ՀՀ ԳԱԱ ԻՆՖՈՐՄԱՏԻԿԱՅԻ ԵՎ ԱՎՏՈՄԱՏԱՑՄԱՆ ՊՐՈԲԼԵՄՆԵՐԻ ԻՆՍՏԻՏՈՒՏ

Միհրան Մերուժանի Հովսեփյան

ԳԱՂՏՆԱԳՐՎԱԾ ՏՎՅԱԼՆԵՐՈՒՄ ԱՆՎՏԱՆԳ ՈՐՈՆՈՒՄ ԻՐԱԿԱՆԱՑՆԵԼՈՒ ԱԼԳՈՐԻԹՄՆԵՐԻ ՄՇԱԿՈՒՄ

Ե13.05 – «Մաթեմատիկական մոդելավորում, թվային մեթոդներ և ծրագրերի համալիրներ» մասնագիտությամբ տեխնիկական գիտությունների թեկնածուի գիտական աստիճանի հայցման ատենախոսության

ՍԵՂՄԱԳԻՐ

Երևան – 2016

ИНСТИТУТ ПРОБЛЕМ ИНФОРМАТИКИ И АВТОМАТИЗАЦИИ НАН РА

Овсепян Мигран Меружанович

РАЗРАБОТКА БЕЗОПАСНЫХ АЛГОРИТМОВ ПОИСКА НА ЗАШИФРОВАННЫХ ДАННЫХ

АВТОРЕФЕРАТ

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

Ереван – 2016

Ատենախոսության թեման հաստատվել է Հայ-Ռուսական (Սլավոնական) համալսարանում

Գիտական ղեկավար՝	տեխ. գիտ. դոկտոր	Գ. Հ. Խաչատրյան
Պաշտոնական ընդդիմախոսներ՝	ֆիզ. մաթ. գիտ. դոկտոր	Լ. Հ. Ասլանյան
	ֆիզ. մաթ. գիտ. թեկնածու	Հ. Է. Դանոյան
Առաջատար կազմակերպություն՝	Հայաստանի ազգային պոլիտեխնիկական համալսարան	

Պաշտպանությունը կայանալու է 2016թ. հունիսի 6-ին, ժամը 16:00-ին ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում գործող 037 «Ինֆորմատիկա և հաշվողական համակարգեր» մասնագիտական խորհրդի նիստում հետևյալ հասցեով՝ Երևան, 0014, Պ. Սևակի 1:

Ատենախոսությանը կարելի է ծանոթանալ ՀՀ ԳԱԱ ԻԱՊԻ գրադարանում:

Սեղմագիրն առաքված է 2016թ. մայիսի 6-ին:

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

Тема диссертации утверждена в Российско-Армянском (славянском) университете

Научный руководитель:	доктор тех. наук	Г. Г. Хачатрян
Официальные оппоненты:	доктор физ.-мат. наук	Л. А. Асланян
	кандидат физ.-мат. наук	А. Э. Даноян
Ведущая организация:	Национальный Политехнический Университет Армении	

Защита состоится 6-го июня 2016г. в 16.00 на заседании специализированного совета 037 «Информатика и вычислительные системы» 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 6-го мая 2016 г.

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

ԱՇԽԱՏԱՆՔԻ ԸՆԴՀԱՆՈՒՐ ԲՆՈՒԹԱԳԻՐԸ

Թեմայի արդիականությունը

Մեր օրերում *որոնումը* ինֆորմացիան հասանելի դարձնելու հիմնական միջոցն է, և համացանցի զարգացման ու «տվյալները որպես ծառայություն» (data as a service) տեղակայման մոդելի (deployment model) տարածման շնորհիվ այժմ որոնումը հիմնականում կատարվում է առցանց: Oracle DaaS - ը, Amazon S3 պահոցը, Microsoft Azure պահոցը, տարատեսակ ամպային հանրային պահոցները (օրինակ՝ Dropbox, Google Drive, Box, Microsoft OneDrive և այլն) «տվյալները որպես ծառայություն» մոդելով աշխատող համակարգերից ընդամենը մի քանիսն են: «Տվյալները որպես ծառայություն» մոդելի հիմնական բաղադրիչները հետևյալն են՝

- *Տվյալների օգտատեր*՝ հանդիսանում է տվյալների սեփականատերը և կարող է դրանք կարդալ և փոփոխել: Օգտատերը ունի որոշակի հաշվողական ռեսուրս և հիշողություն, բայց դրանք սերվերի համապատասխան ռեսուրսներից անհամեմատ ավելի քիչ են:
- *Սերվեր*՝ հեռակա ծառայությունների մատակարար, ով պահում է օգտատերերի տվյալները և հատուկ ծրագրային համակարգի՝ ինտերֆեյսի միջոցով ապահովում տվյալների հասանելիությունը օգտվողներին՝ կլիենտներին:
- *Տվյալներից օգտվողներ կամ կլիենտներ*՝ օգտատերը, ինչպես նաև կողմնակի անձինք, ովքեր՝ օգտատիրոջ թույլտվությամբ, իրավունք ունեն տվյալների վերաբերյալ հարցումներ իրականացնել սերվերին: Օրինակ՝ եթե *օգտատերը* կազմակերպություն է, այդ կազմակերպության աշխատակիցները կարող են հանդիսանալ *օգտվողներ*:

Նշված մոդելը ունի մի շարք առավելություններ ծրագրային համակարգերի ավանդական տեղակայման մոդելի նկատմամբ: Դրանցից հիմնականներն են ցածր գինը, պահոցի ապարատային և ծրագրային ենթակառուցվածքների կառավարումը սերվերի կողմից, պահոցի չափը հեշտությամբ մեծացնելու հնարավորությունը (scalability), տվյալների հասանելիությունը տարբեր սարքերից, տրամադրվող ծառայությունների բարձր որակը և այլն: Սակայն այս մոդելն ունի նաև մեծ խնդիր՝ կապված սերվերի կողմից կառավարվող տվյալների անվտանգության հետ: Անկախ տվյալների բնույթից՝ դրանց սեփականատերերը հիմնականում ցանկանում են ունենալ երաշխիքներ, որ դրանք չեն գողացվի և առանց թույլտվության որևէ մեկը դրանք չի կարդա և չի փոփոխի: Այդպիսի երաշխիքներ տալու համար ծառայությունների մատակարարները իրականացնում են օգտվողների *իսկության ստուգման* (authentication), *լիազորությունների ստուգման* (authorization, permission check), *տվյալների պահուստավորման* (data backup) և այլ ընդունված գործողություններ: Այս բոլոր գործողությունները, ճիշտ է, բարձրացնում են սերվերի կողմից պահվող տվյալների անվտանգության մակարդակը, սակայն չեն երաշխավորում տվյալների անվտանգությունը հենց սերվերից՝ սերվերի ծրագրային և ապարատային համակարգերի ադմինիստրատորներից, այդ համակարգերում առկա հնարավոր հետախույզ ծրագրերից ու վիրուսներից: Ակնհայտ է, որ նշված խնդիրը լուծելու համար օգտատերը կարող է պահոց ուղարկելուց առաջ տվյալները գաղտնագրել: Դա թերևս կլուծի խնդիրը, բայց փոխարենը կբարդացնի այդ տվյալներում որոնում

իրականացնելու հնարավորությունը: Խնդրի լուծման համար առաջ է քաշվել որոնելի գաղտնագրման՝ ՈԳ (searchable encryption) սխեմաներ ստեղծելու անհրաժեշտության հարցը[1]: Այս սխեմաները թույլ են տալիս պահել ֆայլերը արտաքին պահոցում գաղտնագրված տեսքով՝ միաժամանակ թույլատրելով իրականացնել դրանցում որոնում ըստ այդ ֆայլերի հիմնաբառերի (keyword): ՈԳ սխեմաները հանդիսանում են կիրառական և տեսական հետազոտությունների արդիական թեմա: **Ատենախոսության առաջին մասը նվիրված է ՈԳ սխեմաներին:**

Այժմ դիտարկենք մեկ այլ խնդիր: Նախորդի նման՝ այստեղ ևս կա երկու կողմ՝ կլիենտ ու սերվեր, և կլիենտը որոնման հարցում է ուղարկում սերվերին: Բայց, ի տարբերություն նախորդի՝ այստեղ սերվերն է հանդիսանում տվյալների սեփականատերը, հետևաբար դրանք սերվերից անվտանգ պահելու խնդիր չկա: Փոխարենը կա որոնման հարցումը՝ նմուշը (pattern), սերվերից գաղտնի պահելու խնդիր: Այսինքն՝ կլիենտը, ինչ-ինչ պատճառներից ելնելով, չի ցանկանում սերվերին փոխանցել հարցումը, սակայն ցանկանում է, որ սերվերը այդ հարցումը կատարի և իրեն ուղարկի կատարման արդյունքը: Բացի այդ, սերվերն էլ իր հերթին ցանկանում է գաղտնի պահել իր տվյալները կլիենտից, այսինքն՝ չի ցանկանում կլիենտի հարցման պատասխանից ավել ինֆորմացիա փոխանցել վերջինիս: Նմանատիպ խնդիրների լուծման համար են նախատեսված նմուշների անվտանգ որոնման՝ ՆԱՈ հաղորդակարգեր (secure pattern search/matching protocols): Այսպիսի խնդիր կարող է ի հայտ գալ հետևյալ իրավիճակներում՝

1. Մի կողմից ոստիկանությունը՝ սերվերն, ունի անցյալում դատապարտված կամ կալանավորված մարդկանց ԴՆԹ-ների ցուցակը, սակայն չի ցանկանում դրանք հանրայնացնել: Մյուս կողմից գենետիկայով զբաղվող գիտնականը՝ կլիենտը, ցանկանում է հասկանալ, թե ինչ ընդհանրություններ կան կրիմինալ հակումներ ունեցող մարդկանց ԴՆԹ-ներում: Իր հետազոտության ընթացքում գիտնականը կառուցում է որոշակի նմուշ, օրինակ՝ կանոնավոր արտահայտություն (regular expression), և ցանկանում է ստուգել, թե ոստիկանության ցուցակի ԴՆԹ-ների քանի՞ տոկոսն է համապատասխանում այդ նմուշին կամ միջինում քանի՞ անգամ է նմուշը հանդիպում յուրաքանչյուր ԴՆԹ-ում և այլ նմանատիպ խնդիրներ: Սակայն գիտնականը ցանկանում է գաղտնի պահել այդ նմուշը սերվերից, քանի որ վերջինս հանդիսանում է իր հետազոտության արդյունքը: Այսպիսով առաջանում է վերոհիշյալ խնդիրը՝ գիտնականը ցանկանում է իր նմուշի վերաբերյալ ինֆորմացիա ստանալ սերվերից առանց այդ նմուշը ուղարկելու և առանց ԴՆԹ-ների ցուցակը ստանալու:
2. Անհատը կամ ընկերությունը ցանկանում է վարկ վերցնել բանկից: Բանկը ունի վարկ տրամադրելու իր պահանջները դիմումատուից, որոնց մի մասը վերաբերում են դիմումատուի վարկային պատմությունը: Այդ պահանջները ստուգելու համար

[1] Որոնելի գաղտնագրման առաջին սխեման համարվում է Սոնգի սխեման, որը հրապարակվել է 2000 թ.-ին "IEEE Symposium on Security and Privacy" սիմպոզիումի շրջանակներում: Դրանից հետո ստեղծվել են և շարունակվում են ստեղծվել բազմաթիվ այլ ՈԳ սխեմաներ:

բանկը՝ կլիենտը, պետք է հարցում ուղարկի վարկային ռեգիստրին՝ սերվերին: Սակայն բանկի՝ վարկեր տրամադրելու քաղաքականությունը, մասնավորապես՝ այդ քաղաքականության մաս հանդիսացող, դիմումատուի վարկային պատմության հանդեպ եղած պահանջները, հանդիսանում են առևտրային գաղտնիք, և բանկը չի ցանկանում դրանք տրամադրել վարկային ռեգիստրին: Մյուս կողմից վարկային ռեգիստրը ինքը չի ցանկանում բանկին տրամադրել դիմումատուի ամբողջ վարկային պատմությունը: Այսպիսով կրկին առաջանում է վերոհիշյալ խնդիրը՝ բանկը ցանկանում է ստուգել իր պահանջները առանց դրանք ուղարկելու վարկային ռեգիստր և առանց վերջինից ստանալու դիմումատուի վարկային պատմությունը:

Այսպիսի իրավիճակ կարող է ի հայտ գալ նաև այլ հեռակա ստուգման/որոնման խնդիրներում: **Ատենախոսության երկրորդ մասը նվիրված է ՆԱՌ հաղորդակարգերին:**

Աշխատանքի նպատակը

- Սիմետրիկ գաղտնագրման, այսպես կոչված, «սպիտակ արկղ» կիրառելով՝ լուծել ՈԳ սիմետրիկ սխեմաների բանալիները օգտագործողներին բաշխելու խնդիրը: Խնդրի լուծումը ցուցադրելու համար նախագծել և իրականացնել կիրառական հատկանիշներ ունեցող ՈԳ սիմետրիկ սխեմա:
- Ուսումնասիրել տվյալների ամպային պահոցներում անվտանգության մակարդակը գաղտնագրության միջոցով բարձրացնող ծրագրային համակարգերը և դրանցում ՈԳ սխեմայի ներդրման հնարավորությունը:
- Նախագծել նոր ՆԱՌ հաղորդակարգ, որը թույլ կտա կլիենտին ստուգել՝ արդյոք իր մոտ եղած կանոնավոր արտահայտությունը համապատասխանում է (match) սերվերի մոտ եղած տողին, առանց սերվերին փոխանցելու այդ կանոնավոր արտահայտությունը և առանց սերվերից վերջինիս տողը ստանալու: Հաղորդակարգում չպետք է օգտագործվի ասիմետրիկ գաղտնագրում. փոխարենը պետք է օգտագործվի սիմետրիկ գաղտնագրում սպիտակ արկղում (white-box cryptography), որի հաշվին էլ այն պետք է էապես գերազանցի այլ նմանատիպ հաղորդակարգերին իր արագագործությամբ: Մշակել նախագծված հաղորդակարգը իրականացնող գրադարան և ծրագիր հաղորդակարգի փաստացի արագագործությունը ստուգելու համար:

Հետազոտման մեթոդները հիմնված են գաղտնագրման դասական հասկացությունների, ինչպես նաև սպիտակ արկղում սիմետրիկ գաղտնագրման հնարավորությունը որպես հանրային բանալուն այլընտրանք օգտագործելու մոտեցման վրա: Բացի այդ մշակված ծրագրային համակարգերի և դրանց արագագործության ստուգման համար օգտագործվել են օբյեկտակողմնորոշված ծրագրավորման մոտեցումները և C++ ծրագրավորման լեզուն:

Գիտական նորույթը

- Նոր մեթոդ, որը թույլ է տալիս առանց ՈԳ սիմետրիկ սխեմաների բանալիները օգտագործողներին բաշխելու վերջիններիս տալ որոնման և/կամ փոփոխման հարցումներ իրականացնելու հնարավորություն:

- Ճարտարապետություն, որը թույլ է տալիս հանրային ամպային պահոցներում (Dropbox, Google Drive և այլն) ֆայլերը պահել գաղտնագրված տեսքով՝ միաժամանակ տրամադրելով դրանցում անվտանգ որոնում իրականացնելու հնարավորություն:
- Նոր ՆԱՈ հաղորդակարգ, որը թույլ է տալիս կլիենտին ստուգել՝ արդյոք իր մոտ եղած կամայական Σ մուտքի այբուբեն ունեցող կանոնավոր արտահայտությունը համապատասխանում է սերվերի մոտ եղած տողին:

Ստացված արդյունքների կիրառական նշանակությունը

ՈԳ սխեմաների թեմայով ստացված արդյունքների հիման վրա մշակվել և իրականացվել է ՈԳ սիմետրիկ սխեմա, որը թույլ է տալիս տվյալների տիրոջը իր կողմից գաղտնագրված և սերվերի վրա պահվող ֆայլերում որոնման և/կամ փոփոխման հնարավորություն տրամադրել երրորդ կողմին, առանց վերջինիս տրամադրելու սխեմայի բանալիները: Սխեման կարող է կիրառվել, հատկապես իրավաբանական և կոմերցիոն արժեք ներկայացնող ֆայլերի հետ աշխատելիս: Օրինակ՝ իրավական խնդիրներով զբաղվող ընկերության աշխատակիցներին ընկերության փասթաթղթերի արխիվում որոնում կատարելու հնարավորություն տրամադրել, առանց վերջիններիս գաղտնագրման բանալիները տրամադրելու:

Մշակված ՆԱՈ հաղորդակարգը, իր արագագործության և բարձր անվտանգության մակարդակի հաշվին կարող է կիրառվել ԴՆԹ-ի ուսումնասիրման, բանկային, հեռակա ախտորոշման և այլ ոլորտներում:

Պաշտպանությանը ներկայացվող հիմնական դրույթները

1. Նոր Մեթոդ, որը թույլ է տալիս առանց ՈԳ սիմետրիկ սխեմաների բանալիները օգտագործողներին բաշխելու վերջիններիս որոնման և/կամ փոփոխման հարցումներ իրականացնելու հնարավորություն տալ:
2. Նոր ՈԳ սխեմա, հիմնված DSSE[2] ՈԳ սիմետրիկ սխեմայի վրա, որը, առաջին կետում նշված մեթոդով, թույլ է տալիս օգտագործողներին որոնման և/կամ փոփոխման հարցումներ իրականացնելու հնարավորություն տրամադրել:
3. C++ գրադարան, որը իրականացնում է մշակված նոր ՈԳ սխեման և այդ գրադարանից օգտվող գրաֆիկական ինտերֆեյսով ծրագիր:
4. Ճարտարապետություն, որը թույլ է տալիս հանրային ամպային պահոցներում (Dropbox, Google Drive և այլն) ֆայլերը պահել գաղտնագրված տեսքով՝ միաժամանակ տրամադրելով դրանցում անվտանգ որոնում իրականացնելու հնարավորություն, որը ներդրվել է SkyCryptor[3] համակարգում որոնման ֆունկցիոնալություն մշակելու համար:
5. Ասիմետրիկ գաղտնագրում չօգտագործող նոր ՆԱՈ հաղորդակարգ, որը թույլ է տալիս կողմերից մեկին՝ կլիենտին, ստուգել՝ արդյոք իր մոտ եղած կամայական Σ

[2] DSSE (dynamic searchable symmetric encryption) սխեման նկարագրված է այստեղ՝ https://eprint.iacr.org/2012/530.pdf:

[3] https://www.skycryptor.com/

մուտքի այբուբենն ունեցող Γ կանոնավոր արտահայտությունը համապատասխանում է մյուս կողմի՝ սերվերի մոտ եղած n երկարության $x \in \Sigma^n$ տողին:

6. C++ գրադարան, որը իրականացնում է նախագծված ՆԱՌ հաղորդակարգը և այդ գրադարանի աշխատանքը ցուցադրող ծրագիր:

Աշխատանքի արդյունքների ներդրումը

Աշխատանքում նկարագրված ուսումնասիրությունների և դրանց արդյունքների հիմման վրա մշակվել և «Ուանքրփթոր» ընկերության SkyCryptor ծրագրային համակարգում ներդրվել է ֆունկցիոնալություն, որը թույլ է տալիս Dropbox կամ Google Drive ամպային պահոցից ընտրել որևէ պանակ (folder) և այն իր մեջ եղած ֆայլերի՝ ներառյալ ենթապանակները (subfolders), հետ միասին գաղտնագրել՝ թուլատրելով պանակում ֆայլերի հիմնաբառերով որոնում իրականացնելու հնարավորություն:

Աշխատանքի արդյունքների զեկուցումները

- Միջազգային գիտաժողով՝ «The NATO Science for Peace and Security Programme: Advanced Research Workshop» (AWR-2015) ARW, Աղվերան, Հայաստան, 2015 թ.:
- Միջազգային գիտաժողով՝ «Կոմպյուտերային գիտություն և տեղեկատվական տեխնոլոգիաներ» (CSIT-2015), Երևան, Հայաստան, 2015 թ.:
- Գիտաժողով՝ «Science And Technology Convergence Forum», Երևան, Հայաստան, 2016 թ.:
- Սեմինարներ Հայ-Ռուսական (Սլավոնական) համալսարանում, Հայաստանի Ամերիկյան համալսարանում և ՀՀ ԳԱԱ Ինֆորմատիկայի և ավտոմատացման պրոբլեմների ինստիտուտում:

Հրատարակումները

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

Աշխատանքի կառուցվածքը և ծավալը

Ատենախոսությունը բաղկացած է ներածությունից, չորս գլուխներից, եզրակացությունից, օգտագործված գրականության ցանկից և A հավելվածից՝ հապավումների բառարանից: Աշխատանքի ընդհանուր ծավալը 105 էջ է՝ ներառյալ 18 պատկեր, 3 աղյուսակ և 110 հղում օգտված գրականության ցանկում:

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 ՈԳ սխեմաների և ՆԱՌ հաղորդակարգերի ընդհանուր ուսումնասիրություն: Այս գլխում նաև ներկայացված է նոր ՆԱՌ հաղորդակարգի և նոր՝ առանց ՈԳ սիմետրիկ սխեմաների բանալիները օգտագործողներին բաշխելու, վերջիններիս որոնման և/կամ փոփոխման հարցումներ իրականացնելու հնարավորություն տվող մեթոդի ստեղծման

համար օգտագործված հիմնական գաղափարը՝ սպիտակ արկղում սիմետրիկ գաղտնագրման հնարավորությունը որպես հանրային բանալուն այլընտրանք օգտագործելը: **1.1 բաժնում** բացատրված է ՈԳ սխեմաների ընդհանուր կառուցվածքը. ցանկացած ՈԳ սխեմա իրենից ներկայացնում է մի քանի հաղորդակարգերի բազմություն, որոնք կարելի է բաժանել երեք խմբի՝

1. *Նախապատրաստման (initialization) հաղորդակարգ:* Այս փուլում ֆայլերի նախնական $\boldsymbol{f} = \{f_1, f_2, \dots, f_n\}$ բազմության բոլոր էլեմենտները $\boldsymbol{K}$ բանալիների[4] միջոցով օգտատիրոջ կողմից գաղտնագրվում են, որի արդյունքում ստացվում է ֆայլերի ծածկագրերի $\boldsymbol{c} = \{c_1, c_2, \dots, c_n\}$ բազմությունը: Բացի ֆայլերի գաղտնագրումից, օգտատիրոջ կողմից՝ կրկին $\boldsymbol{K}$ ի կիրառմամբ, կառուցվում է անվտանգ որոնման ինդեքս $\boldsymbol{I}$, որը պարունակում է ինֆորմացիա այն մասին, թե որ ֆայլը ինչ հիմնաբառեր (keywords) է պարունակում[5], սակայն այդ ինֆորմացիան հասանելի է միայն համապատասխան բանալիների առկայության դեպքում: Նախապատրաստման հաղորդակարգի վերջում օգտատերը սերվերին է ուղարկում $(\boldsymbol{c}, \boldsymbol{I})$ զույգը:
2. *Որոնման (search) հաղորդակարգեր:* Օգտատերը հատուկ ձևով գաղտնագրում է որոնման հարցումը[6]՝ ստանալով որոնման կտրոն (search token), և ուղարկում այդ կտրոնը տվյալների պահոցը սպասարկող սերվերին: Վերջինս, օգտագործելով կտրոնը և տվյալների $\boldsymbol{I}$ ինդեքսը, համապատասխան ալգորիթմի միջոցով գտնում ու օգտատիրոջն է ուղարկում հարցմանը համապատասխան ֆայլերի նույնացուցիչները (id) կամ ծածկագրերը: Որոնման ալգորիթմը և ինդեքսի կառուցվածքը ապահովում են որոնվող հիմնաբառերի գաղտնիությունը սերվերից:
3. *Փոփոխման (modification) հաղորդակարգեր:* Իրականացնում են ֆայլի ավելացման, հեռացման կամ խմբագրման գործողություն: Գլխենտը, օգտագործելով իր $\boldsymbol{K}$ բանալիները, հատուկ ձևով ստեղծում է փոփոխման կտրոն (modification token), որը պարունակում է գաղտնագրված ինֆորմացիա պահանջվող փոփոխության մասին և ուղարկում է այդ կտրոնը տվյալների պահոցը սպասարկող սերվերին: Վերջինս, օգտագործելով կտրոնը, համապատասխան

---

[4] $\boldsymbol{K}$-ն իրենից ներկայացնում է ՈԳ սխեմայի բանալիների բազմությունը: Տարբեր հաղորդակարգերում գաղտնագրման/գաղտնազերծման գործողությունների ժամանակ կարող են օգտագործվել տարբեր բանալիներ:

[5] Ամեն ֆայլի հիմնաբառերի բազմությունը կարելի է ստանալ ծրագրային միջոցով: Օրինակ՝ մարդու համար ընթեռնելի տեքստ պարունակող ֆայլերի դեպքում (.txt, .pdf, MS Office, և այլն) որպես հիմնաբառեր կարելի է ընտրել տեքստում հանդիպող բոլոր տարբեր բառերի ցողունները (stem), այլ ֆայլերի դեպքում որպես հիմնաբառ կարելի է ընտրել ֆայլը բնութագրող բառերի ցողունները:

[6] Հիմնականում հարցումը հետևյալն է՝ «գտնել այն ֆայլերի նույնացուցիչները (id), որոնք պարունակում են w հիմնաբառը», սակայն որոշ ՈԳ սխեմաներ թույլ են տալիս որոնել մի քանի հիմնաբառերի Բուլյան արտահայտություններով: Օրինակ՝ «գտնել այն ֆայլերի նույնացուցիչները, որոնք պարունակում են w_1 և w_2 հիմնաբառերը կամ չեն պարունակում w_3-ը»:

ալգորիթմի օգնությամբ կատարում է պահանջվելիք փոփոխությունները տվյալների ***I*** ինդեքսում և ֆայլերի ծածկագրերի ***c*** բազմությունում:

Այնուհետև կրկին 1.1 բաժնում դասակարգված են ՈԳ սխեմաների հիմնական հատկանիշները՝

1. *Անվտանգության հատկանիշներ:* Գոյություն ունեցող ՈԳ սխեմաների անվտանգության անալիզը մեծամասամբ կատարված է պասիվ կամ հետաքրքրասեր գրոհողի մոդելում (passive adversary or curious intruder model): Այսինքն՝ ենթադրվում է, որ սերվերը ճշգրիտ կերպով է իրականացնում ՈԳ սխեմայի հաղորդակարգերը և չի փորձում որևէ բան փոփոխել, սակայն վերջինս փորձում է պահվող տվյալների և կատարվող հարցումների մասին հնարավորինս շատ իմաստալից ինֆորմացիա քաղել[7]:

   ՈԳ սխեմայի անվտանգության կարևոր հատկանիշ է ապառնի (forward) գաղտնիությունը՝ երաշխիքը, որ նոր ֆայլ ավելացնելուց սերվերը չի կարող հասկանալ՝ արդյոք այդ ֆայլը հանդիսանում է նախկինում կատարված որոշակի հարցման արդյունքի մաս: Մեկ այլ կարևոր անվտանգության հատկանիշ է հետադարձ (backward) գաղտնիությունը՝ երաշխիքը, որ ո՛չ հարցման արդյունքը, ո՛չ էլ հարցման ընթացքում ստացված միջանկյալ արդյունքները չեն պարունակում ջնջված կամ խմբագրված[8] ֆայլերի նույնացուցիչներ: Կան ՈԳ սխեմայի անվտանգությունը բնութագրող բազմաթիվ այլ հատկանիշներ՝ կապված սխեմայի հաղորդակարգերի աշխատանքի ընթացքում ինֆորմացիոն արտահոսքի հետ:
2. *Դինամիկության հատկանիշ*: ՈԳ սխեման կարող է թույլատրել կամ չթույլատրել պահվող ֆայլերի փոփոխման՝ ավելացում, հեռացում, խմբագրում, գործողություններ: Դրանք թույլատրող ՈԳ սխեմաները կոչվում են *դինամիկ*, իսկ մյուսները՝ *ստատիկ*: ՈԳ սխեմաների մեծ մասը դինամիկ են:
3. *Օգտվողների քանակի հատկանիշ:* ՈԳ սիմետրիկ սխեմաներում (symmetric searchable encryption SSE schemes) ***I*** ինդեքսի, որոնման ու փոփոխման կտրոնների կառուցման, ինչպես նաև հետագայում որոնման ու փոփոխման ալգորիթմների կատարման ընթացքում օգտագործվում են սիմետրիկ գաղտնագրման գործողություններ: Այդ պատճառով, եթե օգտատերը ցանկանում է երրորդ կողմին թույլատրել իր ֆայլերում որոնում և/կամ փոփոխում կատարել, նա պետք է վերջինիս փոխանցի իր սեփականությունը հանդիսացող սիմետրիկ գաղտնագրման համապատասխան բանալիները: Այս տեսակետից ՈԳ սիմետրիկ սխեմաները համարվում են միայն մի օգտվողի՝ օգտատիրոջ համար նախատեսված սխեմաներ (single-user schemes), քանի որ բանալիների փոխանցումից հետո սխեմայի տեսանկյունից օգտվողի և օգտատիրոջ միջև որևէ տարբերություն չկա: Փոխարենը հանրային բանալով՝ ասիմետրիկ, ՈԳ

[7] Որպես պասիվ գրոհող կարող է հանդես գալ սերվերի կողմում գտնվող որևէ ծրագիր կամ անհատ, ում հասանելի է ՈԳ սխեմայի հաղորդակարգերի կատարման գործընթացը:

[8] Երբ խմբագրվելուց առաջ ֆայլը բավարարել է տվյալ որոնմանը, բայց այժմ չի բավարարում:

սխեմաները[9] թույլ են տալիս օգտատիրոջը երրորդ կողմերին որոնման գործողություն իրականացնելու լիազորություն տրամադրել՝ առանց վերջիններիս տրամադրելու գաղտնագրման բանալիները, հետևաբար՝ չտալով ֆայլերը կարդալու, ջնջելու և խմբագրելու հնարավորություն:

4. *Արագագործության, **I** ինդեքսի օգտագործած հիշողության, հաղորդակարգերի կատարման ընթացքում օգտագործվող հիշողության, կլիենտից սերվերին և հակառակ փոխանցվող տվյալների չափի և այդ փոխանցումների քանակի՝ փուլերի հատկանիշներ:* Սրանք սխեմայի պրակտիկ կիրառելիությունը որոշող հիմնական հատկանիշներն են, սակայն սովորաբար ինչքան ավելի լավն են այս հատկանիշները, այնքան ավելի վատն են անվտանգության հատկանիշները: Գոյություն ունեցող առավել կիրառելի ՈԳ սխեմաների որոնման գործողությունները ունեն որոնման արդյունքների քանակից գծային կախվածություն, իսկ փոփոխման գործողությունները զգալիորեն ավելի ծանր են: ***I*** ինդեքսի օգտագործած հիշողությունը համեմատական է առանձին ֆայլերի հիմնաբառերի քանակների գումարին: Կլիենտ-սերվեր հաղորդակցման թրաֆիկը ուղիղ համեմատական է հարցման չափին որոնման հաղորդակարգերում, փոփոխման հաղորդակարգերում՝ ավելացող կամ փոփոխվող ֆայլի չափին, իսկ ֆայլը հեռացման դեպքում հաստատուն է: Կլիենտ-սերվեր հաղորդակցությունների քանակը նույնպես հաստատուն է:

Այս ամենից հետո 1.1 բաժնում կատարված է գոյություն ունեցող առավել կիրառական հատկանիշներ ունեցող վեց ՈԳ սխեմաների հակիրճ նկարագրություն:

1.2 բաժնում ձևակերպված է նմուշների անվտանգ որոնման՝ ՆԱՈ խնդիրը: Խնդրի և դրա վարիացիաների հակիրճ ձևակերպումը հետևյալն է. կա երկու կողմ՝ կլիենտ և սերվեր, կլիենտը ունի որոշակի p նմուշ (որպես նմուշ կարող է հանդես գալ տեքստի հատվածը կամ որևէ արտահայտություն, որը բնութագրում է բազմաթիվ այդպիսի հատվածներ. օրինակ՝ կանոնավոր արտահայտություն և այլն), իսկ սերվերը ունի n երկարության S տողը: Կողմերի նպատակն է՝

1. Ստուգել արդյոք S-ը համապատասխանում է p նմուշին (ամբողջական համապատասխանություն (full matching)),
2. Ստուգել արդյոք S-ի որևէ հատվածը համապատասխանում է p նմուշին (մասնակի համապատասխանություն (partial matching)),
3. Հաշվել S-ի այն հատվածների քանակը, որոնք համապատասխանում են p նմուշին (համապատասխանությունների քանակը (count of all matches)),
4. Գտնել S-ի այն դիրքերի ցուցակը որոնք հանդիսանում են p-ին համապատասխանող հատվածի սկիզբ (համապատասխանությունների դիրքեր (positions of all matches)):

[9] Գրականության մեջ *ասիմետրիկ ՈԳ սխեմաները* կոչվում են *հիմնաբառերի որոնում թույլատրող հանրային բանալով գաղտնագրման սխեմաներ* (public key encryption with keyword search):

Այլ կերպ ասած՝ կողմերը պետք է հաշվեն $F(p,S)$ ֆունկցիայի արժեքը, որը առաջին երկու դեպքերում Բուլյան է (1՝ համապատասխանում է և 0՝ չի համապատասխանում), երրորդ դեպքում այն ոչ բացասական ամբողջ թիվ է, իսկ չորրորդ դեպքում դիրքերի ցուցակ: Կա ևս մեկ պահանջ՝ $F(p,S)$ ֆունկցիայի արժեքը պետք է հաշվարկվի այնպես, որ ո՛չ կլիենտը սովորի S-ը, ո՛չ էլ սերվերը սովորի p-ն:

Այնուհետև 1.2 բաժնում քննարկվում է, թե գոյություն ունեցող ՆԱՈ հաղորդակարգերը ինչ նմուշների համար են նախատեսված, և վերջում տրվում է կանոնավոր արտահայտության սահմանումը, որը չորրորդ գլխում նկարագրված նոր ՆԱՈ հաղորդակարգում կատարում է նմուշի դերը:

1.3 բաժնում բացատրվում է, թե ինչպես կարելի է սպիտակ արկղում սիմետրիկ գաղտնագրման հնարավորությունը օգտագործել որպես հանրային բանալուն այլընտրանք: Նաև ներկայացվում է նախագծված նոր ՆԱՈ հաղորդակարգի բաղկացուցիչ մաս հանդիսացող 1-ը-N-ից գաղտնի ընտրման հաղորդակարգերի աշխատանքի սկզբունքը:

Երկրորդ գլխում նկարագրված է առանց ՈԳ սիմետրիկ սխեմաների բանալիները օգտագործողներին բաշխելու վերջիններիս որոնման և/կամ փոփոխման հարցումներ իրականացնելու հնարավորություն տվող նոր մեթոդ: Մեթոդի հիմնական միտքը բանալիների փոխարեն սպիտակ արկղում սիմետրիկ գաղտնագրում իրականացնելու հնարավորության բաշխումն է: Մեթոդը հետևյալն է. ՈԳ սիմետրիկ սխեմայում ինդեքսի կառուցման ընթացքում կատարվող գաղտնագրման գործողությունները կատարվում են սպիտակ արկղի միջոցով, այնուհետև այդ սպիտակ արկղերը տրամադրվում են օգտվողին: Կախված նրանից, թե ինչ լիազորություններ են տրամադրվում օգտվողին՝ նրան կարող են տրամադրվել բոլոր սպիտակ արկղերը կամ դրանց մի մասը: Մեթոդի ցուցադրման համար DSSE ՈԳ սխեմայի հիմման վրա կառուցվել է նոր ՈԳ սխեմա, այդ սխեման իրականացնող գրադարան և այդ գրադարանի աշխատանքը ցուցադրող գրաֆիկական ինտերֆեյսով ծրագիր: Գլխում նկարագրված են առաջարկված ՈԳ սխեման և դրա անվտանգության հիմնավորումները: **2.1 բաժնում** տրված են օգտագործվող սահմանումներն ու նշանակումները, **2.2 բաժնում** նկարագրված են են սխեմայի բաղկացուցիչ 14 ալգորիթմները՝

- Gen (1^k),
- Enc$(\boldsymbol{K}, \boldsymbol{f})$,
- GenSearchGateway(K_1, K_2, K_3),
- GenUpdateGateway$(K_1, K_2, K_3, K_4, K_5, K_6, K_7)$,
- SearchToken(K_1, K_2, K_3, w),
- GatewaySearchToken(T_1, T_2, T_3, w),
- AddToken$(\boldsymbol{K}, f)$,
- GatewayAddToken$(\boldsymbol{T}, f)$,
- DelToken$(\boldsymbol{K}, f)$,
- GatewayDelToken$(\boldsymbol{T}, f)$,
- Search$(\gamma, \boldsymbol{c}, \tau_s)$,
- Add$(\gamma, \boldsymbol{c}, \tau_a)$,
- Del$(\gamma, \boldsymbol{c}, \tau_d)$,
- Dec(f)

Բաժին 2.3-ում քննարկված են սխեմայի անվտանգության հետևյալ հարցերը՝

1. Բանալու անվտանգությունը օգտվողներից:
2. Տվյալների անվտանգությունը օգտվողներից:
3. Բանալու անվտանգությունը արդար, բայց հետաքրքրասեր (honest but curious) սերվերից:
4. Տվյալների անվտանգությունը արդար, բայց հետաքրքրասեր սերվերից:

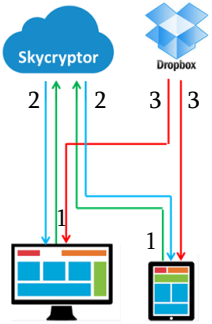
Վերջապես **2.4 բաժնում** նկարագրված է մշակված գրադարանը և ծրագիրը, որը իրականացնում է սխեման՝ օգտագործելով սիմետրիկ գաղտնագրման սպիտակ արկղեր: DSSE անունը կրող գրադարանը տրամադրում է API ՈԳ սխեմայի կլիենտ և սերվեր հավելվածների (client and server application) ծրագրավորման համար: Կլիենտի կողմի API-ի հիմնական դասը (class) QueryTokenGenerator-ն է: Այս դասի օբյեկտները հնարավորություն են տալիս ստեղծել SearchToken, AddToken և DeleteToken դասերի օբյեկտներ: Բոլոր *Token դասերը ունեն serialize() և deserialize() ֆունկցիաները, որպեսզի հնարավոր լինի դրանց օբյեկտները ուղարկել կլիենտից սերվերին: Սերվերի կողմի API-ի հիմնական դասը FileDbManager-ն է: Այս դասը կառավարում է անվտանգ ինդեքսը և ֆայլերի ծածկագրերը: Այն տալիս է որոնման, ֆայլի հեռացման, ավելացման հնարավորություն համապատասխան կտրոնի կիրառմամբ: Ստեղծված ցուցադրական ծրագիրը իրականացնում է և՛ կլիենտի, և՛ սերվերի գործառույթները: Գրաֆիկական ինտերֆեյսի միջոցով այն հնարավորություն է տալիս ստեղծել դատարկ պահոց և հետագայում հարցումների միջոցով ֆայլեր ավելացնել/հեռացնել դրանից, ինչպես նաև որոնել նշված հիմնաբառերը գաղտնագրված ֆայլերում:

Երրորդ գլխում քննարկված են ամպային պահոցներում տվյալների անվտանգությունը գաղտնագրման միջոցով բարձրացնող մի շարք ծրագրային համակարգերի՝ Sookasa, nCryptedCloud, Boxcryptor, աշխատանքի սկզբունքները:

- Sookasa՝ նորահայտ ամպային պահոցների գաղտնագրման համակարգ է (gateway)՝ նախատեսված հիմնականում կարգավորվող ոլորտների՝ առողջապահություն, ֆինանսներ և այլն, ինչպես նաև պետական կազմակերպություններին օգնելու նպատակով, որպեսզի վերջիններս հնարավորություն ունենան պահել իրենց տվյալները ամպային պահոցներում՝ առանց խախտելու պետության պահանջները տվյալների անվտանգության վերաբերյալ՝ HIPAA, FERPA, PCI DSS, GLBA, FINRA:
- nCryptedCloud՝ աշխատում է ամպային պահոցների մեծ մասի հետ: Թույլ է տալիս գաղտնագրել պահոցի ցանկացած ֆայլ ցանկացած պանակում: Թերություններից հիմնականը այն է, որ գաղտնագրված ֆայլը երրորդ անձին հասանելի դարձնելու համար օգտատերը ստիպված է ֆայլի գաղտնագրման բանալին հասանելի դարձնել nCryptedCloud-ին. ինչպես նաև այս համակարգում չի թույլատրվում որոնում գաղտնագրված ֆայլերում:
- Boxcryptor՝ միակ, գաղտնագրված ֆայլերի և բանալու մասին զրո-գիտելիք (zero-knowledge) ստացող, նմանատիպ ծառայությունն է: Այստեղ բանալու անվտանգ կառավարումը հիմնված է RSA ասիմետրիկ գաղտնագրման համակարգի վրա. ամեն օգտատեր ունի հանրային և գաղտնի բանալիների զույգ: Սակայն այստեղ ևս բացակայում է որոնման հնարավորությունը:

Այնուհետև այս գլխում նկարագրված է SkyCryptor համակարգում ներդրված որոնման սխեմայի աշխատանքի սկզբունքը օգտագործողի տեսակետից, որը ցուցադրված է **Նկար 1**-ում:

1. Օգտագործողը (հավելված կամ web-կայք) ուղարկում է անվտանգ որոնման հարցում:
2. SkyCryptor ի սերվերը կատարում է որոնումը անվտանգ ինդեքսում և վերադարձնում է համապատասխան ֆայլերի նույնացուցիչների բազմությունը:
3. Օգտագործողը բեռնում է համապատասխան ֆայլերը ամպային պահոցից (օր. Dropbox):



Նկար 1

Ատենախոսության **չորրորդ գլխում** ներկայացված է նոր ՆԱՈ հաղորդակարգ: Հաղորդակարգը նախատեսված է հետևյալ խնդրի լուծման համար. կլիենտը՝ որպես հարցում, ունի Σ այբուբենով կանոնավոր արտահայտություն r, իսկ սերվերը ունի $X \in \Sigma^n$ n երկարությամբ տող: Կլիենտը ուզում է ստուգել՝ արդյոք X-ը ամբողջությամբ համապատասխանում է r-ին (արդյոք X ը պատկանում է r-ի առաջացրած լեզվին՝ $X \in L(r)$) այնպես, որ r-ը անհայտ մնա սերվերից, իսկ X-ը՝ կլիենտից: Ավելի ֆորմալ՝ հաղորդակարգը թույլ է տալիս կլիենտին և սերվերին, սեփական արգումենտը մյուս կողմից գաղտնի պահելով, համատեղ հաշվել հետևյալ ֆունկցիայի արժեքը՝

$$f(r, X) = \begin{cases} 1, \text{if } X \in L(r) \\ 0, \text{if } X \notin L(r)' \end{cases}$$

որը նույնպես՝ կլիենտի ցանկությամբ, կարող է գաղտնի պահվել սերվերից: Հաղորդակարգում չեն օգտագործվում ասիմետրիկ գաղտնագրման գործողություններ, ինչի հաշվին այն նմանատիպ այլ հաղորդակարգերից ավելի արագ է: Ինչպես նաև ցույց է տրված, որ որոշ փոփոխություններից հետո մշակված հաղորդակարգը հնարավոր կլինի կիրառել ՆԱՈ խնդրի մյուս երեք վարիացիաներից յուրաքանչյուրի համար (մասնակի համապատասխանություն, համապատասխանությունների քանակ և դիրքեր):

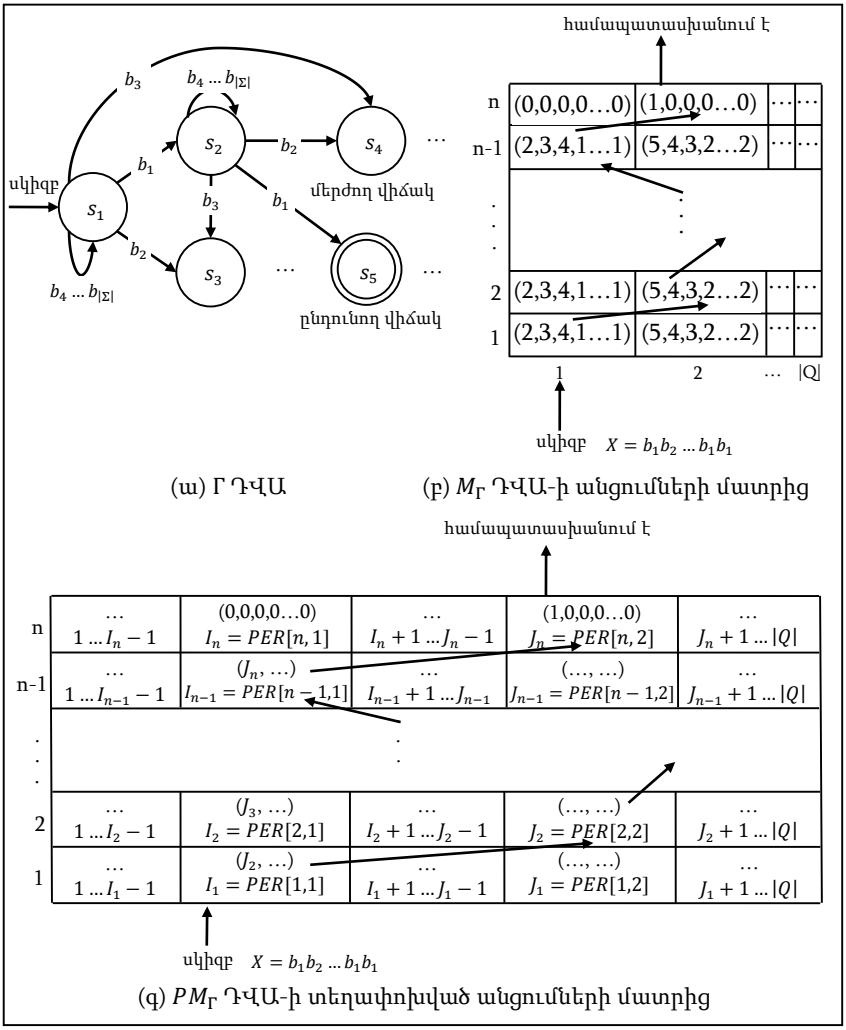
4.1 բաժնում տրված են սահմանումներ և բերված են նշանակումներ հաղորդակարգի աշխատանքն ու անվտանգությունը նկարագրելու համար: Մասնավորապես, քանի որ կանոնավոր արտահայտությունները և դետերմինիստիկ վերջավոր ավտոմատները հանդիսանում են համարժեք ֆորմալիզմներ, ենթադրվում է, որ կլիենտը r-ի փոխարեն ունի $\Gamma(Q, \Sigma, \Delta, s_1, F)$ դետերմինիստիկ վերջավոր ավտոմատ և հետագայում r-ի փոխարեն խոսվում է միայն Γ ի մասին: Այստեղ $Q = \{s_1, s_2, \dots, s_{|Q|}\}$ ավտոմատի վիճակների բազմությունն է, $\Sigma = \{b_1, b_2, \dots, b_{|\Sigma|}\}$՝ մուտքի այբուբենը, $\Delta_{Q \times \Sigma}$՝ անցումների աղյուսակը, որտեղ $\Delta[s, b] \in Q$, s_1-ը՝ սկզբնական վիճակը և F-ը՝ ընդունող կամ վերջնական վիճակների բազմությունը: Իսկ սերվերն ունի $X \in \Sigma^n$ n երկարության տող: Ասելով $\Gamma(X)$-ը 1 է նկատի ունենք, որ

$$\Delta\Big[\dots\Delta\Big[\Delta[s_1, X[1]], X[2]\Big]\dots, X[n]\Big] - \text{ը}$$

ընդունող վիճակ է, հակառակ դեպքում $\Gamma(X)$-ը 0 է: **4.2 բաժնում** նկարագրված է հաղորդակարգի աշխատանքը երկուական մուտքային այբուբենի դեպքում (երբ $\Sigma = \{0,1\}$), իսկ ընդհանուր դեպքի նկարագրությունը բերված է **4.3 բաժնում**: Հաղորդակարգի հիմնական քայլերը հետևյալն են՝

1. *Կլիենտ* – Γ ի հիման վրա կառուցել հատուկ M_Γ, $n \times |Q| \times |\Sigma|$ չափանի ԴՎԱ-ի անցումների մատրից, որը թույլ կտա ստուգել n-երկարության տողերի պատկանելիությունը $L(\Gamma)$ ին.
2. *Կլիենտ* – Կառուցել աղավաղված (garbled) ԴՎԱ անցումների GM_Γ մատրից՝ սկզբում M_Γ-ի ամեն տողի վրա կիրառելով պատահական տեղափոխություն և ստանալով ԴՎԱ-ի տեղափոխված անցումների մատրից PM_Γ, այնուհետև գաղտնագրելով ստացված մատրիցի ամեն մի վանդակը: Արդյունքում $\Gamma(X)$-ը հաշվելու համար բավարար կլինի սերվերին ունենալ GM_Γ մատրիցը և համապատասխան բանալիներ X ի ամեն հերթական տառի համար (տես **Նկար 2**):
3. *Սերվեր* – X-ի ամեն տառի համար ստեղծել 1-ը-$|\Sigma|$-ից գաղտնի ընտրման պրոտոկոլի համապատասխան հարցման կտրոն և ուղարկել այն կլիենտին:
4. *Կլիենտ* – Յուրաքանչյուր ստացված գաղտնի ընտրման կտրոնի համար կառուցել պատասխան, որը կպարունակի համապատասխան բանալին, և ուղարկել դրանք կլիենտին GM_Γ մատրիցի հետ միասին:
5. *Սերվեր* – X-ի ամեն տառի համար ստացված պատասխանից գաղտնազերծել համապատասխան բանալին և հաշվել $\Gamma(X)$-ը՝ օգտագործելով այդ բանալիները և GM_Γ աղավաղված ԴՎԱ անցումների մատրիցը: Կախված նրանից, թե արդյոք կլիենտ ցանկանում է սերվերից գաղտնի պահել $\Gamma(X)$–ը՝ GM_Γ մատրիցի վերջին տողը կարող է լինել գաղտնագրված, այդ դեպքում սերվերը հաշվարկի արդյունքում կստանա ոչ թե $\Gamma(X)$-ը, այլ՝ դրա ծածկագիրը, որը կլիենտը ստանալուն պես կգաղտնազերծի:

4.4 բաժնում բացատրված է, թե ինչու է հաղորդակարգը ապահով չարակամ սերվերի և ամբողջությամբ անվտանգ չարակամ կլիենտի նկատմամբ (private against a malicious server and fully secure against a malicious client), երկկողմանի հաղորդակցման ապահովության ու անվտանգության ստանդարտ՝ սիմուլյացիոն, սահմանման դեպքում (simulation-based definitions for secure and privacy): **4.5 բաժնում** ներկայացված են հաղորդակարգը իրականացնող C++ գրադարանը և այդ գրադարանի աշխատանքը ցուցադրող ծրագիրը:



(ա) Γ ԴՎԱ (բ) M_Γ ԴՎԱ-ի անցումների մատրից

(գ) PM_Γ ԴՎԱ-ի տեղափոխված անցումների մատրից

Նկար 2

4.6 բաժնում ասվում է, որ կատարվել է մշակված ծրագրի արագագործության ստուգում $|\Sigma| = 2$ դեպքի համար 64-բիթանոց Windows 7 ՕՀ-ով աշխատող Intel® Core™ 2 Quad Q6600 2.4 GHz պրոցեսորով և 4GB օպերատիվ հիշողությամբ համակարգչի վրա: Ստուգումների արդյունքները բերված են առաջին և երկրորդ աղյուսակներում: Առաջին աղյուսակում ցուցադրված են աղավաղված ԴՎԱ անցումների մատրիցի ստեղծման համար ծախսված միջին ժամանակը n-ի և $|Q|$-ի տարբեր արժեքների դեպքում: Երկրորդ աղյուսակում ցուցադրված են հաղորդակարգում օգտագործվող 1-ը-N-ից ենթահաղորդակարգի, ինչպես նաև աղավաղված ԴՎԱ անցումների մատրիցի կիրառման (evaluation) վրա ծախսված ժամանակները:

Աղյուսակ 1՝ աղավաղված ԴՎԱ անցումների մատրիցի կառուցման ժամանակը (վրկ.)

$\lvert Q\rvert$ \ n	10	100	1000	10000	100000
10	<0.001	0.002	0.017	0.17	1.7
100	0.002	0.017	0.17	1.7	17
1000	0.017	0.17	1.7	17	>100
10000	0.17	1.7	17	>100	>100
100000	1.7	17	>100	>100	>100
1000000	17	>100	>100	>100	>100

Աղյուսակ 2՝ 1-ը-N-ից գաղտնի ընտրման փուլի ժամանակը (վրկ.)

n	1-ը-N-ից հարցման կառուցում և պատասխանի վերծանում (սերվեր)	1-ը-N-ից հարցման պատասխանի կառուցում (client)	Աղավաղված ԴՎԱ անցումների մատրիցի կիրառում (սերվեր)
10	<0.001	<0.001	<0.001
100	0.002	<0.001	<0.001
1000	0.021	0.007	<0.001
10000	0.21	0.07	0.002
100000	2.1	0.7	0.02
1000000	21	7	0.2

Երկրորդ դեպքում կախվածությունը արտահայտված է միայն n-ից, քանի որ դրանում ներկայացված գործողությունները որևէ կերպ կախված չեն ԴՎԱ-ից, այլ կախված են միայն X տողից:

Աղյուսակ 3՝ նմանատիպ հաղորդակարգերի բարդությունները

	Հաղորդակցման փուլեր	Կլիենտ		Սերվեր		Ցանցի թրաֆիկը
		Ասիմետրիկ	Սիմետրիկ	Ասիմետրիկ	Սիմետրիկ	
Troncoso	$O(n)$	$O(n\lvert Q\rvert)$	չկա	$O(n\lvert Q\rvert)$	$O(n\lvert Q\rvert)$	$O(n\lvert Q\rvert k)$
Frikken	2	$O(n+\lvert Q\rvert)$	$O(n\lvert Q\rvert)$	$O(n+\lvert Q\rvert)$	$O(n\lvert Q\rvert)$	$O(n\lvert Q\rvert k)$
Gennaro	$O(\min\{\lvert Q\rvert, n\})$	$O(n\lvert Q\rvert)$	չկա	$O(n\lvert Q\rvert)$	չկա	$O(n\lvert Q\rvert k)$
Yao	1	$O(n)$	$O(n\lvert Q\rvert\log\lvert Q\rvert)$	$O(n)$	$O(n\lvert Q\rvert\log\lvert Q\rvert)$	$O(kn^2)$
Ishai	1	$O(n)$	չկա	$O(n\lvert Q\rvert)$	չկա	$O(n\lvert Q\rvert k)$
Mohassel	1	$O(n)$	$O(n)$	$O(n)$	$O(n\lvert Q\rvert)$	$O(n\lvert Q\rvert k)$
սա՝ $\lvert\Sigma\rvert=2$ դեպքում	1	չկա	$O(n\lvert Q\rvert)$	չկա	$O(n)$	$O(n\lvert Q\rvert k)$

4.7 բաժինը չորրորդ գլխի ամփոփումն է. այնտեղ բերված է ներկայացվող և այլ հաղորդակարգերի տարբեր գործողությունների բարդությունների աղյուսակ:

ԱՇԽԱՏԱՆՔԻ ՀԻՄՆԱԿԱՆ ԱՐԴՅՈՒՆՔՆԵՐԸ

1. Մշակվել է մեթոդ՝ որը սիմետրիկ գաղտնագրման սպիտակ արկղի կիրառմամբ, թույլ է տալիս առանց ՈԳ սիմետրիկ սխեմաների բանալիները օգտագործողներին բաշխելու վերջիններիս տալ որոնման և/կամ փոփոխման հարցումներ իրականացնելու հնարավորություն: Մեթոդը ցուցադրելու համար մշակվել է նոր ՈԳ սխեմա և այն իրականացնող ծրագիր [1]:
2. «Ուանքրփթոր» ընկերության SkyCryptor ծրագրային համակարգում ներդրվել է գաղտնագրված ֆայլերում որոնում իրականացնելու ճարտարապետություն [2]:
3. Նախագծվել է նոր ՆԱՈ հաղորդակարգ, որը թույլ է տալիս կլիենտին գաղտնի կերպով ստուգել՝ արդյոք իր մոտ եղած կամայական մուտքի Σ այբուբենով կանոնավոր արտահայտությունը համապատասխանում է սերվերի մոտ եղած n-երկարության տողին $X \in \Sigma^n$: Ստեղծվել է հաղորդակարգը իրականացնող գրադարան և գրադարանի աշխատանքը ցուցադրող ծրագիր [3,4]:

ԱՏԵՆԱԽՈՍՈՒԹՅԱՆ ԹԵՄԱՅՈՎ ՏՊԱԳՐՎԱԾ ՀՈԴՎԱԾՆԵՐԸ

[1] Hovsepyan M. ***Review of Searchable Encryption Algorithms.*** «Вестник» Российско-Армянский (Славянский) Университет, 2015 № 2, стр. 39-53.

[2] Jivanyan A., Hovsepyan M. ***Implementation Aspects of Search Functionality over Encrypted Cloud Data.*** Transactions of IIAP of the NAS RA, Mathematical Problems of Computer Science, vol. 44, pp. 101-108, 2015.

[3] Khachatryan G., Hovsepyan M., Jivanyan A. ***Efficient Secure Pattern Search Algorithm.***

a. In Proc. of 10th International Conference on Computer Science and Information Technologies (CSIT15). Yerevan, Armenia, 2015, pp. 147-151.

b. IEEE: Conference, Sept. 28 2015-Oct. 2 2015, pp. 90 - 94.

[4] Khachatryan G., Hovsepyan M., Jivanyan A. ***Two-Party Regular Expression Matching Protocol without Asymmetric Cryptography Operations.*** Transactions of IIAP of the NAS RA, Mathematical Problems of Computer Science, vol. 45, pp. 77-89, 2016.

Овсепян Мигран Меружанович

Разработка безопасных алгоритмов поиска на зашифрованных данных

РЕЗЮМЕ

В последние годы спектр предоставляемых онлайн IT услуг сильно увеличился, в частности появились облачные хранилища (например, Dropbox, Google Drive, Microsoft OneDrive и т.д.). Услугами таких хранилищ активно пользуются как частные лица, так и компании для хранения личной и конфиденциальной информации. Однако нет никакой гарантии, что провайдер облачного хранилища не будет пытаться несанкционированно использовать данные пользователей или предоставлять третьим лицам доступ к ним. Поэтому, чтобы гарантировать защиту пользовательских данных, они должны храниться в зашифрованном виде. Стандартные методы шифрования гарантируют нужную защиту данных, однако строго ограничивают пользовательские возможности, в частности делая невозможным поиск нужной информации по зашифрованным данным. В такой ситуации, чтобы получить доступ к требуемой информации, пользователь должен либо точно знать в каких файлах нужная информация находится – загрузить, а потом расшифровать эти файлы, либо загрузить все файлы, расшифровать их, а затем сделать поиск по расшифрованным файлам. В случае большого количества файлов первый вариант не реалистичный, а второй требует много ресурсов, так как данных может быть очень много. В этих обстоятельствах и появилась нужда в разработке схем *шифрования допускающих поиск* (ШДП), при этом поиск должен производиться без нарушения конфиденциальности, то есть провайдер хранилища не должен узнать, что пользователь искал и что он нашел в результате своего поиска.

Задача *безопасного шаблонного поиска* (БШП) является другой задачей, связанной с шифрованием и поиском. В этой задаче сервер хранит и является владельцем данных. Клиент же хочет, не отправляя серверу свой шаблон (кусок текста, регулярное выражение и т.д.), узнать, соответствуют ли данные (или их часть) на сервере этому шаблону. При этом, по желанию клиента, ответ запроса также должен быть скрыт от сервера. Сервер же не хочет предоставлять клиенту информацию больше, чем ответ на его запрос.

Основная цель

- С использованием, так называемого белого ящика симметричного шифрования, без предоставления пользователям ключей симметричной схемы ШДП, предоставить им возможность делать запросы поиска и/или изменения. Для иллюстрации метода разработать и реализовать схем ШДП с хорошими, с точки зрения практичности, характеристиками.
- Исследовать программные системы улучшающие качество безопасности облачных хранилищ (Google Drive, Dropbox и т.д.) с помощью шифрования и возможность внедрения в них схем ШДП.
- Разработать новый эффективный протокол БШП, который позволит клиенту проверить соответствует ли его регулярному выражению строка у сервера, без предоставления серверу само регулярное выражение и не получая от него эту строку.

Протокол не должен использовать операции ассиметричного шифрования, взамен он должен использовать белый ящик симметричного шифрования, за счет которого протокол должен значительно превышать похожие протоколы по быстродействию. Разработать библиотеку и программу для тестирования эффективности нового протокола.

Научная новизна

- Новый метод, который позволяет, без предоставления пользователям ключей симметричной схемы ШДП, предоставить им возможность делать запросы поиска и/или изменения.
- Архитектура, которая позволяет хранить файлы в облачных хранилищах (Google Drive, Dropbox и т.д.) в зашифрованном виде, при этом, предоставляя возможность безопасного поиска над этими файлами.
- Новый эффективный протокол БШП, который позволяет клиенту проверить соответствует ли его регулярному выражению с произвольным входным алфавитом Σ строка у сервера.

Внедрение: Была разработана новая схема ШДП, которая была внедрена в систему SkyCryptor, предназначенного для улучшения уровня безопасности облачных хранилищ Dropbox и Google Drive.

Практическая значимость полученных результатов

В рамках диссертации была разработана новая схема ШДП, которая позволяет владельцу хранить свои данные на сервере в зашифрованном виде, при этом, имея возможность предоставлять третьим лицам возможность делать безопасные запросы поиска и/или изменения над этими данными. Схема может быть использована в работе с файлами, которые имеют юридическую или коммерческую ценность. Например: предоставить возможность работникам юридической фирмы делать поиск среди архивов компании, хранимых в зашифрованном виде не предоставляя им ключи шифрования.

Разработанный новый протокол БШП за счет своего быстродействия и высокого уровня предоставляемой безопасности может быть использован в исследовании ДНК, банковском деле, удаленной диагностике и в других сверах.

Mihran Hovsepyan

Development of Secure Search Algorithms over Encrypted Data

ABSTRACT

In recent years, the range of online IT-services has dramatically increased. Particularly this is depended on the appearance of the public cloud storages (e.g. Dropbox, Google Drive, Microsoft OneDrive etc.). These storages are actively used by both individuals and companies to store personal and confidential information. However, there is no guarantee that the cloud storage provider will not attempt to illegally use the stored user data or make it accessible to third parties. Therefore, to ensure the protection of the user data, they should be stored in encrypted form. The standard encryption techniques ensure the necessary protection of the data, but strictly limit the user experience, making it impossible to search the information over the encrypted data. In such a situation, to gain access to the desired information, the user must either exactly know which files contain the necessary information and download then decrypt these files or download all the files, decrypt them, and then search over decrypted files. In the case of the large number of files, the first option is not realistic, and the second one requires a huge computational and memory resources. Under these circumstances, there was a need for the appearance of the searchable encryption (SE) schemes. These schemes allow encrypting files and constructing a special secure-index based on the keywords of these files. This secure-index allows searching files by keywords in such a way that the storage provider does not learn what was searched and what was found as a result.

Secure pattern matching/search (SPM) is another problem related to *encryption* and *search*. In this problem, the server is the owner of the data and the client wants to check whether his pattern (a piece of text, regular expressions, etc.) matches the data (or its part) without sending the pattern to the server. And the server itself does not want to send the data to the client as well as any information more than the answer of the request.

The main objective

- Using so called white-box cryptography delegate the ability of running search and modification queries to the third party users of a symmetric SE scheme without proving them the keys of the scheme. For illustrating the method design and implement a practical symmetric SE schemes.
- Investigate systems intended for securing public cloud storages (Google Drive, Dropbox, etc.) using encryption mechanisms and possibility of embedding an SE scheme into them.
- Develop a new efficient SPM protocol which allows a client to check whether his regular expression matches to the string of the server without sending the regular expression to the server and without getting the string from the later. The protocol must not use public-key operations. This should make it significantly faster than the other similar protocols. Develop a library and corresponding application to test the efficiency of the new protocol.

Scientific Novelty

- A novel method for delegating the ability of running search and modification queries to the third party users of a symmetric SE scheme without proving them the keys of the scheme.
- Architecture which allows to store files in public cloud storages (Google Drive, Dropbox etc.) in encrypted form wherein allows to run secure search queries over those files.
- A new efficient SPM protocol which allows the client to check if his regular expressions with arbitrary input alphabet matches to the input string of the server.

Integration: Search functional architecture was designed and embedded into the SkyCryptor system intended for securing Dropbox and Google Drive public storages using encryption.

Practical Significance of Results

The new SE schemes designed and implemented within this dissertation allows the data owner to store his files on the server in encrypted form and give ability to third parties to run secure search and/or modification queries over these encrypted files without providing them the encryption keys. The scheme can be used mostly when working with legally or commercially sensitive files. For example a law firm wants to give ability to his employees to search over the archive of the documents encrypted and stored in the server without providing them the encryption keys.

The designed new SPM protocol due to its high performance and security characteristics may be used in privacy preserving DNA computations, banking, remote diagnostics etc.