Міністерство освіти і науки України
Харківський національний університет імені В. Н. Каразіна

ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ

Конспект лекцій
з дисципліни для здобувачів вищої освіти першого (бакалаврського) рівня
за спеціальністю 151 «Автоматизація та комп’ютерно-інтегровані технології»
(174 «Автоматизація, комп’ютерно-інтегровані технології та робототехніка»)
освітньої програми «Автоматизація та комп’ютерно-інтегровані технології»

Електронний ресурс

Харків – 2025

УДК 556.11:504.064
Т 38

Рецензенти:

І. В. Філіпенко – кандидат технічних наук, доцент кафедри автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки;

О. П. Нарєжній – кандидат технічних наук, доцент кафедри кібербезпеки інформаційних систем, мереж і технологій Харківського національного університету імені В. Н. Каразіна.

Затверджено до розміщення в мережі Інтернет рішенням Науково-методичної ради Харківського національного університету імені В. Н. Каразіна (протокол № 9 від 23 квітня 2025 року)

Т 38 **Технологі**ї захисту інформації : конспект лекцій з дисципліни для здобувачів вищої освіти першого (бакалаврського) рівня за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології» (174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка») освітньої програми «Автоматизація та комп'ютерно-інтегровані технології» [Електронний ресурс] / уклад. К. Є. Лисицький, Є. П. Колованова, Д. Ю. Узлов. – Харків : ХНУ імені В. Н. Каразіна, 2025. – (PDF 165 с.)

Конспект лекцій створено з метою надати студентам знання з сучасних технологій захисту інформації, зокрема з криптографічних методів захисту інформації, ознайомити з найбільш поширеними математичними алгоритмами, які використовуються в криптографії, симетричними та несиметричними криптографічними системами, методиками розрахунку параметрів криптоалгоритмів та специфічними методами криптоаналізу.

УДК 556.11:504.064

Зміст

Вступ

Задача захисту інформації в комп'ютерних системах на сьогоднішній день є актуальною внаслідок широкої розповсюдженості таких систем, розширення комп'ютерних мереж, якими передаються великі обсяги інформації.

Забезпечення безпечної діяльності комп'ютерних систем необхідне для будь-яких підприємств і установ, починаючи від державних організацій і закінчуючи невеликими приватними фірмами, незалежно від виду їх діяльності. Розходження полягає лише в засобах, методах та в обсязі забезпечення безпеки. В лекціях відображено основні розділи курсу «Технології захисту інформації»: основні засади інформаційної безпеки, класифікація та принципи побудови систем захисту, методи і засоби захисту інформації в комп'ютерних системах. Для освоення даної дисципліни необхідні знанні із курсів: «Вища математика», «Дискретна математика», «Лінійна алгебра та аналітична геометрія», «Теорія ймовірності і математична статистика. Студенти при вивченні дисципліни повинні сформувати погляд на захист інформації і криптографію як на систематичну науково-практичну діяльність, що носить прикладний характер. Сформувати базисні теоретичні поняття, що лежать в основі процесу захисту інформації. У результаті вивчення дисципліни студент повинен знати: особливості інформації як об'єкту захисту, перелік та особливості основних загроз інформації в комп'ютерних системах, підходи до розробки політики безпеки комп'ютерної системи, основні принципи захисту інформації, порядок формування комплексу засобів захисту інформації, криптографічні методи і засоби захисту інформації, порядок визначення вимог щодо захисту інформації в комп'ютерній системі, основні положення нормативної бази системи захисту інформації в комп'ютерних системах. Вміти: застосовувати математичні методи описання і дослідження криптосистем; оцінювати криптографічну стійкість шифрів.

Розділ 1 Інформаційна безпека та криптологія. Класичні симетричні криптосистеми

Лекція 1

Вступ до дисципліни. Основні поняття й означення інформаційної безпеки. Основні загрози безпеці АСОІ

Цивілізація переживає чергову революцію–інформаційну, яка в значної мірі торкнулась й України. Неможна уявити собі систему керування складними технологічними процесами, не кажучи вже про військові системи, енергетичні, платіжні та багато інших без використання засобів обчислювальної техніки.

Мережі телекомунікацій все більш інтегруються з комп′ютерними засобами обробки та архівного зберігання інформації.

Будемо розуміти під інформацією сукупність даних та програм, які використовуються у автоматизованій системі.

Особливо важливу роль інформаційно-телекомунікаційні та інформаційно - аналітичні системи грають в таких сферах як державне управління, економіка, освіта, наука, оборона, безпека життєдіяльності і т. і.

При функціонуванні цих систем відбувається обробка інформації.

Під обробкою інформації в системі мається на увазі виконання однієї або декількох функцій, а саме: збір, введення, запис, перетворення, зчитування, зберігання, знищення, реєстрація, прийом, передача, які здійснюються в системі за допомогою технічних і програмних засобів.

При цьому обмін інформацією здійснюється з використанням інформаційно - телекомунікаційних систем як внутрішнього так і загального користування, у тому числі при підключенні через глобальну світову інформаційну інфраструктуру.

Широке застосування знаходить використання електронних документів і здійснення електронного документообігу.

При цьому під електронним документом розуміється інформація, яка зафіксована у вигляді електронних даних, включаючи обов'язкові реквізити документа.

Однією з головних вимог до електронних документів є забезпечення їх цілісності на всіх етапах їх життєвого циклу, а також підтвердження авторства. Для виконання цих вимог в Україні як і в інших технологічно розвинутих державах прийняті відповідні закони. " Про електронні документи і електронний документообіг" та "Про електронний цифровий підпис". Ці закони набули чинності з 1 січня 2004 року. "Про захист інформації в інформаційно-комунікаційних системах", прийнятий 16 грудня 2020 року.

Питання безпеки інформації - важлива частина процесу впровадження нових інформаційних технологій у всі сфери життя суспільства. Широкомасштабне використовування обчислювальної техніки і телекомунікаційних систем в рамках територіально-розподілених інформаційних систем, перехід на цій основі до безпаперової технології, збільшення кількості інформації, яка обробляється, і розширення круга користувачів приводять до якісно нових можливостей несанкціонованого доступу до ресурсів і даних інформаційної системи, до їх високої уразливості.

Реалізація загроз несанкціонованого використання інформації наносить зараз набагато більший збиток, ніж, наприклад, "випадкові" пожежі в приміщеннях або фізична дія на співробітників. Проте витрати на побудову системи захисту інформації ще поки непомірно малі в порівнянні з витратами на захист від грабіжників або на протипожежний захист.

До того ж в сучасному бізнесі спостерігається поступовий перехід від чисто фізичних методів дії на конкурентів до більш інтелектуальних, у тому числі з використанням новітніх засобів і способів добування інформації.

Проводячи аналіз положення інформаційної безпеки різних відомств, організацій і фірм, можна прийти до висновку, що об'єктом захисту, який викликає найбільшу тривогу і акумулює всі проблеми інформаційної безпеки є інформаційно-телекомунікаційні системи, які будуються на базі комп'ютерів.

За даними федерального бюро розвідки США, наприклад, в 2001 році фінансові втрати від комп'ютерного злодійства склали 400 млн. дол.

Втрати від НСД до інформації, яка пов'язана з діяльністю фінансових інститутів США в тому ж році склала 1 млрд. дол.

Крім того, проведений аналіз підтверджує, що з кожним роком число комп'ютерних атак, які скоюють зловмисники, зростає. У сучасному світі витрати на кібербезпеку сягають астрономічних цифр. Однак цифри у звітах про кіберзлочинність, зазвичай, сильно занижені, тому статистика досить далека від реальної картини. Згідно з дослідженнями, щорічно, більше 70 мільйонів людей у світі стають жертвами кіберзлочинів.

Американські дослідники встановили, що завдані хакерами у 2020 році збитки у світі становлять понад один відсоток світового ВВП. Ще один приклад. З січня по червень 2023 року криптоіндустрія зіткнулася як мінімум з 395 зламами, у результаті втратила $479,4 млн.

Вище надані дані підтверджують проблемність і складність забезпечення інформаційної безпеки в різних інформаційнио - телекомунікаційних системах, інформаційних технологіях і системах інформаційних технологій.

Отже, перелічимо основні причини, які обумовили актуальність та важливість проблеми забезпечення безпеки автоматизованих систем обробки інформації (АСОІ):

– різке збільшення потужності сучасних комп′ютерів із спрощенням їх експлуатації;

– збільшення обсягів інформації, яка накопичується, зберігається та обробляється за допомогою засобів автоматизації;

– зосередження у єдиних базах даних інформації різного призначення та належності;

– розширення кола користувачів, які мають доступ до обчислювальних ресурсів;

– бурхливий розвиток програмних засобів, які не задовольняють навіть мінімальним вимогам безпеки;

– поширення мережевих технологій, об′єднання локальних мереж у глобальні. Розвиток Internet.

Таким чином проблеми безпеки кіберпросторів набувають все більшої значимості та актуальності. Одним із свідчень розуміння важливості цих проблем у житті держави – поява в учбових планах та програмах вузів України нових спеціальностей щодо підготовки фахівців відповідного профілю. Наша дисципліна може бути названа ″Технологіїї захисту інформації ″.

З цієї дисципліни ви дізнаєтесь про загальні відомості щодо методів захисту інформації. Зокрема із криптографічним методом захисту інформації.

Познайомимося з важливими поняттями курсу.

Захист інформації – сукупність організаційно-технічних заходів, правових та морально–етичних норм, адміністративних заходів, фізичних та програмно–технічних засобів, спрямованих на протидію загрозам АСОІ чи зведення до мінімуму можливості шкоди.

Безпека АСОІ – захищеність від випадкового чи навмисного втручання у нормальний процес функціонування, а також від спроб крадіжки, змінювання чи порушення її компонентів.

Природа впливів на АСОІ може бути самою різною: стихійні лиха, помилки персоналу, дії шахраїв.

Під доступом до інформації розуміється знайомство з інформацією, її обробка, копіювання, модифікація, знищення.

Розрізняють санкціонований та несанкціонований доступ.

Санкціонований доступ до інформації – доступ, не руйнуючий установлені правила розмежування доступу.

Правила розмежування доступу – служать для регламентації прав доступу суб′єктів доступу до об′єктів доступу.

Несанкціонований доступ – доступ до інформації, який характеризується порушенням установлених правил розмежування доступу. Найбільш поширений тип комп′ютерних порушень.

Конфіденційність даних - статус, який надається даним, та визначає вимоги до степені їх захисту.

Суб′єкт - активний компонент системи, який може стати причиною потоку інформації від об′єкта к суб′єкту чи зміни становища системи.

Об′єкт - пасивний компонент системи, зберігаючий, приймаючий чи передаючий інформацію. Доступ до об′єкту- це доступ до інформації, яка в ньому зберігається.

Цілісність інформації забезпечується у тому разі, коли дані в системі, у семантичному сенсі, не відрізняються від даних в вихідних документах.

Цілісність компонента чи ресурсу - властивість компоненту чи ресурсу бути незмінним у семантичному сенсі при функціонуванні системи в умовах випадкових чи навмисних викривлень чи порушуючих втручань.

Під загрозой безпеці АСОІ розуміють можливі впливи на АСОІ, які прямо чи побічно можуть нанести шкоду її безпеці.

По цілі впливу розрізнюють 3 типи загроз АСОІ:

- загрози порушення конфіденційності;
- загрози порушення цілісності;
- загрози порушення працездатності.

1. Спрямовані на розголошення конфіденційної чи таємної інформації. Інформація стає відомою лицям , які не мають до неї доступу. Має місце кожен раз коли стався НСД.

2. Спрямовані на змінення та викривлення інформації, яке приводить до порушення її кількості чи повному знищенню. Може бути порушена шахраєм та через об′єктивні впливи середовища. Актуальна для мереж передачі інформації, комп′ютерних мереж та телекомунікацій.

3. Визначені дії чи знижують працездатність системи, чи блокують доступ до її ресурсів.

Небезпечні впливи на АСОІ можна поділити на випадкові та навмисні.

Випадкові:

–аварійні ситуації через стихійні лиха;

–відмови та збої апаратури;

–помилки у програмному забезпеченні;

–помилки у роботі обслуговуючого персоналу та користувачів;

–перешкоди у лініях зв′язку через впливи навколишнього середовища.

Навмисні загрози зв′язані із цілеспрямованими діями порушника.

Щодо банківських АСОЇ, то можна виділити наступні навмисні загрози:

– НСД та ознайомлення з конфіденційною інформацією;

– ознайомлення банківських службовців із інформацією, до якої вони не повинні мати доступ;

– несанкціоноване копіювання програм та даних;

– крадіжка магнітних носіїв, які містять конфіденційну інформацію;

– крадіжка надрукованих банківських документів;

– навмисне знищення інформації.

– несанкціонована модифікація фінансових документів;

– фальсифікація документів, які передаються по каналу зв′язку;

– відмова від авторства повідомлення, які передаються по КЗ;

– відмова від факту одержання інформації;

– нав′язування до цього переданого повідомлення;

– руйнування інформації вірусами;

– руйнування архівної інформації;

– крадіжка обладнання.

Уразливість АСОІ – деяка невдала властивість, яка робить можливим виникнення та реалізацію загрози.

Атака на комп′ютерну систему - дія, яка здійснюється шахраєм, та складається у пошуку та використанні тої чи іншої вразливості системи.

Атака – реалізація загрози.

Безпечна або захищена система – система із засобами захисту, які успішно та ефективно протидіють загрозам безпеці.

Комплекс засобів захисту – сукупність програмних та технічних засобів, які створюються та підтримуються для забезпечення інформаційної безпеки АСОІ.

Склад комплексної системи захисту визначається на основі вивчення всіх інформаційних потоків системи телекомунікацій і, як наслідок розробці такої моделі загроз, що забезпечить мінімальні втрати. На основі моделі загроз має бути розроблена та запроваджена концепція та політика інформаційної безпеки

Політика безпеки - сукупність норм, правил, практичних рекомендацій, які регламентують роботу засобів захисту АСОІ від заданої множини загроз безпеці.

Комплексна система захисту інформації має забезпечувати такі послуги безпеки:

– конфіденційність інформації – властивість інформації, коли неавторизовані особи, які не мають доступу до інформації, не можуть розкрити зміст цієї інформації;

– цілісність інформації – властивість інформації, яка полягає в тому, що вона не може бути змінена навмисно чи випадково користувачем чи процесом, а також жоден з компонентів не може бути усуненим, модифікованим або доданий з порушенням політики безпеки;

– спостережливість – властивість ресурсу інформаційної технології, що дозволяє реєструвати всі дії користувачів. Здійснювати доступ поіменно, відповідно до ідентифікаторів та повноважень, а також реагувати на ці дії з метою мінімізації можливих втрат в системі, що також здійснюється за рахунок застосування криптографічного захисту інформації;

– доступність – властивість ресурсу системи (інформації), яка полягає в тому, що авторизований користувач може отримати доступ до ресурсу тільки із заданою якістю.

Існують принаймні два підходи до проблеми забезпечення безпеки АСОІ:

«Фрагментарний» – спрямований на протидію чітко визначеним загрозам. Перевага - високе обрання до конкретної загрози.

Комплексний підхід – орієнтований на створення захищеного середовища обробки інформації в АСОІ, який єднає у єдиний комплекс різнопланові міри протидії загрозам. Дозволяє гарантувати визначений рівень безпеки АСОІ.

Під системой захисту АСОІ розуміють єдину сукупність правових та морально–етичних норм, адміністративно–організаційних мір, фізичних та програмно–технічних засобів, спрямованих на протидію загрозам АСОІ із ціллю зведення до мінімуму можливостей шкоди.

По способам здійснення усі міри забезпечення безпеки комп′ютерних систем поділяються на:

– правові;

– морально–етичні;

– адміністративні;

– фізичні;

– апаратно-програмні.

До правових мір відносяться усі діючі у державі закони, укази та нормативні акти, які регламентують правила поводження із інформацією обмеженого користування та відповідальності за їх порушення .

Морально-етичні міри. Найрізноманітніші норми поведінку, які традиційно склались чи складаються по мірі розповсюдження комп′ютерів у державі.

Адміністративні міри включають:

– розробку правил обробки інформації в АСОІ;

– сукупність дій при проектуванні та обладнанні ОЦ;

– сукупність дій при доборі та підготовці персоналу;

– організація надійного пропускного режиму;

– організацію обліку, зберігання, використання та знищення документів та носіїв конфіденційної інформації;

– розподіл реквізитів розмежування доступу;

– організацію прихованого контролю за роботою користувачів та персоналу.

Фізичні міри контролю. Різного роду механічні та електро та електронно-механічні установи та спорудження, які призначені задля створення фізичних перешкод.

Апаратно-програмні засоби. Деякі електронні пристрої та спеціальні програми. які реалізують самостійно чи с другими засобами захисту:

– ідентифікацію;

– аутентифікацію;

– розподіл доступу к ресурсам АСОІ;

– контроль цілісності даних;

– забезпечення конфіденційності;

– реєстрацію та аналіз подій, які трапляються в АСОІ;

– резервування ресурсів та компонентів АСОІ.

Більшість з перелічених способів захисту реалізується криптографічними методами захисту інформації, які ми будемо вивчати більш детально на наступних лекціях.

Контрольні запитання та завдання

1. Надайте поняття інформації.
2. Що розуміється під обробкою інформації.
3. Що таке електронний документ?
4. Основні причини, які обумовлюють актуальність та важливість проблеми забезпечення безпеки автоматизованих систем обробки інформації.

5. Надайте визначення захисту інформації в інформаційній системі.
6. Надайте визначення санкціонованого та несанкціонованого доступу.
7. Атака на комп′ютерну систему.
8. Уразливість АСОІ.
9. Цілісність інформації.
10. Цілісність компонента чи ресурсу.
11. Види загроз автоматизованій системі обробки інформації.
12. Суб’єкт та об’єкт системи.
13. Політика безпеки.
14. Підходи до проблеми забезпечення безпеки АСОІ.
15. Комплексна система захисту інформації.
16. Морально-етичні міри.
17. Адміністративні міри
18. Фізичні міри контролю
19. Апаратно-програмні засоби.
20. Правові міри.

Лекція 2

Структурна схема та математична модель системи КЗІ. Принципи криптографічного захисту інформації. Основні групи шифрів. Класичні симетричні криптосистеми

Класична задача криптографії виникає тоді, коли двоє збираються обмінятися конфіденційним повідомленням у присутності третьої недружньої сторони.

Криптологія (від слов kryptos – таємний, logos – наука) займається проблемами захисту інформації шляхом її перетворення.

Слід відзначити, що криптологія поділяється на криптографію та криптоаналіз. Шифрування та розшифрування , яке виконують криптографи, та розробка та розкриття шифрів, яке виконують криптоаналітики, складають предмет науки криптології.

Щодо телекомунікаційного каналу можна розглядати передачу повідомлень між двома законними (санкціонованими) користувачами каналу при наявності незаконного (несанкціонованого) користувача, який хоче перехопити повідомлення.

Санкціоновані користувачі перш ніж обмінятися конфіденційним повідомленнями повинні надати друг другу відомості про ключі. Кожен з санкціонованих користувачів може зашифрувати та розшифрувати повідомлення, передаючи його по каналу зв′язку у зашифрованому вигляді.

Одразу надаємо деякі означення.

Ключ – деяка послідовність символів, яка керує процедурами шифрування- розшифрування.

Шифрування – процес перетворення відкритих даних у шифртекст по закону ключа.

Розшифрування – процес протилежний шифруванню.

Шифртекст – перетворені дані із закритим семантичним змістом.

Зрозуміло, що шифр повинен бути надійним, тобто не дозволяти несанкціонованому користувачу уявити зміст переданого повідомлення, а також повинен бути ефективним, тобто виконувати процедури шифрування-розшифрування достатньо швидко. У той чи іншої мірі цім вимогам задовольняють:

– шифри заміни;
– шифри перестановки;
– шифри гамування;
– шифри з використанням аналітичних перетворень даних, які шифруються.

Приклади таких шифрів будуть нами розглянуті у наступних лекціях. Можна одразу відзначити, що ідеального шифру не існує. З другого боку, вимоги до захисту інформації змінюються у широких межах. У залежності від ситуації використовуються шифри з тими чи іншими властивостями.

Так біржова інформація, перестає бути таємницею вже через 20 хвилин. Тобто повинна бути зашифрована та передана в секунди. Є інформація, яку зберігають десятиріччями, але немає необхідності особливо турбуватися про швидкість її шифрування.

Класична криптографічна схема захищеної передачі інформації у односторонньому каналі зв′язку може бути схематично надана у наступному вигляді.

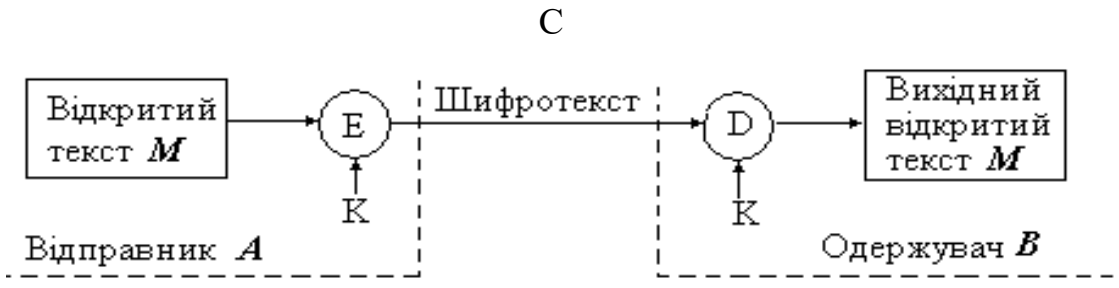

Рисунок 2.1 – Класична криптографічна схема захищеної передачі інформації у односторонньому каналі зв′язку

Де : M – відкритий текст;

C– зашифрований текст;

K– ключ;

E,D – процедури шифрування-розшифрування.

Процедуру перетворення відкритого повідомлення M у шифртекст C математично можна описати:

$$C = E_K (M)$$

Це означає, що шифртекст C одержують із відкритого повідомлення M через вплив на повідомлення M оператора E та ключа шифрування K.

Подібним чином можна описати процедуру розшифрування:

$$M = D_K (C)$$

Відкрите повідомлення M одержують із шифртекста C через дію оператора D та ключа K.

Щоб уявити та відчути предмет та задачі дисципліни розглянемо кілька прикладів найпростіших криптосистем.

Самий стародавній шифр - шифр ″Скітала″. На валик визначеного діаметру намотувалась стрічка виток до витка. Далі писали повідомлення завдовжки валику (по однієї літері на витку). Прочитати повідомлення можна було тільки якщо намотати стрічку виток до витка на валик такого ж діаметру.

У даному разі ключ - діаметр валику.

Римський імператор Юлій Цезар (100-44 р. до н.е.) шифрував свої повідомлення способом, при якому кожна буква тексту замінювалась деякою другою, яка стоїть у абетці на 3 позиції пізніше. Для української мови це означає а→г; б→д ; в→е. Останні літери абетки зміщувалися циклічно. У даному разі ключ K=3. Говорять, що шифр Цезаря- шифр зсуву на 3 позиції (чи заміни).

Шифр ″частоколу ″.

Одразу приклад. Слово криптографія можна записати по-різному:

Р П О Р Ф Я

К И Т Г А І

И О А Я

Р Т Р І

К П Г Ф

У даному випадку бачимо, ключ ″висота″ частоколу. Шифри ″Скітала″ та ″частоколу″ – шифри перестановки. Взагалі говорячи, перетворення шифрування може бути симетричним чи несиметричним відносно до перетворення розшифрування. Існують два класи криптосистем:

– симетричні (одноключові);

– асиметричні (двухключові або системи із відкритим ключем)

Відповідні загальні схеми цих систем можна надати наступним чином (рисунки 2.2 та 2.3).

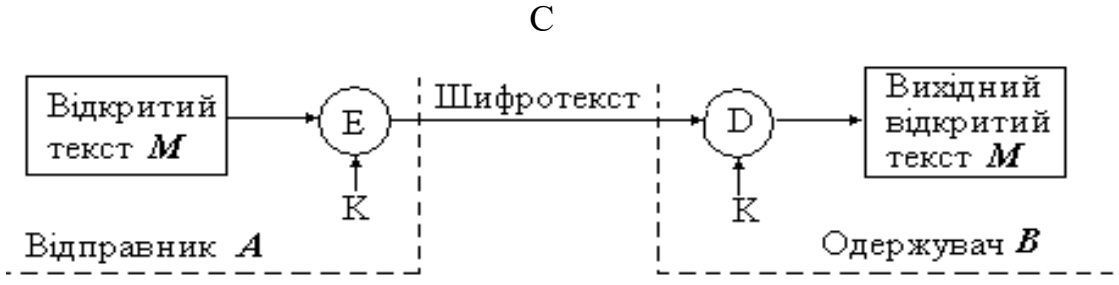

Рисунок 2.2 – Симетрична криптосистема

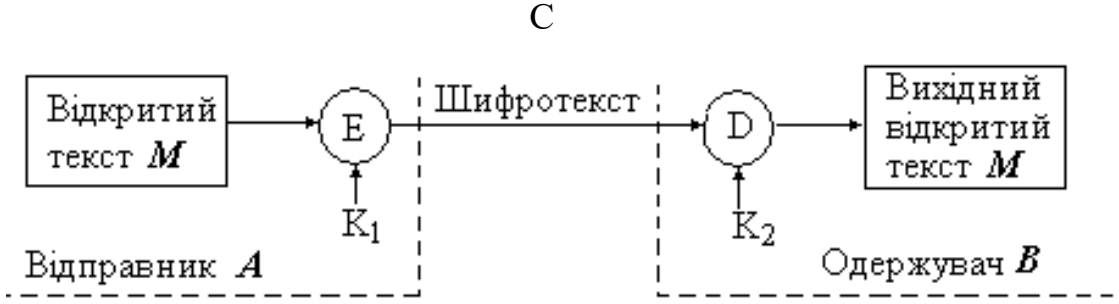

Рисунок 2.3 – Асиметрична криптосистема

Як видно, в симетричній криптосистемі використовують однакові ключі для шифрування і розшифрування. Типовими прикладами, крім вже розглянутих нами, можуть бути симетричні криптосистеми DES, IDEA, RJNDAEL.

В симетричній криптосистемі ключ треба передавати відправнику та одержувачу по захищеному каналу зв′язку.

В асиметричних криптосистемах ключ шифрування не дорівнює ключа розшифрування. По незахищеному каналу зв′язку передається тільки відкритий ключ, а таємний зберігається на місці його генерації.

Ми будемо розглядати асиметричні киптосистеми RSA, Єль-Гамала, Діфі-Хелмана та криптосистеми з використанням перетворень в групах точок еліптичних кривих.

Будь-яка спроба з боку перехоплювача розшифрувати зашифроване повідомлення С для одержання відкритого тексту М, чи зашифрувати власний текст М* для отримання правдоподібного тексту С*, не маючи справжнього ключа, зветься криптоаналітичною атакою.

Коли криптоаналітична атака не досягає цілі, то система виявляється криптостійкою.

Криптоаналіз – наука про розкриття вихідного тексту без знання ключа.

Фундаментальне правило криптоаналізу вперше сформульовано голландцем А. Керкхофом у XIX віці.

Воно укладається у тому, що стійкість шифру повинна визначатися тільки таємністю ключа. Увесь алгоритм шифрування, окрім значення таємного ключа, відомий криптоаналітику супротивника.

Існують чотири типи криптоаналітичних атак. Вони формулюються за припущенням, що криптоаналітику відомий алгоритм та шифртексти повідомлень.

1. Криптоаналітична атака при наявності тільки відомого шифртексту.

Робота криптоаналітика укладається у тому, щоб розкрити вихідні тексти більшості повідомлень, ще краще, обчислити ключ, для того, щоб розшифрувати усі повідомлення, зашифровані цім ключем.

2. Криптоаналітична атака при наявності відомого відкритого тексту.

Робота криптоаналітика укладається у знаходженні ключа, чи алгоритму розшифрування.

3**.** Криптоаналітична атака при можливості вибору відкритого тексту.

Більш потужний криптоаналіз.

4. Криптоаналітична атака з адаптивним вибором відкритого тексту.

Ця атака надає криптоаналітику ще більш можливостей.

Крім цього можна відзначити атаку повного перебору та атаку із використанням обраного шифртексту.

Клод Шеннон, а точніше його книга «Теорія зв'язку в секретних системах», зробила визначальний внесок у сучасну криптографічну науку. Вважають, що зазначена праця визначила онови та сформулювала сутність сучасної криптографії.

Шенон у своїх трактатах відповів на дуже важливі питання:

1. Чи можна створити систему захисту інформації із стійкістю, яка вимагається, абсолютно стійку, якщо криптоаналітик має необмежені ресурси часу (енергія Сонця);

2. Чи можна утворити систему захисту, якщо криптоаналітик має обмежені ресурси щодо часу та працездатності.

Шенон показав, що можна створити такі системи захисту:

1. Теоретично недешифруємі.

2. Обчислювально стійки.

3. Обмеженої стійкості.

4. Доказуємо стійки.(Ель-Гамала, RSA).

Теорема 2.1 Необхідною та достатньою умовою теоретичної недешифруємості є умова:

$$P(C_J / M_I) = P(C_J)$$

Тобто ймовірність виникнення криптограми не залежить від сформованого відправником повідомлення. Будь-яке повідомлення повинно з рівною ймовірністю відображатися у будь-яку криптограму.

Якщо ввести позначення:

Відправник інформації формує на виході повідомлення M_I та полагається відомою апріорна статистика $P(M_I)$ ймовірностей виникнення повідомлень для $I = 1, n_c$, n_c - кількість повідомлень.

Вважається відомим алфавіт джерела ключів m_k, ймовірності виникнення ключів $P(K_I)$, $I = 1, n_k$, n_k - кількість ключів.

На виході шифратора-аутентифікатора формуються криптограми C_J, та уявляється відомою апріорна статистика $P(C_J / M_I)$ для усіх i та j, $j = 1, n_k, n_k$ - кількість криптограм.

Криптограми передаються по КЗ чи записуються на носії інформації. Коли приймають криптограму, роблять зворотне перетворення.

Криптоаналітик із ймовірністю ≈1 перехоплює усі криптограми та намагається спочатку визначити:

$$P(M_i / C_j) \text{ чи } P(K_i / C_j),$$

тобто

1. Що в C_j криптограмі міститься M_i повідомлення.

2. C_j криптограма отримана із використанням K_i ключа.

Доведення:

$$P(M_i/Cj) = \frac{P(M_i) * P(C_j/M_i)}{P(C_j)} = \frac{P(M_i) * P(K_{ij})}{P(C_j)}$$

$$P(C_j) = \sum_{i=1}^{N} P(M_i) * P(C_j / M_i)$$

$$P(M_i / C_j) = P(M_i).$$

Тобто система теоретично недешифруєма. Криптоаналітік у результаті перехоплення нічого не дізнався.

$$\frac{P(M_i / C_j)}{P(M_i)} = \frac{P(C_j / M_i)}{P(C_j)} = 1$$

$$P(C_j/M_i) = P(C_j)$$

Шенон упровадив поняття ентропії – середньої кількості інформації, яка міститься у одному повідомленні, знаку та т. і.

$$H(M_i) = -\sum_{i=1}^{n} P(M_i) * \log_2 P(M_i)$$

$$H(K_i) = -\sum_{i=1}^{n} P(K_i) * \log_2 P(K_i)$$

До перехоплення криптоаналітик знаходиться у апріорній невизначеності H (M). Після перехоплення великої кількості криптограм криптоаналітик знаходиться у невизначеності H (M/C)

Кількість інформації, яку одержав криптоаналітик відносно джерела повідомлень:

$$H(M) - H(M/C) = \Delta I(M,C),$$

чи

$$H(K) - H(K/C) = \Delta I(K,C)$$

Граничні випадки:

1. $\Delta I (M/C) = 0$ $H (M) = H (M/C)$ – ентропія не змінилась, нічого о системі не дізнався.

2. $H (M/C) = 0$ $\Delta I (M/C) = H (M)$ – відомо про систему все (система зламана).

3. $0 < H (M/C) \leq H (M)$ – у реальних ситуаціях.

Чім менше H (M) то й менше успіх.

Відстань єдності – яку кількість символів треба перехопити, щоб мати єдине рішення.

Для теоретично недешифруємої системи довжина ключа повинна бути не менш ніж довжина повідомлення.

Шифри простої заміни перетворюють відкритий текст таким чином, що кожний його символ замінюється на якій - небудь інший.

При цьому однаковим символам відкритого тексту відповідають однакові символи шифртексту, а різним - різні. Ключем є таблиця, яка указує в який саме символ шифртекста переходить символ відкритого тексту.

Без втрати спільності можна вважати, що повідомлення і шифртекст записують в одному і тому ж алфавіті, оскільки використовування екзотичних символів не зробить шифр надійніше.

Зробивши це припущення, можна легко підрахувати кількість всіх можливих ключів для шифру підстановки. Для алфавіту з 33 букв це 33! ключів.

Слід зазначити, що комбінація шифрів заміни - перестановки утворюють все різноманіття вживаних на практиці симетричних шифрів.

На наступній лекції ми починаємо розглядати ідеї найпростіших шифрів заміни та перестановки.

Контрольні запитання та завдання

1. Криптологія як наука.
2. Класична задача криптографії.

3. Основні означення криптографіїї (ключ, шифрування, розшифрування, шифртекст).

4. Класична криптографічна схема захищеної передачі інформації у односторонньому каналі зв′язку.

5. Два класи криптосистем.

6. Шифри "Скітала"та частоуолу.

7. Визначення криптоаналізу.

8. Типи криптоаналітичних атак.

9. Правило А. Керкхофа.

10. Системи захисту по Шеннону.

11. Теоретично недешифруємі криптосистеми, приклад.

12. Обчислювально стійки криптосистеми

13. Криптосистеми лбмеженої стійкості.

14. Доказуємо стійки криптосистеми.

15. Криптоаналітична атака повного перебору.

16. Види криптоаналітичних атак.

17. Теорема про необхідну та достатню умову теоретичної недешифруємості.

18. Поняття ентропії.

19. Відстань єдності.

20. Шифри простої заміни.

Лекція 3

Найпростіші шифри заміни та перестановки. Моноалфавітні та поліалфавітні шифри

Шифр підстановки (заміни) – це шифр, у якому кожен символ відкритого тексту у шифртексті замінюється іншим символом.

Найчастіше виділяють такі типи шифрів підстановки:

– проста підстановка, або моноалфавітна заміна – це шифр, який кожен символ відкритого тексту замінює відповідним символом шифртексту, при чому, конкретній літері відкритого повідомлення відповідає єдина, завжди одна і та сама, літера шифртексту;

– однозвучний шифр підстановки схожий на простий шифр підстановки за винятком того, що один символ відкритого тексту замінюється на один з декількох можливих символів шифртексту;

– поліграмний шифр підстановки – це шифр, який блоки символів шифрує по групах (біграма – це група з двох символів, триграма – з трьох символів і т.д).;

– поліалфавітна підстановка складається з декількох простих шифрів підстановки, тобто одна і та сама літера відкритого тексту може бути замінена кожен раз по різному (відбувається циклічне застосування декількох моноалфавітних шифрів);

– шифр перестановки – це шифр, у якому символи повідомлення переставляються місцями безпосередньо у відкритому тексті за певним правилом, що залежить від ключа.

За технологією шифрування розрізняють:

– блокові шифри здійснюють шифрування блоків фіксованої довжини, що складаються з послідовності символів відкритого тексту;

– потокові шифри здійснюють шифрування окремих символів відкритого тексту.

Поступово розглянемо приклади таких шифрів. Типовим прикладом шифрів перестановки є табличні шифри.

Магічні квадрати

Магічним квадратом називається квадратна таблиця з вписаними в комірки натуральними числами, починаючи з 1, які дають в сумі по кожному стовпцю, кожному рядку і кожній діагоналі одне і те ж число. Шифрований текст вписується в магічні квадрати відповідно до нумерації комірок (таблиця 3.1), а зчитується вміст таблиці по рядках і одержується шифртекст.

Наприклад.

Відкритий текст: ПРИЛІТАЮ ВОСЬМОГО.

Таблиця 3.1- Магічний квадрат

16	3	2	13
5	10	11	8
9	6	7	12
4	15	14	1

О	И	Р	М
І	О	С	Ю
В	Т	А	Ь
Л	Г	О	П

Шифртекст: ОИРМІОСЮВТАЬЛГОП.

Кількість магічних квадратів швидко зростає із збільшенням розміру квадрата. Наприклад, існують 880 квадратів з даними властивостями розміром 4×4, а розміром 5×5 вже близько 250 000.

Полібіанській квадрат

За два століття до нашої ери грецький письменник і історик Полібій запропонував для цілей шифрування квадратну таблицю розміром 5×5, заповнену буквами грецького алфавіту у випадковому порядку.

При шифруванні поступали таким чином. В полібіанському квадраті знаходили чергову букву відкритого тексту і замінювали її на букву, розташовану нижче в тому ж стовпці. Якщо буква опинялася в нижньому рядку таблиці, то брали саму верхню букву з того ж стовпця. Концепція полібіанського квадрата знайшла застосування в криптосистемах подальшого часу.

Здійснимо математичний аналіз шифрів простої заміни

Підстановка в алфавіті $\overline{Z_m}$ є взаємно однозначним відображенням π з $\overline{Z_m}$ на $\overline{Z_m}$: $\pi\colon\ t \longrightarrow \pi(t)$, яке замінює букву t відкритого тексту на букву $\pi(t)$ шифртексту. Множина всіх підстановок на $\overline{Z_m}$ називається симетричеськоюю групою $\overline{Z_m}$ і позначається $\overline{SYM}$ $(\overline{Z_m})$.

Симетрическа група $\overline{SYM}$ $(\overline{Z_m})$ володіє наступними властивостями.

1. Замкнутість. Добуток підстановок π_1 і π_2 є підстановкою.

$\pi\colon\ \overline{Z_m} \xrightarrow{\pi_2} \overline{Z_m} \xrightarrow{\pi_1} \overline{Z_m}$

$\pi\colon\ t \longrightarrow \pi_1(\pi_2(t))$.

2.Асоціативність. Обидва способи надання добутку підстановок $\pi_1\pi_2\pi_3$ дають однаковий результат.

$\pi_1(\pi_2\pi_3) = (\pi_1\pi_2)\pi_3$.

3. Існування одиничного елемента. Підстановка δ, що

$\delta(t) = t\ ,\ 0 \le t < m$ є єдиним одиничним елементом групи $\overline{SYM}$ $(\overline{Z_m})$ по множенню.

$\delta\pi = \pi$ для всіх

$\pi \in \overline{SYM}$ $(\overline{Z_m})$.

4.Існування зворотних елементів

$\pi\pi^{-1} = \delta$.

Зворотна підстановка позначається π^{-1}.

Вказані властивості – аксіоми групи.

Ключ K підстановки для алфавіту $\overline{Z_m}$ є послідовністю елементів симетрической групи з $\overline{Z_m}$.

Підстановка, визначувана ключем, є криптографічним перетворенням E_k, яке шифрує n - граму $(x_0, x_1, \ldots, x_{n-1})$ відкритого тексту в n - граму $(y_0, y_1, \ldots, y_{n-1})$ шифртекста, де $y_i = \pi_i(x_i)$,$0 \le i < n$ для кожного n, $n = 0,1,2, \ldots$

Криптографічне перетворення E_k називається одноалфавітною підстановкою, якщо значення π_i однаково для кожного i, $i = 0,1,2, \ldots$. Інакше E_k- багатоалфавітна підстановка.

Окрім розглянутих одноалфавітних підстановок розглянемо наступні.

Афінна система підстановок Цезара

Розглянемо один з найдавніших та найбільш поширених шифрів простої (моноалфавітної) заміни – шифр Цезара

Визначимо перетворення:

$E_{a,b}: \overline{Z_m} \longrightarrow \overline{Z_m}$

$E_{a,b} : t \longrightarrow E_{a,b}(t)$

$E_{a,b}(t) = at + b(mod\, m)$.

a, b - цілі числа $0 \le a, b < m$ і $НСД(a, m)$ =1.

В даному перетворенні буква, відповідна числу t замінюється на букву, відповідну числовому значенню $at + b(mod\, m)$.

Наприклад.

Нехай m=33, a=4, b=5. Виконується умова $НСД$ (4,33) = 1. Ми одержуємо наступну відповідність між кодами букв. Ми використовуватимемо для прикладу український алфавіт.

Наприклад слову КАЛАМБУР відкритого тексту відповідає шифртекст ШДЯДЄЗЮП (згідно з побудованою таблицею 3.2). Афінна система використовувалася на практиці декілька століть назад, зараз її застосування обмежується ілюстрацією основних криптологічних положень.

Таблиця 3. 2 - Афінна система підстановок Цезара

Символ	t	$4t+5(\mathrm{mod}\ 33)$
а	0	5
б	1	9
в	2	13
г	3	17
ґ	4	21
д	5	25
е	6	29
є	7	0
ж	8	4
з	9	8
и	10	12
і	11	16
ї	12	20
й	13	24
к	14	28
л	15	32
м	16	3
н	17	7
о	18	11
п	19	15
р	20	19
с	21	23
т	22	27
у	23	31
ф	24	2
х	25	6
ц	26	10
ч	27	14
ш	28	18
щ	29	22
ь	30	26
ю	31	30
я	32	1

Система Цезара з ключовим словом

Система Цезара з ключовим словом також є моноалфавітною підстановкою.

Наприклад.

Нехай обрано слово DIPLOMAT в латинському алфавіті як ключове слово (слово без повторень букв) і числовий ключ $K = 5$.

Ключове слово записується під буквами алфавіту, починаючи з букви, числовий код якої співпадає з вибраним числом $K = 5$.

0 1 2 3 4 5 6 …

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

V W X Y Z D I P L O M A T B C E F G H J K N Q R S U

Решта букв алфавіту підстановки записується після ключового слова в алфавітному порядку, виключаючи букви, які вже використані у ключовому слові.

Відкритий текст *SEND MORE MONEY* перетворюється в шифртекст *HZBY TCGZ TCBZS.*

Поширений метод зламу всіх розглянутих шифрів – частотний аналіз. Будь-який алфавіт володіє надмірністю. Можна скласти таблиці вірогідності різних символів в тексті для будь-якої мови (біграм, символів і т.і.). По цих таблицях буде легко відновити відкритий текст з шифртекста.

Система омофонів

Система омофонів забезпечує найпростіший захист від криптоаналітичних атак. Ідея заснована на підрахунку частот появи букв в шифртексті. Система омофонів є одноалфавітною, хоча при цьому букви початкового повідомлення мають декілька замін. Число замін береться пропорційним вірогідності появи букви у відкритому тексті.

Дані про розподіли вірогідності букв в англійському тексті для прикладу приведено в табл. 3.3. Букви в таблицях вказані в порядку убування вірогідності

їх появи в тексті. Наприклад, англійська буква E зустрічається в 123 раз частіше, ніж буква Z.

Таблиця 3.3 - Розподіл ймовірностей у англійських текстах

Буква	Ймов	Буква	Ймов	Буква	Ймов
E	0,123	L	0,040	B	0,016
T	0,096	D	0,036	G	0,016
A	0,081	C	0,032	V	0,009
O	0,079	U	0,031	K	0,005
N	0,072	P	0,023	Q	0,002
I	0,071	F	0,023	X	0,002
S	0,066	M	0,022	J	0,001
R	0,060	W	0,020	Z	0,001
H	0,051	Y	0,019		

Шифруючи букву початкового повідомлення, вибирають випадковим чином одну з її замін. Заміни (часто звуться омофонами) можуть бути представлений трьохрозрядними числами від 000 до 999.

Наприклад, в англійському алфавіті букві E привласнюється 123 випадкові номери, буквам B і G - по 16 номерів, а буквам J і Z - по 1 номеру.

Якщо омофони (заміни) привласнюються випадковим чином різним появам однієї і тієї ж букви, тоді кожний омофон з'являється у шифртексті рівноймовірно.

При такому підході до формування шифртекста простий підрахунок частот вже ніщо не дає криптоаналітику. Проте у принципі корисна також інформація про розподіл пар і трійок букв в різних природних мовах. Якщо цю інформацію використовувати при криптоаналізі, він буде проведений успішніше.

Шифр "подвійний квадрат" Уїнстона

В 1854 р. англієць Чарльз Уїтстон розробив новий метод шифрування біграмами, який називають "подвійним квадратом". Свою назву цей шифр

отримав по аналогії з полібіанським квадратом. Шифр Уїтстона відкрив новий етап в історії розвитку криптографії.

На відміну від полібіанського шифр "подвійний квадрат" використовує відразу дві таблиці, розміщені по одній горизонталі, а шифрування йде біграмами, як в шифрі Плейфейра. Ці не такі складні модифікації привели до появи якісно нової криптографічної системи ручного шифрування.

Шифр "подвійний квадрат" виявився дуже надійним і зручним і застосовувався Німеччиною навіть в роки другої світової війни.

Пояснимо процедуру шифрування цим шифром на прикладі. Таблиці для шифрування 3.4. Нехай є дві таблиці з випадково розташованими в них українськими алфавітами.

Перед шифруванням початкове повідомлення розбивають на біграми. Кожна біграма шифрується окремо. Першу букву біграми знаходять в лівій таблиці, а другу букву - в правій таблиці. Потім будують прямокутник так, щоб букви біграми лежали в його протилежних вершинах. Інші дві вершини цього прямокутника дають букви біграми шифртекста. Припустимо, що шифрується біграма ІЛ початкового тексту . Буква І знаходиться в стовпці 2 і рядку 1 лівої таблиці. Буква Л знаходиться в стовпці 3 і рядку 6 правої таблиці.

Таблиця 3.4 - Шифр "подвійний квадрат"

А	І	Р	Ч	Ю	Я		Ь	Ш	З	А	В	Б
И	Б	Ї	С	Ш	,		Р	Ю	Щ	Ж	Ґ	Г
П	З	В	Й	Т	Щ		С	П	Я	Ч	Є	Д
Ц	Ж	О	Г	К	У		Й	О	Т	_	Е	Ц
_	Х	Є	Н	Ґ	Л		Ї	К	Н	У	Ф	Х
.	Ь	Ф	Д	Е	М		І	И	Л	М	,	.

Це означає, що прямокутник освічений рядками 1 і 6, а також стовпцями 2 лівої таблиці і 3 правої таблиці.

Отже, в біграму шифртекста входять буква З, розташована в стовпці3 і рядку 1 правої таблиці, і буква Ь, розташована в стовпці 2 і рядку 6 лівої таблиці, тобто одержуємо біграму шифртекста ЗЬ. Якщо обидва букви біграми повідомлення лежать в одному рядку, то і букви шифртекста беруть з цього ж рядка.

Першу букву біграми шифртекста беруть з лівої таблиці в стовпці, відповідному другій букві біграми повідомлення. Друга ж буква біграми шифртекста береться з правої таблиці в стовпці, відповідному першій букві біграми повідомлення.

Тому біграма повідомлення ТЄ перетворюється в біграму шифртекста ЗЄ. Аналогічним чином шифруються всі біграми повідомлення:

Повідомлення ПР ИЛ ІТ АЮ

Шифртекст СИ Щ . ЗЖ ШИ

Шифрування методом "подвійного квадрата" дає вельми стійкий до розкриття і простій в застосуванні шифр. Злам шифртекста "подвійний квадрат" вимагає великих зусиль. При цьому довжина повідомлення повинна бути не менш тридцяти рядків.

Багатоалфавітні системи

Багатоалфавітні (поліалфавітні) підстановочні шифри були винайдені Ліном Баттістою (Lean Battista) в 1568 році. Основна ідея багатоалфавітних систем полягає в тому, що впродовж всього тексту одна і та ж літера може бути зашифрована по-різному. Тобто заміни для літери вибираються із багатьох алфавітів залежно від положення в тексті. Це є хорошим захистом від простого підрахунку частот, тому що не існує єдиного маскування для кожної літери в криптотексті.

Шифр Віженера

Однією із старих і найбільш відомих багатоалфавітних криптосистем є система Віженера, названа на честь французького криптографа Блейза Віженера (Vigenere).

Таблиця Віженера для латинського алфавіту наведена у таблиці 3.5. Цей метод був вперше опублікований в 1586 році. У даному шифрі ключ задається набором з d літер. Такі набори підписуються із повторенням під повідомленням, а, потім, отриману послідовність складають із відкритим текстом за модулем *n* (потужність алфавіту).

Тобто виходить наступна формула:

$$Vig_d(m_i) = (m_i + k_{i \bmod d}) \pmod{n}$$

Літеру шифротексту можна знаходити також із таблиці, як перетин стовпця, визначуваного літерою відкритого тексту, і рядка, визначуваною літерою ключа. В окремому випадку, при *d*=1, отримуємо шифр Цезаря.

Таблиця 3.5 - Квадрат Віженера для латинського алфавіту

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Наприклад повідомлення meeting point, ключ cipher.

Пишемо повідомлення і підписуємо ключ. Результат в таблиці 3.5 на перетині символів..

M E E T I N G P O I N T повідомлення

C I P H E R C I P H E R ключ

O MT A M E I X D P R K шифртекст

Шифр Гронсфельда

Для шифрування тут використовується цифровий ключ. Але кожна буква зміщується не на постійне число позицій, а на число позицій, що відповідає значенню ключа.

Ключ не обов'язково повинен бути таким же довгим як повідомлення, що шифрується. Якщо ключ коротше повідомлення, його просто повторюють по циклу.

Так, наприклад, якщо в тексті 10 символів, а довжина ключа 5 символів, то для шифрування ключ використовуватиметься з повтореннями.

Приклад:

Вихідний текст: "швидка перемога"

Ключ 14352

Ш В И Д К А П Е Р Е М О Г А
1 4 3 5 2 1 4 3 5 2 1 4 3 5
Зашифрований текст: «ЩЕЙИМБУЗХНТЕД»

Одноразова система шифрування

Майже всі застосовувані на практиці шифри характеризуються як умовно надійні, оскільки вони можуть бути переважно розкриті за наявності необмежених обчислювальних можливостей.

Абсолютно надійні шифри не можна зруйнувати навіть за використання необмежених обчислювальних можливостей. Існує єдиний такий шифр, застосовуваний на практиці, – одноразова система шифрування. Характерною

рисою одноразової системи шифрування є одноразове використання ключової послідовності.

Одноразову систему винайдено 1917 року американцями Дж. Моборном та Г. Вернамом. Для реалізації цієї системи іноді використовують одноразовий нотатник. Цей нотатник складено з відривних листків, на кожному з яких надруковано таблицю з випадковими числами (ключами) *К*і.

Нотатник виконується у двох екземплярах: один використовується відправником, а другий – одержувачем. Для кожного символу *Х*і повідомлення використовується власний ключ *К*і з таблиці лише одноразово.

Після того як таблицю використано, її має бути вилучено з нотатника і знищено. Шифрування нового повідомлення розпочинається з нового листка.

Цей шифр буде абсолютно надійний, якщо набір ключів *К*і буде насправді випадковий і непередбачуваний. Якщо криптоаналітик спробує використовувати для заданого шифртексту всі можливі набори ключів і відновити всі можливі варіанти вихідного тексту, то вони усі виявляться рівноймовірними.

Контрольні запитання та завдання

1. Які шифри називають шифрами заміни?
2. Що є ключем шифру заміни?
3. Наведіть приклади шифрів простої заміни. Опишіть алгоритм одного з них.
4. Які основні недоліки шифрів простої заміни?
5. У чому відмінність шифрів простої і складної заміни?
6. Які існують шифри складної заміни?
7. Система омофонів.
8. Магічні квадрати.
9. Одноразова система шифрування.
10. Система Цезаря з ключовим словом.
11. Афінна система підстановок Цезара.
12. Математичний аналіз шифрів простої заміни.

7. Яким чином для шифрування використовують «подвійний квадрат» Уітстона?

8. У чому полягає шифрування з використанням системи Віженера?

9. Зашифрувати повідомлення КРИПТОГРАФІЯ шифром Цезара з ключем $(m+1) \bmod 17$, де m - номер за списком у журналі.

10. Зашифрувати повідомлення КРИПТОГРАФІЯ афінною системою підстановок Цезара $(at+b) \bmod m$, де $1 < a, b < m,$ НСД$(a, m) = 1$ в українському алфавіті. Параметри a та b обрати самостійно згідно з вимогами.

11. Зашифрувати перші три букви свого імені , використавши шифр одноразового блокноту з ключем 110000 011110 010100. Букви імені для цього спочатку надати в двійковій формі (кожен блок з шести цифр є номером відповідної букви у двійковому запису).

12. Дана система Цезаря з ключовим словом. Алфавіт український. Зашифрувати повідомлення надане в наступній таблиці.

Ключ також надано в таблиці 3.6. Ключове слово ФОРЕЛЬ.

Таблиця 3.6

№ вар	Ключ	Відкрите повідомлення	№ вар	Ключ	Відкрите повідомлення
1	7	КАЛАМБУР	16	7	ЛІЦЕЙ
2	8	МАТЕМАТИКА	17	8	КНИГА
3	9	КРИПТОГРАФІЯ	18	9	ГАЗЕТА
4	10	КРИПТОАНАЛІЗ	19	10	ЖУРНАЛ
5	11	ШИФР	20	11	ФІЗИКА
6	12	АЛГЕБРА	21	12	ПРІНТЕР
7	13	ГЕОМЕТРІЯ	22	13	ТЕЛЕВІЗОР
8	14	ПРАВИЛО	23	14	АУДИТОРІЯ
9	15	ДИПЛОМ	24	15	ДЕКАНАТ
10	1	БІОЛОГІЯ	25	16	ФАКУЛЬТЕТ
11	2	ГЕОГРАФІЯ	26	17	СТУДЕНТ
12	3	ХІМІЯ	27	18	ЕКЗАМЕН
13	4	ІНСТИТУТ	28	19	ПРАКТИКА
14	5	ШКОЛА	29	20	КРЕДИТ
15	6	ГІМНАЗІЯ			

13. Зашифрувати шифром Гронсфельда повідомлення надане в таблиці завдання 6 згідно з варіантом. Ключ розрахувати за формулою 15148 – 16k, де k – номер за списком у журналі.

14. Зашифрувати шифром Віженера повідомлення надане в таблиці у відповідності з номером по списку в журналі. Ключ також надано у таблиці 3.7.

Таблиця 3.7

1	ДОБРИЙ РАНОК	ЯБЛУКО
2	ВАЖКА СПРАВА	ФОРЕЛЬ
3	У НЕДІЛЮ СЬОМОГО	ЯКОР
4	ЗАВТРА ВРАНЦІ	КЛЮЧ
5	ДОПОМОГА ПРИЙДЕ	ДИПЛОМ
6	ЧЕКАЙТЕ ПРИЇЗДУ	СЕЗОН
7	СПРАВУ ЗАКІНЧЕНО	ЛЕМОН
8	ЗАЛИШАЙТЕСЬ НА МІСЦІ	БЛИСК
9	ВАЖКИЙ ІСПИТ	БРАТ
10	НЕСПОДІВАНА ЗУСТРІЧ	МОРЕ
11	ТРИВАЛЕ ВІДРЯДЖЕННЯ	КНИГА
12	ЗАГАЛЬНІ ЗБОРИ	ЯБЛУКО
13	ШВИДКИЙ ПОРЯТУНОК	ФОРЕЛЬ
14	ОСТАННЯ НАДІЯ	ЯКОР
15	ЧЕКАТИ ЗВІСТКИ	КЛЮЧ
16	ЗАЛИШИТЬ МІСТО	ДИПЛОМ
17	РЯТУВАТИ МАЙНО	СЕЗОН
18	ВІДШКОДУВАТИ ВТРАТИ	ЛЕМОН
19	КІЛЬКОРАЗОВЕ ШИФРУВАННЯ	БЛИСК
20	КРИПТОГРАФІЧНЕ ЗАСТОСУВАННЯ	БРАТ
21	СИМЕТРИЧНА КРИПТОСИСТЕМА	МОРЕ
22	ШИФР ВІЖЕНЕРА	КНИГА
23	ДОБРИЙ ВЧИНОК	СЕЗОН
24	МІСТО ЗУСТРІЧІ	ЛЕМОН
25	ВАЖКА СПРАВА	БЛИСК
26	У НЕДІЛЮ СЬОМОГО	БРАТ
27	ЗАВТРА ВРАНЦІ	МОРЕ
28	ОСТАННЯ НАДІЯ	ЯКОР
29	ЧЕКАТИ ЗВІСТКИ	КЛЮЧ
30	ЗАЛИШИТЬ МІСТО	ДИПЛОМ

Лекція 4

Математичні алгоритми, які найчастіше використовуються в криптографії

Розглянемо математичні алгоритми, які найчастіше використовуються в криптографічних протоколах.

Алгоритм Евкліда

Нагадаємо, що два числа називають взаємнопростими, якщо вони не мають спільних дільників, крім 1. Іншими словами, якщо найбільший спільний дільник чисел a і n дорівнює 1, то ці числа називають взаємнопростими і записують $HCD\ (a,n)=1$. Один з шляхів обчислення найбільшого спільного дільника двох чисел – використання алгоритму Евкліда. Евклід описав цей алгоритм у своїй книзі „Елементи”, датованій 300 роком до н. е.

Однак алгоритм був створений не Евклідом. Історики вважають, що алгоритм на 200 років старіше.

Це найдавніший нетривіальний алгоритм, що дійшов до наших днів. Будемо позначати найбільший спільний дільник чисел A і B через (A,B).

Нагадаємо, що алгоритм Евкліда полягає у послідовному виконанні операції ділення із залишком до отримання нульового залишку.

Нехай $A>B>0$. Позначимо $A=r_{-1}$, $B=r_0$ і $r_{i-2}=d_i r_{i-1}+r_i$ при $i=1,\dots,k$ і $r_{k-1}=d_{k+1}r_k$. Тоді $(A,B)=r_k$ і до отримання залишку $r_{k+1}=0$ необхідно виконати $k+1$ ділення.

Оцінимо число ділень, що виконується в алгоритмі Евкліда. Для цього розглянемо послідовність чисел Фібоначчі

$$f_0=0, f_1=1, f_k=f_{k-1}+f_{k-2}, k\geq 2.$$

Лема 4.1 При $k>1$ вірна нерівність $f_k \geq R^{k-2}$, *де* $R=\dfrac{1+\sqrt{5}}{2}$.

Доведення:

Застосуємо індукцію за n. При $k = 2$ твердження є очевидним.

Далі, використовуючи припущення індукції, маємо:

$$f_{k+1} = f_k + f_{k-1} \geq R^{k-2} + R^{k-3} = R^{k-3}(R+1) = R^{k-3}R^2 = R^{k-1}.$$

Так як R є додатнім коренем рівняння $x^2 = x + 1$.

Дійсно, $R + 1 = \frac{3+\sqrt{5}}{2}$ и $R^2 = \frac{3+\sqrt{5}}{2}$.

Теорема 4. 1 (Ламе, 1844). Для будь-якого натурального числа $N > 0$ число ділень в алгоритмі Евкліда для знаходження найбільшого спільного дільника чисел *A і B,* $0 < A < B \leq N$ не перевершує $1 + \lfloor log_R N \rfloor$.

Доведення:

Доведемо, що $f_i \leq r_{k+1-i}$ при $i = 1, \ldots, k+2$. При $i = 1$ вірно. Для $i + 1$, пам'ятаючи про індукцію, маємо

$$r_{k-i} = d_{k-i+2}r_{k-i+1} + r_{k-i+2} \geq r_{k-i+1} + r_{k-i+2} \geq f_i + f_{i-1} = f_{i+1}.$$

Тому, $A \geq r_{-1} \geq f_{k+2} \geq R^k$, звідки виходить шукана оцінка числа ділень $k + 1 \leq 1 + \lfloor log_R N \rfloor$.

$$(N \geq R^k \Rightarrow log_R N \geq k\, log_R R \Rightarrow k \leq log_R N \Rightarrow k + 1 \leq 1 + \lfloor log_R N \rfloor)$$

Можна також відмітити, що алгоритм Евкліда широко використовується не тільки при визначенні спільних дільників чисел, але й поліномів. А саме, у теорії подільності його використовують при тестуванні взаємної простоти двох поліномів з коефіцієнтами, наприклад, з кінцевого поля.

Задача обчислення зворотних значень за модулем трохи складніша. Іноді вона має розв'язок, іноді – ні.

Наприклад, зворотне значення 5 за модулем 14 дорівнює 3. З іншого боку, число 2 не має зворотного значення за модулем 14.

У загальному випадку рівняння $a^{-1} \equiv x (mod\, n)$ має єдиний розв'язок, якщо a і n взаємнопрості. Якщо a і n не є взаємнопростими., то порівняння $a^{-1} \equiv x (mod\, n)$ не має рішень. Зворотне значення за модулем n теж можна обчислити за допомогою розширеного алгоритму Евкліда.

Розширений алгоритм Евкліда

Розглянемо алгоритм, що дозволяє не тільки знаходити *НСД* чисел A і B, а ще й знаходити цілі числа x і y , що задовольняють рівності $Ax + By = (A, B)$.

Від звичайного алгоритму Евкліда він відрізняється тим, що разом з послідовністю залишків r_i обчислюються ще дві допоміжні послідовності x_i і y_i.

Наприклад потрібно вирішити діофантове рівняння вигляду

$5x + 7y = НСД\ (5,7)$

1. $r_{-1} = 5;\ r_0 = 7$

$x_{-1} = 1;\ y_{-1} = 0;\ x_0 = 0;\ y_0 = 1$

$d_1 = \lfloor r_{-1} / r_0 \rfloor = \lfloor 5 / 7 \rfloor = 0$

$r_1 = r_{-1} - d_1 r_0 = 5 - 0 \times 7 = 5$

$x_1 = x_{-1} - d_1 x_0 = 1 - 0 \times 0 = 1$

$y_1 = y_{-1} - d_1 y_0 = 0 - 0 \times 1 = 0$

2. $d_2 = \lfloor r_0 / r_1 \rfloor = \lfloor 7 / 5 \rfloor = 1$

$r_2 = r_0 - d_2 r_1 = 7 - 1 \times 5 = 2$

$x_2 = x_0 - d_2 x_1 = 0 - 1 \times 1 = -1$

$y_2 = y_0 - d_2 y_1 = 1 - 1 \times 0 = 1$

3. $d_3 = \lfloor r_1 / r_2 \rfloor = \lfloor 5 / 2 \rfloor = 2$

$r_3 = r_1 - d_3 r_2 = 5 - 2 \times 2 = 1$

$x_3 = x_1 - d_3 x_2 = 1 - 2 \times (-1) = 3$

$y_3 = y_1 - d_3 y_2 = 0 - 2 \times 1 = -2$

4. $d_4 = \lfloor r_2 / r_3 \rfloor = \lfloor 2 / 1 \rfloor = 2$

$r_4 = r_2 - d_4 r_3 = 2 - 2 \times 1 = 0$

Дійсно, знайдені числа x та y, що задовольняють рівності

НСД(5,7)

$5 \times 3 + 7 \times (-2) = НСД\ (5,7)$

Сам алгоритм можна подати таким чином:

$r_{-1} = A;\ r_0 = B$

$x_{-1} = 1; y_{-1} = 0; x_0 = 0; y_0 = 1$

for $i = 1$ *until* $r_i > 0$ *do*

begin

$d_i = \lfloor r_{i-2} / r_{i-1} \rfloor;$

$r_i = r_{i-2} - d_i r_{i-1};$

$x_i = x_{i-2} - d_i x_{i-1};$

$y_i = y_{i-2} - d_i y_{i-1};$

$i = i + 1;$

end

Значення x_k та y_k, за яких $r_k = (A, B)$ будуть шуканими в силу наступного твердження:

Лема 4.2

При всіх i , $-1 < i \leq k$,виконується рівність

$$x_i A + y_i B = r_i.$$

Застосуємо індукцію по i . *При* $i = -1,0$ рівність очевидна.

Якщо рівності доведені для всіх значень індексів менших від i, то для $i + 1$, користуючись індуктивним припущенням, отримуємо

$$x_i A + y_i B = (x_{i-2} - d_i x_{i-1})A + (y_{i-2} - d_i y_{i-1})B =$$
$$= (x_{i-2}A + y_{i-2}B) - d_i(x_{i-1}A + y_{i-1}B) = r_i.$$

Використання модульної арифметики

Модулярні обчислення часто використовують в криптографії, оскільки обчислення дискретних логарифмів та квадратних коренів за модулем n може стати складною задачею.

До того ж, модулярну арифметику простіше реалізувати на комп'ютерах, так як вона обмежує діапазон проміжних значень результату. Для k -бітних модулів довжина проміжних результатів будь-якої операції додавання, віднімання або множення не перевищує $2k$ бітів. Тому за допомогою модулярної арифметики можна, наприклад, виконати піднесення до степені без громіздких проміжних результатів.

Обчислення степені деякого числа за модулем іншого числа, $a^x(mod\, n)$ являє собою просту послідовність операцій множення і ділення, однак відомі методи їх прискорення. Один з таких методів прагне мінімізувати кількість множень за модулем. Оскільки операції дистрибутивні, піднесення до степеню виконується швидше як послідовність множень, кожного разу отримуючи лишки. Зараз ви не відчуваєте різниці, але вона стане помітною за множення 200-розрядних чисел.

Наприклад, обчислюючи $a^8\, mod\, n$, можна не виконувати сім множень і одне величезне приведення за модулем:

$$(a * a * a * a * a * a * a * a)\, mod\, n.$$

Замість цього Виконайте три менших множення і три менших приведення за модулем:

$$((a^2\, mod\, n)^2\, mod\, n)^2\, mod\, n.$$

Під час обчислень із цілими числами часто використовують наступне.

Якщо відомо, що початкові числа і результати обчислень обмежені деяким числом M (при цьому припускають нерівності двох видів $0 \le N \le< M$ або $-\frac{M}{2} < N < \frac{M}{2}$, то обчислення можна проводити в кільці лишків Z_M, ототожнюючи числа з указаних інтервалів і відповідні лишки.

Щодо M, то його можна обирати різними шляхами, причому цей вибір переважно визначає складність обчислень. Найбільш ефективним такий підхід є у випадку, коли число M може бути представлене у виді добутку невеликих взаємнопростих чисел $M = m_1, m_2, ..., m_k$, оскільки в даному випадку можна скористуватися ізоморфізмом кілець.

$$Z_M \cong Z_{m_1} + Z_{m_2} + ... + Z_{m_k}.$$

При цьому у співвідношенні кожному числу з інтервалу $0 \le U < M$ відповідає набір $(u_1, u_2, \ldots, u_k)$, *де* $u_i = u(mod\, m_i)$, $i = 1, \ldots, k$. У даному випадку маємо на увазі найменший невід'ємний залишок від ділення числа u на число m. Замість обчислень з початковими числами можна перейти до їх

залишків і проводити усі обчислення в кільцях Z_{m_i}, $i = 1, \ldots, k$, а потім, отримавши результат, виконати зворотній перехід і відновити шукане число за залишками. Для виконання зворотного переходу застосовують китайську теорему про залишки. Більш докладно теорема про залишки розглядається в теорії чисел. Використання модулярної арифметики вигідне тільки якщо у розпорядженні є засоби швидкого переходу до модулярного стану та навпаки.

Теорема 4.2 Нехай $M = m_1 \cdot m_2 \cdot \ldots \cdot m_k$, *де* m_i попарно взаємнопрості, і

$$c_i = m_1 \cdot \ldots \cdot m_{i-1} \cdot m_{i+1} \cdot \ldots \cdot m_k = \frac{M}{m_i},$$

$d_i = c_i^{-1} \, mod \, m_i, i = 1, \ldots, k$. *(*тут і далі запис $c \, mod \, m$, на відміну від запису правої частини порівняння $c(mod \, m)$, означає найменший невід'ємний залишок від ділення числа c на число m *).*

Тоді розв'язок системи порівнянь

$$u \equiv u_i(mod \, m_i), i = 1, \ldots, k,$$

існує однозначно за модулем M і знаходиться за формулою

$$U = \sum_{i=1}^{k} c_i d_i u_i \bmod M.$$

Доведення легко витікає з наступних властивостей обраних чисел:

$$c_i d_i \equiv 0(\bmod m_i), \text{ при } j \neq i,$$

$$c_i d_i \equiv 1(\bmod m_i), i, j = 1, \ldots, k$$

Алгоритм Монтгомері

Ефективний шлях багаторазового приведення за модулем – використання методу Монтгомері, який було запропоновано в 1985 році. Цей метод особливо ефективний при апаратній реалізації алгоритмів. Дуже зручно відмовитися від операцій множення і ділення та замінити їх операціями додавання. Метод полягає в наступному. Нехай N - непарне число, потрібно помножити лишки $A = \sum_{i=0}^{n-1} 2^i a_i$ і B.

Розглянемо алгоритм:

R = 0;

for i = 0 until i < n do

begin

if a_i = 1 then R = R + B;

if R – непарне then R = R + N;

R = R / 2;

end

if R ≥ N then R = R – N.

Суть даного алгоритму полягає в тім, що в силу рівності

$$A = \sum_{i=0}^{n-1} 2^i a_i = (\ldots(2a_{n-1} + a_{n-2})2 + \ldots + a_1)2 + a_0$$

множення числа B на число A зводиться до обчислення

$$AB = a_0 B + 2(a_1 B + \ldots 2(a_{n-2} B + 2a_{n-1} B)\ldots).$$

Воно виконується за n кроків, на кожнім з яких здійснюється додавання до поточного значення R значення $a_i B$, $i = 0, \ldots, n-1$, з наступним діленням на 2. Завдяки цьому діленню отримані значення завжди знаходяться в інтервалі $0 < R < N$. У результаті роботи даного алгоритму виходить число $2^{-n} AB \pmod N$.

Тепер для одержання числа $AB \pmod N$ необхідно застосувати ще один раз даний алгоритм до чисел $2^{2n} \pmod N$ і $2^{-n} AB \pmod N$. Оскільки число $2^{2n} \pmod N$ обчислюється за допомогою зрушень і вирахувань, його можна обчислити заздалегідь і зберігати отримане значення).

Наприклад:

Нехай $\boldsymbol{A}$ = 1×2^0 + 0×2^1 + 1×2^2 + 0×2^3 + 1×2^4 = 1 + 4 + 16 = **21** (10101)

B = 18

N = 41

Зрозуміло, що AB(mod N) = 21×18 (mod 41) = 9

Обчислимо добуток цих чисел за допомогою вищевказаного алгоритму.

1. R = 0

$a_0 = 1$

$R = R + B = 0 + 18 = 18;$

R – парне;

$R = R / 2 = 9.$

2. $a_1 = 0;$

R – непарне;

$R = R + N = 9 + 41 = 50;$

$R = R / 2 = 25;$

3. $a_2 = 1$

$R = R + B = 25 + 18 = 43;$

R - непарне;

$R = R + N = 43 + 41 = 84;$

$R = R / 2 = 42;$

4. $a_3 = 0;$

R - парне;

$R = R / 2 = 21;$

5. $a_4 = 1$

$R = R + B = 21 + 18 = 39;$

R – непарне;

$R = R + N = 39 + 41 = 80;$

$R = R / 2 = 40;$

$R < N \rightarrow R = 40.$

Це ми одержали 2^{-n} AB (mod N).

Тепер ми повинні ще раз скористатися цим алгоритмом для обчислення AB (mod N).

A' = 2^{2n} (mod N) = $2^{2 \times 5}$ (mod N) = 1024 (mod 41) = 40 = $0\times2^0 + 0\times2^1 + 0\times2^2 + 1\times2^3 + 0\times2^4 + 1\times2^5$

B '= 40;

N = 41.

1. R = 0

   $a_0 = 0$

   R – парне;

   R = R / 2 = 0.

2. $a_1 = 0$;

   R – парне;

   R = R / 2 = 0;

3. $a_2 = 0$

   R – парне;

   R = R / 2 = 0;

4. $a_3 = 1$;

   R = R + B = 0 + 40 = 40;

   R – парне;

   R = R / 2 = 20;

5. $a_4 = 0$;

   R – парне;

   R = R / 2 = 10;

6. $a_5 = 1$;

   R = R + B = 10 + 40 = 50;

   R = R - N = 50 – 41 = 9.

Перевірка показує, що рішення вірне.

Контрольні питання та задачі

1. У чому полягає алгоритм Евкліда?
2. Навести розширений алгоритм Евкліда.
3. Застосування розширеного алгоритму Евкліда в криптографії. Навести приклади.
4. Як використовується модульна арифметики в криптографії?

5. За допомогою алгоритму Евкліда знайти НСД пар чисел: (187,34), (841,160), (2613,2171).

6. Знайти НСД поліномів $f(x) = x^3 + x^2 - 10x + 2$ та $g(x) = x^3 + 3x^2 - x + 3$ у кільці $Q[x]$.

7. Написати програму, яка реалізує алгоритм Евкліда, розширений алгоритм Евкліда.

8. Яка сутність методу Монтгомері знаходження добутку двох чисел?

9. За допомогою розширеного алгоритму Евкліда знайти цілі числа x та y, які задовольняють рівнянню $Ax + By = (A, B)$. Рівняння надані у таблиці 4.1.

10. Написати програму, яка реалізує алгоритм Монтгомері.

Таблиця 4.1.

№ вар.	Рівняння.	№ вар.	Рівняння.
1	$3x + 4y = \text{НОД}(3,4)$	15	$3x + 12y = \text{НОД}(3,12)$
2	$3x + 7y = \text{НОД}(3,7)$	16	$7x + 12y = \text{НОД}(7,12)$
3	$5x + 3y = \text{НОД}(3,5)$	17	$5x + 12y = \text{НОД}(5,12)$
4	$2x + 3y = \text{НОД}(2,3)$	18	$11x + 12y = \text{НОД}(11,12)$
5	$2x + 5y = \text{НОД}(2,5)$	19	$3x + 16y = \text{НОД}(3,16)$
6	$2x + 7y = \text{НОД}(2,7)$	20	$8x + 17y = \text{НОД}(8,17)$
7	$2x + 9y = \text{НОД}(2,9)$	21	$7x + 15y = \text{НОД}(7,15)$
8	$5x + 13y = \text{НОД}(5,13)$	22	$9x + 19y = \text{НОД}(9,19)$
9	$10x + 11y = \text{НОД}(10,11)$	23	$9x + 17y = \text{НОД}(9,17)$
10	$13x + 12y = \text{НОД}(12,13)$	24	$2x + 7y = \text{НОД}(2,7)$
11	$2x + 4y = \text{НОД}(2,4)$	25	$2x + 9y = \text{НОД}(2,9)$
12	$4x + 8y = \text{НОД}(4,8)$	26	$5x + 13y = \text{НОД}(5,13)$
13	$5x + 10y = \text{НОД}(5,10)$	27	$10x + 11y = \text{НОД}(10,11)$
14	$2x + 11y = \text{НОД}(2,11)$	28	$2x + 5y = \text{НОД}(2,5)$

11. За допомогою алгоритму Монтгомери знайти добуток чисел $A \times B (\text{mod}\, N)$. Дані надані в таблиці 4.2 у залежності від номеру у журналі.

Таблиця 4.2.

№	*A*	*B*	*N*
1	21	18	37
2	19	16	37
3	23	18	37
4	25	16	37
5	25	18	37
6	21	16	41
7	27	20	41
8	25	20	41
9	27	18	41
10	29	16	41
11	21	14	29
12	21	16	29
13	21	12	29
14	19	16	29
15	19	14	29
16	27	18	31
17	27	20	31
18	27	16	31
19	27	14	31
20	25	16	31
21	33	18	47
22	33	16	47
23	35	18	47
24	35	16	47

Лекція 5

Класичні симетричні криптосистеми. Шифрування методом гамування. Стандарт блокового симетричного шифрування DES

Під гамуванням розуміють процес накладення по певному закону гами шифру на відкриті дані.

Гама-шифру – псевдовипадкова послідовність, вироблена по заданому алгоритму для зашифрування відкритих даних та розшифрування зашифрованих.

Перед зашифруванням, відкриті дані розбивають на блоки $T_0^{(i))}$ однакової довжини (звичайно по 64 бита). Гамма шифру виробляється у вигляді послідовності блоків $Г_ш^{(i)}$ аналогічної довжини.

$$T_ш^{(i)} = Г_ш^{(i)} \oplus T_0^{(i)}, \; i = 1,...,M.$$

M – кількість блоків відкритого тексту.

Процес розшифрування зводиться до повторної генерації гами і накладенні цієї гами на зашифровані дані.

$$T_0^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)}, \; i = 1,...,M.$$

Криптостійкість шифру визначається довжиною ключа.

Для генерації псевдовипадкових послідовностей застосовуються криптографічний стійкі генератори. До них пред'являються наступні вимоги:

– період гами повинен бути достатньо великим;

– гама повинна бути достатньо непередбачуваною;

– генерація гами не повинна викликати великих технічних складнощів.

На думку Шеннона в практичних шифрах необхідно використовувати два загальні принципи – розсіювання і перемішування.

Розсіювання – розповсюдження впливу одного знаку відкритого тексту на багато знаків шифртекста, що дозволяє приховати статистичні властивості відкритого тексту.

Перемішування – використовування таких шифруючих перетворень, які ускладнюють відновлення взаємозв'язку статистичних властивостей відкритого та шифрованого текстів.

Сучасні симетричні криптосистеми – це складові шифри, реалізовані у вигляді деякої послідовності простих шифрів, кожний з яких вносить свій внесок в сумарне розсіювання і перемішування.

Як найпростіші шифри частіше за все використовують шифри заміни і перестановки. В сучасному блоковому шифрі блоки є двійковими послідовностями завдовжки 64,128,192 або 256 біт. Одержують достатньо стійкі шифри. Типовими прикладами симетричних криптосистем є стандарти IDEA, DES, Калина, RIJNDAEL, Camellia .

DES (Date Encryption Standard) опубліковано в 1977 році. Він призначений для захисту від НСД важливої, але несекретної інформації, в державних і комерційних структурах США. В даний час стандарт все ще використовується в системах захисту комерційної інформації, хоча з розвитком інформаційних технологій в майбутньому, можливо, носитиме тільки ілюстративний характер.

Основні достоїнства DES:

– використовується тільки один ключ завдовжки 56 біт;

– відносна простота алгоритму забезпечує високу швидкість;

– достатньо висока стійкість алгоритму.

Алгоритм DES здійснює шифрування 64-бітових блоків даних за допомогою 64-бітового ключа, в якому значущими є 56 біт (8 біт для контролю на парність). Узагальнена схема алгоритму надана на рисунку 5.1.

Можливі 4 режима роботи DES:

– ECB (Electronic Code Book) - електронна кодова книга;

– CBC (Cipherblock Chainsng) - зчеплення блоків шифру;

– CFB (Cipher Feedback) - зворотний зв'язок по криптотексту;

– OFB (Output Feedback) - зворотний зв'язок по виходу.

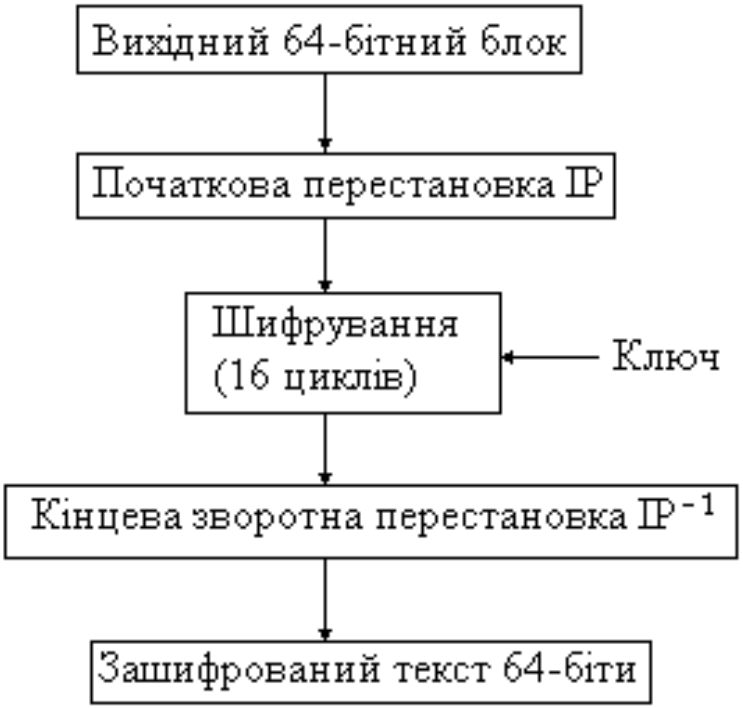

Рис 5.1 – Загальна схема алгоритму DES

В режимі ECB послідовність двійкових символів відкритого тексту розбивається на 64-бітові блоки і потім здійснюється їх незалежне шифрування за допомогою одного і того ж 64-бітного ключа. Для шифрування використовуються 56 біт ключа з 64, інші 8 використовуються для контролю спотворень ключа. Структура перетворень циклової функції представлена на рис. 5.2.

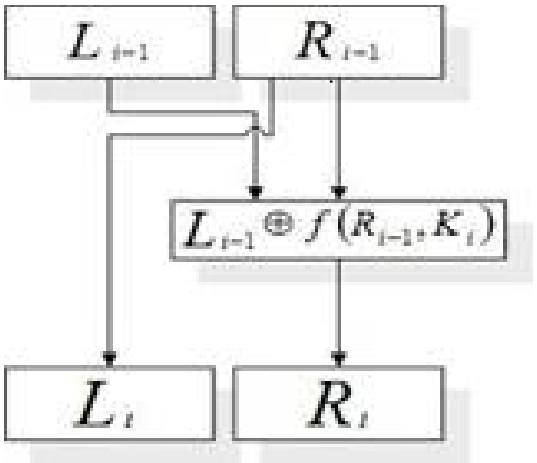

Рис 5.2 – Циклова функція DES

Розглянемо основні операції процедури шифрування E. Вхідний блок M з 64 двійкових символів розбивається на дві рівні частини по 32 біта: ліву L_0 та праву R_0. Потім здійснюються 16 циклів перетворення повідомлення $M = (L_0 R_0)$. Тут запис $(L_0 R_0)$ позначає операцію конкатенації (об'єднання) блоків.

$L_{i-1}, L_i, R_{i-1}, R_i$ – ліві і праві 32-бітові напівбокси;

K_i, $i = 1, \ldots, 16$ – 48 бітні двійкові послідовності (циклові підключи), формовані з ключа шифрування K;

$f(R_{i-1}, K_i)$ – циклова функція перетворення 32-бітового слова в 32-бітове;

СМ – побітовий суматор за модулем 2.

Як випливає з рис.5.2, в i-тому циклі слово $(L_{i-1}\ R_{i-1})$ перетвориться в $(L_i\ R_i)$ за правилами:

$$L_i = R_{i-1},\ R_i = L_{i-1} \oplus f(R_{i-1}, K_i).$$

Тут символ $\oplus$ позначає операцію побітового додавання за модулем 2. Схема виконання перетворення $f(R_{i-1}, K_i)$ на рис. 5. 3.

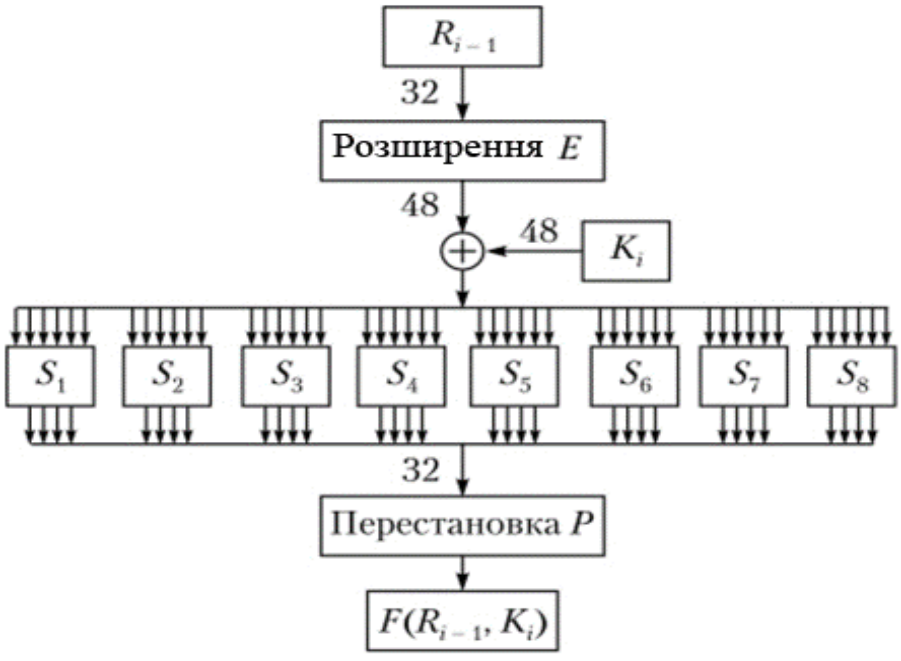

Рис.5.3 – Схема обчислення перетворення $f(R_{i-1}, K_i)$

Обчислення починається з так званого розширення, що перетворює 32 - бітовий напівблок в 48 бітовий відповідно до розширюючої таблиці E (фіксованого вигляду), яка фактично додатково вставляють копії 16 позицій.

Отримані 48 біт гамуються (побітово додаються) до 48 бітового підключа K_i. Результат гамування розбивається на 8 блоків по 6 біт, що поступають на так звані $S-$блоки $S_1, S_2, \ldots, S_8$. В кожному з $S-$ блоків вхідні 6 біт замінюються на 4 вихідних, причому взаємозв'язку між вхідними і вихідними символами визначається за допомогою 8 фіксованих таблиць – таблиць підстановок.

Вісім 4-бітових двійкових блоків, що поступають з виходів блоків, утворюють 32-бітовий напівблок, який піддається перестановці, ще однією фіксованою таблицею, що задається P.

Для повного опису алгоритму DES залишається відзначити ще декілька деталей.

Відповідно до стандарту процедура шифрування починається з початкової фіксованої перестановки і завершується застосуванням до отриманого результату зворотної перестановки IP^{-1}.

З початкових 64 біт ключа K алгоритму DES для шифрування використовується тільки 56. Практично 64 біт ключа розбивається на вісім 8-бітових блоків.

Восьмий біт кожного блоку служить для контролю парності, який дозволяє виявляти помилки типу спотворення непарного числа бітів (такі помилки можуть виникати при зберіганні або безпосередньо в процесі формування ключів) Ці контрольні біти в процесі формування ключів не беруть участь.

Кожний з циклових підключей $K_1, \ldots, K_{16}$ виходить з 56 бітів, що залишилися, ключа K за допомогою операцій перестановок і циклічних зсувів на відповідне для кожного циклового підключа K_i число позицій.

Помітимо також, що в самому кінці виконання всі 16 циклів алгоритму DES лівий і правий напівблоки міняються місцями, так що результатом роботи алгоритму є $E_K(M) = (R_{16} L_{16})$.

Розшифрування. Алгоритм розшифрування DES достатньо очевидний: Достатньо ″прогнати ″ DES з тим же ключем у зворотному напрямі. При цьому

$$R_{i-1} = L_i,$$

$$L_{i-1} = R_i \oplus f\left(R_{i-1}, K_i\right).$$

Схема алгоритму в режимі ECB надана на рисунку 5.4.

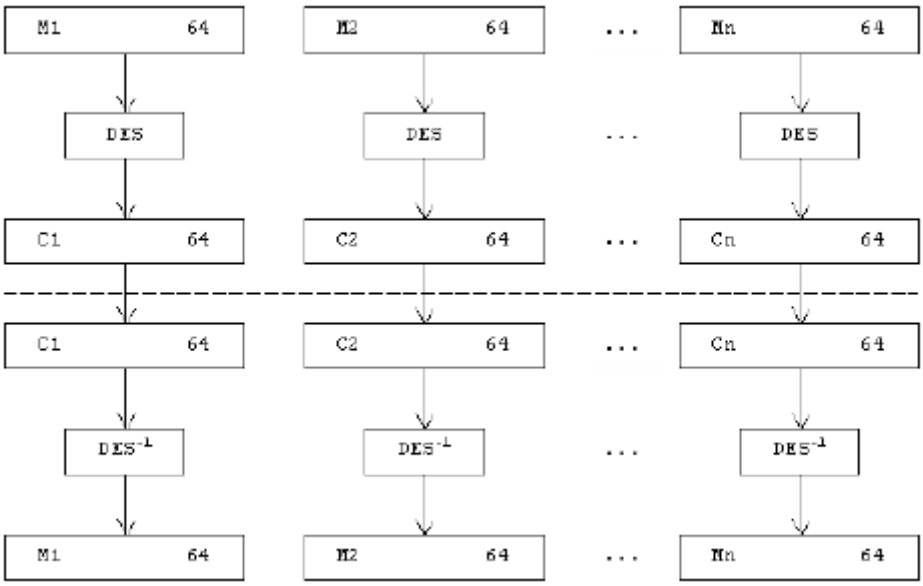

Рис. 5.4 – Робота алгоритму DES в режимі ECB

Розглянутий режим ECB володіє тим недоліком, що результати шифрування однакових блоків, за допомогою одного і того ж ключа співпадають.

З урахуванням того, що довжина блоку криптотекста (криптограми) постійна і відносно невелика, наявність у криптотексті ідентичних блоків служить явною вказівкою на присутність співпадаючих блоків і у відповідних фрагментах відкритого тексту, що в загальному випадку може істотно полегшити криптоаналіз.

З другого боку, спостереження за інформаційним обміном між двома абонентами стосовно ситуації, коли відправник повторно посилає відкритий текст, зашифрований одним і тим же ключем, дозволяє імітувати повернення тієї

ж криптограми-відповіді, який був переданий одержувачем в першому випадку. Як видно, при цьому DES не забезпечує імітостійкості.

Нарешті, алгоритм DES в режимі ECB володіє властивістю так званого розмноження помилок, яке полягає в тому, що спотворення одного біта криптограми приводить до спотворення декількох (близько половини) біт у відкритому тексті, отриманому в результаті розшифрування.

Для нейтралізації цих недоліків розроблений спеціальний режим використовування DES, званий CBC (режим зчеплення блоків шифру).

Схема алгоритму в режимі CBC надана на рисунку 5.5.

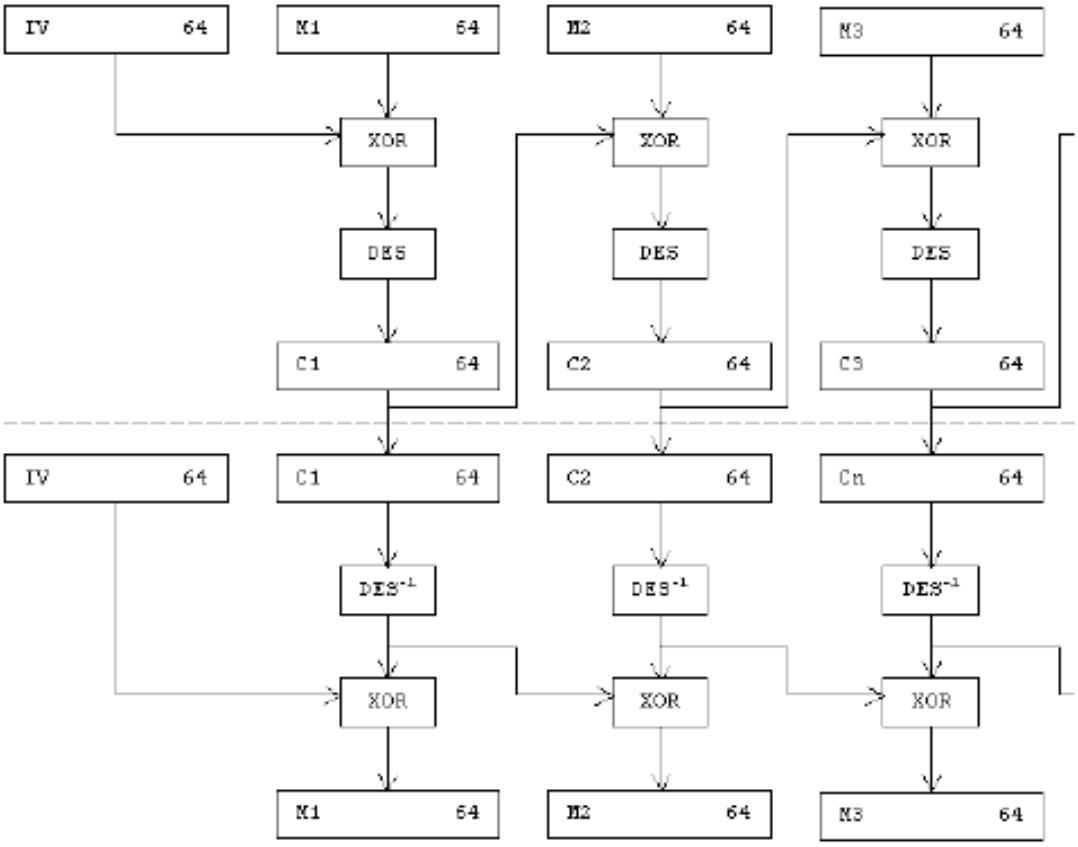

Рис. 5.5 – Робота алгоритму в режимі CBC

В цьому режимі повідомлення M розбивається на 64-бітові блоки $M_1, \ldots, M_n$. До кожного 64 бітового блоку M_i відкритого тексту на поточному кроці шифрування, перед подачею його на вхід алгоритму шифрування, за

допомогою побітового додавання за модулем 2 додається зашифрований блок C_{i-1}, отриманий на попередньому кроці, так що

$$C_i = E_K\left(M_i \oplus C_{i-1}\right).$$

Якщо два різні повідомлення починаються однаковими блоками, то співпадаючі перші блоки матимуть і відповідні криптотексти. Щоб захиститися і від цього недоліку, перед шифруванням кожному повідомленню M приписується початкове випадкове 64-бітове число – так званий вектор ініціалізації C_0.

Розшифрування виконується за алгоритмом

$$M_i \oplus C_{i-1} = E_K^{-1}\left(C_i\right)$$

або, що те ж саме

$$M_i = C_{i-1} \oplus E_K^{-1}\left(C_i\right)$$

Як випливає з приведених співвідношень, кожний черговий блок криптограми, є функцією попередніх. Тому спотворення одного біта в криптотексті спотворює два суміжні блоки, отримані в результаті розшифрування.

Проте, оскільки спотворений блок криптограми гамується з наступним блоком відкритого тексту, число спотворень в наступному блоці, що розшифровується, дорівнює числу спотворень у криптотексті. Що стосується криптостійкості, то режим CBC вважається більш стійким, ніж ECB з двох причин. По-перше, криптотекст є функцією не тільки відкритого тексту і ключа, але і зрештою початкового вектора C_0. Це, звичайно, ускладнює криптоаналіз. З другого боку, ідентичним блокам відкритого тексту в загальному випадку відповідають різні блоки криптограми, що нейтралізує основний недолік ECB.

У відмінності від режимів ECB і CBC, оперуючих 64-бітовими блоками, режими CFB і OFB оперують з k - бітовими блоками $(1 < k < 63)$. Це, зокрема, дозволяє шифрувати текстові дані посимвольно (для коду EBCDIC достатньо узяти, $k = 8$, а для коду ASCII – $k = 7$).

Режим CFB (Зворотний зв'язок за криптотекстом). Схема алгоритму в режимі CFB надана на рисунку 5.6.

В цьому режимі базовий алгоритм E служить для породження блоків псевдовипадкових бітів B_i завдовжки 64 бита: $B_i = E_K(I_i)$, $i = 0, \ldots, l$, де I_0 – довільний ініціюючий вектор, а I_{i+1} виходить з I_i відкиданням (методом зсуву) перших k бітів і приписуванням (заповненням) справа до блоку перших (лівих) k біт блоку B_{i-1}. Блок зашифрованого повідомлення передається у вигляді $C_i^{(k)} = m_i^{(k)} \oplus B_i^{(k)}$. Тут $C_i^{(k)}$ – k бітів тексту на i - тому кроці шифрування; $m_i^{(k)}$ – відповідні k бітів відкритого тексту; $B_i^{(k)}$ – ліві k біт результату шифрування на i - тому кроці; $\oplus$ – символ побітового додавання за модулем 2.

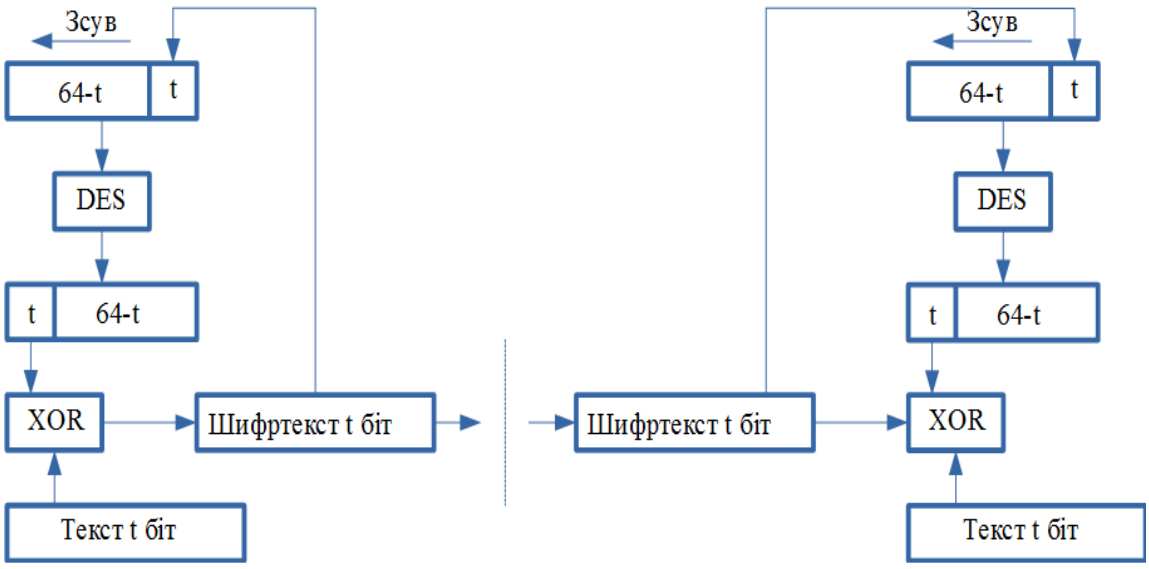

Рисунок 5.6 – Робота алгоритму DES в режимі CFB

Зворотний зв'язок за виходом (OFB). Схема алгоритму в режимі ECB надана на рисунку 5.4.

Цей режим схожий на попередній, тільки для оновлення I_i використовується C_{i-1} замість B_{i-1}. Як видно, обидва розглянуті режими використовування DES із зворотним зв'язком, як і режим CBC, вільні від головного недоліку ECB: ідентичним блокам відкритого тексту в загальному

випадку відповідають різні блоки криптотекста (що стосується OFB, то дане твердження вірне по відношенню до будь-яких двох ідентичних блоків, розділених у відкритому тексті блоками, кількість яких не кратна періоду повторення, зменшеному на одиницю). Схема алгоритму в режимі OFB надана на рисунку 5.7.

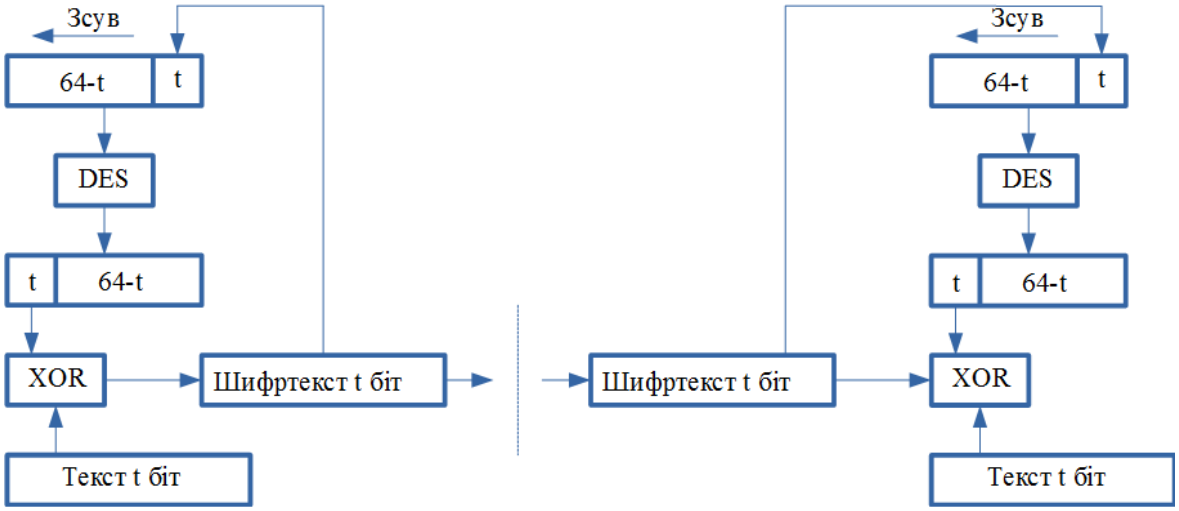

Рис. 5.7 – Блок-схема алгоритму DES в режимі OFB

Істотною гідністю OFB є відсутність явища розмноження спотворень, що мали місце при передачі криптотекста в процесі розшифрування останнього.

В той же час спотворення одного біта k-бітного блоку криптотекста, отриманого в режимі CFB, впливатиме на k-бітні блоки відкритого тексту до тих пір, поки спотворений біт не виштовхне за межі вхідного 64-бітового блоку в результаті послідовності зсувів його вмісту, виконуваних в процесі розшифрування.

Такі основні особливості алгоритму шифрування DES і різних режимів його використовування. Стисло зупинимося на питаннях застосування вказаних режимів для забезпечення безпеки інформації, переданої і береженої в ІВС.

Звичайно розглядаються три області застосування DES: передача даних по каналах зв'язку (ІВС), зберігання і доступ до файлів (базам даних) ІВС, а також організація системи платежів і обміну комерційною інформацією.

Вважається, що для забезпечення безпеки каналів зв'язку ІВС можуть застосовуватися всі описані вище режими, хоча в більшості апаратних реалізацій (платні криптографічного закриття) використовується режим CFB з $k = 1$, забезпечуючий побітове шифрування. Це пояснюється його більшою криптостійкістю у порівнянні з OFB, особливо в плані криптографічної підтримки біт- орієнтованих протоколів, характерних для ІВС, функціонуючих на основі високошвидкісних широкосмугових каналів зв'язку. Відносно недорога платня, реалізовуючи CFB з $k = 1$, забезпечує швидкість шифрування-розшифрування, достатню для передачі даних з швидкістю до 19,2 Кбіт/с.

В додатки, пов'язаних із зберіганням інформації в зовнішній пам'яті ЕОМ, розрізняють три основні напрями застосування DES: криптографічне закриття файлів прямого і послідовного доступу, а також окремих полів запису таких файлів.

Контрольні питання та задачі

1. Поняття про гамування.
2. Розсіювання та перемішування.
3. Режими роботи алгоритму DES.
4. Основні операції процедури шифрування E.
5. Циклова функція DES.
6. Особливості кожного з режимів DES.

Лекція 6

Основні типи перетворень, що використовуються в перспективних симетричних криптосистемах. Блоковий симетричний шифр AES

Розглянемо типові криптографічні та табличні перетворення на прикладі перспективного симетричного стандарту. В 2000 році закінчено міжнародний проект створення перспективного блокового симетричного шифру AES (Advanced Encryption Standard), за результатами якого прийнято стандарт блокового симетричного шифрування США FIPS-197. При розробці перспективного стандарту БСШ до нього було висунуто такі вимоги:

1. Блоковий симетричний шифр розробляється відкрито, тобто стійкість має забезпечуватися при умові, що невідомий тільки ключ.
2. Довжина блоку $l_б$ має бути 128, 192 та 256 бітів.
3. Довжина ключа $l_к$ має бути 128,192 та 256 бітів.
4. Алгоритм повинен обов'язково працювати в таких режимах $l_б / l_к$: 128/128, 128/192, 128/256.
5. Алгоритм має застосовуватись в таких режимах:

– блокового шифрування;

– потокового шифрування;

– потокового шифрування зі зворотним зв'язком;

– виробки імітоприкладки;

– формування псевдовипадкових послідовностей.

6. Реальна криптостійкість для алгоритму: не має існувати жодної атаки, складність якої менша, ніж складність атаки ″грубої сили″.

7. Статистична безпечність.

8. Надійність математичної бази (у розумінні, що розробник не зможе закласти математичну лазівку в шифр).

9. Можливість реалізувати алгоритм програмно, апаратно і апаратно-програмно при практично однаковій складності.

10. Можливість реалізувати алгоритм на процесорах різної розрядності (8,16,32 і т. д. бітів).

11. В системі має бути реалізований принцип розгортання ключів.

Всім цім вимогам відповідає симетричний шифр RIJNDAEL, який став стандартом США.

В даній криптосистемі реалізовано принцип розгортання ключів. При цьому множина ключів, що називаються цикловими, є деякою функцією від початкового ключа:

$$K_{ц} = \psi(K_{П}). \tag{6.1}$$

Довжина кожного із циклових ключів завжди дорівнює довжині блоку повідомлення. Сутність схеми розгортання ключів – кількість циклових ключів на один більше кількості циклів $n_{ц}$, тому довжину розгорнутого ключа $l_{РК}$ можна визначити як:

$$l_{РК} = (n_{ц} + 1) \cdot l_{б}, \tag{6.2}$$

де $l_{б}$ - довжина зашифрованого блоку.

Початковий ключ $K_{П}$ має формуватися випадково, рівноймовірно та незалежно. Перший режим можна надати наступним чином. Джерело повідомлень та ключів задається станом:

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

Якщо довжина блоку 128 біт, то стан джерела задається матрицею 4 на 4 байтів, де a_{ij} – байт повідомлення. Якщо довжина блоку 192 біта, то стан джерела задається матрицею 6 на 4 (6 стовпців і 4 рядки), де a_{ij} – байт

повідомлення. Якщо довжина блоку 256 біт, то стан джерела задається матрицею 8 на 4 (8 стовпців і 4 рядки), a_{ij} – байт повідомлення.

На рисунку вище зображено матриця джерела повідомлень для блоку 192 біта та матриця джерела ключів для довжини ключа 128 бітів. При перетворенні A_{ij} блоку з матричного вигляду в бітову послідовність інформація зчитується по стовпцях:

$$A_{ij} = a_{00}a_{10}a_{20}a_{30}a_{01}...a_{33}...$$

На рис. 6.1 наведено блоковий криптоалгоритм зашифрування.

Основний цикл перетворення містить два види перетворень – табличні та криптографічні.

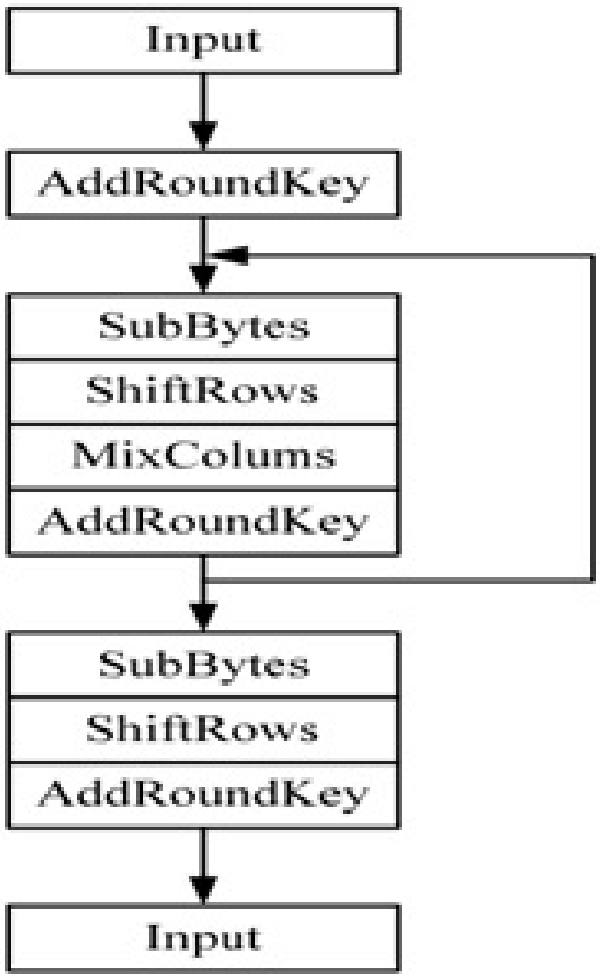

Рис. 6.1 – Алгоритм зашифрування AES

Сутність табличних перетворень

Bitesub (заміна байтів) – це два табличних перетворення, здійснюються над байтами послідовно по стовпцях.

1. Перше перетворення здійснюється засобом заміни a_{ij} байта на a_{ij}^{-1} обернений, що досягається розв'язком порівняння:

$$a_{ij} \cdot a_{ij}^{-1} \equiv 1(\mathrm{mod}(x^8 + x^4 + x^3 + x + 1)). \qquad (6.3)$$

2. Друге табличне перетворення кожного байта здійснюється як афінне:

$$Y = (C \cdot x + C_1) \bmod (x^8 + 1), \qquad (6.4)$$

де C, C_1 - константи, що мають вигляд:

$$C = \begin{vmatrix} 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \\ 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \end{vmatrix}, \quad C_1 = \begin{vmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{vmatrix},$$

$x = (x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7)$ - біти a_{ij}^{-1} байта.

В сукупності (6.3) та (6.4) задають деяку нелінійну підстановку типу байт в байт. Тому реально в алгоритмі перетворення (6.3) та (6.4) не виконуються, а задаються фіксованою таблицею підстановки.

Shiftrow (зсув рядків) – здійснює відносний зсув байтів в рядках. В залежності від кількості стовбців (n) зсув здійснюється згідно з таблицею 6.1.

Табиця 6.1 – таблиця зсуву

n_{ij}	N_2	N_3	N_4
4	1	2	3
6	1	2	3
8	1	3	4

В алгоритме RD число циклів n перетворення залежить від довжини інформаційного блоку l_m тв довжини початкового ключа l_k В таблиці 6.2 наведено значення n_c циклів перетворення як функція l_u і l_k.

Таблиця 6.2– кількість циклів перетворення

Nr	*Nb* = 4	*Nb* = 6	*Nb* = 8
Nk = 4	10	12	14
Nk = 6	12	12	14
Nk = 8	14	14	14

Mixcjlumn (перемішування в стовпцях) – кожний стовпець можна подати у вигляді полінома з коефіцієнтами:

$$a_{ij} = a_{3j}x^3 + a_{2j}x^2 + a_{1j}x + a_{0j}, \qquad (6.5)$$

де $a_{3j}; a_{2j}; a_{1j}; a_{0j}$ – елементи (байти) стовпця.

Сутність Mixcjlumn полягає в помноженні кожного стовпця a_{ij} на константу $C(x)$:

$$a'(x) = C(x) \cdot a(x) \ (\mathrm{mod}(x^4 + 1)), \qquad (6.6)$$

де $a(x)$ використовується згідно з (6.5);

$C(x)$ - константа, що задається поліномом :

$$C(x) = "03"x^3 + "01"x^2 + "01"x + "02"$$

Коефіцієнти константи є байтами, при цьому, наприклад, "03" = 00000011.

При табличному перетворенні Mixcjlumn кожний стовпчик подається у вигляді поліному не вище третього степеня з коефіцієнтами над полем $GF(2^8)$.

Константа $C(x)$ вибрана з умови найкращого перемішування в стовпцях. Коефіцієнти полінома $a(x)$ складаються з чотирьох байтів, які є коефіцієнтами (6.5). Результат перемноження (6.6) не може бути вище третього степеня.

Зашифрування стану A_{ij} здійснюється за правилом:

$$A'_{ij} = A_{ij} \oplus K_{ij},$$

де K_{ij} представляється станом ключа (6.1). Довжина ключа має дорівнювати довжині блоку перетворення.

Алгоритм блокового розшифрування

Криптоалгоритм, що реалізований у $FIPS-197$, не є інволютивним. Інволютивними називаються шифри, у яких алгоритм прямого і зворотного перетворення є одними і тими ж.

У алгоритмі AES розшифрування здійснюється у зворотному порядку. По відношенню до алгоритму зашифрування можна говорити, що алгоритм розшифрування здійснюється в зворотному напрямку (знизу вверх).

Розглянемо задачу

Знайти афінне перетворення виду $a'_{ij} = C \cdot a_{ij}^{-1} + C_1$, де C та C_1 - константи, що мають вигляд:

$$C = \begin{vmatrix} 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \\ 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \end{vmatrix} \qquad C_1 = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}$$

При цьому $a_{ij} \cdot a_{ij}^{-1} \equiv 1(\mathrm{mod}(x^8 + x^4 + x^3 + x + 1))$, якщо $a_{ij} = 71$. Знайти відстань Хемінга між вхідними та вихідними елементами.

Розв′язок.

Знайдемо a_{ij}^{-1}. Для цього зведемо розв’язок порівняння $a_{ij} \cdot a_{ij}^{-1} \equiv 1(\mathrm{mod}(x^8 + x^4 + x^3 + x + 1))$ до розв’язку порівняння виду $E_k \cdot D_k \equiv 1(\mathrm{mod}\,\varphi(N))$, яке в свою чергу можна звести до розв’язку Діофантового рівняння виду $ax + by = 1$, тобто виділити

$$(-k)\cdot\varphi(N)+a_{ij}\cdot a_{ij}^{-1}=1,$$

де $b=a_{ij}=71$, $a=\varphi(N)=x^8+x^4+x^3+x+1$. Треба знайти $y=a_{ij}^{-1}$ та $x=(-k)$.

Діофантове рівняння має розв'язок, якщо $\varphi(N)\geq E_k$ та $((\varphi(N),E_k)=1$.

Подамо цей розв'язок у вигляді ланцюгового дробу.

Для цього запишемо $a_{ij}=71$ у вигляді поліному:

$a_{ij}=01000111=x^6+x^2+x+1$.

Тоді наш ланцюговий дріб матиме вигляд:

$$\frac{x^8+x^4+x^3+x+1}{x^6+x^2+x+1}=x^2+\frac{x^2+x+1}{x^6+x^2+x+1};$$

$$\frac{x^6+x^2+x+1}{x^2+x+1}=(x^4+x^3+x)+\frac{1}{x^2+x+1};$$

$$a_0=r_0=x^2;\quad b_0=1;$$
$$a_1=r_0r_1+1=x^2(x^4+x^3+x)+1=x^6+x^5+x^3+1;$$
$$b_1=r_1=x^4+x^3+x.$$

Тоді, якщо $\mu-$порядок ланцюгового дробу, а $a,b-$його параметри, то

$$\left\{\begin{array}{l} y=(-1)^{\mu}a_{\mu-1}=(-1)^2\cdot a_1=x^6+x^5+x^3+1 \\ x=(-1)^{\mu}b_{\mu-1}=(-1)^2\cdot b_1=x^4+x^3+x. \end{array}\right\}$$

Отже, $a_{ij}^{-1}=y=x^6+x^5+x^3+1=01101001_2=105_{10}$.

Перевірку правильності розв'язку рівняння виконуємо, підставивши значення a_{ij} та a_{ij}^{-1} в $a_{ij}\cdot a_{ij}^{-1}\equiv 1(\mathrm{mod}(x^8+x^4+x^3+x+1))$.

Маємо

$$(x^6+x^2+x+1)(x^6+x^5+x^3+1)=1(\mathrm{mod}(x^8+x^4+x^3+x+1)).$$

Дійсно

$(x^6+x^2+x+1)(x^6+x^5+x^3+1)(mod(x^8+x^4+x^3+x+1)=x^{12}+x^{11}+x^9+x^6++x^8+x^7+x^5+x^2+x^7+x^6+x^4+x+x^6+x^5+x^3++1(mod(x^8+x^4+x^3+x+1)).$

Знайдемо залишок:

$$
\begin{array}{l|l}
x^{12}+x^{11}+x^9+x^8+x^2+x^4+x+x^6+x^3+1 & x^8+x^4+x^3+x+1 \\
\hline
x^{12}+x^8+x^7+x^5+x^4 & x^4+x^3+x \\
\hline
x^{11}+x^9+x^7+x^6+x^5+x^3+x^2+x+1 & \\
x^{11}+x^7+x^6+x^4+x^3 & \\
\hline
x^9+x^5+x^4+x^2+x+1 & \\
x^9+x^5+x^4+x^2+x & \\
\hline
\end{array}
$$

1

Таким чином, лишок =1. Елементи x^6+x^2+x+1 та $x^6+x^5+x^3+1$ є зворотними.

Знайдемо афінне перетворення $a_{ij}^{'}=C\cdot a_{ij}^{-1}+C_1$. Для цього запишемо $a_{ij}^{'}=01101001$ у вигляді матриці-стовпця:

$$a_{ij}^{-1}=\begin{vmatrix}1\\0\\0\\1\\0\\1\\1\\0\end{vmatrix}.$$

Позначимо $C*=C\cdot a_{ij}^{-1}$:

$$C* = \begin{vmatrix} 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \\ 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \end{vmatrix} \cdot \begin{vmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 0 \end{vmatrix} = \begin{vmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{vmatrix};$$

Тепер знайдемо $a_{ij}^{'} = C* + C_1$:

$$a_{ij}^{'} = \begin{vmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \end{vmatrix} \oplus \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix} = \begin{vmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{vmatrix}$$

Отже, $a_{ij}^{-1} = 11111010_{\,2} = 250_{10}$.

Контрольні питання та задачі

1. Вимоги до сучасних блокових симетричних шифрів.
2. Поясніть алгоритм звшифрування Rijndael.
3. Принцип розгортання ключів.
4. Режими роботи алгоритму Rijndael.
5. Перетворення Bitesub (заміна байтів).
6. Перетворення Shiftrow (зсув рядків).
7. Перетворення Mixcjlumn (перемішування в стовпцях).
8. Поняття про інволютивний шифр.
9. Знайти афінне перетворення $a_{ij}^{'} = C \cdot a_{ij}^{-1} + C_1$, де C та C_1 - константи, що мають вигляд:

$$C = \begin{vmatrix} 11111000 \\ 01111100 \\ 00111110 \\ 00011111 \\ 10001111 \\ 11000111 \\ 11100011 \\ 11110001 \end{vmatrix}, \qquad C_1 = \begin{vmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{vmatrix}.$$

При цьому $a_{ij} = a_{ij}^{-1} \equiv 1(\mathrm{mod}(x^8 + x^4 + x^3 + x + 1))$, якщо $a_{ij} = 99 + 3k$, де k - номер з журналу. Знайдіть відстань Хемінга між вхідними та вихідними елементами.

Розділ 2 Класичні криптосистеми з відкритим ключем та їх використання

Лекція 7

Вступ в теорію асиметричних криптоперетворень. Концепція криптосистем з відкритим ключем

1976 рік відкрив новий (сучасний) етап у криптографії. Характерною рисою цього етапу є поява принципова нових криптографічних завдань, а також принципово нових рішень завдань класичних.

Тому часто говорять про революцію в області криптографії. Зрушення, що відбулися в 1976 році, зв'язується з іменами американських математиків Вайтфілда Діфі, Мартіна Гелмана й Ральфа Меркле, які розвили ідеологію відкритого ключа.

Кардинальна відмінність криптосистеми з відкритим ключем від системи симетричної полягає в тому, що в криптосистемах з відкритим ключем процедура шифрування стає загальнодоступною. Це, однак, не означає як у традиційних системах шифрування, що загальнодоступною є й процедура розшифрування.

Поняття ключа розбивається на дві частини (включає тепер два поняття): ключ відкритий, і ключ таємний. Загальнодоступний відкритий ключ використовується для шифрування, але розшифрування може здійснити тільки той, хто володіє таємним ключем.

Повернемося до нашої базової криптографічної схеми, розглянутої на першій лекції. Вона припускає поняття ключа, що використовується як при шифруванні, так і при розшифруванні.

У всіх криптосистемах, які ми розглядали дотепер, ключ шифрування або збігався із ключем розшифрування або діставався із ключа шифрування досить просто.

Раніше ми (як і все криптографічне співтовариство до 1976 року) і не замислювалися про те, що таке положення речей можна змінити, і навіть із якою-небудь користю.

А саме в допущенні того, що знаходження ключа розшифрування по відомому ключу шифрування може бути важким завданням. В цьому і полягає ідея, що визначила подальший напрямок розвитку криптографії.

Поняття криптосистеми з відкритим ключем містить у собі такі об'єкти (див. рис. 7.1).

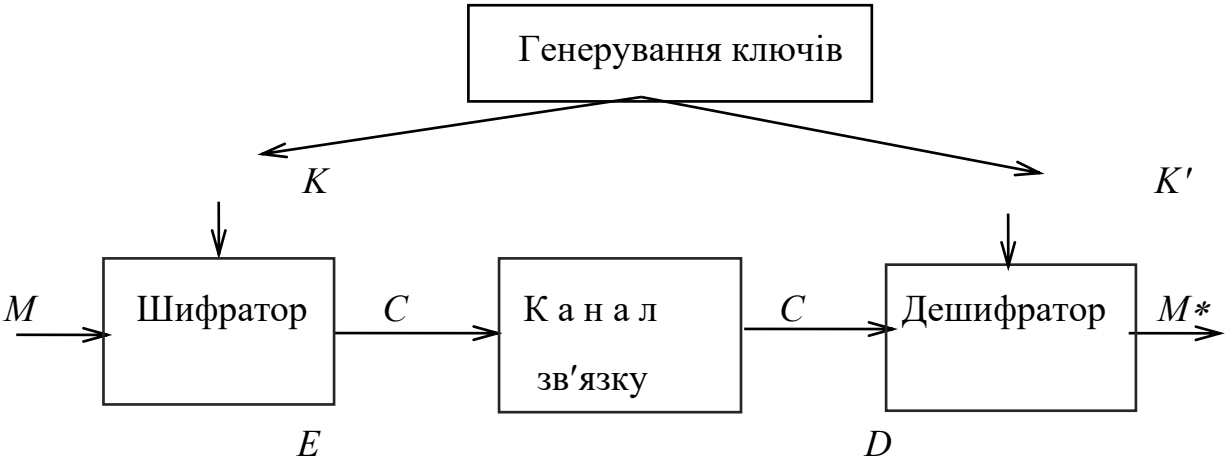

Рис.7.1 – Нова криптографічна схема захищеної передачі інформації

– Алфавіт A, у якому записуються повідомлення (відкриті тексти), і алфавіт B, у якому записуються криптотексти.

– Простір ключів K (безліч слів у деякому алфавіті).

– Алгоритм генерування ключів. Це поліноміальний (ефективний) імовірнісний алгоритм, що видає (дозволяє сформувати) випадкову пару $K, K' \in \boldsymbol{K}$. Компонента K називається відкритим ключем і використовується для шифрування, а компонента K' називається таємним ключем і використовується для розшифрування.

– Поліноміальний (ефективний) детермінований алгоритм шифрування E, що одержує на вхід повідомлення M й відкритий ключ K, а видає криптотекст C, що записується як $C = E_k(M)$.

– Поліноміальний (ефективний) детермінований алгоритм розшифрування D, що одержує на вхід криптотекст C і секретний ключ K', а видає відкритий текст M, що записуємо як $M = D_{K'}(C)$.

Перераховані алгоритми задовольняють таким умовам:

– Якщо пара (K, K') породжена алгоритмом генерування ключів, то із $C = E_k(M)$ можна отримати $M = D_{K'}(C)$ для будь-якого відкритого тексту M.

– Немає (або, принаймні, невідомо) хоча б одного ефективного алгоритму, який би по відомим $C = E_k(M)$ і K знаходив би M.

Остання умова забезпечує надійність криптосистеми навіть тоді, коли з відкритого ключа K не робиться секрету. Із цієї умови витікає, зокрема, що пари (K, K') могли б були породжені алгоритмом генерування ключів.

Помітимо, що коли відкритий ключ є доступним для зловмисника, останній має практично ті ж можливості криптоаналіза, які для традиційних криптосистем називалися атакою з обраним відкритим текстом.

Як і раніше, найсильнішим видом криптоаналіза залишається атака з обраним криптотекстом.

Криптосистеми з відкритим ключем ще називають й асиметричними. У цьому контексті класичні криптосистеми називають симетричними.

Криптосистема шифрування RSA

Ця система запропонована в 1977 році, і є однією з найвідоміших криптосистем з відкритим ключем. Назва системи утворена з перших букв імен її творців (Рональда Райвеста, Ади Шаміра й Леонарда Адлемана).

Генерування ключів.

Вибирають два досить великих простих числа p і q. Для їхнього добутку $n = pq$ функція Ойлера дорівнює $\varphi(n) = (p-1)(q-1) = n - p - q + 1$ (у теорії чисел використовується поняття функції Ойлера $\varphi(m)$, під якою розуміється число чисел менш ніж m і взаємно простих з m).

Потім випадковим чином обирають число *e*, що не перевищує значення $\varphi(n)$ й взаємнопросто з ним.

Для цього з числа *e* за допомогою розширеного алгоритму Евкліда знаходять елемент *d*, зворотний до *e*, тобто такий, що $d < \varphi(n)$ й

$$ed \equiv 1(\mathrm{mod}\,\varphi(n)) \qquad (7.1)$$

Цей запис у теорії чисел позначає що добуток ed при діленні на число $\varphi(n)$ дає залишок рівний 1 (читається ed порівняно з одиницею за модулем $\varphi(n)$).

У результаті полагають:

Як відкритий ключ пара чисел e й n.

В якості таємного ключа – число d.

Також числа p і q неможна відкривати.

Шифрування здійснюється блоками.

Для цього повідомлення записують у цифровому вигляді й розбивають на блоки так, що кожен блок представляє число, що не перевищує n.

Скажемо, якщо блок *M* записаний у двійковій формі довжини *m*, то повинне виконуватися умова $2^m < n$.

Алгоритм шифрування *E* в системі RSA складається у зведенні двійкового числа *M* у степінь e. Запишемо це так

$$E(M) = M^e \,(\mathrm{mod}\, n)$$

У результаті виходить блок криптотексту $C = E(M)$.

Розшифрування.

Алгоритм розшифрування D блоку криптотексту C складається у зведенні числа C в степінь d, тобто

$$D(C) = C^d \,(\mathrm{mod}\, n).$$

Приклад

Нехай $p = 53$ й $q = 67$. Тоді $n = 3551$ й $\varphi(n) = 3432$. Візьмемо $e = 1021$ – можна перевірити, що *НСД*(1021,3432)=1. Одночасно обчислюємо $d = 1021^{-1} (\text{mod } 3432) = 1237$. Ключі знайдені.

Відкритий ключ $e = 1021$ й $n = 3551$ опубліковуємо. Припустимо, що один з ділових партнерів вирішив послати нам вказівку НОВИНИ.

Спочатку він перетворить своє повідомлення до цифрового виду, замінивши кожну букву її цифровим десятковим номером в алфавіті 17 18 02 10 17 10 (тут нумерація букв алфавіту починається з нуля).

Видно, що для нашого модуля $n = 3551$ цифрове повідомлення доцільно розбивати на блоки по 4 цифри, тобто 1718 0210 1710.

При шифруванні перший блок перетвориться в число $1718^{1021}((\text{mod } 3531) = 139$. У такий же спосіб шифруються й інші блоки, і в результаті виходить шифртекст 0139 ... .

Розшифрування цієї шифрограми виконується зведенням кожного блоку в степінь $d = 1237$ за модулем $n = 3551$. Легко перевірити, що $139^{1237} (\text{mod } 3551) = 1718$ й т.д.

Коректність

Варто переконатися, що $D\big(E(M)\big) = M$ для будь-якого повідомлення *M*. Сформулюємо це у вигляді твердження.

Твердження 7.1 Нехай $n = pq$ є добутком двох різних простих чисел. Якщо $ed \equiv 1\big(\text{mod } \varphi(n)\big)$, то для всіх $x \in Z_n$ справедливо порівняння

$$x^{ed} \equiv x\big(\text{mod } n\big). \qquad (7.2)$$

Цей результат безпосередньо випливає з відповідної теореми Ойлера, що стверджує, що для будь-якого натурального числа $n > 1$ і числа x, взаємно простого з n, тобто $(a, n) = 1$, справедливе порівняння

$$x^{\varphi(n)} \equiv 1(\text{mod } n).$$

Дійсно, у цьому випадку з (7.1) випливає рівність $ed = k\varphi(n) + 1, \quad k \in N$

($N -$ простір натуральних чисел), з якого у свою чергу випливає справедливість (7.2).

Ефективність

Алгоритм генерування ключів використовує процедуру породження простих чисел і розширений алгоритм Евкліда для обчислення $НСД\big(e, \varphi(n)\big)$ й $d = e^{-1} \bmod \varphi(n)$. В алгоритмах шифрування й розшифрування зведення в степінь виконується за допомогою бінарного методу.

Надійність

Щоб криптосистема вважалася надійною, необхідно, щоб завдання визначення повідомлення по криптотексту й відкритого ключа були важкими. У нашому випадку останнє завдання можна сформулювати так.

РОЗКРИТТЯ RSA

Задано*:* e, n, y, де $n = pq$ й $НСД\big(e, \varphi(n)\big) = 1$.

Знайти*:* x таке, що $x^e \equiv y (\bmod n)$.

Як бачимо, завдання розкриття RSA є дослівно завданням добування із заданого цілого числа кореня $e-$того степеня за модулем $n = pq$ (при перерахованих умовах).

На сьогоднішній день для цього завдання невідомо ніякого ефективного алгоритму. Однак не доведено, що такого алгоритму не існує. Нажаль, така ситуація характерна для всіх криптосистем з відкритим ключем, які мають практичний інтерес.

Єдине, що ми можемо зробити, це розглянути деякі специфічні методи криптоаналізу для RSA й оцінити їхню перспективність.

Будь-яку асиметричну криптосистему можна зламати, якщо вказати ефективний спосіб визначення секретного ключа по відкритому. У розглянутому випадку це означає, що розкриття RSA зводиться до завдання

ЗНАХОДЖЕННЯ СЕКРЕТНОГО КЛЮЧА ДЛЯ RSA

Задано: e, n, де $n = pq$ й $НОД\big(e, \varphi(n)\big) = 1$.

Знайти: d таке, що $ed \equiv 1(\mathrm{mod}\,\varphi(n))$.

З алгоритму генерування ключів безпосередньо випливає, що завдання зводиться до обчислення значення функції Ойлера $\varphi(n)$. Обчислення функції Ойлера від аргументів такого типу є завданням еквівалентним знаходженню співмножників p і q числа n. Таким чином, спроба факторизувати модуль $n = pq$ є самим природнім шляхом розкриття алгоритму RSA. На щастя на сьогодні це завдання є безнадійним для n порядку 10^{200}.

Тому при генеруванні ключа, p і q рекомендується вибирати із приблизно сотнею десяткових цифр кожне. Вибір таких чисел повинен бути дійсно випадковим, щоб уникнути можливої факторизації яким-небудь вузько спеціалізованим методом.

Звідси такі високі вимоги до якості генератора псевдовипадкових бітів, що використовується алгоритмом породження простих чисел.

Природно виникає питання, чи не можна вирішити завдання визначення секретного ключа, обходячись без факторизації. Це завдання можна переформулювати як завдання знаходження такого d, що число $ed - 1$ ділиться на $\varphi(n)$. Але для такого d число $m = ed - 1$ можна використати для розкладання числа n на множники за допомогою імовірнісної процедури.

Таким чином, знаходження секретного ключа для RSA є таким же важким, як факторизація модуля n.

Підбиваючи підсумок нашим аргументам, можемо сформулювати

Твердження 7.2 Знаходження секретного ключа для RSA є завданням, еквівалентної факторизації чисел, що є добутком двох простих, яка на сьогоднішній день є обчислювально складною задачею.

Помітимо, що для деяких повідомлень M має місце рівність $E(M) = M$. Числові еквіваленти таких повідомлень є рішенням порівняння $x^e = x(\mathrm{mod}\,n)$ в Z_n (безліч відрахувань $\{0, 1, \ldots, n-1\}$), число яких нескладно підрахувати.

Можна назвати й ще кілька каналів уразливості системи RSA.

Алгоритм вважається поліноміальним тоді й тільки тоді, коли час його роботи для числа на вході n (довжиною $\lfloor \log_k n \rfloor + 1$) обмежено функцією $c(\log_k n)^d$ для деяких констант $c > 0$ й $d > 1$.

Алгоритм вважається експоненціальним, якщо для входу n (точніше, для нескінченної послідовності таких входів) час його роботи перевищує cn^d для деяких констант $c, d > 0$.

Криптосистема Ель- Гамала

Генерування ключів .Вибирають велике просте число p, а також число g $1 < g < p - 1$, що має в мультиплікативній групі $Z_p^\bullet$ великий порядок.

В ідеальному випадку g є первісним коренем за модулем p. Числа p і g не є секретними й перебувають у загальному користуванні. Кожен абонент вибирає собі випадкове число a в проміжку від 1 до $p - 1$, і обчислює

$$h = g^a \bmod p.$$

Відкритий ключ: p, g, h.

Секретний ключ: a.

Шифрування здійснюється блоками. Кожен блок M вважається елементом з $Z_p^\bullet$ (мультиплікативна група елементів, для яких у Z_n є зворотні щодо множення елементи (є мультиплікативні інверсії)).

Повідомлення $M \in Z_p^\bullet$ перетвориться в криптотекст $C \in \left(Z_p^\bullet\right)^2$ у такий спосіб.

– Вибирається випадкове число r таке, що $1 \le r \le p - 1$.

– Обчислюють $C = (c_1, c_2)$, де

$$c_1 = g^r \bmod p,\ c_2 = Mh^r \bmod p.$$

Розшифрування Маючи секретний ключ a і криптотекст $C = (c_1, c_2)$, обчислюють

$$D(C) = c_2 \cdot \left(c_1^a\right)^{-1} \bmod p$$

Приклад

Як і у всіх попередніх випадках, ми жертвуємо реалізмом для простоти обчислень. Тобто числа у прикладах достатньо малі.

Нехай $p = 23, g = 5, a = 6$. Обчислюємо $h = 5^6 \bmod 23 = 8$. Відкритий і секретний ключі сформовані.

Припустимо, що шифрується числова інформація й необхідно зашифрувати повідомлення $M = 7$. Нехай обрано $r = 10$. Тоді $c_1 = 5^{10} \bmod 23 = 9$ й $c_2 = \left(7 \cdot 8^{10}\right) \bmod 23 = 21$.

Одержуємо криптотекст $C = (9, 21)$. Легко перевірити, що при розшифруванні цього криптотекста дійсно $D(9,21) = 21 \cdot \left(9^6\right)^{-1} \bmod 23 = 7$.

Коректність. Перевірка рівності $D(C) = M$ для криптотекста C, отриманого з повідомлення M за допомогою алгоритму шифрування з довільним r, перевіряється безпосередньо

$$D(C) = c_2 \cdot \left(c_1^a\right)^{-1} (\text{mod } p) = Mh^r \cdot \left(g^{ra}\right)^{-1} (\text{mod } p) = M\left(g^a\right)^r \left(g^{ra}\right)^{-1} (\text{mod } p) = M$$

.

Ідея криптосистеми досить прозора: повідомлення M маскується, здобуваючи вид c_2, і разом з тим посилається підказка c_1, що дозволяє відтворити M з c_2.

Ефективність

Зведення в степінь у $Z_p^{\bullet}$ за модулем виконується за допомогою бінарного методу. Алгоритм вибору великого простого p вважається вже добре освоєним. Однак наше завдання складніше – необхідно вибрати також число g.

У найкращому разі необхідно мати в якості g первісний корінь за модулем p. На жаль, це завдання не має простого обчислювального рішення. Тому варто

ставити завдання генерування пари p, g, для рішення якої існують ефективні процедури.

Надійність

Розкриття системи Ель Гамала.

Задано: p, g, h, c_1, c_2, де $1 < g < p - 1$,

$h = g^a \bmod p, \quad c_1 = g^r \bmod p, \quad c_2 = Mh^r \bmod p$,

для деяких $a, r, M \in Z_p^{\bullet}$.

Обчислити: M.

Легко показати, що це завдання еквівалентне наступному

Задано: $p, g, x, y \in N$, де $1 < g < p - 1$, $x = g^a \bmod p$,

$y = g^b \bmod p$, для деяких a й b,

Обчислити: $z = g^{ab} \bmod p$.

Друге завдання, як відомо, зводиться до дискретного логарифмування за модулем p. Тому варто виключити ті випадки, при яких логарифмування можна провести ефективно. Зокрема, p варто вибирати таке, щоб число $p - 1$ мало великий простий співмножник, інакше зловмисник може скористатися алгоритмом Сільвера-Поліга-Геллмана.

Протолкол Діффі- Геллмана

Протокол Діффі-Геллмана був першим з алгоритмів роботи з відкритими ключами (1976 г.). Безпека даного протоколу заснована на труднощі обчислення дискретних логарифмів в кінцевому полі.

Протокол Діффі-Геллмана (англ. Diffie–Hellman key exchange (D–H)) — це метод обміну криптографічними ключами.

Один з перших практичних прикладів узгодження ключа, що дозволяє двом учасникам, що не мають жодних попередніх даних один про одного, отримати спільний секретний ключ з використанням незахищеного каналу зв'язку.

Цей ключ можна використати для шифрування наступних сеансів зв'язку, що використовують шифр з симетричним ключем.

Схему вперше оприлюднили Вітфілд Діффі і Мартін Геллман у 1976, хоча пізніше стверджувалось, що її кількома роками раніше винайшов Малколм Вільямсон у GCHQ, британській розвідувальній агенції.

Хоча протокол Діффі-Геллмана є анонімним (без автентифікації) протоколом встановлення ключа, він забезпечує базу для різноманітних протоколів з автентифікацією, і використовується для забезпечення цілковитої прямої секретності в недовговічних режимах Transport Layer Security (відомих як EDH або DHE залежно від комплектації шифру).

Припустимо обом абонентам відомі деякі два числа g та p, які не є секретними та можуть бути розповсюджені.

Для того, щоб побудувати невідомий більш нікому секретний ключ, обидва абоненти генерують великі випадкові числа: перший абонент — число a, другий абонент — число b.

Далі перший абонент обчислює значення $A = g^a \bmod p$ та надсилає його другому абоненту, а другий абонент обчислює $B = g^b \bmod p$ та передає першому.

Передбачується, що зловмисник може отримати обидва ці значення, та не модифікувати їх (у нього немає можливості втручання в процес передачі).

На другому етапі перший абонент на основі a (яке у нього є) та отриманого з мережі B обчислює значення $B^a \bmod p = g^{ab} \bmod p$, а другий абонент на основі b (яке у нього є) та отриманого з мережі A обчислює значення $A^b \bmod p = g^{a\,b} \bmod p$.

Як можна бачити, у обох абонентів побудовано одне и те ж число: $K = g^{ab} \bmod p$. Його вони можуть використовувати у якості секретного ключ. Тут зловмисник зустрічається з проблемою обчислення $g^{ab} \bmod p$ з перехоплених $g^a \bmod p$ и $g^b \bmod p$ (за реальний час). Числа p, a, b обирають достатньо великими.

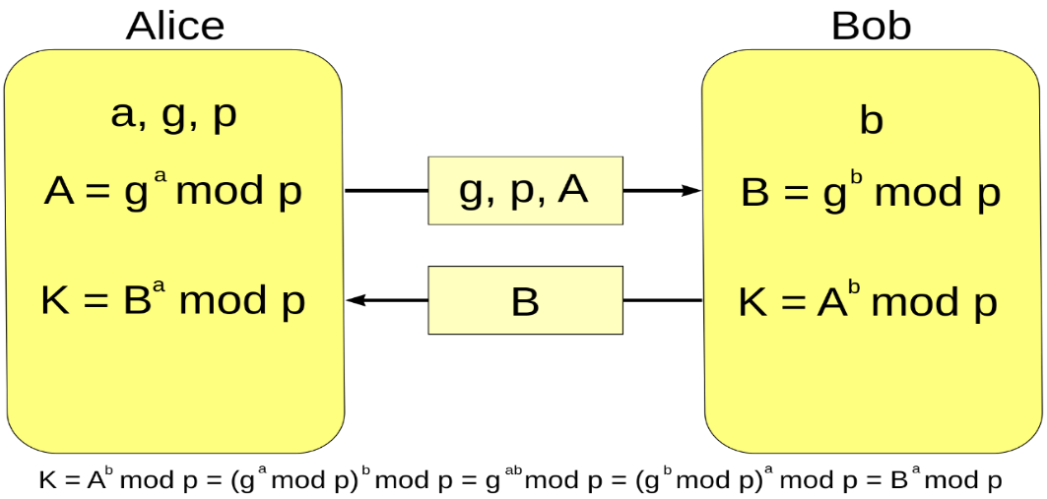

Рис 7.1 – Алгоритм Діффі-Геллмана, де K — загальний секретний ключ

При роботі алгоритму, кожна сторона:

1. Генерує випадкове натуральне число *a* — закритий ключ.
2. Сумісно з віддаленою стороною установлює відкриті параметри *p* та *g* (звичайно значення *p* та *g* генеруються на одній стороні та передаються другій), де

p є випадковим простим числом

g є первісним коренем за модулем *p*.

3. Обчислюють відкритий ключ *A*, використовує перетворення закритого ключа

$A = g^a \mod p$.

4. Обмінюються відкритими ключами з віддаленою стороною.
5. Обчислюють загальний секретний ключ *K*, використовує відкритий ключ віддаленої сторони *B* та свій закритий ключ *a*.

$K = B^a \mod p$.

K получаємо рівним з обох сторін, тому що:

$B^a \mod p = (g^b \mod p)^a \mod p = \boldsymbol{g^{ab} \mod p} = (g^a \mod p)^b \mod p = A^b \mod p$

Приклад

Припустимо модуль p=47, а примітивний елемент g=23. Припустимо користувачі A та B обрали свої секретні ключі a=12 (mod 47) та b=33 (mod 47).

Для того, щоб мати загальний ключ, обидва користувачі обчислюють значення відкритих ключів:

$A = g^a \ (\ mod\ p) = 23^{12} \ (\text{mod } 47) = 27$

$B = g^b \ (\ mod\ p) = 23^{33} \ (\text{mod } 47) = 33$

Після цього користувачі обмінюються A та B та обчислюють незалежно загальний ключ.

Користувач A обчислює:

$K = B^a \ (\ mod\ p) = 33^{12} \ (\text{mod } 47) = 25.$

Користувач B обчислює:

$K = A^b \ (\ mod\ p) = 27^{33} \ (\text{mod } 47) = 25.$

Як бачимо ключі рівні $K = g^{ab}\ mod\ p = g^{ba}\ mod\ p$.

Тепер можна використовувати цей загальний ключ, наприклад, в RSA.

Тобто $K_1=25$ – перший ключ для RSA.

Другий ключ можна знайти як $K_1\ K_2 \equiv 1 \ (\text{mod } \varphi(p))$

$25\ K_2 \equiv 1 \ (\text{mod } (47 -1))$

$K_2 = 35.$

Припустимо, треба зашифрувати системою RSA повідомлення M=16.

Шифруємо ключем K_1 :

$C=M^{к1}(\text{mod p}) = 16^{25} \ (\text{mod } 47)=21$

Розшифровуємо ключем K_2:

$M=C^{к2}(\text{mod p}) = 21^{35} \ (\text{mod } 47)=16$

Були запропоновані варіанти протоколу Діффі-Геллмана для різних множин. Зокрема: мультиплікативні групи над великими скінченними полями (поля простих чисел або розширення), мультиплікативна група залишків за модулем складеного числа, еліптичні криві над скінченними полями, якобіан гіпереліптичних кривих над скінченним полем, та факторгрупи уявних квадратичних полів.

Контрольні запитання та завдання

1. Концепція криптосистем з відкритим ключем.
2. Криптосистема шифрування RSA.
3. Генерування ключів в криптосистемі RSA.
4. Методи криптоаналізу RSA.
5. Математична задача, яка лежить в основі криптографічної стійкості криптосистеми RSA.
6. Ефективність криптосистеми RSA.
7. У чому відмінність симетричних і асиметричних алгоритмів шифрування
8. Що таке секретний ключ?
9. Як виконати розшифрування методом RSA
10. Математична задача, яка лежить в основі криптографічної стійкості криптосистеми Ель- Гамала.
11. Криптосистема Ель- Гамала.
12. Зв'язок між відкритим і таємним ключами в криптосистемі Ель- Гамала.
13. Протолкол Діффі- Геллмана.
14. Зашифрувати та розшифрувати повідомлення M=2 з використанням RSA криптоалггоритму, якщо надані параметри:

P=11,Q=13 (0);

P=11,Q=23 (1);

P=7,Q=23 (2);

P=7,Q=19 (3).

Вибрати та розрахувати ключові параметри $(\varphi(n), E, D)$. P та Q обирати у залежності від номеру журналу k. № вар.$=(k+1) \bmod 4$.

15. Зашифрувати повідомлення $M = 2$ з використанням алгоритму Ель – Гамала та розшифрувати його. Таємний ключ та випадкове число k обрати самостійно. P та G обирати у залежності від номеру у журналі.

$1. P = 11, G = 3;$
$2. P = 13, G = 5;$

$3. P = 11, G = 5;$
$4. P = 11, G = 4;$
$5. P = 17, G = 5;$
$6. P = 17, G = 4;$
$7. P = 19, G = 5;$
$8. P = 17, G = 3;$
$9. P = 19, G = 3;$
$10. P = 19, G = 4;$
$11. P = 13, G = 4;$
$12. P = 23, G = 2;$
$13. P = 11, G = 3$;
$14. P = 13, G = 5;$
$15. P = 11, G = 5;$

16. Продемонструвати роботу алгоритму відкритого розповсюдження ключів Діффі- Геллмана, якщо к1, к2 та g надані в наступній таблиці. N=41.

№	k1	k2	g
1	2	13	2
2	3	19	3
3	5	13	2
4	7	11	3
5	11	5	2
6	13	3	3
7	19	2	2
8	2	11	3
9	2	17	2
10	3	11	3
11	3	13	2
12	5	11	3
13	5	17	2
14	5	19	3
15	7	13	2
16	7	17	3
17	11	13	2
18	11	17	3
19	11	19	2
20	13	17	3

Лекція 8

Проблема аутентифікації даних і електронний цифровий підпис

Метою аутентифікації електронних документів є захист документів від можливих видів злочинних дій, до яких відносять:

– активне перехоплення – порушник, що підключився до мережі, перехоплює документи (файли) і змінює їх;

– маскарад – абонент *C* посилає документ абоненту *B* від імені абонента *A*;

– ренегатство – абонент *A* заявляє, що не посилав повідомлення абоненту *B*, хоча насправді послав;

– підміна – абонент *B* змінює або формує новий документ і заявляє, що одержав його від абонента *A*;

– повтор – абонент *C* повторює раніше переданий документ, що абонент *A* посилав абонентові *B*.

Ці види злочинних дій можуть завдати істотної шкоди банківським і комерційним структурам, державним підприємствам та організаціям, приватним особам, що застосовують у своїй діяльності комп'ютерні інформаційні технології.

Після того, як з'єднання встановлене, необхідно забезпечити виконання вимог захисту при обміні повідомленнями:

1) одержувач повинен бути впевнений у істинності джерела даних;

2) одержувач повинен бути впевнений у істинності переданих даних;

3) відправник повинен бути впевнений у доставці даних одержувачу;

4) відправник повинен бути впевнений у істинності доставлених даних.

Для виконання вимог 1-2 засобом захисту є цифровий підпис. Для виконання вимог 3-4 відправник повинен одержати за допомогою пошти повідомлення, що засвідчує вручення (certified mail).

Засобами захисту в такій процедурі є цифровий підпис повідомлення, що підтверджує доставку повідомлення, яке у свою чергу є доказом пересилання вихідного повідомлення.

Якщо ці чотири вимоги реалізовані в криптосистемі, то гарантується захист даних при передачі каналом зв'язку і забезпечується функція захисту, яку називають функцією підтвердження (незаперечності) передачі. У цьому випадку відправник не може заперечувати ні факту посилки повідомлення, ні його змісту, а одержувач не може заперечувати ні факту одержання повідомлення, ні істинності його змісту.

При обробці документів в електронній формі зовсім непридатні традиційні способи встановлення істинності по рукописному підпису й відбитках пальців на паперовому документі.

Принципово новим рішенням є електронний цифровий підпис (ЕЦП).

Електронний цифровий підпис використовується для аутентифікації текстів, що передаються по телекомунікаційних каналах. Функціонально ЕЦП аналогічний звичайному рукописному підпису і володіє її основними перевагами:

– засвідчує, що підписаний документ виходить від особи, яка поставила підпис;

– не дає особі, яка відправила повідомлення, можливості відмовитися від зобов'язань, пов'язаних з підписаним текстом;

– гарантує цілісність підписаного тексту.

Цифровий підпис являє собою відносно невелику кількість додаткової цифрової інформації, яка передається разом з текстом, що підписується.

Система ЕЦП враховує дві процедури:

– процедуру підписання;

– процедуру перевірки підпису.

У процедурі підписання використовується таємний ключ відправника повідомлення, у процедурі перевірки підпису - відкритий ключ відправника (рис 8.1).

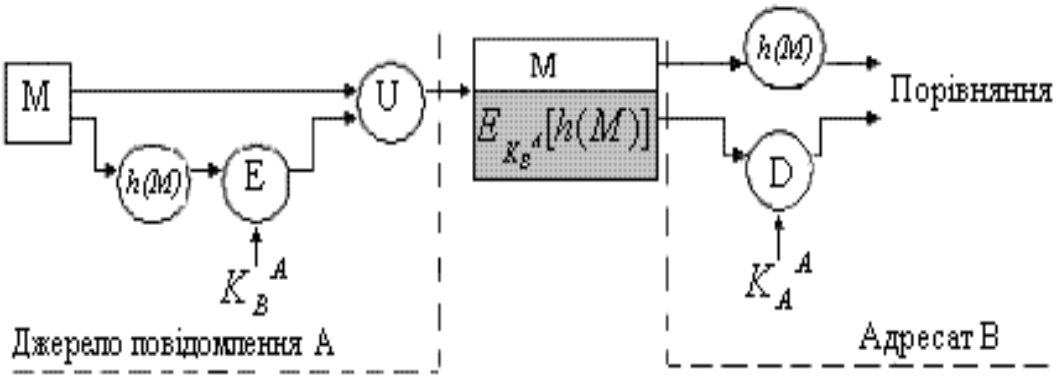

Рисунок 8.1 – Схема реалізації ЕЦП

При формуванні ЕЦП відправник насамперед обчислює хеш-функцію *h(m)* тексту *M*, який необхідно підписати. Хеш функція повідомлення *m=h(m)* являє собою один короткий блок інформації, що характеризує весь текст *M* у цілому. Потім *m* шифрується таємним ключем відправника $К_B^A$. Шифрований дайджест і є ЕЦП для даного тексту *M*.

При перевірці ЕЦП одержувач повідомлення знову обчислює хеш-функцію *m=h(m)* прийнятого тексту *M*, після чого за допомогою відкритого ключа відправника $К_A^A$. перевіряє, чи відповідає отриманий підпис обчисленому значенню *m* хеш-функції. Принциповим моментом у системі ЕЦП є неможливість підробки ЕЦП користувача без знання його таємного ключа.

Як документ, що підписується ЕЦП, може бути використаний будь-який файл. Підписаний файл створюється з вихідного файлу шляхом додавання до нього однієї або більше електронних підписів. Кожен підпис містить таку інформацію: дату підпису; строк закінчення дії ключа даного підпису; інформацію про особу, що підписала файл (П. І. П/б, посада, стисла назва фірми); ідентифікатор, що підписав (ім’я відкритого ключа); цифровий підпис.

До цифрового підпису ставляться такі вимоги:

– підпис повинен бути двійковим кодом, який залежить від повідомлення, що підписується;

– підпис повинен використовувати якусь інформацію, унікальну для відправника, щоб запобігти фальсифікації й відмову від авторства;

– простота виконання цифрового підпису;

– цифровий підпис повинен досить легко розпізнаватися та перевірятися;

– з погляду обчислень нереально фальсифікувати цифровий підпис ні за допомогою наявного повідомлення, ні за допомогою створення фальшивого цифрового підпису для наявного повідомлення;

– екземпляр цифрового підпису зручно зберігати на запам'ятовувальному пристрої.

Функція хешування

Функція хешування (хеш-функція) представляє собою перетворення, до входу якого подається повідомлення змінної довжини C, а виходом є рядок фіксованої довжини $h(C)$. Тобто хеш-функція h(.) приймає в якості аргументу повідомлення (документ) C довільної довжини і повертає хеш-значення (хеш) $H= h(C)$фіксованої довжини.

Хеш-значення $h(C)$ – це дайджест повідомлення C, тобто зжате двійкове представлення основного повідомлення C довільної довжини. Функція хешування дозволяє зжати документ C, який має бути підписано, до 128 чи 256 біт, у той час коли C може бути розміром в мегабайт і більше. Слід відзначити, що значення хеш-функції $h(C)$ залежить складним образом від документу C і не дозволяє відновити сам документ C.

Функція хешування повинна володіти наступними властивостями:

1. Хеш-функція може бути застосована до аргументу будь-якої довжини.

2. Результуюче значення хеш-функції має фіксований розмір.

3. Хеш-функцію $h(C)$ доволі просто визначити для будь-якого C.

Швидкість визначення хеш-функції повинна бути такою, щоб швидкість створення і перевірки ЕЦП у разі використання хеш-функції була значно більша за випадок використання самого повідомлення.

4. Хеш-функція повинна бути чутливою до усяких змін у тексті С, таким як вставки, викиди, перестановки і т. і.

5. Хеш-функція повинна мати властивість необоротності, тобто задача підбору документа C^*, який мав би потрібне значення хеш-функції, повинна бути нерозв'язною математично.

6. Вірогідність того, що значення хеш-функцій двох різних документів (незалежно від їх довжин) співпадають, повинна бути дуже мала; тобто для будь-якого фіксованого C неможливо знайти $C^* \neq C$, таке що $h(C^*) = h(C)$.

Теоретично можливо, що два різних повідомлення може бути зжато у одну і ту саму згортку (такий випадок називають колізія чи зіткнення). Тому для забезпечення стійкості функції хешірування необхідно передбачити спосіб уникати зіткнень. Зовсім зіткнень уникнути неможливо, оскільки в загальному випадку кількість можливих повідомлень перевищує кількість можливих вихідних значень функції хешірування.

Однак вірогідність зіткнень повинна бути низькою. Властивість 5 означає, що *h(.)* є однобічною функцією. Властивість 6 гарантує, що не може бути знайдено інше повідомлення, яке дає таку саму згортку, що запобігає фальсифікації повідомлення.

Таким чином, функція хешірування може використовуватись з метою виявлення змін повідомлення, тобто вона може служити для створення криптографічної контрольної суми, яку ще називають кодом виявлення змін чи кодом аутентифікації повідомлень.

В такому випадку хеш-функція використовується для контролю цілісності повідомлення, під час створення і перевірки електронного цифрового підпису.

Найбільш популярними хеш-функціями є MD2, MD4, MD5, SHA.

MD2, MD4, MD5 – алгоритми визначення дайджесту повідомлень, які розроблено Р. Райвестом. Кожен з них виробляє 128-бітовий хеш-код. Алгоритм MD2 – самий повільний, MD4 – самий швидкий.

Алгоритм MD5 є модифікацією MD4, в якої пожертвували швидкістю заради збільшення безпеки. SHA (Secure Hash Algorithm) – алгоритм визначення дайджесту повідомлень, який виробляє 160-бітовий хеш-код.

Алгоритм SHA декілька надійніший за MD4 і MD5.

Алгоритм цифрового підпису RSA

Першою і найбільш відомою у всьому світі конкретною системою ЕЦП стала система RSA , математичну схему якої було розроблено у 1977 році в Массачуссетському технологічному інституті.

Генерація ключів

Для того, щоб згенерувати пари ключів виконуються наступні дії:

– вибираються два великих простих числа p та q;

– обчислюється їх добуток (модуль) $n=p\,q$;

– обчислюється функція Ойлера $\varphi(n)=(p-1)(q-1)$;

– вибирається ціле e таке, що $1 < e < \varphi(n)$ та e взаємно просте з $\varphi(n)$;

– за допомогою розширеного алгоритма Евкліда знаходиться число dтаке, що e d ≡ 1(mod ($\varphi(n)$).

Число n називається модулем, а числа e і d — відкритою й секретною експонентами, відповідно. Пари чисел (n, e) є відкритою частиною ключа, а (p, q, d) — секретною. Числа p і q після генерації пари ключів можуть бути знищені, але в жодному разі не повинні бути розкриті.

Підпис s повідомлення з хеш – функцією m обчислюється з використанням секретного ключа за формулою:

$$s = m^d (mod\ n).$$

Для перевірки правильності підпису потрібно переконатися, що виконується рівність

$$m = s^e (mod\ n).$$

Електронний цифровий підпис по Ель-Гамалю

Порівняно з RSA алгоритм цифрового підпису Ель Гамаля є більш надійним і зручним для реалізації на персональних комп'ютерах. В розробленому Ель Гамалем (1984 р.) алгоритмі, який називають EGSA, вдалося уникнути слабкості алгоритму цифрового підпису RSA, зв'язаної з можливістю підробки цифрового підпису під деякими повідомленнями без знання секретного ключа. У 1991 році Національний інститут стандартів і технологій США

(National Institute of Standards and Technology – NIST) обґрунтував вибір алгоритму цифрового підпису Ель Гамаля в якості основи для національного стандарту цифрового підпису.

Формування і перевірка підпису відбувається за етапами.

1.Генерується випадкове просте число p довжини n бітів.

2. Вибирається випадковий примітивний елемент g поля Z_p.

3. Вибирається випадкове ціле число x таке, що $1< x < p-1$

4. Обчислюється y, за формулою

$$Y = g^x mod\ p$$

g- випадковий примітив поля;

x- випадкове ціле число;

p- випадкове просте число довжини n бітів;

5. Відкритим ключем є трійка ($p,\ g\ ,\ y$), закритим (таємним) ключем – число x. При роботі в режимі підпису передбачається наявність фіксованої хеш-функції h, значення якої лежать в інтервалі ($1,\ p-1$).

6.Генерування підпису:

Обчислюється хеш значення повідомлення $h(M) =m$. При цьому хеш значення лежить в інтервалі $1 < m < (p – 1)$.

Обирається випадкове число k (це рандомізатор, таємний параметр) з інтервалу $(1; p - 1)$, та взаємно просте з $(p -1)$, потім обчислюються $a = g^{\,k}$ $(mod\ p)$ та b з рівняння

$m = ax + kb$ (mod $(p-1)$)

Пара чисел (a, b) є цифровим підписом.

Отримувачу відправляється (M, a, b).

7.Перевірка підпису:

Перевіряють умови $1 < a < p$, $0 < b < (p – 1)$. Якщо хоча б одно невиконане – підпис недійсна.

Обчислюється хеш значення $m=h(M)$.

Підпис приймається при умові:

$Y^a\ a^b \pmod p = g^m \pmod p$.

Контрольні питання та задачі

1. Визначення аунтефікації.
2. Для чого потрібен цифровий підпис?
3. Злочинні дії, проти яких можна захиститися цифровим підписом.
4. Процедура підписання.
5. Процедура перевірки підпису.
6. Функція хешірування (хеш-функція) та її властивості.
7. Алгоритм цифрового підпису RSA.
8. Алгоритм цифрового підпису Ель Гамала
9. Який порядок використання відкритого та закритого ключів при створенні і перевірці ЦП?
10. Як здійснюється підпис RSA?
11. Яка відмінність підпису RSA від алгоритму шифрування RSA?
12. Як здійснюється підпис по Ель-Гамалю?
13. Як здійснюється перевірка на дійсність підпису Ель Гамала?
14. З використанням алгоритму RSA підписати та перевірити підпис повідомлення M, хеш-значення якого $h(M) = 88$. Обираються параметри $p = 17$ і $q = 11$. Ключі обрати та розрахувати самостійно згідно з вимогами алгоритму.
15. З використанням алгоритму Ель-Гамаля підписати та перевірити підпис повідомлення M . хеш-значення, якого $h(M) = 14$. Обираються $p = 19$ і $g = 10$. Нехай $x = 16$ – закритий ключ. Відритий ключ і додаткові параметри обрати та розрахувати самостійно згідно з вимогами алгоритму.

Лекція 9

Тестування чисел на простоту, імовірнісні алгоритми з однобічною помилкою

Задача визначення є число простим або складовим, є однією з фундаментальних проблемних задач теорії чисел. Пошуком рішення цієї задачі математики займалися багато століть, однак до появи криптографії з відкритим ключем ефективних алгоритмів перевірки чисел на простоту знайдено не було.

Більшість криптосистем, що базуються на асиметричній криптографії, використовують у якості одного або декількох параметрів прості числа. До них належать криптографічні системи , що використовують перетворення в кільцях (Рабина, RSA), полях Галуа (Ель - Гамала), групах точок еліптичних кривих та ін. Стійкість цих криптосистем залежить від правильності побудови простих чисел різної довжини (від 128 до 2048 і більше біт).

Простим числом називається число p, що не має нетривіальних дільників, тобто ділиться тільки на 1 і на само себе. Кількість простих чисел в інтервалі можна оцінити за допомогою нерівності Чебишева:

Теорема 9.1 Існують дві постійні c_1 і c_2, такі, що $0 < c_1 < 1$, $c_2 > 1$ і для всіх $x \geq 2$ виконуються нерівності

$$c_1 x/\ln x < \pi(x) < c_2 x/\ln x.$$

Функція $\pi(x)$ відображає кількість простих чисел, які менше або дорівнюють x. У нулі вона дорівнює 0, і стрибком зростає на одиницю в точках,що відповідають простому числу. Закономірність функції $\pi(x)$ вчені намагаються розгадати давно. Над цією проблемою працювали Гаус, Лежандр, Чебишев, Адамар. Чебишев установив границі, між якими перебувають значення функції $\pi(x)$. Він показав, що при $0 \leq x \leq \infty$ виконуються умови: $0{,}92129 \dfrac{x}{\ln x} < \pi(x) < 1{,}10555 \dfrac{x}{\ln x}$.

Задача тестування чисел на простоту формулюється таким чином.

Нехай задане ціле число n , необхідно визначити, чи є n простим числом. Алгоритми, які призначені для рішення цієї задачі, називають тестами на простоту. Якщо алгоритм приймає на вхід n, то говорять, що n проходить або витримує тест на простоту.

Всі алгоритми тестування на простоту можна розділити на три великих класи:

– імовірнісні алгоритми з однобічною помилкою;

– алгоритми з імовірнісним часом виконання;

– детерміновані алгоритми.

Якщо n проходить тест імовірнісним алгоритмом з однобічною помилкою, то n є простим не достовірно, а тільки з деякою імовірністю $\delta < 1$. У випадку алгоритму з імовірнісним часом виконання, якщо число n проходить тест, то воно достовірно є простим, однак число кроків алгоритму, а, отже, і час виконання алгоритму залежать від деякого випадкового числа.

Детермінований алгоритм розпізнає простоту числа n з імовірністю 1 й із заздалегідь відомим числом кроків.

Алгоритми тестування чисел на простоту можна також класифікувати по складності виконання:

*експоненційні - алгоритми з оцінкою складності порядку $O(c^{\log n})$ для деякої константи $c > 1$;

*субекспоненційні- $O(c^{(\log n)^{\nu}(\log\log n)^{1-\nu}})$ для $c > 1$ й $0 < \nu < 1$;

*квазіполіноміальні - $O((\log n)^{c\log\log n})$ для $c > 0$;

*поліноміальні - $O(\log^{c} n)$ для $c \geq 1$.

Експоненційні й субекспоненційні алгоритми є алгоритмами факторизації й у випадку, якщо число n складене, знаходять його нетривіальний дільник. Першими алгоритмами з поліноміальною складністю виконання стали імовірнісні алгоритми з однобічною помилкою.

Критерій Вільсона

В 1770 р. Е. Варинг опублікував наступну теорему, приписувану Д. Вільсону.

Теорема 9.2

Для будь-якого n наступні умови еквівалентні:

(а) n - просте*;*

(б) $(n-1)!\equiv -1(\mathrm{mod}\, n)$.

Доведення.

У випадку $n=2$ твердження очевидно.

(Нагадуємо наступну теорему теорії чисел. Теорема. Елемент a із множини Z_n має зворотний по множенню елемент a^{-1} тоді й тільки тоді, коли $НОД(a,n)=1$, тобто a й n - взаємнопрості.) .

Якщо $n=p>2$ - просте, то кожен елемент a поля , відмінний від 1 і -1 має зворотний a^{-1} , причому $a\neq a^{-1}$. Тому

$$(n-1)!\equiv (-1)\prod_{a\neq 1,-1} aa^{-1}\equiv -1(\mathrm{mod}\, n).$$

Якщо $n=ab$ - складене, $1<a<n$, то $a\,|\,(n-1)!$, і отже, $(n-1)$ є необоротним елементом кільця Z_n .

Тому $(n\not-1)!\equiv -1(\mathrm{mod}\, n)$. Теорема доведена.

Даний критерій іноді буває зручний у доведеннях, але застосовувати його для перевірки простоти неможливо через велику трудомісткість.

Тест Лемана

Тест Лемана один з перших алгоритмів класу імовірнісних алгоритмів з однобічною помилкою. Цей тест використовує властивості малої теореми Ферма, що стверджує, що якщо *n* просте, то виконується умова: при всіх $a\in\{1,2,3,...,n-1\}$ має місце порівняння

$$a^{n-1}\equiv \pm 1(\mathrm{mod}\, n). \qquad (9.1)$$

Зворотне твердження неправильно.

Із цього твердження випливає, що якщо порівняння (9.1) не виконане хоча б для одного числа a в інтервалі $\{1,2,3,...,n-1\}$ тоді n складене. Тому можна запропонувати такий імовірнісний тест простоти:

1) обираємо випадкове число з інтервалу $\{1,2,3,...,n-1\}$ й перевіряємо за допомогою алгоритму Евкліда умову $(a,n)=1$;

2) якщо вона не виконується, то відповідь « n - складене » ;

3) перевіряємо виконання порівняння (9.1) ;

4) якщо порівняння не виконане, то відповідь « n - складене » ;

5) якщо порівняння виконане, то відповідь невідома, але можна повторити тест ще раз.

Якщо виконується порівняння (9.1), то говорять, що число n є псевдопростим на підставі a. Помітимо, що існує нескінченно багато пар чисел $(a,n)=1$, де n - складене й псевдопросте на підставі a. Наприклад, при $(a, n) = (2, 341)$ одержуємо $2^{340} = (2^{10})^{34} \equiv 1 \pmod{n}$, хоча $341 = 11 \times 31$.

Особливий випадок становлять складені числа, для яких умова (9.1) виконується при будь яких a. Вони називаються псевдопростими числами, або числами Кармайкла. Таким чином, при застосуванні вищеописаного тесту на основі малої теореми Ферма можуть виникнути три ситуації:

– число n - просте й тест завжди говорить « невідомо»;

– число n складене й не є числом Кармайкла, тоді з імовірністю успіху не менше 1/2 тест дає відповідь « n - складене »;

– число n складене і є числом Кармайкла, тоді тест завжди дає відповідь « невідомо ».

Приймемо без доказу наступну теорему, що знадобиться надалі.

Теорема 9.3 (Кармайкл, 1912).

Нехай n непарне складене. Тоді

(а) якщо $p^2 \mid n$, $p>1$ тоді n не є числом Кармайкла;

(б) якщо $n = p_1 \cdot p_2 \cdot ... \cdot p_k$, $p_i \neq p_j$, тоді n число Кармайкла в тім і тільки в тому випадку, коли при всіх *i* виконана умова $(p_i - 1)/(n-1)$;

(в) якщо $n = p_1 \cdot p_2 \cdot ... \cdot p_k$, $p_i \neq p_j$, число Кармайкла, тоді $k \geq 3$.

Тест Соловея - Штрассена

Теорема 9.4

Для будь-якого непарного *n* такі умови еквівалентні:

(а) n - просте;

(б) для кожного $a \in Z_n$ виконується порівняння

$$a^{\frac{n-1}{2}} \equiv \left(\frac{a}{n}\right) (\mathrm{mod}\, n) - \qquad (9.2)$$

Доведення:

Якщо n просте, тоді дане порівняння, мабуть, виконується в силу властивостей символу Лежандра. (Твердження теорії чисел. Символ Лежандра виражається як $\left(\frac{x}{p}\right) \equiv x^{(p-1)/2} (\mathrm{mod}\, p)$).

Нехай тепер виконана умова (б), але n не просте. Тоді,

$$a^{n-1} \equiv \left(a^{\frac{n-1}{2}}\right)^2 \equiv \left(\frac{a}{n}\right)^2 = 1 (\mathrm{mod}\, n).$$

Тому *n* є числом Кармайкла, і по властивості (б) твердження (9.2) повинне мати вигляд $n = p_1 \cdot p_2 \cdot ... \cdot p_k$, $p_i \neq p_j$. Виберемо елемент b, *що* не є квадратичним лишком по модулю p_1. По китайській теоремі про залишки знайдеться елемент a, *що* задовольняє умовам

$$\begin{cases} a \equiv b (\mathrm{mod}\, p_1) \\ a \equiv 1 (\mathrm{mod}\, p_2) \\ \ldots\ldots\ldots\ldots\ldots \\ a \equiv 1 (\mathrm{mod}\, p_k) \end{cases}.$$

Для цього елемента повинне виконуватися

$$\left(\frac{a}{n}\right)=\left(\frac{a}{p_1}\right)\left(\frac{a}{p_2}\right)\ldots\left(\frac{a}{p_k}\right)=\left(\frac{a}{p_1}\right)=\left(\frac{b}{p_1}\right)=-1.$$

За умовою, для даного елемента повинне виконуватися порівняння (9.2) , звідки $a^{\frac{n-1}{2}}=\left(\frac{a}{n}\right)\equiv -1\left(\mathrm{mod}\, p_2\right)$. Разом з тим, відповідно до вибору елемента a повинне бути $a\equiv 1(\mathrm{mod}\, p_2)$. Протиріччя.

Р. Соловей й У. Штрассен запропонували наступний імовірнісний тест для перевірки простоти чисел:

1) обираємо випадкове число a з інтервалу $\{1,2,...,n-1\}$й перевіряємо за допомогою алгоритму Евкліда умову $(a,n)=1$;

2) якщо вона не виконується, то відповідь « n - складене »;

3) перевіряємо виконання порівняння (9.2);

4) якщо порівняння не виконане, то відповідь « n - складене »;

5) якщо порівняння виконане, то відповідь невідома (тест можна повторити ще раз).

Даний тест аналогічний тесту на основі малої теореми Ферма, але має вирішальну перевагу - при його використанні виникає тільки дві ситуації:

– число n - просте й тест завжди говорить « невідомо»;

– число n складене й тест із імовірністю успіху не менше 1/2 дає відповідь « n - складене »;

Після повторення тесту k раз імовірність невідбраковування складеного числа не перевершує $1/2^k$.

Тест Рабіна-Міллера

Нехай n - непарне й $n-1=2^s t$, t - непарне. Якщо число n є простим, то при всіх $a\geq 2$ виконується порівняння $a^{n-1}\equiv 1(\mathrm{mod}\, n)$. Тому, розглядаючи елементи a^t , a^{2t} ,..., $a^{2^{s-1}t}$ можна помітити, що або серед них знайдеться рівний $-1(\mathrm{mod}\, n)$, або $a^t\equiv 1(\mathrm{mod}\, n)$.

На цьому зауваженні заснований наступний тест простоти:

1) обираємо випадкове число a з інтервалу $\{1,2,..., n-1\}$ й перевіряємо за допомогою алгоритму Евкліда умова $(a, n) = 1$;

2) якщо воно не виконується, то відповідь « n - складене »;

3) обчислюємо $a^t \pmod n$;

4) якщо $a^t \equiv \pm 1 (\mathrm{mod}\, n)$, тоді переходимо до п. 1;

5) обчислюємо $(a^t)^2$, $(a^t)^4$, ..., $\left(a^t\right)^{2^{s-1}} (\mathrm{mod}\, n)$ доти , поки не з'явиться -1;

6) якщо жодне із цих чисел не дорівнює –1, то відповідь «n - складене » ;

7) якщо ми досягли -1, то відповідь невідома, і тест можна повторити.

М. Рабін в 1980 р. запропонував даний імовірнісний варіант тесту й довів, що частка чисел $a \in Z_n^t$, для яких число n є псевдопростим по даній підставі, не менше *3/4*. Тому після повторення даного тесту k раз імовірність невідбраковування складеного числа не перевершує $1/4^k$.

Контрольні питання та завдання

1. Пояснити актуальність для криптографії задачі тестування чисел на простоту.
2. Сформульувати теорему Чебишева щодо розподілу простих чисел.
3. Поняття про імовірнісні алгоритми з однобічною помилкою.
4. Сформулювати основні теореми, які лежать в основі імовірнісних алгоритмів з однобічною помилкою.
5. Поясніть сутність тесту Лемана.
6. Поясніть сутність тесту Соловея-Штрассена.
7. Поясніть сутність тесту Рабіна –Міллера.

Лекція 10

Побудова великих простих чисел

Розглянемо тепер такі методи перевірки чисел на простоту, при застосуванні яких можна стверджувати, що числа дійсно є простими. На відміну від попередніх тестів, які використовували необхідні умови простоти й давали відповіді типу $''n$ - не просте $''$, $''$або $''$ не знаю $''$або $''$ імовірність того, що n - не просте , не вище заданого як завгодно малого значення $''$, дані тести засновані на застосуванні достатніх умов простоти.

Тому вони можуть давати як відповіді типу $''n$ - не просте $''$, $''$не знаю$''$, так й $''n$ - просте $''$. Ця властивість застосовується для побудови простих чисел.

Загальна схема в цьому випадку така: обирається деяка послідовність чисел спеціального виду, серед яких потрібно знайти просте число, потім до чисел із цієї послідовності застосовується тест доти, поки він не дасть позитивну відповідь. Якщо ця відповідь $''n$ - не просте$''$, то обирається наступне число. Якщо отримано відповіді $''n$ - просте $''$, то шукане просте число побудоване.

Розглянемо достатні умови простоти чисел, які ,звичайно, застосовуються в алгоритмах побудови доказово простих чисел.

Критерій Люка

Теорема, уперше доведена Люка в 1876 р., перетворює малу теорему Ферма в критерій простоти числа n, достатня умова якого може бути ефективно використана для доказу простоти цього числа.

Теорема 10.1 (Люка)

Натуральне число n є простим у тому і тільки в тому випадку, коли виконується умова

$$\exists a \in Z_n^*, (a^{n-1} \equiv 1(\bmod n)) \wedge (\forall q | (n-1), a^{\frac{n-1}{q}} \neq 1(\bmod n)). \quad (10.1).$$

Доведення:

Якщо n просте, то в полі Z_n є примітивний елемент, що і буде шуканим. Навпаки, нехай для елемента a виконується умова (10.1). Якщо $ord(a) = m$, те $m|(n-1)$, причому умова (10.1) гарантує, що $m = n-1$. Отже, $\psi(n) = n-1$ і n - просте. Теорема доведена.

Зауваження (Селфридж).

Умова (10.1) у даній теоремі можна замінити на кожну з наступних умов:

$$\exists a \in Z_{n,} ord(a) = n-1; \qquad (10.2)$$

$$\forall q|(n-1), \exists a \in Z_n^*, (a^{n-1} \equiv 1(\bmod n)) \wedge (a^{\frac{n-1}{q}} \neq 1(\bmod n)). \qquad (10.3)$$

Дійсно, те, що (7.1) =>(7.2) й (7.1)=>(7.3) , очевидно.

Доведемо, що (10.3)=>(10.2) . Нехай $n-1 = q_1^{k1} \ldots q_s^{ks.}$. За умовою для кожного i знайдеться a_i таке, що $ord(a_i)|(n-1)$, але $ord(a_i)$ не ділить число $\frac{n-1}{q_i}$. Отже, $q_i^{ki} | ord(a_i)$. Виходить, знайдуться елементи b_i, такі, що $ord(b_i) = q_i^{ki}$. Тепер елемент $a = b_1 \ldots b_s$ буде шуканим, тому що порядки елементів b_i взаємно прості й

$$ord(b_1 \ldots b_s) = q_1^{k1} \ldots q_s^{ks} = n-1.$$

Теорема Люка дозволяє довести простоту числа n у випадку, коли відоме розкладання на прості співмножники числа $n-1$. Для цього можна використати детермінований алгоритм, заснований на переборі всіх можливих значень $a \in Z_n^*$, або скористатися наступним імовірнісним методом:

1) обираємо випадкові числа $a_1, \ldots, a_s \in Z_n^*$ й перевіряємо для них умову (10.1);

2) якщо умова (10.1) виконана хоча б для одного із цих чисел, то n просте, якщо ні, те відповідь невідома.

Аналогічний метод можна побудувати, використовуючи умову (10.3).

Проілюструємо цей метод стосовно до чисел Ферма. Числами Ферма називають числа виду $F_k = 2^{2^k} = 1, k = 1,2,\ldots$(Покажіть , що число виду $2^m = 1$ може бути простим у тому і тільки в тому випадку, коли $m = 2^k$.)

Ферма висловлював припущення, що всі числа такого виду - прості. При $n = 0,1,2,3,4$ це дійсно так. Але при $n = 5$, як показав Ейлер в 1732 р., справедливе розкладання

$$F_5 = 2^{2^5} + 1 = 4294967297 = 641 \times 6700417$$

В 1878р. Іван Михейович Первушин показав, що числа F_{12} й F_{23} також не є простими. (Помітимо , що число F_{23} має 2525223 цифри. При відтворенні такого числа знадобився б рядок довжиною в 5 км або книга об'ємом 1000 сторінок.)

Теорема 10. 2 (Пепін, 1877).

Числа $F_k = 2^{2^k} + 1$ при $k \geq 1$ є простими в тому і тільки в тому випадку, коли виконується умова $3^{\frac{(F^k - 1)}{2}} \equiv -1 (\mathrm{mod}\, F_k)$.

Доведення:

Тому що єдиним простим дільником числа $F_k - 1$ є 2 , то досить перевірити умову теореми Люка при $q = 2$. Покажемо, що в якості числа a можна взяти число 3, тобто досить перевірити умову $3^{\frac{(F^k - 1)}{2}} \neq 1 (\mathrm{mod}\, F_k)$.

Використовуючи формулу Ейлера для обчислення значень квадратичних лишків і квадратичний закон взаємності Гауса одержуємо, що при простому F_k повинне бути

$$3^{\frac{(F_k - 1}{2}} \equiv (\frac{3}{F_k}) \equiv (-1)^{\frac{(F_k - 1}{2}} (\frac{F_k}{3}) \equiv (\frac{F_k}{3}) (\mathrm{mod}\, F_k).$$

Тепер помітимо, що $F_k \not\equiv 1 (\mathrm{mod}\, 3)$, і тому умова $F_k \neq 0 (\mathrm{mod}\, 3)$ рівносильна рівності $F_k \equiv 2 \equiv -1 (\mathrm{mod}\, 3)$. Теорема доведена.

Теорема Люка послужила відправним пунктом для побудови цілої групи тестів, що дозволяють перевіряти простоту числа. Багато хто з них одержали назву $(n-1)$- методів, тому що припускають знання повної або часткової факторизації числа $(n-1)$. Ще одне узагальнення теореми Люка засновано на розгляді інших груп замість мультиплікативної групи Z_n^*.

Фактично, доказ простоти числа n в теоремі Люка засновано на вивченні властивостей групи Z_n^*: якщо яким-небудь чином вдається встановити, що її порядок дорівнює $n-1$, то число n-просте. Дана ідея лежить в основі таких методів, як метод еліптичних кривих і метод числового поля.

Теорема Поклінгтона

В 1914 р., Х. Поклінгтоном була доведена наступна теорема.

Теорема 10.3 (Поклінгтон). Нехай $n = q^k R + 1 > 1$, де q - просте, що не є дільником R. Якщо існує ціле a таке, що $a^{n-1} \equiv 1 (\mathrm{mod}\, n)$ й $(a^{\frac{n-1}{q}} - 1, n) = 1$, то кожен простий дільник p числа n має вигляд $p = q^k r + 1$ при якомусь r.

Доведення:

Нехай p- простий дільник числа n. Тоді з умови теореми випливає , що $a^{n-1} \equiv 1 (\mathrm{mod}\, p)$ й $a^{\frac{n-1}{q}} \neq 1 (\mathrm{mod}\, p)$. Звідси одержуємо, що порядок m елемента a за модулем p задовольняє умовам: $m | (n-1)$ і m не ділить $\frac{n-1}{q}$. Тому $q^k | m$. У силу малої теореми Ферма $m | (p-1)$. Отже, $p - 1 = q^k r$. Теорема доведена.

Застосовуючи дану теорему для всіх дільників q числа $n-1$, одержуємо наступну теорему, що є узагальненням теореми Люка на випадок $R > 1$.

Теорема 10.4 Нехай $n = FR + 1 > 1$, де $0 < R < F$. Якщо для будь-якого простого дільника q числа F існує ціле a таке, що $a^{n-1} \equiv 1 (\mathrm{mod}\, p)$ й $(a^{\frac{n-1}{q}} - 1, n) = 1$, тоді число n-просте.

Доведення:

Нехай n- складене й p- нетривіальний простий дільник числа n. Помітимо, що завжди можна вибрати дільник так, що $p \le \sqrt{n}$. Тоді з умови теореми випливає, що для всіх простих дільників q числа F існує ціле a таке, що $a^{n-1} \equiv 1(\mathrm{mod}\, p)$ й $a^{\frac{n-1}{q}} \neq 1(\mathrm{mod}\, p)$. Міркуючи аналогічно зауваженню до теореми Люка, одержуємо, що повинен знайтися елемент, що має порядок рівний F по модулю p. У силу малої теореми Ферма $F \le p-1$.

Отже, справедливий ланцюжок нерівностей

$$p^2 \ge (F+1)^2 > R(F+1) \ge RF+1 \ge n.$$

Але $p \le \sqrt{n}$, протиріччя.

Дана теорема показує, що якщо вдалося частково факторизувати число $n-1$, причому факторизовано частина задовольняє умові $F \ge \sqrt{n}$, те n - просте.

Перш ніж переходити до подальшого, приведемо дві класичні частки випадку даної теореми.

Теорема 10. 5 (Прот, 1878). Нехай $n = 2^k R + 1$, де $R < 2^k$.

Якщо існує число a, для якого виконується умова

$$a^{\frac{n-1}{2}} \neq 1(\mathrm{mod}\, n),$$

то n- просте.

Теорема 10.6 (Прот,1878)). Нехай $n = 2^k R = 1$, де $R < 2^k$, $3 < 2^k + 1$ і 3 не ділить R. Тоді n просте в тому і тільки в тому випадку, коли виконується умова

$$3^{\frac{n-1}{2}} \equiv -1(\mathrm{mod}\, n).$$

Доведення:

У силу теореми Поклінгтона досить перевірити умову $a^{\frac{n-1}{q}} \neq 1(\mathrm{mod}\, n)$ при $a = 3$ й $q = 2$. Тому що за умовою $n = 2^k R + 1 \neq 1(\mathrm{mod}\, 3)$, то умова $3^{\frac{n-1}{2}} \neq 1(\mathrm{mod}\, n)$ рівносильна виконанню рівності

$$3^{\frac{n-1}{2}} \equiv \left(\frac{3}{2^k R+1}\right) \equiv (-1)^{\frac{n-1}{2}} \left(\frac{2}{3}\right) \equiv -1(\mathrm{mod}\, n).$$

Помітимо, що якщо в теоремі Поклінгтона замінити рівність $(a^{\frac{n-1}{q}} - 1, n) = 1$ на більш слабку умову $a^{\frac{n-1}{q}} \neq 1(\mathrm{mod}\, n)$, то можна одержати наступний результат.

Лема 10.1 Нехай $n = q^k R + 1$, де q - просте число, що не є дільником R. Якщо існує ціле a таке, що $a^{n-1} \equiv 1(\mathrm{mod}\, n)$ й $a^{\frac{n-1}{q}} \neq 1(\mathrm{mod}\, n)$, то знайдеться простий дільник p числа n виду $p = q^k r + 1$ при якомусь r.

Доведення

Нехай $n = p_1^{m_1} \ldots p_k^{m_k}$. Тоді за умовою теореми в силу китайської теореми про залишки випливає, що існує таке i , що $a^{n-1} \equiv 1(\mathrm{mod}\, p_i^{m_i})$ й $a^{\frac{n-1}{q}} \neq 1(\mathrm{mod}\, p_i^{m_i})$.

Звідси одержуємо, що порядок t елемента a за модулем $p_i^{m_i}$ задовольняє умовам: $t|(n-1)$ і t не ділить $\frac{n-1}{q}$. Тому $q^k | t$.

У силу леми Гауса про циклічність мультиплікативної групи кільця $Z_{p_i}^{*m_i}$ одержуємо $t | p_i^{m_i - 1}(p_i - 1)$. Помітимо, що числа p_i й q взаємно прості як дільники сусідніх чисел. Тому $q^k | (p_i - 1)$. Отже, $p_i - 1 = q^k r$.

Хоча цей результат слабкіше, ніж теорема Поклінгтона, даний підхід, як показав Н. Дієметко в 1988 р., також може бути ефективно використаний для доказу простоти чисел.

Теорема 10.7 (Дієметко).Нехай $n = qR + 1$, де q - просте, R - парне й $R < 4(q+1)$. Якщо існує ціле a таке, що $a^{n-1} \equiv 1(\mathrm{mod}\, n)$ й $a^{\frac{n-1}{q}} \neq 1(\mathrm{mod}\, n)$, то n - просте.

Доведення:

Нехай n - не просте й $n = p_1^{m1} \ldots p_k^{mk}$. Тоді по лемі одержуємо, що існує таке i, що $q|(p_i - 1)$. Позначимо $n = p_i Q$. Тоді $n \equiv p_i Q (\mathrm{mod}\, q)$, де $n \equiv 1 (\mathrm{mod}\, q)$ й $p_i \equiv 1 (\mathrm{mod}\, q)$. Звідси $Q \equiv 1 (\mathrm{mod}\, q)$. Отже, $Q = qt + 1 \geq 2q + 1$, де t - не може дорівнювати 0, інакше n - просте, або 1, тому що Q - непарне. Аналогічно, $p_i = qs + 1 \geq 2q + 1$. Таким чином,

$$n = p_i Q \geq (1 + 2q)^2 = q \times 4(q + 1) + 1 > qR + 1.$$

Протиріччя. Теорема доведена.

Помітимо, що за умовою теореми числа n й R можуть бути не взаємно прості. Ця теорема лежить в основі алгоритму генерації простих чисел у вітчизняному стандарті на цифровий підпис Р 34.10-94. У ньому в якості a обираються не дуже високі степені числа 2, а R перебуває перебором. (З 1 липня 2002 р. цей стандарт був замінений на новий Р 34.10-2001.)

Метод Маурера

В 1995 р. У. Маурер опублікував швидкий алгоритм генерації доказово простих чисел, близьких до випадкового. У його основі лежить посилення теореми Поклінгтона на випадок, коли факторизовано частину F числа $n - 1$ задовольняє нерівності $F \geq \sqrt[3]{n}$. Крім того, йому вдалося оцінити ймовірність успіху при випадковому пошуку числа a в умові теореми Поклінгтона.

Наступна лема є спеціальним випадком теореми Поклінгтона.

Лема 10.2 Нехай $n = 2FR + 1 > 1$. Якщо існує ціле a таке, що для будь-якого простого дільника q числа F виконані умови $a^{n-1} \equiv 1 (\mathrm{mod}\ n)$

і $(a^{\frac{(n-1)}{q}} - 1, n) = 1$, те кожен простий дільник p числа n має вигляд $p = mF + 1$ при деякому цілому m. Якщо , крім того , $F > \sqrt{n}$, або F - парне й $R < F$, те n - просте.

Доведення:

Нехай n - складене й p - нетривіальний простий дільник числа n. Тоді за умови теореми випливає, що $a^{n-1} \equiv 1 (\mathrm{mod}\ p)$ й $a^{\frac{(n-1)}{q}} \neq 1 (\mathrm{mod}\ p)$. Міркуючи

аналогічно зауваженню до теореми Люка, одержуємо, що повинен знайтися елемент, що має порядок рівний F за модулем p. У силу малої теореми Ферма $F|(p-1)$.

Для доказу другого твердження, припустимо, що $p < n$. Тоді $p \le \sqrt{n}$. Якщо $F > \sqrt{n}$, то $p = mF + 1 > \sqrt{n}$. Якщо F - непарне й $R < F$, те $p \ge 2F + 1$ й

$$p^2 \ge (2F+1)^2 > (2R+1)(2F+1) > 2RF + 1 = n.$$

Протиріччя.

Наступна лема доведена Д. Коувером і Дж. Куіскуотером в 1992 р.

Лема 10. 3 Нехай n, F, R і a задовольняють умові леми 1. Визначимо числа $x \ge 0$ й $0 \le y < F$ рівністю $2R = xF + y$. Якщо $F \ge \sqrt[3]{n}$ й число $y^2 - 4x$ не дорівнює нулю й не є повним квадратом, то n - прості.

Доведення:

Відповідно до леми 1 для кожного простого дільника p числа n виконується нерівність $p \ge F + 1$. За умовою $n \le F^3$. Тому, якщо число n - складене, то воно не може мати більше двох простих дільників. Нехай

$$n = 2FR + 1 = (m_1 F + 1)(m_2 F + 1) \text{ и. } m_1 \ge m_2$$

Маємо $m_1 m_2 < F$ інакше $n > F^3$.

Якщо $m_1 + m_2 \le F$, то $F > m_1 m_2 \ge m_1 (F - m_2) \ge F - 1$.

Звідси $m_1 = F - 1, m_2 = 1$, однак у цьому випадку $n = F^3 + 1$. Тому $m_1 + m_2 < F$.

Отже, $m_1 m_2 = x$ і $m_1 + m_2 = y$. По теоремі Вієта $m_{1,} m_2$ є коренями квадратного рівняння $m^2 - ym + x = 0$, що має рішення в цілих числах у тому і тільки в тому випадку, якщо $y^2 - 4x$ є повним квадратом або нулем. Лема доведена.

Помітимо, що переконатися, що задане число не є повним квадратом, можна обчисливши символ Лежандра для декількох маленьких простих модулів.

Якщо при деякому модулі число не буде квадратичним відрахуванням, то воно не буде й повним квадратом.

Нехай $\psi(x)$- функція Ейлера.

Лема 10.4 Нехай p - просте число й $d|(p-1)$. Позначимо через T число елементів $x \in Z_p^*$, порядок яких ділиться на d. Тоді справедлива оцінка

$$T \geq \frac{\psi(d)}{d}(p-1),$$

причому рівність виконується в тому й у тільки в тому випадку, коли $(d,(p-1)/d)=1.$

Доведення:

Використовуючи властивості функції Ейлера, одержуємо

$$T = \sum_{d|d^*|(p-1)} \psi(d^*) = \sum_{k|(p-1)/d} \psi(kd) \geq \sum_{k|(p-1)/d} \psi(k)\psi(d) = \psi(d) \sum_{k|(p-1)/d} \psi(k) = \psi(d)\frac{p-1}{d},$$

причому рівність виконана в тому і тільки в тому випадку, коли $(d,(p-1)/d)=1.$

Контрольні питання та задачі

1. Проблема побудови великих простих чисел та актуальність її для криптографії.
2. Навести достатні умови простоти.
3. Критерій Люка.
4. Теорема Поклінгтона.
5. Теорема Діеметко.
6. Навести метод Маурера побудови великих простих чисел.

Лекція 11

Загальні відомості відносно методів криптоаналізу двоключових криптосистем, алгоритми факторизації

Метод Полларда

Найбільш популярним імовірнісним алгоритмом факторизації є ,так званий метод, запропонований Дж. Поллардом в 1975 р.

Алгоритм 11.1

Крок 1. Обираємо багаточлен $f(x) \in Z[x]$.

Крок 2. Випадково обираємо $x_0 \in Z_n$ й, обчислюючи значення $x_i = f(x_{i-1}) \bmod n,\ i = 1,\ldots,m,$ перевіряємо тест кроку 3.

Крок 3. Полигаємо $j = 2^h - 1$ й для кожного $2^h \le k < 2^{h+1}$ обчислюємо $d = (x_j - x_k, n)$. Якщо $1 < d < n$, то нетривіальний дільник числа n знайдений.

Якщо $d = 1$ або $d = n$, то переходимо до наступного значення h.

Щоб зрозуміти сутність алгоритму розглянемо його на простому прикладі з невеликими значеннями чисел.

Наприклад

Нехай потрібно факторизувати число $n = 209$. Нехай обрано багаточлен з цілими коефіцієнтами $f(x) = x^2 + 2,\ x_0 = 1,\ n = 209$.

Обчислюємо $x_i = f(x_{i-1}) \bmod n,\ i = 1,\ldots,m.$

$x_1 = 1^2 + 2 = 3 (\bmod 209)$,

$x_2 = 3^2 + 2 = 11 (\bmod 209)$,

$x_3 = 11^2 + 2 = 123 (\bmod 209)$,

$x_4 = 123^2 + 2 = 15131 (\bmod 209) = 83 (\bmod 209)$,

$x_5 = 83^2 + 2 = 203 (\bmod 209)$,

$x_6 = 203^2 + 2 = 38 (\bmod 209)$,

$x_7 = 38^2 + 2 = 192 (\mathrm{mod}\, 209)$,

$x_8 = 192^2 + 2 = 82 (\mathrm{mod}\, 209)$,

$x_9 = 82^2 + 2 = 38 (\mathrm{mod}\, 209)$,

$x_{10} = 38^2 + 2 = 192 (\mathrm{mod}\, 209)$.

Знаходимо $d = (x_j - x_k, n)$.

$(x_1 - x_2, n) = (11 - 3, 209) = 1$

$(x_1 - x_3, n) = (123 - 3{,}209) = 1$

$(x_3 - x_4, n) = (123 - 83{,}209) = 1$

$(x_3 - x_5, n) = (203 - 123{,}209) = 1$

$(x_3 - x_{6,} n) = (123 - 38{,}209) = 1$

$(x_3 - x_7, n) = (192 - 123{,}209) = 1$

$(x_7 - x_8, n) = (192 - 82{,}209) = (110{,}209) = 11$

$1 < 11 < 209$ – нетривіальний дільник знайдено.

209=11 19 – другий дільник знайдено.

Варто помітити, що ми відразу вдало підібрали багаточлен, при якому метод швидко дав результат. Це буває не завжди.

Помітимо, що сучасні методи факторизації на практиці часто виконуються в 3 етапи.

Етап 1. Пробні ділення на 1 - 2 тисячі перших простих чисел.

Етап 2. Знаходження маленьких простих дільників методом Полларда або Полларда - Штрассена (у якому число z підбирають із міркувань оптимізації загальної трудомісткості алгоритму).

Етап 3. Для знаходження більших простих дільників застосовується один із субекспоненційних алгоритмів.

p-1 метод факторизаії Полларда

Припустимо, що n - непарне складене число, що не має невеликих простих дільників. Позначимо через p найменший простий дільник числа n.

Наша задача полягає в його знаходженні.

Припустимо, що число $(p-1)$ розкладається в добуток невеликих простих дільників. Виберемо число k, що є параметром методу.

Для успішної роботи алгоритму потрібно, щоб виконувалася умова

$$(p-1)|M(k),$$

де $M(k) = \text{НСК}(1,2,...,k\ )$ (замість $M(k)$ можна використати $k!$ або добуток $p_1^{\alpha_1}...p_2^{\alpha_2}$ перших k простих чисел у деяких степенях $\alpha_1 \geq ... \geq \alpha_k$, які обираються з евристичних міркувань).

У силу малої теореми Ферма виконується порівняння

$$2^{M(k)} \equiv 1(\text{mod}\ p).$$

Якщо при цьому

$$2^{M(k)} \neq 1(\text{mod}\ n),$$

то

$$p|(2^{M(k)}-1,n),$$

де $1 < p,(2^{M(k)}-1,n) < n.$ Таким чином, $d = (2^{M(k)}-1,n)$ - є власним дільником числа n кратним p.

На цій ідеї заснований наступний метод знаходження власного дільника числа n. Число k невідомо, тому використовується послідовний перебір малих значень до деякого фіксованого значення.

Нехай k – ціле число, наприклад $k < 10^6$, і c – невелике ціле з умовою $(c,n) = 1,$ наприклад $c = 2.$

Крок 1. Для кожного i від 1 до k обчислюється $m_i = c^{M(i)} \bmod n$ й перевіряємо тест кроку 2.

Крок 2. Обчислити $d = (m_i - 1,n).$ Якщо $1 < d < n,$ те знайдений нетривіальний дільник числа n. У противному випадку полагаємо $i = i + 1.$

Наприклад

Методом Полларда факторизувати число 247.

Нехай $M(i) = k!, c = 2.$

Обчислюємо для кожного i від 1 до k $m_i = c^{M(i)} \bmod n$.

$M_1 = 2;$

$M_2 = 2^2 = 4$ $d = (4 - 1{,}247) = 1;$

$M_3 = 2^{2\cdot 3} = 64$ $d = (64 - 1{,}247) = 1;$

$M_4 = 2^{2\cdot 3\cdot 4} = 16777216$ $d = (16777216 - 1{,}247) = 13.$

т. к. $16777216(\bmod 247) = 235(\bmod 247)$, НОД(235-1,247)=13.

Нетривіальний дільник знайдений.

$$247 = 13 \cdot 19$$

Можна модифікувати алгоритм, використовуючи одночасно кілька різних c. Так як $(c, n) = 1$, то $(c, p) = 1.$

Тому як тільки $(p-1) | M(i),$ то $m_i \equiv 1(\bmod p),$ тобто $p|(m_i - 1)$, і на кроці 2 буде знайдений нетривіальний дільник n.

Параметр k повинен обиратися не дуже великим, інакше можуть бути виконані умови $m_i \equiv 1(\bmod n)$ й дільник не буде знайдений.

При виконанні алгоритму піднесення в степінь $c^{M(i)}$ треба здійснювати в кільці Z_n методом повторного піднесення у квадрат.

Факторизація Ферма

Досить плідною ідеєю при побудові алгоритмів факторизації є пошук чисел x й y, для яких виконується співвідношення $x^2 \equiv y^2 (\bmod n), x \neq \pm y (\bmod n).$

Якщо при цьому $x \neq \pm y(\bmod n)$, то числа НСД$(x + y, n)$ й НСД$(x - y, n)$ суть нетривіальні дільники числа n.

Безумовно , першим у цьому напрямку є метод факторизації, застосований П. Ферма. Він заснований на теоремі Ойлера про подання числа у вигляді різниці двох квадратів.

Теорема 11.1

Якщо $n > 1$ – непарне, то існує взаємно однозначна відповідність між розкладаннями на множники $n = a \cdot b$, $a \geq b > 0$ і поданнями у вигляді різниці квадратів $n = x^2 - y^2$, $x > y > 0$.

Ця відповідність має вигляд

$$x = \frac{a+b}{2}, \quad y = \frac{a-b}{2}.$$
$$a = x + y \quad b = x - y.$$

Доведення очевидне.

$$n = a \cdot b = x^2 - y^2 = \left(\frac{a+b}{2}\right)^2 - \left(\frac{a-b}{2}\right)^2 = \frac{a^2 2ab + b}{4} - \frac{a^2 - 2ab + b^2}{4} =$$

$$= \frac{4ab}{4} = ab.$$

Метод Ферма полягає в тім, що при малих значеннях параметра y в поданні $n = x^2 - y^2$ можна знайти пару (x, y), перебираючи як кандидатів на значення числа x $\lfloor \sqrt{n} \rfloor + 1, \lfloor \sqrt{n} \rfloor + 2, \ldots$ й перевіряючи для кожного з них рівності $(\lfloor \sqrt{n} \rfloor + i)^2 - n = y^2$.

Для відбраковування помилкових значень x можна скористатися тим, що якщо число не є квадратом, то воно з великою ймовірністю не буде й квадратичним відрахуванням для одного з невеликих простих чисел p.

Остання властивість легко перевіряється шляхом обчислення відповідного символу Лежандра.

Алгоритм Ферма

Вхід: $n -$ непарне число, $p_1, \ldots, p_k -$ невеликі прості числа.

Крок 0. Перевірити $p_i | n, i = 1, \ldots, k$. Якщо так, то дільник знайдений.

Крок 1. Для кожного x від $\lfloor \sqrt{n} \rfloor + 1$ до $\lfloor \sqrt{n} \rfloor + n_0$ обчислити величини

$$t = x^2 - n, t_i = t \bmod p_i, i = 1, \ldots, k.$$

Крок 2. Якщо хоча б для одного $i = 1,\ldots,k$ виконано одне з умов:

$-\ t_i = 0$ і p_i^2 не ділить t;

або

$$-\ t_i \neq 0 \text{ і } \left(\frac{t_i}{p_i}\right) = -1,$$

то перейти до наступного x на кроці 1.

У противному випадку перейти до кроку 3.

Крок 3. Перевірити , чи є $t = x^2 - n$ повним квадратом. Якщо $x^2 - n = y^2$, то видати відповідь: «$n-$ складене, ».

Відповідь буде $n = a \cdot b$, де a = НСД$(x + y, n)$, b = НСД $(x - y, n)$.

Якщо $t = x^2 - n$ - не повний квадрат, то перейти до наступного x на кроці 1.

Наприклад

Нехай потрібно факторизувати число $n = 377$. Вхід $n = 377, p_1 = 2, p_2 = 3, p_3 = 5, p_4 = 7, p_5 = 11.$

Крок 0. 377 не ділиться на жодне з p_i.

Крок 1. $\lfloor \sqrt{n} \rfloor = \lfloor \sqrt{377} \rfloor = 19$

$x = \lfloor \sqrt{n} \rfloor + 1$

$x_1 = 20$

$x_2 = 21$

$x_3 = 22 \ldots$

$t = x_1^2 - n = 400 - 377 = 23$

$t_{i1} = 23 (\text{mod}\, 2) = 1$

$\left(\frac{t_{i1}}{p_i}\right) = \left(\frac{1}{2}\right) = \left(\frac{1}{3}\right) = \left(\frac{1}{5}\right) = \left(\frac{1}{7}\right) = \left(\frac{1}{11}\right) = 1.$(т. к. $\left(\frac{1}{p}\right) = 1$);

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}};$$

$t_{i2} = 23(\mathrm{mod}\,3) = 2$

$$\left(\frac{t_{i2}}{p_1}\right) = \left(\frac{2}{3}\right) = (-1)^{\frac{3^2-1}{8}} = -1$$

Виконано умову $-t_i \neq 0$ й $\left(\frac{t_i}{p_i}\right) = -1,$ переходимо до наступного $x.$

$t = 21^2 - 377 = 64$

$t_{i1} = 64(\mathrm{mod}\,2) = 0$ $\quad 64{:}2^2 = 16$ ($t_i = 0$,але p_i^2 ділить t)

$t_{i2} = 64(\mathrm{mod}\,3) = 1$ $\quad \left(\frac{t_{i2}}{p_i}\right) = \left(\frac{1}{2}\right) = \left(\frac{1}{3}\right) = \left(\frac{1}{5}\right) = \left(\frac{1}{7}\right) = \left(\frac{1}{11}\right) = 1;$

$t_{i3} = 64(\mathrm{mod}\,5) = 4$ $\quad \left(\frac{t_{i3}}{p_i}\right) = \left(\frac{4}{3}\right) = \left(\frac{4}{5}\right) = \left(\frac{4}{7}\right) = \left(\frac{4}{11}\right) = 1$ (т. к. $\left(\frac{a^2}{p}\right) = 1$);

$t_{i4} = 64(\mathrm{mod}\,7) = 1$ $\quad \left(\frac{t_{i4}}{p_i}\right) = \left(\frac{1}{2}\right) = \left(\frac{1}{3}\right) = \left(\frac{1}{5}\right) = \left(\frac{1}{7}\right) = \left(\frac{1}{11}\right) = 1;$

$t_{i5} = 64(\mathrm{mod}\,11) = 9$

$$\left(\frac{t_{i5}}{p_i}\right) = \left(\frac{9}{2}\right) = \left(\frac{9}{5}\right) = \left(\frac{9}{7}\right) - \left(\frac{9}{11}\right) = 1$$

Умови $-t_i = 0$ й p_i^2 не ділить t; або

$-t_i \neq 0$ і $\left(\frac{t_i}{p_i}\right) = -1,$ не виконуються ні для одного $i = 1,\ldots,k.$

Переходимо до кроку 3.

Перевіряємо, чи є $t = x^2 - n$ повним квадратом. $t = 21^2 - 377 = 64 = 8^2$

n - складене, , $a = x + y = 21 + 8 = 29, b = x - y = 21 - 8 = 13.$

По суті даний алгоритм , подібно методу «пробних ділень» , є різновидом методу перебору всіх можливих дільників. Параметр n_0 визначається з конкретних обчислювальних можливостей.

Чим більше значення n_0, тим більше число можливих дільників буде перевірено.

Однак, на відміну від методу «пробних ділень» за допомогою якого перебуває найменший дільник, даний метод дозволяє знаходити найбільший дільник числа n, не переважаючий $\sqrt{n}.$

Помітимо, що замість $\sqrt{n}$ можна брати $\sqrt{kn}$ при малих значеннях k. У багатьох випадках перехід до рівнянь $(\lfloor \sqrt{kn} \rfloor + i)^2 - kn = x^2$, $k = 3,5,\ldots.$ дозволяє знайти шукане значення x шляхом більш короткого перебору значень i, чим при $k = 1.$

При цьому, у випадку успіху, таке подання також дозволяє факторизувати число n, тому що числа $(\lfloor \sqrt{kn} \rfloor + i + x), (\lfloor \sqrt{kn} \rfloor + i - x)$ лежать в інтервалі $(0, n)$ й задовольняють рівності

$$(\lfloor \sqrt{kn} \rfloor + i + x), (\lfloor \sqrt{kn} \rfloor + i - x) = kn.$$

Алгоритм Діксона

У багатьох сучасних алгоритмах факторизації для знаходження дільників використовується ідея Лежандра (1798 р.), що полягає в пошуку чисел x й y, що задовольняють умовам

$$x^2 \equiv y^2 \pmod{n}, x \neq \pm y \pmod{n}.$$

Цей підхід є узагальненням методу Ферма, у якому потрібне виконання строгої рівності. Для пошуку таких чисел використається поняття факторної бази.

Назвемо факторною базою деяка множина $B = \{p_1, p_2, \ldots, p_h\}$ невеликих простих чисел. Звичайно в якості $\{p_1, p_2, \ldots, p_h\}$ беруть прості числа, що не перевершують деякої границі $M, h = \pi(M)$.

Будемо говорити, що ціле число $b \in N \in B$ - числом, якщо число $b^2 \bmod n$ розкладається в добуток простих чисел з факторної бази.

$$b^2 \bmod n = \prod_{p \in B} p^{\alpha_p(b)}.$$

Зіставимо кожному B - числу вектор показників із цього розкладання

$$\vec{\alpha}(b) = (\alpha_{p_1}(b), \ldots, \alpha_{p_h}(b)),$$

а також двійковий вектор, отриманий з вектора $\vec{\alpha}(b)$ приведенням всіх його координат за модулем 2,

$$\vec{\varepsilon}(b) = (\alpha_{p_1}(b) \bmod 2, \ldots, \alpha_{p_h}(b) \bmod 2).$$

Якщо тепер яким-небудь способом підібрати таку множину різних B - чисел $b_1, \ldots, b_m$, при якому виконується лінійне співвідношення

$$\vec{\varepsilon}(b_1) \oplus \ldots \oplus \vec{\varepsilon}(b_m) = \vec{0},$$

те для добутку $x = b_1 \ldots b_m$ виконується співвідношення

$$x^2 \equiv y^2 \pmod n,$$

де число y визначається по векторах показників рівністю

$$y = \prod_{p \in B} p^{\frac{1}{2}(\vec{\alpha}_p(b_1) + \ldots + \vec{\alpha}_p(b_m))}.$$

Алгоритм Діксону полягає в наступному.

Крок 1. Вибрати випадкове $b, 1 < b < n$, і обчислити $b^2 \bmod m$.

Крок 2. Пробними діленнями спробувати розкласти $b^2 \bmod m$ на прості множники з факторної бази.

Крок 3. Якщо B є B - числом, тобто $b^2 \bmod n = \prod_{p \in B} p^{\alpha_p}(b)$,

те запам'ятати $\vec{\alpha}(b)$ й $\vec{\varepsilon}(b)$. Повторити процедуру генерації чисел b доти не буде знайдено $m = h + 1$ B - чисел $b_1, \ldots, b_m$.

Крок 4. Знайти, наприклад вирішуючи за допомогою алгоритму послідовного виключення невідомих Гауса, однорідну систему лінійних рівнянь $x_1 \vec{\varepsilon}(b_1) \oplus \ldots \oplus x_m \vec{\varepsilon}(b_m) = \vec{0}$ щодо невідомих $(x_1, \ldots, x_m))$ співвідношення лінійної залежності

$$\vec{\varepsilon}(b_{i_1}) \oplus \ldots \oplus \vec{\varepsilon}(b_{i_t}) = \vec{0}, 1 < t \le m.$$

Покласти

$$x = b_{i_1} \ldots b_{i_{t_y}}, \qquad y = \prod_{p \in B} p^{\frac{1}{2}(\vec{\alpha}_p(b_1) + \ldots + \vec{\alpha}_p(b_m))}.$$

Крок 5. Перевірити $x \equiv \pm y (\bmod n)$. Якщо це так, то повторити процедуру генерації. Якщо ні, то знайдено нетривіальне розкладання

$$n = u \cdot v, u = (x + y, n), v = (x - y, n).$$

Наприклад

Нехай потрібно факторизувати число $n = 143$.

Нехай $B = 17$. Фактор база 2,3,5.

1. $B^2 = 289 = 3 + 2 \cdot 143 = 3 (\bmod 143)$ $S_{17} = 3$

$17^2 \equiv 2^0 \cdot 3^1 \cdot 5^0 (\bmod 143)$

$(b+1)^2 = b^2 + 2b + 1$

2. $B = 18$

$18^2 \equiv 17^2 + 2 \cdot 17 + 1 \equiv 3 + 35 = 38 = 2 \cdot 19 (\bmod 143)$ $S_{18} = 2 \cdot 19$

3. $B = 19$

$19^2 = 18^2 + 2 \cdot 18 + 1 = 38 + 37 = 75 (\bmod 143)$

$19^2 = 2^0 \cdot 3^1 \cdot 5^2$

$S_{19} = 75$

4. $B = 20$

$20^2 = 19^2 + 2 \cdot 19 + 1 \equiv 75 + 39 = 114 (\mathrm{mod}\, 143) = 2 \cdot 3 \cdot 19 (\mathrm{mod}\, 143)$

$S_{20} = 2 \cdot 3 \cdot 19$

5. $B = 21$

$21^2 = 20^2 + 2 \cdot 20 + 1 \equiv 114 + 41 = 155 (\mathrm{mod}\, 143) =$

$= 12 (\mathrm{mod}\, 143) = 2^2 \cdot 3^1 \cdot 5^0 (\mathrm{mod}\, 143)$

$(17 \cdot 19)^2 \equiv 2^0 \cdot 3^2 \cdot 5^2 (\mathrm{mod}\, 143)$

$x = 17 \cdot 19$

$y = 3 \cdot 5$

$\mathrm{НСД}(17 \cdot 19 - 3 \cdot 5, 143) = 11$

$\mathrm{НСД}(17 \cdot 19 + 3 \cdot 5, 143) = 13$

$(19 \cdot 21)^2 \equiv 2^2 \cdot 3^2 \cdot 5^2 (\mathrm{mod}\, 143)$

Однак не завжди така комбінація дає відповідь. Наприклад розглянемо:

$x = 19 \cdot 21$

$y = 2 \cdot 3 \cdot 5$

$\mathrm{НСД}(19 \cdot 21 - 2 \cdot 3 \cdot 5, 143) = 1$

$\mathrm{НСД}(19 \cdot 21 - 2 \cdot 3 \cdot 5, 143) = 143$

На трудомісткість алгоритму Діксону істотно впливає вибір факторної бази. Якщо число h обране так, що $M \approx \frac{\sqrt{n}}{2}$, то майже кожне b буде B-числом, але одержувані при цьому порівняння $x^2 \equiv y^2 (\mathrm{mod}\, n)$ будуть тривіальними.

Крім того, потрібно буде вирішувати системи від дуже великого числа невідомих. Якщо h - мало, то B - числа будуть з'являтися дуже рідко.

Помітимо, що якщо n - складене, то рівняння $x^2 \equiv a^2 (\mathrm{mod}\, n)$ має принаймні 4 рішення. Тому ймовірність появи пари (x, y) з $x \equiv \pm y (\mathrm{mod}\, n)$ не перевершує 1/2.

Отже, повторюючи процедуру набору для одержання потрібної пари k раз ми одержимо, що надійність даного методу знаходження дільника не менше, ніж $1 - 2^{-k}$.

Алгоритм Діксону може бути вдосконалений по декількох напрямках:

можна замінити процедуру генерації B - чисел так, щоб імовірність їхнього породження була більшою;

– можна оптимізувати вибір факторної бази для того, щоб зменшити число невідомих у системі рівнянь;

– можна вдосконалити процедуру відсівання «поганих» чисел b, що не є B-числами;

– можна використати швидкий алгоритм рішення системи лінійних рівнянь (наприклад, алгоритм Відемана для розріджених матриць), і т. і.

Контрольні питання та задачі

1. Алгоритми факторизації в криптографії.
2. Які алгоритми факторизації вам відомі?
3. Метод факторизації Полларда.
4. Метод факторизації Ферма.
5. Метод факторизації Діксонаю
6. Метод факторизації Полларда.
7. Символ Лежандра та його властивості.
8. Метод Полларда факторизувати число *N*= 299, *K*= 391.
9. Методом Полларда фауторизувати число *N*= 221, *K*= 187.
10. Метод Ферма факторизувати *N*= 323, *K*= 143.
11. Методом Діксона факторизувати число *N*= 221, *K*= 391.
12. Напишіть програми для реалізації алгоритмів факторизації.

Розділ 3 Криптосистеми на еліптичних кривих

Лекція 12

Еліптичні криві та операції у групах точок еліптичних кривих

Еліптичні криві застосовуються в криптографії з 1985 року, причому як для факторизації чисел і перевірки простоти, так і для побудови криптографічних протоколів. Інтерес до них обумовлений, з одного боку, тим, що вони є багатим джерелом кінцевих абелевих груп, що володіють корисними структурними властивостями, так і тим, що на їх основі забезпечуються ті ж криптографічні властивості, якими володіють числові і поліноміальне криптосистеми, але при істотно меншому розмірі ключа.

Нагадаємо, що еліптичною кривою E над полем F називається гладка крива, що задається рівнянням вигляду

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, a_i \in F. \qquad (12.1)$$

Іноді замість (10.1) зручно користуватися рівнянням алгебри для функції двох змінних.

$$F(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6 = 0. \qquad (12.2)$$

Крива E називається сингулярною (особливою), якщо існує хоча б одна точка з координатами (x, y), в якій приватні похідні функції (12.2) одночасно звертаються в 0, тобто $\partial F / \partial x = \partial F / \partial y = 0$.

Інакше крива E називається несингулярної (неособливою).

Позначатимемо ε F безліч точок (x, y) ∈ F^2 , які задовольняють цьому рівнянню і що містить крім того нескінченно видалену точку. Точка на нескінченості позначається О. Якщо K - розширення поля F, то ε K позначає безліч точок (x, y) ∈ K^2, що задовольняють рівнянню разом з точкою О. Щоб крива, що задається рівнянням (12.1) була еліптичною кривою в F^2 або K^2, вона

повинна бути гладкою (несингулярною). Це означає, що в F^2 або в K^2 не повинно бути точок, в яких рівні нулю обидві приватні похідні. Іншими словами два рівняння

$$-a_1 y + 3x^2 + 2a_2 x + a_4 = 0$$

$$2y + a_1 x + a_3 = 0$$

не повинні задовольнятися ні в одній точці $(x, y) \in E(F^2)$ або $(x, y) \in E(K^2)$.

З рівнянням еліптичної кривої можна зв'язати поняття дискримінанту

$$d = -16\,(4a^2 + 27b^2).$$

Еліптична крива над полем R (дійсних чисел) з ненульовим дискримінантом є гладкою кривою (несингулярною), в кожній точці якої можна провести дотичну. Замість загального запису рівняння часто розглядають канонічні рівняння трьох типів кривих

$$E : y^2 = x^3 + ax + b, p \neq 2,3;$$
$$E_S : y^2 + y = x^3 + a_4 x + b, p = 2;$$
$$E_N : y^2 + xy = x^3 + a_2 x^2 + a_6, a_6 \neq 0, p = 2,$$

де p – характеристика поля.

Нехай E еліптична крива над полем дійсних чисел, задана рівнянням (12.2), і нехай P і Q - дві точки на еліптичній кривій. Визначимо протилежний елемент до P (тобто зворотний елемент) і суму $P + Q$ за наступним правилом::

1) Якщо P є точкою O , то ми визначимо $-P$ як O. Для кожної точки Q ми вважатимемо $Q + O = Q$, тобто точка O виконує роль одиниці по додаванню;

2) Точка $-P$ є точкою з тією ж x - координатою, але із запереченням y-координати, тобто – (x, y) = (x,–y). Якщо $Q = -P$, то ми визначаємо суму $P + Q$ як точку на нескінченності O;

3) Якщо P і Q мають різні x-координати, то можна показати, що лінія, що проходить через P та Q, перетинає криву тільки в одній точці R. (Ця точка

може співпадати з P або Q і тоді пряма є дотичною до кривої в точці P або Q і тоді ми вважатимемо $R = P$ або $R = Q$ відповідно).

Потім ми визначаємо $P = Q$ як $-R$, тобто як дзеркальне щодо осі x відображення точки перетину R;

4) Остання можливість це $P = Q$. Нехай L є дотичною до кривої в точці P , та нехай R – єдина точка перетину прямої з еліптичною кривою, тоді ми вважатимемо, що $2P = -R$. (В цьому випадку точка P є точкою інфлекції).

Сума будь-яких трьох точок на еліптичній кривій, що належать одній прямій, рівна O.

Визначимо групову операцію, традиційно звану додаванням точок еліптичної кривої. Сумою двох точок $P = (x_1, y_1)$ и $Q = (x_2, y_2)$ називається точка $R = P + Q = (x_3, y_3)$, зворотна третій точці перетину EC прямою лінією, що проходить через точки P та Q. Знайдемо координати точки $R = P + Q = (x_3, y_3)$. Будемо виражати їх через координати точок P та Q. При цьому точки P та Q можуть бути різними або співпадати. Ми опустимо доведення цих формул.

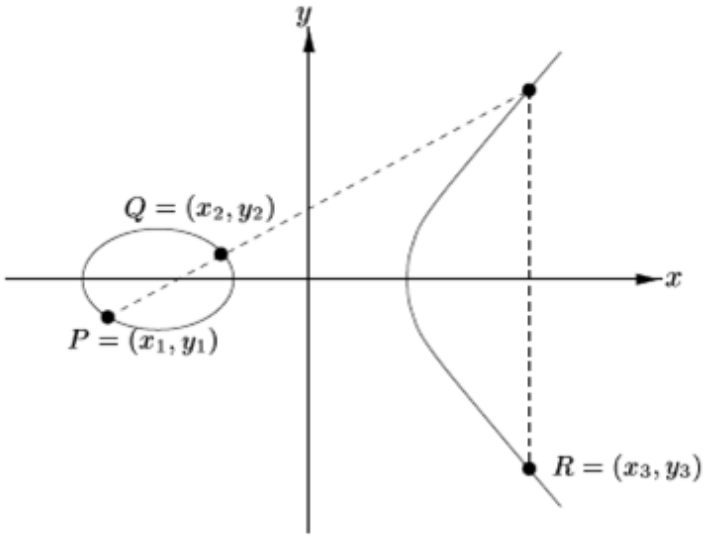

Рис.12.1 – Геометрична інтерпретація додавання двох різних точок P та Q

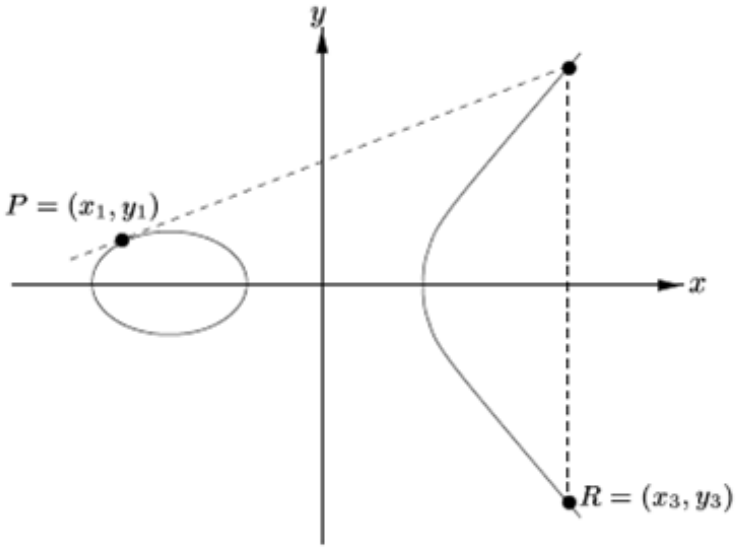

Рис.12.2 ─ Геометрична інтерпретація подвоєння точки кривої

Закони додавання та подвоєння точок для кривої E

$$1.\begin{cases} x_3 = \lambda^2 - x_1 - x_{2,} P \neq Q; \\ y_3 = -y_1 - \lambda(x_3 - x_1), \lambda = \dfrac{y_2 - y_1}{x_2 - x_1} \end{cases} \tag{12.3}$$

$$2.\begin{cases} x_3 = v^2 - 2x_1, P = Q; \\ y_3 = -y_1 - v(x_3 - x_1), v = \dfrac{3x_1^2 + a}{2y_1} \cdot \end{cases} \tag{12.4}$$

Ці формули справедливі для кривих E над всіма полями, у тому числі і кінцевими, окрім полів характеристик 2 і 3.

Закони додавання та подвоєння точок для кривої E_S

$$1.\quad \begin{cases} x_3 = \lambda^2 + x_1 + x_{2,} P \neq Q; \\ y_3 = y_1 + 1 + \lambda(x_3 + x_1), \lambda = \dfrac{y_1 + y_2}{x_1 + x_2} \end{cases} \tag{12.5}$$

$$2.\quad \begin{cases} x_3 = v^2, v = x_1^2 + a_4, P = Q; \\ y_3 = y_1 + 1 + v(x_3 + x_1). \end{cases} \tag{12.6}$$

Подвоєння точок кривої E_S істотно простіше додавання різних точок, оскільки в даному випадку немає обчислення зворотного елемента. Для великих полів ця операція досить трудомістка і може зажадати в десятки разів більше обчислень ніж множення. В цілому операції з точками кривої E_S порівняно прості, проте ці криві є криптографічно слабкими.

Закони додавання та подвоєння точок для кривої E_N

$$1.\begin{cases} x_3 = \lambda^2 + \lambda + x_1 + x_2 + a_2, P \neq Q; \\ y_3 = y_1 + x_3 + \lambda(x_3 + x_1), \lambda = \dfrac{y_1 + y_2}{x_1 + x_2}. \end{cases} \tag{12.7}$$

$$2.\begin{cases} x_3 = v^2 + v + a_2, v = x_1 + \dfrac{y_1}{x_1}, P = Q; \\ y_3 = y_1 + x_3 + v(x_3 + x_1) = x_1^2 + x_3(v+1). \end{cases} \tag{12.8}$$

Як бачимо, у всіх формулах додавання точок (окрім формули для подвоєння точки кривої E_S) є одна операція ділення (вона зводиться до інверсії або визначення зворотного елемента поля з подальшим множенням). Це сама трудомістка операція при багатократному додаванні точок.

Якщо ми запишемо рівняння кривої в афінних координатах: $E_N : y^2 + xy = x^3 + a_2 x^2 + a_6, a_{2,6} \in F$, координату x_3 в (12.8) можна також виразити інакше

$$x_3 = x_1^2 + a_6 x_1^{-2}.$$

Простіше всього вона обчислюється для кривих Коблиця з коефіцієнтом $a_6 = 1$, при цьому послідовно виконуються три операції: інверсія x_1^{-1}, додавання

і зведення в квадрат. Помітимо, що останнє співвідношення може бути корисним при криптоаналізі методом послідовного подвоєння точок кривої без визначення y-координат точок.

Його також можна використовувати при рішенні зворотної задачі: визначення координати x_1 точки ділення на два при відомому значенні x_3.

Помітимо також, що при вивченні властивостей EC у ряді практичних додатків часто корисним виявляється перехід від афінних координат (x, y) до проективних (X, Y, Z), зв'язуючи точки кривої E в цих координатах відношенням еквівалентності. При цьому в операціях над кінцевими полями вдається уникнути трудомістких обчислень зворотного елемента при додаванні точок.

В афінних координатах рівняння кривої $E: y^2 = x^3 + ax + b, p \neq 2,3$ запишеться як

$$F(x, y) = y^2 - x^3 - ax - b, a, b \in K.$$

Формально введемо нові змінні підстановкою

$$x = X/Z, y = Y/Z, z = Z/Z = 1. \text{ Тоді при } Z \neq 0$$

$$F(x, y) = \left(\frac{Y}{Z}\right)^2 - \left(\frac{X}{Z}\right)^3 - a\left(\frac{X}{Z}\right) - b = 0.$$

Умножаючи це рівняння на Z^3, маємо

$$F^*(X, Y, Z) = Z^3 F(x, y) = Y^2 Z - X^3 - aXZ^2 - bZ^3 = 0. \qquad (12.9)$$

Це так зване однорідне кубічне рівняння (все мономи 3-їй ступені). Виключаючи з розгляду початок координат (0,0,0), Для будь-якої трійки відношення (X, Y, Z) еквівалентності задається проективної точкою $(\lambda X, \lambda Y, \lambda Z)$, де λ-скаляр, X, Y, Z - фіксовані.

В тривимірному просторі цей клас є прямою лінією, що проходить через початок координат. При $Z \neq 0$ будь-яку таку пряму перетинає площина $Z = 1$, в якій ми повертаємося до запису кривої в афінних координатах. При $\lambda = Z^{-1}$

еквіваленти точки $(X,Y,Z) \approx (\lambda X, \lambda Y, \lambda Z) = (x, y, 1)$, де остання належить площині $Z = 1$.

Проективні координати часто називають проективною площиною і позначають P_k^2. Це зв'язано з тим, що точки всього простору проектуються відношенням еквівалентності на площину, площини – на прямі, а прямі – в точки. Нескінченно видалена пряма в P_k^2 задається рівнянням $Z = 0$. Єдиною точкою, що задовольняє при цьому рівнянню (12.9) над будь-яким полем K, є точка $O = (0,1,0)$. В проективній площині вона задає координати нескінченно видаленої точки або нульового елемента групи E.

Точка O є третьою точкою перетину точок P і $-P$ на нескінченності.

Наприклад, кривій

$$y^2 = x^3 - x = x(x-1)(x+1)$$

проективних координатах відповідає крива

$$Y^2 Z = X^3 - XZ^2 = X(X-Z)(X+Z).$$

Нехай $P = (-1/2, \sqrt{3/8})$. Пряма, яка проходить через точки P і $-P$, задається в афінних координатах як $x = -1/2$, і в проективних як $X = -Z/2$.

Підставимо останню рівність в рівняння кривої, тоді $Y^2 Z = 3Z^3/8$. Це рівняння має три рішення: $Z = 0, Y = \pm\sqrt{3/8}Z$.

Таким чином, пряма $X = -Z/2$ перетинає криву в проективних координатах в трьох точках: $(-1/2, \sqrt{3/8}, 1); (-1/2, -\sqrt{3/8}, 1); (0,1,0)$. Остання точка - це точка на нескінченності O. Проективна площина дозволяє задати координати цієї точки.

При обчисленні точок з багатократними операціями додавання (віднімання) і подвоєння часто більш продуктивні групові операції не в афінних координатах, а різного роду проективних координатах. Це дозволяє уникнути

обчислення зворотного елемента в полі як самої трудомісткої операції і заощадити тимчасові обчислювальні ресурси. В стандартних проективних координатах проективна точка (X, Y, Z), $Z \neq 0$, відповідає афінній точці $(x = X/Z, y = Y/Z)$. Однорідне рівняння кривої після заміни змінних і множення на Z^3 приймає вигляд $E_j : Y^2 Z = X^3 + aXZ^2 + bZ^3 \pmod p$.

Точка на нескінченності $O = (0:1:0)$ є вже одним з рішень даного рівняння. Зворотна точка тут, як і раніше, визначається інверсією знаку Y – координати: $-P = (X:-Y:Z)$. Подібно тому, як в афінних координатах, додавання точок $P_1 = (X_1 : Y_1 : Z_1)$ і $P_2 = (X_2 : Y_2 : Z_2)$ при $P_1 \neq \pm P_2$ називається точка $P_3 = P_1 + P_2 = (X_3 : Y_3 : Z_3)$, координати якої (позначення $\pmod p$ надалі опускається для скорочення запису) рівні:

$$P_1 \neq \pm P_2 : X_3 = VG;$$
$$Y_3 = u(v^2 X_1 Z_2 - g) - v^3 Y_1 Z_2;$$
$$Z_3 = v^3 Z_1 Z_2,$$

де $v = X_2 Z_1 - X_1 Z_2; u = Y_2 Z_1 - Y_1 Z_2; g = u^2 Z_1 Z_2 - v^3 - 2v^2 X_1 Z_2$.

Операцію додавання двох однакових точок $P_1 = P_2$ називають подвоєнням, а координати точки $P_3 = 2P_1 = (X_3 : Y_3 : Z_3)$ рівні:

$$P_1 = P_2 : X_3 = 2hs;$$
$$Y_3 = w(4d - h) - 8s^2 Y_1^2;$$
$$Z_3 = 8s^3,$$

де $w = aZ_1^2 + 3X_1^2; s = Y_1 Z_1; d = sX_1 Y_1; h = w^2 - 8d$.

Наступний вид проективних координат - якобіанові координати.

До них можна перейти ізоморфним перетворенням координат, помноживши рівняння $y^2 \equiv x^3 + ax + b \pmod p$ на Z^6, при цьому отримаємо:

$(yZ^3)^2 = (xZ^2)^3 + aZ^4(xZ^2) + bZ^6$ або

$$E_j : Y^2 = X^3 + aXZ^4 + bZ^6 \pmod p,$$

де $y = Y/Z^3$, x $= X/Z^2$.

Додаванням точок $P_1 = (X_1 : Y_1 : Z_1)$ і $P_2 = (X_2 : Y_2 : Z_2)$ при $P_1 \neq \pm P_2$ є точка $P_3 = P_1 + P_2 = (X_3 : Y_3 : Z_3)$, координати якої визначаються як:

$$P_1 \neq \pm P_2 : X_3 = -H^3 - 2U_1H^2 + r^2;$$
$$Y_3 = -S_1H^3 + r(U_1H^2 - X_3);$$
$$Z_3 = HZ_1Z_2,$$

де $U_1 = X_1Z_2^2; U_2 = X_2Z_1^2; S_1 = Y_1Z_2^3; S_2 = Y_2Z_1^3; H = U_2 - U_1; r = S_2 - S_1.$

При подвоєнні точки кривої отримаємо $P_3 = 2P_1 = (X_3 : Y_3 : Z_3)$:

$$P_1 = P_2 : X_3 = T;$$
$$Y_3 = -8Y_1^4 + M(S - T);$$
$$Z_3 = 2Y_1Z_{1,}$$

де $S = 4X_1Y_1^2; M = 3X_1^2 + aZ_1^4; T = -2S + M^2.$

Замість трьох якобіанових координат точки Чудновський запропонував використовувати п'ять: $(X : Y : Z : Z^2 : Z^3)$. Рівняння кривої описується формулою $E_j : Y^2Z = X^3 + aXZ_2 + bZ^3 (\mathrm{mod}\, p)$, а сума точок $P_1 = (X_1 : Y_1 : Z_1 : Z_1{}^2 : Z_1{}^3)$ и $P_2 = (X_2 : Y_2 : Z_2 : Z_2{}^2 : Z_2{}^3)$ при $P_1 \neq \pm P_2$ визначається як точка $P_3 = P_1 + P_2 = (X_3 : Y_3 : Z_3 : Z_3^2 : Z_3^3)$, координати Чудновського якій рівні:

$$P_1 \neq \pm P_2 : X_3 = -H^3 - 2U_1H^2 + r^2;$$
$$Y_3 = -S_1H^3 + r(U_1H^2 - X_3);$$
$$Z_3 = HZ_1Z_2;$$
$$Z_3^2 = Z_3^2;$$
$$Z_3^3 = Z_3^3,$$

де $U_1 = X_1Z_2^2; U_2 = X_2Z_1^2; S_1 = Y_1Z_2^3; S_2 = Y_2Z_1^3; H = U_2 - U_1; r = S_2 - S_1.$

При подвоєнні точки кривої отримаємо

$$P_3 = 2(X_1 : Y_1 : Z_1 : Z_1^2 : Z_1^3) = (X_3 : Y_3 : Z_3 : Z_3^2 : Z_3^3):$$

$$P_1 = P_2 : X_3 = T;$$
$$Y_3 = -8Y_1^4 + M(S - T);$$
$$Z_3 = 2Y_1 Z_{1,}$$
$$Z_3^2 = Z_3^2 :$$
$$Z_3^3 = Z_3^3,$$

де $S = 4X_1 Y_1^2; M = 3X_1^2 + aZ_1^4; T = -2S + M^2.$

Модифіковані якобіанові координати для рівняння

$E_j : Y^2 Z = X^3 + aXZ_2 + bZ^3 (\mathrm{mod}\, p)$ кривій містять чотири координати $(X : Y : Z : aZ^4)$. Додаванням точок $P_1 = (X_1 : Y_1 : Z_1 : aZ_1{}^4)$ і $P_2 = (X_2 : Y_2 : Z_2 : aZ_2{}^4)$ при $P_1 \neq \pm P_2$ визначається як точка $P_3 = P_1 + P_2 = (X_3 : Y_3 : Z_3 : aZ_3^4)$, модифіковані якобіанові координати якої рівні:

$$P_1 \neq \pm P_2 : X_3 = -H^3 - 2U_1 H^2 + r^2;$$
$$Y_3 = -S_1 H^3 + r(U_1 H^2 - X_3);$$
$$Z_3 = HZ_1 Z_2;$$
$$aZ_3^4 = aZ_3^4,$$

де $U_1 = X_1 Z_2^2; U_2 = X_2 Z_1^2; S_1 = Y_1 Z_2^3; S_2 = Y_2 Z_1^3; H = U_2 - U_1; r = S_2 - S_1.$

Мінімальна обчислювальна складність додавання досягається за допомогою координат Чудновського, а подвоєння – в модифікованих якобіанових координатах. \

Тому звичайно користуються змішаними координатами з метою оптимізації обчислень при багатократному додаванні точки. Після обчислення точки kP в змішаних координатах необхідно повернутися в афінні координати, для чого в кінці обчислень потрібна одна інверсія.

Контрольні питання та задачі

1. Дати визначення ЕК.

2. Який загальний вигляд має крива, що використовується в криптографічних системах, заснованих на еліптичних кривих?

3. Дайте визначення порядку групи точок еліптичної кривої.

4. Дайте визначення порядку точки еліптичної кривої.

5. Яка математична проблема забезпечує стійкість криптосистем, побудованих на еліптичних кривих?

6. Які основні операції виконуються над точками еліптичних кривих при їх використанні в криптографічних системах?

7. Запишіть та поясніть формулу додавання точок еліптичної кривої над простим полем.

8. Запишіть та поясніть формулу подвоєння точок еліптичної кривої над простим полем.

9. Запишіть та поясніть рівняння еліптичної кривої над простим полем GF(P).

10. Запишіть та поясніть рівняння еліптичної кривої над розширеним полем $GF(2^m)$.

11. Запишіть та поясніть рівняння еліптичної кривої в проективному вигляді.

12. Що забезпечує використання проективного базису?

13. Як можна знайти зворотні елементи в полі GF(q)?

14. Порівняйте афінне та проективні подання ЕК.

15. Нехай є ЕК з рівнянням: $y^2 = (x^3 + x + 1) \bmod 23, a = 1, b = 1, p = 23.$ Перевірити, чи належать точки, що наведені в таблиці нижче еліптичній кривавій.

N	1	2	3	4	5	6	7	8
P_n	(3,13)	(3,10)	(4,0)	(5,4)	(5,19)	(6,4)	(6,19)	(7,11)
N	9	10	11	12	13	14	15	16
P_n	(9,16)	(17,3)	(17,20)	(18,20)	(19,5)	(13,16)	(9,7)	(17,3)

Знайти $P_n + Q$, якщо $Q = 2 \cdot P_n$.

16. Знайти порядки точок кривих:

1. $P = (3, 8)$ на $y^2 = x^3 - 43x + 166$
2. $P = (0,4)$ на $y^2 = 4x^3 + 16$
3. $P = (2, 8)$ на $y^2 = 4x^3 + 16x$
4. $P = (2, 3)$ на $y^2 = x^3 + 1$
5. $P = (2, 4)$ на $y^2 = x^3 + 4x$
6. $P = (7,2)$ на $y^2 = x^3 + 3x + 3 \pmod{11}$
7. $P = (7,9)$ на $y^2 = x^3 + 3x + 3 \pmod{11}$
8. $P = (0,6)$ на $y^2 = x^3 + 3x + 3 \pmod{11}$
9. $P = (3, 6)$ на $y^2 = x^3 + x+6 \pmod{11}$
10. $P = (8.3)$ на $y^2 = x^3 + x+6 \pmod{11}$
11. $P = (3, 12)$ на $y^2 = x^3 - 14x^2 + 81x$;
12. $P = (2,2)$ на $y^2 = x^3 +2x + 6 \pmod 7$
13. $P = (2,5)$ на $y^2 = x^3 +2x + 6 \pmod 7$
14. $P = (3,5)$ на $y^2 = x^3 +2x + 6 \pmod 7$
15. $P = (4,1)$ на $y^2 = x^3 +2x + 6 \pmod 7$
16. $P = (4,6)$ на $y^2 = x^3 +2x + 6 \pmod 7$
17. $P = (0,5)$ на $y^2 = x^3 + 3x + 3 \pmod{11}$
18. $P = (0, 0)$ на $y^2 + y = x^3 - x^2$.
19. $P = (4,9)$ на $y^2 = x^3 + 17$.
20. $P = (2,5)$ на $y^2 = x^3 + 17$.
21. $P = (-1,4)$ на $y^2 = x^3 + 17$.
22. $P = (2, 2)$ на $y^2 = x^3 +x+ 1 \pmod 7$
23. $P = (2, 2)$ на $y^2 = x^3 + 1 \pmod 5$
24. $P = (1, 3)$ на $y^2 = x^3 + 1 \pmod 7$
25. $P = (3, 4)$ на $y^2 = x^3 + x \pmod 7$

17. Перевірити , чи належать точки кривій. Знайти координати точки $R = P + Q = (x_3, y_3)$ та $2P\ (x_4, y_4)$ на кривій $y^2 = x^3 + 3x+3 \pmod{11}$, якщо

№	1	2	3	4	5	6
P	(7,9)	(7,2)	(8,0)	(9,0)	(0,6)	(0,5)
Q	(5,0)	(9,0)	(0,5)	(7,9)	(7,2)	(7,9)

Лекція 13

Сліди та базиси розширеного поля. Поліноміальний та нормальний базиси

Від ідеї створення ECC до сьогоднішнього дня поряд із криптоаналізом цих систем фахівці безупинно і плідно працюють над підвищенням ефективності ECC.

Насамперед це відноситься до швидкодії криптосистеми або швидкості обчислень . Одним з напрямків робіт у цій сфері було вивчення і порівняльний аналіз арифметики в поліноміальному і нормальному базисах поля F_2^m.

Сліди і базиси розширеного поля

Операції в розширених полях вимагають введення таких понять, як слід елемента поля та базису поля.

Нехай $F = F_p$ - просте поле і $K = F_p^n$- його розширення.

Визначення

Слідом елемента $\alpha \in K$ над полем F називається сума сполучених елементів поля K

$$Tr_{K/F}(\alpha) = \alpha + \alpha^p + \alpha^{p^2} + \ldots + \alpha^{p^{m-1}}.$$

Зокрема, слід елемента над полем F_2 визначається сумою

$$Tr(\alpha) = \alpha + \alpha^2 + \alpha^4 + \ldots + \alpha^{2^{n-1}}.$$

Розширене поле Галуа F_p^n є n - мірним векторним простором над полем F_p. Базисом цього поля називається будь-яка множина з n лінійно незалежних елементів поля $\{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n\}$.

Кожен елемент поля надається як n - мірний вектор з координатами з поля F_p (або поліном степеня $n-1$ з коефіцієнтами з F_p).

Його також можна виразити як лінійну комбінацію векторів базису.

$$\beta = c_1\alpha_1 + c_2\alpha_2 + c_3\alpha_3 + \ldots + c_n\alpha_n, c_i \in F_p.$$

Теорема 13.1

Елементи $\{\alpha_1, \alpha_2, \alpha_3, \ldots, \alpha_n\}$ поля F_p^n утворюють його базис над полем F_p тоді і тільки тоді, коли визначник матриці Вандермонда

$$det(A) = \begin{vmatrix} \alpha_1 & \alpha_2 & \ldots & \alpha_n \\ \alpha_1^p & \alpha_2^p & \ldots & \alpha_n^p \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ \alpha_1^{p^{n-1}} & \alpha_2^{p^{n-1}} & \ldots\ldots\ldots & \alpha_n^{p^{n-1}} \end{vmatrix} \neq 0,$$

або визначник

$$\Delta \quad (\alpha_1, \alpha_2, \ldots, \alpha_n) = \begin{vmatrix} Tr(\alpha_1\alpha_1) & Tr(\alpha_1\alpha_2) & \ldots & Tr(\alpha_1\alpha_n) \\ Tr(\alpha_2\alpha_1) & Tr(\alpha_2\alpha_2) & \ldots & Tr(\alpha_2\alpha_n) \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ Tr(\alpha_n\alpha_1) & Tr(\alpha_n\alpha_2) & \ldots & Tr(\alpha_n\alpha_n) \end{vmatrix} \neq 0.$$

Із множини всяких базисів найбільш розповсюдженими є поліноміальний і нормальний базиси поля F_p^n.

Поліноміальний базис

Поліноміальний базис, звичайно, будується за допомогою послідовних степенів примітивного елементу поля $\{1, \alpha, \alpha^2, \alpha^3, \ldots, \alpha^{n-1}\}$. Його назва зв'язана з тим , що при $x = \alpha$ всі операції в полі здійснюються за модулем мінімального полінома елемента α. Примітивний елемент α тут є утворюючим елементом мультиплікативної групи поля.

Наприклад

Розглянемо поле F_2^4. Елементами цього поля є 16 векторів.

0000	0001	0010	0011	0100	0101	0110	0111
1000	1001	1010	1011	1100	1101	1110	1111

Використовуємо при обчисленнях поліном $f(x) = x^4 + x + 1$ (незвідний)

Додавання:

(0101)+(1101)=(1000).

Множення :

(0101)(1101)=$(x^2+1)\cdot(x^3+x^2+1)(mod\, f\,(x)) = x^5+x^4+x^2+x^3+x^2+1(mod\, f\,(x)) = (x^4+x+1)(x+1)+(x^3+x^2)(mod\, f\,(x)) = x^3+x^2 = 1100$.

Зведення в степінь: $(0010)^2 = (0010)(0010) = x\cdot x(mod\, f\,(x)) = (x^2)(mod\, f\,(x)) = x^2 = (0100)$.

$(0010)^4 = (0010)^2\cdot(0010)^2 = x^2\cdot x^2\big(mod\, f\,(x)\big) =$

$= (x^4+x+1)\cdot 1+(x+1)(mod\, f\,(x)) = x+1 = (0011)$

$(0010)^5 = (0010)^4\cdot(0010) = (0011)\cdot(0010) = (x+1)\cdot x(mod\, f\,(x)) =$

$= (x^2+x)\, mod\, f\,(x) = (x^4+x+1)\cdot 0+(x^2+x)\big(mod\, f\,(x)\big) = x^2+x =$

$= (0110)$

Мультиплікативна інверсія:

Мультиплікативною інверсією для $g^7 = (1011)$ є $g^{-7(mod\,15)} = g^{8(mod\,15)} = (0101)$.

Дійсно $g^7\cdot g^8 = (1011)\cdot(0101) = g^0$.

Нормальний базис

Нормальний базис над полем F_p визначається як множина сполучених елементів поля $N = \{\alpha, \alpha^p, \alpha^{p^2}, \ldots, \alpha^{p^{n-1}}\}$ з підходящим вибором елемента α.

Розглянемо далі властивості НБ $N = \{\beta, \beta^2, \beta^4, \ldots, \beta^{2^{n-1}}\}$ над полем F_2.

На елемент β тут накладається необхідна умова: $Tr(\beta) = 1$. У той же час β не обов'язково є примітивним. У будь-якому полі F_2^n існує елемент зі слідом 1, тому в будь-якому полі існує і НБ. Елементи НБ можна представити n-мірними векторами.

$\beta = 1000\ldots0,$

$\beta = 0100\ldots0,$

$\beta = 0010\ldots0,$

……

$\beta = 0000\ldots1.$

Помітимо, що молодший розряд НБ звичайно записується ліворуч (на відміну від поліноміального, у якому молодший розряд прийнятий записувати праворуч). Кожен наступний елемент базису є циклічним зрушенням праворуч попереднього. Так як $Tr(\beta) = \beta + \beta^2 + \beta^4 + \beta^8 + \ldots + \beta^{2^{n-1}} = 1$, елемент 1 поля F_2^n визначається координатами $1 = (1,1,1,\ldots,1)$. Як бачимо, векторне надання елемента 1 поля F_2^n в поліноміальному і нормальному базисах різні. Для порівняння двійкове надання елементів у поліноміальному і нормальному базисах дані в наступній таблиці.

Таблиця 13.1 - Двійкове надання елементів у поліноміальному і нормальному базисах

α^i	$x^3, x^2, x,\ 1$	$\beta, \beta^2, \beta^4, \beta^8$	α^i	$x^3, x^2, x,\ 1$	$\beta, \beta^2, \beta^4, \beta^8$
0	0000	0000	α^7	1011	1110
1	0001	1111	α^8	0101	0011
α	0010	1001	α^9	1010	0001
α^2	0100	1100	α^{10}	0111	1010
α^3	1000	1000	α^{11}	1110	1101
α^4	0011	0110	α^{12}	1111	0010
α^5	0110	0101	α^{13}	1101	1011
α^6	1100	0100	α^{14}	1001	0111

Довільний елемент поля в базисі N представляється як

$$A = a_0\beta + a_1\beta^2 + a_2\beta^4 + \ldots + a_{n-1}\beta^{2^{n-1}}, a_i \in F_2.$$

Зведення у квадрат елемента A в нормальному базисі дає

$$A^2 = a_0\beta^2 + a_1\beta^4 + a_2\beta^8 + \ldots + a_{n-2}\beta^{2^{n-1}} + a_{n-1}\beta^{2^n}$$
$$= a_{n-1}\beta + a_0\beta^2 + a_1\beta^4 + a_2\beta^8 + \ldots + a_{n-2}\beta^{2^{n-1}}, a_i \in F_2.$$

Таким чином, операція зведення у квадрат або витягу кореня квадратного зводиться до циклічного зрушення вправо (або вліво) векторного представлення елемента. Це одне з важливих технологічних переваг нормального базису перед поліноміальним. Іншим його достоїнством є простота визначення сліду елемента.

Дійсно:

$$Tr(A) = A + A^2 + A^4 + A^8 + \ldots + A^{2^{n-1}} = (a_0 + a_1 + a_2 + \ldots + a_{n-1}) \, mod \, 2.$$

Отже, слід елемента дорівнює 0 при парній вазі його векторного представлення в НБ , або 1 – при непарній вазі. Це властивість радикально спрощує визначення сліду елемента в НБ. Наприклад: елемент $\alpha^4 = 0110$ у нормальному базисі (парна вага векторного надання). Слід елемента дорівнює 0, дійсно $Tr(\alpha) = \alpha + \alpha^2 + \alpha^4 + \ldots + \alpha^{2^{n-1}} = \alpha^4 + \alpha^8 + \alpha^{16} + \alpha^{32} = \alpha^4 + \alpha^8 + \alpha^1 + \alpha^2 = 0011 + 0101 + 0010 + 0100 = 0000.$

Приклад 1

Визначимо елементи поля $F_q, q = 2^4$, прийняв у якості утворюючого незвідний поліном $P(x) = x^4 + x + 1 = 10011$.

Рішення

Нуль поля можна надати вектором 0000, одиницю – 0001. Нехай ненульові елементи поля визначаються як степені x^k, $k = 1,2,3, \ \ldots, 14$. Тоді $x = 0010$, $x^2 = 0100$, $x^3 = 1000$, и далі

$$x^4 \, mod \, P\,(x) = res\{x^4/(x^4 + x + 1)\} = x + 1 = 0011.$$

$$x^5 \, mod \, P\,(x) = res\{x^5/(x^4 + x + 1)\} = x^2 + 1 = 0110 \qquad (*)$$

Подібним чином визначаються всі елементи.

Можна помітити , що при нульовому старшому розряді наступний елемент утворюється зсувом ліворуч попереднього на один розряд з додаванням праворуч нуля. Співвідношення (*) означає, що $x = \alpha$ є коренем рівняння $x^4 + x + 1 = 0$, для котрого $\alpha^4 = \alpha + 1$. Таким чином ненульові елементи поля можна розглядати як степені кореня утворюючого полінома. Згідно з обчисленнями можна заповнити таблицю 11.2.

Таблиця 13.2 - Експоненціальна та векторна форми надання елементів F_2^4

0	0000	α^3	1000	α^7	1011	α^{11}	1110
Я 1	0001	α^4	0011	α^8	0101	α^{12}	1111
α	0010	α^5	0110	α^9	1010	α^{13}	1101
α^2	0100	α^6	1100	α^{10}	0111	α^{14}	1001

Тут $\alpha^{15} = 1$, тобто порядок елемента α максимальний і дорівнює порядку $q - 1 = 15$ мультиплікативної циклічної групи поля. Такий елемент поля

максимального порядку зветься примітивним елементом. Он же є й генератором мультиплікативної групи поля. Примітивний елемент завжди є коренем примітивного незвідного полінома.

Приклад 2

Побудувати нормальний базис поля F_2^4 над полем F_2. Поліноміальне надання поля надано в попередньому завданні. Тобто заповнити третю колонку наступної таблиці.

Таблиця 13.3 - Поліноміальне і нормальне надання елементів поля F_2^4

α^i	$x^3, x^2, x, 1$	$\beta, \beta^2, \beta^4, \beta^8$	α^i	$x^3, x^2, x, 1$	$\beta, \beta^2, \beta^4, \beta^8$
0	0000	0000	α^7	1011	1110
1	0001	1111	α^8	0101	0011
α	0010	1001	α^9	1010	0001
α^2	0100	1100	α^{10}	0111	1010
α^3	1000	1000	α^{11}	1110	1101
α^4	0011	0110	α^{12}	1111	0010
α^5	0110	0101	α^{13}	1101	1011
α^6	1100	0100	α^{14}	1001	0111

У якості утворюючого елементу для переходу до нормального базису потрібно взяти елемент зі слідом 1. Половина елементів в таблиці має слід «0», половина «1».

Для $\beta = \alpha^2$ рахуємо слід за формулою:

$$Tr(\alpha) = \alpha + \alpha^2 + \alpha^4 + ... + \alpha^{2^{n-1}}$$

$$Tr(\alpha^2) = \alpha^2 + \alpha^4 + \alpha^8 + \alpha^{16-15} =$$

$$Tr(\alpha^2) = \alpha^2 + \alpha^4 + \alpha^8 + \alpha^1 = \quad 0100$$

$$0011$$

$$0101$$

$$0010$$

$$0000$$

Елементи виписуємо з таблиці 13.3 (поліноміальне надання).

Додавання за правилом: 0⊕0=0, 0⊕1=1, 1⊕0=1, 1⊕1=0.

Слід $\alpha^2 = 0$. Його неможна взяти за утворюючий.

Перевіряємо α^3.

$$Tr(\beta) = Tr(\alpha^3) = \beta + \beta^2 + \beta^4 + \beta^8 + ... + \beta^{2^{n-1}} = \alpha^3 + \alpha^6 + \alpha^{12} + \alpha^{24} =$$
$$\alpha^3 + \alpha^6 + \alpha^{12} + \alpha^9 = 1000 + 1100 + 1010 + 1111 = 1 \quad)$$

Слід $\alpha^3 = 1$.

Так як $Tr(\alpha^3) = 1$, елемент $\beta = \alpha^3$ можна взяти за утворюючий.

У якості нормального базису потрібно взяти набір сполучених елементів поля. Перевіримо такий набір сполучених елементів $N = \{\beta, \beta^2, \beta^4, \beta^8\}$. ($F_2^4$, p=2).

Розрахуємо визначник матриці Вандермонда за допомогою теореми 13.1.

Елементи $\{\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n\}$ поля F_p^n утворюють базис над полем F_p тоді і тільки тоді, коли визначник матриці Вандермонда

$$\Delta(\alpha_1, \alpha_2, ..., \alpha_n) = \begin{vmatrix} Tr(\alpha_1\alpha_1) & Tr(\alpha_1\alpha_2) & ... & Tr(\alpha_1\alpha_n) \\ Tr(\alpha_2\alpha_1) & Tr(\alpha_2\alpha_2) & ... & Tr(\alpha_2\alpha_n) \\ ..... & ..... & ..... & ..... \\ Tr(\alpha_n\alpha_1) & Tr(\alpha_n\alpha_2) & ... & Tr(\alpha_n\alpha_n) \end{vmatrix} \neq 0.$$

Перевіряємо:

α_1 α_2 α_3 α_4

β, β^2, β^4, β^8

Тут буде подвійна підстановка:

Tr ($\alpha_1\alpha_1$) = Tr (β^2).

Tr ($\alpha_1\alpha_2$) = Tr (β^3).

…

$Tr(\beta^2) = \beta^2 + \beta^4 + \beta^8 + \beta^{16-15} = \beta^2 + \beta^4 + \beta^8 + \beta^1 =$

$= (\alpha^3)^2 + (\alpha^3)^4 + (\alpha^3)^8 + (\alpha^3)^1 = \alpha^6 + \alpha^{12} + \alpha^{24-15} + \alpha^3 =$

$= \alpha^6 + \alpha^{12} + \alpha^9 + \alpha^3 =$

1100

1111

1010

1000

0001

І так далі. Переходимо до визначника з нуликами та одиничками.

Згідно з теоремою:

$$\Delta(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \begin{vmatrix} Tr(\beta^2) & Tr(\beta^3) & Tr(\beta^5) & Tr(\beta^9) \\ Tr(\beta^3) & Tr(\beta^4) & Tr(\beta^6) & Tr(\beta^{10}) \\ Tr(\beta^5) & Tr(\beta^6) & Tr(\beta^8) & Tr(\beta^{12}) \\ Tr(\beta^9) & Tr(\beta^{10}) & Tr(\beta^{12}) & Tr(\beta) \end{vmatrix} = \begin{vmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \end{vmatrix} =$$

$$\begin{vmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{vmatrix} + \begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{vmatrix} + \begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{vmatrix} = 1.$$

Сукупність елементів $N = \{\beta, \beta^2, \beta^4, \beta^8\}, \beta = \alpha^3$,утворюють базис і є системою лінійно незалежних векторів. $N = \{\beta, \beta^2, \beta^4, \beta^8\} -$ можна взяти у якості базису з утворюючим елементом $\beta = \alpha^3$. Для переходу до нормального базису стандартно пишеться 3 рівняння.

1. $M_1(\alpha) = \alpha^4 + \alpha + 1 = 0$

2. $\beta + \beta^2 + \beta^4 + \beta^8 = 1$.

3. $\beta = \alpha^3$

Визначимо елементи нормального базису.

$\alpha^6 = \beta^2$.

$\alpha^{12} = \beta^4$.

$\alpha^{24-16=9} = \beta^8$.

$\alpha^0 = 1 \quad \beta + \beta^2 + \beta^4 + \beta^8 = 1$.

Перше рівняння помножимо почлено на α^2.

$\alpha^6 + \alpha^3 = \alpha^2$.

$\alpha^6 = \beta^2$, а $\alpha^3 = \beta$.

$\alpha^2 = \alpha^6 + \alpha^3 = \beta + \beta^2$.

$\alpha^4 = \beta^2 + \beta^4$.

$\alpha^8 = \beta^4 + \beta^8$.

$\alpha^{16-15=1} = \beta + \beta^8$.

Неважко впевнитися, що зведення у квадрат довільного елемента поля зводиться у НБ к циклічному зсуву праворуч двійкового вектору. Крім того, всі елементи у НБ парної ваги мають слід 0, а непарної ваги – слід 1.

Контрольні питання та задачі

1. Поняття сліду елемента у полі.
2. Теорема про необхідні та достатні умови існування базису поля F_p^n над полем F_p.
3. Як будується поліноміальний базис.
4. Обчислення у поліноміальному базисі (додавання, множення, зведення в степінь, мультиплікативна інверсія).
5. Визначення нормального базису.
6. Зведення у квадрат у нормальному базисі.
7. Визначення сліду елемента у нормальному базисі.
8. Скільки елементів в полі GF(2^m), якщо m = 4, 9, 81, 160?
9. Скільки елементів в полі GF(P), якщо P = 17, 31, 47, 89, 107, 257?
10. Запишіть елементи поля GF(7).
11. Запишіть елементи поля GF(2^3).
12. Побудувати поліноміальне надання у полі F_2^3 . Взяти незвідні поліноми

а) $f(x)=x^3+x+1$;

б) $f(x)=x^3+x^2+1$.

Лекція 14

Оптимальний нормальний базис поля F_2^m

Вже наприкінці 80-х років були сформульовані і доведені теореми, що визначають необхідні і достатні умови існування оптимального нормального базису (ОНБ) у розширеному полі F_p^m.

Використання ОНБ у подальших апаратних реалізаціях ECC підтвердило його переваги у швидкості і технологічності обчислень.

Означення

Нормальний базис $\left\{\beta, \beta^2, \beta^4, \ldots, \beta^{2^{n-1}}\right\}$ поля F_2^m називається оптимальним , якщо виконується умова

$$\beta^{2^i}\beta^{2^k} = \beta^{2^r} + \beta^{2^s}, \quad i \neq k, \quad r \neq s.$$

Відповідно до означення нормальний базис складається із сполучених елементів поля.

У криптосистемах використовуються непарні (більше того, прості) степені розширення m. При непарних m існує так званий ОНБ 2-го типу. Необхідні й достатні умови його існування такі:

1) $2m+1 = p$ – просте число;

2) у простому полі Галуа F_p $Ord(2) \geq m$, або, інакше кажучи, елемент 2 має порядок m або $2m$ (в останньому випадку 2 – примітивний елемент мультиплікативної групи поля F_p^*).

Обом умовам задовольняють, наприклад, непарні числа при $m < 645$, наведені в таблиці 14.1. Зірочками позначені прості числа, прийнятні для криптосистем.

Таблиця 14.1– Приклади елементів непарних ступенів m поля F_2^m, при яких існує ОНБ 2 -го типу

3*	5*	9	11*	23*	29*	33	35	39	41*
89*	95	99	105	113*	119*	131*	135	155	173*
251*	261	273	281*	293*	299	303	309	323	329
419*	429	431*	441	443*	453	473	483	491*	495
575	585	593*	611	615	629	639	641	645	

У розширенні F_2^{2m} при виконанні умов 1) і 2) завжди існує елемент θ, що має порядок $p = 2m + 1$. Дійсно, у полі F_p

$$2^{p-1} = 2^{2m} \equiv 1(\mathrm{mod}\, p) \Rightarrow p \,|\, (2^{2m} - 1).$$

Так як число p повинне ділити порядок мультиплікативної групи поля F_2^{2m}, то в цьому полі існує корінь θ p-ого степеня з 1, для якого

$$\theta^p = 1, \quad \theta \in F_2^{2m},\ Ord(\theta) = p.$$

Виявляється, що генератор β ОНБ поля F_2^m виражається за допомогою елемента θ p-го порядку розширення F_2^{2m} як $\beta = \theta^1 + \theta^{-1}$.

З рівності $2^{2m} \equiv 1(\mathrm{mod}\, p)$ витікає $2^m \equiv \pm 1(\mathrm{mod}\, p)$, тому $\theta^q = \theta$ або $\theta^q = \theta^{-1}$, $q = 2^m$. Тоді $\beta^q = \beta$ і тому β – елемент поля F_2^m.

Нагадаємо тепер як виконується множення у нормальному базисі. Нехай ми маємо нормальний базис

$$N = \left\{\beta, \beta^2, \beta^4, \ldots, \beta^{2^{n-1}}\right\}.$$

Нехай елементи A і B в оптимальному базисі представлені у вигляді лінійних комбінацій або векторів $A = (a_0, a_1, a_2, ..., a_{n-1})$, $B = (b_0, b_1, b_2, ..., b_{n-1})$, причому $A = AN^T$, $B = BN^T$, де T – знак транспонування. Тоді добуток елементів A і B поля $C = AB = \sum_{i=0}^{n-1} c_i \beta^{2^i}$ з урахуванням, що

$$A = a_0\beta + a_1\beta^2 + a_2\beta^4 + ... + a_{n-1}\beta^{2^{n-1}},$$
$$B = b_0\beta + b_1\beta^2 + b_2\beta^4 + ... + b_{n-1}\beta^{2^{n-1}},$$

можна представити в матричній формі

$$C = A\Delta B^T,$$

де Δ-матриця визначається наступним чином

$$\Delta = N^T N = \left[\beta^{2^i+2^k}\right] = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \cdots & \beta^{2^{n-1}+1} \\ \beta^3 & \beta^4 & \beta^6 & \cdots & \beta^{2^{n-1}+2} \\ \cdots & \cdots & \cdots & \cdots & \\ \beta^{2^{n-1}+1} & \beta^{2^{n-1}+2} & \cdots & \beta & \end{bmatrix}, \quad i, \quad k = 0, \quad n-1.$$

Можемо обчислити елементи Δ-матриці. Елементами її першого рядка дорівнюватимуть:

$$\begin{aligned} \beta^3 &= \beta\beta^2 = (\theta + \theta^{-1})(\theta^2 + \theta^{-2}) = (\theta^3 + \theta^{-3}) + (\theta + \theta^{-1}); \\ \beta^5 &= \beta\beta^4 = (\theta + \theta^{-1})(\theta^4 + \theta^{-4}) = (\theta^5 + \theta^{-5}) + (\theta^3 + \theta^{-3}); \\ \beta^9 &= \beta\beta^8 = (\theta + \theta^{-1})(\theta^8 + \theta^{-8}) = (\theta^9 + \theta^{-9}) + (\theta^7 + \theta^{-7}). \end{aligned} \qquad (14.1)$$

Виявляється, щоб виразити елементи Δ-матриці у вигляді суми , немає необхідності визначати елемент θ розширення.

Суть методу, запропонованого А. Бессаловим й А. Теліженко, складається в побудові таблиці відповідності послідовних степенів елементів β й θ, при цьому степені елемента β зростають як $2^i \bmod (2^m - 1)$, тоді як редукція степенів θ береться за $\bmod p$, тому що $\theta^p = 1$.

Умова означає, що степені $\deg(\theta) \pmod p$ пробігають або всі значення від 1 до $2m$ мультиплікативної циклічної групи F_p^* при послідовному подвоєнні (якщо $2^m = -1$), або половину всіх значень (якщо $2^m = 1$).

В обох випадках в області $i = 0,1,\ldots,m-1$ половина інших значень степенів $r = 2^i \bmod p$ може бути представлена від'ємними значеннями

$(-r) \bmod p \equiv (p-r) \bmod p$. Співвідношення дозволяє при цьому однозначно виразити елементи Δ-матриці через елементи нормального базису у вигляді

$$\begin{bmatrix} \beta \\ \beta^2 \\ \beta^4 \\ \beta^8 \\ \beta^{16} \end{bmatrix} \cdot \left[\beta\ \beta^2\ \beta^4\ \beta^8\ \beta^{16}\right] = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 & \beta^{17} \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} & \beta^{18} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} & \beta^{20} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16} & \beta^{24} \\ \beta^{17} & \beta^{18} & \beta^{20} & \beta^{24} & \beta \end{bmatrix}$$

Важливо відзначити, що якщо ОНБ існує при даному m, то він єдиний (з точністю до циклічного зрушення елементів β^{2^i}).

Приклад 1

Розглянемо процедуру визначення $\Delta^{(0)}$ матриці ОНБ у полі F_2^5. У цьому випадку $p = 1m + 1 = 11, \quad 2^5 = -1(\text{mod}11)$, і таким чином, 2 примітивний елемент поля F_{11} з порядком $2m = 10$.

Вибираючи степені елементів згідно (14.1) і зіставляючи їх з виразом генератора β в НБ, одержимо:

$$\begin{aligned} \beta^3 &= \beta^8 + \beta, \\ \beta^5 &= \beta^{16} + \beta^8. \end{aligned} \qquad \beta^3 = \beta^8,\ \beta^5 = \beta^{16}\ (\text{у НБ}) \qquad (14.2)$$

У загальному випадку досить визначити $(m-1)/2$ елементів 1-го рядка Δ-матриці, тому що їхні степені визначають перші сполучені елементи степенів різних циклотоматичних класів, які заповнюють матрицю уздовж головної діагоналі.

Запишемо послідовно у двійковій формі сполучені елементи двох циклотоматичних класів $\left\{\beta^3, \beta^6, \beta^{12}, \beta^{24}, \beta^{17}\right\}$, $\left\{\beta^5, \beta^{10}, \beta^{20}, \beta^9, \beta^{18}\right\}$:

$\beta^3 = 10010,$ $\beta^5 = 00011,$

$\beta^6 = 01001,$ $\beta^{10} = 10001,$

$\beta^{12} = 10100,$ $\beta^{20} = 11000,$

$\beta^{24} = 01010,$ $\beta^9 = 01100,$

$\beta^{17} = 00101,$ $\beta^{18} = 00110,$

які заповнюють діагоналі Δ-матриці. У нормальному базисі ці елементи утворюються послідовними циклічними зсувами двійкових векторів праворуч.

Відбираючи елементи першого степеня β (ліві позиції векторів), одержимо допоміжні вектори.

$$(\beta^3)' = 10100, \quad (\beta^5)' = 01100, \tag{14.3}$$

які утворюються з (14.2) як реверсні послідовності при фіксації початкового символу. Вектори (14.3) далі вписуються в діагоналі матриці $\Delta^{(0)}$ (індекси i й k матриці послідовно зростають на одиницю за $\mathrm{mod}\, m$). У результаті одержимо матрицю

$$\Delta^{(0)} = \begin{bmatrix} \beta \\ \beta^2 \\ \beta^4 \\ \beta^8 \\ \beta^{16} \end{bmatrix} \begin{bmatrix} \beta & \beta^2 & \beta^4 & \beta^8 & \beta^{16} \end{bmatrix} = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 & \beta^{17} \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} & \beta^{18} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} & \beta^{20} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta^{16} & \beta^{24} \\ \beta^{17} & \beta^{18} & \beta^{20} & \beta^{24} & \beta \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}.$$

Ця симетрична матриця має мінімальну вагу $C_N = 2m - 1 = 9$, тобто має по дві одиниці в кожному рядку (стовпці) , крім першого, і, таким чином, є матрицею ОНБ. Згідно $\Delta = \sum_{i=0}^{n-1} \Delta^{(i)} \beta^{2^i}$, координати вектора-добутку дорівнюють:

$$c_0 = a_0 b_1 + a_1 (b_0 + b_3) + a_2 (b_3 + b_4) + a_3 (b_1 + b_2) + a_4 (b_2 + b_4);$$
$$c_1 = a_1 b_2 + a_2 (b_1 + b_4) + a_3 (b_4 + b_0) + a_4 (b_2 + b_3) + a_0 (b_3 + b_0);$$
$$c_2 = a_2 b_3 + a_3 (b_2 + b_0) + a_4 (b_0 + b_1) + a_0 (b_3 + b_4) + a_1 (b_4 + b_1);$$
$$c_3 = a_3 b_4 + a_4 (b_3 + b_1) + a_0 (b_1 + b_2) + a_1 (b_4 + b_0) + a_2 (b_0 + b_2);$$
$$c_4 = a_4 b_0 + a_0 (b_4 + b_2) + a_1 (b_2 + b_3) + a_2 (b_0 + b_1) + a_3 (b_1 + b_3),$$

де індекси наступного рядка збільшуються на $1(\mathrm{mod}\, m)$ в порівнянні з попередньої. Обчислювальна складність операції множення $A \cdot B$, як бачимо мінімальна.

Приклад 2

Визначимо Δ-матрицю для нормального базису $N = \left\{\beta, \beta^2, \beta^4, \beta^8\right\}$, $\beta = \alpha^3$. Далі знайдемо добуток елементів $A = 1011$, $B = 0111$ у нормальному базисі.

Для нашого прикладу Δ - матриця має вигляд :

$$\begin{bmatrix} \beta \\ \beta^2 \\ \beta^4 \\ \beta^8 \end{bmatrix} \cdot \begin{bmatrix} \beta & \beta^2 & \beta^4 & \beta^8 \end{bmatrix} = \begin{bmatrix} \beta^2 & \beta^3 & \beta^5 & \beta^9 \\ \beta^3 & \beta^4 & \beta^6 & \beta^{10} \\ \beta^5 & \beta^6 & \beta^8 & \beta^{12} \\ \beta^9 & \beta^{10} & \beta^{12} & \beta \end{bmatrix}.$$

Так як β – елемент 5-го порядку $\left(\beta^5 = 1\right)$, і відповідно до таблиці 14.1 попередньої лекції маємо:

$$\begin{aligned} &\beta^3 = \alpha^9 = \beta^8; \\ &\beta^6 = \left(\beta^3\right)^2 = \left(\alpha^9\right)^2 = \alpha^{18} = \alpha^3 = \beta; \\ &\beta^{10} = \left(\beta^5\right)^2 = 1^2 = 1; \\ &\beta^{12} = \left(\beta^6\right)^2 = \beta^2; \\ &\beta^9 = \beta^6 \beta^3 = \beta\beta^3 = \beta^4. \end{aligned} \quad .$$

Дану матрицю можна виразити через елементи НБ:

$$\Delta = \begin{bmatrix} \beta^2 & \beta^8 & 1 & \beta^4 \\ \beta^8 & \beta^4 & \beta & 1 \\ 1 & \beta & \beta^8 & \beta^2 \\ \beta^4 & 1 & \beta^2 & \beta \end{bmatrix}.$$

Тут $\beta+\beta^2+\beta^4+\beta^8=1$, тому в розкладанні $\Delta=\sum_{i=0}^{n-1}\Delta^{(i)}\beta^{2^i}$ матриця $\Delta^{(0)}$ має одиниці на позиціях, у яких є доданок β.

$$\Delta^{(0)}=\begin{bmatrix}0 & 0 & 1 & 0\\ 0 & 0 & 1 & 1\\ 1 & 1 & 0 & 0\\ 0 & 1 & 0 & 1\end{bmatrix}.$$

Інші матриці $\Delta^{(i)}$, $i=1,2,3$ утворюються із $\Delta^{(0)}$ циклічним зсувом позицій уздовж головної діагоналі $(i,k)\to(i+1,k+1)\ (\mathrm{mod}\, n)$.

Наприклад,

$$\Delta^{(1)}=\begin{bmatrix}1 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\\ 1 & 0 & 0 & 1`\\ 0 & 1 & 1 & 0\end{bmatrix}.$$

Ваги всіх матриць $\Delta^{(i)}$ (число одиниць) однакові. У нашому прикладі вага матриць мінімальна й дорівнює $W_0=2n-1=7$. Нормальні базиси з мінімальною вагою матриці $\Delta^{(0)}$ називають оптимальними (тому що гарантують мінімальний об'єм обчислень).

Тепер просто записати формули для обчислення коефіцієнтів c_j добутку $C=AB$.

$$\begin{aligned}
c_0 &= a_0b_2+a_1(b_2+b_3)+a_2(b_0+b_1)+a_3(b_1+b_3);\\
c_1 &= a_1b_3+a_2(b_3+b_0)+a_3(b_1+b_2)+a_0(b_2+b_0);\\
c_2 &= a_2b_0+a_3(b_0+b_1)+a_0(b_2+b_3)+a_1(b_3+b_1);\\
c_3 &= a_3b_1+a_0(b_1+b_2)+a_1(b_3+b_0)+a_2(b_0+b_2).
\end{aligned}$$

Вирази для розрахунку кожного наступного коефіцієнта c_j рекурентно пов'язане з попереднім приростом на $1(\mathrm{mod}\,4)$ всіх індексів коефіцієнтів a_i,b_k. У нашому прикладі $A=1011,\ B=0111$. Тоді

$$c_0 = 1\cdot 1 + 0\cdot(1+1) + 1\cdot(0+1) + 1\cdot(1+1) = 0;$$
$$c_1 = 0\cdot 1 + 1\cdot(1+0) + 1\cdot(1+1) + 1\cdot(1+0) = 0;$$
$$c_2 = 1\cdot 0 + 1\cdot(0+1) + 1\cdot(1+1) + 1\cdot(1+1) = 1;$$
$$c_3 = 1\cdot 1 + 1\cdot(\ 1+1) + 0\cdot(1+0) + 1\cdot(0+1) = 0.$$

У такий спосіб $C = 0010$. Перевіримо цей результат, користуючись експонентним поданням елементів. Відповідно до таблиці $A = \alpha^{13}, B = \alpha^{14}$, тоді $C = \alpha^{13+14} = \alpha^{12} = 0010.$

Іншими словами, кожен елемент Δ-матриці (крім елементів головної діагоналі) представляється лише двома доданками в НБ, що забезпечує мінімальне число одиниць у матриці $\Delta^{(0)}$, що дорівнює $C_N = 2m - 1$ (одна одиниця в першому рядку і по двох – в інших). Тим самим досягається мінімальне число парціальних додавань в операції множення або, іншими словами, мінімальна обчислювальна складність множення.

Контрольні питання та задачі

1. Умови існування оптимального нормального базису.
2. Множення елементів у оптимальному нормальному базисі.
3. Визначення Δ-матриці для нормального базису.
4. Переваги оптимального нормального базису.
5. Мінімально можлива вага Δ-матриці.
6. За допомогою незвідного поліному у полі F_2^3 $f(x)=x^3+x+1$ побудувати поліноміальне надання. Обрати утворюючій елемент, перейти до нормального базису. Перемножити елементи A та B у ОНБ $\{\beta,\ \beta^2,\ \beta^4\}$ поля F_2^3.

а) $A=\beta^5, B=\beta^6$;
б) $A=\beta^4, B=\beta^6$;
в) $A=\beta^5, B=\beta^6$.

7. Дивиться умови попереднього завдання, але взяти незвідний поліном

$f(x)=x^3+x^2+1$

Лекція 15

Проблема дискретного логарифмування у групі точок еліптичної кривої

В пошуках криптографічних алгоритмів з відкритим розповсюдженням ключів з експоненціальною складністю криптоаналізу спеціалісти зупинились на криптографічних перетвореннях, що виконуються в групі точок ЕК. Відповідно до прогнозів ці перетворення ще довго забезпечуватимуть необхідний рівень стійкості.

Розглянемо основні задачі криптоаналізу для систем, в яких перетворення здійснюються в групі точок ЕК, методи їх розв'язання та дамо оцінку стійкості для відомих нам методів криптоаналізу.

Під час аналізу стійкості необхідно розглянути дві проблеми стійкості - розв’язання задачі дискретного логарифму та задачі Діффі – Хеллмана.

Проблема дискретного логарифму формується в наступному вигляді. Нехай задано точку G на еліптичній кривій $E(F(q))$, де $q = p$ ($p -$ просте число) або $q = p^m$ ($p -$ просте число, $m -$ натуральне, $m \in N$). Відомо також значення відкритого ключа Q, причому

$$Q = dG(\mathrm{mod}\, q). \qquad (15.1)$$

Необхідно знайти конфіденційний (особистий) ключ d.

Проблема Діффі – Хеллмана формується у наступному вигляді. Нехай дано ЕК $E(F(q))$, відомо значення точки $G \in E(F(q))$, а також відкритий ключ $Q = dG$. Необхідно знайти загальний секрет

$$K_{12} = d_1 \cdot d_2 G(\mathrm{mod}\, q), \qquad (15.2)$$

де d_1 та d_2 - особисті ключі відповідно першого та другого користувачів.

На сьогодні для аналізу стійкості та проведення криптоаналізу знайшли розповсюдження декілька методів Полларда – λ, ρ та оптимальний $\rho_{опт}$.

Поллард запропонував замість детерміністського псевдоймовірнісний алгоритм розв'язання DLP в полі F_q. Це дозволило істотно знизити вимоги до обсягу пам'яті при практично тій же стійкості алгоритму. Ідея методу заснована на випадковому пошуку двох співпадаючих точок серед точок криптосистеми.

У теорії ймовірностей добре відомі задачі про випадкові блукання. Одна із задач ставиться так.

Є n ящиків і $k < n$ куль, які випадковим образом розміщені по ящиках. Процедура закінчується при першому влученні кулі у вже зайнятий ящик. Потрібно визначити медіану розподілу ймовірностей $P_n(k)$.

Більше простою моделлю є задача про співпадаючі дні народження. Якщо n – число днів у році, то скільки чоловік k з рівноймовірними днями народження в році потрібно відібрати, щоб з імовірністю $1/2$ дні народження хоча б двох людей співпадали ?

Очевидно, що ймовірність такої події дорівнює

$$P_n(k) = 1 - \left(1 - \frac{1}{n}\right)\left(1 - \frac{2}{n}\right)...\left(1 - \frac{k-1}{n}\right).$$

При $k << n$ неважко одержати наближене значення цієї імовірності

$$P_n(k) \approx \frac{1 + 2 + ... + (k-1)}{n} = \frac{k(k-1)}{2n} \approx \frac{k^2}{2n}.$$

Приймаючи $P_n(k) = 1/2$, одержимо оцінку числа $k \approx \sqrt{n}$. Інакше кажучи, щоб при випадковому переборі великої множини із n чисел з імовірністю $50\,\%$ двічі з'явилося те саме число, буде потрібно в середньому порядку $\sqrt{n}$ спроб. Збіг елементів або точок в аналізі прийнято називати колізією.

Нехай $Q = kG$, де генератор G криптосистеми має великий простий порядок n. Алгоритм ρ- методу в застосуванні до еліптичних кривих складається в послідовному обчисленні точок

$$Q_{i+1} = \begin{cases} Q_i + G, |x_i| \in S_1, \\ 2Q_i, |x_i| \in S_2, \\ Q_i + Q_0, |x_i| \in S_3, i = 0,1,2,..., \end{cases}$$

де $|x_i|$ – якась міра координати x_i точки Q_i, S_1, S_2, S_3 – три рівноймовірні області, у які може потрапити ця міра. Виберемо випадкові значення $a_0, b_0 \in [1, n-1]$ й визначимо початкову точку як $Q_0 = a_0 G + b_0 Q.$ Ітераційна послідовність обчислень дає послідовність $Q_i = a_i G + b_i Q$, таку що

$$(Q_{i+1,} a_{i+1}, b_{i+1}) = \begin{cases} (Q_i + G, a_i + 1, b_i), |x_i| \in S_1, \\ (2Q_i, 2a_i, 2b_i), |x_i| \in S_2, \\ (Q_i + Q, a_i, b_i + 1), |x_i| \in S_3. \end{cases}$$

На кожному кроці обчислене значення x_i порівнюється з попереднім аж до збігу (колізії) $x_{m+s} = x_m$ або

$$Q_{m+s} = \pm Q_m$$

Алгоритм разом з колізією дозволяє скласти рівняння $a_{m+1} G + b_{m+s} Q = a_m G + b_m Q, Q = kG,$ з якого визначається значення дискретного логарифма

$$k = -\frac{a_{m+s} \pm a_m}{b_{m+s} \pm b_m} \bmod n.$$

Походження терміну (ρ - метод) пов'язане із графічною інтерпретацією алгоритму, представленої на рисунку 15.1. При замиканні петлі виникає періодичний цикл. Це обумовлено детермінованістю алгоритму. Його називають імовірнісним лише у зв'язку з непередбачуваністю шляху, по якому виконується одне із трьох обчислень.

Реалізація методу пов'язана з нарощуванням пам'яті, у яку записуються x - координати точок, що Q_i обчислюють. У міру збільшення порядку n криптосистеми він незабаром стає практично нереалізованим. Позбутися від цього недоліку вдається за допомогою методу Флойда.

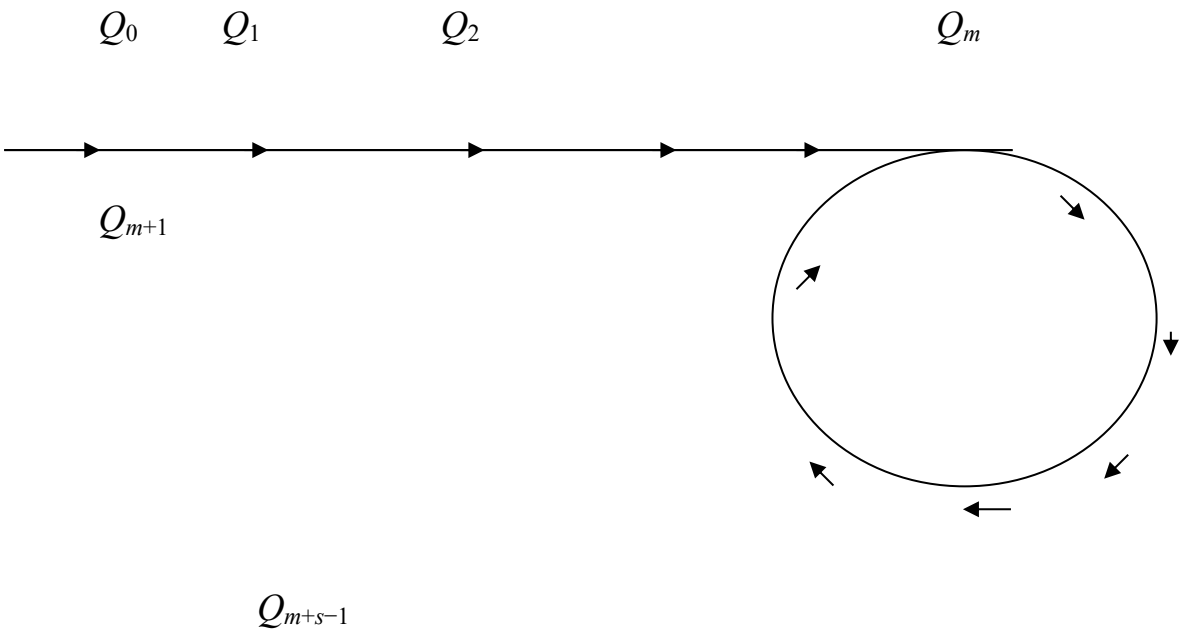

Рис. 15.1 – Графічна інтерпретація ρ-методу Полларда

Ідея методу проста й елегантна. На циферблаті секундна стрілка завжди обганяє хвилинну, а хвилинна – годинну. При влученні усередину петлі в ρ-методі Полларда якась точка Q_{2k} наздоганяє точку Q_k (колізія $Q_{2k} = Q_k$), що дає розв'язання *ECDLP*.

У такий спосіб замість порівняння чергової обчисленої точки з усіма попередніми досить у пам'яті зберегти для порівняння лише дві точки: Q_i і Q_{2i}. Точка колізії при цьому зрушується усередину петлі на відстань, що не перевищує половини довжини петлі. Тим самим відбувається обмін необхідної пам'яті на час обчислень.

Кожен цикл у методі Флойда вимагає обчислення трьох точок відповідно до алгоритму й порівняння двох з них. Вихідні дані – точки Q_i й Q_{2i}, обчислені

в попередньому циклі. На їхній основі згідно розраховуються точки Q_{i+1}, Q_{2i+1} й Q_{2i+2} і рівняються x - координати першої й останньої точок. При їхньому збігу має місце колізія $Q_{2k} = \pm Q_k$, де знак визначається з порівняння y - координат обчислених точок.

Найпростіша ілюстрація цього методу – спрощений алгоритм із обчисленням $Q_{i+1} = Q_i + G$. Колізія на s-тім циклі $(k+s)\cdot G = 2\cdot(k+s)\cdot G$ відразу дає розв'язання дискретного логарифму $k \equiv -s \bmod n$. По суті це прямий метод визначення дискретного логарифма з експоненційною складністю $O(n)$.

В іншому окремому випадку алгоритму маємо $Q_{i+1} = 2Q_i$. Колізія на s-тому кроці приведе до рівняння $2^{s-1}k \equiv 2^s k \bmod n$ або $2^{s-1}k \equiv 0 \bmod n$. Воно не має розв'язання $k \neq 0$. Якщо модернізувати алгоритм так, що на кожній ітерації порівнювати точки $2^i kG$ й генератор G, то при виконанні $2^s kG = G$ можна одержати розв'язання $k \equiv 2^{n-s-1} \bmod n$ за умови, що 2 є примітивним елементом поля F_n.

На практиці для виявлення колізій замість методу Флойда знайшла застосування його модифікація, запропонована Шнором і Ленстрой. У цієї модифікації пам'ять містить 8 осередків, зрушення вмісту яких здійснюється при $i \geq 3j$, де i, j – номера ітерацій в останньому й першому осередках відповідно.

Алгоритм ρ- методу Полларда з розбивкою на три області S_i є споконвічним і найбільш простим у реалізації. Подальші вдосконалення алгоритму пропонують використання r рівноймовірних областей з вибором, наприклад, ітераційної функції

$$Q_{i+1} = \begin{cases} Q_i + m_1 G + n_1 Q, |x_i| \in S_1, \\ Q_1 + m_2 G + n_2 Q, |x_i| \in S_2, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ Q_i + m_r G + n_r Q, |x_i| \in S_r, i = 0,1,2,\ldots. \end{cases}$$

Число областей, як правило, не перевищує 20, тому що подальше їхнє збільшення практично не впливає на статистичні характеристики алгоритму.

Очевидно колізію точок можна одержати й іншим шляхом, рухаючись із двох (або більше) різних точок Q_m і Q_n до збігу $Q_{m+s} = Q_{n+s}$. Ця ситуація відбивається на рисунку 15.2. Даний метод одержання колізії зветься λ-Методом Полларда. Походження терміна прийняте з рисунка.

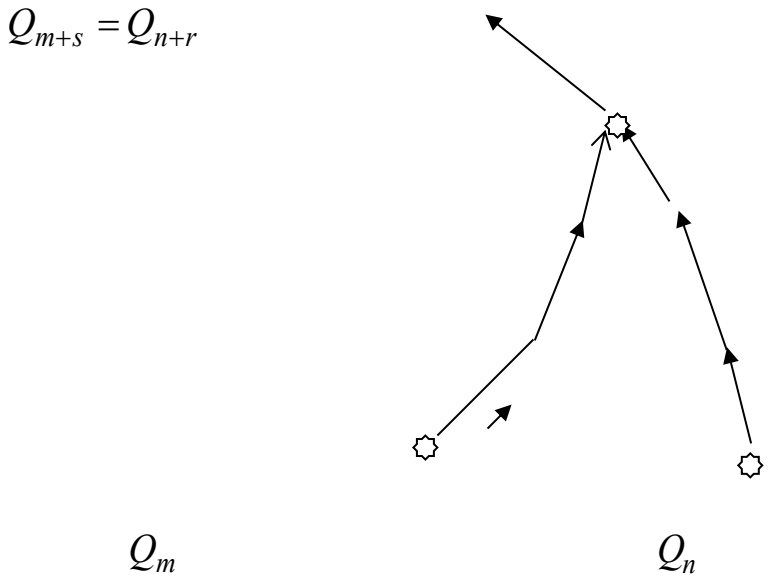

Рис. 15.2 – Графічна інтерпретація λ-методу Полларда

Контрольні запитання та завдання

1. Як формулюється задачіа дискретного логарифму у групі точок ЕК?
2. Постановка задачі Діффі – Хеллмана у групі точок ЕК?
3. Яка сутність ρ-методу Полларда розв′язання DLP в полі F_q?

Лекція 16

Проблема дискретного логарифмування у групі точок еліптичної кривої (продовження)

Розглянемо ρ-метод Полларда на прикладі *ЕК* над простим полем Галуа $GF(p)$, тобто

$$y^2 = x^3 + ax + b(\mathrm{mod}\, p). \tag{16.1}$$

Для всіх точок $(x_i, y_i) \in E(F(q))$ задано операції додавання та подвоєння.

Наприклад, якщо $G_1 = (x_1, y_1)$ а $G_2 = (x_2, y_2)$, то

$$G_1 + G_2 = (x_1, y_1) + (x_2, y_2) = (x_3, y_3),$$

де

$$\begin{aligned} x_3 &= \lambda^2 - x_1 - x_2 (\mathrm{mod}\, p); \\ y_3 &= \lambda(x_1 - x_3) - y_1 (\mathrm{mod}\, p); \end{aligned}$$

$$\begin{cases} \lambda = \dfrac{y_2 - y_1}{x_2 - x_1} (\mathrm{mod}\, p),\ G_1 \neq G_2; \\ \lambda = \dfrac{3x_1^2 + a}{2y_1} (\mathrm{mod}\, p),\ G_1 = G_2. \end{cases} \tag{16.2}$$

Для *ЕК* над полем $F(2^m)$ виду

$$y^2 + xy = x^3 + ax^2 + b(\mathrm{mod}\, f(x), 2),$$

причому $b \neq 0$, то для двох точок G_1 та G_2 таких, що

$$G_1 + G_2 = (x_1, y_1) + (x_2, y_2) = (x_3, y_3)$$

виходить

$$\begin{aligned} x_3 &= \lambda^2 + \lambda + x_1 + x_2 + a(\mathrm{mod}\, f(x), 2); \\ y_3 &= \lambda(x_1 + x_3) + x_3 + y_1 (\mathrm{mod}\, f(x), 2); \end{aligned} \tag{16.3}$$

$f(x)$ примітивний поліном *m*-го степеня;

$$\lambda = \begin{cases} \dfrac{y_1 + y_2}{x_1 + x_2} (\mathrm{mod}\, f(x), 2, G_1 \neq G_2; \\ \dfrac{x_1^2 + y_1}{x_1} (\mathrm{mod}\, f(x), 2), G_1 = G_2. \end{cases} \tag{16.4}$$

Для розв'язання задачі пошуку конфіденційного ключа d в порівнянні (16.1) розглянемо ρ - метод Полларда над простим полем $GF(p)$. Нехай G-базова точка, Q – відкритий ключ, шукатимемо парі цілих $(A_i, B_i) \bmod n$ та $(A_j, B_j) \bmod n$, таких що

$$A_i G + B_i Q = A_j G + B_j Q (\mathrm{mod}\, p). \tag{16.5}$$

Позначимо в загальному вигляді

$$Z_i = A_i G + B_i Q (\mathrm{mod}\, p). \tag{16.6}$$

Суть ρ-методу Полларда розв'язання порівняння (16.1) міститься в наступному. Знайдемо деяку функцію $f(Z)$, вибравши $A_0, B_0 \in [1, n-1]$, де n – порядок точки G на ЕК

$$Z_0 = A_0 G + B_0 Q. \tag{16.7}$$

Далі знайдемо Z_i послідовність:

$$Z_1 = f(Z_0), Z_2 = f(Z_1), Z_3 = f(Z_2), ..., Z_i = f(Z_{i-1})$$

для пар A_i, B_i, таких що:

$$Z_i = A_i G + B_i Q. \tag{16.8}$$

Рекомендується в простих випадках (при відносно невеликих p) послідовність Z розраховувати у вигляді:

$$f(Z_{i+1}) = \begin{cases} 2Z_i, \left|Z_i\right|_X \in S_1; \\ Z_i + G, \left|Z_i\right|_X \in S_2; \\ Z_i + Q, \left|Z_i\right|_X \in S_3. \end{cases} \tag{16.9}$$

При цьому S_1, S_2 та S_3 складають частини області $[1, p-1]$. Якщо область $[1, p-1]$ рівномірно ділиться, то (16.9) має вигляд:

$$f(Z_{i+1}) = \begin{cases} 2Z_i, 0 < |Z_i| \le \left\lceil p/3 \right\rceil; \\ Z_i + G, \left\lceil p/3 \right\rceil < |Z_i| \le \left\lceil 2p/3 \right\rceil; \\ Z_i + Q, \left\lceil 2p/3 \right\rceil < |Z_i| \le p-1. \end{cases} \tag{16.10}$$

При побудові множини Z_{i+1} пошук буде успішним, якщо ми знайдемо

$$Z_i \equiv Z_j \pmod p,$$

що еквівалентно знаходженню

$$A_i G + B_i Q = A_j G + B_j Q. \tag{16.11}$$

Зробивши прості перетворення, маємо:

$$(A_i - A_j)G \equiv (B_i - B_j)Q \pmod p \tag{16.12}$$

і далі

$$Q = (A_i - A_j)(B_i - B_j)^{-1} G \pmod p. \tag{16.13}$$

З (15.1) та (16.13) випливає, що

$$\begin{aligned} &d = (A_i - A_j)(B_i - B_j)^{-1} G \pmod n, \\ &B_i \ne B_j \pmod n. \end{aligned} \tag{16.14}$$

Більш ефективним є розрахунок Z_i з розбиванням інтервалу p на t інтервалів. Для реальних значень p рекомендується $t \approx 20$. У цьому випадку замість (16.10) маємо

$$f(Z_{i+1}) = \begin{cases} Z_i + a_1 G + b_1 Q, Z_i \in S_1; \\ Z_i + a_2 G + b_2 Q, Z_i \in S_2; \\ \\ Z_i + a_t G + b_t Q, Z_i \in S_t. \end{cases} \tag{16.15}$$

причому a_i та b_i є випадкові цілі з інтервалу $[1, n-1]$.

При розв'язанні задач важливо успішно вибрати Z_0. Значення Z_0 рекомендується вибирати у вигляді

$$G, Q, G+Q(\mathrm{mod}\, p), G-Q(\mathrm{mod}\, p).$$

Z_0 також можна вибрати як

$$Z_0 = a_0 G + b_0 Q(\mathrm{mod}\, p),$$

де $(a_0, b_0) \in [1, n-1]$.

***Задача* 1**. Нехай точка $G(13,7)$ належить ЕК

$$y^2 = x^3 + ax + b(\mathrm{mod}\, 23),$$

причому $a = 1$ і $b = 1$, тобто

$$y^2 = x^3 + x + 1(\mathrm{mod}\, 23).$$

Відкритий ключ $Q = (17,20)$. Порядок точки $n = 7$, порядок *ЕК* $u = n \cdot k = 4 \cdot 7 = 28$, де k - кофактор. Необхідно знайти секретний ключ d із порівняння

$$Q = dG(\mathrm{mod}\, p), d \in [1, n-1].$$

У нашому випадку

$$(17,20) = d \cdot (13,7)(\mathrm{mod}\, p), d \in [1,6].$$

Розв'язання задачі. Використовуючи співвідношення, отримаємо:

$$f(Z_{i+1}) = \begin{cases} 2Z_i, 0 < |Z_i| \le \left\lceil p/3 \right\rceil; \\ Z_i + G, \left\lceil p/3 \right\rceil < |Z_i| \le \left\lceil 2p/3 \right\rceil; \\ Z_i + Q, \left\lceil 2p/3 \right\rceil < |Z_i| \le p-1. \end{cases}$$

Таблиця 16.1 - Результати розв'язку задачі 1

Z_i	a_i	b_i	$a_iG + b_iQ$
$Z_0 = (13,7)$	1	0	$1 \cdot (13,7) + 0 \cdot (17,20)$
$Z_1 = (5,4)$	2	0	$2 \cdot (13,7) + 0 \cdot (17,20)$
$Z_2 = (17,20)$	4	0	$4 \cdot (13,7) + 0 \cdot (17,20)$
$Z_3 = (13,7)$	4	1	$4 \cdot (13,7) + 1 \cdot (17,20)$

$$\lceil p/3 \rceil = 7; S_1 \in [1,7],$$
$$\lceil 2p/3 \rceil = 15; S_2 \in [8,15],$$
$$p = 23; S_3 \in [16,22].$$

Виберемо як $Z_0 = G,$ тоді $|Z_0|_X = 13$ належить S_2, тому

$$Z_1 = Z_0 + G = (13,7) + (13,7) = 2 \cdot (13,7) = (x_3, y_3).$$

$$\lambda = \frac{3 \cdot x_1^2 + a}{2 \cdot y_1} (\mathrm{mod}\, p) = \frac{3 \cdot 13^2 + 1}{2 \cdot 7} (\mathrm{mod}\, 23) = \frac{24}{7} (\mathrm{mod}\, 23) = \frac{1}{7} (\mathrm{mod}\, 23)$$

$7 \cdot x \equiv 1 (\mathrm{mod}\, 23)$.

Розв'язуємо це рівняння, використовуючи алгоритм Евкліда

$$\frac{23}{7} = 3 + \frac{1}{3 + \frac{1}{2 + 0}}$$

Отже $\mu = 2.$ Таким чином, $r_0 = 3; r_1 = 3; r_2 = 2; \mu = 2.$

$$x = (-1)^2 P_1 \cdot 1 (\mathrm{mod}\, 23) = 1 \cdot 10 (\mathrm{mod}\, 23) = 10$$
$$P_1 = r_0 \cdot r_1 + 1 = 3 \cdot 3 + 1 = 10$$

У результаті маємо, що

$$\lambda = \frac{1}{7}(\mathrm{mod}\,23) = 1 \cdot 10(\mathrm{mod}\,23) = 10;$$

$$x_3 = \lambda^2 - x_1 - x_2\,(\mathrm{mod}\,P) = 100 - 2 \cdot 13(\mathrm{mod}\,23) = 74(\mathrm{mod}\,23) = 5;$$

$$y_3 = \lambda(x_1 - x_3) - y_1\,(\mathrm{mod}\,P) = 10 \cdot (13 - 5) - 7(\mathrm{mod}\,23) = 73(\mathrm{mod}\,23) = 4;$$

Таким чином $Z_1 = Z_0 + G = 2 \cdot (13,7) = (5,4).$

Другий крок: $Z_1 \in S_1$. Знаходимо $2 \cdot Z_1 = 2 \cdot (5,4).$

$$\lambda = \frac{3 \cdot x_1^2 + a}{2 \cdot y_1}(\mathrm{mod}\,p) = \frac{3 \cdot 5^2 + 1}{2 \cdot 4}(\mathrm{mod}\,23) = \frac{19}{2}(\mathrm{mod}\,23).$$

Мультиплікативно зворотний елемент числу 2 у полі $GF(23)$ знаходимо з рівняння

$$x \cdot 2 \equiv 1(\mathrm{mod}\,23);$$

дійсно

$$\frac{23}{2} = 12 + \frac{1}{2};\ \mu = 1;\ P_0 = 12;$$

$$\lambda = 19 \cdot 12(\mathrm{mod}\,23) = 21;$$

$$x_3 = \lambda^2 - x_1 - x_2\,(\mathrm{mod}\,p) = 21^2 - 10(\mathrm{mod}\,23) = 17;$$

$$y_3 = \lambda(x_1 - x_3) - y_1\,(\mathrm{mod}\,p) = 21 \cdot (5 - 17) - 4(\mathrm{mod}\,23) = 20.$$

Таким чином,

$$Z_3 = Z_2 + Q = (17,20) + (17,20) = 2 \cdot (17,20).$$

Далі знаходимо $2 \cdot Z_2 = 2 \cdot (17,20)$

$$\lambda = \frac{3 \cdot x_1^2 + a}{2 \cdot y_1}(\mathrm{mod}\,p) = \frac{3 \cdot 17^2 + 1}{2 \cdot 20}(\mathrm{mod}\,23) = \frac{217}{10}(\mathrm{mod}\,23) = 1;$$

$$x_3 = \lambda^2 - 2x_1(\mathrm{mod}\,p) = 1^2 - 34(\mathrm{mod}\,23) = -33(\mathrm{mod}\,23) = 13;$$

$$y_3 = \lambda(x_1 - x_3) - y_1(\mathrm{mod}\,p) = 1 \cdot (17 - 13) - 20(\mathrm{mod}\,23) = -16(\mathrm{mod}\,23) = 7.$$

Таким чином, у таблиці ми знайшли, що

$$Z_3 = Z_0(mod\,p);$$

$$a_3 = 4;\ b_3 = 1; \quad i = 0; \quad j = 3;$$
$$a_0 = 1; b_0 = 0.$$

Знаходимо

$$d = \frac{a_i - a_j}{b_j - b_i} (\mathrm{mod}\, n) = \frac{1-4}{1-0} (\mathrm{mod}\, 7) = -3 (\mathrm{mod}\, 7) = 4.$$

Перевіряємо

$$Q' = dG = 4 \cdot (13,7).$$
$$2 \cdot (13,7) = (5,4);$$
$$2 \cdot (5,4) = (17,20).$$

Таким чином

$$Q' = Q;$$
$$(17,20) = (17,20).$$

Контрольні запитання та завдання

1. Знайти таємний ключ d, якщо відомий відкритий ключ $Q = dG$ для кривої E: $y^2 = x^3 + 10x + 7 \left(\mathrm{mod}\, 23\right)$ над простим полем F_{23}, якщо $G = (15,6)$ точка порядку $n = 25$ та $Q = (8,1)$.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Горбенко І. Д. ″ Криптографічний захист інформації ″. Навч. посібник Харків, ХНУРЕ, 2004 р.

2. Вербіцький О. В. Всуп до криптології. - Львів.: Видавництво науково-технічної літератури, 1998. - 247 с.

3. Бессалов А.В., Телиженко А.Б. Криптосистеми на еліптичних кривих: Навч. посібн. – К.: ІВЦ «Політехніка», 2004. – 224с.

4. Криптологія: навч. посібник / М.Н. Курко, П.М. Лісовський, Ю.П. Лісовська. — К.: Видавничий дім «Кондор», 2020. — 248 с.

5. Безпека інформаційних систем і технологій: Навч. посібник / В. І. Єсін, О. О. Кузнецов, Л. С. Сорока. – Х. : ХНУ імені В. Н. Каразіна, 2013. – 632 с.

6. Горбенко І. Д. Прикладна криптологія. Теорія. Практика. Застосування : Монографія / І. Д. Горбенко, Ю. І. Горбенко. – Харків : Видавництво “Форт”, 2012. – 880 с.: іл.

7. Богуш В. М. Криптографічні застосування елементарної теорії чисел : Навч. посібник / В. М. Богуш, В. А. Мухачов. – К. : Державний ун-т інформаційно-комунікаційних технологій, 2006. – 126 с.: іл.

Електронне навчальне видання комбінованого використання
Можна використовувати в локальному та мережному режимі

Лисицький Костянтин Евгенійович
Колованова Євгенія Павлівна
Узлов Дмитро Юрійович

ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ

Конспект лекцій
з дисципліни для здобувачів вищої освіти першого (бакалаврського) рівня
за спеціальністю 151 «Автоматизація та комп'ютерно-інтегровані технології»
(174 «Автоматизація, комп'ютерно-інтегровані технології та робототехніка»)
освітньої програми «Автоматизація та комп'ютерно-інтегровані технології»

В авторській редакції

Підписано до розміщення 23.04.2025. Гарнітура Times New Roman.
Ум. друк. арк. 6,21. Обсяг 3,692 Мб. Зам. № 159/25.

Харківський національний університет імені В. Н. Каразіна,
61022, м. Харків, майдан Свободи, 4.
Свідоцтво суб'єкта видавничої справи ДК № 3367 від 13.01.2009
Видавництво ХНУ імені В. Н. Каразіна