Лабораторна робота № 3

**Тема: Реалізація основних асиметричних криптосистем**

**Мета роботи: Дослідження можливостей побудови загальних та спеціальних криптографічних протоколів за допомогою асиметричних криптосистем.**

**Завдання на лабораторну роботу:**

Для першого типу лабораторних робіт – дослідити можливість реалізації одного з чотирьох криптографічних протоколів: розділення секрету, сліпого цифрового підпису, несуперечливого цифрового підпису та розподілу ключів для симетричної криптосистеми за допомогою різних асиметричних алгоритмів (не менше як двох) та порівняти їх ефективність за обраним критерієм.

Підгрупа 1А. Розподіл секрету.

Підгрупа 1B. Розподіл ключів.

Підгрупа 1C. Несуперечливий цифровий підпис

Загальний огляд потенційних алгоритмів реалізації

Криптографічний обмін секретами дійсно може бути реалізований за допомогою різних асиметричних алгоритмів шифрування, щоб забезпечити безпечний обмін і відновлення секрету. Ось кілька методів, що використовують різні асиметричні алгоритми:

**Обмін секретами на основі RSA:**

RSA можна використовувати для поділу секретів, використовуючи властивості CRT(Китайська Теорема про лишки). За допомогою CRT секрет розбивається на частки, і для відновлення секрету потрібна мінімальна кількість цих часток.

Нехай кількість користувачів в групі дорівнює �n. У схемі Міньотта вибирається деяка множина попарно взаємно простих чисел {�1,�2,...,��}{m1, m2, … mn} таких, що добуток �−1k-1 менше, ніж добуток �k найменших з цих чисел. Нехай ці добутки рівні �MN�, відповідно. Число �k називається порогом для схеми, що конструюється по множині {�1,�2,...,��}{m1, m2, … mn}. В якості секрету обирається число �S таке, для якого виконується співвідношення �<�<�M<S<N. Частини секрету розподіляються між учасниками групи наступним чином: кожному учаснику видається пара чисел (��,��)(ri,mi), де ri = S(mod mi)��≡�(mod��)/.

Щоб відновити секрет, необхідно об'єднати �≥�t>=k фрагментів. В цьому випадку отримаємо систему порівнянь виду x = ri(mod mi)�≡��(mod��), множину рішень якої можна знайти, використовуючи китайську теорему про лишки. Секретне число ��<�1⋅�2⋅...⋅��SS<m1\*m2\*…\*mt. Також не складно показати, що якщо число фрагментів менше �k, то, для того щоб знайти секрет �S, необхідно перебрати близько ��N/M цілих чисел. При правильному виборі чисел ��mi такий перебір практично неможливо реалізувати. Наприклад, якщо розрядність ��mi буде від 129 до 130 біт, а �<15k<15, то відношення ��N/M буде мати порядок 21002^100.

**Схема Шаміра**

Ідея схеми полягає в тому, що двох точок достатньо для задання прямої, трьох крапок — для завдання параболи, чотирьох точок — для кубічної параболи, і так далі. Щоб задати многочлен ступеня �k потрібно �+1k+1 точок.

Для того, щоб після поділу секрет могли відновити тільки �k учасників, його «ховають» у формулу многочлена ступеня (�−1)k-1 над кінцевим полем �G. Для однозначного відновлення цього многочлена необхідно знати його значення �k точках, причому, використовуючи меншу кількість точок, однозначно відновити початковий многочлен не вийде. Кількість різних точок многочлена не обмежена (на практиці воно обмежується розміром числового поля �G, в якому ведуться розрахунки).

Щоб відновити секрет можна скористатися інтерполяційної формули, наприклад формулою Лагранжа.

Важливою перевагою схеми Шаміра є те, що вона легко масштабована Щоб збільшити число користувачів в групі, необхідно лише додати відповідне число несекретних елементів до вже існуючих, при цьому має виконуватися умова ��≠��ri != rj при �≠�i!=j. У той же час, компрометація однієї частини секрету переводить схему з ((�,�)n,t) -порогової в (�−1,�−1)(n-1,t-1)-порогову.

**Схема Берклі**

Дві непаралельні прямі на площині перетинаються в одній точці. Будь-які дві некомпланарні площини перетинаються по одній прямій, а три некомпланарні площини в просторі теж перетинаються в одній точці. Взагалі *n* n-мірних гіперплощин завжди перетинаються в одній точці. Одна з координат цієї точки буде секретом. Якщо закодувати секрет як декілька координат точки, то вже по одній частці секрету (однієї гіперплощини) можна буде отримати якусь інформацію про секреті, тобто про взаємозалежності координат точки перетину.

Схема Блеклі менш ефективна, ніж схема Шаміра: у схемі Шаміра кожна частка такого ж розміру, як і секрет, а в схемі Блеклі кожна частка в *t* разів більше. Існують поліпшення схеми Блеклі, що дозволяють підвищити її ефективність.

**Криптографія еліптичних кривих (ECC):**

Порогова ECDSA: На основі ECC можуть бути реалізовані варіанти порогової криптографії, де секрет розбивається на частки за допомогою операцій ECC. Об'єднання порогової кількості часток відновлює секрет, не розкриваючи його окремим учасникам.

**Пост-квантова криптографія:**

Пост-квантові схеми обміну секретами: З появою квантових обчислень, пост-квантові криптографічні алгоритми, такі як NTRUEncrypt або решітчасті конструкції, досліджуються для обміну секретами, щоб забезпечити стійкість до квантових атак.

**Криптографія на основі ідентифікації:**

Порогове шифрування на основі ідентифікаційних даних (IBE): Використовуючи принципи шифрування на основі ідентифікаційних даних, можна реалізувати спільний доступ до секретів, коли центральний орган генерує секретні частки на основі ідентифікаційних даних учасників. Для відновлення секретності необхідна порогова кількість часток.

**Багатовимірні квадратні рівняння:**

Криптографія багатовимірних квадратних рівнянь (БКР): Цей підхід використовує системи багатовимірних поліноміальних рівнянь для розподілу секрету. Секрет ділиться на частки на основі рішень цих рівнянь.

Реалізація обміну секретами за допомогою цих асиметричних алгоритмів, як правило, передбачає розробку спеціальних протоколів, які відповідають властивостям і складності кожного алгоритму. При виборі асиметричного алгоритму для обміну криптографічними секретами необхідно ретельно враховувати припущення щодо безпеки, ефективність і конкретні вимоги програми. Крім того, важливо забезпечити, щоб обраний алгоритм був стійким до відомих атак і підтримував бажаний рівень безпеки.

Реалізація протоколу обміну секретами на основі RSA:

Розподіл секретів на основі RSA передбачає використання алгоритму RSA, зокрема його властивостей, пов'язаних з китайською теоремою про залишок (CRT), для розділення секрету на частки. Ось огляд того, як це працює і процес його реалізації:

**Процес поділу секрету на основі RSA:**

Генерація пари ключів:

Спочатку генерується пара ключів (відкритий і закритий ключі) для шифрування і дешифрування RSA. Відкритий ключ буде використовуватися для шифрування секрету, а закритий ключ буде використовуватися для розшифрування секрету.

Генерування частки:

Секрет розбивається на декілька частин за допомогою властивостей CRT в RSA.

Наприклад, якщо ви хочете розділити секрет S на n частин, ви можете згенерувати n випадкових чисел x1, x2, … xn, такі, що xi < p а потім обчислити Si = S + xi (mod p).

Розподіл часток:

Кожен учасник отримує частку Si не знаючи початкового секрету S.

Відновлення секрету:

Щоб відновити секрет, потрібно зібрати певну кількість фрагментів початкового секрету.

Використовуючи китайську теорему про залишок, враховуючи k дійсних створених фрагментів, початковий секрет S можна відновити за допомогою рівнянь CRT.

**Споживання часу та простору:**

Витрати часу: Часова складність розбиття секрету на частки в схемі розподілу секретів на основі RSA відносно невелика. Вона включає прості модульні операції додавання для генерації часток.

Витрата простору: Кожна частка має складність простору, еквівалентну розміру модуля, що використовується в парі ключів RSA, плюс деякі додаткові метадані для відстеження індексу частки та інформація для реконструкції.

Розмір ключа: Безпека схеми обміну секретними даними на основі RSA залежить від розміру модуля, що використовується. Більші розміри модуля (наприклад, 2048, 4096 біт) забезпечують вищий захист від атак грубої сили.

Хоча обмін секретами на основі RSA пропонує простоту і ефективність з точки зору обчислень, він може бути не таким ефективним, як деякі інші схеми (наприклад, Схема поділу секрету Шаміра) з точки зору споживання простору і гнучкості у визначенні структур доступу. Важливо оцінити конкретні вимоги та міркування безпеки вашої програми перед тим, як вибрати схему спільного використання секретів.

**Переваги обміну секретами на основі RSA:**

Знайомство з алгоритмом і його усталеність:

RSA є добре відомим і широко вивченим алгоритмом асиметричного шифрування, що робить його реалізацію і розуміння більш доступним для криптографів і розробників.

Безпека через математичні властивості:

Безпека RSA полягає в складності факторизації великих псевдопростих чисел. Поки ключі RSA мають відповідний розмір, схема обміну секретними даними користується перевагами безпеки RSA проти відомих атак.

Реконструкція на основі порогових значень:

Дозволяє реконструкцію на основі порогових значень, що означає, що початковий секрет може бути відновлений тільки тоді, коли буде об'єднано певний поріг часток. Цей поріг можна регулювати залежно від потреб безпеки.

Простота поділу секретів:

Поділ секрету на частки за допомогою RSA включає відносно прості операції, такі як модульна арифметика, що робить процес поділу ефективним з точки зору обчислень.

**Недоліки обміну секретами на основі RSA:**

Відсутність гнучкості в структурі доступу:

Розподіл секретів на основі RSA може бути недостатньо гнучким у визначенні складних структур доступу. Він може бути не таким універсальним, як інші методи, наприклад, Схема поділу секрету Шаміра, яка дозволяє більш детально контролювати доступ.

Потенційні проблеми з управлінням ключами:

Управління ключами RSA, особливо в розподіленому середовищі, де необхідно безпечно розподіляти ресурси між декількома сторонами, може створювати проблеми з точки зору безпеки ключів і їх розподілу.

Великі розміри ресурсів:

Ресурси RSA зазвичай мають розмір, еквівалентний розміру по модулю, який використовується в ключах RSA. Це може призвести до більшого розміру ресурсу порівняно з іншими схемами обміну секретними даними, що впливає на вимоги до зберігання та передачі даних.

Проблеми з ефективністю:

Хоча поділ секрету є ефективним, процес відновлення може вимагати більше обчислювальних ресурсів у порівнянні з іншими більш спеціалізованими схемами обміну секретами, особливо при збільшенні кількості розділів.

Потенційна можливість викриття ключів:

Якщо не керувати належним чином, розкриття приватного ключа RSA, який використовується для поділу секрету, може поставити під загрозу безпеку всієї схеми спільного доступу.

При виборі відповідного методу обміну секретними даними, чи то на основі RSA, чи іншої схеми, слід ретельно враховувати вимоги конкретного випадку використання, потреби в безпеці, обчислювальну ефективність, а також простоту реалізації та управління.

Протокол поділу секрету з використанням ECC:

Реалізація протоколу з використанням криптографії еліптичних кривих (ECC) передбачає використання властивостей еліптичних кривих для розбиття секрету на частки і подальшого відновлення секрету, коли достатня кількість часток буде об'єднана.

**Алгоритм поділу секрету з використанням ECC:**

Генерація ключів:

ECC покладається на генерацію пари публічних і приватних ключів. Безпека ECC базується на складності розв'язання задачі дискретного логарифмування на еліптичних кривих.

Закритий ключ генерується випадковим чином і використовується для обчислення відповідного відкритого ключа.

Розділення секрету:

Щоб розділити секрет S на частки за допомогою ECC, можна застосувати порогову схему. Випадкові значення генеруються для створення окремих часток, і ці частки розподіляються між учасниками.

Частка кожного учасника обчислюється за допомогою операцій ECC над секретом і випадковими значеннями.

Розподіл часток:

Кожен учасник отримує свою частку, не маючи можливості самостіно розкрити початковий секрет. Ці частки повинні бути надійно розподілені, щоб зберегти конфіденційність і цілісність секрету.

Відтворення секрету:

Для відновлення секрету збирається заздалегідь визначена кількість часткою (поріг). Алгоритми на основі ECC використовуються для об'єднання цих часток, щоб відновити початковий секрет.

**Споживання часу та простору:**

Часова складність: Операції ECC, такі як додавання та множення точок на еліптичних кривих, як правило, більш ефективні, ніж еквівалентні операції в RSA. Розділення секрету та його відновлення за допомогою ECC включає в себе ці операції і зазвичай вимагає менше обчислювального часу, ніж багато інших асиметричних криптографічних алгоритмів.

Просторова складність: Розмір кожної частки, згенерованої за допомогою ECC, залежить від розміру параметрів кривої та пов'язаних з нею ключів. Частки ECC зазвичай менші за розміром порівняно з частками RSA при однаковому рівні безпеки, що призводить до меншого споживання простору.

**Міркування щодо реалізації:**

Вибір параметрів кривої: Вибір відповідних параметрів еліптичної кривої має вирішальне значення для безпеки та ефективності. Вибір кривої включає такі міркування, як рівень безпеки, продуктивність і сумісність зі стандартами.

Управління ключами: Безпечна генерація, зберігання та розповсюдження ключів ECC та спільних ресурсів мають вирішальне значення для запобігання несанкціонованого доступу до секрету.

Визначення порогових значень: Встановлення порогу для відновлення секрету має важливе значення для балансування між безпекою та практичністю. Вищий поріг підвищує безпеку, але вимагає більшої кількості часткою для відновлення.

Обмін секретними даними на основі ECC має переваги з точки зору ефективності, меншого розміру частки та високого рівня безпеки. Однак важливо зазначити, що реалізація протоколів на основі ECC вимагає глибокого розуміння математики еліптичних кривих і належного управління криптографічними ключами і частками, щоб забезпечити конфіденційність і цілісність спільного секрету.

**Переваги обміну секретами на основі ECC:**

Надійний захист з меншим розміром ключа:

ECC забезпечує такий самий рівень криптографічної стійкості, як і інші асиметричні схеми шифрування (наприклад, RSA), але з меншим розміром ключа. Це призводить до більш ефективного зберігання та передачі ключів і даних.

Ефективність обчислень:

Операції ECC (додавання точок, множення) на еліптичних кривих, як правило, більш ефективні, ніж їхні аналоги в традиційних алгоритмах асиметричного шифрування, таких як RSA. Ця ефективність означає більш швидкі обчислення для розділення секретів та їх відновлення.

Зменшення вимог до сховища:

Обмін на основі ECC, як правило, має менший розмір у порівнянні з іншими криптографічними схемами для того ж рівня безпеки. Це призводить до зменшення вимог до сховища для зберігання і передачі даних.

Стійкість до квантових атак:

ECC є одним з асиметричних методів шифрування, який вважається стійким до атак квантових комп'ютерів, що робить його перспективним вибором для забезпечення довгострокової безпеки.

**Недоліки обміну секретами на основі ECC:**

Складність реалізації:

Розуміння та реалізація протоколів обміну секретними даними на основі ECC вимагає хорошого розуміння математики еліптичних кривих та криптографічних принципів. Ця складність може створити проблеми під час впровадження.

Проблеми управління ключами:

Безпечне управління ключами та ресурсами ECC має вирішальне значення. Неправильне поводження з ключами може призвести до компрометації ключів, що потенційно може поставити під загрозу конфіденційність спільного секрету.

Вибір алгоритму та параметрів:

Вибір відповідних параметрів еліптичної кривої та алгоритмів має вирішальне значення. Вибір слабких параметрів або незахищених алгоритмів може поставити під загрозу безпеку всієї системи.

Потенціал для атак побічних каналів:

Реалізації ECC можуть бути вразливими до атак побічних каналів (таких як синхронізація або аналіз потужності), якщо вони не захищені належним чином. Для зменшення таких вразливостей необхідна належна практика реалізації.

Обмежене застосування в деяких середовищах:

Хоча ECC пропонує значні переваги, його впровадження може бути обмеженим у певних застарілих системах або середовищах через необхідність використання нових криптографічних бібліотек або апаратної підтримки.

Вибір методу обміну секретами на основі ECC передбачає збалансування цих переваг і недоліків на основі конкретних вимог програми, враховуючи такі фактори, як безпека, обчислювальна ефективність, можливості керування ключами та сумісність з існуючими системами. При правильному впровадженні та управлінні, спільний доступ до секретів на основі ECC може забезпечити надійний захист та ефективні криптографічні операції.

Порівняння RSA та ECC

|  |  |
| --- | --- |
| RSA | ECC |
| **Витрати часу** | |
| Часова складність поділу секретів є відносно низькою через використання модульних арифметичних операцій.  Час відновлення може збільшуватися при більшій кількості розділів через обчислення CRT. | Операції ECC (точкове додавання, множення), як правило, більш ефективні, ніж аналоги RSA.  Поділ і відновлення секретів з використанням ECC зазвичай вимагає менше обчислювального часу в порівнянні з RSA. |
| **Витрати пам’яті** | |
| Розмір частки RSA еквівалентний розміру модуля, що призводить до більшого розміру частки.  Більший розмір модуля (для більшої безпеки) також призводить до більшого розміру частки і більшого споживання простору. | Частки ECC мають менший розмір у порівнянні з RSA при однаковому рівні безпеки.  Менші розміри ключів і частки призводять до зменшення вимог до простору, що робить його більш ефективним з точки зору зберігання даних. |
| **Безпека** | |
| Безпека RSA базується на складності факторизації великих напівпростих чисел.  Вразливий до атак, якщо розмір модуля недостатньо великий. | Безпека ECC базується на складності розв'язання задачі дискретного логарифмування на еліптичних кривих.  Забезпечує надійний захист з меншим розміром ключа, стійкий до атак, включаючи потенційну стійкість до квантових атак |
| **Управління ключами** | |
| Керування ключами RSA та спільними ресурсами передбачає роботу з великими розмірами ключів і потенційно великими спільними ресурсами.  Розподіл і зберігання ключів повинні бути безпечними, щоб запобігти компрометації. | Ключі та ресурси ECC мають менший розмір, що робить управління ключами потенційно більш керованим.  Належне поводження з ключами та їх розподіл мають вирішальне значення для підтримання безпеки. |
| **Складність і реалізація** | |
| Відносно простіший у розумінні та впровадженні порівняно з ECC для деяких розробників через його широке розповсюдження та знайомство з ним.  Проблеми можуть виникнути з великими розмірами модулів або складними пороговими схемами. | Вимагає глибокого розуміння математики еліптичних кривих і криптографічних принципів для правильної реалізації.  Більш ефективний і потенційно більш безпечний, але складність реалізації може бути вищою. |

Вибір між RSA- і ECC-шифруванням залежить від різних факторів, включаючи обчислювальну ефективність, вимоги до безпеки, вимоги до місця, можливості управління ключами і конкретний контекст застосування. ECC, як правило, має переваги з точки зору ефективності, меншого розміру ключів і потенційної стійкості до квантових атак, тоді як RSA може бути простішим у реалізації, але може вимагати більшого розміру ключів для забезпечення еквівалентного рівня безпеки.