Әл-Фараби атындағы Қазақ ұлттық университеті

ӘОЖ 004.056 Қолжазба құқығында

САҚАН ҚАЙРАТ САҚАНҰЛЫ

Итерациялық блоктық шифрларға негізделген хеш алгоритмдерін құру және олардың криптоберіктілігін зерттеу

8D06301 – Ақпараттық қауіпсіздік жүйелері

Философия докторы (PhD)
дәрежесін алу үшін дайындалған диссертация

Отандық ғылыми кеңесші:
Нысанбаева С.Е.
т.ғ.д., доцент
Шетелдік ғылыми кеңесші:
Andrzej Smolarz
т.ғ.д., профессор
(Польша, Люблин техникалық университеті)

Қазақстан Республикасы
Алматы, 2023

МАЗМҰНЫ

НОРМАТИВТІК СІЛТЕМЕЛЕР

Бұл диссертацияда келесі нормативтік құқықтық актілерге сілтемелер қолданылды:

1. Халықаралық стандарттау ұйымы мен Халықаралық электротехникалық комиссия бірлесіп әзірлеген ISO/IEC 27001:2013 халықаралық ақпараттық қауіпсіздік стандарты;
2. «Электрондық құжат және электрондық цифрлық қолтаңба туралы» 2003 жылғы 7 қаңтардағы Қазақстан Республикасының Заңы.
3. «Дербес деректер және оларды қорғау туралы» 2013 жылғы 21 мамырдағы Қазақстан Республикасының Заңы;
4. Қазақстан Республикасы Президентінің 2023 жылғы 20 наурыздағы № 145 Жарлығымен бекітілген Қазақстан Республикасының Ақпараттық Доктринасы;
5. ҚР Үкіметінің 2017 жылғы 30 маусымдағы № 407 Қаулысымен бекітілген Киберқауіпсіздік («Қазақстанның киберқалқаны») тұжырымдамасы»;
6. «Мәліметтерді таратылуы шектелген қызметтік ақпаратқа жатқызу және онымен жұмыс істеу қағидаларын бекіту туралы» Қазақстан Республикасы Үкіметінің 2022 жылғы 24 маусымдағы № 429 Қаулысы;
7. «Ақпараттық-коммуникациялық технологиялар және ақпараттық қауіпсіздікті қамтамасыз ету саласындағы бірыңғай талаптарды бекіту туралы» Қазақстан Республикасы Үкіметінің 2016 жылғы 20 желтоқсандағы № 832 Қаулысы.
8. СТ РК 1073-2007 – Ақпаратты криптографиялық қорғау құралдары.
9. «Диссертацияларды және авторефераттарды рәсімдеу бойынша нұсқаулық», ҚР БҒМ, Жоғары аттестаттау комитеті, Алматы, 2004. МЕСТ 7.1-2003. Библиографиялық жазба.
10. ГОСТ 7.32-2001 – Ғылыми-зерттеу жұмысының есебі.

БЕЛГІЛЕУЛЕР МЕН ҚЫСҚАРТУЛАР

АЕТИ	–	Ақпараттық және есептеуіш технологиялар институты
АҚЗ	–	Ақпараттық қауіпсіздік зертханасы
АККҚ	–	Ақпаратты криптографиялық қорғау құралдары
АҚЖ	–	Ақпараттық қауіпсіздік жүйелері
ҒЗЖ	–	Ғылыми-зерттеу жұмыстары
ДҚФ	–	дизъюнктивті қалыпты форма
КҚФ	–	конъюнктивті қалыпты форма
ҚР ҒЖБМ	–	Қазақстан Республикасы Ғылым және жоғарғы білім министрлігі
ПЛИС	–	Программируемая логическая интегральная схема, бағдарламаланатын логикалық интегралды сұлба
ЭЕМ	–	Электронды есептеуіш мәшине
ЭЦҚ	–	Электрондық цифрлық қолтаңба
FAT	–	File Allocation Table, файлдардың үлестірілу кестесі
GF	–	Galois field, Галуа өрісі
ISO	–	International Organization of Standardization, халықаралық стандарттау Ұйымы
LAT	–	Linear Approximation Table, сызықтық жуықтау кестесі
MAC	–	Message Authenticate Code, хабарламаны аутентификациялау коды
MD	–	Message Digest, хабарлама ізі
NIST	–	National Institute of Standards and Technology, АҚШ-тың Ұлттық стандарттар және технологиялар институты
SAT	–	SATisfiability problem, логикалық формулалардың орындалуы туралы мәселе
SHA	–	Secure Hash Algorithm, қауіпсіз хештеу алгоритмі

КІРІСПЕ

Күнделікті өмірде ақпараттық технологиялардың кең қолданысқа енуі оң нәтиже ретінде қабылданып қана қоймай, «цифрлық гигиена» шараларын ескермеу салдарынан ақпараттың бөтен қолға түсіп, түрленіп, зиянды әсер факторын туығызуы мүмкін. Ақпараттық жүйелерге шабуылдардың жаңа түрлерінің пайда болуы және бар шабуылдардың модификацияға ұшырауы, сонымен қатар, ақпараттық технологиялардың мүмкіндіктерінің артуы қорғаныс жүйелерін ұдайы дамытуды және жаңартуды талап етеді. Шабуылдаушылар ақпараттың электронды жүйесіне ене отырып, өздеріне қажетті деректерді салыстырмалы түрде оңай ала алу мүмкіндігінің жоқ екендігіне ешкім кепілдік бере алмайды. Бұл тұжырым деректерді пайдалану және тасымалдау кезінде қауіпсіздік мәселелерін тудырады. Сондықтан, ақпараттық қауіпсіздіктің маңызды механизмдерінің бірі – деректердің құпиялылығын, тұтастығын және қолжетімділігін қамтамасыз етудің ең көп қолданылатын криптографиялық әдістері болып табылатын шифрлау және хештеу жүйелерін пайдалану ұсынылады [1].

Хештеу механизмі бастапқы уақытта мәліметтің тұтастығын тексеру үшін пайдаланылды, бірақ қазір информатикада және маңызды деректер мен операцияларды оңтайландыру үшін бағдарламалау саласында да кеңінен қолданылады. Хеш функциялары аутентификацияны, ақпараттың тұтастығын тексеруді, авторлықты растау және бас тарта алмау құқығын, деректер мен файлдарды қорғауды, соның ішінде кейбір жағдайларда зиянды бағдарламаларды анықтауды және т.б. орындау үшін пайдаланылады.

Бүгінгі күні жаңа конструкциялар мен оларды құру әдістерін қолдану арқылы көптеген жаңа хеш-функциялар құрылған. Шартты түрде хеш функцияларын құруды үш санатқа бөлуге болады: блоктық шифрларға негізделген хеш функциялары, арифметикалық функцияларға негізделген хеш функциялары және арнайы хеш функциялары.

Әзірленетін хеш функциялар қатаң қауіпсіздік қасиеттері бойынша тексерулерінен өтуі керек. Блоктық шифрлар негізінде тиімді хеш-функцияны жобалау кезінде оларды бағдарламалық және бағдарламалы-аппараттық қамтамасыз етуді жүзеге асыруға мүмкіндік беретін жақсы зерттелген криптографиялық түрлендірулер мен конструкцияларды пайдалану ұсынылады.

Зерттеу тақырыбының өзектілігі ақпараттық технологиялардың үдемелі дамуымен және ақпараттық қауіпсіздікті, оның ішінде ақпараттардың тұтастығы (бүтіндігі) мен авторлықты растау және бас тарта алмау құқығын қамтамасыз ету мақсатында ақпаратты қорғаудың қолданыстағы модельдерін жетілдіру қажеттілігінен туындайды. Қазіргі жағдай ақпаратты қорғау қажеттілігі тек мемлекеттік секторға ғана емес, сонымен қатар үкіметтік емес ұйымдарға және қарапайым пайдаланушыға да қажет.

Қазақстанда электрондық ақпаратты қорғау үшін қолданыстағы электрондық жүйелерде негізінен халықаралық стандарттар мен шетелдік криптографиялық құралдар және бағдарламалық жасақтамалар қолданылады, сондықтан отандық криптографиялық қорғау құралдарын құру сөзсіз өзекті және

күн тәртібіндегі кейінге шегеруге болмайтын мәселе. Ақпараттың құпиялығы, бүтіндігі мен авторлықты растау және бас тарта алмау құқығын бақылау үшін отандық өнімдерді құру біздің еліміз үшін шұғыл міндет болып табылады. Бұл жұмыс отандық ақпараттық қауіпсіздік жүйелерін әзірлеуге және оларды практикалық тұрғыда пайдалану үшін бағдарламалық-аппараттық кешендерді құруға бағытталуы тиіс. Осы бағытта елімізде бірқатар статегиялық басымдықтар анықталып, соның негізінде қабылданған төменгі нормативтік құқықтық актілер ұсынылған диссертациялық жұмыстың өзектілігін растайды [2-8]:

– Қазақстан Республикасы Президентінің 2023 жылғы 20 наурыздағы № 145 Жарлығымен бекітілген Қазақстан Республикасының Ақпараттық Доктринасы;

– «Электрондық құжат және электрондық цифрлық қолтаңба туралы» 2003 жылғы 7 қаңтардағы Қазақстан Республикасының Заңы;

– «Дербес деректер және оларды қорғау туралы» 2013 жылғы 21 мамырдағы Қазақстан Республикасының Заңы;

– «Мәліметтерді таратылуы шектелген қызметтік ақпаратқа жатқызу және онымен жұмыс істеу қағидаларын бекіту туралы» Қазақстан Республикасы Үкіметінің 2022 жылғы 24 маусымдағы № 429 Қаулысы;

– «Ақпараттық-коммуникациялық технологиялар және ақпараттық қауіпсіздікті қамтамасыз ету саласындағы бірыңғай талаптарды бекіту туралы» Қазақстан Республикасы Үкіметінің 2016 жылғы 20 желтоқсандағы № 832 қаулысы;

– ҚР Үкіметінің 2017 жылғы 30 маусымдағы № 407 қаулысымен бекітілген «Киберқауіпсіздік («Қазақстанның киберқалқаны») тұжырымдамасы»;

– Криптографиялық қорғау құралдарына мемлекеттік стандарт (СТ РК 1073-2007).

Хештеу механизмінің қолдану аясы өте кең екендігін, ақпараттық технологиялар саласының қарқынды дамуын ескере отырып, хештеудің жаңа әдістерін әзірлеу және оларды зерттеу, оның ішінде параллелизмнің жоғары дәрежесі бар жаңа есептеуіш архитектураларды пайдалану ақпаратты қорғау саласында өзекті міндеттің бірі болып табылады. Қазіргі таңда хеш-функциялардың алуан түрлері бар және осы саладағы зерттеулер әлі де жалғасуда. Хеш-функцияны құрудың итерациялық тізбекті схемаларын блоктық шифрлау алгоритмі негізінде құру және оны сенімділікке зерттеу диссертациялық жұмыстың негізгі бағыты болады.

Диссертациялық жұмыстың мақсаты. Симметриялы блоктық шифрлау алгоритмі негізінде сенімділігі мен өнімділігі жағынан жоғары, бағдарламалы-аппараттық жүзеге асыруға және параллелдік есептеуге икемделген хештеу алгоритмін құру және оның қауіпсіздік қасиеттері мен тиімділігін зерттеу.

Зерттеу міндеттері:

1. Заманауи хеш функцияларға сараптама жүргізу, коллизияларды зерттеу әдістерін талдау, шабуылдардың үлгілері мен криптоталдауды зерделеу;

2. блоктық шифрға негізделген жаңа хеш алгоритмнің архитектурасын құру;

3. қысу функциясы ретінде қолданылатын блоктық шифрлау алгоритмін құру;

4. құрылған хештеу алгоритмінің қауіпсіздік қасиеттерін статистикалық сынақтар және криптоталдау әдістері арқылы зерттеу;

5. құрылған хештеу алгоритмінің бағдарламалық және бағдарламалы-аппараттық жүзеге асыру, сондай-ақ тиімділігін талдау.

Зерттеу нысаны. Криптографиялық хештеу және шифрлау жүйелері.

Зерттеу пәні. Блоктық шифрларға негізделген хеш функциялар және олардың қауіпсіздік қасиеттері.

Зерттеу құралы мен әдісі. Жұмыста бульдік функция теориясы, сызықтық алгебра, ықтималдықтар теориясы және математикалық статистика, хеш алгоритмге жүргізілетін криптографиялық талдау әдістері мен шабуылдар түрлері, биттік шашырау критерийлері қолданылды.

Жұмыстың ғылыми жаңалығы:

- хеш функцияларға тән қасиеттерге ие болатын және оларға қойылатын жалпы талаптарға сай келетін, блоктық шифрларға негізделген, паралелльді есептеуге икемделген жаңа хештеу алгоритмі құрылды;
- төрт 4-биттік S-блок ауыстыру түйіндерін элементтің индекстеріне қатысты жұптастырып қолданудың жаңа сұлбасы ұсынылды, оны қолдану алгоритмнің қауіпсіздігін арттыруға және аппараттық жүзеге асыруда микросхеманың жадын тиімді пайдалануға мүмкіндік береді;
- қысу функциясындағы сызықты емес түрлендіруді қолданудың жаңа сұлбасы ұсынылып, оның раундтар санын азайтуға мүмкіндік беретіні көрсетілді;
- хабарлама блогының ұзындығын оның көлеміне байланысты k бөліктер санын өзгерту мүмкіндігі ұсынылды, ол өз кезегінде есептеу өнімділігін арттыратыны анықталды (k=3,…, 8, k - бөліктер саны).

Зерттеудің теориялық және практикалық құндылығы. Жүргізілген ғылыми зерттеулердің теориялық және алынған нәтижелердің практикалық құндылығы электрондық құрылғыларда, деректерді тасымалдаудың және сақтаудың арнайы жүйелерінде ақпаратты қорғаудың криптографиялық құралдарын пайдалану мүмкіндігін арттырады және нәтижесінде отандық ақпараттық жүйелерді дамыту үшін жаңа мүмкіндіктер ашады.

Жасалған HBC-256 хештеу алгоритмі 2022 жылы Алматы қаласындағы «Guppyprint» баспасынан т.ғ.д., профессор Р.Бияшевтың басшылығымен жарық көрген «Разработка и исследование алгоритмов хеширования произвольной длины» монографиясында жеке бөлім ретінде енгізілді (ISBN 978-601-08-2549-9, 95 бет).

Зерттеу жұмысы нәтижелері SCOPUS және Web of Science халықаралық деректер қорына кірген журналдарда, сондай-ақ ҚР ҒЖБМ-нің Білім және ғылым саласы бойынша бақылау комитетімен ұсынылған басылымдарда ғылыми мақалалар болып жарияланды (Қосымша А).

Аталған хештеу алгоритмнің тәуелсіз зерттеулері нәтижелері Ресей Федерациясы Новосибирск мемлекеттік университеті, «Криптографиялық Орталық» (Новосибирск қ.) және «Академгородок халықаралық математикалық Орталығы» ұйымдастыруымен 2022 жылы өткен «Криптография и информационная безопасность» жаздық мектеп-конференцияның еңбектері жинағына «Исследование криптографических свойств новых функций хэширования HBC и HAS01» (авторлары – А.Е. Доронин, Д.А. Зюбина, Е.А. Ищукова, Н.А. Коломеец, А.В. Куценко, Э.А. Пивнева, И.А. Сутормин) тақырыбымен енгізілді.

Онымен қоса, ғылыми туынды ретінде ҚР ӘМ Ұлттық зияткерлік меншік институтынан «Алгоритм хеширования данных «HBC-256» 2021 жылғы 20 қыркүйектегі № 20318 авторлық куәлігі, алгоритмнің бағдарламалық жасақтамасы жасалып, «ISL_HASH 1.0» 2021 жылғы 5 қазандағы № 20661 авторлық куәлігі, «CSP_HASH 1.0» 2022 жылғы 21 ақпандағы № 23886 авторлық куәлігі «Криптопровайдер ISL_CSP 1.0» 2022 жылғы 12 қазандағы № 29379 авторлық куәлігі алынды (Қосымша Ә).

Қорғауға шығарылған негізгі тұжырым. Заманауи хеш функцияларға тән қасиетке ие болатын, оларға қойылатын жалпы талаптарға сай келетін, блоктық шифрларға негізделген жаңа хештеу алгоритмі құрылды. Құрылған хештеу алгоритмінің қауіпсіздігі қасиеттерін статистикалық сынақтары, лавиндік әсер критерийі, «жақын коллизиялар» әдісі, сондай-ақ дифференциалдық, сызықтық және алгебралық криптоталдаулар түрлері бойынша зерттелді.

Сенімділік дәрежесі мен апробациялау нәтижелері. Диссертациялық жұмыс бойынша жүргізілген зерттеулер мен нәтижелерінің сенімділігі екінші, үшінші және төртінші бөлімдерде көрсетілген.

Зерттеулер нәтижесі төменде көрсетілген ғылыми-практикалық конференцияларда, сондай-ақ отандық және шетелдік ғылыми-зерттеу институттары мен оқу орындарындағы ғылыми семинарларда баяндалды және талқыланды (Қосымша Б):

1) «Информатика және қолданбалы математика» V Халықаралық ғылыми-тәжірибелік конференциясында (Алматы, 29 қыркүйек – 1 қазан 2020);

2) «Қазақстандағы ақпараттық қауіпсіздіктің өзекті мәселелері» Халықаралық ғылыми-тәжірибелік конференциясында (АПБИК-2021, Алматы, 11 маусым 2021);

3) «Информатика және қолданбалы математика» VI Халықаралық ғылыми-тәжірибелік конференциясында (Алматы, 29 қыркүйек – 2 қазан 2021);

4) Әл-Фараби атындағы Қазақ ұлттық университеті профессорі У.А. Тукеевтің 75 жылдық мерейтойына арналған ақпараттық технологиялар саласындағы Халықаралық ғылыми конференцияда (Алматы, 8 қазан 2021);

5) IV халықаралық «Минские научные чтения-2021. Передовые технологии и материалы будущего» ғылыми-техникалық конференциясында (Минск, Беларусь, 9 – 10 желтоқсан 2021);

6) “Computer Data Analysis and Modeling: Stochastics & data Science” (CDAM-2022) халықаралық конференцияда (Минск, Беларусь, 6 – 9 қыркүйек 2022);

7) «Информатика және қолданбалы математика» VII Халықаралық ғылыми-тәжірибелік конференциясында (Алматы, 20 – 21 қазан 2022);

8) Украина Ұлттық авиациялық университеті «Киберқауіпсіздік, компьютерлік және бағдарламалық инженерия» факультетінің (ФКбКПИ НАУ) ғылыми семинарында (Киев, Украина, 3 желтоқсан 2021);

9) Беларусь мемлекеттік университеті «Математика және информатиканың қолданбалы мәселелері» ғылыми зерттеу институты ғылыми семинарында (НИИ ППМИ БГУ) (Минск, Беларусь, 6 қыркүйек 2022);

10) Electrical Engineering and Computer Science Department of Khalifa University ғылыми семинарында (Абу-Даби, БАӘ, 13 желтоқсан 2022);

11) «Ақпараттық және есептеуіш технологиялар» институты ғылыми семинарларында (2020 – 2023жж., Алматы);

12) Әл-Фараби атындағы Қазақ ұлттық университетінің «Ақпараттық технологиялар» факультеті ғылыми семинарларында (2020 – 2023 жж., Алматы).

Диссертациялық тақырыптың ғылыми бағдарламалармен байланысы. Диссертациялық жұмыс Қазақстан Республикасының Ғылым және жоғарғы білім министірлігі Ғылым комитетінің Ақпараттық және есептеуіш технологиялар институтында бекітілген PhD докторлық диссертациялар жоспарына және ЖТН – OR11465439 «Электрондық цифрлы қолтаңба үшін еркін ұзындықтағы хештеу алгоритмін құру мен зерттеу және олардың беріктілігін бағалау» бағдарламалық-нысаналы қаржыландыру жобасының ғылыми-зерттеу жұмыстарының аясында орындалды. Диссертациялық жұмыс бойынша жүргізілген зерттеу жұмыстарының нәтижесі аталған БНҚ жобасының 2021-2022 жылдарындағы есебіне енгізіліп, «Енгізу актісі» алынды (Қосымша В).

Жұмыс көлемі мен құрылымы. Диссертациялық жұмыс кіріспе, 4 бөлім, қорытынды және пайдаланылған әдебиеттер тізімі мен қосымшалардан тұрады. Диссертацияның толық көлемі: 103 бет жазба мәтіні, оның ішінде 30 сурет, 29 кесте, 99 пайдаланылған әдебиеттер тізімінен және 6 қосымшадан тұрады.

Нәтижелердің жариялаңымдары. Зерттеу жұмыстарын орындау кезеңінде ғылыми зерттеу нәтижесі ретінде жалпы саны 24 ғылыми еңбек жарық көрді. Оның ішінде 7 мақала Scopus және Web of Science базаларында индекстелген журналдарда, 1 отандық монография, 7 мақала Қазақстан Республикасы ғылым және жоғарғы білім министрлігінің білім және ғылым саласы бойынша бақылау комитетімен ұсынылған басылымдарда, 10 мақала халықаралық және отандық ғылыми-практикалық конференциялар жинақтарында және басқа да ғылыми журналдарда жарияланды.

Кіріспеде диссертациялық жұмыс тақырыбы бойынша алғысөз және тақырып өзектілігінің негіздемесі баяндалады. Осы бөлімде ғылыми-зерттеу жұмысының мақсаттары, нысаны және зерттеу пәні көрсетілді. Сондай-ақ, жұмыстың ғылыми жаңалығы, тәжірибелік маңызы және жұмыстардың нәтижелерінің апробациясымен қоса жариялаңымдары туралы мағлұматтар беріледі.

Бірінші бөлімде ақпаратты қорғауда қолданыстағы хештеу алгоритмдердің түрлері және жалпы криптографияда хеш функцияларға қатысты пайдаланылған

негізгі ұғымдар мен анықтамаларға тоқталған. Хеш функцияларға қойылатын негізгі және қосымша талаптарға сипаттама жүргізіліп, олардың негізгі қасиеттері сараланады. Бөлімнің соңында хеш алгоритмдерінің сапасын бағалау критерийлері мен хеш алгоритмдерге жасалатын шабуылдарға жіктеу жасап, сараптама жүргізіледі.

Екінші бөлімде хеш функцияларға қойылатын талаптарды ескере отырып, оның негізгі қасиеттеріне ие бола алатындай етіп, итерациялы симметриялы блоктық шифрлау алгоритмі негізінде HBC-256 хештеу алгоритмі ұсынылады. Блоктық шифрлау алгоритмі ретінде SP-желісі негізінде құрылған CF алгоритмі қарастырылады. Құрылған CF шифрлау алгоритмде қолданылған криптографиялық примитивтер мен түрлендірулерге жеке-жеке сипаттамалар беріледі. Есептеу өнімділігін арттыру мақсатында раундтар санын неғұрым минималды ету үшін жаңа сұлба ұсынылады. Жадыдағы орынды үнемдеу үшін 4-биттік төрт S-блок ауыстыру кестесі қолданылды және оларды тиімді пайдалану қағидасы көрсетіледі.

Үшінші бөлімде құрылған HBC-256 хештеу алгоритмінің қауіпсіздігі қасиеттеріне зерттеулер жүргізіледі. Атап айтқанда, алгоритмнің қауіпсіздік қасиеттері теориялық тұрғыдан бағаланып, одан әрі NIST және Д.Кнут сатистикалық сынақтар жиынтығы бойынша хеш-мәннің псевдокездейсоқтық қасиетке ие болу деңгейіне баға беріледі. Келесі кезекте хабарлама мен хеш-мән арасындағы қатынасты сипаттайтын талап – лавиндік және қатаң лавиндік әсері зерттеледі. Одан кейін, «жақын коллизияларға», дифференциалдық, сызықтық және алгебралық криптоталдау әдістері бойынша коллизияларды табу мүмкіндіктері бағаланады.

Төртінші бөлімде құрылған хештеу алгоритмі бағдарламалық және бағдарламалы-аппараттық жасақтамалары жайлы мәліметтер көрсетіледі. Осы бөлімде HBC-256 алгоритмін жүзеге асыру түрлеріне байланысты ерекшеліктеріне сипаттамалар беріліп, олардың есептеу өнімділігіне баға беріледі және осы бағытта басқа хештеу алгоритмдеріне қатысты салыстырмалық талдаулар нәтижелері жарияланады.

Қорытындыда ғылыми жұмыстың зерттеу нәтижелері көрсетіліп, олардың бағалаулары тұжырымдалды.

Ғылыми тағылымдамалар. Люблин техникалық университеті, Люблин қаласы, Польша, 2022 жылғы 24 мамыр – 8 тамыз аралығында.

1 ЗАМАНАУИ ХЕШТЕУ АЛГОРИТМДЕРІН ЖОБАЛАУ ЖӘНЕ ОЛАРДЫ ЗЕРТТЕУ ӘДІСТЕРІ

1.1 Заманауи хештеу алгоритмдеріне шолу, хеш функцияға қойылатын талаптар

Хеш функциялар информатикадағы іргелі тұжырымдама болып табылады және деректер құрылымы, криптография және цифрлық қолтаңбалар сияқты әртүрлі салаларда көптеген қолданыстарға ие. Олар әртүрлі пайдалану жағдайларында қауіпсіз және пайдалы ететін детерминирленген және қайтымсыздық сияқты қасиеттерімен танымал және осы жағынан алғанда ақпараттардың қауіпсіздігі, бүтіндігі мен аутентивтігін қамтамасыз ету бағытында үлкен сұранысқа ие.

Хеш функциялар 20 ғасырдың басында деректердің тұтастығын тексеру мақсатында жасалған. 1953 жылы Харви Дэвис пен Гаррет Мейер деректердің тұтастығын тексеру үшін алғашқы хеш функциясын ұсынды. 1964 жылы К. Евклид пен С. Лампорт хеш-кесте деп аталатын мәліметтер құрылымын құру үшін хеш функцияны қолданды. Осы уақыттан бері олар деректер құрылымы, ақпараттық жүйелердің қауіпсіздігі, криптография сияқты әртүрлі салаларда қолданылып келеді. Уақыт өте келе криптографияның дамуымен бірге хеш функциялар сандық қолтаңбаларды жасау үшін, сондай-ақ әртүрлі қосымшалардағы деректердің тұтастығын бақылау үшін қолданыла бастады. 1980 жылдары кеңінен қолданыла бастаған MD4, MD5 және SHA1 сияқты әртүрлі хеш алгоритмдері жасалды. Кейінгі жылдары бұл алгоритмдердегі кейбір әлсіздіктер белгілі болды, бұл SHA-2 және SHA-3 сияқты заманауи және қауіпсіз алгоритмдердің дамуына әкелді. Соңғысы 2015 жылы жасалған және бүгінгі күнге дейін ең сенімді хеш алгоритмдерінің бірі болып табылады. Жалпы, хеш функцияларын дамыту олардың қауіпсіздік қасиеттері мен орындалу жылдамдығын арттыру бағытында жүреді. Қазіргі заманғы хеш функциялары жоғары қауіпсіздік қасиеттері мен өнімділікке ие. Заманауи хеш функцияларының дизайны жылдамдық, коллизияға төзімділік және шабуылдан қорғау сияқты әртүрлі факторларды қамтиды.

IT-индустрияға арналған стандарттар халықаралық техникалық регламенттермен үйлесуі тиіс, өйткені әр мемлекет әлемдік экономикаға интеграциялануда. Ақпараттық қауіпсіздік саласында әрбір мемлекет криптография саласында өзінің ұлттық стандарттарын әзірлеуге ұмтылады. Көптеген ұйымдар мен елдерде хештеудің өзіндік стандарттары бар. 2018 жылы ISO және МЭК (Халықаралық электротехникалық комиссия) ұсынған ISO/IEC 10118-3:2018(E) дүниежүзілік стандарттауға арналған мамандандырылған жүйесінде 17 алгоритм халықаралық хештеу алгоритмдері мәртебесіне ие болды, атап айтқанда, RIPEMD-160, RIPEMD-128, SHA-1, SHA-256, SHA-512, SHA-384, Whirlpool, SHA-224, SHA-512/224, SHA-512/256, Streebog-512, Streebog-256, SHA3-224, SHA3-256, SHA3-384, SHA3-512, SM3 [9].

АҚШ-тың NIST – Ақпараттық технологиялар және криптография саласындағы стандарттарды әзірлейтін ұйымы SHA-1, SHA-2 және SHA-3 сияқты көптеген хеш алгоритмдерінің стандарттарын жариялады. Ұйым 2015

жылы SHA-3 (Keccak хеш функциясы) айнымалы ұзындықты хештеу алгоритмі негізінде FIPS 202 мемлекеттік стандартын бекітіп жариялады. Алгоритм ең көп қолданылатын сұлбалардың бірі болып саналатын Sponge криптографиялық сорғыш құрылымына негізделген [10]. Қазіргі уақытта ғылыми қауымдастық өзінің соңғы нұсқасының беріктігі туралы көптеген зерттеулер жүргізетіні белгілі, өйткені SHA-3-тің алдыңғы нұсқалары бұзылған немесе осалдықтары анықталған. SHA-3-те хештеу процесі екі кезеңнен тұрады: сіңіру және қысу. Бірінші қадамда әрбір тұрақты ұзындықтағы хабарлама блогы R бит матрицаның ағымдағы күйіне қосылады және *f* қысу функциясының 24 раунды орындалады. Екінші қадамда күй матрицасы *f* қысу функциясын итеративті орындау арқылы қажетті хеш-мән ұзындығына дейін кесіледі [11].

2016 жылы Қытай елі GB/T 32905-2016 «Ақпараттық қауіпсіздік технологиясы. SM3 криптографиялық хештеу алгоритмі» стандартын бекітті [12]. SM3-256 биттік хеш алгоритмі *M* хабарламасы үшін 256 биттік хеш-мәнін жасайды және Меркл-Дамгард құрылымын қолданады. Ол негізінен электрондық қолтаңбаларда, криптографиялық бақылау сомаларында және жалған кездейсоқ сандар генераторларында қолданылады.

Ресей Федерациясында ГОСТ Р 34.11-2018 мемлекетаралық стандарты қолданысқа енгізілді. Бұл стандарт 2012 жылғы қабылданған «ГОСТ Р 34.11-2012. Ақпараттық технология. Ақпаратты криптографиялық қорғау. Хештеу функциясы» стандартын қолдану негізінде түзетулер мен толықтырулар бойынша дайындалған. Алгоритм кіріс блогының ұзындығы 512 бит және хеш-мәннің ұзындығы 256/512 бит болатын хеш функциясын сипаттайды. Ол үш түрлендіруге негізделген қысу функциясын қолданады: сызықтық емес биективті түрлендіру, байттық ауыстыру, сызықтық түрлендіру [13]. Бұл стандарт Армения, Қырғызстан, Түрікменстан Республикаларында және Тәжікстанда қолданылады.

2015 жылдан бастап Украинада «DSTU 7564:2014. Ақпараттық технологиялар. Ақпаратты криптографиялық қорғау. Хештеу функциясы» мемлекеттік стандарты енгізілді [14]. Ол ГОСТ 28147:2009 мемлекетаралық стандартын біртіндеп ауыстыруға арналған. Стандарттағы «Купина» хештеу функциясы Дэвис-Мейер схемасын қолданады және оның примитивтері «Калина» блоктық шифрлау алгоритміне негізделген.

2020 жылдан бастап Беларусь Республикасында СТБ 34.101.77-2016 стандарты орнына «СТБ 34.101.77-2020. Ақпараттық технологиялар және қауіпсіздік. Sponge функциясына негізделген криптографиялық алгоритмдер» жаңа стандарты күшіне енді [15]. 1536 бит ұзындықтағы екілік сөздердің күрделі биективті түрленуін анықтайтын Sponge-функциясы bash-f және bash-s алгоритмдерінен тұрады. Хеш-мән ұзындығы беріктілік деңгейі $l \in \{128, 192, 256\}$ параметрімен анықталады, яғни осы мәндерге тең.

Біздің елімізде хештеу алгоритмі бойынша Қазақстан Республикасының Ұлттық стандарты болып ҚР СТ ГОСТ Р 34.11-2015 «Ақпараттық технология. Ақпаратты криптографиялық қорғау. Хэштеу функциясы» 2017 жылдың 1 қаңтарынан бастап қолданысқа енгізілді. Ескере кететіні, бұл ұлттық стандарт Ресей Федерациясының «ГОСТ Р 34.11-2012. Ақпараттық технология.

Ақпаратты криптографиялық қорғау. Хештеу функциясы» стандартының көшірмесі болып табылады [16, 17].

Қазақстан Республикасы негізінен халықаралық стандарттарды және шетелдік аппараттық және бағдарламалық қамтамасыз ету жасақтамаларын пайдаланады. Ақпаратты криптографиялық қорғаудың отандық алгоритмдерін, соның ішінде хештеу алгоритмдерін құру өзекті және қажетті міндет болып табылады.

Хеш функциялар информатика және ақпараттық қорғау саласында негізгі ұғым болып табылады және деректер құрылымы, криптография және цифрлық қолтаңбалар сияқты әртүрлі қолданбаларда кеңінен қолданылады. Қарапайым тілмен айтқанда, хеш функциялар ақпарат тізбегі (хабарлама) немесе кілт (пароль) деп аталатын кірістерді қабылдайды және хеш-мән (кейде хеш-кескін немесе дайджест) деп аталатын тұрақты ұзындықтағы шығыс мәнді шығарады.

Анықтама 1. *H(M)* хеш функциясы – еркін ұзындықтағы *M* ақпарат тізбегін (жолын) кіріс ретінде қабылдайтын және нәтиже ретінде алдын-ала белгіленген ұзындықтағы *h* ақпарат тізбегін (жолын) беретін функция.

Анықтама 2. *M* ақпарат тізбегін хештеу нәтижесі *h=H(M)* – хеш-мән деп аталады.

M ақпарат тізбегі мен *H(M)* хеш-мән ұзындықтары ара-қатынасы әртүрлі болуы мүмкін: мысалы, $|M| > |H(M)|$ немесе $|M| < |H(M)|$ немесе $|M| = |H(M)|$, мұндағы $|*|$ – белгілеуі ақпараттың немесе хеш-мәннің ұзындығын білдіреді. Көп жағдайда ақпарат тізбегінің ұзындығы хеш-мән ұзындығынан едәуір үлкен болып жатады.

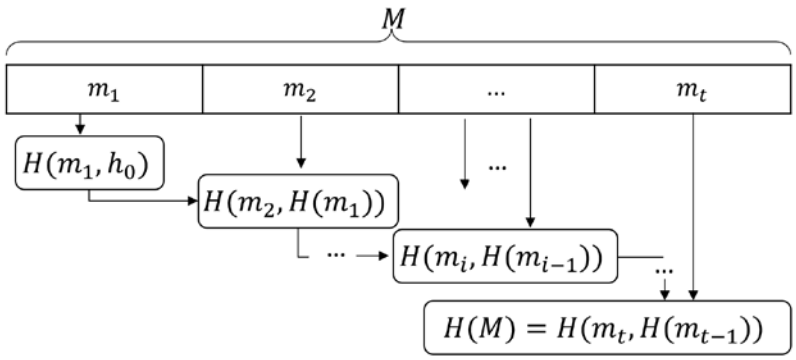
Анықтама 3. Хеш функцияның нәтижесі хеш-мән деп аталатындықтан, M ақпараттар тізбегі кейде түпбейне (прототип немесе бірінші прототип) деп аталады.

Анықтама 4. *m* ұзындықтағы барлық екілік тізбектер жиыны $\{0, 1\}^m$ деп, ал барлық ақырлы ұзындықтағы екілік тізбектер жиыны $\{0, 1\}^*$ деп белгіленсін. Онда *H* хеш функциясы деп мына түрдегі түрлендіруді айтады:

$$H : \{0, 1\}^* \rightarrow \{0, 1\}^m, \qquad (1.1)$$

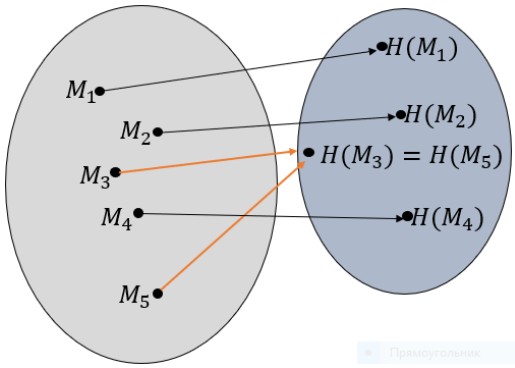
мұндағы m – хеш-мәннің ұзындығы. *H* хеш функциясы кейде h болып белгіленеді.

Хеш функцияның жалпы сұлбасы төменгі Сурет 1.1-де көрсетілген. Мұнда: $M = \{m_0, m_1, \dots, m_t\}$ – итерациялық түрде хештеу үшін *M* хабарламаны бекітілген ұзындықтағы *t* бөлікшелерге бөлінуі, $i=1,2,\dots,t$, t – *M* хабарламадағы бөлікшелер саны, h_0 – инициализациялық вектор (*IV*). *H(M)* – *M* хабарламасының хеш-мәні.



Сурет 1.1 – *H(M)* хеш функцияның жалпы сұлбасы

Анықтама 5. Хеш функциядағы коллизия деп әртүрлі екі M_1 және M_2 хабарламалар үшін $H(M_1)=H(M_2)$ орындалу жағдайын айтады. Басқаша айтқанда, екі әртүрлі кіріс үшін хеш функцияның мәндері тең келеді. Келесі Сурет 1.2-де M_3 и M_5 коллизия тудыратын хабарламалар.



Сурет 1.2 – *H(M)* хеш функциясының M_3 және M_5 хабарламалары тудырған коллизия

Анықтама 6. Әдетте, бекітілген нүкте деп мына жағдайды айтады: $h_i=H(h_{i-1},M_i)=h_{i-1}$, $i=1,2,\ldots,t$, t – M хабарламадағы бөлікшелер саны, $i=1$ болған жағдайда, $h_0=IV$.

Хеш функциясының қауіпсіздігі қасиеттеріне уақыт өте келе есептеу қуаты артқан сайын қауіп төнуі мүмкін екенін ескеру маңызды. Сондықтан олардың әлі де қауіпсіз қасиеттерге ие екеніне көз жеткізу үшін қауіпсіздік қолданбаларында қолданылатын хеш функциялардың қасиеттері мен мүмкіндіктерін үнемі қайта қарап, жаңартып отыру тиіс.

Кез-келген хештеу алгоритмдерін құру кезінде келесідегідей қасиеттерге ие болуы өамтамасы етілуі тиіс:

1. Анықталғандық (детерминирленген) – бірдей кіріс хабарламалар үшін хеш функция әрқашан бірдей нәтижелер беруі.
2. Бірбағыттылық (қайтымсыздық) – бұл хеш функцияның шығыс нәтижесі бойынша бастапқы кірістерді (хабарламаны) математикалық тұрғыдан есептеу алудың мүмкін еместігі.

3. Бекітілген ұзындық – кез-келген ұзындықтағы деректерді белгіленген ұзындықтағы хеш-мәнге түрлендірудің орындалуы.

4. Тиімділік (өнімділік) – үлкен есептеу ресурстарын қажет етпеуі және жылдам орындалуы.

5. Лавиндік эффекті – кіріс хабарламаның шамалы өзгеруі шығыс хеш-мәннің айтарлықтай өзгеруіне әкелуі.

6. Ашықтық пен қолжетімділік – хеш функцияның криптографиялық қауіпсіздік қасиеттерін зерттеу үшін оның алгоритмі қолжетімді болуы.

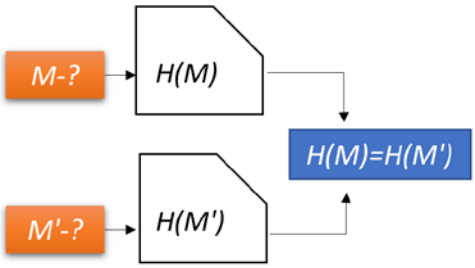
Қазіргі таңда кез-келген хештеу алгоритмдерге қойылатын негізгі классикалық талаптар бар. Бұл талаптар хеш функциясының қажетті қауіпсіздік қасиеттерін қамтамасыз етуін және әртүрлі қолданбаларда тиімді пайдаланылуын қамтамасыз етеді. Осы талаптардың кез-келгенін орындамау қауіпсіздік осалдығына немесе басқа мәселелерге әкелуі мүмкін. Бұл талаптарға мыналар жатады:

1. Коллизияға төзімділік;
2. Түпбейнеге (бірінші прототипке) төзімділік;
3. Екінші түпбейнеге (екінші прототипке) төзімділік.

Енді осы көрсетілген үш талапқа кеңінен тоқталып кетейік.

1) Коллизияға төзімділік (CR – Collision-Resistance):

Хеш функциядағы коллизия кез-келген әртүрлі екі хабарламалардың хеш-мәні бірдей болған кезде пайда болады. *H* хеш функциясы коллизияға төзімділік талабы орындалуы үшін *H(M)* белгілі болған жағдайда *H(M)=H(M')* орындалатын кез-келген бір-біріне тең емес *M* және *M'* хабарламаларын *(M≠M')* есептеп табу мүмкін болмауы керек (Сурет 1.3). Формальды түрде, *H* хеш функциясындағы коллизияны анықтаудағы А қарсыластың мүмкіндігі келесідей анықталады: $Adv_H^{CR}(A) = Pr\left[(M, M') \overset{\$}{\leftarrow} A : M \neq M' \wedge H(M) = H(M')\right]$, мұндағы Adv (қысқ., advantage) – артықшылық, мүмкіндік. Аталған талаптың графикалық нұсқасын келесі Сурет 1.3-тен көруге болады.



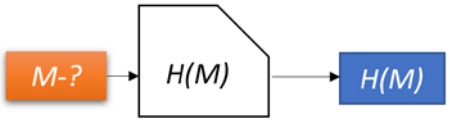
Сурет 1.3 – Коллизияға төзімділік (CR – Collision-Resistance) жағдайы

2) Түпбейнеге (Pre – Pre-image Resistance, бірінші прототипке) төзімділік:

Түпбейнеге төзімділік деп – *H* хеш функциясына кері функцияны есептеудің тиімді полиномиалдық алгоритмінің болмауы, яғни нақты уақытта (қайтымсыздық) белгілі хеш-мән арқылы *M* түпбейнесін есептеп қалпына келтіру мүмкін емес. Бұл қасиет хеш функция бір жақты функцияға пара-пар дегенді білдіреді. Яғни, әзірше белгісіз *M* хабарламаның белгілі *H(M)* хеш-мәні

арқылы кері қарай *M* түпбейнесіне көше алмауы немесе кез-келген басқа *M'≠M* түпбейнесін (мұндағы *H(M')=H(M)*) есептеп шығарудың мүмкін болмауы жағдайы. Әдетте, хеш функция коллизияға төзімділік танытса, ол түпбейне төзімділігіне кепілдік бермейді, бірақ хеш функция түпбейнеге жеткілікті түрде төзімділік танытса, онда ол колизияға төзімділікті де қамтамасыз ете алады, яғни 1-ші талап орындалса, 2-ші талап орындала бермейді, ал керісінше – орындалады [18, 19]. *H* хеш функциясындағы түпбейнені анықтаудағы А қарсыластың мүмкіндігі келесідей анықталады:

$$Adv_H^{Pre[m]}(A) = Pr\left[M \xleftarrow{\$} \{0,1\}^m; Y \leftarrow H(M); M' \xleftarrow{\$} A(Y) : H(M') = Y\right].$$

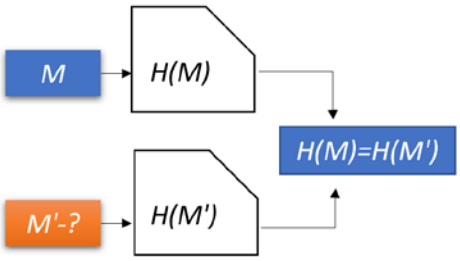


Сурет 1.4 – Түпбейнеге (Pre – Pre-image Resistance) төзімділік жағдайы

3) Екінші түпбейнеге (Sec – 2nd Pre-image Resistance, екінші прототипке) төзімділік:

Алдын-ала белгілі *M* түпбейнесінен *H* хеш функциясы арқылы алынған *h=H(M)* хеш-мәнін пайдаланып, *H(M) = H(M')* орындалатын қандай да бір *M'* түпбейнесін (мұндағы *M≠M'*) нақты уақытта есептеп шығару мүмкіндігі болмаса, *H* хеш функциясы екінші түпбейнеге төзімділік танытады деп айтады. Осы жағдайды А қарсыласқа қатысты былай сипаттауға болады:

$$Adv_H^{sec[m]}(A) = Pr\left[M \xleftarrow{\$} \{0,1\}^m; M' \xleftarrow{\$} A(M): M \neq M' \wedge H(M) = H(M')\right].$$



Сурет 1.5 – Екінші түпбейнеге (Sec – 2-nd Pre-image Resistance) төзімділік жағдайы

Жоғарғы сипатталған үш талап хештелген ақпараттың дәйектілігін қамтамасыз ету үшін маңызды роль атқарады. Тұтастай алғанда, хеш функциясын құру және енгізу оның мақсатты пайдалану жағдайына сәйкес келуін қамтамасыз ету үшін қауіпсіздік, тиімділік және қарапайымдылық арасындағы тепе-теңдікті қамтамасыз етуі керек. Сенімді жасалған хеш функциялардың қауіпсіздігі қасиеттерінің жоғарғы үш талаптың қатысымен төмендегідей бағаланады:

1) Хеш функцияның коллизияға төзімділік (CR – Collision-Resistance) жағдайы үшін коллизияны анықтауға арналған ең жақсы шабуыл туған күн шабуылынан жақсы болмауы керек. Яғни, хеш осы талапқа сай болу үшін коллизияны іздеудегі ең тиімді шабуыл «Туған күн парадоксі» шабуылы болуы керек. Бұл шабуылдан басқа тиімді шабуыл табылмауы және барлық шабуылдар түрлерінің есептеу күрделілігі $2^{n/2}$ -ден жоғары болуы керек, мұндағы n – хеш-мән ұзындығы (битпен).

2) Жақсы жасалған қауіпсіз хеш функция үшін екінші талап мына тұрғыда орындалуы керек: хеш функцияның түпбейнені қалпына келтіруге төзімді болуы үшін хеш функциясына қарсы ең жақсы шабуыл «дөрекі күшті шабуыл» болуы керек, яғни, n ұзындықтағы хеш-мән беретін хеш функциясы үшін операциясының күрделілігі 2^n-нен жоғары болуы керек.

3) Жақсы жасалған қауіпсіз хеш функция үшін үшінші шарт келесідегідей сипатталады: хеш функцияның екінші түпбейнені қалпына келтіруге төзімді болуы үшін хеш функциясына қарсы ең жақсы шабуыл «дөрекі күшті шабуыл» болуы керек.

Хеш функциялардың қауіпсіздігін қамтамасыз ететін басқа да қосымша қажетті қасиеттер бар [20]. Дегенмен, бұл қасиеттердің кейбіреулерін кейбір хеш қатысатын қолданбалар қажет етпеуі мүмкін.

- «Жақын коллизияларға» төзімділік (Near-collision resistance): бір-бірінен өзгеше екі хабарламаның хеш мәндері бір-бірінен айтарлықтай ерекшеленуі керек (тіпті хабарламалар бір-бірінен сәл ғана өзгеше болса да). Басқаша айтқанда, екі түрлі M және M' ($M \neq M'$) хабарламалары үшін олардың хеш-мәндері $H(M)$ және $H(M')$ бір-бірінен аз ғана биттерге ерекшеленетін болса,онда осындай жағдайда «жақын коллизиялар» туындайды. Жақсы құрылған хеш функция осындай жағдай туындатпауы тиіс.
- Псевдоколлизияларға төзімділік: псевдоколлизия (немесе еркін іске қосу коллизиясы) екі хабарлама арасындағы коллизияны тек инициалицациялық векторды (*IV*-ді) басқару арқылы анықтауға болатын кезде пайда болады. Бұл шабуыл іс жүзінде көп қолданбайды, өйткені хеш функцияларының көпшілігінде *IV* алдын-ала бекітілген. Сол себепті, қарсылас *IV*-ді өзгерте алмайды.
- Түпбейнеге (прототипке) ішінара төзімділік: белгілі хеш-мән арқылы бастапқы хабарламаның бір бөлігін қалпына алу, сондай-ақ хабарламаның бір бөлігі бұрыннан белгілі болса да, бүкіл хабарламаны қалпына келтіре алу есептеу жағынан қиын болуы тиіс.

1.2 Хештеу алгоритмдердің қауіпсіздік қасиеттерін бағалау критерийлері және хеш функцияға бағытталған шабуылдар

Хеш функциялар криптография және цифрлық қолтаңбалар қосымшаларында дұрыс және қауіпсіз жұмыс істейтініне көз жеткізу үшін оның сапасын бағалау – міндетті атқарылатын зерттеу жұмыстардың бірі болып саналады. Хеш функцияның сапасын бағалау критерийлері ретінде оған қойылатын жоғарыда айтылған талаптар мен қасиеттерін қарастыруға болады, атап айтқанда:

Коллизияға төзімділік: жақсы хеш функция бірдей хеш-мән беретін екі кірісті табуды қиындатуы керек.

Бірінші және екінші прототипке төзімділік: жақсы хеш функциясы көрсетілген хеш нәтижесін беретін кірістерді табуды қиындатуы керек.

Лавиндік эффекті: хабарламаның шамалы өзгеруі хеш-мәннің айтарлықтай өзгеруіне әкелуі керек.

Диффузия: хеш функция хабарламаны шығыс биттеріне біркелкі таратуы керек.

Тиімділік: жақсы хеш функциясы есептеуде жылдам болуы керек және аз операция орындалуы қажет етеді.

Әмбебаптық: жақсы хеш функциясы кірістер мен кірістердің кең ауқымы үшін жақсы жұмыс істеуі керек.

Анықталғандық: әрқашан бірдей кіріс үшін хеш функциясының шығысы бірдей болуы керек.

Хеш-мәннің ұзындығы: шығыс ұзындығы қолданбаға сәйкес келуі керек және коллизия шабуылдарының алдын-алу үшін жеткілікті үлкен болуы керек.

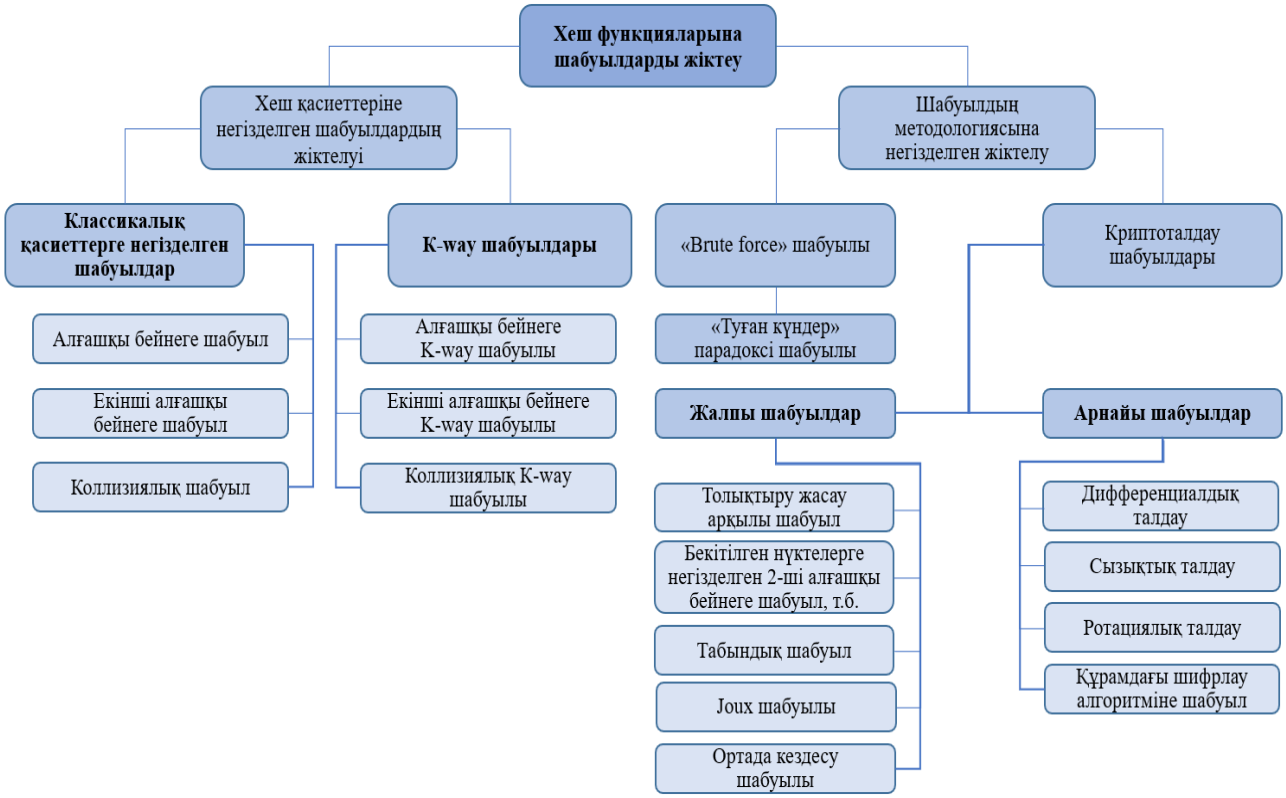
Криптографиялық қауіпсіздік қасиеттері: криптографиялық қауіпсіздікті қажет ететін қосымшалар үшін хеш функция берік математикалық негізге ие болуы немесе дәлелденген криптографиялық примитивтерден жасалуы керек және мұқият талдау мен тестілеу арқылы қауіпсіз түрде дәлелденуі керек.

Ашықтық: хеш функциясында шабуылдаушы қолдана алатын жасырын функциялар немесе әлсіз жерлер болмауы керек.

Криптографиялық және статистикалық талдауларға негізделген (дифференциалды, сызықтық, алгебралық және т.б.) шабуылдарға төзімділік: жақсы хеш функциясы аталған шабуылдарға осал болмауы керек, алгоритм сұлбасы негізінде шабуылдаушы кірістер (хабарламалар, кілттер, инициализациялық векторлар) мен шығыстар (хеш-мәндер) арасындағы айырмашылықтарды талдай отырып, кіріс туралы ақпаратты ала алмауы керек.

Бұл критерийлер хеш мүмкіндіктерінің қауіпсіз және нақты қолданбаларда пайдалануға жарамды екеніне көз жеткізуге көмектеседі.

Хеш функциясына шабуыл оның сапасы мен қауіпсіздік қасиеттерін бағалап қана қоймай, оны одан әрі жетілдірудің бірден-бір жолы ретінде қарастыруға болады. Хеш функциясына шабуыл – бұл хеш функцияларының қауіпсіздік қасиеттері мен талаптарының бірінің немесе бірнешеуінің дұрыс орындалмауын дәлелдеуге бағытталған әс-әрекеттер. Мысалы, түпбейнеге төзімділікті бұзу (сындыру) дегеніміз, шабуылдаушы түпбейнеге қалпына келтіріле алмау қасиетін бұзуы, яғни ол белгілі хеш-мән арқылы хабарламаны жасай алады. Шабуылдар көбінесе хеш функциясының құрылымына немесе қысу функциясының алгоритміне бағытталады. Жалпы, хеш функцияларда оған жасалатын шабуылдарды төменгі көрсетілген сұлба бойынша сипаттауға болады (Сурет 1.6). Сұлба бойынша хеш функцияларына шабуылдарды екі үлкен санатқа бөлуге болады [21].



Сурет 1.6 – Хеш функцияларға жасалатын шабуыл түрлерінің жіктелуі

«Brute force» шабуылдары. «Brute force» шабуылдары алгоритмдердің құрылымдарына және кез-келген басқа сипаттамаларға қарамастан барлық хеш функцияларына жүргізуге болады. Олар шифрлау алгоритмдеріндегі құпия кілтті алу үшін толық іздеу немесе кілтті қалпына келтіру әдістеріне ұқсас. Кез-келген хеш функциясының қауіпсіздігі оның хеш-мән ұзындығына байланысты. Төменгі Кесте 1.1-де осы қауіпсіз хеш функциялар үшін ең тиімді шабуыл түрлері, сондай-ақ олардың есептеу күрделілігі көрсетілген.

Кесте 1.1 – Жақсы жасалған қауіпсіз хеш функция үшін тиімді шабуылдар

	Талаптар		
	Коллизияға төзімділік	Түпбейнеге төзімділік	Екінші түпбейнеге төзімділік
Тиімді шабуыл	«Туған күндер парадоксы» әдісі	Толық теру әдісі	Толық теру әдісі
Есептеу күрделілігі	$2^{n/2}$	2^n	2^n

Криптоталдаулық шабуылдар.

Бұл шабуыл түрлері хеш функциясының негізгі сұлбасындағы, сондай-ақ, оның құрамындағы қысу функциясының алгоритміне зерттеуге бағытталады. Хеш-мәннің ұзындығының бекітілген болуы және оның ұзындығы хабарлама ұзындығы кіші болуы себепті, хеш функция міндетті түрде коллизия тудырады. Шабуылдаушының негізгі мақсаты коллизия немесе түпбейнені таба алу

мүмкіндігі есептеу тұрғыдан мүмкін болатын жағдайларды анықтап, мүмкіндігінше түпбейнені қалпына келтіре алу немесе коллизияны таба алу.

Әдетте, егер хеш функцияның кем дегенде бір қасиетін бұзу үшін жасалған ең тиімді криптоталдау шабуылының есептеу күрделілігі «толық теру» әдісінің есептеу күрделілігімен салыстырғанда төмен болса, онда қарастырылған хеш функция бұзылды немесе сынды деп ұйғарым жасалады. Бұл тұжырымда бұл шабуылдың есептеу мүмкіндігі үшін маңызды емес. Мысалы, хеш-мән ұзындығы 256 бит болатын хеш функцияның коллизиясын табу үшін есептеу күрделілігі 2^{90}, осындай мөлшердегі есептеулер жүргізілетін болсын. Бүгінгі күні тәжірибедегі есептеу техникаларының және технологиялардың осындай дәрежедегі есептеуді жүргізу қауқары болмаса да, бұл коэффициент «туған күн» парадоксі шабуылында талап етілетін 2^{128} көрсеткішінен аз болғандықтан, теория жүзінде хеш функциясы бұзылған (сынған, нашар жасалған) деп айтылады.

Жалпы шабуылдар. Хеш функция құрылысында жұмыс істейтін шабуылдар жалпы шабуылдар деп аталады. Мысалы, Меркл-Дамгард конструкциясы арқылы жобаланған барлық хеш-функцияларда осы конструкцияға қатысты жасалатын шабуылдар жалпы шабуылдар болып табылады. Оларға мыналарды жатқызуға болады:

1. Толықтыру жасау арқылы шабуыл.
2. Мультиколлизиялар шабуылы.
3. Екінші түпбейне бойынша шабуыл (2nd Pre-image Attack).
4. Табындық шабуыл (Herding attack).

1. Толықтыру жасау арқылы шабуыл хабарламаның соңына жаңа биттерді тіркеу негізінде жүргізіледі [22]. Бұл шабуылға осал хеш функциялардың ішіне MD5 және SHA-1 хештеу алгоритмдері жатады. Бұл шабуылдың 4 нұсқасы бар. Олар:

1) Коллизиялық шабуыл (Collision attack): $L=|M|$ ұзындықтағы M хабарламасы берілсін. Осы M хабарламадан Меркл-Дамгард конструкциясы негізіндегі H хеш-функциясы арқылы $H(M)$ хеш-мәні алынсын. Бұдан әрі мынандай түрде коллизия тауып алуға болады: $H(M||pad||x) = H(H(M)||x)$, мұндағы pad - хештеу алдында M хабарламасына тіркелген толтыру, $|pad|=L \bmod m$, мұндағы m - M-дегі бір блоктың ұзындығы.

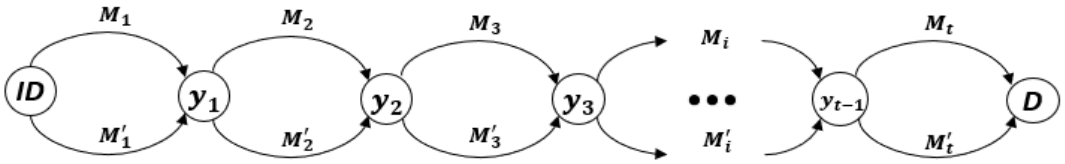
2) Екінші коллизия арқылы шабуыл (Second collision attack). Бұл шабуылда келесі коллизияны бірдей ұзындықтағы өзара соқтығысатын хабарламаларды кеңейту арқылы оңай анықтауға болады. Яғни, егер $H(M)=H(N)$, ал $M\neq N$ және $|M|=|N|$ болса, екінші коллизияны M және N-ді ($suffix$) S – еркін жолмен кеңейту арқылы алуға болады: $H(M||S) = H(N||S)$.

3) Байланысқан хабарламалар негізіндегі шабуыл. Осы шабуыл арқылы Меркл-Дамгард конструкциясы көмегімен ұзындығы L және хеш-мәні $H(M)$ белгілі, ал өзі белгісіз M хабарламасы үшін онымен байланысқан M' хабарламасын есептеуге болады, және $H(M)$, яғни ол хабарлама M және $suffix\ L||x$ кеңейтілімінен тұратын хабарламаның хеш-мәні $H(M||L||x)$ болады.

4) Жалған MAC негізінде шабуыл (MAC Forgery Attack). Бұл шабуыл барысында Меркл-Дамгард негізінде MAC кілтті хеш-функциясында

қолданылатын *K* құпия кілтін білмей-ақ, нақты хабарламаны есептеп шығуға болады.

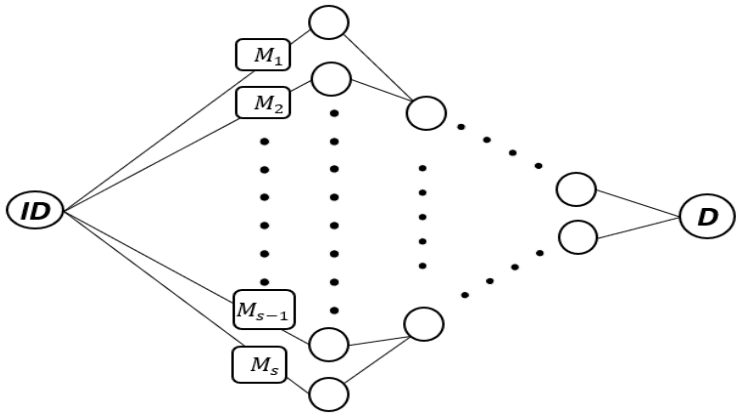
2. Мультиколлизиялар шабуылы (The Multi-collision Attack). Жукс (Joux) Меркла-Дамгард негізіндегі хеш функциядағы мультиколлизияларды (бірдей мәні бар екіден көп хештелген хабарламалар) анықтау бір коллизияны анықтаудан гөрі қиын емес екенін көрсетті. Осы шабуыл *IV* – инициализациялық векторы арқылы алғашқы коллизия құра алатын екі хабарламаны (туған күндер шабуылы арқылы) ала алады [23]. Мультиколлизиялар шабуылы төменгі Сурет 1.7-де көрсетілген.



Сурет 1.7 – Мультиколлизиялар шабуыл құрылымы

3. Екінші түпбейне бойынша шабуыл кеңейтілетін хабарламалар жиынтығының бар деп тұжырым жасайды, хабарламалар әр түрлі ұзындықта болуы да мүмкін, алайда бекітілген *IV* үшін бірдей аралық хеш-мән шығарып отырады [24]. Егер хеш алгоритмде бекітілген нүктелер табылатын болса, бұл хабарламаларды оңай табуға болады. Әдетте, бекітілген нүкте ретінде мына жағдайды қарастырады: $h_i=H(h_{i-1},M_i)=h_{i-1}$.

4. Табындық шабуыл (Herding attack). Бұл жоғарыда қарастырылған мультиколлизиялар шабуылы және екінші түпбейне шабуылдарымен тығыз байланысты [25]. Бұл шабуыл хеш-мәнді табу үшін ромб тәрізді құрылымды және Меркл-Дамгард конструкциясын пайдаланады (Сурет 1.8).



Сурет 1.8 – Табындық шабуылдағы ромб құрылымы

Алдын ала оған хеш-мән белгілі болуы керек. Шабуыл екі кезеңде өтеді. Бірінші кезеңде шабуылдаушы ромб тәрізді құрылымды құрады. Ромбтың төбелері – сәйкес аралық хеш-мәндер, ал қабырғалары – хабарламалар болып саналады. Егер екі хабарлама бір төбеде кездессе, олар осы аралық хеш-мән арқылы коллизия тудыратын болады. Бастапқыда шабуылдаушы кездейсоқ

жолмен өте көп мөлшерде $\{M_1, M_2, \ldots, M_s\}$ хабарламалар жиынын жасайды, бұл хабарламаларды хештейді және коллизияларды табуға тырысады. Осы процесті *D* ақырғы төбеге жеткенге дейін жалғастырады. Ромд құрылымы жасалғаннан соң, бастапқы хабарламалардан *D*-ге дейінгі кез-келген жол өзінше *D*-ге дейін тізбектеліп хештеледі. 2-кезеңде шабуылдаушы *D*-де хештеу үшін алынған *P* префикстерін (алдыңғы хеш-мәнді) келесідей жинайды: шабуылдаушы алдымен *P* префиксі мен 1-ші блоктың қандайда бір *S* суффиксі (келесі хеш-мән) біріктірілгенде, бастапқы хабарламалардың $H(M_i)$ хеш-мәнімен салыстырғанда коллизия тудыратындай болатын *S* суффиксті іздейді. Сәйкестік табылған жағдайда *P, S* және сәйкес келетін $H(M_i)$-ден *D*-ге дейінгі хабарламалар тізбегі біріктіріледі және бұл жолдың барлығы хештеледі.

2 БЛОКТЫҚ ШИФРҒА НЕГІЗДЕЛГЕН ХЕШТЕУ АЛГОРИТМІН ҚҰРУ

Хеш алгоритмдеріндегі тиімділік мәселесі оның сенімділігіне, яғни коллизияларға төзімділігіне, жылдамдығына, сондай-ақ есептеуге қажетті ішкі жедел жады регистрлеріне тәуелді екендігіне байланысты. Осы уақытқа дейін ұсынылған хеш функциялардың көпшілігі бағдарламалық жасақтамаға бағытталған және күнделікті қолданыста коллизияны іздеуге арналған кез-келген әдіс-тәсілдер қолжетімсіз немесе тиімсіз болу үшін хеш ұзындығы кемінде 256 бит болу керек деп ұйғарылған. Қауіпсіз және тиімді хеш функцияны құру барысында көптеген жылдар бойы жақсы тексерілген, сонымен қатар бағдарламалық және аппараттық құралдарға енгізу үшін тиімді криптографиялық компоненттерді пайдалану қажет [26, 27]. MD5 және SHA-1 хеш функциясы конструкциясы жағынан ерекше және арнайы хеш функциялар деп аталады. Кейбір хеш функциялар бастапқыда хештеу үшін пайдалануға арналмаған, бірақ бейімделуі мүмкіндігі бар криптографиялық немесе математикалық құрамдастарға негізделген. Қысу функцияларын қалай жасау керектігін нақты талқылаған кезде біз «жақсы» қысу функциясы бар деп алып және сәйкесінше конструкция жасауға болады. Осылайша, хеш функцияларын жасаудағы кейбір ұсыныстар қандайда бір тиімді X қасиетіне ие қысу функциясын құруға, содан кейін барып осы қысу функциясын пайдалана отырып, X қасиетін дәлелді түрде сақтайтын қолайлы конструкцияны әзірлеуге қатысты болды, мысалы, егер қысу функциясы коллизияға төзімді болса, Меркл-Дамгард конструкциясының модификациясы негізінде жасалған хештеу алгоритмі де коллизияға төзімді болады.

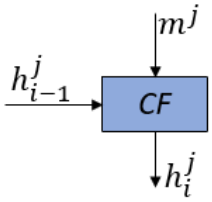
Блоктық шифрлар негізіндегі хеш функцияларды жасау – ең танымал және қалыптасқан бағыт. Бұл бағытта қысу функциясы хабарлама блогы мен кілтті қабылдайтын екі кірісі бар блоктық шифр болып қарастырылады. Пренель, Говертс және Вандевалль (Preneel, Govaerts and Vandewalle) $E : \{0, 1\}^n \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ блоктық шифрды пайдаланып, одан хеш функцияларды құрудың 64 ықтимал әдісін зерттеді [28]. Бұл 64 әдісті кейде осы құрылымның қауіпсізік қасиеттерін зерттеумен шұғылданған авторлардың аттарымен PGV деп атайды. Алғашқыда 64 PGV әдістің ішінде коллизияға төзімділері деп 12-сі есептелген. Кейіннен Блэк және т.б дәлелді тәсілді қолдана отырып, тағы 8-ін қолданылатын қысу функциясының коллизияға төзімді болмаса да, бұлардың хештеу схемасында дұрыс таңдап ала алынса, бұлар да коллизияға төзімді екенін көрсетті [29]. Жоғарғы 64 PGV әдістің 20-сын Дэвис пен Мейердің атымен байланыстырады және олардың жалпы нұсқасы мына түрде: $y_i = f(h_{i-1}, M_i) \oplus y_{i-1}$, мұндағы y_{i-1} және M_i қысу функциясының кірістері, ал y_i – шығысы [30]. Бұл PGV құрылымдарының идеалды шифрлық модельдегі коллизиялары мен түпбейнеге төзімділігінің қосымша талдаулары мен дәлелдерін [31] және [32]-ден танысуға болады.

PGV функциялары дәлелді қауіпсіз болғанымен, олар тиімсіз, себебі кілт (қысу функциясының хабарлама блогының кірісі) қысу функциясы шақырылған сайын өзгеріп, бұл блоктық шифрлар үшін аса тиімді емес, өйткені жұмыс істеу үшін үлкен көлемдегі есептеулер қажет. Сонымен, ұсынылған тағы бір тәсіл –

тұрақты бекітілген кілті бар блоктық шифр негізіндегі қысу функцияларын қолдану [33-37]. Бұл тәсілде блоктық шифр үшін қысу функциясымен шақырылған кезде қандайда бір тұрақты кілттер жинағы пайдаланады. Дегенмен, Black және тағы басқалары тұрақты кілтті мұндай конструкция тиімді болғанымен, коллизияға төзімді бола алмайтынын дәлелдеді [33, б. 327].

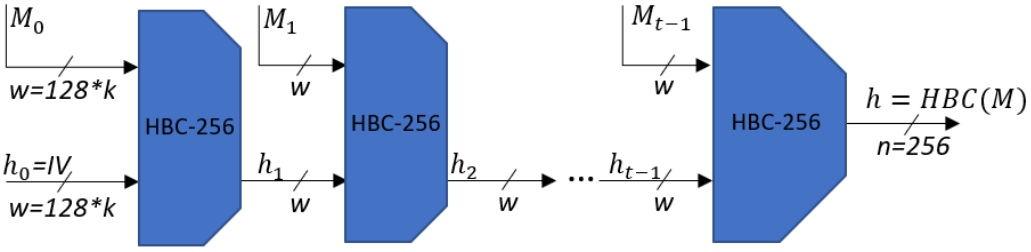
2.1 Блоктық шифрға негізделген хештеу алгоритмінің құрылымдық бөліктері

НВС-256 (Hash based on Block Cipher) мәліметтерді хештеу алгоритмі CF блоктық шифрлау алгоритмі негізінде жасалды [38]. Хештеу механизмдегі қысу функциясының жұмысын CF алгоритмі атқаратын болады. Бүгінгі күні қысу функциясын қолданатын тәсіл ең танымал және жақсы орныққан болып саналады. Қарастырылатын қысу функциясы кірісінде екі мән қабылдайды – 128-биттік хабарлама бөлікшесі m^j арқылы жасалатын шифрлаудың 128-биттік раундтық кілттері және алдыңғы $(i-1)$-ші итерациядағы 128-биттік аралық хеш-мән h_{i-1}^j, ал шығыс мәні ретінде 128-биттік аралық хеш-мән h_i^j қабылданған, яғни $CF: \{0,1\}^{128} \times \{0,1\}^{128} \to \{0,1\}^{128}$, мұндағы $\{0,1\} \in GF(2)$ [39, 40]. CF шифрлау функциясы арқылы жүзеге асырылатын қысу функциясының кескінін Сурет 2.1-ден көруге болады.



Сурет 2.1 – НВС-256 алгоритміндегі қысу функциясы

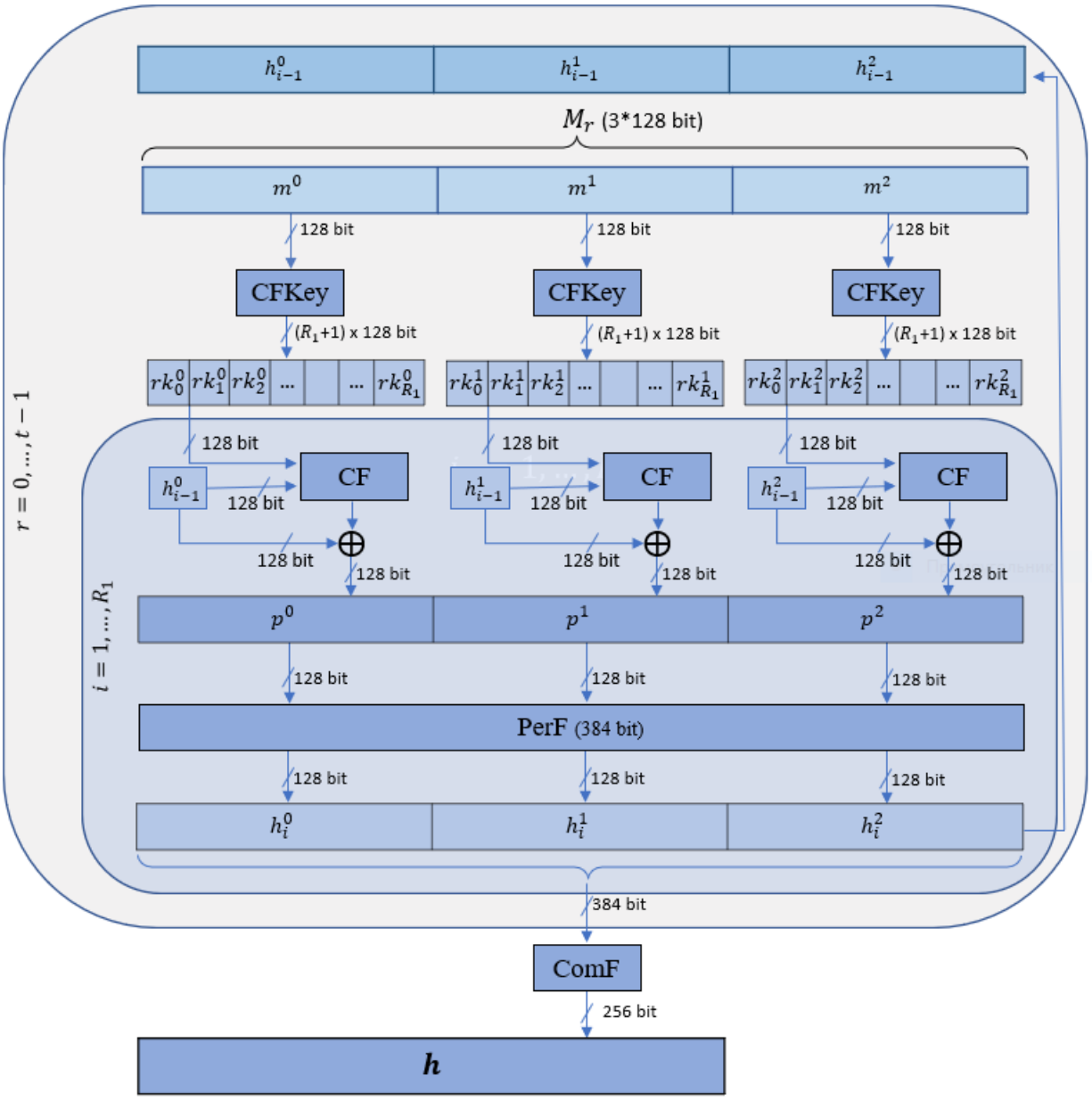
Хештеу алгоритмін жасау барысында коллизияларды табуға арналған «Толықтыру жасау арқылы шабуылға» (Length extension attack) қарсы тұра алатын Меркл-Дамгард конструкциясының кеңінен таралған Wide-pipe модификациясын қолданатын боламыз [41]. Бұл жағдайда ақырғы *n*-битті хеш-мәнді алу үшін аралық хеш-мәндердің ұзындығы және хабарламалар блогінің (бөлігінің) ұзындығы w битке тең болу керек, сондай-ақ $n < w$ болуы шарт (Сурет 2.2).



Сурет 2.2 – НВС-256 алгоритміндегі wide-pipe конструкциясы

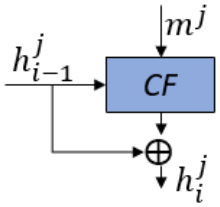
Wide-pipe модификациясының талаптары орындалуы үшін хештеудің бір циклінде бір уақытта CF қысу функциясын әртүрлі $m^j \in M_r$ үшін k рет орындайтын боламыз, мұндағы $r = 0, 1, \dots, t-1$; $j = 0, \dots, k-1$. Сондықтан, аралық хеш-мәндер ұзындығы *128*k* битке тең.

$M(M_0, M_1, \dots, M_{t-1})$ хабарламаны хештеудің жалпы сұлбасы Сурет 2.3-те көрсетілген, мұнда $M_r(m^0, m^1, \dots, m^{k-1})$, $r = 0,1, \dots, t-1$, суретте және алгоритмде $k=3$ болып алынған. Ұсынылған алгоритмде *CF* қысу функциясындағы кілт ретінде M хабарламаның M_r бөлігі, ал ашық мәтін ретінде h_{i-1}^j – алдыңғы итерация нәтижесіндегі хеш-мән алынады. M_r бөлігінің m^j бөлікшелерін пайдаланып, қажетті мөлшерде раундтық кілттер жасалып отырады.



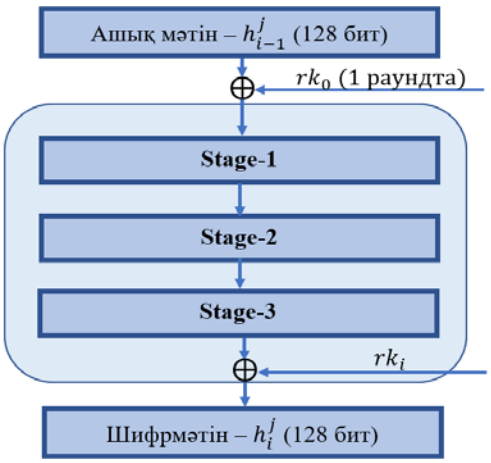
Сурет 2.3 –НВС-256 хештеу алгоритмінің сұлбасы

НВС-256 хештеу алгоритмінде коллизияға төзімділікті арттыру үшін Дэвис-Майер схемасы қолданылады, мұнда *CF* функциясының шығысы алдыңғы хештеу итерациясының нәтижесі h_{i-1}^{j}-пен XOR операциясы арқылы қосылады. p^{j} мәні Дэвис-Майер схемасы негізінде хеш функциясының *i*-ші итерациясының нәтижесі болып табылады. Бұл Дэвис-Майер схемасы блоктық шифрларға негізделген хештеу алгоритмдерінде жиі қолданылады және бірбағытты қысу функциясы ретінде әрекет етеді. Дэвис-Мейер схемасының қауіпсіздігін алғаш рет Винтерниц дәлелдеген. Сурет 2.4-те НВС-256 хештеу алгоритміндегі Дэвис-Мейер схемасының сұлбасы кескінделген.



Сурет 2.4 – НВС-256 алгоритміндегі Дэвис-Мейер схемасы

CF блоктық шифрлау алгоритмі. CF шифрлау алгоритмі блок пен кілт ұзындығы 128 бит болатын симметриялық блоктық шифрлау алгоритмі класына жатады. Алгоритмде сызықтық (модульдік 2 бойынша қосу, циклдік солға жылжу) және сызықты емес (төрт S-блокты ауыстыру) түрлендірулер қолданылады. Шифрдың құрылымы төрт раундты (R_1=4) ауыстыру-алмастыру желісінің (SP-желі) нұсқасында құрылған [42, 43]. Шифрлаудың бір раунды Stage-1, Stage-2 және Stage-3 түрлендірулерінен тұрады және оның құрылымы Сурет 2.5-те көрсетілген. Шифрлауда алдымен раундтық кілтпен ағарту жүргізіледі. 1-ші раундтық кілт ретінде мастер-кілт пайдаланады. Әр раундта тізбектей орындалатын Stage-1, Stage-2 және Stage-3 түрлендіруінен кейін алынған шифрмәтінді раундтық кілтпен модуль екі бойынша биттік қосу операциялары орындалып отырады. Раундтық кілттерді жасау CFKey алгоритмі арқылы іске асырылады.



Сурет 2.5 – CF алгоритмінің жалпы сызбасы

Ашық мәтін h_0^j мәндерін массив $A(a_0, a_1, a_2, \dots a_{15})$ ретінде қабылдап, оны 4x4 өлшемді квадраттық A матрицасы түрінде жазуға болады:

$$A = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 \\ a_4 & a_5 & a_6 & a_7 \\ a_8 & a_9 & a_{10} & a_{11} \\ a_{12} & a_{13} & a_{14} & a_{15} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix}.$$

Матрицаның әр элементі бір байт ретінде қарастырылады.

Stage-1 түрлендіруі. Бұл екі қадамнан тұратын түрлендіру берілген A матрицасындағыдай өлшемдегі жаңа матрицаны алу үшін қолданылады. Stage-1 түрлендіруінің бір ерекшелігі болып шифрлау арысында сызықты және сызықты емес криптографиялық түйіндер қатарласып жұмыс істеуі саналады. Матицаның әр элементтерін есептеу барысында бұл екі түйіндер жұмыс кезінде төмендегідей қадамдармен анықталып, сол элемент үшін екеуі бірінен кейін бірі кезектесіп орындалып отырады.

1-қадам. Бұл қадам – сызықты түйін қадамы. A матрицасы арқылы есептелінетін c_{ij} аралық мәндері матрица құрылымы бойынша солдан оңға, жоғарыдан төмен бағытта есептелініп алынады, мұндағы, $i, j = 0,1,2,3$. c_{ij} аралық мәндері A матрицасының i-ші жолының төрт элементі мен j-ші бағанының i-ші жолындағы элементінен басқа үш элементімен 2-модуль қосу операциясы арқылы есептеледі. Мысалы ретінде c_{01} элементін есептеуде қатысатын матрица элементтері Сурет 2.6 ерекше бояумен көрсетілген.

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix};$$

Сурет 2.6 – c_{01} есептеуге қатысатын A матрицасының элементтері

2 қадам. Бұл қадамда есептелетін c_{ij} аралық мәні S-блок ауыстырулар (SBOX процедурасы) арқылы өзгертіліп, A матрицасының осы орындағы жаңа мәні болып қабылданады.

1-ші және 2-ші қадамдардан тұратын Stage-1 түрлендіруін былай жазуға болады:

$$\left.\begin{array}{c} c_{ij} = \oplus\sum_{k=0}^{3} a_{ik} \oplus \left(\oplus\sum_{\substack{k=0 \\ k\neq i}}^{3} a_{kj} \right); \\ a_{ij} = SBOX(c_{ij}); \end{array}\right\} \; i = 0,1,2,3; \;\; j = 0,1,2,3. \qquad (2.1)$$

мұндағы c_{ij} – A матрицасының аралық мәні,

SBOX – *S*-блок ауыстыру кестесін орындайтын процедура,

$\oplus\sum$ – модуль 2 бойынша қосу (биттік қосу) операциясы.

SBOX процедурасы. Сызықты емес биективті S-блок ауыстыру түрлендіруі *SBOX* процедурасы арқылы анықталады. S_0, S_1, S_2, S_3 ауыстыру түрлендіруі берілген, мұнда $S_i: \mathbb{Z}_{2^4} \to \mathbb{Z}_{2^4}$, $i = 0, \ldots, 3$. Алгоритм үшін Кесте 2.1-де көрсетілген төрт «алтын» S-блок ауыстырулары таңдап алынды [44].

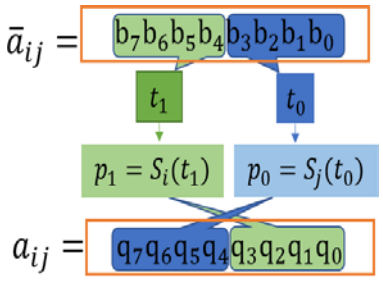
Кесте 2.1 – Төрт «алтын» S-блок ауыстырулары

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
$S_0(x)$	0	F	B	8	C	9	6	3	D	1	2	4	A	7	5	E	Serpent, S_3
$S_1(x)$	2	E	F	5	C	1	9	A	B	4	6	8	0	7	3	D	HB-1, S_2
$S_2(x)$	7	C	E	9	2	1	5	F	B	6	D	0	4	8	A	3	HB-2, S_0
$S_3(x)$	4	A	1	6	8	F	7	C	3	0	E	D	5	9	B	2	HB-2, S_1

Кесте 2.1-де Serpent – жеңілсалмақты Serpent шифры, HB-1 – жеңілсалмақты Hummingbird-1 шифры, HB-2 – жеңілсалмақты Hummingbird-2 шифры.

SBOX процедурасының жұмыс істеу тәртібі Сурет 3.7-те көрсетілген. Кіріске A матрицасының $\bar{a}_{ij}$ элементі беріледі, $\bar{a}_{ij} = (b_7 b_6 b_5 b_4 b_3 b_2 b_1 b_0)_2 - \bar{a}_{ij}$ байтының екілік жазбасы. S-блок ауыстырулар жартыбайттық деңгейде жүргізіледі, оларда төмендегідей белгілеулер енгізілсін: $t_1 = b_7 b_6 b_5 b_4$ – сол жақ жарты байт, $t_0 = b_3 b_2 b_1 b_0$ – оң жақ жартыбайт (жазбалар екілік санау жүйесінде). Бұдан әрі, Кесте 2.2 көмегімен $p_1 = S_i(t_1)$ және $p_0 = S_j(t_0)$ анықталатын болады. Матрица элементтерінің индекстері i және j S-блоктың реттік нөмірлеріне сәйкес келеді. Алынған жартыбайттар өзара бір-бірімен орын алмасып, бір байтқа біріктіріледі. Осы тәртіппен алынған байт шығыс болып саналады: $a_{ij} = (q_7 q_6 q_5 q_4 q_3 q_2 q_1 q_0)_2$, яғни, $a_{ij} = SBOX(\bar{a}_{ij})$.

SBOX процедурасы жұмысына мысал: Кіріс байт ретінде $\bar{a}_{13} = 39_{10} = 00100111_2 = 27_{16}$ берілген болсын. Кесте 2.2 бойынша, $p_1 = S_1(2_{16}) = F_{16}$, $p_0 = S_3(7_{16}) = C_{16}$. Екі жарты байттардың орнын ауыстырып, біріктіргенде: $a_{13} = p_0 \parallel p_1 = CF_{16} = 11001111_2 = 207_{10}$. Нәтижесінде $a_{13} = SBOX(39) = 207$.



Сурет 2.7 –SBOX жұмысының тәртібі

Stage-2 түрлендіруі. Бұл түрлендіру екі операциядан тұрады: циклдық жылжыту және XOR. Stage-1-ден алынған A матрицаның элементтері бір өлшемді массив ретінде жазылады: (a_{00},a_{01},a_{02},a_{03},a_{10},a_{11},a_{12},a_{13},a_{20},a_{21},a_{22},a_{23},a_{30}, a_{31},a_{32},a_{33}). Бұл байттардың екілік жүйедегі жазбалары бір-бірімен конкатенация

операциясы арқылы біріктіріліп жазылады: $W=a_{00}||a_{01}||a_{02}||a_{03}||a_{10}||a_{11}||a_{12}||a_{13}||a_{20}||a_{21}||a_{22}||a_{23}||a_{30}||a_{31}||a_{32}||a_{33}$, $|W|=128$ бит. Осы тізбекке солға қарай 1 бит циклдық жылжыту орындалып $V=W<<<1$, алынған тізбек он алты байтты жаңа нәтиже аламыз: $V=b_{00}||b_{01}||b_{02}||b_{03}||b_{10}||b_{11}||b_{12}||b_{13}||b_{20}||b_{21}||b_{22}||b_{23}||b_{30}||b_{31}||b_{32}||b_{33}$. Келесі кезеңде алынған V мен W массивтері xor операциясымен қосылады: $A = W \oplus V$. Ақырғы алынған нәтиже A матрицасының жаңа элементтері болып солдан оңға, жоғарыдан төмен ретпен жазылады.

Stage-3 түрлендіруі. Stage-3 түрлендіруі құрылымы жағынан жоғарыда көрсетілген Stage-1 түрлендіруіне өте ұқсас. Ол түрлендірудегідей, Stage-3 түрлендіруіндегі A матрицасы мәндері біріншісі – сызықты, екіншісі – сызықты емес криптографиялық түйінге жататын екі қадамнан тұратын операциялар арқылы есептелінеді, нәтижесінде осындай өлшемдегі жаңа матрица алынады. Жұмыс істеу тәртібіндегі өзгешелік – жаңа матрица элементтерін есептеудегі бағытта, яғни матрица элементтерін есептеу төменнен жоғары, оңнан сол бағытта жүргізіледі. Осы жердегі S-блок ауыстырулары ретінде Кесте-1-де көрсетілген «Алтын S-блоктар» қолданылады. S-блоктар жұмыс реті *SBOX* процедурасымен жүзеге асырылады. Әр элементті есептеу барысында қадам-1 мен қадам-2 тізбектеліп жүргізіледі. Есептеу матрицаның a_{33} элементінен бастап, a_{00} элементіне дейін өтеді.

Қадам-1 мен Қадам-2-ден тұратын есептеуді алгебралық түрде мына формулалармен жүргіземіз:

$$\left.\begin{array}{c} c_{ij} = \oplus\sum_{k=0}^{3} a_{ik} \oplus \left(\oplus\sum_{\substack{k=0 \\ k\neq i}}^{3} a_{kj}\right); \\ a_{ij} = SBOX(c_{ij}); \end{array}\right\} \; i = 3,2,1,0; \;\; j = 3,2,1,0. \qquad (2.2)$$

Аралық мән c_{ij} есептеу кезінде сәйкесінше матрицаның i-ші жолындағы төрт элемент пен j-ші бағандағы үш элементтердің (i-жол мен j-баған қиылысындағы элементтен басқа) модуль екі бойынша биттік қосындысы бойынша жүреді. Сурет-2.8-де мысал ретінде c_{32} аралық мәнін есептеуге қатысатын матрица элементтері көрсетілген. Stage-3 түрлендіруі нәтижесінде шифрланған 16 байтты блок аламыз.

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix};$$

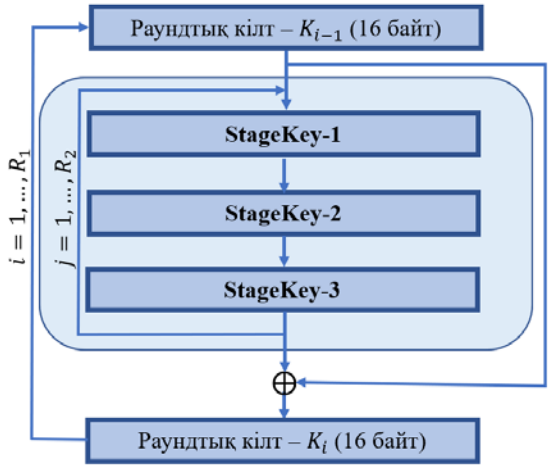
Сурет 2.8 – c_{32} есептеуге қатысатын A матрицасының элементтері

CFKey раундтық кілттерді жасау алгоритмі. 16 байт ұзындықтағы $K(k_0, k_1, k_2, \ldots k_{15})$ құпия кілттен осы ұзындықтағы раундтық кілттерді дайындау алгоритмі *CFKey* құрылған. K құпия кілті K_0 раундтық кілт ретінде саналады.

Раундтық кілттердің жалпы саны *CF* шифрлау алгоритміндегі R_1 раунд санына байланысты. $K_0(k_0, k_1, k_2, \dots k_{15})$ раундық кілтті A квадрат матрицасы түрінде төмендегідей жазуға болады:

$$A = \begin{pmatrix} k_0 & k_1 & k_2 & k_3 \\ k_4 & k_5 & k_6 & k_7 \\ k_8 & k_9 & k_{10} & k_{11} \\ k_{12} & k_{13} & k_{14} & k_{15} \end{pmatrix} = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix};$$

CFKey раундтық кілттерді жасау алгоритмі StageKey-1, StageKey-2 және StageKey-3 түрлендірулерінен тұрады. *CFKey* кілт жасау алгоритмі жұмысы Сурет-2.9-де кескінделген. *CFKey* алгоритмі жұмыс істеу тәртібі бойынша *CF* шифрлау алгоритміне өте ұқсас келеді: яғни сәйкес жұмыс компоненттері Stage-1 түрлендіруі StageKey-1 түрендіруімен, Stage-2 түрлендіруі StageKey-2 және Stage-3 түрлендіруі StageKey-3. Айырмашылық тек StageKey-2-де ғана, ол Stage-2-дағыдай екі операциядан емес, тек бір ғана операциядан тұрады: солға қарай циклдық 1 бит жылжыту операциясы. *CFKey* алгоритмі сурет-2.9-де көрсетілгендегідей, келесі K_i раундтық кілтті алу үшін *CFKey* $R_2 = 8$ рет қайталанады, одан соң алынған нәтиже K_{i-1} раундтық кілтімен модуль 2 бойынша қосылады, мұнда $i = 1, \dots, R_1$ дейін.



Сурет 2.9 – *CFKey* раундтық кілттер жасау алгоритмінің жалпы сызбасы

2.2 НВС-256 хештеу алгоритмінің жұмыс істеу тәртібі

Хабарламаға толықтыру жүргізу. *M* хабарламасына толықтыру мына түрде жүргізіледі. Егер *M* хабарламасының көлемі *128*k* битке еселік болса, онда хабарлама соңына тағы бір *128*k* биттен тұратын жаңа бөлік жалғанады. Жалғанатын жаңа екілік жазбадағы бөлік бірінші және соңғы биттері «1», қалған биттері «0»-дерден тұратын биттік тізбектен құралған. Ал, *M* хабарламасы көлемі *128*k* битке еселік болмаған жағдайда, яғни $l = length(M) \neq 0 \; mod \; (128 * k)$ болғанда, онда соңғы бөлікке *128*k*-ке еселік болатындай етіп толықтыру жүргізіледі. Ол үшін l ұзындықтағы *M* хабарламасының соңына $s+2$ ұзындықтағы биттік тізбек жалғанады. Бұл тізбекте де бірінші және соңғы биттері «1» мәнін қабылдайды, ал s ұзындықтағы ортаңғы биттері «0» мәнін

қабылдайтын болады. Мұндағы s саны $(-l-2) \equiv s \, mod \, (128 * k)$ өрнегі арқылы табылады. Соңында, $M = M \parallel Pad(M)$ түрде толықтырылған жаңа M хабарламасын алынады. Мұнда, $Pad(M)$- толықтырылған соңғы бөлік, $\parallel$ – конкатенация операциясы, Pad – «padding» сөзінен қысқартылған белгілеу.

Хабарламаны бөліктерге бөлу. Алгоритм M хабарламаны оның көлеміне байланысты итеративті жолмен хештеуі себепті толықтырылған M хабарламасы *128*k*-битті t бөліктерге бөлінеді: $M(M_0, M_1, \dots, M_{t-1}) = M \parallel Pad(M) = M_0 \parallel M_1 \parallel \cdots \parallel M_{t-1}$.

Хештеу процесі. Бөлікше саны $k = 3$ ретінде қабылдап, 384 бит ұзындықтағы M_r $(r = 0,1, \dots, t-1)$ хабарламаның хештелуі Сурет 2.4-те көрсетілген. Хештеудің басында M хабарламасының алғашқы 384 битінен тұратын M_0 хабарлама бөлігі 128 биттен үш m^0, m^1, m^2 бөлікшелерге бөлінеді. Әр m^j хабарлама бөлімшесі негізінде *CFKey* раундтық кілттерді жасау алгоритмін пайдаланып, rk_i^j раундтық кілттер жасалады, мұндағы $i = 1,2, \dots, R_1$ және $j = 0,1, \dots, k-1$. Хештеу процесінде M_0 хабарлама бөлігі үшін инициализациялық вектор немесе бастапқы хеш-мән «0» мәнін қабылдайды, яғни $h_0^j = 0^{128}$. Шифрлаудың ең басында rk_0^j кілті кілттік ағарту жасау үшін пайдаланады. Бұдан әрі, әрбір үш бөлікше үшін кіріс мәндері ретінде rk_1^j және h_0^j қабылдайтын *CF* шифрлау алгоритмі бір уақытта қатарласа орындалады. *CF* алгоритмінен алынған h_1^j нәтижесі Девис-Мейер схемасы арқылы h_0^j-мен модуль 2 бойынша қосылып, p^j нәтижесін береді. p^j мәні *PerF* процедурасы әсерінен үш бөліктер арасында өзара орын ауыстыру жүргізіліп, p^j көршілес бөлікшелер мәндерімен 60% жаңартылған мәндер қабылдайды. Байттық орын ауыстыру процедурасы *PerF* төмендегі өрнек бойынша орындалады:

$$h_{ki+j} = h_{i+16j}, \;\; i = 0, \dots,15; \; j = 0, \dots, k-1. \tag{2.3}$$

Дербес жағдайда, $k = 3$ кезінде, жоғарғы формуланы төмендегідей өрнектеуге болады:

$$\left.\begin{matrix} h_{3i} = h_i \\ h_{3i+1} = h_{i+16} \\ h_{3i+2} = h_{i+32} \end{matrix}\right\} \;\; i = 0, \dots,15, \tag{2.4}$$

немесе Кесте 2.2 бойынша орын ауыстыруға болады.

Кесте 2.2 – Байттық орын ауыстыру

Байттық орын ауыстыру , *x* – байттың орны																
x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PerF(x)	0	16	32	1	17	33	2	18	34	3	19	35	4	20	36	5
x	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
PerF(x)	21	37	6	22	38	7	23	39	8	24	40	9	25	41	10	26
x	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
PerF(x)	42	11	27	43	12	28	44	13	29	45	14	30	46	15	31	47

Алынған h_1^j бөлікше мәндері келесі раундтық есептеулер процестерде h_0^j ретінде қабылданады. Бұдан әрі, аралық хеш-мәннің жаңартылған мәндері h_0^j мен келесі раундтық кілттер мәндері арқылы CF шифрлау алгоритмі қайта орындалатын болады. Осы жұмыс процедурасы әрбір хабарлама бөліктері $M_r(m^0, m^1, \dots, m^{k-1})$ үшін R_1 рет қайталанып жүргізіледі. $r = 0,1, \dots, t-1$. Келесі хабарлама бөлігі $M_{r+1}(m^0, m^1, \dots, m^{k-1})$ үшін аралық хеш-мәнді есептеу кезінде инициализациялық вектордың мәні ретінде $M_r(m^0, m^1, \dots, m^{k-1})$ хабарлама бөлігінен есептелінген h_0^j аралық хеш-мән мәні қабылданады.

Хабарламаның бастапқы $t-2$ бөліктері үшін хештеу алгоритмі сұлбасындағы $R_1 = 4$, ал соңғы M_{t-1} бөлігі үшін $R_1 = 8$ рет қайталанып орындалады (Сурет 2.4 – HBC-256 хештеу алгоритмінің сұлбасы). Соңында алынған *128*k* биттік аралық хеш-мән ComF функциясы арқылы 256 бит ұзындықтағы ақырғы хеш-мән алынатын болады:

$$h = ComF(h_{t-1}^0, h_{t-1}^1, \dots, h_{t-1}^k). \quad (2.5)$$

Ақырғы хеш-мәнді алу процедурасына ComF (Compression Function) дербес жағдай ретінде 256 биттен тұратын ақырғы хеш-мәнді алу үшін бірінші және екінші бөлікшелердің хеш-мәндерін конкатенациялау операциясы қолданылады: $h = h_{t-1}^0 \parallel h_{t-1}^1$.

2.3 HBC-256 хештеу алгоритмін параллелдеуге икемдеу

Блоктық шифрларға негізделген хеш функцияларды жобалаудағы тән мәселе – бұл әдетте блок шифрлардың блок ұзындықтарынының қысқа-ұзын болуында (мысалы, блок ұзындығы 128 бит болғанда, шифрға «туған күндер парадоксі» шабуылын қолдану мүмкіндігі бар-жоғы – 2^{64}). Егер хеш функциясының хеш-мәнін ұзарту мүмкін болмаса, онда хештің қажетті қауіпсіздік қасиеттерін сақтау үшін қиындық туғызады. Осы мәселені шешу жолының бірі болып, осы жұмыс арқылы көрсетілетін бағыт – бір уақытта бірнеше блок үшін блоктық шифрлау алгоритмін бірнеше рет паралелль орындау. Бұл өз кезегінде, Wide-pipe модификациясының қолданылу талаптарына сай келеді және жұмыс өнімділігін арттыруға көмектеседі.

Құрылған CF алгоритмін бір уақытта бірнеше рет қолданылады. Таңдап алынатын *k* параметрі 3-тен 8-ге дейін өзгере алады және CF алгоритмінің неше рет паралельді пайдаланатынын көрсетеді. Жұмыста *k*-ны бөліктер деп қарастырылып, оған қатысты хештеудің блогы ұзындығы 128* *k* бит болады. Жұмыста хештелетін ақпараттың (файлдың) көлеміне байланысты *k*-ны таңдау алу төмендегі кестемен ұсынылады.

Кесте 2.3 – *k* - бөлік санын таңдау

k	Ақпарат көлемі, *V*	*k*	Ақпарат көлемі, *V*
3	$V \leq 100$ КБ	6	10 МБ $< V \leq 100$ МБ
4	100 КБ $< V \leq 1$ МБ	7	100 МБ $< V \leq 1$ ГБ
5	1 МБ $< V \leq 10$ МБ	8	1 ГБ $< V$

3 НВС-256 ХЕШТЕУ АЛГОРИТМІНІҢ ҚАУІПСІЗДІК ҚАСИЕТТЕРІН ЗЕРТТЕУ

Жаңа құрылған хештеу алгоритмінің оның тәжірибедегі тиімділігімен қоса қауіпсіздігі қасиеттерін жан-жақты зерттеу жұмыстарын жүргізу міндетті талаптардың бірі болып саналады. Талдау жүргізудің жан-жақты және объективті болуы ғылыми тұрғыдан негізделіп, сенімділіктің нақты бағалануына үлкен назар аударған жөн. Қауіпсіздік пен тиімділікті бағалау барысында заманауи ақпараттық технологиялардың жетістіктері мен мүмкіндіктері, сондай-ақ ең кемінде, олардың алдағы жақын қысқа мерзімдегі даму қарқынын ескеру маңызды [45].

Ең алдымен, құрылған НВС-256 хештеу алгоритмін қауіпсіздігі қасиеттерін зерттеу барысында алынған хеш-мәндерді танымал статистикалық сынақтар арқылы псевдакездейсоқ тізбектердің сипаттарына ие бола алу мүмкіндігіне тексереміз. Зерттеудің осы бағытында біз неғұрлым көп сынақ жұмыстарын жүргізетін болсақ, соғұрым дәлдікке жақын нәтиже алатын боламыз. Талдау барысында жасалған хештеу алгоритмі арқылы алынған хеш-мәндердің үлкен көлемдегі жиынын қарастыратын боламыз. Егер жиындағы хеш-мәндер алдын-ала болжамды болса, онда бұл қолданылатын ең күшті хештеу алгоритмінің өзі де осал болып шығады – шабуылдаушы ықтимал хеш-мәндер кеңістігі тарыла түскендігін пайдаланып, оларды теру (немесе «сұрыптау») арқылы кейбір ақпаратты, дәлірек айтқанда, шабуылдаушы үшін коллизияны тауып ала алу мүмкіндігі арта түседі. Сол себепті, есептелінетін хеш-мән псевдокездейсоқ тізбекке неғұрым жақын болғаны дұрыс. NIST ұсынатын сынақтардың әрқайсысы кіріс тізбек ретінде хеш-мәндер жиынын қабылдайды. Әрі қарай, осы тізбектің белгілі бір қасиетін сипаттайтын статистика есептеледі – бұл бір мән немесе мәндер жиынтығы болуы мүмкін. Осыдан кейін бұл статистика эталондық статистикамен немесе математикалық тәсілдермен алынған «нөлдік гипотезамен» салыстырылады. Бәрімізге белгілі – іс жүзіндегі статистика біз қанша мықты, идеалды алгоритм пайдалансақ та, ол мән еш уақытта эталондық мәнмен 100% үйлесе бермейді. Сондықтан, 5% ауытқуды шекара ретінде аламыз. Егер іс жүзіндегі статистика мәні осы шекарадан аспаса, тәжірибе сынақтан өтті деп ұйғаратын боламыз [46-47].

Келесі кезекте хабарламаның әрбір элементтеріне аз мөлшерде өзгерістер енгізіп, алынған хеш-мәндердің өзгерістеріне жалпылама және жеке-жеке элементтеріне талдаулар жүргізетін боламыз. Бұл талдау аз өзгерістің лавиндік әсері және қатаң лавиндік әсері деп аталады. Егер алгоритм жеткілікті деңгейде лавин әсеріне ие бола алмаса, онда криптоталдаушы шығыс ақпарат негізінде кіріс ақпараты туралы болжам жасай алу мүмкіндігіне ие болады. Осылайша, шифрлау алгоритмдеріндегідей хештеу алгоримдерінде де лавиндік әсеріне қол жеткізу оны жасаудағы маңызды талап болып табылады және маңызды криптографиялық қасиет болып табылады.

Кез келген жаңа криптографиялық құрылым сияқты, НВС-256 алгоритмі де оның криптографиялық қасиеттерін, атап айтқанда: қайтымсыздығы мен бірінші және екінші тектегі коллизияларға төзімділігін растау үшін мұқият зерттеуді

қажет етеді. НВС-256 хештеу алгоритмі үшін сызықты емес элементтердің (S-блоктардың) дифференциалдық қасиеттері қарастырылады. Жұмыста раундтық сипаттамаларды құрудың әртүрлі нұсқалары қарастырылады. Хештеу үшін раундтық сипаттамаларды құруға және раундтық кілттерді жасау функциясында жұптық айырымдардың болуы туралы гипотеза ұсынылды. Айырымдар тізбегін құрудың ең оңтайлы тәсілі үшін де кіріс мәтіндердің дұрыс жұптарын табу ықтималдығы бастапқы кіріс мәтіннің 128-биттік бір блогының толық теру ықтималдығынан аз болатыны көрсетілген, бұл дифференциалды криптоталдау әдісін коллизялады табу үшін жарамсыз етеді. Сондай-ақ, алгебралық криптоталдау үшін НВС-256 хештеу алгоритмінің бір раунды үшін Transalg құралының көмегімен бульдік теңдеулер жүйесі құрылды. Коллизияны табу мақсатында құрылған жүйені шешу үшін параллельдеп есептеу мүмкіндігі бар plingeling есептеу нұсқасын қамтитын lingeling SAT-шешуші пайдаланылды.

3.1 Алгоритмге жасалатын негізгі шабуылдардың күрделілігін талдау

Кез-келген хештеу функциясының сенімділігін бағалу кезінде төмендегідей үш мәселе зерттеледі [48]:

Түпбейнені іздеу, яғни алдын-ала белгілі хеш-мән $h(M)$ арқылы алғашқы M хабарламасын қалпына келтіру;

Екінші түпбейнені іздеу, яғни хеш-мәндері үшін $h(M_1) = h(M_2)$ орындалатындай бізге белгілі M_1 хабарламасынан бөлек M_2 хабарламасын табу, мұнда $M_1 \neq M_2$.

Коллизияны іздеу, яғни хеш-мәндері үшін $h(M_1) = h(M_2)$ орындалатын кез-келген M_1 және M_2 хабарламаларын табу.

Осы аталған мәселелер НВС-256 алгоритміне қатысты келесі параметрлермен нақтыланған. НВС-256 хеш-мән ұзындығы $n = 256$ бит, сәйкесінше барлық мүмкін болатын хеш-мәндердің саны $N = 2^{256}$. Үш мәселенің әрқайсысын $p = 0.5$ ықтималдықпен іске асырудың ең аз K мәнін анықтаймыз, мұнда K – хабарламалар саны. Кесте 3.1-де әрбір үш мәселе үшін шабуылдардың күрделілігі туралы деректер берілген [49].

Кесте 3.1 – Түпбейне мен коллизияларды іздеудің күрделілігі туралы деректер

	Мәселелер		
	Түпбейнені іздеу	Екінші түпбейнені іздеу	Коллизияны іздеу
$p = 0.5$ және $N = 2^{256}$ болғандағы K-ның мәні	$K = 0.69 \cdot 2^{256}$	$K = 0.69 \cdot 2^{256} + 1$	$K = 0.83 \cdot 2^{128}$
Қолданылатын әдіс	Толық теру әдісі	Толық теру әдісі	«Туған күндер парадоксы» әдісі

3.2 Хеш-мәндердің статистикалық қасиеттерін бағалау

Кез-келген еркін ұзындықтағы M хабарламасы үшін жасалған НВС-256 хештеу алгоритмінің $h(M)$ хеш-мәні жалған кездейсоқ тізбектердің қасиеттерін қанағаттандыруы міндетті шарттардың бірі болып саналады. Бұл хештеу алгоритмдеріне қойылатын негізгі талаптардың бірі, яғни хеш функциясына

негізделген жалғанкездейсоқ сандар генераторын кездейсоқ сандар генераторынан ажырату қиын болуы керек [50-52].

NIST статистикалық сынақтар жиынтығы арқылы бағалау.

Неғұрлым сенімді статистикалық бағалау үшін Ұлттық стандарттар және технологиялар институтының (бұдан әрі – NIST) статистикалық сынақтар жиынтығын қолдана отырып, HBC-256 хеш функциясының кездейсоқтық қасиетін бағалау қажет. NIST ұсынған статистикалық сынақтар жиынтығы құрамына 15 статистикалық тест кіреді, бірақ осы 15 сынақтың бірнешеуі кіріс параметрлерге (күйлерге) байланысты бірнеше сынақтардың жиынтығы болып табылады, сондықтан іс жүзінде жүргізілетін сынақтар саны 15-тен әлдеқайда көп. Сынақтардың мақсаты берілген тізбектің шынайы кездейсоқ тізбектерден ауытқушылық өлшемін анықтау болып табылады. Бұл сынақтар шынайы кездейсоқ тізбектерге тән әртүрлі статистикалық қасиеттерге негізделген. Сынақтар екілік санау жүйесіндегі тізбектерге (екілік тізбектерге) жүргізіледі [53].

Әр сынақ кездейсоқтықты белгілі бір критерийлер бойынша бағалап, сәйкесінше, функцияны *p*-мәндері бойынша бағалады. *p*-мәні – бұл есептелген тексеру статистикасы H_0 нөлдік гипотезадан бас тартуға әкелетін маңыздылық деңгейінің ең кіші мәні (яғни ақиқат гипотезадан бас тарту ықтималдығы). Біздің жағдайда, егер сынақ жүргізу нәтижесінде есептелген $p > \alpha=0,01$ мәніне ие болса, бұл зерттелетін екілік тізбек 99% сенімділікпен кездейсоқ болады дегенді білдіреді. Мұндағы $p \in[0,1]$, α – маңыздылық деңгейі, яғни H_0 нөлдік гипотезаны қабылдамау ықтималдығы [54].

HBC-256 алгоритмінің хеш-мәндердің кездейсоқтығын талдау үшін келесі сынақтар жүргізілді: [55, 56]:

1) Жиілік (монобиттік) сынақ: Бұл сынақтың мақсаты – екілік тізбектегі нөлдер мен бірліктердің ара-қатынасын өлшеу. Өту критерийлері бірлік пен нөлдердің сандарының (жиілігінің) жақындығымен анықталады, олар шамамен бір-біріне жақын болуы керек.

2) Блок ішіндегі жиіліктер сынағы: бұл сынақ жиілік сынағының жалғасы болып табылады, онда бірлік пен нөлдің арақатынасы *M*-биттік блоктың өлшемін қолдану арқылы анықталады. Сынақтан өту критерийі болып *M*-биттік блоктағы бірліктер мен нөлдердің жиілігі *M*/2-ге жақын болуы саналады.

3) Бірдей биттердің тізбектері сынағы: бұл сынақтың мақсаты – әртүрлі ұзындықтағы тек бірліктер немесе тек нөлдерден тұратын тізбекшелердің саны кездейсоқ тізбектегі санына сәйкес келетіндігі туралы қорытынды жасау. Атап айтқанда, қарастырылатын тізбектегі тек бірліктер мен тек нөлдер тізбекшелері өзара жиі немесе сирек ауысып отыратынын тексереді.

4) Блоктағы бірліктердің ең ұзын тізбекшесі сынағы: бұл сынақ «Бірдей биттердің тізбегі сынағының» жалғасы болып табылады. Сынақ мақсаты – мұндай бірлік тізбекшенің ұзындығы қарастырылған екілік тізбек ұзындығындай кездейсоқ екілік тізбектегі бірліктердің ең ұзын бірлік тізбегінің ұзындығына сәйкес келетіндігін-келмейтіндігін анықтау.

5) Екілік матрицаның рангін анықтау сынағы: Мұнда қарастырылған екілік тізбектен алынған қиылыспайтын ішкі екілік матрицалардың рангтері

есептеледі. Бұл сынақтың мақсаты – алғашқы тізбекті құрайтын тұрақты ұзындықтағы ішкі тізбекшелердің өзара сызықтық тәуелділігін тексеру.

6) Дискретті Фурье түрлендіру сынағы (спектрлік сынақ): Бұл сынақтың мақсаты – алғашқы екілік тізбектің периодтық қасиеттерін анықтау, мысалы, бір-біріне жақын орналасқан қайталанатын үзінділерді анықтау.

7) Қабаттаспайтын үлгілерге сәйкестік сынағы: Бұл сынақ алғашқы тізбекте табылған алдын-ала енгізілген үлгілердің негізінде нөлдер мен бірліктердің бірқалыпты үлестірілуін бағалайды.

8) Қабаттасатын үлгілерге сәйкестік сынағы: Бұл бір-біріне сәйкес келмейтін үлгіні тексеруге ұқсас, тек айырмашылығы - үлгіні тапқан кезде, тор келесі іздеуді жалғастырмас бұрын бір битке ауысады.

9) Маурердің «әмбебап статистикалық» сынағы: Мұнда алғашқы екілік тізбектегі бірдей үлгілер арасындағы биттер саны анықталады. Соның негізінде тізбектің айтарлықтай қысылғанын немесе қысылмағанын тексереді. Айтарлықтай қысылған реттілік кездейсоқ емес деп анықталады.

10) Сызықтық күрделілік сынағы: бұл сынақ кездейсоқтықты тексеру үшін берілген екілік тізбектің сызықтық күрделілігін бағалайды. Сынақ кері байланысты жылжудың сызықтық регистрінің жұмыс істеу принципі негізінде жүргізіледі.

11) Сериялық сынақ: бұл сынақ берілген екілік тізбектегі барлық мүмкін болатын қабаттасатын үлгілерді анықтайды. Сынақ мақсаты – берілген биттік тізбектегі M-биттік ұзындықтағы қабаттасатын 2^M үлгілерінің пайда болу саны шынымен кездейсоқ биттік тізбектегімен бірдей ме, жоқ па, соны анықтау.

12) Жуықтау энтропиялық сынақ: Алдыңғы сериялық сынақ сияқты бұл сынақ алғашқы биттік тізбек бойынша M-биттік ұзындықтағы үлгілердің барлық ықтимал қабаттасу жиілігін санауға бағытталған. Сынақтың мақсаты – алғашқы биттік тізбектің ұзындығы M және $M+1$ бит болатын екі тізбектескен блоктарының қабаттасу жиіліктерін кездейсоқ биттік тізбектің сынақ аналогіндегі ұқсас блоктардың қабаттасу жиіліктерімен салыстыру.

13) Жиынтық қосынды сынағы (Cusums сынағы): бұл сынақ кездейсоқтықты тексеру кезінде ішінара тізбекшелердің жиынтық қосындысы тым аз немесе тым үлкен екенін анықтайды, нәтижені кездейсоқ екілік тізбектегі ішінара тізбекшелердің жиынтық қосындысымен салыстырады. Мұны тікелей сынақ немесе сынақ арқылы бағалауға болады.

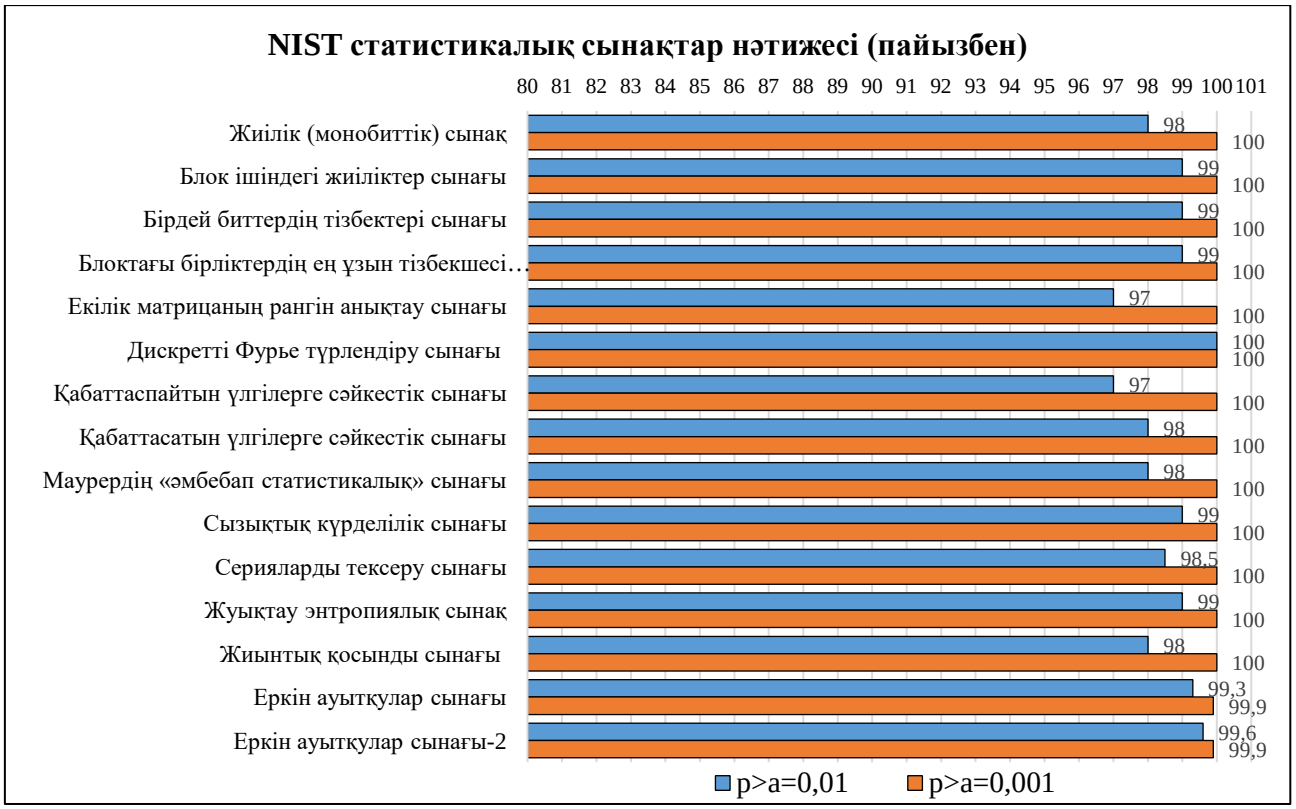
14) Еркін ауытқулар сынағы: Бұл сынақтың мақсаты – кездейсоқтықты өлшеу үшін қандай да бір ауытқу орын алатын немесе болмайтын циклдегі белгілі бір күйге бару санын анықтау. Іс жүзінде бұл сынақ циклдың сегіз (-4, -3, -2, -1, +1, +2, +3, +4) күйінің әрқайсысы үшін жүргізілген сегіз тәуелсіз сынақтан тұрады.

15) Еркін ауытқулар сынағы-2: Алдыңғы сынаққа ұқсас. Ерекшелігі – әртүрлі күйлерге барудың күтілетін санынан ауытқуларды ерікті түрде айналып өту кезіндегі жалпы санын анықтау. Сынақ әр жағдайға арналған 18 сынақтан тұрады: -9, -8, ..., -1, +1, +2, ... , +9.

Талдау үшін NIST ұсынған статистикалық сынақтардың жиынтығы ресми дереккөзден алынған бағдарламалық жасақтама пайдаланылды [56].

Ұсынылған хештеу алгоритмін NIST ұсынған статистикалық сынақтардан өткізу төмендегідей жүргізілді. Тестілеу үшін көлемі 19 200 000 Б болатын файл *.zip форматындағы кездейсоқ архив файл алынды. Бұл форматтағы файл таңдау себебі, файл мазмұнында қайталаулар санын минимизициялау болды. Алгоритмнің сипаттамасына сәйкес файлдағы кезекті әрбір 48 байт ақпарат хештеу алгоритміне хабарлама ретінде енгізіліп, нәтижесінде соған сәйкес 32 байттағы хеш-мән болып белгілі бір файлға тізбектеліп жазылып отырады, яғни 400 мың хабарламаға сәйкес 400 мың хеш-мән жазылған. Шығыс нәтижелер – хеш-мәндер жазылған *.hash форматындағы файлдың көлемі 12 800 000 байтты құрады. Шығыс файлды әрқайсысы 12500 байт болатын 100 файлға (тізбекке) бөлінді. Әрі қарай, алынған файлдарға NIST статистикалық сынақтар жиынтығы қолданылды.

Төменгі Сурет 3.1-де NIST статистикалық сынақтар жиынтығы арқылы құрылған хештеу алгоритмімен алынған хеш-мәндер тізбегінің псевдокездейсоқтыққа қатысты талдау көрсетілген. Көрсетілген суретте α – маңыздылық деңгейінің екі жағдайы қарастырылған және пайыздық көрсеткішпен сынақтан өту дәрежесі бейнеленген.



Сурет 3.1 –NIST статистикалық сынақтар нәтижелері

Алынған нәтиже бойынша барлық сынақтар сәтті өтті. NIST ұсыныстары бойынша, егер сынақтан өткен тізбектердің ең аз саны 100 тізбектің 96 тізбегі болса, сынақ сәтті өтті деп саналады. Сынақ нәтижелері HBC-256 хештеу алгоритмі арқылы алынған тізбектерде ауытқулардың жоқ екендігін көрсетті. Осылайша, алгоритм статистикалық қауіпсіздіктің жоғары деңгейіне ие.

Д.Кнуттың статистикалық сынақтар жиынтығы арқылы бағалау.

NIST ұсынған статистикалық сынақтар жиынтығын пайдаланумен қатар, құрылған хештеу алгоритмінен алынған хеш-мәндердің кездейсоқтығын бағалау үшін американдық ғалым, Стэнфорд университетінің профессоры Дональд Эрвин Кнут ұсынған статистикалық сынақтар жиынтығы қолданылды.

Бұл сынақтар жиындығын 1969 жылы Д. Кнут өзінің "Компьютерге арналған бағдарламалау өнері" атты жұмысында келтірді [57]. Статистиканың есептелген мәні, яғни χ^2-Присонның хи-квадраты кестелік нәтижелермен салыстырылады және мұндай статистиканың пайда болу ықтималдығына байланысты оның сапасы туралы қорытынды жасалады. Бұл сынақтардың артықшылықтарының бірі – олардың аз саны және жылдам орындау алгоритмдерінің болуы.

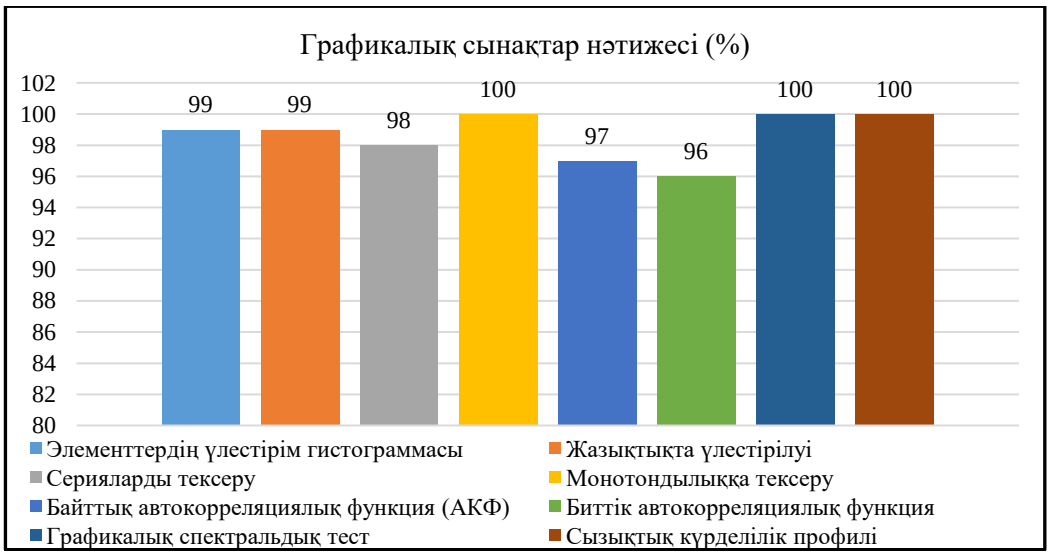
Алынған хеш-мәндердің кездейсоқтығын бағалау үшін статистикалық сынақтар жиынтығын жүзеге асыратын «Д. Кнуттың статистикалық тесттер мен графикалық тестілерді таңдаудың автоматтандырылған жүйесі» бағдарламалық кешенін қолданып, төмендегідей нәтижелер алынды [58]. Салыстыру жүргізу мақсатында осы сынақтардан жоғарғыдағы NIST ұсынған статистикалық сынақтарды жүргізуге пайдаланған әрқайсысында 12500 байт ақпарат бар 100 файлдар (тізбектер) өткізілді.

Статистикалық қасиеттерді анықтау үшін келесі графикалық және бағалау сынақтар қолданылды:

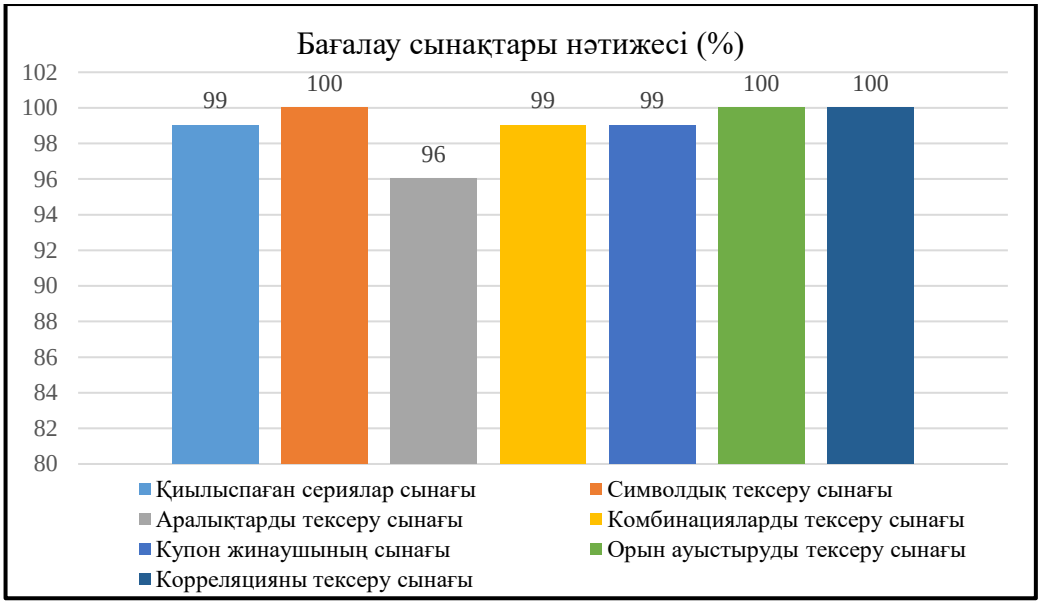
1) Графикалық сынақтар:
   - тізбек элементтерінің таралу гистограммасы сынағы;
   - жазықтықтағы үлестірім сынағы;
   - сериялардың тексеру сынағы;
   - монотондылықты тексеру сынағы;
   - автокорреляция функциясы сынағы;
   - сызықтық күрделілік профилі сынағы;
   - графикалық спектрлік сынақ.
2) Бағалау сынақтары:
   - тіркеспеген сериялар сынағы;
   - символдық тексеру сынағы;
   - аралықтарды тексеру сынағы;
   - комбинацияларды тексеру сынағы;
   - купон жинаушының сынағы;
   - орын ауыстыруды тексеру сынағы;
   - корреляцияны тексеру сынағы.

Біздің бағалау процесімізде сенімділік интервалын 0,01 деп алып, хеш-мән тізбектеріне статистикалық бағалау жүргізілді.

Сурет 3.2 мен Сурет 3.3-те графикалық және бағалау сынақтарынан сәтті өткен файлдар саны туралы мәліметтер диаграмма түрінде көрсетілген.



Сурет 3.2 –Кнуттың статистикалық графикалық сынақтары нәтижелері



Сурет 3.3 –Кнуттың статистикалық бағалау сынақтары нәтижелері

Зерттеу барысында қарастырылған хеш-мәндер тізбегі NIST және Кнут статистикалық сынақтарды жиынтығы бойынша нәтижелер алынды. Нәтижелерді бағалаудан алынған хеш-мәндер тізбегі статистикалық қауіпсіз деп айтуға болады. Осылайша, HBC-256 хештеу алгоритмі жақсы статистикалық қауіпсіздікке ие деп қорытындылауға негіз бар.

3.3 Алгоритмнің лавиндік және қатаң лавиндік әсерін бағалау

Қазіргі уақытта хештеу және шифрлау алгоритмдерінің криптографиялық беріктілігін сызықтық және дифференциалды криптоталдау әдістеріне сүйеніп бағалау кеңінен қолданылады [59]. Дифференциалды криптоталдау әдісі түрлендірудің әрбір раундындағы кіріс биттерінің айырымының өзгеруіне байланысты шығыс биттерінің айырымдарының өзгеруін қадағалау болып табылады. Алгоритмде «лавин әсерінің» болуы дифференциалдық

криптоталдауға криптографиялық беріктілікті қамтамасыз етудің қажетті шарты болып табылатынын ескеру қажет. Егер алгоритм қажетті дәрежеде биттік шашырау әсерімен қамтамасыз етілмесе, онда криптоталдаушы шығыс биттер негізінде кіріс биттер туралы ақпарат алуға мүмкіндік алады [60, 61].

Лавиндік әсерді талдау үшін әдетте келесі екі критерий қолданылады [62]:

– лавиндік критериі;

– қатаң лавиндік критерийі.

Лавиндік критерийі кіріс ашық мәтіннің әрбір битін өзгерткенде, шығыс шифрмәтін биттерінің шамамен жартысына жуығының өзгеру әсері және бұл әсер мына формуламен анықталады:

$$\varepsilon_{a\,i} = |2k_i - 1|, \tag{3.1}$$

мұнда, i – кіріс ашық мәтіндегі өзгертілген биттің нөмірі, k_i – бастапқы (өзгермейтін) кіріс мәтін мен шығыс шифрмәтінді салыстырғанда, кіріс мәтіндегі і-ші бит өзгерген кезде шифрмәтіндегі биттердің жартысына жуығының өзгеру ықтималдығы. Ал, қатаң лавиндік критерий шифрмәтіннің биттерінің өзгерісіне өте қатал талап қояды: яғни өзгертілген кіріс ашық мәтіннің әрбір битіне байланысты әрбір шифрмәтін битінің өзгеру қасиетін қарастырады. Теорияда бұл өзгерістің ықтималдығы 0,5-ке жуықтау болуы қажет. Қатаң лавиндік критерийі төмендегі формуламен анықталады:

$$\varepsilon_{s\,i,j} = \left|2 * k_{s\,i,j} - 1\right|, \tag{3.2}$$

мұндағы i –кіріс ашық мәтіндегі өзгертілген биттің нөмірі, j – шығыс шифрмәтіннің қарастырылатын битінің нөмірі, $k_{s\,i,j}$ –шығыстағы j-биттің кірістегі өзгертілген i-битке қатысты өзгеруінің ықтималдығы. ε_a- лавиндік параметр, ε_s- қатаң лавиндік параметр деп аталады [63].

Лавиндік әсері талдауын алдымен CF шифрлау алгоритміне жүргізейік [64]. Шифрлау алгоритмі сызбасы негізінде лавиндік әсердің таратылуы және лавиндік критерийдің 1-ші, 2-ші және 4-ші раундтардан кейінгі орындалуы зерттелді. Нәтижелер Кесте 3.2, Кесте 3.3 және Кесте 3.4 көрсетілген.

Кесте 3.2 – 1-раундтан кейін CF алгоритмінің лавиндік әсерін талдау

i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i
1	0,45	17	0,45	33	0,44	49	0,52	65	0,42	81	0,54	97	0,54	113	0,52
2	0,54	18	0,45	34	0,48	50	0,52	66	0,49	82	0,47	98	0,43	114	0,53
3	0,55	19	0,45	35	0,51	51	0,57	67	0,52	83	0,54	99	0,53	115	0,58
4	0,47	20	0,45	36	0,53	52	0,55	68	0,58	84	0,48	100	0,50	116	0,52
5	0,58	21	0,45	37	0,45	53	0,55	69	0,49	85	0,49	101	0,52	117	0,47
6	0,57	22	0,45	38	0,52	54	0,45	70	0,52	86	0,46	102	0,45	118	0,61
7	0,41	23	0,45	39	0,54	55	0,55	71	0,51	87	0,43	103	0,51	119	0,46
8	0,48	24	0,45	40	0,55	56	0,48	72	0,48	88	0,55	104	0,46	120	0,47
9	0,50	25	0,45	41	0,52	57	0,51	73	0,52	89	0,52	105	0,46	121	0,52
10	0,51	26	0,45	42	0,51	58	0,40	74	0,48	90	0,47	106	0,52	122	0,52
11	0,52	27	0,45	43	0,52	59	0,42	75	0,48	91	0,56	107	0,54	123	0,53
12	0,50	28	0,45	44	0,56	60	0,46	76	0,55	92	0,52	108	0,55	124	0,51
13	0,48	29	0,54	45	0,47	61	0,45	77	0,48	93	0,51	109	0,52	125	0,47
14	0,55	30	0,55	46	0,52	62	0,56	78	0,53	94	0,50	110	0,52	126	0,56
15	0,53	31	0,53	47	0,59	63	0,55	79	0,49	95	0,54	111	0,57	127	0,45
16	0,47	32	0,59	48	0,46	64	0,45	80	0,52	96	0,44	112	0,46	128	0,36

Кесте 3.3 – 2-раундтан кейін CF алгоритмінің лавиндік әсерін талдау

i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i
1	0,41	17	0,48	33	0,51	49	0,52	65	0,55	81	0,54	97	0,49	113	0,45
2	0,55	18	0,41	34	0,46	50	0,50	66	0,50	82	0,48	98	0,50	114	0,52
3	0,56	19	0,56	35	0,47	51	0,50	67	0,40	83	0,54	99	0,50	115	0,48
4	0,48	20	0,47	36	0,59	52	0,48	68	0,48	84	0,55	100	0,49	116	0,48
5	0,40	21	0,45	37	0,55	53	0,55	69	0,55	85	0,46	101	0,54	117	0,48
6	0,51	22	0,50	38	0,51	54	0,48	70	0,50	86	0,49	102	0,50	118	0,55
7	0,43	23	0,48	39	0,49	55	0,49	71	0,48	87	0,56	103	0,54	119	0,55
8	0,46	24	0,48	40	0,55	56	0,67	72	0,52	88	0,54	104	0,60	120	0,47
9	0,59	25	0,54	41	0,48	57	0,49	73	0,49	89	0,48	105	0,58	121	0,52
10	0,55	26	0,49	42	0,51	58	0,52	74	0,52	90	0,51	106	0,53	122	0,49
11	0,52	27	0,52	43	0,60	59	0,41	75	0,49	91	0,51	107	0,51	123	0,48
12	0,55	28	0,59	44	0,52	60	0,49	76	0,49	92	0,43	108	0,47	124	0,52
13	0,48	29	0,56	45	0,63	61	0,54	77	0,57	93	0,50	109	0,45	125	0,50
14	0,51	30	0,45	46	0,48	62	0,46	78	0,45	94	0,58	110	0,45	126	0,48
15	0,47	31	0,50	47	0,52	63	0,49	79	0,48	95	0,48	111	0,45	127	0,48
16	0,43	32	0,45	48	0,53	64	0,42	80	0,46	96	0,43	112	0,52	128	0,53

Кесте 3.4 – 4-раундтан кейін CF алгоритмінің лавиндік әсерін талдау

i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i	i	k_i
1	0,48	17	0,44	33	0,52	49	0,52	65	0,44	81	0,42	97	0,54	113	0,45
2	0,56	18	0,49	34	0,50	50	0,57	66	0,48	82	0,51	98	0,52	114	0,43
3	0,52	19	0,55	35	0,59	51	0,55	67	0,48	83	0,48	99	0,46	115	0,58
4	0,53	20	0,54	36	0,50	52	0,53	68	0,47	84	0,52	100	0,38	116	0,48
5	0,53	21	0,48	37	0,50	53	0,51	69	0,47	85	0,52	101	0,49	117	0,53
6	0,52	22	0,46	38	0,52	54	0,52	70	0,49	86	0,53	102	0,46	118	0,45
7	0,41	23	0,42	39	0,49	55	0,52	71	0,50	87	0,48	103	0,49	119	0,50
8	0,43	24	0,45	40	0,46	56	0,56	72	0,61	88	0,51	104	0,51	120	0,46
9	0,54	25	0,50	41	0,48	57	0,55	73	0,51	89	0,59	105	0,45	121	0,48
10	0,45	26	0,55	42	0,52	58	0,41	74	0,54	90	0,45	106	0,46	122	0,39
11	0,51	27	0,55	43	0,48	59	0,51	75	0,51	91	0,38	107	0,47	123	0,52
12	0,43	28	0,42	44	0,45	60	0,58	76	0,45	92	0,53	108	0,55	124	0,51
13	0,49	29	0,52	45	0,53	61	0,52	77	0,54	93	0,45	109	0,53	125	0,45
14	0,55	30	0,46	46	0,50	62	0,50	78	0,52	94	0,51	110	0,58	126	0,55
15	0,50	31	0,45	47	0,55	63	0,65	79	0,51	95	0,52	111	0,46	127	0,52
16	0,52	32	0,53	48	0,47	64	0,52	80	0,52	96	0,48	112	0,51	128	0,47

Мысал ретінде талдау үшін 0xCC156C4CE024D5113D680D7CCE6D8B2 мәніне ие 128-биттік кездейсоқ M_0 хабарламасы таңдалды. $1 \le i \le 128$ үшін 128 M_i ашық хабарламасы $M_{129-i} = M_0 \oplus (i \ll 1)$ формуласы арқылы жасалып алынды. Әр M_i бастапқы M_0-ден 1 бит айырмашылығымен жасалған.

Осы 129 M_i $(i = 0,1,\dots,128)$ хабарламаға CF қолданылғаннан кейін сәйкес 128-биттік C_i шифрлық мәтіндері алынды. Содан кейін C_0 шифрмәтіні мен қалған 128 шифрмәтін арасындағы k_i $(i = 1,2,\dots,128)$ ықтималдықтары есептелді. Қарастырылған раундтарда жасалған тәжірибе k_i мәндерінің 98,5%-і (0,41; 0,59) интервалында жатқанын көрсетті. Баршаға белгілі, лавиндік параметрдің өзгеру диапазоны 0-ден 1-ге дейінгі аралықта жатады. Лавин параметрінің мәні неғұрым нөлге жақын болған сайын, лавиндік әсер шифрлау алгоритмінде соғұрлым жоғары болып көрінеді. Жүргізілген талдау 1, 2 және 4 раундтар үшін лавиннің ε_a параметрінің орташа мәндері сәйкесінше 0,074, 0,071 және 0,073 екенін көрсетті. Тәжірибеден байқағанымыздай, CF алгоритмінің лавиндік әсері шифрлаудың тіпті бірінші раундынан кейін де жоғары көрсеткіш көрсетіп тұр. Жасалған эксперименттің нақтылығын растау үшін лавиндік критерий шифрлаудың 8-ші, 16-шы және 24-ші раундтары үшін жүргізілді және алынған нәтижелер CF алгоритмінің лавиндік әсерінің қажетті деңгейде таралу дәрежесін растады.

Әрі қарай HBC-256 хештеу алгоритмінің лавиндік әсерін зерттеп, нәтижелеріне бағалау жүргізу қарастырылады. Бастапқы 48 байттық кездейсоқ M_0 хабарламасының мысалы ретінде төменгі 0x00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000 алынды. Бұдан әрі 328-биттік M_i хабарламалары келесі түрде жасалатын болады: $M_{385-i} = M_0 \oplus (i \ll 1), i = 1,\dots,328.$

Біз 1-ші, 2-ші, 4-ші, 8-ші және 12-ші раундтардан кейін хеш нәтижелерін талдауды қарастырдық. Бірінші раундтан кейін алгоритмнің лавиндік параметрінің орташа мәні 0,66 қабылдап, ең нашар екені анықталды. Дегенмен, CF алгоритмінің лавиндік әсерінің таралу дәрежесінің жоғары болуына байланысты хештеудің 2-ші раундынан бастап-ақ бит шашырауының бізге қажетті жоғарғы деңгейі байқалады. Кесте 3.5-те ε_a лавиндік параметрінің статистикалық көрсеткіштерінің өзгерісі хештеу алгоритмінің раундтарына байланысты қалай өзгеріп отыратыны көрсетілген.

Кесте 3.5 – HBC-256 алгоритмінің лавиндік параметрінің статистикалық көрсеткіштері

Статистикалық көрсеткіштер	1-раунд	2-раунд	4-раунд	8-раунд	12-раунд
Максимальды мән	0,7240	0,1870	0,1770	0,1720	0,1720
Минимальды мән	0,5940	0	0	0	0
Арифметикалық орта мән	0,6645	0,0399	0,0407	0,0405	0,0398
Дисперсия	0,0007	0,0009	0,0009	0,0009	0,0009
Мода	0,6560	0,0310	0,0160	0,0150	0,0260

Жоғарғы Кесте 3.5-тен қарастырылған хеш функция 1-ші раундтан кейін лавиндік әсер қажетті дәрежеде қамтамасыз етілмейтінін көруге болады. Өзгертілген биттің орнына байланысты алынған оның мәндері (0,594, 0,724) интервалында жатады, бұл өз кезегінде 0-ден әлдеқайда қашық. Дегенмен, 2-ші және одан кейінгі раундтардан кейін алынған статистикалық көрсеткіштер бірдей деңгейдегі мәндерді қабылдайды, яғни олардың бір-бірінен ауытқу диапазоны өте кішкентай.

Сурет 3.4 арқылы бірқалыпты үлестіруді көрсететін 4-ші раундтан кейінгі k_i өзгеру ықтималдығы көрсетілген. Осыдан кіріс ашық мәтіннің бір битінің өзгеруі 328 биттік хеш-мәннің 50 пайыздық өзгеруіне әкеледі деген қорытынды жасауға болады. k_i ықтималдықтарының χ^2 (Хи-квадрат) мәні 189,49-ге тең. Әрі қарай, сенімділік мәні $\alpha = 0{,}05$ және еркіндік дәрежесі $df = 383$ болғанда, H_0 нөлдік гипотезамен келісудің қажетті деңгейі $\chi^2_{\alpha=0{,}05,\ df=383}$=429,63. Біздің жағдайда $\chi^2 = 189{,}49 < \chi^2_{\alpha=0{,}05,\ df=383} = 429{,}63$, сондықтан алынған ықтималдылықтар k_i оң нәтиже көрсетіп, сәйкесінше HBC-256 хештеу алгоритмі лавиндік критерийлер талаптарын қанағаттандырып отыр.

Сурет 3.4 – Биттердің өзгеру ықтималдығы

S-блок ауыстыруының қатаң лавиндік әсері қасиеттерін зерттеу.

Қарастырылған төрт S-блоктары үшін қатаң лавиндік әсерін қарастырайық. Лавиндік әсердің қатаң критерийі (SAC) S-блоктарды бағалаудың негізгі критерийлерінің бірі болып табылады. Ол дифференциалды криптоталдауға беріктілікті сипаттайтын S-блоктарды синтездеу процесінде кеңінен қолданылады [65]. Бәрімізге белгілі, бульдік функцияларды S-блок құрылымының бөлігі ретінде қарастыруға болады. SAC-ті қанағаттандыратын бульдік функцияларға негізделген S-блоктардың құрылымдары алғаш рет Карлайл Адамс, Стаффорд Таварес және Гванджо Ким зерттеді. Бульдік функция үшін қатаң лавиндік критерийін зерттеу келесі белгілерге, түсініктерге және анықтамаларға негізделген [66].

Бізде $\mathcal{F}_2^n$ – n өлшемді екілік векторлық кеңістік болсын және мұндағы $\mathcal{F}_2$ – $\{0,1\}$ элементтерінен тұратын Галуа өрісі болсын. n мен m – натурал сандар, F векторлы бульдік функцияны мына түрде анықталады: $F: \mathcal{F}_2^n \longmapsto \mathcal{F}_2^m$.

1-анықтама. $F(x) = (f_1, f_2, \dots, f_m)$ функциясындағы $f_1, f_2, \dots, f_m$ – бульдік функциялары F бульдік функцияның координаталары деп аталады. m *=1* кезінде векторлы бульдік функция шығысында тек бір бит ғана болатын кәдімгі бульдік функцияға эквивалентті.

2-анықтама. $f(x): \mathcal{F}_2^n \longmapsto \mathcal{F}_2$ – n-айнымалысы бар бульдік функция болсын, мұндағы $x = (x_0, x_1, \dots, x_{n-1})$. Онда $f(x)$ функциясының хемминг салмағы былай анықталады:

$$hw(f) = \sum_{x=0}^{2^n-1} f(x). \tag{3.3}$$

3-анықтама. $f(x): \mathcal{F}_2^n \longmapsto \mathcal{F}_2$ бульді функциясы болсын. Онда $f(x)$ функциясының $u \in \mathcal{F}_2^n$ екілік векторы арқылы алынған өсімшесі былай анықталады:

$$D_u f(x) = f(x) \oplus f(x+u). \tag{3.4}$$

4-анықтама. Қандай да бір бульдік функция $f(x)$ қатаң лавиндік критерийді қанағаттандырады деп айтамыз, егер $u \in \mathcal{F}_2^n$ үшін төмендегідей теңдеулер жүйесі орындалса:

$$\begin{cases} hw(u) = 1; \\ \sum_{x=0}^{2^n-1} f(x) \oplus f(x+u) = 2^{n-1}; \end{cases} \tag{3.5}$$

немесе ықтималдықтар түрінде былай жазуға болады:

$$\begin{cases} hw(u) = 1; \\ p\{f(x) = f(x+u)\} = 0.5. \end{cases} \tag{3.6}$$

Енді, негізгі жұмыс – S-блоктарға қатаң лавиндік критерийі көшейік. Түсінікті болу үшін төрт «алтын» S-блоктың біріншісіне (S_1-блок) жүргізілген

талдауды толықтай қадамдап жүргізейік. S_1-блокты декомпозиция арқылы бульдік функция компоненттерімен жазып алайық:

$$S_1 = \begin{Bmatrix} \boldsymbol{0} & \boldsymbol{F} & \boldsymbol{B} & \boldsymbol{8} & \boldsymbol{C} & \boldsymbol{9} & \boldsymbol{6} & \boldsymbol{3} & \boldsymbol{D} & \boldsymbol{1} & \boldsymbol{2} & \boldsymbol{4} & \boldsymbol{A} & \boldsymbol{7} & \boldsymbol{5} & \boldsymbol{E} \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{Bmatrix} \quad (3.7)$$

Енді, (8)-формуладан біз S-box бірінші жолдың компоненттік мәндері негізінде төрт айнымалысы бар ($n=4$) бульдік функциясын қатаң лавиндік критерийіне сәйкестігін зерттеуге көшеміз: $f_1(x_1, x_2, x_3, x_4) = \{0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\}$.

Бұдан әрі 3-ші және 4-анықтамаға сүйене отырып, келесі кестені құрайық. Бұл Кесте 3.6-да $f_1(x)$ бульдік функцияның төрт айнымалысының барлық мүмкін мәндеріндегі (9)-формулаға сәйкес нәтижелері, $f_1(x)$ бульдік функцияның $hw(u) = 1$ өсімшесімен қосылған аргументіндегі мәні және $D_u f_1(x)$ өсімшесінің нәтижелері көрсетілген.

Кесте 3.6 –$f(x)$ бульдік функциясының өсімшелердің мәндерін анықтау

$f_1(x)$	$f_1(x \oplus 0001)$	$D_{0001}f_1(x)$	$f_1(x \oplus 0010)$	$D_{0010}f_1(x)$	$f_1(x \oplus 0100)$	$D_{0100}f_1(x)$	$f_1(x \oplus 1000)$	$D_{1000}f_1(x)$
$f(0000)=0$	$f(0001)=1$	1	$f(0010)=1$	1	$f(0100)=0$	0	$f(1000)=1$	1
$f(0001)=1$	$f(0000)=0$	1	$f(0011)=0$	1	$f(0101)=1$	0	$f(1001)=1$	0
$f(0010)=1$	$f(0011)=0$	1	$f(0000)=0$	1	$f(0110)=0$	1	$f(1010)=0$	1
$f(0011)=0$	$f(0010)=1$	1	$f(0001)=1$	1	$f(0111)=1$	1	$f(1011)=0$	0
$f(0100)=0$	$f(0101)=1$	1	$f(0110)=0$	0	$f(0000)=0$	0	$f(1100)=0$	0
$f(0101)=1$	$f(0100)=0$	1	$f(0111)=1$	0	$f(0001)=1$	0	$f(1101)=1$	0
$f(0110)=0$	$f(0111)=1$	1	$f(0100)=0$	0	$f(0010)=1$	1	$f(1110)=1$	1
$f(0111)=1$	$f(0110)=0$	1	$f(0101)=1$	0	$f(0011)=0$	1	$f(1111)=0$	1
$f(1000)=1$	$f(1001)=1$	0	$f(1010)=0$	1	$f(1100)=0$	1	$f(0000)=0$	1
$f(1001)=1$	$f(1000)=1$	0	$f(1011)=0$	1	$f(1101)=1$	0	$f(0001)=1$	0
$f(1010)=0$	$f(1011)=0$	0	$f(1000)=1$	1	$f(1110)=1$	1	$f(0010)=1$	1
$f(1011)=0$	$f(1010)=0$	0	$f(1001)=1$	1	$f(1111)=0$	0	$f(0011)=0$	0
$f(1100)=0$	$f(1101)=1$	1	$f(1110)=1$	1	$f(1000)=1$	1	$f(0100)=0$	0
$f(1101)=1$	$f(1100)=0$	1	$f(1111)=0$	1	$f(1001)=1$	0	$f(0101)=1$	0
$f(1110)=1$	$f(1111)=0$	1	$f(1100)=0$	1	$f(1010)=0$	1	$f(0110)=0$	1
$f(1111)=0$	$f(1110)=1$	1	$f(1101)=1$	1	$f(1011)=0$	0	$f(0111)=1$	1
	$\sum D_{0001}f_1(x)$=12		$\sum D_{0010}f_1(x)$=12		$\sum D_{0100}f_1(x)$=8		$\sum D_{1000}f_1(x)$=8	

Енді, (8)-формуланы пайдаланып, осындай есептеулерді біз S_1-блоктың екінші, үшінші және төртінші жолдың компоненттік мәндері:

$f_2(x_1, x_2, x_3, x_4) = \{0\ 1\ 1\ 0\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 1\}$,
$f_3(x_1, x_2, x_3, x_4) = \{0\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 0\ 1\ 0\ 1\ 1\ 1\}$,
$f_4(x_1, x_2, x_3, x_4) = \{0\ 1\ 1\ 1\ 1\ 1\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 1\}$.

үшін жүргіземіз. Соңында, S_1-блоктың барлық жолдарының компоненттері арқылы алынған нәтижелер төмендегідей матрица түрінде өрнектейік:

$$K_s = \begin{pmatrix} \sum D_{0001}f_1(x) & \sum D_{0010}f_1(x) & \sum D_{0100}f_1(x) & \sum D_{1000}f_1(x) \\ \sum D_{0001}f_2(x) & \sum D_{0010}f_2(x) & \sum D_{0100}f_2(x) & \sum D_{1000}f_2(x) \\ \sum D_{0001}f_3(x) & \sum D_{0010}f_3(x) & \sum D_{0100}f_3(x) & \sum D_{1000}f_3(x) \\ \sum D_{0001}f_4(x) & \sum D_{0010}f_4(x) & \sum D_{0100}f_4(x) & \sum D_{1000}f_4(x) \end{pmatrix} = \begin{pmatrix} 12 & 12 & 8 & 8 \\ 8 & 12 & 12 & 8 \\ 12 & 8 & 12 & 12 \\ 8 & 12 & 8 & 12 \end{pmatrix} \quad (3.8)$$

Дәл осындай есептеу жолымен біз қолданған S_2-блок, S_3-блок және S_4-блок үшін төмендегідей нәтижелер аламыз:

$$K_{s2}=\begin{pmatrix} 8 & 12 & 12 & 8 \\ 12 & 8 & 12 & 8 \\ 12 & 8 & 12 & 8 \\ 12 & 12 & 8 & 12 \end{pmatrix}, K_{s3}=\begin{pmatrix} 12 & 8 & 12 & 8 \\ 8 & 12 & 12 & 8 \\ 8 & 12 & 12 & 8 \\ 12 & 8 & 8 & 12 \end{pmatrix}, K_{s4}=\begin{pmatrix} 12 & 12 & 8 & 8 \\ 12 & 8 & 8 & 12 \\ 8 & 12 & 12 & 12 \\ 8 & 12 & 8 & 12 \end{pmatrix}. \quad (3.9)$$

(3.6)-формулаға сүйенсек, алынған мәндер оң нәтиже беру үшін олар $N/2 = 8$ саны маңында болуы тиіс, мұндағы $N = 2^4$. (3.8) мен (3.9)-дан байқайтынымыз, таңдап алынған S-блок алмастырумар қатаң лавиндік эффектіні (SAC) орта есеппен тек 70-75% қанағаттандырады, яғни оларды шифрлау алгоритмінің тиімді примитиві ретінде қолдануға болады. Дегенмен, тәжірибеде SAC-ті 100% қанағаттандыратын кейбір S-блоктар дифференциалдық талдауға төзімсіздік танытып жатады: мысалы келесі қарастырылған S-блок – $S = \{4, 7, 2, 14, 1, 13, 8, 11, 15, 12, 6, 10, 5, 9, 3, 0\}$. Сол себепті, алдағы уақытта қарастырып отырған S-блоктарға дифференциалды және сызықты талдау, олардың векторлық бульдік функциялар арқылы жазбасындағы сызықсыздық дәрежесін, қасиеттерін және коррелациялық, алгебралық, статистикалық талдаулар негізіндегі шабуылдарға төзімділігін анықтау бағытында зерттеулер жүргізу қажеттілігі туындады.

CF шифрлау алгоритмінің қатаң лавиндік әсерін зерттеу. Шифрлау алгоритмінің қатаң лавиндік критерийі (3.2) формуласы арқылы анықталады. Бұл критерий лавиндік критерийге қарағанда талапты жоғары қояды: өзгертілген әрбір кіріс битіне байланысты әрбір шығыс битінің өзгеру қасиетін қарастырады. Теорияда бұл өзгерістің ықтималдығы 0,5-ке жуықтау болуы қажет.

Тәжірибеде талдауымызды мына бағытта жүргіземіз. Алдымен P_0^k ашық мәтінді толық 4 раундпен шифрлаймыз, нәтижені C_0^k деп белгілейік, мұндағы k - ашық мәтіндер нөмірі. Талдау үшін ашық мәтіннің әрбір кіріс *i*-битін инверсиялап, оны P_i^k ретінде қарастырып, шифрлау арқылы соған сәйкес C_i^k шифрмәтінін алып отырамыз, мұнда *i=1,…,128*. Әрбір P_i^k үшін C_i^k шифрмәтіндегі *j*-ші битті бастапқы C_0^k шифрмәтіндегі *j*-ші битімен салыстыратын боламыз, мұндағы *j=1,…,128*. Бізге салыстыру нәтижелеріне талдау жүргізу үшін төмендегідей 128×128 өлшемдегі Q^k матрицасы қажет болады:

$$Q^k = \begin{pmatrix} q_{1,1}^k & q_{1,2}^k & \dots & q_{1,128}^k \\ q_{2,1}^k & q_{2,2}^k & \dots & q_{2,128}^k \\ \dots & \dots & \dots & \dots \\ q_{128,1}^k & q_{128,2}^k & \dots & q_{128,128}^k \end{pmatrix}. \quad (3.10)$$

Мұнда, $q_{i,j}^k$ – P_0^k ашық мәтіннің *i*-битін инверсиялап, шифрлау жүргізгенде алынған C_i^k шифрмәтіннің *j*-ші битінің C_0^k шифрмәтіндегі *j*-ші битімен салыстыратын өзгеруі, яғни

$$q_{i,j}^{k} = \begin{cases} 1, & \text{салыстыруда өзгеріс болса;} \\ 0, & \text{салыстыруда өзгеріс болмаса.} \end{cases} \quad (3.11)$$

Қатаң лавиндік критерийдің орындалуын эмипирикалық түрде тексеру үшін біз әртүрлі екі жүз P_0^k ашық мәтін алдық, $k = 1,2,\dots,200$. Әр k үшін жоғарғы процесті жүргізіп, сәйкесінше екі жүз Q^k алатын боламыз. Алынған екі жүз Q^k матрицасының k бойынша сәйкес элементтерінің қосындысын шығарып, оны төмендегідей белгілейік:

$$R = \begin{pmatrix} \sum_{k=1}^{200} q_{1,1}^{k} & \sum_{k=1}^{200} q_{1,2}^{k} & \dots & \sum_{k=1}^{200} q_{1,128}^{k} \\ \sum_{k=1}^{200} q_{2,1}^{k} & \sum_{k=1}^{200} q_{2,2}^{k} & \dots & \sum_{k=1}^{200} q_{2,128}^{k} \\ \dots & \dots & \dots & \dots \\ \sum_{k=1}^{200} q_{128,1}^{k} & \sum_{k=1}^{200} q_{128,2}^{k} & \dots & \sum_{k=1}^{200} q_{128,128}^{k} \end{pmatrix}. \quad (3.12)$$

Бұдан әрі, $k_{s\,i,j}$ ықтималдығын алу үшін *R* матрицасының әр элементін ашық мәтіндер санына – 200-ге бөлеміз, сонда:

$$K_s = \begin{pmatrix} k_{s\,1,1} & k_{s\,1,2} & \dots & k_{s\,1,128} \\ k_{s\,2,1} & k_{s\,2,2} & \dots & k_{s\,2,128} \\ \dots & \dots & \dots & \dots \\ k_{s\,128,1} & k_{s\,128,2} & \dots & k_{s\,128,128} \end{pmatrix}. \quad (3.13)$$

Алынған $k_{s\,i,j}$ негізінде (3.2) формула арқылы CF шифрлау алгоритмінің қатаң лавиндік критерийін қанағаттандыруын бағалайтын боламыз, мұндағы *i=1,…,128, j=1,…,128*. Бұл есептеулерді жүргізу үшін «Ақпараттық қауіпсіздік» зертханасында арнайы компьютерлік бағдарлама әзірленді. Бағдарлама көмегімен таңдап алынған 200 ашық мәтінге қатаң лавиндік критерийін анықтау мақсатында төмендегідей ықтималдықтар матрицасын алдық:

$$K_s = \begin{pmatrix} 0.56 & 0.50 & 0.51 & 0.54 & 0.53 & 0.46 & \dots & 0.53 \\ 0.52 & 0.50 & 0.50 & 0.49 & 0.49 & 0.47 & \dots & 0.47 \\ 0.49 & 0.41 & 0.42 & 0.47 & 0.53 & 0.51 & \dots & 0.57 \\ 0.47 & \mathbf{0.45} & 0.44 & 0.52 & 0.5 & 0.55 & \dots & 0.51 \\ 0.49 & 0.51 & 0.49 & 0.47 & 0.51 & 0.45 & \dots & 0.53 \\ 0.55 & 0.59 & 0.47 & 0.48 & 0.51 & 0.49 & \dots & 0.53 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.45 & 0.56 & 0.54 & 0.47 & 0.51 & 0.61 & \dots & 0.51 \end{pmatrix}. \quad (3.14)$$

Мысалы, 200 ашық мәтіннің әрбір 4-ші битін инвертациялап шифрлағанда, 200 шифрмәтіннің әрқайсының 2-ші битінің бастапқы инвертацияланбаған нұсқасынан өзгеру ықтималдығы тәжірибе жүзінде 0,45-ке тең болды.

(3.2) формула көмегімен төмендегі Кесте 3.7-де көрсетілген $\varepsilon_{s\,i}$ қатаң лавиндік параметрдің статистикалық көрсеткіштерін алдық.

Кесте 3.7 – $\varepsilon_{s\,i}$ қатаң лавиндік параметрдің статистикалық көрсеткіштері

	$k_{s\,i,j}$ ықтималдығы	$\varepsilon_{s\,i}$-нің мәндері
Максимальды мән	0.6562	0,3125
Минималді мән	0.3437	0
Арифметикалық орта мән	0.5004	0,0710
Дисперсия	0,0018	0,0027
Мода	0.5078	0,0156
Медиана	0.5000	0,0625

CF шифрлау алгоритмінің қатаң лавиндік критерийін талдауы бойынша қорыта келгенде, Кесте 3.7-дегі мәндер теориялық тұрғыдан алғанда оң нәтижелер көрсетеді. Осы нәтижелер көрсеткендей, шифрлаудың кірісіндегі әрбір i-ші биттің өзгерісі шифрмәтіннің j–ші битінің өзгерісін 0,5 ықтималдықтен туындатады. Осы себепті аталған алгоритм қатаң лавиндік критерийін толық қанағаттандырады.

3.4 Алгоритмді «Жақын коллизияларды іздеу» тәсілімен бағалау

Берілген хеш функция үшін бірдей хеш-мәнге сәйкес келетін кез-келген екі кірісті табу есептеу тұрғысынан шешілмейтін болса, h хеш функциясы коллизияға төзімді деп аталады. Коллизиялық шабуылдар бірдей $h(M_1) = h(M_2)$ хеш-мән беретін екі түрлі M_1 және M_2 хабарламаларын табу үшін жүзеге асырылады. Классикалық шабуылда криптоталдаушы түпбейнені табудағы шабуылдағыдай хеш-мәнді мақсатты түрде таңдамайды. Егер бұл хеш функция үшін хеш-мәндері $h(M_1)$ және $h(M_2)$ бір-бірлерінен бірнеше битке ғана айырмашылықта болатын кез-келген екі M_1 және M_2 хабарламасын табу қиын болса, онда қарастырылған хеш функция «жақын коллизияға» төзімді деп айтылады [67].

$M_1 \neq M_2$ болатын M_1 және M_2 хабарламаларының жұбы h хеш функциясы үшін «ϵ-жақындықтағы коллизиялар» деп аталады, егер осы хабарламалар үшін мына теңсіздік орындалса: $d\big(h(M_1), h(M_2)\big) \leq \epsilon$, мұндағы d – хэмминг қашықтығы [68]. Теориялық тұрғыда қарағанда, қауіпсіздігі жоғары хештеу алгоритмінің n бит ұзындықтағы хеш-мәндері жұптарының арасындағы хемминг қашықтығы, яғни сәйкес орындардағы бірдей емес биттер саны $n/2$ санының маңына шоғырлануы керек.

Құрылған HBC-256 хештеу алгоритмінің «ϵ-жақындықтағы коллизия» төзімділігіне тәжірибелік жолмен зерттеулер жүргізейік. Осы алгоритм қауіпсіздігі үшін хеш-мәндер жұптары арасындағы хемминг қашықтығы 128 санының маңына топтасуы шарт. Ол үшін өте үлкен көлемдегі хеш-мәндер жиынын қарастыратын боламыз. Статистика жүргізу үшін кездейсоқ түрде 25 мың хабарламалар алынды. Алгоритм көмегімен қарастырылған хабарламаларға сәйкес 25 мың хеш-мән жасалды. Осы хеш-мәндер жиынынан барлық мүмкін болатын $C_{25000}^2 = 312\,487\,500$ мөлшердегі хеш-мәндер жұбын құрастырамыз. Бұл мақсатта аталған хеш-мәндер жұбының хэмминг қашықтығын есептейтін компьютерлік бағдарлама жасалынды. Сонымен қатар, жұмыс нәтижесінде

мәндері минимальды және максимальды хэмминг қашықтықтары 81 және 175 екендігі анықталды. Шектік мәнді $\epsilon = 20$ деп бекітіп алып (128 ± 20 бит), осы талапты орындайтын өте жақсы хэмминг қашықтығына ие хабарламалар жұбының жалпы санын анықталды:

$(108 \leq d(M_i, M_j) \leq 148) = 309\,283\,762, i, j = 1, \ldots, 25000, i \neq j,$ (≈98,9758%).

Зерттеу нәтижесінде 108 бен 148 арасындағы хэмминг қашықтығына ие хеш-мән жұптарының саны барлық мүмкін жұптардың 99% құрайтыны анықталды. Бұл нәтиже хеш-мәндер «жақын коллизиялар» шабуылына төтеп бере алатынын білдіреді. HBC-256 хештеу алгоритміне қатысты «жақын коллизиялар» қасиетіне ие болу үшін үшін екі хабарлама арасындағы Хэмминг қашықтығы 16 битке дейін аз болуы керек [69]. Осыған сәйкес талдау нәтижелері бойынша HBC-256 алгоритмі «жақын коллизиялар» шабуылға қатысты төзімді.

Кесте 3.8 – Хеш функциялардың «жақын коллизияларын» іздеу нәтижелері

Хэмминг қашықтығы диапазоны	ГОСТ Р 34.11-2012, %	MGR, %	HBC-256, %
128 ± 5	50,39	50,49	49,98
128 ± 10	80,86	81,05	81,11
128 ± 15	93,85	94,92	94,01
128 ± 20	99,02	98,83	98,98

Жоғарғы Кесте 3.8-де Ресей Федерациясының Ұлттық Стандарты ГОСТ Р 34.11-2012 болып бекітіліп, 2013 жылдың 1 қаңтарынан қолданысқа енгізілген мәліметтерді хештеу функциясының (256 немесе 521 биттік хеш-мән есептейтін «Стрибог» алгоритмі) және осы функцияның модификацияланған MGR хеш функциясының «жақын коллизияларын» іздеудегі нәтижелері HBC-256 хеш функциясының нәтижелерімен салыстырылып берілген [70].

3.5 Дифференциалдық криптоталдау әдісімен коллизияның табылуын бағалау

Дифференциалды талдау симметриялық блоктық шифрларды және басқа криптографиялық примитивтерді, атап айтқанда, хеш функцияларды және ағындық шифрларды талдау әдісі болып табылады [71, 72]. Талдаудың бұл түрін қолдану үшін алгоритмнің барлық сызықтық емес элементтері – S-блок ауыстыру, модулі 2 бойынша қосу және басқалары үшін дифференциалдық қасиеттердің кестелерін құру қажет. Сызықты емес элементтерді дифференциалды талдаудың егжей-тегжейлі алгоритмін [73, 74] табуға болады.

Жалпы жағдайда дифференциалды криптоталдау әдісін қолдану келесі қадамдардан тұрады:

1) Сызықты емес элементтерді талдау және олар үшін ең ықтимал айырымдарды анықтау.

2) Қарапайымнан күрделіге, яғни 1 раундтан n раундқа дейін көпраундты сипаттаманы (кіріс айырым – шығыс айырым түрін) тұрғызу. Құрылған сипаттаманың пайда болу ықтималдығын анықтау.

3) Мәтіндердің дұрыс жұптарын, яғни кіріс мәндерінің қосындысы кіріс айырымның мәніне, ал шығыс мәндерінің қосындысы шығыс айырымның мәніне сәйкес келетін мәтіндердің жұптарын іздеу.

Талдаудың бірінші кезеңі – сызықтық емес элементтерді талдау және дифференциалдық қасиеттердің нәтижелері көмегімен кестені құру. HBC-256 алгоритмі үшін мұндай сызықты емес элементтер ретінде Кесте 3.9-де көрсетілген S-блок ауыстырулары болып табылады.

Кесте 3.9 – Төрт алтын S-блок ауыстырулар

S-блоктар	*x - байттар*																Ескерту
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
$S_0(x)$	0	F	B	8	C	9	6	3	D	1	2	4	A	7	5	E	Serpent, S_3
$S_1(x)$	2	E	F	5	C	1	9	A	B	4	6	8	0	7	3	D	HB-1, S_2
$S_2(x)$	7	C	E	9	2	1	5	F	B	6	D	0	4	8	A	3	HB-2, S_0
$S_3(x)$	4	A	1	6	8	F	7	C	3	0	E	D	5	9	B	2	HB-2, S_1

Осы S-блоктардың дифференциалдық қасиеттерін талдау нәтижесіне сәйкес Кесте 3.10 – Кесте 3.13 құрастырылды. Осы кестелерден нөлдік емес айырымдардың максималды ықтималдылығы 1/4 тең екенін көруге болады.

Кесте 3.10 – S_0-блоктың айырымдар кестесі

Ауыстыру кестесінің мәндері																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	**4**	2	0	0	0	0	2	2	2	0	2
2	0	0	0	0	0	2	0	2	0	2	**4**	2	0	0	0	**4**
3	0	0	2	2	**4**	0	0	0	2	2	0	0	0	0	0	**4**
4	0	0	0	0	0	0	**4**	**4**	0	0	2	2	2	2	0	0
5	0	2	0	2	0	0	0	0	2	2	2	2	2	0	2	0
6	0	2	0	2	0	0	2	2	**4**	0	0	0	2	0	0	2
7	0	0	2	**4**	**4**	2	0	0	0	2	0	0	0	0	2	0
8	0	0	0	2	0	0	2	0	0	2	0	0	2	**4**	**4**	0
9	0	2	2	2	0	0	2	0	2	0	2	2	0	0	0	2
A	0	0	2	0	2	0	2	2	0	**4**	0	2	2	0	0	0
B	0	2	2	0	2	2	0	0	0	2	2	0	2	2	0	0
C	0	2	0	0	2	0	2	2	**4**	0	2	0	0	0	2	0
D	0	2	2	0	2	**4**	0	2	0	0	0	0	0	**4**	0	0
E	0	**4**	2	0	0	2	0	0	0	0	0	2	0	2	2	2
F	0	0	2	0	0	0	0	2	2	0	2	2	2	0	**4**	0

Кесте 3.11 – S_1-блоктың айырымдар кестесі

Ауыстыру кестесінің мәндері																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	0	2	0	2	2	**4**	2
2	0	0	0	2	0	2	0	0	0	0	2	**4**	2	**4**	0	0
3	0	2	2	2	2	0	2	2	2	0	0	0	0	2	0	0
4	0	0	0	2	0	**4**	2	0	0	0	0	2	0	0	2	**4**
5	0	0	2	2	2	2	0	0	0	0	0	**4**	**4**	0	0	0
6	0	0	0	2	**4**	0	2	0	2	2	0	2	0	0	0	2
7	0	2	0	0	0	0	2	**4**	**4**	2	0	0	0	0	2	0
8	0	0	0	0	0	0	2	2	0	**4**	**4**	0	2	2	0	0
9	0	2	0	2	2	2	2	2	0	2	0	2	0	0	0	0
A	0	2	0	0	**4**	0	2	0	0	2	0	0	2	2	0	2
B	0	2	2	0	0	0	0	0	2	0	**4**	2	0	0	**4**	0
C	0	0	**4**	0	0	2	0	2	2	2	0	0	2	0	0	2
D	0	2	2	0	0	2	2	0	2	0	2	0	2	0	2	0
E	0	2	**4**	2	0	0	0	0	0	2	2	0	0	0	2	2
F	0	2	0	0	2	2	0	2	2	0	0	0	0	**4**	0	2

Кесте 3.12 – S_2-блоктың айырымдар кестесі

Ауыстыру кестесінің мәндері																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	2	2	2	**4**	0	0
2	0	0	0	0	0	2	**4**	2	0	2	0	2	0	0	**4**	0
3	0	0	**4**	0	2	0	0	2	0	0	0	**4**	0	2	2	0
4	0	0	0	2	0	2	2	2	0	0	0	2	0	2	2	2
5	0	2	2	2	0	0	2	0	0	0	2	0	2	0	**4**	0
6	0	2	2	2	0	2	0	0	**4**	2	0	0	2	0	0	0
7	0	0	0	0	2	2	0	0	**4**	2	0	2	2	0	0	2
8	0	0	0	2	0	0	2	0	0	**4**	2	0	**4**	0	0	2
9	0	2	0	0	2	**4**	2	2	0	0	2	0	0	0	2	0
A	0	2	2	0	0	2	0	2	2	0	2	0	2	0	0	2
B	0	**4**	2	0	0	0	0	2	2	0	0	**4**	0	2	0	0
C	0	0	0	2	**4**	0	0	2	2	2	2	0	0	0	0	2
D	0	0	2	2	2	2	0	0	2	0	2	0	0	2	0	2
E	0	**4**	0	0	0	0	0	0	0	2	2	0	0	2	2	**4**
F	0	0	2	2	**4**	0	**4**	0	0	0	0	0	2	2	0	0

Кесте 3.13 – S_3-блоктың айырымдар кестесі

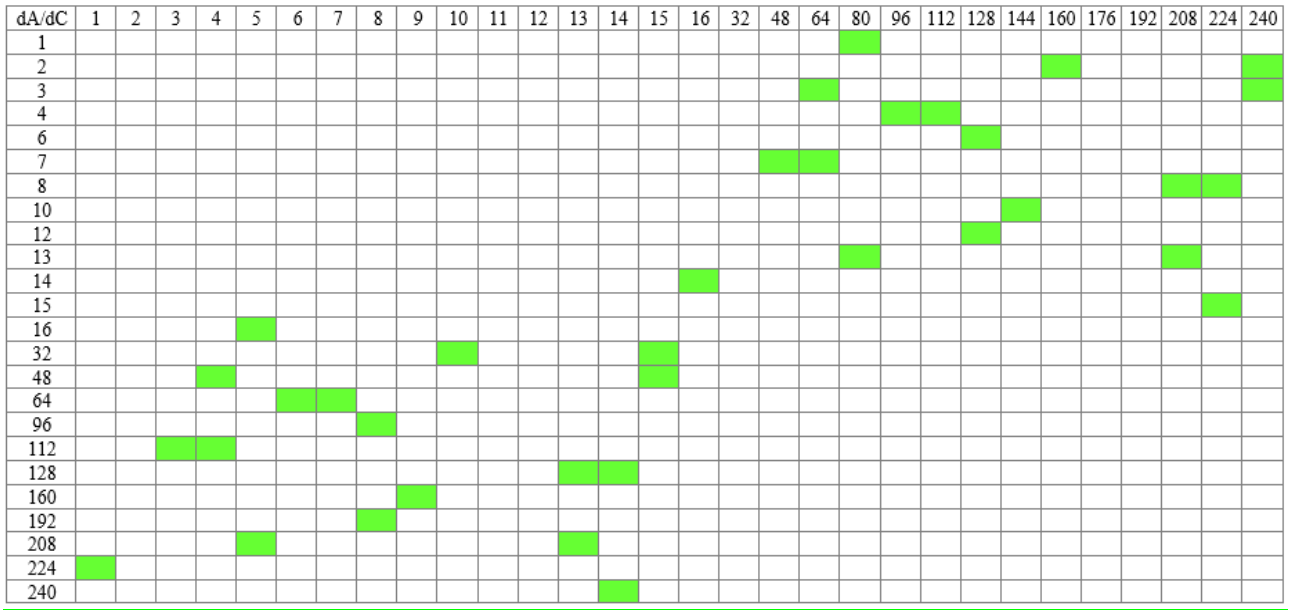
Ауыстыру кестесінің мәндері																
	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**A**	**B**	**C**	**D**	**E**	**F**
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	**4**	0	0	0	**4**	0	2	0	2	2	0	2	0
2	0	0	0	2	0	2	0	0	0	0	0	2	2	**4**	2	2
3	0	0	**4**	0	2	0	0	2	2	0	0	2	0	0	**4**	0
4	0	0	0	0	0	**4**	**4**	0	0	2	2	0	2	0	0	2
5	0	2	2	0	0	2	2	0	0	0	2	2	2	2	0	0
6	0	0	2	2	2	0	2	0	2	**4**	0	2	0	0	0	0
7	0	2	0	0	0	0	0	2	**4**	0	0	2	0	2	**4**	0
8	0	0	0	0	0	0	2	2	0	0	2	2	2	2	2	2
9	0	2	0	0	2	2	0	2	2	2	2	0	2	0	0	0
A	0	0	**4**	2	0	2	2	2	0	0	2	0	0	2	0	0
B	0	2	0	0	**4**	2	0	0	0	**4**	2	0	0	0	2	0
C	0	**4**	0	2	2	0	0	0	0	2	2	2	0	0	0	2
D	0	0	2	2	0	0	0	0	2	0	2	0	2	**4**	0	2
E	0	0	2	0	**4**	0	2	0	2	0	0	0	2	0	0	**4**
F	0	**4**	0	2	0	2	2	2	2	0	0	0	0	0	0	2

Мәліметтерді түрлендіру схемасын, яғни Сурет-2.7-ні мұқият қарастырған кезде, Кесте 3.8-дегі S-блоктар 8x8 разрядты 16 S-блок құрайтынын байқауға болады. Қарастырылатын байтқа байланысты үлкен S-блокты қалыптастыру үшін Кесте 3.8-дегі шағын 4x4 разрядтық (4-биттік) S-блоктардың комбинациялары қолданылады. Осылайша, байттың әрбір күйі (орналасу тәртібіне байланысты) үшін 16 S-блок қарастырылып талданды. Мәселен, мысалы, a_{00} байты үшін S_0-блогін қосарлап қолдану нәтижесінде S_{00}-блогін алуға болады. Нәтижесінде S_{00}-блогі келесідегідей болады:

S_{00} = [0, 240, 176, 128, 192, 144, 96, 48, 208, 16, 32, 64, 160, 112, 80, 224, 15, 255, 191, 143, 207, 159, 111, 63, 223, 31, 47, 79, 175, 127, 95, 239, 11, 251, 187, 139, 203, 155, 107, 59, 219, 27, 43, 75, 171, 123, 91, 235, 8, 248, 184, 136, 200, 152, 104, 56, 216, 24, 40, 72, 168, 120, 88, 232, 12, 252, 188, 140, 204, 156, 108, 60, 220, 28, 44, 76, 172, 124, 92, 236, 9, 249, 185, 137, 201, 153, 105, 57, 217, 25, 41, 73, 169, 121, 89, 233, 6, 246, 182, 134, 198, 150, 102, 54, 214, 22, 38, 70, 166, 118, 86, 230, 3, 243, 179, 131, 195, 147, 99, 51, 211, 19, 35, 67, 163, 115, 83, 227, 13, 253, 189, 141, 205, 157, 109, 61, 221, 29, 45, 77, 173, 125, 93, 237, 1, 241, 177, 129, 193, 145, 97, 49, 209, 17, 33, 65, 161, 113, 81, 225, 2, 242, 178, 130, 194, 146, 98, 50, 210, 18, 34, 66, 162, 114, 82, 226, 4, 244, 180, 132, 196, 148, 100, 52, 212, 20, 36, 68, 164, 116, 84, 228, 10, 250, 186, 138, 202, 154, 106, 58, 218, 26, 42, 74, 170, 122, 90, 234, 7, 247, 183, 135, 199, 151, 103, 55, 215, 23, 39, 71, 167, 119, 87, 231, 5, 245, 181, 133, 197, 149, 101, 53, 213, 21, 37, 69, 165, 117, 85, 229, 14, 254, 190, 142, 206, 158, 110, 62, 222, 30, 46, 78, 174, 126, 94, 238].

Осы тәртіппен басқа он бес 8x8 разрядты S-блоктар жасалынады. Барлық S-блоктар дифференциалдық қасиеттерге талданды. Әрбір S-блок үшін 1/4 ықтималдығымен түрленетін біраз айырымдар бар екені көрсетілді. Бұл ерекшелікті S_0,…,S_3 блоктарын талдау нәтижелерінен-ақ байқауға болады (Кесте 3.9 – Кесте-3.12). Бұл ситуация тек мына жағдайда орын алады: бірінші S-блоктың кірісіне 0000 айырымы (0000 шығыс айырымына 1-ге тең ықтималдықтен түрленетін) түскен жағдайда, екінші S-блоктың кірісіне пайда

болу ықтималдығы ¼ болатын қандай да бір айырым түсетін болса. Дегенмен, мұндай алдын-ала есептеулер хеш функциясының құрылымын талдау уақытын айтарлықтай қысқартады және байттық айырымдардағы өзгерістерді жақсы түсінуге мүмкіндік береді. S_{00}-блогы бойынша жүргізілген талдау нәтижелері бар кесте үзіндісі төменгі Сурет 3.5-те көрсетілген. Бағандар арқылы кейбір кіріс айырымдар, ал жолдар арқылы кейбір шығыс айырымдар белгіленген. Қарастырылатын кіріс айырмасына сәйкес келетін шығыс айырмасының пайда болу ықтималдығы 1/4-ге тең жағдайлар үшін баған мен жолдың қиылысындағы ұяшықтар ерекше бояумен көрсетілген.



Сурет 3.5 – S_{00}-блогі үшін жүргізілген талдау нәтижелері кестесінің үзіндісі

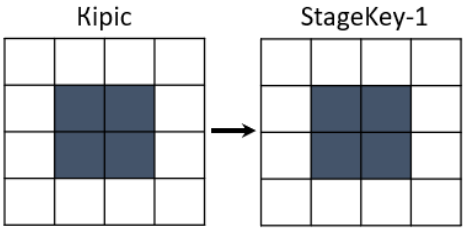
Хештеу функциясының коллизияларын табу үшін дифференциалды криптоталдау жұмысы келесідей жүргізіледі: мәтіндік айырымдардың түрлендіруін кіріс айырымы нөлдік емес мәнге ие, ал шығыс айырымы нөлге тең болатындай етіп құру қажет. Айта кету керек, мұндай жұп мәтінді табу ықтималдығы «дөрекі күш» әдісін қолдану арқылы коллизияны табу ықтималдығынан аз болуы керек.

HBC-256 хештеу алгоритмінің маңызды ерекшелігі раундтық кілттерді генерациялау алгоритмінің кірісіне және CF қысу функцияның өзінің кірісіне (ашық мәтін ретінде) бірдей мәннің алынуы болып табылады. Дегенмен, бұл екі кіріс мәндерді өңдеу біршама ерекшеленеді, бұл ерекшелік талдау жүргізуді күрделендіреді.

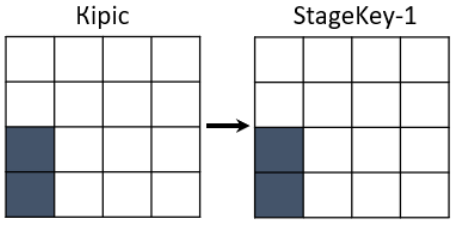
Біз раундтық кілттерді генерациялау функциясын талдаудан бастаймыз. Функцияда StageKey-1, StageKey-2, StageKey-3 түрлендірулері бар. StageKey-2-де Stage-2 түрлендіруіндегідей Xor-сыз: нақтырақ – тек 1 биттік солға циклдық жылжыту операциясы ғана орындайтынын ескерген жөн. Үш түрлендіру де бірінен соң бірі тізбектеліп 8 рет орындалады. Осыдан кейін бастапқы (кіріс) матрица алынған (шығыс) матрицаға Xor операциясы арқылы қосылады.

Кіріс матрицасы StageKey-1 түрлендіруіне кіреді. Түрлендіру 00 индексі бар элементтен басталады. Таңдалған элементі бар жол мен бағандағы барлық элементтер қосылады, таңдалған элемент де есепке алынады. Осыдан кейін жаңа есептелген элемент индекстеріне байланысты екі S-блокқа сәйкес жаңа мәнге ауыстырылады және матрицада осы орынға қайта жазылады. Процесті дәл осылай 01 индексті элементке және тағы сол сияқты жалғастыра отырып, 33 индексті элементке дейін жүргіземіз. Күрделілік тудыратын бір жайт, матрицаның әр элементі орналасу орнына байланысты жеке өзіне ғана тән тәртіппен жұптасқан екі S-блоктан өтетін болады.

Қажетті мәтін жұптарын іздеу идеясы мынандай: түрлендірулерден толық өткеннен кейін матрица элементтерінің ең аз санын пайдаланатын мәтіндердің айырымдарын табу. Жұмыс матрицаның қолайлы элементтермен толтыруын іздеуден басталды. Ол үшін көптеген әртүрлі комбинациялар нұсқалары терілді. Мысал ретінде қажетті мәтін жұптарының бір нұсқасы Сурет 3.6-те көрсетілген. Осы суретте және алдағы суреттерде квадраттың (матрица ретінде қарастырылған) ақ ұяшықтары осы орындарда нөлдік айырымдардың бар екендігін, ал қара ұяшықтар осы орындарда айырымдардың нөлдік мән еместігін көрсетеді. Бұдан арғы жүргізілген талдаулар айырымдарды түрлендірудің одан да қолайлы нұсқасы бар екенін анықтады (Сурет 3.7).

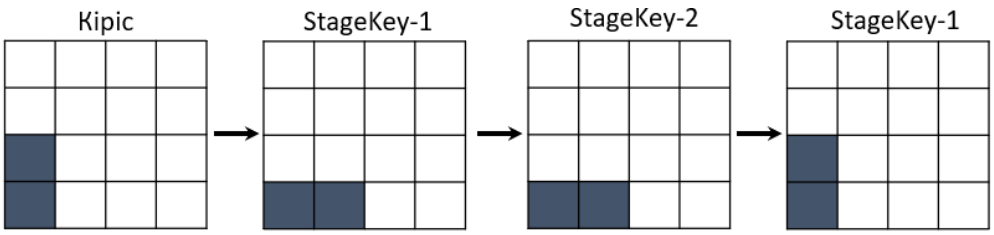


Сурет 3.6 – Матрицаның қажетті мәндермен толтырылу нұсқасы



Сурет 3.7 – Матрицаның қажетті мәндермен толтырылуының қолайлы нұсқасы

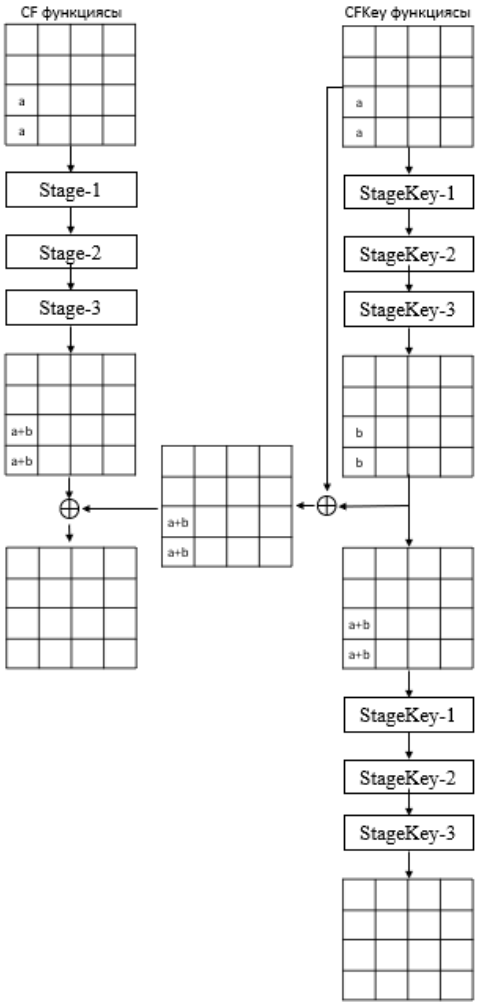
Сурет 3.4-те мәндерді түрлендіруді әрі қарай қарастырған кезде, тіпті StageKey-2 және Stage-2 и StageKey-3 түрлендірулерінен кейін де әр жолы тек үш S-блоктар ғана есептеулерге қатысатыны анықталды, олар – S_{20}, S_{30}, S_{31}. Бұл түрлендірулер Сурет 3.5-те көрсетілген.



Сурет 3.8 – Негізгі үш операциядан өткендегі айырымның түрленулері

Табылған түрлендіру нәтижесіне сүйенсек (Сурет 3.5), мынандай гипотеза қоюға болады: егер 1-раунд түрлендірулерінен соң алынған айырым 1-раундтық кілттердің айырымдарына тең келетін болса, сондай-ақ 2-раундтық кілт түрлендірулер нәтижесінде 0-дік айырымға тең болса, онда бұдан әрі қарайғы түрлендірулер коллизия туындататын болады (Сурет 3.9).

Гипотеза. Хештеудің бір раундынан өткенде матрицаның (2,0) және (3,0) орындардағы мәндер *(a+b)* айырымдық мәндерді қабылдайтын жағдай тудыратын *a* айырымы бар болады. Сонымен қатар, *a* айырымы раундтық кілттерді жасаудың бір раундынан өткенде матрицаның нақ осы (2,0) және (3,0) орындарындағы мәндеріне де *(a+b)* айырымын береді. Сондай-ақ, (2,0) және (3,0) орындарындағы *(a+b)* айырымы раундтық кілттерді жасаудың тағы бір раундынан өткенде өзіне-өзі түрленуі мүмкін.



Сурет 3.9 – HBC-256 хештеу функциясындағы айырымдарды түрлендіру схемасы

Сурет 3.8-ке сәйкес S_{20}, $S_{30,}$ S_{31} блоктары кіріс айырымдардың бірдей мәндерін шығыс айырымдардың бірдей мәндеріне түрлендіруі тиіс. Басқаша айтқанда, кіріске мәндері *a* тең екі айырым кірсе, онда шығыста (2,0) және (3,0) орындарында *b* немесе *a+b* мәніне тең болатын екі айырым шығуы керек (осы орындарда әртүрлі алмастыру блоктары қолданылған жағдайдың өзінде де). Stage-2 кезеңінде айырымдардың мәндері әртүрлі болуы мүмкін. Хештеу функциясындағы Stage-2 раундтық кілт жасау операциясындағы StageKey-2-дан бөлек екенін ескерген жөн. Енді $S_{30,}$ S_{31} блоктарындағы кірістері бірдей мәндер шығысында бірдей мәндер беретін тізбектерді іздеу жүргізіледі. Бұл Stage-1 түрлендіруінде орындалуы тиіс. Бұдан соң олардың мәндері 2-ге көбейтілуі тиіс (себебі ол солға қарай бір битке жылжытумен пара-пар), бұл бізге Stage-2 түрлендіруінің орындалуын береді. Әрі қарай, бұл мәндер $S_{20,}$ S_{30} блоктары бойынша алғашқы мәнге оралуы керек. Егер осының барлығын символдық жазбаға келтірейік. Кіріске A мәні кіретін болсын. Ол мән $S_{30,}$ S_{31} блоктары арқылы түрленіп, C мәнге ие болады. Бұдан әрі ол 2 еселеніп, 2C болады. Әрі қарай ол $S_{20,}$ S_{30} ауыстыру кестелері арқылы B мәніне ие болады.

Осылайша, біз айырымдарды түрлендіру үшін келесі көрсеткіштерге қол жеткізгіміз келеді (Кесте 3.14):

Кесте 3.14 – CF үшін айырымдарды түрлендіру сұлбасы

	Хештеу функциясы	Кілттерді жасау функциясы
1-раунд	$a \to a + b$	$a \to b$
2-раунд	$0 \to 0$	$a + b \to a + b$

Алдымен, Stage-1-ге (2,0) орында нөлге айналатын мән кіретінін ескерген жөн. Содан кейін (3,0) орындағы мән S-блок бойынша қандайда бір мәнге өзгереді. Stage-2 түрлендірілгеннен кейін көрші блоктар өзгермеуі үшін шығыс айырымдарының мәндері 128 мәнінен артық болмауы керек.

Жүргізілген талдау нәтижесінде кілтті түрлендіруге арналған гипотезаға сәйкес тізбектер алынды (Кесте 3.15). Барлық мәндер ондық түрінде жазылған бір байттық айырымдар болып табылады, жазба мына схемаға сәйкес келеді: кіріс айырым → Stage1-ден кейінгі айырым → Stage2-ден кейінгі айырым → Stage3-тен кейінгі айырым, ал жақша ішінде есептеуге қатысатын екі S-блок бойынша айырымды анықтауға арналған көрсеткіштер көрсетілген (256 жағдайдың ішіндегі пайда болулар саны). Сол жақ бағанында 1-раундтағы түрлендірулер, оң жақ бағанда – 2-раундтағы түрлендірулер нәтижелері).

Кесте 3.15 – CFKey-де айырымдарды түрлендіру үшін табылған тізбектер

Кіріс айырым → Stage1-ден кейінгі *a* айырымы → StageKey2-ден кейінгі айырым → Stage3-тен кейінгі *b* айырымы → (Кіріс айырым) *xor* (Stage3-тен кейінгі *a+b* айырымы)	*(a+b)* кіріс айырымы→ Stage1-ден кейінгі айырым → StageKey2-ден кейінгі айырым → Stage3-тен кейінгі *(a+b)* айырымы
1	2
27 (8|8) → 19 → 38 → 245(4|4) → 238	238 (4|4) → 92 → 184 (8|4) → 238
27 (8|8) → 19 → 38 → 53 (4|4) → 46	46 (8|4) → 93 → 186 (4|8) → 46
27 (8|8) → 23 → 46 → 245 (4|4) → 238	238 (4|4) → 92 → 184 (8|4) → 238
27 (4|4) → 43 → 86 → 140 (8|8) → 151	151 (8|8) → 69 → 138 (8|4) → 151
	151 (8|4) → 73 → 146 (4|4) → 151
	151 (4|4) → 85 → 170 (8|4) → 151
27 (4|4) → 43 → 86 → 60 (4|4) → 39	39 (8|8) → 69 → 138 (4|4) → 39
	39 (8|4) → 79 → 158 (4|4) → 39
	39 (4|4) → 85 → 170 (4|4) → 39
27 (4|8) → 46 → 92 → 140 (8|8) →151	151 (8|8) → 69 → 138 (8|4) → 151
	151 (8|4) → 73 → 146 (4|4) →151
	151 (4|4) → 85 → 170 (8|4) → 151
28 (8|8) → 19 → 38 → 59 (4|4) → 39	39 (8|8) → 69 → 138 (4|4) → 39
	39 (8|4) → 79 → 158 (4|4) → 39
	39 (4|4) → 85 → 170 (4|4) → 39
28 (8|8) → 23 → 46 → 190 (4|8) → 162	162 (8|4) → 82 → 164 (8|8) → 162
29 (8|8) → 19 → 38 → 197 (4|4) → 216	216 (4|8) → 50 → 100 (4|8) → 216
29 (8|8) → 23 → 46 → 213 (4|4) → 200	200 (4|4) → 52 → 104 (4|4) → 200

Кесте 3.15 жалғасы

1	2
29 (4|4) → 43 → 86 → 54 (4|4) → 43	43 (8|8) → 45 → 90 (4|4) → 43 43 (4|8) → 69 → 138 (4|4) → 43
39 (8|8) → 69 → 138 → 108 (4|8) → 75	75 (8|8) → 69 → 138 (4|4) → 75 75 (8|16) → 70 → 140 (4|4) → 75
75 (8|16) → 70 → 140 → 143 (8|8) → 196	196 (8|4) → 99 → 198 (4|8) → 196 196 (8|4) → 100 → 200 (4|8) → 196
77 (8|8) → 69 → 138 → 102 (4|4) → 43	43 (8|8) → 45 → 90 (4|4) → 43 43 (4|8) → 69 →138 (4|4) → 43
	43 (8|8) → 45 → 90 (4|4) → 43 43 (4|8) → 69 → 138 (4|4) → 43
77 (8|16) → 70 → 140 → 239 (4|4) → 162	162 (8|4) → 82 → 164 (8|8) → 162
77 (8|16) → 70 → 140 → 31 (4|4) → 82	82 (4|4) → 113 → 226 (4|4) → 82
77 (16|4) → 85 → 170 → 149 (8|8) → 216	216 (4|8) → 50 → 100 (4|8) → 216
97 (4|4) → 99 → 198 → 195 (4|4) → 162	162 (8|4) → 82 → 164 (8|8) → 162
97 (4|4) → 100 → 200 → 147 (4|4) → 242	242 (8|4) → 113 → 226 (8|8) → 242 242 (4|8) → 119 → 238 (4|8) → 242
108 (4|16) → 70 → 140 → 166 (4|4) → 202	202 (4|4) → 35 → 70 (4|4) → 202 202 (4|4) → 43 → 86 (4|4) → 202 202 (4|8) → 74 → 148 (4|4) → 202 202 (4|4) → 99 → 198 (4|4) → 202 202 (4|4) → 100 → 200 (4|4) → 202
154 (4|4) → 73 → 146 → 85 (4|8) → 207	207 (4|4) → 35 → 70 (4|4) → 207
154 (4|8) → 105 → 210 → 82 (4|4) →200	200 (4|4) → 52 → 104 (4|4) → 200
154 (4|8) → 105 → 210 → 189 (8|4) → 39	39 (8|8) → 69 → 138 (4|4) → 39
154 (4|8) → 105 → 210 → 189 (8|4) → 39	39 (8|4) → 79 → 158 (4|4) → 39 39 (4|4) → 85 → 170 (4|4) → 39
202 (4|4) → 43 → 86 → 18 (4|4) → 216	216 (4|8) → 50 → 100 (4|8) → 216
202 (4|8) → 74 → 148 → 215 (4|4) → 29	29 (8|8) → 19 → 38 (8|8) → 29 29 (8|8) → 23 → 46 (16|8) → 29 29 (4|4) → 43 → 86 (4|8) → 29 29 (4|8) →46 → 92 (4|8) → 29
221 (8|4) → 82 → 164 → 181 (4|4) → 104	104 (4|8) → 50 → 100 (8|8) → 104 104 (4|4) → 52 → 104 (4|8) → 104

Дәл осыдай жолмен хештеу функциясы үшін де кірістері бірдей мәндер шығысында бірдей мәндер беретін тізбектерді табамыз (Кесте 3.16). Сонымен қатар, кіріс айырымдар ретінде Кесте 3.14-тегі кіріс айырымдарды ғана қолданамыз және шығысында (a+b) айырымына сәйкес келетін айырымдар пайда болатындай тізбектерді іздейміз. Кесте 3.16-дегі барлық мәндер ондық түрінде жазылған бір байттық айырымдар болып табылады, жазба мына схемаға сәйкес келеді: кіріс айырым → Stage1-ден кейінгі айырым → Stage2-ден кейінгі айырым → Stage3-тен кейінгі айырым.

Кесте 3.16 – CF-те айырымдарды түрлендіру үшін табылған тізбектер

Кіріс айырым → Stage1-ден кейінгі айырым → Stage2-ден кейінгі айырым → Stage3-тен кейінгі айырым	Кіріс айырым → Stage1-ден кейінгі айырым → Stage2-ден кейінгі айырым → Stage3-тен кейінгі айырым
27 (8|8) → 19 → 53 (8|4) → 238	77 (4|4) → 73 → 219 (4|4) → 162
27 (4|4) → 25 → 43 (4|8) → 46	77 (4|4) → 25 → 43 (4|4) → 43
27 (4|4) → 27 → 45 (4|8) → 46	77 (4|4) → 73 → 219 (4|4) → 82
27 (8|8) → 19 → 53 (8|4) → 238	77 (4|4) → 73 → 219 (4|4) → 216
27 (8|8) → 19 → 53 (4|4) → 151	97 (4|4) → 52 → 92 (4|4) → 162
27 (8|8) → 23 → 57 (4|4) → 39	97 (4|8) → 50 → 86 (4|4) → 241
27 (8|8) → 19 → 53 (4|4) → 151	108 (4|8) → 120 → 136 (4|4) → 202
28 (8|8) → 119 → 153 (4|4) → 39	154 (4|4) → 73 → 219 (4|4) → 207
28 (4|4) → 73 → 219 (4|4) → 162	154 (4|8) → 120 → 136 (4|4) → 207
29 (4|4) → 43 → 125 (16|16) → 216	154 (4|4) → 73 → 219 (4|4) → 200
29 (4|4) → 73 → 219 (4|4) → 200	154 (4|4) → 113 → 147 (4|4) → 39
29 (4|4) → 25 → 43 (4|4) → 43	202 (4|4) → 73 → 219 (4|4) → 216
29 (4|4) → 27 → 45 (4|4) → 43	202 (4|4) → 79 → 209 (4|4) → 216
29 (4|4) → 43 → 125 (4|4) → 43	202 (4|4) → 73 → 219 (8|4) → 29
39 (4|4) → 43 → 125 (4|4) → 75	221 (4|8) → 44 → 116 (16|16) → 104
39 (8|8) → 45 → 119 (8|8) → 75	221 (4|4) → 79 → 209 (4|4) → 104
75 (8|16) → 70 → 202 (4|8) → 196	

Кесте 3.15 және Кесте 3.16-тен айырымның бір байтының түрлену ықтималдығы $\frac{1}{2^6}$ нен $\frac{1}{2^4}$-ке дейін өзгеретінін көруге болады. Бірраундты түрлендірудің қорытынды ықтималдығын анықтау үшін S-блок алмастырулар арқылы нөлдік емес айырымдарды түрлендірулер санын есептеу керек. Кілтті түрлендірудің бір раунды үшін төрт нөлдік емес S-блок қатысады. Бір кілтті жасауда 8 раунд жүргізілетіндіктен, 1-раундтық кілтті жасау үшін 32 нөлдік емес S-блок арқылы нөлдік емес байттарды түрлендіру қажет болады. Нәтижесінде бірінші раундтық кілтті жасау кезінде айырымды түрлендіру ықтималдығы $\frac{1}{2^{192}}$ - нен $\frac{1}{2^{128}}$-ге дейін болуы мүмкін. 2-раундтық кілтті жасау ықтималдығы 1-раундтық кілтті жасаудағы ықтималдықпен бірдей болады. Бір раундта да қысу функциясы үшін бір раундты түрлендіру төрт нөлдік емес S-блок қатысады және оның ықтималдығы $\frac{1}{2^6}$ мен $\frac{1}{2^4}$ аралығында болады. Осылайша, алынған ықтималдықтарды Сурет 3.6-ға сәйкес біріктіре отырып, қойылған гипотеза нәтижесінде түрлендіру ықтималдығы ең нашар жағдайда $\frac{1}{2^{390}}$, ал ең жақсы жағдайда $\frac{1}{2^{250}}$ ықтималдықта болады. Нәтижесінде, хештеу функциясының кілті 8 раундта жасалатын конструкциясында HBC-256 хештеу функциясының коллизияларын табу үшін дифференциалды криптоталдау әдісін қолдану негізсіз.

HBC-256 хештеу алгоритмінің дифференциалды талдауының нәтижелері S-блоктарды талдауға негізделген (Кестелер 3.10 – 3.13, Суреттер 3.2 және 3.6). Алынған нәтижелерді бағалау нәтижесінде Кесте 3.15 және Кесте 3.16-дегі мәліметтермен расталатын айырымдарды түрлендірудің (мәтіндерді түрлендіру

және кілт жасау үшін) жұптасқан тізбектерін құру мүмкіндігі туралы гипотеза жасалды.

Зерттеу жүргізудегі ең үлкен шектеу – бұл хештеу алгоритмдері үшін толықөлшемді кірістер мен шығыстарды пайдаланудың күрделілігі, өйткені бұл жағдайда талдау жүргізу есептеу ресурсы мен көп уақытты қажет етеді және күрделі болады. Мұндай мәселені шешудің бір нұсқасы – модельдеуге және нәтижеге жуықтауға мүмкіндік беретін кішірейтілген модельдерді немесе қысқартылған функцияларды қолдану.

Қазіргі уақытта жүргізілген зерттеулер нәтижесі толықраундтық хештеу алгоритмінің осалдығын анықтаған жоқ.

3.6 Алгебралық криптоталдау әдісі арқылы коллизияның табылуын бағалау

Алгебралық талдау әдісі криптографиялық алгоритмдердің көптеген түрлеріне (симметриялы блоктық және ағындық шифрларға, хештеу алгоритмдеріне) қолданылатын әмбебап әдіс болып табылады [75]. Алгебралық шабуылдар құпия кілтті немесе хабарламаны қалпына келтіру мақсатында сызықтық емес теңдеулер жүйесін шешуге негізделген [76, 77]. Хеш функцияларына криптоталдау жүргізуге бағытталған алгебралық шабуылдар, негізінен, коллизияларды анықтауға және нашар қысу функциясы қолданылған жағдайда түпбейнені қалпына келтіру мақсатында жүргізіледі. Алгебралық шабуылдардың негізгі идеясы – хабарламаны, шифрланған мәтінді және құпия кілтті қамтитын сызықтық емес теңдеулерді шешу. Алгебралық шабуыл екі кезеңнен тұрады:

1-кезең. Төменгі дәрежелі сызықтық емес теңдеулердің немесе құрылымдалған (көп өлшемді) сызықтық емес теңдеулерді жеткілікті дәрежеде құру;

2-кезең. Құрылған теңдеулер жүйесін шешу арқылы кілтті есептеу.

Зерттелетін криптографиялық алгоритм үшін бірінші кезең бір рет қана орындалды. Теңдеулерді шешудің ең көп қолданылатын әдістеріне линеаризациялау (сызықтандыру) алгоритмдері [78-80], Гребнер базисі [81] және SAT-шешушілері [82-84] жатады. Линеаризациялау алгоритмі алынған сызықтық емес теңдеулер жүйесіндегі сызықтық емес мүшелерді жаңа айнымалылармен алмастырады, яғни әрбір сызықтық емес моном жаңа айнымалымен ауыстырылады. Нәтижесінде жаңа сызықтық жүйе алынады және оны Гаусс әдісі арқылы шешуге болады. Көп өлшемді алгебралық теңдеулер жүйесін шешудің тағы бір класы Гребнер базистеріне негізделген. Сондай-ақ, егер талданатын хеш алгоритмін сипаттайтын теңдеулер саны тым көп болмаса, іс жүзінде алынған сызықтық емес теңдеулер жүйесін шешу үшін автоматтандырылған құралдарды қолдануға болады, мысалы, төмендегідей SAT шешушілері: Crypto|MiniSat [85], Lingeling [86], Cadical [87] және т.б.

Алгебралық талдау кез-келген шифрлау процесін алгебралық түрлендірулер түрінде ұсынуға және шығыс биттерінің кіріс биттеріне айқын тәуелділігін математикалық түрде сипаттауға болатындығын жорамалдайды. Мұндай жүйені құру процесінің өзі, көбінесе логикалық (бульдік) теңдеулерді құру өте қиын

және көп уақытты алады. Талдаудың бұл түрі статистикалық талдауға жатпайды, яғни берілген жүйені шешу үшін тек бір «ашық мәтін-шифрмәтін» жұбының бар болуы қажет. Құрылған жүйе логикалық (бульдік) болғандықтан, оның құрамындағы айнымалылар екі мән – «0» және «1»-ді қабылдайды, сондықтан жүйені бірнеше логикалық базистер (|,&, ¬), (&, ¬),(|, ¬), (⊕,&) арқылы өрнектеуге болады. Осы соңғы үш нұсқа өрнекті сәйкесінше ДҚФ, ККФ және Жегалкин көпмүшесі түрінде жазуға мүмкіндік береді. Осындай алгебралық сипаттаманы жасағаннан кейін құрылған теңдеулер жүйесін шешу керек, оны SAT шешушілерінің бірінің көмегімен жасауға болады және осының нәтижесі жүйенің берілген шарттарда шешімі бар немесе жоқ екендігін көрсетеді.

Алгебралық талдау жүргізу үшін логикалық теңдеулер жүйесін құру қажет, ол үшін Transalg [88] құралы қолданылады. Бұл бағдарламалық жасақтама шифрлау және хештеу алгоритмін теңдеулер жүйесіне түрлендіреді және ККФ форматында және &, ¬ базисінде жаза алады, сондай-ақ символдық постфикстік көріністегі кіріс биттеріне тәуелділік түрінде жазуға мүмкіндік береді.

Хештеу алгоритмін жүзеге асыратын бағдарлама коды теңдеулер құруға арналған бағдарлама кодына түрлендірілді. Нәтижесінде қысу функциясының бір раунды 82533 теңдеулер мен 16609 айнымалыларды қолдана отырып, ККФ түрінде өрнектелді. Мысал ретінде, бірнеше ККФ түріндегі теңдеулер төменде келтірілген:

$x_{176} = x_{168}x_{102}$
$x_{177} = x_{169}x_{103}$
$x_{178} = x_{170}x_{104}$
$x_{179} = x_{178}x_{177} \oplus x_{178}x_{177}x_{176} \oplus x_{176}x_{178} \oplus x_{176}x_{177}$
$x_{180} = x_{178}x_{177} \oplus x_{176} \oplus x_{176}x_{177}x_{178}x_{175} \oplus x_{175}$
$x_{181} = x_{178}x_{177} \oplus x_{176}x_{178} \oplus x_{175}x_{178} \oplus x_{175}x_{177}x_{178}$
$x_{182} = x_{178}x_{177} \oplus x_{176}x_{177} \oplus x_{175} \oplus x_{175}x_{178} \oplus x_{175}x_{176}$
$x_{183} = x_{174}x_{173} \oplus x_{173}x_{174}x_{172} \oplus x_{172}x_{174} \oplus x_{172}x_{173}$
$x_{184} = x_{174}x_{173} \oplus x_{172} \oplus x_{172}x_{173} \oplus x_{174}x_{171} \oplus x_{171}$
$x_{185} = x_{174}x_{173} \oplus x_{172}x_{174} \oplus x_{171}x_{174} \oplus x_{171}x_{173}x_{174}$
$x_{186} = x_{174}x_{173} \oplus x_{172}x_{173}x_{171} \oplus x_{171}x_{174} \oplus x_{171}x_{172}$
$x_{187} = x_{9}x_{179}$
$x_{188} = x_{10}x_{180}$
$x_{189} = x_{11}x_{181}$
$x_{190} = x_{12}x_{182}$
$x_{191} = x_{13}x_{183}$
$x_{192} = x_{14}x_{184}$
$x_{193} = x_{14}x_{184}.$

Жүйені ішінара өрнектеу және оны шешу үшін SAT шешушісін қолдану қажет. Осы мақсатта SAT lingeling шешушісінің топтамасы таңдалды, оның ішінде есептеулерді жүргізуде параллельдеу мүмкіндігі бар plingeling нұсқасы және кубтық treengeling нұсқасы қарастырылды. Теңдеулер жүйесін шешу процесін қысу функциясы жұмысының бір раунды үшін жүргізейік.

```
c 1  S 1321200.1   15832    81756  60400771 2432040   1940    87   58   131     43     0  524  49   987   5010
c 2  S 1321200.0   10088    78408  26500005 2907539   1735     0   86   123     97    -1  458  48   990   5901
c
c    seconds         irredundant           redundant clauses   glue   iterations"      MB    stability
c            variables clauses conflicts  large ternary binary     jlevel       jlevel'  agility     tlevel
c
c 0  S 1321200.4   10076    78435  55252397 2875160   1795     0   59   143     47   -15  488  50   984   3245
c 5  S 1321200.6   10060    79816  57215567 2911408   1735     0   61   131     45    -2  475  49   988   3020
c 3  S 1321200.7   10080    78413  55188498 2876906   1805     0   63   134     47    -6  466  50   989   5203
c 4  S 1321201.0   10071    78512  90044163 3174600   1827     0   46   177     29   -11  508  49   974   2135
c 1  S 1324800.4   15832    81756  60509350 2430984   1940    87   59   127     43    -1  524  49   990   4429
c 0  S 1324800.2   10076    78435  55326336 2972343   1795     0   64   154     47   -23  545  49   987   5562
c 4  S 1324800.1   10071    78512  90214095 3169680   1830     0   46   168     29     3  548  49   976   2750
c 5  S 1324800.4   10060    79816  57281178 2852573   1735     0   59   164     45    -7  457  50   987   5510
c 2  S 1324800.4   10088    78408  26544507 2938074   1735     0   85   121     97     0  481  50   990   5989
c 3  S 1324800.8   10080    78413  55349526 2937044   1805     0   48   145     47   -14  503  52   987   4699
c 1  S 1328400.1   15832    81756  60596516 2534618   1945    89   62   139     43   -12  590  50   988   8871
c 5  S 1328400.2   10060    79816  57322403 2916374   1735     0   53   137     45    -8  501  48   984   5397
c 4  S 1328400.4   10071    78512  90354947 3146126   1830     0   46   148     29    -9  526  49   973   3105
c 2  S 1328400.6   10088    78408  26587538 2924946   1735     0   85   123     97    -0  469  48   990   6023
c 0  S 1328400.9   10076    78435  55490715 2867430   1795     0   50   190     47    26  467  50   981   2151
c 3  S 1328400.6   10080    78413  55419123 2904285   1805     0   57   138     47    -9  490  50   988   4795
c 0  S 1332000.2   10076    78435  55579671 2975017   1795     0   54   203     46    24  541  49   981   2403
c 1  S 1332000.4   15832    81756  60689363 2523433   1947    90   65   133     43    -1  577  50   989   5703
c 3  S 1332000.1   10080    78413  55508296 2896525   1805     0   57   151     46   -17  473  51   984   5655
c 5  S 1332000.4   10060    79816  57407588 2890312   1735     0   61   134     45    -3  464  49   985   5061
c 4  S 1332000.5   10071    78512  90475820 3113113   1830     0   46   159     29    18  511  49   969   2067
c 2  S 1332000.8   10088    78408  26630505 2913884   1735     0   84   121     97     0  456  50   990   5753
```

Сурет 3.10 – lingeling нұсқасында түпбейнені есептеудегі жұмыс фрагменті

Жұмысты тексеру үшін lingeling нұсқасына бастапқы теңдеулер жүйесі енгізіліп және ол шығыс айнымалыларының мәндерімен шектелумен іске қосылды, яғни lingeling нұсқасы шығыс айнымалылардың мәндері (түпбейнені қалпына келтіру) белгілі деп бекітіліп алып, кіріс айнымалыларды табу үшін жүргізілді. Есептеу деректерін іске қосу шығыс деректердің сынақтық күйіне тең болған жағдайда жүргізілді. Теңдеудің шешімі алдын-ала белгілі деп қарастырдық, яғни негізгі жұмыс – осы шешімге қосымша жанама шешімдердің (коллизиялардың) бар-жоғын тексердік, сондай-ақ белгілі кіріс мәнін есептеу жылдамдығына бағалау (түпбейнені қалпына келтіруді тексеру) жүргіздік. Бір процессорлы ЭЕМ қолдана отырып (ешқандай қосымша опцияларсыз), бұл есепті шешуге 241000 сек = 67 сағат қажет болды және қосымша шешімді таппады (Сурет 3.10).

Шешімді табуды жеделдету үшін кіріс айнымалыларының бір бөлігіне өрнектеулер жүргізіліп, сынақтық мысалда шешімді табу жылдамдығы есептелді. Ішінара өрнектелген мәндер үшін есептеу уақыты туралы мәліметтер келесі Кесте 3.17 пен Кесте 3.18-да келтірілген.

Кесте 3.17 – Бір процессорлік ядросы бар ЭЕМ-да есептеу жылдамдығы (сек.)

SAT шешушісінің топтамасы	Белгісіз биттердің саны			
	0	8	16	24
plingeling	-	15,8	1851	белгісіз
treengeling	0,11	19,54	105,41	207987,68

Кесте 3.18 – Алты процессорлік ядросы бар ЭЕМ-да есептеу жылдамдығы (сек.)

SAT шешушісінің топтамасы	Белгісіз биттердің саны				
	0	8	16	24	32
plingeling	-	0,1	2,9	27807,5	39331,7

Осылайша, құрылған теңдеулер жүйесі белгілі бір ықтималдықпен қысу функциясының бір раунды үшін түпбейнені қалпына келтіруге мүмкіндік береді. Әрі қарайғы жұмыс хештеу функциясының толық раундтық процесін сипаттайтын теңдеулер жүйесін құруға бағытталуы керек. Сондай-ақ, шешімдерді іздеу алгоритмін бірінші типтегі коллизияны іздеу үшін қайта қарастыруға болады.

HBC-256 хештеу алгоритмінің бір раунды үшін хештеу функциясын сипаттайтын логикалық теңдеулер жүйесі құрылды. Құрылған теңдеулер жүйесін шешу үшін есептеуді параллельдеу мүмкіндігі бар plingeling нұсқасын қамтитын lingeling SAT-шешуші пайдаланылды. Қысу функциясының бір раунды 82533 теңдеулер мен 16609 айнымалыларды қолдана отырып, КҚФ жазбасы түрінде сипатталды. Алты процессорлы есептеу ядросын қолдана отырып, хештеудің бір раундындағы 32 биті белгісіз түпбейнесін табу үшін шамамен 11 сағаттық уақыт қажет болды. Айта кететіні, логикалық теңдеулер жүйесін шешуші plingeling нұсқасы өзінің алгоритмдерінде шешімді іздеу процесінде кездейсоқтықтықты (рандомизацияны) пайдаланады. Сондықтан бес эксперименттің біреуі ғана алғашқы толықтай табудың сәтті шешімімен аяқталды. Сонымен қатар, раундтар қосылған сайын теңдеулер мен айнымалылар саны екі есеге артып, шешімді табу уақыты экспоненциалды түрде өсетіні түсінікті. Осылайша, қазіргі уақытта HBC-256 хештеу функциясының толықраундтық алгоритмі үшін шешімдерін табу мүмкін емес [89].

3.7 Сызықтық криптоталдау әдісі негізінде талдау жүргізу

Сызықтық криптоталдау криптографиялық примитивтерді талдаудың маңызды әдістерінің бірі болып табылады. Сызықтық криптоталдау – ашық мәтін, шифрмәтін және кілт арасындағы сызықтық жуықтауды талдауға бағытталған. Егер шифрлау алгоритмі сызықтық криптоталдау үшін кездейсоқ орын ауыстырудан басқаша әрекет ететін болса, онда оны кілтті қалпына келтіру үшін шабуыл ұйымдастыруға пайдалануға болады [90]. Сызықтық криптоталдаудың негізгі идеясы сызықтық жуықтауды іздеу болып табылады, ол ашық мәтін биттері мен шифрмәтін биттерінің жиындарының арасындағы сызықтық жуықтау байланысын зерттейді, яғни кейбір ашық мәтіннің биттері мен шифрмәтіннің және кілттің биттері арасында қандай сызықтық байланыс бар екенін анықтайды [91, 92].

$$A[a_1, a_2, \dots, a_n] \oplus C[c_1, c_2, \dots, c_m] = K[k_1, k_2, \dots, k_l] \qquad (3.15)$$

мұндағы $a_1, a_2, \dots, a_n, c_1, c_2, \dots, c_m$ және $k_1, k_2, \dots, k_l$ белгілеулері биттердің бекітілген орындарын білдіреді, ал (3.15) теңдеу кездейсоқ берілген ашық мәтін

A, оған сәйкес шифрмәтін C және кілтт K үшін p ≠ ½ ықтималдықпен орындалады.

Кілтпен xor немесе биттердің орын ауыстыруы сияқты қарапайым сызықтық операциялар үшін 1-ге тең ықтималдылықпен орындалатын өте қарапайым сызықтық теңдеулерді жазуға болады. Ал, S-блок сияқты сызықты емес түйіндер үшін p ықтималдығымен орындалатын сызықтық жуықтауларды табу керек болады. Бұл ретте, сәтті талдау жүргізу үшін теңдеулердің p ықтималдығы мүмкіндігінше 0,5 мәнінен алшақ жатуы керек.

Алдымен шифр ішінде жеке операциялар үшін жуықтау ізделеді, содан кейін олар шифрдың бір раунды үшін ақиқат жуықтаулармен біріктіріледі. Бір раундтық жуықтауларды сәйкес конкатенациялау арқылы шабуылдаушы ақыр соңында толықраундтық шифр үшін жуықтауды алады [93].

Шабуылдың күрделілігін анықтау үшін сызықтық сипаттаманың ықтималдығын бағалау қажет. Бір раундтағы сызықтық жуықтауды кілт биттеріне байланысты 0 немесе 1 мәнін қабылдайтын $\alpha_1 X_1 \oplus \alpha_2 X_2 \oplus \ldots \oplus \alpha_n X_n \oplus \beta_1 Y_1 \oplus \beta_2 Y_2 \oplus \ldots \oplus \beta_m Y_m$ түрдегі кездейсоқ шама ретінде қарастыруға болады. Содан кейін осы кездейсоқ шамалардың сызықтық сипаттамасы мен сызықтық сипаттаманың ықтималдығы таңбаларлың жүгірісі туралы лемманы (лемма о набегании знаков) пайдаланып есептелуі мүмкін (Лемма 1). Екі кездейсоқ шама X_1 және X_2 қарастырылсын. Осыдан $i \in \{1, 2\}$ үшін $P(X_i = 0) = p_i$ и $P(X_i = 1) = 1 - p_i$ болсын. Бұдан X_1 мен X_2 тәуелсіздігінен $P(X_1 = 0, X_2 = 0) = p_1 p_2$ және $P(X_1 = 1, X_2 = 1) = (1 - p_1)(1 - p_2)$ екендігі шығады. Осылайша, $P(X_1 \oplus X_2 = 0) = p_1 p_2 + (1 - p_1)(1 - p_2)$ [94].

Лемма 1. $X_i \in \mathbb{Z}_2$ – тәуелсіз кездейсоқ шамалар және олардың мәндерінің 0-ге тең болу ықтималдығы $P(X_i = 0) = \frac{1}{2} + \varepsilon_i$ болсын, мұндағы $0 \le \varepsilon_i \le 1/2, 1 \le i \le n$. Онда $X_1 \oplus X_2 \oplus \ldots \oplus X_n = 0$ болуы ықтималдығы мынаған тең болады: $P(X_1 \oplus X_2 \oplus \ldots \oplus X_n = 0) = 1/2 + 2^{n-1} \prod_{i=1}^{n} \varepsilon_i$.

Лемма 2. N – берілген ашық мәтіндер саны болып, (3.15) теңдеуінің орындалу ықтималдығы p болсын, сондай-ақ $\left|p - {}^{1}/_{2}\right|$ өте аз мәнге ие болсын. Онда алгоритмнің сәтті болу ықтималдығы $\int_{-2\sqrt{N}|p-\frac{1}{2}|}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx$ тең.

CF шифрлау алгоритмінде жалғыз сызықты емес түйін – бұл S-блок ауыстыруы. Біз берілген 4-биттік төрт S-блок үшін сызықтық жуықтау кестесін (ағылш. LAT) құрамыз. Кесте құру барысында кіріс пен шығыс екілік векторлардың барлық комбинациялары қадағаланып отырады. Әрбір екілік векторлар жұбы орынтірек ретінде қолданады, бұл орынтіректер S-блоктың барлық мүмкін болатын кіріс-шығыс жұбына қолданылады және келесі қатынаспен анықталады:

$$LAT(\alpha, \beta) \stackrel{\text{def}}{=} \left\{ X \mid X \in Z_2^8, \bigoplus_{i=1}^{8} X[i] \cdot \alpha[i] = \bigoplus_{i=1}^{8} S(X[i]) \cdot \beta[i] \right\},$$

мұндағы $\alpha, \beta \in Z_{256}$ және көбейту амалы скалярлық көбейту операциясын білдіреді [95-97].

Кесте 3.19 – S_0-блоктың сызықтық жуықтау кестесі

		Шығыс орынтірек														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Кіріс орынтірек	1	10	8	6	8	10	4	10	8	10	8	6	8	10	12	10
	2	6	10	8	8	10	10	4	6	8	8	10	10	8	12	10
	3	12	10	10	8	8	6	10	6	6	8	12	10	6	8	8
	4	8	10	10	10	10	8	8	8	12	10	6	10	6	8	4
	5	6	6	8	10	8	8	10	8	10	6	12	10	12	8	6
	6	6	8	6	6	8	10	12	10	8	10	8	12	6	8	10
	7	8	12	8	6	10	10	10	10	10	6	10	4	8	8	8
	8	8	8	12	10	6	10	10	6	10	6	6	8	8	8	12
	9	10	8	10	6	4	10	8	10	8	10	8	8	10	12	6
	10	10	10	8	6	8	8	6	8	10	10	8	10	12	4	10
	11	8	10	10	10	10	8	8	12	4	6	6	10	10	8	8
	12	8	6	10	8	12	10	10	6	6	12	8	6	10	8	8
	13	6	10	8	12	6	6	8	10	8	12	10	6	8	8	10
	14	10	4	10	8	10	8	6	12	10	8	10	8	6	8	10
	15	12	8	4	12	8	12	8	8	8	8	8	8	8	8	8

Кесте 3.20 – S_1-блоктың сызықтық жуықтау кестесі

		Шығыс орынтірек														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Кіріс орынтірек	1	8	6	10	10	6	8	8	8	12	6	6	10	10	8	12
	2	10	8	10	8	6	8	6	10	8	10	8	10	12	10	4
	3	10	10	8	10	12	4	10	6	8	8	6	8	10	10	8
	4	10	6	8	6	8	8	10	8	10	10	4	6	8	4	6
	5	6	4	6	8	10	8	6	8	10	8	6	8	6	12	6
	6	8	6	6	10	6	4	8	6	10	8	12	8	8	6	6
	7	4	8	8	12	8	8	8	10	6	6	6	6	10	6	6
	8	8	8	4	8	8	8	4	6	6	10	6	10	10	6	10
	9	8	10	10	6	10	8	4	6	10	4	8	8	8	6	6
	10	10	8	6	8	6	8	10	8	6	4	6	12	6	8	6
	11	10	6	8	6	8	4	6	12	6	6	8	6	8	8	10
	12	6	6	8	6	12	8	10	10	8	8	10	12	10	6	8
	13	10	8	10	12	10	8	6	10	8	10	8	10	4	6	8
	14	12	6	6	10	10	12	8	8	8	6	10	6	10	8	8
	15	8	12	4	8	8	8	8	12	12	8	8	8	8	8	8

Кесте 3.21 – S_2-блоктың сызықтық жуықтау кестесі

		Шығыс орынтірек														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Кіріс орынтірек	1	8	6	6	6	10	4	8	8	8	10	10	6	10	8	4
	2	10	8	10	8	6	8	6	10	8	6	4	6	8	10	4
	3	6	6	12	10	8	8	6	10	12	8	10	8	10	6	8
	4	8	8	8	6	6	10	10	6	10	6	10	8	12	12	8
	5	4	6	10	8	8	10	10	6	6	8	8	10	6	8	4
	6	6	4	10	6	8	6	8	8	6	8	6	6	8	10	12
	7	6	10	8	12	10	10	8	8	6	10	8	4	10	10	8
	8	6	8	6	6	8	10	4	8	6	8	6	10	12	6	8
	9	10	6	8	8	10	10	12	12	6	6	8	8	10	6	8
	10	8	8	8	10	10	6	6	10	6	6	10	12	8	12	8
	11	8	6	6	8	12	10	6	6	10	4	8	6	6	8	8
	12	10	8	10	8	6	8	6	6	4	6	12	6	8	6	8
	13	10	6	8	6	8	12	6	10	8	12	10	8	6	10	8
	14	8	12	12	4	12	8	8	8	8	8	8	8	8	8	8
	15	4	10	6	6	6	8	8	12	8	6	10	6	6	8	8

Кесте 3.22 – S_3 блоктың сызықтық жуықтау кестесі

		Шығыс орынтірек														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Кіріс орынтірек	1	6	8	10	8	6	8	10	10	4	10	8	6	8	6	4
	2	8	10	10	10	10	8	8	8	8	10	10	6	6	4	12
	3	10	10	8	6	8	4	10	10	8	4	6	8	6	6	8
	4	10	8	6	8	6	8	10	10	8	10	12	10	4	10	8
	5	8	8	8	8	4	8	4	8	8	8	8	12	8	4	8
	6	10	6	12	10	8	4	6	10	8	8	10	8	10	10	8
	7	4	6	10	6	6	8	8	8	12	6	10	6	6	8	8
	8	10	8	10	6	8	10	4	8	6	8	6	6	4	10	8
	9	8	12	8	10	10	10	6	10	10	6	10	8	8	8	4
	10	10	6	8	8	10	6	8	4	10	10	8	8	6	6	4
	11	12	10	10	8	4	10	10	6	10	8	8	6	10	8	8
	12	8	8	4	10	6	6	6	10	10	10	6	4	8	8	8
	13	6	12	10	6	8	6	8	8	10	12	6	10	8	10	8
	14	8	10	6	4	8	6	6	6	6	8	12	6	10	8	8
	15	10	6	8	4	10	10	8	12	10	10	8	8	10	6	8

Сызықтық жуықтау кестесінде (Кесте 3.19-3.22) екінші бағанда кіріс орынтіректер, ал екінші жолда шығыс орынтіректер жазылған. Егер 4 биттік сызықтық теңдеу 0 рет орындалса, онда осы S-блок үшін осы 4 биттік сызықтық қатынас жоқ деп тұжырымдалады. Егер 4 биттік сызықтық теңдеу 16 рет орындалса, онда осы нақты 4 биттік S-блокқа қатысты осы 4 биттік сызықтық қатынас бар деп есептеледі. Екі жағдайда да толық мәлімет криптоталдаушыға жіберіледі. 4 биттік сызықтық қатынастың бар немесе жоқ болуының ықтималдығы 1/2-ден әлдеқайда қашық болуы, сондай-ақ 0-ге немесе 1-ге жақын болуы криптоталдаушы үшін қолайлы деп саналады. Егер барлық 4 биттік

сызықтық қатынастардың бар немесе жоқ болуы ықтималдығы 1/2 немесе осыған жақын болса, онда 4 биттік S-блок үшін сызықтық криптоталдау жүргізу қиындық тудырады. Сондықтан, жоғарғы кестелерде сегіз саны аз болған сайын, криптоталдаушы үшін шабуыл жасау оңайға түседі. Егер кестеде сегіз санының жалпы саны басқа сандардан әлдеқайда көп болса, онда 4 биттік S-блок сызықты криптоталдауға берік деп саналады [98, 99]. Жоғарыдағы сызықты жуықтау кестелерінде 8 санынан алшақ жатқан 12 және 4 сандары болды. Сондықтан, кестелердегі сандарға сүйене отырып, талдауды одан әрі жалғастыруға мүмкіндік беретін тиімді сызықтық теңдеулерді 3/4 ықтималдықпен ақиқат деп құруға болады. Бұл теңдеулер жүйесін Қосымша Д-ден көруге болады.

Лавиндік әсер талдауының нәтижесіне сүйенсек, шығыс биттер барлық кіріс биттерге 1-раундтан кейін-ақ тәуелді болады. Талдау жүргізу үшін айнымалылары ең аз теңдеуді қарастырайық. Талдау нәтижесінде раунд соңында модуль 2 бойынша қосылатын раундтық кілттерді анықтау үшін жоғарыдағы тиімді теңдеулер (орындалу ықтималдығы 3/4) айнымалылары неше рет S-блоктан өтетіндігін есептеу керек. CF шифрлау алгоритмі сұлбасына сүйеніп, шығыс байттар 2 өлшемді массивте орналасу орнына байланысты төмендегідей мөлшердегі белсенді S-блок қатысатынын есептелінді. Ескере кететіні, сұлба бойынша бір раунд ішінде Stage-1 және Stage-3 түрлендірулері әсерінен белсенді S-блоктар саны N_S өте үлкен мәнге ие болады:

$$N_S = \begin{pmatrix} 27046 & 9470 & 3101 & 952 \\ 9470 & 3512 & 1320 & 484 \\ 3101 & 1320 & 560 & 266 \\ 952 & 484 & 266 & 196 \end{pmatrix}.$$

Алгоритмдегі Stage-2 түрлендіруі арқылы кіріс мәндерге қосуды сызықты криптоталдауда жеке қарастырмаса да болады. Өйткені, Stage-2 түрлендіруі – циклдық жылжытудан ғана тұратын сызықты функция.

Криптоталдаушыға ең жақсы жағдай ретінде белсенді S-блок саны ең аз мөлшерде қатысатын жағдайды (массивтегі [3,3] позициясындағы 196 мәнін) бағалайық. Жоғарыдағы лемма 1 бойынша CF шифрлау алгоритмі сұлбасына сәйкес толық бір раунд нәтижесінде тиімді теңдеудердің орындалу ықтималдығы былай бағаланады. Леммадағы ықтималдықтың 0,5 мәнінен қаншалықты ауытқу болатынын есептейік: $\varepsilon = 2^{196-1} \cdot \left(\frac{1}{2} - \frac{3}{4}\right)^{196} = 2^{-197}$. Яғни, 4 биттік сызықтық қатынастардың бар болуы ықтималдығы 0,5 мәніне өте жақын болып, криптоталдаушы үшін сызықтық криптоталдау жүргізу тіпті 1-раунд үшін де қиындық тудыратыны анықталды. Ал, лемма 2 сүйеніп, раундтық кілттер k_i анықтау үшін төмендегі мөлшердегі кіріс және шығыс мәтіндер жұбы қажет болады: $N = \left(\frac{1}{0{,}5+2^{-197}-0{,}5}\right)^2 = 2^{394}$.

Қорыта айтқанда, CF шифрлау алгоритмінің 1 раундына сызықтық криптоталдау арқылы тиімді шабуыл жасау үшін 2^{394} жұп ашық/жабық мәтін қажет. Талдау барысында келесі 3 раундын және HBC-256 хештеу алгоритміндегі бөліктер арасындағы PerF прецедурасын ескергенде, сызықтық талдауды аталған хештеу алгоритміне пайдалану тиімсіз екені анықталды.

4 ҚҰРЫЛҒАН АЛГОРИТМДІ ЖҮЗЕГЕ АСЫРУ ҮШІН БАҒДАРЛАМАЛЫҚ ЖӘНЕ БАҒДАРЛАМАЛЫ-АППАРАТТЫҚ ЖАСАҚТАМАЛАР ҚҰРУ

4.1 HBC-256 алгоритмін бағдарламалық жүзеге асыру

Құрылған HBC-256 хештеу алгоритмін бағдарламалық қамтамасыз ету екі тәуелсіз бағдарлама түрінде жүзеге асырылды:

1) «ISL_HASH 1.0» деректерді хештеу бағдарламасы.

2) «CSP_HASH 1.0» деректерді хештеу бағдарламасы.

Екі бағдарлама да хеш-мәндерді салыстыру негізінде еркін ұзындықтағы деректердің хеш-мәнді алуға және файлдардың тұтастығын тексеруге арналған. Екі бағдарламаның айырмашылығы мынада: «ISL_HASH 1.0» бағдарламасы HBC-256 хештеу алгоритмін тікелей жүзеге асырады, ал «CSP_HASH 1.0» бағдарламасы алдын-ала құрылған «ISL_CSP 1.0» криптопровайдеріне жүгінетін CryptoAPI 1.0 функцияларын шақыру арқылы жұмыс істейді.

«ISL_HASH 1.0» және «CSP_HASH 1.0» деректерді хештеу бағдарламалары хеш-мәндерді алуды жүзеге асыратын төменде көрсетілген негізгі функциялардың орындалуын қамтамасыз етеді:

- деректердің хеш-мәндерін алу функциясы;
- файлдардың хеш-мәндерін салыстыру функциясы.

Кіріс дерек ретінде сыртқы тасымалдағышта сақталған кез-келген файлдың немесе экрандық форма арқылы енгізілген мәтіннің мазмұны алынады. Шығыс деректер экрандық формада көрсетіледі немесе «*.hash» форматындағы файлда сақталады.

«ISL_HASH 1.0» және «CSP_HASH 1.0» деректерді хештеу бағдарламалары пайдаланушымен өзара әрекеттесуді жүзеге асыратын төменде көрсетілген қосалқы функцияларды орындау мүмкіндігін қамтамасыз етеді:

- диалогтық режимде файлдарды таңдау функциясы;
- диалогтық режимде файлдарды сақтау функциясы;
- мәтінді кесу функциясы;
- мәтінді көшіру функциясы;
- мәтінді кірістіру функциясы;
- мәтінді тазарту функциясы;
- хештелетін деректер түрін таңдау функциясы (файл немесе мәтін);
- бағдарлама жасаушысының сайтын көрсету функциясы;
- бағдарламаның нұсқасын көрсету функциясы;
- параметрлерді автоматты түрде енгізу функциясы;
- бағдарлама параметрлерін сақтау функциясы.

Бұл бағдарламалардың әрқайсысы үш функционалды модульден тұрады:

- хеш-мәнді қалыптастыру модулі;
- хеш-мәндерді салыстыру модулі;
- графикалық интерфейс модулі.

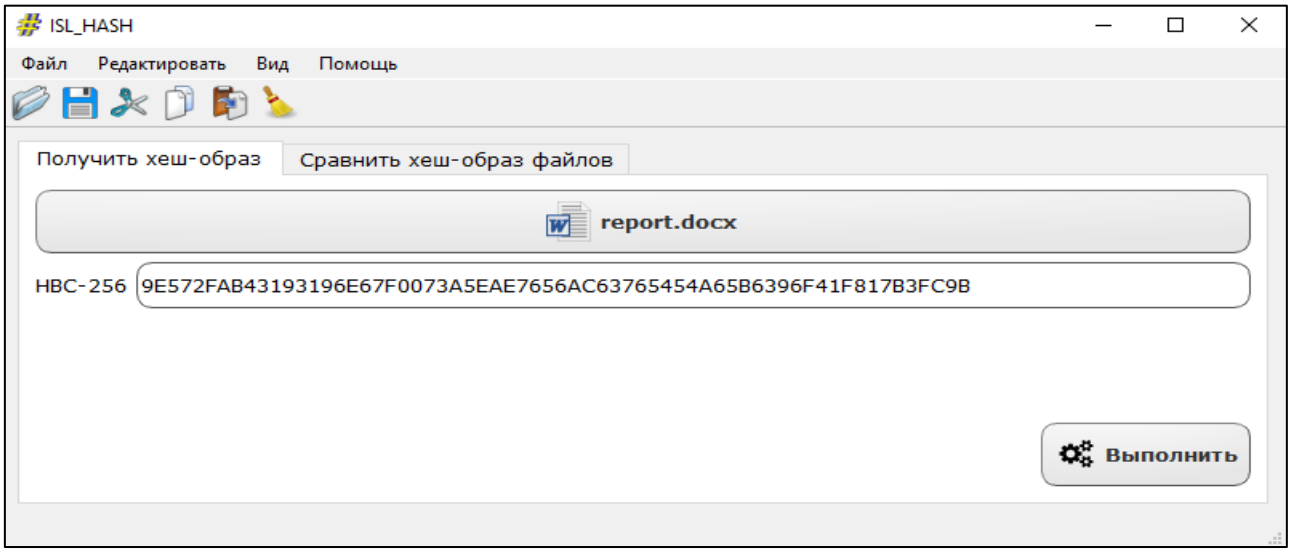
Хеш-мәнді қалыптастыру модулі деректерден ұзындығы 256 бит (HBC-256 болатын хеш-мән алуға арналған.

Хеш-мәндерді салыстыру модулі берілген файлдың жаңа хеш-мәнінқалыптастыру және оны «*.hash» форматында бұрын сақталған файлдағы мәндермен салыстыру арқылы файлдардың тұтастығын тексеруге арналған.

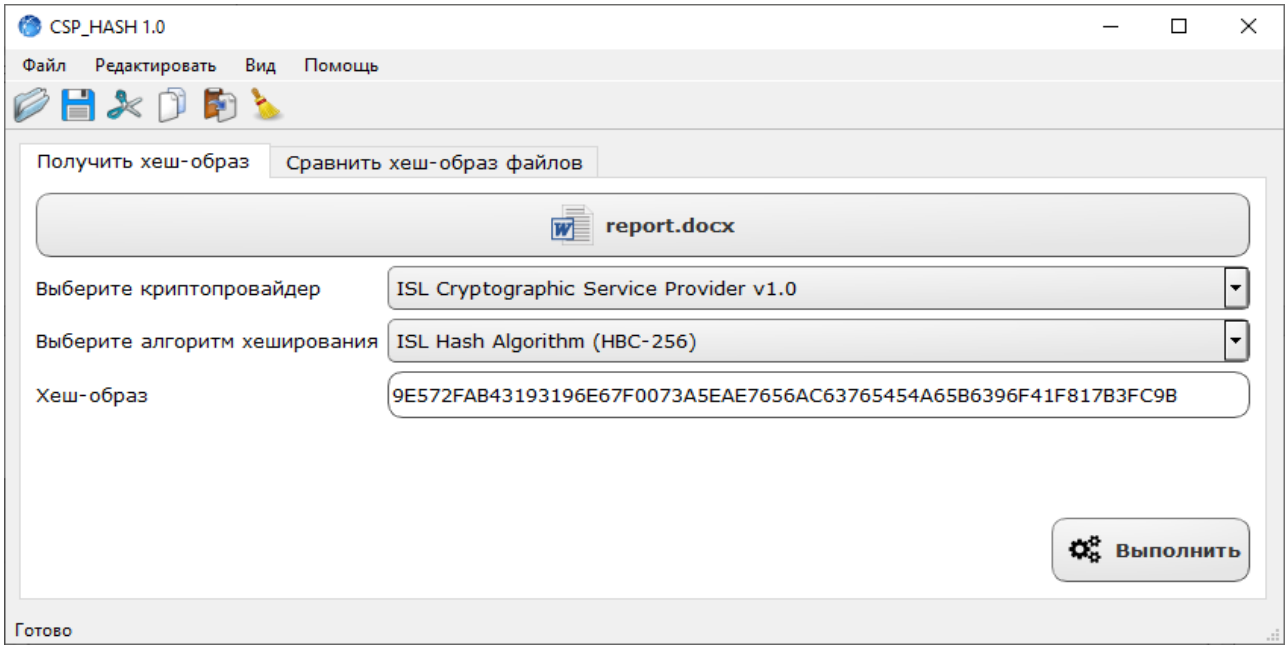
Графикалық интерфейс модулі пайдаланушының интерактивті режимде жоғарыда сипатталған функционалды модульдермен өзара әрекеттесуін қамтамасыз етеді.

Оператордың бағдарламамен өзара әрекеттесуі графикалық пайдаланушы интерфейсін қолдана отырып, интерактивті режимде жүзеге асырылады.

«ISL_HASH 1.0» бағдарламасының негізгі терезесі 4.1-суретте, ал «CSP_HASH 1.0» бағдарламасының негізгі терезесі 4.2-суретте көрсетілген.



Сурет 4.1 – «ISL_HASH 1.0» бағдарламасының негізгі терезесі



Сурет 4.2 – «CSP_HASH 1.0» бағдарламасының негізгі терезесі

«ISL_HASH 1.0» және «CSP_HASH 1.0» деректерді хештеу бағдарламалары Qt Creator 4.15.2 интеграциялық әзірлеу ортасы арқылы Qt 5.15.2 (Open Source Edition) кітапханасын пайдалана отырып, C++ бағдарламалау тілінде жүзеге асырылды.

«ISL_CSP 1.0» криптопровайдері Microsoft Windows операциялық жүйесін қолдайтын CryptoSPI (Cryptographic Service Provider Interface) бағдарламалық интерфейсінің талаптарына сәйкес HBC-256 алгоритмін жүзеге асырады. Қосымшалар криптопровайдерлермен тікелей әрекеттеспейді. Оның орнына қосымшалар Advapi32.dll және Crypt32.dll жүйелік кітапханалары ұсынатын CryptoAPI 1.0 функцияларын шақырып пайдаланады. «ISL_CSP 1.0» криптопровайдері Microsoft CryptoSPI бағдарламалық интерфейсінің келесі функцияларын жүзеге асырады:

– CPCreateHash() функциясы хеш-нысанды жасайды және деректер ағынын хештеуді бастайды;

– CPDestroyHash() функциясы бұрын жасалған хеш-нысанды жояды;

– CPDuplicateHash() функциясы хеш-нысанның мен оның күйінің көшірмесін жасайды;

– CPGetHashParam() функциясы ағымдағы хеш-нысан параметрлерін алады. Дәл осы функцияның көмегімен нақты хеш-мәнді алуға болады;

– CPHashData() функциясы хеш-нысанына хештеу үшін деректерді жібереді;

– CPSetHashParam() функциясы хеш-нысан параметрлерін анықтауға мүмкіндік береді.

«ISL_CSP 1.0» криптопровайдері Microsoft Visual Studio 2015 (Community Edition) интеграциялық әзірлеу ортасы арқылы C++ бағдарламалау тілінде жүзеге асырылды және динамикалық түрде қосылатын кітапхана (DLL) болып табылады.

«ISL_HASH 1.0», «CSP_HASH 1.0» және «ISL_CSP 1.0» криптопровайдер деректерін хештеу бағдарламаларына авторлық құқықпен қорғалатын объектілерге құқықтардың мемлекеттік тізіліміне мәліметтерді енгізу туралы куәлік алынған (авторлық куәліктер 2021 жылғы 5 қазандағы № 20661, 2022 жылғы 24 ақпандағы № 23886 және 2022 жылғы 12 қазандағы № 29379).

4.2 HBC-256 алгоритмін бағдарламалы-аппараттық жүзеге асыру

Аппараттық платформаны таңдау. Хештеу алгоритмін аппараттық жағынан іске асыру үшін MYIR Z-turn түзету тақшасы (отладочная плата) таңдалды. Бұл тақша бір кристалды Xilinx Zynq XC7Z010 жүйесімен (бұдан әрі SoC, немесе өнім), жоғарғы жылдамдықты USB OTG интерфейсті микросхемасымен, 1 ГБ жедел жадымен және 16 МБ NAND Flash микросхемасымен жабдықталған (Сурет 4.3).

SoC құрамына мыналар кіреді:

– Artix-7 архитектурасындағы бағдарламаланатын логикалық интегралды схемасы (Программируемая логическая интегральная схема, бұдан әрі - ПЛИС);

– Cortex A9 ядролы микропроцессоры.

Аппараттық іске асыру технологияларын таңдау. Cortex процессорына арналған бағдарламалық код ассамблерлік кірістірулерді қолдана отырып, Си бағдарламалау тілінде орындалды.

ПЛИС дизайны VHDL технологиялық белгілеу тілін қолдана отырып жасалған.

Өнімнің жұмыс принципі. Cortex процессоры компьютермен өзара әрекеттесу функцияларын жүзеге асыруға, USB интерфейсін қолдауға және HBC-256 алгоритмін аппараттық іске асыруға негізделген ПЛИС-ті басқаруға арналған.

Сурет 4.3 – Xilinx Zynq XC7Z020 жүйесі орнатылған MYIR Z-turn түзету тақшасы

USB интерфейсі арқылы компьютермен деректер алмасу және қуат алу жүзеге асырылады. Инициализациялау кезінде өнім компьютерге ақпарат сақтау режимінде (Mass Storage Device, MSD) жалғанады. Деректерді сақтау үшін ақпарат жинақтаушы ретінде жедел жады қолданылады, онда осы мақсатта 512 МБ аумақ бөлінген. Cortex процессоры FAT файлдық жүйесіне сәйкес осы жад аймағын жаңа файлдардің бар-жоғын үздіксіз қадағалап отырады. Операциялық жүйенің көмегімен көшірілген жаңа файл пайда болғаннан кейін, процессор жадқа тікелей қол жеткізу технологиясын (direct Memory Access, DMA) қолдана отырып, ішкі AXI шинасы арқылы ПЛИС-ке деректер блоктарын жібереді. ПЛИС кезекті хабарламалар блогын ала отырып, оған HBC-256 алгоритмінің сипаттамасына сәйкес түрлендіру жүргізеді. Соңғы блокты түрлендіруді аяқталғаннан кейін орталық процессор ПЛИС-тен хештеу алгоритмін жұмысы нәтижесін оқиды және деректерді сақтауға арналған аймақта бастапқы файлдың атына сәйкес келетін, «hash» форматында жаңа файл жасайды. Сондай-ақ, бұл файлға қосымша мына ақпараттар жазылады: бастапқы файлдың көлемі, блоктар саны, хештеу операциясының орындалу уақыты және түрлендіру жылдамдығы.

Түзету ресурстарының статистикасы. Cortex процессоры 667 МГц жиілікте, ал ПЛИС 150 МГц жиілікте жұмыс істейді. Тақшаның энергия тұтынудың жалпы шығыны шамамен 0.3 Вт құрайды. ПЛИС келесі ресурстар: 2370 логикалық ұяшық, 384 биттік бір блокты түрлендіруге 32 такт жұмсалады (Кесте 4.1).

Кесте 4.1 – Нәтижелер көрсеткіші

Нәтиже нөмірі	Файл көлемі, байт	Орындалу жылдамдығы, Мб/с
1	384000	179,832≈22,5 МБ
2	1024000	179,885
3	64000000	179,917

Жасалған іске асыру тақшасы қолданыстағы алгоритмдерге сәйкес хеш түрлендіруді орындайтын аналогтармен бәсекеге түсе алады. Өнімділік жылдамдығы мен ПЛИС ресурстарының саны бойынша өнім қолданыстағы аналогтармен тепе-тең келеді немесе асып түседі.

4.3 HBC-256 алгоритмінің есептеу өнімділігін бағалау және оны арттырудың жолдары

Құрылған хештеу алгоритмінің есептеу өнімділігін бағалау барысында бағдарламалық және бағдарламалы-аппараттық жасақтамалар жасалып, олардың жылдамдық бойынша көрсеткіштеріне талдау жүргізілді. Алгоритмдегі бір және екі өлшемдегі массивтердің элементтерін есептеу – жалпы есептеу өнімділігіне едәуір әсер ететіні белгілі.

Бағдарламалық жасақтамада есептеу өнімділігі сипаттамасына ие есептеу машинесі (жеке компьютер) қолданылды. Келесі кестеде HBC-256 хештеу алгоритмінің бағдарламалық және бағдарламалы-аппараттық іске асырудағы есептеу жылдамдығы көрсетілген.

Кесте 4.2 – HBC-256 алгоритмінің есептеу жылдамдығы

Іске асыру түрі	Сипаттамалар	Есептеу жылдамдығы, МБ/сек
Бағдарламалық	Intel(R) Core i7-8700 2.90 GHz and 4 GB RAM	3,35
Бағдарламалы-аппараттық	Cortex A9 ядролы микропроцессорымен MYIR Z-Turn түзету тақшасы	22,5

Құрылған HBC-256 хештеу алгоритмінің жұмыс өнімділігін бағдарламалық іске асыру бойынша салыстырмалы талдау үшін блоктық шифрларға негізделген келесі екі хештеу алгоритмі қарастырылды:

1) 2013 жылы Ресей Федерациясында ГОСТ-Р 34.11-2012 мемлекеттік стандарт ретінде қабылданған хеш функциясын есептеуге арналған «Стрибог» алгоритмі. Талдау үшін хеш-мән өлшемі 256 биттік алгоритм нұсқасы таңдалды.

2) Үнді ғалымдары Khushboo Bussi, Dhananjoy Dey және басқалары ұсынған хеш функциясын есептеуге арналған MGR алгоритмдері [70]. Бұл хеш функциясы AES-текті блоктық шифрды қысу функциясы ретінде пайдаланатын «Стрибог» алгоритмінің модификациясы болып табылады. Есептеулер Intel(R) Core i7-8700 2.90 GHz and 4 GB RAM сипаттамасына ие есептеу техникасында жүргізілген.

Кесте 4.3 – Хештеу алгоритмдердің есептеу жылдамдығы

Ақпарат көлемі	«Стрибог»	MGR	HBC-256
1 МБ	3.34 сек.	1.2 сек.	0.58 сек.
5 МБ	16.52 сек.	5.84 сек.	2.98 сек.
10 МБ	33.01 сек.	11.50 сек.	5.6 сек.
20 МБ	66.13 сек.	22.95 сек.	11.94 сек.

Жоғарғы кестеден байқалатыны, HBC-256 алгоритмінің бағдарламалық жасақтамасы «Стрибог» және MGR алгоритмдерімен салыстырғанда өнімділік тұрғысынан жақсы нәтиже көрсеткенін көруге болады.

Есептеу өнімділігін арттырудың жолдары.

Осы мақсатта, алгоритмнің ерекшелігі, яғни алгоритм схемасындағы *k* параметрін 3-тен 8-ге дейінгі аралықта манипуляциялау арқылы параллель есептеу арқылы өнімділікті арттыру үшін салынған. Сондай-ақ, сызықсыздық дәрежесін көтеру үшін S-блоктарды бір раундтың ішінде бірнеше рет пайдалану қарастырылған және 16 байттық төрт S-блоктарды матрица элементтерінің индекстеріне байланысты жұптастырып пайдалану принципі қолданған.

Ескере кететіні, аппараттық платформа ретінде арнайы жасалған тақшадан гөрі нарықта бар реттеу тақшасы таңдалды. Дегенмен, өнімнің бірқатар параметрлерін жақсартуға болады, атап айтқанда:

– өлшемдік-габариттік сипаттамалар;

– SoC-ті аз функционалды нұсқасына ауыстыру (жұмыста ПЛИС ресурстарының 10% ғана жеткілікті болды, сол себепті Cortex процессорының бір ядросы өшірілді), бұл қуат тұтынуды үнемдеуге оң әсер етеді;

– оңтайлы тепе-теңдікке қол жеткізу үшін аппаратты іске асыруды оңтайландыру және процесті параллельдеу;

– үлкен көлемдегі файлдарды өңдеу үшін жедел жад көлемін ұлғайту.

ҚОРЫТЫНДЫ

Диссертациялық жұмыста блоктық шифрлау алгоритмі негізінде қауіпсіздігі қасиеттері мен өнімділігі жағынан жоғары, бағдарламалы-аппараттық жүзеге асыруға және параллельдік есептеуге икемделген жаңа хештеу алгоритмі әзірленіп, оған жан-жақты зерттеу жұмыстары жүргізілді. Ғылыми жұмысты орындау барысында төмендегідей нәтижелерге қол жеткізілді.

1) Криптографиялық хеш функцияларды құру және талдау бойынша қазіргі уақытта жүргізілген ғылыми-зерттеу жұмыстарына шолу, сонымен қатар заманауи хеш функциялардың қасиеттері мен оларға қойылатын талаптарға, құрылымдары мен қауіпсіздік деңгейіне және оларға бағытталған жалпы және арнайы шабуыл түрлеріне талдау жүргізілді.

2) Симметриялы блоктық шифрлау алгоритміне негізделген HBC-256 жаңа хештеу алгоритмі әзірленді. Алгоритмдегі қысу функциясы ретінде 128-биттік ұзындықтағы хабарлама блогы мен осы ұзындықтағы раундтық кілттен тұратын кірісі бар CF блоктық шифры жасалынды. Қысу функциясы шығысы болып 128-биттік аралық хеш-мән шығарады. Хеш функцияны құру үшін қолданыста кең тараған Merkle-Damgard конструкциясының Wide-pipe модификациясы қолданылды. Алгоритм құрылымы хабарламаның көлеміне байланысты бөліктер саны k параметрін өзгерте отырып, параллельді есептеу арқылы өнімділікті жақсартатындай етіп құрастырылды. Хеш функцияның қасиеттерінің бірі – қайтымсыздықты қамтамасыз ету мақсатында Девис-Мейер схемасы пайдаланды.

3) Құрылған HBC-256 хештеу алгоритмінің қауіпсіздігі қасиеттерін анықтау үшін төмендегідей зерттеулер жүргізілді:

а) HBC-256 хештеу алгоритмінің лавиндік және қатаң лавиндік әсерге қатысты зерттеу бағалары алынды. Зерттеу нәтижесі бойынша HBC-256 алгоритмінің лавиндік әсері хештеудің бірінші раундынан кейін-ақ жоғары болатыны көрсетілді.

ә) HBC-256 алгоритмінің хеш-мәндері NIST және D. Knuth статистикалық сынақтар жинағы арқылы жалғанкездейсоқтыққа зерттелді. Алынған нәтижелерге сүйене отырып, алынған хеш-мәндер тізбегі жалғанкездейсоққа өте жақын екендігін расталды.

б) Жүргізілген зерттеулер нәтижелеріне сәйкес, HBC-256 алгоритмі «жақын коллизиялар» шабуылға қатысты төзімді екеніне көз жеткізілді.

в) HBC-256 хештеу алгоритмінің дифференциалды талдауының нәтижелері S-блоктарды талдауға негізделген. Бірраундтық зерттеу бойынша коллизияны тудыру мүмкіндігі өте аз ықтималдықта болатындығы анықталып, толықраундтық HBC-256 хештеу функциясының коллизияларын табу үшін дифференциалды криптоталдау әдісін қолдану негізсіз екені дәлелденді.

г) Алгебралық криптоталдау жүргізу барысында HBC-256 хештеу алгоритмінің бір раунды үшін Transalg құралының көмегімен логикалық теңдеулер жүйесі құрылып, оны шешу үшін SAT шешушісі пайдаланылды. ЭЕМ-мүмкіндігін ескергенде, HBC-256 хештеу функциясының толықраундтық

алгоритмі үшін шешімдерін есептеу тұрғысынан табу мүмкін еместігі анықталды.

д) HBC-256 хештеу алгоритмінің бір раунды үшін сызықтық криптоталдау жүргізіліп, аталған хештеу алгоритміне осы талдауды жүргізу тиімсіз екені анықталды.

4) Құрылған HBC-256 хештеу алгоритмінің жасақтамасы бағдарламалық және бағдарламалы-аппараттық тұрғыда жүзеге асырылды. Бағдарламалық қамтамасыз ету тұрғыда ақпаратты хештеу үшін «ISL_HASH 1.0» және «CSP_HASH 1.0» жасақтамалары жасалынды. Сонымен бірге, аталған алгоритмді бағдарламалы-аппараттық тұрғыда жүзеге асыратын MYIR Z-turn түзету тақшасында макеттік үлгісі жасалды.

Сондай-ақ, зерттеу жұмысы барысында алынған нәтижелер халықаралық ғылыми конференциялардан бөлек Украина Ұлттық авиациялық университеті «Киберқауіпсіздік, компьютерлік және бағдарламалық инженерия» факультетінің ғылыми семинарында, Беларусь мемлекеттік университеті «Математика және информатиканың қолданбалы мәселелері» ғылыми зерттеу институты ғылыми семинарында және Біріккен Араб Әмірліктері Халифа университетінің «Electrical Engineering and Computer Science» факультетінің ғылыми семинарында талқыланды және оң бағаланды (Қосымша Б).

ПАЙДАЛАНЫЛҒАН ӘДЕБИЕТТЕР ТІЗІМІ

1 Khompysh A., Kapalova N.A., Algazy K.N., Dyusenbayev D.S., Sakan K.S. Design of substitution nodes (S-Boxes) of a block cipher intended for preliminary encryption of confidential information // Cogent Engineering. – 2022. – Vol. 9. – № 1. doi: 10.1080/23311916.2022.2080623.

2 Қазақстан Республикасы. ҚР Заңы. Электрондық құжат және электрондық цифрлық қолтаңба туралы: 2003 жылғы 7 қаңтарда қабылданған. https://adilet.zan.kz/kaz/docs/Z030000370_ 03.05.2023.

3 Қазақстан Республикасы. ҚР Заңы. Дербес деректер және оларды қорғау туралы: 2013 жылғы 21 мамырда қабылданған. https://adilet.zan.kz/kaz/docs/Z1300000094 03.05.2023.

4 Қазақстан Республикасы Президентінің Қаулысы. Қазақстан Республикасының Ақпараттық Доктринасы: 2023 жылғы 20 наурызда бекіт., № 145. https://akorda.kz/kz/kazakstan-respublikasynyn-akparattyk-doktrinasyn-bekitu-turaly-2025635 09.06.2023.

5 Қазақстан Республикасы Үкіметінің Қаулысы. Мәліметтерді таратылуы шектелген қызметтік ақпаратқа жатқызу және онымен жұмыс істеу қағидаларын бекіту туралы: 2022 жылғы 24 маусымда бекіт., № 429. https://adilet.zan.kz/kaz/docs/P2200000429 03.05.2023.

6 Қазақстан Республикасы Үкіметінің қаулысы. Ақпараттық-коммуникациялық технологиялар және ақпараттық қауіпсіздікті қамтамасыз ету саласындағы бірыңғай талаптарды бекіту туралы: 2016 жылғы 20 желтоқсанда бекіт., № 832. https://adilet.zan.kz/kaz/docs/P1600000832 03.05.2023.

7 Концепция кибербезопасности («Киберщит Казахстана»): утв. постановлением Правительства Республики Казахстан от 30 июня 2017 года, № 407. https://adilet.zan.kz/rus/docs/P1700000407 03.05.2023.

8 Криптографиялық қорғау құралдарына мемлекеттік стандарт (СТ РК 1073-2007). https://online.zakon.kz/Document/?doc_id=30615266 03.05.2023

9 International Standart ISO/IEC 10118-3:2018, IT Security techniques – Hash functions// https://standards.iteh.ai/catalog/standards/sist/972301fb-e6ca-4414-9a40- deb2fc3ba89a/iso-iec-10118-3-2018 05.04.2023.

10 SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions // https://www.nist.gov/publications/sha-3-standard-permutation-based-hash-and-extendable-output-functions?pub_id=919061: 14.01.2022.

11 Dworkin, M. J. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions // NIST Pub Series. doi: 10.6028/nist.fips.202.

12 The SM3 Cryptographic Hash Function // https://tools.ietf.org/id/draft-oscca-cfrg-sm3-02.html 10.01.2022.

13 ГОСТ 34.11-2018. Межгосударственный стандарт. Информационная технология. Криптографическая защита информации. Функция хэширования. https://docs.cntd.ru/document/1200161707 14.01.2021.

14 DSTU 7564:2014. Information Technologies. Cryptographic Data Security. Hash function // https://usts.kiev.ua/wp-content/uploads/2020/07/dstu-7564-2014.pdf 14.03.2022.

15 СТБ 34.101.77-2020. Государственный стандарт Республики Беларусь. Информационные технологии и безопасность. Криптографические алгоритмы на основе Sponge-функции // http://www.apmi.bsu.by/assets/files/std/bash-spec24.pdf 12.02.2022.

16 ҚР СТ ГОСТ Р 34.11-2015 «Ақпараттық технология. Ақпаратты криптографиялық қорғау. Хэштеу функциясы» // https://egfntd.kz/rus/tv/384070.html?sw_gr=-1&sw_str=&sw_sec=0 04.03.2022.

17 Молдабеков Д. Евразийский киберсоюз. История о несамостоятельности Казахстана в области кибер-безопасности // https://vlast.kz/obsshestvo/31791-evrazijskij-kibersouz.html 20.02.2022.

18 Damgard I. Collision Free Hash Functions and Public Key Signature Schemes // Eurocrypt’87. – Vol. 304 of LNCS. – P. 203-216. Springer-Verlag, 1987.

19 Rogaway P., Shrimpton T. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance // FSE’04. – Vol. 3017 of LNCS. – P. 371-388. Springer-Verlag, 2004.

20 Menezes A., Oorschot P., Vanstone S. Handbook of Applied Cryptography, chapter Hash Functions and Data Integrity. – CRC Press, 1996. – P. 321-384.

21 Kapalova N.A., Nyssanbayeva S.Y., Varennikov A.V., Dyusenbayev D.S., Sakan K.S. Higher professional and postgraduate training of information security specialists // Global Journal of Engineering Education. – 2022. – Vol. 24. – № 3. – P. 232-238.

22 Rajeev S., Geetha G. Cryptographic Hash Functions: A Review. International Journal of Computer Science Issues // ISSN (Online):1694-0814. – 2012. – Vol. 9. – P. 461-479.

23 Ferguson N., Schneier B. Practical Cryptography. – New York: John Wiley & Sons, 2003. – P. 410.

24 Joux A. Multicollisions in Iterated Hash Functions: Application to Cascaded Con- structions // Crypto'04. – Vol. 31252 of LNCS. – P. 306-316. Springer-Verlag, 2004.

25 Kelsey J., Kohno T. Herding Hash Functions and the Nostradamus Attack // Eurocrypt'06. – Vol. 4004 of LNCS. – P. 183-200. Springer-Verlag, 2006.

26 Bartkewitz T.S. Building hash functions from block ciphers. Their security and implementation properties // Ruhr-university Bochum. – 2009. – P. 1-22.

27 Winternitz R. A secure one-way hash function built from DES // IEEE Press. – 1984. – P. 88-90.

28 Preneel B., Govaerts R., Vandewalle J. Hash Functions Based on Block Ciphers: A Synthetic Approach // Crypto'93. – Vol. 773 of LNCS. – P. 368-378. Springer-Verlag, 1993.

29 Black J., Rogaway P., Shrimpton T. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV // Crypto'02. – Vol. 2442 of LNCS. – P. 320-335. Springer-Verlag, 2002.

30 Menezes A., Oorschot P., Vanstone S. Handbook of Applied Cryptography, chapter Hash Functions and Data Integrity. – Boca Raton, Florida: CRC Press, 1996. – P. 321-384.

31 Duo L., Li C. Improved Collision and Preimage Resistance Bounds on PGV Schemes // Cryptology ePrint Archive, Report 2006/462. – 2006. https://ia.cr/2006/462 03.05.2023.

32 Stam M. Blockcipher-Based Hashing Revisited // FSE'09. – Vol. 5665 of LNCS. – P. 69-83. Springer-Verlag, 2009.

33 Black J., Cochran M., Shrimpton T. On the Impossibility of Highly-E cient Blockcipher-Based Hash Functions // Journal of Cryptology. – 2009. – P. 311-329.

34 Rogaway P., Steinberger J. Constructing Cryptographic Hash Functions from Fixed-Key Blockciphers // Crypto'08. – Vol. 5157 of LNCS. – P. 433-450. Springer- Verlag, 2008.

35 Shrimpton T., Stam T. Building a Collision-Resistant Compression Function from Non-compressing Primitives // ICALP'08. – Vol. 5126 of LNCS. –P. 643-654. Springer-Verlag, 2008.

36 Ищукова Е.А., Маро Е.А., Бабенко Л.К., Алгазы К.Т, Сакан К.С. Исследование свойств хеширования «HBC-256» // Матер. VII междунар. науч.-практ. конф. «Информатика и прикладная математика». – Алматы, 2022. – С. 362-373.

37 Нысанбаева С.Е., Сақан Қ.С. Хештеу алгоритмдерін жасаудағы блокты шифрлау алгоритмдерін қолдану ерекшеліктері // Матер. VI междунар. науч.-практ. конф. «Информатика и прикладная математика». – Алматы, 2021. – С. 406-410.

38 Sakan K.S., Nyssanbayeva S.Y., Kapalova N.A., Algazy K.T., Khompysh A., Dyusenbayev D.S. Development and analysis of the new hashing algorithm based on block cipher // Eastern-European Journal of Enterprise Technologies. – 2022. – № 2/9 (116). – P. 60-73. doi:10.15587/1729-4061.2022.252060.

39 Algazy K.N., Sakan K.S., Kapalova N.A., Nyssanbayeva S.Y., Dyusenbayev D.S. Differential analysis of a cryptographic hashing algorithm HBC-256 // Appl. Sci. –2022. – Vol. 12, 10173. doi:10.3390/app121910173.

40 Бияшев Р.Г., Нысанбаева С.Е., Капалова Н.А., Дюсенбаев Д.С., Алгазы К.Т., Сакан К.С. Разработка и исследование алгоритмов хеширования произвольной длины: монография – Алматы, Guppyprint, 2022. – 95 с.

41 Сакан К.С., Алгазы К.Т. Криптографиялық хеш алгоритмдер жасаудың әртүрлі жолдарын қарастыру // V междунар. науч.-практ. конф. "Информатика и прикладная математика". – Алматы, 2020. – С. 374-377.

42 Нысанбаева С.Е., Сақан Қ.С. Хештеу алгоритмдерін жасаудағы блокты шифрлау алгоритмдерін қолдану ерекшеліктері // VI междунар. науч.-практ. конф. "Информатика и прикладная математика". – Алматы, 2021. – С. 406-410.

43 Алгазы К.Т., Сакан К.С. Принципы построения блочных шифров и требования к ним // VI междунар. науч.-практ. конф. "Информатика и прикладная математика". – Алматы, 2020. – С. 378-384.

44 Saarinen MJ.O. Cryptographic Analysis of All 4 × 4-Bit S-Boxes // Selected Areas in Cryptography. Lecture Notes in Computer Science. – Berlin, Heidelberg, 2012. – Vol. 7118. doi:10.1007/978-3-642-28496-0_7 20.09.2021.

45 Kapalova N.A., Sakan K.S., Haumen A., Suleimenov O.T. Requirements for symmetric block encryption algorithms developed for software and hardware

implementation // Journal «KazNU Bulletin. Series Mathematics, Mechanics and Computer Science». – 2021. – № 4(112). – P. 134-147.

46 Kapalova N.A., Algazy K.S., Sakan K.S., Dyussenbayev D. The algorithm of block encryption «Al03» and the results of its analysis // Bulletin of KazNPU. Series of Physics & Mathematical Sciences. – 2021. – № 3(75). – P. 108-114.

47 Kapalova N.A., Dyusenbayev D.S., Sakan K.S. A new hashing algorithm - HAS01: development, cryptographic properties and inclusion in graduate studies // Global Journal of Engineering Education. – 2022. – Vol. 24, № 2. – P. 155-164.

48 Al-Kuwari, S., Davenport, J., Bradford, R. Cryptographic Hash Functions: Recent Design Trends and Security Notions // IACR. https://eprint.iacr.org/2011/565.pdf 12.10.2021.

49 Иванов М. А., Стариковский А. В. Хеш-функции. Теория, применение и новые стандарты (часть 1). http://www.aha.ru/~msa/papers7.pdf 21.08.2021.

50 Dobrovolsky Y., Prokhorov G., Hanzhelo M., Hanzhelo D., Trembach D. Development of a hash algorithm based on cellular automata and chaos // Eastern-European Journal of Enterprise Technologies. – 2021. – № 5(9 (113). – P. 48-55. doi:10.15587/1729-4061.2021.242849.

51 Kapalova N.A., Khompysh A., Arici M., Algazy K.T. A block encryption algorithm based on exponentiation transform // Cogent Engineering. – 2020. – Vol. 7 (1788292). – P. 1-12. doi:10.1080/23311916.2020.1788292.

52 Algazy K.T., Babenko L.K., Biyashev R.G., Ishchukova E.A., Kapalova N.A., Nysynbaeva S.E., Smolarz A. Differential Cryptanalysis of New Qamal Encryption Algorithm // International journal of electronics and telecommunications. – 2020. – № 4. – P. 647-653.

53 Nysanbayeva S.E., Kapalova N.A., Dyusenbayev D.S., Algazy K.T., Sakan K.S. Investigation of the stastical security of a pseudo-random sequence generator // International Conference “Computer Data Analysis and Modeling: Stochastics & Data Science” (CDAM-2022). – Minsk. September 06-10, 2022. – P. 137-144.

54 Upadhyay D., Gaikwad N., Zaman M., Sampalli S. Investigating the Avalanche Effect of Various Cryptographically Secure Hash Functions and Hash-Based Applications // IEEE Access. – 2022. – Vol. 10. – P. 112472-112486. doi: 10.1109/access.2022.3215778.

55 Andrew R., Juan S. A statistical Test Suite foe Random and Pseudo Random Number Generators for Cryptographic Applications // https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf 21.03.2021.

56 Bassham L., Rukhin A., Soto J., Nechvatal J., Smid M., Leigh S., Levenson M., Vangel M., Heckert N., Banks D. A Statistical test suite for random and pseudorandom number generators for cryptographic applications, special publication (NIST SP), National institute of standards and technology, Gaithersburg, MD, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=906762 23.01.2023.

57 Кнут Э.Д. Искусство программирования. Получисленные методы. том 2. Издание 3. – Москва: Диалектика, 2019. – 832 с.

58 Капалова Н.А., Нысанбаева С.Е. Анализ статистических свойств алгоритма генерации псевдослучайных последовательностей // Матер. X

международ. науч.-практ. конф. «Информационная безопасность». – Таганрог: Изд-во ТТИ ЮФУ, 2008. – с. 169-172.

59 Kocarev L. Chaos-based cryptography: a brief overview // IEEE Circuits and Systems Magazine. – 2001. – Vol. 1. – № 3. – P. 6-21.

60 Saarinen, Markku-Juhani O. Cryptographic Analysis of All 4 x 4 - Bit S-Boxes // IACR Cryptol. ePrint Arch.2011(2011):218.

61 Anderson, Biham E., Knudsen L. Serpent: A Proposal for the Advanced Encryption Standard // http://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf 19.04.2022.

62 Vergili I., Yücel M. D. Avalanche and Bit Independence Properties for the Ensembles of Randomly Cho-sen× S-Boxes // Turk J Elec Engin. – 2001. – Vol. 9. – № 2. – P. 137-145.

63 Мулярчик К. С. Лавинный эффект в алгоритмах шифрования на основе дискретных хаотических отображений // Доклады БГУИР. – Минск, 2013. – № 6 (76). – С. 86-91.

64 Нысанбаева С.Е., Алғазы К.Т., Сақан Қ.С., Хомпыш А., Дуйсенбаев Д.С. CF блокты шифрлау алгоритмі және оны биттік шашырау эффектіне зерттеу // Вестник Евразийского национального университета имени Л.Н. Гумилева. Серия Математика. Информатика. Механика. – 2022, Т. 138, № 1. – С. 6-22.

65 Сейткулов Е., Оспанов Р., Ергалиева Б. О криптографических свойствах S-блоков // Вестник КазНИТУ. – 2021. – № 143(4). – С. 96-103. doi:10.51301/vest.su.2021.i4.12.

66 Соколов А., Жданов О. Строгий лавинный критерий четырехзначных функций как качественная характеристика стойкости криптографических алгоритмов // Сибирский научно-технический журнал. – 2019, № 20. – С. 183-190. doi:10.31772/2587-6066-2019-20-2-183-190.

67 Lamberger M., Mendel F., Rijmen V., Simoens K. Memoryless near-collisions via coding theory // Designs, Codes and Cryptography. – 2012. – № 62. – P. 1-18. doi:10.1007/s10623-011-9484-2.

68 Maram B.K., Gnanasekar J.M. Evaluation of Key Dependent S-Box Based Data Security Algorithm using Hamming Distance and Balanced Output // TEM Journal. – 2016. – № 5. – P. 67-75.

69 Kumar M., Dey D., Pal S.K., Panigrahi A. HeW: A Hash Function based on Lightweight Block Cipher FeW // Defence Science Journal. – 2017. – Vol. 67. – № 6. – P. 636-644. doi:10.14429/dsj.67.10791.

70Bussi K., Dey D., Mishra P.R., Dass B.K. (2019) MGR Hash Functions // Cryptologia. – 2019. – № 43:5. –P. 372-390. doi:10.1080/01611194.2019.1596995.

71 Biham E., Dunkelman O. Differential Cryptanalysis in Stream Ciphers // Cryptology ePrint Archive, Report 2007/218 // http://eprint.iacr.org/ 14.04.2022.

72 Kapalova N.A., Sakan K.S., Algazy K.T., Dyusenbayev D.S. Development and study of an encryption algorithm // Computation. – 2022. – Vol. 10. – № 198. Doi: 10.3390/computation10110198.

73 Babenko L., Ischukova E., Maro E. GOST Encryption Algorithm and Approaches to its Analysis // Theory and Practice of Cryptography Solutions for Secure Information Systems, IGI Global book series, Advances in Information Security,

Privacy, and Ethics (AISPE), USA: Information Science Reference, 2013. – P. 34-61. Doi:10.4018/978-1-4666-4030-6.ch002.

74 Ishchukova E., Tolomanenko E., Babenko L. Differential ananlysis of 3 round Kuznyechik // Proceedings of the 10th International Conf. on Security of Information and Networks. – 2017. – P. 29-36.

75 Gregory V. Bard. Algebraic Cryptanalysis. – New York: Springer, NY, 2009. Doi:10.1007/978-0-387-88757-9 17.09.2022.

76 Дюсенбаев Д.С., Алгазы К.Т., Сакан К.С. Исследование алгоритмов шифровании «Al01» и «Qamal» на основе алгебраического криптоанализа // Вестник КазНИТУ. – Алматы. – 2020. – № 5. – С. 620-629.

77 Алгазы К.Т., Капалова Н.А., Сакан К.С., Хомпыш А. Модификация алгоритма шифрования «Al01» // Вестник АУЭС, Алматы. – 2022. – № 1. – С. 162-170.

78 Courtois N. T., Pieprzyk J. Cryptanalysis of block ciphers with overdefined systems of equations // Advances in Cryptology - ASIACRYPT 2002: 8th International Conf. on the Theory and Application of Cryptology and Information Security Queenstown. –Springer Nature. – 2002. – Vol. 2501. – P. 267-287. Doi.org/10.1007/3-540-36178-2_17.

79 Shamir A., Patarin J., Courtois N., Klimov A. Efficient Algorithms for solving Overdefined Systems of Multivariate Polynomial Equations // Eurocrypt'2000, LNCS 1807. – Springer. – 2000. – P. 392-407.

80 Biyashev R.G., Dyusenbayev D.S., Algazy K.T., Kapalova K.A. Algebraic Cryptanalysis of Block Ciphers // Proceedings of the 2019 International Conf. on Wireless Communication, Network and Multimedia Engineering (WCNME 2019). Doi:10.2991/wcnme-19.2019.30.

81 Bardet M., Faugère J.-C, Salvy B. On the complexity of the F5 Gröbner basis algorithm // Journal of Symbolic Computation. – 2015. – Vol. 70. – P. 49-70.

82 Ishchukova E. Maro E. Pristalov P. Algebraic Analysis of a Simplified Encryption Algorithm GOST R 34.12-2015 // Computation. – 2020. – № 8(2):51. Doi:10.3390/computation8020051 05.06.2022.

83 Stachowiak, S., Kurkowski, M., Soboń, A. SAT-Based Cryptanalysis of Salsa20 Cipher // Progress in Image Processing, Pattern Recognition and Communication Systems. CORES IP&C ACS 2021. Lecture Notes in Networks and Systems. – Springer, Cham. – 2021. – Vol. 255. doi:10.1007/978-3-030-81523-3_25 07.07.2022.

84 Kochemazov, S. Exploring the Limits of Problem-Specific Adaptations of SAT Solvers in SAT-Based Cryptanalysis // Parallel Computational Technologies. PCT 2021. Communications in Computer and Information Science. – Springer, Cham, 2021. – Vol. 1437. Doi:10.1007/978-3-030-81691-9_11 19.05.2022.

85 Soos M. Enhanced Gaussian Elimination in DPLL-based SAT Solvers // POS@SAT. – 2010. – P. 2-14.

86 Armin B. Lingeling, Plingeling and Treengeling Entering the SAT Competition 2013 // http://fmv.jku.at/papers/Biere-SAT-Competition-2013-L: 16.07.2022.

87 Armin B. CADICAL at the SAT Race 2019 // https://cca.informatik.uni-freiburg.de/papers/Biere-SAT-Race-2019-solvers.pdf 17.09.2022.

88 Otpuschennikov I., Semenov A., Gribanova I. et al. Encoding Cryptographic Functions to SAT Using TRANSALG System // 22nd European Conf. on Artificial Intelligence - ECAI 2016, Frontiers in Artificial Intelligence and Applications. IOS Press. – 2016. – Vol. 285. – P. 1594-1595.

89 Algazy K.T., Sakan K.S., Kapalova N.A. Evaluation of the strength and performance of a new hashing algorithm based on a block cipher // International Journal of Electrical and Computer Engineering. – 2023. – Vol.13. – № 3. – P. 3124-3130. Doi: http://doi.org/10.11591/ijece.v13i3.

90 Yu L., Huicong L., Wei W., Meiqin W. New Linear Cryptanalysis of Chinese Commercial Block Cipher Standard SM4 // Security and Communication Networks, Hindawi. – Vol. 2017, Article ID 1461520, 10 pages, doi:10.1155/2017/1461520.

91 Matsui M. Linear cryptanalysis method for DES cipher // Advances in Cryptology – Eurocrypt'93. – Berlin: Springer, 1994. – P. 386-397.

92 Zhengbin Liu. Differential-linear cryptanalysis of PRINCE cipher // Chinese Journal of Network and Information Security. – 2021. – № 7(4). – P. 131-140. doi:10.11959/j.issn.2096-109x.2021072.

93 Biryukov A., Canniere C. Linear cryptanalysis for block ciphers // Encyclopedia of Cryptography and Security. – 2011. – P. 722-725. http://hdl.handle.net/10993/17077.

94 Borghoff J. Cryptanalysis of lightweight ciphers // Technical University of Denmark. – 2011. – P. 60-65, https://backend.orbit.dtu.dk/ws/portalfiles/ /portal/5456432/phd-thesis_Julia_Borghoff.pdf.

95 O'Connor, L. Properties of linear approximation tables // Preneel, B. (eds) Fast Software Encryption. FSE 1994. Lecture Notes in Computer Science. – Vol. 1008. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-60590-8_10.

96 Кузнецов А.А., Лисицкая И.В., Исаев С.А. Линейные свойства блочных симметричных шифров, представленных на украинский конкурс // Прикладная радиоэлектроника. – 2011. – Том 10. – № 2. – С. 135-140.

97 Dey S, Ghosh R. 2017. A review of existing 4-bit crypto S-box cryptanalysis techniques and two new techniques with 4-bit Boolean functions for cryptanalysis of 4-bit crypto S-boxes // Advances in Pure Mathematics. – 2018. – Vol. 8. – № 3. https://doi.org/10.7287/peerj.preprints.3441v1.

98 Heys H.M. A tutorial on linear and differential cryptanlysis // Cryptologia. – 2002. – № 26. – P. 189-221.

99 Khompysh A., Kapalova N.A., Algazy K.T., Sakan K.S. Study of the cryptographic strength of the S-box obtained on the basis of exponentiation modulo // Scientific Journal of Astana IT University. – 2022. – Vol. 12. Doi:10.37943/12DZLQ4553

ҚОСЫМША А

Жарияланымдар тізімі

1 Sakan K., Nyssanbayeva S., Kapalova N., Algazy K., Khompysh A., Dyusenbayev D. Development and analysis of the new hashing algorithm based on block cipher // Eastern-European Journal of Enterprise Technologies. Ukraine. – 2022. – № 2/9(116), https://doi.org/10.15587/1729-4061.2022.252060, percentile – 48. – pp. 60-73.

2 Ardabek Khompysh, Nursulu Kapalova , Kunbolat Algazy, Dilmukhanbet D yusenbayev, and Kairat Sakan. Design of substitution nodes (S-Boxes) of a block cipher intended for preliminary encryption of confidential information // Cogent Engineering. – 2022. – V. 9, № 1. https://doi.org/10.1080/23311916.2022.2080623, percentile – 66.

3 Nursulu Kapalova, Dilmukhanbet Dyusenbayev & Kairat Sakan. A new hashing algorithm - HAS01: development, cryptographic properties and inclusion in graduate studies // Global Journal of Engineering Education, Australia. – 2022. – V. 24, № 2, percentile – 62. – pp. 155-164, http://www.wiete.com.au/journals/GJEE/Publish/vol24no2/09-Sakan-K.pdf.

4 Algazy K., Sakan K., Kapalova N., Nyssanbayeva S. and Dyusenbayev D. Differential analysis of a cryptographic hashing algorithm HBC-256 // Appl. Sci. – 2022, 12(19), 10173. https://doi.org/ 10.3390/app121910173, percentile – 59.

5 Kapalova N., Sakan K. Algazy K. and Dyusenbayev D. Development and study of an encryption algorithm // Computation. – 2022, 10(11), 198. https://doi.org/ 10.3390/computation10110198, percentile – 70.

6 Nursulu Kapalova, Saule Nyssanbayeva, Andrey Varennikov, Dilmukhanbet Dyusenbayev & Kairat Sakan. Higher professional and postgraduate training of information security specialists // Global Journal of Engineering Education, Australia. – 2022. – Vol. 24, № 3, percentile – 62. – pp. 232-238, http://wiete.com.au/journals/GJEE/Publish/vol24no3/10-Sakan-K.pdf.

7 Kunbolat Algazy, Kairat Sakan, Nursulu Kapalova. Evaluation of the strength and performance of a new hashing algorithm based on a block cipher // International Journal of Electrical and Computer Engineering, Vol.13, № 3, June 2023, pp. 3124-3130, DOI: http://doi.org/10.11591/ijece.v13i3.pp3124-3130, percentile – 66.

8 Бияшев Р.Г., Нысанбаева С.Е., Капалова Н.А., Дюсенбаев Д.С., Алгазы К.Т., Сакан К.С. Разработка и исследование алгоритмов хеширования произвольной длины: монография – Алматы, Guppyprint, 2022. – 95 с. ISBN 978-601-08-2549-9.

9 Дюсенбаев Д.С., Алгазы К.Т., Сакан К.С. Исследование алгоритмов шифровании «Al01» и «Qamal» на основе алгебраического криптоанализа // Вестник КазНИТУ. – Алматы. – 2020. – № 5. – С. 620-629.

10 Kapalova N., Algazy K., Sakan K., Dyussenbayev D. The algorithm of block encryption «Al03» and the results of its analysis // Bulletin of KazNPU. Series of Physics & Mathematical Sciences. – Almaty. – 2021. – № 3(75). –pp. 108-114.

11 N.A. Kapalova, K.S. Sakan, A.Haumen, O.T.Suleimenov. Requirements for

symmetric block encryption algorithms developed for software and hardware implementation // Journal «KazNU Bulletin. Series Mathematics, Mechanics and Computer Science». – Almaty. – 2021. – № 4(112). –pp. 134–147.

12 К.Т. Алгазы, Н.А. Капалова, К.С. Сакан, А. Хомпыш. Модификация алгоритма шифрования «Al01» // Журнал «Вестник Алматинского университета энергетики и связи» – Алматы. – 2022. – № 1. – С. 162-70.

13 С.Е.Нысанбаева, К.Т.Алғазы, Қ.С.Сақан, А.Хомпыш, Д.С.Дуйсенбаев. CF блокты шифрлау алгоритмі және оны биттік шашырау эффектіне зерттеу // Вестник Евразийского национального университета имени Л.Н. Гумилева. Серия Математика. Информатика. Механика, 2022, Т. 138, № 1, – С. 6-22.

14 Ardabek Khompysh, Nursulu Kapalova, Kunbolat Algazy, Kairat Sakan. Study of the cryptographic strength of the S-box obtained on the basis of exponentiation modulo // Scientific Journal of Astana IT University, Vol. 12. 2022. DOI: 10.37943/12DZLQ4553.

15 Хомпыш А., Капалова Н.А., Сакан К.С., Дюсенбаев Д.С., Алгазы К.Т. Жаңа 4 биттік S-блок алу әдісі және алынған S-блокты қатал лавиндік критерийі бойынша зерттеу // Университет еңбектері КарТУ – Қарағанды, 2022. – № 4(89) – 411-417 б.

16 Сакан К.С., Алгазы К.Т. Криптографиялық хеш алгоритмдер жасаудың әртүрлі жолдарын қарастыру // Материалы V международной научно-практической конференции «Информатика и прикладная математика», 29 сентября – 01 октября 2020. – Алматы. – С. 374-378.

17 Алгазы К.Т., Сакан К.С. Принципы построения блочных шифров и требования к ним // Материалы V международной научно-практической конференции «Информатика и прикладная математика», 29 сентября – 01 октября 2020. – Алматы. – С. 378-384.

18 K.S.Sakan, K.T.Algazy Cryptographic attack to encryption algorithm “AL01” by the boomerang method // Advanced technologies and computer science. – Almaty. – 2020. – № 2. – pp. 21-25.

19 Д.С.Дюсенбаев, К.Т.Алғазы, Қ.С.Сақан. Симметриялы шифрларда қолданылатын сызықты емес түйіндерді зерттеу // Материалы международной научно-практической конференции «Актуальные проблемы информационной безопасности в Казахстане». – Алматы. 11 июня 2021 г., – С. 34-38.

20 С.Нысанбаева, Қ. Сақан. Хештеу алгоритмдерін жасаудағы блокты шифрлау алгоритмдерін қолдану ерекшеліктері // Материалы VI международной научно-практической конференции «Информатика и прикладная математика», 29 сентября – 02 октября 2021. – Алматы. – С. 406-410.

21 К.Алгазы, К.Сакан, Н.Капалова, Д.Дюсенбаев. HAS03 хеш алгоритмін құру және зерттеу // Материалы международной научной конференции в области информационных технологий, посвященной 75-летию профессора У.А. Тукеева. – Алматы. 08 октября 2021 г., – С. 55-61.

22 К.С. Сакан, Д.С. Дюсенбаев, К.Т. Алгазы, О.А. Лизунов, Хомпыш Ардабек. Разработка и анализ алгоритма хеширования «HAS01» // Сборник статей IV международной научно-технической конференции «Минские научные чтения-2021». – Минск. 09 декабря 2021 г. – Т. 3. – С. 181-187.

23 S.E. Nysanbayeva, N.A. Kapalova, D.S. Dyusenbayev, K.T. Algazy, K.S. Sakan. Investigation of the stastical security of a pseudo-random sequence generator // International Conference “Computer Data Analysis and Modeling: Stochastics & data Science” (CDAM-2022). – Minsk. September 06-10, 2022, P. 137-144.

24 Ищукова Е.А., Маро Е.А., Бабенко Л.К., Алгазы К.Т, Сакан К.С. Исследование свойств хеширования «HBC-256» // Материалы VII международной научно-практической конференции «Информатика и прикладная математика», 20 октября – 21 октября 2022. – Алматы. – С. 362-373.

25 С. Нысанбаева, Қ.Сақан. «HBC-256» хештеу алгоритміндегі S-блоктардың дифференциалды криптоталдаудағы критикалық нүктелерін зерттеу // Advanced technologies and computer science. – Almaty. – 2022. – № 4, – С. 15-24.

26 А. к. 20318. Ақпаратты хештеу алгоритмі «HBC-256» / Сақан Қ.С., Нысанбаева С.Е.; жариял. 20.09.2021. – 1 б.

27 А. к. 20661. Ақпаратты хештеу бағдарламасы «ISL_HASH 1.0» / Варенников А.В., Нысанбаева С.Е., Капалова Н.А., Дюсенбаев Д.С., Сақан Қ.С., Лизунов О.А.; жариял. 05.10.2021. – 1 б.

28 А. к. 23886. Ақпаратты хештеу бағдарламасы «CSP_HASH 1.0» / Лизунов О.А., Нысанбаева С.Е., Капалова Н.А., Дюсенбаев Д.С., Сақан Қ.С., Варенников А.В.; жариял. 24.02.2022. – 1 б.

29 А. к. 29379. ЭВМ-ге арналған бағдарлама «Криптопровайдер ISL_CSP 1.0» / Варенников А.В., Капалова Н.А., Алғазы К.Т., Дюсенбаев Д.С., Лизунов О.А., Сақан Қ.С.; жариял. 12.10.2022. – 1 б.

ҚОСЫМША Ә

Лицензия және авторлық куәліктер

Ақпаратты криптографиялық қорғау құралдарын (оның ішінде басқа да түрлерін) әзірлеуге және енгізуге мемлекеттік лицензия

17012125

ГОСУДАРСТВЕННАЯ ЛИЦЕНЗИЯ

03.07.2017 года **549**

Выдана **Республиканское государственное предприятие на праве хозяйственного ведения "Институт информационных и вычислительных технологий" Комитета науки Министерства образования и науки Республики Казахстан**

050010, Республика Казахстан, г.Алматы, УЛИЦА ПУШКИНА, дом № 125., БИН: 040740002672

(полное наименование, местонахождение, бизнес-идентификационный номер юридического лица (в том числе иностранного юридического лица), бизнес -идентификационный номер филиала или представительства иностранного юридического лица – в случае отсутствия бизнес-идентификационного номера у юридического лица/полностью фамилия, имя, отчество (в случае наличия), индивидуальный идентификационный номер физического лица)

на занятие **На осуществление деятельности по разработке и реализации (в том числе иной передаче) средств криптографической защиты информации**

(наименование лицензируемого вида деятельности в соответствии с Законом Республики Казахстан «О разрешениях и уведомлениях»)

Особые условия

(в соответствии со статьей 36 Закона Республики Казахстан «О разрешениях и уведомлениях»)

Примечание **Неотчуждаемая, класс 1**

(отчуждаемость, класс разрешения)

Лицензиар **Комитет национальной безопасности Республики Казахстан**

(полное наименование лицензиара)

Руководитель (уполномоченное лицо) **МАСИМОВ КАРИМ КАЖИМКАНОВИЧ**

(фамилия, имя, отчество (в случае наличия)

Дата первичной выдачи

Срок действия лицензии

Место выдачи **г.Астана**

ПРИЛОЖЕНИЕ К ГОСУДАРСТВЕННОЙ ЛИЦЕНЗИИ

Номер лицензии 549

Дата выдачи лицензии 03.07.2017 год

Подвид(ы) лицензируемого вида деятельности:

- Реализация (в том числе иная передача) средств криптографической защиты информации
- Разработка средств криптографической защиты информации

(наименование подвида лицензируемого вида деятельности в соответствии с Законом Республики Казахстан «О разрешениях и уведомлениях»)

Лицензиат **Республиканское государственное предприятие на праве хозяйственного ведения "Институт информационных и вычислительных технологий" Комитета науки Министерства образования и науки Республики Казахстан**

050010, Республика Казахстан, г.Алматы, УЛИЦА ПУШКИНА, дом № 125., БИН: 040740002672

(полное наименование, местонахождение, бизнес-идентификационный номер юридического лица (в том числе иностранного юридического лица), бизнес-идентификационный номер филиала или представительства иностранного юридического лица – в случае отсутствия бизнес-идентификационного номера у юридического лица/полностью фамилия, имя, отчество (в случае наличия), индивидуальный идентификационный номер физического лица)

Производственная база

(местонахождение)

Особые условия действия лицензии

(в соответствии со статьей 36 Закона Республики Казахстан «О разрешениях и уведомлениях»)

Лицензиар **Комитет национальной безопасности Республики Казахстан**

(полное наименование органа, выдавшего приложение к лицензии)

Руководитель (уполномоченное лицо) МАСИМОВ КАРИМ КАЖИМКАНОВИЧ

(фамилия, имя, отчество (в случае наличия))

Номер приложения 001

Срок действия

Дата выдачи приложения 03.07.2017

Место выдачи г.Астана

Авторлық куәлік № 20318 –
Ақпаратты хештеу алгоритмі «HBC-256»

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2021 жылғы «20» қыркүйек № 20318

Автордың (лардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
САҚАН ҚАЙРАТ САҚАНҰЛЫ, НЫСАНБАЕВА САУЛЕ ЕРКЕБУЛАНОВНА

Авторлық құқық объектісі: ғылыми туынды

Объектінің атауы: Алгоритм хеширования данных «HBC-256»

Объектіні жасаған күні: 15.09.2021

Құжат түпнұсқалығын http://www.kazpatent.kz/ru сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

ЭЦҚ қол қойылды

Оспанов Е.К.

Авторлық куәлік 20661 –
Ақпаратты хештеу бағдарламасы «ISL_HASH 1.0»

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2021 жылғы «5» қазан № 20661

Автордың (лардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
ВАРЕННИКОВ АНДРЕЙ ВЛАДИСЛАВОВИЧ, НЫСАНБАЕВА САУЛЕ ЕРКЕБУЛАНОВНА, ҚАПАЛОВА НУРСУЛУ АЛДАЖАРОВНА, ДЮСЕНБАЕВ ДИЛМУХАНБЕТ САМУРАТОВИЧ, САҚАН ҚАЙРАТ САҚАНҰЛЫ, ЛИЗУНОВ ОЛЕГ АЛЕКСАНДРОВИЧ

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: Программа хеширования данных «ISL_HASH 1.0»

Объектіні жасаған күні: 23.09.2021

Құжат түпнұсқалығын http://www.kazpatent.kz/ru сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

ЭЦҚ қол қойылды

Оспанов Е.К.

Авторлық куәлік 23886 –
Ақпаратты хештеу бағдарламасы «CSP_HASH 1.0»

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2022 жылғы «24» ақпан № 23886

Автордың (лардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе): **ЛИЗУНОВ ОЛЕГ АЛЕКСАНДРОВИЧ, НЫСАНБАЕВА САУЛЕ ЕРКЕБУЛАНОВНА, КАПАЛОВА НУРСУЛУ АЛДАЖАРОВНА, ДЮСЕНБАЕВ ДИЛМУХАНБЕТ САМУРАТОВИЧ, САКАН ҚАЙРАТ САҚАНҰЛЫ, ВАРЕННИКОВ АНДРЕЙ ВЛАДИСЛАВОВИЧ**

Авторлық құқық объектісі: **ЭЕМ-ге арналған бағдарлама**

Объектінің атауы: **Программа хеширования данных «CSP_HASH 1.0»**

Объектіні жасаған күні: **21.02.2022**

Құжат тұтастығын http://www.kazpatent.kz/ сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

ЭЦҚ қол қойылды

Е. Куантыров

Авторлық куәлік 29379 –
ЭВМ-ге арналған бағдарлама «Криптопровайдер ISL_CSP 1.0»

ҚАЗАҚСТАН РЕСПУБЛИКАСЫ

РЕСПУБЛИКА КАЗАХСТАН

АВТОРЛЫҚ ҚҰҚЫҚПЕН ҚОРҒАЛАТЫН ОБЪЕКТІЛЕРГЕ ҚҰҚЫҚТАРДЫҢ МЕМЛЕКЕТТІК ТІЗІЛІМГЕ МӘЛІМЕТТЕРДІ ЕНГІЗУ ТУРАЛЫ

КУӘЛІК

2022 жылғы «12» қазан № 29379

Автордың (лардың) жөні, аты, әкесінің аты (егер ол жеке басын куәландыратын құжатта көрсетілсе):
ВАРЕННИКОВ АНДРЕЙ ВЛАДИСЛАВОВИЧ, КАПАЛОВА НУРСУЛУ АЛДАЖАРОВНА, АЛҒАЗЫ КҮНБОЛАТ ТІЛЕУХАНҰЛЫ, ДЮСЕНБАЕВ ДИЛМУХАНБЕТ САМУРАТОВИЧ, ЛИЗУНОВ ОЛЕГ АЛЕКСАНДРОВИЧ, САҚАН ҚАЙРАТ САҚАНҰЛЫ

Авторлық құқық объектісі: ЭЕМ-ге арналған бағдарлама

Объектінің атауы: «Криптопровайдер ISL_CSP 1.0»

Объектіні жасаған күні: 30.09.2022

Құжат түпнұсқалығын http://www.kazpatent.kz/ru сайтының "Авторлық құқық" бөлімінде тексеруге болады https://copyright.kazpatent.kz

Подлинность документа возможно проверить на сайте kazpatent.kz в разделе «Авторское право» https://copyright.kazpatent.kz

ЭЦҚ қол қойылды

Е. Оспанов

ҚОСЫМША Б

Ғылыми семинарлар хаттамалары

1) Украина Ұлттық авиациялық университеті «Киберқауіпсіздік, компьютерлік және бағдарламалық инженерия» факультетінің (ФКбКПИ НАУ) ғылыми семинары хаттамасы (Киев, Украина, 3 желтоқсан 2021)

Выписка из протокола № 2 от 03 декабря 2021 года
заседания научного семинара Факультета кибербезопасности, компьютерной и программной инженерии Национального авиационного университета

03 декабря 2021 г. г. Киев

Присутствовали: 16 человек.
Д.т.н., профессор Гнатюк С.А., д.т.н., профессор Одарченко Р.С., д.т.н., профессор Фауре Э.В., д.т.н., профессор Смирнов А.А., д.т.н., профессор Бабенко В.Г., д.т.н., профессор Белецкий А.Я., д.т.н., профессор Василиу Е.В., к.т.н., доцент Фесенко А.А., к.т.н., доцент Охрименко Т.А., PhD Явич М.П., докторант КазНУ им. аль-Фараби Иманбаев А., докторант КазНУ им. аль-Фараби Зиро А., лаборант Горбаха Б.М., PhD-аспирант Евченко Я.В., сотрудник ИИВТ КН МОН РК Алгазы К.Т., сотрудник ИИВТ КН МОН РК Сакан К.С.

ПОВЕСТКА ДНЯ:

3. Представление и обсуждение результатов научно-исследовательских работ представителей Института информационных и вычислительных технологий КН МОН РК

СЛУШАЛИ:

Сотрудники Института информационных и вычислительных технологий МОН РК представили результаты научно-исследовательских работ по разработке криптографических алгоритмов по темам:

- «Особенности проектирования симметричных алгоритмов блочных шифрования, применяемых для алгоритмов хеширования» – научный сотрудник ИИВТ КН МОН РК Алгазы К.Т.;
- «Разработка и исследование алгоритма хеширования HBC-256» – младший научный сотрудник ИИВТ КН МОН РК Сакан К.

РЕКОМЕНДОВАЛИ:

1. Провести углубленные анализы по исследованию стойкости разработанных алгоритмов (стойкость коллизиям);
2. Изучить особенности конструкции алгоритма хеш-функции Wide-pipe с обоснованиями криптографических свойств (различные подходы функции ComF);
3. Детально изучить строение хеш-функции и схемы Дэвисе-Мейера, после чего построить их сравнительный анализ. На основе найденных различий произвести поиск уязвимости в структуре хеш-функции и пострить атаку;
4. Рассмотреть возможность применения алгоритма в технологии блокчейн;
5. Провести дополнительные исследования по повышению производительности при программно-аппаратной реализации алгоритмов. Рассмотреть возможность функционирования алгоритма при разных *k*.

Заместитель декана Факультета кибербезопасности,
компьютерной и программной инженерии
Национального авиационного университета, д.т.н.,
профессор Гнатюк С.А.

Секретарь Горбаха Б.М.

2) Беларусь мемлекеттік университеті «Математика және информатика қолданбалы мәселелері» ғылыми зерттеу институты ғылыми семинары хаттамасы (НИИ ППМИ БГУ) (Минск, Беларусь, 6 қыркүйек 2022)

ВЫПИСКА
из протокола научного семинара
НИИ прикладных проблем математики и информатики
Белорусского государственного университета (НИИ ППМИ БГУ)

06 сентября 2022 г. г. Минск

ПРИСУТСТВОВАЛИ: сотрудники НИЛ проблем безопасности информационных технологий НИИ ППМИ БГУ и сотрудники Института информационных технологий МНВО РК: к.т.н. Капалова Н.А., PhD Алгазы К.Т., Сакан К.С., Дюсенбаев Д.С.

ПОВЕСТКА ДНЯ:

1. Представление и обсуждение результатов научно-исследовательских работ представителей Института информационных технологий МНВО РК.

СЛУШАЛИ:

Соотрудники Института информационных технологий МНВО РК представили результаты научно-исследовательских работ по разработке криптографических алгоритмов по темам:

– «Разработка алгоритма хеширования HBC-256»;
– «Алгоритм хеширования HAS01».

РЕКОМЕНДОВАЛИ:

1. Провести дополнительные исследования по обоснованию стойкости разработанных алгоритмов и, в частности, оценке снизу числа активаций (активных *S*-блоков).
2. Провести дополнительные исследования по выбору *S*-блоков. Учесть дополнительные критерии выбора, в том числе критерии, связанные с защитой от алгебраических атак.

Заведующий НИЛ проблем безопасности информационных технологий НИИ ППМИ БГУ канд. физ.-мат. наук С.В. Агиевич

06.09.2022

Личную подпись Агиевича С.В. удостоверяю
Ведущий специалист по кадрам НИИ прикладных проблем математики и информатики Янина М.В.
«06» сентября 2022

3) Electrical Engineering and Computer Science Department of Khalifa University ғылыми семинары хаттамасы (Абу-Даби, БАӘ, 12 желтоқсан 2022)

Extract from Minutes

of the meeting of the Scientific Seminar of the Electrical Engineering and Computer Science (EECS) Department of the Khalifa University

December 13, 2022, Abu Dhabi

Attendees: Prof. Thanos Stouraitis, Prof. Chan Yeun, Cand. Tech. Sc. N.A. Kapalova, PhD K.T. Algazy, K.S. Sakan, D.S. Dyusenbayev and other members of the EECS.

AGENDA:

1. Presentation and discussion of the results of the research work of representatives of the Institute of Information and Computational Technologies of the Ministry of Science and Higher Education of the Republic of Kazakhstan.

HEARD:

Members of the Institute of Information and Computational Technologies of the RK MSHE presented the results of research work on the development of cryptographic algorithms on the topics:

- The main directions of R&D in the Laboratory of Information Security,
- The HBC-256 Hashing Algorithm,
- The QAMAL Symmetric Block Encryption Algorithm and LBC-3 Lightweight Encryption Algorithm.

DISCUSSED:

1. In-depth study of the strength of the developed algorithms.
2. Additional research to improve performance in hardware-software implementation of algorithms.
3. Possibility for work on post-quantum cryptographic algorithms.

Electrical Engineering and Computer Science

Professor Thanos Stouraitis

ҚОСЫМША В

Енгізу актісі

УТВЕРЖДАЮ
Зам. генерального директора
РГП на ПХВ «Институт информационных и
вычислительных технологий» КН МНВО РК
_____ Айнакулов С.Ж
«28» 02 2023 г.

АКТ
о внедрении результатов диссертационной работы
Сақан Қайрат Сақанұлы

Экспертная комиссия РГП на ПХВ «Институт информационных и вычислительных технологий» (ИИВТ) Комитета науки МНВО РК в составе:

председатель: заместитель генерального директора ИИВТ по науке, ассоц. проф. Мамырбаев О.Ж.:

члены:

д.т.н., доцент, ГНС ИИВТ Нысанбаева С.Е.;

к.т.н., ассоц.проф., ВНС ИИВТ Капалова Н.А.;

PhD, ученый секретарь ИИВТ Усатова О.А.

составили настоящий акт о том, что результаты диссертационной работы «Разработка алгоритмов хеширования на основе итеративных блочных шифров и исследование их криптостойкости» МНС ЛИБ ИИВТ Сақан Қ.С. были получены при выполнении проекта РГП на ПХВ ИИВТ КН МНВО РК:

– проект программно-целевого финансирования (ПЦФ) «Разработка и исследование алгоритмов хеширования произвольной длины цифровых подписей и оценка их стойкости» на 2021-2022 годы, OR № 11465439, источник финансирования Комитет науки МНВО РК.

Результаты включены в отчёты указанного проекта ПЦФ, № гос.рег. 0121РК00795.

Краткое содержание внедренных результатов:

1. разработан новый симметричный алгоритм блочного шифрования CF;
2. разработан алгоритм хеширования HBC-256 на основе алгоритма шифрования CF;
3. проведены исследования безопасности разработанного алгоритма хеширования HBC-256 с применением методов линейного, дифференциального и алгебраического криптоанализа, а также наборов статистических тестов NIST и Д.Кнута;
4. осуществлена программная и программно-аппаратная реализация разработанного алгоритма хеширования HBC-256.

Материалы к настоящему акту были рассмотрены на Ученом Совете института (протокол № 2 от 27 февраля 2023 год).

Председатель комиссии _____ Мамырбаев О.Ж.

Члены комиссии _____ Нысанбаева С.Е.

_____ Капалова Н.А.

_____ Усатова О.А.

ҚОСЫМША Г

Хештеу алгоритміне мысал

Өзіміз таңдап алған ашық мәтін (хабарлама) *M*-ді қалай хештелетінін қарастырайық, мұндағы: *M*={Republic of Kazakhstan}= $(52656275626C6963206F66204B617A616B687374616E)_{16}$. Хабарламаның көлемі: 22 байт. Бөлікшелер $k = 3$ болғанда, бөліктер саны 1-ге тең. Инициализациядық вектор, яғни бастапқы хеш-мән – h_0^j = 0000000000000000000000000000000, $j = 0, \dots, k-1$. Түсінуге және қабылдауға ыңғайлы болуы үшін мәндер он алтылық жүйеде көрсетілген. Бір бөлік 48 байттан тұратындықтан, алғашқы хабарламаға толық бөлікке дейін толықтыру жүргіземіз:

$M = M \parallel Pad(M)$= =52656275626C6963206F66204B617A616B687374616E800000 00000000000000000000000000000000000000000000001. $k = 3$ болғандықтан, бөлікшелердің мәні төмендегідей болады:

m^0= 52 65 62 75 62 6C 69 63 20 6F 66 20 4B 61 7A 61,

m^1= 6B 68 73 74 61 6E 80 00 00 00 00 00 00 00 00 00,

m^2= 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01.

Бұдан арғы процесті раундтар бойынша Кесте Г.1 – Кесте Г.8 арқылы көрсетейік:

Кесте Г.1 – Хештеудің 1-раундтағы жұмыс нәтижелері

Параметрлер			Мәндер
1-бөлікше	$rk_0^0 := m^0$		52 65 62 75 62 6C 69 63 20 6F 66 20 4B 61 7A 61
	$CFKey: rk_1^0$		CF 23 72 43 F8 E2 B5 58 6C DD AA E5 AC 65 B6 07
	h_0^0		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	$h_0^0 := h_0^0 \oplus rk_0^0$		52 65 62 75 62 6C 69 63 20 6F 66 20 4B 61 7A 61
	CF:	h_1^0	5E 26 08 ED BA 01 DE 75 B0 51 02 02 1B 79 14 D4
		$h_1^0 := h_1^0 \oplus rk_1^0$	91 05 7A AE 42 E3 6B 2D DC 8C A8 E7 B7 1C A2 D3
2-бөлікше	$rk_0^1 := m^1$		6B 68 73 74 61 6E 80 00 00 00 00 00 00 00 00 00
	$CFKey: rk_1^1$		6D EC 87 FC 3D 92 D6 E6 2A 6F 8F 84 6E 74 AB 19
	h_0^1		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	$h_0^1 := h_0^1 \oplus rk_0^1$		6B 68 73 74 61 6E 80 00 00 00 00 00 00 00 00 00
	CF:	h_1^1	84 01 68 B6 6F DD 02 D9 ED DA 93 47 55 1D DB 6D
		$h_1^1 := h_1^1 \oplus rk_1^1$	E9 ED EF 4A 52 4F D4 3F C7 B5 1C C3 3B 69 70 74
3-бөлікше	$rk_0^2 := m^2$		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01
	$CFKey: rk_1^2$		83 6B 6B AA 8E 91 48 0A F4 01 D8 BD ED D1 82 CB
	h_0^2		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	$h_0^2 := h_0^2 \oplus rk_0^2$		00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01
	CF:	h_1^2	53 50 67 1E B3 7A 55 2E 08 CA 97 BE EC AA E9 B5
		$h_1^2 := h_1^2 \oplus rk_1^2$	D0 3B 0C B4 3D EB 1D 24 FC CB 4F 03 01 7B 6B 7E
$h_1^0 := h_0^0 \oplus h_1^0$			91 05 7A AE 42 E3 6B 2D DC 8C A8 E7 B7 1C A2 D3
$h_1^1 := h_0^1 \oplus h_1^1$			E9 ED EF 4A 52 4F D4 3F C7 B5 1C C3 3B 69 70 74
$h_1^2 := h_0^2 \oplus h_1^2$			D0 3B 0C B4 3D EB 1D 24 FC CB 4F 03 01 7B 6B 7E
$PerF(h_1^0, h_1^1, h_1^2)$ – бөлікшелер арасында өзара орын алмастырудан соң:			
h_1^0			91 E9 D0 05 ED 3B 7A EF 0C AE 4A B4 42 52 3D E3
h_1^1			4F EB 6B D4 1D 2D 3F 24 DC C7 FC 8C B5 CB A8 1C
h_1^2			4F E7 C3 03 B7 3B 01 1C 69 7B A2 70 6B D3 74 7E

Кесте Г.2 – Хештеудің 2-раундтағы жұмыс нәтижелері

Параметрлер			Мәндер
1-бөлікше	$CFKey: rk_2^0$		0E FF 0B ED 4A 09 72 71 3B B0 46 92 23 8F 51 B2
	h_0^0:=h_1^0		91 E9 D0 05 ED 3B 7A EF 0C AE 4A B4 42 52 3D E3
	CF:	h_1^0	46 B0 8C CD D9 48 2B DA B0 15 F2 0E 02 62 5F 39
		$h_1^0 \coloneqq h_1^0 \oplus rk_2^0$	48 4F 87 20 93 41 59 AB 8B A5 B4 9C 21 ED 0E 8B
2-бөлікше	$CFKey: rk_2^1$		14 18 1F 77 EA 40 6A 72 D6 9B 01 FD 48 27 80 86
	h_0^1:= h_1^1		4F EB 6B D4 1D 2D 3F 24 DC C7 FC 8C B5 CB A8 1C
	CF:	h_1^1	DB 52 B3 BD FA F3 C5 83 CE BC 26 40 6A 3A 45 DB
		$h_1^1 := h_1^1 \oplus rk_2^1$	CF 4A AC CA 10 B3 AF F1 18 27 27 BD 22 1D C5 5D
3-бөлікше	$CFKey: rk_2^2$		1C 77 4D 0F 94 BF B7 5B 03 19 A7 15 99 EE A6 A7
	$h_0^2 := h_1^2$		4F E7 C3 03 B7 3B 01 1C 69 7B A2 70 6B D3 74 7E
	CF:	h_1^2	2E 03 30 DB 29 06 BD AE 83 6A FC AC 21 8D 8A B8
		$h_1^2 := h_1^2 \oplus rk_2^2$	32 74 7D D4 BD B9 0A F5 80 73 5B B9 B8 63 2C 1F
$h_1^0 := h_0^0 \oplus h_1^0$			D9 A6 57 25 7E 7A 23 44 87 0B FE 28 63 BF 33 68
$h_1^1 := h_0^1 \oplus h_1^1$			80 A1 C7 1E 0D 9E 90 D5 C4 E0 DB 31 97 D6 6D 41
$h_1^2 := h_0^2 \oplus h_1^2$			7D 93 BE D7 0A 82 0B E9 E9 08 F9 C9 D3 B0 58 61
$PerF(h_1^0, h_1^1, h_1^2)$ – бөлікшелер арасында өзара орын алмастырудан соң:			
h_1^0			D9 80 7D A6 A1 93 57 C7 BE 25 1E D7 7E 0D 0A 7A
h_1^1			9E 82 23 90 0B 44 D5 E9 87 C4 E9 0B E0 08 FE DB
h_1^2			F9 28 31 C9 63 97 D3 BF D6 B0 33 6D 58 68 41 61

Кесте Г.3 – Хештеудің 3-раундтағы жұмыс нәтижелері

Параметрлер			Мәндер
1-бөлікше	$CFKey: rk_2^0$		3C 22 9E AA 3F F6 B4 85 F4 4C 2B 53 62 CB CC C4
	h_0^0:=h_1^0		D9 80 7D A6 A1 93 57 C7 BE 25 1E D7 7E 0D 0A 7A
	CF:	h_1^0	25 05 0F 1D D8 68 85 25 34 FE A3 B4 9E DD 95 DD
		$h_1^0 \coloneqq h_1^0 \oplus rk_2^0$	19 27 91 B7 E7 9E 31 A0 C0 B2 88 E7 FC 16 59 19
2-бөлікше	$CFKey: rk_2^1$		85 71 C4 B1 36 2E 3A F1 8D 91 3D 9F 18 A6 70 B2
	h_0^1:= h_1^1		9E 82 23 90 0B 44 D5 E9 87 C4 E9 0B E0 08 FE DB
	CF:	h_1^1	1B A6 4D F1 33 69 AA D9 33 3C B2 9C AD F9 A4 1E
		$h_1^1 := h_1^1 \oplus rk_2^1$	9E D7 89 40 05 47 90 28 BE AD 8F 03 B5 5F D4 AC
3-бөлікше	$CFKey: rk_2^2$		E5 85 98 10 5D FB 93 3F 82 62 A0 34 4C 6B D8 AD
	$h_0^2 := h_1^2$		F9 28 31 C9 63 97 D3 BF D6 B0 33 6D 58 68 41 61
	CF:	h_1^2	A4 3A 97 A8 F8 50 78 31 55 83 E5 6C 4A 56 F4 FA
		$h_1^2 := h_1^2 \oplus rk_2^2$	41 BF 0F B8 A5 AB EB 0E D7 E1 45 58 06 3D 2C 57
$h_1^0 := h_0^0 \oplus h_1^0$			C0 A7 EC 11 46 0D 66 67 7E 97 96 30 82 1B 53 63
$h_1^1 := h_0^1 \oplus h_1^1$			00 55 AA D0 0E 03 45 C1 39 69 66 08 55 57 2A 77
$h_1^2 := h_0^2 \oplus h_1^2$			B8 97 3E 71 C6 3C 38 B1 01 51 76 35 5E 55 6D 36
$PerF(h_1^0, h_1^1, h_1^2)$ – бөлікшелер арасында өзара орын алмастырудан соң:			
h_1^0			C0 00 B8 A7 55 97 EC AA 3E 11 D0 71 46 0E C6 0D
h_1^1			03 3C 66 45 38 67 C1 B1 7E 39 01 97 69 51 96 66
h_1^2			76 30 08 35 82 55 5E 1B 57 55 53 2A 6D 63 77 36

Кесте Г.4 – Хештеудің 4-раундтағы жұмыс нәтижелері

Параметрлер			Мәндер
1-бөлікше	$CFKey: rk_2^0$		E4 3D 29 52 5B EA 33 60 34 6E 88 E6 15 B4 ED F0
	h_0^0:=h_1^0		C0 00 B8 A7 55 97 EC AA 3E 11 D0 71 46 0E C6 0D
	CF:	h_1^0	E1 B6 D4 58 6E C5 97 4B 84 DE B8 43 8D E0 52 C8
		$h_1^0 := h_1^0 \oplus rk_2^0$	05 8B FD 0A 35 2F A4 2B B0 B0 30 A5 98 54 BF 38
2-бөлікше	$CFKey: rk_2^1$		45 1E 03 42 E1 C7 9B 80 BA D9 84 F1 C1 B0 DA 7E
	h_0^1:= h_1^1		03 3C 66 45 38 67 C1 B1 7E 39 01 97 69 51 96 66
	CF:	h_1^1	F0 55 2F CE 32 66 6A 95 DA B1 03 74 C4 9F 2D 4D
		$h_1^1 := h_1^1 \oplus rk_2^1$	B5 4B 2C 8C D3 A1 F1 15 60 68 87 85 05 2F F7 33
3-бөлікше	$CFKey: rk_2^2$		35 AB D2 6D 8E 7F 0F 35 55 4D 4E CB 61 F4 09 4D
	$h_0^2 := h_1^2$		76 30 08 35 82 55 5E 1B 57 55 53 2A 6D 63 77 36
	CF:	h_1^2	C7 16 A8 DB C1 36 39 EA C4 19 EE D4 2B E8 32 BA
		$h_1^2 := h_1^2 \oplus rk_2^2$	F2 BD 7A B6 4F 49 36 DF 91 54 A0 1F 4A 1C 3B F7
$h_1^0 := h_0^0 \oplus h_1^0$			C5 8B 45 AD 60 B8 48 81 8E A1 E0 D4 DE 5A 79 35
$h_1^1 := h_0^1 \oplus h_1^1$			B6 77 4A C9 EB C6 30 A4 1E 51 86 12 6C 7E 61 55
$h_1^2 := h_0^2 \oplus h_1^2$			84 8D 72 83 CD 1C 68 C4 C6 01 F3 35 27 7F 4C C1
$PerF(h_1^0, h_1^1, h_1^2)$ – бөлікшелер арасында өзара орын алмастырудан соң:			
h_1^0			C5 B6 84 8B 77 8D 45 4A 72 AD C9 83 60 EB CD B8
h_1^1			C6 1C 48 30 68 81 A4 C4 8E 1E C6 A1 51 01 E0 86
h_1^2			F3 D4 12 35 DE 6C 27 5A 7E 7F 79 61 4C 35 55 C1

Кесте Г.5 – Хештеудің 5-раундтағы жұмыс нәтижелері

Параметрлер			Мәндер
1-бөлікше	$CFKey: rk_2^0$		28 AB 2E 5E 3F 31 6D C4 E3 54 CD CC 41 53 02 AC
	h_0^0:=h_1^0		C5 B6 84 8B 77 8D 45 4A 72 AD C9 83 60 EB CD B8
	CF:	h_1^0	F5 35 03 25 88 8C 48 6B 42 20 8D F3 19 DE 2C 9B
		$h_1^0 := h_1^0 \oplus rk_2^0$	DD 9E 2D 7B B7 BD 25 AF A1 74 40 3F 58 8D 2E 37
2-бөлікше	$CFKey: rk_2^1$		63 F9 B3 CB 83 40 E1 AD 78 C2 89 CE BE E3 15 44
	h_0^1:= h_1^1		C6 1C 48 30 68 81 A4 C4 8E 1E C6 A1 51 01 E0 86
	CF:	h_1^1	D4 56 5B 88 56 FE 52 58 D2 1F FB 06 18 CB 28 5B
		$h_1^1 := h_1^1 \oplus rk_2^1$	B7 AF E8 43 D5 BE B3 F5 AA DD 72 C8 A6 28 3D 1F
3-бөлікше	$CFKey: rk_2^2$		A1 C5 48 B8 7F A5 D2 FA 06 C9 89 0A E8 48 4E 98
	$h_0^2 := h_1^2$		F3 D4 12 35 DE 6C 27 5A 7E 7F 79 61 4C 35 55 C1
	CF:	h_1^2	1B 88 53 5D 19 D0 FE 82 BB 23 59 84 E2 4D 70 04
		$h_1^2 := h_1^2 \oplus rk_2^2$	BA 4D 1B E5 66 75 2C 78 BD EA D0 8E 0A 05 3E 9C
$h_1^0 := h_0^0 \oplus h_1^0$			18 28 A9 F0 C0 30 60 E5 D3 D9 89 BC 38 66 E3 8F
$h_1^1 := h_0^1 \oplus h_1^1$			71 B3 A0 73 BD 3F 17 31 24 C3 B4 69 F7 29 DD 99
$h_1^2 := h_0^2 \oplus h_1^2$			49 99 09 D0 B8 19 0B 22 C3 95 A9 EF 46 30 6B 5D
$PerF(h_1^0, h_1^1, h_1^2)$ – бөлікшелер арасында өзара орын алмастырудан соң:			
h_1^0			18 71 49 28 B3 99 A9 A0 09 F0 73 D0 C0 BD B8 30
h_1^1			3F 19 60 17 0B E5 31 22 D3 24 C3 D9 C3 95 89 B4
h_1^2			A9 BC 69 EF 38 F7 46 66 29 30 E3 DD 6B 8F 99 5D

Кесте Г.6 – Хештеудің 6-раундтағы жұмыс нәтижелері

Параметрлер			Мәндер
1-бөлікше	$CFKey: rk_2^0$		C7 FC 44 1A 7E C0 CC 25 BA 19 27 42 AC B0 E8 1C
	h_0^0:=h_1^0		18 71 49 28 B3 99 A9 A0 09 F0 73 D0 C0 BD B8 30
	CF:	h_1^0	C5 89 CB 7A 27 CA F7 2E EA 14 13 A0 C5 04 2F 97
		$h_1^0 \coloneqq h_1^0 \oplus rk_2^0$	02 75 8F 60 59 0A 3B 0B 50 0D 34 E2 69 B4 C7 8B
2-бөлікше	$CFKey: rk_2^1$		C5 F3 30 A4 7D B6 BA BD AA 6C 48 B2 00 59 66 2D
	h_0^1:= h_1^1		3F 19 60 17 0B E5 31 22 D3 24 C3 D9 C3 95 89 B4
	CF:	h_1^1	C7 CA 54 49 7A B9 F9 55 6F 35 02 92 63 81 8C 77
		$h_1^1 := h_1^1 \oplus rk_2^1$	02 39 64 ED 07 0F 43 E8 C5 59 4A 20 63 D8 EA 5A
3-бөлікше	$CFKey: rk_2^2$		80 E7 C0 2E C5 3D 19 20 B4 58 68 A1 66 FA EE EE
	$h_0^2 := h_1^2$		A9 BC 69 EF 38 F7 46 66 29 30 E3 DD 6B 8F 99 5D
	CF:	h_1^2	CE EC 0C B4 25 67 AB 1A 04 09 2C 01 BF 43 0D F8
		$h_1^2 := h_1^2 \oplus rk_2^2$	4E 0B CC 9A E0 5A B2 3A B0 51 44 A0 D9 B9 E3 16
$h_1^0 := h_0^0 \oplus h_1^0$			1A 04 C6 48 EA 93 92 AB 59 FD 47 32 A9 09 7F BB
$h_1^1 := h_0^1 \oplus h_1^1$			3D 20 04 FA 0C EA 72 CA 16 7D 89 F9 A0 4D 63 EE
$h_1^2 := h_0^2 \oplus h_1^2$			E7 B7 A5 75 D8 AD F4 5C 99 61 A7 7D B2 36 7A 4B
$PerF(h_1^0, h_1^1, h_1^2)$ – бөлікшелер арасында өзара орын алмастырудан соң:			
h_1^0			1A 3D E7 04 20 B7 C6 04 A5 48 FA 75 EA 0C D8 93
h_1^1			EA AD 92 72 F4 AB CA 5C 59 16 99 FD 7D 61 47 89
h_1^2			A7 32 F9 7D A9 A0 B2 09 4D 36 7F 63 7A BB EE 4B

Кесте Г.7 – Хештеудің 7-раундтағы жұмыс нәтижелері

Параметрлер			Мәндер
1-бөлікше	$CFKey: rk_2^0$		3E 5A E1 23 A4 48 0C 61 0E B1 1A E5 E8 EE 17 63
	h_0^0:=h_1^0		1A 3D E7 04 20 B7 C6 04 A5 48 FA 75 EA 0C D8 93
	CF:	h_1^0	CB D8 71 6C 9C F9 C2 71 09 48 8B 20 B3 2F BA B3
		$h_1^0 \coloneqq h_1^0 \oplus rk_2^0$	F5 82 90 4F 38 B1 CE 10 07 F9 91 C5 5B C1 AD D0
2-бөлікше	$CFKey: rk_2^1$		AD 68 38 24 EC 92 C4 75 7C 4D 51 7F 8F 39 0C 89
	h_0^1:= h_1^1		EA AD 92 72 F4 AB CA 5C 59 16 99 FD 7D 61 47 89
	CF:	h_1^1	A7 98 C5 2B 52 D2 78 A4 AB 26 53 98 07 6E 8B CE
		$h_1^1 := h_1^1 \oplus rk_2^1$	0A F0 FD 0F BE 40 BC D1 D7 6B 02 E7 88 57 87 47
3-бөлікше	$CFKey: rk_2^2$		15 7A EE 34 E4 60 62 18 92 6F 9A 85 2B 7D 9B F0
	$h_0^2 := h_1^2$		A7 32 F9 7D A9 A0 B2 09 4D 36 7F 63 7A BB EE 4B
	CF:	h_1^2	E9 6D 36 2E B6 F4 88 29 01 85 D3 D3 AB 60 31 2B
		$h_1^2 := h_1^2 \oplus rk_2^2$	FC 17 D8 1A 52 94 EA 31 93 EA 49 56 80 1D AA DB
$h_1^0 := h_0^0 \oplus h_1^0$			EF BF 77 4B 18 06 08 14 A2 B1 6B B0 B1 CD 75 43
$h_1^1 := h_0^1 \oplus h_1^1$			E0 5D 6F 7D 4A EB 76 8D 8E 7D 9B 1A F5 36 C0 CE
$h_1^2 := h_0^2 \oplus h_1^2$			5B 25 21 67 FB 34 58 38 DE DC 36 35 FA A6 44 90
$PerF(h_1^0, h_1^1, h_1^2)$ – бөлікшелер арасында өзара орын алмастырудан соң:			
h_1^0			EF E0 5B BF 5D 25 77 6F 21 4B 7D 67 18 4A FB 06
h_1^1			EB 34 08 76 58 14 8D 38 A2 8E DE B1 7D DC 6B 9B
h_1^2			36 B0 1A 35 B1 F5 FA CD 36 A6 75 C0 44 43 CE 90

Кесте Г.8 – Хештеудің 8-раундтағы жұмыс нәтижелері

Параметрлер			Мәндер
1-бөлікше	$CFKey: rk_2^0$		84 2D EF 84 FD CA 7D ED A2 5C 61 13 3F F9 F5 CB
	h_0^0:=h_1^0		EF E0 5B BF 5D 25 77 6F 21 4B 7D 67 18 4A FB 06
	CF:	h_1^0	B7 FA 97 66 3A 77 AF 14 04 5B 64 D7 D0 A7 F0 41
		$h_1^0 \coloneqq h_1^0 \oplus rk_2^0$	33 D7 78 E2 C7 BD D2 F9 A6 07 05 C4 EF 5E 05 8A
2-бөлікше	$CFKey: rk_2^1$		55 44 D5 25 5E 1F 0D 1F 01 03 F0 18 EC DD 96 E8
	h_0^1:= h_1^1		EB 34 08 76 58 14 8D 38 A2 8E DE B1 7D DC 6B 9B
	CF:	h_1^1	F5 17 81 90 E6 92 B1 09 DC A8 BD 76 61 A8 41 B4
		$h_1^1 := h_1^1 \oplus rk_2^1$	A0 53 54 B5 B8 8D BC 16 DD AB 4D 6E 8D 75 D7 5C
3-бөлікше	$CFKey: rk_2^2$		1E 53 2E 25 B1 1D 8A 3D 3F 2C 00 86 F3 22 3A 41
	$h_0^2 := h_1^2$		36 B0 1A 35 B1 F5 FA CD 36 A6 75 C0 44 43 CE 90
	CF:	h_1^2	46 B5 B2 80 C4 36 DB 10 B6 3F FC EF F4 7E D2 C7
		$h_1^2 := h_1^2 \oplus rk_2^2$	58 E6 9C A5 75 2B 51 2D 89 13 FC 69 07 5C E8 86
$h_1^0 := h_0^0 \oplus h_1^0$			DC 37 23 5D 9A 98 A5 96 87 4C 78 A3 F7 14 FE 8C
$h_1^1 := h_0^1 \oplus h_1^1$			4B 67 5C C3 E0 99 31 2E 7F 25 93 DF F0 A9 BC C7
$h_1^2 := h_0^2 \oplus h_1^2$			6E 56 86 90 C4 DE AB E0 BF B5 89 A9 43 1F 26 16
$PerF(h_1^0, h_1^1, h_1^2)$ – бөлікшелер арасында өзара орын алмастырудан соң:			
h_1^0			DC 4B 6E 37 67 56 23 5C 86 5D C3 90 9A E0 C4 98
h_1^1			99 DE A5 31 AB 96 2E E0 87 7F BF 4C 25 B5 78 93
h_1^2			89 A3 DF A9 F7 F0 43 14 A9 1F FE BC 26 8C C7 16
$ComF(h_1^0, h_1^1, h_1^2)$ ақырғы хеш-код алу:			
Хеш-код (256 бит): h			DC 4B 6E 37 67 56 23 5C 86 5D C3 90 9A E0 C4 98 99 DE A5 31 AB 96 2E E0 87 7F BF 4C 25 B5 78 93

Нәтиже. *H(M*={Republic of Kazakhstan}) = (DC 4B 6E 37 67 56 23 5C 86 5D C3 90 9A E0 C4 98 99 DE A5 31 AB 96 2E E0 87 7F BF 4C 25 B5 78 93)$_{16}$**.**

ҚОСЫМША Д

Сызықтық криптоталдау теңдеулері

Сызықтық криптоталдауда құрылған сызықтық теңдеулер жүйесі:

S_0 блок үшін теңдеулер:

$$x_4 \oplus y_2 \oplus y_3 = 1$$
$$x_4 \oplus y_1 \oplus y_2 \oplus y_3 = 0$$
$$x_3 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_3 \oplus y_1 \oplus y_2 \oplus y_3 = 0$$
$$x_3 \oplus x_4 \oplus y_4 = 0$$
$$x_3 \oplus x_4 \oplus y_1 \oplus y_3 \oplus y_4 = 0$$
$$x_2 \oplus y_1 \oplus y_4 = 0$$
$$x_2 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_2 \oplus x_4 \oplus y_1 \oplus y_3 \oplus y_4 = 0$$
$$x_2 \oplus x_4 \oplus y_1 \oplus y_2 \oplus y_4 = 0$$
$$x_2 \oplus x_3 \oplus y_2 \oplus y_3 \oplus y_4 = 0$$
$$x_2 \oplus x_3 \oplus y_1 \oplus y_2 = 0$$
$$x_2 \oplus x_3 \oplus x_4 \oplus y_3 = 0$$
$$x_2 \oplus x_3 \oplus x_4 \oplus y_1 \oplus y_2 = 1$$
$$x_1 \oplus y_3 \oplus y_4 = 0$$
$$x_1 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 0$$
$$x_1 \oplus x_4 \oplus y_2 \oplus y_4 = 1$$
$$x_1 \oplus x_4 \oplus y_1 \oplus y_2 \oplus y_3 = 0$$
$$x_1 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_4 = 0$$
$$x_1 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_3 = 1$$
$$x_1 \oplus x_3 \oplus x_4 \oplus y_1 = 0$$
$$x_1 \oplus x_3 \oplus x_4 \oplus y_1 \oplus y_4 = 1$$
$$x_1 \oplus x_2 \oplus y_2 \oplus y_4 = 0$$
$$x_1 \oplus x_2 \oplus y_1 \oplus y_3 = 0$$
$$x_1 \oplus x_2 \oplus x_4 \oplus y_2 = 0$$
$$x_1 \oplus x_2 \oplus x_4 \oplus y_1 \oplus y_3 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_3 = 1$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_1 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_4 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_2 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_2 \oplus y_3 = 0.$$

S_1 блок үшін теңдеулер:

$$x_4 \oplus y_1 \oplus y_4 = 0$$
$$x_4 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 0$$
$$x_3 \oplus y_1 \oplus y_2 \oplus y_4 = 0$$
$$x_3 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_3 \oplus x_4 \oplus y_2 \oplus y_4 = 0$$
$$x_3 \oplus x_4 \oplus y_2 \oplus y_3 = 1$$
$$x_2 \oplus y_1 \oplus y_3 \oplus y_4 = 1$$
$$x_2 \oplus y_1 \oplus y_2 \oplus y_3 = 1$$
$$x_2 \oplus x_4 \oplus y_3 = 1$$
$$x_2 \oplus x_4 \oplus y_1 \oplus y_2 \oplus y_3 = 0$$

$$x_2 \oplus x_3 \oplus y_2 \oplus y_3 = 1$$
$$x_2 \oplus x_3 \oplus y_1 \oplus y_3 \oplus y_4 = 0$$
$$x_2 \oplus x_3 \oplus x_4 \oplus y_4 = 1$$
$$x_2 \oplus x_3 \oplus x_4 \oplus y_2 = 0$$
$$x_1 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus x_4 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus x_4 \oplus y_1 \oplus y_3 = 1$$
$$x_1 \oplus x_3 \oplus y_1 \oplus y_3 = 1$$
$$x_1 \oplus x_3 \oplus y_1 \oplus y_2 = 0$$
$$x_1 \oplus x_3 \oplus x_4 \oplus y_2 \oplus y_3 = 1$$
$$x_1 \oplus x_3 \oplus x_4 \oplus y_1 = 0$$
$$x_1 \oplus x_2 \oplus y_2 \oplus y_4 = 0$$
$$x_1 \oplus x_2 \oplus y_1 \oplus y_2 = 0$$
$$x_1 \oplus x_2 \oplus x_4 \oplus y_2 = 0$$
$$x_1 \oplus x_2 \oplus x_4 \oplus y_1 \oplus y_2 \oplus y_4 = 1$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_4 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_2 \oplus y_3 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_3 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_1 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_1 \oplus y_4 = 0.$$

S_3 блок үшін теңдеулер:

$$x_4 \oplus y_2 \oplus y_3 = 1$$
$$x_4 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_3 \oplus y_1 \oplus y_3 \oplus y_4 = 1$$
$$x_3 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_3 \oplus x_4 \oplus y_3 \oplus y_4 = 0$$
$$x_3 \oplus x_4 \oplus y_1 \oplus y_4 = 0$$
$$x_2 \oplus y_1 \oplus y_2 \oplus y_4 = 0$$
$$x_2 \oplus y_1 \oplus y_2 \oplus y_3 = 0$$
$$x_2 \oplus x_4 \oplus y_4 = 1$$
$$x_2 \oplus x_4 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_2 \oplus x_3 \oplus y_3 = 1$$
$$x_2 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 0$$
$$x_2 \oplus x_3 \oplus x_4 \oplus y_2 = 0$$
$$x_2 \oplus x_3 \oplus x_4 \oplus y_1 \oplus y_2 = 1$$
$$x_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus y_1 \oplus y_2 \oplus y_4 = 0$$
$$x_1 \oplus x_4 \oplus y_2 \oplus y_3 \oplus y_4 = 0$$
$$x_1 \oplus x_4 \oplus y_1 = 0$$
$$x_1 \oplus x_3 \oplus y_1 \oplus y_2 = 0$$
$$x_1 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_3 = 0$$
$$x_1 \oplus x_3 \oplus x_4 \oplus y_2 \oplus y_4 = 0$$
$$x_1 \oplus x_3 \oplus x_4 \oplus y_1 \oplus y_3 = 1$$
$$x_1 \oplus x_2 \oplus y_1 \oplus y_4 = 1$$
$$x_1 \oplus x_2 \oplus y_1 \oplus y_3 \oplus y_4 = 0$$
$$x_1 \oplus x_2 \oplus x_4 \oplus y_2 \oplus y_3 = 0$$
$$x_1 \oplus x_2 \oplus x_4 \oplus y_1 \oplus y_3 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_3 = 0$$

$$x_1 \oplus x_2 \oplus x_3 \oplus y_3 \oplus y_4 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_2 = 1$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_2 \oplus y_4 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_4 = 1$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_1 = 0.$$

S_4 блок үшін теңдеулер:

$$x_4 \oplus y_1 \oplus y_4 = 1$$
$$x_4 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_3 \oplus y_1 \oplus y_2 \oplus y_3 = 1$$
$$x_3 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 0$$
$$x_3 \oplus x_4 \oplus y_2 \oplus y_3 = 1$$
$$x_3 \oplus x_4 \oplus y_1 \oplus y_3 = 1$$
$$x_2 \oplus y_1 \oplus y_3 \oplus y_4 = 0$$
$$x_2 \oplus y_1 \oplus y_2 \oplus y_4 = 1$$
$$x_2 \oplus x_4 \oplus y_2 \oplus y_4 = 1$$
$$x_2 \oplus x_4 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_2 \oplus x_4 \oplus y_1 \oplus y_2 = 0$$
$$x_2 \oplus x_4 \oplus y_1 \oplus y_2 \oplus y_3 = 1$$
$$x_2 \oplus x_3 \oplus y_3 \oplus y_4 = 0$$
$$x_2 \oplus x_3 \oplus y_2 \oplus y_3 = 1$$
$$x_2 \oplus x_3 \oplus x_4 \oplus y_4 = 1$$
$$x_2 \oplus x_3 \oplus x_4 \oplus y_1 \oplus y_4 = 0$$
$$x_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus y_1 \oplus y_2 \oplus y_4 = 1$$
$$x_1 \oplus x_4 \oplus y_3 = 0$$
$$x_1 \oplus x_4 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus x_3 \oplus y_1 = 1$$
$$x_1 \oplus x_3 \oplus y_1 \oplus y_2 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus x_3 \oplus x_4 \oplus y_4 = 0$$
$$x_1 \oplus x_3 \oplus x_4 \oplus y_2 \oplus y_4 = 1$$
$$x_1 \oplus x_2 \oplus y_3 \oplus y_4 = 1$$
$$x_1 \oplus x_2 \oplus y_1 \oplus y_2 = 1$$
$$x_1 \oplus x_2 \oplus x_4 \oplus y_3 = 0$$
$$x_1 \oplus x_2 \oplus x_4 \oplus y_1 \oplus y_3 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_2 = 1$$
$$x_1 \oplus x_2 \oplus x_3 \oplus y_1 \oplus y_3 \oplus y_4 = 0$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_2 = 1$$
$$x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus y_1 = 0.$$

x_i – S-блокқа кіріс мәндер, y_i – S-блоктан шығыс мәндер, $i = \overline{1,4}$.