МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ФАХОВИЙ КОЛЕДЖ ПРОМИСЛОВОЇ АВТОМАТИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ОДЕСЬКОГО НАЦІОНАЛЬНОГО ТЕХНОЛОГІЧНОГО УНІВЕРСИТЕТУ»

ЗАТВЕРДЖУЮ
Заступник директора з
навчально-методичної роботи
ФКПАІТ ОНТУ
_____ Вікторія ОКСАНІЧЕНКО
_____._____.20___ року

Конспект лекцій

ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ

(шифр за ОПП і назва навчальної дисципліни)

галузь знань _____12 «Інформаційні технології»_____
(шифр і назва галузі знань)

спеціальність_____122 «Комп'ютерні науки»_____
(шифр і назва спеціальності)

освітня програма_____Комп'ютерні науки_____
(назва освітньої програми)

(Шифр за ОПП_____)

Одеса 2022

Конспект лекцій з дисципліни «Технології захисту інформації» для підготовки фахового молодшого бакалавра за спеціальністю 122 «Комп’ютерні науки».

Укладач: викладач вищої категорії ФКПАІТ ОНТУ Юлія БУРМАКІНА

Конспект лекцій затверджено на засіданні циклової комісії «Комп’ютених наук та інженерії програмного забезпечення» ФКПАІТ ОНТУ

Протокол від ___._____.20___ року, № ____

Голова циклової комісії

_____Тетяна КОСТИРЕНКО

(підпис) (власне ім’я та прізвище)

___._____.20___ року

Лекція №1 (2 години)

Тема: Поняття інформаційної безпеки.

Мета: Ознайомитися з поняттями й визначенням курсу, розглянути актуальність проблеми захисту інформації й визначити необхідність її для комп'ютерних систем.

ПЛАН

1. Поняття курсу основ захисту інформації
2. Інформаційна система безпеки
3. Завдання захисту користувальницької інформації
4. Засоби вирішення завдання забезпечення безпеки в інформаційно-телекомунікаційних системах.

Стрімкий розвиток засобів обчислювальної техніки й відкритих мереж передачі даних обумовило їхнє широке поширення в повсякденному житті й підприємницькій діяльності. Потужні обчислювальні можливості й оперативність передачі інформації не тільки дуже вплинули на принципи ведення бізнесу, що склалися в більшості традиційних галузей, але й відкрили нові напрямки розвитку підприємницької діяльності. В сучасних умовах автоматизація банківської діяльності й керування підприємствами є «modus vivendi», а такі слова, як «Internet-banking», «e-commerce» і «smart-cards», уже не викликають загального подиву й жарких дебатів.

Однак останні досягнення людської думки в області комп'ютерних технологій пов'язані з появою не тільки персональних комп'ютерів, мереж передачі даних і електронних грошей, але й таких понять, як *хакер, інформаційна зброя, комп'ютерні віруси* й т.п. Виявляється, що під *інформаційною безпекою* мається на увазі один із провідних напрямків розвитку інформаційних технологій - коло завдань, які розв'язуються у цій області, постійно розширюється як у кількісному, так і в якісному відношенні.

Сучасні методи нагромадження, обробки й передачі інформації сприяли появі погроз, пов'язаних з можливістю втрати, розкриття, модифікації даних, що належать кінцевим користувачам.

На практиці погрози для *інформаційно-телекомунікаційних систем* (ІТС) можуть бути реалізовані безпосереднім впливом як на інформацію, що представляє інтерес для кінцевих користувачів подібних систем, так і на інформаційні ресурси й телекомунікаційні служби, забезпечувані в рамках даної ІТС.

Під *інформаційною безпекою* варто розуміти стан захищеності оброблюваних, збережених і переданих в ІТС даних від незаконного ознайомлення, перетворення й знищення (як крайній випадок модифікації), а також стан захищеності інформаційних ресурсів від впливів, спрямованих на порушення їхньої працездатності.

Основними завданнями захисту користувальницької інформації є:

•забезпечення конфіденційності інформації;

•забезпечення цілісності інформації;

•забезпечення вірогідності інформації;

•забезпечення оперативності доступу до інформації;

•забезпечення юридичної значимості інформації, представленої у вигляді електронного документа;

•забезпечення невідстежуваності дій клієнта.

Конфіденційність - властивість інформації бути доступною тільки обмеженому колу користувачів інформаційної системи, у якій циркулює дана інформація.

Під *цілісністю* розуміється властивість інформації або програмного забезпечення зберігати свою структуру й/або зміст у процесі передачі й/або зберігання.

Розглядаючи питання передачі інформації у вигляді повідомлень через мережу, можна дійти висновку, що кожне повідомлення по своєму змісту значення утворить деякий клас. Інакше кажучи, зміст кінцевого повідомлення залишиться таким же, як і початкового, навіть якщо форма подання інформації в електронному вигляді істотно зміниться. Таким чином, кожне повідомлення російською мовою буде мати свій клас еквівалентності, і для даного випадку властивість збереження цілісності інформації можна сформулювати в такий спосіб: передане повідомлення X вважається таким, що зберегло цілісність, якщо отримане в результаті передачі повідомлення X1 належить класу еквівалентності повідомлення X.

Вірогідність - властивість інформації, що виражається в строгій приналежності об'єкту, що є її джерелом, або тому об'єкту, від якого ця інформація прийнята.

Оперативність - здатність інформації або деякого інформаційного ресурсу бути доступним для кінцевого користувача відповідно до його тимчасових потреб.

Юридична значимість означає, що документ має юридичну силу. Із цією метою суб'єкти, які мають потребу в підтвердженні юридичної значимості переданого повідомлення, домовляються про повсюдне прийняття деяких атрибутів інформації, що виражають її здатність бути юридично значимою. Дана властивість інформації особливо актуальна в системах електронних платежів, де здійснюється операція з переводу коштів. Виходячи зі сказаного, можна сформулювати деякі вимоги до атрибутів інформації, що виражають її властивість бути юридично значимою. Насамперед її необхідно сформувати таким чином, щоб з формальної точки зору було виразно ясно, що тільки відправник, якому належить даний платіжний документ, міг його створити.

Невідстежуваність - здатність робити деякі дії в інформаційній системі непомітно для інших об'єктів. Актуальність даної вимоги стала очевидною завдяки появі таких понять, як електронні гроші й Internet-banking. Так, для авторизації доступу до електронної платіжної системи користувач повинен надати деякі

відомості, які однозначно його ідентифікують. У міру розвитку даних систем може з'явитися реальна небезпека, що, наприклад, всі платіжні операції будуть контролюватися, тим самим виникнуть умови для тотального стеження за користувачами інформаційних систем.

Існує кілька шляхів вирішення проблеми невідстежуваності:

- заборона за допомогою законодавчих актів усякого тотального стеження за користувачами інформаційних систем;
- застосування криптографічних методів для підтримки невідстежуваності.

Як вже говорилося, інформаційна безпека може розглядатися не тільки стосовно деяких конфіденційних відомостей, але й стосовно здатності інформаційної системи виконувати задані функції.

Основні задачі, що розв'язуються в рамках інформаційної безпеки стосовно працездатності ІТС, повинні забезпечувати захист від:

- порушення функціонування телекомунікаційної системи, що виражається у впливі на інформаційні канали, канали сигналізації, керування й вилученого завантаження баз даних комутаційного встаткування, системне й прикладне програмне забезпечення;
- несанкціонованого доступу до інформаційних ресурсів і від спроб використання ресурсів мережі, що приводять до витоку даних, порушенню цілісності мережі та інформації, зміні функціонування підсистем розподілу інформації, доступності баз даних;
- руйнування що вбудовуються і зовнішніх засобів захисту;
- неправомочних дій користувачів і обслуговуючого персоналу мережі.

Пріоритети серед перерахованих завдань інформаційної безпеки визначаються індивідуально для кожної конкретної ІТС і залежать від вимог, пропонованих безпосередньо до інформаційних систем.

Варто врахувати, що з погляду державних структур захисні заходи в першу чергу покликані забезпечити конфіденційність, цілісність і доступність інформації. (Зрозуміло, для режимних державних організацій на першому місці завжди стоїть конфіденційність відомостей, а цілісність розуміється винятково як їхня незмінність.) Комерційним структурам, імовірно, важливіше всього цілісність і доступність даних і послуг по їхній обробці. У порівнянні з державними, комерційні організації більше відкриті й динамічні, тому ймовірні погрози для них відрізняються не тільки кількістю, але і якістю.

Для вирішення завдання забезпечення безпеки в інформаційно-телекомунікаційних мережах необхідно:

- захистити інформацію при її зберіганні, обробці й передачі по мережі;
- підтвердити дійсність об'єктів даних і користувачів (аутентифікація сторін, що встановлюють зв'язок);
- виявити й попередити порушення цілісності об'єктів даних;

- захистити технічні пристрої й приміщення;
- захистити конфіденційну інформацію від витоку й від впроваджених електронних пристроїв знімання інформації;
- захистити програмні продукти від впровадження програмних закладок і вірусів;
- захистити від несанкціонованого доступу до інформаційних ресурсів і технічних засобів мережі, у тому числі й до засобів керування, щоб запобігти зниженню рівня захищеності інформації й самої мережі в цілому;
- організувати заходи, що вимагаються для спрямування на забезпечення збереження конфіденційних даних.

Конкретна реалізація загальних принципів забезпечення інформаційної безпеки може виражатися в організаційних або технічних мірах захисту інформації.

Контрольні питання:

1. Розкрити поняття й визначення курсу.
2. Визначити поняття інформаційної безпеки системи.
3. Перелічити й розкрити завдання захисту користувальницької інформації.
4. Перелічити способи рішення завдання забезпечення безпеки в інформаційно-телекомунікаційних система

Лекція №2 (2 години)

Тема: Основні поняття і положення захисту інформації в комп'ютерних системах. Об'єкти захисту інформації.

Мета: Ознайомитися з поняттями й визначенням курсу, розглянути актуальність проблеми захисту інформації й визначити необхідність її для комп'ютерних систем.

ПЛАН

1. Поняття курсу основ захисту інформації
2. Інформаційна система безпеки
3. Завдання захисту користувальницької інформації
4. Засоби рішення завдання забезпечення безпеки в інформаційно-телекомунікаційних системах.

Предмет захисту

В **якості предмета захисту** розглядається інформація, що зберігається, оброблюється і передається в комп'ютерних системах.

Термін інформація походить від латинського **infonnatio**, що означає роз'яснення, інформування, виклад.

З позиції матеріалістичної філософії інформація є відображенням реального світу за допомогою відомостей (повідомлень).

Повідомлення - це форма представлення інформації у вигляді мови, тексту, зображення, цифрових даних, графіків, таблиць і тп.

У неживій природі поняття інформації зв'язують з поняттям відбиттям, відображення.

У побуті під інформацією розуміють відомості, які нас цікавлять, тобто відомості про навколишній світ і процесах, що протікають у ньому, сприйняті людиною або спеціальними пристроями (суб'єктивний підхід).

Інформація для людини - це знання, які він отримує з різних джерел. За допомогою усіх своїх органів чуття людина отримує інформацію із зовнішнього світу.

У теорії зв'язку під інформацією прийнято розуміти будь-яку послідовність символів, не враховуючи їх сенс.

У теорії інформації під інформацією розуміють не будь-які відомості, а лише ті, які знімають повністю або зменшують ту, що існує до їх отримання невизначенність. За визначенням К. Шенона, інформація - це знята невизначенність.

У математиці і кибернетиці - кількісна міра усунення невизначенності (ентропії), міра організації системи.

Інформація - відомості про об'єкти і явища довкілля, їх параметрах, властивостях і стані, які зменшують наявну в них міру невизначенності, неповноту знань.

Цінність інформації є критерієм при ухваленні будь-якого рішення про її захист. Відоме наступне розділення інформації по рівню важливості :

- життєво важлива незамінна інформація, наявність якої потрібна для функціонування комп'ютерної системи, організації і т. п.;

- важлива інформація - інформація, яка може бути замінена або відновлена, але процес відновлення якої дуже важкий або пов'язаний з великими витратами;

- корисна інформація - інформація, яку важко відновити, проте комп'ютерна система або організація може ефективно функціонувати і без неї;

- несуттєва інформація.

Цінність інформації зазвичай змінюється з часом і залежить від міри відношення до неї різних груп осіб, що беруть участь в процесі обробки інформації:

- джерел, або постачальників, інформації;
- утримувачів або володарів, власників інформації;

- власників систем передачі, збору і обробки інформації;

- законних користувачів або споживачів інформації;

- порушників, прагнучих отримати інформацію, до якої в нормальних умовах не мають доступу.

Відношення цих груп осіб до значущості однієї і тієї ж інформації може бути різним: для однієї - важлива, для іншої - ні. Наприклад:

- важлива оперативна інформація, така, наприклад, як список замовлень на даний тиждень або графік виробництва, може мати високу цінність для утримувача, тоді як для джерела (наприклад, замовника) або порушника - низьку;

- персональна інформація, наприклад медична, має значно більшу цінність для джерела (обличчя, до якого відноситься інформація), чим для утримувача або порушника;

- інформація, використовувана керівництвом для вироблення рішень, наприклад про перспективи ринку, може бути значно ціннішою для порушника, чим для джерела або її утримувача, який вже завершив аналіз цих даних.

Існує ділення інформації по рівню секретності, конфіденційності. Ознаками секретної інформації є наявність:

- законних користувачів, які мають право володіти цією інформацією

- не законних користувачів (порушників, супротивників), які прагнуть опанувати цю інформацію з тим, щоб обернути її собі на благо, а законним користувачам в шкоду.

Для найбільш типових, ситуацій такого типу, що часто зустрічаються, введені навіть спеціальні поняття:

- державна таємниця

- військова таємниця

- комерційна таємниця

- юридична таємниця

- лікарська таємниця і тд.

Об'єктом захисту інформації являється комп'ютерна система (КС) або автоматизирована система обробки даних (АСОД) або автоматизована система обробки інформації (АСОІ). Будемо користуватися терміном комп'ютерна система (КС).

Комп'ютерна система - це комплекс апаратних і програмних засобів, призначених для автоматизованого збору, зберігання, обробки, передачі і отримання інформації.

Компоненти КС можна розбити на наступні групи:

- апаратні засоби - ПК і їх складові частини (процесори, монітори, термінали, периферійні пристрої-дисководи, принтери, контроллери, кабелі, лінії зв'язку) і так далі;

- програмне забезпечення - придбані програми, початкові, об'єктні, завантажувальні модулі; операційні системи і системні програми (компіляторы, компонувальники та ін.), утиліти, діагностичні програми і так далі;

- дані, що зберігаються тимчасово і постійно, на магнітних носіях, друкарські архіви, системні журнали і тд.;

- персонал - обслуговуючий персонал і користувачі.

При розгляді КС з точки зору захисту інформації слід звернути увагу, що комп'ютерні системи відносяться до класу людино-машинних систем.

Обслуговуючий персонал і користувачі можуть бути як об'єктом, так і джерелом несанкціонованої дії на інформацію.

Таким чином, забезпечення безпеки інформації в КС повинно передбачати захист усіх компонентів від зовнішніх і внутрішніх дій (загроз).

Під загрозою безпеки інформації розуміється потенційно можлива подія, процес або явище, які можуть привести до знищення, втрати цілосності, конфіденційності або доступності інформації.

Уся безліч потенційних загроз безпеки інформації в КС може бути розділена на два класи (рисунок 1).

Рисунок 1 – Загрози безпеки інформації в комп'ютерних системах

***Під системою захису інформації** в КС* розуміється єдиний комплекс правових норм, організаційних заходів, технічних, програмних і криптографічних засобів, що забезпечує захищеність інформації в КС відповідно до прийнятої політики безпеки.

ТЕРМІНОЛОГІЯ

Криптографічні методи захисту інформації є предметом дослідження сучасної криптології, що включає два основні розділи, цілі яких прямо протилежні:

- криптографія - наука, що вивчає способи перетворення (шифрування) інформації з метою її захисту від незаконних користувачів (супротивників);

- криптоаналіз – наука (і практика її застосування) про методи і способи розкриття шифрів.

Стенографія - набір засобів і методів приховання факту передачі повідомлення.

Стенографія приховує сам факт передачі повідомлення, а криптографія вважає, що повідомлення (у шифрованому виді) доступно незаконному користувачеві, але він не може витягнути з цього повідомлення інформацію, що захищається.

Комплексне використання стенографії і шифрування багаторазово підвищує складність рішення задачі виявлення і розкриття конфіденційності інформації.

Символ - це будь-який знак, у тому числі буква, цифра або розділовий знак.

Алфавіт - кінцева безліч використовуваних символів для шифрування інформації.

Наприклад, російський алфавіт містить 33 букви від А до Я (32 букви, виключаємо Ё і додаємо пропуск) – Z_{33}.

Алфавіт арабських цифр - це символи 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. Цей алфавіт містить 10 знаків і з його допомогою можна записати будь-яке натуральне число - $\mathbf{Z_{10} = \{0,1,2,3,4,5,6,7,8,9\}}$.

Будь-яке повідомлення може бути записане також за допомогою двійкового алфавіту, тобто з використанням тільки нулів і одиниць - $\mathbf{Z_2 = \{0,1\}}$.

Текст (повідомлення) – це впорядкований набір з елементів алфавіту.

Початкове повідомлення називають відкритим текстом (plaintext).

Відкритий текст - це не засекречена (не зашифрована) інформація, яка необхідна для передачі каналом зв'язку.

Повідомлення, отримане після перетворення з використанням будь-якого шифру, ***називається шифртекстом (закритим текстом, криптограмою, ciphertext).***

Шифр - спосіб, метод перетворення інформації з метою її захисту від незаконного користувача.

Шифр - сукупність оборотних перетворень безлічі можливих відкритих даних на безліч зашифрованих даних, що задаються ключем і алгоритмом криптографічного перетворення (криптоалгоритмом).

Ключ - змінний елемент шифру, який застосовується для шифрування конкретного повідомлення.

Ключ - конкретний секретний стан деяких параметрів алгоритму криптографічного перетворення даних, що забезпечує вибір одного перетворення з сукупності всіляких, для цього алгоритму, перетворень.

Зазвичай ключ є послідовним рядом символів алфавіту. Слід відрізняти поняття "ключ" і "пароль".

Пароль також є секретною послідовністю символів алфавіту, однак використовується не для шифрування (як ключ), а для аутентифікації суб'єктів.

Алгоритм криптографічний - алгоритм, що реалізує обчислення однієї з криптографічних функцій.

Функція криптографічна – функція, необхідна для реалізації криптографічної системи.

Криптографічна система, або шифрсистема - є сімейством оборотних перетворень відкритого тексту в шифрований.

Шифрування (enciphering) - процес застосування шифру до інформації, що захищається, тобто перетворення відкритого тексту в шифроване повідомлення (шифртекст) за допомогою певних правил, що містяться в шифрі.

Дешифрування (deciphering) - процес перетворення зашифрованих даних (шифртекста) у відкритий текст за допомогою певних правил, що містяться в шифрі.

Розкриття (взлом) шифру - процес перетворення закритих даних у відкриті, при невідомому ключі і/або невідомому алгоритмі.

Методи кодування спрямовані не на те, щоб приховати відкрите повідомлення, а на те, щоб подати його в зручнішому виді для передачі засобами зв'язку, для зменшення довжини повідомлення, зменшення надмірності, підвищення пропускної спроможності каналу.

Кодування це представлення інформації у вигляді символів алфавіту.

Простір відкритих текстів PT - безліч всіляких початкових повідомлень pt(plaintext). (Для повідомлень використовується також позначення m(message))

Простір ключів K - набір можливих значень ключа.

Кожен ключ $k \in K$ визначає деяку підстановку (алгоритм шифрування) Ek (enciphering, encryption) на просторі PT і зворотне перетворення (дешифрування) Dk(deciphering, decryption).

Простір шифртекстів CT - складається з усіх шифртекстів ct (ciphertext). Використовується також позначення c.

Криптосистема складається з простору ключів, простору відкритих текстів, простору шифртекстів, алгоритмів шифрування і дешифрування.

Криптостійкістю називається характеристика шифру, що визначає його стійкість до дешифрування без знання ключа (стійкість до взлому).

Контрольні питання:

1. Розкрити поняття й визначення курсу.
2. Визначити поняття інформаційної безпеки системи.
3. Перелічити й розкрити завдання захисту користувальницької інформації.
4. Перелічити способи рішення завдання забезпечення безпеки в інформаційно-телекомунікаційних системах
5. Що є об’єктом захисту інформації.
6. Що таке комп’ютерна система.

7. Компоненти комп’ютерної системи.
8. Що таке шифрування та дешифрування.

Лекція №3 (2 години)

Тема: Принципи криптографічного захисту. Класифікація методів криптографічного перетворення інформації.

Мета: Ознайомитися з поняттями криптографічного захисту інформації та його принципами.

ПЛАН

1. Основні завдання захисту інформації призначеної для користувача
2. Види атак
3. Умови криптографічної стійкості по К.Шенону
4. Причини здійснення успішних атак.

Основними завданнями захисту інформації призначеної для користувача є:

- забезпечення конфіденційності інформації;

- забезпечення цілісності інформації;

- забезпечення достовірності інформації;

- забезпечення оперативності доступу до інформації;

- забезпечення юридичної значущості інформації, уявленої про вид електронного документу;

- забезпечення невідстежуваних дій клієнта.

Основу забезпечення інформаційної безпеки комп'ютерних системах складають криптографічні методи і засоби захисту інформації.

Криптографічні алгоритми повинні забезпечувати:

1. Конфіденційність - властивість інформації бути доступною тільки обмеженому колу користувачів інформаційної системи, в якій циркулює ця інформація.

2. Цілісність - властивість інформації або програмного забезпечення зберігати свою структуру і/або зміст в процесі передачі і/або зберігання.

Автентичність інформації - достовірність авторства і цілісності.

Аутентифікація джерела даних - перевірка і підтвердження того, що набір даних (повідомлення, документ) був створений саме заявленим джерелом. Припускає перевірку цілісності.

3. Невідстежуваність - властивість, що означає неможливість отримання противником і/або порушником відомостей про дії учасників (протоколу).

4. Криптостійкість

Засновником криптографії є Клод Шеннон. Він розглянув узагальнену модель криптографічної системи:

1. Джерело повідомлень (посилач) генерує з алфавіту А повідомлення (відкритий текст) М, який має бути переданий законному одержувачеві по незахищеному каналу зв'язку.

2. Джерело ключів генерує ключ К.

3. Шифратор робить шифрування E_K , тобто перетворює відкритий текст **М** за допомогою ключа **К** в шифртекст **С**.

Робиться перетворення З=f(M, K) або $C = E_K(M)$

де f або Е - функція шифрування.

4. Посилач відправляє шифртекст одержувачеві

5. Законний одержувач, прийнявши шифртекс **С** проводить дешифрування $\mathbf{D = E_K^{-1}}$ і отримує початкове повідомлення у вигляді відкритого тексту **М**.

Робиться перетворення М=f(C, K) або $\mathbf{M = D_K(C) = E_K^{-1}(E_K(M))}$

де f або D - функція дешифрування.

6. За каналом стежить перехоплювач, метою якого є перехопити, розкрити або змінити передане повідомлення.

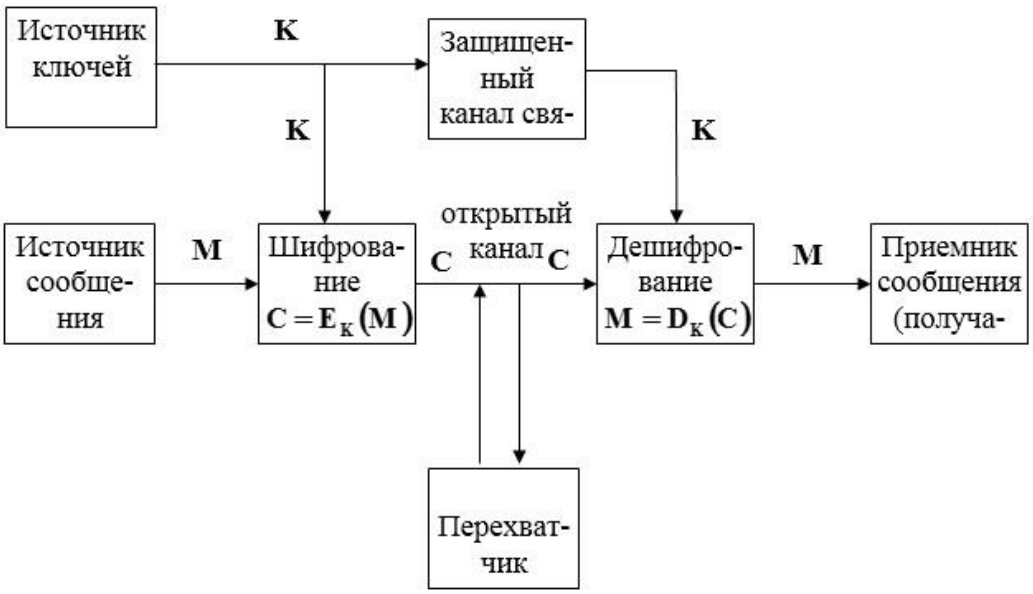


Рисунок 2 - Узагальнена модель криптографічної системи

Перетворення $\mathbf{E_K}$ вибирається з сімейства криптографічних перетворень, що називаються криптоалгоритмами.

Будь-яка спроба з боку перехоплювача розшифрувати шифртекст **C** для отримання відкритого тексту **M** або зашифрувати свій власний текст **M'** для отримання правдоподібного шифртекста **C**, не маючи справжнього ключа, ***називається криптоаналітичною атакою.***

Під атакою на шифр розуміють процес розшифрування закритої інформації без знання ключа і, можливо, за відсутності відомостей про алгоритм.

Якщо зроблені криптоаналітичні атаки не досягають поставленої мети і криптоаналітик не може, не маючи справжнього ключа, вивести **M** із **C** або **C** з **M'**, то вважають, що така криптосистема є криптостійкою.

Правило Кірхгофа свідчить, що стійкість шифру (криптосистеми) забезпечується секретністю ключа, а не секретністю алгоритму шифрування.

Іншими словами, правило Кірхгофа полягає в тому, що увесь алгоритм шифрування, окрім значення секретного ключа, відомий криптоаналітику супротивника.

Такий підхід відбиває дуже важливий принцип технології захисту інформації: захищеність системи не повинна залежати від секретності чого-небудь такого, що неможливо швидко змінити у разі витоку секретної інформації.

Стійкість криптографічного алгоритму необхідно розглядати відносно пари "атака-мета", де під метою супротивника розуміється планована загроза.

Класифікація різних типів атак на криптографічні алгоритми:

- атака з відомим шифртекстом (ciphertext - only attack). Передбачається, що супротивник знає алгоритми шифрування, але не знає секретний ключ. Крім того, йому відомий лише набір перехоплених криптограм. Робота криптоаналітика полягає в тому, щоб розкрити початкові тексти по можливості більшість повідомлень або, ще краще, вичислити ключ, який використовується для шифрування цих повідомлень, з тим, щоб розшифрувати і інші повідомлення, зашифровані цим ключем.;

- атака з відомим відкритим текстом (known plaintext attack). Те ж, що попередня, але супротивник отримує у своє розпорядження ще деякий набір криптограм і відкритих текстів, що відповідають їм. Його робота заключається в знаходженні ключа, що використовується при шифруванні цих повідомлень, або алгоритму раозшифровки будь-яких нових повідомлень, зашифрованих тим самим ключем;

- проста атака з вибором відкритого тексту (chosen - plaintext attack). Противник має можливість вибрати необхідну кількість відкритих текст і отримати ів криптограми, що відповідають їм. Такий криптоаналіз виходить потужнішим в порівнянні з криптоаналізом з відомим відкритим текстом, тому що криптоаналітик може вибрати для шифрування такі блоки відкритого тексту, які дадуть більше інформації про ключ.

- адаптивна атака з вибором відкритого тексту (adaptive - chosen - plaintext attack). В цьому випадку супротивник має можливість вибирати відкриті тексти з урахуванням того, що криптограми усіх попередніх відкритих текстів йому відомі.

При криптоаналізі з простим вибором відкритого тексту криптоаналітик зазвичай може вибирати декілька великих блоків відкритого тексту для їх шифрування; при криптоаналізі з адаптивним вибором відкритого тексту він має можливість вибрати спочатку дрібніший пробний блок відкритого тексту, потім вибрати наступний блок залежно від результатів першого вибору, і так далі

- атака з вибором шифртекста (chosen - ciphertext attack). Супротивник має можливість вибрати необхідну кількість криптограм і отримати відповідні їм відкриті тексти. Наприклад, криптоаналітик отримав доступ до захищеного від несанкціонованого розкриття блоку, який виконує автоматичне розшифрування. Робота криптоаналітика полягає в знаходженні ключа. Цей тип криптоаналізу представляє особливий інтерес для розкриття алгоритмів з відкритим ключем;

- атака з вибором ключа (chosen - key attack). Ця атака припускає використання криптоаналітиком відомого шифртекста і здійснюється засобом повного перебору усіх можливих ключів з перевіркою, чи є осмисленим відкритий текст, що виходить. Такий підхід вимагає прилучення граничних обчислювальних ресурсів і іноді *називається силовою атакою.*

У цьому переліку атаки представлені у міру зростання їх сили.

Аналізуючи атаки і їх цілі, можна дійти висновку, що найбільшою стійкістю алгоритм володіє у тому випадку, коли він здатний протистояти самій сильній атаці,

що проводиться супротивником, за умови, що він переслідує найслабкішу з можливих цілей атаки (загрозу).

Теоретично існують абсолютно стійкі алгоритми шифрування.

Умови абсолютної стійкості шифру (по К. Шенону) :

- довжина ключа і довжина відкритого повідомлення мають бути однакові;

- ключ повинен використовуватися тільки один раз;

- вибір ключа з ключового простору повинен здійснюватися рівновірогідно.

Єдиним таким шифром є яка-небудь форма так званої стрічки одноразового використання, в якій відкритий текст "об'єднується" з повністю випадковим ключем такої ж довжини.

У разі порушення хоча б однієї з цих умов шифр перестає бути абсолютно стійким, і з'являються принципові можливості для його розкриття (хоча вони можуть бути такими, що важко реалізовуються).

Але ці умови і роблять абсолютно стійкий шифр дуже дорогим і непрактичним. Перш ніж користуватися таким шифром, ми повинні забезпечити усіх абонентів достатнім запасом випадкових ключів великої довжини і унеможливити їх повторне застосування. А це зробити надзвичайно важко і дорого.

Тепер уже зрозуміло, що найчастіше для захисту своєї інформації законні користувачі вимушені застосовувати неабсолютно стійкі шифри. Такі шифри, принаймні теоретично, можуть бути розкриті. Питання тільки в тому, чи досить у супротивника сил, засобів і часу для розробки і реалізації відповідних алгоритмів. Зазвичай цю думку виражають так: супротивник з необмеженими ресурсами може розкрити будь-який неабсолютно стійкий шифр.

Найпоширеніші на сьогодні причини здійснення успішних атак на алгоритми шифрування :

- наявність статистичної структури мов, що історично склалися. Тобто існують певні символи або комбінації символів, що найчастіше зустрічаються в природній мові. Таким чином, при перехопленні зашифрованного повідомлення для деяких типів алгоритмів шифрування, можна підрахувати частоту появи певних символів і зіставити їх з вірогідністю появи певних символів або їх комбінацій (біграми, триграми і так далі)

- наявність вірогідних слів. Йдеться про слова або вирази, появу яких можна чекати в перехопленому повідомленні. Так, в діловому листуванні є присутніми шаблонні слова; у англійській мові, наприклад, найбільш часто зустрічаються "and", the", "are" і так далі.

При побудові стійких шифрів необхідно використати два основних принципа - розсіювання і перемішування.

Розсіювання припускає поширення впливу одного знаку відкритого тексту на безліч знаків шифротекста, що дозволяє приховати статистичні властивості відкритого тексту.

Перемішування припускає використання таких шифруючих перетворень, які ускладнюють відновлення взаємозв'язку статистичних властивостей відкритого і шифртекста.

Розвитком принципу розсіювання став принцип заплутування, в якому вплив одного символу ключа поширюється на безліч символів зашифрованого повідомлення.

Висунена ідея створення складеного шрифту, що задовольняє принципам розсіювання і перемішування. Такий шифр, реалізований у вигляді послідовності простих традиційних шифрів, кожен з яких вносить свій вклад в сумарне розсіювання і перемішування. Найчастіше при цьому використовують традиційні шифри перестановки і заміни.

При багатократному чергуванні простих перестановок і замін, керованих достатньо довгим секретним ключем, можна отримати дуже стійкий шифр з хорошим розсіюванням і перемішуванням. Більшість існуючих стандартів шифрування побудована в повній відповідності з цією методологією.

Класифікація методів шифрування (криптоалгоритмів) може бути здійснена за наступними ознаками:

1. За типом ключів:

а) симетричні криптоалгоритми;

б) асиметричні криптоалгоритми;

2. За способом перетворення :

а) метод заміни;

б) метод перестановки;

б) аналітичні методи;

в) шифри гамміровання;

г) комбіновані методи.

Під симетричними криптографічними системами розуміються такі криптосистеми, в яких для шифрування і дешифрування використовується один і той же ключ, що зберігається в секреті.

Криптографічний алгоритм є симетричним, якщо для шифрування і дешифрування використовується один і той же ключ, який є закритим.

Основним недоліком симетричного шифрування є те, що секретний ключ має бути відомий і відправнику, і одержувачу. З одного боку, це ставить нову проблему

розсилки ключів. З іншого боку, одержувач на підставі наявності шифрованого і розшифрованого повідомлення не може довести, що він отримав це повідомлення від конкретного відправника, оскільки таке ж повідомлення він міг згенерувати.

У ассиметричных криптосистемах (з відкритим ключем) для зашифрування використовується один ключ, а для розшифровки - інший ключ. Перший ключ є відкритим і може бути опублікований для використання усіма користувачами системи. Для розшифровки одержувач інформації використовує секретний ключ.

Симетричні алгоритми можна розділити за розміром блоку інформації:

а) потокові шифри;

б) блокові шифри;

в) складені.

Потокові шифри - за один раз обробляється один елемент даних (біт або буква).

Блокові шифри - за один крок обробляється група елементів даних. (наприклад 64 біт).

Блокові шифри - один блок відкритого тексту замінюється одним блоком закритого тексту.

Шифри перестановки - змінюють тільки порядок отримання символів або інших елементів початкового тексту.

При шифруванні перестановкою символи шифрованого тексту переставляються за відповідним правилом в межах блоку цього тексту.

При шифруванні заміною символи шифрованого тексту замінюються символами того ж або іншого алфавіту із заздалегідь встановленим правилом заміни.

Шифрування гамміюванням полягає в тому, що символи шифрованого тексту складаються з символами деякої випадкової послідовності, що іменується гаммою шифру. Стійкість шифрування визначається в основному завдовжки (періодом) частини гамми шифру, що не повторюється. Оскільки за допомогою ПК можна генерувати практично нескінченну гамму шифру, то цей спосіб є одним з основних для шифрування інформації в автоматизированных системах.

Шифрування аналітичним перетворенням полягає в тому, що шифрований текст перетвориться за деяким аналітичним правилом (формулі).

Основною характеристикою шифру є криптостійкість, яка визначає його стійкість до розкриття методами криптоаналізу.

Зазвичай ця характеристика визначається інтервалом часу, необхідним для розкриття шифру.

До шифрів, використовуваних для криптографічного захисту інформації, пред'являється ряд вимог:

- достатня криптостійкість (надійність закриття даних);
- простота процедур шифрування і розшифрування;
- незначна надмірність інформації за рахунок шифрування;
- нечутливість до невеликих помилок шифрування та ін.

В тій чи іншій мірі цим вимогам відповідають:

- шифри перестановок;
- шифри заміни;
- шифри гамміювання;
- шифри, грунтовані на аналітичних перетвореннях шифрованих даних.

Контрольні питання:

1. Що таке надійність та конфіденційність.
2. Що таке аутентифікація.
3. Умови стійкості шифрів.
4. Що таке атака на шифр, які вони бувають.
5. Симетричні та ассиметричні шифри.
6. Потокові та блокові шифри.
7. Методи перестановки та заміни.
8. Вимоги до шифрів.

Лекція №4 (2 години)

Тема: Симетричні криптосистеми. Шифри перестановки. Шифр маршрутної перестановки. Шифр поодинокої перестановки по ключу.

Мета: Ознайомитися з симетричними криптосистемами.

ПЛАН

1. Блочний шифр як приклад симетричної криптосистеми
2. Режими застосування блочних шифрів

Для шифрування вихідного тексту довільної довжини блочні шифри можуть використовуватись в декількох режимах. Існує чотири основних режими застосування блочних шифрів, що найбільш часто зустрічаються в системах криптографічного захисту інформації. Це режими *електронної кодувальної книги* (*ECB*), *зчеплення блоків шифрованого тексту* (*CBC*), *зворотного зв'язку по шифрованому тексту* (*CFB*) і *зворотного зв'язку по виходу* (*OFB*).

В режимі *електронної кодувальної книги* кожен блок вихідного тексту шифрується блоковим шифром незалежно від інших. Стійкість режиму *ECB* дорівнює стійкості самого шифру. Однак, структура вихідного тексту при цьому не приховується. Кожен однаковий блок вихідного тексту приводить до появи однакового блоку шифрованого тексту. Вихідним текстом можна легко маніпулювати шляхом видалення, чи повторення перестановки блоків. Швидкість шифрування дорівнює швидкості блочного шифру. Режим *ECB* допускає просте розпаралелювання для збільшення швидкості шифрування.

В режимі *зчеплення блоків шифрованого тексту (CBC)* кожен блок вихідного тексту складається порозрядно по модулю 2 з попереднім блоком шифрованого тексту, а потім шифрується. Для початку процесу шифрування використовується *синхропосилка* (початковий вектор), що передається в канал зв'язку у відкритому виді. Стійкість режиму *CBC* дорівнює стійкості блочного шифру, що лежить у його основі. Структура вихідного тексту приховується за рахунок додавання попереднього блоку шифрованого тексту з черговим блоком відкритого тексту. Стійкість шифрованого тексту збільшується, оскільки стає неможливою пряма маніпуляція вихідним текстом. Швидкість шифрування дорівнює швидкості роботи блочного шифру, але простого способу розпаралелювання процесу шифрування не існує, хоча розшифровка може проводитися паралельно.

В режимі *зворотного зв'язку по шифрованому тексті (CFB)* попередній блок шифрованого тексту шифрується ще раз, і для одержання чергового блоку шифрованого тексту результат складається порозрядно по модулю 2 з блоком вихідного тексту. Для початку процесу шифрування також використовується початковий вектор. Стійкість режиму *CFB* дорівнює стійкості блочного шифру, що лежить у його основі. Структура вихідного тексту приховується за рахунок використання операції додавання по модулю 2. Маніпулювання вихідним текстом шляхом видалення блоків з початку чи кінця шифрованого тексту неможлививе. В режимі *CFB* якщо два блоки шифрованого тексту ідентичні, то результати їхнього шифрування на наступному кроці також будуть ідентичні, що створює можливість витоку інформації про вихідний текст. Швидкість шифрування дорівнює швидкості роботи блочного шифру. Простого способу розпаралелювання процесу шифрування не існує.

Режим *зворотного зв'язку по виходу* (*OFB*) подібний режиму *CFB* за винятком того, що величини, що складаються по модулю 2 з блоками вихідного тексту, генеруються незалежно від вихідного чи шифрованого тексту. Для початку процесу шифрування також використовується початковий вектор. Режим *OFB* має перевагу перед режимом *CFB* у тому, що будь-які бітові помилки, що виникли в процесі передачі, не впливають на розшифровку наступних блоків. Однак, можлива проста маніпуляція вихідним текстом шляхом зміни шифрованого тексту. Хоча в цьому випадку простого способу розпаралелювання процесу шифрування також не існує, час

можна заощадити, виробивши ключову послідовність заздалегідь.

При шифруванні перестановкою символи шифрованого тексту переставляються за певним правилом в межах блоку цього тексту.

Шифр Сцитала (Скитала)

У Спарті в V - IV віках до н.е. використовувалося одно з перших шифрувальних засобів - Сцитала. (скитала або сцитала від грецького σκυτάλη, жезл).

Це був жезл циліндричної форми, на який намотувалася смужка пергаменту.

Уздовж осі циліндра на пергамент відрядковий записувався текст, після цього стрічка змотувалась з жезла і передавалася адресатові, який мав таку саму Сциталу.

Такий спосіб шифрування здійснював перестановку букв в повідомленні, ключем до котрого служив діаметр Сцитали.

Метод розкриття цього шифру приписується Арістотелю. Він запропонував ув'язнити конусом довгий брус і, обернувши навколо нього стрічку, почати зрушувати її по конусу від малого діаметру до найбільшого. У тому місці, де діаметр конуса співпадав з діаметром Сцитали букви складалися в склади і слова. Після цього залишалося тільки виготовити жезл потрібного діаметру.

Шифр "Сцитала" реалізує не більше за n перестановки (n - довжина повідомлення). Є ще і чисто фізичні обмеження, що накладаються реалізацією шифру "Сцитала".

Наприклад, використовуючи паличку, по довжині кола якої поміщається 4 символи, а довжина палички дозволяє записати 6 символів, початковий текст :

" **ЭТО ШИФР ДРЕВНЕЙ СПАРТЫ** "

Схематично це можна зображувати так:

		Э **Ф** **В** **П**	**Т** **Р** **Н** **А**	**О** **Д** **Е** **Р**	**Ш** **Р** **Й** **Т**	**И** **Е** **С** **Ы**	

Таким чином, після розмотування стрічки, шифротекст буде наступним: «**ЭФВПТРНАОДЕРШРЙТИЕСЫ**».

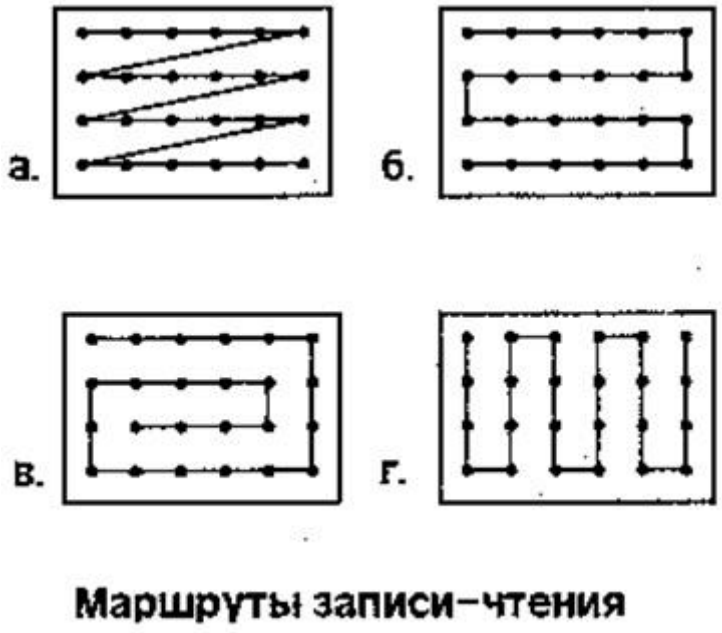


Рисунок 3 – Маршрути запису та читання

Шифруючі таблиці

У шифрах перестановки початку епохи Відродження (кінець XIV століття) часто застосовували шифруючі таблиці, які по суті задають правила перестановки букв в повідомленні.

В якості ключа в шифруючих таблицях використовуються:

- розмір таблиці;

- слово або фраза, задаючі перестановку

- особливості структури таблиці.

Перетворення з цього шифру полягають в тому, що у фігуру вписується по ходу одного "маршруту" початковий текст, а потім по ходу іншого виписується з неї.

Шифр маршрутної перестановки

Шифр перестановки, для якого ключем служить розмір таблиці і маршрут перестановки (особливості структури).

Алгоритм шифрування

Наприклад, зашифруємо повідомлення **ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ**

Використовуємо алфавіт з 32 символів (32 російських букви російського алфавіту).

1) Початкове повідомлення записується в таблицю по горизонталі, починаючи з лівого верхнього кута по черзі, зліва направо і справа наліво (маршрут б на рис 1).

У початковому повідомленні 35 символів, тому вибираємо таблицю з 5 рядків і 7 стовпців (5*7=35 символів). Результат заповнення таблиці показаний на рис.

Т	**Е**	**Р**	**М**	**И**	**Н**	**А**
Б	**И**	**Р**	**П**	**Р**	**О**	**Т**
Ы	**В**	**А**	**Е**	**Т**	**С**	**Е**
В	**О**	**Г**	**О**	**М**	**Ь**	**Д**
П	**О**	**Л**	**Н**	**О**	**Ч**	**Ь**

2) Для формування шифртекста прочитують вміст таблиці по вертикалі, починаючи з верхнього правого кута і рухаючись по черзі зверху вниз і від низу до верху (маршрут г на рис. 3).

Якщо шифртекст записувати групами по п'ять букв, виходить таке шифроване повідомлення: **АТЕДЬ ЧЬСОН ИРТМО НОЕПМ РРАГЛ ООВИЕ ТБЫВП**

Звісно, посилач і одержувач повідомлення повинні заздалегідь умовитися про загальний ключ у вигляді розміру таблиці і маршрут. Об'єднання букв шифртекста в 5-буквені групи не входить в ключ шифру і здійснюється для зручності запису несмислового тексту.

Алгоритм дешифрування

При розшифруванні дії виконують в зворотному порядку.

Дещо більшу стійкість до розкриття має метод шифрування, що називається поодинокою перестановкою по ключу.

Шифр поодинокої перестановки по ключу

Цей метод відрізняється від попереднього тим, що стовпці таблиці переставляються по ключовому слову, фразі або набору чисел завдовжки в рядок таблиці.

Алгоритм шифрування

Зашифруємо повідомлення **ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ**

В якості ключа використаємо слово **ПЕЛИКАН.**

Використовуємо алфавіт з 32 символів(32 російських букви російського алфавіту).

1	2	3	4	5	6	7	8	9	10
А	**Б**	**В**	**Г**	**Д**	**Е**	**Ж**	**З**	**И**	**Й**

11	12	13	14	15	16	17	18	19	20
К	**Л**	**М**	**Н**	**О**	**П**	**Р**	**С**	**Т**	**У**

21	22	23	24	25	26	27	28	29	30
Ф	**Х**	Ц	**Ч**	**Ш**	**Щ**	**Ъ**	**Ы**	**Ь**	**Э**

31	32

Ю	Я

1) По ключу і початковому повідомленню формуються таблиця.

Кількість стовпців в таблиці відповідає кількості символів ключового слова.

$$\mathbf{n_{столб}} = \ell_{\mathbf{k}}.$$

В нашому випадку $\mathbf{n_{столб}} = 7$

Кількість рядків в таблиці визначається відношенням кількості символів відкритого повідомлення до довжини ключового слова плюс два рядки (одна на ключове слово, друга для нумерації символів ключа).

$$\mathbf{n_{строк}} = \frac{\ell_{\mathbf{m}}}{\ell_{\mathbf{k}}} + 2$$

$$\mathbf{n_{строк}} = \frac{35}{7} + 2 = 5 + 2 = 7$$

2) Заповнюється ліва таблиця:

а) перший рядок - ключ;

б) другий рядок - записуються номери букв ключового слова, які визначені відповідно до природного порядку їх розташування в алфавіті

в) далі початкове повідомлення записується в таблицю по черзі по стовпцях.

3) В правій таблиці стовпці переставлені відповідно до впорядкованих номерів букв ключа.

На рис 4 показані дві таблиці, заповнені текстом повідомлення і ключовим словом, при цьому ліва таблиця відповідає заповненню до перестановки, а права таблиця - заповненню після перестановки.

ПЕЛИКАН.

П	Е	Л	И	К	А	Н
7	2	5	3	4	1	6
Т	Н	П	В	Е	Г	Л
Е	А	Р	А	Д	0	Н
Р	Т	И	Е	Ь	В	О
М	о	Б	Т	М	П	Ч
И	Р	Ы	С	О	О	Ь

До перестановки

А	Е	И	К	Л	Н	П
1	2	3	4	5	6	7
Г	Н	В	Е	П	Л	Т
0	А	А	Д	Р	Н	Е
В	Т	Е	Ь	И	0	Р
П	0	Т	М	Б	Ч	М
О	Р	С	О	Ы	Ь	И

После перестановки

Рисунок 4 – Таблиці заповнені ключовим словом і текстом повідомлення

4) Для формування шифртекста прочитують вміст правої таблиці по рядках (рядки, де знаходилося початкове повідомлення).

При записі шифртекста групами по п'ять букв отримаємо шифроване повідомлення: **ГНВЕП ЛТОАА ДРНЕВ ТЕЬИО РПОТМ БЧМОР СОЫЬИ**

Примітка

Якби в ключі зустрілися однакові букви, вони б були пронумеровані зліва направо.

Алгоритм дешифрування

Наприклад:

шифртекст

ЬЛНМФТИТОНААЕНЕГЫЦКХФТИХИ УНО_Й _ОЛОР

ключ: КАФЕДРА

Використовуємо алфавіт з 33 символів (32 російських букви російського алфавіту і пропуск.

1	2	3	4	5	6	7	8	9	10
А	**Б**	**В**	**Г**	**Д**	**Е**	**Ж**	**З**	**И**	**Й**

11	12	13	14	15	16	17	18	19	20
К	**Л**	**М**	**Н**	**О**	**П**	**Р**	**С**	**Т**	**У**

21	22	23	24	25	26	27	28	29	30
Ф	**Х**	Ц	**Ч**	**Ш**	**Щ**	**Ъ**	**Ы**	**Ь**	**Э**

31	32	пробел
Ю	**Я**	_

1) По ключу і шифртексту формуються таблиця.

Кількість стовпців в таблиці відповідає кількості символів ключового слова.

$$\mathbf{n_{столб} = \ell_k}$$

В нашому випадку $\mathbf{n_{столб} = 7}$

Кількість рядків в таблиці визначається відношенням кількості символів шифртекста до довжини ключового слова плюс два рядки(при підрахунку кількості символів, відкидаємо пробіли між групами з п'яти символів).

$$\mathbf{n_{строк} = \frac{\ell_c}{\ell_k} + 2}$$

В нашому випадку $\mathbf{n_{строк} = \frac{35}{7} + 2 = 5 + 2 = 7}$

2) Заповнюється права таблиця:

а) перший рядок - символи ключа по порядку в алфавіті;

б) другий рядок - записуються номери букв ключового слова, які визначені відповідно до природного порядку їх розташування в алфавіті

в) далі шифртекст записується в таблицю по черзі по рядках.

3) В лівій таблиці стовпці переставлені так, щоб у верхньому рядку вийшло ключове слово.

К	**А**	**Ф**	**Е**	**Д**	**Р**	**А**
5	1	7	4	3	6	2
Ф	Ь	И	М	Н	Т	Л
А	Т	Н	А	Н	Е	О
К	Е	Ф	Ц	Ы	Х	Г
У	Т	О	И	Х	Н	И
Л	_	Р	О	_	О	Й
левая						

А	**А**	**Д**	**Е**	**К**	**Р**	**Ф**
1	2	3	4	5	6	7
Ь	Л	Н	М	Ф	Т	И
Т	О	Н	А	А	Е	Н
Е	Г	Ы	Ц	К	Х	Ф
Т	И	Х	И	У	Н	О
_	Й	_	О	Л	О	Р
правая						

←————————————————————

4) Для формування початкового повідомлення прочитують вміст лівої таблиці по стовпцях (рядки, де знаходився шифртекст).

Початкове повідомлення:

ФАКУЛЬТЕТ_ИНФОРМАЦИОННЫХ_ТЕХНОЛОГИЙ

Шифр вертикальної перестановки

Алгоритм шифрування

Використовуємо алфавіт з 32 символів (32 російських букви російського алфавіту).

Застосуємо в якості ключа, наприклад, слово **ПЕЛИКАН,** а текст повідомлення **ТЕРМИНАТОР ПРИБЫВАЕТ СЕДЬМОГО В ПОЛНОЧЬ**

1) По ключу і початковому повідомленню формуються таблиця.

Кількість стовпців в таблиці відповідає кількості символів ключового слова.

$$\mathbf{n_{столб} = \ell_k}$$

В нашому випадку $\mathbf{n_{столб} = 7}$

Кількість рядків в таблиці визначається відношенням кількості символів відкритого повідомлення до довжини ключового слова плюс два рядки(одна на ключове слово, друга для нумерації символів ключа)

$$n_{строк} = \frac{\ell_m}{\ell_k} + 2$$

В нашому випадку $\mathbf{n_{строк} = \frac{35}{7} + 2 = 5 + 2 = 7}$

2) Заповнюється ліва таблиця:

а) перший рядок - ключ;

б) другий рядок - записуються номери букв ключового слова, які визначені відповідно до природного порядку їх розташування в алфавіті

в) далі початкове повідомлення записується в таблицю по черзі по рядках.

3) В правій таблиці стовпці переставлені відповідно до впорядкованих номерів букв ключа.

П	**Е**	**Л**	**И**	**К**	**А**	**Н**
7	2	5	3	4	1	6
Т	Е	Р	М	И	Н	А
Т	О	Р	П	Р	И	Б
Ы	В	А	Е	Т	С	Е
Д	Ь	М	О	Г	О	В
П	О	Л	Н	О	Ч	Ь
левая до перестановки						

А	**Е**	**И**	**К**	**Л**	**Н**	**П**
1	2	3	4	5	6	7
Н	Е	М	И	Р	А	Т
И	О	П	Р	Р	Б	Т
С	В	Е	Т	А	Е	Ы
О	Ь	О	Г	М	В	Д
Ч	О	Н	О	Л	Ь	П
правая после перестановки						

——————————————————————————————→

4) Для формування шифртекста прочитують вміст правої таблиці по стовпцях(рядки, де знаходилося початкове повідомлення).

При записі шифртекста групами по п'ять букв отримаємо шифроване повідомлення:

НИСОЧ ЕОВЬО МПЕОН ИРТГО РРАМЛ АБЕВЬ ТТЫДП

Примітка: Якби в ключі зустрілися однакові букви, вони б були пронумеровані зліва направо.

Алгоритм дешифрування

Використовуємо алфавіт з 33 символів(32 російських букви російського алфавіту і пропуск.

Наприклад:

ключ: кафедра

шифртекст:

АТМЫО ТОНХЙ ЛНИТГ УИЦ_О ФЕРНН ЬФОЕИ К_АХЛ

1) По ключу і шифртексту формуються таблиця.

Кількість стовпців в таблиці відповідає кількості символів ключового слова.

$$\mathbf{n_{столб}} = \ell_{\mathbf{k}}$$

В нашому випадку $\mathbf{n_{столб} = 7}$

Кількість рядків в таблиці визначається відношенням кількості символів шифртекста до довжини ключового слова плюс два рядки(при підрахунку кількості символів, відкидаємо пробіли між групами з п'яти символів)

$$n_{строк} = \frac{\ell_c}{\ell_k} + 2$$

В нашому випадку $\mathbf{n_{строк} = \frac{35}{7} + 2 = 5 + 2 = 7}$

2) Заповнюється права таблиця:

а) перший рядок - символи ключа по порядку в алфавіті;

б) другий рядок - записуються номери букв ключового слова, які визначені відповідно до природного порядку їх розташування в алфавіті

в) далі шифртекст записується в таблицю по черзі по стовпцях.

3) В лівій таблиці стовпці переставлені так, щоб у верхньому рядку вийшло ключове слово.

К	А	Ф	Е	Д	Р	А
5	1	7	4	3	6	2
Ф	А	К	У	Л	Ь	Т
Е	Т	_	И	Н	Ф	О
Р	М	А	Ц	И	О	Н
Н	Ы	Х	_	Т	Е	Х
Н	О	Л	О	Г	И	Й
левая						

А	А	Д	Е	К	Р	Ф
1	2	3	4	5	6	7
А	Т	Л	У	Ф	Ь	К
Т	О	Н	И	Е	Ф	_
М	Н	И	Ц	Р	О	А
Ы	Х	Т	_	Н	Е	Х
О	Й	Г	О	Н	И	Л
правая						

←————————————————

4) Для формування початкового повідомлення прочитують вміст лівої таблиці по рядках(рядки, де знаходився шифртекст).

Початкове повідомлення:

ФАКУЛЬТЕТ_ИНФОРМАЦИОННЫХ_ТЕХНОЛОГИЙ

Контрольні питання:

1. Опишіть блочний шифр як приклад симетричної криптосистеми
2. Перелічить режими застосування блочних шифрів
3. Що таке шифри перестановки.
4. Шифруючі таблиці.
5. Що таке маршрутна перестановка.
6. Алгоритм шифрування та дешифрування.
7. Що таке поодинока перестановка по ключу.
8. Алгоритм шифрування та дешифрування.

Лекція №5 (2 години)

Тема: Метод шифрування подвійною перестановкою.

Мета: Ознайомитися з алгоритмами шифрування та дешифрування у шифрі подвійної перестановки.

ПЛАН

1. Шифр подвійної перестановки
2. Алгоритм шифрування
3. Алгоритм дешифрування

Для забезпечення додаткової скритності можна повторно зашифрувати повідомлення, яке вже пройшло шифрування. Такий метод шифрування називається подвійною перестановкою. У разі подвійної перестановки стовпців і рядків таблиці перестановки визначаються окремо для стовпців і окремо для рядків.

Алгоритм шифрування

Початкове повідомлення

ПРИЛЕТАЮ ВОСЬМОГО

Ключем до шифру подвійної перестановки служить послідовність номерів стовпців і номерів рядків початкової таблиці (у нашому прикладі послідовності 4132 і 3142 відповідно).

$$\mathbf{k_{столб} = 4132 \ \ k_{строк} = 3142}$$

1) Спочатку в початкову таблицю відрядковий записується текст повідомлення

Рис. 2.3. Пример выполнения шифрования методом двойной перестановки

2) По черзі переставляються стовпці

3) По черзі переставляються рядки

4) Прочитують шифртекст з правої таблиці(таблиця перестановки рядків) відрядковими блоками по чотири букви:

ТЮАЕ ООГМ РЛИП ОЬСВ

При розшифровці порядок перестановок має бути зворотним.

Алгоритм дешифрування

Шифртекст

АЗЮЖ Е_СШ ГТОО ИПЕР

Ключем до шифру подвійної перестановки служить послідовність номерів стовпців і номерів рядків початкової таблиці(у нашому прикладі послідовності 2413 і 4123 відповідно).

$$\mathbf{k_{столб} = 2413 \;\; k_{строк} = 4123}$$

1) Спочатку в праву таблицю перестановок рядків записується текст шифртекста відрядковий

	2	4	1	3
4	П	Р	И	Е
1	З	Ж	А	Ю
2	_	Ш	Е	С
3	Т	О	Г	О

Исходное сообщение

Левая таблица

	1	2	3	4
4	И	П	Е	Р
1	А	З	Ю	Ж
2	Е	_	С	Ш
3	Г	Т	О	О

Перестановка строк

Правая таблица

	1	2	3	4
1	А	З	Ю	Ж
2	Е	_	С	Ш
3	Г	Т	О	О
4	И	П	Е	Р

2) По черзі переставляються рядки

3) По черзі переставляються стовпці

4) Прочитують початкове повідомлення з лівої таблиці відрядковий:

ПРИЕЗЖАЮ_ШЕСТОГО

Число варіантів подвійної перестановки швидко зростає при збільшенні розміру таблиці :

- для таблиці 3x3 36 варіантів;

- для таблиці 4x4 576 варіантів;

- для таблиці 5x5 14400 варіантів.

Проте подвійна перестановка не відрізняється високою стійкістю і порівняно просто "зламується" при будь-якому розмірі таблиці шифрування.

Контрольні питання:

1. Що таке подвійна перестановка.
2. Алгоритм шифрування та дешифрування.

Лекція №6 (2 години)

Тема: Шифри заміни. Шифр Цезаря. Афінна система підстановок Цезаря. Мультиплікативна інверсія. Розширений алгоритм Евкліда. Система Цезаря з ключовим словом.

Мета: Ознайомитися з шифрами заміни на прикладі шифру Цезаря.

ПЛАН

1. Шифри заміни. Моноалфавітні шифри
2. Шифр «Атбаш»
3. Шифр простої заміни. Система шифрування Цезаря
4. Алгоритм шифрування та дешифрування

При шифруванні заміною (підстановкою) символи шифрованого тексту замінюються символами того ж або іншого алфавіту із заздалегідь встановленим правилом заміни.

Шифри підстановки можуть бути розбиті на дві категорії: моноалфавітні або багатоалфавітні шифри.

МОНОАЛФАВІТНІ (ОДНОАЛФАВІТНІ) ШИФРИ

У моноалфавітній підстановці буква (чи символ) в початковому тексті завжди змінюється на одну і ту ж саму букву (чи символ) в зашифрованому тексті незалежно від його позиції в тексті.

Букви в початковому тексті і зашифрованому тексті знаходяться у відношенні "один до одного".

Приклад 1

Початковий текст: H E L L O

Зашифрований текст: K H O O R

Приклад 2

Початковий текст: H E L L O

Зашифрований текст: A B N Z F

Шифр "Атбаш"

Правило шифрування полягало в заміні 1-ої букви алфавіту буквою з номером $\mathbf{n-i+1}$, де n - число букв алфавіту.

Це слово складене з букв Алеф, Тае, Бет і Шин, тобто першою і останньою, другою і передостанньою буквою давньосемітського алфавіту.

Приклад

У алфавіті A, B, C, кома, точка, плюс.

Таблиця заміни :

Зашифруємо текст: A+ B

Криптограма: +A.

Якщо шифр атбаш використовуваний для алфавіту з непарним числом символів, символ, який посередині алфавіту, не замінюється.

Шифр атбаш є шифром без ключа, тобто веде заміну символів завжди однаково при заданому алфавіті.

Це означає, що вимагається виключити знання супротивником, що використовується саме цей шифр, інакше навіть вручну зашифрований текст швидко прочитується.

Шифр атбаш є шифром простої заміни.

ШИФРИ ПРОСТОЇ ЗАМІНИ

У шифрі простої заміни кожен символ початкового тексту замінюється на один і той же символ з того ж алфавіту незалежно від його позиції в тексті.

Шифр простої заміни є моноалфавітним шифром.

СИСТЕМА ШИФРУВАННЯ ЦЕЗАРЯ

Шифр Цезаря є часткою випадком шифру простої заміни.

Цезарь замінював букви відповідно до підстановки, нижній рядок якої є алфавітом відкритого тексту, зрушеним циклічно на три букви вліво.

A	B	C	D	E	F	G	H	I	J
X	Y	Z	A	B	C	D	E	F	G

K	L	M	N	O	P	Q	R	S	T
H	I	J	K	L	M	N	O	P	Q

U	V	W	X	Y	Z
R	S	T	U	V	W

Таким чином при шифруванні початкового тексту кожна буква замінювалася на іншу букву того ж алфавіту шляхом зміщення за абеткою від початкової букви на три.

Після закінчення алфавіту здійснювався циклічний перехід до його початку. Тобто Цезарь використав шифр заміни з ключем зміщення K=3.

У системі шифрування Цезаря кожен символ початкового тексту замінюється символом того ж алфавіту, віддаленим від початкового на певне число позицій в алфавіті залежно від значення ключа.

Алгоритм шифрування

1. Задамо алфавіт з 26 символів (букви англійського алфавіту, m=26)

2. №	0	1	2	3	4	5	6	7	8	9
символ	A	B	C	D	E	F	G	H	I	J

№	10	11	12	13	14	15	16	17	18	19
символ	K	L	M	N	O	P	Q	R	S	T

№	20	21	22	23	24	25
символ	U	V	W	X	Y	Z

і значення ключа k . $\mathbf{k = \overline{1,m-1}}$

2. Для кожного символу відкритого тексту $\mathbf{PT = \{p_i\}}$, $\mathbf{i = \overline{1,n}}$:

а) визначаємо порядковий номер символу відкритого повідомлення в алфавіті $\mathbf{j(p_i)}$;

б) визначаємо порядковий номер символу шифртекста по рівнянню шифрування $\mathbf{j(c_i) = j(p_i) + k}$;

в) по відомому порядковому номеру $\mathbf{j(c_i)}$ визначаємо символ криптограми $\mathbf{CT = \{c_i\},\ i = \overline{1,n}}$.

Примітка:

Досягши кінця алфавіту виконувався циклічний перехід до його початку. Якщо номер символу шифртекста перевищує довжину алфавіту (m), то отримане значення зменшують на довжину алфавіту.

Приклад 1

Зашифруємо послання Цезаря

VENI VIDI VICI

Ключ k=3

V	**E**	**N**	**I**	**V**	**I**	**D**	**I**	**V**	**I**	**C**	**I**
21	4	13	8	21	8	3	9	21	8	2	8

21+3 =24	7	16	11	24	11	6	11	24	11	5	11
Y	**H**	**Q**	**L**	**Y**	**L**	**G**	**L**	**Y**	**L**	**F**	**L**

Криптограма: **Y H Q L Y L G L Y L F L**

Алгоритм дешифрования

1. Задамо алфавіт з 26 символів (букви англійського алфавіту, **m**=26)

№	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
символ	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**

№	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
символ	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**

№	**20**	**21**	**22**	**23**	**24**	**25**
символ	**U**	**V**	**W**	**X**	**Y**	**Z**

і значення ключа k . $\mathbf{k = \overline{1, m-1}}$

2. Для кожного символу шифртекста $\mathbf{CT = \{c_i\},\ i = \overline{1,n}}$:

а) визначимо порядковий номер символу шифртекста в алфавіті $\mathbf{j(c_i)}$;

б) визначимо порядковий номер символу відкритого повідомлення по рівнянню дешифрування $\mathbf{j(p_i) = j(c_i) - k}$;

в) по відомому порядковому номеру $\mathbf{j(p_i)}$ визначаємо символ відкритого повідомлення $\mathbf{PT = \{p_i\}}$.

Примітка:

Якщо порядковий номер символу відкритого повідомлення менше нуля, то отримане значення збільшують на довжину алфавіту (m).

Приклад 2

Розшифруємо криптограму - I X E V Z U M X G V N E

Ключ дорівнює 6

I	**X**	**E**	**V**	**Z**	**U**	**M**	**X**	**G**	**V**	**N**	**E**
8	23	4	21	25	20	12	23	6	21	13	4
8-6 =2	23-6 =17	4-6 =-2 26-2=24	21-6 =15	25-6 =19	20-6 =14	12-6 =6	23-6 =17	6-6 =0	21-6 =15	13-6 =7	3-6 =-3 26-3 =23
C	**R**	**Y**	**P**	**T**	**O**	**G**	**R**	**A**	**P**	**H**	**Y**

Початкове повідомлення: **C R Y P T O G R A P H Y**

Шифр Цезаря називають адитивним шифром. Термін адитивний шифр показує його математичний сенс. Кожному символу зіставлено ціле число $\mathbf{Z_{26}}$. Засекречений ключ - також ціле число в $\mathbf{Z_{26}}$. Алгоритм кодування додає ключ до символу початкового тексту; алгоритм дешифрування віднімає ключ з символу зашифрованого тексту. Усі операції проводяться в $\mathbf{Z_{26}}$.

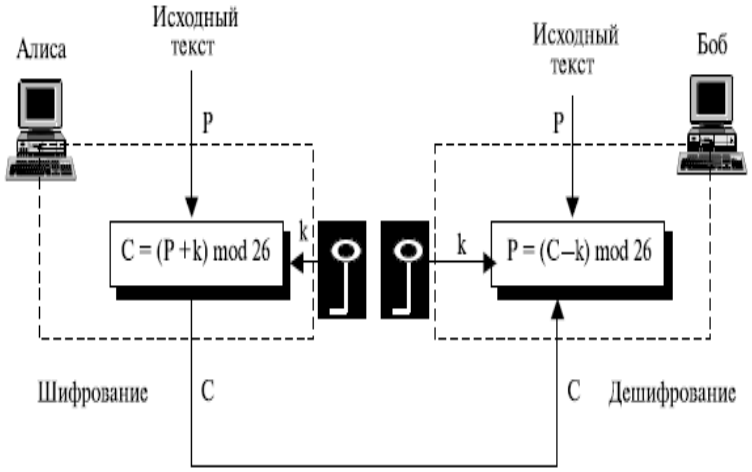
Використовуючи поняття модулярної арифметики можна алгоритм шифрування для шифру Цезаря описати наступним рівнянням криптоперетворення $\mathbf{E_K}$: $\mathbf{j \to (j + k) (mod\ m)}$, $\mathbf{1 \le k < m - 1}$

де K - значення ключа;

m - кількість букв у використовуваному алфавіті;

j - числовий код букви відкритого тексту;

j +k - числовий код відповідної букви шифр тексту "по модулю m".



Для нашого першого прикладу рівняння криптоперетворення

$$E_з: \ j \to (j+3) (\mathbf{mod}\ 26)$$

Вживаний алгоритм шифрування до початкового тексту, буква за буквою:

Початковий текст: **V** $\Rightarrow$ **21**

Шифрування $(21+3) \mathbf{mod}\ 26 \equiv 24 (\mathbf{mod}\ 26)$

Шифртекст: **24** $\Rightarrow$**Y**

Початковий текст: **E** $\Rightarrow$ **4**

Шифрування $(4+3) \mathbf{mod}\ 26 \equiv 7 (\mathbf{mod}\ 26)$

Шифртекст: **7** $\Rightarrow$**H**

Початковий текст: **N** $\Rightarrow$ **13**

Шифрування $(13+3) \mathbf{mod}\ 26 \equiv 16 (\mathbf{mod}\ 26)$

Шифртекст: **16** $\Rightarrow$**Q**

Початковий текст: **I**$\Rightarrow$ **8**

Шифрування $(8+3) \mathbf{mod}\ 26 \equiv 11 (\mathbf{mod}\ 26)$

Шифртекст: **11** $\Rightarrow$**L**

і так далі.

Аналогічно алгоритм дешифрування

$$D_K: \ j \to (j-k) (\mathbf{mod}\ m), \ 1 \le k < m-1$$

де **k** - значення ключа;

m - кількість букв у використовуваному алфавіті;

j - числовий код букви шифртекста;

j - k - числовий код відповідної букви відкритого тексту "по модулю m".

Для другого прикладу рівняння криптоперетворення

$$D_6 : j \to (j-6)(\mathrm{mod}\ 26)$$

Вживаний алгоритм дешифрування до криптограми, буква за буквою:

Криптограма: **I** $\Rightarrow$ **8**

Дешифрування $(8-6)\mathrm{mod}\ 26 \equiv 2(\mathrm{mod}\ 26)$

Відкритий текст: **2** $\Rightarrow$**C**

Криптограма: **X** $\Rightarrow$ **23**

Дешифрування $(23-6)\mathrm{mod}\ 26 \equiv 17(\mathrm{mod}\ 26)$

Відкритий текст: **17**$\Rightarrow$**R**

Криптограма: **E** $\Rightarrow$ **4**

Дешифрування $(4-6)\mathrm{mod}\ 26 \equiv -2(\mathrm{mod}\ 26) \equiv 24(\mathrm{mod}\ 26)$

Відкритий текст: **24** $\Rightarrow$**Y** і так далі.

Гідністю системи шифрування Цезаря є простота шифрування і дешифрування.

До недоліків системи Цезаря слід віднести наступні:

- підстановки, що виконуються відповідно до системи Цезаря, не маскують частот появи різних букв початкового відкритого тексту;
- зберігається алфавітний порядок в послідовності замінюючих букв;
- при зміні значення k змінюються тільки початкові позиції такої послідовності;
- число можливих ключів k мало;
- шифр Цезаря легко розкривається на основі аналізу частот появи букв в шифртексте.

Афінна система підстановок Цезаря

У системі шифрування Цезаря використовувалися тільки адитивні властивості великої кількості цілих Z_m. Проте символи великої кількості множини Z_m можна

також множити по модулю m. Аффінний шифр є комбінацією аддитивних і мультиплікативних шифрів.

Визначимо перетворення в такій системі:

$$\mathbf{E_{k_1,k_2}(j)}: \mathbf{Z_m} \rightarrow \mathbf{Z_m}$$

Де m - кількість букв у використовуваному алфавіті;

j - числовий код букви відкритого тексту;

$\mathbf{k_1}$, $\mathbf{k_2}$ - пара ключів, цілі числа, $\mathbf{0 \le k_1, k_2 < m}$, **НОД (** $\mathbf{k_1}$ **, m) = 1**;

$\mathbf{k_1 \cdot j + k_2}$ - числовий код відповідної букви шифртекста "по модулю m".

У цьому перетворенні буква, що відповідає числу j, замінюється на букву, що відповідає числовому значенню $\mathbf{(k_1 \cdot j + k_2)}$ по модулю m.

Перший ключ $\mathbf{k_1}$ використовується мультиплікативним шифром, другий ключ $\mathbf{k_2}$ - аддитивним шифром. Тобто афінний шифр це два шифри вживаних один за іншим.

Слід зауважити, що перетворення $\mathbf{E_{k_1,k_2}(j)}$ є взаємно однозначним відображенням на множині $\mathbf{Z_m}$ тільки у тому випадку, якщо **НОД(** $\mathbf{k_1}$ **,m)=1**.

Визначимо криптоперетворення при дешифруванні:

$$\mathbf{D_{k_1,k_2}(j) \rightarrow \left((j - k_2) \cdot k_1^{-1}\right) (mod\, m)}$$

де

m - кількість букв у використовуваному алфавіті;

j - числовий код букви шифртекста;

$\mathbf{k_1}$, $\mathbf{k_2}$, - пара ключів, цілі числа, $\mathbf{0 \le k_1, k_2 < m}$, **НОД (** $\mathbf{k_1}$ **, m) = 1;**

$\mathbf{(j - k_2) \cdot k_1^{-1}}$ - числовий код відповідної букви відкритого тексту "по модулю m";

$\mathbf{k_1^{-1}}$ - мультиплікативна інверсія $\mathbf{k_1}$;

$-\mathbf{k}_2$ - аддитивна інверсія $\mathbf{k}_2$.

У $\mathbf{Z}_\mathbf{m}$ аддитивна інверсія числу **a** може бути вичислена як $\mathbf{x = m - a}$. Наприклад, аддитивна інверсія 4 в $\mathbf{Z}_{\mathbf{10}}$ рівна 10 - 4 = 6.

У модульній арифметиці кожне ціле число має аддитивну інверсію. Сума цілого числа і його аддитивної інверсії порівнянна з 0 по модулю m, тобто $\mathbf{a + x \equiv 0 (mod\, m)}$

У модульній арифметиці ціле число може або не може мати мультиплікативної інверсії. Ціле число і його мультиплікативна інверсія порівнянні з 1 по модулю m, тобто $\mathbf{a \cdot x \equiv 1 (mod\, m)}$.

Ціле число **a** в $\mathbf{Z}_\mathbf{m}$ має мультиплікативну інверсію тоді і тільки тоді, якщо **НОД (m, a) = 1**

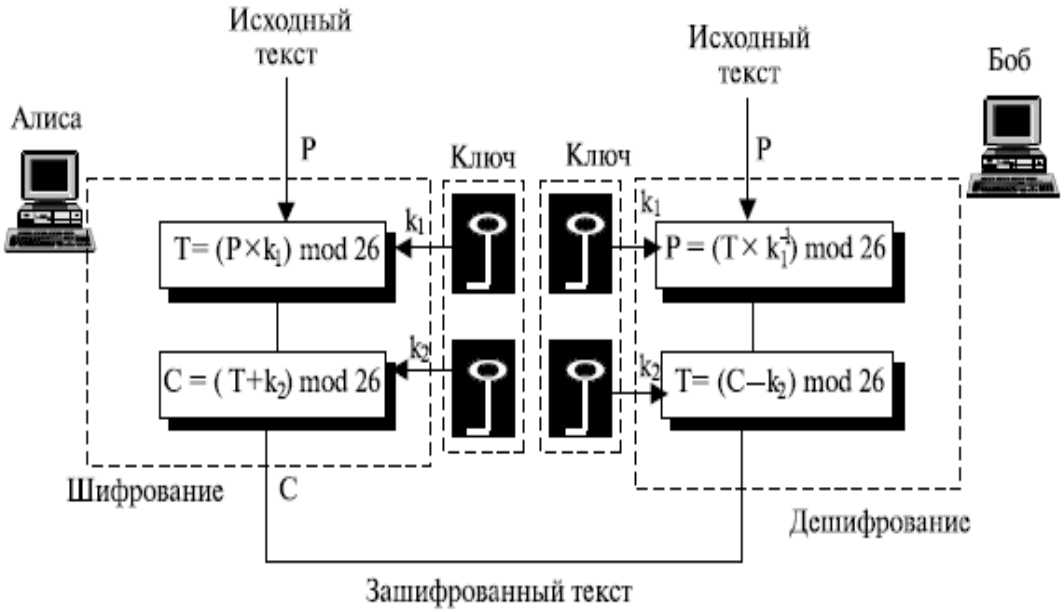


Рисунок 5 - Афінний шифр

Наприклад, нехай m = 26, = 3, = 5. Тоді, очевидно, НОД(3,26) =1, і ми отримуємо наступну відповідність між числовими кодами букв:

j	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
$\mathbf{3 \cdot j + 5}$	**5**	**8**	**11**	**14**	**17**	**20**	**23**	**0**	**3**	**6**

j	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
$\mathbf{3 \cdot j + 5}$	**9**	**12**	**15**	**18**	**21**	**24**	**1**	**4**	**7**	**10**

j	20	21	22	23	24	25
$3 \cdot j + 5$	13	16	19	22	25	2

Перетворюючи числа у букви англійської мови, отримуємо наступну відповідність для букв відкритого тексту і шифртексту:

P_i	A	B	C	D	E	F	G	H	I	J
C_i	F	I	L	O	R	U	X	A	D	G

P_i	K	L	M	N	O	P	Q	R	S	T
C_i	J	M	P	S	V	Y	B	E	H	K

P_i	U	V	W	X	Y	Z
C_i	N	Q	T	W	Z	C

Приклад 4

Використовуємо алфавіт з 26 символів (26 англійських букв). Отже m = 26.

Нехай $k_1 = 3$, $k_2 = 5$.

Перевіряємо НОД (m, k_1)=НОД (26,3) = 1.

Початкове повідомлення: **H O P E**

Рівняння шифрування $E_{3,5}(j): \quad j \to (3 \cdot j + 5) (\mathrm{mod}\ 26)$

m_i	код символа сообщения	Шифрование	код символа шифртекста	c_i
H	7	$(3 \cdot 7 + 5) (\mathrm{mod}\ 26) \equiv 0 (\mathrm{mod}\ 26)$	0	**A**
O	14	$(3 \cdot 14 + 5) (\mathrm{mod}\ 26) \equiv 47 (\mathrm{mod}\ 26)$	21	**V**
P	15	$(3 \cdot 15 + 5) (\mathrm{mod}\ 26) \equiv 50 (\mathrm{mod}\ 26)$	24	**Y**
E	4	$(3 \cdot 4 + 5) (\mathrm{mod}\ 26) \equiv 17 (\mathrm{mod}\ 26)$	17	**R**

Шифртекст: **A V Y R**

Використовуємо алфавіт з 26 символів (26 англійських букв). Отже m = 26.

Нехай $\mathbf{k_1 = 3}$, $\mathbf{k_2 = 5}$.

Перевіряємо НОД (m, $\mathbf{k_1}$)=НОД (26,3) = 1,

Шифртекст: **A V Y R**

Щоб знайти символи початкового тексту додамо аддитивну інверсію від $\mathbf{-5 \equiv 21 (mod\ 26)}$ (т.е. $\mathbf{26 - 5 = 21}$) до отриманого зашифрованого тексту(аддитивна інверсія існує оскільки $\mathbf{5 + 21 \equiv 0 (mod\ 26)}$).

Потім помножимо на мультиплікативну інверсію від $\mathbf{3^{-1} \equiv 9 (mod\ 26)}$.

№	m	$\mathbf{k_1}$	$\mathbf{m \bmod k_1}$	$\mathbf{int\left(\frac{m}{k_1}\right)}$	d	u	v
1	26	3	2	8	1	–1	9
2	3	2	1	1	1	1	–1
3	2	1	0	2	1	0	1
4	1	0	–	–	1	1	0

Рівняння дешифрування:

$$\mathbf{D_{3,5}(j) \rightarrow ((j-5) \cdot 3^{-1}) (mod\ 26)}$$

$\mathbf{c_i}$	код символа шифртекста	Шифрование	код символа сообщения	$\mathbf{m_i}$
A	0	$\mathbf{((0+21) \cdot 9) (mod\ 26) \equiv 189 (mod\ 26)}$	7	**H**
V	21	$\mathbf{((21+21) \cdot 9) (mod\ 26) \equiv (189 (mod\ 26) + 189 (mod\ 26)) (mod\ 26) \equiv 14 (mod\ 26)}$	14	**O**
Y	24	$\mathbf{((24+21) \cdot 9) (mod\ 26) \equiv 15 (mod\ 26)}$	15	**P**
R	17	$\mathbf{((17+21) \cdot 9) (mod\ 26) \equiv 4 (mod\ 26)}$	4	**E**

Гідністю афінної системи є зручне управління ключами - ключі шифрування і розшифрування представляються в компактній формі у вигляді пари чисел(($\mathbf{k}_1$, $\mathbf{k}_2$).

Недоліки афінної системи аналогічні недолікам системи шифрування Цезаря.
Потоковий, симетричний.

Мультиплікативна інверсія. Розширений алгоритм Евкліда.

Інверсії

Коли ми працюємо в модульної арифметики, нам часто потрібно знайти операцію, яка дозволяє обчислити величину, зворотну заданому числу. Ми зазвичай шукаємо аддитивную інверсію (оператор, зворотний додаванню) або мультипликативную інверсію (оператор, зворотний множенню).

Аддитивна інверсія
В Zn два числа a і b адитивно інверсний один одному, якщо b = n - a. наприклад,
a + b \ equiv 0 (mod \ n)
В Zn аддитивная інверсія числа a може бути обчислена як b = n - a. Наприклад, аддитивная інверсія 4 в Z10 дорівнює 10 - 4 = 6.

В модульній арифметиці кожне ціле число має адитивну інверсію. Сума цілого числа і його адитивної інверсії порівняння з 0 по модулю n.
Зверніть увагу, що в модульній арифметиці кожне число має аддитивну інверсію, і ця інверсія унікальна; кожне число має одну і тільки одну аддитивную інверсію. Однак інверсія числа може бути безпосередньо тим же самим числом.

Наприклад:

Знайдіть всі взаємно зворотні пари по додаванню в Z10.

Рішення

Дано шість пар адитивних інверсій - (0, 0), (1, 9), (2, 8), (3, 7), (4, 6) і (5, 5). У цьому списку 0 - інверсія самому собі; так само і 5. Зверніть увагу: адитивні інверсії назад один одному; якщо 4 - адитивна інверсія 6, тоді 6 - також аддитивная інверсія числа 4.

мультиплікативна інверсія
В Zn два числа a і b мультиплікативно інверсний один одному, якщо a \ times b \ equiv 1 (mod \ n)
Наприклад, якщо модуль дорівнює 10, то мультипликативная інверсія 3 є 7. Іншими словами, ми маємо (3 \ times 7) \ bmod 10 \ equiv 1.

В модульній арифметиці ціле число може або не може мати мультиплікативну інверсію. Ціле число і його мультипликативная інверсія порівнянні з 1 по модулю n.
Може бути доведено, що a має мультипликативную інверсію в Zn, якщо тільки НОД (n, a) = 1. У цьому випадку говорять, що a і n взаємно прості.

Наприклад:

Знайти мультиплікативную інверсію 8 в Z10.

Рішення

Мультиплікативна інверсія не існує, тому що HOD \ left ({{\ text {1}} 0, {\ text {8}}} \ right) = {\ text {2}} \ ne {\ text {1}}. Іншими словами, ми не можемо знайти число між 0 і 9, таке, що при множенні на 8 результат можна порівняти з 1 по mod 10.

Наприклад

Знайти всі мультиплікативні інверсії в Z10.

Рішення

Є тільки три пари, що задовольняють умовам існування мультипликативной інверсії: (1, 1), (3, 7) і (9, 9). Числа 0, 2, 4, 5, 6 і 8 не мають мультипликативной інверсії.

Ми можемо перевірити, що (1 x 1) mod 10 = 1 (3 x 7) mod 10 = 1 (9 x 9) mod 10 = 1

Наприклад:

Знайти всі мультиплікативні зворотні пари в Z11.

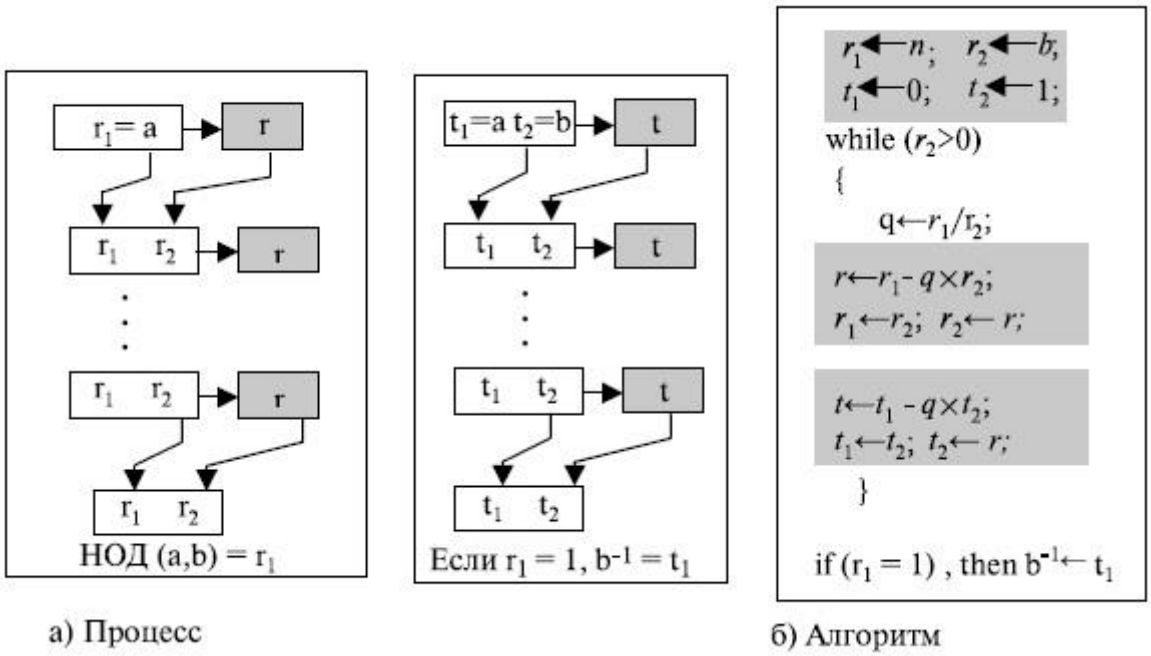
Рішення

Ми маємо такі пари: (1, 1), (2, 6), (3, 4), (5, 9), (7, 8) і (10, 10). При переході від Z10 до Z11 число пар збільшується. При Z11 НСД (11, a) = 1 (взаємно прості) для всіх значень a, крім 0. Це означає, що всі цілі числа від 1 до 10 мають мультиплікативний інверсії.

Ціле число a в Zn має мультипликативную інверсію тоді і тільки тоді, якщо НСД (n, a) = 1 (mod n)
Розширений алгоритм Евкліда, який ми обговорювали раніше в цій лекції, може знайти мультипликативную інверсію b в Zn, коли дані n і b і інверсія існує. Для цього нам треба замінити перший ціле число a на n (модуль). Далі ми можемо стверджувати, що алгоритм може знайти s і t, такі, що s \ times n + b \ times t = HOD \ left ({n, b} \ right). Однак якщо мультипликативная інверсія b існує, НОД (n, b) повинен бути 1. Так що рівняння буде мати вигляд (S x n) + (b x t) = 1.
Тепер ми застосовуємо операції по модулю до обох сторін рівняння. Іншими словами, ми відображаємо кожну сторону до Zn. Тоді ми будемо мати
(S x n + b x t)mod n = 1 mod n
[(S x n) mod n] + [(b x t) mod n] = 1 mod n
0 + [(b x t) mod n] = 1
(B x t) mod n = 1 ->
Це означає, що t - це мультиплікативна інверсія b в Zn

Зверніть увагу, що [(s \ times n) \ bmod n] на третьому рядку - 0, тому що, якщо ми ділимо (s \ times n) n, приватне - s, а залишок - 0.

Розширений алгоритм Евкліда знаходить мультиплікативні інверсії b в Zn, коли дані n і b і НОД (n, b) = 1. Мультиплікативна інверсія b - це значення t, відображене в Zn.
Рисунок 6 показує, як ми знаходимо мультипликативную інверсію числа, використовуючи розширений алгоритм Евкліда.



а) Процесс б) Алгоритм

Рисунок 6 - Розширений алгоритм Евкліда

Наприклад:

Знайти мультипликативную інверсію 11 в Z26.

Рішення

Ми використовуємо таблицю, аналогічну однією з тих, які ми вже застосовували перш за даних r1 = 26 і r2 = 11. Нас цікавить тільки значення t.

q	r_1	r_2	r	t_1	t_2	t
2	26	11	4	0	1	-2
2	11	4	3	1	-2	5
1	4	3	1	-2	5	-7
3	3	1	0	5	-7	26
	1	0		-7	26	

НСД (26, 11) = 1, що означає, що мультиплікативна інверсія 11 існує. Розширений алгоритм Евкліда дає t1 = (-7).

Мультиплікативна інверсія дорівнює (-7) mod 26 = 19. Іншими словами, 11 і 19 - мультиплікативна інверсія в Z26. Ми можемо бачити, що (11 \ times 19) \ bmod 26 = 209 \ bmod 26 = 1.

Наприклад:

Знайти мультипликативную інверсію 23 в Z100.

Рішення

Ми використовуємо таблицю, подібну до тієї, яку застосовували до цього при r1 = 100 і r2 = 23. Нас цікавить тільки значення t.

q	r_1	r_2	r	t_1	t_2	t
4	100	23	8	0	1	-4
2	23	8	7	1	-4	19
1	8	7	1	-4	9	-13
7	7	1	0	9	-13	100
	1	0		-13	100	

НСД (100, 23) = 1, що означає, що інверсія 23 існує. Розширений Евкліда алгоритм дає t1 = -13. Інверсія - (-13) mod 100 = 87. Іншими словами, 13 і 87 - мультиплікативні інверсії в Z100. Ми можемо бачити, що (23 \ times 87) \ bmod 100 = 2001 \ bmod 100 = 1.

Наприклад:

Знайти мультипликативную інверсію 12 в Z26.

Рішення

Ми використовуємо таблицю, подібну до тієї, яку ми застосовували раніше при r1 = 26 і r2 = 12.

q	r_1	r_2	r	t_1	t_2	t
2	26	12	2	0	1	
6	12	2	0	1	-2	
	2	0		-2	13	

$HOD\,(26, 12) = 2 \neq 1$, що означає отсутвствіе для числа 12 мультипликативной інверсії в Z26.

Система Цезаря з ключовим словом.

Система шифрування Цезаря з ключовим словом є моноалфавітною системою

підстановки. Особливістю цієї системи є використання ключового слова для зміщення і зміни порядку символів в алфавіті підстановки.

Алгоритм шифрування

1. Задамо алфавіт з 26 символів(букви англійського алфавіту, m=26)

№	0	1	2	3	4	5	6	7	8	9
символ	A	B	C	D	E	F	G	H	I	J

№	10	11	12	13	14	15	16	17	18	19
символ	K	L	M	N	O	P	Q	R	S	T

№	20	21	22	23	24	25
символ	U	V	W	X	Y	Z

і ключове число **k,** $\mathbf{k = 0,\overline{m-1}}$ і ключове слово або фраза.

2. Ключове слово записується під буквами алфавіту, починаючи з букви, числовий код якої співпадає з вибраним числом **k.**

3. Букви алфавіту підстановки, що залишилися, записуються після ключового слова в алфавітному порядку:

4. У відповідності отриманої підстановки шифруємо відкрите повідомлення.

Приклад

Ключове число k=5, ключове слово DIPLOMAT

	0	1	2	3	4	5	6	7	8	9
початковий алфавіт	A	B	C	D	E	F	G	H	I	J
алфавіт підставляння	V	W	X	Y	Z	D	I	P	L	O

	10	11	12	13	14	15	16	17	18	19
початковий алфавіт	K	L	M	N	O	P	Q	R	S	T
алфавіт підставляння	M	A	T	B	C	E	F	G	H	J

	20	21	22	23	24	25
початковий	U	V	W	X	Y	Z

алфавіт						
алфавіт підставляння	**K**	**N**	**Q**	**R**	**S**	**U**

Відкритий текст: S E N D M O R E M O N E Y

Шифртекст: H Z B Y T C G Z T C B Z S

Примітка:

Якщо в ключовому слові або фразі повторюються букви, то запис ключового слова або фрази проводиться без повторів.

Приклад

якщо у БЛОНДИНКА, то записуватися в таблицю підстановок буде слово БЛОНДИКА

Гідністю системи Цезаря з ключовим словом є те, що кількість можливих ключових слів практично невичерпна.

Недоліком цієї системи є можливість злому шифртекста на основі аналізу частот появи букв.

Контрольні питання:

1. Що таке шифр заміни.
2. Шифр «Атбаш».
3. Алгоритм шифрування та дешифрування у системі Цезаря.
4. Гідності та недоліки шифру Цезаря.
5. Що таке афінна система підстановок Цезаря.
6. Алгоритм шифрування та дешифрування у системі підстановок Цезаря.
7. Гідності та недоліки системи підстановок Цезаря.
8. Аддитивна інверсія.
9. Мультиплікативна інверсія.
10. Розширений алгоритм Евкліда.
11. Особливості шифру Цезаря з ключовим словом.

Лекція №7 (2 години)

Тема: Шифр Полібія. Шифруючі таблиці Трисемуса.
Мета: Ознайомитися з алгоритмом шифрування Полібія.

ПЛАН

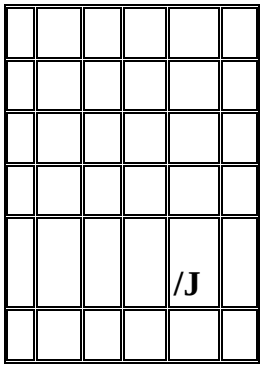
1. Шифр Полібія
2. Алгоритм шифрування
3. Гідності та недоліки шифру

Шифрований символ відкритого повідомлення замінюється двома цифровими символами, що означають координати клітинки квадрата, перше число означає рядок, друге – стовпець (шифрування зводиться до заміни букви парою цифр).

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I,J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

При розшифруванні кожна така пара визначала відповідну букву повідомлення.

Причому заповнювати квадрат можна вибираючи символи у будь-якому порядку.



Тобто ключем такого шифру Полібія є розташування букв в таблиці.

Можуть бути алфавіти, де m "проблемно" для складання таблиці. Припустимо, m= 31 - просте число, його не розкласти на множники; m= 46 розкласти можна, а саме: 46 = 23 * 2, але ось тільки якесь ідіотське відношення розмірів. У таких випадках варто якось розширити алфавіт: додати, скажімо, розділові знаки або арифметичні знаки.

Ідею квадрата Полібія проілюструємо таблицею з російськими буквами. Число букв в російському алфавіті відрізняється від числа букв в грецькому, тому розмір таблиці вибраний іншим(6×6).

i / j	1	2	3	4	5	6

1	А	Б	В	Г	Д	Е Ё
2	Ж	З	И Й	К	Л	М
3	Н	О	П	Р	С	Т
4	У	Ф	Х	Ц	Ч	Ш
5	Щ	Ь	Ъ	Ы	Э	Ю
6	Я	_	.	,	:	;

Зашифруємо повідомлення

ВЕСНА_ПРИШЛА

В	**Е**	**С**	**Н**	**А**	**_**	**П**	**Р**	**И**	**Ш**	**Л**	**А**
13	**16**	**35**	**31**	**11**	**62**	**33**	**34**	**23**	**46**	**25**	**11**

На перший погляд шифр здається дуже нестійким, але для його реальної оцінки слід враховувати два чинники:

- можливість заповнити квадрат Полібія буквами довільно, а не тільки строго за абеткою;

- можливість періодично замінювати квадрати.

Тоді аналіз попередніх повідомлень нічого не дає, оскільки до моменту розкриття шифру він може бути замінений.

Букви можуть вписуватися в таблицю в довільному порядку - заповнення таблиці в цьому випадку і є ключем. Для латинського алфавіту в першу клітину можна вписати одну з 25 букв, в другу - одну з 24, в третю - одну з 23 і так далі. Отримуємо максимальну кількість ключів для шифру на таблиці латинського алфавіту : $N = 25 * 24 * 23 * ... * 2 * 1 = 25!$

Відповідно для дешифрування повідомлення знадобиться не лише знання алфавіту, але і ключа, за допомогою якого складалася таблиця шифрування. Але довільний порядок букв важко запам’ятати, тому користувачеві шифру необхідно постійно мати при собі ключ - квадрат. З'являється небезпека таємного ознайомлення з ключем сторонніх осіб. В якості компромісного рішення був запропонований ключ - пароль.

Пароль виписується без повторів букв в квадрат; у клітини, що залишилися, в алфавітному порядку виписуються букви алфавіту, відсутні в паролі.

Слово-шифр повинно складатися з унікальних символів. Таке слово поміщається спочатку квадрата Полібія. Елементи квадрата, що йдуть за ключем, заповнюються буквами в порядку їх проходження в алфавіті з пропуском тих букв, які вже були використані в ключовому слові. Сам алгоритм криптоперетворень залишається такими ж, як і в шифрі Полібія.

Помітимо, що ефективність шифрування залежить від ключового слова, оскільки саме воно задає якість переміщення символів в квадраті Полібія. Для якіснішого перемішування в ключове слово доцільно включати останні символи базового алфавіту.

Зашифруємо повідомлення, використовуючи ключове слово РЕСПУБЛИКА

Т Е Р П Е Н Ь Е _ В О З Н А Г Р А Ж Д А Е Т С Я

i / j						
1		/Ё				
2		/Й				
3						
4						
5						
6						

4112 1114 1235 5312 6325 3633 3524 2611 3231 2412 4113 62

У шифрі використовується заміна один символ початкового тексту на два символи шифротекста. Це симетричний метод шифрування.

Ключовий простір впливає з довжина слова і максимально можлива кількість ключів.

Максимальна довжина ключа дорівнює довжині алфавіту, максимальна кількість варіантів ключа визначається факторіалом від довжини алфавіту. У нашому випадку 36.

Криптостійкість шифру середня, оскільки простір ключів хоч і великий, але обмежений.

Ефективність шифрування залежить від ключового слова.

Зашифруємо теж повідомлення, використовуючи ключ ОБВАЛ

i / j	1	2	3	4	5	6
1	О	Б	В	А	Л	Г
2	Д	Е/Ё	Ж	З	И/Й	К
3	М	Н	П	Р	С	Т
4	У	Ф	Х	Ц	Ч	Ш
5	Щ	Ь	Ъ	Ы	Э	Ю
6	Я	_	.	,	:	–

Т Е Р П Е Н Ь Е_В О З Н А Г Р А Ж Д А Е Т С Я

3622 3433 2232 5222 6213 1124 3214 1634 1423 2114 2236 3561

Шифруючі таблиці Трисемуса.

Для отримання такого шифру заміни зазвичай використовувалися таблиця для запису букв алфавіту і ключове слово (чи фраза).

У таблицю спочатку вписувалося по рядках ключове слово, причому букви, що повторюються, відкидалися.

Потім ця таблиця доповнювалася буквами алфавіту, що не увійшли до неї, по порядку.

Оскільки ключове слово або фразу легко зберігати в пам'яті, то такий підхід спрощував процеси шифрування і дешифрування.

Приклад

Для російського алфавіту шифруюча таблиця може мати розмір 4*8

Виберемо в якості ключа слово БАНДЕРОЛЬ. Шифруюча таблиця з таким ключем показана на рисунку.

Б	**А**	**Н**	**Д**	**Е**	**Р**	**О**	**Л**
Ь	**В**	**Г**	**Ж**	**З**	**И**	**Й**	**К**
М	**П**	**С**	**Т**	**У**	**Ф**	**Х**	**Ц**
Ч	**Ш**	**Щ**	**Ы**	**Ъ**	**Э**	**Ю**	**Я**

Шифрування:

- знаходять в таблиці букву відкритого тексту і записують в шифртекст букву, розташовану нижче її в тому ж стовпці;

- якщо буква тексту опиняється в нижньому рядку таблиці, тоді для шифртекста беруть саму верхню букву з того ж стовпця.

Наприклад, при шифруванні за допомогою цієї таблиці повідомлення

В Ы Л Е Т А Е М П Я Т О Г О

отримуємо шифртекст

П Д К З Ы В З Ч Ш Л Ы Й С Й

Такі табличні шифри називаються монограмними, оскільки шифрування виконується по одній букві.

Трисемус першим помітив, що шифруючі таблиці дозволяють шифрувати відразу по дві букви. Такі шифри називаються біграмними.

Контрольні питання:

1. Особливості шифру Полібія.
2. Алгоритм шифрування.
3. Гідності і недоліки шифру.
4. Особливості шифру Трисемуса.
5. Алгоритм шифрування.
6. Гідності і недоліки шифру.

Лекція №8 (2 години)

Тема: Багатоалфавітні шифри. Шифр Хасегаві.

Мета: Ознайомитися з багатоалфавітними шифрами.

ПЛАН

1. Багатоалфавітні шифри
2. Автоключовий шифр
3. Криптоаналіз

У багатоалфавітній підстановці кожна поява символу може мати різну заміну.

Стосунки між символом в початковому тексті і символом в зашифрованому тексті - "один до багатьом".

Багатоалфавітні шифри мають перевагу: вони приховують частоту появи символу основної мови.

У багатоалфавітній підстановці для шифрування кожного символу початкового повідомлення застосовують свій шифр простої заміни.

Щоб створити багатоалфавітний шифр, необхідно зробити кожен символ зашифрованого тексту залежним від відповідного символу початкового тексту і позиції символу початкового тексту в повідомленні.

Це має на увазі, що наш ключ k має бути потоком підключів (k1, k2, k3.), в яких кожен підключ так чи інакше залежить від позиції символу початкового тексту, який використовується для вибору підключа шифрування.

Іншими словами, ми повинні мати ключовий потік k = (k1, k2, k3...), в якому ki застосовується, щоб зашифрувати i- тый символ в початковому тексті і створити i- тый символ в зашифрованому тексті.

Автоключовий шифр

Щоб зрозуміти залежність ключа від позиції, обговоримо простий багатоалфавітний шифр, названий "автоключовим". У цьому шифрі ключ - потік підключіі, в якому кожен підключ використовується, щоб зашифрувати відповідний символ в початковому тексті. Перший підключ - визначене заздалегідь значення, таємно погоджене Алісою і Бобом. Другий підключ - значення першого символу

початкового тексту(між 0 і 25). Третій - і- тое значення другого початкового тексту. І так далі.

$$PT = p_1 p_2 p_3 \dots$$
$$CT = c_1 c_2 c_3 \dots$$
$$K = (k_1, p_1, p_2, p_3, \dots)$$

Шифрування Ci = (Pi ki) mod 26

Дешифрування Pi = (Ci - ki) mod 26

Назва шифру, автоключовий, має на увазі, що підключі створюються автоматично залежно від символів шифру початкового тексту в процесі шифрування.

Наприклад:

Припустимо, що Аліса і Боб погодилися використати автоключовий шифр з початковим ключовим значенням k1= 12. Тепер Аліса хоче передати Бобу повідомлення "Attack is today"("Атака - сьогодні"). Шифрування проводиться символ за символом. Кожен символ в початковому тексті спочатку замінюється його значенням цілого числа, як показано на рисунку, перший підключ додається, щоб створити перший символ зашифрованого тексту. Інша частина ключа створюється у міру читання символів початкового тексту. Зверніть увагу, що шифр є багатоалфавітним, тому що ці три появи "a" в результаті тексти зашифровані по-різному. Три виникнення "t" також зашифровані по-різному.

Исходный текст	a	t	t	a	c	k	i	s	t	o	d	a	y
Значения P	00	19	19	00	02	10	08	18	19	14	03	00	24
Поток ключей	12	00	19	19	00	02	10	08	18	19	14	03	00
Значения C	12	19	12	19	02	12	18	00	11	07	17	03	24
Зашифрованный текст	M	T	M	T	C	M	S	A	L	H	R	D	Y

Криптоаналіз

Автоключовий шифр дійсно приховує статистику частоти окремого символу.

Проте він так само уразливий при атаці за допомогою грубої сили, як і аддитивний шифр.

Перший підключ може бути тільки одним з 25 значень(1 - 25). Ми потребуємо багатоалфавітних шифрів, які не лише приховують характеристики мови, але і мають велику безліч ключів.

Шифр Хасегаві.

Алфавітне складання Хасегаві є модифікацією шифру Цезаря. Особливістю цієї системи є наявність ключового слова. Ключове слово може складатися з будь-яких символів і бути будь-якої довжини.

У випадку якщо ключове слово коротше за повідомлення його повторюють циклічно.

1. Задамо алфавіт з 26 символів(букви англійського алфавіту, m=26)

№	0	1	2	3	4	5	6	7	8	9
символ	A	B	C	D	E	F	G	H	I	J

№	10	11	12	13	14	15	16	17	18	19
символ	K	L	M	N	O	P	Q	R	S	T

№	20	21	22	23	24	25
символ	U	V	W	X	Y	Z

і ключове слово $\mathbf{K = \{k_i\}}$.

2. Для кожного символу відкритого тексту $\mathbf{PT = \{p_i\}}$, $\mathbf{i = \overline{1,n}}$:

а) визначуваний порядковий номер символу відкритого повідомлення в алфавіті $\mathbf{j(p_i)}$;

б) Визначуваний порядковий номер символу ключа $\mathbf{j(k_i)}$;

в) Визначуваний порядковий номер символу шифртекста по рівнянню шифрування $\mathbf{j(c_i) = j(p_i) + j(k_i)}$;

г) По відомому порядковому номеру $\mathbf{j(c_i)}$ визначаємо символу шифротекста.

Примітка:

Досягши кінця алфавіту виконувався циклічний перехід до його початку. Якщо номер символу шифротекста перевищує довжину алфавіту(m), то отримане значення зменшують на довжину алфавіту.

Приклад

Зашифруємо повідомлення

I LOVE YOU

ключове слово

LOVE

№ п/п символа текста	8	11	14	21	4	24	14	20
Текст	I	L	O	V	E	Y	O	U
№ п/п символа ключа	11	14	21	4	11	14	21	4

ключ	L	O	V	E	L	O	V	E
сумма номеров	19	25	35	25	15	38	35	24
№ символа шифртекста	19	25	9	25	15	12	9	24
Шифртекст	T	Z	J	Z	P	M	J	Y

Шифрування ci = (pi +ki) mod m

Дешифрування pi = (ci - ki) mod m

Алгоритм дешифрування

самостійно

В шифрі використовується складна заміна: один і той же символ повідомлення замінюється різними символами шифртекста.

Криптостійкість середня. Для підвищення ефективності необхідно ключ вибирати довше. Найкращий результат виходить, якщо ключ дорівнює довжині повідомлення.

Контрольні питання:

1. Що таке багатоалфавітні шифри.
2. Автоключовий шифр.
3. Гідності і недоліки шифру.
4. Шифр Хасегаві.
5. Алгоритм шифрування і дешифрування.

Лекція №9 (2 години)

Тема: Асиметричні криптосистеми. Концепція криптосистеми з відкритим ключем. Однонапрямлені функції.

Мета: Ознайомитися з поняттям асиметричних криптосистем.

ПЛАН

1. Асиметричні криптосистеми
2. Ознаки асиметричних криптосистем

Ефективними системами криптографічного захисту даних є асиметричні криптосистеми, що називаються також криптосистемами (КС) з відкритим ключем.

Криптографія з симетричними ключами базується на спільному використанні ключів; асиметрично-ключова криптографія базується на персональному ключі.

Є деякі інші аспекти безпеки окрім шифрування, які властиві асиметрично-ключовій криптографії. Вони включають встановлення достовірності і цифрові підписи.

Криптографія з симетричними ключами базується (в основному) на підстановці і перестановці символів (символів або біт), а асиметрично-ключова криптографія - на застосуванні математичних функцій до чисел.

Повідомлення повинне перед шифруванням кодуватися як ціле число (чи безліч цілих чисел). Після дешифрування воно має бути розшифроване як ціле число (чи безліч цілих чисел).

У асиметричних системах для шифрування даних використовується один ключ, а для дешифрування - інший ключ (звідси і назва - асиметричні).

Для шифрування даних використовується перший ключ, який є відкритим і може бути опублікований для використання усіма користувачами системи.

Дешифрування даних за допомогою відкритого ключа неможливе.

Для дешифрування даних одержувач зашифрованої інформації використовує другий ключ, який є секретним.

Ключ дешифрування не може бути визначений з ключа шифрування.

У цій криптосистемі застосовують два різні ключі: $\mathbf{K_{B1}}$ - відкритий ключ посилача **A**; $\mathbf{K_{B2}}$ - секретний ключ одержувача **B**. Генератор ключів доцільно розташовувати на стороні одержувача **B** (щоб не пересилати секретний ключ $\mathbf{K_2}$ по незахищеному каналу). Значення ключів $\mathbf{K_{B1}}$ і $\mathbf{K_{B2}}$ залежать від початкового стану генератора ключів.

Розкриття секретного ключа $\mathbf{K_{B2}}$ по відомому відкритому ключу $\mathbf{K_{B1}}$ має бути обчислювальне нерозв'язним завданням.

Характерні особливості асиметричних криптосистем :

1. Відкритий ключ $\mathbf{K_{B1}}$ і криптограма **CT** можуть бути відправлені по незахищених каналах, тобто супротивникові відомі $\mathbf{K_{B1}}$ і **CT**.

2. Алгоритм шифрування

$$\mathbf{E_B: \ PT \ --> \ CT}$$

є відкритим.

Захист інформації в асиметричній криптосистемі грунтований на секретності ключа $\mathbf{K_{B2}}$.

У. Діффі і М. Хеллман сформулювали вимоги, виконання яких забезпечує безопасность асиметричної криптосистеми :

1. Обчислення пари ключів($\mathbf{K_{B1}}$, К $\mathbf{K_{B2}}$) одержувачем **B** на основі початкової умови має бути простим.

2. Посилач **A**, знаючи відкритий ключ $\mathbf{K_{B1}}$ і повідомлення **PT**, може легко вичислити криптограму

$$CT = E_{K_{B1}}(PT) = E_B(PT) \quad (1)$$

3. Одержувач **B**, використовуючи секретний ключ K_{B2} і криптограму **CT**, може легко відновити початкове повідомлення

$$PT = D_{K_{B2}}(CT) = D_B(CT) = D_B(E_B(PT)) \quad (2)$$

4. Супротивник, знаючи відкритий ключ K_{B1}, при спробі вичислити секретний ключ K_{B2} натрапляє на непереборну обчислювальну проблему.

5. Супротивник, знаючи пару (K_{B1}, **CT**), при спробі вичислити початкове повідомлення **pt** натрапляє на непереборну обчислювальну проблему.

Однонапрямлені функції.

Концепція асиметричних криптографічних систем з відкритим ключем грунтована на застосуванні однонапрямлених функцій .

Неформальне визначення однонапрямленої функції (OWF - One Way Function):

Нехай X і Y - деякі довільні множини. Функція

$$f: X \rightarrow Y$$

є однонапрямленою, якщо

1. для усіх $x \in X$ можна легко вичислити функцію

$$y = f(x), \text{ де } y \in Y;$$

2. для більшості $y \in Y$ досить складно отримати значення $x \in X$, таке, що $f(x) = y$ (при цьому вважають, що існує, принаймні, одно таке значення x).

Основним критерієм віднесення функції f до класу однонапрямлених функцій є відсутність ефективних алгоритмів зворотного перетворення $Y \rightarrow X$.

Перший приклад однонапрямленої функції - цілочисельне множення. Пряме завдання - обчислення твору двох дуже великих цілих чисел P і Q, т. е. знаходження значення

N = P*Q (3)

є відносно нескладним завданням для ПК.

Зворотне завдання - розкладання на множники великого цілого числа, тобто знаходження дільників P і Q великого цілого числа N=P*Q, є практично нерозв'язним завданням при досить великих значеннях N.

За сучасними оцінками теорії чисел при цілому $N \approx 2^{664}$ і $P \approx Q$ ідля розкладання числа N знадобиться близько 10^{23} операцій, тобто завдання практично нерозв'язне на сучасних ПК.

Другий приклад однонапрямленої функції - це модульна експонента з фіксованими основою і модулем.

Нехай A і N - цілі числа, такі, що $1 \le A < N$. Визначимо безліч Z_N

$Z_N = \{0, 1, 2, ..., N-1\}$.

Тоді модульна експонента з основою **A** по модулю **N** є функцією

$f_{A,N} : Z_N \rightarrow Z_N$,

$$f_{A,N}(x) = A^x \pmod N$$

де x - ціле число $1 \le x \le N-1$.

Існують ефективні алгоритми, що дозволяють досить швидко вичислити значення функції $f_{A,N}(x)$.

Якщо $y = A^x$, то $x = \log_A y$.

Тому завдання звернення функції $f_{A,N}(x)$ називають завданням знаходження дискретного логарифма або завданням дискретного логарифмування.

Завдання дискретного логарифмування формулюється таким чином.

Для відомих цілих **A**, **N**, y знайти ціле число x, таке, що $A^x \pmod N = y$

Алгоритм обчислення дискретного логарифма за прийнятний час доки не знайдений. Тому модульна експонента вважається однонапрямленою функцією. За сучасними оцінками теорії чисел при цілих числах $A \approx 2^{664}$ і $N \approx 2^{664}$ рішення задачі дискретного логарифмування(знаходження показника міри x для відомого y) зажадає близько 10^{26} операцій, тобто це завдання має в 10^3 раз більшу обчислювальну складність, чим завдання розкладання на множники. При збільшенні довжини чисел різниця в оцінках складності завдань зростає.

Другим важливим класом функцій, використовуваних при побудові криптосистем з відкритим ключем, є так звані однонапрямлені функції з "потайним ходом"(з "лазівкою").

Неформальне визначення такої функції. Функція

$f: X \rightarrow Y$

відноситься до класу однонапрямлених функцій з "потайним ходом" у тому випадку, якщо

1. вона є однонапрямленою;

2. можливе ефективне обчислення зворотної функції, якщо відомий "потайний хід"(секретне число, рядок або інша інформація, що асоціюється з цією функцією).

Як приклад однонапрямленій функції з "потайним ходом" можна вказати використовувану в криптосистемі **RSA** модульну експоненту з фіксованими модулем і показником міри. Змінна основа модульної експоненти використовується для вказівки числового значення повідомлення **PT** або криптограми **CT**.

Контрольні питання:

1. Асиметричні криптосистеми.
2. Ознаки асиметричних криптосистем.
3. Вимоги до асиметричних криптосистем.
4. Однонапрямлені функції.
5. Дискретний логарифм.
6. Алгоритм обчислення дискретного логарифма.

Лекція № 10 (4 години)

Тема: Криптосистема Віжинера

Мета: Ознайомитися із принципом шифрування за допомогою криптосистеми Вижинера, правилами складання таблиці шифрування й загальним алгоритмом роботи системи

ПЛАН

1. Поняття криптосистеми Віжинера
2. Принцип шифрування
3. Таблиця Віжинера

Система Віжинера вперше була опублікована в 1586р. і є однією з найстарших і найбільш відомих багатоалфавітних систем. Свою назву вона одержала по імені французького дипломата XVI століття Блеза Віжинера, що розвивав і вдосконалював криптографічні системи.

Система Віжинера подібна до такої системи шифрування Цезаря, у якої ключ підстановки міняється від букви до букви. Цей шифр багатоалфавітної заміни можна описати таблицею шифрування, що називається таблицею (квадратом) Віжинера. На мал. 1.1 і 1.2 показані таблиці Віжинера для російського й англійського алфавітів

відповідно.

Таблиця Віжинера використовується для шифрування й розшифрування. Таблиця має два входи:

- верхній рядок підкреслених символів, який використовується для зчитування чергової букви вихідного відкритого тексту;.
- крайній лівий стовпець ключа.

Послідовність ключів зазвичай одержують із числових значень букв ключового слова.

При шифруванні вихідного повідомлення його виписують у рядок, а під ним записують ключове слово (або фразу). Якщо ключ виявився коротше повідомлення, то його циклічно повторюють. Буква шифротексту перебуває на перетинанні стовпця, обумовленого буквою, що шифрується, і рядка, обумовленого числовим значенням ключа.

Нехай ключова послідовність має довжину r. тоді ключ r-алфавітної підстановки є r-рядок

$$\bar{\pi} = (\pi_0, \pi_1, ..., \pi_{r-1}).$$

Система шифрування Віжинера перетворить відкритий текст $\bar{x} = (x_0, x_1, ..., x_{n-1})$ у шифротекст $\bar{y} = (y_0, y_1, ..., y_{n-1})$ за допомогою $\bar{\pi} = (\pi_0, \pi_1, ..., \pi_{r-1})$. відповідно до правила

$$T_\pi : \bar{x} = (x_0, x_1, ..., x_{n-1}) \to \bar{y} = (y_0, y_1, ..., y_{n-1}),$$
$$(y_0, y_1, ..., y_{n-1}) = (\pi_0(x_0), \pi_1(x_1), ..., \pi_{n-1}(x_{n-1})),$$

де $\pi_i = \pi(i \bmod r)$

Розглянемо приклад одержання шифротекста за допомогою таблиці Віжинера. Нехай обране ключове слово АМБРОЗІЯ. Необхідно зашифрувати повідомлення ПРИЛІТАЮ СЬОМОГО.

Випишемо вихідне повідомлення в рядок і запишемо під ним ключове слово з повторенням. У третій рядок будемо виписувати букви шифротекста, обумовлені з таблиці Віжинера

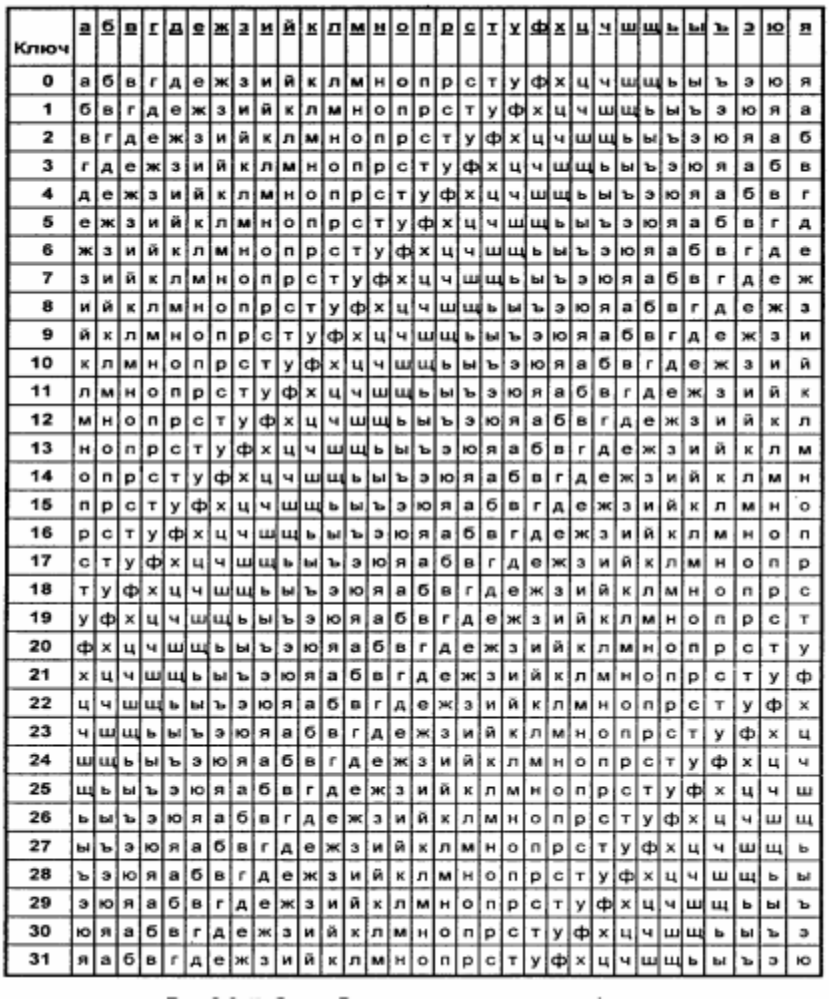
Повідомлення **П Р И Л Е Т А Ю**

С Е Д Ь М О Г О

Ключ **А М Б Р О З И Я**

А М Б Р О З И Я

Шифротекст ПЪЙЫУЩИЗ ССЕКЬХЛН



Ключ	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я
0	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я
1	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а
2	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б
3	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в
4	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г
5	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д
6	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е
7	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж
8	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з
9	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и
10	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й
11	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к
12	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л
13	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м
14	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н
15	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о
16	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п
17	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р
18	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с
19	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т
20	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у
21	х	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф
22	ц	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х
23	ч	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц
24	ш	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч
25	щ	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш
26	ь	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ
27	ы	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь
28	ъ	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы
29	э	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ
30	ю	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э
31	я	а	б	в	г	д	е	ж	з	и	й	к	л	м	н	о	п	р	с	т	у	ф	х	ц	ч	ш	щ	ь	ы	ъ	э	ю

Рис. 1.2. Таблиця Віжинера для російського алфавіта

Зруйнувати статистичні залежності в закодованих повідомленнях і тим самим підвищити надійність кодування можна за допомогою методу Віжинера. Алгоритм застосування цього методу наведений нижче:

1) Символи вихідного алфавіту нумеруються, починаючи з нуля, наприклад:

А Б В Г Д Е Ж З И К Л М Н О П Р С Т У Ф Х Ц Ч Ш Щ Ъ Ы Ь Э Ю Я

0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Одержують таблицю відповідності;

2) задаються ключем кодування - словом у вихідному алфавіті, наприклад, АСУ;

3) виписують повідомлення, підмет кодуванню, наприклад, нехай це буде повідомлення ІНФОРМАТИКА, і виконують наступні кроки:

а) під кожним його символом записують порядковий номер з таблиці

відповідності:

И Н Ф О Р М А Т И К А

8 12 19 13 15 11 0 17 8 9 0

б) під повідомленням виписують ключове слово, а під символами ключа виписують їхні порядкові номери з таблиці відповідності:

А С У А С У А С У А С

0 16 18 0 16 18 0 16 18 0 16

в) порядкові номери символів складаються по модулю, рівному числу символів вихідного алфавіту (у нашім випадку - 31):

8 28 6 13 0 29 0 2 26 9 16

Нагадаємо, що додавання по модулю (позначається ⊕) виконується без переносу одиниці переносу в старший розряд. Так ми одержали при додаванні по модулі 31, наприклад, чисел 17 і 16 (сума дорівнює 33, що на 2 перевищує модуль 31) значення 2;

4) отриманий числовий ряд перетвориться в символи вихідного алфавіту по таблиці відповідності. Так маємо:

И Ь Ж О А Ь А В Ю К С .

Очевидно, що статистика не допоможе декодувати це повідомлення, оскільки повторюються зовсім не ті символи, що у вихідному повідомленні.

Для декодування подібних повідомлень потрібна таблиця відповідності й ключ. Тоді виконують описані вище процедури кодування у зворотному порядку. Складність може представляти тільки операція вирахування з урахуванням модуля. При цьому варто пам'ятати, що не повинні виходити негативні значення. Якщо таке відбувається, потрібно зайняти число, що відповідає модулю.

Приклад 1. Декодувати повідомлення И Ь Ж О А Ь А В Ю К С , задавшись ключем АСУ й знаючи таблицю відповідності.

Рішення:

а) виписуємо під закодованим повідомленням порядкові номери символів з таблиці відповідності (див. вище):

И Ь Ж О А Ь А В Ю К С

8 28 6 13 0 29 0 2 26 9 16

б) виписуємо під повідомленням ключ із порядковими номерами символів:

А С У А С У А С У А С

0 16 18 0 16 18 0 16 18 0 16

в) віднімаємо з урахуванням модуля 31 із чисел у закодованому повідомленні

числа для ключа:

8 12 19 13 15 11 0 17 8 9 0

г) перетворимо числа в символи по таблиці відповідності:

И Н Ф О Р М А Т И К А

При декодуванні виникла складність в одержанні кодів символів Т, Ф, Р. Справді, при вирахуванні з 2 числа 16 виходило -14. Тоді до 2 додали модуль 31, одержали 33 і вже з 33 відняли 16. Одержали 17 - порядковий номер символу Т. Аналогічно чинимо й із символами Ф и Р.

Цей вид кодування використовується для зменшення обсягів інформації на носії - сигналі.

Для кодування символів вихідного алфавіту використають двійкові коди змінної довжини: чим більше частота символу, тим коротше його код. **Ефективність коду** визначається середнім числом двійкових розрядів для кодування одного символу - $I_{ср}$ по формулі

$$I_{ср} = \sum_{i=1}^{k} f_s, n_s ,$$

Де k - число символів вихідного алфавіту;

n_s - число двійкових розрядів для кодування символу s;

f_s - частота символу s; причому

$$\sum_{i=1}^{k} f_s = 1$$

При ефективному кодуванні існує межа стиску, нижче якого не «спускається» жоден метод ефективного кодування - інакше буде загублена інформація. Цей параметр визначається **граничним значенням двійкових розрядів** можливого ефективного коду - $I_{пр}$.

$$I_{пр} = -\sum_{i=1}^{n} f_i \log_2 f_i ,$$

де n - потужність кодируемого алфавіту, що кодується;

F_I - частота і-го символу алфавіту, що кодується.

Існують два класичних методи ефективного кодування: метод Шеннона-Фано й метод Хаффмена. Вхідними даними для обох методів є задана безліч вихідних символів для кодування з їхніми частотами; результат - ефективні коди.

Контрольні питання:

1. Описати принцип методу шифрування криптосистеми Вижинера.
2. Визначити тип даної криптосистеми й на основі типу дати характеристику
3. Описати типи залежності в шиврованиии криптосистемою Вижинера.

4. Алгебраїчно описати систему.
5. Використовуючи таблицю для російського алфавіту зашифрувати наступне вихідне повідомлення: «Головна умова існування вірусів універсальна інтерпретація інформації в обчислювальних системах». К=ЕФІОП.

Лекція №11 (4 години)

Тема: Криптосистема шифрування даних RSA.
Мета: Ознайомитися з алгоритмом шифрування та дешифрування RSA.

ПЛАН

1. Криптосистема RSA
2. Алгоритм шифрування
3. Алгоритм дешифрування
4. Гідності та недоліки шифру

Алгоритм RSA запропонований в 1978 р. Автори: Р. Райвест (Rivest), А. Шамир (Shamir) і А. Адлеман (Adleman). Алгоритм дістав свою назву по перших буквах прізвищ його авторів.

Алгоритм RSA став першим повноцінним алгоритмом з відкритим ключем, який може працювати як в режимі шифрування даних, так і в режимі електронного цифрового підпису.

Надійність алгоритму грунтується на важкості факторизації великих чисел і трудності обчислення дискретних логарифмів.

У криптосистемі RSA відкритий ключ $\mathbf{K_{B1}}$, секретний ключ $\mathbf{K_{B2}}$, повідомлення **PT** і криптограма **CT** належать безлічі цілих чисел

$$\mathbf{Z_N = \{0,1,2, ..., N-1\},} \quad (1)$$

де **N** - модуль:

$$\mathbf{N = P * Q} \quad (2)$$

Тут **P** і **Q** - випадкові великі прості числа.

Для забезпечення максимальної безпеки вибирають **P** і **Q** рівної довжини і зберігають в таємниці.

Безліч $\mathbf{Z_N}$ з операціями складання і множення по модулю **N** утворює арифметику по модулю **N**.

Відкритий ключ $\mathbf{K_{B1}}$ вибирають випадковим чином так, щоб виконувалися умови:

$$1 < K_{B1} \le \varphi(N), \quad (3)$$

$$НОД(K_{B1}, \varphi(N)) = 1 \quad (4)$$

$$\varphi(N) = (P-1)(Q-1) \quad (5)$$

де $\varphi(N)$ – функція Ейлера.

Функція Ейлера $\varphi(N)$ *вказує кількість позитивних цілих чисел в інтервалі від* ***1*** *до* ***N****, які взаємно прості з* ***N****.*

Нагадуємо: умова (7) означає, що відкритий ключ K_{B1} *і функція Ейлера* $\varphi(N)$ *мають бути взаємно простими.*

Використовуючи алгоритм знаходження мультиплікативної інверсії (зворотного значення), обчислюють секретний ключ K_{B2}, такий, що

$$K_{B1} * K_{B2} \equiv 1 (\mathrm{mod}\ \varphi(N)) \quad (6)$$

або

$$K_{B2} \equiv K_{B1}^{-1} (\mathrm{mod}\ (P-1)(Q-1)) \quad (7).$$

Це можна здійснити, оскільки одержувач В знає пару простих чисел (**P, Q**) і може легко знайти $\varphi(N)$.

Помітимо, що $НОД(K_{B2}, N) = 1$ - взаємно прості.

Відкритий ключ K_{B1} використовують для шифрування даних, а секретний ключ K_{B2} - для розшифрування.

Перетворення шифрування визначає криптограму **CT** через пару (відкритий ключ K_{B1}, повідомлення **PT**) відповідно до наступної формули:

$$CT = E_{K_{B1}}(PT) = E_B(PT) = PT^{K_{B1}} (\mathrm{mod}\ N) \quad (8)$$

В якості алгоритму швидкого обчислення значення **CT** використовують відомі властивості модулярной арифметики(ряд послідовних зведень в квадрат цілого числа і множень на ціле число з приведенням по модулю **N**).

Звернення функції $C = M_i^{K_{B1}} (\mathrm{mod}\ N)$, тобто визначення значення **Mi** по відомих значеннях **Ci,** K_{B1} і **N**, практично не здійсненно при $N \approx 2^{512}$.

Проте зворотне завдання, тобто завдання дешифрування криптограми **CT**, можна вирішити, використовуючи пару(секретний ключ K_{B2}, криптограма ct) по наступній формулі:

$$\mathbf{PT = D_{K_{B2}}(CT) = D_B(CT) = CT^{K_{B2}} (mod\ N)} \quad (9)$$

Процес дешифрування можна записати так:

$$\mathbf{D_B(E_B(PT)) = PT} \quad (10)$$

Підставляючи в (10) значення (9) і (8), отримуємо: $\mathbf{\left(PT^{K_{B1}}\right)^{K_{B2}} = PT(mod\ N)}$

Чи $\mathbf{PT^{K_{B1}K_{B2}} = PT(mod\ N)}$ (11)

Величина $\varphi(N)$ відіграє важливу роль в теоремі Ейлера, яка стверджує, що якщо $НОД(x, N) = 1$ то $\mathbf{x^{\varphi(N)} \equiv 1(mod\ N)}$

чи в дещо загальнішій формі, якщо **d** - ціле число

$$\mathbf{x^{d\cdot\varphi(N)+1} \equiv x(mod\ N)} \quad (12)$$

Зіставляючи вирази (16) і(15), отримуємо $\mathbf{K_{B1} \cdot K_{B2} = d \cdot \varphi(N) + 1}$

чи, що те ж саме $\mathbf{K_{B1} \cdot K_{B2} \equiv 1(mod\ \varphi(N))}$

Саме тому для обчислення секретного ключа K_{B2} використовують співвідношення (10).

Таким чином, якщо криптограму $\mathbf{CT = PT^{K_{B1}} (mod\ N)}$

піднести до степеня K_{B2} те в результаті відновлюється початковий відкритий текст **PT**, оскільки

$$\mathbf{\left(PT^{K_{B1}}\right)^{K_{B2}} = PT^{K_{B1}K_{B2}} = PT^{d\cdot\varphi(N)+1} \equiv PT(mod\ N)}$$

Тобто, одержувач **B**, який створює криптосистему, захищає два параметри:

1) секретний ключ K_{B2};

2) пару чисел(**P, Q**), твір яких дає значення модуля **N**.

З іншого боку, одержувач **У** відкриває значення модуля **N** і відкритий ключ K_{B1}.

Супротивникові відомі лише значення K_{B1} і **N**.

Якби він зміг розкласти число **N** на множники **P** і **Q**, то він упізнав би "потайний хід" - трійку чисел {**P, Q,** K_{B1}}, вичислив значення функції Ейлера $\varphi(N) = (P-1)(Q-1)$ і визначив значення секретного ключа K_{B2}.

Проте, розкладання дуже великого **N** на множники обчислювальне не здійснено (за умови, що довжини вибраних **P** і **Q** ≥ 100 десяткових знаків).

Процедури шифрування і дешифрування в криптосистемі RSA

А - відправник

У - отримувач

Криптосистему RSA повинні сформувати одержувач повідомлення - користувач **В**.

Послідовність дій :

I. Дії користувача **В**

1) Вибирає два довільних великих простих числа **P** і **Q**.

2) Обчислює значення модуля **N = P * Q**.

3) Обчислює функцію Ейлера $\varphi(N) = (P-1)(Q-1)$

Вибирає випадковим чином значення відкритого ключа K_{B1} з урахуванням виконання умов : $1 < K_{B1} \le \varphi(N)$, $НОД(K_{B1}, \varphi(N)) = 1$

4) Обчислює значення секретного ключа K_{B2}, використовуючи властивості арифметики по модулю (наприклад, розширений алгоритм Евкліда) при рішенні порівняння

5) Пересилає користувачеві **А** пару чисел (**N**, K_{B1}) по незахищеному каналу.

II. Дії користувача А

6) Розбиває початковий відкритий текст РТ на блоки, кожен з яких може бути представлений у вигляді числа

pi = 0,1, 2, ..., N - 1.

7) Зашифровує текст, представлений у вигляді послідовності чисел **pi**, по формулі $c_i = p_i^{K_{B1}} (\mathbf{mod}\ N)$

і відправляє криптограму

c1, c2, c3 ..., ci...

користувачеві B.

III. Дії користувача B

8) Розшифровує прийняту криптограму

c1, c2, c3 ..., ci...

використовуючи секретний ключ K_{B2}, по формулі $\mathbf{p_i = c_i^{K_{B2}} (mod\ N)}$

В результаті буде отримана послідовність чисел **pi**, які є початковим повідомленням **PT**.

Щоб алгоритм RSA мав практичну цінність, необхідно мати можливість без істотних витрат генерувати великі прості числа, уміти оперативно обчислювати значення ключів K_{B1} і K_{B2}.

Приклад

Шифрування повідомлення CAB.

Заданий англійський алфавіт

1	*2*	*3*	*...*	*26*
A	*B*	*C*	*...*	*Z*

Для простоти обчислень використовуватимуться невеликі числа. На практиці застосовуються дуже великі числа.

Дії користувача B.

1) Вибирає P = 3 і Q =11.

2) Обчислює модуль N = P*Q = 3*11 = 33.

3) Обчислює значення функції Ейлера для N = 33 :

$$\varphi(N) = (P-1)(Q-1) = 2*10 = 20$$

4) Вибирає в якості відкритого ключа K_{B1} довільне число з урахуванням виконання умов :

$$1 < K_{B1} \le 20 \text{ , } НОД(K_{B1}, 20) = 1$$

Нехай K_{B1}=7.

4) Обчислює значення секретного ключа K_{B2}, використовуючи розширений алгоритм Евкліда при рішенні порівняння $\mathbf{K_{B2} \equiv 7^{-1} (mod\ 20)}$

$\varphi(N)$	K_{B1}	$\varphi(N) \bmod K_{B1}$	int ($\varphi(N) / K_{B1}$)	u	v
20	7	6	2		
7	6	1	1		
6	1	0	6		
1	0	–	–		

Перші три колонки це звичайний алгоритм Евкліда, отже обчислюємо $\mathbf{НОД(7, 20)}$. Останнє значення a=1, при якому b=0 дає результат $\mathbf{НОД(7, 20) = 1}$.

int($\varphi(N) / K_{B1}$) - ціла частина ділення.

$\varphi(N) \bmod K_{B1}$ - залишок від ділення

$\varphi(N)$	K_{B1}	$\varphi(N) \bmod K_{B1}$	int ($\varphi(N) / K_{B1}$)	u	v
20	7	6	2	–1	3
7	6	1	1	1	–1
6	1	0	6	0	1
1	0	–	–	1	0

Рішення дає K_{B2} = 3.

5) Пересилає користувачеві **А** пару чисел (N = 33. K_{B1} =7).

Дії користувача А.

6) Представляє початкове повідомлення як послідовність цілих чисел в діапазоні 0.32. (0.N - 1)

Нехай буква А представляється як число 1, буква В - як число 2, буква З - як число 3. Тоді со-общение САВ можна представити як послідовність чисел 312, тобто p1 = 3, p2 = 1, p3 = 2.

7) Зашифровує текст, представлений у вигляді послідовності чисел p1, p2 і p3, використовуючи ключ K_{B1}=7 і N = 33, по формулі

$$c_i = p_i^{K_{B1}} (\mathrm{mod}\ N) = p_i^7 (\mathrm{mod}\ 33)$$

Отримує

а) $c_1 = 3^7 (\mathrm{mod}\ N) = \ldots = 9$

б) $c_2 = 1^7 (\mathrm{mod}\ N) = 1 (\mathrm{mod}\ 33) = 1$

в) $c_3 = 2^7 (\mathrm{mod}\ N) = 128 (\mathrm{mod}\ 33) = 29$

Відправляє користувачеві **B** криптограму

c1, c2. c3 = 9, 1, 29.

Дії користувача B.

1) Розшифровує прийняту криптограму c1, c2. c3, використовуючи секретний ключ K_{B2}= 3. по формулі

$$p_i = c_i^{K_{B2}} (\mathrm{mod}\ N)$$

Отримує

а) $p_1 = 9^3 (\mathrm{mod}\ 33) = \ldots = 3$

б) $p_2 = 1^3 (\mathrm{mod}\ 33) = 1 (\mathrm{mod}\ 33) = 1$

в) $p_3 = 29^3 (\mathrm{mod}\ 33) = \ldots = 2$

Таким чином, відновлено початкове повідомлення:

C A B

3 1 2

Безпека і швидкодія криптосистеми RSA

Безпека алгоритму RSA базується на труднощі рішення задачі факторизації (розкладання на множники) великих чисел.

Криптостійкість алгоритму RSA визначається тим, що після формування секретного ключа K_{B2} і відкритого ключа K_{B1} "стираються" значення простих чисел **P і Q**, і тоді виключно важко визначити секретний ключ K_{B2} по відкритому ключу K_{B1}, оскільки для цього необхідно вирішити за-дачу знаходження дільників **P і Q** модуля **N**.

Розкладання величини **N** на прості множники **P і Q** дозволяє вичислити функцію $\varphi(N)=(P-1)(Q-1)$ і потім визначити секретне значення K_{B2}, використовуючи рівняння $K_{B2} \equiv K_{B1}^{-1}(\mathrm{mod}\ \varphi(N))$

Інший спосіб криптоаналізу алгоритму RSA - обчислення або підбір значення функції $\varphi(N)=(P-1)(Q-1)$

Якщо встановлено значення $\varphi(N)$, то співмножники P і Q обчислюються з системи

$$x = P + Q = N + 1 - \varphi(N),$$
$$y = (P - Q)^2 = (P + Q)^2 - 4 * N.$$

Знаючи, можна визначити x, y; потім можна визначити числа P і Q з наступних співвідношень:

$$P = 1/2\ (x + \sqrt{y}\ ), \quad Q = 1/2\ (x - \sqrt{y}\ ).$$

Проте ця атака не простіше завдання факторизації модуля N.

Завдання факторизації є важко вирішуваним завданням для великих значень модуля N.

Спочатку автори алгоритму RSA пропонували для обчислення модуля N вибирати прості числа P і Q випадковим чином, по 50 десяткових розрядів кожне. Вважалося, що такі великі числа N дуже важко розкласти на прості множники. Один з авторів алгоритму RSA, P. Райвест, вважав, що розкладання на прості множники числа з майже 130 десяткових цифр, приведеного в їх публікації, зажадає більше 40 квадрилліонів років машинного часу.

Проте цей прогноз не оправдилися через порівняно швидкий прогрес комп'ютерів і їх обчислювальної потужності, а також поліпшення алгоритмів факторизації.

Один з найбільш швидких алгоритмів, відомих нині, алгоритм NFS(Number Field Sieve) може виконати факторизацію великого числа N(з числом десяткових розрядів більше 120) за число кроків, що оцінюються величиною

$$e^{2(\ln n)^{1/3}(\ln(\ln n))^{2/3}}$$

У 1994 р. було факторизовано число з 129 десятковими цифрами. Це вдалося здійснити математикам А. Ленстра і М. Манасси за допомогою організації розподілених обчислень на 1600 комп'ютерах, об'єднаних мережею, впродовж восьми місяців. На думку А. Ленстра і М. Манасси, їх робота компрометує криптосистеми RSA і створює велику загрозу їх подальшим застосуванням.

Тепер розробникам криптоалгоритмов з відкритим ключем на базі RSA доводиться из-бегать застосування чисел завдовжки менше 200 десяткових розрядів. Самі останні публікації пропонують застосовувати для цього числа завдовжки не менше 250-300 десяткових розрядів.

Тепер розробникам криптоалгоритмів з відкритим ключем на базі RSA доводиться уникати застосування чисел завдовжки менше 200 десяткових розрядів. Самі останні публікації пропонують застосовувати для цього числа завдовжки не менше 250-300 десяткових розрядів.

Зроблена спроба розрахунку оцінок безпечних довжин ключів асиметричних криптосистем на найближчі 20 років виходячи з прогнозу розвитку комп'ютерів і їх обчислювальної потужності, а також можливого вдосконалення алгоритмів факторизації. Ці оцінки (таблиця 1) дані для трьох груп користувачів (індивідуальних користувачів, корпорацій і державних організацій), відповідно до відмінності вимог до їх інформаційної безпеки.

Звичайно, ці оцінки слід розглядати як суто приблизні, як можливу тенденцію змін безпечних довжин ключів асиметричних криптосистем з часом.

Таблиця 1 - Оцінки довжин ключів для асиметричних криптосистем, біт

Год	**Отдельные пользователи**	**Корпорации**	**Государственные организации**
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

Криптосистеми RSA реалізуються як апаратним, так і програмним шляхом.

Для апаратної реалізації операцій зашифрования і расшифрования RSA розроблені спеціальні процесори. Ці процесори, реалізовані на надвеликих інтегральних схемах(СБИС), дозволяють виконувати операції RSA, пов'язані з піднесенням великих чисел до колосально великого степеня по модулю N, за відносно короткий час.

Та все ж апаратна реалізація RSA приблизно в 1000 разів повільніше за апаратну реалізацію симетричного криптоалгоритма DES.

Одна з найшвидших апаратних реалізацій RSA з модулем N=512 біт на надвеликій інтегральній схемі має швидкодію 64 Кбит/с. Кращими з тих, що серійно випускаються СБИС являються процесори фірми CYLINK, що виконують 1024-бітове шифрування RSA.

Недолік

асиметрична криптосистема RSA має малу швидкодію в порівнянні з симетричними криптосистемами.

Програмна реалізація RSA приблизно в 100 разів повільніше за програмну реалізацію DES.

З розвитком технології ці оцінки можуть дещо змінюватися, але асиметрична криптосистема RSA ніколи не досягне швидкодії симетричних криптосистем.

Слід зазначити, що мала швидкодія криптосистем RSA обмежує сферу їх застосування, але не перекреслює їх цінність.

Контрольні питання:

1. Криптосистема RSA.
2. Алгоритм шифрування.
3. Алгоритм дешифрування

Лекція №12 (2 години)

Тема: Засоби керування системою.

Мета: Ознайомитися із поняттям комплексу засобів захисту та системою захисту інформації

ПЛАН

1. Поняття комплексу засобів захисту (КЗЗ)
2. Засоби керування системою
3. Взаємодія об’єктів системи
4. Комплексна система захисту інформації (КСЗІ)

Комп'ютерна система, як правило, складається з безлічі компонентів. Деякі з компонентів можуть бути спеціально призначені для реалізації політики безпеки (наприклад, засоби ізоляції процесів або керування потоками інформації). Інші можуть впливати на безпеку опосередковано, наприклад, забезпечувати функціонування компонентів першого типу. І, нарешті, треті можуть взагалі не бути задіяні під час вирішення завдань забезпечення безпеки. Множина всіх компонентів перших двох типів називається комплексом засобів захисту.

Іншими словами, КЗЗ — це сукупність всіх програмно-апаратних засобів, в тому числі програм ПЗП, задіяних під час реалізації політики безпеки. Частина КС, що складає КЗЗ, визначається розробником. Будь- який компонент КС, який внаслідок якого-небудь впливу здатний спричинити порушення політики безпеки, повинен розглядатись як частина КЗЗ.

Комплекс засобів захисту розглядає ресурси КС як об'єкти і керує взаємодією цих об'єктів відповідно до політики безпеки інформації, що реалізується. Як об'єкти

ресурси характеризуються двома аспектами: логічне подання (зміст, семантика, значення) і фізичне (форма, синтаксис). Об'єкт характеризується своїм станом, що в свою чергу характеризується атрибутами і поводженням, яке визначає способи зміни стану. Для різних КС об'єкти можуть бути різні. Наприклад, для СУБД в якості об'єктів можна розглядати записи БД, а для операційної системи — процеси, файли, кластери, сектори дисків, сегменти пам'яті і т. ін.

Все, що підлягає захисту відповідно до політики безпеки, має бути визначено як об'єкт.

При розгляді взаємодії двох об'єктів КС, що виступають як приймальники або джерела інформації, слід виділити пасивний об'єкт, над яким виконується операція, і активний об'єкт, який виконує або ініціює цю операцію. Далі розглядаються такі типи об'єктів КС: об'єкти-користувачі, об'єкти-процеси і пасивні об'єкти. Прийнятий у деяких зарубіжних документах термін "суб'єкт" є суперпозицією об'єкта-користувача і об'єкта-процеса.

Об'єкти-користувачі і об'єкти-процеси є такими тільки всередині конкретного домену — ізольованої логічної області, всередині якої об'єкти володіють певними властивостями, повноваженнями і зберігають певні відносини. В інших доменах об'єкти залишаються в пасивному стані. Це дозволяє одному об'єкту-процесу керувати іншим об'єктом-процесом або навіть об'єктом-користувачем, оскільки останній залишається "пасивним" з точки зору керуючого об'єкта. Іншими словами, об'єкти можуть знаходитись в одному з трьох різних станів: об'єкт-користувач, об'єкт-процес і пасивний об'єкт. Перехід між станами означає, що об'єкт просто розглядається в іншому контексті.

Пасивний об'єкт переходить в стан об'єкта-користувача, коли індивід (фізична особа-користувач) «входить» в систему. Цей об'єкт- користувач виступає для КЗЗ як образ фізичного користувача. Звичайно, за цим процесом іде активізація об'єкта-процесу за ініціативою користувача. Цей об'єкт-процес є керуючим для пасивних об'єктів всередині домену користувача. Об'єкти-користувачі, об'єкти-процеси і пасивні об'єкти далі позначаються просто як користувачі, процеси і об'єкти, відповідно. Взаємодія двох об'єктів КС (звернення активного об'єкта до пасивного з метою одержання певного виду доступу) приводить до появи потоку інформації між об'єктами і/або зміни стану системи. Як потік інформації розглядається будь-яка порція інформації, що передається між об'єктами КС.

Для забезпечення безпеки інформації під час її обробки в АС створюється КСЗІ, процес управління якою повинен підтримуватись протягом всього життєвого циклу АС. На стадії розробки метою процесу управління КСЗІ є створення засобів захисту, які могли б ефективно протистояти ймовірним загрозам і забезпечували б надалі дотримання політики безпеки під час обробки інформації. На стадії експлуатації АС метою процесу управління КСЗІ є оцінка ефективності створеної КСЗІ і вироблення додаткових (уточнюючих) вимог для доробки КСЗІ з метою забезпечення її

адекватності при зміні початкових умов (характеристик ОС, оброблюваної інформації, фізичного середовища, персоналу, призначення АС, політики безпеки і т.ін.).

На кожному етапі мають бути виконані збирання і підготовка даних, їх аналіз і прийняття рішення. При цьому результати виконаного на певному етапі аналізу і прийняті на їх підставі рішення нарівні з уточненими вимогами слугують вихідними даними для аналізу на наступному етапі. На будь-якій стадії або будь-якому етапі може постати необхідність уточнення початкових умов і повернення на більш ранні етапи.

Створення КСЗІ має починатись з аналізу об'єкта захисту і можливих загроз. Передусім мають бути визначені ресурси АС, що підлягають захисту. Загрози мають бути визначені в термінах ймовірності їх реалізації і величини можливих збитків. На підставі аналізу загроз, існуючих в системі вразливостей, ефективності вже реалізованих заходів захисту для всіх ресурсів, що підлягають захисту, мають бути оцінені ризики. Ризик являє собою функцію ймовірності реалізації певної загрози, виду і величини завданих збитків. Величина ризику може бути виражена в грошовому вимірі або у вигляді формальної оцінки (високий, низький і т. ін.). На підставі виконаної роботи мають бути вироблені заходи захисту, перетворення яких в життя дозволило б знизити рівень остаточного ризику до прийнятного рівня.

Підсумком даного етапу робіт повинна стати сформульована або скоригована політика безпеки.

На підставі проведеного аналізу ризиків сформульованої політики безпеки розробляється план захисту, який включає в себе опис послідовності і змісту всіх стадій і етапів життєвого циклу КСЗІ, що мають відповідати стадіям і етапам життєвого циклу АС.

Вартість заходів щодо захисту інформації має бути адекватною розміру можливих збитків.

Контрольні питання:

1. Дайте визначення поняття комплексу засобів захисту (КЗЗ)
2. Опишіть засоби керування системою
3. Розкажіть про взаємодію об’єктів системи
4. Опишіть комплексну систему захисту інформації (КСЗІ)

Лекція № 13 (2 години)

Тема: Погрози безпеки.

Мета: Ознайомитися із поняттям безпеки та основними погрозами.

ПЛАН

1. Поняття інформаційної безпеки

2. Погрози безпеки
3. Класифікація погроз безпеки

Під *інформаційною безпекою* ми будемо розуміти захищеність інформації і підтримуючої інфраструктури від випадкових чи навмисних впливів природного чи штучного характеру, що можуть завдати неприйнятної шкоди суб'єктам інформаційних відносин, у тому числі власникам і користувачам інформації і підтримуючої інфраструктури.

Захист інформації — це комплекс заходів, спрямованих на забезпечення інформаційної безпеки.

Загрози інформаційної безпеки — це зворотний бік використання інформаційних технологій.

Загроза — це потенційна можливість певним чином порушити інформаційну безпеку.

Спроба реалізації загрози називається *атакою*, а той, хто починає таку спробу, — *зловмисником.* Потенційні зловмисники називаються *джерелами загрози.*

Найчастіше загроза є наслідком наявності вразливих місць у захисті інформаційних систем (наприклад, можливість доступу сторонніх осіб до критично важливого обладнання чи помилки в програмному забезпеченні).

Проміжок часу від моменту, коли з'являється можливість використовувати слабке місце, і до моменту, коли пробіл ліквідується, називається *вікном небезпеки,* асоційованим з даним вразливим місцем. Поки існує вікно небезпеки, можливі успішні атаки на ІС.

Загрози можна класифікувати по декількох критеріях:

- по аспектам інформаційної безпеки (приступність, цілісність, конфіденційність), на порушення яких загрози націлені в першу чергу;
- по компонентах інформаційних систем, на які загрози націлені (дані, програми, апаратура, що підтримує ІС, інфраструктура);
- по способу здійснення (випадкові/навмисні дії природного/техногенного характеру);
- по розташуванню джерела загроз (всередині/поза ІС).

Найбільш розповсюджені погрози приступності

Найчастішими і найнебезпечнішими з погляду розміру збитку є ненавмисні помилки штатних користувачів, операторів, системних адміністраторів і інших облич, що обслуговують інформаційні системи. Це можуть бути або неправильно введені чи дані помилка в програмі, що викликала крах системи. До 65% утрат — наслідок таких ненавмисних помилок.

Найрадикальніший спосіб боротьби з ненавмисними помилками — максимальна автоматизація і строгий контроль.

Інші погрози доступності можна класифікувати по компонентах ІС, на які націлені погрози:

- відмова користувачів;
- внутрішнє відмова інформаційної системи;
- відмова підтримуючої інфраструктури.

Стосовно підтримуючої інфраструктури рекомендується розглядати наступні погрози:

- порушення роботи (випадкове чи навмисне) систем зв'язку, електроживлення, водо- і/чи теплопостачання, кондиціонування;
- руйнування або ушкодження приміщень;
- неможливість небажання обслуговуючого персоналу або користувачів виконувати свої обов'язки (цивільні безладдя, аварії на транспорті, терористичний акт чи його загроза, страйк і т.п.).

Погрозами динамічної цілісності є порушення атомарності транзакцій, крадіжка, дублювання чи даних внесення додаткових повідомлень (мережних пакетів і т.п.). Відповідні дії в мережному середовищі називаються *активним прослуховуванням.*

Конфіденційну інформацію можна розділити на *предметну* і *службову.* Службова інформація (наприклад, паролі користувачів) не відноситься до визначеної предметної області, в інформаційній системі вона відіграє технічну роль, але її розкриття особливе небезпечно, оскільки воно чревате одержанням несанкціонованого доступу до всієї інформації, у тому числі предметної.

Можна виділити наступні загрози конфіденційності:

1. розміщення конфіденційних даних у середовищі, де їм не забезпечено необхідний захист (багатьом людям доводиться виступати як користувачем не однієї, а цілого ряду систем (інформаційних сервисів). Якщо для доступу до таких систем використовуються багаторазові паролі чи інша конфіденційна інформація, то напевно ці дані будуть зберігатися не тільки в голові, але й у записній чи книжці на листках папера, що користувач часто залишає на робочому столі, а то і попросту втрачає. Неможливо пам'ятати багато різних паролів; рекомендації з їх регулярної (по можливості — частої) зміні тільки збільшують положення, змушуючи застосовувати нескладні схеми чи чергування взагалі намагатися звести справу до двох-трьох (і настільки ж легко вгадуються) паролів, що легко запам'ятовуються. Погроза ж полягає в тому, що хтось не відмовиться довідатися секрети, що самі просяться в руки.
2. Передача конфіденційних даних у відкритому виді (у розмові, у листі, по мережі), що уможливлює перехоплення даних. Для атаки можуть використовуватися різні технічні засоби (підслуховування чи прослуховування розмов, пасивне прослуховування мережі і т.п.), але ідея

одна — здійснити доступ до даних у той момент, коли вони найменш захищені.

3. Збереження даних на резервних носіях. Для захисту даних на основних носіях застосовуються розвинуті системи керування доступом; копії ж нерідко просто лежать у шафах і одержати доступ до них може будь-хто.
4. *Перехоплення даних* — дуже серйозна погроза, і якщо конфіденційність дійсно є критичною, а дані передаються по багатьом каналам, їхній захист може виявитися дуже складним і дорогим. Технічні засоби перехоплення добре пророблені, доступні, прості в експлуатації, а установити їх, наприклад на кабельну мережу, може хто завгодно, так що цю погрозу потрібно брати до уваги по відношенню не тільки до зовнішніх, але і до внутрішніх комунікацій.
5. Крадіжки устаткування (є загрозою не тільки для резервних носіїв, але і для комп'ютерів, особливо портативних. Часто ноутбуки залишають без догляду на чи роботі в автомобілі, іноді просто втрачають).
6. *маскарад* — виконання дій під видом обличчя, що володіє повноваженнями для доступу до даних.
7. Зловживання повноваженнями (на багатьох типах систем привілейований користувач (наприклад системний адміністратор) здатний прочитати будь-який файл, одержати доступ до пошти будь-якого користувача і т.п.)

Контрольні питання:

1. Дайте визначення поняття інформаційної безпеки
2. Перерахуйте погрози безпеки
3. Класифікуйте та опишіть погрози безпеки

Лекція № 14 (2 години)

Тема: Шкідливе програмне забезпечення.

Мета: Ознайомитися із поняттям та типами шкідливого програмного забезпечення

ПЛАН

1. Поняття шкідливого програмного забезпечення
2. Класифікація шкідливого програмного забезпечення
3. Поняття вікна небезпеки

Одним з найнебезпечніших способів проведення атак є впровадження в системи шкідливого програмного забезпечення. Виділено наступні грані шкідливого ПО:

- шкідлива функція;
- спосіб поширення;
- зовнішнє представлення.

Частина, що здійснює руйнівну функцію, називається "бомбою" (хоча, можливо, більш удалими термінами були б "заряд" чи "боєголовка"). Спектр шкідливих функцій необмежений, оскільки "бомба", як і будь-яка інша програма, може володіти якою завгодно складною логікою, але звичайно "бомби" призначаються для:

- впровадження іншого шкідливого ПЗ;
- одержання контролю над системою, що атакується;
- агресивного споживання ресурсів;
- чи зміни руйнування програм і/чи даних.

По механізму поширення розрізняють:

- віруси — код, що володіє здатністю до поширення (можливо, зі змінами) шляхом впровадження в інші програми;
- "хробаки" — код, здатний самостійно, тобто без впровадження в інші програми, викликати поширення своїх копій по ІС і їхнє виконання (для активізації вірусу потрібний запуск зараженої програми).

Віруси зазвичай поширюються локально, у межах вузла мережі; для передачі по мережі їм потрібна зовнішня допомога, така як пересилання зараженого файлу. "Хробаки", навпаки, орієнтовані в першу чергу на подорожі по мережі.

Іноді саме поширення шкідливого ПЗ викликає агресивне споживання ресурсів - отже, є шкідливою функцією. Наприклад, "хробаки" "з'їдають" смугу пропущення мережі і ресурси поштових систем. З цієї причини для атак на приступність вони не мають потреби у вбудовуванні спеціальних "бомб".

Шкідливий код, що виглядає як функціонально корисна програма, називається троянським. Наприклад, звичайна програма, будучи ураженою вірусом, стає троянською; часом троянські програми виготовляють вручну і підсовують довірливим користувачам у якому-небудь привабливому упакуванні.

Відзначимо, що дані нами визначення і приведена класифікація шкідливого ПЗ відрізняються від загальноприйнятих. Наприклад, У Д С Т Р 51275-99 "Захист інформації. Об'єкт інформатизації. Фактори, що впливають на інформацію. Загальні положення" міститься наступне визначення:

"Програмний вірус — виконаний це чи інтерпретований програмний код, що володіє властивістю несанкціонованого поширення і самовідтворення в автоматизованих чи системах телекомунікаційних мереж з метою змінити чи знищити програмне забезпечення і/чи дані, що зберігаються в автоматизованих системах".

Подібне визначення досить невдале, оскільки в ньому змішані функціональні і транспортні аспекти.

Вікно небезпеки для шкідливого ПЗ з'являється з випуском нового різновиду "бомб", вірусів і/чи "хробаків" і перестає існувати з відновленням бази даних антивірусних програм і накладенням інших необхідних латок.

За традицією з усього шкідливого ПЗ найбільша увага громадськості приходиться на частку вірусів. Однак до березня 1999 року з повним правом можна

було затверджувати, що "незважаючи на експонентний ріст числа відомих вірусів, аналогічного росту кількості інцидентів, викликаних ними, не зареєстроване. Дотримання нескладних правил "комп'ютерної гігієни" практично зводить ризик зараження до нуля. Там, де працюють, а не грають, число заражених комп'ютерів складає лише частки відсотка".

Контрольні питання:

1. Дайте визначення поняття шкідливого програмного забезпечення
2. Класифікуйте шкідливе програмне забезпечення
3. Дайте визначення поняття вікна небезпеки

Лекція № 15 (2 години)

Тема: Комп’ютерні віруси та їх властивості.

Мета: Ознайомитися із поняттям комп’ютерних вірусів, їх ознаками та властивостями

ПЛАН

1. Поняття комп’ютерних вірусів
2. Властивості комп’ютерних вірусів
3. Класифікація комп’ютерних вірусів

Практична робота на персональному комп'ютері із прикладними програмами й реальні ситуації в комп'ютері часто чекають від вас уміння зберігати в цілісності інформацію, захищати її від можливих руйнувань, викликаних дефектами магнітних дисків, збоями в роботі комп'ютера, впливами програмних вірусів або особисто вашими помилками.

На жаль, сьогодні масове застосування персональних комп'ютерів, використання мережі Інтернет виявилося пов'язаним з появою програм-вірусів, що перешкоджають нормальній роботі комп'ютера, що руйнують файлову структуру дисків і наносять збиток збереженій в комп'ютері інформації.

Незважаючи на прийняті в багатьох країнах закони про боротьбу з комп'ютерними злочинами й розробку спеціальних програмних засобів захисту від вірусів, кількість нових програмних вірусів постійно росте. Це вимагає від користувача персонального комп'ютера знань про природу вірусів, способах зараження вірусами й захисту від них.

Комп'ютерні віруси і їхні властивості

Насамперед комп'ютерний вірус - це програма, що володіє здатністю до самовідтворення. Така здатність є єдиним засобом, властивим всім типам вірусів.

Вірус не може існувати в «повній ізоляції»: не можна уявити собі вірус, що не

використовує код інших програм, інформацію про файлову структуру або навіть просто імена інших програм. Причина зрозуміла: вірус повинен яким-небудь способом забезпечити передачу собі керування.

Комп'ютерний вірус - спеціально написана програма, здатна мимовільно приєднуватися до інших програм, створювати свої копії й впроваджувати їх у файли, системні області комп'ютера й в обчислювальні мережі з метою порушення роботи програм, псування файлів і каталогів, створення всіляких перешкод у роботі комп'ютера.

У наш час відомі тисячі комп'ютерних вірусів, їх можна класифікувати по наступних ознаках:

- середовищу перебування;
- способу зараження середовища перебування;
- впливу;
- особливостям алгоритму.

Залежно від середовища перебування віруси можна розділити на мережні, файлові, завантажувальні й файлово-завантажувальні. Мережні віруси поширюються по різних комп'ютерних мережах. Файлові віруси впроваджуються головним чином у виконувані модулі, тобто у файли, що мають розширення COM і EXE. Файлові віруси можуть впроваджуватися і в інші типи файлів, але, як правило, записані в таких файлах, вони ніколи не одержують керування, а отже - втрачають здатність до розмноження. Завантажувальні віруси впроваджуються в завантажувальний сектор диска (Boot-сектор) або в сектор, що містить програму завантаження системного диска (Master Boot Record). Файлово-завантажувальні віруси заражають як файли, так і завантажувальні сектори дисків.

По способу зараження віруси діляться на такі, що при зараженні (інфікуванні) комп'ютера залишають в оперативній пам'яті свою частину, що потім перехоплює обіг операційної системи до об'єктів зараження (файлам, завантажувальним секторам дисків і т.п.) і впроваджуються в них, перебувають у пам'яті і є активними аж до вимикання або перезавантаження комп'ютера, і такі віруси, які не вражають пам'ять комп'ютера і є активними обмежений час.

По ступеню впливу віруси можна розділити на наступні види:

- безпечні, що не заважають роботі комп'ютера, але зменшують обсяг вільної оперативної пам'яті й пам'яті на дисках, дії таких вірусів проявляються в яких-небудь графічних або звукових ефектах;
- небезпечні віруси, які можуть привести до різних порушень у роботі комп'ютера;
- дуже небезпечні, вплив яких може привести до втрати програм, знищенню даних, стиранню інформації в системних областях диска.

По особливостях алгоритму віруси важко класифікувати через велику розмаїтість. Найпростіші віруси - паразитичні, вони змінюють вміст файлів і секторів

диска й можуть бути досить легко виявлені та знищені. Можна відзначити віруси-реплікатори, що називаються хробаками, які поширюються по комп'ютерних мережах, обчислюють адреси мережних комп'ютерів і записують по цих адресах свої копії. Відомі віруси-невидимки, що називаються стелс-вірусами, які дуже важко виявити й знешкодити, тому що вони перехоплюють обіги операційної системи до уражених файлів і секторів дисків і підставляють замість свого тіла незаражені ділянки диска. Найбільш важко виявити віруси-мутанти, що містять алгоритми шифровки-розшифровки, завдяки яким копії того самого вірусу не мають ні одного повторюваного ланцюжка байтів. Є й так звані квазівірусні, або троянські програми, які хоча й не здатні до самопоширення, але дуже небезпечні, тому що, маскуючись під корисну програму, руйнують завантажувальний сектор і файлову систему дисків.

Основні види вірусів та схеми їх функціонування.

Завантажувальні віруси

Завантажувальні віруси заражають завантажувальні (Boot) сектори гнучких дисків і Boot-сектора або Master Boot Record (MBR) жорстких дисків (рис. 12.1). У секторі початкового завантаження зберігається інформація про з’ємний пристрій - кількість поверхонь, кількість доріжок, кількість секторів та ін.

Завантажувальні віруси є резидентними. Зараження відбувається при завантаженні операційної системи з дисків. У завантажувальних вірусах виділяють дві частини – голову і хвіст. Хвіст може бути порожнім.

Нормальна схема початкового завантаження наступна:

ППЗ (ПЗП) □ ППЗ (диск) □ СИСТЕМА

Після включення ЕОМ здійснюється контроль її працездатності за допомогою програми, записаної в постійному запам’ятовуючому пристрої. Якщо перевірка завершилася успішно, то здійснюється прочитування першого сектора з гнучкого або жорсткого диска. Порядок використання дисководів для завантаження задається користувачем за допомогою програми Setup. Якщо диск, з якого проводиться завантаження ОС заражений завантажувальним вірусом, то зазвичай виконуються наступні кроки:

Крок 1. Зчитаний з 1-го сектора диска завантажувальний вірус (частина вірусу) отримує управління, зменшує об'єм вільної пам'яті ОП і прочитує з диска тіло вірусу.

Крок 2. Вірус переписує сам себе в іншу область ОП, найчастіше - в старші адреси пам'яті.

Крок 3. Встановлюються необхідні вектора переривань (вірус резидентний).

Крок 4. При виконанні певних умов проводяться деструктивні дії.

Крок 5. Копіюється Boot-сектор в ОП і передається йому управління.

Рис. 12.1. Розміщення завантажувального вірусу на диску

Якщо вірус був активізований з гнучкого диска, то він записується в завантажувальний сектор жорсткого диска. Активний вірус, постійно знаходячись в ОП, заражає завантажувальні сектори всіх гнучких дисків, а не тільки системні диски.

Зараження робочих гнучких дисків завантажувальними вірусами виконується з розрахунку на помилкові дії користувача ЕОМ у момент завантаження ОС. Якщо встановлений порядок завантаження ОС спочатку з гнучкого диска, а потім - з жорсткого, то за наявності гнучкого диска в накопичувачі буде зчитаний 1-й сектор з гнучкого диска. Якщо диск був заражений, то це досить для зараження ЕОМ. Така ситуація найчастіше має місце при перезавантаженні ОС після «зависань» або відмов ЕОМ.

При завантаженні комп'ютера з вінчестера першою бере на себе керування програма початкового завантаження в MBR **(Master Boot Record** - головний завантажувальний запис). Якщо жорсткий диск розбитий на кілька розділів, то лише один з них позначений як завантажувальний **(boot).** Програма початкового завантаження в **MBR** знаходить завантажувальний розділ вінчестера й передає керування на програму початкового завантаження цього розділу. Код останньої збігається з кодом програми початкового завантаження, що втримується на звичайних дисках, а відповідні завантажувальні сектори відрізняються тільки таблицями параметрів. Таким чином, на вінчестері є два об'єкти атаки завантажувальних вірусів - програма початкового завантаження в **MBR** і програма початкового завантаження в boot-секторі завантажувального диска.

Файлові віруси

До даної групи відносяться віруси, які при своєму розмноженні тим або іншим способом використовують файлову систему якоїсь операційної системи. Файлові віруси можуть впроваджуватися практично у всі виконувані файли всіх популярних операційних систем.

Є віруси, що заражають файли, які містять вихідні тексти програм, бібліотечні або об'єктні модулі. Можливий запис вірусу й у файли даних, але це трапляється або в результаті помилки у вірусі, або при прояві його агресивних властивостей.

За способом зараження файлів віруси діляться на

- overwriting, паразитичні (parasitic);
- компаньйони-віруси (companion);
- link-віруси;
- віруси, що заражають об'єктні модулі (OBJ), бібліотеки компіляторів (LIB) і вихідні тексти програм.

Схема роботи файлового вірусу. Отримавши управління, вірус вчиняє наступні дії:

- резидентний вірус перевіряє оперативну пам'ять на наявність своєї копії і інфікує пам'ять комп'ютера, якщо копія вірусу не знайдена; нерезидентний вірус шукає незаражені файли в поточному і (або) кореневому каталогах, в каталогах, зазначених командою PATH, сканує дерево каталогів логічних дисків, а потім заражає виявлені файли;
- виконує, якщо вони є, додаткові функції: деструктивні дії, графічні чи звукові ефекти і т.д. (Додаткові функції резидентного вірусу можуть викликатися через деякий час після активізації в залежності від поточного часу, конфігурації системи, внутрішніх лічильників вірусу або інших умов; в цьому випадку вірус при активізації обробляє стан системних годин, встановлює свої лічильники і т.д.)
- повертає управління основній програмі (якщо вона є). Паразитичні віруси при цьому або відновлюють програму (але не файл) в початковому вигляді, або лікують файл, виконують його, а потім знову заражають.

Приклади файлових вірусів:

Вірус VIENNA (Відень). Інші назви вірусу: 648, Restart (перезавантаження), Time Bomb (часова бомба) та ін. Один із перших найбільш примітивних вірусів. Знайдений спочатку у Відні, потім заполонив увесь світ. При завантаженні у пам'ять комп'ютера проглядає всі COM-програми у поточному каталозі та у доступних через PATH (шляхи пошуку, що звичайно встановлені в AUTOEXEC.BAT). Первісний варіант цього вірусу збільшував довжину жертви на 648 байт. Першу знайдену ще не заражену програму або заражає, або, з ймовірністю 1/8 (в залежності від системного часу), псує таким чином, що вона при запуску призводить до перезавантаження системи. В останньому випадку в початок жертви записується код EAF0FF00F0, який на машинній мові означає теплий рестарт (еквівалентне до дії клавіш Ctrl+Alt+Del). Якщо зіпсована таким чином програма викликається з AUTOEXEC.BAT, процедура початкового завантаження операційної системи зациклюється. Як ознаку зараження, вірус ставить у часі створення жертви неіснуюче число секунд (62). Надалі з'явилось багато різновидів вірусу VIENNA (більше 20), що відрізняються від нього довжинами та шкідливими діями.

Вірус CASCADE (Каскад, водоспад). Інші назви вірусу: LetterFall (буквопад), Letter та ін. Існує два варіанти вірусу за довжиною (1701 або 1704 байт). Заражає тільки COM-програми, резидентний. Спричиняє обсипання символів на екрані, що супроводжується характерним шелестінням. При цьому блокується можливість роботи з клавіатурою. Зберігає працездатність тільки на машинах типу PC XT/AT.

Вірус BLACK FRIDAY (Чорна п'ятниця). Інші назви вірусу: Israeli Virus (ізраїльський вірус), Ierusalem (Єрусалим), Black Hole (чорна діра) та ін. Вірус одержав вказані назви, оскільки вперше був виявлений в ізраїльському університеті та через свої характерні дії. Він заражає EXE- та COM-файли, збільшуючи їх розміри на 1813 байт, і залишається резидентним у пам'яті ПК. При цьому зараження може відбуватися неодноразово, що приводить до неймовірного розростання заражених

файлів. Інфікований даним вірусом ПК сповільнює свою роботу в декілька тисяч разів. При виведенні інформації на дисплей у нижньому лівому куті екрана з'являється чорний прямокутник (дірка). Нарешті, якщо час роботи приходиться на п'ятницю 13-го числа, то заражені файли знищуються. Характерною ознакою вірусу є наявність в його тілі сполучень MsDos а також COMMAND.COM.

Вірус DARK AVENGER (Чорний месник). Інші назви вірусу: Eddie, Sofia. Вірус одержав свої назви по текстовому рядку "Eddie lives . somewhere in time. This program was written in the sity of Sofia (C) 1988-89 Dark avenger", що міститься у його тілі. Вірус заражає EXE- та COM-файли, є резидентним, його довжина в байтах 1800. Вірус дуже небезпечний, оскільки на інфікованому комп'ютері файли заражаються не тільки при виконанні, але і під час їх проглядання та копіювання. Він також знищує COM-файли, довжина яких лежить у межах від 64K1800байт до 64K. Періодично знищує інформацію в одному із секторів вінчестера.

Завантажувально-Файлові віруси

Типовим представником таких вірусів є «популярний» завантажувально-файловий вірус **One_Half,** що заражає головний завантажувальний сектор (MBR) і файли, що виконуються. Основна руйнівна дія - шифрування секторів вінчестера. При кожному запуску вірус шифрує чергову порцію секторів, а, зашифрувавши половину жорсткого диска, радісно сповіщає про це.

One_Half . Інші назви вірусу: Free Love, One_half, One Half.3544. One_Half це файлово-бутовий, резидентний, поліморфний вірус, який використовує Stealth-технологію. Заражає COM- та EXE-файли, збільшуючи їх довжину на 3544 байт та Master Boot запис жорсткого диска. Під час холодного (пере)завантаження системи з інфікованого жорсткого диска One_Half зашифровує два циліндра у кінці жорсткого диска. При кожному наступному (пере)завантаженні системи кількість зашифрованих циліндрів зростає. Поки вірус знаходиться у пам'яті, інформація, що міститься у цих циліндрах, доступна. Коли вірус зашифрує приблизно половину жорсткого диска, він виводить на екран повідомлення "This is one half. Press any key to continue.". Шифрування інформації, що проводить One_Half, значно ускладнює роботу антивірусних програм, яким треба не тільки вилікувати комп'ютер, але і відновити зашифровану інформацію. One_Half не заражає деякі антивірусні програми (SCAN, CLEAN, FINDVIRU, GUARD, NOD, VSAFE, MSAV).

Основна проблема при лікуванні даного вірусу полягає в тому, що недостатньо просто видалити вірус із MBR і файлів, треба розшифрувати зашифровану ним інформацію. Найбільш радикальна дія – просто переписати новий здоровий MBR.

Поліморфні віруси

Більшість питань пов'язані з терміном «поліморфний вірус». Цей вид комп'ютерних вірусів є на сьогоднішній день найнебезпечнішим. Поліморфні віруси - віруси, що модифікують свій код у заражених програмах таким чином, що два екземпляри того самого вірусу можуть не збігатися в жодному біті. Такі віруси не

тільки шифрують свій код, використовуючи різні шляхи шифрування, але й містять код генерації шифрувальника й розшифровувача, що відрізняє їх від звичайних шифрувальних вірусів, які також можуть шифрувати ділянки свого коду, але мають при цьому постійний код шифрувальника й розшифровувача.

Поліморфні віруси – це віруси з розшифровувачами, що самомодифікуються. Ціль такого шифрування: захист від розшифровки. Маючи заражений і оригінальний файли, буде однаково неможливо проаналізувати його код. Цей код зашифрований і являє собою безглуздий набір команд. Розшифровка виробляється самим вірусом уже безпосередньо під час виконання. При цьому можливі варіанти: він може розшифрувати себе всього відразу, а може виконати таку розшифровку «по ходу справи», може знову шифрувати вже ділянки, що вже пропрацьовані. Все це робиться для ускладнення аналізу коду вірусу.

Яскравий приклад - Tchechen 1912, 1914. Ці віруси не входять до десятки найбільш поширених у світі, але на території СНД, мабуть, стоять чи не найпершому місці. Дуже небезпечні резидентні поліморфні віруси. При старті віруси зчитують 2-й сектор жорсткого диска і записують у нього слово "МИР" та число 4, яке надалі буде лічильником стартів інфікованих програм. Потім намагаються знайти в ROM BIOS текстові рядки Megatrends, AWARD. Якщо цей пошук успішний, то віруси вимикають у CMOS-пам'яті опцію Virus Warning on Boot (контроль запису до Boot-сектору). При досягненні лічильника у 2-у секторі значення 0 віруси замінюють слово "МИР" на непотрібне слово із 3 літер та записують в MBR жорсткого диска "троянський" код. Цей код при завантаженні системи самостійно віддає управління активному Boot-сектору жорсткого диска, але приблизно через місяць після запису даного коду в MBR знищує вміст всього першого жорсткого диска. Після чого планувалося виведення на екран тексту. Tchechen.1914 непрацездатний на процесорі Pentium

Контрольні питання:

1. Дайте визначення поняття комп’ютерних вірусів
2. Охарактеризуйте властивості комп’ютерних вірусів
3. Класифікуйте комп’ютерні віруси за критеріями
4. Опишіть види комп’ютерних вірусів
5. Опишіть схеми функціонування комп’ютерних вірусів.

Лекція № 16 (2 години)

Тема: Підвиди хробаків та троянських програм.

Мета: Ознайомитися з підвидами хробаків, троянських програм, їх схемами функціонування

ПЛАН

1. Поняття вірусу «троян»

2. Робота вірусу у системі
3. Поняття вірусу «хробак», підвиди
4. Робота вірусу у системі

Троян (троянський кінь) — тип шкідливих програм, основною метою яких є шкідливий вплив стосовно комп'ютерної системи. Трояни відрізняються відсутністю механізму створення власних копій. Деякі трояни здатні до автономного подолання систем захисту КС, з метою проникнення й зараження системи. У загальному випадку, троян попадає в систему разом з вірусом або хробаком, у результаті необачних дій користувача або ж активних дій зловмисника.

Життєвий цикл

У силу відсутності в троянів функцій розмноження й поширення, їхній життєвий цикл украй короткий - усього три стадії: • Проникнення на комп'ютер • Активація • Виконання закладених функцій. Це, саме собою, не означає малого часу життя троянів. Навпроти, троян може тривалий час непомітно перебувати в пам'яті комп'ютера, ніяк не видаючи своєї присутності, доти, поки не буде виявлений антивірусними засобами.

Способи проникнення. Завдання проникнення на комп'ютер користувача трояни вирішують звичайно одним із двох наступних методів.

1.Маскування — троян видає себе за корисний додаток, що користувач самостійно завантажує з Інтернет і запускає. Іноді користувач виключається із цього процесу за рахунок розміщення на web-сторінці спеціального скрипта, що використовуючи діри в браузері автоматично ініціює завантаження й запуск трояна. Приклад. Trojan.SymbOS.Hobble.a є архівом для операційної системи Symbian (SIS-архівом). При цьому він маскується під антивірус Symantec і має ім'я Symantec.sis. Після запуску на смартфоні троян підмінює оригінальний файл оболонки FExplorer.app на ушкоджений файл. У результаті при наступному завантаженні операційної системи більшість функцій смартфона виявляються недоступними.

Одним з варіантів маскування може бути також впровадження зловмисником троянського коду в код іншого додатка. У цьому випадку розпізнати троян ще складніше, тому що заражений додаток може відкрито виконувати які-небудь корисні дії, але при цьому тайкома завдавати шкоди за рахунок троянських функцій. Розповсюджений також спосіб впровадження троянів на комп'ютери користувачів через веб-сайти. При цьому використається або шкідливий скріпт, що завантажує й запускає троянську програму на комп'ютері користувача, використовуючи уразливість у веб-браузері, або методи соціальної інженерії - наповнення й оформлення веб-сайту провокує користувача до самостійного завантаження трояна. При такому методі впровадження може використатися не одна копія трояна, а поліморфний генератор, що створює нову копію при кожнім завантаженні. Застосовувані в таких генераторах

технології поліморфізму звичайно не відрізняються від вірусних поліморфних технологій.

2.Кооперація з вірусами й хробаками — троян подорожує разом із хробаками або, рідше, з вірусами. У принципі, такі пари хробак-троян можна розглядати цілком як складеного хробака, але в сформованій практиці прийнято троянську складову хробаків, якщо вона реалізована окремим файлом, уважати незалежним трояном із власним ім'ям. Крім того, троянська складова може попадати на комп'ютер пізніше, ніж файл хробака.

Приклад. Використовуючи backdoor-функціонала хробаків сімейства Bagle, автор хробака проводив сховану інсталяцію трояна SpamTool.Win32.Small.b, що збирав і відсилав на певну адресу адреси електронної пошти, що були у файлах на зараженому комп'ютері.

Нерідко спостерігається кооперація хробаків з вірусами, коли хробак забезпечує транспортування вірусу між комп'ютерами, а вірус поширюється по комп'ютері, заражаючи файли.

Приклад. Відомий у минулому хробак Email-Worm.Win32.Klez.h при зараженні комп'ютера також запускав на ньому вірус Virus.Win32.Elkern.c. Навіщо це було зроблено, сказати важко, оскільки вірус сам по собі, крім зараження й пов'язаних з помилками в коді шкідливих проявів (явно виражених шкідливих процедур у ньому немає), ніяких дій не виконує, тобто не є "посиленням" хробака в якому б те не було змісті.

На відміну від вірусів і хробаків, розподіл яких на типи виробляється по способах розмноження/поширення, трояни діляться на типи по характері виконуваних ними шкідливих дій. Найпоширеніші наступні види троянів:

- Клавіатурні шпигуни - трояни, що постійно перебувають у пам'яті й дані, що зберігають всі, вступники від клавіатури з метою наступної передачі цих даних зловмисникові. Звичайно в такий спосіб зловмисник намагається довідатися паролі або іншу конфіденційну інформацію. Сучасні програми-шпигуни оптимізиовані для збору інформації, переданої користувачем в Інтернет, оскільки серед цих даних можуть зустрічатися логіни й паролі до банківських рахунків, PIN-коди кредитних карт й інша конфіденційна інформація, що ставиться до фінансової діяльності користувача. Trojan-Spy.Win32.Agent.fa відслідковує відкриті вікна Internet Explorer і зберігає інформацію з відвідуваних користувачем сайтів, уведення клавіатури в спеціально створений файл servms.dll із системному каталозі Windows.
- Викрадачі паролів - трояни, також призначені для одержання паролів, але без спостереження за клавіатурою. У таких троянах реалізовані способи добування паролів з файлів, у яких ці паролі зберігаються різними додатками
- Утиліти вилученого керування - трояни, що забезпечують повний вилучений контроль над комп'ютером користувача. Найбільш популярна утиліта вилученого керування - Back Orifice.

• Люки (backdoor) - трояни які надають зловмисникові обмежений контроль над комп'ютером користувача. Від утиліт вилученого керування відрізняються більше простим пристроєм і, як наслідок, невеликою кількістю доступних дій. Проте, зазвичай однііїю з дій є можливість завантаження і запуску будь-яких файлів по команді зловмисника, що дозволяє при необхідності перетворити обмежений контроль у повний. Останнім часом backdoor-функціонал став характерною рисою хробаків.

• Анонімні smtp-сервера й проксі - трояни, що виконують функції поштових серверів або проксі й, що використаються в першому випадку для спам-розсилань, а в другому для замітання слідів хакерами.

• Утиліти дозвону - порівняно новий тип троянів, що представляє собою утиліти dial-up доступу в Інтернет через дорогі поштові служби. Такі трояни прописуються в системі як утиліти дозвону за замовчуванням і спричиняють величезні рахунки за користування Інтернетом.

• Модифікатори настроювань браузера - трояни, які міняють стартову сторінку в браузері, сторінку пошуку або ще які-небудь налаштування, відкривають додаткові вікна браузера, імітують натискання на банери і т.п.

• Логічні бомби - частіше не стільки трояни, скільки суміш хробаків і вірусів, суть роботи яких полягає в тому, щоб за певних умов (дата, час доби, дії користувача, команда ззовні) зробити певну дію: наприклад, знищення даних.

Хробак (мережний хробак) — тип шкідливих програм, що поширюються по мережних каналах, здатних до автономного подолання систем захисту автоматизованих і комп'ютерних мереж, а також до створення й подальшого поширення своїх копій, що не завжди збігаються з оригіналом, і здійсненню іншого шкідливого впливу. Життєвий цикл

Так само як для вірусів, життєвий цикл хробаків можна розділити на певні стадії: 1. Проникнення в систему; 2. Активація; 3. Пошук "жертв"; 4. Підготовка копій; 5. Поширення копій.

Стадії 1 й 5, загалом кажучи, симетричні й характеризуються в першу чергу використовуваними протоколами й додатками.

Стадія 4 - Підготовка копій - практично нічим не відрізняється від аналогічної стадії в процесі розмноження вірусів. Сказане про підготовку копій вірусів без змін застосовно й до чирв.

Канали поширення

На етапі проникнення в систему хробаки діляться переважно по типах використовуваних протоколів:

• Мережні хробаки - хробаки, що використовують для поширення протоколи Інтернет і локальні мережі. Зазвичай цей тип хробаків поширюється з використанням неправильної обробки деякими додатками базових пакетів стека протоколів tcp/ip;

- Поштові хробаки - хробаки, що поширюються у форматі повідомлень електронної пошти;
- IRC-хробаки - хробаки, що поширюються по каналах IRC (Internet Relay Chat);
- P2P-хробаки - чирви, що поширюються за допомогою пірінгових (peer-to-peer) файлообмінних мереж;
- IM-хробаки - хробаки, що використають для поширення системи миттєвого обміну повідомленнями (IM, Instant Messenger - ICQ, MSN Messenger, AIM й ін.);

Приклади. Класичними мережними хробаками є представники сімейства Net-Worm.Win32.Sasser. Ці хробаки використають уразливість у службі LSASS Microsoft Windows. При розмноженні, хробак запускає FTP-службу на TCP-порту 5554, після чого вибирає IP-адресу для атаки й відсилає запит на порт 445 по цій адресі, перевіряючи, чи запущена служба LSASS. Якщо атакується комп'ютер, що відповідає на запит, хробак посилає на цей же порт експлойт уразливості в службі LSASS, у результаті успішного виконання якого на вилученому комп'ютері запускається командна оболонка на TCP-порту 9996. Через цю оболонку хробак віддалено виконує завантаження копії хробака по протоколі FTP із запущеного раніше сервера й віддалено ж запускає себе, завершуючи процес проникнення й активації.

Контрольні питання:

1. Охарактеризуйте поняття вірусу «троян»
2. Розкажіть про роботу вірусу у системі
3. Дайте визначення поняття вірусу «хробак» та перелічить його підвиди
4. Розкажіть про роботу «хробаків» у системі

Лекція № 17 (2 години)

Тема: Шляхи проникнення вірусів у ПК.

Мета: Ознайомитися зі шляхами проникнення вірусів на носії даних та персональні комп’ютери

ПЛАН

1. Шляхи проникнення вірусів
2. Ознаки прояву вірусів
3. Заходи щодо уникнення та профілактики вірусів

Основними шляхами проникнення вірусів у комп'ютер є знімні диски (гнучкі й лазерні), а також комп'ютерні мережі. Зараження жорсткого диска вірусами може відбутися при завантаженні програми зі з’ємного пристрою, що містить вірус. Таке зараження може бути й випадковим, наприклад, якщо з’ємний пристрій не вийняли з входу і перезавантажили комп'ютер, при цьому з’ємний пристрій може бути й не системним. Заразити з’ємний пристрій набагато простіше. На нього вірус може потрапити, навіть якщо його просто приєднали до зараженого комп'ютера і, наприклад, прочитали його зміст.

Вірус, як правило, впроваджується в робочу програму таким чином, щоб при її запуску керування спочатку передалося йому й тільки після виконання всіх його команд знову повернулося до робочої програми. Одержавши доступ до керування, вірус насамперед переписує сам себе в іншу робочу програму й заражає її. Після запуску програми, що містить вірус, стає можливим зараження інших файлів. Начастіше вірусом заражається завантажувальний сектор диска і виконуються файли, що мають розширення EXE, COM, SYS, BAT. Украй рідко заражаються текстові файли.

Після зараження програми вірус може виконати яку-небудь диверсію, не занадто серйозну, щоб не привернути уваги і, нарешті, не забуває повернути керування тій програмі, з якої був запущений. Кожне виконання зараженої програми переносить вірус у наступну. Таким чином, заразиться все програмне забезпечення.

При зараженні комп'ютера вірусом важливо його виявити. Для цього треба знать про основні ознаки прояву вірусів. До них можна віднести наступні:

- припинення роботи або неправильна робота раніше успішно функціонуючих програм;
- повільна робота комп'ютера;
- неможливість завантаження операційної системи;
- зникнення файлів і каталогів або спотворення їх вмісту;
- зміна дат і часу модифікації файлів;
- зміна розмірів файлів;
- несподіване значне збільшення кількості файлів на диску;

- істотне зменшення розміру вільної оперативної пам'яті;
- вивід на екран непередбачених повідомлень або зображень;
- подача непередбачених звукових сигналів;
- часті зависання і збої в роботі комп'ютера .

Слід зазначити, що перераховані вище явища необов'язково викликаються присутністю вірусу, а можуть бути наслідком інших причин. Тому завжди утруднена правильна діагностика стану комп'ютера.

Виявлення вірусів і заходи щодо захисту і профілактики

Отже, якийсь «вірусописатель» створює вірус і запускає його в «життя». Якийсь час він, можливо, погуляє досхочу, але рано або пізно хтось запідозрить що-небудь негарне. Як правило, віруси виявляють звичайні користувачі, які зауважують ті або інші аномалії в поводженні комп'ютера. Вони в більшості випадків не здатні самостійно впоратися із заразою, але цього від них і не потрібно. Необхідно лише, щоб якомога швидше вірус потрапив у руки фахівців. Професіонали будуть його вивчати, з'ясовувати, «що він робить», «як він робить», «коли він робить» і т.п. У процесі такої роботи збирається вся необхідна інформація про даний вірус, зокрема, виділяється *сигнатура вірусу* - послідовність байтів, що цілком виразно його характеризує. Для побудови сигнатури звичайно беруться найбільш важливі й характерні ділянки коду вірусу. Одночасно стають ясні механізми роботи вірусу, наприклад, у випадку завантажувального вірусу важливо знати, де він ховає свій хвіст, де перебуває оригінальний завантажувальний сектор, а у випадку файлового - спосіб зараження файлу. Отримана інформація дозволяє з'ясувати:

– як виявити вірус, для цього уточнюються методи пошуку сигнатур у потенційних об'єктах вірусної атаки - файлах і/або завантажувальних секторах;

– як знешкодити вірус, якщо це можливо, розробляються алгоритми видалення вірусного коду з уражених об'єктів.

Для виявлення, видалення вірусів і захисту від них розроблено кілька видів спеціальних програм, які дозволяють виявляти і знищувати віруси. Такі програми називаються антивірусними. Розрізняють наступні види антивірусних програм:

- програми-детектори;
- програми-доктори, або фаги;
- програми-ревізори;
- програми-фільтри;
- програми-вакцини, або іммунізатори.

Програми-Детектори здійснюють пошук характерної для конкретного вірусу сигнатури в оперативній пам'яті веб-файлах і при виявленні видають відповідне повідомлення. Недоліком таких антивірусних програм є те, що вони можуть знаходити тільки ті віруси, які відомі розроблювачам таких програм.

Програми-Доктори, або фаги, а також програми-вакцини не тільки знаходять заражені вірусами файли, але й «лікують» їх, тобто видаляють із файлу тіло програми-

вірусу, повертаючи файли у вихідний стан. На початку своєї роботи фаги шукають віруси в оперативній пам'яті, знищуючи їх, і тільки потім переходять до «лікування» файлів. Серед фагів виділяють поліфаги, тобто програми-доктори, призначені для пошуку й знищення великої кількості вірусів. Найбільш відомі з них: Aidstest, Scan, Norton AntiVirns, Doctor Web.

З огляду на те, що постійно з'являються нові віруси, програми-детектори і програми-доктори швидко застарівають, і потрібне регулярне відновлення версій.

Програми-Ревізори відносяться до найнадійніших засобів захисту від вірусів. Ревізори запам'ятовують вихідний стан програм, каталогів і системних областей диска тоді, коли комп'ютер не заражений вірусом, а потім періодично або за бажанням користувача порівнюють поточний стан з вихідним. Виявлені зміни виводяться на екран монітора. Як правило, порівняння станів роблять відразу після завантаження операційної системи. При порівнянні перевіряються довжина файлу, код циклічного контролю (контрольна сума файлу), дата й час модифікації, інші параметри. Програми-Ревізори мають досить розвинені алгоритми, виявляють стелс-віруси й можуть навіть очистити програму, що перевіряється, від змін, внесених вірусом. До числа програм-ревізорів відноситься широко розповсюджена програма ADinf.

Програми-Фільтри, або *«сторожі»* являють собою невеликі резидентні програми, призначені для виявлення підозрілих дій при роботі комп'ютера, характерних для вірусів. Такими діями можуть бути:

- спроби корекції файлів з розширеннями COM, EXE;
- зміна атрибутів файлу;
- прямий запис на диск по абсолютній адресі;
- запис у завантажувальні сектори диска;
- завантаження резидентної програми.

При спробі будь-якої програми зробити зазначені дії «сторож» посилає користувачеві повідомлення і пропонує заборонити або дозволити відповідну дію. Програми-Фільтри досить корисні, тому що здатні виявити вірус на самій ранній стадії його існування до розмноження. Однак вони не «лікують» файли й диски. Для знищення вірусів потрібно застосувати інші програми, наприклад фаги. До недоліків програм-сторожів можна віднести їх «настирливість» (наприклад, вони постійно видають попередження про будь-яку спробу копіювання файлу, що виконується), а також можливі конфлікти з іншим програмним забезпеченням. Прикладом програми-фільтра є програма Vsafe.

Вакцини, або *іммунізатори* - це резидентні програми, що запобігають зараження файлів. Вакцини застосовують, якщо відсутні програми-доктори, «лікуючі» цей вірус. Вакцинація можлива тільки від відомих вірусів. Вакцина модифікує програму або диск таким чином, щоб це не відбивалося на їхній роботі, а вірус буде сприймати їх зараженими й тому не впровадиться. У цей час програми-вакцини мають обмежене застосування.

Своєчасне виявлення заражених вірусами файлів і дисків, повне знищення виявлених вірусів на кожному комп'ютері дозволяють уникнути поширення вірусної епідемії на інші комп'ютери.

Контрольні питання:

1. Опишіть шляхи проникнення вірусів
2. Пелелічить ознаки прояву вірусів

Лекція № 18 (2 години)

Тема: Компю’терні злочини та засоби захисту інформації у законодавстві.

Мета: Ознайомитися з поняттям комп’ютерних злочинів та засобами захисту інформації від них, які надає законодавство України

ПЛАН

1. Статті Уголовного Кодексу України

Р о з д і л XVI

ЗЛОЧИНИ У СФЕРІ ВИКОРИСТАННЯ ЕЛЕКТРОННО-ОБЧИСЛЮВАЛЬНИХ МАШИН (КОМП'ЮТЕРІВ), СИСТЕМ ТА КОМП'ЮТЕРНИХ МЕРЕЖ І МЕРЕЖ ЕЛЕКТРОЗВ'ЯЗКУ

(Назва розділу XVI із змінами, внесеними згідно із Законом N 908-IУ (908-15) від 05.06.2003)

Чинний КК, а саме розділ XVI “Злочини у сфері використання електронно-обчислювальних машин (комп’ютерів), систем та комп’ютерних мереж і мереж електрозв’язку” Особливої частини КК містить шість статей:

1. несанкціоноване втручання в роботу електронно-обчислювальних машин (комп’ютерів), автоматизованих систем, комп’ютерних мереж чи мереж електрозв’язку (ст. 361 КК);

2. створення з метою використання, розповсюдження або збуту шкідливих програмних чи технічних засобів, а також їх розповсюдження або збут (ст. 361^1 КК);

3. несанкціоновані збут або розповсюдження інформації з обмеженим доступом, яка зберігається в електронно-обчислювальних машинах (комп.ютерах), автоматизованих системах, комп’ютерних мережах або на носіях такої інформації (ст. 361^2 КК);

4. несанкціоновані дії з інформацією, яка оброблюється в електронно-обчислювальних машинах (комп’ютерах), автоматизованих системах, комп’ютерних мережах або зберігається на носіях такої інформації, вчинені особою, яка має право доступу до неї (ст. 362);

5. порушення правил експлуатації електронно-обчислювальних машин (комп’ютерів), автоматизованих систем, комп’ютерних мереж чи мереж електрозв’язку

або порядку чи правил захисту інформації, яка в них оброблюється (ст. 363 КК);

6. перешкоджання роботі електронно-обчислювальних машин (комп'ютерів), автоматизованих систем, комп'ютерних мереж чи мереж електрозв'язку шляхом масового розповсюдження повідомлень електрозв'язку (ст. 363[1] КК).

Стаття 361. Незаконне втручання в роботу електронно-обчислювальних машин (комп'ютерів), систем та комп'ютерних мереж і мереж електрозв'язку

1. Незаконне втручання в роботу автоматизованих електронно-обчислювальних машин, їх систем чи комп'ютерних мереж, що призвело до перекручення чи знищення комп'ютерної інформації або носіїв такої інформації, а також розповсюдження комп'ютерного вірусу шляхом застосування програмних і технічних засобів, призначених для незаконного проникнення в ці машини, системи чи комп'ютерні мережі і здатних спричинити перекручення або знищення комп'ютерної інформації чи носіїв такої інформації, а так само незаконне втручання в роботу мереж електрозв'язку, що призвело до знищення, перекручення, блокування інформації або до порушення встановленого порядку її маршрутизації, - караються штрафом до сімдесяти неоподатковуваних мінімумів доходів громадян або виправними роботами на строк до двох років, або обмеженням волі на той самий строк.

2. Ті самі дії, якщо вони заподіяли істотну шкоду або вчинені повторно чи за попередньою змовою групою осіб, - караються штрафом від ста до чотирьохсот неоподатковуваних мінімумів доходів громадян або обмеженням волі на строк до п'яти років, або позбавленням волі на строк від трьох до п'яти років.

Примітка. Під істотною шкодою, якщо вона полягає в завданні матеріальних збитків, слід розуміти таку шкоду, яка в триста і більше разів перевищує неоподатковуваний мінімум доходів громадян.

(Стаття 361 в редакції Закону N 908-IУ (908-15) від 05.06.2003)

Стаття 362. Викрадення, привласнення, вимагання комп'ютерної інформації або заволодіння нею шляхом шахрайства чи зловживання службовим становищем

1. Викрадення, привласнення, вимагання комп'ютерної інформації або заволодіння нею шляхом шахрайства чи зловживання службовою особою своїм службовим становищем - караються штрафом від п'ятдесяти до двохсот неоподатковуваних мінімумів доходів громадян або виправними роботами на строк до двох років.

2. Ті самі дії, вчинені повторно або за попередньою змовою групою осіб, - караються штрафом від ста до чотирьохсот неоподатковуваних мінімумів доходів громадян або обмеженням волі на строк до трьох років, або позбавленням волі на той самий строк.

3. Дії, передбачені частинами першою або другою цієї статті, якщо вони заподіяли істотну шкоду, - караються позбавленням волі на строк від двох до п'яти років.

Стаття 363. Порушення правил експлуатації автоматизованих електронно-обчислювальних систем

1. Порушення правил експлуатації автоматизованих електронно-обчислювальних машин, їх систем чи комп'ютерних мереж особою, яка відповідає за їх експлуатацію, якщо це спричинило викрадення, перекручення чи знищення комп'ютерної інформації, засобів її захисту, або незаконне копіювання комп'ютерної інформації, або істотне порушення роботи таких машин, їх систем чи комп'ютерних мереж, - карається штрафом до п'ятдесяти неоподатковуваних мінімумів доходів громадян або позбавленням права обіймати певні посади чи займатися певною діяльністю на строк до п'яти років, або виправними роботами на строк до двох років.

2. Те саме діяння, якщо воно заподіяло істотну шкоду, - карається штрафом до ста неоподатковуваних мінімумів доходів громадян або виправними роботами на строк до двох років, або обмеженням волі на строк до п'яти років, з позбавленням права обіймати певні посади чи займатися певною діяльністю на строк до трьох років або без такого.

Контрольні питання:

1. Ознайомтеся з відповідальністю за комп’ютерні злочини