МІНІСТРЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Сумський державний університет
Факультет електроніки та інформаційних технологій
Кафедра електроніки і комп'ютерної техніки

«До захисту допущено»
Завідувач кафедри
_____Анатолій ОПАНАСЮК
(підпис)
_____ _____ 2023р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня бакалавра

зі спеціальності 172 «Телекомунікації та радіотехніка»,
освітньо-професійної програми «Мережеві та інтернет технології»
На тему: «Телекомунікаційний пристрій шифрування даних на основі алгоритму Ель Гамаля»

Здобувачки групи ТК-91 Приходіної Поліни Анатоліївни

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело.

_____ Поліна ПРИХОДІНА
(підпис)

Керівник, старший викладач,
кандидат фізико-математичних наук,
доцент _____ Олексій Д'ЯЧЕНКО
_____
(підпис)

Суми – 2023

ЗМІСТ

ВСТУП

У сучасному світі, де швидкість та безпека обміну даними є надзвичайно важливими, необхідність захисту конфіденційної інформації стає критичною. Особливо в контексті пандемії та війни, які відбуваються в Україні, забезпечення безпеки передачі даних та захисту приватності стає надзвичайно актуальною задачею. У таких умовах, використання телекомунікаційних пристроїв шифрування даних на основі алгоритму Ель-Гамаля стає необхідним для забезпечення безпеки комунікацій та збереження конфіденційної інформації.

Метою даної бакалаврської роботи є дослідження, розробка та реалізація телекомунікаційного пристрою шифрування даних на основі алгоритму Ель-Гамаля з метою забезпечення безпеки передачі даних. Об'єктом дослідження є сам алгоритм Ель-Гамаля та його використання в телекомунікаційних системах.

Електронні підписи є важливим елементом електронної комунікації та транзакцій, але їх безпека та недоступність для несанкціонованих осіб є найважливішими факторами. Використання телекомунікаційного пристрою шифрування даних на основі алгоритму Ель-Гамаля в електронних підписах дозволяє забезпечити надійну аутентифікацію та цілісність електронних підписів. Це означає, що при використанні телекомунікаційного пристрою шифрування на основі алгоритму Ель-Гамаля, дані, які підписуються, будуть захищені від несанкціонованого доступу, модифікації та підробки.

Застосування телекомунікаційних пристроїв шифрування даних на основі алгоритму Ель-Гамаля в електронних підписах має велике значення для різних сфер, включаючи фінансові установи, електронну комерцію, урядові організації та багато інших. Це дозволяє забезпечити довіру і надійність електронних транзакцій, підписаних за допомогою цих пристроїв.

Отже, розробка та використання телекомунікаційних пристроїв шифрування даних на основі алгоритму Ель-Гамаля має велике значення в контексті пандемії та війни в Україні, а також для забезпечення безпеки електронних підписів. Це дослідження принесе значний внесок у покращення захисту конфіденційної інформації та безпеки комунікацій, що є важливим у сучасному цифровому світі.

1 КРИПТОГРАФІЯ ТА ЕЛЕКТРОННІ ПІДПИСИ

1.1 Визначення криптографії та її роль у забезпеченні безпеки даних

Криптографія є наукою про захист інформації шляхом шифрування та розшифрування даних. Вона використовує математичні та алгоритмічні методи для забезпечення конфіденційності, цілісності та автентичності даних під час їх передачі та зберігання. Роль криптографії полягає в тому, щоб зробити інформацію незрозумілою та недоступною для несанкціонованих осіб, забезпечуючи тим самим безпеку даних.

Одним з важливих аспектів криптографії є електронний підпис. Електронний підпис використовується для підтвердження автентичності документа або повідомлення, а також забезпечення невідмовності автора від вмісту. Він складається з криптографічного хешу документа або повідомлення, який підписується приватним ключем особи. Цей підпис може бути перевірений за допомогою відкритого ключа, що забезпечує впевненість у тому, що документ або повідомлення не були змінені після підпису та що вони були створені автентичною особою.

Розробка та використання телекомунікаційних пристроїв шифрування даних на основі алгоритму Ель-Гамаля має велике значення в контексті криптографії та електронних підписів. Вони дозволяють захистити дані від несанкціонованого доступу та забезпечити автентичність та невідмовність підписаного вмісту. Це є важливим інструментом для захисту конфіденційної інформації та забезпечення безпеки електронних комунікацій та транзакцій.

1.2 Огляд електронних підписів та їх використання в бізнесі та фінансових установах

Криптографія та електронні підписи відіграють важливу роль в сфері бізнесу та фінансових установ, забезпечуючи безпеку та надійність електронних транзакцій. Огляд електронних підписів та їх використання в цих сферах є необхідним для розуміння їх практичного значення.

Електронний підпис - це електронний еквівалент звичайного підпису на папері, що підтверджує автентичність та цілісність документа або повідомлення. Використання електронних підписів дозволяє уникнути фальсифікації, змін або

відмови автора від підписаного вмісту. Вони базуються на криптографічних алгоритмах та використовують приватні та відкриті ключі для підпису та перевірки підпису.

У бізнесі та фінансових установах використання електронних підписів має кілька важливих переваг. По-перше, вони забезпечують швидку та зручну обробку документів та транзакцій, оскільки вони можуть бути виконані електронним шляхом без необхідності фізичної присутності. По-друге, вони забезпечують високий рівень безпеки, оскільки електронні підписи є важкозламними та важкопідробними. Це дозволяє знизити ризики фальсифікації та зловживання у фінансових операціях.

Крім того, електронні підписи допомагають уникнути зайвих витрат на друк та зберігання фізичних документів, а також зменшують негайність фізичного переміщення документів для їх підпису. Вони виявляються особливо корисними в ситуаціях, коли потрібно здійснити дистанційні операції, спілкування або підписання важливих угод. Електронні підписи сприяють ефективному та безпечному функціонуванню бізнес-процесів, спрощують взаємодію з клієнтами та партнерами, а також забезпечують довіру та легальність електронних документів.

Таким чином, огляд електронних підписів та їх використання в бізнесі та фінансових установах підкреслює їх важливість як інструменту для забезпечення безпеки, автентичності та ефективності електронних транзакцій. Ця технологія стає все більш поширеною у сучасному світі, де електронні комунікації та транзакції стають невід'ємною частиною нашого повсякденного життя.

Телекомунікаційні пристрої відіграють ключову роль у забезпеченні електронних підписів. Вони забезпечують безпечну комунікацію, зберігання та передачу електронних підписів, що використовуються для підтвердження автентичності та цілісності документів та повідомлень. Телекомунікаційні пристрої включають в себе мобільні пристрої, комп'ютери, смартфони, мережеві пристрої та інше обладнання, а також криптографічні модулі та програмне забезпечення для генерації та перевірки електронних підписів. Вони є необхідною інфраструктурою для впровадження електронних підписів у різних сферах, забезпечуючи надійність, автентичність та цілісність підписаних даних у цифровому середовищі.

2 АЛГОРИТМ ЕЛЬ-ГАМАЛЯ

2.1 Передумови створення алгоритму та його автор

Алгоритм Ель-Гамаля є одним з важливих асиметричних криптографічних алгоритмів, який був розроблений Др. Тахіром Ель-Гамалем у 1984 році. Цей алгоритм був створений як альтернатива алгоритму RSA, з метою покращення безпеки та забезпечення ефективного використання криптографії у сфері електронних комунікацій.

Др. Тахір Ель-Гамал є відомим криптографом, який народився в Іраку. Він отримав свою освіту у США, де вивчав математику та криптографію. В 1984 році він опублікував свою роботу "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", в якій вперше представив алгоритм Ель-Гамаля.

Алгоритм Ель-Гамаля базується на математичних властивостях дискретного логарифму та арифметиці над конечними полями. Його основна ідея полягає у використанні двох ключів: публічного ключа для шифрування даних та приватного ключа для розшифрування та підписування повідомлень.

Алгоритм Ель-Гамаля відкрив широкі можливості для застосування криптографії у телекомунікаційних системах, включаючи шифрування даних, цифровий підпис та безпечний обмін ключами. Він став основою для багатьох інших криптографічних протоколів та систем, що використовуються в сучасних телекомунікаційних мережах.

Алгоритм Ель-Гамаля завоював велику популярність та визнання у криптографічному співтоваристві завдяки своїм математичним основам, високому рівню безпеки та широкому спектру застосування. Його безпека базується на складності оберненого обчислення дискретного логарифму, що робить його вразливим до атак на основі розкладу на прості множники, які застосовуються у квантовій криптографії.

Алгоритм Ель-Гамаля знайшов широке застосування у різних сферах, включаючи захист інформації в електронних комунікаціях, електронні підписи, безпеку електронних транзакцій та зберігання конфіденційних даних. Його використання особливо актуально у контексті зростання кількості цифрових транзакцій та обміну даними, а також забезпечення приватності та безпеки в онлайн-середовищі.

Застосування алгоритму Ель-Гамаля в телекомунікаційних системах дозволяє забезпечити конфіденційність, цілісність та автентичність передаваних даних. Він дозволяє створювати безпечні канали зв'язку, шифрувати повідомлення та здійснювати електронний підпис, що використовується для підтвердження автентичності відправника та недоступності модифікації даних під час передачі.

Алгоритм Ель-Гамаля є важливим елементом криптографічної системи, що забезпечує безпеку та захист інформації у телекомунікаційних системах. Його розробка та використання відкриває широкі перспективи для розвитку безпечних та надійних систем електронних комунікацій у сучасному цифровому світі.

2.2 Важливі моменти у розвитку алгоритму

Алгоритм Ель-Гамаля, окрім своєї важливості та застосування в криптографії та телекомунікаціях, має кілька цікавих фактів та важливих моментів у своєму розвитку. Ось декілька з них:

а) Внутрішній зв'язок з дискретним логарифмом: Алгоритм Ель-Гамаля базується на складності обчислення дискретного логарифму в групі простого поля. Це одна з основних математичних операцій, на яких побудований алгоритм. Дискретний логарифм є складно обчислюваним, що забезпечує стійкість алгоритму проти атак зламу.

б) Потреба в безпечному обміні ключами: Алгоритм Ель-Гамаля вимагає безпечного обміну публічним ключем перед шифруванням або підписом даних. Це може бути складним завданням, оскільки потрібно забезпечити конфіденційність та цілісність переданого ключа. Рішення цієї проблеми включають в себе використання інших криптографічних протоколів або довірені центри для обміну ключами.

в) Практична складність атак: Алгоритм Ель-Гамаля вважається стійким до багатьох атак, включаючи атаки на основі підрахунку дискретного логарифму. Проте, існують певні атаки, такі як атака на основі перебору, які можуть бути ефективними при недостатньо великих ключах. Тому важливо вибирати достатньо довгі ключі для забезпечення безпеки алгоритму.

г) Застосування в електронних підписах: Один з важливих аспектів алгоритму Ель-Гамаля - його використання в електронних підписах. Алгоритм Ель-Гамаля може бути використаний для створення електронного підпису, який дозволяє перевірити автентичність та цілісність даних. Електронні підписи на основі алгоритму Ель-Гамаля знайшли широке застосування в електронній комерції, фінансових транзакціях та інших сферах, де безпека даних є критично важливою.

д) Розширення алгоритму: Алгоритм Ель-Гамаля має кілька розширень та варіацій, які розроблені для покращення його ефективності та безпеки. Наприклад, існують розширені версії алгоритму, що використовують еліптичні криві (еліптичну криптографію), які забезпечують більшу безпеку при коротших ключах. Також існують модифіковані варіанти алгоритму, які використовуються для специфічних застосувань та потреб.

е) Альтернативи алгоритму Ель-Гамаля: Незважаючи на те, що алгоритм Ель-Гамаля є ефективним та стійким, існують інші криптографічні алгоритми, які також використовуються для шифрування та електронних підписів, такі як RSA, Шифр Рівеста-Шамира-Адлемана та Еліптична криптографія. Вибір алгоритму залежить від конкретних вимог до безпеки, швидкодії та розміру ключа.

В цілому, алгоритм Ель-Гамаля має багато цікавих аспектів у своєму розвитку, від його математичних основ до застосування в різних сферах. Вивчення і розуміння цих аспектів розвитку алгоритму Ель-Гамаля допомагає нам краще оцінити його значення та потенціал у криптографії та телекомунікаціях. Використання цього алгоритму дозволяє забезпечити конфіденційність, цілісність та автентичність даних, що є критично важливими аспектами в цифровому світі.

Дослідження алгоритму Ель-Гамаля та його застосування у телекомунікаційних пристроях для шифрування даних та створення електронних підписів відкриває широкі перспективи для покращення безпеки та захисту інформації. Розуміння принципів роботи алгоритму, його сильних та слабких сторін, а також його взаємозв'язку з телекомунікаційними системами допомагає

розробляти нові методи та алгоритми для захисту даних у сучасному цифровому світі.

Таким чином, вивчення алгоритму Ель-Гамаля та його застосування у телекомунікаційних пристроях для шифрування даних належить до важливих напрямків досліджень у сфері криптографії та телекомунікацій. Результати цих досліджень можуть мати велике практичне значення для розвитку безпеки інформаційних систем і забезпечення конфіденційності та цілісності даних у сучасному цифровому середовищі.

2.3 Огляд інших відомих алгоритмів шифрування та їх переваги та недоліки

У порівняння з іншими алгоритмами шифрування та електронних підписів, алгоритм Ель-Гамаля має свої переваги та недоліки. Давайте розглянемо огляд деяких відомих алгоритмів шифрування та їх особливості:

- RSA (Rivest-Shamir-Adleman): RSA є одним з найпоширеніших алгоритмів шифрування та створення електронних підписів. Він базується на складності факторизації великих простих чисел. Основна перевага RSA полягає у високій швидкодії шифрування та простоті реалізації. Проте, RSA має певні обмеження, зокрема пов'язані з безпекою факторизації та розміром ключів.
- AES (Advanced Encryption Standard): AES є симетричним алгоритмом шифрування, що використовується для захисту конфіденційності даних. Він забезпечує високий рівень безпеки та швидкодії. Однак, AES вимагає спільного обміну секретним ключем між сторонами, що може бути складним у деяких сценаріях.
- ECC (Elliptic Curve Cryptography): ECC є асиметричним алгоритмом шифрування, який базується на обчисленнях на еліптичних кривих. Він відомий своєю високою безпекою та ефективністю застосування ключів малих розмірів. ECC дозволяє досягти такого ж рівня безпеки при використанні коротших ключів порівняно з іншими алгоритмами, що робить його особливо привабливим для обмежених ресурсів пристроїв.

Кожен з цих алгоритмів має свої переваги та недоліки залежно від конкретних вимог та сценаріїв використання. Ось кілька загальних переваг та недоліків алгоритму Ель-Гамаля у порівнянні з іншими алгоритмами:

Переваги алгоритму Ель-Гамаля:

- Висока безпека: Алгоритм Ель-Гамаля забезпечує високий рівень безпеки, оскільки базується на обчисленнях у складних математичних групах. Це робить його стійким до багатьох атак, включаючи факторизацію чисел.
- Асиметрична криптографія: Ель-Гамаль є асиметричним алгоритмом, що означає, що він використовує різні ключі для шифрування та розшифрування. Це дозволяє забезпечити безпеку передачі даних без потреби в обміні секретним ключем.
- Використання в електронних підписах: Алгоритм Ель-Гамаля є ефективним для створення електронних підписів, що забезпечує автентичність та цілісність даних.

Недоліки алгоритму Ель-Гамаля:

- Обчислювальна складність: Одним з недоліків Ель-Гамаля є висока обчислювальна складність порівняно з іншими алгоритмами шифрування, особливо при використанні довгих ключів. Це може вплинути на швидкодію операцій шифрування та розшифрування.
- Потреба в великому розмірі ключа: Ель-Гамаль вимагає використання ключів великого розміру для забезпечення безпеки. Це може мати вплив на об'єм пам'яті, необхідний для збереження, запам'ятовування та обробки ключів. Більші розміри ключів також можуть вплинути на швидкодію алгоритму.
- Вразливість до атак на базові примітиви: Алгоритм Ель-Гамаля може бути вразливим до атак, якщо базові примітиви, такі як генератори випадкових чисел, не використовуються належним чином або якщо ключі недостатньо випадкові.
- Складність реалізації: Реалізація алгоритму Ель-Гамаля може бути вимогливою та складною, особливо для новачків у галузі криптографії. Відповідне розуміння математичних концепцій та правильна імплементація можуть бути викликом.

Враховуючи ці переваги та недоліки, вибір алгоритму шифрування та електронних підписів залежить від конкретних вимог, потреб безпеки, швидкодії та ресурсів системи. Кожен алгоритм має свої унікальні особливості та застосування, тому важливо ретельно оцінити їх перед вибором для конкретного використання.

2.4 Порівняння алгоритму Ель-Гамаля з іншими алгоритмами щодо безпеки, ефективності та використання у телекомунікаційних системах

Алгоритм Ель-Гамаля має свої унікальні характеристики щодо безпеки, ефективності та використання у телекомунікаційних системах, порівняно з іншими алгоритмами шифрування та електронних підписів. Нижче наведено порівняння алгоритму Ель-Гамаля з деякими іншими відомими алгоритмами:

а) RSA (Rivest-Shamir-Adleman):

- Безпека: Якщо використовуються достатньо довгі ключі, обидва алгоритми забезпечують високий рівень безпеки. Однак, RSA вважається більш стійким до атак на основі факторизації чисел, в той час як Ель-Гамаль може бути вразливим до атак на дискретний логарифм.
- Ефективність: Шифрування та розшифрування RSA зазвичай є швидшими, ніж Ель-Гамаля, особливо при використанні великих чисел. Однак, Ель-Гамаля може бути швидшим для підписування повідомлень.
- Використання у телекомунікаційних системах: Обидва алгоритми широко використовуються для захисту комунікацій у телекомунікаційних системах, зокрема для шифрування даних та електронних підписів.

б) ECC (Elliptic Curve Cryptography):

- Безпека: ECC вважається більш ефективним з точки зору безпеки, оскільки забезпечує той самий рівень безпеки при коротших ключах порівняно з RSA і Ель-Гамаля.

- Ефективність: ECC відомий своєю високою ефективністю, оскільки вимагає меншу кількість обчислень для досягнення того самого рівня безпеки порівняно з RSA та Ель-Гамаля.
- Використання у телекомунікаційних системах: Використання ECC у телекомунікаційних системах стає все більш популярним завдяки своїй ефективності та малому розміру ключів. Він знаходить широке застосування у мобільних пристроях, де обмежені обчислювальні ресурси. ECC також використовується у протоколах безпеки, таких як SSL/TLS для захисту з'єднань у Інтернеті.

в) DSA (Digital Signature Algorithm):

- Безпека: DSA забезпечує сильну безпеку, але він може бути вразливим до атак на основі квантових обчислень. У порівнянні з Ель-Гамаля, DSA використовує менше обчислювальних ресурсів для отримання того самого рівня безпеки.
- Ефективність: Ефективність DSA подібна до Ель-Гамаля при підписуванні повідомлень, але DSA може бути трохи швидшим у випадку перевірки підпису.
- Використання у телекомунікаційних системах: DSA широко використовується для електронних підписів у телекомунікаційних системах, таких як автентифікація та забезпечення цілісності даних.

Кожен з цих алгоритмів має свої переваги та недоліки, і вибір конкретного алгоритму залежить від потреб безпеки, ефективності та вимог конкретного застосування у телекомунікаційних системах. Враховуючи унікальні особливості алгоритму Ель-Гамаля, він може бути вигідним в певних сценаріях, особливо з використанням дискретного логарифму та ефективних телекомунікаційних пристроїв.

3 ВИКОРИСТАННЯ АЛГОРИТМУ ЕЛЬ-ГАМАЛЯ В ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМАХ

3.1 Роль телекомунікаційних пристроїв з шифруванням на основі алгоритму Ель-Гамаля у сучасних комунікаційних мережах

У сучасних комунікаційних мережах телекомунікаційні пристрої з шифруванням на основі алгоритму Ель-Гамаля відіграють важливу роль у забезпеченні безпеки та конфіденційності передачі даних. Вони забезпечують захист інформації, що передається по мережі, від несанкціонованого доступу та перехоплення.

Роль телекомунікаційних пристроїв з шифруванням на основі алгоритму Ель-Гамаля полягає у виконанні наступних завдань:

- Шифрування даних: Телекомунікаційні пристрої використовують алгоритм Ель-Гамаля для шифрування переданих даних. Вони обчислюють криптографічні ключі, використовуючи приватний ключ і отримують публічний ключ для передачі. За допомогою публічного ключа, дані шифруються перед відправкою, що забезпечує їхню конфіденційність під час трансляції по мережі.
- Розшифрування даних: Телекомунікаційні пристрої також використовують алгоритм Ель-Гамаля для розшифрування отриманих зашифрованих даних. За допомогою приватного ключа, який залишається виключно власністю власника, пристрій може розкодувати шифровані дані, що дозволяє отримати оригінальну інформацію.
- Електронний підпис: Телекомунікаційні пристрої на основі алгоритму Ель-Гамаля також можуть використовуватися для створення електронних підписів. Електронний підпис забезпечує автентифікацію та цілісність даних, дозволяючи отримувачу перевірити, що дані не були змодифіковані та підписані власником приватного ключа. Телекомунікаційні пристрої можуть генерувати електронний підпис для документів, повідомлень або транзакцій, що забезпечує невідомість відправника і гарантує, що дані не були змінені під час передачі.

Телекомунікаційні пристрої з шифруванням на основі алгоритму Ель-Гамаля мають різні практичні застосування, такі як захист електронної пошти, мобільних додатків, фінансових транзакцій, корпоративних даних, IoT-пристроїв і мережевих з'єднань. Вони забезпечують конфіденційність, цілісність та недоступність інформації, підтверджують автентичність даних і створюють безпечні комунікаційні з'єднання.

Таким чином, телекомунікаційні пристрої з шифруванням на основі алгоритму Ель-Гамаля мають велике значення в телекомунікаційних системах. Вони забезпечують конфіденційність, цілісність та автентичність передачі даних. Шифрування на основі алгоритму Ель-Гамаля використовується в електронних підписах, зв'язку між користувачами, безпечних протоколах передачі даних та інших аспектах телекомунікаційних систем. Це сприяє підвищенню рівня безпеки та довіри в мережах зв'язку.

3.2 Застосування алгоритму Ель-Гамаля для шифрування та розшифрування даних у телекомунікаційних системах

Алгоритм Ель-Гамаля має широке застосування для шифрування та розшифрування даних у телекомунікаційних системах. Він забезпечує безпеку передачі інформації та конфіденційність даних, що є критичними аспектами в комунікаційних мережах. Деякі основні застосування алгоритму Ель-Гамаля включають:

а) Захищений обмін ключами: Алгоритм Ель-Гамаля може використовуватися для безпечного обміну секретними ключами між взаємодіючими сторонами. За допомогою алгоритму, дві сторони можуть створити спільний секретний ключ, який може бути використаний для подальшого шифрування та розшифрування даних.

б) Шифрування повідомлень: Алгоритм Ель-Гамаля дозволяє шифрувати повідомлення перед їхньою передачею через комунікаційну мережу. Відправник може використовувати публічний ключ отримувача для шифрування повідомлення, тоді як отримувач використовує свій приватний ключ для розшифрування його.

в) Електронні підписи: Алгоритм Ель-Гамаля також може використовуватися для створення електронних підписів. Це дозволяє власнику приватного ключа підписувати повідомлення або документи, щоб підтвердити їх автентичність та цілісність. Отримувач може перевірити підпис, використовуючи відповідний публічний ключ.

г) Захист інформації у мобільних та безпровідних мережах: Алгоритм Ель-Гамаля може бути використаний для захисту інформації у мобільних та безпровідних мережах. Завдяки своїй безпеці та можливості застосування на різних платформах, алгоритм Ель-Гамаля є важливим інструментом для шифрування та розшифрування даних у телекомунікаційних системах.

д) Захист приватної інформації у хмарних обчисленнях: У хмарних обчисленнях, де дані зберігаються та обробляються на віддалених серверах, захист приватності є критичним. Алгоритм Ель-Гамаля може бути використаний для шифрування даних перед їхнім збереженням у хмарних сервісах, що забезпечує додатковий рівень безпеки та конфіденційності.

е) Безпечна передача даних у веб-протоколах: У веб-протоколах, таких як HTTPS, застосування алгоритму Ель-Гамаля дозволяє шифрувати дані, що передаються між веб-сервером та клієнтом. Це забезпечує безпеку передачі конфіденційної інформації, такої як особисті дані, фінансова інформація тощо.

Застосування алгоритму Ель-Гамаля у телекомунікаційних системах дозволяє забезпечити конфіденційність, цілісність та автентичність передачі даних. Він знайшов широке застосування в захищених комунікаційних протоколах, електронних підписах, мобільних та безпровідних мережах, хмарних обчисленнях та інших сферах, де безпека даних є пріоритетом.

3.3 Використання алгоритму для забезпечення електронних підписів та підтвердження автентичності даних у комунікаціях

Алгоритм Ель-Гамаля також може бути використаний для забезпечення електронних підписів та підтвердження автентичності даних у комунікаціях. Електронний підпис використовується для підтвердження, що певний документ

або повідомлення були створені конкретним відправником і не піддалися зміні під час передачі.

При використанні алгоритму Ель-Гамаля для електронних підписів, відправник генерує свій приватний ключ, який використовується для підпису повідомлення, і публічний ключ, який розповсюджується серед отримувачів. Для підписування повідомлення використовується функція хешування, яка перетворює повідомлення на унікальний хеш-код. Потім приватний ключ використовується для створення цифрового підпису, який прикріплюється до повідомлення.

Отримувачі повідомлення можуть перевірити автентичність даних, використовуючи публічний ключ відправника. Вони використовують публічний ключ для перевірки підпису, використовуючи той самий алгоритм Ель-Гамаля. Якщо підпис співпадає зі згенерованим хеш-кодом повідомлення, то це підтверджує, що повідомлення не було змінене під час передачі і є автентичним. Принцип роботи алгоритму Ель- Гамаля наведено на рисунку 3.1.

Застосування алгоритму Ель-Гамаля для електронних підписів має декілька переваг. По-перше, цей алгоритм забезпечує високий рівень безпеки. Він базується на обчислювально складних операціях, таких як дискретний логарифм, що робить його важким для зламування. Крім того, алгоритм Ель-Гамаля володіє операційною незалежністю, що дозволяє ефективно використовувати його у різних телекомунікаційних системах.

Додатковою перевагою використання алгоритму Ель-Гамаля для електронних підписів є його відкритий ключ, який може бути розповсюджений серед багатьох користувачів. Це спрощує процес підписання та перевірки підпису, оскільки відправник не повинен зберігати приватний ключ для кожного одержувача.

Отже, використання алгоритму Ель-Гамаля для електронних підписів дозволяє забезпечити автентичність та цілісність даних у комунікаціях. Він надає високий рівень безпеки, ефективність та можливість використання у різних телекомунікаційних системах.

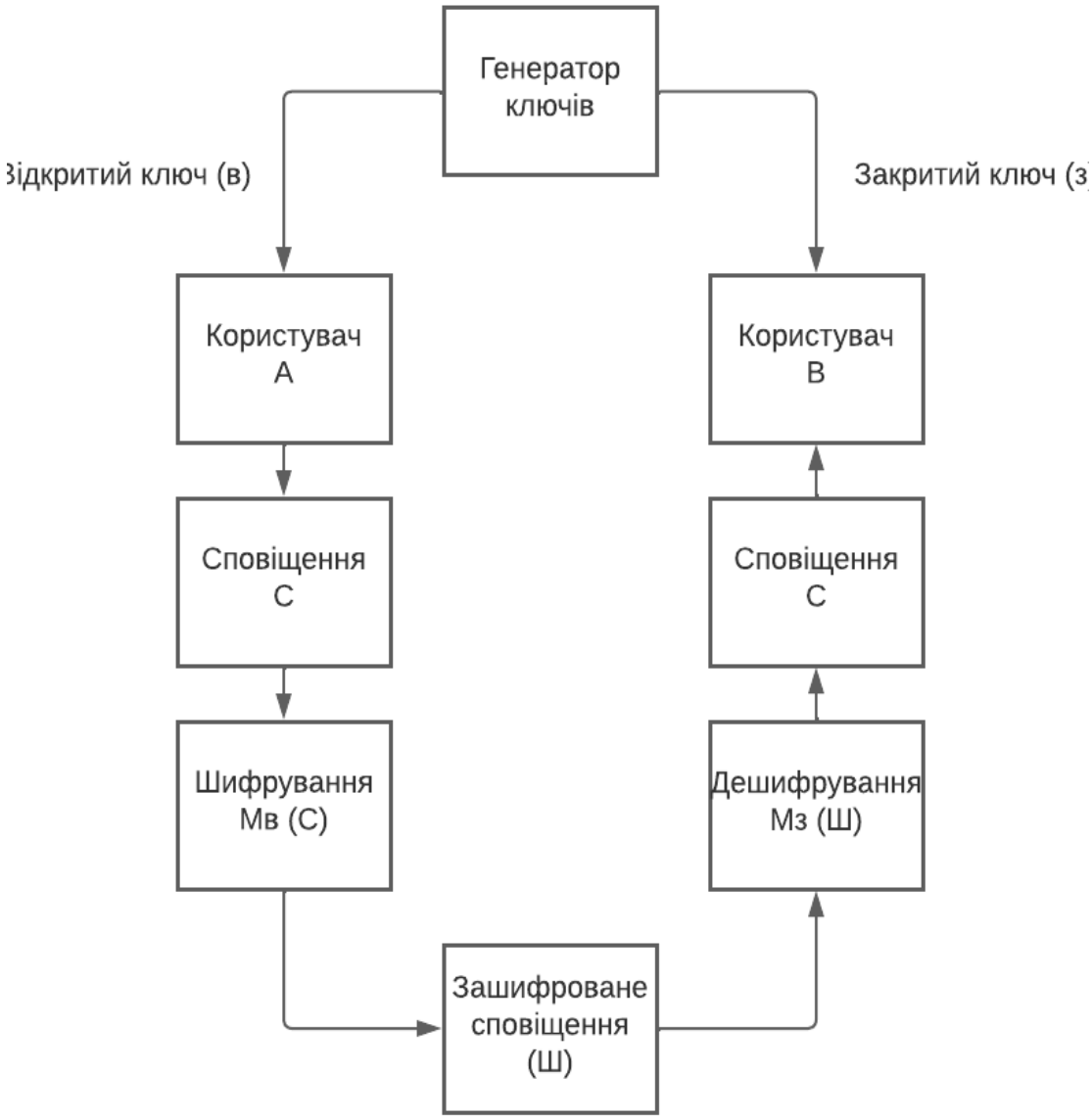

Рисунок 3.1 – Принцип роботи алгоритму Ель-Гамаля

4 АНАЛІЗ БЕЗПЕКИ АЛГОРИТМУ ЕЛЬ-ГАМАЛЯ У ТЕЛЕКОМУНІКАЦІЙНИХ ПРИСТРОЯХ

4.1 Виявлення потенційних загроз та вразливостей алгоритму Ель-Гамаля

Аналіз безпеки алгоритму Ель-Гамаля у телекомунікаційних пристроях включає виявлення потенційних загроз та вразливостей, які можуть вплинути на безпеку його застосування. Деякі з основних аспектів, що потребують уваги, включають:

а) Вразливості алгоритму: Ель-Гамаля може бути вразливий до атак, таких як перебір, статистичні атаки, факторизація чисел та інші. Аналіз безпеки має на меті виявлення вразливостей та розробку захисних заходів.

б) Розмір ключів: Ефективність Ель-Гамаля залежить від розміру ключів. Малі ключі можуть бути вразливими до перебірних атак, а великі ключі можуть сповільнити телекомунікаційні пристрої. Оптимальний розмір ключів повинен забезпечувати безпеку та ефективність.

в) Захист приватного ключа: Безпека Ель-Гамаля залежить від захисту приватного ключа. Компрометація приватного ключа може призвести до порушення безпеки та розкриття конфіденційної інформації. Необхідно надійно захищати приватний ключ у телекомунікаційних пристроях.

г) Можливість аналізу трафіку: Зловмисники можуть аналізувати шифрований трафік, що проходить через Ель-Гамаля, і здобувати інформацію про ключі та параметри. Це загрожує конфіденційності та безпеці передачі даних. Важливо запобігати аналізу трафіку та забезпечувати захист конфіденційної інформації.

д) Реалізаційні атаки: Ель-Гамаля може бути вразливий до атак, які використовують помилки у реалізації. Недоліки в програмному забезпеченні, генерація ключів або некоректне виконання можуть створити вразливості. Важливо правильно та безпечно реалізувати Ель-Гамаля.

е) Квантові обчислення: З'явлення квантових комп'ютерів створює загрозу для алгоритмів шифрування, включаючи Ель-Гамаля. Квантові комп'ютери можуть зламати математичні проблеми, які забезпечують безпеку Ель-Гамаля. Це вимагає розробки нових криптографічних методів, щоб забезпечити стійкість шифрування в епоху квантових обчислень.

Аналіз безпеки алгоритму Ель-Гамаля у телекомунікаційних пристроях допомагає виявити загрози та розробити заходи для забезпечення безпеки передачі даних. Це робить алгоритм Ель-Гамаля надійним і безпечним для використання. Аналіз включає тестування стійкості, виявлення вразливостей та розробку контрзаходів для забезпечення безпеки передачі даних. Контрзаходи можуть включати додаткові шари захисту, довші ключі, вдосконалення генерації ключів та перевірку безпеки алгоритму.

4.2 Огляд заходів забезпечення безпеки для захисту використання алгоритму

Огляд заходів забезпечення безпеки для захисту використання алгоритму Ель-Гамаля в телекомунікаційних пристроях включає:

- Ключове управління: Ефективне керування ключами є важливим для забезпечення безпеки алгоритму Ель-Гамаля. Це включає безпечні протоколи ключового обміну, шифрування ключів і системи управління ключами.
- Захист від атак на ключі: Ключі шифрування повинні бути захищені від різних видів атак, таких як криптоаналітичні атаки, підслуховування ключів і атаки на структуру алгоритму. Генерація безпечних ключів і регулярне їх оновлення є важливими для запобігання складним атакам.
- Захист від побічних каналів: Алгоритм Ель-Гамаля може бути вразливим до побічних каналів, таких як енергозалежний аналіз, електромагнітний аналіз та аналіз часу виконання. Застосування технік, таких як затримка, маскування і фільтрація шуму, може допомогти зменшити витік інформації через побічні канали.
- Аутентифікація та контроль доступу: Встановлення механізмів аутентифікації та контролю доступу є важливим для забезпечення безпеки використання алгоритму Ель-Гамаля. Це може включати використання цифрових підписів, сертифікатів, протоколів аутентифікації та механізмів перевірки цілісності даних.
- Захист від відомих атак: Врахування можливих атак, таких як атаки на підсилення публічного ключа, атаки на перебір ключа і атаки на витік інформації, повинно бути частиною заходів забезпечення безпеки для

алгоритму Ель-Гамаля. Застосування випадкових чисел, безпечних хеш-функцій і регулярний аудит безпеки можуть допомогти у запобіганні відомим атакам.

- Шифрування комунікаційного каналу: Захист комунікаційного каналу, через який передаються зашифровані дані, є важливим аспектом безпеки алгоритму Ель-Гамаля. Використання шифрування на рівні транспортного протоколу, наприклад TLS (Transport Layer Security), може забезпечити додатковий рівень захисту шифрованих даних під час їх передачі.
- Аналіз і оновлення безпеки: Постійний аналіз безпеки алгоритму Ель-Гамаля та його впровадження у телекомунікаційні пристрої є важливим кроком для забезпечення безпеки. Слід відстежувати нові відомості про вразливості та атаки на алгоритм, оновлювати безпекові протоколи та алгоритми за необхідності та вживати відповідних заходів для запобігання новим загрозам безпеки.

В цілому, використання алгоритму Ель-Гамаля в телекомунікаційних пристроях вимагає комплексного підходу до забезпечення безпеки.

4.3 Аналіз перспектив та майбутнього розвитку використання алгоритму Ель-Гамаля в телекомунікаційних системах

Аналіз перспектив та майбутнього розвитку використання алгоритму Ель-Гамаля в телекомунікаційних системах є важливим аспектом, оскільки шифрування та забезпечення безпеки даних стають все більш критичними у сучасному цифровому світі. Деякі перспективи та напрями розвитку використання алгоритму Ель-Гамаля в телекомунікаційних системах включають:

- Покращення швидкодії алгоритму: Запровадження оптимізаційних технік та алгоритмічних покращень для зниження обчислювальної складності алгоритму та підвищення ефективності його використання в телекомунікаційних системах з великим обсягом даних.
- Інтеграція з іншими криптографічними примітивами: Поєднання алгоритму Ель-Гамаля з іншими криптографічними примітивами,

такими як симетричне шифрування, хеш-функції або інші алгоритми підпису, для створення комплексних систем шифрування та аутентифікації.

- Використання у квантових системах: Розробка квантово-стійких варіантів алгоритму Ель-Гамаля, що відповідають вимогам безпеки в квантовому світі, з огляду на появу квантових комп'ютерів та квантової криптографії.
- Застосування в розподілених системах: Використання алгоритму Ель-Гамаля для забезпечення конфіденційності та цілісності даних під час їх передачі між різними вузлами або довіреними сторонами в розподілених системах.
- Адаптація до мобільних пристроїв: Розробка оптимізованих версій алгоритму Ель-Гамаля, призначених для обмежених ресурсів мобільних пристроїв, що дозволить забезпечити безпеку комунікацій на мобільних платформах із збереженням ефективності.
- Стандартизація та інтеграція: Робота над стандартизацією та інтеграцією алгоритму Ель-Гамаля в різноманітні протоколи та системи безпеки, що сприятиме його широкому використанню та забезпеченню сумісності між різними рішеннями та продуктами, що використовуються в телекомунікаційних системах.
- Дослідження нових математичних основ: Вивчення нових математичних структур та методів, які можуть покращити безпеку алгоритму Ель-Гамаля та призвести до винайдення нових варіацій, що забезпечують ще більшу безпеку та ефективність.
- Адаптація до квантових обчислювальних систем: Розробка варіантів алгоритму Ель-Гамаля, що відповідають вимогам безпеки в квантовому середовищі, враховуючи швидкий розвиток квантових обчислювальних систем.
- Застосування у розподілених реєстрах: Використання алгоритму Ель-Гамаля в розподілених реєстрах, наприклад, в блокчейні, для забезпечення безпеки та аутентифікації транзакцій в таких системах.

Аналіз використання алгоритму Ель-Гамаля в телекомунікаціях показує потенціал його розвитку. Продовження досліджень та інновацій може поліпшити ефективність та безпеку алгоритму. Нові математичні основи, інтеграція з

іншими примітивами та адаптація до нових вимог є важливими аспектами розвитку. Крім того, використання його в квантових системах та розподілених реєстрах відкриває нові можливості. Розвиток алгоритму є перспективним, забезпечуючи надійне шифрування та аутентифікацію. Дослідження, стандартизація та інтеграція допоможуть підтримувати його актуальність та ефективність.

5 РОЗРОБКА СТРУКТУРНОЇ СХЕМИ ТА АЛГОРИТМ ФУНКЦІОНУВАННЯ ПРИСТРОЮ, ЩО РЕАЛІЗУЄ АЛГОРИТМ ЕЛЬ-ГАМАЛЯ

5.1 Опис принципу роботи алгоритму Ель-Гамаля

Алгоритм Ель-Гамаля базується на математичних властивостях дискретного логарифму та арифметиці над конечними полями. Принцип роботи алгоритму Ель-Гамаля можна розділити на кроки шифрування та розшифрування, які забезпечують безпеку та конфіденційність даних, що передаються.

Шифрування: Користувач, який бажає надіслати повідомлення, обирає випадкове число (криптографічний ключ) із певного діапазону.

За допомогою цього числа обчислюється публічний ключ за формулою: публічний ключ = g^ криптографічний ключ mod p, де g є генератором групи простого поля p.

Повідомлення перетворюється на числову форму, наприклад, за допомогою кодування символів або хеш-функцій.

Для кожного повідомлення обчислюється шифрований текст шляхом застосування наступних формул:

a = g^r mod p, де r є випадковим числом з того ж діапазону, що і криптографічний ключ.

b = (повідомлення * (публічний ключ)^r) mod p.

Отримані a та b є шифрованим текстом, який можна надіслати отримувачу.

Розшифрування: Отримувач отримує шифрований текст (a, b).

Використовуючи свій приватний ключ (криптографічний ключ), отримувач обчислює спільний ключ за формулою: спільний ключ = (a^(-криптографічний ключ)) mod p.

За допомогою спільного ключа отримувач відновлює повідомлення за формулою: повідомлення = (b * (спільний ключ^(-1))) mod p, де ^(-1) позначає обернене значення модуля p.

Отримане повідомлення може бути перетворене у текстовий або символьний вигляд для подальшого розуміння та використання отриманої інформації. На рисунку 5.1 представлена схема алгоритму, а на рисунку 5.2- структурна електрична схема.

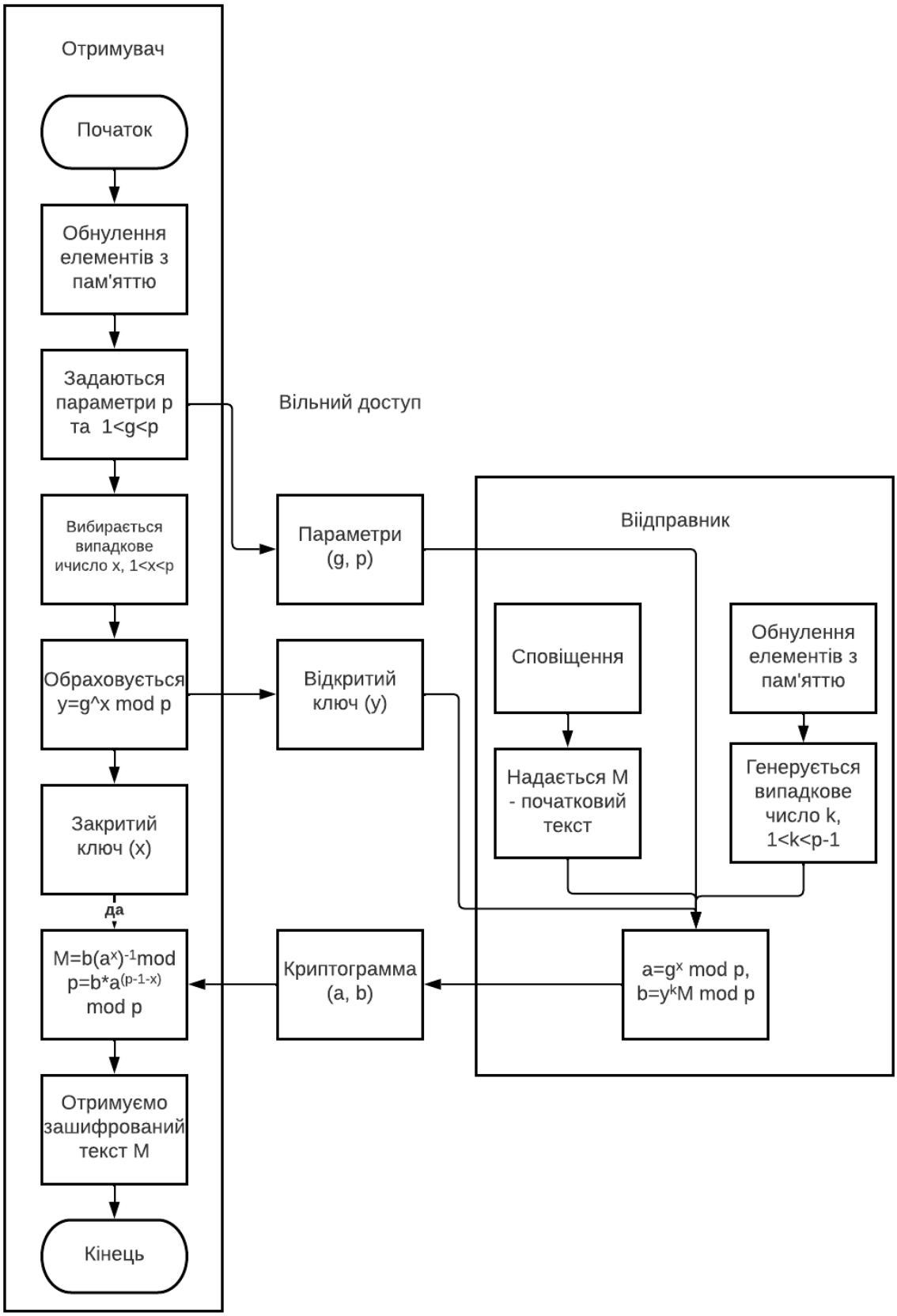

Рисунок 5.1 - Розробка схеми алгоритму

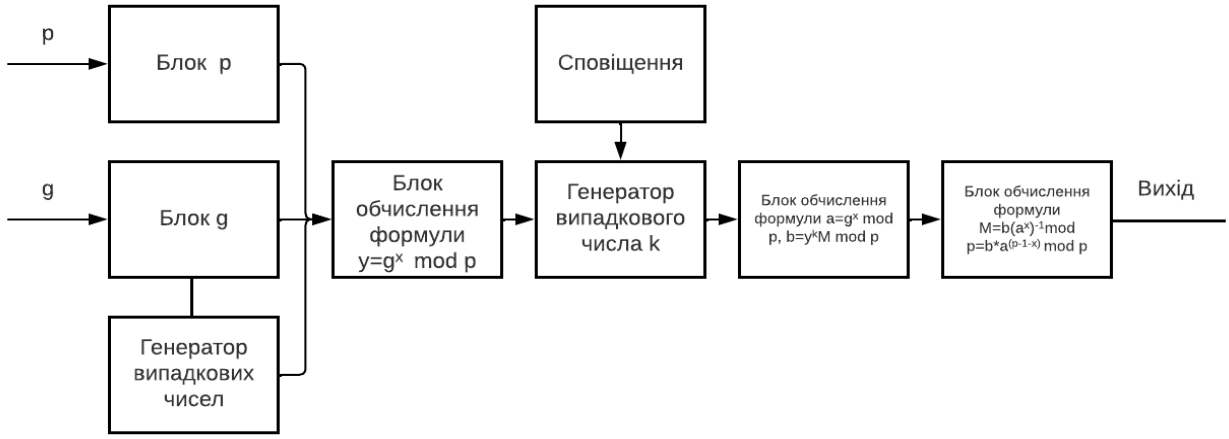

Рисунок 5.2 - Розробка структурної електричної схеми

Принцип роботи алгоритму Ель-Гамаля забезпечує безпеку шляхом використання математичних властивостей, які роблять його важким для зламу атаками, такими як підбір ключа чи перебір значень. Криптографічна стійкість алгоритму базується на складності оберненого обчислення дискретного логарифму в групі простого поля.

Алгоритм Ель-Гамаля має широке застосування у сферах, де конфіденційність та безпека даних є критичними, таких як електронні підписи, безпека електронних транзакцій, обмін конфіденційною інформацією тощо. Він дозволяє створювати безпечні канали зв'язку, шифрувати повідомлення та здійснювати електронні підписи, що забезпечують автентичність, цілісність та недоступність модифікації даних під час їх передачі.

Приклад. Припустимо, ми хочемо зашифрувати повідомлення "Hello" за допомогою алгоритму Ель-Гамаля з використанням наступних параметрів:

- Просте число p = 23 (використовується як загальний модуль)
- Примітивний корінь g = 5 (вибирається як генератор групи)
- Випадкове ціле число x = 6 (приватний ключ)
- Відкритий ключ y = (g^x) mod p = (5^6) mod 23 = 8 (публічний ключ)

Перший крок - шифрування:

а) Обираємо випадкове ціле число k = 3.

б) Обчислюємо a = (g^k) mod p = (5^3) mod 23 = 10.

в) Обчислюємо b = (y^k * m) mod p, де m - числове представлення повідомлення.

Припустимо, що числове представлення повідомлення "Hello" дорівнює 10.

b = (8^3 * 10) mod 23 = (512 * 10) mod 23 = 15.

Тепер ми отримали шифрований текст (a, b) = (10, 15).

Другий крок - розшифрування:

а) Обчислюємо s = (a^x) mod p = (10^6) mod 23 = 3.

б) Знаходимо обернене число s^(-1) mod p.

в) За допомогою алгоритму розширеного алгоритму Євкліда, ми знаходимо, що s^(-1) mod p = 16.

г) Розшифровуємо повідомлення m = (b * s^(-1)) mod p = (15 * 16) mod 23 = 12.

Таким чином, розшифроване повідомлення дорівнює 12, що відповідає числовому представленню повідомлення "Hello".

Це простий приклад роботи алгоритму Ель-Гамаля з використанням невеликих чисел. У реальних застосуваннях використовуються набагато більші числа.

Алгоритм Ель-Гамаля є одним з важливих алгоритмів в сучасній криптографії та телекомунікаційних системах. Його ефективність та стійкість до криптоаналізу роблять його привабливим вибором для забезпечення безпеки даних в різних сферах життя та бізнесу. Розуміння принципу роботи алгоритму Ель-Гамаля є важливим для розробки та використання безпечних систем телекомунікацій та забезпечення конфіденційності та цілісності даних.

На малюнках 5.3 та 5.4 наведено структурна електрична схема та блок схема алгоритму для конкретного прикладу.

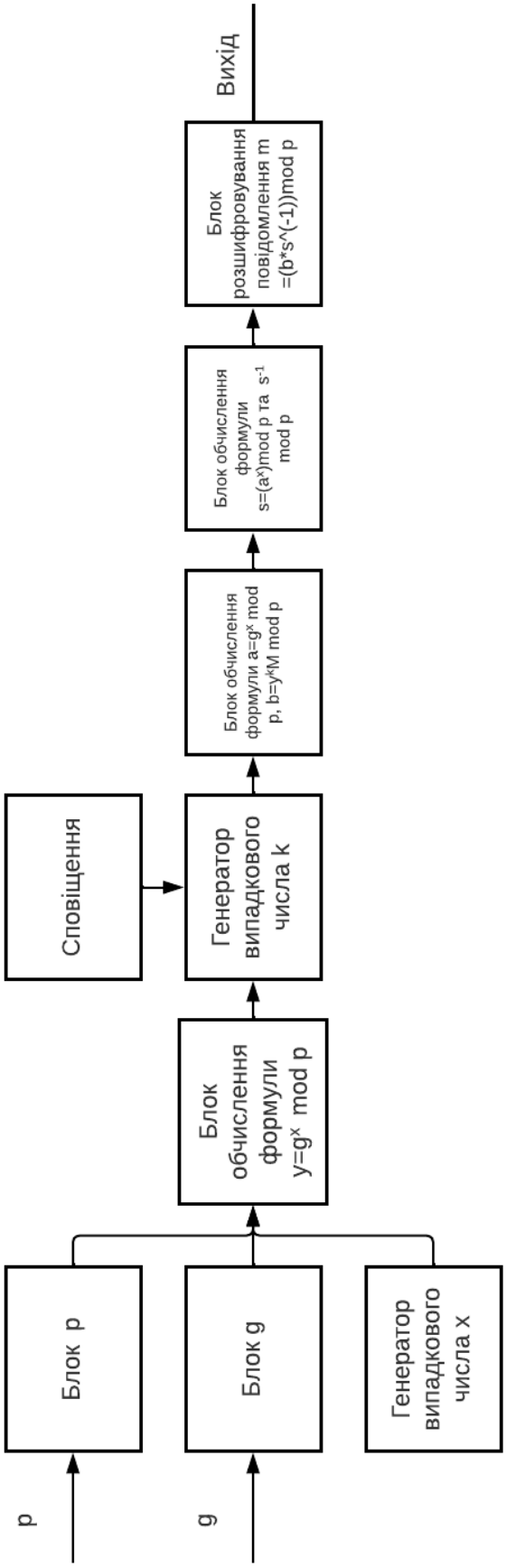

Рисунок 5.3 - Структурна електрична схема

Рисунок 5.4 – Блок- схема лгоритму

6 СИНТЕЗ ВУЗЛА ВВЕДЕННЯ В СТУПЕНЬ

6.1 Розробка алгоритму функціонування та функціональної схеми вузла зведення в ступінь

Зведення в ступінь за модулем - одна з операцій над натуральними числами - зведення в ступінь, - виконується за модулем. Знаходить застосування в інформатиці, особливо в галузі криптографії з відкритим ключем.

Зведення в ступінь за модулем - це обчислення залишку від ділення натурального числа g (основа), зведеного в ступінь x (показник ступеня), на натуральне число p (модуль). Позначається:$y = g^x (mod\ p)$.

Розглянемо докладно операцію піднесення до степені.

Розглянемо приклад.

Нехай необхідно виконати операцію: 5^4 – це значить, що п’ятірку необхідно помножити на саму себе чотири рази, тобто все буде виглядати так:

							1	0	1
						*	1	0	1
							1	0	1
					+	0	0	0	
				+	1	0	1		
		5*5=25			1	1	0	0	1
				*			1	0	1
					1	1	0	0	1
			+	0	0	0	0		
		+	1	1	0	0	1		
	25*5=12		1	1	1	1	1	0	1
	5								
		*					1	0	1
			1	1	1	1	1	0	1
	+	0	0	0	0	0			
+	1	1	1	1	1	0	1		
1	0	0	1	1	1	0	0	0	1

Останній рядок таблиці відповідає значенню 125*5=625 – тобто 5^4.

Таким чином, для операцію віднесення до степені можна замінити додаванням вмісту регістрів, при чому на кожному кроці вміст регістру треба зсувати на один такт.

Крім того, необхідно контролювати розрядність числа, для якого знаходиться ступінь, а також перевіряти чи вже виконався крок, що контролює необхідне значення степені.

Таким чином, необхідні регістри, куди перед початком перетворення необхідно записати значення параметрів:

- g – основа, тобто число, яке підноситься до степені;
- n – розрядність цього числа, тобто кількість розрядів;
- x – показник ступеня.

Доцільніше буде вибрати паралельні регістри. Перед початком виконання операції за сигналом «Пуск» в регістри запишуться відповідні значення. В пристрої також необхідний регістр зсуву, він буде виступати основним елементом, а також в складі накопичувача буде ще один регістр – паралельний регістр результату виконання операцій. В нього буде записуватися сума – результат додавання з виходів суматора, але тільки в тому випадку, якщо множення відбувалося на одиницю, якщо ж відповідний розряд, на який множили був рівним нулю – запису результату не буде.

Таким чином перед початком роботи в регістри записані відповідні значення. При цьому необхідно скинути в нуль як робочий регістр зсуву, так і вихідний регістр результату.

Операцією множення $g * g$ буде керувати лічильник розрядів, в якості якого вибираємо звичайний лічильник додавання, а нарощення операції множення, тобто саме піднесення до степені буде контролювати лічильник степені – також лічильник додавання.

Операція піднесення до степені в даному випадку буде реалізована як операція множення, повторення декілька разів.

Перший крок – це реалізація $g * g$.

В регістр зсуву записується число g, одночасно це число знаходиться на входах даних керуючого мультиплексора. З виходів лічильника розрядів на адресні входи подається сигнал 00, який дозволяє появу значення молодшого розряду числа g на виході мультиплексора. Одночасно на входах суматора знаходиться число g и нуль з виходів регістра результатів. Комбінаційний суматор виконує операцію додавання, ця операція буде постійно виконуватися суматором. А от необхідно записати цей результат в регістр додавання чи ні буде залежати від значення відповідного розряду числа g. Якщо розряд дорівнює

одиниці сума запишеться в накопичувач, в противному разі – ні. Але в будь якому разі в регістрі зсуву відбудеться зсув вмісту регістра.

Відбувається перехід до наступного розряду числа g, тепер на виходах лічильника розрядів з'являється наступне значення 01, це значення подається на адресні входи мультиплексора, дозволяючи появі на його виході значення наступного розряду числа g. Це значення дозволяє чи забороняє запис результату додавання в накопичувач. Ця операція відбувається аналогічно розглянутій. Кроки повторюються поки не будуть проаналізовані всі розряди числа g. Тобто виконана перша операція, реалізовано добуток числа g на себе. Кінцем цієї операції є поява на виході схеми співпадіння, яка аналізує вміст лічильника розрядів і порівнює його з значенням, записаним в регістрі розрядів. В процесі виконання операції множення на виході компаратора знаходиться значення «менше» і тільки в кінці операції – «дорівнює». За цим сигналом відбувається скидання лічильника розрядів в нуль, одночасно цей імпульс подається на вхід лічильника степені збільшуючи його значення на одиницю. Таким чином виконується перехід до наступного значення степені. Вміст лічильника степені також перевіряється. Якщо його значення співпаде з значенням степені, записаним в регістрі степені, то операція надійшла до свого логічного завершення, вся схема скидається в нуль і готова до запису наступних значень. Якщо ж вміст лічильника степені менше значення, записаного в регістрі степені, то операція повторюється. При переході до значення лічильника ступені «+1» регістр зсуву встановлюється в нульовий стан, потім в нього переписується значення регістру результату. Далі операція відбувається аналогічно розглянутій раніше.

На рисунку 6.1 наведена докладна схема алгоритму, а на рисунку 6.2 – функціональна схема, що реалізує розглянутий алгоритм.

Добавить сюда схемы 6.1 и 6.2.

6.2 Розробка схеми електричної принципової

6.2.1 Вибір елементної бази. Метою вибору елементної бази є обґрунтування серії (або серій) інтегральних мікросхем, а також інших електрорадіоелементів, необхідних для раціональної реалізації пристрою, що

проектується.

Критеріями вибору серії (серій) ІМС є:

- наявність необхідних функціональних вузлів у складі серії ІМС;
- мала споживана потужність;
- виконання вимог по швидкодії (граничній робочій частоті) і умовам експлуатації;
- низька вартість;
- можливість керувати необхідними елементами, наприклад, індикаторами без додаткових підсилень і перетворень вихідних сигналів і т.п.

Вибір елементної бази необхідно проводити в наступній послідовності:

- за функціональною схемою пристрою визначаються необхідні функціональні вузли (лічильники, регістри, шифратори, перетворювачі коду тощо) та їх параметри;
- по довідниках визначаються серії ІМС, що містять всі або частину відповідних функціональних вузлів. При відсутності функціональних вузлів визначається можливість їх побудови за допомогою вхідних до складу серії елементів;
- на основі проведеного аналізу визначається одна або декілька серій, що застосовуються для побудови пристрою.

При виборі дискретних елементів (індикаторів, електромагнітних реле і т.д.), які входять до складу пристрою, що проектується, доцільно використовувати ті, які керуються сигналами з мікросхем або спеціальними мікросхемами сполучення, що входять до складу серій. Інакше проводиться розрахунок схем сполучення на дискретних елементах.

Вибір елементної бази доцільно ілюструвати таблицями, наприклад:

- таблиця відповідності складу серій потрібним функціональним вузлам і можливість реалізації функціональних елементів на дискретних логічних елементах серії;
- таблиця характеристик обраних серій ІМС;
- таблиця характеристик необхідних дискретних елементів.

На підставі аналізу даних таблиць проводиться вибір елементної бази.

Для реалізації вузлів та блоків принципової схеми будемо застосовувати серію мікросхем К1533, оскільки вона повністю задовольняє нас за своїми характеристиками – достатня швидкодія та незначне споживання потужності.

6.2.2 Вибір регістрів числа, розрядів та степені. Доцільно в якості

названих регістрів застосувати паралельні регістри, побудовані на тригерних збірках К 1533 ТМ8, які містять по чотири тригери затримки.

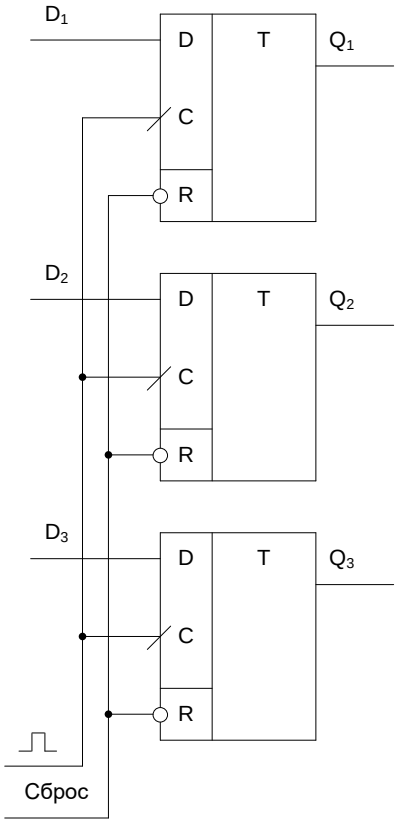

Рисунок 6.3 – Паралельний регістр

6.2.3 Вибір регістру зсуву. В якості регістру зсуву доцільніше буде вибрати паралельно-послідовний регістр, схема якого наведена на рисунку 6.4.

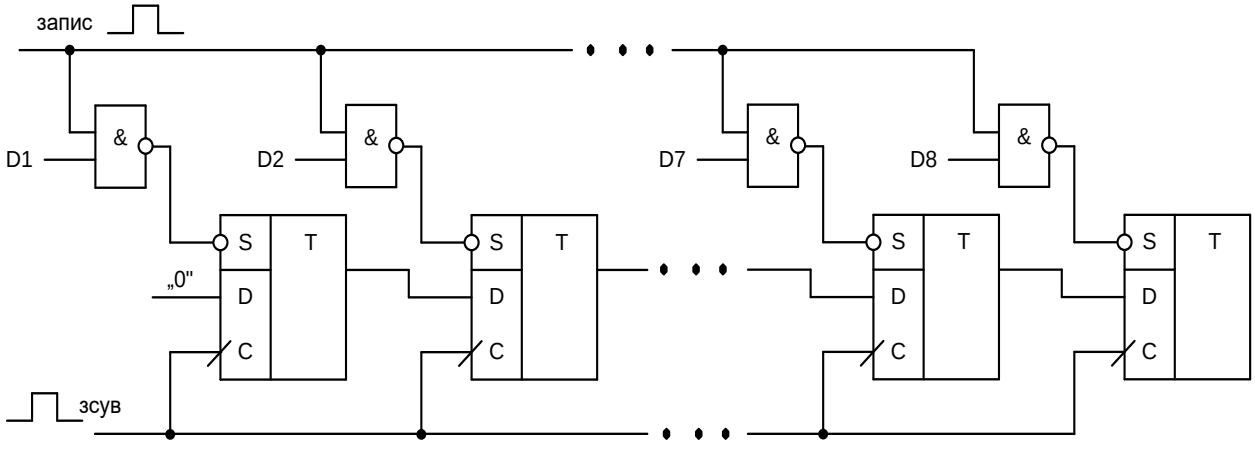

Рисунок 6.4 – Паралельно-послідовний регістр

По приходу сигналу «запис», що надходить одночасно на всі схеми І-НІ на виходах цих схем з'являються сигнали, протилежні значенням даних, що подаються на другі входи схем І-НІ. Ці сигнали надходять на входи більш високого рівня пріоритету - входи установки тригерів в одиничний стан. Так як керування цих входів інверсне, при рівності одиниці сигналу даних на виході схеми І-НІ з'являється нульовий сигнал, який і переводить тригер в потрібний одиничний стан. Оскільки при включенні всі елементи пам'яті примусово були встановлені в нульовий стан, то розряди відповідні вхідним сигналам, рівним одиниці встановлюються в одиничний стан, а інші залишаються в нулі. Після запису паралельного вхідного коду проводиться зсув інформації, подачею на синхровхід сигналу «зсув». Одночасно на вхід D крайнього лівого розряду подається нуль. При зсуві вправо на послідовному виході послідовно з'являтиметься двійкова кодова комбінація. Одночасно з кожним тактовим сигналом в регістр буде рухатись нульовий сигнал. Через *n* тактів (за кількістю розрядів вихідного числа) регістр заповниться нулями. Схема готова до прийому наступної двійковій кодової комбінації.

Паралельно-послідовний регістр доцільніше побудувати на мікросхемах К1533 ТМ2, оскільки тригери в цій мікросхемі мають як входи скидання, так і входи встановлення тригера в одиничний стан.

6.2.4 Розробка накопичувача на базі двійкового суматора. Суматори входять в номенклатуру декількох серій мікросхем ТТЛ. У складі серії К 155 випускаються три типи повних суматорів: однорозрядний К 155 ІМ1, двухрозрядний К 155 ІМ2 і чотирьохрозрядний К 155 ІМ3. Всі вони відносяться до розряду комбінаційних пристроїв, і сигнали суми і перенесення присутні на виході, поки діють вхідні сигнали.

Розглянуті суматори побудовані на основі повних суматорів, які функціонують згідно з таблицею істинності:

Таблиця 6.1 Таблиця істинності повного суматора

№	Входи			Виходи	
п/п	A_i	B_i	P_i	P_{i+1}	S_i
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1

3	0	1	1	1	0
4	1	0	0	0	1
5	1	0	1	1	0
6	1	1	0	1	0
7	1	1	1	1	1

За таблицею істинності запишемо рівняння:

$$S_i = A_i \overline{P}_i \vee B_i \overline{P}_i \vee P_{i-1} \overline{P}_i \vee A_i B_i P_{i-1}$$

$$P_i = B_i P_{i-1} \vee A_i P_{i-1} \vee A_i B_i .$$

У мікросхемах - суматорах в якості базового вузла використовується суматор, реалізований на основі отриманих формул. На рисунку 6.5 приведено функціональне позначення суматора і цоколівка виводів.

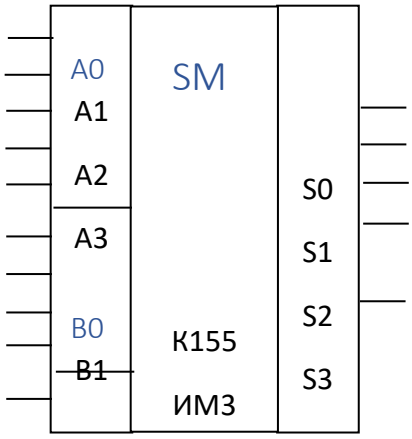

Рисунок 6.5 – Мікросхема К 155 ІМ3

Принцип роботи суматора К155 ІМ3 заснований на паралельному підсумовуванні даних в різних розрядах при послідовному перенесення з розряду в розряд. Вхід перенесення P_0 є тільки у молодшого розряду, а вихід - тільки у старшого. Особливо зручна в застосуванні в апаратурі мікросхема К155 ІМ3: наявність чотирьох розрядів і можливість нарощування дозволяють використовувати її для виконання різних арифметичних операцій.

Результат на виходах суми і перенесення описується наступними виразами:

$$\sum_{A,B} = P_0 + A_1 + B_1 + 2(A_2 + B_2) + 4(A_3 + B_3) + 8(A_4 + B_4) =$$

$$= S_1 + 2S_2 + 4S_4 + 8S_4 + 16P_4.$$

На рисунку 6.6 наведена схема суматора - накопичувача, виконаного на основі повного комбінаційного суматора ІМ3 і регістра пам'яті, виконаного на основі мікросхеми К1533 ТМ8.

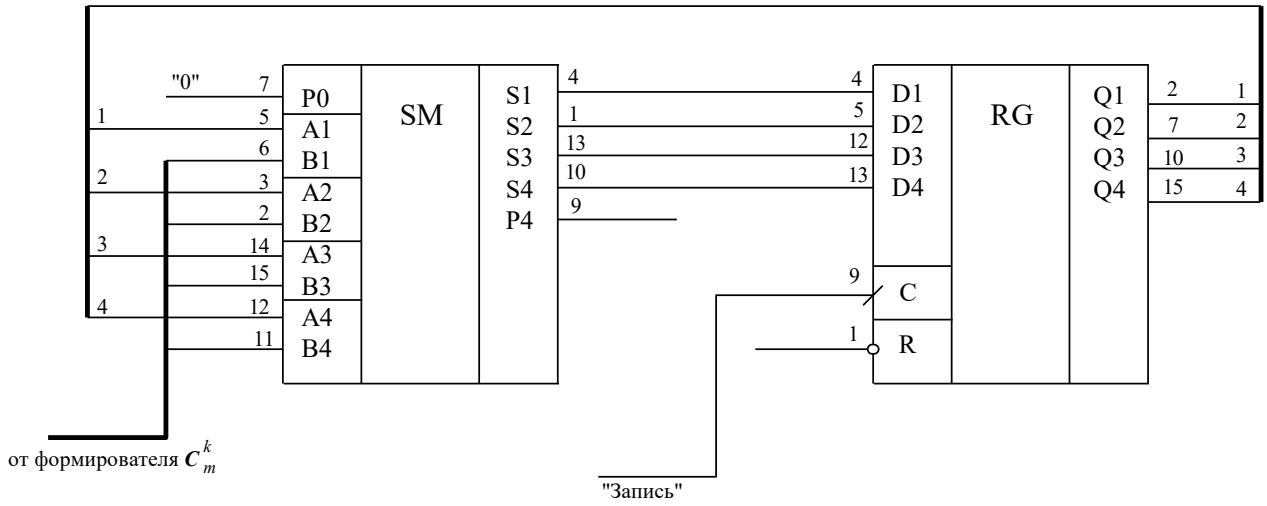

Рисунок 6.6 – Схема накопичувача

У початковому стані регістр знаходиться в нулі. Ланцюгом зворотного зв'язку його стан подається на одну групу входів комбінаційного суматора. На іншу групу входів подається інформація з виходу регістра зсуву. У суматорі проводиться підсумовування двійкових чисел, результат якого присутній на виході суматора, поки двійкові числа надходять на входи суматора. З виходу суматора інформація подається в паралельному коді на входи даних регістра результату. Прийняття рішення про запис інформації в регістр результату докладно розглянуто при розробці функціональної схеми. З приходом імпульсу запису отриманий двійковий код результату записується в регістр.

6.2.5 Розробка схеми управління. Вибір мультиплексора. При керуванні процесом множення значення розрядів повинні подаватися в послідовному коді, хоча в регістрі значень вони записані в коді паралельному. Перетворення паралельного коду в послідовний можна виконати на основі мультиплексора.

Призначення мультиплексорів (від англ. Multiplex - багаторазовий) - комутувати у бажаному порядку інформацію, що надходить з декількох вхідних

шин на одну вихідну. За допомогою мультиплексора здійснюється часовий поділ інформації, що надходить на різних каналах. Мультиплексор виконує функцію безконтактного багатопозиційного перемикача.

Мультиплексори володіють двома групами входів і одним, рідше двома - взаємодоповнюючими виходами. Одні входи інформаційні, а інші служать для керування. До них відносяться адресні та ті, що вирішують (стробуючі) входи. Якщо мультиплексор має *n* адресних входів, то число інформаційних входів буде 2^n. Набір сигналів на адресних входах визначає конкретний інформаційний вхід, який буде з'єднаний з вихідним виводом.

Дозволяючий (стробуючий) вхід керує одночасно всіма інформаційними входами незалежно від стану адресних входів. Заборонний сигнал на цьому вході блокує дію всього пристрою. Наявність дозволяючого входу розширює функціональні можливості мультиплексора, дозволяючи синхронізувати його роботу з роботою інших вузлів. Дозволяючий вхід використовується також для нарощування розрядності мультиплексорів.

Мультиплексори ТТЛ, виконані у вигляді самостійних мікросхем, розрізняються головним чином числом інформаційних і адресних входів, наявністю або відсутністю дозволяючого входу, а також характером вихідних сигналів (щодо вхідних інформаційних), які можуть бути прямими, інверсними або парними.

Для вирішення нашого завдання - можна вибрати мікросхему універсального мультиплексора, наприклад мультиплексор К155 КП1 (рисунок 6.7).

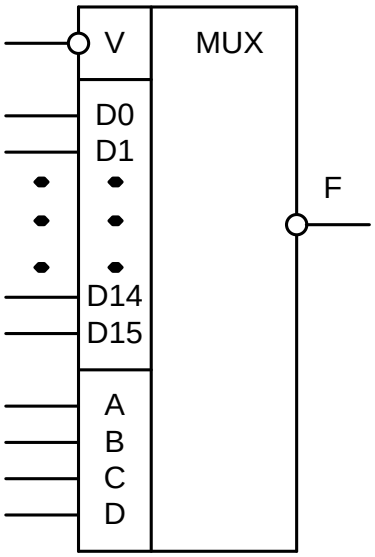

Рисунок 6.7 – Умовне позначення мікросхеми К155 КП1

Він має 16 інформаційних входів (D_0 - D_{15}) і чотири керуючі входи A, B, C, D, вхід дозволу V і один інверсний вихід F. Залежно від цифрової комбінації на керуючих входах сигнали з відповідного інформаційного входу проходять в інвертованому вигляді на вихід мікросхеми. Передача інформації можлива, якщо на вхід дозволу діє напруга низького рівня. При високому рівні на вході дозволу схема блокується і на виході мікросхеми виникає напруга високого рівня.

Логічна функція, реалізована мікросхемою К155 КП1, має вигляд:

$$\bar{F} = \bar{V}(\bar{D}\bar{C}\bar{B}\bar{A}x_0 \vee \bar{D}\bar{C}\bar{B}Ax_1 \vee \cdots \vee DC\bar{B}Ax_{14} \vee DCBAx_{15})$$

Робота мультиплексора описується таблицею 6.2.

Таблиця 6.2 - Таблиця істинності мікросхеми К155 КП1

V	D	C	B	A	D0	D1	D2	D13	D14	D15	$\bar{F}$
0	0	0	0	0	1/0	*	*	*	*	*	0/1
0	0	0	0	1	*	1/0	*	*	*	*	0/1
0	0	0	1	0	*	*	1/0	*	*	*	0/1
…	…	…	…	…	…	…	…	…	…	…	…
0	1	1	0	1	*	*	*	1/0	*	*	0/1
0	1	1	1	0	*	*	*	*	1/0	*	0/1
0	1	1	1	1	*	*	*	*	*	1/0	0/1
1	*	*	*	*	*	*	*	*	*	*	1

Двійкова кодова комбінація, що відповідає значенню основи подається на інформаційні входи, до адресних входів необхідно підключити виходи двійкового лічильника, який буде перебирати в порядку зростання кодові комбінації, відповідні адресами (номерами інформаційних входів). Цю задачу вирішуємо за допомогою керуючого лічильника розрядів.

Керуючий лічильник, перебираючи свої стани, буде послідовно підключати до виходу мультиплексора його інформаційні входи. Як тільки лічильник, виконавши всі кроки, буде скинутий сигналом зворотного зв'язку з виходу компаратора, в початковий - нульовий стан, то схема буде готова для аналізу

наступної кодової комбінації. До початку перетворення лічильник обов'язково повинен знаходитися в нульовому стані, щоб забезпечувати послідовний перебір кодів адрес, починаючи з нульового.

6.2.6 Вибір схеми скидання. Схема керування необхідна для керування роботою вимірювача і узгодження окремих вузлів пристрою між собою.

Найголовніша функція пристрою керування - це встановлення всіх елементів пам'яті при вмиканні в початковий стан. Це здійснюється за допомогою кнопки К1 (див. рис. 6.8).

Тепер пристрій готовий до виконання операції піднесення до степені.

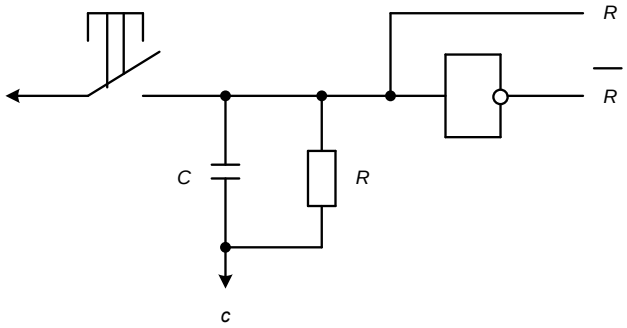

Рисунок 6.8 – Схема формування сигналів установки в нульовий стан

7 РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

7.1 Алгоритм Ель-Гамаля на мові Python

Listing program, що наведена в додатку А реалізує алгоритм Ель-Гамаля для шифрування та розшифрування повідомлень з використанням мови програмування Python. Виконуються такі кроки:

- Визначаються функція modexp для обчислення піднесення до степеня за модулем.
- Визначаються функції generate_keypair для генерації публічного та приватного ключів, encrypt для шифрування повідомлення та decrypt для розшифрування шифротексту.
- Виконуються інструкції для отримання вхідних даних від користувача, генерації ключів, шифрування та розшифрування повідомлення.
- Результати шифрування та розшифрування виводяться на екран.

Цей код дозволяє використовувати алгоритм Ель-Гамаля для шифрування та розшифрування повідомлень. Користувач може ввести параметри алгоритму, такі як просте число p, примітивний корінь g і приватний ключ x. Публічний ключ обчислюється на основі цих параметрів, а потім використовується для шифрування повідомлення за допомогою функції encrypt. Розшифрування повідомлення здійснюється за допомогою приватного ключа за допомогою функції decrypt.

Важливо зауважити, що у цьому коді повідомлення шифрується і розшифровується у вигляді цілих чисел. Якщо потрібно шифрувати текстові повідомлення, необхідно виконати перетворення між символами та числами ASCII.

7.2 Розробка програмного забезпечення на основі алгоритму Ель-Гамаля для управління пристроєм та шифрування/дешифрування повідомлень

Listing program, що наведена в додатку Б реалізує алгоритм Ель-Гамаля для шифрування та дешифрування повідомлень. Ось опис його основних частин:

Функція modexp виконує обчислення модульного піднесення до степеня. Вона використовується для ефективного обчислення великих чисел у модулярному просторі.

Функція generate_keypair створює публічний та приватний ключі на основі переданих параметрів p (просте число) та g (примітивний корінь). Вона вираховує випадкове значення x, обчислює y як g піднесене до степеня x за модулем p, і повертає ці значення як ключі.

Функція encrypt виконує шифрування повідомлення за допомогою публічного ключа. Вона вираховує випадкове значення k, обчислює a як g піднесене до степеня k за модулем p, і обчислює b як добуток повідомлення на y піднесене до степеня k за модулем p. Результатом є пара (a, b).

Функція decrypt виконує дешифрування зашифрованого повідомлення за допомогою приватного ключа. Вона обчислює розшифроване повідомлення за допомогою формули (a^(p-1-x) * b) mod p, де x - приватний ключ. Результат - розшифроване повідомлення. У головній частині програми користувач вводить просте число p та примітивний корінь g. Потім викликаються функції generate_keypair, encrypt і decrypt, щоб створити ключі, зашифрувати та розшифрувати повідомлення. Результати виводяться на екран.

Апаратна частина телекомунікаційного пристрою повинна включати в себе наступні компоненти:

- Процесор: центральний обчислювальний блок, який відповідає за обробку даних та виконання програм.
- Оперативна пам'ять: пам'ять, яку комп'ютер використовує для тимчасового зберігання даних та інструкцій програм.
- Флеш-пам'ять: пам'ять для зберігання програм та даних, які мають бути збережені навіть після вимкнення пристрою.
- Сетевий інтерфейс: інтерфейс, який дозволяє пристрою підключатися до мережі та отримувати/надсилати дані.
- Інтерфейси для підключення зовнішніх пристроїв: інтерфейси, які дозволяють підключати до пристрою різноманітні зовнішні пристрої, такі як мікрофони, камери, динаміки тощо.

Для проектування апаратної частини телекомунікаційного пристрою можна використати наступну схему: Процесор: Raspberry Pi 4; Оперативна

пам'ять: 2 ГБ; Флеш-пам'ять: 16 ГБ; Сетевий інтерфейс: Wi-Fi та Ethernet; Інтерфейси для підключення зовнішніх пристроїв: HDMI, USB, аудіовхід.

Апаратна частина телекомунікаційного пристрою може варіюватися залежно від конкретних вимог проекту та бюджету. Зокрема, можна розглядати різні варіанти процесора, ОЗУ, флеш-пам'яті, а також різноманітні інтерфейси для підключення зовнішніх пристроїв. На рисунку 5.5 наведено алгоритм роботи Program.

Рисунок 5.5 – Алгоритм роботи Program

ВИСНОВКИ

Звідки почати при оцінці алгоритму Ель-Гамаля? Цей алгоритм є одним з важливих криптографічних алгоритмів, який використовується для шифрування та електронних підписів в телекомунікаційних системах. Він забезпечує безпеку та конфіденційність передачі даних, а також перевірку автентичності і цілісності.

Алгоритм Ель-Гамаля базується на складних математичних операціях, зокрема на операціях дискретного логарифмування. Це дозволяє йому бути стійким до атак зламу, включаючи атаки з використанням суперкомп'ютерів.

Один з важливих аспектів використання алгоритму Ель-Гамаля полягає у його застосуванні для шифрування та електронних підписів. При шифруванні дані за допомогою алгоритму Ель-Гамаля стають незрозумілими для несанкціонованих осіб, що забезпечує конфіденційність. Зворотній процес - розшифрування - може бути виконаний лише за допомогою спеціального приватного ключа.

Крім того, алгоритм Ель-Гамаля може використовуватися для створення електронних підписів, що дозволяє перевірити автентичність і цілісність даних. Це особливо важливо в електронних комунікаціях, де важливо переконатися, що дані не були підроблені і походять від вірного джерела.

Порівняння алгоритму Ель-Гамаля з іншими криптографічними алгоритмами показує, що він має свої переваги і недоліки. Один з його великих плюсів - стійка до обчислювальних атак, зокрема до атак з використанням суперкомп'ютерів і квантових обчислювальних систем. Це робить його цікавим для застосування в довгострокових криптографічних сценаріях.

Однак, алгоритм Ель-Гамаля також має певні обмеження та виклики. Наприклад, в порівнянні з деякими іншими алгоритмами, він може бути менш ефективним з точки зору швидкості обробки даних. Використання довгих ключів може впливати на продуктивність системи. Також важливо враховувати практичні аспекти реалізації алгоритму та безпеку зберігання приватних ключів.

У майбутньому розвитку використання алгоритму Ель-Гамаля в телекомунікаційних системах можна очікувати появу нових підходів та оптимізації. Наприклад, можуть бути розроблені нові варіації алгоритму, що покращують його ефективність або стійкість. Також можливе поєднання алгоритму Ель-Гамаля з іншими криптографічними методами для створення більш потужних систем захисту.

Однак, варто враховувати, що безпека криптографічних алгоритмів завжди є постійним пріоритетом. Розвиток алгоритму Ель-Гамаля повинен супроводжуватися постійним аналізом потенційних загроз та вразливостей, а також застосуванням відповідних заходів безпеки для забезпечення безпеки його використання.

Висновки можна зробити, що алгоритм Ель-Гамаля є важливим криптографічним інструментом для забезпечення безпеки, конфіденційності та автентичності даних у телекомунікаційних системах. Він використовує складні математичні операції, що робить його стійким до обчислювальних атак. Алгоритм Ель-Гамаля має успішні приклади використання в шифруванні та електронних підписах, забезпечуючи безпеку і надійність передачі даних.

В цілому, алгоритм Ель-Гамаля є важливим компонентом сучасної криптографії, який допомагає забезпечити безпеку і захист електронних комунікацій, і його значення виростатиме в майбутньому, коли безпека даних стає все більш критичною у цифровому світі.

СПИСОК ЛІТЕРАТУРИ

1. Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644-654.

2. ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31(4), 469-472.

3. Paar, C., & Pelzl, J. (2010). Understanding Cryptography: A Textbook for Students and Practitioners. Springer.

4. Stinson, D. R. (2006). Cryptography: Theory and Practice. CRC Press.

5. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of Applied Cryptography. CRC Press.

6. Katz, J., & Lindell, Y. (2014). Introduction to Modern Cryptography. CRC Press.

7. Boneh, D., & Shoup, V. (2003). A Graduate Course in Applied Cryptography. Retrieved from URL: http://crypto.stanford.edu/~dabo/cryptobook/

8. Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons.

9. Smart, N. (2003). Cryptography: An Introduction. McGraw-Hill.

10. Galbraith, S. D. (2012). Mathematics of public key cryptography. Cambridge University Press.

11. Hoffstein, J., Pipher, J., & Silverman, J. H. (2008). An introduction to mathematical cryptography. Springer Science & Business Media.

12. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (2010). Handbook of applied cryptography. CRC Press.

13. Boneh, D. (2000). Twenty years of attacks on the RSA cryptosystem. Notices of the AMS, 47(2), 160-166.

14. Neal Koblitz. (1987). Elliptic curve cryptosystems. Mathematics of Computation, 48(177), 203-209.

15. Victor S. Miller. (1986). Use of elliptic curves in cryptography. In Advances in Cryptology – CRYPTO’85 Proceedings (pp. 417-426). Springer.

16. Smart, N. (2003). The discrete logarithm problem on elliptic curves of trace one. Journal of Cryptology, 16(3), 205-216.

17. Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, 26(5), 1484-1509.

18. Rivest, R. L., Shamir, A., & Adleman, L. M. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

19. ElGamal, T. (1985). A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 31(4), 469-472.

20. Boneh, D., & Shoup, V. (2000). A graduate course in applied cryptography. Retrieved from URL: https://crypto.stanford.edu/~dabo/cryptobook/

21. Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. CRC Press.

22. Paar, C., & Pelzl, J. (2009). Understanding cryptography: A textbook for students and practitioners. Springer Science & Business Media.

23. Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography engineering: Design principles and practical applications. Wiley.

24. Katz, J., & Lindell, Y. (2014). Introduction to modern cryptography. CRC Press.

25. Протасова Т.О., Приходіна П.А. Схемотехнічна реалізація алгоритма Ель Гамаля // Наукове видання «ФІЗИКА, ЕЛЕКТРОНІКА, ЕЛЕКТРОТЕХНІКА ФЕЕ :: 2023 Матеріали та програма міжнародної конференції молодих вчених», СумДУ, Суми, 95-96 с. URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/91551/1/Conf_FEE_2023.pdf (дата звернення: 25.05.2023)

Додаток А.

Listing program Алгоритм Ель-Гамаля на мові програмування Python

```
import random

def modexp(base, exp, modulus):
    if modulus == 1:
        return 0
    result = 1
    base = base % modulus
    while exp > 0:
        if exp % 2 == 1:
            result = (result * base) % modulus
        exp = exp >> 1
        base = (base * base) % modulus
    return result

def generate_keypair(p, g, x):
    y = modexp(g, x, p)
    return (p, g, y), x

def encrypt(pk, plaintext):
    p, g, y = pk
    k = random.randint(1, p - 1)
    a = modexp(g, k, p)
    b = (modexp(y, k, p) * plaintext) % p
    return a, b

def decrypt(sk, ciphertext):
    p, _, _ = sk
    a, b = ciphertext
    x = sk[1]
    plaintext = (modexp(a, p - 1 - x, p) * b) % p
    return plaintext
```

```
if __name__ == '__main__':
    print("ElGamal Encrypter/Decrypter")
    p = int(input("Enter a prime number (p): "))
    g = int(input("Enter a primitive root (g) modulo p: "))
    x = int(input("Enter a private key (x) less than p: "))
    public_key, private_key = generate_keypair(p, g, x)
    print("Your public key is ", public_key)
    print("Your private key is ", private_key)
    message = int(input("Enter a message to encrypt (an integer): "))
    encrypted_msg = encrypt(public_key, message)
    print("Your encrypted message is: ")
    print(encrypted_msg)
    print("Decrypting message with private key ", private_key, " . . .")
    decrypted_msg = decrypt(private_key, encrypted_msg)
    print("Your decrypted message is:")
    print(decrypted_msg)
```

Додаток Б.

Listing program на основі алгоритму Ель-Гамаля для управління пристроєм та шифрування/дешифрування повідомлень.

```
import random

def modexp(base, exp, modulus):
    if modulus == 1:
        return 0
    result = 1
    base = base % modulus
    while exp > 0:
        if exp % 2 == 1:
            result = (result * base) % modulus
        exp = exp >> 1
        base = (base * base) % modulus
    return result

def generate_keypair(p, g):
    x = random.randint(1, p-1)
    y = modexp(g, x, p)
    return (p, g, y), (p, g, x)

def encrypt(public_key, plaintext):
    p, g, y = public_key
    k = random.randint(1, p-1)
    a = modexp(g, k, p)
    b = (modexp(y, k, p) * plaintext) % p
    return a, b

def decrypt(private_key, ciphertext):
    p, g, x = private_key
    a, b = ciphertext
    plaintext = (modexp(a, p-1-x, p) * b) % p
```

```
    return plaintext

if __name__ == '__main__':
    print("Ель-Гамаль Encrypter/Decrypter")
    p = int(input("Введіть просте число p: "))
    g = int(input("Введіть примітивний корінь g: "))

    public_key, private_key = generate_keypair(p, g)
    print("Ваш публічний ключ: ", public_key)
    print("Ваш приватний ключ: ", private_key)

    message = int(input("Введіть повідомлення для шифрування (ціле число): "))
    ciphertext = encrypt(public_key, message)
    print("Зашифроване повідомлення: ", ciphertext)

    decrypted_message = decrypt(private_key, ciphertext)
    print("Розшифроване повідомлення: ", decrypted_message)
```