ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Մարտուն Միքայելի Կարապետյան

ԹԱՓԱՆՑԻԿ ԾԱԾԿԱԳՐՈՒԹՅԱՆ ՍԽԵՄԱՆԵՐԻ ՈՐՈՇ ԿԻՐԱՌՈՒԹՅՈՒՆՆԵՐ

Ե 13.05 – «Մաթեմատիկական մոդելավորում, թվային մեթոդներ և ծրագրերի համալիրներ» մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման ատենախոսության

Երևան – 2017

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА

Карапетян Мартун Микаелович

ПРОЗРАЧНЫЕ КРИПТОСИСТЕМЫ И ИХ ПРИМЕНЕНИЯ

А В Т О Р Е Ф Е Р А Т

диссертации на соискание ученой степени кандидата технических наук по специальности 05.13.05 – «Математическое моделирование, численные методы и комплексы программ»

Ереван 2017

Ատենախոսության թեման հաստատվել է ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում

Գիտական ղեկավար՝	տեխ. գիտ. դոկտոր	Գ. Հ. Խաչատրյան
Պաշտոնական ընդդիմախոսներ՝	ֆիզ. մաթ. գիտ. դոկտոր	Լ. Հ. Ասլանյան
	ֆիզ. մաթ. գիտ. թեկնածու	Ս. Ե. Աբրահամյան

Առաջատար կազմակերպություն՝ Հայաստանի ազգային պոլիտեխնիկական համալսարան

Պաշտպանությունը կայանալու է 2017թ. Հունիսի 6-ին, ժամը 16:00-ին ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում գործող 037 «Ինֆորմատիկա» մասնագիտական խորհրդի նիստում հետևյալ հասցեով՝ Երևան, 0014, Պ. Սևակի 1:

Ատենախոսությանը կարելի է ծանոթանալ ՀՀ ԳԱԱ ԻԱՊԻ գրադարանում:
Սեղմագիրն առաքված է 2017թ. Մայիսի 6-ին:

Մասնագիտական խորհրդի գիտական քարտուղար՝ ֆիզ. մաթ. գիտ. դոկտոր Հ. Գ. Սարուխանյան

Тема диссертации утверждена в Институте проблем информатики и автоматизации НАН РА

Научный руководитель:	доктор тех. наук	Г. Г. Хачатрян
Официальные оппоненты:	доктор физ.-мат. наук	Л. А. Асланян
	кандидат физ.-мат. наук	С. Е. Абраамян
Ведущая организация:	Национальный политехнический университет Армении	

Защита состоится 6-го июня 2017г. в 16:00 на заседании специализированного совета 037 «Информатика» Института проблем информатики и автоматизации НАН РА по адресу: 0014, г. Ереван, ул. П. Севака 1.

С диссертацией можно ознакомиться в библиотеке ИПИА НАН РА.
Автореферат разослан X-го мая 2017 г.

Ученый секретарь специализированного совета, доктор физ.-мат. наук А. Г. Саруханян

Աշխատանքի ընդհանուր բնութագիրը

Թեմայի արդիականությունը

Ծածկագրության ավանդական ալգորիթմները ապահով են «սև տուփի» հարձակման միջավայրում, որտեղ հարձակվողը տեսնում է ծածկագրության ալգորիթմի մուտքն ու ելքը, բայց չի կարող հետևել ծրագրի աշխատանքի ընթացքում առաջացած միջանկյալ արժեքներին: Սակայն, որոշ դեպքերում, ծածկագրության ալգորիթմ պարունակող ծրագիրը աշխատում է չպաշտպանված միջավայրում, որտեղ հարձակվողը մուտքի ու ելքի տվյալներից բացի հասանելիություն ունի նաև ծրագրի կատարման ընթացքում առաջացած միջանկյալ տվյալներին և ըստ ցանկության կարող է փոփոխել ծրագրի կատարման ընթացքը: Թափանցիկ ծածկագրության ալգորիթմները նախագծված են այսպիսի չպաշտպանված սարքավորումների վրա աշխատելու համար: Այս ալգորիթմները ստեղծվում են «սև տուփի» ալգորիթմների հիման վրա, բայց ծածկագրման բանալին ուղղակիորեն օգտագործելու փոխարեն ծածկագրման ալգորիթմը օգտագործում է հատուկ աղյուսակներ, որոնք ստեղծվել են բանալու հիման վրա: Ծածկագրության բոլոր գործողությունները իրականացվում են օգտագործելով այս աղյուսակները այնպես, որ հարձակվողը չկարողանա դուրս բերել աղյուսակների կառուցման համար օգտագործված ծածկագրման բանալին, անգամ եթե նրան հասանելի լինեն աղյուսակները, հնարավորություն ունենա հետևել և նույնիսկ ըստ ցանկության փոփոխել ծածկագրման ընթացքի բոլոր գործողությունները: Թափանցիկ ծածկագրման ալգորիթմները կիրառվում են գաղտնի ընտրման (Oblivious transfer), ծածկագրված տվյալների վրա փնտրման ալգորիթմների մեջ, ինչպես նաև որպես հանրային բանալիով ծածկագրման համակարգերի արագագործ այլընտրանք:

AES ծածկագրի առաջին թափանցիկ իրականացումը ներկայացվել է 2002 թվականին Չոի կողմից: Չոի AES թափանցիկ ծածկագրման ալգորիթմի վրա հաջող հարձակման սխեմա ներկայացրեց Բիլետը 2004 թվականին, որը կոչվեց BGE հարձակում: 2009-ին AES-ի անվտանգ թափանցիկ իրականացում ստեղծելու ևս մեկ փորձ իրականացվեց, որը հիմնված էր Չոի աղյուսակներում առկա խառնող արտապատկերումների (Mixing Bijection) ոչնչացումը ալգորիթմի մի փուլից դեպի հաջորդը տեղափոխելու վրա: Սակայն 2013-ին համակարգի նկատմամբ հաջողությամբ կիրառվեց Յոնի Մուլդերի հարձակումը:

2010-ին Կարումին առաջարկեց AES-ի մեկ այլ թափանցիկ ծածկագրման իրականացում, հիմնված դուալ ծածկագրերի վրա, որի նկատմամբ հարձակում հրապարակվեց 2013-ին: Հարձակումը իրենից ներկայացնում էր BGE հայտնի հարձակման ձևափոխված տարբերակ: 2013-ին առաջարկվեց մեկ այլ հարձակման մեթոդ, որը հիմնված էր աղյուսակների արժեքների համընկնումների վրա: Այս հարձակումը թույլ էր տալիս դուրս բերել գաղտնի բանալին բոլոր ստեղծված թափանցիկ իրականացումներից 2^{22} գործողության միջոցով: Հարձակումը ավելի պարզ էր քան BGE հարձակումը, և միաժամանակ աշխատում էր ավելի արագ:

Թվարկված ալգորիթմները և նրանց նկատմամբ հաջող կիրառված հարձակումները ուսումնասիրվել են նոր թափանցիկ ծածկագրի կառուցման նպատակով:

Աշխատանքի նպատակը

- Ուսումնասիրել նախկինում նախագծված սիմետրիկ ծածկագրերի, այդ թվում AES-ի թափանցիկ իրականացումները, դրանց նկատմամբ հաջողությամբ կիրառված հարձակման մեթոդները:
- Ստեղծել SAFER+ սիմետրիկ ծածկագրության ալգորիթմի թափանցիկ իրականացում, որը հնարավոր չի լինի կոտրել գոյություն ունեցող հարձակման մեթոդներով:
- Ուսումնասիրել Գ. Խաչատրյանի և Մ. Կյուրեղյանի հանրային բանալիով ծածկագրության համակարգը, որը հիմնված է տեղափոխության բազմանդամների վրա: Բարելավել նշված համակարգի անվտանգությունը և աշխատանքի արագությունը:

Հետազոտման մեթոդները հիմնված են ծածկագրման դասական հասկացությունների, գոյություն ունեցող թափանցիկ ծածկագրության ալգորիթմների և դրանց նկատմամբ կիրառված հարձակման մեթոդների ուսումնասիրության վրա: Մշակված ծրագրային համակարգերի արագագործության և աշխատունակության ստուգման նպատակով, ինչպես նաև դրանց համեմատության համար այլ մրցակից համակարգերի հետ, օգտագործվել են օբյեկտակողմնորոշված ծրագրավորման մոտեցումները, C++ ծրագրավորման լեզուն և Crypto++ բաց գրադարանը:

Գիտական նորույթը

- SAFER+ սիմետրիկ ծածկագրության ալգորիթմի նոր թափանցիկ իրականացում, որն անվտանգ է նմանատիպ իրականացումների վրա հարձակման գոյություն ունեցող ալգորիթմների նկատմամբ:
- Բարելավված անվտանգությամբ և արագագործությամբ բաց բանալիով թափանցիկ ծածկագրության համակարգ հիմնված տեղափոխության բազմանդամների վրա:

Ստացված արդյունքների կիրառական նշանակությունը և ներդրումները

Ստեղծված SAFER+ ալգորիթմի թափանցիկ իրականացումը կարող է օգտագործվել իրավունքների թվային կառավարման համակարգերում (DRM), ներառյալ օնլայն վիդեո վարձույթի համակարգերում, ինչպես նաև գաղտնի ընտրության սխեմաներում, որոնք հանդիսանում է ֆունկցիայի ամպային հաշվարկման համակարգերի հիմքը: Ստեղծված թափանցիկ ծածկագիրը օգտագործվում է նաև ծածկագրված փնտրման ալգորիթմներում: Նախկինում ծածկագրված տվյալների փնտրման ալգորիթմները օգտագործում էին հանրային բանալիով ծածկագրություն, որի արդյունքում համակարգի արագործությունը թույլ չէր տալիս գործնականում կիրառել դրանք: Որոշ նորագույն արդյունքներում հանրային բանալիով ծածկագրության ալգորիթմները

փոխարինվում են թափանցիկ ծածկագրության ալգորիթմներով, որի արդյունքում համակարգի արագագործությունը աճում է մի քանի կարգով: Այսպիսով ծածկագրված տվյալների փնտրման ալգորիթմները հանդիսանում են ներկայացված SAFER+ ալգորիթմի թափանցիկ իրականացման ևս մեկ կիրառություն:

Ատենախոսության 2-րդ մասում ներկայացված տեղափոխության բազմանդամների վրա հիմնված հանրային բանալիով ծածկագրության լավացված սխեման կարող է փոխարինել հանրային բանալիով ծածկագրման այլ սխեմաների, ինչպիսիք են RSA և էլիպտիկ կորերի համակարգերը, բերելով արագագործության զգալի ավելացման:

Աշխատանքում նկարագրված թափանցիկ ծածկագրության 2 ալգորիթմները ներդրվել են «Ուանքրփթոր» ընկերության ծածկագրված ֆայլերի որոնման համակարգում, որը թույլ է տալիս Dropbox կամ Google Drive ամպային պահոցում պահել ծածկագրված ֆայլեր և միևնույն ժամանակ ունենալ հիմնաբառերով որոնում իրականացնելու, ինչպես նաև ֆայլերը այլ օգտատերերի հետ կիսելու հնարավորություն:

Պաշտպանությանը ներկայացվող դրույթները

1. SAFER+ սիմետրիկ ծածկագրության ալգորիթմի նոր թափանցիկ իրականացում:
2. Բարելավված անվտանգությամբ և արագագործությամբ բաց բանալիով թափանցիկ ծածկագրության համակարգ հիմնված տեղափոխության բազմանդամների վրա:
3. C++ գրադարան, որը իրականացնում է SAFER+ ալգորիթմի մշակված նոր թափանցիկ ծածկագրության սխեման:
4. C++ գրադարան, որը իրականացնում է տեղափոխության բազմանդամների վրա հիմնված հանրային բանալիով ծածկագրության բարելավված թափանցիկ սխեման:

Աշխատանքի արդյունքների հավաստիությունը հիմնավորվում է մշակված ծրագրային համակարգի կիրառմամբ ստացված մի շարք փորձնական արդյունքներով:

Աշխատանքի արդյունքների զեկուցումները

Աշխատանքի արդյունքները զեկուցվել են "From Information Age to Big Data Era", միջազգային գիտաժողովում, Երևան, Հայաստան, հոկտեմբեր 3-5 2016թ., և ՀՀ ԳԱԱ ԻԱՊԻ ընդհանուր սեմինարներում:

Հրատարակումները

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

Աշխատանքի կառուցվածքը և ծավալը

Ատենախոսությունը բաղկացած է ներածությունից, չորս գլուխներից, եզրակացությունից, օգտագործված գրականության ցանկից և մեկ հավելվածից՝

հապավումների բառարանից: Աշխատանքի ընդհանուր ծավալը 105 էջ է՝ ներառյալ 33 պատկեր, 5 աղյուսակ և օգտագործված գրականության 78 հղում:

ԱՇԽԱՏԱՆՔԻ ԲՈՎԱՆԴԱԿՈՒԹՅՈՒՆԸ

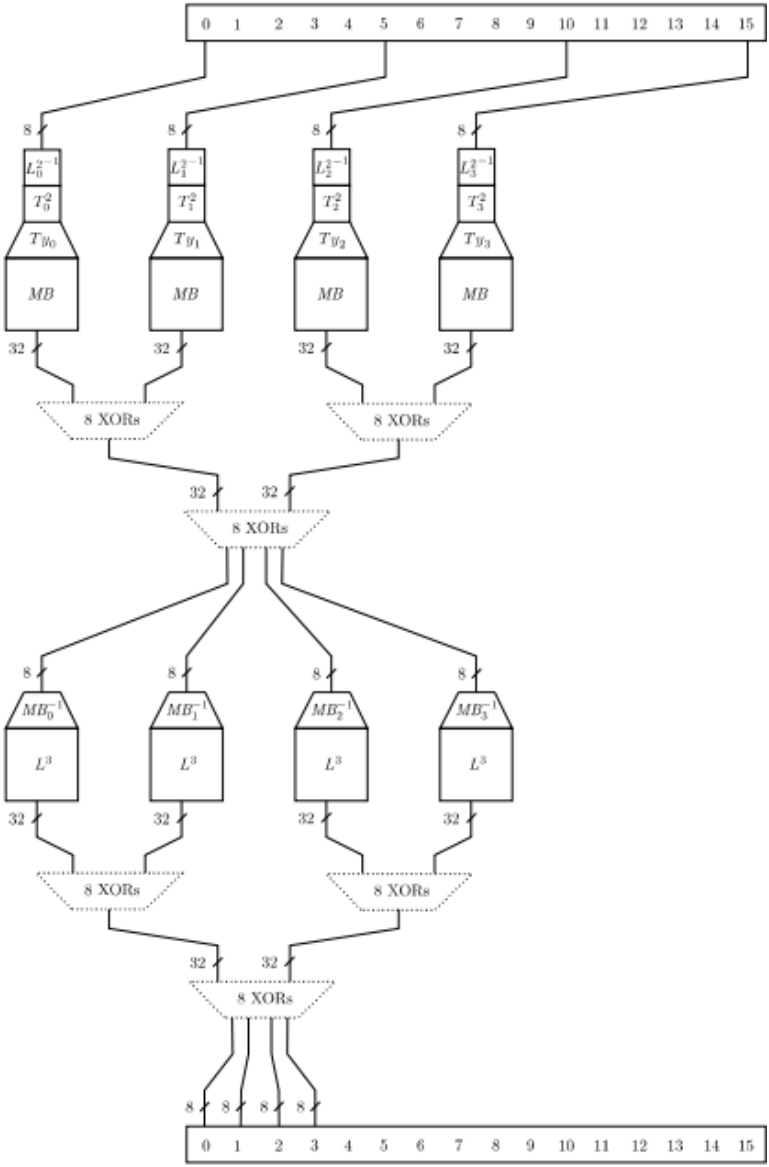
Ներածության մեջ հիմնավորված են թեմայի արդիականությունը, ձևակերպված են աշխատանքի նպատակները, հետազոտման մեթոդները, գիտական նորույթները և հիմնական դրույթները, որոնք ներկայացված են պաշտպանությանը:

Ատենախոսության **առաջին գլխում** կատարված է գոյություն ունեցող թափանցիկ ծածկագրության ալգորիթմների և դրանց նկատմամբ հաջող կիրառված հարձակման մեթոդների ընդհանուր վերլուծություն, բերված են թափանցիկ ալգորիթմների կիրառության օրինակներ:

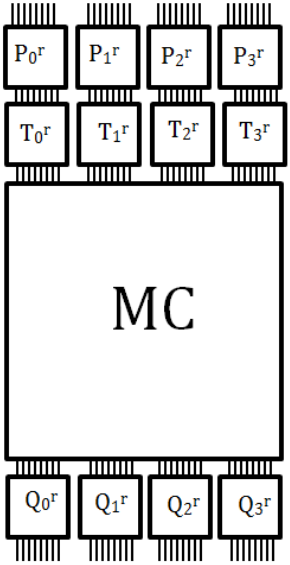
1.1 բաժնում ներկայացված են AES-ի մի շարք թափանցիկ իրականացումներ, որոնք բոլորը հիմնված են Չոի կողմից 2002-ին առաջարկված իրականացման վրա: Չոն դիտարկում է AES ալգորիթմի յուրաքանչյուր փուլ որպես 4 բայթ մուտքով և 4 բայթ ելքով 4 արտապատկերումների խումբ: Անվտանգության ապահովման նպատակով մուտքային և ելքային բայթերի վրա կիրառվում են պատահական տեղափոխություններ, և E_K ծածկագրման ֆունկցիայի փոխարեն հաշվարկվում է $E'_K = IDE * E_K * ODE^{-1}$ ֆունկցիան, որտեղ IDE-ն և ODE-ն համապատասխանաբար մուտքային և ելքային տվյալների վրա կիրառվող պատահական տեղափոխության ֆունկցիաներ են: Չոն իրականացնում է AES ծածկագրի փուլերը 4 տեսակի աղյուսակների միջոցով: Պատկեր 1-ում պատկերված են բոլոր աղյուսակները, ինչպես նաև դրանց միջոցով ծածկագրման 2-րդ փուլի 4 բայթերի ծածկագրման պրոցեսը:

1.1.4 բաժնում բերված է Չոի իրականացման վրա հաջող կիրառված BGE հարձակման նկարագրությունը: BGE հարձակումը Չոի աղյուսակներով AES-ի իրականացումը դիտում է որպես 4 բայթ մուտքով և 4 բայթ ելքով արտապատկերումների խումբ, ինչպես պատկերված է Պատկեր 2-ում: Ինչպես տեսնում ենք, յուրաքանչյուր փուլի յուրաքանչյուր 4 բայթի վրա կիրառվում են P_i^r և Q_i^r մուտքային և ելքային արտապատկերումները, $T_i^r(x)$ ֆունկցիան և MC մատրիցան:
BGE հարձակումը աշխատում է 3 փուլով՝

1. Աղյուսակների հիման վրա վերականգնվում են P_i^r և Q_i^r արտապատկերումների ոչ գծային մասերը, որից հետո կառուցվում են նոր աղյուսակներ, որտեղ P_i^r-ը և Q_i^r-ը փոխարինվում են $\tilde{P}_i^r$ և $\tilde{Q}_i^r$ աֆինական ձևափոխություններով:
2. $\tilde{P}_i^r$ և $\tilde{Q}_i^r$ ձևափոխությունների լիարժեք վերականգնում:
3. AES-ի գաղտնի բանալու արժեքի վերականգնում:



Պատկեր 1: AES ծածկագրի Չոի թափանցիկ իրականման 2-րդ փուլը:



Պատկեր 2: AES ծածկագրի Չոի իրականացումը BGE հարձակման համար:

BGE հարձակմանը հաջորդում են Չոի իրականացման մի քանի զարգացումներ և աղյուսակի արժեքների համընկման վրա հիմնված հարձակումը:
1.1.5, 1.1.6 և 1.1.7 բաժիններում բերված են Չոի ալգորիթմի փոփոխված տարբերակները: 1.1.8 բաժինը նվիրված է Չոի աղյուսակների ելքերի համընկնումների վրա հիմնված հարձակման նկարագրությանը: Հարձակումը դիտարկում է AES-ի փուլը որպես 4 բայթը 4բայթի արտապատկերող ֆունկցիա՝

$$f : (x_0, x_5, x_{10}, x_{15}) \mapsto \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \oplus \begin{pmatrix} S(x_0 \oplus k_0) \\ S(x_5 \oplus k_5) \\ S(x_{10} \oplus k_{10}) \\ S(x_{15} \oplus k_{15}) \end{pmatrix},$$

որտեղ S-ը AES-ի SubBytes ֆունկցիան է: Նշանակենք $E_i = \left(P_{i,0} || P_{i,1}\right) \circ L_i$ և $E_i' = \left(P_{i,0}' || P_{i,1}'\right) \circ L_i'$, որտեղ $P_{i,j}$-ները մուտքային տեղափոխություններն են իսկ L_i-ները Խառնող արտապատկերումները (mixing bijection): Չոի աղյուսակների միջոցով հնարավոր է հաշվել $f' = (E_0'||E_1'||E_2'||E_3') \circ f \circ (E_0^{-1} || E_5^{-1} || E_{10}^{-1} || E_{15}^{-1})$ ֆունկցիան: f_i'-ով նշանակենք f-ի կոորդինատ ֆունկցիաները՝ $f' = (f_0', f_1', f_2', f_3')$, իսկ S_j–ով $S_j(x) = S\left(k_j \oplus E_j^{-1}(x)\right)$ ֆունկցիան: Ալգորիթմի առաջին քայլում վերականգնվում են S_0 և S_5 ֆունկցիաները $f_0'(\alpha, 0,0,0) = f_0'(0, \beta, 0,0)$ տեսակի հանընկնումների հիման վրա,

որից հետևում է $02 \cdot S_0(\alpha) \oplus 03 \cdot S_5(0) = 02 \cdot S_0(0) \oplus 03 \cdot S_5(\beta)$: f_0'-ի ելքերը միշտ ունեն բավարար քանակով համընկնումներ S_0-ի և S_5-ի 512 արժեքների նկատմամբ գծային հավասարումների համակարգ կառուցելու և լուծելու համար: Մյուս $S_j(x)$-երի արժեքները վերականգնվում են նմանատիպ կերպով, որից հետո վերականգնվում են E_i' ֆունկցիաները և գաղտնի բանալին: 1.2 բաժինը նվիրված է թափանցիկ ծածկագրերի կիրառություններին:

Երկրորդ գլխում նկարագրված է SAFER+ ալգորիթմի թափանցիկ իրականացումը: 2.1 բաժնում բերված է Safer+ ալգորիթմի համառոտ նկարագրությունը: Ալգորիթմը բաղկացած է գծային և ոչ գծային շերտերից: Ոչ գծային շերտը նկարագրվում է հետևյալ բանաձևերով՝

$r = 1$ փուլի համար

$$T_i^1(x) := log\big(x + k_{i1}^1\big) \oplus k_{i2}^1, i \in B,$$
$$T_i^1(x) := exp\big(x \oplus k_{i1}^1\big) + k_{i2}^1, i \in A,$$

$2 \le r \le 6$ փուլերի համար

$$T_i^r(x) := exp(x \oplus k_{i1}^r) + k_{i2}^r, i \in A,$$
$$T_i^r(x) := log(x + k_{i1}^r) \oplus k_{i2}^r, i \in B,$$

$r = 7$ փուլի համար

$$T_i^7(x) := x \oplus k_i^{13}, 1 \le i \le 16,$$

որտեղ $A = \{1,4,5,8,9,12,13,16\}$ and $B = \{2,3,6,7,10,11,14,15\}$, exp-ը 45^x ֆունկցիան է, իսկ log-ը $\log_{45}(x)$-ը մնացորդով ըստ 257-ի: Գծային շերտը իրենից ներկայացնում է բայթերի զույգերի բազմապատկում H = $\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ մատրիցայով 4 անգամ, յուրաքանչյուր երկուսի բազմապատկումների միջև Հայկական տեղափոխության կիրառմամբ ([9 ,12 ,13 ,16 ,3 ,2 ,7 ,6 ,11 ,10 ,15 ,14 ,1 ,8 ,5, 4]):

2.2 բաժնում նաև նկարագրված է բանալիների գեներացման նոր ալգորիթմը, որը հիմնված է SHA-256 հեշավորման ֆունկցիայի վրա: Ի տարբերություն SAFER+-ի բանալիների գեներացման ալգորիթմի, ներկայացված ալգորիթմը հակադարձելի չէ, և ունենալով որևէ մեկ փուլի բանալիները հնարավոր չէ հաշվարկել մյուս բոլոր փուլերի բանալիները:

2.3 բաժնում նկարագրված է թափանցիկ ծածկագրման ալգորիթմը: Թափանցիկ իրականացման մեջ աղյուսակների ելքերը բաժանված են 2 մասի, և ոչ գծային շերտը ստանում է հետևյալ տեսքը՝

r=1 փուլի համար

$$T_{i,1}^1(x) := exp\big(\mathbf{IP}_i(x) \oplus k_{i1}^1\big) + k_{i2}^1 + R_i^1,\ i \in A,$$
$$T_{i,1}^1(x) := log\big(\mathbf{IP}_i(x) + k_{i1}^1\big) \oplus k_{i2}^1 + R_i^1(x),\ i \in B,$$
$$T_{i,2}^1(x) := -R_{i1}^1(x),\ 1 \le i \le 16,$$

$2 \le r \le 6$ փուլերի համար

$$T^r_{i,1}(x_1, x_2) := exp((x_1 + x_2 - S_i) \oplus k^r_{i1}) + k^r_{i2} + R^r_i(x_1, x_2),\ i \in A,$$

$$T^r_{i,1}(x_1, x_2) := log(x_1 + x_2 - S_i + k^r_{i1}) \oplus k^r_{i2} + R^r_i,\ i \in B,$$

$$T^r_{i,2}(x_1, x_2) := -R^r_i(x_1, x_2), 1 \le i \le 16,$$

$r = 7$ փուլի համար

$$T^7_i(x_1, x_2) := \mathbf{OP}_i((x_1 + x_2 - S_i) \oplus k^{13}_i), 1 \le i \le 16,$$

որտեղ $R^1_i(x)$ և $R^r_i(x_1, x_2)$ պատահական ֆունկցիաներ են, գեներացված բանալիներից անկախ: Լրացուցիչ պաշտպանվածության նպատակով բոլոր աղյուսակների մուտքային և ելքային բայթերի վրա կիրառվում են պատահական արտապատկերումներ $\mathrm{f}^r_1, \mathrm{f}^r_2, \dots, \mathrm{f}^r_{32},\ \mathrm{f}^r_i : Z_{256} \to Z_{256}$, և $\mathrm{g}^r_1, \mathrm{g}^r_2, \dots, \mathrm{g}^r_{32},\ \mathrm{g}^r_i : Z_{256} \to Z_{256}$ այնպես, որ յուրաքանչյուր հաջորդ աղյուսակի մուտքին կիրառվում է նախորդ աղյուսակի ելքի վրա կիրառված արտապատկերման հակադարձը: Այսպիսով այս արտապատկերումները փոխադարձաբար ոչնչացնում են միմյանց և որևէ կերպով չեն ազդում ալգորիթմի աշխատանքի վրա:

Այսպիսով թափանցիկ ալգորիթմը կիրառում է 2 տեսակի աղյուսակներ, $E - box$ և $2 - PHT$. $E - box$-երը նկարագրվում են հետևյալ բանաձևերով՝

$r = 1$ փուլի համար

$$E^1_{i1}(x) \stackrel{\text{def}}{=} \mathrm{f}^r_{2*i-1}\left(T^1_{i1}(x)\right),$$
$$E^1_{i2}(x_1, x_2) \stackrel{\text{def}}{=} \mathrm{f}^r_{2*i}\left(T^1_{i2}(x)\right),$$

$2 \le r \le 6$ փուլերի համար

$$E^r_{i1}(x_1, x_2) \stackrel{\text{def}}{=} \mathrm{f}^r_{2*i-1}\left(T^r_{i1}\left({\mathrm{g}^r_{2*i-1}}^{-1}(x_1), {\mathrm{g}^r_{2*i}}^{-1}(x_2)\right)\right),$$
$$E^r_{i2}(x_1, x_2) \stackrel{\text{def}}{=} \mathrm{f}^r_{2*i}\left(T^r_{i2}\left({\mathrm{g}^r_{2*i-1}}^{-1}(x_1), {\mathrm{g}^r_{2*i}}^{-1}(x_2)\right)\right),$$

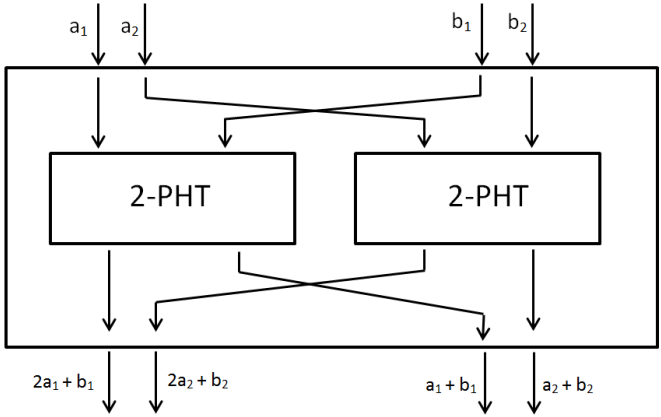
$r = 7$ փուլի համար

$$E^1_i(x_1, x_2) = T^7_i\left({\mathrm{g}^r_{2*i-1}}^{-1}(x_1), {\mathrm{g}^r_{2*i}}^{-1}(x_2)\right).$$

Պատկեր 3-ում երևում է, թե ինչպես կարելի է կառուցել 4 բայթանի 2-PHT աղյուսակ 2 հատ 2 բայթանի աղյուսակների միջոցով: 2 բայթանի $2 - PHT$ աղյուսակները կառուցվում են հետևյալ բանաձևերով՝

$$2 - \mathrm{PHT}_1(x_1, x_2) = \mathrm{g}_1\left(2 * f_1^{-1}(x_1) + f_2^{-1}(x_2) + S^{rl}_B\right),$$

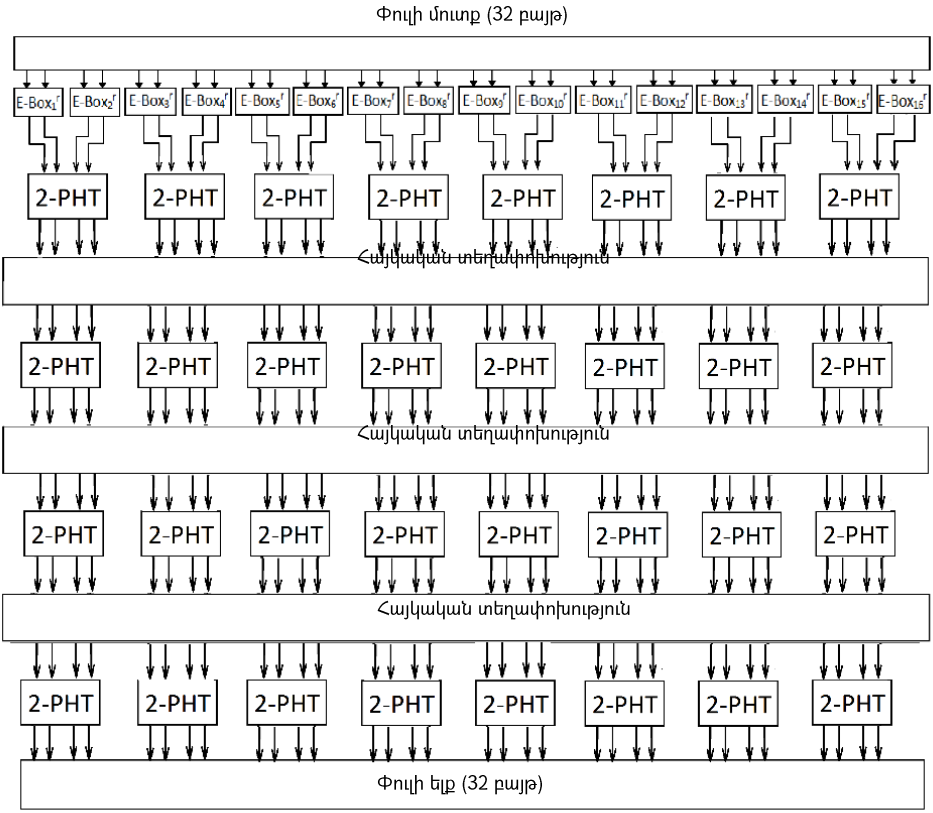
$$2 - \mathrm{PHT}_2(x_1, x_2) = \mathrm{g}_2\left(f_1^{-1}(x_1) + f_2^{-1}(x_2) + S^{rl}_{B+1}\right).$$



Պատկեր 3: 4 բայթանի 2-PHT աղյուսակի կառուցումը 2 հատ 2 բայթանի աղյուսակների միջոցով:

SAFER+ ալգորիթմի թափանցիկ ծածկագրման փուլը պատկերված է Պատկեր 4-ում: Այսպիսի տեսք ունեն բոլոր փուլերը բացառությամբ 1-ինի, որտեղ փուլի մուտքը 16 բայթ է:

2.4 բաժնում նկարագրված է ապածածկագրման ալգորիթմը: 2.5 բաժնում բերված է ալգորիթմի անվտանգության անալիզը: Նկարագրված են մի շարք հարձակումներ, որոնք չեն կարող կիրառվել ներկայացված իրականացման նկատմամբ: 2.5.1 բաժնում բերված են օգտագործվող աղյուսակների բազմազանության և երկիմաստության արժեքները, որոնք բերված են Աղյուսակ 1-ում: Ի տարբերություն բազմազանության, որը կարող է ուղղակիորեն հաշվվել, գոյություն չունի մեթոդ երկիմաստության հաշվարկի համար, ուստի բերված արժեքները ստորին գնահատականներ են: 2.5.2 բաժնում ներկայացված է պաշտպանությունը BGE հարձակումից: 2.5.3 բաժնում քննարկվում է 2-PHT աղյուսակների բազմանկի օգտագործման անհնարինությունը: Աղյուսակների բազմակի օգտագործումը ծածկագրման տարբեր փուլերում կարող էր նպաստել աղյուսակների ընդհանուր ծավալի փոքրացմանը, սակայն ատենախոսության մեջ բերված է հարձակում կրկնվող 2-PHT աղյուսակներով իրականացման վրա: 2.5.4 և 2.5.5 բաժինները նվիրված են համապատասխանաբար 2-PHT և E-box աղյուսակների ելքերի հանընկնումների վրա հիմնված հնարավոր հարձակումներին (collision attacks): 2.5.6 բաժնում ցույց է տրված, որ հավանականային հարձակումներ հնարավոր չեն ի շնորհիվ նրա, որ E-box և 2-PHT աղյուսակների ելքերը հավասարահավանական են: 2.6 բաժնում նկարագրված է ալգորիթմի C++ իրականացումը: Բերված են հիմնական ինտերֆեյսային դասերի և դրանց հիմնական մեթոդների նկարագրությունները:

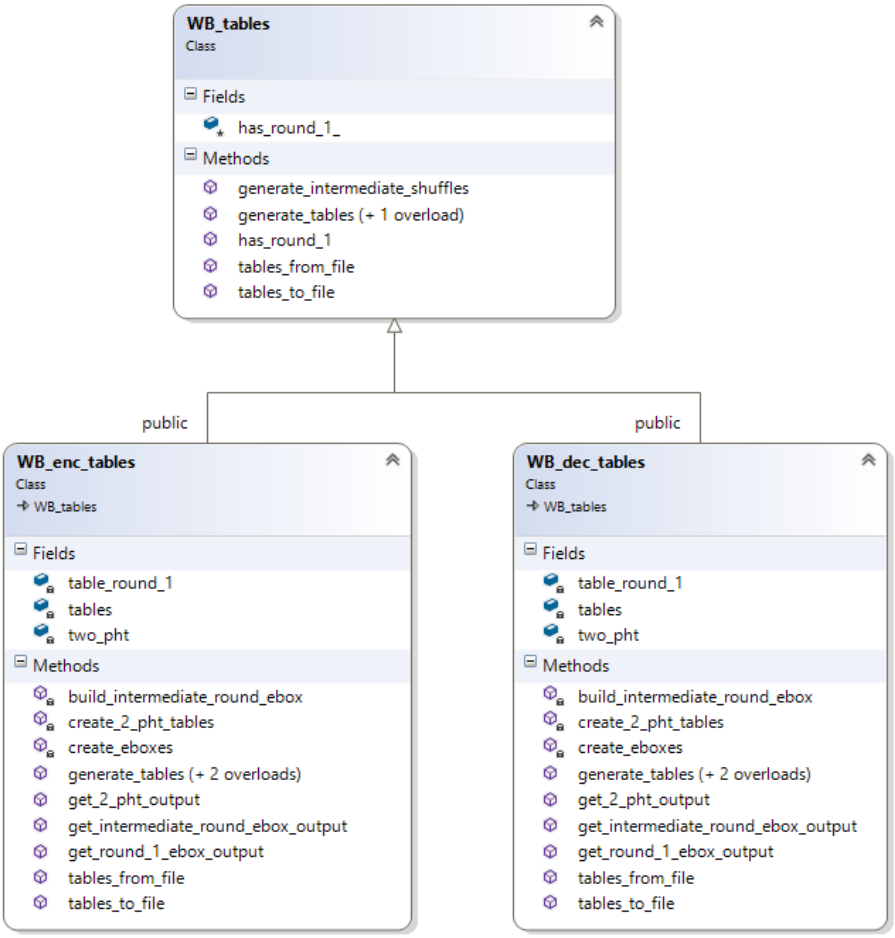


Պատկեր 4: SAFER+ ալգորիթմի թափանցիկ ծածկագրման փուլը:

Աղյուսակ 1: Safer+ թափանցիկ աղյուսակների և Չոի աղյուսակների բազմազանության և երկիմաստության գնահատականները:

	բազմազանություն	երկիմաստություն
$E-box$	2^{72288}	2^{68920}
$2-PHT$	2^{1692}	2^{3376}
Չոի 1-ին տիպի	2^{2420}	2^{546}
Չոի 2-րդ տիպի	2^{769}	2^{129}
Չոի 3-րդ տիպի	2^{699}	2^{117}
Չոի 4-րդ տիպի	2^{133}	2^{48}

Պատկեր 5-ում բերված են “WB_enc_tables” և “WB_dec_tables” դասերը իրենց հիմնական ֆունկցիաներով, որոնք իրականացնում են գաղտնի բանալիների հիման վրա թափանցիկ աղյուսակների ստեղծման, պահման և օգտագործման ֆունկցիաները:



Պատկեր 5: Թափանցիկ աղյուսակների կառուցման և օգտագործման դասերը:

Աղյուսակ 2: Արագագործության չափումները միլիվայրկյաններով:

	Ծածկագրում	Ապածածկագրում
Թափանցիկ Safer+	$6.5 * 10^{-5}$	$6.7 * 10^{-5}$
«Սև տուփի» Safer+	$1.02 * 10^{-5}$	$1.07 * 10^{-5}$
RSA-2048	0.12	5.46
RSA-4096	0.64	16.1

2.7 բաժնում բերված են ալգորիթմի արագագործության չափումները: Համեմատության համար բերված են նաև «Սև տուփի» Safer+ իրականցման, և 2048 և 4096 բիթանի RSA ալգորիթմի արագությունները: Աղյուսակ 2-ից երևում է, որ թափանցիկ իրականացումը դանդաղ է սև տուփի իրականացումից մոտ 6.5 անգամ, ինչը սակայն թույլ է տալիս մեզ օգտագործել այլ այնպիսի սարքերի վրա, որտեղ սև տուփի ալգորիթմը անվտանգ չէ: Մասնավորապես, ստացված 5.2 ՄԲ/վրկ արագությունը բավարար է առցանց վիդեո վարձույթի համար: 2.8 բաժնում բերված են թափանցիկ աղյուսակների չափսերը, որոնք ներկայացված են աղյուսակ 3-ում: 2.9-ում նշված են թե աղյուսակներից որոնք են հաստատուն (static), իսկ որոնք բանալուց կախված(dynamic): Որոշ համակարգեր օգտագործում են բանալու պարբերական թարմացումներ: Օրինակ, վիդեո ֆայլերի վարձույթները կարող են ֆայլի մասերը ծածկագրել տարբեր բանալիներով, այսպիսով դժվարացնելով անբողջական ֆայլի ապածածկագրումը: Այս նպատակով գեներացվում են մի քանի թափանցիկ աղյուսակներ, տարբեր բանալիներով, սակայն թարմացվում են միայն այն աղյուսակները, որոնք բանալի են պարունակում: 2.10 բաժինը 2-րդ գլխի ամփոփումն է:

Աղյուսակ 3: Թափանցիկ աղյուսակների չափերը և անհրաժեշտ քանակը:

Աղյուսակի տեսակը	Չափսը	անհրաժեշտ քանակը	Ընդհանուր հիշողությունը
1-ին փուլի $E-box$	512 բայթ	16	8 KB
2-6 փուլերի $E-box$	128 ԿԲ	80	10 MB
7-րդ փուլի $E-box$	64 ԿԲ	16	1 MB
$2-PHT$	128 ԿԲ	320	40 MB

Աշխատանքի **երրորդ գլուխը** նվիրված է տեղափոխության բազմանդամների վրա հիմնված բաց բանալիով թափանցիկ ծածկագրության համակարգին:
$f(x)$ բազմանդամը կկոչվի տեղափոխության բազմանդամ, եթե $x \mapsto f(x)$ արտապատկերումը փոխմիարժեք է: $GF(q)$ դաշտում $f^{-1}(x)$ բազանդամը $f(x)$-ի հակադարձն է, եթե $f(f^{-1}(x)) = f^{-1}(f(x)) = x$:

$F(x) = \sum_{u=0}^{n} a_u x^u \in GF(2)$ բազմանդանդամի համար $P(x) = \sum_{u=0}^{n} a_u x^{2^u} \in GF(2^n)$ կկոչենք դրա գծային 2-ասոցացված բազմանդամ: $GF(2^n)$-ում գործողությունները կատարվում են մոդուլով ըստ $g(x) \in GF(2)$ պրիմիտիվ բազմանդամի:

3.1 բաժինը նվիրված է Խաչատրյան-Կյուրեղյան կրիպտոհամակարգի համառոտ նկարագրությանը: Խաչատրյանի և Կյուրեղյանի հոդվածում ներկայացվում է $P(x)$-ի հակադարձի՝ $P^{-1}(x)$-ի հաշվման ալգորիթմը:

Համակարգը գործում է հետևյալ կերպ՝

- Բանալու ստեղծման նպատակով, ունենալով $g(x), L(x), L_i(x)$ գաղտնի բազմանդամները հաշվվում են հետևյալ արժեքները՝
  - $R_N(x) = x^N \bmod g(x)$, որտեղ $N = 2^i(2k+1), i = \overline{1,128}, r = 2k+1, k = \overline{0,63}$:
  - $B_N(x) = \left(R_N(x) \cdot L_0(x) \bmod L(x)\right) \oplus L_{k+1}(x)$, որտեղ $N = 2^i(2k+1)$, $i = \overline{1,128}, r = 2k+1, k = \overline{0,63}$:
- $m(x)$ բազմանդամի ծածկագրման համար հաշվվում է $P\big(m(x)\big) = c(x) \bmod g(x)$, որից հետո թափանցիկ հաշվվում են հետևյալ արժեքները, որոնք միասին կազմում են ծածկագիրը՝
  - $R(x)$ - $c(x)$-ի 128-ից փոքր անդամները:
  - $\sum B_N(x)$ - $c(x)$-ի 127-ից մեծ անդամների համապատասխան $B_N(x)$ արժեքների գումարը:
  - B= $(b_0, b_1, .., b_{63})$, որտեղ $b_k = 1$, եթե $c(x)$-ի 127-ից մեծ անդամների մեջ $N = 2^i(2k+1)$ տեսքի անդամների քանակը կենտ է և 0 հակառակ դեպքում:
- $m(x)$ բազմանդամի ապածածկագրման համար ($R(x)$, $\sum B_N(x)$, B) արժեքների հիման վրա հաշվվում է

$$R(x) \oplus \left(\sum B_N(x) \oplus \sum_{i=1}^{128} b_i L_i(x) \right) \cdot L_0(x)^{-1} = c(x) \bmod L(x),$$

որից հետո $P^{-1}\big(c(x)\big) = P^{-1}\left(P\big(m(x)\big)\right) = m(x) \bmod g(x)$.

3.2 բաժնում նկարագրված են ալգորիթմի լավացումները: Նկարագրված համակարգի օգտագործման դեպքում, $P(x)$-ի որոշակի արժեքների դեպքում հարձակվողը, $R(x)$-ի արժեքի հիման վրա կարող է ինֆորմացիա քաղել $m(x)$-ի վերաբերյալ: Դա կանխելու նպատակով կատարվել են հետևյալ փոփոխությունները՝

- Ծածկագրության ժամանակ հաշվվում է $P(m(x) * x^{128}) = c(x) \bmod g(x)$ բազմանդամը, որից հետո նույն ձևով կառուցվում է ($\sum B_N(x)$, B) զույգը:
- Ապածածկագրության ժամանակ ($\sum B_N(x)$, B) զույգի հիման վրա վերականգնվում է $c(x)$-ի արժեքը, որից հետո հաշվվում է $P^{-1}\big(c(x) * x^{2^{128-8}}\big) = m(x) \bmod g(x)$.

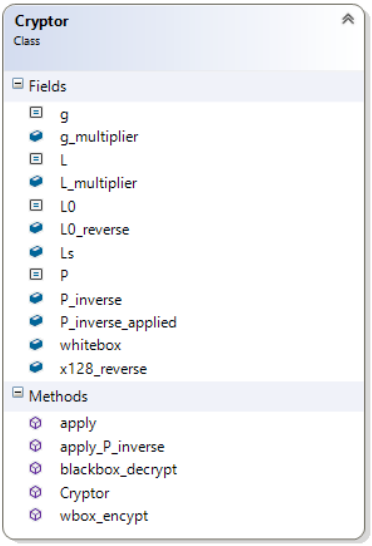
Կարելի է նկարտել, որ նոր համակարգում $R(x)$ բազմանդամի կարիք չկա, քանի որ $c(x)$-ը չունի 128-ից փոքր անդամներ, որի արդյունքում նվազում է ծածկագրի չափսը, սակայն $B_N(x)$ արժեքները հաշվվում են $k = \overline{0,127}$-ի համար, հետևաբար աճում է թափանցիկ աղյուսակների չափսը:

3.3 բաժնում բերված է փոփոխված համակարգի կիրառման թվային օրինակ, որը կարող է օգտագործվել ծրագրային ապահովման թեսթավորման համար:
3.4 բաժինը նվիրված է ալգորիթմի արագագործության բարելավման նպատակով իրականացված փոփոխություններին: Եթե |P(x)|-ով նշանակենք P(x)-ի անդամների քանակը, ապա $P(m(x)\cdot x^{128})$-ը կունենա $|P(x)|*|m(x)|$ անդամ, ուստի $\sum B_N(x)$-ի հաշվարկման համար կպահանջվի $|P(x)|*|m(x)|$ հատ բազմանդամային գումարում, և ևս $|P(x)|*|m(x)|$ գործողություն կպահանջվի B վեկտորի հաշվարկման համար: $|B|$-ով նշանակենք B վեկտորում 1-երի քանակը: Ապածածկագրման գործողության ժամանակ $c(x)$-ի հաշվարկը կպահանջի $|B|$ հատ բազմանդամների գումարում $L_i(x)$-ի արժեքների գումարման համար, և մեկ բազմապատկում $L_0(x)^{-1}$-ով, եթե $L_0(x)^{-1}$-ը նախապես հաշվվել և պահվել է: Ապա $P^{-1}(c(x))\cdot x^{2^{120}} = m(x)\ mod\ g(x)$-ի հաշվարկի համար պետք է հաշվել $c(x)^{2^i}$-ը բոլոր $i=\overline{0,127}$-ի համար, ապա միմյանցով բազմապատկել անհրաժեշտ արժեքները, որի համար կպահանջվի $128+|P^{-1}(x)|$ բազմանդամային բազմապատկման գործողություն: Կարելի է նկատել, որ $P(x)=\sum_{u=0}^{n} a_u x^{2^u}$ և $m(x)=\sum_{v=0}^{n} a_v x^v$ բազմանդամների համար $P(m(x))=\sum_{v=0}^{n} a_v P(x^v)$: Նման կերպով, եթե $c(x)=\sum_{l=0}^{n} a_l x^l$, ապա $P^{-1}(c(x))=\sum_{l=0}^{n} a_l P^{-1}(x^l)$:
Նշանակենք $B_{P(x^v)}(x)$-ով $B_N(x)$ բազմանդամների գումարը, որտեղ N-ը $P(x^v)$-ի որևէ թերմի աստիճանն է: Կարելի է նկատել, որ $m(x)=\sum_{v=0}^{n} a_v x^v$ բազմանդամի ծածկագրման ընթացքում հաշվվող $\sum B_N(x)$-ը կարելի ներկայացնել հետևյալ տեսքով՝

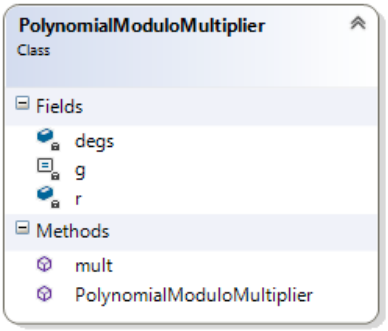
$$\sum B_N(x)=\sum_{v=0}^{n} a_v B_{P(x^v)}(x):$$

Սա նշանակում է, որ $v=\overline{0,127}$-ի համար $B_{P(x^v)}(x)$ արժեքները նախապես հաշվելու դեպքում, $\sum B_N(x)$-ի հաշվարկի համար կպահանջվի ընդամենը $|m(x)|$ գործողություն: Յուրաքանչյուր $P(x^v)$-ի ազդեցությունը B վեկտորի վրա նույնպես կարող է նախապես հաշվարկվել, որի արդյունքում B վեկտորի արժեքի հաշվարկը նույնպես կպահանջի $|m(x)|$ գործողություն: Ապածածկագրման ժամանակ $P^{-1}(x^l)$-ի արժեքների նախապես հաշվումը բոլոր $l=\overline{0,127}$-ի համար թույլ է տալիս հաշվել $P^{-1}(c(x))$-ը ընդամենը $|c(x)|$ գործողությամբ:
3.5 բաժնում բերված է ալգորիթմի անվտանգության անալիզը, մասնավորապես մի շարք հնարավոր հարձակումների նկարագրությունը: Ցույց է տրված, որ ունենալով $B_N(x)$ բաղմանդամների արժեքները հնարավոր չէ հաշվել գաղտնի $g(x)$, $L(x)$ կամ $L_i(x)$ բազմանդամների արժեքները: Ցույց է տրված, որ կամայական $B_N(x)$ և $g(x)$-ի համար հնարավոր է ընտրել $L(x)$ և $L_i(x)$ այնպես, որ ստացված արժեքներով կառուցված $B_N(x)$-երը համապատասխանեն տրված արժեքներին: Այսպիսով անհնար է ունենալով $B_N(x)$ բազմանդամները գտնել $g(x)$-ի արժեքը:
3.6 բաժնում նկարագրված է ալգորիթմի C++ իրականացումը: Պատկեր 6-ում բերված է հիմնական ինտերֆեյսային “Cryptor” դասը, որը տրամադրում է թափանցիկ աղյուսակների ստեղծման, ծածկագրման և ապածածկագրման ֆունկցիաները: Պատկեր 7-ում բերված “PolynomialModuloMultiplier” դասը կիրառվում է օպտիմալ

կերպով բազամանդամների բազմապատկման համար, մոդուլով ըստ նախապես որոշված $g(x)$ բազմանդամի, որոշ նախապես հաշվված արժեքների միջոցով:



Պատկեր 6: “Cryptor” դասը:



Պատկեր 7: “PolynomialModuloMultiplier” դասը:

3.7 բաժնում բերված են հին և նոր համակարգերի և մրցակից RSA համակարգի արագագործության տվյալները: Ինչպես երևում է աղյուսակ 4-ից, մի շարք օպտիմիզացիաներից հետո ալգորիթմի ծածկագրման և ապածածկագրման գործողությունները արագացել են մոտ 2.9 անգամ համեմատած սկզբնական Խաչատրյան-Կյուրեղյան համակարգի հետ: Փոփոխված համակարգը նաև մոտ 3.75

անգամ արագ է RSA-2048 ալգորիթմից ծածկագրման և 133 անգամ ապածածկագրման գործողությունների վրա: 3.8 բաժինը երրորդ գլխի ամփոփումն է:

Աղյուսակ 4: Արագագործության չափումները միլիվայրկյաններով:

	Ծածկագրում	Ապածածկագրում
Խաչատրյան-Կյուրեղյան	0.094	0.12
Փոփոխված ալգորիթմ	0.032	0.041
RSA-2048	0.12	5.46
RSA-4096	0.64	16.1

4-րդ գլխում բերված է աշխատանքի հիմնական արդյունք հանդիսացող երկու թափանցիկ ալգորիթմների «Ուանքրփթոր» ընկերության ծածկագրված ամպային տվյալների վրա փնտրման համակարգին ինտեգրման մանրամասները: Նկարագրված են համակարգի հիմնական գործողությունները, որոնք են հաշվի ստեղծումը, ֆայլի ավելացումը, ջնջումը, փնտրումը ավելացված ֆայլերի վրա և ֆայլի կիսումը այլ օգտատերերի հետ: Safer+ի թափանցիկ իրականացումը կիրառվում է ֆայլի ավելացման և փնտրման գործողություններում՝ Պատկեր 7-ում պատկերված ուղիղ և հակադարձ ինդեքսների կառուցման և դրանց հիման վրա փնտրման համար, իսկ բաց բանալիով ծածկագրման ալգորիթմը՝ այլ օգտատերերի հետ ֆայլով կիսվելու գործողության մեջ:

f_{id}	Ծածկագրված հիմնաբառեր
1	$E_{WB}(w_1), E_{WB}(w_8), E_{WB}(w_9)$
2	$E_{WB}(w_1), E_{WB}(w_6), E_{WB}(w_4)$
...	...
n	$E_{WB}(w_1), E_{WB}(w_2)$

(a) Ուղիղ ինդեքս

Ծածկագրված հիմնաբառեր	f_{id}-ների ցուցակ
$E_{WB}(w_1)$	$1,2,n$
$E_{WB}(w_2)$	n
...	...
$E_{WB}(w_m)$	$3,15$

(b) Հակադարձ ինդեքս

Պատկեր 8: Ուղիղ և հակադարձ ինդեքսները:

Ինչպես երևում է Պատկեր 8-ից, ուղիղ ինդեքսը թույլ է տալիս տրված համարի ֆայլի համար գտնել ֆայլում եղած հիմնաբառերի ծածկագրերը, իսկ հակադարձ ինդեքսը՝ յուրաքանչյուր ծածկագրված հիմնաբառի համար գտնել այն ֆայլերը, որոնցում կա տվյալ բառը: Այսպիսով, հակադարձ ինդեքսը օգտագործվում է փնտրման, իսկ ուղիղ ինդեքսը ֆայլի հեռացման ժամանակ: Թափանցիկ ծածկագրերի կիրառումը

ինդեքսների կառուցման մեջ թույլ է տալիս ապահովել համակարգի անհրաժեշտ արագագործությունը:

ԱՇԽԱՏԱՆՔԻ ՀԻՄՆԱԿԱՆ ԱՐԴՅՈՒՆՔՆԵՐԸ

1. Ուսումնասիրվել են գոյություն ունեցող թափանցիկ ծածկագրության համակարգերը և ստեղծվել է SAFER+ սիմետրիկ ծածկագրության ալգորիթմի նոր թափանցիկ իրականացում, որը անվտանգ է այլ թափանցիկ ծածկագրության ալգորիթմների նկատմամբ հաջողությամբ կիրառված հարձակման մեթոդների նկատմամբ [1-2]:

2. Բարելավվել է տեղափոխության բազմանդամների վրա հիմնված բաց բանալիով թափանցիկ ծածկագրության Խաչատրյան-Կյուրեղյան համակարգի անվտանգությունը և արագագործությունը [3]:

3. Իրականցվել են վերը նշված 2 թափանցիկ ծածկագրության սխեմաները իրականացնող գրադարաններ C++ լեզվով, որոնք ինտեգրվել են «Ուանքրփթոր» ընկերության ծածկագրված ամպային տվյալների վրա փնտրման համակարգին:

ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ԹԵՄԱՅՈՎ ՏՊԱԳՐՎԱԾ ՀՈԴՎԱԾՆԵՐԸ

[1] M. Karapetyan, "Review of White-box Implementations of AES Block," *Mathematical Problems of Computer Science 46,* p. 26–36, 2016.

[2] Gurgen Khachatrian, Martun Karapetyan, "White-box Encryption Algorithm based on SAFER+," in proceedings of international workshop on Information theory and Data Science, "From Information Age to Big Data Era", Yerevan, Armenia, October 3-5, pp. 77-88, 2016.

[3] Gurgen Khachatrian, Martun Karapetyan, "On a public key encryption algorithm based on Permutation Polynomials and performance analyses," *International Journal Information Theories and Applications (ITHEA), Vol. 23, Number 1, pp.34-38,* 2016.

Martun Karapetyan

White-box cryptography and applications

ABSTRACT

Conventional encryption algorithms are designed to be secure in the "black-box" attack context, i.e. the attacker has access to the input and output of the encryption algorithm, but cannot observe the intermediate values generated during the software execution.

Yet in some cases, the encryption algorithm runs in a hostile environment, where the attacker can see not only the input and output values but also has full access to all the intermediate values generated during the software execution and can change the execution at will. This means that an attacker can easily analyze the binary code of the application, and the corresponding memory pages during execution; the attacker can intercept system calls, tamper with the binary and its execution; and use any kind of attack tool such as IDA Pro, debuggers, disassemblers, emulators, etc. Any cryptographic software running on these type of devices is operating in the white-box attack context, and conventional black-box ciphers don't provide the necessary level of security for these devices.

White-box cryptography algorithms are based on the black-box algorithms, but instead of directly using the secure cryptographic key, the encryption algorithm uses special look-up tables which are built based on the key. All the operations are performed using these tables in such a clever way, that the attacker having access to all the tables and being able to observe the process of each encryption operation and even modify it at will, is unable to extract the cryptographic key used for the construction of the tables.

White-box cryptography is used in several commercial products such as Digital Rights management applications, Oblivious transfer protocols, search over encrypted data and as a fast alternative to public key cryptography.

The first white-box implementation of AES block cipher was presented by Chow, Eisen, Johnson and van Oorschot in 2002, which was effectively attacked by the BGE attack presented by Billet, Gilbert and Ech-Chatbi in 2004. Later several white-box implementations of AES block cipher were presented, but all those ciphers were effectively attacked by different modifications of BGE attack, and attacks based on collisions in the output values of the tables. This creates base for research to build a secure white-box implementation of a block cipher.

The main objective is to design a secure white-box implementation of SAFER+ block cipher, resistant to modifications of the known attacks and to improve security and performance for public key encryption scheme designed by G. Khachatryan and M. Kyureghyan.

Scientific Novelty

- A novel white-box algorithm was created based on Safer+ block cipher [2]. The general security analyses of the cryptosystem were presented, including comments regarding resistance to the widely successful BGE attack.
- The public key encryption algorithm based on permutation polynomials was modified for improved security and performance. The algorithm uses white-box reduction for encryption operation [3].

Integration: Based on the results of the research a white-box algorithm based on Safer+ and another white-box algorithm based on permutation polynomials were implemented and integrated into SkyCryptor's secure search engine.

Practical Significance of the Results

The Safer+ white-box presented in this dissertation can be used in different digital right management systems, including video-on-demand services and applications requiring distribution control for protected content playback. It can also be used in secure search schemes and in oblivious transfer protocols, which are playing an essential role in secure multiparty computation protocols. The improved public key encryption scheme based on permutation polynomials can replace conventional public key systems like RSA, Diffie-Helman or Elliptic curve cryptosystem, resulting to significant performance improvements.

Provisions Presented to the Defense

- A novel white-box algorithm based on SAFER+ block cipher, secure against the widely successful BGE attack [2].
- An improved public key encryption scheme based on permutation polynomials with white-box reduction on encryption procedure [3].
- C++ libraries which implement the novel SAFER+ white-box and the new public key system based on permutation polynomials.

Карапетян Мартун Микаелович
Прозрачные криптосистемы и их применения

РЕЗЮМЕ

Стандартные криптограгические алгоритмы рассчитаны на примение в защищеной среде в контексте “черного ящика” где атакующий видит входные и выходные данные алгоритма но не имеет доступа к промежуточным значениям генерируемым алгоритмом. Но в некоторых случаях алгоритм шифрования работает в незащищеной среде, где атакующий может использовать различные дебагеры, дизасемблеры и эмуляторы для чтения страниц памяти устройства или иземенения програмного кода алгоритма. Таким образом атакующий получает доступ ко всем промежуточным значениям генерируемыми алгоритмом. Стандартные алгоритмы шифрования не расчитаны для работы на таких устройствах, что привело к созданию прозрачный криптосистем. Прозрачные алгоритмы шифрования создаются на основе стандартных алгоритмов, но вместо ключа используются так называемые прозрачные таблицы, которые создаются на основе ключа. Таблицы создаются специальным способом, чтобы атакующий, имеющий полный доступ как к самим таблицам, так и ко всем значениям, создающимся во время работы шифрующего алгоритма, не может извлечь какую либо информацию о защищеном ключе.

Впервые прозрачный алгоритм шифрования основаный на симетричном алгоритме AES был создан в 2002 году, который был успешно атакован в 2004 году атакой BGE, которая использует группу таблиц, которые вместе формируют одну стаднию шифрования AES. В последствии были созданы различные модификации алгоритма, но к сожалению все они были успешно атакованы различными модификациями атаки BGE, а также более эфективным и простым методом подбора совпадений среди выходных значений некоторых прозрачных таблиц.

Целью дистертации является разработка прозрачного алгоритма шифрования основаного на симетричном алгоритме SAFER+ а так же усовершенствование безопасности и повышение скорости работы алгоритма шифрования с открытым ключом основаном на так называемых перестановочных многочленах созданый Г. Хачатряном и М. Кюрегяном.

Научная новизна

- Разработан новый прозрачный алгоритм шифрования основаный на симетричном алгоритме SAFER+.

- Была усовершенствована безопасность и повышена эфективность работы алгоритма шифрования с открытым ключом созданый Г. Хачатряном и М. Кюрегяном.

Внедрение

Созданые алгоритмы были недрены в систему SkyCryptor, предназначенного для хранения зашифрованых данный в облачных хранилищах Dropbox и Google Drive, с возможностью поиска и деления дайлами с другими пользователями.

Практическая значимость полученных результатов

Разработаный алгоритм прозрачного шифрования основаный на симетричном алгоритме SAFER+ может быть использован в цифровых системах управления правами, таких как онлайн видеопрокат и системы контролирования доступа к защищеным видео файлам. Такими системами являются ITunes, Amazon Videos, которые не могут существовать без алгоритмов прозрачного шифрования. Другим важным применением алгоритма являются протоколы секретной передачи, которые играют важнейщую роль в протоколах удаленного вычисления функций. Обычно протоколы секретной передачи были основаны на алгоритмах шифрования открытым ключом, замена которых на прозрачный алгоритм увеличила скорость передачи, сделав возможмым практическое применение протоколов.

С увеличением количества данных хранящихся на облачных хранилищах, все больше и больше компаний нуждаются в системе поиска на зашифрованых файлах. Прозрачные алгоритмы шифрования являются важной частью таких систем, и разработаный алгоритм прозрачного шифрования основаный на SAFER+ может применяться в таких системах. Также важно иметь функцию деления зашифрованым файлом с другими пользователями, где может использоваться усовершенстовованый прозрачный алгоритм с открытым ключом. Усовершенстовованый прозрачный алгоритм с открытым ключом также может служить как быстрая альтернатива стандартным алгоритмам шифрования с открытым ключом.

Положения, выносимые на защиту

- Новый прозрачный алгоритм шифрования основаный на симетричном алгоритме SAFER+ [2].
- Усовершенствована версия алгоритма шифрования с открытым ключом основаном на так называемых перестановочных многочленах созданый Г. Хачатряном и М. Кюрегяном [3].
- C++ библиотеки реализующие созданые прозрачные алгоритмы.