МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
СУМСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ
КАФЕДРА ЕЛЕКТРОНІКИ І КОМП'ЮТЕРНОЇ ТЕХНІКИ

ПОЯСНЮВАЛЬНА ЗАПИСКА

до кваліфікаційної роботи бакалавра на тему:

«Пристрій захисту інформації на основі реалізації алгоритму Ель-Гамаля»

Завідувач кафедри **А.С. Опанасюк**

Керівник кваліфікаційної роботи **Т.О. Протасова**

Виконав студент гр. ТК-61 **М.А. Клещев**

Суми 2020 р.

Сумський Державний Університет

Факультет ЕлІТ
Кафедра електроніки і комп'ютерної техніки
Спеціальність 172 "Телкомунікації та радіотехніка"

ЗАТВЕРДЖУЮ**:**
Зав. кафедри Опанасюк А.С.
«___» _____ 20___ р.

ЗАВДАННЯ
на кваліфікаційну роботу бакалавра
студенту **Клещеву Миколі Андрійовичу**

1. Тема роботи: «Пристрій захисту інформації на основі реалізації алгоритму Ель-Гамаля»
затверджено наказом по кафедрі від «21» ___Квітня_____ 20 20 р. № _№0544/III
2. Термін здачі студентом закінченої роботи 5.06.2020 р.
3. Вихідні дані до роботи 1. Функціональна схема пристрою. 2. Алгорим генерації ключів. 3. Алгоритм шифрування. 4. Алгоритм дешифрування. 5.Структурна схема

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці) 1. Розглянути основні принципи інформаційної безпеки. 2. Розглянути методи захисту інформації. 3. Розглянути недоліки алгоритму. 4. Розробити алгоритми генерації ключів, шифрування та дешифрування. 5. Розробити функціональну схему пристрою захисту інформації. 6. Оцінити актуальність та доцільність реалізації даного пристрою.
5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
1. Презентація з 13 слайдів. 2. Алгоритм генерації ключів. 3. Алгоритм шифрування. 4. Алгоритм дешифрування. 5. Структурна схема пристрою шифрування. 6. Функціональна схема пристрою шифрування.

Дата видачі завдання: 10.03.2020 р.
Завдання прийняв до виконання: _____ Клещев М.А.

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Назва етапів кваліфікаційної роботи	Термін виконання етапів кваліфікаційної роботи	Примітка
1	Огляд літератури відповідно до теми дослідження	25.03.2020	
2	Аналіз виявленої літератури	10.04.2020	
3	Опис алгоритму роботи методу Ель-Гамаля	25.04.2020	
4	Розробка алгоримів генерації ключів, шифрування та дешифрування	30.04.2020	
5	Аналіз недоліків методу	10.05.2020	
6	Розробка функціональної та структурної схеми пристрою шифрування методом Ель-Гамаля	17.05.2020	
7	Структуризація всього матеріалу та оформлення кваліфікаційної роботи	24.05.2020	
8	Представлення кваліфікаційної роботи для захисту	10.06.2020	

Студент Клещев М.А.

Керівник кваліфікаційної роботи Протасова А.І.

«___» _____ 2020 р

РЕФЕРАТ

У кваліфікаційній роботі бакалавра спроектований пристрій захисту інформації методом Ель-Гамаля.

Кваліфікаційна робота бакалавра складається з п’яти розділів, містить 48 сторінок тексту, 12 рисунків, 3 таблиці, графічний матеріал у вигляді презентації з 13 слайдів, алгоримів генерації ключів, шифрування та дешифрування, структурної та функціональної схеми пристрою шифрування.

В першому розділі розглянуто основи інформаційної безпеки а саме її категорії та моделі захисту інформації.

У другому розділі детально присвячений огляду класифікації криптоалгоритмів та розглянуто різновид симетричних криптоалгоритмів.

У третьому розділі побудовано алгоритми для генерації відкритих ключів, шифрування повідомлення та дешифрування повідомлення.

У четвертому розділі присвячений складність апаратної реалізації методів захисту інформації з відкритим ключем.

П’ятый розділ присвячений побудові функціональну схему пристрою шифрування на основі методу Ель Гамаля та детальному опису її принципів функціонування.

У кінці пояснювальної записки зроблені висновки та приведений перелік літературних джерел.

Кількість літературних джерел – 8.

ЗМІСТ

ВСТУП

У сучасному комп'ютерному співтоваристві атаки на інформацію стали повсякденною практикою. Зловмисники використовують як помилки в написанні і адмініструванні програм, так і методи соціальної психології для отримання бажаної інформації. Криптографія - наука про способи двонаправленого перетворення інформації з метою конфіденційної передачі її по незахищеному каналу між двома станціями, розділеними в просторі і/або часу. Криптографія, використовуючи досягнення в першу чергу математики, дозволяє модифікувати дані таким чином, що ніякі найсучасніші ЕОМ за певний період часу не можуть відновити вихідний текст, відомий тільки відправнику і одержувачу.

Останнім часом повідомлення про атаки на інформацію, про хакерів і комп'ютерних зловмисників наповнили всі засоби масової інформації. Дати визначення цього дійства насправді дуже складно, оскільки інформація, особливо в електронному вигляді, представлена сотнями різних видів. Інформацією можна вважати і окремий файл, і базу даних, і один запис в ній, і цілком програмний комплекс. І всі ці об'єкти можуть піддатися і піддаються атакам з боку деякої соціальної групи осіб.

При зберіганні, підтримці і наданні доступу до будь-якого інформаційного об'єкту його власник, або уповноважена ним особа, накладає за собою явний або самоочевидний набір правил по роботі з нею. Умисне їх порушення класифікується як атака на інформацію.

З масовим впровадженням комп'ютерів в усі сфери діяльності людини обсяг інформації, що зберігається в електронному вигляді виріс у мільйони разів. І тепер скопіювати за півхвилини і віднести флеш-накопичувач з файлом, що містить план випуску продукції, набагато простіше, ніж копіювати або переписувати стос паперів. А з появою комп'ютерних мереж навіть відсутність фізичного доступу до комп'ютера перестало бути гарантією збереження інформації.

Які можливі наслідки атак на інформацію? В першу чергу, звичайно, багатьох будуть цікавити економічні втрати:

— Розкриття певної інформації, що становить комерційну таємницю може привести до серйозних прямих збитків для компаній та окремих приватних осіб;

— Звістка про крадіжку великого обсягу інформації зазвичай серйозно впливає на репутацію будь-якої компанії, або приватної особи, приводячи побічно до втрат грошових коштів, та спрямовані на погіршення різних ситуацій;
— Фірми-конкуренти можуть скористатися крадіжкою інформації, якщо та залишилася непоміченою, для того щоб повністю розорити фірму, нав'язуючи їй фіктивні або завідомо збиткові угоди;\
— Підміна інформації як на етапі передачі, так і на етапі зберігання в фірмі може привести до величезних збитків;
— Багаторазові успішні атаки на фірму, яка надає будь-якої вид інформаційних послуг, знижують довіру до фірми у клієнтів, що позначається на обсязі доходів;
— Природно, комп'ютерні атаки можуть принести і величезний моральний збиток.

Криптографія - наука про захист доступу до інформації сторонніх осіб або організацій. Захист досягається шифруванням, тобто перетворенням, які роблять захищеними вхідні дані які неможливо зчитати, скопіювати, прочитати або знищити без знання спеціальної ключової інформації - ключа. Під ключем розуміється легко змінна частина криптосистеми, що зберігається в таємниці і визначає, яке шифрувальні перетворення з усіх можливих виконується в даному випадку. Криптосистема – система вибирає за допомогою ключа оборотних перетворень, які перетворять захищається відкритий текст в шифрограму і назад.

Бажано, щоб методи шифрування мали мінімум дві властивості:
— компанія, або приватна особа може виконати дешифрування, і отримати інформацію у первинному виді;
— криптоаналітичні пристрої зловмисника, яких може перехопити повідомлення, не зможуть зробити зворотне дешифрування для отримання доступу до інформації.

1 ОСНОВИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

1.1 Категорії інформаційної безпеки

Інформація з точки зору інформаційної безпеки має такі складові:

— конфіденційність - гарантія того, що конкретна інформація доступна тільки тим особам, як фізичним, так і юридичним, для яких вона призначена; порушення цієї категорії називається розкраданням або розкриттям інформації.

— цілісність - гарантія того, що інформація зараз існує в її сталому та незмінному вигляді, тобто при її зберіганні або передачі не було проведено ніяких несанкціонованих змін або втручань; порушення цієї категорії називається порушенням цілісності блоку інформації.

— автентичність - гарантія того, що джерелом інформації є саме та особа, яка заявлена як її автор та власник; порушення цієї категорії також називається фальсифікацією, але вже автора повідомлення.

— апельованість - досить складна категорія, але часто застосовується в електронній комерції - гарантія того, що при необхідності можна буде довести, що автором інформації є саме заявлена людина, і не може бути ніхто інший; відмінність цієї категорії від попередньої в тому, що при підміні автора, хтось інший намагається заявити, що він автор даної інформації, а при порушенні апельованності - сам автор намагається заперечувати свої права автентичності, підписані ним одного разу.

Відносно інформаційних систем застосовуються інші категорії:

— надійність - гарантія того, що система працює в нормальному і позаштатному режимах так, що її ефективність не менше 85%.

— точність - гарантія точного і повного дотримання всіх визначених інструкцій.

— контроль доступу - гарантія того, що різні групи осіб мають різний доступ до інформаційних об'єктів, і ці обмеження доступу непохитно виконуються.

— контрольованість - гарантія того, що в будь-який момент може бути проведена повноцінна перевірка будь-якого компонента програмного комплексу.

— контроль ідентифікації - гарантія того, що клієнт, підключений в даний момент до системи, є саме тим, за кого себе видає.

— стійкість до навмисних збоїв - гарантія того, що при навмисному внесенні помилок в межах заздалегідь обговорених норм система буде вести себе так, як обумовлено заздалегідь.

1.2 Абстрактні моделі захисту інформації

Однією з перших моделей була опублікована у 1977 модель Біба (Biba). Відповідно до неї всі суб'єкти і об'єкти попередньо поділяються за кількома рівнями доступу, а потім на їх взаємодії накладаються наступні обмеження:

1. Суб'єкт не може викликати на виконання суб'єкти з більш низьким рівнем доступу.
2. Суб'єкт не може модифікувати об'єкти з більш високим рівнем доступу. Як бачимо, ця модель дуже нагадує обмеження, введені в захищеному режимі мікропроцесорів Intel 80386+ щодо рівнів привілеїв.

Модель Гогена-Мезігера (Goguen-Meseguer), представлена ними в 1982 році, заснована на теорії автоматів. Відповідно до неї будь-яка система може при кожній дії переходити з одного дозволеного стану тільки в кілька інших. Суб'єкти і об'єкти в даній моделі захисту розбиваються на групи - домени, і перехід системи з одного стану в інший виконується тільки відповідно до так званої таблицею дозволів, в якій вказано які операції може виконувати суб'єкт, скажімо, з домена C над об'єктом з домену D. У даній моделі при переході системи з одного дозволеного стану в інше використовуються транзакції, що забезпечує загальну цілісність системи.

Сазерлендська (від англ. Sutherland) модель захисту, опублікована в 1986 році, робить акцент на взаємодії суб'єктів і потоків інформації. Так само як і в попередній моделі, тут використовується класифікація станів з безліччю дозволених комбінацій станів і деяким набором початкових позицій. У даній моделі досліджується поведінка множинних композицій функції переходу з одного стану в інший.

Важливу роль в теорії захисту інформації грає модель захисту Кларка-Вільсона (Clark-Wilson), опублікована в 1987 році і модифікована в 1989 році. Спрямована дана модель на повсюдне використання транзакцій і ретельному оформленні прав доступу суб'єктів до об'єктів. Але в даній моделі вперше досліджена захищеність третьої сторони в даній проблемі - сторони, що підтримує всю систему безпеки. Цю роль в інформаційних системах зазвичай грає програма-супервізор. Крім того, в моделі Кларка-Вільсона транзакції вперше були побудовані за методом верифікації, тобто

ідентифікація суб'єкта проводилася не тільки перед виконанням команди від нього, але і повторно після виконання. Це дозволило зняти проблему підміни учасника в момент між його ідентифікацією і власне командою. Модель Кларка-Вільсона вважається однією з найдосконаліших у відношенні підтримки цілісності інформаційних систем.

1.3 Огляд найбільш поширених методів "зламу"

Зловмисники досить ретельно вивчають систему перед проникненням в неї. Дуже часто вони знаходять очевидні і дуже прості методи "зламу" системи, які просто пропустили розробники які створювали її, створюючи можливо дуже хорошу систему ідентифікації або шифрування.

— *Термінали захищеної інформаційної системи*

Термінали - це точки входу користувача в інформаційну мережу. У тому випадку, коли до них мають доступ кілька людей або взагалі будь-який бажаючий, при їх проектуванні та експлуатації необхідно ретельне дотримання цілого комплексу заходів безпеки.

— *Отримання пароля на основі помилок адміністратора і користувачів*

Подальші дії зловмисника, який отримав доступ до термінальної точці входу, можуть розвиватися за двома основними напрямками: а) спроби з'ясування пароля напряму або обхідними шляхами; б) спроби входу в систему абсолютно без знання пароля, ґрунтуючись на помилках допущених при розробці програмного або апаратного забезпечення.

— *Соціальна психологія і інші способи отримання ключа*

Іноді зловмисники вступають і в прямий контакт з особами, що володіють потрібною їм інформацією, розігруючи досить переконливі сцени. "Жертва" обману, що повірила в реальність розказаної їй по телефону або в електронному листі ситуації, сама повідомляє пароль зловмисникові.

— Термінали захищеної інформаційної системи як такі

Незважаючи на самоочевидність, все-таки найбільш поширеним способом входу в систему при атаках на інформацію залишається вхід через офіційний log-in запит системи. Обчислювальна техніка, яка дозволяє зробити вхід в систему, називається в теорії інформаційної безпеки терміналом. Термінологія сягає часів суперЕОМ і тонких "термінальних" клієнтів. Якщо система складається всього з

одного персонального комп'ютера, то він одночасно вважається і терміналом і сервером. Доступ до терміналу може бути фізичним, в тому випадку, коли термінал - це ЕОМ з клавіатурою і дисплеєм, або віддаленим - найчастіше по телефонній лінії (в цьому випадку терміналом є модем, підключений або безпосередньо до системи, або до її фізичного терміналу).

При використанні терміналів з фізичним доступом необхідно дотримуватися таких вимог:

- Захищеність терміналу повинна відповідати захищеності приміщення: термінали без пароля можуть бути присутніми тільки в тих приміщеннях, куди мають доступ особи відповідного або вищого рівня доступу;
- відсутність імені реєстрації можливо тільки в тому випадку, якщо до терміналу має доступ тільки одна людина, або якщо на групу осіб, які мають до нього доступ, поширюються загальні заходи відповідальності;
- термінали, встановлені в публічних місцях повинні завжди запитувати ім'я реєстрації і пароль.

Системи контролю за доступом в приміщення з встановленим терміналом повинні працювати повноцінно і відповідно до загальної схемою доступу до інформації.

У разі установки терміналу в місцях з широким скупченням народу клавіатура, а якщо необхідно, то і дисплей повинні бути обладнані пристроями, що дозволяють бачити їх тільки працює в даний момент клієнтові (непрозорі скляні або пластмасові огорожі, шторки, "утоплена" модель клавіатури).

При використанні віддалених терміналів необхідно дотримуватися таких правил:

- Будь віддалений термінал повинен запитувати ім'я користувача і пароль. Того, що нібито ніхто не знає шестизначного номера вашого службового модему, аж ніяк не достатньо для конфіденційності вашої системи. Вся справа в тому, що при наявності програмного забезпечення, яке не важко буде знайти в мережі Інтернет, і тонового набору для одного дзвінка достатньо 4 секунд. Це означає, що за 1 хвилину можна перебрати близько 15 номерів телефонної станції з тим, щоб дізнатися чи існує на цьому телефонний номер модем. За годину таким чином можна перебрати 1000 номерів, а за робочий день з повтором в нічний час (це стандартна методика) - всю АТС (10.000 номерів).

— Своєчасне відключення всіх модемів, які не потрібні в даний момент фірмі (наприклад, вечорами, або під час обідньої перерви), або не контрольованих в даний момент Вашими співробітниками.

По можливості рекомендується використовувати схему зворотного дзвінка від модему, оскільки вона гарантує з рівнем надійності АТС то, що віддалений клієнт отримав доступ з певного телефонного номера.

З log-in запиту терміналу рекомендується прибрати всі безпосередні згадки імені фірми, її логотипи і т.п. - це не дозволить комп'ютерним вандалам, просто перебирати номери з модемами, дізнатися log-in екран якої фірми вони виявили. Для перевірки правильності з'єднання замість імені фірми можна використовувати неординарну вітальну фразу, якої-небудь афоризм або просто фіксовану послідовність літер і цифр, які будуть запам'ятовуватися у постійних операторів цього терміналу.

Також на вході в систему рекомендується виводити на екран попередження про те, що вхід в систему без повноважень на це переслідується по закону. По-перше, це послужить ще одним застереженням початківцям зловмисникам, а по-друге, буде надійним аргументом на користь атакований фірми в судовому розгляді, якщо таке буде проводитися.

Безсумнівно від фізичного або комутованого доступу до терміналу, лінія, яка з’єднує термінал (комутований, або встановлений в публічному місці) із зоною ядра інформаційної системи повинна бути захищена від прослуховування, або ж весь обмін інформацією повинен вестися по конфіденційної схемою ідентифікації та надійній схемі аутентифікації клієнта – цим займаються криптосистеми.

1.3.1 Отримання пароля на основі помилок адміністратора і користувачів. Перебір паролів по словнику був деякий час однієї з найпоширеніших технік підбору паролів. В даний час, як хоч найменший результат пропаганди інформаційної безпеки, він став здавати свої позиції. Хоча розвиток швидкодії обчислювальної техніки і все більш складні алгоритми складання слів-паролів не дають "загинути" цього методу.

Технологія перебору паролів народилася в той час, коли найскладнішим паролем було скажімо слово "brilliant", а в русифікованих ЕОМ воно ж, але для "хитрощів" набране в латинському режимі, але дивлячись на російські літери (ця тактика на жаль досі надзвичайно поширена, хоча і збільшує інформаційну насиченість пароля всього на 1 біт). У той час простенька програма зі словником в

5000 іменників давала позитивний результат в 60% випадків [4]. Величезне число інцидентів зі зломами систем змусило користувачів додавати до слів 1-2 цифри з кінця, записувати першу і/або останню букву в верхньому регістрі, але це збільшило час на перебір варіантів з урахуванням зростання швидкодії ЕОМ всього в кілька разів.

Наступною модифікацією підбору паролів є перевірка паролів, які встановлюються в системах за замовчуванням. У деяких випадках адміністратор програмного забезпечення, змонтували або отримавши новий продукт від розробника, не спромагається перевірити, з чого складається система безпеки. Як наслідок, пароль, встановлений в фірмі розробника за умовчанням, залишається основним паролем в системі. У мережі Інтернет можна знайти величезні списки паролів за замовчуванням практично до всіх версій програмного забезпечення, якщо вони встановлюються на ньому виробником.

Основні вимоги до інформаційної безпеки, засновані на аналізі даного методу, такі:

- Вхід всіх користувачів в систему повинен підтверджуватися введенням унікального для клієнта пароля.
- Пароль повинен ретельно підбиратися так, щоб його інформаційна ємність відповідала часу повного перебору пароля. Для цього необхідно детально інструктувати клієнтів про поняття "простий до підбору пароль", або передати операцію вибору пароля у відання інженера з безпеки.
- Паролі за умовчанням повинні бути змінені до офіційного запуску системи і навіть до скільки-небудь публічних випробувань програмного комплексу. Особливо це відноситься до мережевого програмного забезпечення.
- Всі помилкові спроби увійти в систему повинні враховуватися, записуватися в файл журналу подій і аналізуватися через "розумний" проміжок часу.

Якщо в системі передбачена можливість блокування клієнта або всієї системи після певної кількості невдалих спроб входу, цією можливістю необхідно скористатися. Якщо ж Ви є розробником системи безпеки, дану можливість безсумнівно необхідно передбачити, так як вона є основним бар'єром до підбору паролів повним перебором. Розумно блокувати клієнта після 3-ої поспіль неправильної спроби набору пароля, і, відповідно, блокувати систему після K = max (int (N * 0.1 * 3) +1, 3) невдалих спроб входу за деякий період (годину, зміну, добу). У цій формулі N - середня кількість підключаються за цей період до системи клієнтів, 0.1 - 10% -ва межа "забудькуватості пароля", 3 - ті ж самі три спроби на згадування

пароля. Природно, інформація про блокування клієнта або системи повинна автоматично надходити на пульт контролю за системою. У момент відправки пакета підтвердження або відхилення пароля в системі повинна бути встановлена розумна затримка (2-5 секунд). Це не дозволить зловмиснику, потрапивши на лінію з хорошим зв'язком до об'єкта атаки перебирати по сотні тисяч паролів за секунду.

Всі дійсні в системі паролі бажано перевіряти сучасними програмами підбору паролів, або оцінювати особисто адміністратору системи .

Через певні проміжки часу необхідна примусова зміна пароля у клієнтів. Найбільш часто використовуваними інтервалами зміни пароля є рік, місяць і тиждень (в залежності від рівня конфіденційності інформації та частоти входу в систему).

Всі невживані протягом довгого часу імена реєстрації користувачів повинні переводитися в закритий (недоступне для реєстрації) стан. Це відноситься до даних тих співробітників, які перебувають у відпустці, на лікарняному, у відрядженні, а також до імен реєстрації, створеним для тестів, випробувань системи і т.п.

Від співробітників і всіх операторів терміналу необхідно вимагати суворе нерозголошення паролів, відсутність будь-яких взаємозв'язків пароля з широковідомими фактами і даними, і відсутність паперових записів пароля "через погану пам'яті".

Отримання пароля на основі помилок в реалізації

Наступною за частотою використання є методика отримання паролів з самої системи. Однак, тут вже немає можливості дати будь-які загальні рекомендації, оскільки всі методи атаки залежать тільки від програмної і апаратної реалізації конкретної системи. Основними двома можливостями з'ясування пароля є несанкціонований доступ до носія, який містить його, або використання недокументованих можливостей і помилок в реалізації системи.

Перша група методів заснована на тому, що будь-якій системі доводиться де-небудь зберігати оригінали паролів всіх клієнтів для того, щоб звіряти їх в момент реєстрації. При цьому паролі можуть зберігатися як у відкритому текстовому вигляді, як це має місце в багатьох клонах UNIX, так і представлені у вигляді малозначних контрольних сум (хеш-значень), як це реалізовано в ОС Windows, Novell NetWare і багатьох інших. Проблема в тому, що у випадку для зберігання паролів на носії не може бути використана основна методика захисту - шифрування. Дійсно, якщо всі паролі зашифровані будь-яким ключем, то цей ключ теж повинен зберігатися в самій системі для того, щоб вона працювала автоматично, не питаючи кожного разу у

адміністратора дозвіл «Пускати чи не пускати певного користувача». Тому, отримавши доступ до подібної інформації , зловмисник може або відновити пароль в читабельному вигляді (що буває досить рідко), або відправляти запити, підтверджені даними хеш-значенням, що не раскодіруя його. Всі рекомендації щодо запобігання розкрадань паролів складаються в перевірці не доступний файл з паролями, або таблиця в базі даних, що зберігає ці паролі, кому-небудь ще крім адміністраторів системи, чи не створюється системою резервних файлів, в місцях доступних іншим користувачам і т.п.. В принципі, оскільки крадіжка паролів є самим грубим вторгненням в систему, розробники приділяють їй досить пильну увагу, і дотримання всіх рекомендацій по використанню системи зазвичай досить для запобігання подібних ситуацій.

Отримання доступу до паролів завдяки недокументованим можливостям систем зустрічається в даний час вкрай рідко. Раніше ця методика використовувалася розробниками набагато частіше в основному з метою налагодження, або для екстреного відновлення працездатності системи.

Наступною поширеною технологією отримання паролів є копіювання буфера клавіатури в момент набору пароля на терміналі. Цей метод використовується рідко, так для нього необхідний доступ до термінальної машині з можливістю запуску програм. Але якщо зловмисник все-таки отримує подібний доступ, дієвість даного методу дуже висока:

Робота програми-перехоплювача паролів (так званого "троянського коня") на робочій станції непомітна.

Подібна програма сама може відправляти результати роботи на заздалегідь задані сервера або анонімним користувачам, що різко спрощує саму процедуру отримання паролів хакером, і ускладнює пошук і доказ його провини. У нас в Україні, наприклад, широке поширення набула подібна троянська програма, підписувати до архівів.

Основними методами боротьби з копіюванням паролів є адекватний захист робочих станцій від запуску сторонніх програм:

a) відключення змінних носіїв інформації (дискети).
b) спеціальні драйвера, що блокують запуск здійсненних файлів без відома оператора, або адміністратора.
c) монітори, що повідомляють про будь-які зміни системних налаштувань і списку автоматично запускаються, дуже потужна, але незручна міра - система

одноразових паролів (кожного разу при входженні в системі клієнтам з дуже високим рівнем відповідальності самою системою генерується новий пароль) .

Сканування сучасними антивірусними програмами також може допомогти у виявленні "троянських" програм, але тільки тих з них, які набули широкого поширення по країні. А отже, програми, написані зловмисниками спеціально для атаки на Вашу систему, будуть пропущені антивірусними програмами без будь-яких сигналів.

Наступний метод отримання паролів відноситься тільки до мережевого програмного забезпечення. Проблема полягає в тому, що в багатьох програмах не враховується можливість перехоплення будь-якої інформації, що йде по мережі - так званого мережевого трафіку.

Спочатку, з впровадженням локальних комп'ютерних мереж так воно і було. Мережа розташовувалася в межах 2-3 кабінетів, або будівлі з обмеженим фізичним доступом до кабелів. Однак, стрімкий розвиток глобальних мереж зажадало на загальний ринок ті ж версії програмного забезпечення без будь-яких зволікань для посилення безпеки. Тепер ми пожинаємо плоди цієї тенденції. Більше половини протоколів мережі Інтернет передають паролі в нешифрованому вигляді - відкритим текстом. До них відносяться протоколи передачі електронної пошти SMTP і POP3, протокол передачі файлів FTP, одна зі схем авторизації на WWW-серверах.

Сучасне апаратне і програмне забезпечення дозволяє отримувати всю інформацію, що проходить по сегменту мережі, до якого підключений конкретний комп'ютер, і аналізувати її в реальному масштабі часу. Можливі кілька варіантів прослуховування трафіку: 1) це може зробити службовець компанії зі свого робочого комп'ютера, 2) зловмисник, який підключився до сегменту за допомогою портативної ЕОМ або більш мобільного пристрою. Нарешті, трафік, що йде від Вас до Вашого партнеру або в інший офіс по мережі Інтернет, технічно може прослуховуватися з боку Вашого безпосереднього провайдера, з боку будь-якої організації, що надає транспортні послуги для мережі Інтернет (листування всередині країни в середньому йде через 3-4 компанії , за межі країни - через 5-8). Крім того, якщо в належній мірі буде реалізовуватися план СОРМ (система оперативно-розшукових заходів в комп'ютерних мережах), то можливе прослуховування і з боку силових відомств країни.

Для комплексного захисту від подібної можливості крадіжки паролів необхідно виконувати наступні заходи:

— Фізичний доступ до мережевих кабелів повинен відповідати рівню доступу до інформації.
— При визначенні топології мережі слід за будь-яких можливостях уникати широкомовних топологій.
— Оптимальною одиницею сегментації є група операторів з рівними правами доступу, або якщо ця група становить більше 10 осіб, то кімната або відділ всередині групи. Ні в якому разі на одному кабелі не повинні знаходитися оператори з різними рівнями доступу, якщо тільки весь переданий трафік не шифрується, а ідентифікація не виробляється по прихованої схемою без відкритої передачі пароля.
— До всіх інформаційних потоків, які виходять за межі фірми, повинні застосовуватися ті ж правила, що і тільки що описані вище для об'єднання різнорівневих терміналів.

Будова файлів, їх заголовки і розташування в будь-якій операційній системі може бути прочитано при використанні відповідного програмного забезпечення. Дана проблема вирішується набагато простіше і дешевше - за допомогою криптографії.

2 КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ

2.1 Класифікація криптоалгоритмів

Залежно від наявності або відсутності ключа кодують алгоритми діляться на тайнопис і криптографію. Залежно від відповідності ключів шифрування і дешифрування - на симетричні і асиметричні. Залежно від типу використовуваних перетворень - на підстановочні і перестановки. Залежно від розміру шифруемого блоку - на потокові та блокові шифри.

Відносно криптоалгоритмів існує кілька схем класифікації, кожна з яких заснована на групі характерних ознак. Таким чином, один і той же алгоритм "проходить" відразу за кількома схемами, опиняючись в кожній з них в будь-якої з підгруп.

Основною схемою класифікації всіх криптоалгоритмів є наступна:

- Тайнопис. Відправник і одержувач виробляють над повідомленням перетворення, відомі тільки їм двом. Стороннім особам невідомий сам алгоритм шифрування.
- Криптографія з ключем. Алгоритм впливу на дані, що передаються відомий всім стороннім особам, але він залежить від деякого параметра - "ключа", яким володіють тільки відправник і одержувач.
- Симетричні криптоалгоритми. Для кодування і розшифровки повідомлення використовується один і той же блок інформації (ключ).
- Асиметричні криптоалгоритми. Алгоритм такий, що для шифрування повідомлення використовується один ("відкритий") ключ, відомий всім бажаючим, а для розшифровки - інший ("закритий"), що існує тільки в одержувача.

Надалі докладніше розглянемо криптографії з ключем, так як більшість фахівців саме по відношенню до цих криптоалгоритмів використовують термін криптографія, що цілком виправдано.

Залежно від характеру впливів, вироблених над даними, алгоритми поділяються на:

- перестановки.

Блоки інформації (байти, біти, більші одиниці) не змінюються самі по собі, але змінюється їх порядок проходження, що робить інформацію недоступною сторонньому спостерігачеві.

— Символи.

Самі блоки інформації змінюються за законами криптоалгоритма. Переважна більшість сучасних алгоритмів належить цій групі.

Залежно від розміру блоку інформації криптоалгоритми діляться на:

— Потокові шифри.

Одиницею кодування є один біт. Результат кодування не залежить від минулого раніше вхідного потоку. Схема застосовується в системах передачі потоків інформації, тобто в тих випадках, коли передача інформації починається і закінчується в довільні моменти часу і може випадково перериватися. Найбільш поширеними представників поточних шифрів є скремблери.

— Блокові шифри

Одиницею кодування є блок з декількох байтів (в даний час 4-32). Результат кодування залежить від усіх вихідних байтів цього блоку. Схема застосовується при пакетної передачі інформації і кодування файлів.

2.2 Симетричні криптоалгоритми

2.2.1 Скремблери. Скремблерами називаються програмні або апаратні реалізації алгоритму, що дозволяє шифрувати побитно безперервні потоки інформації. Сам скремблер вдає із себе набір біт, що змінюються на кожному кроці за певним алгоритмом. Після виконання кожного чергового кроку на його виході з'являється шифрує біт - або 0, або 1, який накладається на поточний біт інформаційного потоку операцією XOR.

Останнім часом сфера застосування скремблюється алгоритмів значно скоротилася. Це пояснюється в першу чергу зниженням обсягів побітної послідовної передачі інформації, для захисту якої були розроблені дані алгоритми. Практично повсюдно в сучасних системах застосовуються мережі з комутацією пакетів, для підтримки конфіденційності якої використовуються блокові шифри. А їх криптостійкість перевершує, і часом досить значно, криптостійкість скремблерів.

Суть скремблювання полягає в побітному зміні проходить через систему потоку даних. Практично єдиною операцією, використовуваної в скремблерами є XOR - "побітно виключаюче АБО". Паралельно проходженню інформаційного потоку в скремблер за певним правилом генерується потік біт - кодує потік. Як пряме, так і зворотне шифрування здійснюється накладенням по XOR кодує на вихідну.

Генерація кодує біт проводиться циклічно з невеликого початкового об'єму інформації - ключа за наступним алгоритмом. З поточного набору біт вибираються значення певних розрядів і складаються по XOR між собою. Всі розряди зсуваються на 1 біт, а тільки що отримане значення ("0" або "1") поміщається в звільнився наймолодший розряд. Значення, що знаходилося в самому старшому розряді до зсуву, додається в послідовність, що кодує, стаючи черговим її бітом.

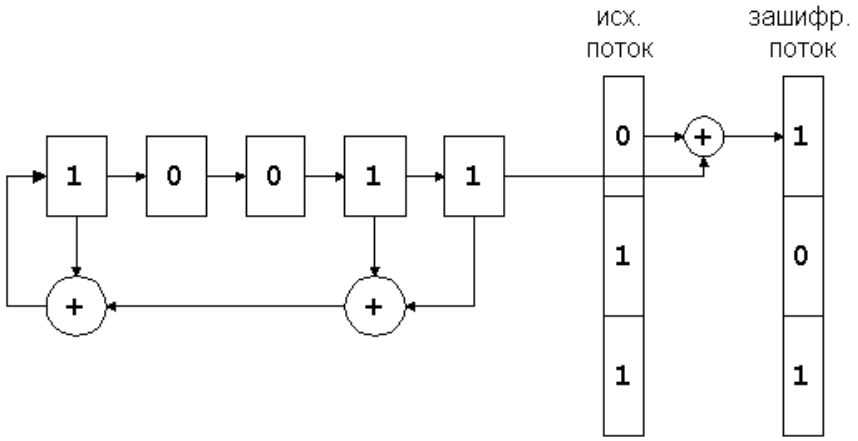


Рисунок 2.1 - Схема скреблювання

З теорії передачі даних криптографія запозичила для запису подібних схем двійкову систему запису. По ній зображений на малюнку скремблер записується комбінацією "100112" - одиниці відповідають розрядам, з яких знімаються біти для формирования зворотного зв'язку.

Розглянемо приклад кодування інформаційної послідовності 010111_2 скремблер 101_2 з початковим ключем 110_2.

скремблер код.біт інф.біт рез-т

1 1 0 _

\ \ \ _

1 + 1 1 _ \ _

\ \ \ _ 0 XOR 0 = 0

0 1 + 1 _ \ _

\ \ \ _ 1 XOR 1 = 0

1 0 1 \ _

\ \ 1 XOR 0 = 1

Пристрій скремблера гранично простий. Його реалізація можлива як на електронній, так і на електричній базі, що й забезпечило його широке застосування в

польових умовах. Більш того, той факт, що кожен біт вихідної послідовності залежить тільки від одного вхідного біта, ще більше зміцнило становище скремблеров в захисті потокової передачі даних. Це пов'язано з неминуче виникають в каналі передачі перешкодами, які можуть спотворити в цьому випадку тільки ті біти, на які вони припадають, а не пов'язану з ними групу байт, як це має місце в блокових шифри [25].

Декодування заскремблірованних послідовностей відбувається за тією ж самою схемою, що і кодування. Саме для цього в алгоритмах застосовується результуюче кодування по "виключаюче АБО" - схема, однозначно відновити події при розкодування без будь-яких додаткових обчислювальних витрат. Зробимо декодування отриманого фрагмента.

Головна проблема шифрів на основі скремблеров - синхронізація передавального (кодує) і приймає (декодуючого) пристроїв. При пропуску або помилковому вставлянні хоча б одного біта вся передана інформація необоротно втрачається. Тому, в системах шифрування на основі скремблеров дуже велика увага приділяється методам синхронізації.

На практиці для цих цілей зазвичай застосовується комбінація двох методів: а) додавання в потік інформації синхронізуючих бітів, заздалегідь відомих приймальній стороні, що дозволяє їй при незнаходження такого біта активно почати пошук синхронізації з відправником, і б) використання високоточних генераторів тимчасових імпульсів, що дозволяє в моменти втрати синхронізації виробляти декодування прийнятих бітів інформації "по пам'яті" без синхронізації.

Число біт, охоплених зворотним зв'язком, тобто розрядність пристрою пам'яті для породжують послідовність, що кодує біт називається розрядністю скремблера.

Можуть бути різні типи графів стану скремблера. На малюнку 2.2. наведені приблизні варіанти для 3-розрядного скремблера. У разі "А" крім завжди присутнього циклу "000" >> "000" ми бачимо ще два цикли - з 3-ма станами і 4-ма. У разі "Б" ми бачимо ланцюжок, яка сходиться до циклу з 3-х станів і вже ніколи звідти не виходить. І нарешті, в разі "В" всі можливі стани крім нульового, об'єднані в один замкнутий цикл. Очевидно, що саме в цьому випадку, коли все 2^N-1 станів системи утворюють цикл, період повторення вихідних комбінацій максимальний, а кореляція між довжиною циклу і початковим станом скремблера (ключем), яка привела б до появи більш слабких ключів, відсутня.

Наприклад, в 8-бітному скремблер, при охопленні 0-го, 1-го, 6-го і 7-го розрядів дійсно за час генерації 255 біт послідовно проходять всі числа від 1 до 255, не повторюючись ні разу.

Схеми з вибраними за цим законом зворотними зв'язками називаються генераторами послідовностей максимальної довжини (ПМД), і саме вони використовуються в скремблюючій апаратурі. З безлічі генераторів ПМД заданої розрядності за часів, коли вони реалізовувалися на електричної або мінімальної електронній базі вибиралися ті, у яких число розрядів, що беруть участь у створенні чергового біта, було мінімальним. Зазвичай генератора ПНД вдавалося досягти за 3 або 4 зв'язку. Сама ж розрядність скремблерів перевищувала 30 біт, що давало можливість передавати до 240 біт = 100 Мбайт інформації без побоювання початку повторення послідовності, що кодує.

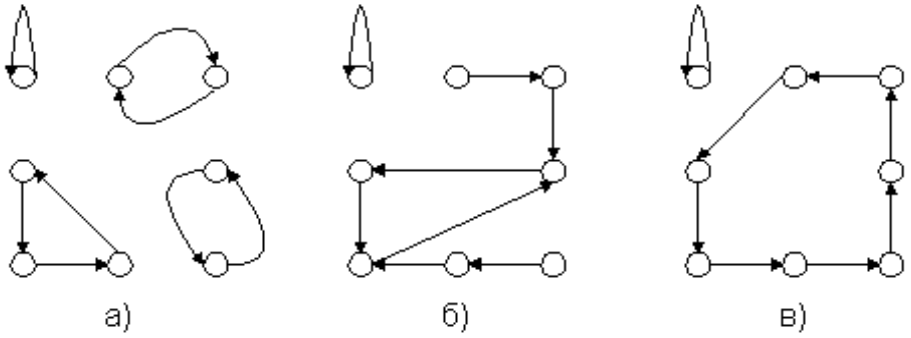


Рисунок 2.2 Графи стану скремблеру 3 розряду

ПНД нерозривно пов'язані з математичною теорією непріводімих полиномів. Виявляється, достатньо щоб поліном ступеня N не був представлений по модулю 2 в вигляді добутку ніяких інших поліномів, для того, щоб скремблер, побудований на його основі, створював ПМД. Наприклад, єдиним неприводимим полиномом ступеня 3 є x3 + x + 1, в двійковому вигляді він записується як 1011_2 (одиниці відповідають присутнім розрядам). Скремблери на основі непріводімих полиномів утворюються відкиданням найстаршого розряду (він завжди присутній, а отже, несе інформацію тільки про ступінь полінома), так на основі зазначеного полінома, ми можемо створити скремблер 0112 з періодом зациклення 7 (= 23-1). Природно, що на практиці застосовуються поліноми значно більш високих порядків. А таблиці неперевіних полиномів будь-яких порядків можна завжди знайти в спеціалізованих математичних довідниках.

Істотним недоліком скремблювальних алгоритмів є їх нестійкість до фальсифікації

2.2.2 Блокові шифри. На сьогоднішній день розроблено досить багато стійких блокових шифрів. Практично всі алгоритми використовують для перетворень певний набір бієктивне (оборотних) математичних перетворень.

Мережею Фейштеля називається метод оборотних перетворень тексту, при якому значення, обчислене від однієї з частин тексту, накладається на інші частини. Часто структура мережі виконується таким чином, що для шифрування і дешифрування використовується один і той же алгоритм - відмінність полягає тільки в порядку використання матеріалу ключа.

Блоковий алгоритм TEA наведено як приклад одного з найпростіших в реалізації стійких криптоалгоритмів.

У 1998 році був оголошений відкритий конкурс на криптостандарта США на кілька перших десятиліть XXI століття. Переможцем конкурсу був визнаний бельгійський блоковий шифр Rijndael. Він претендує на стандарт де-факто блочного шифрування в усьому світі.

Характерною особливістю блокових криптоалгоритмів є той факт, що в ході своєї роботи вони виробляють перетворення блоку вхідної інформації фіксованої довжини і отримують результуючий блок того ж обсягу, але недоступний для прочитання стороннім особам, які не володіють ключем. Таким чином, схему роботи блочного шифру можна описати функціями Z = EnCrypt (X, Key) і X = DeCrypt (Z, Key)

Ключ Key є параметром блокового криптоалгоритму і являє собою деякий блок двійковій інформації фіксованого розміру. Вихідний (X) і зашифрований (Z) блоки даних також мають фіксовану розрядність, рівну між собою, але необов'язково рівну довжині ключа.

Блокові шифри є основою, на якій реалізовані практично всі криптосистеми. Методика створення ланцюжків із зашифрованих блоковими алгоритмами байт дозволяє шифрувати ними пакети інформації необмеженої довжини. Така властивість блокових шифрів, як швидкість роботи, використовується асиметричними криптоалгоритмами, повільними за своєю природою. Відсутність статистичної кореляції між бітами вихідного потоку блочного шифру використовується для обчислення контрольних сум пакетів даних і в хешуванні паролів.

Наступні розробки всесвітньо визнані стійкими алгоритмами і публікацій про універсальні методи їх злому в засобах масової інформації на момент створення матеріалу не зустрічалося.

Криптоалгоритм іменується ідеально стійким, якщо прочитати зашифрований блок даних можна тільки перебравши всі можливі ключі, до тих пір, поки повідомлення не опиниться осмисленим. Так як по теорії ймовірності шуканий ключ буде знайдений з вірогідністю 1/2 після перебору половини всіх ключів, то на злом ідеально стійкого криптоалгоритму з ключем довжини N потрібно в середньому 2^N-1 перевірок. Таким чином, в загальному випадку стійкість блокового шифру залежить тільки від довжини ключа і зростає експоненціально з її ростом. Навіть припустивши, що перебір ключів проводиться на спеціально створеній багатопроцесорної системі, в якій завдяки діагональному паралелізму на перевірку 1 ключа йде тільки 1 такт, то на злом 128 бітного ключа сучасній техніці буде потрібно не менше тисячі двадцять один років. Природно, все сказане стосується лише ідеально стійким шифрів, якими, наприклад, з великою часткою впевненості є наведені в таблиці вище алгоритми.

Крім цього умови до ідеально стійких криптоалгоритмів застосовується ще одна дуже важлива вимога, якій вони повинні обов'язково відповідати. При відомих вихідному і зашифрованому значеннях блоку ключ, яким зроблено це перетворення, можна дізнатися також тільки повним перебором. Ситуації, в яких сторонньому спостерігачеві відома частина вихідного тексту зустрічаються повсюдно. Це можуть бути стандартні написи в електронних бланках, фіксовані заголовки форматів файлів, досить часто зустрічаються в тексті довгі слова або послідовності байт. У світлі цієї проблеми описане вище вимога не є нічим надмірним і також строго виконується стійкими криптоалгоритмами, як і перше.

Таким чином, на функцію стійкого блокового шифру Z = EnCrypt (X, Key) накладаються наступні умови:

— Функція EnCrypt повинна бути оборотною.

— Не повинно існувати інших методів прочитання повідомлення X за відомим блоку Z, крім як повним перебором ключів Key.

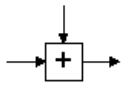
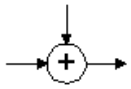
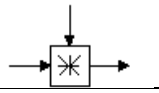
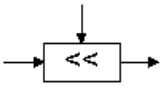
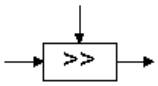
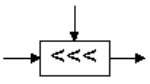
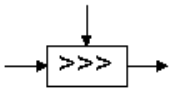
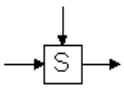
— Не повинно існувати інших методів визначення яким ключем Key було вироблено перетворення відомого повідомлення X в повідомлення Z, крім як повним перебором ключів.

Всі дії, вироблені над даними блоковим криптоалгоритмом, засновані на тому факті, що перетворений блок може бути представлений у вигляді цілого невід'ємного

числа з діапазону, відповідного його розрядності. Так, наприклад, 32-бітний блок даних можна інтерпретувати як число з діапазону 0..4'294'967'295. Крім того, блок, розрядність якого зазвичай є "ступенем двійки", можна трактувати як кілька незалежних невід'ємних чисел з меншого діапазону (розглянутий вище 32-бітний блок можна також представити у вигляді 2 незалежних чисел з діапазону 0..65535 або у вигляді 4 незалежних чисел з діапазону 0..255).

Над цими числами блоковим криптоалгоритмом і виробляються за певною схеме наступні дії (див. таблицю 2.1. зліва дані умовні позначення цих операцій на графічних схемах алгоритмів):

Таблиця 2.1 Схема блочного криптоалгоритму

Бієктивні математичні функції		
	Додавання	X'=X+V
	Виключаюче АБО	X'=X XOR V
	Множення по модулю 2^N+1	X'=(X*V) mod (2^N+1)
	Множення по модулю 2^N	X'=(X*V) mod (2^N)
Бітові зрушення		
	Арифметичний зсув вліво	X'=X SHL V
	Арифметичний зсув вправо	X'=X SHR V
	Циклічний зсув вліво	X'=X ROL V
	Циклічний зсув вправо	X'=X ROR V
Табличні підстановки		
	S-box (англ. substitute)	X'=Table[X,V]

Як параметр V для будь-якого з цих перетворень може використовуватися:

— фіксоване число (наприклад, X '= X + 125);

— число, що отримується з ключа (наприклад, X '= X + F (Key));

— число, що отримується з незалежної частини блоку (наприклад, X2 '= X2 + F (X1)).

Останній варіант використовується в схемі, названої на честь її творця мережею Фейштеля (нім. Feistel).

Послідовність виконуваних над блоком операцій, комбінації перерахованих вище варіантів V і самі функції F і складають "ноу-хау" кожного конкретного блокового криптоалгоритму.

Характерною ознакою блокових алгоритмів є багаторазове і непряме використання матеріалу ключа. Це диктується в першу чергу, вимогою неможливості зворотного декодування щодо ключа при відомих вихідному і зашифрованому текстах. Для вирішення цього завдання в наведених вище перетвореннях найчастіше використовується не саме значення ключа або його частини, а деяка, іноді необоротна (небіективна) функція від матеріалу ключа. Більш того, в подібних перетвореннях один і той же блок або елемент ключа використовується багаторазово. Це дозволяє при виконанні умови оборотності функції щодо величини X зробити функцію незворотною щодо ключа Key.

Оскільки операція зашифровуваної або розшифровки окремого блоку в процесі кодування пакета інформації виконується багаторазово (іноді до сотень тисяч разів), а значення ключа і, отже, функцій Vi (Key) залишається незмінним, то іноді стає доцільно заздалегідь однократно обчислити дані значення і зберігати їх в оперативної пам'яті спільно з ключем. Оскільки ці значення залежать тільки від ключа, то вони в криптографії називаються матеріалом ключа. Необхідно відзначити, що дана операція не змінює ні довжину ключа, ні криптостійкості алгоритму в цілому. Тут відбувається лише оптимізація швидкості обчислень шляхом кешування (англ. Caching) проміжних результатів. Описані дії зустрічаються практично у багатьох блокових криптоалгоритмах і носять назву розширення ключа (англ. Key scheduling).

2.2.2Мережа Фейштеля. Мережа Фейштеля є подальшою модифікацією описаного вище методу змішування поточної частини шифруемого блоку з результатом деякої функції, обчисленої від іншої незалежної частини того ж блоку. Ця методика отримала широке поширення, оскільки забезпечує виконання вимоги про багаторазовому використанні ключа і матеріалу вихідного блоку інформації.

Незалежні потоки інформації, породжені з вихідного блоку, називаються гілками мережі. У класичній схемі їх дві. Величини V_i іменуються параметрами мережі, зазвичай це функції від матеріалу ключа. Функція F називається утворює. Дія, що складається з одноразового обчислення утворює функції і подальшого накладення її результату на іншу гілку з обміном їх місцями, називається циклом або раундом (англ. Round) мережі Фейштеля. Оптимальне число раундів K - від 8 до 32. Важливим є те, що збільшення кількості раундів значно збільшує кріптоскойстость будь-якого

блочного шифру до криптоанализу. Можливо, ця особливість і вплинула на настільки активне поширення мережі Фейштеля - адже при виявленні, скажімо, будь-якого слабкого місця в алгоритмі, майже завжди досить збільшити кількість раундів на 4-8, що не переписуючи сам алгоритм. Часто кількість раундів не фіксується розробниками алгоритму, а лише вказуються розумні межі (обов'язково нижній, і не завжди - верхній) цього параметра.

Мережа Фейштеля володіє такою властивістю, що навіть якщо в якості утворення функції F буде використано необоротне перетворення, то і в цьому випадку весь ланцюжок буде відновлений. Це відбувається внаслідок того, що для зворотного перетворення мережі Фейштеля не потрібно обчислювати функцію F-1.

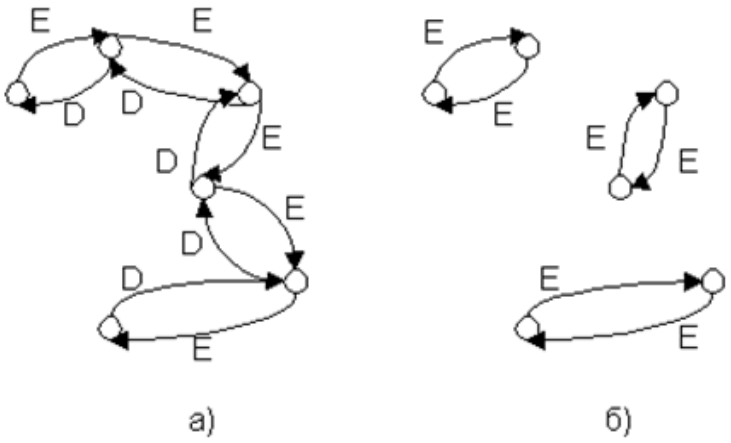


Рисунок 2.3 Мережа Фейштеля

Більш того, як неважко помітити, мережа Фейштеля симетрична. Використання операції XOR, оборотної своїм же повтором, і інверсія останнього обміну гілок роблять можливим розкодування блоку тієї ж мережею Фейштеля, але з інверсним порядком параметрів V_i. Зауважимо, що для оборотності мережі Фейштеля не має значення чи є число раундів парних або непарних числом. У більшості реалізацій схеми, в яких обидва вищеперелічені умови (операція XOR і знищення останнього обміну) збережені, пряме і зворотне перетворення виробляються однією і тією ж процедурою, якою як параметр передається вектор величин V_i або в вихідному, або в інверсному порядку.

З незначними доробками мережу Фейштеля можна зробити і абсолютно симетричною, тобто виконує функції шифрування і дешифрування одним і тим же

набором операцій. Математичною мовою це записується як "Функція EnCrypt тотожно дорівнює функції DeCrypt". Якщо ми розглянемо граф станів криптоалгоритма, на якому точками відзначені блоки вхідний і вихідний інформації, то при якомусь фіксованому ключі для класичної мережі Фейштеля ми матимемо картину, зображену на рисунку (2.3), а в другому випадку ка ждая пара точок отримає унікальну зв'язок, як зображено на рисунку (2.3б). Модифікація мережі Фейштеля, що володіє подібними властивостями наведена на рисунку (2.4).

Як бачимо, основна її хитрість в повторному використанні даних ключа в зворотному порядку в другій половині циклу. Необхідно зауважити, однак, що саме через цю недостатньо дослідженою специфіки такої схеми (тобто потенційної можливості ослаблення зашифрованого тексту зворотними перетвореннями) її використовують в криптоалгоритмах з великою обережністю.

Модифікацію мережі Фейштеля для більшого числа гілок застосовують набагато частіше. Це в першу чергу пов'язано з тим, що при великих розмірах кодованих блоків (128 і більше біт) стає незручно працювати з математичними функціями по модулю 64 і вище. Як відомо, основні одиниці інформації оброблювані процесорами на сьогоднішній день - це байт і подвійне машинне слово 32 біта. Тому все частіше і частіше в блокових криптоалгоритмах зустрічається мережу Фейштеля з 4-ма гілками. Найпростіший принцип її модифікації зображений на рисунку 2.4(а). Для більш швидкого перемішування інформації між гілками (а це основна проблема мережі Фейштеля з великою кількістю гілок) застосовуються дві модифіковані схеми, звані "type-2" і "type-3". Вони зображені на рисунку 2.4(б,в).

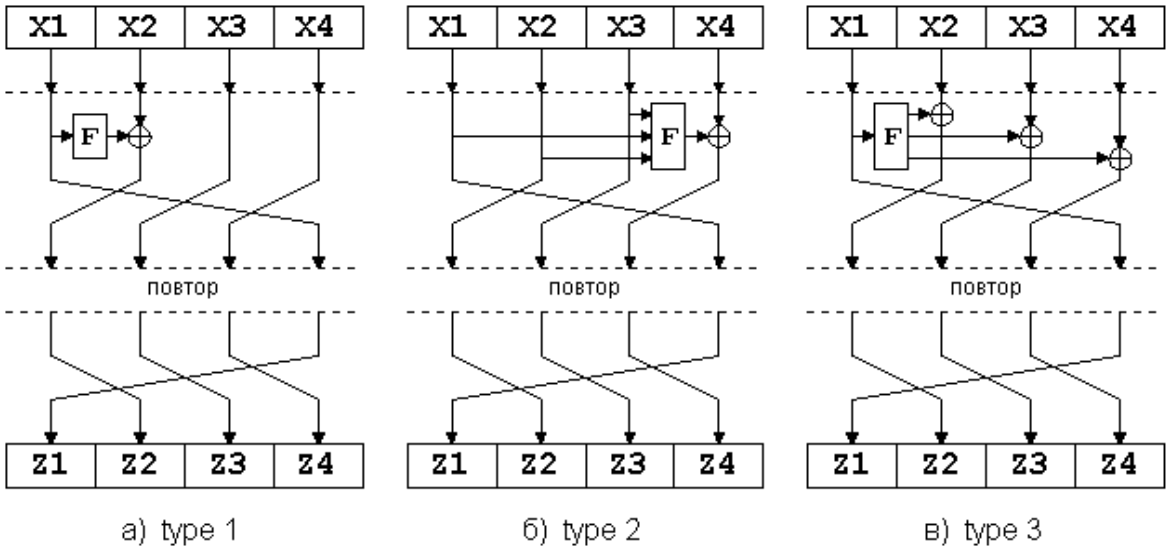


Рисунок 2.4 Модифіковані схеми, так звані "type-2" і "type-3

Мережа Фейштеля надійно зарекомендувала себе як крипостійкістна схема добутку перетворень, і її можна знайти практично в будь-якому сучасному блоковому шифрі. Незначні модифікації стосуються зазвичай додаткових початкових та кінцевих перетворень (англомовний термін - whitening) над зашифрованих блоком. Подібні перетворення, що виконуються зазвичай також або "виключаюче АБО", або складанням мають на меті підвищити початкову рандомізацію вхідного тексту. Таким чином, криптостійкість блочного шифру, що використовує мережу Фейштеля, визначається на 95% функцією F і правилом обчислення Vi з ключа. Ці функції і є об'єктом все нових і нових досліджень фахівців в області криптографії.

2.2.3 Блоковий шифр TEA. Розглянемо один з найпростіших в реалізації, але визнано стійких криптоалгоритмів - TEA (Tiny Encryption Algorithm).

Параметри алгоритму:

Розмір блока - 64 біта.

Довжина ключа - 128 біт.

В алгоритмі використана мережу Фейштеля з двома гілками в 32 біта кожна. Утворює функція F оборотна.

Мережа Фейштеля несиметрична через використання в якості операції накладення що не виключає "АБО", а арифметичного додавання.

Відмінною рисою криптоалгоритма TEA є його розмір. Простота операцій, відсутність табличних підстановок і оптимізація під 32-розрядну архітектуру процесорів дозволяє реалізувати його на мові ASSEMBLER в гранично малому обсязі коду. Недоліком алгоритму є деяка повільність, викликана необхідністю повторювати цикл Фейштеля 32 рази (це необхідно для ретельного "перемішування даних" через відсутність табличних підстановок).

2.2.4 AES -RC6: стандарт блочних шифрів США с 2000 року. Алгоритм є продовженням криптоалгоритма RC5, розробленого Рональдом Ривестом (англ. Ron Rivest) - дуже відомою особою в світі криптографії. RC5 був незначно змінений для того, щоб відповідати вимогам AES по довжині ключа і розміром блоку. При цьому алгоритм став ще швидше, а його ядро, успадковане від RC5.

Перетворення T (x) дуже просто: $T(X) = (X * (X + 1)) \bmod 2^N$. Воно використовується в якості нелінійного перетворення з хорошими показниками перемішування бітового значення вхідної величини.

2.2.5 Шифр Rijndael. Даний алгоритм розроблений двома фахівцями з криптографії з Бельгії. Він є нетрадиційним блоковим шифром, оскільки не використовує мережу Фейштеля для криптоперетворень. Алгоритм представляє кожен блок кодованих даних у вигляді двовимірного масиву байт розміром 4x4, 4x6 або 4x8 в залежності від встановленої довжини блоку. Далі на відповідних етапах перетворення виробляються або над незалежними стовпцями, або над незалежними рядками, або взагалі над окремими байтами в таблиці.

Всі перетворення в шифрі мають суворе математичне обґрунтування. Сама структура і послідовність операцій дозволяють виконувати даний алгоритм ефективно як на 8-бітних так і на 32-бітних процесорах. У структурі алгоритму закладена можливість паралельного виконання деяких операцій, що на багатопроцесорних робочих станціях може ще підняти швидкість шифрування в 4 рази.

Алгоритм складається з певної кількості раундів (від 10 до 14 - це залежить від розміру блоку і довжини ключа), в яких послідовно виконуються наступні операції: ByteSub - таблична підстановка 8x8 біт представлена на рисунку (2.5).

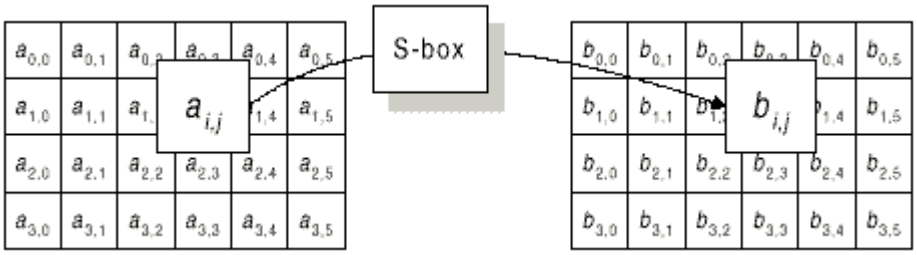


Рисунок 2.5 ByteSub

ShiftRow - зсув рядків в двовимірному масиві на різні зміщення Рисунок (2.6),

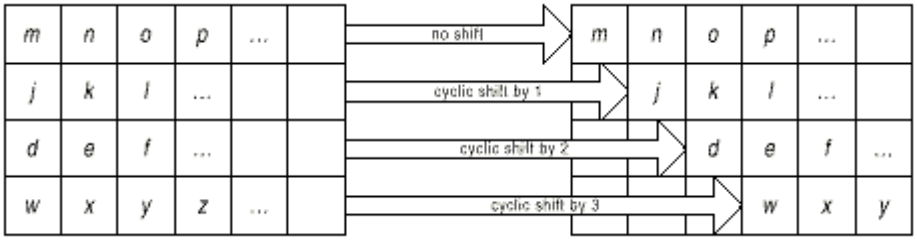


Рисунок 2.6 ShiftRow

MixColumn - математичне перетворення, переміщуючий дані всередині стовпчика рисунок (2.7),

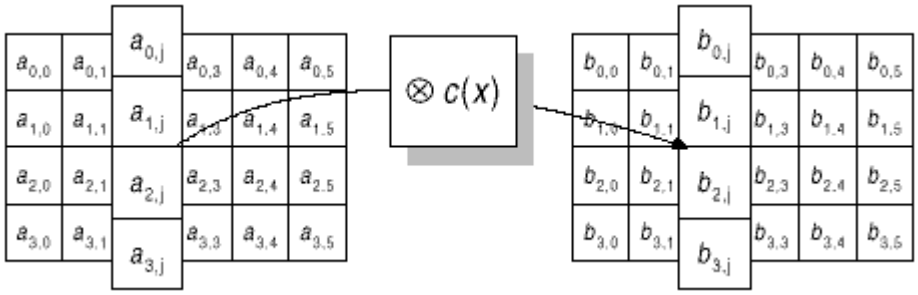


Рисунок 2.7 MixColumn

AddRoundKey - додавання матеріалу ключа операцією XOR ().

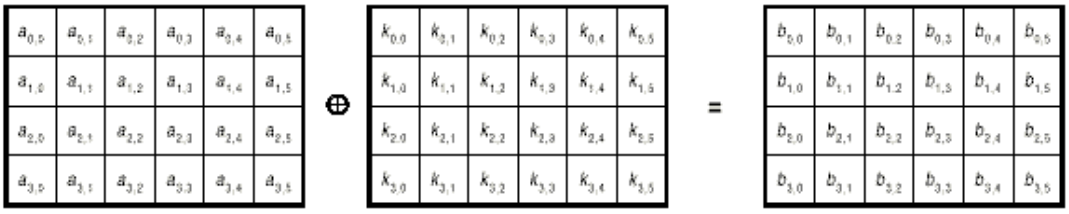


Рисунок 2.8 AddRoundKey

В після вдень раунді операція перемішування стовпців відсутній, що робить всю послідовність операцій симетричною.

2 РОЗРОБКА СХЕМИ АЛГОРИТМУ ДЛЯ МЕТОДУ ЕЛЬ-ГАМАЛЯ

3.1 Алгоритм Ель-Гамаля для шифрування

Схема була запропонована Тахера Ель-Гамаль в 1984 році. Ель-Гамаль розробив один з варіантів алгоритму Діффі-Хеллмана. Він удосконалив систему Діффі-Хеллмана і отримав два алгоритма, які використовувалися для шифрування і для забезпечення аутентифікації. На відміну від RSA алгоритм Ель-Гамаля ні запатентований і, тому, став більш дешевою альтернативою, тому що не була потрібна оплата внесків за ліцензію. Вважається, що алгоритм потрапляє під дію патенту Діффі-Хеллмана.

3.2 Алгоритми генерації ключів, шифрування та дешифрування

Алгоритм Ель-Гамаля - двоключового алгоритма, призначений як для шифрування/дешифрування повідомлень, так і для генерації електронного підпису. В основі секретності алгоритму лежить висока складність операцій обчислення цілочисельних логарифмів в порівнянні з операцією піднесення до ступеня в кінцевих полях.

При використанні алгоритму Ель-Гамаля для шифрування інформації зашифроване повідомлення матиме вдвічі більший розмір у порівнянні з вхідним.

3.2.1 Генерація ключів.

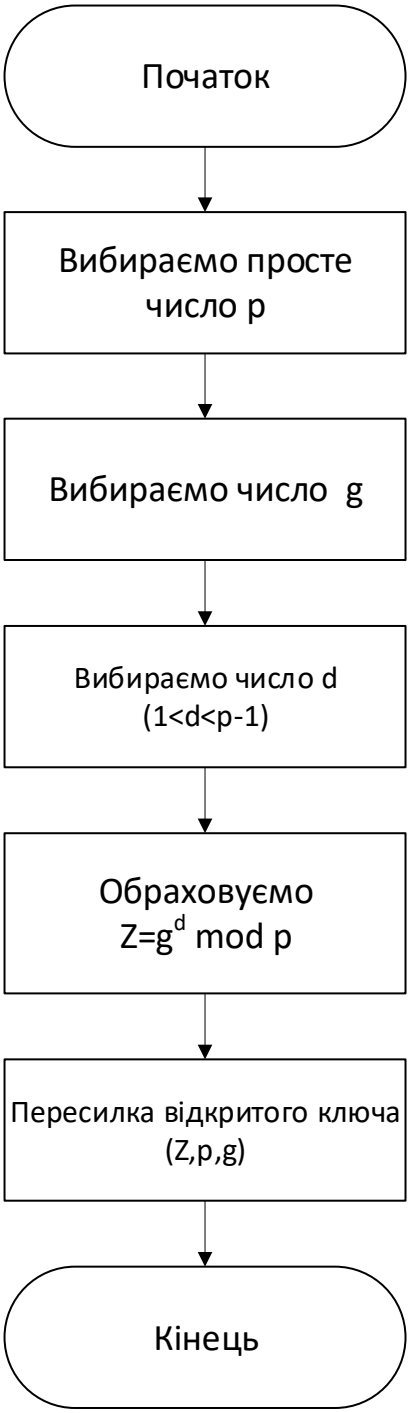


Рисунок 3.1 Алгоритм генерації відкритого ключа

Для генерації пари ключів, - відкритого і індивідуального, - необхідно вибрати велике просте число p і два довільних числа g і d, але число d повинно виконувати умову:

$1 < d < p-1$

Далі належить обчислити:

$$Z = g^d \ mod \ p$$

Відкритий ключ утворюють числа Z, g, і p, причому g і p можуть бути загальними для групи користувачів.

Індивідуальним (закритим) ключем є число d.

3.2.2 Шифрування. Тепер розглянемо шифрування, як таке, яке буде застосовуватися у нашому пристрої для шифрування даних. Перелік буде йти покроково.

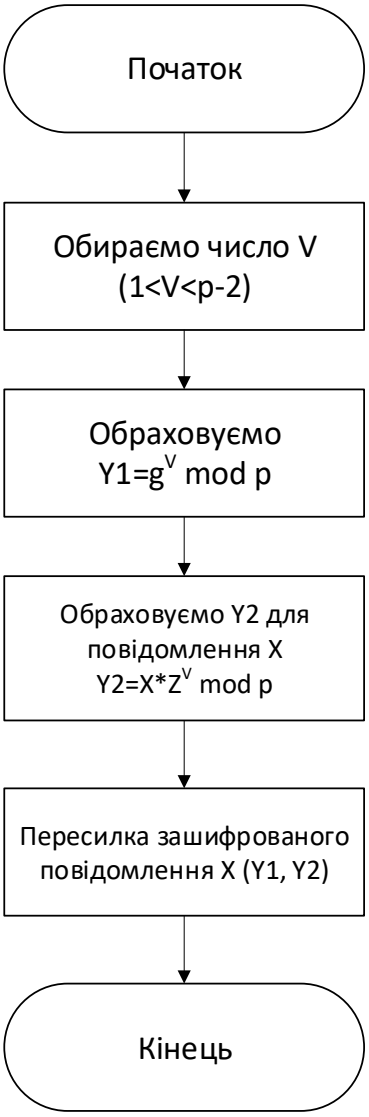


Рисунок 3.2 Алгоритм шифрування повідомлення

Для шифрування повідомлення, фрази, абощо, позначимо його X вибираємо будь-яке число V, взаємнопросте с (p-1) і обраховуємо:

$$Y1 = g^V \ mod \ p$$

$$Y2 = (Z^V \cdot X) \ mod \ p$$

Y1 та Y2 утворюють зашифроване повідомлення, його розмір удвічі перевищує розмір вихідного незашифрованого повідомлення X.

3.2.3Дешифрування.

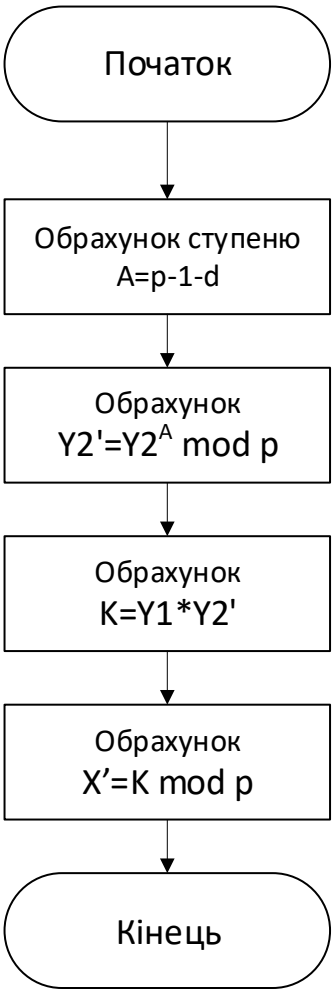


Рисунок 3.3 Алгоритм дешифрування повідомлення

Дешифрування повідомлення відбувається у відповідності з виразом:

$$X' = Y1 * Y2^{d-1-p} mod \ p$$

Доказ вірності:

$$X' = (Y2/(Y1^d))\ mod\ p$$

$$Y2 = (Z^V \cdot X)\ mod\ p = (g^{V \cdot d} \cdot X)\ mod\ p$$

$$Y1^d = g^{V \cdot d}\ mod\ p$$

$$X' = [(g^{V \cdot d} \cdot X)/(g^{V \cdot d})]\ mod\ p = X$$

4 ПРОБЛЕМИ РЕАЛІЗАЦІЇ ШИФРУВАННЯ З ВІДКРИТИМ КЛЮЧЕМ НА АПАРАТНОМУ РІВНІ

В основі систем шифрування з відкритим ключем є використання односторонніх функцій. Одностороння функція – така математична функція функція яка легко обчислюється для будь якого вхідного значення, але зворотню дію виконати за невеликий час слкадно.

Одними із основних недоліків алгоритмів несиметричного шифрування в порівнянні з симетричним є:

- в алгоритм складніше внести зміни;
- довші ключі - нижче наведена таблиця, що зіставляє довжину ключа симетричного алгоритму з довжиною ключа RSA з аналогічною криптостійкістю;
- шифрування-розшифрування з використанням пари ключів проходить на два-три порядки повільніше, ніж шифрування-розшифрування того ж тексту симетричним алгоритмом;
- потрібні істотно більші обчислювальні ресурси, тому на практиці асиметричні криптосистеми використовуються в поєднанні з іншими алгоритмами:
  - для ЕЦП повідомлення попередньо піддається хешування, а за допомогою асиметричного ключа підписується лише відносно невеликий результат хеш-функції;
  - для шифрування вони використовуються в формі гібридних криптосистем, де великі обсяги даних шифруються симетричним шифром на сеансовому ключі, а за допомогою асиметричного шифру передається тільки сам сеансовий ключ.

Таблиця 4.1 Довжина ключів шифрування

Довжина симметричного ключа, біт	Довжина ключа RSA, біт
56	384
64	512
80	768
112	1792
128	2304

В основі надійності методу Ель Гамаля полягає функція дискретного логарифмування - $g^x=a$ (знаходження ступеня x при відомих g і результату оперції mod (залишку від ділення) a. Тому для надійності треба використовувати великі числа для того щоб знаходження результату дискретного логарифмування було досить складним.

Для шифрування потрібний відкритий ключ для генерації якого потрібно обрати число p, яке буде модулем в усі подальших операціях. Воно повинно відповідати деяким вимогам – воно повинно бути простим і найбільшим з усіх випадкових чисел котрі вибираються при генерації ключа та шифруванні повідомлення, все це ставить під питання доцільність використання цього методу для реалізації на програмному рівні.

Рекомендована довжина ключа для надійного шифрування методом Ель Гамаля –1024біт
(179769313486231590772930519078902473361797697894230657273430081157732675805500963132708477322407536021120113879871393357658789768814416622492847430639474124377767893424865485276302219601246094119453082952085005768838150682342462881473913110540827237163350510684586298239947245938479716304835356329624224137216 – число довжиною 1024 біти), що є досить великим значенням для апаратної реалізації.

Іншою проблемою апаратної реалізації являється використання методом операцій модульної експоненціації

$$A^x \pmod{M} = y,$$

та модульного множення

$$M = A \times B \bmod n.$$

Апаратна реалізація данних операцій над великими числами є дуже складною.

Апаратне забезпечення для прискорення точної експоненції (крім прискорення множення) зустрічається рідко, оскільки у продуктивності з таким обладнанням або без нього буде домінувати час, необхідний для здійснення складових множень.

5 РОЗРОБКА ФУНКЦІОНАЛЬНОЇ СХЕМИ ПРИСТРОШУ ЗАХИСТУ ІНФОРМАЦІЇ МЕТОДОМ ЕЛЬ-ГАМАЛЯ.

Входи та виходи блоку верхнього рівня пристрою шифрування є показано в таблиці 5.1, і функціональна схема на рисунку 5.1.

Пристрій керування (CU – control unit) підключений до входів та виходів всіх інших вузлів для керування їх виконанням згідно з алгоритмом виконання шифрування методом Ель-Гамаля.

Пристрій шифрування виконає два процеси експоненції і одне множення. Для оптимального дизайну ми будемо використовувати один пристрій модульної експоненціації (EXP), а не два, щоб зменшити розмір пристрою, а отже, потрібно використовувати пристрій контролю (CU).

Вихід мультиплексора (MUX) підключений до вхідного порту (b) в пристрій модульної експоненціації(EXP) і два його входи підключені до двох очікуваних основ g і Z. Аналогічним чином вхід демультиплексора (DEMUX) підключений до виходу пристрою модульної експоненціації і два його виходи підключені до відповідних входів які потребують резульату експоненціації. Пристрій контролю (CU) підключений до всіх інших компонентів у входи/виходи в які подаються сигнали старту та закінчення.

Пристрій починає шифрування коли отримує на вхід сигналу початку шифрування (start). Пристрій контролю (CU) встановлює значення «нуль» свого порту (S1), який підключено до лінії вибору мультиплексора (S). У результаті чого, вихід мультиплексор буде значенням на вхідному порту (I0) що є g (основа експоненціації). Пристрій управління (CU) посилає значення логічної одиниці на порту 4 пристрою модульної експоненціації для початку першої експоненції з основою g, модулем p і експонентою (V). Після закінчення пристрій модульної експоненціації(EXP) відправляє на пристрій контролю (CU) сигнал finish exp зі значенням 1.

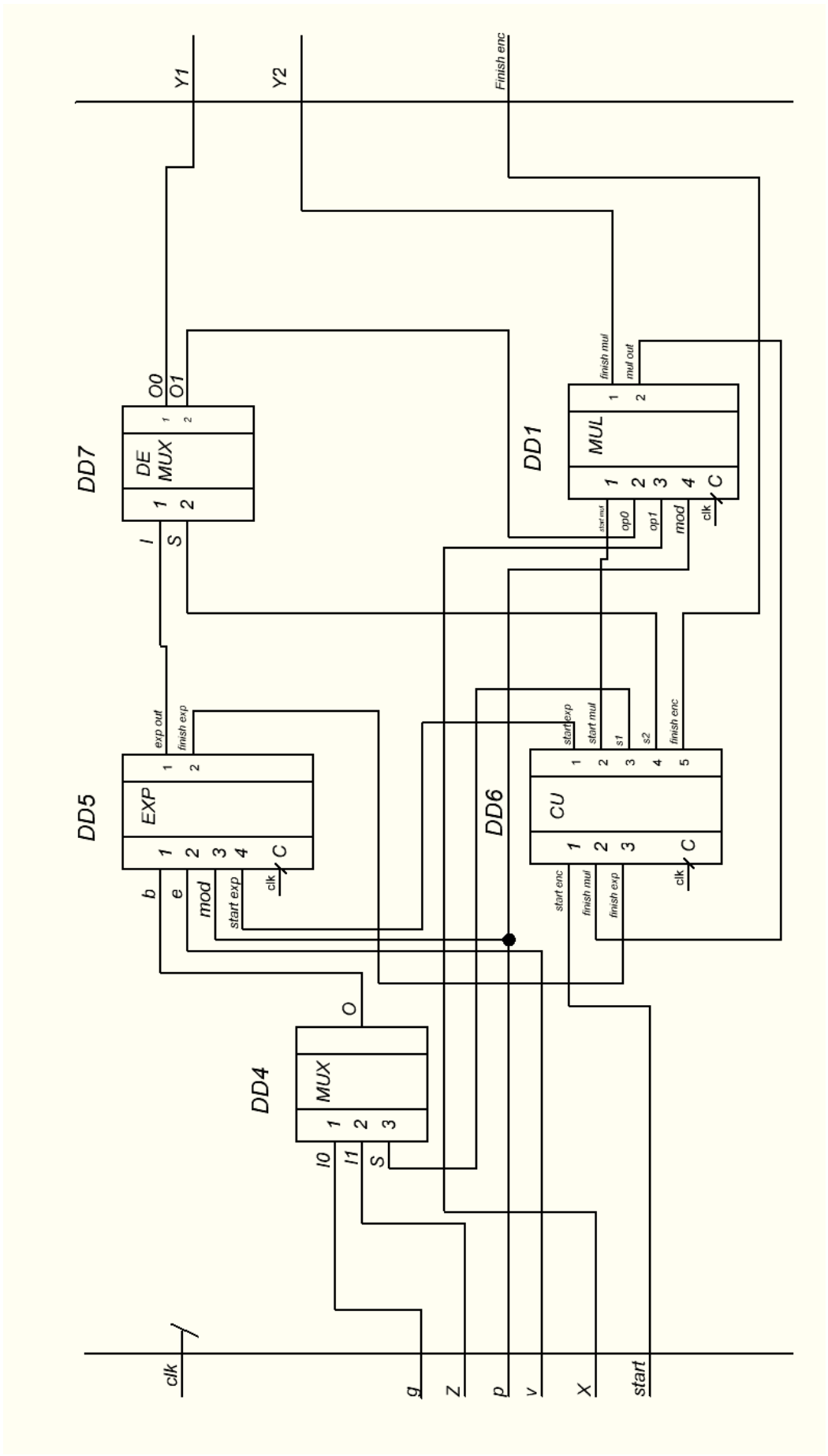


Рисунок 5.1 Функціональна схема пристрою шифрування

Пристрій контролю (CU) встановлює startexp в логічний нуль і разом з ним посилає логічний нуль на вихід (S2), для того щоб результат пристрою модульної експоненціації (EXP) вийшов з вихідного порту (O0) демультиплексора (DEMUX), який підключений до виводу порт (Y1). Перше вихідне значення ($Y1=g^V \bmod p$) готове.

Другу експоненціацію починається коли пристрій контролю (CU) змінює значення (S1) на логічну одиницю, як результат на вхід пристрою модульної екпоненціації (EXP) поступає значення з поступає значення з іншого вхідного порту мультиплексора (MUX) з порту I1 (Z) - нова основа експоненціації для Exp. Знову використовується вхідне довільне значення V ($1<V<p-1$) як показник ступеню, як у перший раз. Пристрій контролю (CU) встановлює значення startexp в «один» і, отже, пристрою модульної екпоненціації (EXP) починає працювати. Закінчивши, пристрій модульної екпоненціації (EXP) надсилає сигнал finish exp до пристрій контролю (CU) який посилає сигнал «один» з виходу (S2), тому демультиплексор (DEMUX) бере результат модульної експоненціації і відправляє його через вихідний (O1), який підключений на вхідний порт op0 пристрою модульного множення (MUL). Інший вхідний порт пристрою модульного множення (MUL) - (op1) підключений до вхідного порту верхнього рівня (повідомлення X яке потрібно зашифрувати). Пристрій контролю (CU) починає роботу пристрою модульного множення (MUL) за допомогою стартового сигналу (start mul) встановлюючи його в «один». Вихідний результат модульного множення підключений до порту верхнього рівня Y2. Друге значення ($Y2=X*Z^V \bmod p$), яке містить оригінальне повідомлення доступне. Пристрій модульного множення (MUL) посилає сигнал закінчення модульного множення до пристрою контролю (CU) , який, у свою чергу, встановлює значення вихідного порту верхнього рівня finish enc в «один», що вказує що процес шифрування закінчений. І два необхідні зашифровані значення (Y1 і Y2) готові для відправки одержувачу.

Таблиця 5.1 Входи і виходи верхнього рівня пристрою ширфування

Назва виходу	Опис
Clk	Вхід тактового імульсу
g	Перша частина відкритого ключа
Z	Друга частина відкритого ключа
p	Модуль
X	Вхідне повідомлення
V	Секретне довільне чилсо для шифрування
Start	Вхід сигналу для початку шифрування
Y1	Перша частина шифротексту
Y2	Друга частина шифротексту
Finish enc	Сигнал індикації закінчення шифрування

ВИСНОВОК

В першому розділі було розглянуто основи інформаційної безпеки її категорії та моделі захисту інформації.

У другому розділі було детально оглянуто класифікацію криптоалгоритмів та різновид симетричних криптоалгоритмів.

У третьому розділі було побудовано алгоритми для генерації відкритих ключів, шифрування повідомлення та дешифрування повідомлення.

У четвертому розділі було оглянуто складність апаратної реалізації методів захисту інформації з відкритим ключем.

У п’ятому розділі було побудовано функціональну схему пристрою шифрування на основі методу Ель Гамаля та детально описано її принцип функціонування.

В результаті аналізу проведеного під час роботи можливо зробити висновок що, актуальність апаратної реалізації даного пристрою досить сумнівна по декільком причинам. Перш за все треба дати відповідь на питання де було б доцільно використовувати даний пристрій ? Я не можу відповісти на це запитання.

Із за складності арифметичних операцій які виконуються при генерації ключів, шифруванні та дешифруванні доцільно використовувати мікроконтролери на базі архітектури RISC. Але якщо використовувати невеликі та дешеві мікроконтролери на базі AVR архітектури тоді генерація ключів, шифрування і дешифрування займає досить тривалий час до 20 секунд. Якщо зменшити довжину ключа і повідомлення тоді значно зменшується криптостійкість алгоритму і доцільність практичної реалізації такого пристрою зникає.

Для практичної апаратної реалізації більш підходять симетричні системи шифрування наприклад AES.

ЛІТЕРАТУРА

1.https://www.intuit.ru/studies/professional_retraining/966/courses/102/lecture/2993?page=5.

2. https://math.wikia.org/ru/wiki/ Дискретное_логарифмирование

3. Е. Баранова, А. Бабаш "Информационная безопасность и защита информации" 3-е изд. (2016) - 30 с.

4.https://www.researchgate.net/publication/233374408_Hardware_Design_and_Implementation_of_ElGamal_Public-Key_Cryptography_Algorithm.

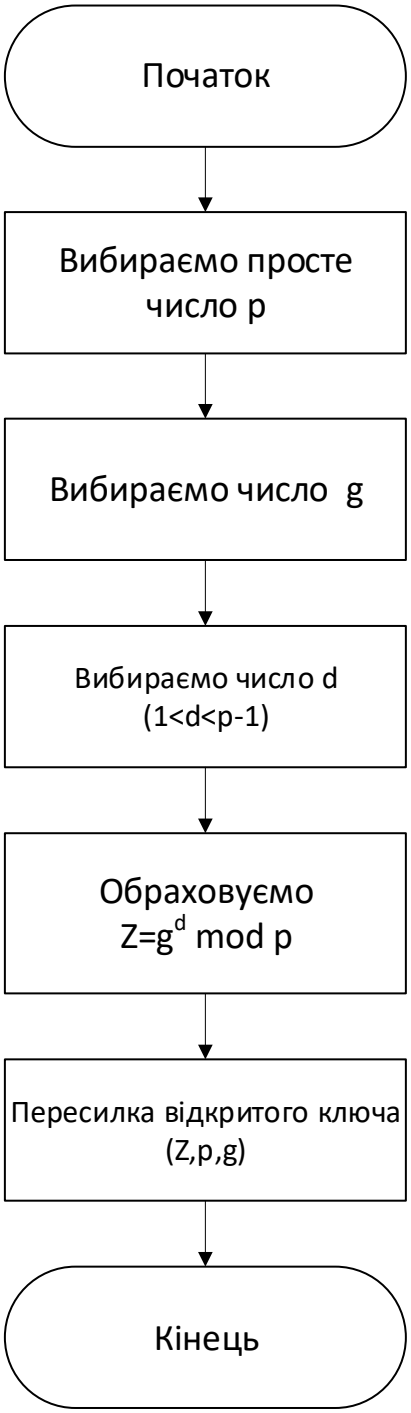
5. ElGamal, T. (1998, July). A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Trans.- Information Theory, IT-31(4), 469–472..

6. https://it.rfei.ru/course/~k017/~V8u3Fj4l/~hIGNMjZS.

7. https://studme.org/239583/informatika/shifr_gamalya.

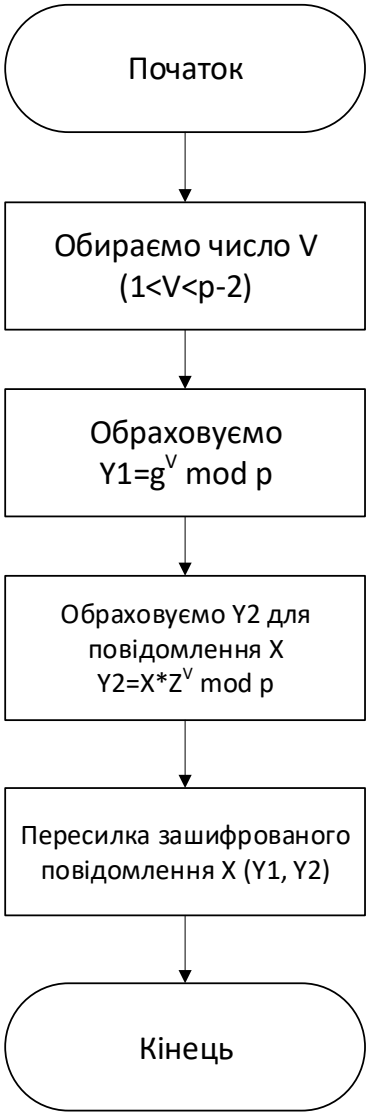
8. https://ru.wikipedia.org/wiki/Криптосистема_с_открытым ключом

Додаток А



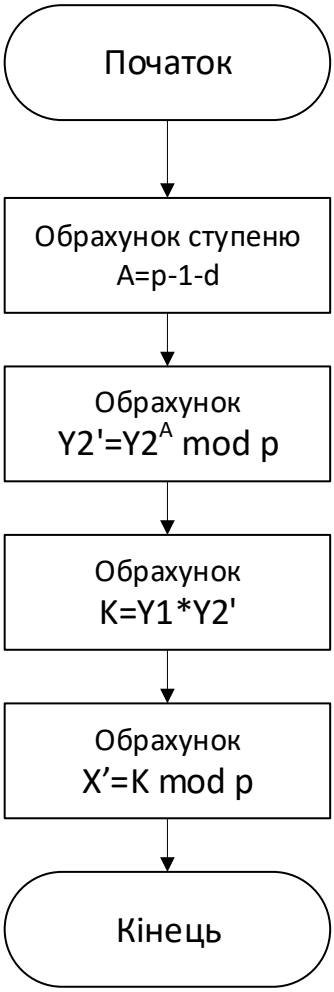
					ЕлІТ 6.172.00.10.533	
Змн.	*Лист*	*№ документа*	*Підпис*	*Дата*		
Розроб.		*Клещев М.А.*			Пристрій захисту інформації методом Ель-Гамаля. Алгоритм генерації ключів	*Літ.* / *Маса* / *Маштаб*
Пеоевіо		*Протасова Т.О.*				
Т. Контр.						*Аркуш. 44* / *Аркушів. 48*
		.				
Н. Контр.		*Протасова Т.О.*				*СумДУ, ТК-61*
Утверд.		*Опанасюк А.С.*				

Додаток Б



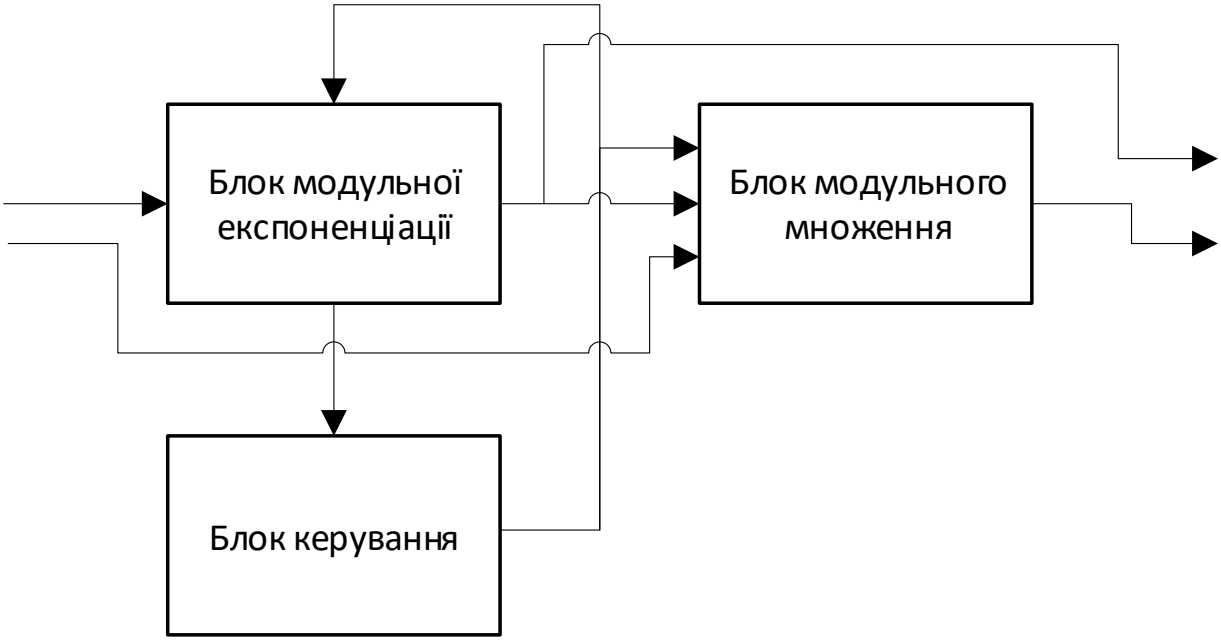
					ЕлІТ 6.172.00.10.533		
					Пристрій захисту інформації методом Ель-Гамаля. Алгоритм шифрування	Літ.	Маса / Маштаб
Змн.	Лист	№ документа	Підпис	Дата			
Розроб.		Клещев М.А.					1
Пеоевіо		Протасова Т.О.				Аркуш. 45	Аркушів. 48
Т. Контр.							
		.				СумДУ, ТК-61	
Н. Контр.		Протасова Т.О.					
Утверд.		Опанасюк А.С.					

Додаток В



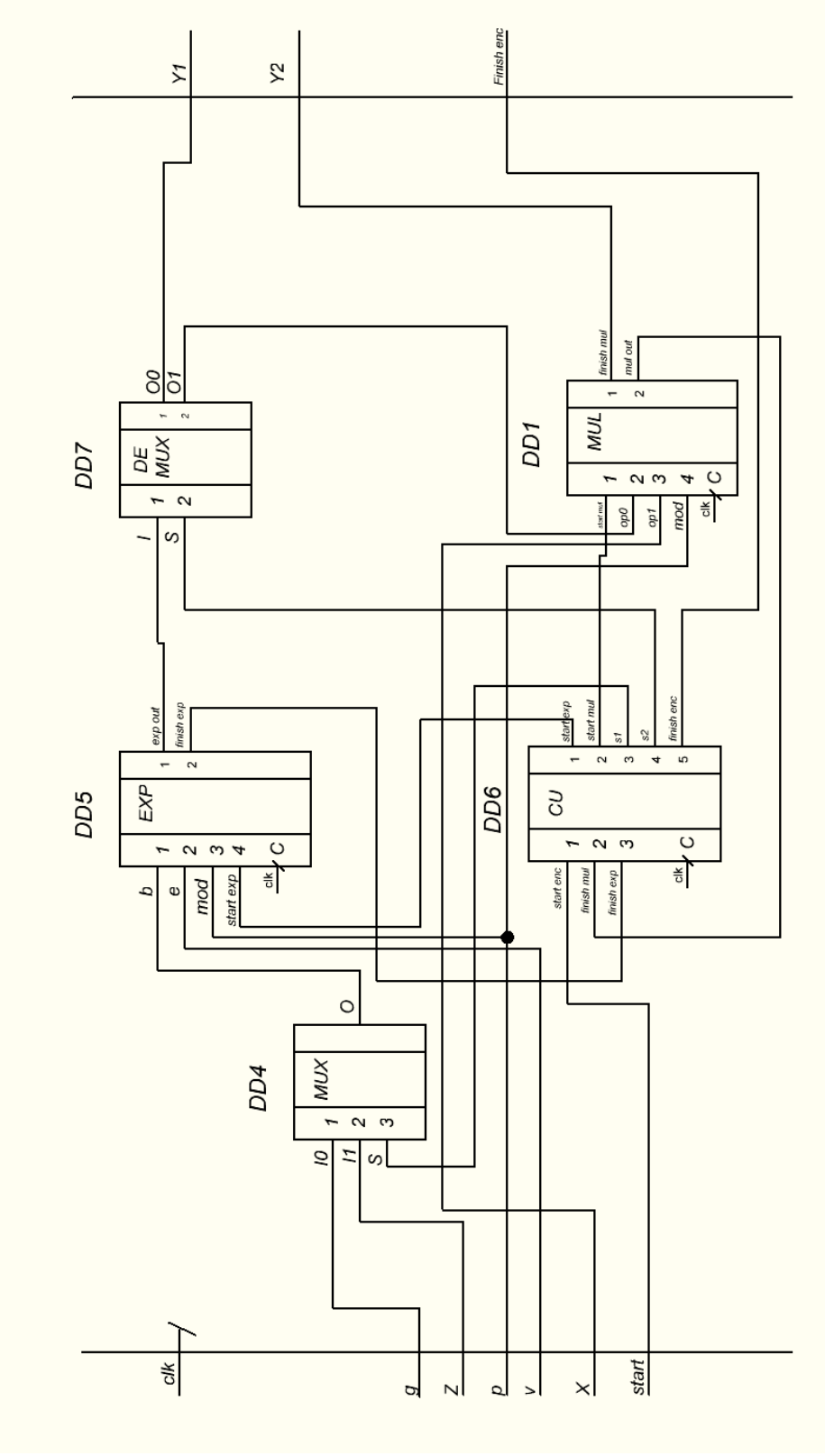
					ЕлІТ 6.172.00.10.533	
Змн.	*Лист*	*№ документа*	*Підпис*	*Дата*	Пристрій захисту інформації методом Ель-Гамаля. Алгоритм дешифрування	*Літ.* / *Маса* / *Маштаб*
Розроб.		*Клещев М.А.*				
Пеоевіо		*Протасова Т.О.*				
Т. Контр.						*Аркуш. 46* / *Аркушів. 48*
		.				
Н. Контр.		*Протасова Т.О.*				*СумДУ, ТК-61*
Утверд.		*Опанасюк А.С.*				

Додаток Г



					ЕлІТ 6.172.00.10.533	
Змн.	*Лист*	*№ документа*	*Підпис*	*Дата*	Пристрій захисту інформації методом Ель-Гамаля. Структурна схема пристрою шифування	*Літ.* / *Маса* / *Маштаб*
Розроб.		*Клещев М.А.*				
Пеоевіо		*Протасова Т.О.*				
Т. Контр.						*Аркуш. 47* / *Аркушів 48*
		.				
Н. Контр.		*Протасова Т.О.*				*СумДУ, ТК-61*
Утверд.		*Опанасюк А.С.*				

Додаток Д



					ЕлІТ 6.172.00.10.533	
Змн.	Лист	№ документа	Підпис	Дата	Пристрій захисту інформації методом Ель-Гамаля. Функіональна схема пристрою шифування	Літ. / Маса / Маштаб
Розроб.		Клещев М.А.				
Пеоевіо		Протасова Т.О.				
Т. Контр.						Аркуш. 48 / Аркушів 48
		.				
Н. Контр.		Протасова Т.О.				СумДУ, ТК-61
Утверд.		Опанасюк А.С.				