մոդելն օգտագործում է (λ + 1) × (λ + 1) հանրային G մատրիցը և (λ + 1) × (λ + 1) գաղտնի D մատրիցը՝ վերջավոր Fq դաշտի վրա, որտեղ q-ն դարձյալ պարզ թիվ է [37]: Վերոհիշյալ երկու մատրիցներն օգտագործվում են երրորդ A մատրիցը գեներացնելու համար.

$$\mathbf{A = (D \cdot G)^T} \qquad \mathbf{(2.2)}$$

Եթե D մատրիցը համաչափ է, ապա K մատրիցը նույնպես համաչափ է, որը հաշվարկվում է հետևյալ կերպ.

$$\mathbf{K = A \cdot G} \qquad \mathbf{(2.3)}$$

(2.3) հավասարումից կարող ենք ստանալ $k_{ij} = k_{ji}$, որտեղ k_{ij}–ն K մատրիցի i-րդ տողի և j-րդ սյան տարրն է՝ i, j = 1,2,3, … , λ + 1:

Եթե I հանգույցին հատկացված է ընդհանուր բանալի, որը պարունակում է A-ի i-րդ տողը և G-ի i-րդ տողը, ապա i և j երկու հանգույցները կարող են հաշվարկել իրենց ընդհանուր kij բանալին՝ G-ի սյունակների փոխանակման միջոցով:

Բացի այդ, եթե G մատրիցը նախագծված է օգտագործելով Vander Monde Matrix, ապա միայն հիմքային մասը պետք է փոխանցվի հանգույցների միջև ողջ սյունակի փոխարեն [37]: Հարկ է նշել, որ բանալու կառավարման առաջարկված սխեման կարող է աշխատել բանալիների ենթակառուցվածքի գեներացման երկու մոդելի հետ:

Ծառայության հանգույցը դերակատարման փուլի ավարտից հետո կատարում է ենթակառուցվածքի համաչափ գաղտնագրում և բանալու բաշխում օգտագործելով գաղտնիքի բաշխման մեթոդը, որի շեմային արժեքն է [λ – 1, λ] [35]:

2.3. Ինտերնետ իրերի դասակարգումը համապատասխան խմբերում

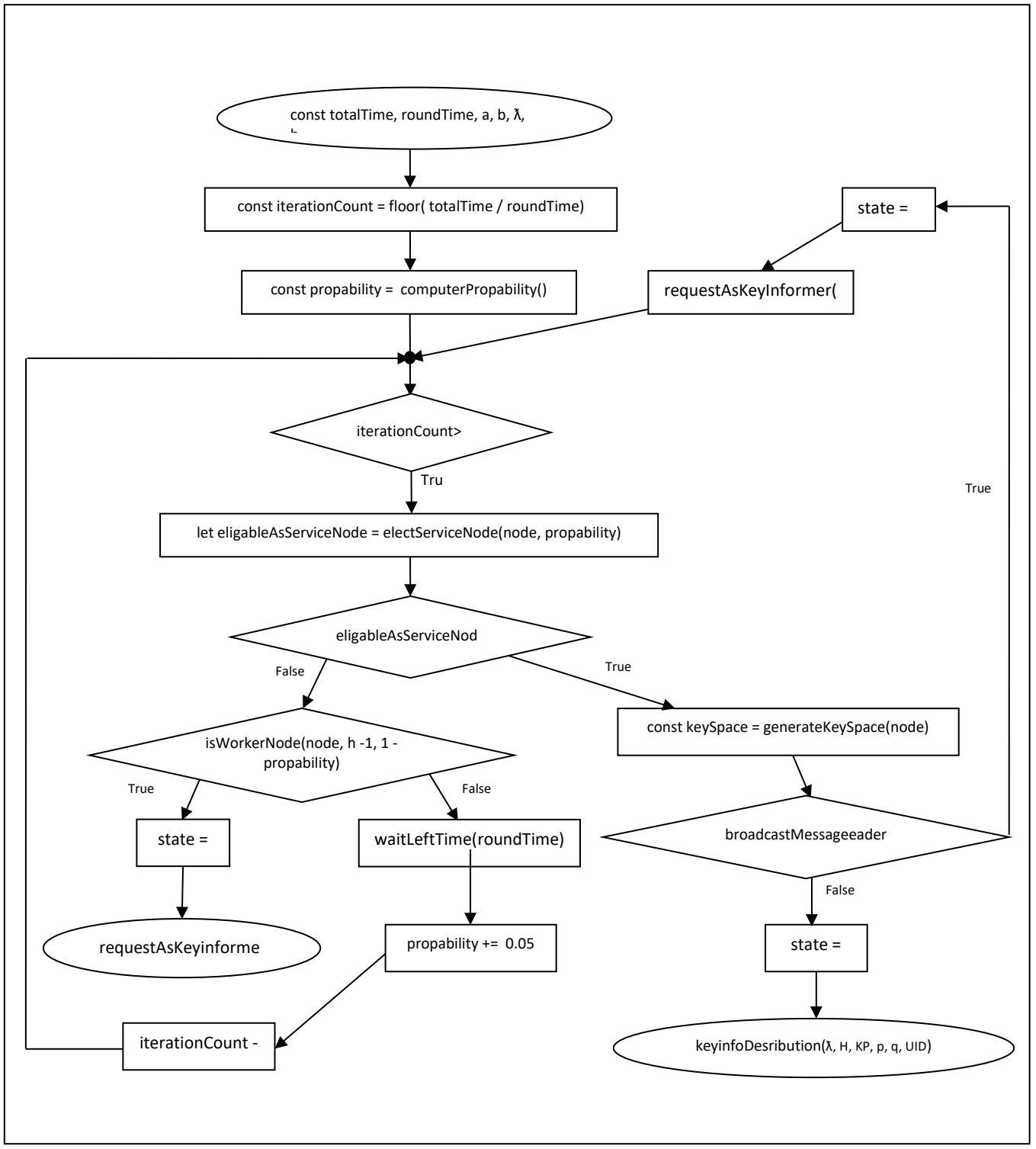
Իրերի ինտերնետ միջավայրի առաջարկվող մոդելում հանգույցների գործարկումը կատարվում է ինքնաբերաբար, որի ընթացքում հանգույցներին վերագրվում են համապատասխան դերեր (նկ.4):

Հանգույցը ցանցին միանալուն պես՝ փորձում է ընտրել իրեն որպես ծառայության հանգույց՝ ալգորիթմ 1-ում բերված քայլերի համաձայն, երբ տվյալ հանգույցը կարող է ընտրվել որպես ծառայության, այլընտրանքային կամ աշխատանքային հանգույց: Պարզության համար ենթադրենք, որ $T_{total} = T_s \cdot t$, որտեղ t-ն հանգույցի ընտրության և ինքնակարգավորման փուլերի ընդհանուր քանակն է: Ինքնակարգավորումը բխում է հանգույցների ինքնակազմակերպման բնույթից:

Այսպիսով, N_i-րդ հանգույցը նախ փորձում է ինքնուրույն ընտրել իրեն որպես ծառայության հանգույց՝ P_s հավանականությամբ: Եթե փորձը հաջողություն է ունենում, ապա N_i-րդ հանգույցը դառնում է ծառայության հանգույցի թեկնածու և ստեղծում է համապատասխան KP_i բանալու ենթակառուցվածքը: Այնուհետև, N_i-րդ հանգույցը ստուգում է, թե արդյոք գոյություն ունի՞ հեռարձակված հաղորդագրություն:

Եթե N_i հանգույցը ստանում է հեռարձակման հաղորդագրություն, նշանակում է՝ միջակայքում առկա է (H-1) ցատկ ծառայության հանգույց: Հանգույցի ողջ հաշվողական ռեսուրսի օգտագործման համար N_i-րդ հանգույցն ընտրվում է որպես այլընտրանքային ծառայության հանգույց:

Եթե N_i հանգույցը չի ստանում հեռարձակման հաղորդագրություն, ապա N_i –րդ հանգույցը դառնում է ծառայության հանգույց: Այնուհետև այն իր կարգավիճակը՝ H ցատկի միջակայքում հայտնում է իր հարևաններին, որից հետո հանգույցն ավարտում է ընտրության ընթացակարգը:



Նկ.4. Հանգույցների դերերի որոշման ալգորիթմի բլոկ-սխեման

Եթե հանգույցը հնարավորությունն չի ստանում դառնալ ծառայության հանգույց, ապա նրան վերագրվում է աշխատանքային հանգույցի կարգավիճակ:

N_i –րդ հանգույցը ստուգում է, թե արդյոք առկա՞ է եղել ծառայության հանգույց, որի միջակայքում արդեն գոյություն է ունեցել (H-1)-ցատկ: Եթե ընթացիկ փուլում ոչ մի ծառայության հանգույց չի հայտնաբերվում (H-1)-ցատկ միջակայքում, ապա N_i-րդ հանգույցը մասնակցում է հաջորդ փուլին, և՝ այդպես շարունակ:

2.4. Անվտանգ կապուղու հաստատումն իրերի ինտերնետ միջավայրում

Նախքան աշխատանքային հանգույցները կպահանջեն բանալիների վերաբերյալ ինֆորմացիա՝ իրենց համապատասխանող ծառայության հանգույցներից, հաստատվում է անվտանգ կապուղի դեպի ծառայության հանգույցը:

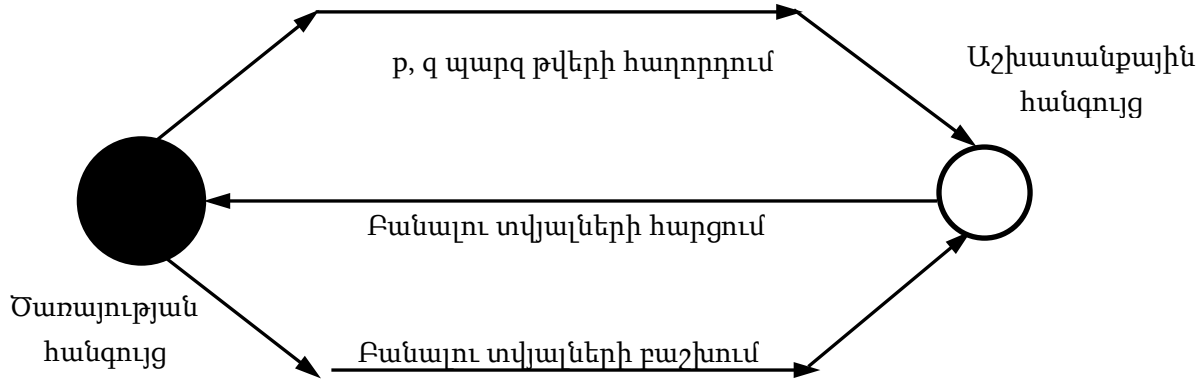
Ինչպես ցույց է տրված նկ.5-ում, նախ հեռարձակվում է հանգույցի նույնականացման եզակի նշիչը (uid), n բաց բանալին և LH վերահասցեավորման սահմանը(left h) իրեն համապատասխանող հանգույցների միջակայքում: LH սկզբնական արժեքը հավասար է H՝ վերահասցեավորման սահմանին: Ցանկացած հանգույց, ստանալով հաղորդագրությունը, ստուգում է. եթե LH > 0-ից, ապա 1-ով փոքրացնում է արժեքը և վերահասցեավորում հաղորդագրությունը: Այնուհետև, աշխատանքային հանգույցն ընտրում է պատահական k թիվ և գաղտնագրում է պահանջված հաղորդագրությունը՝ օգտագործելով անհամաչափ գաղտնահամակարգը [38]

$$E_n (k \,||\, B) = (k \,||\, B)^2 \bmod n, \qquad (2.4)$$

(2. 4) հավասարման լուծման արդյունքում $E_n (k \,||\, B) \,||\, B$ գաղտնագիրն ուղարկվում է ծառայության հանգույցին՝ կցելով այն ուղարկվող փաթեթին: Ի վերջո, ծառայության հանգույցը վերծանում է պահանջված K բանալու ինֆորմացիան՝ հաշվարկելով $D_{p,a} (E_n (k \,||\, B))$: Ինչպես նշվել է ծառայության հանգույցը, բանալիների ենթակառուցվածքի ձևավորումից հետ կատարում է ստացված արդյունքի գաղտնագրում, այնուհետև Շամիրի մեթոդի օգնությամբ բաշխում այդ բանալին λ մասի: Օգտագործելով ստացված K բանալին՝ ծառայության հանգույցը գաղտնագրում է λ_0 մասնիկը, հեռացնում է այն առկա մասնիկների ցանկից և փոխանցում աշխատանքային հանգույցին: Հետագայում օգտագործելով այդ մասնիկիները՝ խափանումների դեպքում այլընտրանքային հանգույցը կարող է փոխարինել ծառայության հանգույցին: Այսպիսով, k-ն կարող է օգտագործվել

որպես գաղտնի բանալի՝ աշխատանքային հանգույցի և դրան համապատասխանող ծառայության հանգույցի միջև: Գաղտնի բանալին կարող է ստեղծվել հայտնի համաչափ ալգորիթների համար, որոնք են՝ AES, DES և այլն:

Աղյուսակ 2-ը ներկայացնում է հաղորդագրությունների ձևաչափը, որոնք օգտագործվում են աշխատանքային հանգույցի և դրան համապատասխանող ծառայության հանգույցի միջև հաղորդագրության փոխանակման գործընթացում՝ անվտանգ կապուղու հաստատման նպատակով:



Նկ.5 Ծառայության հանգույցի և դրան համապատասխանող աշխատանքային հանգույցի միջև հաղորդակցության սխեման

Աղյուսակ 2. Հաղորդագրությունների ձևաչափը

Ծառայության հանգույցից դեպի բոլոր աշխատանքային հանգույցներ		
UID	n=p.q	LH

Ծառայության հանգույցից դեպի աշխատանքային հանգույց			
UID	Գաղտնագրված տվյալներ	*CtrlFlag*	LH

Բանալիների բաշխումը կատարվում է հետևյալ կերպ.

Նախ՝ ծառայության հանգույցը հեռարձակում է իր UID և $\boldsymbol{n}$-ը՝ համապատասխան միջակայքում գտնվող հանգույցներին: Այնուհետև, ծառայության հանգույցը պատրաստվում է սպասարկել աշխատանքային հանգույցների կամ այլընտրանքային ծառայության հանգույցների կողմից ուղարկվող հարցումներին, քանի դեռ այդ հարցումների քանակը չի գերազանցում λ մեծությունը:

Եթե ստացված հարցման հաղորդագրությունն ուղարկված է աշխատանքային հանգույցից, ծառայության հանգույցը վերծանում է հաղորդագրությունը, այնուհետև գտնում է չօգտագործած բանալի տվյալ հարցման համար և ուղարկում այն համապատասխան աշխատանքային հանգույցին:

Եթե հարցումն ուղարկվել է այլընտրանքային հանգույցից, որի եզակի նշիչը առկա չէ ASList-ում, ապա ծառայության հանգույցը փոխանցում է գաղտնագրված բանալիների տիրույթը և ավելացնում եզակի նշիչը ASList-ում:

2.5. Բանալիների բաշխման առաջարկված սխեմայի կայունության գնահատականը

Բանալիների ենթակառուցվածքի ստեղծման և կառավարման սխեման գնահատվում է հետևյալ բնութագրերով.

- հիշողության ծանրաբեռնում,
- հաղորդակցում,
- հաշվողական գերբեռնում,
- կայունություն հարձակումների նկատմամբ:

Հիշողության ծանրաբեռնում: Բանալիների առաջարկված կառավարման սխեման կարող է նպաստել անլար ցանցերում իրերը իրերին կապի անվտանգության ապահովմանը և բարելավել իրերի ինտերնետ միջավայրում ծառայության որակի բարձրացմանը: Առաջարկված սխեմայում յուրաքանչյուր աշխատանքային հանգույց կարող է ձեռք բերել բանալու տեղեկատվությունը ծառայության հանգույցից:

Ենթադրենք՝ N հանգույցները պատահականորեն բաշխված են ցանցում: Այդ դեպքումընտրման առաջին ծառայության հանգույցների թիվը կազմում է՝ $\boldsymbol{N^1_{service} = N * P_S}$

Երկրորդ և i-րդ փուլում ծառայության հանգույցների թիվը կազմում է՝

$$\boldsymbol{N^2_{service} = \left(N - N^1_{service}\right)\left(1 - P_S\right)^{D_{H-1}} \cdot P_S}$$

$$\boldsymbol{N^i_{service} = \left(N - \sum_{j-1}^{i-1} N^j_{serivce}\right)\left(\left(1 - P_S\right)^{(i-1)}\right)^{D_{H-1}} \cdot P_S\ ,} \qquad \textbf{(2.5)}$$

որտեղ D_H-1-ը H-1 միջակայքում հարևան հանգույցների թիվն է: Ինչպես արդեն քննարկվել է, կա t փուլ հանգույցի ընտրման և ինքնակարգավորման համար: Յուրաքանչյուր աշխատանքային հանգույցում պահպանված բանալիների միջին թիվը հաշվելու

նպատակով այլընտրանքային ծառայության բանալու համար բուֆերի չափն ընդունենք L-ով: Այնուհետև, աշխատանքային հանգույցների թիվը կազմում է՝

$$N_W = \left(1 - \frac{L}{\lambda}\right) \cdot \left(N - \sum_{i=1}^{t} N_{service}^{i}\right) \quad (2.6)$$

Հետևաբար, բանալու ինֆորմացիայի միջին քանակը, որը պետք է պահպանվի յուրաքանչյուր աշխատանքային հանգույցում, կարող է գնահատվել հետևյալ մեծությամբ

$$Num_{keys} = \frac{\lambda \cdot \sum_{i=1}^{t} N_{service}^{i}}{N_w} = \frac{\lambda \cdot \sum_{i=1}^{t} N_{service}^{i}}{\left(1-\frac{L}{\lambda}\right)\cdot\left(N - \sum_{i=1}^{t} N_{service}^{i}\right)} \quad (2.7)$$

Առաջարկվող սխեմայում ընդհանուր հնարավոր ամբողջ ցանցային հաղորդակցման գերբեռնվածությունը հաշվարկվում է

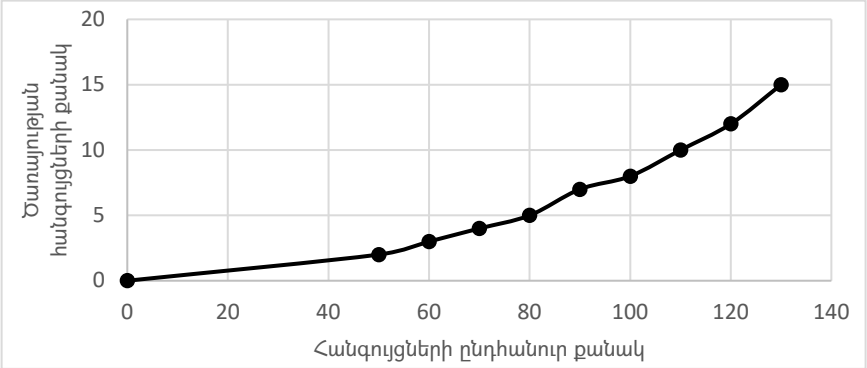
$$\sum_{i=1}^{t} N_{service}^{i} \cdot (B + \lambda + L) + \bar{H} \cdot \left(N - \sum_{i=1}^{t} N_{service}^{i}\right) \cdot \lambda + \bar{D} \quad (2.8)$$

բանաձևով, որտեղ B-ն ծառայության հանգույցի կողմից հեռարձակվող հաղորդագրությունների թիվն է, իսկ L-ը՝ ծառայության հանգույցի կողմից պահպանված այլընտրանքային ծառայության հանգույցների թիվը: Հետևաբար, առաջին գործոնն ամբողջ ծառայության հանգույցների հաղորդակցման գերբեռնվածությունն է [39]:

Առաջարկվող սխեմայում աշխատանքային հանգույցի հաշվողական գերբեռնվածության արժեքը կախված է հետևյալ երեք գործոնից՝

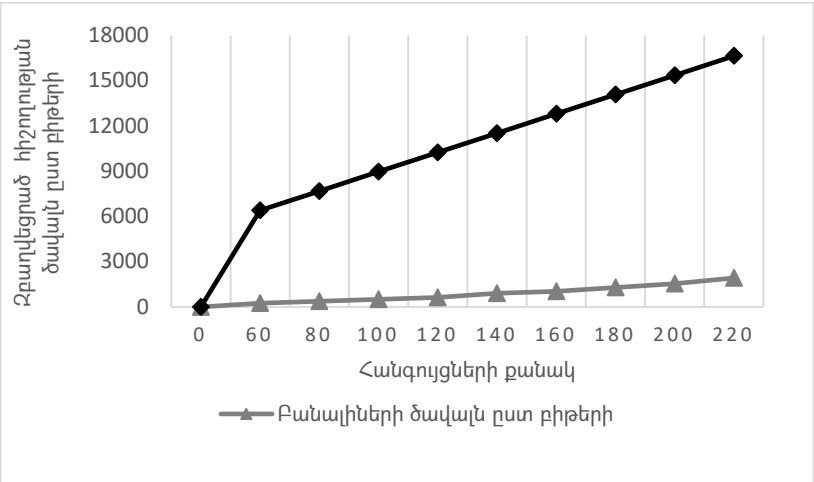
- անվտանգ կապուղու հաստատման համար աշխատանքային հանգույցի և համապատասխան ծառայության հանգույցի միջև k ընդհանուր բանալու գաղտնագրման ժամանակը,
- բանալու ինֆորմացիայի ձեռքբերման փուլում համապատասխան ծառայության հանգույցներից ձեռք բերված բանալու ինֆորմացիայի վերծանման ժամանակը,
- ընդհանուր բանալու դուրսբերման փուլում հարևանների հետ ընդհանուր բանալիների հաշվարկման ժամանակը:

Նկար 6-ում պատկերված է ծառայության հանգույցների և հանգույցների ընդհանուր քանակի հարաբերակցությունը: Ինչպես նաև նկար 7-ում բերված է բանալիների պահպանման համար անհրաժեշտ հիշողության ծավալը՝ համեմատած դասական բանալիների պահպանման մեթոդի հետ:



Նկ.6. Ծառայության հանգույցների քանակի կախվածությունը հանգույցների ընդհանուր քանակից

Բանալիների ենթակառուցվածքի ստեղծման երկու մոդելն էլ ունեն λ դիմացկունություն: Դա նշանակում է, որ λ քանակով հանգույցները պետք է գրավված լինեն՝ ցանցում բանալիները բացահայտելու նպատակով: Քանի որ յուրաքանչյուր ծառայության հանգույց գեներացնում է բանալու ենթակառուցվածքը պատահականորեն և անկախորեն, ուստի առաջարկվող մեթոդը տրամադրում է հուսալի կայունություն:



Նկ.7. Բանալիների պահպանման համար անհրաժեշտ հիշողության ծավալի համեմատականը դասական և առաջարկված մեթոդների դեպքում

2.6. Պարզ ինտերնետ իրերի տվյալների անվտանգ փոխանակման գաղտնահամակարգի մշակում

ԻՆչպես նշվեց գոյություն ունեն պարզ ինտերնետ իրեր, որոնց սահմանափակումները թույլ չեն տալիս կիրառել անվտանգ հաղորդակցման ստանդարտները: Նույն HTTPS արձանագրության օգտագործումը ենթադրում է հավելյալ հիշողություն, բարձր հաշվողականություն և համապատասխան ծրագրային ապահովում: Այս պահանջները բավական մեղմ են, բայց կարող են պահանջել սարքի տեխնիկական պարամետրերի փոփոխությունների կատարում, որոնք զգալիորեն կբարձրացնեն նրա ինքնարժեքը:

Տեղեկատվական անվտանգության տեսանկյունից պարզ ինտերնետ իրերի անվտանգության հիմնական խնդիրները կապված են տվյալների փոխանակման հետ [41]: Տվյալների փոխանակումը բաց տեսքով արգելք է տվյալ տեսակի պարզ ինտերնետ իրերի հետագա զարգացման և կիրառման համար:

Պարզ ինտերնետ իրերը, որոնք իրականացնում են հստակ գործառույթներ, հիմնականում կատարում են տվյալների փոխանակում միայն հրամանների տեսքով՝ ցանցի ծանրաբեռնումից խուսափելու նպատակով: Այդ հրամանները որպես կանոն հաստատուն թվային կամ տեքստային արժեքներ են, որոնք ուղարկվում են հարցման մարմնին: Ակնհայտ է, որ ցանկացած ինտերնետ իրին տրված հրամանների վերծանումը բարդություն չի ներկայացնում և անվտանգության ապահովման կարևոր խնդիր է [40]:

Վերոնշյալ խնդիրների լուծման համար մշակվել է գաղտնահամակարգ, որը հեշավորման միջոցով ապահովում է ելքային/մուտքային հարցումներում առկա հրամանների գաղտնիությունը, ինչպես նաև փոխանցվող տեղեկատվական տվյալների անվտանգությունը կատարելով գաղտնագրային գործառույթներ: Այդ

գաղտնահամակարգը ներդրվելու է լրացուցիչ սարքում, որը շարժական բանալի է համացանցի և ինտերնետ իրերի միջև[41]:

Հաշվի առնելով այն փաստը, որ առաջարկված գաղտնահամակարգը աշխատելու է պարզ իրերի ինտերնետ միջավայրում և ապահովելու է միաժամանակ տարբեր քանակի ինտերնետ իրերի անվտանգությունը, ուստի այդ գաղտնահամակարգը պետք է բավարարի ստորև թվարկված պահանջները.

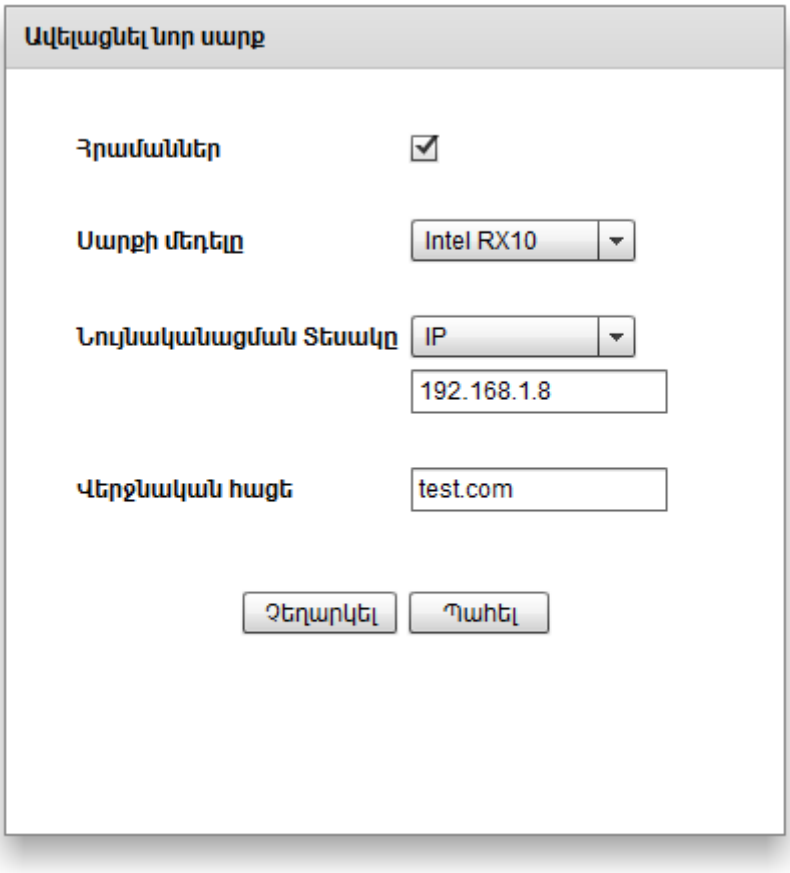
- միաժամանակ ապահովի մի քանի պարզ ինտերնետ իրերի անվտանգությունը:
- կատարի բանալիների անվտանգ փոխանակում:
- արտադրողական սահմանափակումներից ելնելով՝ փոխանցվող հրամանների անվտանգության ապահովման համար խուսափի գաղտնագրման/վերծանման գործառույթներից:

Անհրաժեշտ անվտանգության մակարդակն ապահովելու համար առաջարկված գաղտնահամակարգը պետք է կատարի հետևյալ գործառույթները.

- հրամանների արտապատկերում,
- հարցումների վավերացում,
- մուտքային և ելքային տվյալների գաղտնագրում,
- բանալիների կառավարում:

Այսպիսով շարժական բանալի սարքը, ստեղծում է անլար ցանց, որին կարող են միանալ պարզ տեսակի ինտերնետ իրերը: Այնուհետև կատարվում են հետևյալ քայլերը:

Քայլ 1 Շարժական բանալու կառավարման վահանակի միջոցով ավելացվում են ցանցին միացված ինտերնետ իրերի կարգաբերումները (նկ.8), որի հիման վրա սերվերային համակարգը ձևավորում է գաղտնի բանալին և փոխանցում շարժական բանալուն:

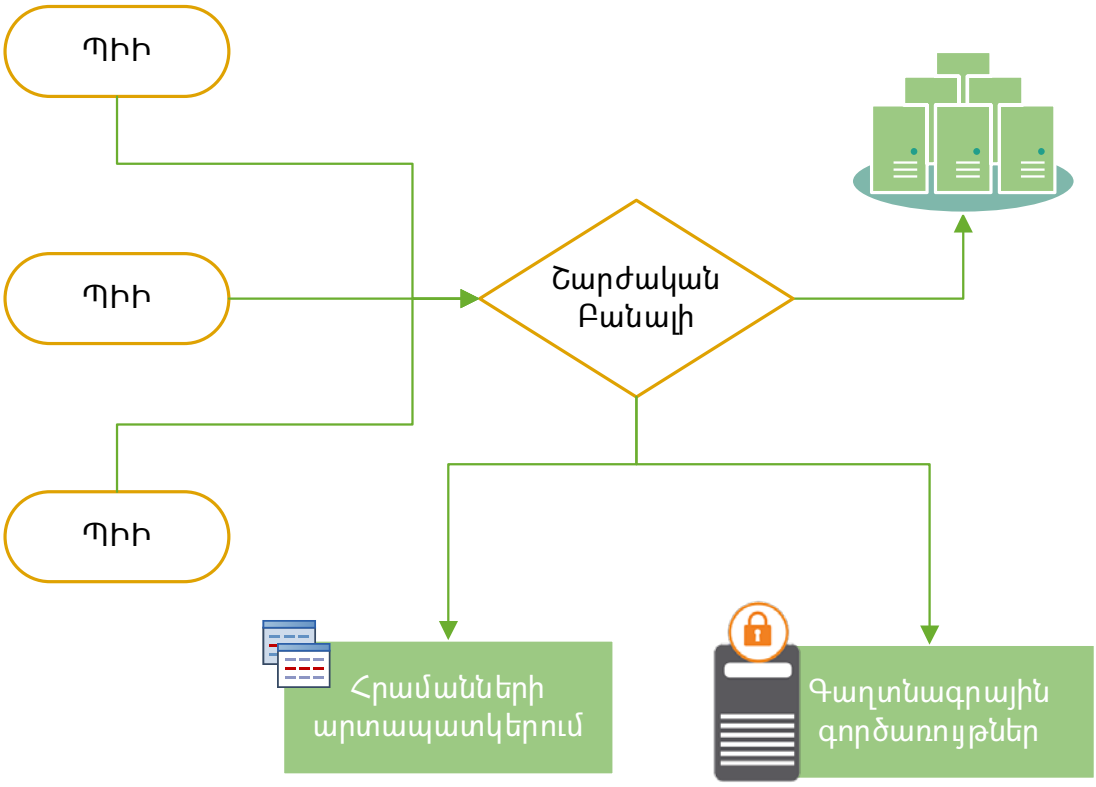


Նկ.8. Շարժական բանալու կառավարման վահանակում նոր կարգաբերումների ավելացման պատուհանը

Քայլ 2. Շարժական բանալու միջոցով ուղարկվող և ստացվող հարցումները դասակարգվում են 2 դասի՝ հեշային և գաղտնագրային. հարցումներից կախված կատարվում է կամ հրամանների/հեշերի արտապատկերում կամ տվյալների գաղտնագրում/վերծանում:

Քայլ 3.1. Եթե հարցումը դասակարգվում է որպես հեշային, ապա շարժական բանալին կատարում է հրամանների / հեշերի հաշվարկում և հարցումների նույնականացում:

Քայլ 3.2. Եթե հարցումը դասակարգվում է որպես գաղտնագրային, ապա կատարվում է ուղարկվող հարցման վերահասցեավորում https արձանագրության միջոցով:



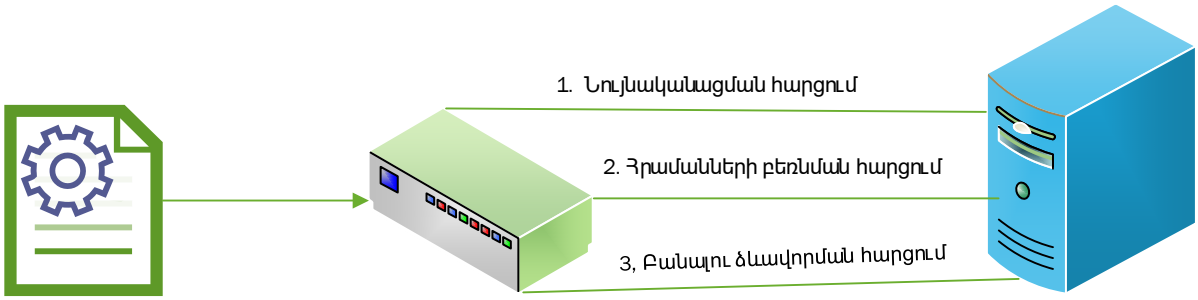
Նկ.9. Գաղտնահամակարգի աշխատանքի սխեման

Քայլ 1-ում շարժական բանալուն միացված պարզ ինտերնետ իրի կարգաբերումները ավելացնելիս նկ.8-ում ներկայացրած «հրամաններ» կարգաբերումը պետք է նշվի միայն այն դեպքում, երբ սարքի մոդելը առկա է համապատասխան դաշտի մոդելների ցուցակում, այլապես անհրաժեշտ է ավելացնել սարքը ինքնուրույն՝ նշելով սարքի կողմից ընդունվող բոլոր հրամանները, որոնք անմիջապես ուղարկվում են սերվերին:

Շարժական բանալին ստանալով համապատասխան կարգաբերումները և ստուգելով սարքի պատկանելությունը հեշավորման դասին՝ հարցում է կատարում սերվերայն համակարգին, ուղարկելով կառավարման վահանակում նշված օգտատիրոջ անունը և գաղտնաբառը՝ նույնականացում կատարելու նպատակով:

Նույնականացումը հաջող ավարտելուց հետո, եթե շարժական բանալիում առկա չէ կարգաբերումներում նշված սարքի ամբողջ հրամանների ցանկը, կատարվում է լրացուցիչ հարցում այդ հրամանները ստանալու և պահեստավորելու նպատակով:

Այնուհետև կատարվում է ևս մեկ հարցում՝ ի պատասխան որի շարժական բանալին ստանում է գաղտնի բանալին և այնուհետև օգտագործում է այն մուտքային և ելքային հարցումները սպասարկելու համար:



Նկ.10. Շարժական բանալու կողմից կատարվող հարցումները

Սերվերը, ստանալով նույնականացում կատարած սարքից գաղտնի բանալու ձևավորման հարցում, նախ ձևավորում է N * M չափի ASCII-ի սիմվոլներից բաղկացած երկչափ R զանգվածը (աղյուսակ 3) օգտագործելով «/dev/urandom/» հրամանը, որը անվտանգ է համարվում գաղտնագրային գործընթացներում օգտագործելու համար: Դասական գաղտնի բանալու ձևավորման գործընթացում N = M = 8: Կարևոր է նշել, որ վերոհիշյալ զանգվածը բաղկացած եզակի սիմվոլներից, որոնք չեն կրկնվում ամբողջ զանգվածում: Տվյալ զանգվածի ձևավորման հնարավոր տարբերակների քանակը բավական մեծ թիվ է, որը կարելի է հաշվել հետևյալ բանաձևով՝ $\prod_{i=0}^{M*N}(256-i)$, որտեղ M և N փոփոխականները ձևավորված R մատրիցի տողերի և սյուների քանակն են: Այնուհետև պատահական սկզբունքով ընտրվում է նախօրոք գեներացված G մատրիցը, որը

իր մեջ պարունակում է M x N չափի ինդեքսների զանգված, որտեղ յուրաքանչյուր ինդեքսը եզակի է և գտնվում է 0 < Index < M * N միջակայքում: Այս զանգվածը ձևավորվում է Համելտոնի կամ այսպես կոչված շախմատային ձիու շրջագայության խնդիրը լուծելիս, որը համարվում է NP բարդության խնդիր և ունի լուծման 9,591,828,170,979,904 եղանակ: Վերոհիշյալ G մատրիցը հնարավոր է ստանալ՝ հարցում կատարելով ընդհանուր ինդեքսային բանալիների կենտրոնին կամ հաշվարկել հատարկման տարբերակով:

Աղյուսակ 3. Ձևավորված R մատրիցի օրինակը

a	6	t	S	,	*		)
]	!	#	f	[	s	G	4
q	d	0	*	?	7	V	3
8	m	y	U	x	-	]	=
b	2	K	w	!	5	g	.
h	+	4	O	r	z	C	v
j		/	p	e	\	D	(
9	I	?	1	[	X	I	h

Ընդհանուր ինդեքսային բանալիների կենտրոնը ամպային միջավայր է, որում կատարվում է G մատրիցների ձևավորում՝ օգտագործելով այսպես կոչված Վարնսդորֆի կանոնը, ըստ որի հաջորդ քայլը կատարելիս գնահատվում է բոլոր հնարավոր քայլերը, այնուհետև ընտրվում է նվազագույն գնահատական ստացած քայլը: Քայլերի գնահատումը կատարվում է՝ հաշվարկելով այդ քայլի կատարման դեպքում առկա բոլոր հնարավոր քայլերի քանակը: Արդյունքում ձևավորվում է G_i ինդեքսներից / քայլերից բաղկացած M * N չափի մատրից, որտեղ G_i <= M * N: Շնորհիվ այդ քայլերի հերթականության շարժական

բանալին և սերվերը կարող են դուրս բերել ընդհանուր հեշավորման բանալի, որը կօգտագործվի հարցումները նույնականացնելու նպատակով:

Այսպիսով շարժական բանալու հարցման արդյունքում ձևավորվում է գաղտնի բանալու ձևաչափը, որը բաղկացած է M * N բիթ ծավալ ունեցող պատահականության սկզբունքով ձևավորված R մատրիցից, R մատրիցի երկարությամբ G մատրիցից, N և M փոփոխականներից, որոնց ծավալը կազմում է 2 բիթ: Վերոհիշյալ ձևաչափը ներկայացված է աղյուսակ 5-ում:

Աղյուսակ 4. Հրամանների և հեշերի արտապատկերումը

	Հրամաններ	5	MAKE_TEA
1	TURN_ON	6	MAKE_COFFE
2	TURN_OFF	7	CANCEL_MAKING
3	TIMER_30_MIN	.....	.....
4	TIMER_15_MIN	S_{total}	CANCEL_MAKING

Բանալու ձևավորման հարցմանը ուղարկվող պատասխանում, բացի ձևավորված բանալուց, հարցման գլխամասին ավելացվում է նաև այդ բանալու 128 բիթ երկարությամբ եզակի նույնականացման նշիչից՝ PKID-ին: Այսպիսով ֆայլի ընդհանուր ծավալը համարժեք է W = |R| + |G| + 2 բանաձևի արժեքին, որտեղ |R|, |G| և |PKID| համապատասխան տվյալների երկարություններն են: Դասական տարբերակում, որտեղ M = N = 8, ծավալը դառնում է W = 130: Արդյունքում գաղտնի բանալու ձևավորման գործընթացն ավարտելուց հետո այդ բանալին և PKID ուղարկվում են որպես պատասխան բանալու ձևավորման հարցման համար:

Աղյուսակ 5. Ձևավորված բանալու ձևաչափը

Տվյալների հերթականությունը	M	N	R	G
Անհրաժեշտ ծավալը	1	1	M * N	M * N

Քայլ 2-ում նշված հարցումների դասակարգումն իրականացվում է գաղտնահամակարգին տրված կարգաբերումների միջոցով: Հարցում կատարած սարքի կարգաբերումների ստացման համար օգտագործվում է հարցման մեջ առկա ինտերնետ իրի հասցեն կամ հնարավորության դեպքում՝ գլխամասում առկա եզակի MAC հասցեն: Դասակարգման արդյունքում պարզ է դառնում հարցման տեսակը, որը կարող է լինել կամ հեշային, կամ գաղտնագրային: Հեշային հարցման պարագայում հարցման մեջ պետք է առկա լինեն նաև KFI(KeyFramIndex), PKID, Salt և VH(ValidationHash) տվյալները, որոնք օգտագործում են հարցման վավերականացման համար:

Աղյուսակ 6. Հաղորդագրությունների ձևաչափը

Գլխամաս	Մարմին		
Referrer | UID | MAC	KFI 64 bit	VH 128 bit	Salt N / 2 bit

Քայլ 3-ում օգտագործելով հարցման գլխամասում առկա PKID-ում գտնվում է համապատասխան բանալին, որի միջոցով կատարվում է հարցման նույնականացում և հրամանի բացահայտում: Վերոհիշյալ գործողություններն իրականացվում են ատենախոսությունում մշակված ալգորիթմի միջոցով, որը բաղկացած է չորս հիմնական փուլից:

Առաջին փուլում օգտագործելով G մատրիցի (K * N) դիրքից մինչ (K * M + N) դիրքում առկա ինդեքսները՝ ձևավորվում է KK բազմությունը, որտեղ K փոփոխականը համարժեք է տվյալ բանալու օգտագործման քանակին: KK բազմության ցանկացած K_i անդամին R

մատրիցում համապատասխանում է $R_{K[i]}$ սիմվոլ, արդյունքում ձևավորվում է N սիմվոլներից բաղկացած PP թվերի բազմություն կամ P տող: Նախ հաշվարկվում է հրամանի բանալին՝ C_k, հաշվարկելով KK և PP բազմությունների համապատասխան մասնիկների XOR արժեքների գումարը (2.9): Օգտագործելով հարցման մարմնում առկա KFI փոփոխականը՝ հաշվարկվում է տվյալ հրամանի C_i ինդեքսը ստորև գրված բանաձևով, որտեղ C_N առկա հրամանների քանակն է :

$$C_k = \sum_{l=0}^{N} PP_i \oplus KK_i;$$

$$C_i = (C_k * C_N) \oplus \text{KFI}; \quad \textbf{(2.9)}$$

Այնուհետև կատարվում է բացահայտած հրամանի հեշավորում՝ օգտագործելով P տողը որպես բանալի: Հարցման մարմնում առկա VH հեշի արժեքի և H_p(C + Salt) հեշի արժեքի հավասարության դեպքում հարցումը համարվում է վավեր, և շարժական բանալին նոր հարցում է կատարում այդ հրամանով համապատասխան ինտերնետ իրին: Հակառակ դեպքում հարցումը համարվում է անվավեր: Գոյություն ունի անվավեր հարցումների սահմանափակում, որի գերազանցման դեպքում կատարվում է հաջորդ բանալու հաշվարկումը:

Շարժական բանալուն հարցում ուղարկող կողմը ունենալով նույն գաղտնի բանալին կարող է հաշվարկել C_k-ն: KFI արժեքի հաշվարկը բերված է ստորև, որտեղ C_i դա կատարվող հրամանի համարն/ինդեքսն է, իսկ $* C_N$ ընդհանուր հրամանների քանակն է: $\text{KFI} = (C_k * C_N) \oplus C_i$: Մինչ հարցում ուղարկելը նաև գեներացվում է պատահական ascii սիմվոլներից բաղկացած 10 բիթ երկարությամբ SALT-ը և կատարվում է հրամանի հեշավորում օգտագործելով KK բանալին: Որպես հեշավորման հաղորդագրություն օգտագործվում է C_i հրամանի SALT-ի միավորումը, որը բացառում է հեշ արժեքի կապի բացահայտումը: Ստացված հեշը, KFI և SALT-ը ուղարկվում է շարժական բանալուն:

Ցանկացած վավեր հարցումից հետո K փոփոխականի արժեքը մեծացվում է մեկով, եթե K < N, այլապես K ստանում է 0 արժեք, և R մատրիցի $R_{[n, m-1]}$ մասնիկը տեղափոխվում է առաջին ինդեքս և վերադասավորում այն, որի շնորհիվ G մատրիցում նշված ինդեքսների արժեքներից ձևավորված KK բազմությունը դառնում է ամբողջովին տարբեր համեմատած մինչև R-մատրիցում կատարված ձևափոխությունների հետ: Երբ տեղափոխումների քանակը հասնում է N * M – 1, ավարտվում է առաջին փուլը և մինչ 2-րդ փուլին ացնելը կատարվում R մատրիցի «transpose» փոխարկում: Նախ տվյալ փոխարկման համար օգտագործելով միաչափ ինդեքսը, հաշվարկվում է ամեն մասնիկի երկչափ ինդեքսը հետևյալ ձևով $[i = Math.floor(index/N),\ j = index\ \%\ N]$ այնուհետև կատարվում է նոր ինդեքսի հաշվարկում $newIndex = j * N + i$ և այդ 2 ինդեքսներին համապատասխան արժեքների փոխարկում: Տվյալ գործողությունը կատարվում է ամեն մատրիցի մասնիկի համար: Արդյունքում հնարավոր է դառնում կիրառել առաջին փուլում նկարագրված նույն գործողությունները ձևափոխված մատրիցի վրա և ստանալ տարբեր բանալիներ առանց կրկնվելու հավանականության:

Երկրորդ փուլի ընթացքում կրկնվում են առաջին փուլում կատարած բոլոր գործողությունները մինչ «transpose» փոխարկումը, որի փոխարեն օգտագործվում է մեկ այլ փոխարկում հետևյալ բանաձևերով $i = M - Math.floor(index/N) - 1;\ j = N - index\ \%\ N - 1;\ newIndex = j * N + i$:

3-րդ փուլում նույնպես կատարվում են առաջին փուլում նկարագրած գործողությունները՝ ներառյալ «transpose» փոխարկումը, որին որպես լրացում կատարվում է մատրիցի տողերի փոխարկումը: Վերոհիշյալ փոխարկումը ստացվում է յուրաքանչյուր մասնիկի՝ իրեն համապատասխան մասնիկով փոխարկմամբ, որը հաշվարկվում է հետևյալ կերպ՝ $(N - i - 1) * N + j$: Այս փոխարկումը նման է մատրիցի 90°-ի թեքմանը:

4-րդ փուլում կատարվող փոխարկումը նման է մատրիցի -90°-ի թեքմանը, որի համար կատարվում է մատրիցի «transpose» փոխարկումը և սյուների փոխարկում: Սյան փոխարկման համար ամեն փոխարկվող մասնիկին համապատասխան մասնիկը հաշվարկվում է հետևյալ կերպ $i * N + N - j - 1$:

5-րդ փուլում կատարվում է մատրիցի 180°-ի թեքում, որի համար կատարվում է նախ մասնիկների փոխարկում ըստ տեղերի, այնուհետև՝ ըստ սյուների:

Նշված հինգ փուլերի ընթացքում կատարվող ձևափոխությունների օրինակը ներկայացված է նկ. 11-ում: Այս ձևափոխությունների արդյունքում հնարավոր է ձևավորել 2520 եզակի բանալի և կատարել հարցումների նույնականացում:

Օգտվելով շարժական բանալու կառավարման վահանակից՝ հնարավոր է կարգավորել այն հարցումները, որոնք պետք է հսկվեն առաջարկած գաղտնահամակարգի կողմից: Կարգավորման համար օգտագործվում է պարզ ինտերնետ իրերի տվյալների փոխանակման վերջավոր հասցեն, որը հսկման ցուցակում ավելացնելուց հետո ապահովում է ինտերնետ իրերի ելքային և մուտքային հարցումների անվտանգությունը: Առաջարկված հեշավորման դասի վրա հիմնված հրամանների փոխանակման համակարգը համեմատվել է նաև այլ հայտնի ալգորիթմների հետ արագագործության առումով, որի արդյունքները ներկայացված են աղյուսակ 7-ում: CMP (Command Message Protocol) դա առաջարկված գաղտնահամակագրի կարճ հապավումն է:

Նաև կատարվել է գաղտնահամակարգի կայունության վերլուծություն: Ութը սիմվոլներից բաղկացած բանալին հատարկման տարբերակով գտնելու համար անհրաժեշտ է կատարել առավելագույնը 2^{64} քանակի փորձ, որը հաշվարկվում է հետևյալ բանաձևով՝ $\prod_{i=0}^{M*N}(256 - i)$: Հաշվի առնելով այն փաստը, որ գոյություն ունի անվավեր հարցումներից սահմանափակում, որից հետո կատարվում է բանալու վերահաշվարկ, այս

տարբերակով բանալու դուրս բերումը կարելի է համարել գրեթե անհնար: Հարցման մեջ առկա տվյալները առանց փակ բանալու իմացության չեն տրամադրում տեղեկատվություն ուղարկվող հրամանների մասին և ունեն պահանջված գաղտնակայունություն, մասնավորապես՝ VH հեշը ունի 2^n բարդության աստիճան: R և G մատրիցի հնարավոր տարբերակների մոտավոր քանակն հավասար է համապատասխանաբար 2^{512}-ի և 2^{53}-ի: Այստեղից կարելի է եզրակացնել, որ հատարկման տարբերակով փակ բանալու բացահայտման համար անհրաժեշտ առավելագույն քանակը հավասար է 2^{565}:

Փուլ 1

```
0  10 4  14 31 46 63 53
47 62 52 58 48 33 16 1
11 5  15 21 6  23 38 55
61 51 57 40 25 8  2  17
32 42 59 49 43 60 50 56
41 26 36 30 20 3  9  24
18 35 45 39 54 37 27 12
22 7  13 28 34 19 29 44
```

Փուլ 2

```
53 1  55 17 56 24 12 44
63 16 38 2  50 9  27 29
46 33 23 8  60 3  37 19
31 48 6  25 43 20 54 34
14 58 21 40 49 30 39 28
4  52 15 57 59 36 45 13
10 62 5  51 42 26 35 7
0  47 11 61 32 41 18 22
```

Փուլ 3

```
22 18 41 32 61 11 47 0
7  35 26 42 51 5  62 10
13 45 36 59 57 15 52 4
28 39 30 49 40 21 58 14
34 54 20 43 25 6  48 31
19 37 3  60 8  23 33 46
29 27 9  50 2  38 16 63
44 12 24 56 17 55 1  53
```

Փուլ 4

```
44 29 19 34 28 13 7  22
12 27 37 54 39 45 35 18
24 9  3  20 30 36 26 41
56 50 60 43 49 59 42 32
17 2  8  25 40 57 51 61
55 38 23 6  21 15 5  11
1  16 33 48 58 52 62 47
53 63 46 31 14 4  10 0
```

Փուլ 5

```
44 12 24 56 17 55 1  53
29 27 9  50 2  38 16 63
19 37 3  60 8  23 33 46
34 54 20 43 25 6  48 31
28 39 30 49 40 21 58 14
13 45 36 59 57 15 52 4
7  35 26 42 51 5  62 10
22 18 41 32 61 11 47 0
```

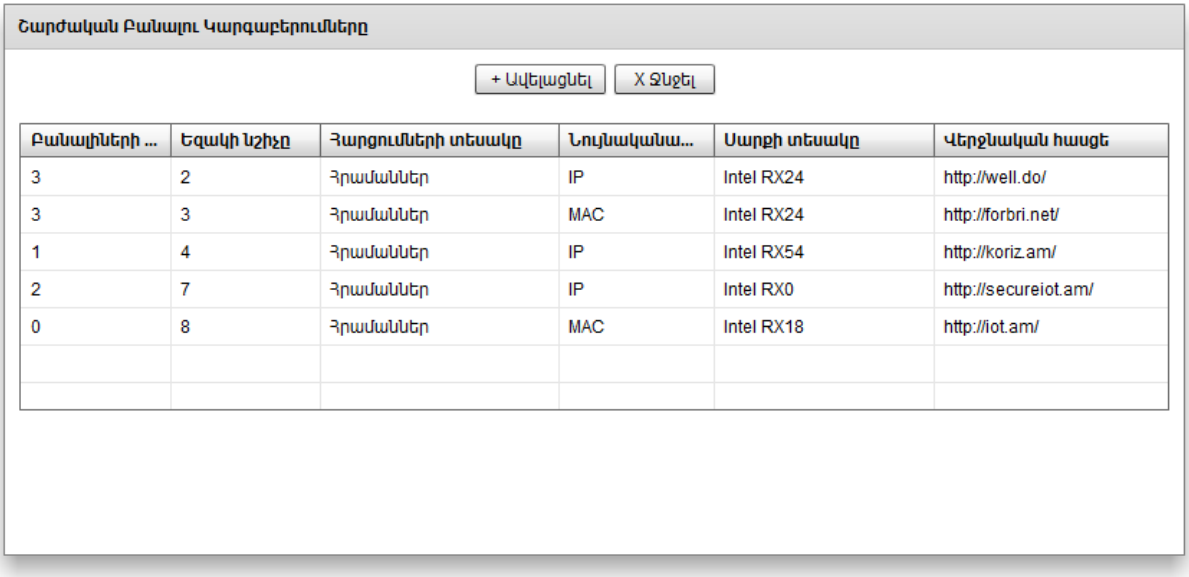
Նկ.11. Փուլերի ընթացքում R մատրիցի ձևափոխման օրինակը

Աղյուսակ 7. Առաջարկված գաղտնահամակարգի արագագործության համեմատականը

	Կատարման քանակը	Ժամանակը միլիվայրկյաններով
AES 128	1000	2821
RSA 1024	1000	15321
CMP SHA 256	1000	287
CMP SHA 384	1000	720
CMP SHA 512	1000	850

Եթե հարցումը պարունակում է այնպիսի տվյալներ, որոնք հրամաններ չեն, ապա իրականացվում է համապատասխան վերահասցեավորում՝ օգտագործելով HTTPS արձանագրությունը, որի կապի հաստատման գործառույթի (handshake) ընթացքում որպես գաղտնագրման ալգորիթմ ընտրվում է AES: Շարժական բանալին, ստանալով անլար ցանցին միացած սարքավորումներից հարցումներ, նախքան վերահասցեավորումը ստուգում է այդ հարցումների վերջնական հասցեներին համապատասխան կարգաբերումների գոյությունը: Եթե այդպիսի կարգաբերումներ գոյություն չունեն, հարցումն ուղարկվում է առանց փոփոխության[44]:

HTTPS արձանագրության մեջ օգտագործվող AES ալգորիթմի արագացման համար օգտագործվել են Intel AES-NI լրացումները, որը թողարկվել է հատուկ Intel ընկերության պրոցեսորների, մասնավորապես՝ շարժական բանալու սարքի համար: Շնորհիվ այդ լրացումների հնարավոր է AES ալգորիթմում օգտագործվող հաշվողական մի քանի հրամաններ կատարել՝ օգտագործելով ընդամենը մեկ հատուկ հրաման: Հատուկ հրամաններից են ՝ AESENC(AES Encrypt Round), AESENCLAST (AES Encrypt Last Round), AESDEC (AES Decrypt Round), AESDECLAST(AES Decrypt Last Round), AESKEYGENASSIST (AES Key Generation Assist), AESIMC(AES Inverse Mix Columns):



Նկ. 12. Շարժական բանալիում առկա կարգաբերումների օրինակ

Քանի որ մշակված գաղտնահամակարգը ներդրվելու է ցանցում գտնվող լրացուցիչ սարքում, կատարվել է գոյություն ունեցող սարքերի հետազոտում՝ հնարավոր սահմանափակումները հայտնաբերելու նպատակով: Հաշվի առնելով այդ հետազոտության արդյունքները և վերոհիշյալ պահանջները, ստեղծվել են շարժական բանալու անհրաժեշտ բնութագրերը, որոնք են՝ բարձր արտադրողականությունը, անլար ցանց տրամադրելու հնարավորությունը և լրացուցիչ ծրագրային ապահովման ներդրման հնարավորությունը:

Վերոհիշյալ պահանջներին համապատասխանում են Intel-ի կողմից արտադրվող և համեմատաբար բարձր հաշվողական ռեսուրսներ ունեցող ինտերնետ իրերը: Տվյալ սարքերը տրամադրում են համապատասխան միջավայր՝ տեխնիկական փոփոխություններ և լրացումներ կատարելու համար: Գոյություն ունեցող հավելվածների միջոցով հնարավոր է հեշտությամբ փոխել այդ սարքերի կարգավորումները և կառավարել բոլոր ծրագրային գործընթացները: Տվյալ սարքերում հնարավոր է տեղադրել Ubilinux, որը հնարավորություն է տալիս առանց բարդությունների ինտեգրել անլար ցանցային կապի համար նախատեսված ծրագրային փաթեթները:

2.7. Գլուխ 2-ի եզրակացություններ

- Հետազոտվել են տարատեսակ ինտերնետ իրեր՝ միմյանց հետ փոխազդելու եզակի գլոբալ հասցեավորման սխեմայի և ծառայությունների տրամադրման մեխանիզմների բացահայտման նպատակով:
- Հիմնվելով իրերի ինտերնետ միջավայրում իրերի բնութագրերի վրա՝ առաջարկվել է բանալիների կառավարման անվտանգ սխեմա՝ տարատեսակ իրերի միջև անլար կապի հաստատման համար:
- Ցույց է տրվել, որ բանալիների կառավարման առաջարկված սխեմայում յուրաքանչյուր հանգույց ընտրվում է որպես ծառայության հանգույց, այլընտրանքային ծառայության հանգույց կամ աշխատանքային հանգույց՝ հանգույցների ինքնակազմակերպման բնույթի շնորհիվ՝ ինքնակարգավորման մշակված ալգորիթմի միջոցով:
- Հիմնավորվել է, որ ընտրման գործընթացից անմիջապես հետո ծառայության հանգույցները գեներացնում են բանալիների ենթակառուցվածք՝ համապատասխան աշխատանքային հանգույցների համար: Արդյունքում աշխատանքային հանգույցների զույգերը հաշվարկում են ընդհանուր բանալին:
- Գնահատվել է մեթոդի պիտանիությունը՝ հիշողության, հաղորդակցման, հաշվողական գերբեռնվածության և հարձակումների նկատմամբ կայունության առումներով:
- Մշակվել է գաղտնագրման նոր համակարգ պարզ ինտերնետ իրերի համար: Տվյալ համակարգը, ստուգելով հարցումներում առկա տվյալները, գործածում է, կամ գաղտնագրային կամ արտապատկերման միջոցով հրամանների նույնացման գործառույթը: Առաջարկած գաղտնահամակարգն աշխատում է պարզ ինտերնետ իրերից անկախ. այսինքն՝ չի պահանջում գոյություն ունեցող սարքի ապարատային կամ ծրագրային փոփոխություն:

ԳԼՈՒԽ 3

ԲԱՇԽՎԱԾ ՑԱՆՑԵՐՈՒՄ ԳՈՐԾԱՌՈՒՅԹՆԵՐԻ ԱՄԲՈՂՋԱԿԱՆՈՒԹՅԱՆ ՈՒ ԱՆՎՏԱՆԳՈՒԹՅԱՆ ԱՊԱՀՈՎՄԱՆ ՄԵԹՈԴԻ ՄՇԱԿՈՒՄ

3. 1. Գործառույթների ամբողջականությունն ապահովող արձանագրության մշակում

Իրերի ինտերնետ միջավայրը տարատեսակ իրերից բաղկացած բաշխված ցանց է, որի կողմից տրամադրվող գործառույթը/ծառայությունն իր հերթին բաղկացած է տարբեր իրերի կողմից կատարած գործառույթներից: Վերոնշյալ բաշխված ցանցի կառավարումը, գործառույթների հաջորդականության վերահսկումը և գործառույթների ամբողջականության ապահովումը կարևոր խնդիր է :

Հաշվի առնելով վերոգրյալը՝ առաջարկվում է իրերի ինտերնետի մոդելը կառուցել ուղղորդված ծառի տեսքով, որտեղ հանգույցներն ինքնակազմակերպվող են, իսկ հրահանգների և տվյալների փոխանակումը հանգույցների միջև վերահսկվում է տեղեկատվության անվտանգության և ամբողջականության մեխանիզմների միջոցով:

Այդ նպատակով միջավայրի գործառույթների ավտոմատացման համար առաջարկվում է ներմուծել հանգույցներին հաղորդվող հաղորդագրությունների՝ աղյուսակ 8-ում բերված ձևաչափը, որը ներկայացնում է նաև յուրաքանչյուր հանգույցին ուղղված գաղտնիքի (մյուսների համար անվերծանելի) մասնաբաժինը:

Աղյուսակ 8. Հաղորդագրության ձևաչափը

Առանձին հանգույցների գաղտնագրված տվյալները					Գլխամաս
E_1	E_2	E_3	E_4	E_5	Hashcas, N_1 -N_m

Առաջարկվող մոդելն օգտագործում է իրերի ինտերնետ միջավայրի համար մշակված արձանագրությունը և ձևաչափը, որը հնարավորություն է տալիս իրագործել՝ գործառույթների կառավարումը, տվյալների գաղտնագրումը, սխալների հայտնաբերումը և գործառույթների չեղարկումը:

Գործառույթների կառավարումը կատարվում է ձևաչափում նկարագրված հաջորդականության շնորհիվ, որը M չափի զանգված է բաղկացած 168 բայթ երկարություն ունեցող N_i տեղեկատվությունից (աղյուսակ 9), որտեղ առաջին 128 բայթը եզակի նշիչ (UID) է կամ IP6 հասցե: Հասցեին հաջորդող 32 բայթ երկարությամբ L տվյալը ցույց է տալիս ձևաչափի մարմնում գտնվող գաղտնագրված տեղեկատվության չափը: Վերջին 8 բայթը՝ D-ն, ցույց է տալիս համապատասխան ինտերնետ իրի գործառույթի խորությունը գործառույթների ծառում:

Աղյուսակ 9. Հաղորդագրության գլխամասի ձևաչափը

N_1 գործառույթ			N_2 գործառույթ			N_m գործառույթ		
UID| IP6	L	D	UID | IP6	L	D	UID | IP6	L	D
128բ	32բ	8բ	128բ	32բ	8բ	128բ	32բ	8բ

Տվյալների գաղտնագրումը կատարվում է 1-ից մինչև N քանակի բանալիներով, որտեղ N պարամետրը փոքր կամ հավասար է ցանցում առկա ինտերնետ իրերի թվին: Շնորհիվ տարբեր բանալիների կիրառության՝ հնարավոր է ապահովել առանձին ինտերնետ իրերին պատկանող գաղտնագրված ինֆորմացիայի անվտանգությունը ցանցում՝ նույնիսկ այլ ինտերնետ իրերի խոցելիության դեպքում: Գաղտնագրման բանալու ստեղծման համար օգտագործվում է հետևալ բանաձևը՝ K = H * Ki, որտեղ Ki-ն ինտերնետ իրում նախաբեռնված բանալին է, իսկ H պարամետրը՝ տվյալ ինտերնետ իրին հարցում ուղարկած հանգույցին պատկանող բաց տվյալների հեշ արժեքը, որը ստացվում է կատարած գործառույթի ելքային արժեքը օգտագործելով որպես հեշավորման բանալի: Շնորհիվ L պարամետրի՝ հաղորդագրության ձևաչափի մարմնում կարելի է գտնել կոնկրետ ինտերնետ իրի հաղորդագրության սկզբնական և վերջնական արժեքները հետևյալ բանաձևով:

$$L_{start} = L_0 + L_1 + \cdots + L_{i-1}, \quad L_{end} = L_{start} + L_{end}, \qquad D(N_i) = L_{start, \ldots,} L_{end} \qquad \textbf{(3.1)}$$

Հարցում ուղարկող հանգույցը գտնվում է D–1 կամ D +1 խորության վրա, որտեղ D-ն տվյալ հանգույցի խորությունն է: Նկարագրած ձևաչափի կիրառումը հնարավորություն է տալիս ապահովել ինտերնետ իրերին պատկանող տեղեկատվական անվտանգությունն այլ ինտերնետ իրերից:

Սխալների հայտնաբերումը և գործառույթների չեղարկումը կատարվում է՝ հիմնվելով գաղտնագրված տեղեկատվության վերծանման արդյունքի վրա: Ինչպես նշվել է, ինտերնետ իրերին պատկանող տեղեկատվության վերծանման համար անհրաժեշտ է D–1 ինտերնետ իրերին պատկանող բաց տվյալի հեշը և տվյալ ինտերնետ իրի բանալին: Կարևոր է նշել, որ հեշը ստացվում է կատարած գործառույթի ելքային արժեքը օգտագործելով, որպես բանալի: Հեշավորման բացակայության, անհաջող վավերականացման կամ սխալ վերծանման դեպքում ինտերնետ իրն սկսում է

գործառույթի չեղարկման գործնթացը, որտեղ չեղարկման հարցումները կատարվում են հակառակ ուղղությամբ:

Այսպիսով, ներկայացրած արձանագրության շնորհիվ, հարցումից կախված, ինֆորմացիան փոխանցվում է ցանցի հանգույցին, որը.

- իրեն հասանելի մասնաբաժնի համար իրականացնում է վերծանում և դրան համապատասխան գործառույթ,
- ձևավորում է վերծանված մասնաբաժնի հեշը, որը հավելվում է վերոհիշյալ հաղորդագրության մեջ՝ գաղտնիքի մասնաբաժնի ամբողջականության վերահսկման նպատակով:

Քանի որ ձևաչափը տեղեկություն է պարունակում տվյալ հանգույցից դեպի այլ՝ տրամաբանորեն հաջորդ հանգույց տանող ճանապարհի վերաբերյալ, որը փաստորեն ենթադրում է ցատկ դեպի ծառի ավելի ստորին մակարդակում/մակարդակներում տեղակայված մեկ կամ մի քանի հանգույցներ, ապա ստորին մակարդակի ամեն հանգույց՝ ստանալով նշված ձևաչափի հաղորդագրությունը, նախ՝ փորձում է իր կարգային համարը գտնել այդ հաղորդագրության մեջ, որից հետո վերծանելով գաղտնիքի իր մասնաբաժինը և արտադրելով համապատասխան հեշը, կառավարումը փոխանցում է հաջորդ հանգույցին: Արդյունքում բոլոր անհրաժեշտ հանգույցների կողմից մշակված և արտադրված հեշ արժեքների համախումբը վերահսկում է ցանցով փոխանակվող տվյալների և գործառույթների ամբողջականությունը, իսկ բաշխվող բանալիների ենթակառուցվածքը՝ այդ նույն տվյալների և գործառույթների անվտանգությունը:

Հարկ է նշել, որ ցանցով տվյալների և հրահանգների փոխանակման առաջարկվող գործընթացը հաջորդական է և ենթադրում է սեսսիոն բանալիների ենթակառուցվածք,

որը բաշխում է սեսսիոն բանալիները՝ ծառի մակարդակներին համապատասխան:

Այն դեպքում, երբ հանգույցը, որին փոխանցվել է վերոհիշյալ ձևաչափի հաղորդագրությունը և, որն իր կարգային համարը չի հայտնաբերում այդ հաղորդագրության մեջ, ուստի ազատ է գաղտնիքի որևէ մասնաբաժին վերծանելուց, կառավարումը փոխանցում է հաջորդ հանգույցին՝ ապահովելով միայն հեշավորումը:

Հեշ արժեքների ստուգման անհաջողության դեպքում տվյալ հանգույցի գործառույթի ամբողջ շղթան, չեղարկման հետ, ազդանշան է ուղարկում նույն հաջորդականությամբ:

Հաջողության դեպքում հանգույցը որոշում է ստացված հաղորդագրությունը փոխանցել անհրաժեշտ հանգույցին՝ ըստ UID-ի: Ամեն հաջորդ հանգույցում ձևավորված հեշերի շղթան երաշխիք է՝ ցանցի գործողությունների վավեր լինելը հավաստիացնելու համար:

Այսպիսով, առաջարկվել են իրերի ինտերնետ միջավայրի համար մշակված արձանագրությունը, ձևաչափը, որն ունակ է տեղեկություն հաղորդել այլ հանգույցների՝ տվյալ հանգույցում անոմալիաների իրազեկման համար և արգելակել ամբողջ համակարգի աշխատանքը՝ թույլ օղակի գործոնով բացառել համակարգի հետագա գործողությունը: Առաջարկված մեթոդը, որպես հեշային ֆունկցիա կարող է օգտագործել ցանկացած գաղտնակայուն հեշ ֆունկցիաներից մեկը, բայց հաշվի առնելով ինտերնետ իրերի առանձնահատկությունների փաստը, առաջանում է գոյություն ունեցող հեշ ֆունկցիաները հետազոտելու և տվյալ մեթոդին համապատասխան հեշ ֆունկցիա գտնելու անհրաժեշտություն:

3.2. Հեշ և POW ֆունկցիաների հետազոտում

Տեղեկատվության անվտանգության ապահովման համար կարևոր նշանակություն ունի հեշ ֆունկցիաների կիրառումը: Տվյալ ոլորտում առաջարկված լավագույն լուծումներից կարելի է առանձնացնել քաոսային տեսության վրա հիմնված հեշ ֆունկցիաները: Այդ ֆունկցիաներն օգտագործում են միաչափ արտապատկերումներ, ինչպիսիք են լոգիստիկային և տենտային արտապատկերումները, կամ կիրառում են բարդ բազմաչափ արտապատկերումներ, որոնք, որպես կանոն, անվտանգ չեն և հիմնականում հեշտությամբ են գրոհի ենթարկվում [45]:

Հեշ ֆունկցիան համարվում է միակողմանի, եթե M կամայական երկարության հաղորդագրությունը փոխակերպվում է ոչ գծային հաշվարկների միջոցով և արդյունքում ստեղծում է h(M) հեշավորումը, որն էլ կարող է ունենալ հաստատուն կամ փոփոխական երկարություն: Այս ֆունկցիաները կարող են դասակարգվել երկու դասերի՝ գաղտնագրային և ոչ գաղտնագրային: Վերոհիշյալ ոչ գաղտնագրային հեշ ֆունկցիաներն օգտագործվում են հիշողության և պահեստավորված տվյալների ինդեքսավորման համար:

Գաղտնագրային հեշ ֆունկցիաների կարևոր հատկություններից են բախումը և կայունությունը: Հեշ ֆունկցիայի բախումը երկու տարբեր հաղորդագրության (M1 և M2) հեշերի արժեքների համընկնումն է, երբ h (M1) = h (M2): Հեշ ֆունկցիայի բախումների բացահայտումն իրականացվում է պատահական մուտքային արժեքների փորձարկման եղանակով [46, 47, 48]: Կարևոր է նշել, եթե հեշավորված տվյալի իմացությամբ հարձակվողը կարող է գտնել M հաղորդագրությունը, որն արտահայտվում է h (M) = H բանաձևով, ապա հեշավորման ֆունկցիան կայունություն ապահովող չի համարվում: Այսպիսով, նախօրոք տրված ելքի համար հաշվողական առումով անհնար է հաշվարկել այն մուտքը, որն արտադրում է միևնույն ելքը: Նաև կայունություն ցածր ցուցանիշի

արդյունք է համարվում հեշ ֆունկցիայի մուտքային և ելքային արժեքների իմացության դեպքում մեկ այլ մուտքային արժեք գտնելու հնարավորությունը, որի ելքային հեշը կհամընկնի սկզբնական հաղորդագրության ելքային հեշի հետ:

Հեշ ֆունկցիան կարող է դասակարգվել բանալիով կամ առանց բանալու հեշավորման ֆունկցիաների: Բանալիով հեշ ֆունկցիան ունի լրացուցիչ մուտք, որն ընդունված է նշանակել k պարամետրով: Արդյունքում առանց k-ի իմացության hk(M) հեշավորման աշխատանքն անհնար է իրականացնել: Ինչպես նաև հաղորդագրության և ելքային հեշի իմացության պարագայում բանալու որոնումը հաշվողական առումով նույնպես անհնար է:

Հեշավորման ֆունկցիաների մեծամասնությունը, ինչպիսիք են՝ MD4, MD5, MD6, SHA1 և SHA2 հեշ ֆունկցիաները, օգտագործում են Merkle–Damgård սխեման և բիթային գործողություններ կատարում մուտքային հաղորդագրության վրա՝ վերջնական հեշ արժեքը ստանալու նպատակով [47, 49, 50]:

Վանգը և Յուն ցույց են տվել, որ Merkle–Damgård սխեման անվտանգ չէ, քանի որ վերոհիշյալ պահանջների կատարումը չի ապահովում [50]: Հեշ ֆունկցիան նաև պետք է ունենա Շենոնի շփոթության և դիֆուզիայի կանոններին համապատասխան հեշավորման դրսևորում [51]: Շփոթությունն ապահովվում է այն դեպքում, երբ հաղորդագրության հեշավորած ելքի ցանկացած բիթ ստացվում է բանալու տարբեր մասերի օգնությամբ, ինչը դժվարացնում է բանալու և հաղորդագրության կապի բացահայտումը: Միաժամանակ հեշավորած ելքի ցանկացած բիթ փոխելու դեպքում պետք է փոխվի հաղորդագրության բիթերի կեսը և պահպանվի հաղորդագրության և հեշ ֆունկցիայի համապատասխանությունը:

Քաոսային մեթոդները բավարարում են վերոհիշյալ կանոններին, սակայն քաոսային հեշ ֆունկցիաների մեծամասնությունը որպես ոչ անվտանգ սխեմա օգտագործում է Merkle–Damgård կամ այդ սխեմայի տարբերակը:

Վերոհիշյալ դրույթների հետ մեկտեղ, բազմաչափ կամ կիսապատահական քաոսային արտապատկերումները նվազեցնում են դրանց կատարողականության ցուցանիշները մասնավորապես՝ արագությունը և անվտանգությունը: Այս ֆունկցիաների վրա հաջող գրոհներ են իրականացվել վերջիններիս թույլ գաղտնագրային բնույթի պատճառով: Վերոնշյալ հատկություններով հեշ ֆունկցիան կիրառվում է որպես գաղտնագրային արձանագրությունների, անվտանգ գործարքների և կրիպտոարժույթների հիմնական տարր:

Որոշ կրիպտոարժույթներ օգտագործում են Sha256 HashCash ֆունկցիան, որպես՝ այսպես կոչված «աշխատանքի ապացույց», որն ապահովում է հանգույցների կատարած գործառույթների անվտանգությունը: HashCash-ն օգտագործվում է նաև սպամերի կանխարգելման համար՝ ստիպելով գրոհող կողմին ծախսել որոշակի ժամանակ ցանկացած հաղորդագրության և էլեկտրոնային նամակին կից ուղարկվող հեշի ստեղծման հաշվարկների վրա: Եթե հաղորդագրությունը կամ էլեկտրոնային նամակը ստացվում է առանց հեշի, կամ ստացված հեշը վավերականացում չի ացնում, ապա այն մերժվում է և համարվում՝ սպամ: HashCash-ը Proof-of-Work (POW) ֆունկցիայի իրականացումն է, որը հայտնի է նաև, որպես աշխատանքային ֆունկցիա, հետաձգման ֆունկցիա կամ գլուխկոտրուկներ, առաջին անգամ առաջարկվել է Dwork և Noar կողմից սպամերի դեմ պայքարի համար: Այս ֆունկիցայի գաղափարը այն է, որ նախքան հարցում ուղարկելը այդ հարցմանը պետք է կցել լուծված որոշակի հաշվողական խնդիր, որն այնուհետև հնարավոր է վավերացնել հարցումն ընդունող կողմում: Այսպիսով, սահմանափակ հաշվողական ռեսուրսների պատճառով վերոհիշյալ հարցումների ընդհանուր քանակը, որը կարող է

ուղարկել մեկ համակարգիչը, բավականին նվազում է և դառնում սահմանափակ: Այս լուծումն օգտագործվում է էլեկտրոնային փոստով ուղարկվող սպամերի քանակը նվազեցնելու համար:

Գոյություն ունեն տարբեր POW ֆունկցիաներ, և անհրաժեշտ է դիտարկել ստորև թվարկված այդ ֆունկցիաների ընդհանուր բնութագրերը.

- POW ֆունկցիաները պետք է լուծեն հաշվողական խնդիրներ, որոնց լուծման համար պահանջում են որոշակի քանակությամբ ռեսուրսներ:
- Պետք է բացառվի որևէ լուծման գոյությունը, որը թույլ կտա գեներացնել արդյունքներն ավելի արագ առանց հստակ որոշված հաշվարկների իրականացման:
- Կատարած հաշվարկների լուծման արդյունքը պետք է հնարավոր լինի հեշտությամբ և արագ վավերացնել, քանի որ այն կատարվելու է սերվերային միջավայրում, որտեղ վավերացում պահանջող հարցումների քանակը կարող է բավականին շատ լինել:
- POW ֆունկցիայի լուծման արդյունքում առաջացած ելքային տվյալները պետք է ունենան փոքր ծավալ, քանի որ այն պետք է փոխանցվի համացանցով այլ վերջնակետ, որտեղ կկատարվի ստուգումը:
- POW ֆունկցիայի լուծման դժվարությունները պետք է լինեն հնարավորինս կարգավորվող:

POW ֆունկցիաների առավել տարածված տեսակն օգտագործում է պրոցեսորի հաշվողական ռեսուրսները: Բացի այդ, առաջարկվել են նաև POW այլ ֆունկցիաներ, ինչպիսիք են հիշողության ֆունկցիաները: Այս տեսակի ֆունկցիաները ավելի դժվար է իրացնել, քանի որ հիշողության ծանրաբեռնման համար անհրաժեշտ ժամանակը ավելի շատ է, քան CPU ծանրաբեռնման ժամանակը: Նաև հիշողության ֆունկցիաների գործածումը բարդություն է ստեղծում ցածր պարամետրերով համակարգիչների կիրառման

դեպքում: Հաշվի առնելով վերոհիշյալ թերությունները՝ ատենախոսությունում նախընտրությունը տրվել է պրոցեսորի հետ աշխատող POW ֆունկցիաներին, որն այսուհետև կնշվի որպես դասական POW ֆունկցիաներ:

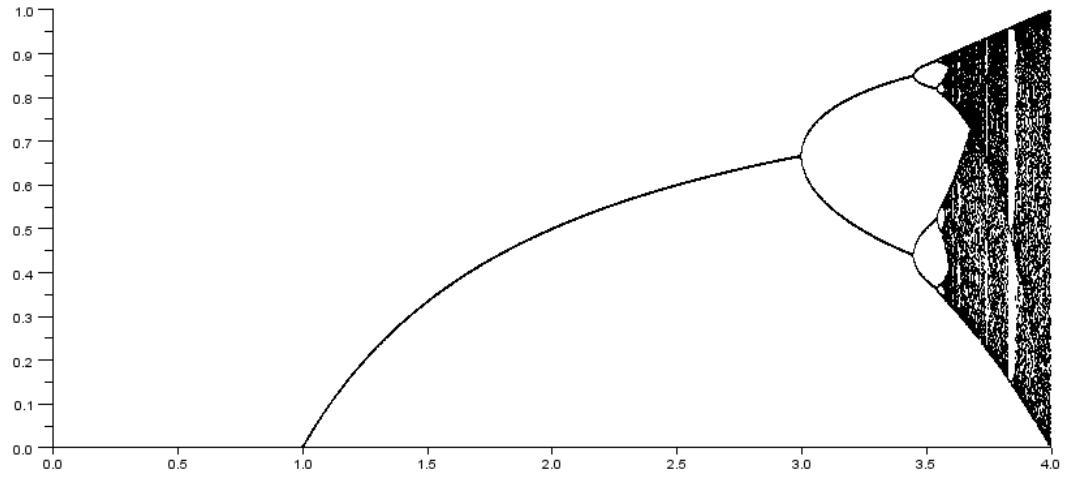
Դասական POW ֆունկցիաների լուծման խնդիրները հիմնված են NP որոնողական խնդիրների վրա, երբ խնդրի լուծման համար պետք է կատարվի որոնում, մինչդեռ սերվերը պետք է ստուգի միայն մեկ առաջարկված լուծում: Սովորաբար վերոնշյալ խնդիրների բնույթից ելնելով լուծման համար անհրաժեշտ հաշվողական գործառույթըկարող է իրականացվել միայն վիճակագրականորեն:

Կրիպտոարժութների գործառույթները կատարվում են ցանցում առկա հանգույցների հաստատմամբ, որը կատարվում է ստացված տվյալների հեշավորման և ցանցի մյուս հանգույցներին ուղարկելու միջոցով [52]: Հեշավորման հզորությունը, որը չափվում է GH/s-ով, օգտագործվում է կրիպտոարժութային ցանցի հեշավորման արագության չափման համար: Արագագործ ցանցերում ապահովվում է առավել արագ գործառույթների իրականցում, որը պայմանավորված է մասնակցող հանգույցների թվով և արագ հաշվարկմամբ: Վերոհիշյալ հեշ ֆունկցիայի հիմնական թերություններն են՝ բանալու բացակայությունը և զուգահեռացման իրականացման բարդությունը: Ստեղծվել են տարբեր Sha256-ի իրականացումներ զուգահեռացման ապահովման նպատակով, բայց արդյունքում միայն հաջողվել է տվյալները բաժանել մի քանի մասի և հեշավորել այդ մասերը՝ գործածելով ամեն մասի համար Sha256 ծրագրի առանձին օրինակ:

Իրականացման պարզության և միաչափ քաոսային արտապատկերումների բարդ վարքագիծի շնորհիվ քաոսային հեշ ֆունկցիաները լայն տարածում են գտել: Լոգիստիկ արտապատկերումը նկարագրած արտապատկերումներից մեկն է, որը սահմանվում է ստորև բերված բանաձևով:

$$X_{n+1} = L_r(X_n) = r \cdot X_n \cdot (1 - X_n) \qquad (3.2)$$

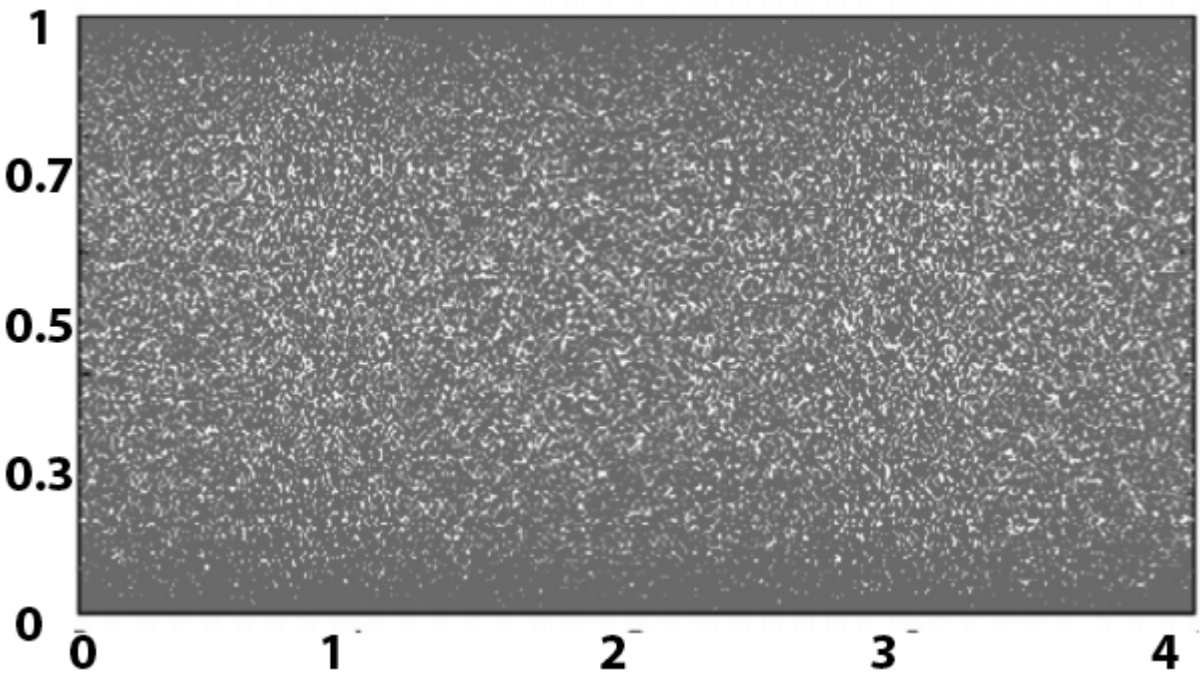
Չնայած այս արտապատկերումն ունի (0, 4] միջակայք r պարամետրի համար, այն քաոսային վարքագիծ է դրսևորում միայն [3.57, 4] միջակայքում, և նույնիսկ այս միջակայքում արտապատկերումը որոշ չափանիշների դեպքում կորցնում է իր քաոսային վարքագիծը: Նկ. 13-ում ցույց է տրված երկատման դիագրամը:



Նկ. 13. Լոգիստական արտապատկերման երկատման դիագրամը

Միաչափ քաոսային արտապատկերումների այլ օրինակներ են սինուսային արտապատկերումները, որոնք լոգիստական արտապատկերման հետ ունեն միանման դրսևորում, մասնավորապես սինուսային արտապատկերումն իր քաոսային վարքագծում ունի լոգիստակայինի արտապատկերմանը նման երկճյուղավորման դիագրամ և միևնույն խնդիրները: Այս խնդիրները լուծելու համար Զհոուն և մյուսներն առաջարկել են քաոսային համակարգ՝ երկչափ արտապատկերումներով, որոնք ընդունված է նշանակել LS պարամետրով [53,54]: Գաղտնագրային համակարգերում ցանկացած ոչ հավասարաչափ բաշխում պատահական թվերի գեներատորով կարող է օգտագործվել գրոհների կողմից ծածկագրի կամ ալգորիթմի բացահայտման համար՝ օգտագործելով դիֆերենցիալ գրոհները:

Բետա բաշխումների համար, որտեղ α=β-ի և առավելապես բաշխված արժեքները գտնվում են [0, 0.1) և (0.9, 1] միջակայքում, հեշտ է գրոհման ենթարկել առաջարկված մեթոդը: LS արտապատկերման մեկ այլ թերությունն այն է, որ սինուսոյիդային արտապատկերումից ելքի ստացումը լոգիստական արտապատկերման համեմատ, կատարվում է ավելի դանդաղ:



Նկ. 14. LS արտապատկերման երկատման դիագրամը

Մեկ այլ լուծում ներկայացվել է Սան Յում և Սրիչավենգսափի կողմից [54]: Այն հիմնված է սինուսոիդային ոչ գծային քաոսային համակարգի ստեղծման վրա: Բազմապատկման, գումարման և հանման գործողությունները սովորաբար հեշտությամբ կատարում է պրոցեսորը ցածր մակարդակում, բայց եռանկյունաչափական գործողությունները, ինչպիսիք են sin և cos, չեն իրականացվում պրոցեսորի կողմից ուղղակի հրամաններով, այլ օգտագործում են գրադարաններ, որոնք իրենց հերթին օգտագործում են ցածր մակարդակի գործողություններ: Հեշ ֆունկցիաներում այսպիսի գործողությունների կիրառումը զգալիորեն նվազեցնում է հեշավորման արագությունը: Եթե

բազմապատկման գործողության համար անհրաժեշտ է 4 ժամանակային ցիկլ, ապա sin գործողության համար այն կարող է հասնել 90 – 100 ժամանակային ցիկլ:

Իրականացման պարզությունը, պատահական դրսևորումը, արագությունը և անվտանգությունը հեշ ֆունկցայի կարևոր բնութագրերից են, որոնց միաժամանակյա ապահովումը հեշ ֆունկցիայում բարդ խնդիր է:

Քաոսային հեշ ֆունկցիաները կարելի է դասակարգել տարբեր դասերի՝ կախված իրենց քաոսային սխեմայից, հաշվարկների զուգահեռացման կարողությունից և այլն: Բայց հաշվի առնելով, որ քաոսային հեշ ֆունկցիաները հիմնված են քաոսային արտապատկերման վրա, հեշտությամբ կարելի է դասակարգել [55, 56].

- արտապատկերման վրա հիմնված պարզ հեշ ֆունկցիաներ,
- արտապատկերման վրա հիմնված բարդ հեշ ֆունկցիաներ,
- նոր քաոսային համակարգի վրա հիմնված հեշ ֆունկցիաներ,
- զուգահեռացում ապահովող և բարդ կառուցվածք ունեցող քաոսային հեշ ֆունկցիաներ,
- քաոսային արտապատկերման միջոցով ձևափոխված դասական հեշ ֆունկցիաներ:

Արտապատկերումների վրա հիմնված պարզ քաոսային հեշ ֆունկցիաների մեծամասնությունը մյուս դասերի համեմատ օգտագործում է պարզ ալգորիթմ՝ միաչափ քաոսային արտապատկերումներով: Այս դասի որոշ հեշ ֆունցիաներում կիրառվում է տենտային արտապատկերում, որի ելքային շարքը ստացվում է Merkle–Damgård սխեմայով: Տենտային արտապատկերման խնդիրները, ինչպիսիք են ոչ քաոսային մուտքային միջակայքերը, նույնպես առկա են այս դասի հեշ ֆունկցիաներում: Գոյություն ունեն նաև տենտային արտապատկերումներ XOR սխեմայով: Ցանկացած հեշ ֆունկցիա, որն ունի 256 բիթից փոքր ելքային չափս, անվտանգ չէ: Վերոհիշյալ հեշ ֆունկցիան ունի 128 բիթ

երկարությամբ ելքային հեշ արժեք, որը հնարավոր է դարձնում գրոհողի համար գտնել հնարավոր բախումները կամ հաշվարկել արտապատկերումը 2^{64} հաշվողական բարդությամբ [57]:

Արտապատկերման վրա հիմնված բարդ հեշ ֆունկցիաներն օգտագործում են բազմաչափ քաոսային համակարգեր իրենց հեշավորման ալգորիթմում: Այս դասին պատկանող քաոսային ֆունկցիաները պահանջում են առավել բարձր արտադրողականություն, որը կապված է մատրիցների բազմապատկման և գումարման գործողությունների հետ, և մեծ արտապատկերումների դեպքում առաջանում են նաև հիշողության հետ կապված խնդիրներ:

Հեշ ֆունկցիաները, որոնք կիրառում են նոր քաուսային համակարգերը, ներկայացված են [54]- ում: Այն հեշավորման համար օգտագործում է նախորդ H_i արժեքը և հաղորդագրության i-րդ բլոկը: H_i արժեքը հաշվարկվում է յուրաքանչյուր փուլի վերջում հաջորդ իտերացիայի համար՝ օգտագործելով ոչ գծային փոխակերպում: Այս հեշ ֆունկցիան չի ապահովում բազմահոսքայնությունը, իսկ բանալին օգտագործվում է որպես թվային արժեք ASCII փոխարեն: Չնայած Merkle-Damgård փոփոխված սխեմային՝ ներկայացված հեշ ֆունկցիան չունի արագություն և համապատասխան անվտանգություն: Այս դասին է պատկանում նաև մեկ այլ օրինակ, որն օգտագործում է սինուսոյիդային ոչ գծայնությունը: Զուգահեռացման բացակայությունը, ցածր հեշավորման արագությունը և քաոսային համակարգի խնդիրները, որոնք արդեն նկարագրվել են, համարվում են այս մեթոդի հիմնական թերությունները: Համաձայն հեղինակների՝ 5,000 մուտքային հաղորդագրությունների համար հեշավորված ելքերի վրա հայտնաբերվել է միևնույն արժեքով 509 ASCII սիմվոլ, ինչը բավականին բարձր ցուցանիշ է և կասկածի տակ է դնում հեշ ֆունկցիայի անվտանգությունը:

Գոյություն ունեն նաև նեյրոնային ցանցերի հիման վրա աշխատող քաոսային բարդ կառույցներ: Քաոսային նեյրոնային ցանցն օգտագործում է քաոսային արտապատկերումը վերջնական արդյունքի ստացման համար: Խիաոյի կողմից առաջարկված հեշ ֆունկցիան, որն օգտագործում է երկշերտ քաոսային նեյրոնային ցանց, հեշտությամբ գրոհման է ենթարկվել [56]:

Լիի Դենգի կողմից առաջարկված ֆունկցիաների կառուցվածքն ավելի բարդ է Merkle–Damgård սխեմայի համեմատ՝ նպատակ ունենալով ապահովել ավելի մեծ անվտանգություն դրանց բարդ կառուցվածքի և քաոսային վարքագծի շնորհիվ: Նրանց կողմից առաջարկվել է զուգահեռ հեշավորման սխեման՝ ոչ գծային քաոսային արտապատկերման օգտագործմամբ: Այս սխեմայի վիճակագրական թեստերը և գաղտնագրային վերլուծությունը ցույց են տալիս բավական բարձր արդյունքներ տարբեր մուտքային տվյալների համար: Այս մեթոդով առանձին հոսքերը հաշվարկում են միջանկյալ Hi արժեքները հեշավորելով առանձին 1024 բիթային բլոկներ, որոնք հետագայում միավորվում են XOR-ով վերջնական H արժեքի ստացման համար: Ըստ Շենոնի՝ անվտանգության առումով առաջանում է թերություն կապված ֆունկցիայի մուտքի և վերջնական ելքի միջև [51]: Այս թերության պատճառը կապված է MITM գրոհման հետ, որը հարձակվողին թույլ է տալիս կոտրել հեշ ֆունկցիան, տրոհել մասերի և հաշվարկել միջանկյալ տվյալները առանց հաղորդագրությունների ամբողջական հաշվարկման անհրաժեշտության: Այսպիսով այս հեշ ֆունկցիան չունի բավարար զգայունություն մուտքային հաղորդագրության նկատմամբ և ունի բախումների բարձր ցուցանիշ: Գոյություն ունեն նաև այնպիսի հեշ ֆունկցիաներ, որոնք օգտագործում են պատահական փոխանակման ցանց, որն իր պատահական վարքագիծը ստանում է ոչ գծային քաոսային արտապատկերմամբ: Փոխանակման ցանցի նախամշակումից հետո կատարվում է եռաքայլ մշակում: Առաջին փուլում օգտագործվող քաոսային արտապատկերումը հեշ

ֆունկցիային թույլ է տալիս ապահովել բարձր կատարողականություն և գաղտնագրային վերլուծություն: Այս ֆունկցիան օգտագործելով չորս առանձին հոսքեր, ապահովում է բարձր հեշավորման արագությունը: Այն կարող է աշխատել ինչպես մեկ հոսքով, այնպես էլ մի քանի հոսքով՝ զուգահեռաբար: Անվտանգությունը և հեշավորման արագության պարամետրերի կարգավորման բացայկությունը համարվում է նշված մեթոդի հիմնական թերությունը:

Հեշ ֆունկցաիների վերաբերյալ կատարված այլ հետազոտություններում սովորական հեշ ֆունկցիաների անվտանգության բարելավման համար առաջարկվել է քաոսային արտապատկերումների օգտագործում: Վերոնշյալ հեշավորման ալգորիթմը SHA հեշ ֆունկցիայի փոփոխված տարբերակն է, որը նույնպես հեշտությամբ ենթարկվել է գրոհման, և համարվում է ոչ անվտանգ հեշավորման սխեմա [58]: Քաոսային հեշ ֆունկցիային վերաբերող աշխատությունները հիմնականում ուղղված են ելքային հեշ արժեքների վիճակագրական վերլուծության, գաղտնագրային վերլուծության և պատահական վարքագծի վրա, մինչդեռ հեշ ֆունկցիայի ծրագրային լուծումները, ինչպիսիք են կրիպտոարժույթները և սպամերի հայտնաբերումը, պահանջում են ավելի մեծ արագություն և անվտանգություն:

3.3. Ինտերնետ իրերի գործառույթների ամբողջականության վերահսկման հեշ ալգորիթմը

Հետազոտված քաոսային համակարգը (Դինամիկ քաոսային համակարգը) օգտագործում է α և β պարամետրերը [0, 1] միջակայքում: Այս պարամետրերը ստեղծում են քաոսային համակարգերի դասակարգ, որը նշանակվում է $C_{\alpha,\beta}$: Միաչափ քաոսային լոգիստիկ արտապատկերման համար այս համակարգը նշանակվում է $C_{\alpha,\beta}$ (w_α, w_β, r, X_n), որտեղ r-ը լոգիստիկ արտապատկերման մուտքային չափանիշն է, X_n-ը ուղեծիրն է, իսկ մյուս երկու չափանիշը՝ w_α և w_β α և β փոփոխականի կշիռն են: Առաջարկված համակարգն օգտագործում է մոդուլ 1, որպեսզի վերականգնի համակարգի ելքը դեպի (0, 1) միջակայք:

Հավասարում (3.3)-ը նկարագրում է այս համակարգի օրինակը՝ լոգիստիկ արտապատկերումը ընդունելով որպես հիմք:

$$X_{n+1} = CL_{\alpha,\beta}(w_\alpha, w_\beta, r, X_n, r) = \left(w_\alpha \alpha\left[L(r, X_n) + w_\beta \beta L(16 - r, X_n)\right] mod 1\right) = \left(w_\alpha \alpha\left[rX_n(1 - X_n) + w_\beta \beta(16 - r)X_n(1 - X_n)\right]\right) mod 1 \quad (3.3)$$

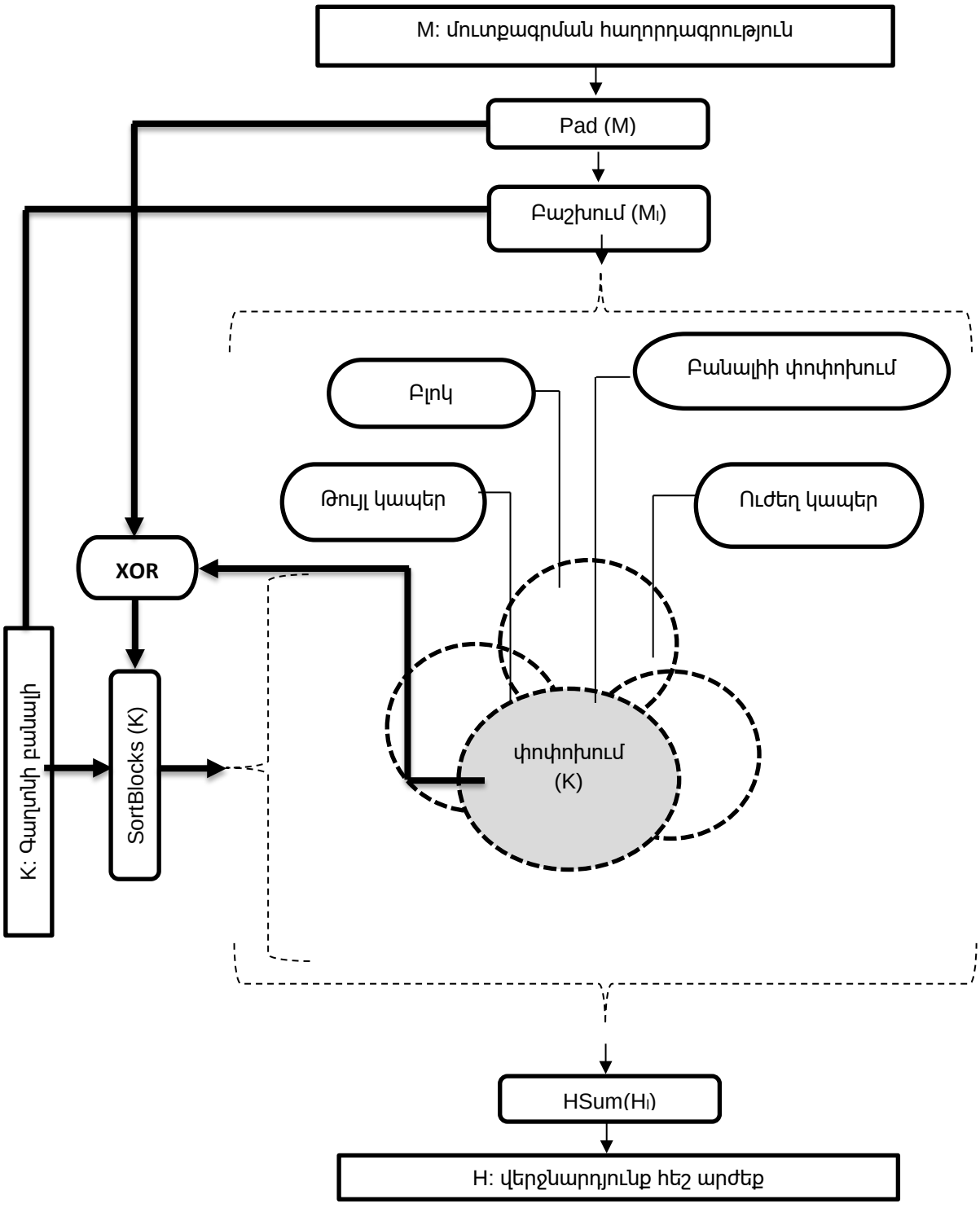
Առաջարկված համակարգը վեցը մեկին արտապատկերում է: w_α, α, w_β և β չափորոշիչների ֆիքսումը փոխում է այս արտապատկերը << երկուսը մեկին>> քաոսային արտապատկերման: Պարամետրերի ցանկացած համադրության ֆիքսումը կարող է փոխել այս արտապատկերումը՝ հաշվի առնելով $w_\alpha \times \alpha$ և $w_\beta \times \beta$ միջակայքերը: Տվյալ հեշավորման ալգորիթմը հավասարաչափ բաշխում ունի r և X_n բոլոր հնարավոր պարամետրերով:

Օպտիմալ բաշխումը գաղտնագրային պատահական գեներատորի համար համարվում է հավասարաչափ: Ելքային շարքերով ոչ հավասարաչափ բաշխումը հանգեցնում է նրան, որ հարձակվողն ունենում է պատահական թվերի գեներացման վիճակագրություն [59]:

Առաջարկված հեշ ֆունկցիայի մոդելում տվյալների ցանկացած 4 բայթ փոխակերպվում է 32 բիթային տասնորդական արժեքի, այսինքն դիտարկում են չմշակված այնպիսի տվյալներ, ինչպիսիք են տասնորդական արժեքների 32 բիթային բառերը:

Pad ֆունկցիան M-ը ընդունում է որպես մուտքային տվյալներ և բաժանում դրանք 32 բիթային բլոկների: Եթե վերջին բլոկն ունենում է 32-ից պակաս բիթ, սկզբում ավելացվում է '1' արժեքի բիթը, այնուհետև ավելացվում են '0' արժեքի բիթերը: Արտադրված բլոկները, որոնք նշանակվում են M_i- ով, իրենցից ներկայացնում են XOR՝ պարզ թվեր առաջին 32 բիթերով:

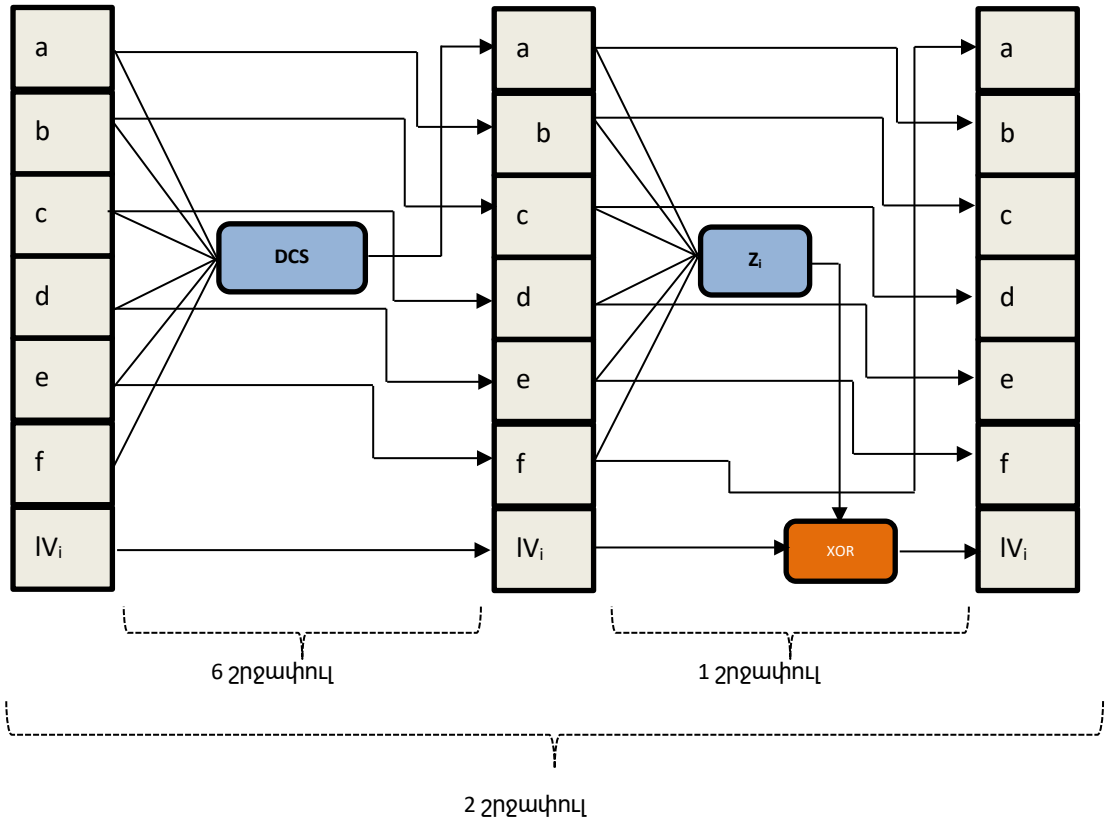
Վերոհիշյալ հեշ ֆունկցիայի բլոկ-սխեման ներկայացված է նկ. 15-ում: Գաղտնի բանալին, որը նշանակված է K-ով, ունի 256 բիթ երկարություն և ազդում է բոլոր ներքին ֆունկցիաների վրա, բացառությամբ Pad և Distribute: K-ի սկզբնական արժեքը իտերացիայի յուրաքանչյուր փուլում փոփոխվում է KeyMutate ֆունկցիայի օգնությամբ: Այս ֆունկցիան վերջին փուլի համար ընդունում է K և Distribute ելքային տվյալները՝ որպես մուտքային տվյալներ և արտադրում է նոր փուլային բանալի, որը նշանակվում է RKj –ով, որը ցույց է տալիս գաղտնի բանալին իտերացիայի j-րդ փուլի համար: SortBlocks ֆունկցիան ընդունում է RKj և XOR արժեքները: Այս ֆունկցիան գեներացնում է 8 յուրատեսակ թվային արժեքներ, որոնք գտնվում են [0, 7] միջակայքում: Այս արժեքները ցույց են տալիս ֆունկցիաների կարգավորվածությունը ֆունկցիաների զանգվածում: Ֆունկցիաների զանգվածը հիմնական գործառույթը ստացված թվային արժեքին համապատասխան ճիշտ ֆունկցիայի տրամադրումն է: Ֆունկցիաների դինամիկ զանգվածի համար մշակվել է 8 յուրատեսակ ֆունկցիա: Արժեքները, որոնք ներկայացված են աղյուսակում, a, b, c, d, e և f ցույց են տալիս համապատասխան M_i արժեքները յուրաքանչյուր f_i-ի համար ըստ իտերացիաների: Այս քաոսային փոխակերպումն ապահովում է փոխադարձ կապ բոլոր արժեքների միջև:



Նկ. 15. Հեշ ֆունկցիայի բլոկ-սխեման

Նկ. 16-ում ներկայացվել է f_i հաշվարկման սխեման: Իտերացիայի յուրաքանչյուր փուլի համար ցանկացած f_i ֆունկցիա ընդունում է RK_j և դրա հետ կապված՝ M_i արժեքները և հաշվարկում դրանց համապատասխան a, b, c, d, e և f պարամետրերը: Այս հաշվարկը զբաղեցնում է 6 ներքին փուլեր, իսկ հաշվարկի վերջում ևս մեկ փուլ՝ օգտագործելով Zi ֆունկցիաները, և տալիս է IV –ի համապատասխան արժեքը: Բոլոր միջանկյալ արժեքները և IV վեկտորը փոխադարձաբար ազդում են միմյանց վրա՝ օգտագործելով DCS և XOR գործողությունները: Բացի այդ, յուրաքանչյուր իտերացիայի վերջում ձևավորվում է կապ այս ֆունկցիաների արժեքների միջև, և արժեքները փոխանցվում են յուրաքանչյուր հարևանին: Թույլ կապեր են համարվում այն f արժեքները, որոնք XOR են կատարած երկու հարևանների արժեքներով, իսկ ուժեղ կապերը նշանակում են, որ եթե fi ֆունկցիայի f արժեքը մեծ է f_i+1 ֆունկցիայի f արժեքից, ապա դրա բոլոր արժեքների հանդեպ կատարվում է XOR գործողություն f_i+1-ի համապատասխան արժեքներով, հակառակ դեպքում այդ գործողությունը կատարվում է f_i և f_i-1 -ի միջև: Այս սխեմայում յուրաքանչյուր ֆունկցիայի համար օգտագործվում է երկու հարևան արժեքները և 8 փուլ, որոնց միջոցով կատարվում է ֆունկցիաների միջև արժեքների փոխանցում: Յուրաքանչյուր ֆունկցիա գործարկվում է առանձին հոսքի մեջ, իսկ ցանկացած փուլի վերջում արժեքները միավորվում են թույլ և ուժեղ կանոնների օգնությամբ:

Այս ընթացակարգը տևում է 8 փուլ, և հաշվարկման վերջում mutate ֆունկցիան ստեղծում է նոր փուլային բանալու վեկտոր: Յուրաքանչյուր իտերացիայի սկզբում RK_j փոխանցվում է SortBlocks, և այն փոխակերպվում է XOR-ի օգնությամբ՝ գտագործելով Mi բոլոր արժեքները, որոնք նախատեսված են SortBlocks ֆունկցիայի մուտքային տվյալների ստեղծման համար: Կատարվում է ֆունկցիաների նոր տեսակավորում, և սկսվում է հաջորդ իտերացիան: Բոլոր իտերացիաների վերջում HSum ֆունկցիան ընդունում է f_i ֆունկցիաների ելքային արժեքները և արտադրում է հեշի ելքային արժեքը:



Նկ. 16. f_i ֆունկցիաների կառուցվածքը

Հեշ ֆունկցիայի կատարողականությունը իրականացվել է ըստ վիճակագրական վերլուծության, արագության և գաղտնավերլուծության: B_i համարվում է երկու արժեքի միջև Հեմմինգի հեռավորությունը: γ և δ երկու տասնվեցերորդական արժեքի համար դրանք ունեն զրո B_i, եթե դրանք չունեն հավասար երկուական արժեք միևնույն դիրքում: Այս պարամետրը կարող է հաշվարկվել տարբեր արժեքների համար՝ հաշվարկելով դրանց բոլորի միջև Հեմմինգի հեռավորությունը (3.4 բանաձև)[60], որոնց միջին արժեքը կլինի Հեմմինգի միջին հեռավորության արժեքը (N իտերացիաների քանակն է)

$$\overline{B} = \frac{1}{N}\sum_{i=1}^{N} B_i \qquad (3.4)$$

Ելքային արժեքների տարբերության տոկոսային քանակը հաշվարկվում է հավասարում (3.5)-ում, որը ցույց է տալիս Հեմմինգի հեռավորության հավանականությունը:

Հեշ ֆունկցիայի համար այս հավանականությունը պետք է մոտ լինի 50%-ին: Հավասարում (3.5)-ում l-ը նշանակում է հեշավորման ելքի երկարությունը:

$$\boldsymbol{P = \left(\frac{\overline{B}}{l}\right) \times 100\%} \qquad \textbf{(3.5)}$$

Վիճակագրական վերլուծությունը: Փոքր փոփոխություններով հինգ տողերը S տողից և դրանց հետ կապված B_i-ը ընդդեմ S-ի, ցույց է տրվել աղյուսակ 10-ում: Օգտագործվել է ‘abcdefghijklmnopqrstuvwxyz 123456’ գաղտնի բանալին այս թեստի համար: Ներկայացվել են յուրաքանչյուր նախադասության համապատասխան հեշավորված արժեքները, իսկ աղյուսակ 11-ում տրվել են վիճակագրական վերլուծությունները 256, 512, 1024 և 2048 տողերի համար՝ 2000 երկարությամբ և 256 բիթային հեշ արժեքով: Այս տողերը գեներացվում են պատահականորեն:

Աղյուսակ 10. Համապատասխան հեշ արժեքները

i	Հեշ արժեք			
1	A 4 E 07782	80 AB 88 C 2	76466278	3 ACA 9 BB 4
	7 E 6E 0185	2 E 34 C 3 D 1	A 2C43 8EB	3 E 1 DF 802
2	FA1 B 0 B 8 F	9 D 7 DD 4 FC	7051 0BC 7	F 3266 B 08
	307 C 09 AC	9 A 605 F 4 A	10 A 63 B 67	B 28 A 0 F 3 F
3	2 E 637 FA 3	E 5313 F 67	B 73599 DE	BB 70 DA 6 D
	65 D 187 DD	E 2 DBD 26 A	4 DA 7 F 605	E 81 E 00 AA
4	A 14 B 02 C 5	1536 B 08 F	D 82734 FD	513 C 368 A
	2 DAB 9 ADE	F 8 EA 015 F	C 4 A 579 FA	764 CFF 01

Աղյուսակ 11. Փորձարկման վիճակագրական արդյունքները՝ օրինակների N քանակով

	256	**512**	**1024**	**2048**
B_{min}	109	107	104	101
B_{max}	161	163	165	168
$\overline{B}$	129.63	128.43	127.98	128.08
P(%)	50.63	50.17	49.99	50.03
ΔB	7.86	7.99	8.04	8.12

Գաղտնավերլուծության արդյունքները: Հեշ ֆունկցիաների նկատմամբ կիրառվել են տարբեր գաղտնագրային գրոհներ՝ ալգորիթմից անկախ և կախված: Ալգորիթմից անկախ հարձակման տեսակը չի պահանջում որևէ տեղեկատվություն հեշավորման կամ գաղտնագրային ալգորիթմի մասին, մինչդեռ, ալգորիթմից կախված հարձակումները պահանջում են լիարժեք կամ մասնակի տեղեկատվություն ալգորիթմի և դրա ներքին գործողությունների մասին: Անկախ հարձակումների օրինակներ են պատահական գրոհը, սպառիչ բանալու որոնումը և <<ծննդյան օրվա>> գրոհը: MITM և ֆիքսված կետի հարձակումը համարվում են ալգորիթմից կախված հարձակումների օրինակներ [61, 62]:

Պատահական գրոհը, օգտագործելով M հաղորդագրությունը, փորձում է գտնել բախումները՝ պատահականորեն փոփոխելով հաղորդագրության բիթերը կամ բլոկերը: Այս գրոհը որևէ գիտելիք չի պահանջում հեշ ֆունկցիայի մասին և կարող է կատարվել

հեշտությամբ: Կայունությունը այս գրոհի նկատմամբ պայմանավորված է նկարագրված ալգորիթմի պատահական և հիմնական ֆունկցիաների քաոսային վարքագծով:

«Ծննդյան օրվա» գրոհը հայտնաբերում է բախումները՝ հաշվարկելով բոլոր հնարավոր հեշ արժեքները: Այս հարձակման իրականացման համար պահանջվում է 2n/2 տարբերակի փորձարկում: Վերոհիշյալ հեշ ֆունկցիան արտադրում է 256 բիթից ավելի մեծ երկարությամբ հեշ, այդ պատճառով այս հարձակումը այդքան էլ պրակտիկ չէ:

Բանալու որոնումը կարող է իրականացվել ցանկացած ծածկագրի կամ ալգորիթմի վրա, որն ընդունում է բանալին որպես մուտքային արժեք: Բանալու երկարությունը այս հարձակումից հիմնական պաշտպանվածության բնութագիրն է: Ձևավորված հեշի և բանալու միջև կապը պետք է լինի նաև բարդ այդ՝ անհաշվարկելի՝ հարձակումից պաշտպանվելու համար: Վերոհիշյալ հեշ ֆունկցիան օգտագործում է 256 բիթային բանալի, որը նշանակում է, որ որոնման համար հնարավոր բանալիների քանակը չափազանց մեծ է: Կատարվել է թեստավորում այս տեսակի գրոհի դեմ, որի համար հեշավորվել է հետևյալ հաղորդագրությունը «*Quick but same time slow lazy dog*» 1000 անգամ: Որպես հեշավորման բանալու հիմք օգտագործվել է հետևյալ «qwertyuiopasdfghjklzxcvbnm 456789» տողը, և յուրաքանչյուր հեշավորված արժեքի համար այդ տողում կատարվել են 1 բիթային փոփոխություններ: Արդյունքում ստացվել է 52% տարբերության նվազագույն շեմ, որը բավարար է, որպեսզի ալգորիթմը համարվի կայուն այդ տեսակի գրոհի դեմ:

Ֆիքսված կետի գրոհումն իրականացվում է հեշ ֆունկցիայի կամ ծածկագրի վրա գտնելով դրա ֆիքսված կետերը: Ֆիքսված կետեր են համարվում այն մուտքային արժեքեքները, որոնք f ֆունկիցայի միանգամյա կամ ցիկլիկ օգտագործման դեպքում հավասար են իրենց ելքային արժեքներին: Օրնակ, երբ A մուտքային արժեքի դեպքում f(A) ֆունկցիայի ելքային B արժեքը հավասար է A-ին կամ f(A) = B և f(B) = A: Վերոհիշյալ հեշ

ֆունկցիայի ներքին ֆունկիցաների համար 0, 0.5 և 1 ֆիքսված կետեր են: Այս հեշ-ի մուտքի սահմանները գտնվում են (0, 1) միջակայքում, և if պարզ պայմանի օգնությամբ 0.5 արժեքը կարող է հեռացվել մուտքային տվյալներից: Այլ օգտագործվող ֆունկցիաները համարվում են ոչ գծային և չունեն այլ ֆիքսված կետեր: Այս եզրակացությունը ցույց է տալիս առաջարկված սխեմայի պաշտպանվածությունը ֆիքսված կետի գրոհների նկատմամբ:

MITM(Meet-in-the-middle) գրոհը նման է <<ծննդյան օրվա>> գրոհի, բայց այն օգտագործում է IV ներքին արժեքները: Առաջարկված հեշավորման սխեման օգտագործվում է DCS IV արտադրելու համար, իսկ յուրաքանչյուր փուլի վերջում բոլոր IV արժեքները ազդում են յուրաքանչյուր f_i ելքի վրա: Ենթադրելով, որ ֆունկցիաների զանգվածը պատահականորեն ընտրում է fi ֆունկցիաները, և յուրաքանչյուր ֆունկցիա օգտագործում է թույլ կամ ուժեղ հարաբերություն ֆունկցիաների մյուս ելքերի հետ՝ այս գրոհը չի կարող կիրառվել առաջարկված հեշավորման սխեմայի նկատմամբ [62]:

Բարձր մակարդակի գրոհները, ինչպիսիք են պարզ գրոհների համադրությունը, դիֆերենցիալ գրոհը, վիճակագրական դիֆերենցիալ գրոհը և ծիածանային հեշ աղյուսակների գրոհը, հիմնված են դիֆուզիայի սկզբունքի վրա: Հեշավորման ալգորիթմի պատահական վարքագծից պատճառով այս հարձակումների իրականացումը չափազանց բարդ խնդիր է և հեշավորման ալգորիթմն կարելի է համարել կայուն նման հարձակումների նկատմամբ :

Հեշավորման արագությունը: Ներքին f_i ֆունկցիաները իրականացվում են առանձին հոսքերում աշխատանքի համար՝ հեշավորման արագության մեծացման նպատակով: 300 ՄԲ պատահական տվյալների համար հեշավորման արագությունը հոսքային ռեժիմում կազմում է 60 .ՄԲ/վ՝ 6 հոսքերով, մինչդեռ սովորական ռեժիմում (մեկ հոսք) այն կազմում է 24 ՄԲ/վ:

3.4. Գլուխ 3-ի եզրակացություն

Այս գլխում առաջարկվել է իրերի ինտերնետ միջավայրում գործառույթների ամբողջականության և անվտանգության ապահովման մեթոդ, որը հիմնված է հատուկ մշակված արձանագրության և քաոսային հեշ ֆունկցիայի վրա: Արդյունքները ցույց են տալիս, որ առաջարկված քաոսային համակարգը, որը պատահական թվերի գեներատոր է, լուծում է առկա թերությունները այլ քաոսային հեշ ֆունկցիաներում: Եզակի միաչափ քաոսային արտապատկերման օգտագործումը՝ որպես մեկ աղբյուր մի քանիսի փոխարեն, որոնք ունեն տարբեր հաշվողական արագություններ, թույլ է տալիս մեծացնել արագությունը և անվտանգությունը:

Ներկայացրած հեշ ֆունկցիան, որն օգտագործում է ֆունկցիաների դինամիկ պատահական զանգված, ոլորտում առկա այլ լուծումների համեմատ, ապահովում է ավելի բարձր կատարողականություն հեշավորման արագության առումով, և այն անձեռնմխելի է հայտնի գրոհների նկատմամբ:

Մշակված արձանագրության հիման վրա հնարավոր է ապահովել գործառույթների անվտանգությունը և ամբողջականությունը:

ԳԼՈՒԽ 4.

ԻՐԵՐԻ ԻՆՏԵՐՆԵՏ ՄԻՋԱՎԱՅՐՈՒՄ ԲԱՆԱԼԻՆԵՐԻ ԲԱՇԽՄԱՆ ԵՎ ԳՈՐԾԱՌՈՒՅԹՆԵՐԻ ԱՄԲՈՂՋԱԿԱՆՈՒԹՅԱՆ ԱՊԱՀՈՎՄԱՆ ԾՐԱԳՐԱՅԻՆ ՀԱՄԱԿԱՐԳԻ ԻՐԱԿԱՆԱՑՈՒՄԸ

4.1. Մշակված ծրագրային իրականացման ընդհանուր բնութագիրը

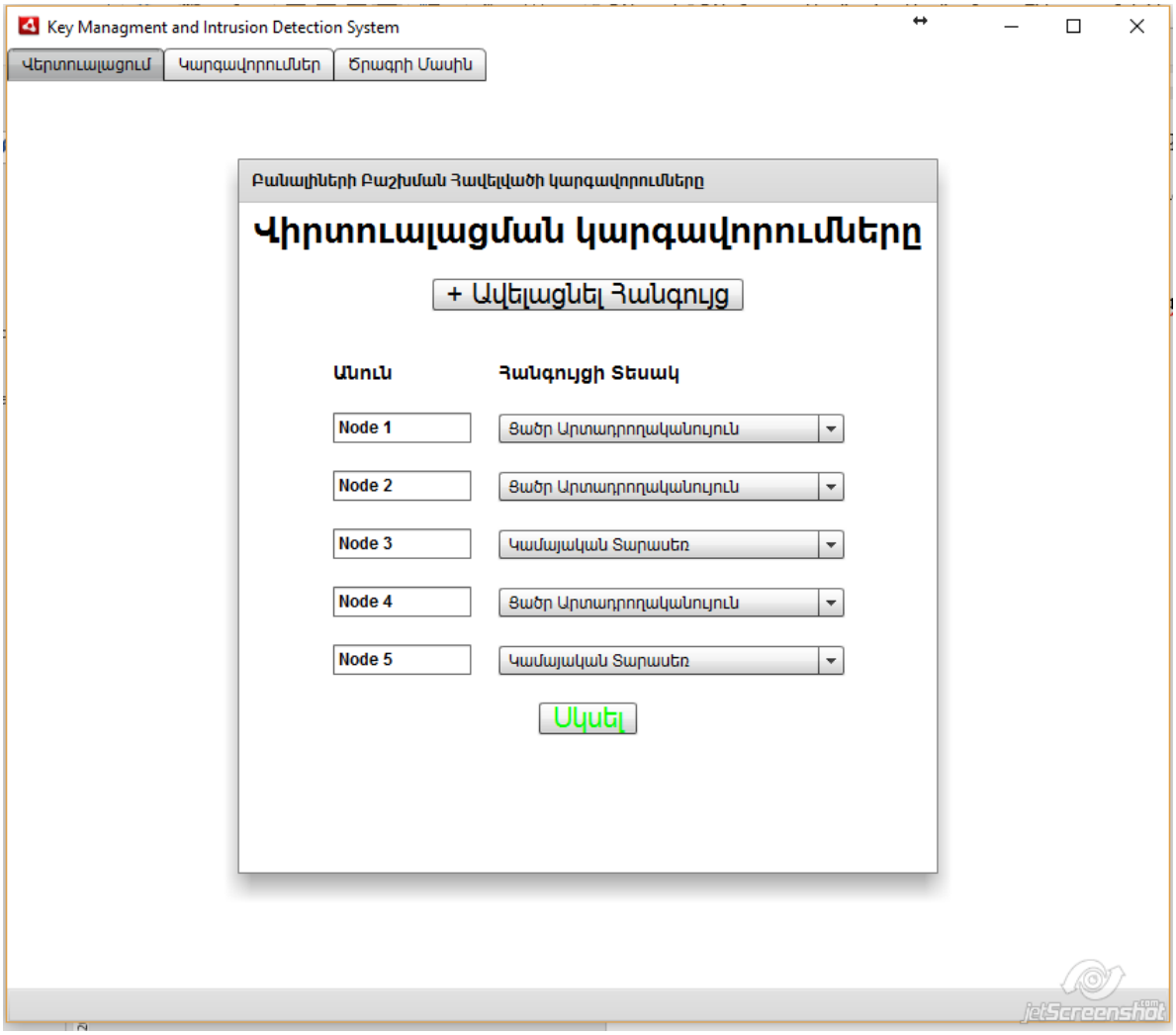
Ատենախոսության մեջ բերված հետազոտությունների հիման վրա մշակվել և փորձարկվել են հետևյալ ծրագրային լուծումները.

- **Ցանցային մոդուլը**, որն իրականացնում է ցանցի վիճակի փոփոխության գրանցումը և բացահայտում է ցանցին միացված սարքերը:
- **IKS Բանալիների բաշխման մոդուլը**, որը հիմնվել է աղյուսակ 1-ում թվարկված նախաբեռնված կարգավորումների վրա, նախ՝ գործարկում է հանգույցների դերակատարման ծրագրային ենթամոդուլը, այնուհետև՝ և բանալիների ենթակառուցվածքի ստեղծման ենթամոդուլը:
- **SIT մոդուլը,** որը տեղադրվում և գործարկվում է նախապես որոշված հանգույցում՝ շարժական բանալիում: Տվյալ կիրառության հիմնական գործառույթն է՝ փաթեթների գաղտնագրումը ստացված այլ հանգույցներից, որոնք չունեն բավարար ռեսուրսներ՝ այն ինքնուրյուն գաղտնագրելու համար:
- **Իրերի ինտերնետ միջավայրում գործառույթների ամբողջականությունը ապահովող մոդուլը,** որը մշակված արձանագրության և քաոսային հեշ ֆունկցիայի իրացման շնորհիվ ապահովում է իրերի ինտերնետ միջավայրում գործառույթների անվտանգությունը և ամբողջականությունը:

Վերոհիշյալ ծրագրային լուծումների փորձարկման համար ստեղծվել է համապատասխան վիրտուալացման համակարգ, որը հիմնված է Docker ծրագրային միջավայրի վրա: Այն հնարավորություն է տալիս ստեղծել արհեստական իրերի ինտերնետ միջավայր, որտեղ օգտատերն առաջադրում է գործարկվող հանգույցների քանակը և կարողանում է հետևել համակարգի աշխատանքին իրական ժամանակում:

4.2. *Իրերի ինտերնետ միջավայրի վիրտուալացման համակարգը*

Մշակված մեթոդների փորձարկման համար ստեղծվել է իրերի ինտերնետ միջավայրի վիրտուալացման համակարգ, որն ստեղծում է իրարից անկախ աշխատող հանգույցներ:



Նկ.17. Վիրտուալացման համակարգի հանգույցների ղեկավարման և գործարկման գրաֆիկական ինտերֆեյսը

Նշված վիրտուալացման համակարգի ստեղծման համար օգտագործվել է docker հիմնահարթակը [66]: Docker հիմնահարթակը հնարավորություն է ընձեռում մեկուսացնել

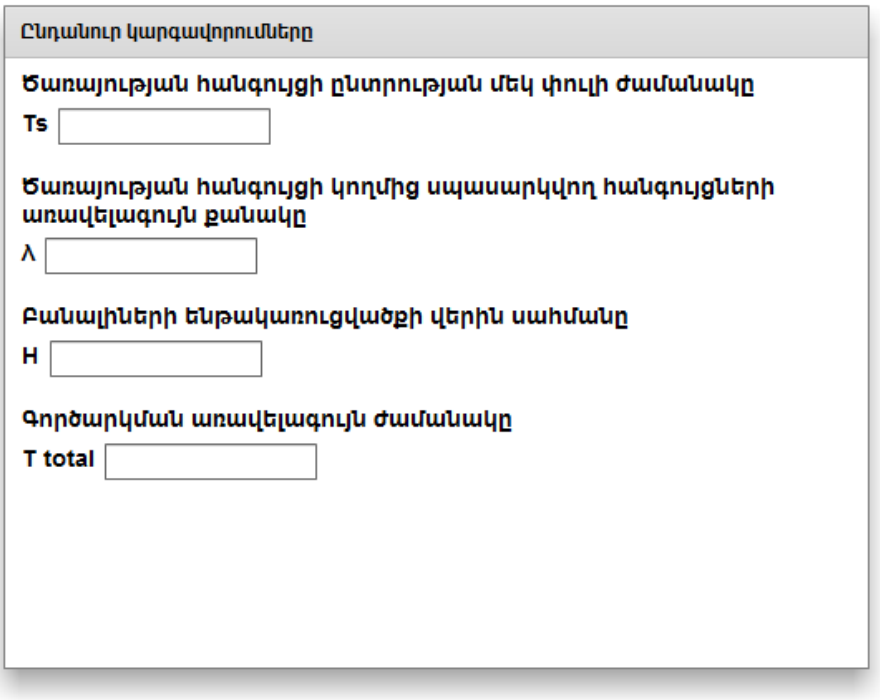
յուրաքանչյուր հանգույցի աշխատանքը: Մեկուսացման հիմքում ընկած են Linux միջավայրի հետևալ հասկացությունները՝ cgroup, union ֆայլային տիրույթը և միջուկի պրոցեսների անվան տիրույթները [67]:

Ստեղծվել է բանալիների բաշխման մեթոդի վրա հիմնված IKS ծրագրային համակարգը, որն ընձեռում է տարատեսակ ինտերնետ իրերի միջև տվյալների անվտանգ փոխանակում ապահովելու հնարավորություն: Այն ներդրվում է տարատեսակ ինտերնետ իրերում որոնցում հնարավոր է կատարել գաղտնագրային ընթացակարգեր և չի պահանջում նախնական հարմարեցում ցանցին՝ մինչ այդ ցանցին միանալը: Նաև ստեղծվել է SIT ծրագրային իրականացումը, որը նախատեսված է օգտագործել իրերի ինտերնետ միջավայրին կից գտնվող համապատասխան սարքում (շարժական բանալիում): Այն հիմնված է մշակված գաղտնահամակարգի վրա և ապահովում է տվյալների անվտանգ փոխանակում այն ինտերնետ իրերի համար, որոնք չեն ապահովում գաղտնագրային ընթացակարգեր:

Վիրտուալացման միջավայրի բնութագրման համար օգտագործվում են այն պատկերները(images), որոնք պարունակում են վիրտուալացրած միջավայրի ծրագրային անհրաժեշտ կարգավորումները և հավելվածները: Ստեղծվել են 4 տարբեր տեսակի պատկերներ.

- Կամայական տարատեսակ պատկերը, որը նախատեսված է՝ գլուխ 2-ում նկարագրած բանալիների բաշխման սխեմայի հիման վրա ստեղծված IKS ծրագրի փորձարկման համար: Տվյալ տեսակի հանգույցում տեղադրվում է IKS ծրագրային մոդուլը՝ անհրաժեշտ նախաբեռնված կարգավորումներով, որը հնարավոր է փոփոխել վիրտուալացումից առաջ՝ համապատասխան պատուհանում: Նշված տեսակի հանգույցի տեխնիկական բնութագրերը,

մասնավորապես՝ հիշողությունը և պրոցեսորի արտադրողականությունն ընտրվում են պատահականության սկզբունքով՝ տարատեսակ միջավայր ստանալու նպատակով: Այս տեսակի հանգույցների ստեղծման քանակը սահմանափակ է, որը կապված է համակարգչի հաշվողական ռեսուրսների հետ:



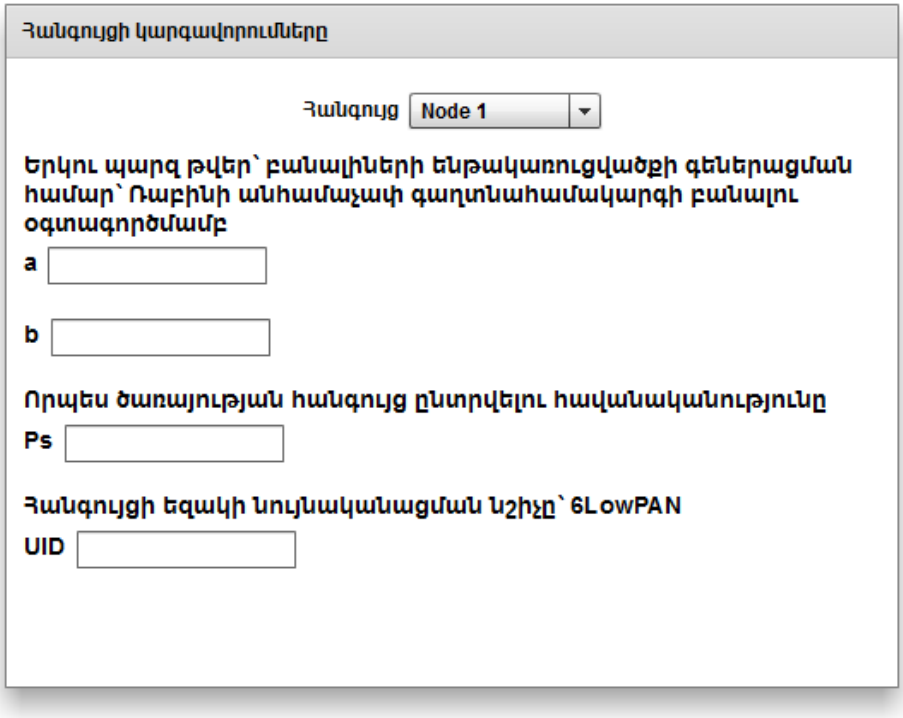
Նկ. 18. Համակարգի ընդհանուր հնարավոր կարգավորումները

- Ցածր արտադրողականության պատկերը, որը նախատեսված է քիչ ռեսուրսներ ունեցող ինտերնետ իրերի նմանակման համար: Տվյալ տիպի հանգույցները զուրկ են գաղտնագրման որևէ գործառույթներից և հասարակ հարցումներ են կատարում նշված պարբերությամբ: Նաև այս հանգույցները կարողանում են ընդունել 100 տեսակի տարբեր հրամաններ՝ առաջարկված գաղտնահամակարգը փորձարկելու նպատակով: Եթե ընտրված է գոնե մեկ

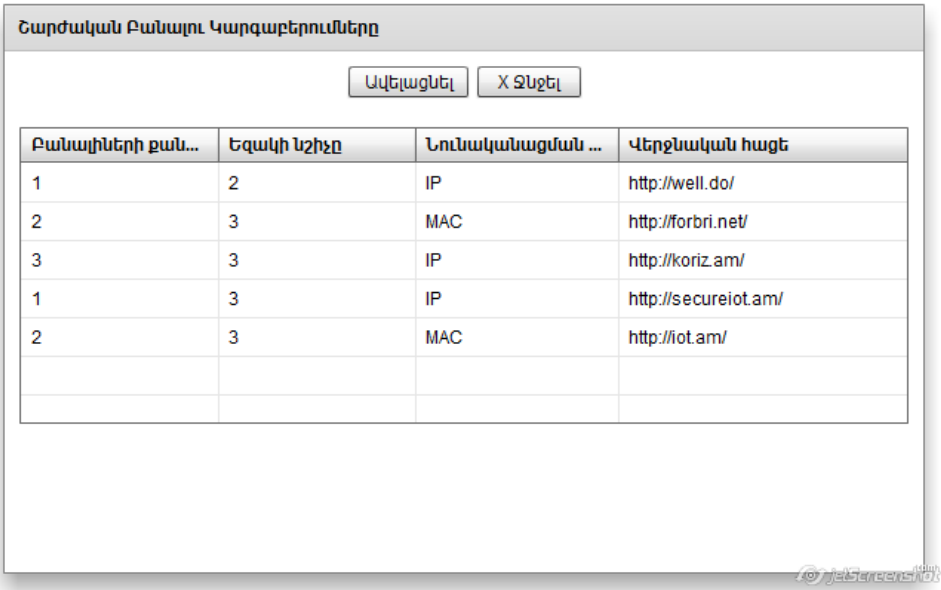
ցածր արտադրողականության հանգույց, դա նշանակում է, որ նույնպես կստեղծվի շարժական բանալու վերտուալացում:

- Շարժական բանալու պատկերը, որում նախապես ներդրված SIT ծրագրային իրականացումը: Շարժական բանալին ցածր արտադրողականություն ունեցող հանգույցներից ստացված և դեպի այդ հանգույցներ ուղարկված հարցումներում առկա տվյալները, հեշային դասին պատկանելու դեպքում հաշվարկում և արտապատկերում է, իսկ տեղեկատվության դեպքում գաղտնագրում կամ վերծանում է՝ ապահովելով տվյալների անվտանգ փոխանակման գործընթացը: Շարժական բանալու վեբ գրաֆիկական կառավարման վահանակը հասանելի է ցանկացած շարժական բանալու պատկեր օգտագործած հանգույցում՝ կիրառելով այդ հանգույցի հասցեի 8086 փորտը [65]: Հարկ է նշել, որ այս կառուցվածքում սերվերային համակարգի դերը կատարում է նույն շարժական բանալին, քանի որ նա է ձևավորում դեպի պարզ իրեր ուղարկվող հարցումները: Տվյալ որոշումը կատարվել է՝ ավելորդ կառուցվածքային բարդություններից խուսափելու նպատակով: Շարժական բանալու մեջ նախօրոք ներդրվել է G և R մատրիցի 10000 օրինակ, որը բավարար է առաջարկված մեթոդի փորձարկումների համար:
- Գործառույթների ամբողջականությունը և անվտանգությունն ապահովող համակարգի պատկերը, որը մշակված արձանագրության շնորհիվ ապահովում է բոլոր հանգույցների միջև կատարվող գործառույթների ամբողջականությունը և գաղտնիությունը: Իրերի ինտերնետ վերտուալացրած միջավայրում տվյալ հանգույցը կարող է գոյություն ունենալ մինչ 15 օրինակով: Սյս պատկերների հիման վրա ստեղծված հանգույցների վիրտուալացման ավարտից հետո կատարվում է բոլոր այս հանգույցների հասցեների

պահեստավորում և T պարբերությամբ համապատասխան հարցումներից ձևավորում է փորձարկում: Հարցման առաջին հանգույցը և ամբողջ շղթայի հանգույցների հաջորդականություն ընտրվում է կամայական սկզբունքով:



Նկ. 19. Հանգույցների հնարավոր կարգավորումները



Նկ. 20. Շարժական բանալու կառավարման ինետրֆեյսը

4.3. Գլուխ 4-ի եզրակացություն

- Մշակվել է IKS ծրագրային համակարգը, որը հիմնված է բանալիների բաշխման մեթոդի վրա և հնարավորություն է ընձեռում ապահովել տվյալների անվտանգ փոխանակում տարատեսակ ինտերնետ իրերի միջև:
- Մշակվել է SIT ծրագրային համակարգը, որի ներդրումը համապատասխան սարքում ապահովում է տվյալների անվտանգ փոխանակում այն ինտերնետ իրերի համար, որոնք գաղտնագրային ընթացակարգեր չեն ապահովում:
- Մշակվել է իրերի ինտերնետ միջավայրում գործառույթների ամբողջականությունն ապահովող ծրագրային գործիքամիջոց, որն օգտագործելով մշակված արձանագրությունը, ապահովում է ինտերնետ իրերի անվտանգությունը միջավայրին հատուկ գրոհների նկատմամբ:
- Մշակված ծրագրային համակարգը հնարավորություն է տալիս ստեղծել տարատեսակ իրերի ինտերնետ միջավայր և փորձարկել վերոհիշյալ ծրագրային մոդուլները տարբեր տեխնիկական և ծրագրային ապահովման պայմաններում:

Եզրակացություն

- Մշակվել է գաղտնագրային բանալիների կառավարման մեթոդ, որն ապահովում է սարքերի ինքնակարգավորումը և տվյալների անվտանգ փոխանակումը [41, 63]:
- Մշակվել է գաղտնահամակարգ՝ ամպային միջավայրում պարզ ինտերնետ իրերի հետ անվտանգ հաղորդակցման նպատակով, որն ապահովելով անհրաժեշտ անվտանգություն, սարքերի տեխնիկական բնութագրերի փոփոխություն չի պահանջում [40, 41, 65]:
- Մշակվել է ամպային բաշխված ցանցերում գործառույթների ամբողջականությունն ապահովող մեթոդ, որը բացահայտում և կանխարգելում է ամպային միջավայրին ուղղված հատուկ գրոհները [35, 40, 64]:
- Մշակվել է բանալիների կառավարման IKS ծրագրային մոդուլը, որը հնարավորություն է ընձեռում, առանց սարքի նախնական կարգաբերման, միացնել այն ցանցին և ապահովել այլ ինտերնետ իրերի հետ անվտանգ կապը: Գաղտնագրման հնարավորություններից զուրկ սարքերի միջև անվտանգ տվյալների փոխանակման նպատակով ստեղծվել է SIT ծրագրային ապահովումը, որը ներդրվում է ցանցում գտնվող լրացուցիչ սարքում և ապահովում է գաղտնագրման գործառույթները: Մշակվել է նաև գլուխ 3-ում նկարագրված արձանագրության ծրագրային իրացումը, որի կիրառման արդյունքում հնարավոր է ապահովել իրերի ինտերնետ միջավայրում կատարվող գործառույթների անվտանգությունն ու ամբողջականությունը [41, 40, 64]:

Օգտագործված գրականության ցանկ

1. Mohapatra Smaranika, Kusum Lata Jain (2017) Analytical Solutions For Security Issues In Cloud Environment. International Journal Of Engineering And Computer Science ISSN: 2319-7242 Volume 6 Issue 1, https://www.ijecs.in/issue/v6-i1/9ijecs.pdf
2. R. C. D. Jr., C. Carver, and A. J. Ferguson, "Phishing for user security awareness," Computers & Security, vol. 26, no. 1, pp. 73 – 80, 2007.
3. Weinhardt C, Anandasivam A, Blau B, Borissov N, Meinl T, Michalk W, Stößer J (2009) Cloud computing—a classification business models, and research directions. Bus Inf Syst Eng 1(5): pp 391–399
4. Tippit Inc. (2008) WebHostingUnleashed: Cloud-computing services comparison guide. http://www.itsj.com/Resources/cloudcomputing-comparison.pdf.
5. Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation Computer Systems, 599–616.
6. Management Association, Information Resources (2008) Web-Based Services: Concepts, Methodologies, Tools, and Applications.*https://books.google.am/books?isbn=146669467X* p.572.
7. Mell P, Grance T (2009) The NIST definition of cloud computing (v15). Tech Rep, National Institute of Standards and Technology, p.2. http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf
8. Hilley D (2009) Cloud computing: a taxonomy of platform and infrastructure-level offerings. Tech Rep GIT-CERCS-09-13, CERCS, Georgia Institute of Technology
9. Youseff L, Butrico M, Da Silva D (2008) Toward a unified ontology of cloud computing. In: Grid computing environments workshop, GCE'08, pp 1–10
10. Viega J (2009) Cloud computing and the common man. IEEE Computer, 42(8).pp:106–108.
11. Mather T, Kumaraswamy S (2009) Cloud Security and privacy: An Enterprise Perspective on Risks and Compliance. 1st edition. O'Reilly Media.

12. Ramireddy S, Chakraborthy R, Raghu TS, Rao HR (2010) Privacy and Security Practices in the Arena of Cloud Computing - A Research in Progress. In: AMCIS 2010 Proceedings, AMCIS.

13. Oltsik J (2010) Information security, virtualization, and the journey to the cloud. Tech. rep., Cloud Security Alliance.

14. Chen D. and Zhao H. (2012). Data Security and Privacy Protection Issues in Cloud Computing. International Conference on Computer Science and Electronics Engineering, pp.647-651.

15. Smiley S. (2016) Active RFID vs. Passive RFID: What's the Difference?, RFID Insider.

16. Chris, S. (2011) An internet of things that do not exist. *Interactions*, Vol. 18, No. 3, pp.18–21.

17. Giusto, D., Iera, A., Morabito, G. and Atzori, L. (Eds) (2010) *The Internet of Things*, Springer, Berlin.

18. Xiaofeng Lu, Zhaowei Qu, Qi Li, Pan Hui (2015) **Privacy information security classification for internet of things based on internet data.** International Journal of Distributed Sensor Networks -Special issue on Big Data and Knowledge Extraction for Cyber-Physical Systems Volume 2015, January 2015 Article No. 23

19. H. Chan and A. Perrig, PIKE: Peer intermediaries for key establishment in sensor networks, in *Proceedings of IEEE Infocom*, Miami, March 2005.

20. Luigi, A., Antonio, I. and Giacomo, M. (2010) 'The internet of things: a survey', Computer Networks, Vol. 54, No. 15.

21. Eschenauer, L. and Gligor, V.D. (2002) 'A key-management scheme for distributed sensor networks', Proceedings of the 9th ACM Conference on Computer and Communication Security, 41-47.

22. Liu, D., Ning, P. and Li, R.F. (2005) 'Establishing pair-wise keys in distributed sensor networks', ACM Transactions on Information and Systems Security, Vol. 8, No.1. pp.41-77.

23. Zhu, S., Seia, S. and Jajodia, S. (2006) 'LEAP+: efficient security mechanisms for large-scale distributed sensor networks', ACM Transactions on Sensor Networks, Vol. 2, No. 4, pp.500-528.

24. Loree, P., Nygard, K. and Du, X.J. (2009) 'An efficient postdeployment key establishment scheme for heterogeneous sensor networks', Proceedings of 2009 Global Telecommunications Conference, GLOBECOM, pp.1–6.

25. Azarderskhsh, R. and Reyhani-Masoleh, A. (2011) 'Secure clustering and symmetric key establishment in heterogeneous wireless sensor networks', EURASIP Journal on Wireless Communications and Networking, pp.1-12.

26. Huang, J.Y., Liao, I.E. and Tang, H.W. (2011) 'A forward authentication key management scheme for heterogeneous sensor networks', EURASIP Journal on Wireless Communications and Networking, pp.1–10.

27. Sun, Y., Trappe, W. and Liu, K.J.R. (2004) 'A scalable multicast key management cheme for heterogeneous wireless networks', IEEE/ACM Transactions on Networking, Vol.12, No. 4, pp.653–666.

28. "Internet of Things: (2015). How Much are We Exposed to Cyber Threats? - InfoSec Resources."http://resources.infosecinstitute.com/internet-things-much-exposed-cyber-threats/.

29. Dou, W., Chen, Q. and Chen, J. (2013). A confidence-based filtering method for DDoS attack defense in cloud environment. Future Generation Computer Systems, 1838–1850.

30. Mattern, F. and Floerkemeier, C. (2010) 'From the internet of computers to the internet of things', *Lecture Notes in Computer Science*, Vol. 6462, pp.242–259.

31. Hui, J.W. and Culler, D.E. (2008) 'Extending IP to low-power wireless personal area networks', *IEEE Internet Computing*, Vol. 12, No .4, pp.37–45.

32. Du, W., Deng, J., Han, Y.S., Chen, S.G. and Varshney, P.K. (2004) 'A key management scheme for wireless sensor networks using deployment knowledge', IEEE INFOCOM'04, pp.586–597.

33. Djamel, D. and Nadjib, B. (2010) ‘A gradual solution to detect selfish nodes in mobile ad hoc networks’, International Journal of Wireless and Mobile Computing, Vol. 4, No. 4 pp.264–274.

34. Tashkova, K., Korosec, P. and Silc, J. (2011) A distributed multilevel ant-colony algorithm for the multi-way graph partitioning’, International Journal of Bio-Inspired Computation, Vol. 3, No. 5, pp.286–296.

35. 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, pp. 101-105.

36. Blundo, C., Santis, A.D., Herzberg, A., Kutten, S., Vaccaro, U. and Yung, M. (1992) ‘Perfectly-secure key distribution for dynamic conferences’, Proceeding of 12th Annual International Cryptology Conference, pp.471–486.

37. Blom, R. (1984) ‘An optimal class of symmetric key generation systems’, Proceeding of Workshop Theory and Application of Cryptographic Techniques, pp.335–338.

38. Mark, M. and Ahmad-Reza, S. (2010) ‘Key agreement for heterogeneous mobile ad-hoc groups’, *International Journal of Wireless and Mobile Computing*, Vol. 4, No. 1, pp.17–30.

39. Ei, T. and Susumu, S. (2011) ‘A study of the relationship between scale-freeness and evolution of cooperation’, *International Journal of Bio-Inspired Computation*, Vol. 3, No. 3, pp.142–150.

40. Hovsepyan V., “Securing data transfer in IOT environment”, Науковий журнал Безпека інформації, Киев, Украина, pp. 131-134, Kiev, Ukraine 2016.

41. Հովսեփյան Վ.,"Շարժական բանալու կիրառումը ամպային միջավայրում", ՀԱՊՀ Լրաբեր, գիտական և մեթոդական հոդվածների ժողովածու, Երևան, Հայաստան, 2016, pp. 162-167.

42. Chan H. and. Perrig A, (2005) PIKE: Peer intermediaries for key establishment in sensor networks, in Proceedings of IEEE Infocom, Miami.

43. Zhou L.,J., Ravishankar C. V., (2005) Efficient Key Establishment for Group-Based Wireless Sensor Networks, pp. 1-10, 2005.

44. Tan Y. K. and. Panda S. K, (2011) "Optimized wind energy harvesting system using resistance emulator and active rectifier for wireless sensor nodes," Power Electronics, IEEE Transactions on, vol. 26, no. 1, pp. 38–50.

45. Kocarev L, Lian S. (2011) Chaos-based cryptography: theory, algorithms and applications. Springer.

46. Amin M, Faragallah OS, El-Latif AAA. (2009) Chaos-based hash function (CBHF) for cryptographic applications. Chaos, Solitons. pp. 767–772.

47. Wang X, Yu H. How to break md5 and other hash functions. In: Cramer R, (2005). Advances in cryptology EUROCRYPT 2005. Lecture Notes in Com- puter Science. Berlin Heidelberg: Springer; pp.19–35.

48. Preneel B. (1994) Cryptographic hash functions. Eur Trans Telecomm.

49. Eastlake DE, Jones PE. (2001) US secure hash algorithm 1 (sha1). http://www.ietf.org/rfc/rfc3174.

50. Stevens M, Lenstra AK, Weger BD. (2012) Chosen-prefix collisions for md5 and applications. Int J Appl Cryptogr.

51. Shannon C. Communication theory of secrecy systems. Bell Syst Tech: pp. 656–715.

52. Gandal N., Halaburda H. (2014) Competition in the cryptocurrency market. CEPR Discussion Paper.

53. Zhou Y, Bao L, Chen CP. (2014) A new 1d chaotic system for image encryption. Signal Process;

54. San-Um W, Srichavengsup W. (2016) A robust hash function using cross-coupled chaotic maps with absolute-valued sinusoidal nonlinearity. Int J Adv Comput Sci Appl.

55. Kwok-Wo Wong. (2003) A combined chaotic cryptographic and hashing scheme, Volume 307.

56. Xiao D, Liao X, Wang Y. (2009) Parallel keyed hash function construction based on chaotic neural network. Neurocomputing.

57. Li Y, Xiao D , Deng S, Han Q, Zhou G. (2011) Parallel hash function construction based on chaotic maps with changeable parameters. Neural Comput Appl.

58. Menezes AJ, Van Oorschot PC, Vanstone SA. (1996) Handbook of applied cryptography. CRC Press.

59. Bard G. Algebraic cryptanalysis. Springer Science & Business Media.2009

60. James Fiedler, "Hamming Codes", 2014

61. Aoki K, Sasaki Y. (2009) Meet-in-the-middle preimage attacks against reduced sha-0 and sha-1. In: Proceedings of the advances in cryptology–CRYPTO. Springer. p. 70–89.

62. Ohta K, Koyama K. Meet-in-the-middle attack on digital signature schemes. In: Proceedings of the advances in cryptology AUSCRYPT'. Springer.

63. Hovsepyan V.,"Secure Real-Time Data Transfer in the Cloud", Meeting Security Challenges Through Data Analytics and Decision Support. "NATO Science for Peace and Security" Series - D:Information and Communication Security – 2016, Volume47, pp.271-276.

64. Hovsepyan V., "Files full life cycle protection and secure distribution", Proceedings of the Conference Computer Science and Information Technologies (CSIT-2015), Yerevan 2016, pp. 217-219.

65. Hovsepyan V., "Securing of IOT environment", Тезисы докладов международной научно-практической конференции молодых ученых и студентов, Киев, Украина, 2016, С. 224-226.

66. Madiha H. Syed and Eduardo B. Fernandez "The Software Container pattern" Florida Atlantic University, Computer Science Congress.

67. Rüdiger Landmann 2010, Resource Management Guide, Managing system resources on Red Hat Enterprise Linux 6

Նկարների ցանկ

Նկարի անվանումը	**Էջը**
Նկ. 1. Ամպային ծառայությունների հիմնական մակարդակները	13
Նկ. 2. Ամպային համակարգերի անվտանգության կարգաբանությունը	17
Նկ. 3. Ցանցային հանգույցները և դրանց կապերը	37
Նկ. 4. Հանգույցների դերակատարումների վերագրման ալգորիթմի բլոկ-սխեման	44
Նկ. 5 Ծառայության հանգույցի և դրան համապատասխանող աշխատանքային հանգույցի միջև հաղորդակցության սխեման	47
Նկ. 6. Ծառայության հանգույցների քանակի կախվածությունն հանգույցների ընդհանուր քանակից	51
Նկ. 7. Բանալիների պահպանման համար անհրաժեշտ հիշողության ծավալի համեմատականը դասական և առաջարկված մեթոդների դեպքում	51
Նկ. 8. Շարժական բանալու կառավարման վահանակում նոր կարգաբերումներ ավելացման պատուհանը	54
Նկ. 9. Գաղտնահամակարգի աշխատանքի սխեման	55
Նկ.10. Շարժական բանալու կողմից կատարվող հարցումները	56
Նկ.11. Փուլերի ընթացքում R մատրիցի ձևափոխման օրինակը	63
Նկ. 12. Շարժական բանալիում առկա կարգաբերումների օրինակ	65
Նկ. 13. Լոգիստական արտապատկերման երկատման դիագրամը	77
Նկ. 14. LS արտապատկերման երկատման դիագրամը	78

Նկ. 15. Հեշ ֆունկցիայի բլոկ-սխեման	85
Նկ. 16. f_i ֆունկցիաների կառուցվածքը	87
Նկ. 17. Վիրտուալացման համակարգի հանգույցների ղեկավարման և գործարկման գրաֆիկական ինտերֆեյսը	95
Նկ. 18. Համակարգի ընդհանուր հնարավոր կարգավորումները	97
Նկ. 19. Հանգույցներ հնարավոր կարգավորումները	99
Նկ. 20 Շարժական բանալու կարգաբերումների փոփոխման իներֆեյսը	99

Աղյուսակների ցուցակ

Աղյուսակի անվանումը	**Էջը**
Աղյուսակ 1. Հանգույցների նախասահմանված և նախաբեռնված պարամետրերը	39
Աղյուսակ 2. Հաղորդագրությունների ձևաչափը	47
Աղյուսակ 3. Ջևավորված R մատրիցի օրինակը	57
Աղյուսակ 4. Հրամանների և հեշերի արտապատկերումը	58
Աղյուսակ 5. Ձևավորված բանալու ձևաչափը	59
Աղյուսակ 6. Հաղորդագրությունների ձևաչափը	59
Աղյուսակ 7. Առաջարկված գաղտնահամակարգի արագագործության համեմատականը	64
Աղյուսակ 8. Հաղորդագրության ձևաչափը	68
Աղյուսակ 9. Հաղորդագրության գլխամասի ձևաչափը	68
Աղյուսակ 10. Համապատասխան հեշ արժեքները	88
Աղյուսակ 11. Փորձարկման վիճակագրական արդյունքները՝ օրինակների N քանակով	89

Հավելված 1

ԿԱՆԽԱՐԳԵԼԻՉ ՍՐՏԱԲԱՆՈՒԹՅԱՆ ԿԵՆՏՐՈՆ
CENTER OF PREVENTIVE CARDIOLOGY

ԲԱՐԵԿԱՄ ՍՊԸ
BAREKAM LLC

Հայաստանի
Հանրապետություն
ք. Երևան 0014,
Պ. Սևակի 5
ՀՎՀՀ 02574766
Հեռ. 055 288-616
010 288-616

Republic of Armenia
Yerevan
P. Sevak 5
Tax Cod:02574766
Tel 055 288-616
010 288-616

E-mail:
zelveian@cardio.am
Web Site:
www.cardio.am

N 4293/1 22.01.2016թ.

ԱԿՏ
Վլադիմիր Հովսեփի Հովսեփյանի «Ամպային տեխնոլոգիաներով անվտանգ աշխատելու լրացուցիչ միջոցների մշակում» ատենախոսության արդյունքների ներդրման մասին

Վլադիմիր Հովսեփի Հովսեփյանի «Ամպային տեխնոլոգիաներով անվտանգ աշխատելու լրացուցիչ միջոցների մշակում» ատենախոսության շրջանակներում մշակված մեթոդը ներդրվել է «Կանխարգելիչ Սրտաբանության Կենտրոնում» տվյալների անվտանգ փոխանցման նպատակով:

Մշակված մեթոդը կիրառվել է «Կանխարգելիչ Սրտաբանության Կենտրոնում» ինտերնետ իրերի սենսորների միջոցով հիվանդի առողջական վիճակի վերաբերյալ տվյալների անվտանգ փոխանցման համար, որը թույլ է տալիս ապահովել հիվանդի առողջական վիճակի վերաբերյալ տվյալների գաղտնիությունը:

Կենտրոնի գիտական ղեկավար,
ք. Երևան գլխավոր սրտաբան,
բ.գ.դ., պրոֆեսոր **Պ.Հ. Զելվեյան**

Հավելված 2

Հատված հրամանների հաշվարկում կատարող ծրագրի կոդից

```
const rounds = [
  { name: "Roud 2", transformFunc: njUtils.rotate90 },
  { name: "Roud 3", transformFunc: njUtils.rotate270 },
  { name: "Roud 4", transformFunc: njUtils.rotate180 },
  { name: "Roud 5", transformFunc: njUtils.matrixTranspose },
];

class KeyProvider {
  constructor (keyLength, symbols, moves, M = 8, N = 8, offset = 0, round = 0) {
    this.keyLength = keyLength;
    this.symbols = symbols;    this._symbols = symbols.concat();
    this.moves = moves;    this._moves = moves.concat();
    this.offset = offset;    this.round = round;
    this.M = M;    this.N = N;    this.symbolsShift = 0;
  }
  getCk() {
    const length = this.keyLength + this.offset;
    let offset = this.offset;    let result = 0;
    while(offset < length) {
      const index = this.moves[offset];
      const symbols = this.symbols[index];
      result += index ^ symbols;
      offset ++;
    }
    return result;
  }
  getKey( ) {
    let results = [];    let offset = this.offset;
    const length = this.keyLength + this.offset;
    while(offset < length) {
      const index = this.moves[offset];
      results.push(this.symbols[index])
      offset ++;
    }
    return results;
  }
  nextKey(pure = false) {
    this.offset += this.keyLength;
    this.checkOffset();
    this.checkRound();
    const key = this.getKey();
    return key;
  }
  checkOffset() {
    if (this.offset === this.moves.length) {
      this.offset = 0;
      this.symbolsShift ++;
      const lSymbole = this.symbols.pop();
      this.symbols.unshift(lSymbole);
      }
  }
  checkRound() {
    if (this.symbolsShift === 64) {
      this.symbolsShift = 0;
      if (rounds.length === this.round) {
        this.symbols = this._symbols;
        this.round = 0;
        return;
      }
      const transformFunc  = rounds[this.round].transformFunc;
      if (this.round === rounds.length) {
        this.round = 0;
        return;
      }
      this.symbols = transformFunc(this._symbols, this.N, this.M);
      this.round ++;
```