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

ЗАТВЕРДЖУЮ
Заступник директора
з НМР ФКПАІТ ОНТУ
_____/Вікторія ОКСАНІЧЕНКО/
“___”_____2022 року

МЕТОДИЧНІ ВКАЗІВКИ
ДЛЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ
з дисципліни
ТЕХНОЛОГІЇ ЗАХИСТУ ІНФОРМАЦІЇ

для спеціальності 122 «Комп’ютерні науки»

Одеса
2022 рік

Методичні вказівки до самостійної роботи студентів розроблено викладачем Бурмакіною Ю. В. згідно з навчальною програмою дисципліни «Технології захисту інформації».

Схвалено цикловою комісією «Комп'ютерних наук та інженерії програмного забезпечення»

Протокол № « _____» від «___»_____20__ __ р.

Голова циклової комісії ___ _____ Тетяна КОСТИРЕНКО
(підпис) (власне ім'я та прізвище)

Тема № 1 (2 години)

Інформаційна безпека комп'ютерних систем. Основні поняття і визначення.

Мета: навчитись поняттям безпеки комп’ютерних систем, знати основні поняття та визначення інформаційної безпеки

План

1. Поняття безпеки та погроз
2. Основні поняття і визначення типів погроз

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Інформатизація є характерною межею життя сучасного суспільства. Нові інформаційні технології активно упроваджуються у всі сфери народного господарства. Комп'ютери управляють космічними кораблями й літаками, контролюють роботами атомних електростанцій, розподіляють електроенергію й обслуговують банківські системи. Комп'ютери є основою безлічі автоматизованих систем обробки інформації (АСОІ), здійснюючe зберігання й обробку інформації, представленій її споживачам, реалізовуючи тим самим сучасні інформаційні технології.

У міру розвитку й ускладнення засобів, методів і форм автоматизації процесів обробки інформації підвищується залежність суспільства від ступеня безпеки інформаційних технологій, що використовуються ними, від яких деколи залежить благополуччя, а іноді й життя багатьох людей.

Актуальність і важливість проблеми забезпечення безпеки інформаційних технологій обумовлені наступними причинами:

- різке збільшення обчислювальної потужності сучасних комп'ютерів при одночасному спрощенні їх експлуатації;
- різке збільшення об'ємів інформації, що накопичується, зберігається і оброблюється за допомогою комп'ютерів та інших засобів автоматизації;
- зосередження в єдиних базах даних інформації різного призначення й різної приналежності;
- високі темпи зростання парку персональних комп'ютерів, що перебувають в експлуатації в самих різних сферах діяльності;
- різке розширення кола користувачів, що мають безпосередній доступ до обчислювальних ресурсів і масивів даних;
- бурхливий розвиток програмних засобів, не задовольняючих навіть мінімальним вимогам безпеки;
- повсюдне розповсюдження мережевих технологій й об'єднання локальних мереж у глобальних;
- розвиток глобальної мережі Internet, що практично не перешкоджають

порушенням безпеки систем обробки інформації у всьому світі.

Введемо й визначимо основні поняття інформаційної безпеки комп'ютерних систем .

Під безпекою АСОІ розуміють її захищеність від випадкового або навмисного втручання в нормальний процес її функціонування, а також від спроб розкрадання, зміни або руйнування її компонентів.

Природа дій на АСОІ може бути самою різноманітною. Це і стихійні лиха (землетрус, ураган, пожежа), і вихід з ладу складових елементів АСОІ, і помилки персоналу, і спроба проникнення зловмисника.

Безпека АСОІ досягається вживанням заходів по забезпеченню конфіденційності й цілісності оброблюваної нею інформації, а також доступності й цілісності компонентів і ресурсів системи.

Під доступом до інформації розуміється ознайомлення з інформацією, її обробка, зокрема копіювання, модифікація або знищення інформації.

Розрізняють санкціонований і несанкціонований доступ до інформації.

Санкціонований доступ до інформації - це доступ до інформації, що не порушує встановлені правила розмежування доступу.

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

Несанкціонований доступ до інформації характеризується порушенням встановлених правил розмежування доступу. Особа або процес, що здійснюють несанкціонований доступ до інформації, є порушниками правил розмежування доступу. Несанкціонований доступ є найбільш поширеним видом комп'ютерних порушень.

Конфіденційність даних - це статус, наданий даним, що визначає необхідний ступінь їх захисту. Власне кажучи конфіденційність інформації - це властивість інформації бути відомою тільки допущеним суб'єктам системи, що пройшли перевірку (авторизованим) (користувачам, процесам, програмам). Для решти суб'єктів системи ця інформація повинна бути невідомою.

Суб'єкт - це активний компонент системи, який може стати причиною потоку інформації від об'єкту до суб'єкта або зміни стану системи.

Об'єкт - пасивний компонент системи, що зберігає, приймає або передає інформацію. Доступ до об'єкту означає доступ до інформації, що міститься в ньому.

Цілісність інформації забезпечується в тому випадку, якщо дані в системі не відрізняються в семантичному відношенні від даних у початкових документах, тобто якщо не відбулося їх випадкового або навмисного спотворення або руйнування.

Цілісність компоненту або ресурсу системи - це властивість компоненту або ресурсу бути незмінними в семантичному сенсі при функціонуванні системи в умовах випадкових або навмисних спотворень або руйнуючих дій.

Доступність компоненту або ресурсу системи - це властивість компоненту або ресурсу бути доступним для авторизованих законних суб'єктів системи.

Під загрозою безпеки АСОІ розуміються можливі дії на АСОІ, які прямо або побічно можуть завдати збитку її безпеки. Збиток безпеки має на увазі порушення стану захищеності інформації, що міститься й обробляється в АСОІ. З поняттям загрози безпеці тісно пов'язано поняття вразливості АСОІ.

Вразливість АСОІ - це деяка невдала властивість системи, яка робить можливим виникнення й реалізацію загрози.

Атака на комп'ютерну систему - це дія, що здійснюється зловмисником, яка полягає в пошуку та використанні тієї або іншої вразливості системи. Таким чином, атака - це реалізація загрози безпеці.

Протидія погрозам безпеці є метою захисту систем обробки інформації

Безпечна або захищена система - це система із засобами захисту, які успішно й ефективно протистоять погрозам безпеці.

Комплексом засобів захисту є сукупність програмних і технічних засобів, що створюються і підтримуються для забезпечення інформаційної безпеки АСОІ. Комплекс створюється й підтримується відповідно до прийнятої в даній організації політики безпеки.

Політика безпеки - це сукупність норм, правил і практичних рекомендацій, що регламентують роботу засобів захисту АСОІ від заданої безлічі погроз безпеці.

Питання щодо самоконтролю:

1. Визначити поняття інформаційної безпеки системи.
2. Перелічити й розкрити завдання захисту користувальницької інформації.
3. Перелічити способи рішення завдання забезпечення безпеки в інформаційно-телекомунікаційних системах

Тема №2 (2 години)

Основні погрози безпеки АСОІ. Забезпечення безпеки АСОІ.

Мета: Ознайомитись з основними поняттями і визначеннями безпеки комп’ютерних систем та основами забезпечення безпеки АСОІ.

План

1. Основні типи погроз безпеки
2. Основні визначення типів погроз та їх класифікація

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Основні погрози безпеці АСОІ

По меті дії розрізняють три основні типи погроз безпеці АСОІ:

- погрози порушення конфіденційності інформації;
- погрози порушення цілісності інформації;
- погрози порушення працездатності системи (відмови в обслуговуванні).

Погрози порушення конфіденційності направлені на розголос конфіденційної або секретної інформації. При реалізації цих погроз інформація стає відомою особам, які не повинні мати до неї доступу. У термінах комп'ютерної безпеки загроза порушення конфіденційності має місце щораз, коли здійснено несанкціонований доступ до деякої закритої інформації, що зберігається в комп'ютерній системі або передається від однієї системи до іншої.

Погрози порушення цілісності інформації, що зберігається в комп'ютерній системі або передається по каналі зв'язку, направлені на її зміну або спотворення, що приводять до порушення її якості або повного знищення. Цілісність інформації може бути порушена умисне зловмисником, а також у результаті об'єктивних дій з боку середовища, що оточує систему. Ця загроза особливо актуальна для систем передачі інформації - комп'ютерних мереж і систем телекомунікацій. Умисні порушення цілісності інформації не слід плутати з її санкціонованою зміною, яка виконується повноважними особами з обгрунтованою метою (наприклад, такою зміною є періодична корекція деякої бази даних).

Погрози порушення працездатності (відмова в обслуговуванні) направлені на створення таких ситуацій, коли певні навмисні дії або знижують працездатність АСОІ, або блокують доступ до деяких її ресурсів. Наприклад, якщо один користувач системи запрошує доступ до деякої служби, а інший робить дії з блокування цього доступу, то перший користувач дістає відмову в обслуговуванні. Блокування доступу до ресурсу може бути постійним або тимчасовим.

Порушення конфіденційності й цілісності інформації, а також доступності й цілісності певних компонентів і ресурсів АСОІ можуть бути викликані різними небезпечними діями на АСОІ.

Сучасна автоматизована система обробки інформації є складною системою, що складається з великого числа компонентів різного ступеня автономності, які зв'язані між собою й обмінюються даними. Практично кожен компонент може піддатися зовнішній дії або вийти з ладу. Компоненти АСОІ можна розбити на наступні групи:

- апаратні засоби - ЕОМ і їх складові частини (процесори, монітори, термінали, периферійні пристрої, дисководи, принтери, контролери, кабелі, лінії зв'язку) і т.д.;
- програмне забезпечення - придбані програми, вихідні, об'єктні, завантажувальні модулі; операційні системи й системні програми (компілятори, компонувальники й ін.), утиліти, діагностичні програми й т.д.;
- дані - тимчасово і постійно збережені, на магнітних носіях, друкарські, архіви, системні журнали й т.д.;
- персонал - обслуговуючий персонал і користувачі.

Небезпечні дії на АСОІ можна підрозділити на випадкові й навмисні. Аналіз

досвіду проектування, виготовлення й експлуатації АСОІ показує, що інформація піддається різним випадковим діям на всіх етапах циклу життя й функціонування АСОІ. Причинами випадкових дій при експлуатації АСОІ можуть бути:

- аварійні ситуації через стихійні лиха і відключення електроживлення;
- відмови й збої апаратури;
- помилки в програмному забезпеченні;
- помилки в роботі обслуговуючого персоналу й користувачів;
- перешкоди в лініях зв'язку через дії зовнішнього середовища.

Навмисні погрози пов'язані з цілеспрямованими діями порушника.

Як порушник можуть виступати службовець, відвідувач, конкурент, найманець і так далі. Дії порушника можуть бути обумовлені різними мотивами: незадоволеністю службовця своєю кар'єрою, суто матеріальним інтересом (хабар), цікавістю, конкурентною боротьбою, прагненням самоствердитися за всяку ціну й тому подібне.

Виходячи з можливості виникнення найбільш небезпечної ситуації, обумовленої діями порушника, можна скласти гіпотетичну модель потенційного порушника:

- кваліфікація порушника може бути на рівні розробника даної системи;
- порушником може бути як стороння особа, так і законний користувач системи;
- порушникові відома інформація про принципи роботи системи;
- порушник вибере найбільш слабку ланку в захисті.

Зокрема, для банківських АСОІ можна виділити наступні навмисні погрози:

– несанкціонований доступ сторонніх осіб,що не належать до банківських службовців, і ознайомлення з конфіденційною інформацією, що зберігається;
– ознайомлення банківських службовців з інформацією, до якої вони не повинні мати доступу;
– несанкціоноване копіювання програм і даних;
– крадіжка магнітних носіїв, що містять конфіденційну інформацію;
– крадіжка роздрукованих банківських документів;
– умисне знищення інформації;
– несанкціонована модифікація банківськими службовцями фінансових документів, звітності й баз даних;
– фальсифікація повідомлень, що передаються по каналах зв'язку;
– відмова від авторства повідомлення, переданого по каналах зв'язку;
– відмова від факту отримання інформації;
– нав'язування раніше переданого повідомлення;
– руйнування інформації, викликане вірусними діями;
– руйнування архівної банківської інформації, що зберігається на магнітних носіях;
– крадіжка устаткування.

Несанкціонований доступ (НСД) є найбільш розповсюдженим і багатообразним видом комп'ютерних порушень. Суть НСД полягає в отриманні користувачем

(порушником) доступу до об'єкту, порушуючи правила розмежування доступу, встановлені відповідно до прийнятої в організації політики безпеки. НСД використовує будь-яку помилку в системі захисту і можливий при нераціональному виборі засобів захисту, їх некоректній установці і налаштуванні. НСД може бути здійснений як штатними засобами АСОІ, так і спеціально створеними апаратними і програмними засобами.

Перерахуємо основні канали несанкціонованого доступу, через які порушник може дістати доступ до компонентів АСОІ й здійснити розкрадання, модифікацію і/або руйнування інформації:

- всі штатні канали доступу до інформації (термінали користувачів, оператора, адміністратора системи; засоби відображення і документування інформації; канали зв'язку) при їх використанні порушниками, а також законними користувачами поза межами їх повноважень;
- технологічні пульти управління;
- лінії зв'язку між апаратними засобами АСОІ;
- побічні електромагнітні випромінювання від апаратури, ліній зв'язку, мереж електроживлення і заземлення та ін.

Із всієї різноманітності способів і прийомів несанкціонованого доступу зупинимося на наступних поширених і зв'язаних між собою порушеннях:

- перехоплення паролів;
- "маскарад";
- незаконне використання привілеїв.

Перехоплення паролів здійснюється спеціально розробленими програмами. При спробі законного користувача увійти до системи програма-перехоплювач імітує на екрані дисплея введення імені й пароля користувача, які відразу пересилаються власнику програми-перехоплювача, після чого на екран виводиться повідомлення про помилку і управління повертається операційній системі. Користувач припускає, що допустив помилку при введенні пароля. Він повторює введення й дістає доступ у систему. Власник програми-перехоплювача, що отримав ім'я й пароль законного користувача, може тепер використовувати їх у своїх цілях. Існують й інші способи перехоплення паролів.

"Маскарад"- це виконання яких-небудь дій одним користувачем від імені іншого користувача, що володіє відповідними повноваженнями. Метою "маскараду" є приписування яких-небудь дій іншому користувачеві або привласнення повноважень і привілеїв іншого користувача.

Прикладами реалізації "маскараду" є:

- вхід у систему під ім'ям і паролем іншого користувача (цьому "маскараду" передує перехоплення пароля);
- передача повідомлень у мережі від імені іншого користувача.

"Маскарад" особливо небезпечний у банківських системах електронних платежів, де неправильна ідентифікація клієнта через "маскарад" зловмисника може

привести до великих збитків законного клієнта банку.

Незаконне використання привілеїв. Більшість систем захисту встановлюють певні набори привілеїв для виконання заданих функцій. Кожен користувач отримує свій набір привілеїв: звичайні користувачі - мінімальний, адміністратори - максимальний. Несанкціоноване захоплення привілеїв, наприклад за допомогою "маскараду", приводить до можливості виконання порушником певних дій в обхід системи захисту. Слід зазначити, що незаконне захоплення привілеїв можливе або за наявності помилок у системі захисту, або через халатність адміністратора при управлінні системою і призначенні привілеїв.

Особливо слід зупинитися на погрозах, яким можуть піддаватися комп'ютерні мережі. Основна особливість будь-якої комп'ютерної мережі полягає в тому, що її компоненти розподілені в просторі. Зв'язок між вузлами (об'єктами) мережі здійснюється фізично за допомогою мережевих ліній зв'язку й програмно за допомогою механізму повідомлень. При цьому керівники повідомлення і дані мережі, що пересилаються між об'єктами, передаються у вигляді пакетів обміну. При вторгненні в комп'ютерну мережу зловмисник може використовувати як пасивні, так й активні методи вторгнення.

При пасивному вторгненні (перехопленні інформації) порушник тільки спостерігає за проходженням інформації по каналі зв'язку, не вторгаючись ні в інформаційний потік, ні в зміст інформації, що передається. Як правило, зловмисник може визначити пункти призначення та ідентифікатори або тільки факт проходження повідомлення, його довжину й частоту обміну, якщо вміст повідомлення не розпізнаваний, тобто виконати аналіз трафіку (потоку повідомлень) у даному каналі.

При активному вторгненні порушник прагне підмінити інформацію, що передається в повідомленні. Він може вибірково модифікувати, змінити або додати правильне або помилкове повідомлення, видалити, затримати або змінити порядок проходження повідомлень. Зловмисник може також анулювати і затримати всі повідомлення, що передаються по каналі. Подібні дії можна кваліфікувати як відмова в передачі повідомлень.

Комп'ютерні мережі характерні тим, що окрім звичайних локальних атак, здійснюваних у межах однієї системи, проти об'єктів мереж роблять так звані видалені атаки, що обумовлене розподілом мережевих ресурсів та інформації. Зловмисник може знаходитися за тисячі кілометрів від об'єкту, що атакується, при цьому нападу може піддаватися не тільки конкретний комп'ютер, але й інформація, що передається по мережевих каналах зв'язку. Під видаленою атакою розуміють інформаційну руйнуючу дію на розподілену комп'ютерну мережу, програмно здійснене по каналах зв'язку.

У табл. 1.1 показані основні шляхи реалізації погроз безпеки АСОІ при дії на її компоненти. Звичайно, табл. 1.1 дає найзагальнішу картину того, що може відбутися з системою. Конкретні обставини та особливості повинні розглядатися окремо.

Таблиця 1.1 Шляхи реалізації погроз безпеці АСОІ

Об'єкти впливу	Порушення конфіденційності інформації	Порушення цілісності інформації	Порушення працездатності системи
Апаратні засоби	нсд-підключення; використання ресурсів; крадіжка носіїв	нсд-підключення; використання ресурсів; модифікація, зміна режимів	нсд-зміна режимів; виведення з ладу; руйнування
Програмне забезпечення	нсд-копіювання; крадіжка; перехват	нсд. впровадження "троянского коня", "вірусів", "хробаків"	нсд-спотворення; видалення; підміна
Дані	нсд-копіювання; крадіжка; перехват	нсд-спотворення; модифікація	нсд-спотворення; видалення; підміна
Персонал	Розголос; передача відомостей про захист; халатність	"Маскарад"; вербовка; підкуп персонала	вихід з робочого місця; фізичне усунення

У табл. 1.1 приведені специфічні назви й терміни: "троянський кінь", "вірус", "черв'як", які вживаються для іменування деяких поширених погроз безпеці АСОІ. Хоча ці назви мають жаргонний відтінок, вони вже ввійшли в загальноприйнятий комп'ютерний лексикон. Дамо коротку характеристику цих поширених погроз безпеці АСОІ.

"Троянський кінь" є програмою, яка поряд з діями, описаними в її документації, виконує деякі інші дії, що ведуть до порушення безпеки системи й деструктивним результатам. Аналогія такої програми із старогрецьким "троянським конем" цілком виправдана, оскільки в обох випадках оболонка, що не викликає підозр, таїть серйозну загрозу.

Термін "троянський кінь" був вперше використаний хакером Даном Едварсом, пізніше співробітником Агентства Національної Безпеки США. "Троянський кінь" використовує в сутності обман, щоб спонукати користувача запустити програму з

прихованою усередині загрозою. Зазвичай для цього затверджується, що така програма виконує деякі вельми корисні функції. Зокрема, такі програми маскуються під які-небудь корисні утиліти.

Небезпека "троянського коня" полягає в додатковому блоці команд, вставленому в початкову нешкідливу програму, яка потім надається користувачам АСОІ. Цей блок команд може спрацьовувати при настанні якої-небудь умови (дати, стани системи) або по команді ззовні. Користувач, що запустив таку програму, піддає небезпеці як свої файли, так і всю АСОІ в цілому. Приведемо для прикладу деякі деструктивні функції, що реалізовуються "троянськими кіньми".

- Знищення інформації. Вибір об'єктів і способів знищення визначається фантазією автора шкідливої програми.
- Перехоплення й передача інформації. Зокрема, відома програма, що здійснює перехоплення паролів, що набирають на клавіатурі.
- Цілеспрямована модифікація тексту програми, що реалізує функції безпеки й захисту системи.

Загалом, "троянські коні" завдають збитку АСОІ за допомогою розкрадання інформації і явного руйнування програмного забезпечення системи. "Троянський кінь" є одній з найбільш небезпечних погроз безпеці АСОІ. Радикальний спосіб захисту від цієї загрози полягає в створенні замкнутого середовища виконання програм, які повинні зберігатися й захищатися від несанкціонованого доступу.

Комп'ютерним "вірусом" є своєрідним явищем, що виникло в процесі розвитку комп'ютерної та інформаційної техніки. Суть цього явища полягає в тому, що програми-віруси володіють рядом властивостей, властивих живимо організмам, - вони народжуються, розмножуються й вмирають.

Термін "вірус" у застосуванні до комп'ютерів був запропонований Фредом Коеном з Університету Південної Каліфорнії. Історично перше визначення, дане Ф. Коеном: "Комп'ютерний вірус - це програма, яка може заражати інші програми, модифікуючи їх за допомогою включення в них своєї, можливо, зміненої копії, причому остання зберігає здібність до подальшого розмноження". Ключовими поняттями у визначенні комп'ютерного вірусу є здатність вірусу до саморозмноження й здібності до модифікації обчислювального процесу. Вказані властивості комп'ютерного вірусу аналогічні паразитуванню в живій природі біологічного вірусу.

Вірус зазвичай розробляється зловмисниками таким чином, щоб якомога довше залишатися невиявленим у комп'ютерній системі. Початковий період "дрімоти" вірусів є механізмом їх виживання. Вірус виявляється повною мірою в конкретний момент часу, коли відбувається деяка подія виклику, наприклад п'ятниця 13-го, відома дата й тому подібне.

Комп'ютерний вірус намагається таємно записати себе на комп’ютерні диски. Спосіб функціонування більшості вірусів полягає в такій зміні системних файлів комп'ютера, щоб вірус починав свою діяльність при кожному завантаженні. Наприклад, віруси, що вражають завантажувальний сектор, намагаються інфікувати

частину з’ємного пристрою або жорсткого диска, зарезервовану тільки для операційної системи й зберігання файлів запуску. Ці віруси особливо підступні, оскільки вони завантажуються в пам'ять при кожному включенні комп'ютера. Такі віруси володіють найбільшою здібністю до розмноження й можуть постійно поширюватися на нові диски.

Інша група вірусів намагається інфікувати виконувані файли, щоб залишитися невиявленими. Зазвичай віруси віддають переваги EXE- або COM-файлам, що вживаються для виконання коду програми в комп'ютерній системі. Деякі віруси використовують для інфікування комп'ютерної системи як завантажувальний сектор, так і метод зараження файлів. Це ускладнює виявлення та ідентифікацію таких вірусів спеціальними програмами і веде до їх швидкого розповсюдження. Існують і інші різновиди вірусів. Комп'ютерні віруси завдають збитку системі за рахунок багатообразного розмноження й руйнування місця існування.

Мережевий "хробак" є різновидом програми-віруса, яка розповсюджується по глобальній мережі і не залишає своєї копії на магнітному носієві. Термін "хробак" прийшов з науково-фантастичного роману Джона Бруннера "По бурним хвилях". Цей термін використовується для іменування програм, які подібно до стрічкових хробаків переміщаються по комп'ютерній мережі від однієї системи до іншої.

Спочатку "хробаки" були розроблені для пошуку в мережі інших комп'ютерів з вільними ресурсами, щоб дістати можливість виконати розподілені обчислення. При правильному використанні технологія "хробаків" може бути вельми корисною. Наприклад, "хробак" World Wide Web Worm формує індекс пошуку ділянок Web. Проте "хробак" легко перетворюється на шкідливу програму. "Хробак" використовує механізми підтримки мережі для визначення вузла, який може бути уражений. Потім за допомогою цих же механізмів передає своє тіло в цей вузол і або активізується, або чекає умови, що є зручними, для активізації.

Найбільш відомим представником цього класу шкідливих програм є "хробак" Моріса, який представляв собою програму з 4000 рядків на мові С і вхідній мові командного інтерпретатора системи UNI. Студент Корнеллського Університета Роберт Таппан Морріс-молодший запустив 2 листопаду 1988 р. на комп'ютері Массачусетсського Технологічного Інституту свою програму-хробак. Використовуючи помилки в операційній системі UNI на комп'ютерах VAX й Sun, ця програма передавала свій код з машини на машину. Всього за 6 годин були уражені підключення до мережі Internet 6000 комп'ютерів ряду крупних університетів, інститутів і дослідницьких лабораторій США. Збиток від цієї акції склав багато мільйонів доларів.

Мережеві "хробаки" є найнебезпечнішим видом шкідливих програм, оскільки об'єктом їх атаки може стати будь-який з мільйонів комп'ютерів, підключених до глобальної мережі Internet. Для захисту від "хробака" необхідно прийняти заходи пересторoги проти несанкціонованого доступу до внутрішньої мережі. Слід відзначити, що "троянські коні", комп'ютерні віруси і мережеві "хробаки" відносяться

до найбільш небезпечних погроз АСОІ. Для захисту від вказаних шкідливих програм необхідно застосування ряду заходів:

- виключення несанкціонованого доступу до виконуваних файлів;
- тестування програмних засобів, що придбаються;
- контроль цілісності виконуваних файлів і системних областей;
- створення замкнутого середовища виконання програм.

Питання щодо самоконтролю:

1. Якими причинами обумовлені актуальність і важливість проблеми забезпечення безпеки інформаційних технологій?
2. Інформаційна безпека комп'ютерних систем, основні поняття й визначення. Основні погрози безпеки.
3. Програмні засоби впливу на АСОІ.

Тема №3 (2 години)
Забезпечення безпеки АСОІ

Мета: опанувати поняття та правила забезпечення безпеки АСОІ

План

1. Призначення АСОІ
2. Типові підходи до роботи з АСОІ
3. Система захисту АСОІ

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Основним призначенням АСОІ є перебирання (збір, зберігання, обробка й видача) інформації, тому проблема забезпечення інформаційної безпеки є для АСОІ центральною. Забезпечення безпеки АСОІ припускає організацію протидії будь-якому несанкціонованому вторгнення в процес функціонування АСОІ, а також спробам модифікації, розкрадання, виведення з ладу або руйнування її компонентів, тобто захист всіх компонентів АСОІ - апаратних засобів, програмного забезпечення, даних і персоналу.

Існують два підходи до проблеми забезпечення безпеки АСОІ: "фрагментарний" і комплексний.

"Фрагментарний" підхід чітко направлений на протидію певним погрозам у

заданих умовах. Як приклад реалізації такого підходу можна вказати окремі засоби управління доступом, автономні засоби шифрування, спеціалізовані антивірусні програми й тому подібне.

Перевагою такого підходу є висока вибірковість до конкретної загрози. Істотним недоліком даного підходу є відсутність єдиного захищеного середовища обробки інформації. Фрагментарні заходи захисту інформації забезпечують захист конкретних об'єктів АСОІ тільки від конкретної загрози. Навіть невелика видозміна загрози веде до втрати ефективності захисту.

Комплексний підхід орієнтований на створення захищеного середовища обробки інформації в АСОІ, що об’єднує в єдиний комплекс різнорідні заходи протидії погрозам. Організація захищеного середовища обробки інформації дозволяє гарантувати певний рівень безпеки АСОІ, що є безперечною перевагою комплексного підходу. До недоліків цього підходу відносяться: обмеження на волю дій користувачів АСОІ, велика чутливість до помилок установки і налаштування засобів захисту, складність управління.

Комплексний підхід застосовують для захисту АСОІ великих організацій або невеликих АСОІ, що виконують відповідальні завдання або оброблюють особливо важливу інформацію. Порушення безпеки інформації в АСОІ великих організацій може завдати величезного матеріального збитку як самим організаціям, так і їх клієнтам. Тому такі організації вимушені приділяти особливу увагу гарантіям безпеки і реалізовувати комплексний захист. Комплексного підходу дотримуються більшість державних і великих комерційних підприємств і установ. Цей підхід знайшов своє віддзеркалення в різних стандартах.

Комплексний підхід до проблеми забезпечення безпеки засновано на розробленій для конкретної АСОІ політиці безпеки. Політикою безпеки є набір норм, правил і практичних рекомендацій, на яких будується управління, захист і розподіл інформації в АСОІ. Політика безпеки регламентує ефективну роботу засобів захисту АСОІ. Вона охоплює всі особливості процесу обробки інформації, визначаючи поведінку системи в різних ситуаціях.

Політика безпеки реалізується за допомогою адміністративно-організаційних заходів, фізичних і програмно-технічних засобів і визначає архітектуру системи захисту. Для конкретної організації політика безпеки носить індивідуальний характер і залежить від конкретної технології обробки інформації і програмних і технічних засобів, що використовуються.

Політика безпеки визначається способом управління доступом, що визначає порядок доступу до об'єктів системи. Розрізняють два основні види політики безпеки: виборчу і повноважну.

Виборча політика безпеки заснована на вибірковому способі управління доступом. Виборче управління доступом характеризується заданою адміністратором множиною дозволених відносин доступу (наприклад, у вигляді трійок <об'єкт, суб'єкт, тип доступу>). Зазвичай для опису властивостей виборчого управління доступом

застосовують математичну модель на основі матриці доступу.

Матриця доступу є матрицею, у якій стовпець відповідає об'єкту системи, а рядок - суб'єкту. На перетині стовпця і рядка матриці указується тип дозволеного доступу суб'єкта до об'єкту. Зазвичай виділяють такі типи доступу суб'єкта до об'єкту, як «доступ на читання», «доступ на запис», «доступ на виконання» і тому подібне. Матриця доступу є самим простим підходом до моделювання систем управління доступом. Проте вона є основою для складніших моделей, що описують реальні АСОІ.

Виборча політика безпеки широко застосовується в АСОІ комерційного сектора, оскільки її реалізація відповідає вимогам комерційних організацій по розмежуванню доступу і підзвітності, а також має прийнятну вартість

Повноважна політика безпеки заснована на повноважному (мандатному) способі управління доступом. Повноважне (або мандатне) управління доступом характеризується сукупністю правил надання доступу, визначених на множині атрибутів безпеки суб'єктів і об'єктів, наприклад, в залежності від мітки конфіденційності інформації й рівня допуску користувача. Повноважне управління доступом має на увазі, що:

- всі суб'єкти й об'єкти системи однозначно ідентифіковані;
- кожному об'єкту системи привласнена мітка конфіденційності інформації, що визначає цінність що міститься в цій інформації;
- кожному суб'єктові системи привласнений певний рівень допуску, що визначає максимальне значення мітки конфіденційності інформації об'єктів, до яких суб'єкт має доступ.

Чим важливіше об'єкт, тим вище його мітка конфіденційності. Тому найбільш захищеними виявляються об'єкти з найбільш високими значеннями мітки конфіденційності.

Основним призначенням повноважної політики безпеки є регулювання доступу суб'єктів системи до об'єктів з різними рівнями конфіденційності, запобігання витоку інформації з верхніх рівнів посадової ієрархії на нижні, а також блокування можливих проникнень з нижніх рівнів на верхні.

Крім управління доступом суб'єктів до об'єктів системи проблема захисту інформації має ще один аспект. Для отримання інформації про який-небудь об'єкт системи зовсім необов'язково шукати шляхи несанкціонованого доступу до нього. Необхідну інформацію можна отримати, спостерігаючи за обробкою об'єкту, що потребується, тобто використовуючи канали просочування інформації. У системі завжди існують інформаційні потоки. Тому адміністратору необхідно визначити, які інформаційні потоки в системі є "легальними", тобто не ведуть до просочування інформації, а який ведуть до витоку. Тому виникає необхідність розробки правил, що регламентують управління інформаційними потоками в системі. Звичайне управління інформаційними потоками застосовується в рамках виборчої або повноважної політики, доповнюючи їх і сприяючи підвищенню надійності системи захисту.

Виборче й повноважне управління доступом, а також управління

інформаційними потоками є тими фундаментом, на якому будується вся система захисту.

Під системою захисту АСОІ розуміють єдину сукупність правових і морально-етичних норм, адміністративно-організаційних заходів, фізичних і програмно-технічних засобів, направлених на протидію погрозам АСОІ з метою зведення до мінімуму можливості збитку.

Процес побудови системи захисту включає наступні етапи:

- аналіз можливих погроз АСОІ;
- планування системи захисту;
- реалізація системи захисту;
- супровід системи захисту.

Етап аналізу можливих погроз АСОІ необхідний для фіксації стану АСОІ (конфігурації апаратних і програмних засобів, технології обробки інформації) і визначення врахованих дій на компоненти системи. Практично неможливо забезпечити захист АСОІ від всіх дій, оскільки неможливо повністю встановити всі погрози і способи їх реалізацій. Тому із всієї множини вірогідних дій вибирають тільки такі дії, які можуть реально відбутися й завдати серйозного збитку.

На етапі планування формулюється система захисту як єдина сукупність заходів протидії погрозам різної природи.

По способах здійснення всі заходи забезпечення безпеки комп'ютерних систем підрозділяють на:

- правові (законодавчі);
- морально-етичні;
- адміністративні;
- фізичні;
- апаратно-програмні.

Перераховані заходи безпеки АСОІ можна розглядати як послідовність бар'єрів або рубежів захисту інформації. Для того, щоб дістатися до інформації, що захищається, потрібно послідовно подолати декілька рубежів захисту. Розглянемо їх докладніше.

Перший рубіж захисту, що встає на шляху людини, що намагається здійснити НСД до інформації, є чисто правовим. Цей аспект захисту інформації пов'язаний з необхідністю дотримання юридичних норм при передачі й обробці інформації. До правових заходів захисту інформації відносяться закони, що діють у країні, укази й інші нормативні акти, що регламентують правила поводження з інформацією обмеженого використання і відповідальності за їх порушення. Цим вони перешкоджають несанкціонованому використанню інформації і є стримуючим чинником для потенційних порушників.

Другий рубіж захисту утворюють морально-етичні заходи. Етичний момент у дотриманні вимог захисту має вельми велике значення. Дуже важливо, щоб люди, що мають доступ до комп'ютерів, працювали в здоровому морально-етичному кліматі.

До морально-етичних заходів протидії відносяться всілякі норми поведінки, які традиційно склалися або складаються в суспільстві в міру розповсюдження комп'ютерів у країні. Ці норми переважно не є обов'язковими, як законодавчо затверджені, але їх недотримання зазвичай веде до падіння престижу людини, групи осіб або організації. Морально-етичні норми бувають як неписаними (наприклад, загальновизнані норми чесності, патріотизму і так далі), так й оформленими в якесь зведення правил або розпоряджень. Наприклад, "Кодекс професійної поведінки членів Асоціації користувачів ЕОМ США" розглядає як неетичні дії, які навмисні або ненавмисні:

- порушують нормальну роботові комп'ютерних систем;
- викликають невиправдані витрати ресурсів (машинного часу, пам'яті, каналів зв'язку й тому подібне);
- порушують цілісність інформації (що зберігається і обробляється);
- порушують інтереси інших законних користувачів і тому подібне.

Третім рубежем, що перешкоджає неправомочному використанню інформації, є адміністративні заходи. Адміністратори всіх рангів з урахуванням правових норм і соціальних аспектів визначають адміністративні заходи захисту інформації.

Адміністративні заходи захисту відносяться до заходів організаційного характеру. Вони регламентують:

- процеси функціонування АСОІ;
- використання ресурсів АСОІ;
- діяльність її персоналу;
- порядок взаємодії користувачів з системою, з тим щоб найбільшою мірою утруднити або виключити можливість реалізації погроз безпеці.

Адміністративні заходи включають:

- розробку правил обробки інформації в АСОІ;
- сукупність дій при проектуванні і устаткуванні обчислювальних центрів та інших об'єктів АСОІ (облік впливу стихії, пожеж, охорона приміщень і т.п.);
- сукупність дій при підборі й підготовці персоналу (перевірка нових співробітників, ознайомлення їх з порядком роботи з конфіденційною інформацією, із заходами відповідальності за порушення правил її обробки; створення умов, при яких персоналу було б невигідно допускати зловживання і так далі);
- організацію надійного пропускного режиму;
- організацію обліку, зберігання, використання й знищення документів і носіїв з конфіденційною інформацією;
- розподіл реквізитів розмежування доступу (паролів, повноважень і тому подібне);
- організацію прихованого контролю за роботою користувачів і персоналу АСОІ;
- сукупність дій при проектуванні, розробці, ремонті й модифікації устаткування і програмного забезпечення (сертифікація використовуваних технічних і

програмних засобів, строге санкціонування, розгляд і затвердження всіх змін, перевірка на задоволення вимогам захисту, документальна фіксація змін і тому подібне).

Важливо відзначити, що, поки не будуть реалізовані дієві заходи адміністративного захисту ЕОМ, інші заходь будуть, безсумнівно, неефективні. Адміністративно-організаційні заходи захисту можуть показатися нудними й рутинними в порівнянні з морально-етичними й позбавленими конкретності в порівнянні з апаратно-програмними. Проте вони є потужним бар'єром на шляху незаконного використання інформації і надійною базою для інших рівнів захисту.

Четвертим рубежем є фізичні заходи захисту. До фізичних заходів захисту відносяться різного роду механічні, електро- і електронно-механічні пристрої або споруди, спеціально призначені для створення фізичних перешкод на можливих шляхах проникнення й доступу потенційних порушників до компонентів системи й інформації, що захищається.

П'ятим рубежем є апаратно-програмні засоби захисту. До них відносяться різні електронні пристрої і спеціальні програми, які реалізують самостійно або в комплексі з іншими засобами наступні способи захисту:

– ідентифікацію (розпізнавання) і аутентифікацію (перевірка достовірності) суб'єктів (користувачів, процесів) АСОІ;
– розмежування доступу до ресурсів АСОІ;
– контроль цілісності даних;
– забезпечення конфіденційності даних;
– реєстрацію й аналіз подій, що відбуваються в АСОІ;
– резервування ресурсів і компонентів АСОІ.

Більшість з перерахованих способів захисту реалізується криптографічними методами захисту інформації.

При проектуванні ефективної системи захисту слід враховувати ряд принципів, що відображають основні положення по безпеці інформації. До цих принципів належать наступні:

- Економічна ефективність. Вартість засобів захисту повинна бути менше, ніж розміри можливого збитку.
- Мінімум привілеїв. Кожен користувач повинен мати мінімальний набір привілеїв, необхідний для роботи.
- Простота. Захист тим більш ефективний, чим легше користувачеві з нею працювати.
- Відключення захисту. При нормальному функціонуванні захист не повинен відключатися. Тільки в особливих випадках співробітник із спеціальними повноваженнями може відключити систему захисту.
- Відвертість проектування й функціонування механізмів захисту. Фахівці, що мають відношення до системи захисту, повинні повністю уявляти собі принципи її функціонування і у разі виникнення скрутних ситуацій адекватно на них реагувати.

- Загальний контроль. Будь-які виключення з множини суб'єктів і об'єктів захисту, що контролюються, знижують захищеність автоматизованого комплексу обробки інформації.
- Незалежність системи захисту від суб'єктів захисту. Особа, що займається розробкою системи захисту, не повинна бути в числі тих, кого ця система контролюватиме.
- Звітність і підконтрольність. Система захисту повинна надавати докази коректності своєї роботи.
- Відповідальність. Мається на увазі особиста відповідальність осіб, що займаються забезпеченням безпеки інформації.
- Ізоляція й розділення. Об'єкти захисту доцільно розділяти на групи так, щоб порушення захисту в одній з груп не впливало на безпеку інших груп.
- Повнота і узгодженість. Надійна система захисту повинна бути повністю специфікована, протестована і погоджена.
- Параметризація. Захист стає більш ефективним і гнучким, якщо він допускає зміну своїх параметрів з боку адміністратора.
- Принцип ворожого оточення. Система захисту повинна проектуватися з розрахунку на вороже оточення. Розробники повинні виходити з припущення, що користувачі мають якнайгірші наміри, що вони здійснюватимуть серйозні помилки й шукатимуть шляхи обходу механізмів захисту.
- Залучення людини. Найбільш важливі і критичні рішення повинні прийматися людиною.
- Відсутність зайвої інформації про існування механізмів захисту. Існування механізмів захисту повинне бути по можливості приховано від користувачів, робота яких повинна контролюватися.

Результатом етапу планування є розгорнений план захисту АСОІ, що містить перелік компонентів АСОІ і можливих дій, що захищаються, на них, позначка захисту інформації в АСОІ, правила обробки інформації в АСОІ, що забезпечують її захист від різних дій, а також опис запланованої системи захисту інформації.

Суть етапу реалізації системи захисту полягає в установці і налаштуванні засобів захисту, необхідних для реалізації запланованих правил обробки інформації.

Завершальний етап супроводу полягає в контролі роботи системи, реєстрації подій, що відбуваються в ній, їх аналізі з метою виявлення порушень безпеки, корекції системи захисту.

Питання щодо самоконтролю:

1. Основні підходи і міри забезпечення безпеки АСОІ.
2. Що ви можете розповісти про доступ до інформації?
3. Що таке цілісність компонента або ресурсу системи?
4. Що ви знаєте про доступ інформації і її конфіденційності.
5. Що ви знаєте про апаратно-програмні засоби захисту?

6. Що ви знаєте про фізичні заходи захисту?
7. Які ви знаєте прийоми несанкціонованого доступу до інформації?

Тема №4 (2 години)
Шифрування методом Вернама

Мета: Ознайомитись з шифруванням методом Вернама

План

1. Поняття методу Вернама
2. Основні визначення та шифрування методом Вернама

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Система шифрування Вернама є по суті окремим випадком системи шифрування Віжінера при значенні модуля m = 2. Конкретна версія цього шифру, запропонована в 1926 р. Гілбертом Вернамом, співробітником фірми AT&T США, використовує двійкове представлення символів початкового тексту.

Кожен символ початкового відкритого тексту з англійського алфавіту {A,B,C,D…Z}, розширеного шістьма допоміжними символами (пропуск, повернення каретки й тому подібне), спочатку кодувався в 5-бітовий блок (b_0, $b_1 \ldots b_4$) телеграфного коду Бодо.

Випадкова послідовність двійкових ключів k_0, k_1, k_2... заздалегідь записувалася на паперовій стрічці.

Схема передачі повідомлень з використанням шифрування методом Вернама показана на малий. 2.11. Шифрування початкового тексту, заздалегідь перетвореного в послідовність двійкових символів x, здійснювалося шляхом складання по модулю 2 символів x з послідовністю двійкових ключів K.

Символи шифротексту

$$y = x \oplus k. \qquad (2.13)$$

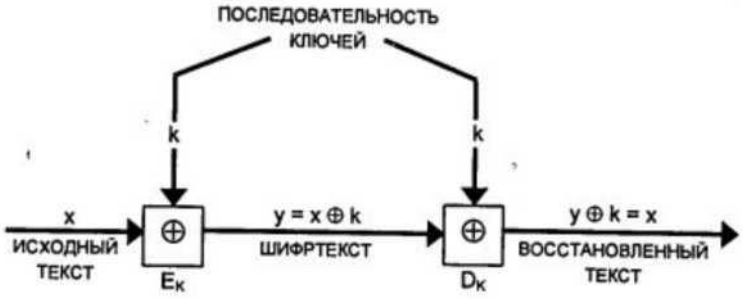

Рис. 2.11. Схема шифрования и расшифрования сообщений по методу Вернама

Розшифрування полягає в складанні по модулю 2 символів шифротекста з тією ж послідовністю ключів *K*:

$y \odot k = k \odot k \odot x = x.$

При цьому послідовності ключів, використані при шифруванні і розшифруванні, компенсують один одного (при складанні по модулі 2), і в результаті відновлюються символи *x* початкового тексту.

При розробці своєї системи Вернам перевіряв її з допомогою закільцьованих стрічок, встановлених на передавачі і приймачі для того, щоб використовувалася одна й та ж послідовність ключів.

Слід зазначити, що метод Вернама не залежить від довжини послідовності ключів і, крім того, він дозволяє використовувати випадкову послідовність ключів. Проте при реалізації методу Вернама виникають серйозні проблеми, пов'язані з необхідністю доставки одержувачеві такої ж послідовності ключів, як у відправника, або з необхідністю безпечного зберігання ідентичних послідовностей ключів у відправника і одержувача. Ці недоліки системи шифрування Вернама подолані при шифруванні методом гамування.

Питання щодо самоконтролю:

1) Розкажіть про шифрування методом Вернама.
2) Що ви знаєте про послідовність ключів у методі Вермана.
3) Недоліки метода Вермана.

Тема №5 (2 години)

Тема: Методи генерації псевдовипадкових послідовностей чисел.

Мета: Ознайомитись з методами генерації псевдовипадкових послідовностей чисел

План

1. Основні визначення псевдовипадкових послідовностей
2. Методи знаходження псевдовипадкових чисел

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Планації послідовностей псевдовипадкових чисел на основі лінійних рекурентних співвідношень.

Розглянемо рекурентні співвідношення і їх різницеві рівняння:

$$\sum_{j=0}^{k} h_j a_{i+j} = 0,$$

$$a_{i+k} = -\sum_{j=0}^{k-1} h_j a_{i+j}$$

де $h_0 \neq 0$ $h_k = 1$ і кожне h(належить полю GF(q).

Вирішенням цих рівнянь є послідовність елементів a_0, a_1, a_2...' поля GF(q). Співвідношення (2.18) визначає правило обчислення a_k по відомих значеннях величин $a_0, a_1, a_2...a_{k-1}$. Потім по відомих значеннях $a_0, a_1, a_2...a_k$ знаходять a_{k+1} і так далі. В результаті по початкових значеннях $a_0, a_1, a_2...a_{k-1}$ можна побудувати нескінченну послідовність, причому кожен її подальший член визначається з k попередніх. Послідовності такого вигляду легко реалізуються на комп'ютері, при цьому реалізація виходить особливо простою, якщо всі h, і a, приймають значення 0 і 1 з поля GF(2).

На мал. 2.12 показана лінійна послідовна перемикаюча схема, яка може бути використана для обчислення суми (2.18) і, отже, для обчислення значення a_k по значеннях до попередніх членів послідовності. Початкові величини $a_0, a_1, a_2...a_{k-1}$ поміщаю ення вмісту якого відповіда

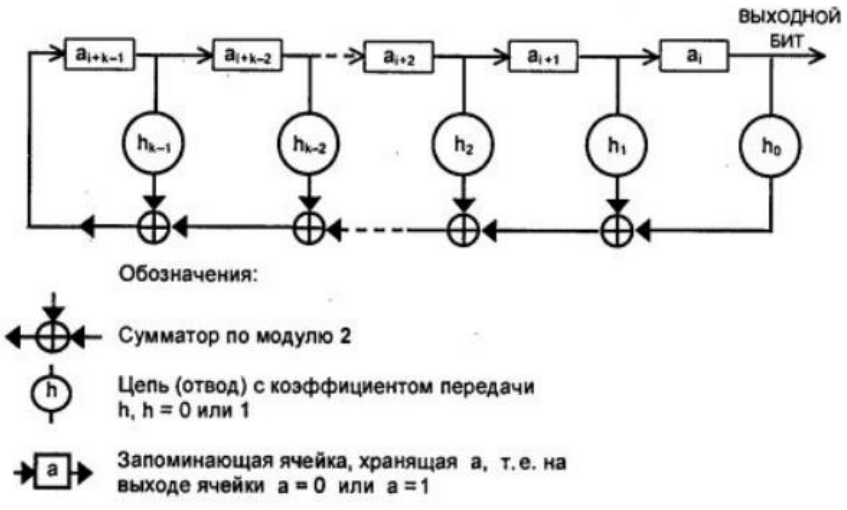

Рис. 2.12. Генератор с регистром сдвига

вихід після *i*-го зрушення рівний a_i. Даний пристрій називають генератором послідовності чисел, побудованим на базі зсувного регістра з лінійним зворотним зв'язком.

Вирішення лінійних рекурентних співвідношень, що реалізовуються генератором з регістром зрушення, описуються наступною теоремой. Нехай многочлен

$$h(X) = \sum_{j=0}^{k} h_j X^j$$

де x - формальна змінна; h_j -коефіцієнт при X^j, що приймає значення 0 або 1; $h_0 \neq 0$. $h_k = 1$, і нехай n - найменше ціле позитивне число, для якого многочлен x^n-1 ділиться на $h(x)$. Крім того, многочлен

$g(X) = (X^n-1)/h(X)$.

Тоді вирішення рекурентних співвідношень

$$\sum_{j=0}^{k} h_j a_{i+j} = 0$$

У вигляді послідовності елементів $a_0, a_1, a_2 ... a_{n-1}$ періодичні з періодом n і сукупність, складена з перших періодів всіх можливих рішень, що розглядаються як многочлени по модулю (X_{n-1}), тобто

$$\boldsymbol{a(X) = a_0 * X^{n-1} + a_1 * X^{n-2} + ... + a_{n-2} * X + a_{n-1}.}$$

співпадає з ідеалом, породженим багаточленом g(X) в алгебрі многочленів по модулю (X_{n-1}).

Відмітимо, що якщо при такому визначенні багаточлена $a(X)$ елементи $a_0, a_1, a_2 ...$ обчислюються в порядку зростання номерів, то коефіцієнти многочлена $a(X)$ обчислюються, починаючи з коефіцієнтів при ступенях вищих порядків. Слід також відзначити, що вид багаточлена

$$h(X) = \sum_{j=0}^{k} h_j X^j$$

визначає конфігурацію зворотних зв'язків (відведень) h_j у генераторі із зсувним регістром. Іншими словами, якщо в багаточлена $h(X)$ коефіцієнт $h_j=l$, це означає, що відведення h_j у схемі генератора присутнє. Якщо ж у багаточлена $h(X)$ коефіцієнт $h_j = 0$, то відведення h_j у схемі генератору відсутнє. Як $h(X)$ необхідно вибирати

примітивний багаточлен, що не приводиться. При такому виборі багаточлена *h(X)* із старшим ступенем *m* генератор забезпечує видачу псевдовипадкової послідовності двійкових чисел з максимально можливим 2m-1.

Розглянемо як приклад трьохрозрядний зсувний регістр з лінійним зворотним зв'язком (рiс.2.13), побудований у відповідності з примітивним багаточленом, що не приводиться

Рис. 2.13. Трехразрядный регистр сдвига с обратными связями (генератор гаммы шифра Гш)

Нехай ключем є 101. Регістр починає працювати з цього стану; послідовність станів регістра наведена на рис. 2.13. Регістр проходить через всі сім ненульових станів і знову повертається у свій початковий стан 101. Це найбільш довгий період даного регістра з лінійною зворотною зв'язком. Така послідовність називається послідовністю максимальної довжини для зсувного регістра (Maximal Lenght Shift Register SEQUENCE- MLSRS). Пітерсон і Уелдон показали, що при будь-якому цілому *m* існує *m*-бітова послідовність MLSRS з періодом *2m-1*. Зокрема, при *m* = 100 послідовність матиме період 2100-1 і не повторитися 1016 разів при передачі її по лінії зв'язку із швидкістю 1 Мбіт/с. У нашому прикладі вихідною послідовністю (гаммою шифру) Гш зсувного регістра із зворотним зв'язком є послідовність 1010011, яка циклічно повторюється. У цій послідовності є чотири одиниці і три нулі, і їх розподіл настільки близько до рівномірного, наскільки це можливо в послідовності, що має довжину 7. Якщо розглянути пари послідовних бітів, то пари 10 і 01 з'являються по два рази, а пари 00 і 11 - один раз, що знову виявляється настільки близьким до рівномірного розподілу, наскільки це можливо. У разі послідовності максимальної довжини для *m*-розрядного регістра це властивість рівного розподілу розповсюджується на трійки, четвірки і так далі бітів, аж до *m*-бітових груп. Завдяки такій близькості до рівномірного розподілу послідовності максимальної довжини часто використовуються як псевдовипадкові послідовності в криптографічних системах, які імітують роботу криптостійкої системи одноразового шифрування.

Хоча така криптографічна система здійснює імітацію свідомо криптостійкої системи одноразового шифрування, сама вона не відрізняється стійкістю й може бути розкрита за декілька секунд роботи комп'ютера за умови наявності відомого відкритого тексту.

Якщо відведення регістра із зворотним зв'язком зафіксовані, то для знаходження початкового стану регістра досить знати *m* бітів відкритого тексту. Щоб знайти *m* бітів ключового потоку, *m* бітів відомого відкритого тексту складають по модулю 2 з відповідними *m* бітами шифротексту. Отримані *m* бітів дають стан зсувного регістра із зворотним зв'язком у зворотному напрямі на деякий момент часу. Потім, моделюючи роботу регістра назад, можна визначити його початковий стан.

Якщо відведення регістра із зворотним зв'язком не фіксовані, а є частиною ключа, то досить *2m* біти відомого відкритого тексту, щоб порівняно швидко визначити розташування відведень регістра і його початковий стан. S(i) - вектор-стовпець, що складається з *m* символів 0 й 1, який визначає стан регістра в *i*-й момент часу. Тоді

$$S(i + 1) = A * S(i) \bmod 2,$$

де *A*-матрица розміром $m \times m$, що визначає положення відводів регістра із зворотним зв'язком.

Для трьохрозрядного регістра (мал. 2.13)

$$A = \begin{vmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{vmatrix}.$$

Матриця *A* завжди має наступну структуру: у її першому рядку відображена послідовність відведень в регістрі, безпосередньо під головною діагоналлю розташовуються одиниці, а в решті позицій розташовуються нулі.

2m бітів відомого відкритого тексту дозволяють обчислити *m* послідовних бітів ключового потоку. Для спрощення позначень припустимо, що це - перші *2m* бітів ключового потоку. Отже

- *S(1)* - перша група *m* відомих бітів ключового потоку;
- *S(2)*- наступна група (починаючи з номера 2) з *m* відомих бітів ключового потоку;
- *S(m +1)* - остання група з *m* відомих бітів ключового потока.

Далі можна утворити дві матриці розміром $m \times m$:

$X(1) = [S(1), S(2), \ldots, S(m)]$,

$X(2) = [S(2), S(3), \ldots, S(m +1)1]$,

які зв'язані співвідношенням

$X(2) = A * X(1) \bmod 2$.

Можна показати, що для будь-якої послідовності максимальної довжини матриця *x(1)* завжди несингулярна, тому матрицю *A* можна обчислити як:

$A = X(2)\,[X(1)]^{-1} \bmod 2$..

Звернення матриці *X(1)* вимагає (найбільше) операцій, тому легко виконується при будь-якому розумному значенні *m*. Для криптографії послідовності максимальної довжини MLSRS можна зробити криптостійкими, використовуючи нелінійну логіку. Зокрема, запропонований варіант, у якому в якості ключового потоку використовується нелінійно "фільтрований" вміст зсувного регістра, а для отримання послідовності максимальної довжини - лінійний зворотний зв'язок, як показаний на

рис. 2.14.

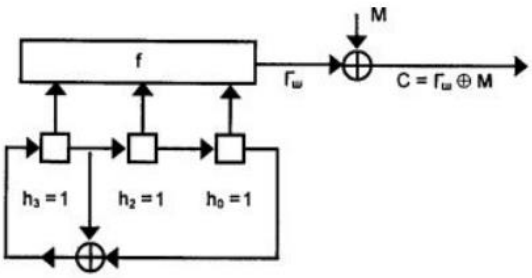

Рис. 2.14. Линейный сдвиговый регистр с нелинейными логическими цепями на выходе

Функція *f* повинна вибиратися так, щоб забезпечити хороший баланс між нулями і одиницями, а фільтрована послідовність мала розподіл, близький до рівномірного. Необхідно також, щоб фільтрована послідовність мала великий період. Якщо (2^m-1) є простим числом (як в прикладі: при $m = 3$ маємо $2^3-1 = 7$), то фільтрована послідовність - може мати період (2^m-1) (при виборі структури зсувного регістра у відповідності з примітивним багаточленом, що не приводиться, *h(X)* ступеня *m*).

До таких значень ш відносяться, зокрема, наступні:

2, 3, 5, 7, 13, 17, 19, 31, 61, 89, 107, 127, 521, 607, 1279, 2203, 2281, а отримані таким чином прості числа називаються простимі числами Мерсенна.

Не дивлячись на те, що фільтровану вихідну послідовність зазвичай не можна отримати за допомогою *m*-розрядного зсувного регістра з лінійним зворотним зв'язком, її завжди можна одержати за допомогою зсувного регістра більшої довжини з лінійним зворотним зв'язком. Регістр завдовжки $(2m-1)$ завжди дозволить це зробити, а іноді придатний і коротший регістр.

Ще привабливіше використання в ланцюзі зворотного зв'язку нелінійної логіки, проте теорія таких схем недостатньо добре освітлена (у відкритій літературі).

Питання щодо самоконтролю:

1. Які основні вимоги висуваються до криптографічного стійкого генератора псевдовипадкової послідовності чисел?
2. В чему полягає суть способу генерації псевдовипадкових чисел?
3. Які основні вимоги висуваються до криптографічного стійкого генератора псевдовипадкової послідовності чисел?
4. Що ви знаєте про довжину періоду гамми?

Тема №6 (2 години)
Математичні основи криптографії

Мета: Ознайомитись з математичними основами криптографії нашого часу

План

1. Поняття Шеннона
2. Принципи Шеннона
3. Принцип роботи шифрів-перестановок

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Великий вплив на розвиток криптографії виявили роботи американського математика Клода Шеннона, що з'явилися в середині XX століття. У цих роботах були закладені основи теорії інформації, а також був розроблений математичний апарат для досліджень у багатьох галузях науки, пов'язаних з інформацією. Більше того, прийнято вважати, що теорія інформації як наука народилася в 1948 році після публікації роботи К. О. Шеннону "Математична теорія зв'язку". У своїй роботі "Теорія зв'язку в секретних системах" Клод Шеннон узагальнив і привів до ладу, накопичений до нього досвід розробки шифрів) - Виявилося , що навіть у дуже складних шифрах як типові компоненти можна виділити такі прості шифри як шифри заміни, шифри перестановки або їхні сполучення.

Шифр заміни є найпростішим, найбільш популярним шифром. Типовими прикладами є шифр Цезаря, "цифирная абетка" Петра Великого й "танцюючі чоловічки" А. Г.,О. Конан Дойла. Як видне із самої назви, шифр заміни здійснює перетворення заміни букв або інших "частин" відкритого тексту на аналогічні "частини" шифрованого тексту. Легко дати математичний опис шифру заміни. Нехай X і Y — два алфавіти (відкритого й шифрованого текстів відповідно), що складаються з однакового числа символів. Нехай також g: X→Y — взаємо-однозначне відображення X в Y. Тоді шифр заміни діє так: відкритий текст $x_1, x_2, \ldots x_n$ перетвориться в шифрований текст $g(x_1)g(x_2)...g(x_n)$. Шифр перестановки, як видно з назви, здійснює перетворення перестановки букв у відкритому тексті. Типовим прикладом шифру перестановки є шифр "Сцитала". Звичайно відкритий текст розбивається на відрізки рівної довжини й кожний відрізок шифрується незалежно. Нехай, наприклад, довжина відрізків рівна n і σ — взаємо-однозначне відображення множини {1,2,..., n} в себе. Тоді шифр перестановки діє так: відрізок відкритого тексту $x_1 \ldots x_n$ перетвориться у відрізок шифрованого тексту $x\sigma(1)... -x\sigma(n)$.

Найважливішим для розвитку криптографії був результат К. О. Шеннона про існування та одиничність абсолютно стійкого шифру. Єдиним таким шифром є яка-небудь форма так званої стрічки однократного використання, у якій відкритий текст

"поєднується " з повністю випадковим ключем такої ж довжини. Цей результат був доведений К. О. Шенноном за допомогою розробленого їм теоретико-інформаційного методу дослідження шифрів. Ми не будемо тут зупинятися на цьому докладно, зацікавленому читачеві рекомендуємо вивчити роботу К. О. Шеннона. Обговоримо особливості будови абсолютно стійкого шифру й можливості його практичного використання. Типовим і найбільш простим прикладом реалізації абсолютно стійкого шифру є шифр Вернама, який здійснює побітове додавання *n*-бітового відкритого тексту й *n*-бітового ключа:

$y_1 = x_1 \odot k_i,\ i=1\ldots n$

Тут $x_1 \ldots x_n$ — відкритий текст, $k_1 \ldots k_n$ — ключ , в $1 \ldots y_n$, шифрований текст. Підкреслимо, що для абсолютної стійкості істотною є кожна з наступних вимог до стрічки однократного використання:

1) повна випадковість (рівноймовірність) ключа (це, зокрема, означає, що ключ не можна виробляти за допомогою якого-небудь детермінованого пристрою);
2) рівність довжини ключа й довжини відкритого тексту;
3) однократність використання ключа.

У випадку порушення хоча б однієї із цих умов шифр перестає бути абсолютно стійким і з'являються принципові можливості для його розкриття (хоча вони можуть бути важко реалізованими). Але, виявляється, саме ці умови й роблять абсолютно стійкий шифр дуже дорогим і непрактичним. Перш ніж користуватися таким шифром, ми повинні забезпечити всіх абонентів достатнім запасом випадкових ключів і виключити можливість їхнього повторного застосування. А це зробити надзвичайно важко й дорого. Як відзначав Д. Кан: "Проблема створення, реєстрації, поширення й скасування ключів може здатися не занадто складною тому, хто не має досвіду передачі повідомлень по каналах військового зв'язку, але у воєнний час обсяг переданих повідомлень заганяє в глухий кут навіть професійних зв'язківців. За добу можуть бути зашифровані сотні тисяч слів. Створення мільйонів ключових знаків зажадало б величезних фінансових витрат і було б сполучене з більшими витратами часу. Тому що кожний текст повинен мати свій власний, єдиний і неповторний ключ, застосування ідеальної системи зажадало б передачі принаймні такої кількості знаків, яка еквівалентна всьому обсягу переданої військової інформації."

У силу зазначених причин абсолютно стійкі шифри застосовуються тільки в мережах зв'язку з невеликим обсягом переданої інформації, звичайно це мережі для передачі особливо важливої державної інформації.

Тепер уже зрозуміло, що найчастіше для захисту своєї інформації законні користувачі змушені застосовувати не абсолютно стійкі шифри. Такі шифри, по крайній мері теоретично, можуть бути розкриті. Питання тільки в тому, чи вистачить в супротивника сил, засобів і часу для розробки й реалізації відповідних алгоритмів. Зазвичай цю думку виражають так: противник з необмеженими ресурсами може розкрити будь-який неабсолютно стійкий шифр. Як же повинен діяти в цій ситуації

законний користувач, вибираючи для себе шифр? Найкраще, звичайно, було б довести, що ніякий супротивник не може розкрити обраний шифр, скажемо, за 10 років і тим самим одержати теоретичну оцінку стійкості. На жаль, математична теорія ще не дає потрібних теорем — вони ставляться до невирішеної проблеми, нижніх оцінок обчислювальної складності завдань . Тому в користувача залишається єдиний шлях — одержання практичних оцінок стійкості. Цей шлях складається з наступних етапів:

– зрозуміти й чітко сформулювати, від якого супротивника ми збираємося захищати інформацію; необхідно усвідомити, що саме супротивник знає або зможе довідатися про систему шифру, а також які сили й засоби він зможе застосувати для його розкриття;

– подумки стати в положення супротивника й намагатися з його позицій атакувати шифр, тобто розробляти різні алгоритми розкриття шифру; при цьому необхідно в максимальній мері забезпечити моделювання сил , засобів і можливостей супротивника;

– найкращий з розроблених алгоритмів використовувати для практичної оцінки стійкості шифру.

Тут корисно для ілюстрації згадати про два найпростіші методи розкриття шифру: випадкове вгадування ключа (він спрацьовує з маленькою ймовірністю, зате має маленьку складність) і перебір усіх підряд ключів аж до знаходження щирого (він спрацьовує завжди, зате має більшу складність). Відзначимо також, що не завжди потрібна атака на ключ: для деяких шифрів можна відразу, навіть не знаючи ключа, відновлювати відкритий текст по шифрованому .

Питання щодо самоконтролю:

1) Після публікації якої роботи К. О.О.Шеннона прийнято вважати , що теорія інформації народилася як наука?

2) Які вимоги до стрічки однократного використання є істотними для абсолютної стійкості?

3) У якій роботі Клод Шеннон узагальнив накопичений досвід розробки шифрів, привів його до ладу?

4) Який шифр є найпростішим і найбільш популярним?

5) Чому необхідно забезпечити всіх абонентів перш ніж користуватися абсолютно стійким шрифтом?

6) Який шифр являє типовий приклад шифру перестановки?

7) Як діє шифр перестановки?

8) Який результат К. О.О.Шеннона був найважливішим для розвитку криптографії?

Тема №7 (2 години)

Нові напрямки у розвитку криптографії

Мета: Ознайомитися з новими напрямками у криптографії

План

1. Однобічна функція
2. Функції з секретом та їх застосування
3. Принципи побудови системи

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

В 1983 році в книзі "Коди й математика" М. Н. Аршинова Л. Е. Садовского було написано: "Прийомів тайнопису - безліч, і швидше за все, це та область, де вже немає потреби придумувати що-небудь суттєво нове." Однак це була чергова велика омана щодо криптографії. Вже в 1976 році була опублікована робота молодих американських математиків У.Диффи й М.Є.Хеллмана "Нові напрямки в криптографії"1′, яка не тільки суттєво змінила криптографію, але й привела до появи й бурхливого розвитку нових напрямків у математиці. Центральним поняттям "нової криптографії" є поняття однобічної функції.

Однобічної називається функція F: X→Y, що володіє двома властивостями:

а) існує поліноміальний алгоритм обчислення значення *F(x)*;

б) не існує поліноміального алгоритму інвертування функції *F* (тобто рішення рівняння *F(x) = y* відносно *x*).

Відзначимо, що однобічна функція суттєво відрізняється від функцій, звичних зі шкільної лави, через обмеження на складність її обчислення й інвертування. Питання про існування однобічних функцій поки відкрите.

Ще одним новим поняттям є поняття функції із секретом. Іноді ще вживається термін функція з пасткою. Функцією із секретом *K* називається функція $F_K : X —> B$, що залежить від параметра *K* і, що володіє трьома властивостями:

а) існує поліноміальний алгоритм обчислення значення $F_K(X)$ для будь-яких *K* і *x*;

б) не існує поліноміального алгоритму інвертування F_K при невідомому *K*;

в) існує поліноміальний алгоритм інвертування F_K при відомому *K*.

Про існування функцій із секретом можна сказати те ж саме, що сказане про однобічні функції. Для практичних цілей криптографії було побудовано кілька функцій, які можуть виявитися функціями із секретом. Для них властивість б) поки строго не доведена, але вважається, що завдання інвертування еквівалентне деякому давно досліджуваному важкому математичному завданню. Найбільш відомою і популярною з них є теоретико-числова функція, на якій побудований шифр RSA.

Застосування функцій із секретом у криптографії дозволяє:

1) організувати обмін шифрованими повідомленнями з використанням тільки відкритих каналів зв'язку, тобто відмовитися від секретних каналів зв'язку для попереднього обміну ключами ;

2) включити в завдання розкриття шифру важке математичне завдання й тим самим підвищити обґрунтованість стійкості шифру;

3) вирішувати нові криптографічні завдання, відмінні від шифрування (електронний цифровий підпис та ін.).

Систему по п.1) називають криптосистемою з відкритим ключем , оскільки алгоритм шифрування F_K є загальнодоступним або відкритим.

Останнім часом такі криптосистеми ще називають асиметричними, оскільки в них є асиметрія в алгоритмах: алгоритми шифрування й дешифрування різні. На відміну від таких систем традиційні шифри називають симетричними: у них ключ для шифрування й дешифрування той самий. Для асиметричних систем алгоритм шифрування загальновідомий, але відновити по ньому алгоритм дешифрування за поліноміальний час неможливо.

Описану вище ідею Диффи й Хеллман запропонували використовувати також для електронного цифрового підпису повідомлень, який неможливо підробити за поліноміальний час. Нехай користувачеві A необхідно підписати повідомлення x. Він, знаючи секрет K, знаходить таке b, що $F_x(y) = x$, і разом з повідомленням x посилає b користувачеві B як свій цифровий підпис. Користувач $У$ зберігає b як доказ того, що A підписав повідомлення x. Повідомлення, підписане цифровий підписом, можна представити як пару (x, y), де x — повідомлення, у рішення рівняння $F_x(y)=x$, $F_k: X \to У$ функція із секретом, відома всім взаємодіючим абонентам. З визначення функції F_k очевидні наступні корисні властивості цифрового підпису:

1) підписати повідомлення x, тобто вирішити рівняння $F_x(y) = x$, може тільки абонент — власник даного секрету K; інакше кажучи, підробити підпис неможливо;

2) перевірити дійсність підпису може будь-який абонент, що знає відкритий ключ , тобто саму функцію F_k;

3) при виникненні суперечок відмовитися від підпису неможливо в силу її непідроблюваності;

4) підписані повідомлення (x,y) можна, не побоюючись збитку, пересилати по будь-яких каналах зв'язку.

Крім принципу побудови криптосистеми з відкритим ключем, Діффі і Хеллман у тій же роботі запропонували ще одну нову ідею — відкритий розподіл ключів . Вони задалися питанням: чи можна організувати таку процедуру взаємодії абонентів A і B по відкритих каналах зв'язку , щоб вирішити наступні завдання :

1) спочатку в A і B немає ніякої загальної секретної інформації, але наприкінці процедури така загальна секретна інформація (загальний ключ) в A и B

з'являється, тобто виробляється;

2) пасивний супротивник, який перехоплює всі передачі інформації і знає, що хочуть одержати *A* і *B*, проте не може відновити вироблений загальний ключ *A* і *B*.

Діффі й Хеллман запропонували вирішувати ці завдання за допомогою функції

$F(x) = a^x \pmod p$,

де *p* - велике просте число, *x* - довільне натуральне число, *a* - деякий примітивний елемент поля GF(p). Загальновизнано, що інвертування функції $ax \pmod p$, тобто дискретне логарифмування, є важким математичним завданням. Сама процедура або, як прийнято говорити, протокол вироблення загального ключа описується в такий спосіб. Абоненти *A* і *B* незалежно один від одного випадково вибирають по одному натуральному числу скажемо *XA XB*. Ці елементи вони тримають у секреті. Далі кожний з них обчислює новий елемент:

$Y_A = a^{xa} \pmod p$, $y_в = a^{xb} \pmod p$.

(Число *p* вважаються загальнодоступним.) Потім вони обмінюються цими елементами по каналу зв'язку. Тепер абонент А, одержавши y_B і знаючи свій секретний елемент *xa*, обчислює новий елемент: $Y^{xa}{}_b \pmod p = (a^{xb})^{xa} \pmod p$.

Аналогічно надходить абонент В:

$Y^{xb}{}_a \pmod p = (a^{xa})^{xb} \pmod p$.

Тим самим в *A* і *B* з'явився загальний елемент поля, рівний *axaxb*. Цей елемент і оголошується загальним ключем *A* і *B*. З опису протоколу видно, що супротивник знає *p,a,axa,axb,* не знає *xa* і *xb* хоче довідатися *axaxb*. У цей час немає алгоритмів дій супротивника, більше ефективних, ніж дискретне логарифмування, а це - важке математичне завдання. Успіхи, досягнуті в розробці схем цифровому підпису й відкритого розподілу ключів, дозволили застосувати ці ідеї також і до інших завдань взаємодії вилучених абонентів. Так виник великий новий напрямок теоретичної криптографії - криптографічні протоколи. Об'єктом вивчення теорії криптографічних протоколів є видалені абоненти, що взаємодіють, как правило, по відкритим каналам зв'язку. Метою взаємодії абонентів є рішення деякої задачі. Є також противник, який переслідує власні цілі. При цьому противник в різних задачах може мати різні можливості: наприклад, може взаємодіяти з абонентами від імені інших абонентів чи втручатися в обмін інформацією між абонентами і т.д. Противником може виявитися навіть один з абонентів чи декілька абонентів, що вступили у змову. Приведемо ще декілька прикладів задач, що вирішаються видаленими абонентами

1. Взаємодіють два абоненти, що не довіряють один одному. Вони хочуть кинути жереб за допомогою монети. Це треба зробити так, щоб абонент, що підкидає монету, не міг змінити результат підкидання після одержання здогаду від абонента, що вгадує цей результат. Протокол рішення цього завдання прийнято називати протоколом підкидання монети. Опишемо один із найпростіших протоколів підкидання монети по телефону (так звана схема Блюма- Микали). Для його реалізації в абонентів *A* і *B* повинна бути однобічна функція $f: X \to B$, що задовольняє наступним

умовам:

1) X — безліч цілих чисел, яке містить однакову кількість парних і непарних чисел;

2) які-небудь числа $x_1\ x_2 \in X$, що мають один образ $f(x) = f(x)$, мають одну парність;

3) по заданому образу $f(x)$ "важко" обчислити парність невідомого аргументу x.

Роль підкидання монети грає випадковий і рівноймовірний вибір елемента $x \in X$, а роль орла й решки — парність і непарність x відповідно. Нехай A — абонент, що підкидає монету, а B — абонент, що вгадує результат. Протокол складається з наступних кроків:

1) A вибирає x ("підкидає монету"), зашифровує x, тобто обчислює $b = f(x)$ і посилає b абонентові B;

2) B одержує b, намагається вгадати парність x і посилає свій здогад з A;

3) A одержує здогад від $У$ и повідомляє B, чи угадав він, посилаючи йому обране число x;

4) $У$ перевіряє, чи не обманює A, зі значення $f(x)$ і порівнюючи його з отриманим на другому кроці значенням $У$.

2. Взаємодіють два абоненти A і B (типовий приклад: A — клієнт банку, B — банк). Абонент A хоче довести абонентові B, що він саме A, а не супротивник. Протокол рішення цього завдання прийнято називати протоколом ідентифікації абонента.

3. Взаємодіють кілька вилучених абонентів, що одержали накази з одного центру. Частина абонентів, включаючи центр, можуть бути противниками. Необхідно виробити єдину стратегію дій, виграшну для абонентів. Це завдання прийнято називати завданням про візантійські генерали, а протокол її рішення протоколом візантійської угоди. Опишемо приклад, якому це завдання зобов'язане своєю назвою. Візантія. Ніч перед великою битвою. Візантійська армія складається з n легіонів, кожний з яких підкоряється своєму генералові. Крім того, в армії є головнокомандуючий, який керує генералами. Однак імперія перебуває в занепаді й до однієї третини генералів, включаючи головнокомандуючого, можуть бути зрадниками. Протягом ночі кожний з генералів одержує від головнокомандуючого наказ про дії на ранок, причому можливі два варіанти наказу: "атакувати" чи "відступати". Якщо всі чесні генерали атакують, то вони перемагають. Якщо всі вони відступають, то їм вдається зберегти армію. Але якщо частина їх атакує, а частина відступає, то вони зазнають поразки. Якщо головнокомандуючий виявиться зрадником, то він може дати різним генералам різні накази, тому накази головнокомандуючого не варто виконувати беззаперечно. Якщо кожний генерал буде діяти незалежно від інших, результати можуть виявитися жалюгідними. Очевидно, що генерали потребують обміну інформацією один з одним (щодо отриманих наказів) для того, щоб дійти згоди. Осмислення різних протоколів і методів їхньої побудови привело в 1985-1986 рр. до

появи двох плідних математичних моделей — інтерактивної системи доказу й докази з нульовим розголошенням. Математичні дослідження цих нових об'єктів дозволили довести багато тверджень, досить корисних при розробці криптографічних протоколів. Під інтерактивною системою доказу (P,V,S) розуміють протокол взаємодії двох абонентів: P (що доводить) і V (що перевіряє). Абонент P хоче довести V, що твердження S істинно. При цьому абонент V самостійно, без допомоги P, не може перевірити твердження S (тому V і називається таким, що перевіряють). Абонент P може бути й супротивником, який хоче довести V, що твердження S істинно, хоча воно неправдиве. Протокол може складатися з багатьох раундів обміну повідомленнями між P і V і повинен задовольняти двом умовам:

1) повнота — якщо Б дійсно істинно, те абонент P переконає абонента V визнати це;

2) коректність - якщо S неправдиве, то абонент P навряд чи переконає абонента V, що S істинно. Тут словами "навряд чи" ми для простоти замінили точне математичне формулювання. Підкреслимо, що у визначенні системи (P, V, S) не допускалося, що V може бути супротивником. А якщо V виявився супротивником, який хоче "вивідати" у P яку-небудь нову корисну для себе інформацію про твердження S? У цьому випадку P, звичайно, може не хотіти , щоб це трапилося в 1 результаті роботи протоколу (P,V,S). Протокол (P,V,S), що вирішує таке завдання, називається доказом з нульовим розголошенням і повинен задовольняти, крім умов 1) і 2), ще й наступній умові:

3) нульове розголошення - у результаті роботи протоколу (P, V,S) абонент V не збільшить свої знання про твердження S або, інакше кажучи, не зможе витягти ніякої інформації про те, чому S істинно.

Питання:

1) У якій книзі М.Н. М.Аршинова й Л.Е. Садового було написано в 1983 році : "Прийомів тайнопису - безліч , і швидше за всі , це та область , де вже немає споживи придумувати що-небудь суттєво нове"?

2) Яка робота молодих американських математиків У.Діффі і М.Є.Хеллмана була опублікована в 1976 році, яка не тільки суттєво змінила криптографію, а й привела до появи й бурхливому розвитку нових напрямків у математикові?

3) Що є центральним поняттям цієї роботи?

4) Якими двома особливостями володіє це поняття?

5) Якими трьома особливостями володіє функція із секретом?

6) Що дозволяє використання функцій із секретом?

7) Чому криптосистему з відкритим ключем так називають?

8) У чому відмінність симетричних і асиметричних криптосистем?

9) Які корисні особливості цифрового підпису?

10) Що є об'єктами вивчення теорії криптографічних протоколів?

Тема №8 (2 години)
Криптографія і гіпотеза P≠NP

Мета: Ознайомитися з гіпотезою P≠NP

План

1. Поняття поліному
2. Правила машини Тюрінга
3. Однобічна функція та її ознаки

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Як правило, знайомство математиків-неспеціалістів з теорією складності обчислень обмежується класами P і NP і знаменитою гіпотезою P≠NP.

Нагадаємо коротко необхідні відомості з теорії складності обчислень. Хай $\sum^*$ – множина всіх кінцевих рядків в двійковому алфавіті $\sum = \{0,1\}$. Підмножини $L \sum^*$ теорії складності прийнято називати мовами. Говорять, що машина Тюрінга M працює за поліноміальний час (або просто, що вона поліноміальна), якщо існує поліном *p* такий, що на будь-якому вхідному слові довжини *n* машина *M* зупиняється після виконання не більш, ніж *p(n)* операцій. Машина Тюрінга *M* розпізнає (інший термін - приймає) мову L, якщо на всякому вхідному слові $x \in L$ машина *M* зупиняється в приймаючому стані, а на всякому слові $x \notin L$ - в тому, що відкидає. Клас *P* - це клас всіх мов, що розпізнаються машинами Тюрінга, що працюють за поліноміальний час. Функція f: $\sum^* \to \sum^*$ вичислюється за поліноміальний час, якщо існує поліноміальна машина Тюрінга така, що якщо на вхід їй подано слово $x \in X$, то момент зупинки на стрічці буде записано значення *f(x)*. Мова *L* належить класу *NP*, якщо о існують предикат P(x,y): $\sum^* \times \sum^* = \{0,1\}$ обчислюваний за поліноміальний час, і поліном *p* такі, що L = { x|зуP(x,y)&|y| ≤ p(|x|)}. Таким чином, мова L, належить *NP*, якщо для всякого слова з *L*, довжиною *n* можна вгадати деякий рядок поліноміальний від *n* довжини і потім за допомогою предиката *P* переконатися в правильності здогадки. Ясно, що P∈NP. Чи є це включення строгим одне з найвідоміших невирішених завдань математики. Більшість фахівців вважають, що воно строге (так звана гіпотеза PNP). У класі NP виділений підклас максимально складних мов, що називаються NP- повними: будь-яка NP-повна мова, що розпізнається за поліноміальний час тоді і тільки тоді, коли *P=np*. Для подальшого нам буде потрібно ще поняття імовірнісної машини Тюрінга. У звичайних машинах Тюрінга (їх називають детермінованими, щоб відрізнити від імовірнісних) новий стан, в який машина переходить на черговому кроці, повністю визначається поточним станом і тим символом, який оглядає головка на стрічці. У імовірнісних машинах новий стан може залежати ще і від випадкової

величини, котра приймає значення 0 і 1 з вірогідністю 1/2 кожне. Альтернативно можна вважати, що імовірнісна машина Тюрінга має додаткову випадкову стрічку, на якій записаний нескінченний двійковий випадковий рядок. Випадкова стрічка може читатися тільки в одному напрямі і перехід в новий стан може залежати від символу, що оглядає на цій стрічці.

Розглянемо тепер наступне звичайне питання: чи немає гіпотеза P-NP необхідною і достатньою умовою для існування стійких криптографічних схем?

Необхідність, і справді, у багатьох випадках майже очевидна. Повернемося на приклад 1. Визначимо наступну мову L = {(K_1,d,i)| існує повідомлення *m* таке, що $E_{k1}(m) = d$ і його *i*-ий біт рівний 1}.

Ясно, що $L \in NP$: замість описаного у введенні повного перебору можна просто вгадати відкритий текст *m* і перевірити за поліноміальний час, що $E_{k1}(m) = d$ і *i*-ий *m* рівний 1. Якщо так, то вхідне слово (K_1,d,i) приймається, інакше - відкидається.

У припущенні $P=np$ існує детермінований поліноміальний алгоритм, що розпізнає мову *L*. Знаючи K_1 і *d*, за допомогою цього алгоритму можна послідовно, по біту, обчислити відкритий текст *m*. Тим самим криптосистема нестійка.

Той же підхід: вгадати секретний ключ і перевірити (за поліноміальний час) правильність здогадки, застосовний в принципі і до інших криптографічних схем. Проте, в деяких випадках виникають технічні труднощі, пов'язані з тим, що за інформацією, яка є у супротивника, шукана величина (відкритий текст, секретний ключ і т. і.) відновлюється неоднозначно.

Що ж до питання про достатність припущення PNP, то тут напрошується наступний підхід: вибрати яку-небудь *Np*-повне завдання і побудувати на її основі криптографічну схему, завдання злому якої (тобто завдання, що стоїть перед супротивником) було б *NP-повною*. Такі спроби робилися на початку 80-х років, особливо відносно криптосистем з відкритим ключем, але до успіху не привели.

Результатом всіх цих спроб стало усвідомлення наступного факту: навіть якщо P≠NP, то будь-яка NP-повне завдання може виявитися важким лише на деякій нескінченній послідовності вхідних слів. Іншими словами, у визначення класу NP закладена міра складності «у гіршому разі». Для стійкості ж криптографічної схеми необхідно, щоб завдання супротивника було складним «майже усюди». Таким чином, стає ясно, що для криптографічної стійкості необхідно істотно сильніше припущення, ніж P≠NP. А саме, припущення про існування односторонніх функцій.

Кажучи неформально, одностороння функція - це ефективно обчислювана функція, для завдання інвертування якої, не існує ефективних алгоритмів. Під інвертуванням розуміється масове завдання знаходження по заданому значенню функції одного (будь-якого) значення з прообразу (відмітимо, що зворотна функція, взагалі кажучи, може не існувати).

Оскільки поняття односторонньої функції - центральне в математичній криптографії, нижче ми даємо його формальне визначення. Хай $\sum^n = \{0,1\}^n$ - безліч всіх двійкових рядків довжини *n*. Під функцією *f* ми розуміємо сімейство $\{f_n\}$, де f_n :

$\sum^n \to \sum$, $m=m(n)$. ДЛЯ простоти викладу ми припускаємо, що n пробігає весь натуральний ряд і що кожна з функцій f_n усюди визначена.

Функція f називається чесною, якщо існує поліном q такий, що $n \le q(m(n))$ для всіх n.

Визначення 1. Чесна функція f називається односторонньою, якщо

1. Існує поліноміальний алгоритм, який для всякого x обчислює $f(x)$.
2. Для будь-якої поліноміальної імовірнісної машини Тюрінга A виконане наступне. Хай рядок x вибраний наугад з множини $\sum^n$ (позначається $x \in \sum^n$). Тоді для будь-якого полінома p і всі досить великих n

$Pr\{f(A(f(x))) = f(x)\} < 1/p(n)$.

Вірогідність тут визначається випадковим вибором рядка X і випадковими величинами, які A використовує в своїй роботі.

Умова 2 якісно означає наступне. Будь-яка поліноміальна імовірнісна машина Тюрінга A може по даному y знайти x з рівняння $f(x)=y$ лише знехтує малою вірогідністю.

Відмітимо, що вимогу чесності не можна опустити. Оскільки довжина вхідного слова $f(x)$ машини A рівна f_n, їй може не вистачити поліноміального (від f_n) часу просто на виписування рядка x, якщо f дуже сильно «стискає» вхідні значення.

Ясно, що з припущення про існування односторонніх функцій витікає, що P≠NP. Проте, не виключена наступна ситуація: P≠NP, але односторонніх функцій немає.

Існування односторонніх функцій є необхідною умовою для стійкості багатьох типів криптографічних схем. У деяких випадках цей факт встановлюється досить просто. Звернемося знову до прикладу 1. Розглянемо функцію f таку, що $f(r)= K_1$. Вона вичисляється за допомогою алгоритму G за поліноміальний час. Покажемо, що якщо f - не одностороння функція, то криптосистема нестійка. Припустимо, що існує поліноміальний імовірнісний алгоритм A, який інвертує f з вірогідністю принаймні $1/p(n)$ для деякого полінома p. Тут n - довжина ключа K_1. Супротивник може подати на вхід алгоритму A ключ K_1 і набути з вказаною вірогідністю деякого значення $r1$ з прообразу. Далі супротивник подає r' на вхід алгоритму G і отримує пару ключів $(K1,K'_1)$. Хоча K'_2 не обов'язково співпадає з K_2, проте, за визначенням криптосистеми DK'_2. $(E_{k1}(m)) = m$ для будь-якого відкритого тексту m. Оскільки K'_2 знайдений з вірогідністю принаймні $1/p(n)$, яка в криптографії не вважається малою, криптосистема нестійка.

Для інших криптографічних схем подібний результат доводиться не так просто. У роботі Імпальяццо і Лубі доведена необхідність односторонніх функцій для існування цілого ряду стійких криптографічних схем. Зі всього сказаного виходить, що припущення про існування односторонніх функцій є найслабкішим криптографічним припущенням, яке може виявитися достатнім для доказу існування стійких криптографічних схем різних типів. На з'ясування того, чи є ця умова і в справді достатньою, направлені значні зусилля фахівців. Трудність завдання побудови криптографічних схем з односторонніх функцій можна пояснити на наступному

прикладі. Хай *f* - одностороння функція і нам потрібно побудувати криптосистему з секретним ключем. У такій криптосистемі є тільки один ключ - секретний, який відомий і відправникові, і одержувачеві шифрованого повідомлення. Алгоритми шифрування і дешифрування *DK* обидва залежать від цього секретного ключа *K* і такі, що *DK (EK (m)) = m* для будь-якого відкритого тексту *m*. Ясно, що якщо криптограму *d* повідомлення *m* обчислювати як *d = f(m)* у то супротивник, *d*, що перехопив, може обчислити *m* лише нехтує малою вірогідністю. Але по-перше, незрозуміло, яким чином зможе відновити повідомлення *m* з криптограми законний одержувач? По-друге, з того, що функція *f* одностороння слідує лише, що противник не може обчислити все повідомлення цілком. А це - вельми низький рівень стійкості. Бажано, щоб супротивник, що знає криптограму *d*, не міг обчислити жодного біта відкритого текста.

На даний момент доведено, що існування односторонніх функцій є необхідною і достатньою умовою для існування стійких криптосистем з секретним ключем, а також стійких криптографічних протоколів декількох типів, включаючи протоколи електронного підпису. З іншого боку, є результат Імпальяццо і Рудіха, який є достатньо сильним аргументом на користь того, що для деяких типів криптографічних схем (включаючи протоколи розподілу ключів типу Діффі-Хеллмана) потрібні сильніші припущення, ніж припущення про існування односторонніх функцій. На жаль, цей результат дуже складний, щоб його можна було роз'яснити в даному розділі.

Питання:

1) Що таке одностороння функція?
2) В якому випадку функція *f* називається чесною?
3) В чому полягає трудність побудови криптографічних схем з односторонніх функцій?
4) Що розуміється під інвертуванням?
5) Яке припущення є найслабкішим криптографічним припущенням, яке може виявитися достатнім для доказу існування стійких криптографічних схем різних типів.

Тема №9 (2 години)

Псевдовипадкові генератори. Докази з нульовим розголошуванням.

Мета: ознайомитися з псевдовипадковими генераторами

План

1. Основні типи погроз безпеки
2. Вірогідність та поняття генератора
3. Робота псевдовипадкового генератора

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні

поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Істотний недолік шифру Вернама полягає в тому, що ключі одноразові. Чи можна позбавитися від цього недоліку за рахунок деякого зниження стійкості? Один із способів вирішення цієї проблеми полягає в наступному. Відправник і одержувач мають загальний секретний ключ *K* довжини *n* і за допомогою деякого достатньо ефективного алгоритму *g* генерують з нього послідовність *r = g(K)* довжини *q(n),* де q деякий поліном. Така криптосистема (позначимо її $C_г$) дозволяє шифрувати повідомлення *m* (або сукупність повідомлень) довжиною до *q(n)* бітів по формулі d=r⊙m, де ⊙ порозрядне складання бітових рядків по модулю 2. Дешифрування виконується по формулі m=d⊙r. З результатів Шеннона витікає, що така криптосистема не може бути абсолютно стійкою, тобто стійкою проти будь-якого супротивника (у чому, втім, не важко переконатися і безпосередньо). Але що буде, якщо потрібно захищатися тільки від поліноміально обмеженого супротивника, котрий може атакувати криптосистему лише за допомогою поліноміальних імовірнісних алгоритмів? Яким умовам повинні задовольняти послідовність *r* і алгоритм *g,* щоб криптосистема $C_г$ була стійкою? Пошуки відповідей на ці питання привели до появи поняття псевдовипадкового генератора, яке було введено Блюмом і Мікалі.

Нехай g: {0,1}*n*{0,1} *q(n)* функція, що обчислюється за поліноміальній (від *n*) час. Така функція називається генератором. Інтуїтивно, генератор *g* є псевдовипадковим, якщо послідовності, що породжуються ним, невідмітні ніяким поліноміальним імовірнісним алгоритмом від випадкових послідовностей тієї ж довжини *q(n)*. Формально цей об'єкт визначається наступним чином.

Нехай *A* поліноміальна імовірнісна машина Тюрінга, котра отримує на вході двійкові рядки довжини *q(n)* і видає в результаті своєї роботи один біт. Нехай

$P_1(A,n) = Pr\{A(r) = 1 | r \in r \{0,1\}^{q(n)}\}$.

Вірогідність тут визначається випадковим вибором рядка *r* і випадковими величинами, які *A* використовує в своїй роботі. Нехай

$P_2(A,n) = Pr\{A(g(S)) = 1 | S \in_я \{0,1\}^n\}$.

Ця вірогідність визначається випадковим вибором рядка *S* і випадковими величинами, які *A* використовує в своїй роботі. Підкреслимо, що функція *g* обчислюється детермінованим алгоритмом.

Визначення 2. Генератор *g* називається криптографічним стійким псевдовипадковим генератором, якщо для будь-якої поліноміальної імовірнісної машини Тюрінга *A*, для будь-якого полінома *p* і всіх достатньо великих *n*

$|P_1(A,n)-P_2\{A,n\}|<1/p(n)$.

Усюди нижче ми скорочено називатимемо криптографічні стійкі псевдовипадкові генератори просто псевдовипадковими генераторами. Таке

скорочення є загальноприйнятим в криптографічній літературі.

Неважко переконатися, що для існування псевдовипадкових генераторів необхідне існування односторонніх функцій. Насправді, сама функція g повинна бути односторонньою. Питання про те, чи є існування односторонніх функцій одночасно і достатньою умовою, довгий час залишалося відкритим. У 1982 р. було побудовано псевдовипадковий генератор, виходячи з припущення про існування односторонніх перестановок, тобто що зберігають довжину взаємо-однозначних односторонніх функцій. За цим послідувала серія робіт, в яких достатня умова все більш і більш ослаблялася, поки нарешті в 1989-1990 рр. Імпальяццо, Льовін і Лубі, Хостад не отримали наступний остаточний результат.

Теорема 1. Псевдовипадкові генератори існують тоді і тільки тоді, коли існують односторонні функції.

Псевдовипадкові генератори знаходять застосування не тільки в криптографії, але і в теорії складності, і в інших областях дискретної математики. Тут же розглянемо описану на початку даного розділу криптосистему C_r, що використовує псевдовипадковий генератор як алгоритм q. Перш за все, нам необхідно дати визначення стійкості криптосистеми з секретним ключем.

Нехай E_k - алгоритм шифрування криптосистеми з секретним ключем. Позначимо результат його роботи $d = E_k(m)$, тут K - секретний ключ довжиною n бітів, а m - відкритий текст довжиною $q(n)$ бітів. Через m позначається i-ий біт відкритого тексту. Нехай A - поліноміальна імовірнісна машина Тюрінга, яка отримує на вхід криптограму d і видає пару (i,σ), де $i \in \{1,...,q(n)\}$, $\sigma \in \{0,1\}$. Інтуїтивно, криптосистема є стійкою, якщо ніяка машина Тюрінга A не може обчислити жоден біт відкритого тексту з вірогідністю успіху, істотно більшою, ніж при простому вгадуванні.

Визначення 3. Криптосистема називається стійкою, якщо для будь-якої поліноміальної імовірнісної машини Тюрінга A для будь-якого полінома p і всі досить великих n

$$Pr\{A(d) = (i,\sigma) \;\&\; \sigma = m_i \mid K \in_R \{0,1\}^n,\ m \in_R \{0,1\}^{q(n)}\} < 1/2 + 1/p(n)$$

Ця вірогідність (усюди нижче скорочено ми її позначаємо просто Pr) визначається випадковим вибором секретного ключа K, випадковим вибором відкритого тексту m з множини всіх двійкових рядків довжини $q(n)$ і випадковими величинами, які A використовує в своїй роботі.

Покажемо, що криптосистема C_r з псевдовипадковим генератором як g є стійкою в сенсі даного визначення. Припустимо осоружне, тобто що існують поліноміальний імовірнісний алгоритм A і поліном p такі, що $Pr > 1/2 + 1/p(n)$ для нескінченно багато n. Розглянемо алгоритм B, який отримує на вході двійковий рядок r довжини $q(n)$, вибирає те $m \in_R \{0,1\}^{q(n)}$, обчислює $d = m \oplus r$ і викликає A як підпрограму, подаючи їй на вхід рядок d. Отримавши від A пару (r,σ), B перевіряє, чи дійсно $m_i = \sigma$ і якщо так, то видає 1, інакше - 0 і зупиняється. Видно, що B працює за поліноміальний (від n) час. Переконаємося, що алгоритм $У$ відрізняє псевдовипадкові рядки, породжені

генераторів g, від випадкових рядків довжини $q(n)$. Насправді, якщо рядки r, що поступають на вхід B, є випадковими, то d - криптограма шифру Вернама і згідно теоремі Шеннона, $Pr = 1/2$. Якщо рядки r породжені генератором g, то криптограми d мають такий же розподіл вірогідності, як в криптосистемі C_r, і, згідно припущенню, $Pr>1/2 + 1/p(n)$ для нескінченно багато n. Отримана суперечність з визначенням псевдовипадкового генератора доводить твердження про стійкість криптосистеми C_r.

Зрозуміло, стійкість криптосистеми з секретним ключем можна визначати різним чином. Наприклад, можна розглядати стійкість проти атаки з вибором відкритого тексту: супротивник може заздалегідь вибрати поліноміальну кількість відкритих текстів і отримати їх криптограми, після чого він отримає ту криптограму, по якій йому потрібно обчислити хоч би один біт відповідного відкритого тексту. Неважко впевнитися, що криптосистема C_r з псевдовипадковим генератором в якості g є стійкою і проти атаки з вибором відкритого текста.

Таким чином, ми переконалися, що за допомогою псевдовипадкових генераторів можна будувати стійкі криптосистеми. Основний напрям досліджень в даній області - пошук методів побудови ефективних псевдовипадкових генераторів на основі різних криптографічних припущень. Показником ефективності тут служить кількість операцій, що витрачаються на обчислення кожного очередного біта псевдовипадкової послідовності.

Питання:

1) Що називається криптографічним стійким псевдовипадковим генератором?
2) У якому випадку криптосистема називається стійкою?
3) Де знаходять застосування псевдовипадкові генератори?
4) Що необхідно для існування псевдовипадкових генераторів?
5) Ким було введено поняття псевдовипадкового генератора?

Тема №10 (2 години)

Блокові і потокові шифри

Мета: Ознайомитися з блоковими і потоковими шифрами

План

1. Основні типи та способи шифрування даних
2. характеристика блокових шифрів
3. характеристика потокових шифрів

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Проектування алгоритмів шифрування даних засноване на раціональному виборі функцій, що перетворюють початкові (незашифровані) повідомлення в шифротекст. Ідея безпосереднього застосування такої функції до всього повідомлення реалізується дуже рідко. Практично всі вживані криптографічні методи пов'язані з розбиттям повідомлення на велике число фрагментів (або знаків) фіксованого розміру, кожен з яких шифрується окремо. Такий підхід істотно спрощує завдання шифрування, оскільки повідомлення зазвичай мають різну довжину.

Розрізняють три основні способи шифрування: потокові шифри, блокові шифри й блокові шифри із зворотним зв'язком. Для класифікації методів шифрування даних слід вибрати деяку кількість характерних ознак, які можна застосувати для встановлення відмінностей між цими методами. Вважатимемо, що кожна частина або кожен знак повідомлення шифрується окремо в заданому порядку.

Можна виділити наступні характерні ознаки методів шифрування даних:

– Виконання операцій з окремими бітами або блоками. Відомо, що для деяких методів шифрування знаком повідомлення, над яким проводять операції шифрування, є окремий біт, тоді як інші методи оперують кінцевою множиною бітів, зазвичай званим блоком.

– Залежність або незалежність функції шифрування від результатів шифрування попередніх частин повідомлення.

– Залежність або незалежність шифрування окремих знаків від їх положення в тексті. У деяких методах знаки шифруються з використанням однієї й тієї ж функції незалежно від їх положення в повідомленні, а в інших методах, наприклад при потоковому шифруванні, різні знаки повідомлення шифруються з урахуванням їх положення в повідомленні. Цю властивість називають позиційною залежністю або незалежністю шифру.

– Симетрія або асиметрія функції шифрування. Ця важлива характеристика визначає істотну відмінність між звичайними симетричними (одноключевими)

криптосистемами й асиметричними (двохключовими) криптосистемами з відкритим ключем. Основна відмінність між ними полягає в тому, що в асиметричній криптосистемі знання ключа шифрування (або розшифрування) недостатньо для розкриття відповідного ключа розшифрування (або шифрування).

У табл. приведені типи криптосистем й їх основні характеристики.

Основные характеристики криптосистем

Тип криптосистемы	Операции с битами или блоками	Зависимость от предыдущих знаков	Позиционная зависимость	Наличие симметрии функции шифрования
Поточного шифрования	Биты	Не зависит	Зависит	Симметричная
Блочного шифрования	Блоки	Не зависит	Не зависит	Симметричная или несимметричная
С обратной связью от шифртекста	Биты или блоки	Зависит	Не зависит	Симметричная

Потокове шифрування полягає в тому, що біти відкритого тексту складаються по модулю 2 з бітами псевдовипадкової послідовності *K* достоїнств потокових шифрів відносяться висока швидкість шифрування, відносна простота реалізації і відсутність розмноження помилок. Недоліком є необхідність передачі інформації синхронізації перед заголовком повідомлення, яка повинна бути прийнята до розшифрування будь-якого повідомлення. Це обумовлено тим, що якщо два різні повідомлення шифруються на одному й тому ж ключі, то для рошифрування цих повідомлень потрібна одна й та ж псевдовипадкова послідовність. Таке положення може створити загрозу криптостійкості системи. Тому часто використовують додатковий, випадково вибраний ключ повідомлення, який передається на початку повідомлення й застосовується для модифікації ключа шифрування. У результаті різні повідомлення шифруватимуться за допомогою різних послідовностей, блоку відкритого тексту, і ніякі два блоки відкритого тексту не можуть бути представлені одним і тим же блоком шифротекста. Алгоритм блокового шифрування може використовуватися в різних режимах. Чотири режими шифрування алгоритму DES фактично застосовні до будь-якого блокового шифру: режим прямого шифрування або шифрування з використанням електронної книги кодів ECB (Electronic code Book), шифрування із зчепленням блоків шифротекста CBC (Cipher block chaining), шифрування із зворотним зв'язком по шифротексту CPB (Cipher feedback) і шифрування з зворотної зв'язком по виходу OFB (Output feedback).

Основною перевагою прямого блокового шифрування ECB є ті, що в спроектованій системі блокового шифрування невеликі зміни в шифротексті викликають великі й непередбачувані зміни у відповідному відкритому тексті, і навпаки. Разом з тим застосування блокового шифру в даному режимі має серйозні недоліки. Перший з них полягає в тому, що унаслідок детермінованого характера шифрування при фіксованій довжині блоку 64 біта можна здійснити криптоаналіз

шифротекста «зі словником» в обмеженій формі. Це обумовлено тим, що ідентичні блоки відкритого тексту завдовжки 64 біта в початковому повідомленні представляються ідентичними блоками шифротекста, що дозволяє криптоаналітику зробити певні висновки про зміст повідомлення. Інший потенційний недолік цього шифру пов'язаний з розмноженням помилок. Результатом зміни тільки одного біта в прийнятому блоці шифротексту буде неправильне розшифрування всього блоку. Це, в свою чергу, приведе до появи спотворених бітів (від 1 до 64) у відновленому блоці вихідного тексту.

Через відмічені недоліки блокові шифри рідко застосовуються у вказаному режимі для шифрування довгих повідомлень. Проте у фінансових установах, де повідомлення часто складаються з одного або двох блоків, блокові шифри широко використовують у режимі прямого шифрування. Таке застосування зазвичай пов'язане з можливістю частої зміни ключа шифрування, тому ймовірність шифрування двох ідентичних блоків відкритого тексту на одному й тому ж ключі дуже мала.

Криптосистема з відкритим ключем також є системою блокового шифрування і повинна оперувати блоками досить великої довжини. Це обумовлено тим, що криптоаналітик знає відкритий ключ шифрування й міг би заздалегідь обчислити і скласти таблицю відповідності блоків відкритого тексту і шифротексту. Якщо довжина блоків мала, наприклад 30 бітів, то число можливих блоків не дуже велике (при довжині 30 бітів це 230 =~109) і може бути складена повна таблиця, що дозволяє моментально розшифрувати будь-яке повідомлення з використанням відомого відкритого ключа. Асиметричні криптосистеми з відкритим ключем детально розбираються в наступному розділі.

Найчастіше блокові шифри застосовуються в системах шифрування із зворотним зв'язком. Системи шифрування зі зворотнім зв'язком зустрічаються в різних практичних варіантах. Як і при блоковому шифруванні, повідомлення розбивають на ряд блоків, що складаються з m бітів. Для перетворення цих блоків в блоки шифротексту, які також складаються з m біт, використовуються спеціальні функції шифрування. Проте якщо в блоковому шифрі така функція залежить тільки від ключа, то в блокових шифрах із зворотним зв'язком вона залежить як від ключа, так і від одного або більше попередніх блоків шифротексту.

Практично важливим шифром із зворотним зв'язком є шифр із зчепленням блоків шифротексту CBC. У цьому випадку m біт попередніх шифротексту підсумовуються по модулю 2 з наступними m біт відкритого тексту, а потім застосовується алгоритм блокового шифрування під управлінням ключа для отримання наступного блоку шифротексту. Ще один варіант шифру зі зворотним зв'язком виходить із стандартного режиму CFB алгоритма DES, тобто режиму із зворотним зв'язком по шифротексту.

Перевагою криптосистем блокового шифрування із зворотним зв'язком є можливість застосування їх для виявлення маніпуляцій повідомленнями, вироблюваних активними перехоплювачами. При цьому використовується факт

розмноження помилок у таких шифрах, а також здатність цих систем легко генерувати код аутентифікації повідомлень. Тому системи шифрування зі зворотнім зв'язком використовують не тільки для шифрування повідомлень, але і для їх аутентифікації. Криптосистемам блокового шифрування із зворотним зв'язком властиві деякі недоліки. Основним з них є розмноження помилок, оскільки один помилковий біт при передачі може викликати ряд помилок у розшифрованому тексті. Інший недолік пов'язаний з тим, що розробка і реалізація систем шифрування із зворотним зв'язком часто виявляються важчими, ніж систем потокового шифрування.

На практиці для шифрування довгих повідомлень застосовують потокові шифри або шифри із зворотним зв'язком. Вибір конкретного типу шифру залежить від призначення системи і вимог, що до неї пред’являються.

Потокові шифри широко застосовуються для шифрування перетворених у цифрову форму мовних сигналів і цифрових даних, що вимагають оперативної доставки споживачеві інформації. До недавнього часу такі застосування були переважними для даного методу шифрування. Це обумовлено, зокрема, відносною простотою проектування й реалізації генераторів шифруючих послідовностей. Але найважливішим чинником, звичайно, є відсутність розмноження помилок у потоковому шифрі. Стандартним методом генерування послідовностей для потокового шифрування є метод, що вживається у стандарті шифрування DES у режимі зворотного зв'язку по виходу (режим OFB).

При блоковому шифруванні відкритий текст спочатку розбивається на рівні по довжині блоки, потім застосовується залежна від ключа функція шифрування для перетворення блоку відкритого тексту довжиною m біт у блок шифротексту такої ж довжини. Перевагою блокового шифрування є те, що кожен біт блоку шифротексту залежить від значень всіх бітів відповідного.

Питання:

1) Які характерні ознаки методів шифрування даних ви знаєте?
2) Які достоїнства потокового шифрування ви знаєте?
3) Де широко застосовуються потокові шифри?
4) Назвіть достоїнства блокового шифрування.
5) В чому полягає суть потокового шифрування?

Тема 11 (2 години)

Режим гамування

Мета: Ознайомитися з роботою режиму гамування

План

1. Основні поняття режиму гамування
2. Шифрування та дешифрування у режимі гамування

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Зашифрування відкритих даних у режимі гамування. Криптосхема, що реалізовує алгоритм зашифрування в режимі гамування показана на рис. 3.12. Відкриті дані розбивають на 64-розрядні блоки

$T_0^{(1)}, T_0^{(2)},\ldots, T_0^{(i)},\ldots, T_0^{(m)},$

де $T_0^{(i)}$ *i*-й 64-розрядний блок відкритих даних, i =1...*m, m* визначається об'ємом шифрованих даних.

Ці блоки по черзі зашифровуються в режимі гамування шляхом порозрядного складання по модулю 2 в суматорі $СМ_5$ з гаммою шифру $Г_ш$ яка виробляється блоками по 64 бита, тобто

$$Г_ш - (Г_ш^{(1)}, Г_ш^{(2)},\ldots, Г_ш^{(i)}, Г_ш^{(m)}),$$

де $Г_ш^{(i)}$ - *i*-й 64-розрядний блок, i = 1...m.

Число двійкових розрядів в блоці $T_0^{(m)}$ може бути менше 64, при цьому невикористана для зашифрування частина гами шифра з блоку $Г_ш^{(m)}$ відкидається.

Рівняння зашифрування даних в режимі гамування має вигляд

$$T_ш^{(i)} = T_0^{(i)} \oplus Г_ш^{(i)}$$

де $Г_ш(i) = A(У_{i-1} \boxplus C_2, Z_{i-1} \boxplus C_1)$, i =1…m; $T_ш(i)$; блоку зашифрованого тексту; A(*) - функція шифрування в режимі простій заміни; C_1 C_2 - 32-розрядні двійкові константи; Y_i, Z_i - 32-розрядні двійкові послідовності.

Величини Y_i, Z_i визначаються ітераційно у міру формування гами $Г_ш$ таким чином:

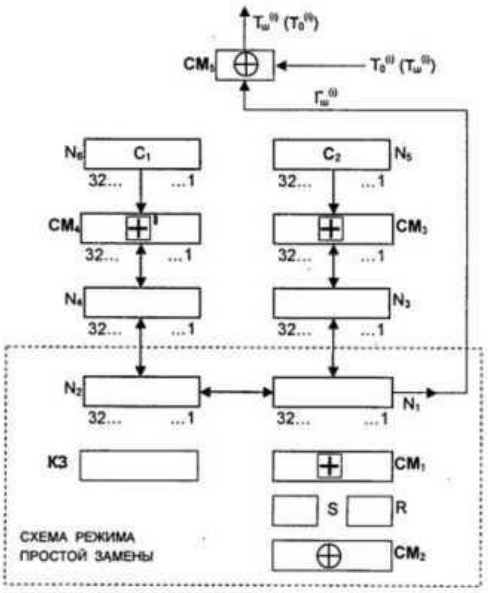

Рис.3.12. Схема реализации режима гаммирования

$(Yi, Zi) = A(\tilde{S})$

де *S*-синхропосилка (64-розрядна двійкова послідовність)

$(Y_i, Z_i)=(У_{i-1} \boxplus C_2, Z_{i-1} \boxplus C_1)$, i =1…m

Розглянемо реалізацію процедури зашифрованої в режимі гамування. У накопичувані N6 і N5 заздалегідь записані 32-розрядні двійкові константи C_1 і C_2, що мають наступні значення (у шестнадцатирічній формі):

C_1= $01010104_{(16)}$, C_2 = $01010101_{(16)}$.

У КЗУ вводиться 256 біт ключа; у накопичувані N1 і N2 - 64-разрядная двійкова послідовність (синхропосилка)

$\tilde{S} = (S_1, S_2, ..., S_{64})$

Синхропосилка *S* є початковим заповненням накопичувачів N1 і N2 для послідовного вироблення *m* блоків гамми шифру

Початкове заповнення накопичувана N1:

(S_{32}, S_{31},…, S_2, S_1);

32, 31,…, 2, 1 ←номер розряда N1

початкове заповнення накопичувана N2:

(S_{64}, S_{63},…, S_{34}, S_{33});

64, 63,…, 34, 33 ←номер розряда N2

Початкове заповнення N1 і N2 (синхропосилка *S*) зашифровується в режимі простої заміни. Результат за шифрування

$A(\tilde{S}) = (Y_0, Z_0)$

переписується в 32-розрядні накопичувані N3 і N4 так, що заповнення N1 переписується в N3, а заповнення N2 - в N4.

Заповнення накопичувана N4 підсумовують по модулю (2^{32} -1) в суматорі $СМ_4$ з 32-розрядною константою 3, з накопичувана N_6. Результат записується в N4. Заповнення накопичувана N3 сумується по модулю 2 в суматорі $СМ_3$ з 32-розрядною константой С2 з накопичувана N5. Результат записується в N3. Заповнення N3 переписують в N1, а заповнення N4 - в N_2. При цьому заповнення N3, N4 зберігаються. Заповнення накопичувачів N1 і N2 зашифровується в режимі простої заміни.

Отримане в результаті зашифрування заповнення накопичувачів N1, N2 утворює перший 64-розрядний блок гамми шифру

$Г_ш^{(1)} = (У_1^{(1)}, У_2^{(1)},...У_{63}^{(1)},У_{64}^{(1)})$, який підсумовують порозрядний по модулю 2 в суматорі СМБ 3 першим 64-розрядним блоком відкритих даних

В результаті підсумовування по модулю 2 значень $Г_ш^{(1)}$ і $Т_0^{(1)}$ отримують перший 64-розрядний блок зашифрованих даних:

Для отримання наступного 64-розрядного блоку гамми шифра $Г_ш^{(2)}$ заповнення N4 підсумовується по модулю (2^{32}-1) в суматорі $СМ_4$ з константою $С_1$ з N6. Результат записується в N4. Заповнення N3 підсумовується по модулю 2 в суматорі $СМ_3$ з константою $С_2$ з N5. Результат записується в N3. Нове заповнення N3 переписується в N1, а нове заповнення N4 - в N2 при цьому заповнення N3 і N4 зберігають. Заповнення N1, N2 зашифровують в режимі простої заміни.

Отримане в результаті зашифрування заповнення накопичувачів N1 і N2 утворює другий 64-розрядний блок гамми шифру $Г_ш^{(2)}$, який підсумовується порозрядний по модулю 2 в суматорі $СМ_5$ з другим блоком відкритих даних $Т_0^{(2)}$

$Т_ш^{(2)}= Г_ш^{(2)} \oplus Т_0^{(2)}$

Аналогічно виробляються блоки гамми шифру $Г_ш^{(3)}$ $Г_ш^{(4)}$ .... $Г_ш^{(m)}$ і зашифровуються блоки відкритих даних $Т_0^{(3)}$, $Т_0^{(4)}$ ...,$Т_0^{(m)}$

У канал зв'язку або пам'ять ЕОМ передаються синхропосилка *S* і блоки зашифрованих даних

$Т_ш^{(1)}, Т_ш^{(2)},\ldots, Т_ш^{(m)},$

Розшифрування в режимі гамування.

При розшифруванні кріптосхема має той же вигляд, що і при шифруванні.

Рівняння розшифруванні:

$$Т_0^{(i)}=Т_ш^{(i)}\oplus Г_ш^{(i)}= Т_ш^{(i)}\oplus A(Y_{i-1} \boxplus C_2, Z_{i-1} \boxplus C_1) \quad i = 1 \ldots m.$$

Слід зазначити, що розшифрування даних можливо тільки за наявності синхропосилки, яка не є секретним елементом шифру і може зберігатися в пам'яті ЕОМ або передаватися по каналах зв'язку разом із зашифрованими даними.

Розглянемо реалізацію процедури розшифрування. У КЗУ вводять 256 біт ключа, за допомогою якого здійснюється шифрування даних $Т_0^{(1)}$,$Т_0^{(2)}$,…, $Т_0^{(m)}$. У накопичувані N1 і N2 вводиться синхропосилка, і здійснюється процес вироблення *m*

блоків гамми шифру $Г_ш^{(1)}$,$Г_ш^{(2)}$…,$Г_ш^{(m)}$. Блоки зашифрованих даних $Т_ш^{(1)}$,$Т_ш^{(2)}$…$Т_ш^{(m)}$ підсумовуються порозрядний по модулю 2 в суматорі $СМ_5$ з блоками гамми шифру $Г_ш^{(1)}$…$Г_ш^{(m)}$. В результаті виходять блоки відкритих даних

$Т_0^{(1)}$, $Т_0^{(2)}$,…, $Т_0^{(m)}$

при цьому $Т_0^{(m)}$ може містити менше 64 розрядів.

Питання щодо самоконтролю:

1) Яким чином реалізується процедура зашифрування даних?

Тема 12 (2 години)
Основні режими роботи алгоритму DES

Мета: Ознайомитись з основними режимами роботи алгоритму DES

План

1. Поняття алгоритму DES
2. Режими алгоритму DES
3. Загальна характеристика режимів

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Алгоритм DES цілком підходить як для шифрування, так і для аутентифікації даних. Він дозволяє безпосередньо перетворювати 64- бітовий вхідний відкритий текст в 64-бітовий вихідний шифрований текст, проте дані рідко обмежуються 64 розрядами.

Щоб скористатися алгоритмом DES для вирішення різноманітних криптографічних завдань, розроблені чотири робочих режими:

- електронна кодова книга ECB (Electronic Code Book);
- зчеплення блоків шифру CBC (Cipher Block Chaining);
- зворотний зв'язок по шифротексту CFB (Cipher Feed Back);
- зворотний зв'язок по виходу OFB (Output Feed Back)

Режим “Електронна кодова книга”. Довгий файл розбивають на 64-бітові відрізки (блоки) по 8 байтів. Кожен з цих блоків шифрують незалежно з використанням одного і того ж ключа шифрування (рис. 3.5).

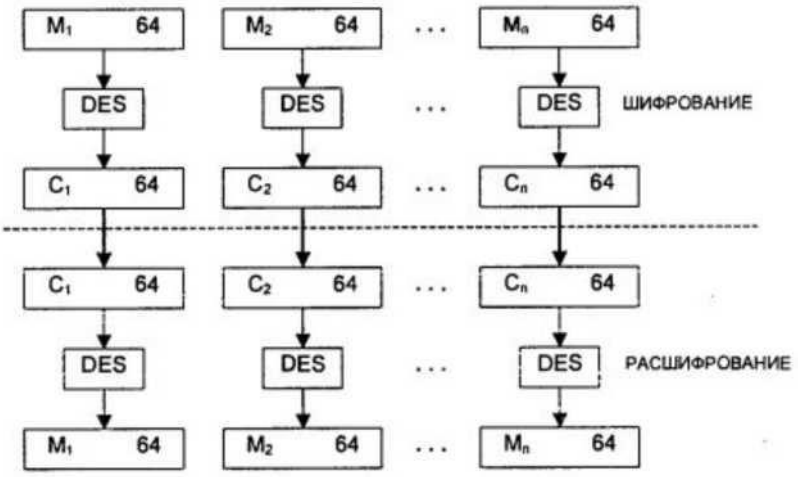

Рис. 3.5. Схема алгоритма DES в режиме электронной кодовой книги

Основна перевага - простота реалізації. Відносно слабка стійкість проти кваліфікованих криптоаналітиків. Через фіксований характер шифрування при обмеженій довжині блоку 64 біта можливо проведення криптоаналіза "зі словником". Блок такого розміру може повторитися в повідомленні внаслідок великої надмірності в тексті на природній мові. Це приводить до того, що ідентичні блоки відткритого тексту в повідомленні будуть представлені ідентичними шифротексту, що дає криптоаналітику деяку інформацію про зміст повідомлення.

Режим "Зчеплення блоків шифру". У цьому режимі початковий файл M розбивається на 64-бітові блоки: M = Ml M2... Mn. Перший блок M, складається по модулю 2 з 64-бітовим початковим вектором IV, який міняється щодня і тримається в секреті (рис. 3.6). Отримана сума потім шифруется з використанням ключа DES, відомого і відправникові, і отримувачу інформації. Отриманий 64-бітовий шифр C1 складається по модулю 2 з другим блоком тексту, результат шифрується і виходить другий 64-бітовий шифр C2, і так далі. Процедура повторюється до тих пір, поки не будуть оброблені всі блоки тексту.

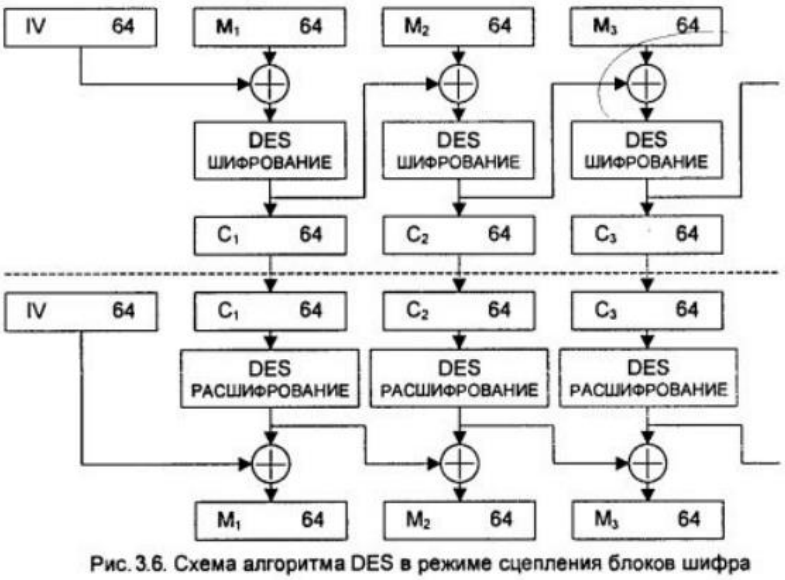

Рис. 3.6. Схема алгоритма DES в режиме сцепления блоков шифра

Таким чином, для всіх i = 1...*n* (*n* - число блоків) результат шифрування C_i,

визначається таким чином: C_i =DES($M_1 \odot C_i$-1), де C_0 = IV - початкове значення шифру, рівне початковому вектору (вектору ініціалізації).

Очевидно, що останній 64-бітовий блок шифротексту є функцією секретного ключа, початкового вектора й кожного біта відкритого тексту незалежно від його довжини. Цей блок шифротексту називають кодом аутентифікації повідомлення (КАП).

Код КАП може бути легко перевірений одержувачем, що володіє секретним ключем і початковим вектором, шляхом повторення процедури, виконаної відправником. Сторонній, проте, не може здійснити генерацію КАП, який сприйнявся б отримувачем як справжній, щоб додати його до помилкового повідомлення, або відокремити КАП від дійсного повідомлення для використання його зі зміненим або помилковим повідомленням. Перевага даного режиму в тому, що він не дозволяє накопичуватися помилкам при передачі.

Блок М1, є функцією тільки C_l-1 і C_i. Тому помилка при передачі приведе до втрати тільки двох блоків початкового тексту.

Режим “Зворотний зв'язок по шифру”. У цьому режимі розмір блоку може відрізнятися від 64 біт (рис. 3.7). Файл, що підлягає шифруванню (розшифрування), зчитується послідовними блоками завдовжки *K* бітів (*K* =1... 64).

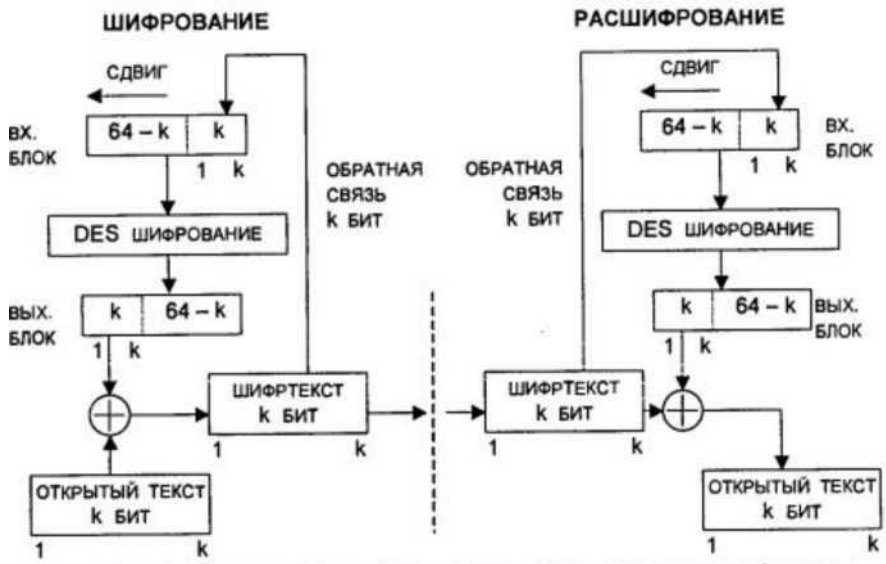

Рис. 3.7. Схема алгоритма DES в режиме обратной связи по шифртексту

Вхідний блок (64-бітовий регістр зрушення) спочатку містить вектор ініціалізації, вирівняний по правому краю.

Припустимо, що в результаті розбиття на блоки ми отримали *n* блоків завдовжки *K* бітів кожен (залишок дописується нулями або пропусками). Тоді для будь-якого i = 1... *n* блок шифротексту

$C_i = M_I \oplus P_{i-1}$,

де P_{i-1}, позначає *K* старших бітів попереднього зашифрованного блоку.

Оновлення зсувного регістра здійснюється шляхом видалення його старших *K*

бітів і запису 3; у регістр. Відновлення зашифрованих даних також виконується відносно просто: P_{i-1} і C_i обчислюються аналогічним чином *i*

$M_i = C_i \oplus P_{i-1}$

Режим "Зворотний зв'язок по виходу". Цей режим теж використовує змінний розмір блоку й зсувний регістр, що ініціалізувався так саме, як у режимі СРВ, а саме-вхідний блок спочатку містить вектор ініціалізації IV, вирівняний по правому краю (рис.3.8). При цьому для кожного сеансу шифрування даних необхідно використовувати новий початковий стан регістра, який повинен пересилатися по каналу відкритим текстом.

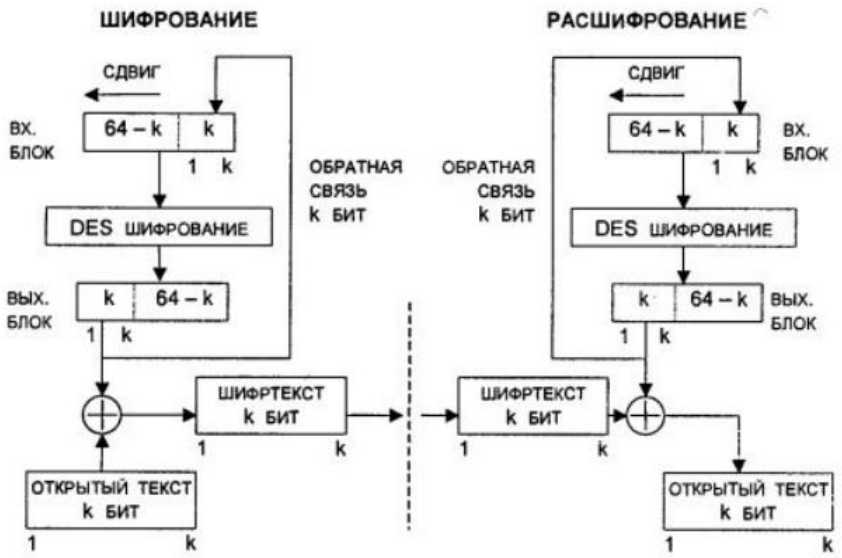

Рис. 3.8. Схема алгоритма DES в режиме обратной связи по выходу

$M = M_1M_2 \ldots M_n$ для всіх I=1.n

$C_i = M_i \oplus P_i$,

де P_i - старші *K* бітів операції DES (C_{i-1}).

Відмінність від режиму зворотного зв'язку по шифротексту полягає в методі оновлення зсувного регістра.

Це здійснюється шляхом відкидання старших *K* бітів і дописування справа P_I

Питання щодо самоконтролю:

1) Що ви знаєте про алгоритм DES?
2) Що ви знаєте про режим "Електронна кодова книга”?
3) Що ви знаєте про режим "Зчеплення блоків шифру "?
4) Що ви знаєте про режим " Зворотний зв'язок по шифру "?
5) Що ви знаєте про режим " Зворотний зв'язок по виходу "?

Тема № 13 (2 години)
Комбінування блокових алгоритмів

Мета: Ознайомитись з комбінуванням блокових алгоритмів

План

1. Основні типи блокових алгоритмів
2. Комбінування блокових алгоритмів

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

В даний час блоковий алгоритм DES вважається відносно безпечним алгоритмом шифрування. Він піддавався ретельному криптоаналізу протягом 20 років, і самим практічним способом його злому є метод перебору всіх можливих варіантів ключа. Ключ DES має довжину 56 біт, тому існує 256 можливих варіантів такого ключа. Якщо припустити, що суперкомп'ютер може випробувати мільйон варіантів ключа за секунду, то буде потрібно 2285 років для знаходження правильного ключа. Якби ключ мав довжину 128 біт, то знадобилось би 10 років (для порівняння: вік Всесвіту біля 10^{10}лет).

Неважко уявити собі, що при постійному прогресі можливостей комп'ютерної техніки недалеким є той час, коли машини пошуку ключа DES методом повного перебору стане економічно для могутніх у фінансовому відношенні державних і комерційних організацій.

Виникає питання: чи не можна використовувати DES як будівельного блоку для створення іншого алгоритму з довшим ключем?

У принципі існує багато способів комбінування блочних алгоритмів для отримання нових алгоритмів. Одним з таких способів комбінування є багатократне шифрування, тобто використання блокового алгоритму кілька разів з різними ключами для шифрування одного й того ж блоку відкритого тексту. Двократне шифрування блоку відкритого тексту одним і тим же ключем не приводити до позитивного результату. При використанні одного й того ж алгоритму таке шифрування не впливає на складність криптоаналітичної атаки повного перебору.

Розглянемо ефективність двократного шифрування блоку відкритого тексту за допомогою двох різних ключів. Спочатку шифрують блок *P* ключем *K1*, шифротекст, що вийшов *Ekl(P)* шифрують ключем *K2*. У результаті двократного шифрування одержують криптограму $C = E_{к2}(E_{к1},(P))$.

Розшифрування є зворотним процесом:

$$P = D_{K1},(D_{K2}(C)).$$

Якщо блоковий алгоритм володіє властивостями групи, то завжди знайдеться такий ключ K_3, що

$$C=E_{K2}(E_{K1}(P))=E_{K3}(P).$$

Якщо ж блоковий алгоритм немає групи, то результуючий двократно

шифрований блок тексту виявиться набагато складнішим для злому методом повного перебору варіантів. Замість 2^n спроб, де n - довжина ключа в бітах, буде потрібно 2^{2n} спроб. Зокрема, якщо n = 64, то двократно зашифрований блок тексту зажадає 2^{128} спроб для знаходження ключа.

Проте Р. Меркль і м. Хеллман показали на прикладі DES, що, використовуючи метод "обміну часу на пам'ять" і криптоаналітичну атаку "зустріч посередині", можна зламати таку схему двократного шифрування за 2^{n+1} спроба. Хоча ця атака потребує дуже великого об'єму пам'яті (для алгоритму з 56-бітовим ключем буде потрібно 256 64бітових блоків або 10^{17} біт пам'яті).

Привабливішу ідею запропонував У. Тачмен. Суть цієї ідеї полягає в тому, щоб шифрувати блок відкритого тексту P три рази за допомогою двох ключів $К_1$ і $К_2$ (рис.3.9). Процедура шифрування:

$C = E_{K1}(D_{K2}(E_{K1}(P)))$,

тобто блок відкритого тексту P спочатку шифрується ключем K_1 потім розшифровується ключем K_2 і остаточно зашифровується ключем K_f.

Рис. 3.9. Схемы трехкратного применения алгоритма DES с двумя разными ключами

Цей режим іноді називають режимом EDE (encrypt-decrypt-encrypt). Введення в дану схему операції розшифрування D_{K2} дозволяє забезпечити сумісність цієї схеми зі схемою однократного використання алгоритму DES. Якщо в схемі трьократного використання DES вибрати всі ключі однаковими, то ця схема перетворюється на схему одноразового використання DES. Процедура розшифрування виконується в зворотному порядку:

$P = D_{K1}(E_{K2}(D_{K1}(C)))$,

тобто блок шифротексту спочатку розшифровується ключем $К_1$ і потім зашифровується ключем $К_2$ і остаточно розшифровується ключем $К_1$.

Якщо початковий блоковий алгоритм має n-бітовий ключ, то схема триразового шифрування має $2n$-бітовий ключ. Чередування ключів $К_1$ і $К_2$ дозволяє запобігти криптоаналітичній атаці "зустріч посередині". Дана схема приводиться в стандартах X9.17 і ISO 8732 як засіб поліпшення характеристик алгоритму DES. При триразовому шифруванні можна застосувати три різних ключа. При цьому зростає загальна довжина результуючого ключа. Процедури шифрування і розшифрування описуються

виразами:

$C = E_{K3}(D_{K2}(E_{K1}(P)))$.
$P = D_{K1}(E_{K2}(D_{K3}(C)))$.

Трьохключовий варіант має ще більшу стійкість. Очевидно, що якщо потрібно підвищити безпеку великого парку устаткування, що використовує DES, то набагато дешевше перемкнутися на схеми триразових DES, чим переходити на інший тип криптосхем.

Питання щодо самоконтролю:

1) Чи не можна використовувати DES як будівельного блоку для створення іншого алгоритму з довшим ключем?
2) Що ви знаєте про багатократне шифрування?
3) Що таке EDE?
4) Що ви знаєте про режим EDE?

Тема № 14 (2 години)
Алгоритм шифрування даних IDEA

Мета: Ознайомитись з алгоритмом шифрування даних IDEA

План

1. Поняття алгоритму IDEA
2. Схема шифрування за допомогою алгоритму IDEA

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Алгоритм IDEA (International Data Encryption Algorithm) є блоковим шифром. Він оперує 64-бітовими блоками відкритого тексту. Безперечною перевагою алгоритму IDEA є те, що його ключ має довжину 128 біт. Один і той же алгоритм використовується і для шифрування, і для дешифрування.

Перша версія алгоритму IDEA була запропонована в 1990 р.. її автори - Х. Лий і Дж. Мессі. Первинна назва алгоритму PES (Proposed Encryption Standard). Покращений варіант цього алгоритму, розроблений в 1991р.. отримав назву IPES (Improved Proposed Encryption Standard). В 1992 р. IPES змінив своє ім'я на IDEA. Як і більшість інших блокових шифрів, алгоритм IDEA використовує при шифруванні процеси змішування й розсіювання, причому всі процеси легко реалізуються апаратними й програмними засобами.

В алгоритмі IDEA використовуються наступні математичні операції:

– порозрядне складання по модулю 2 (операція "виключаюче АБО"); операція позначається як $\oplus$

– складання беззнакових цілих по модулю 2^{16} (модуль 65536); операция позначається як $\boxplus$;

– множення цілих по модулю $(2^{16} + 1)$ (модуль 65537), що розглядаються як беззнакові цілі, за винятком того, що блок з 16 нулів розглядається як 216; операція позначається як $\odot$.

Всі операції виконуються над 16-бітовими субблоками.

Ці три операції несумісні в тому сенсі, що:

• ніяка пара з цих трьох операцій не задовольняє ассоциатівному закону, наприклад $a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus c$

• ніяка пара з цих трьох операцій не задовольняє дистрибутивному закону, наприклад $a \boxplus (b \oplus c) \neq (a \boxplus b) \oplus (a \boxplus c)$

Комбінування цих трьох операцій забезпечує комплексне перетворення входу, істотно утрудняючи криптоаналіз IDEA у порівнянні з DES, який базується виключно на операції "виключає АБО".

Загальна схема алгоритму IDEA наведена на рис.3.10. 64-бітовий блок даних ділиться на чотири 16-бітові субблоки. Ці чотири субблоки стають входом у перший цикл алгоритму. Всього виконується вісім циклів. Між циклами другим й третім субблоки міняються місцями. У кожному циклі має місце наступна послідовність операцій:

(1) $\odot$ – умножение субблока X_1 и первого подключа.
(2) $\boxplus$ – сложение субблока X_2 и второго подключа.
(3) $\boxplus$ – сложение субблока X_3 и третьего подключа.
(4) $\odot$ – умножение субблока X_4 и четвертого подключа.
(5) $\oplus$ – сложение результатов шагов (1) и (3).
(6) $\oplus$ – сложение результатов шагов (2) и (4).
(7) $\odot$ – умножение результата шага (5) и пятого подключа.
(8) $\boxplus$ – сложение результатов шагов (6) и (7).
(9) $\odot$ – умножение результата шага (8) с шестым подключом.
(10) $\boxplus$ – сложение результатов шагов (7) и (9).
(11) $\oplus$ – сложение результатов шагов (1) и (9).
(12) $\oplus$ – сложение результатов шагов (3) и (9).
(13) $\oplus$ – сложение результатов шагов (2) и (10).
(14) $\oplus$ – сложение результатов шагов (4) и (10).

Виходом циклу є чотири субблоки, які отримують як результати виконання кроків (11), (12), (13) і (14). В завершенні циклу переставляють місцями два внутрішні субблоки (за виключенням останнього циклу), і в результаті формується вхід для наступного циклу.

Після восьмого циклу здійснюють завершальне перетворення виходу:

(1) $\odot$ – умножение субблока X_1 и первого подключа.
(2) $\boxplus$ – сложение субблока X_2 и второго подключа.
(3) $\boxplus$ – сложение субблока X_3 и третьего подключа.
(4) $\odot$ – умножение субблока X_4 и четвертого подключа.

Нарешті, ці результуючі чотири субблоки $Y \ldots Y_4$ знов об’єднують для отримання

блока шифротексту.

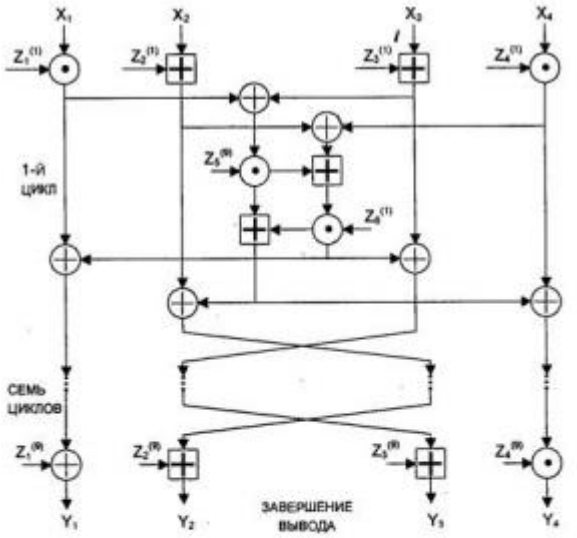

Рис. 3.10. Схема алгоритма IDEA (режим шифрования)

Створення підключень $\boldsymbol{Z}$, також відносно недоладно. Алгоритм використовує всього 52 підключа (по шість для кожного з восьми циклів і ще чотири для перетворення виходу). Спочатку 128-бітовий ключ ділять на вісім 16-бітових підключів. Це перші вісім підключів для алгоритму (шість підключів - для першого циклу й перші два для іншого циклу). Потім 128-бітовий ключ циклічно зрушується вліво на 25 біт і знову ділиться на вісім підключів. Перші чотири з них використовують в іншому циклі; останні в третьому циклі. Ключ знову циклічно зрушується вліво ще на 25 біт для отримання наступних восьми підключів і так далі, поки виконання алгоритму не завершитися.

Розшифрування здійснюють аналогічним чином, за виключенням того, що порядок використання підключів стає зворотним, причому ряд значень підключів замінюється на обернені значення. Підключі розшифрування є в основному або адитивними, або мультиплікативними зворотними величинами підключів шифрування (табл.).

Подключи шифрования и расшифрования алгоритма IDEA

Цикл	Подключи шифрования	Подключи расшифрования
1	$Z_1^{(1)}\ Z_2^{(1)}\ Z_3^{(1)}\ Z_4^{(1)}\ Z_5^{(1)}\ Z_6^{(1)}$	$Z_1^{(9)-1}\ -Z_2^{(9)}\ -Z_3^{(9)}\ Z_4^{(9)-1}\ Z_5^{(8)}\ Z_6^{(8)}$
2	$Z_1^{(2)}\ Z_2^{(2)}\ Z_3^{(2)}\ Z_4^{(2)}\ Z_5^{(2)}\ Z_6^{(2)}$	$Z_1^{(8)-1}\ -Z_3^{(8)}\ -Z_2^{(8)}\ Z_4^{(8)-1}\ Z_5^{(7)}\ Z_6^{(7)}$
3	$Z_1^{(3)}\ Z_2^{(3)}\ Z_3^{(3)}\ Z_4^{(3)}\ Z_5^{(3)}\ Z_6^{(3)}$	$Z_1^{(7)-1}\ -Z_3^{(7)}\ -Z_2^{(7)}\ Z_4^{(7)-1}\ Z_5^{(6)}\ Z_6^{(6)}$
4	$Z_1^{(4)}\ Z_2^{(4)}\ Z_3^{(4)}\ Z_4^{(4)}\ Z_5^{(4)}\ Z_6^{(4)}$	$Z_1^{(6)-1}\ -Z_3^{(6)}\ -Z_2^{(6)}\ Z_4^{(6)-1}\ Z_5^{(5)}\ Z_6^{(5)}$
5	$Z_1^{(5)}\ Z_2^{(5)}\ Z_3^{(5)}\ Z_4^{(5)}\ Z_5^{(5)}\ Z_6^{(5)}$	$Z_1^{(5)-1}\ -Z_3^{(5)}\ -Z_2^{(5)}\ Z_4^{(5)-1}\ Z_5^{(4)}\ Z_6^{(4)}$
6	$Z_1^{(6)}\ Z_2^{(6)}\ Z_3^{(6)}\ Z_4^{(6)}\ Z_5^{(6)}\ Z_6^{(6)}$	$Z_1^{(4)-1}\ -Z_3^{(4)}\ -Z_2^{(4)}\ Z_4^{(4)-1}\ Z_5^{(3)}\ Z_6^{(3)}$
7	$Z_1^{(7)}\ Z_2^{(7)}\ Z_3^{(7)}\ Z_4^{(7)}\ Z_5^{(7)}\ Z_6^{(7)}$	$Z_1^{(3)-1}\ -Z_3^{(3)}\ -Z_2^{(3)}\ Z_4^{(3)-1}\ Z_5^{(2)}\ Z_6^{(2)}$
8	$Z_1^{(8)}\ Z_2^{(8)}\ Z_3^{(8)}\ Z_4^{(8)}\ Z_5^{(8)}\ Z_6^{(8)}$	$Z_1^{(2)-1}\ -Z_3^{(2)}\ -Z_2^{(2)}\ Z_4^{(2)-1}\ Z_5^{(1)}\ Z_6^{(1)}$
Преобразование выхода	$Z_1^{(9)}\ Z_2^{(9)}\ Z_3^{(9)}\ Z_4^{(9)}$	$Z_1^{(1)-1}\ -Z_2^{(1)}\ -Z_3^{(1)}\ Z_4^{(1)-1}$

Для реалізації алгоритму IDEA було прийняте припущення, що нульовий субблок рівний 216= -1; при цьому мультиплікативна зворотна величина від 0 рівна 0. Обчислення значень мультиплікативних зворотних величин вимагає деяких витрат, але це доводиться тільки один раз для кожного ключа розшифрування.

Алгоритм IDEA може працювати в будь-якому режимі блокового шифру, передбаченому для алгоритму DES. Алгоритм IDEA володіє рядям переваг перед алгоритмом DES. Він значно безпечніше за алгоритм DES, оскільки 128-бітовий ключ алгоритму IDEA удвічі більше ключа DES. Внутрішня структура алгоритму IDEA забезпечує кращу стійкість до криптоаналізу. Існуючі програмні реалізації алгоритму IDEA приблизно удвічі швидше за реалізації алгоритму DES. Алгоритм IDEA шифрує дані на IBM Pc/486 із швидкістю 2,4 Мбіт/с. Реалізація IDEA на СБІС шифрує дані зі швидкістю 177 Мбіт/с при частоті 25 Мгц. Алгоритм IDEA запатентовано у Європі й США.

Питання щодо самоконтролю:

1) Які ви знаєте основні характеристики алгоритму IDEA?
2) Що ви знаєте про історію алгоритму IDEA?
3) Які математичні операції виконуються у алгоритмі IDEA?
4) Як відбуваються розшифрування?
5) Які переваги має алгоритм IDEA над алгоритмом DES?

Тема 15 (4 години)

Вітчизняний стандарт шифрування даних. Режим простої заміни. Режим гамування. Режим гамування зі зворотним зв'язком. Режим вироблення імітовставки.

Мета: Ознайомитися з вітчизняними стандартами шифрування даних

План

1. Вітчизняний стандарт шифрування даних
2. Режим простої заміни
3. Режим гамування

4. Режим гамування зі зворотним зв'язком
5. Режим вироблення імітовставки

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

У наший країні встановлений єдиний алгоритм криптографічного перетворення даних для систем обробки інформації в мережах ЕОМ, окремих обчислювальних комплексах й ЕОМ, котрий визначається ДЕРЖСТАНДАРТ 28147-89 [81]. Стандарт обов'язковий для організацій, підприємств й установ, що застосовують криптографічний захист даних, ЕОМ, що зберігаються і передаються у мережах, в окремих обчислювальних комплексах й ЕОМ.

Цей алгоритм криптографічного перетворення даних призначений для апаратної й програмної реалізації, задовольняє криптографічним вимогам і не накладає обмежень на ступінь секретності інформації, що захищається. Алгоритм шифрування даних є 64-бітовим блочний алгоритмом з 256-бітовим ключем.

При описі алгоритму використовуються наступні позначення:

L і R-послідовності бітів;

LR-конкатенація послідовностей L і R, у якої біти послідовності R слідують за бітами послідовності L;

$\oplus$ - операція побітового складання по модулю 2;

$\boxplus$ - операція складання по модулю 2^{32} двох 32-розрядних двійкових чисел;

$\boxplus'$ - операція складання двох 32-розрядних чисел по модулю 2^{32} -1.

Два цілі числа a, b де $0 < a, b < 2^{32} - 1$,

$a=(a_{32}, a_{31} \ldots a_2, a_1)$, $b=(b_{32}, b_{31} \ldots b_2, b_1)$

представлені в двійковому вигляді, тобто

$a \boxplus b = a+b$, якщо $a+b<2^{32}$,

$a \boxplus b = a+b-2^{32}$, якщо $a+b \geq 2^{32}$.

Правила підсумовування чисел по модулю 2-1:

$a \boxplus' b = a+b$, якщо $a+b<2^{32}-1$,

$a \boxplus' b = a+b-(2^{32}-1)$, якщо $a+b \geq 2^{32}-1$.

Алгоритм передбачає чотири режими роботи: шифрування даних у режимі простої заміни; шифрування даних у режимі гамування; шифрування даних у режимі гамування зі зворотним зв'язком; вироблення імітовставки.

Режим простої заміни

Для реалізації алгоритму шифрування даних в режимі простої заміни

використовується тільки частина блоків загальної криптосистеми (рис. 3.11). Позначення на схемі:

N1, N2 - 32-розрядні накопичувані;

$СМ_1$ - 32-розрядний суматор по модулю 2^{32}(⊞);,

$СМ_2$ - 32-розрядний суматор по модулю 2 (⊕);

R - 32-розрядний регістр циклічного зрушення;

КЗУ- ключовий пристрій, що запам'ятовує, на 256 біт, що складається з восьми 32-розрядних накопичувачів X_0, X_1 X_2.... X_7;

S-блок підстановки, що складається з восьми вузлів заміни (S-блоків заміни) S_1, S_2, S_3.... S_7, S_8.

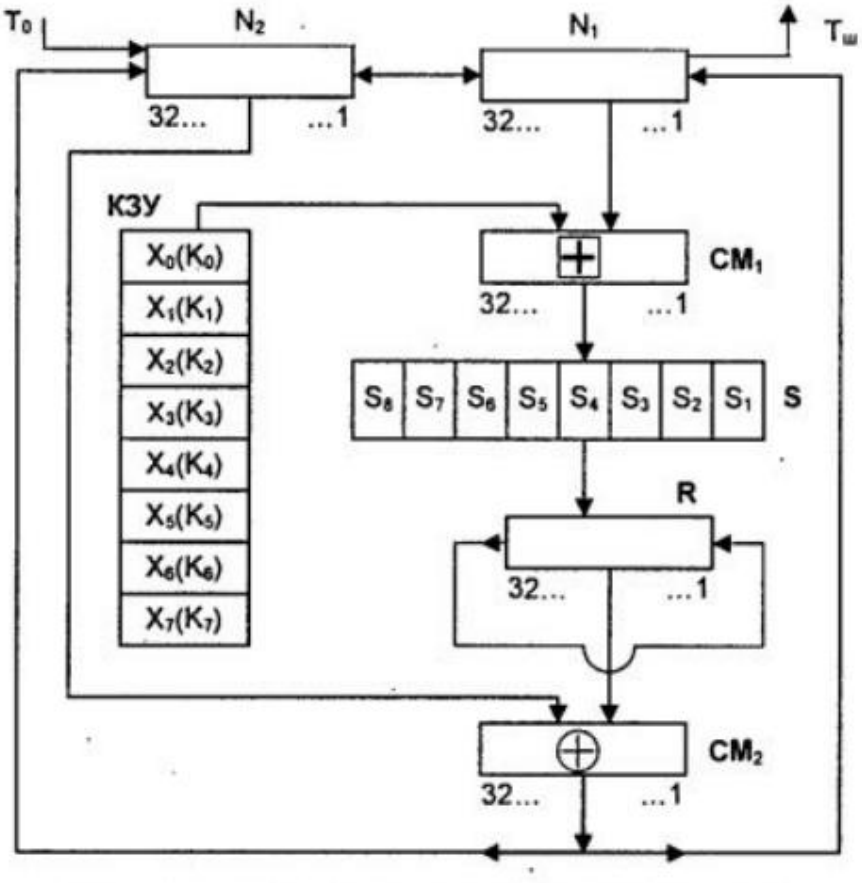

Рис. 3.11. Схема реализации режима простой замены

Зашифрування відкритих даних в режимі простої заміни. Відкриті дані, належні зашифруванню, розбивають на 64-розрядні-блоки T_0. Процедура зашифрування 64-розрядного блоку T_0 в режимі простій заміни включає 32 цикли 0 = 1... 32). У ключовий пристрій, що запам'ятовує, вводять 256 біт ключа *K* у вигляді восьми 32-розрядних підключів (чисел) K_i

$K=K_7K_6K_5K_4K_3K_2K_1K_0$.

Послідовність бітів блоку

$T_0 = (a_1(0), a_2(0) \ldots a_{31}(0), a_{32}(0), b_1(0), b_2(0) \ldots b_{31}(0), b_{32}(0))$

розбивають на дві послідовності по 32 бита: $b(0)$ а, де $b(0)$ - ліві або старші біти,

a(0) - праві або молодші біти.

Ці послідовності вводять в накопичувані N1 і N2 перед початком першого циклу зашифрування. В результаті початкове заповнення накопичувача N1

$a(0)=(a_{32}(0), a_{31}(0),\ldots,a_2(0), a_1(0))$,

32, 31, …, 2, 1 ← номер розряда N1

початкове заповнення накопичувана N2

$b(0)=(b_{32}(0), b_{31}(0),\ldots,b_2(0), b_1(0))$,

32, 31, …, 2, 1 ← номер розряда N2

Перший цикл (j=1) процедури зашифрування 64-розрядного блоку відкритих даних можна описати рівняннями:

$$\begin{cases} a(1) = f(a(0) \boxplus K_0) \oplus b(0), \\ b(1) = a(0). \end{cases}$$

Тут *a(1)* - заповнення N, після 1-го циклу зашифрування; *b*(1) - заповнення N2 після 1 -то циклу зашифрування; *f* - функція шифрування.

Аргументом функції *f* є сума по модулю 2 числа *a(0)* (початкового заповнення накопичувана N) і числа K_0 - підключа, що прочитується з накопичувана X_0 КЗУ. Кожне з цих чисел рівне 32 бітам.

Функція *f* включає дві операції над отриманою 32-разрядной сумою $(a(0) \boxplus K_0)$.

Перша операція називається підстановкою (заміною) і виконується блоком підстановки S. Блок підстановки S складається з восьми вузлів заміни (S-блоков заміни) S_1, S_2 ...,S_8 з пам'яттю 64 біт кожен. Що поступає на блок підстановки S 32-розрядний вектор розбивають на вісім послідовно розташованих 4-розрядних векторів, кожен з яких перетвориться в чотирьохрозрядний вектор відповідним вузлом заміни. Кожен вузол заміни можна представити у вигляді таблиці-перестановки шістнадцяти чотирьохрозрядних двійкових чисел в діапазоні 0000. .1111. Вхідний вектор указує адреса рядка в таблиці, а число в цьому рядку є вихідним вектором. Потім четирьохрозрядні вихідні вектори послідовно об'єднують в 32-розрядний вектор. Вузли заміни (таблиці-перестановки) представляють собою ключові елементи, які є загальними для мережі ЕОМ і рідко змінюються. Ці вузли заміни повинні зберігатися в секреті.

Друга операція - циклічне зрушення вліво (на 11 розрядів) 32-розрядного вектора, отриманого з виходу блоку підстановки S. Циклічне зрушення виконується регістром зрушення R.

Далі результат роботи функції шифрування *f* підсумовують порозрядний по модулю 2 в суматорі $СМ_2$ з 32-розрядним начальним заповненням *b*(0) накопичувана N_2. Потім отриманий на виході $СМ_2$ результат (значення *a(1)*) записують в накопичувач N1, а старе значення N1 (значення *a(0)*) переписують в накопичувач N_2 (значення *b(1)= a(0)*). Перший цикл завершений.

Подальші цикли здійснюються аналогічно, при цьому в другому циклі з КЗУ прочитують заповнення X_1 - підключ K_1 в третьому циклі - підключ K_2 і так далі у восьмому циклі - підключ K_7. У циклах з 9-го по 16-ий, а також в циклах з 17-го по 24-ий підключі з КЗУ прочитуються в тому ж порядку: K_0, K_1 K_2..., K_6, K_7. У останніх восьми циклах з 25-го по 32-й порядок прочитування підключів з КЗУ зворотний:

$$\begin{cases} a(j) = f(a(j-1) \boxplus K_{32-j}) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 25 \ldots 31,$$

K_7, K_6…K_2, K_1 K_0. Таким чином, при зашифруванні в 32 циклах здійснюється наступний порядок вибірки з КЗУ підключів:

K_0, $\mathbf{K_1}$, K_2, K_3, K_4, K_5, K_6, K_7, K_0, $\mathbf{K_1}$, K_2, K_3, K_4, K_5, K_6, K_7,

K_0, $\mathbf{K_1}$, K_2, K_3, K_4, K_5, K_6, K_7, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.

У 32-му циклі результат з суматора $СМ_2$ вводиться в накопичувач N_2, а в накопичувані N1, зберігається колишнє заповнення. Отримані після 32-го циклу зашифрування заповнення накопичувачів N1 і N_2 є блоком зашифрованих даних T_0.

Рівняння за шифрування в режимі простої заміни мають вигляд:

$$\begin{cases} a(j) = f(a(j-1) \boxplus K_{j-1(\mathrm{mod}8)}) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 1 \ldots 24,$$

$$\begin{cases} a(j) = f(a(j-1) \boxplus K_{32-j}) \oplus b(j-1) \\ b(j) = a(j-1) \end{cases} \quad \text{при } j = 25 \ldots 31,$$

$$\begin{cases} a(32) = a(31) \\ b(32) = f(a(31) \boxplus K_0) \oplus b(31) \end{cases} \quad \text{при } j = 32,$$

де $a(j) = (a_{32}(j), a_{31}(j),\ldots,a_1(j))$ - заповнення N_1, після j-го циклу зашифрування; $b(j) = -(b_{32}(j), b_{31}(j),\ldots,b_1(j))$ – заповнення N_2, після j-го циклу за шифрування, $j=1\ldots32$.

Блок зашифрованих даних $T_ш$ (64 розряди) виводиться з накопичувачів N1 N2 в наступному порядку: з розрядів 1...32 накопичувача N1. потім з розрядів 1...32 накопичувача N2 тобто починаючи з молодших розрядів:

$T_ш = (a_1(32), a_2(32),\ldots a_{32}(32), b_1(32), b_2(32),\ldots b_{32}(32))$.

Решта блоків відкритих даних зашифровується в режимі простої заміни аналогічно.

Розшифрування в режимі простої заміни. Криптосхема, що реалізовує алгоритм розшифрування в режимі простій заміни, має той же вигляд, що і при зашифруванні.

У КЗУ вводять 256 біт ключа, на якому здійснювалося зашифрування. Зашифровані дані, належні розшифруванню, розбиті на блоки $T_ш$ по 64 бита в кожному. Введення будь-якого блоку

$T_ш = (a_1(32), a_2(32),\ldots a_{32}(32), b_1(32), b_2(32) \quad ,\ldots b_{32}(32))$.

у накопичувані N1 і N2 проводять так, щоб початкове значення накопичувана N1 мало вигляд

$(a_{32}(32), a_{31}(32),\ldots,a_2(32), a_1(32))$,

32, 31, …, 2, 1 ← номер розряда N1

а початкове заповнення накопичувача N2:

($b_{32}(32), b_{31}(32),…,b_2(32), b_1(32)$),

32, 31, …, 2, 1 ← номер розряда N2

Розшифрування здійснюється по тому ж алгоритму, що і зашифрування, з тією зміною, що заповнення накопичувачів X_0, X_1 X_2.... X_7 прочитуються з КЗУ в циклах розшифрування в наступному порядку:

K_0, **K_1,** K_2, K_3, K_4, K_5, K_6, K_7, K_0, **K_1,** K_2, K_3, K_4, K_5, K_6, K_7,

K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0, K_7, K_6, K_5, K_4, K_3, K_2, K_1, K_0.

$$\begin{cases} a(32-j) = f(a(32-j+1) \boxplus K_{j-1}) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \text{ при } j = 1\ldots 8;$$

$$\begin{cases} a(32-j) = f(a(32-j+1) \boxplus K_{32-j(\mathrm{mod}\,8)}) \oplus b(32-j+1) \\ b(32-j) = a(32-j+1) \end{cases} \text{ при } j = 9\ldots 31;$$

$$\begin{cases} a(0) = a(1) \\ b(0) = f(a(1) \boxplus K_0) \oplus b(1) \end{cases} \text{ при } j = 32.$$

Отримані після 32 циклів роботи заповнення накопичувачів N1 і N2 утворюють блок відкритих даних

$T_0 = (a_1(0), a_2(0)\ldots a_{31}(0), a_{32}(0), b_1(0), b_2(0)\ldots b_{31}(0), b_{32}(0))$.

відповідний блоку зашифрованих даних $T_ш$. При цьому стан накопичувана N1

($a_{32}(0), a_{31}(0),…,a_2(0), a_1(0)$),

32, 31, …, 2, 1 ← номер розряда N1

стан накопичувана N2

($b_{32}(0), b_{31}(0),…,b_2(0), b_1(0)$),

32, 31, …, 2, 1 ← номер розряда N2

Аналогічно розшифровується решта блоків зашифрованих даних.

Якщо алгоритм зашифрування в режимі простої заміни 64-бітового блоку T_0 позначити через *A* то

$A(T_0) = A(a(0), b(0)) = (a(32), b(32)) = T_ш$.

Слід мати на увазі, що режим простої заміни допустимо використовувати для шифрування даних тільки в обмежених випадках - при виробленні ключа і зашифруванні його із забезпеченням імітозахисту для передачі по каналах зв'язку або для зберігання в пам'яті ЕОМ.

Питання щодо самоконтролю:

1) Для чого призначений алгоритм ДЕРЖСТАНДАРТ 28147-89?
2) Що ви знаєте про простий режим заміни алгоритму ДЕРЖСТАНДАРТ 28147-89?
3) Що ви знаєте про шифрування даних у режимі гамування?
4) Що ви знаєте про шифрування даних у режимі гамування зі зворотним

зв'язком?

Тема 16 (2 години)
Режим гамування із зворотним зв'язком

Мета: Ознакомиться з режимом гамування із зворотнім зв’язком

План

1. Основні поняття режиму гамування із зворотнім зв’язком
2. правила шифровки у режимі гамування
3. Правила дешифровки у режимі гамування

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Криптосхема, що реалізовує алгоритм зашифрування в режимі гамування зі зворотним зв'язком, має вигляд, показаний на мал.

Відкриті дані, розбиті на 64-розрядні блоки $T_0^{(1)}$, $T_0^{(2)}$ ...,$T_0^{(m)}$, зашифровуються в режимі гамування із зворотним зв'язком шляхом порозрядного складання по модулю 2 з гаммою шифра Гш, яка виробляється блоками по 64 бита:

$$\Gamma_ш = (\Gamma_ш^{(1)}, \Gamma_ш^{(2)}, \ldots, \Gamma_ш^{(m)}).$$

Число двійкових розрядів в блоці $T_0^{(m)}$ може бути менше 64, при цьому невикористана для шифрування частина гамми шифру з блоку $\Gamma_ш^{(m)}$ відкидається.

Рівняння зашифрування в режимі гамування із зворотним зв'язком мають вигляд:

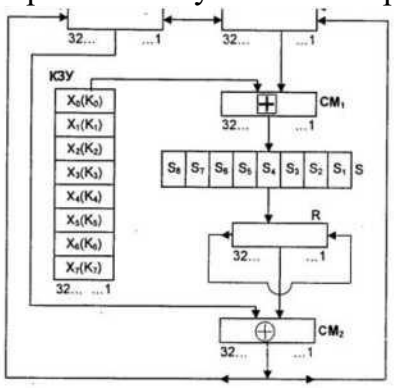

Рис. 3.13. Схема реализации режима гаммирования с обратной связью

$$T_ш^{(1)} = A(\tilde{S}) \oplus T_0^{(1)} = \Gamma_ш^{(1)} \oplus T_0^{(1)},$$
$$T_ш^{(i)} = A(T_ш^{(i-1)}) \oplus T_0^{(i)} = \Gamma_ш^{(i)} \oplus T_0^{(i)}, \quad i = 2 \ldots m.$$

Тут $Т_ш^{(i)}$ - *i*-й 64-розрядний блок зашифрованого тексту; А(*) - функція зашифрування в режимі простої заміни; *m* - визначається об'ємом відкритих даних.

Аргументом функції А(*) на першому кроці ітеративного алгоритма є 64-розрядна синхропосилка S, а на всіх наступних кроках - попередній блок зашифрованих даних $Т_ш^{(i-1)}$.

Процедура зашифрування даних в режимі гамування із зворотним зв'язком реалізується таким чином. У КЗУ вводятся 256 біт ключа. У накопичувачі N1 і N2 вводиться синхролосилка

S = (S_1, S_2.... S_{64}) з 64 біт. Початкове заповнення накопичувачів N1 і N2 зашифровується в режимі простої заміни. Отримане в результаті зашифрування заповнення накопичувачів N1 і N2 утворює перший 64-розрядний блок гамми шифру $Г_ш^{(1)}$= А, який підсумовується порозрядний по модулю 2 в суматорі $СМ_5$ з першим 64-розрядним блоком відкритих даних.

$Т_0^{(1)}$=($t_1^{(1)}$, $t_2^{(1)}$,…, $t_{64}^{(1)}$).

В результаті отримують перший 64-розрядний блок зашифрованих даних.

$Т_ш^{(1)}$=$Г_ш^{(1)}$⊕ $Т_0^{(1)}$, де $Т_ш^{(1)}$=($\tau_1^{(1)}$, $\tau_2^{(1)}$,…, $\tau_{64}^{(1)}$).

Блок зашифрованих даних $Т_ш^{(1)}$ одночасно є також початковим станом накопичувачів N1, N2 для вироблення другого блоку гамми шифру $Г_ш^{(2)}$ і тому по зворотному зв'язку $Т_ш^{(1)}$ записується у вказані накопичувачі N1 і N2.

Заповнення накопичувана N1

($\tau_{32}(1)$, $\tau_{31}(1)$,…, $\tau_2(1)$, $\tau_1(1)$),

32, 31, …, 2, 1 ← номер розряда N1

Заповнення накопичувана N2

($\tau_{64}(1)$, $\tau_{63}(1)$,…, $\tau_{34}(1)$, $\tau_{33}(1)$),

64, 63, …, 34, 33 ← номер розряда N2

Заповнення накопичувачів N1 і N2 зашифровується в режимі простої заміни. Отримане в результаті зашифрування заповнення накопичувачів N1 і N2 утворює другий 64-розрядний блок гамми шифру $Г_ш^{(2)}$, який підсумовується порозрядно по модулю 2 в суматорі $СМ_5$ з другим блоком відкритих даних $Т_0^{(2)}$:

$$Г_ш^{(2)} \oplus Т_0^{(2)} = Т_ш^{(2)}.$$

Вироблення подальших блоків гамми шифру $Г_ш^{(1)}$ і зашифрування відповідних блоків відкритих даних $Т_0^{(i)}$(i = 3.. m) виконується аналогічно.

Якщо довжина останнього *m*-го блоку відкритих даних $Т_0^{(m)}$ менше 64 розрядів, то з $Г_ш^{(т)}$ використовується тільки відповідне число розрядів гамми шифру, решта розрядів відкидаються.

У канал зв'язку або пам'ять ЕОМ передаються синхропосилка S і блоки зашифрованих даних $Т_ш^{(1)}$,

$Т_ш^{(2)}$ …, $Т_ш^{(m)}$

Розшифрування в режимі гамування із зворотнім зв'язком. При розшифруванні криптосхема має той же вигляд, що і при за шифруванні.

Рівняння розшифрування:

$$T_0^{(1)} = A(\tilde{S}) \oplus T_ш^{(1)} = Г_ш^{(1)} \oplus T_ш^{(1)},$$

$$T_0^{(i)} = Г_ш^{(i)} \oplus T_ш^{(i)} = A(T_ш^{(i-1)}) \oplus T_ш^{(i)}, \quad i = 2 \ldots m.$$

Реалізація процедури розшифрування зашифрованих даних в режимі гамування із зворотним зв'язком відбувається наступним чином. У КЗУ вводять 256 біт того ж ключа, на якому здійснювалося за шифрування відкритих блоків $T_0^{(1)}, T_0^{(2)}, \ldots, T_0^{(m)}$.

У накопичувані N1 і N2 вводиться синхропосилка S. Початкове заповнення накопичувачів N1 і N2 (синхропосилка S) зашифровується в режимі простої заміни. Отримане в результаті зашифрування заповнення N1 і N2 утворює перший блок гамми шифру

$$Г_ш^{(1)} = A(\tilde{S}),$$

який підсумовується порозрядний по модулю 2 в суматорі $СМ_5$ з блоком зашифрованих даних $T_ш^{(1)}$

В результаті виходить перший блок відкритих даних

$T_0^{(1)} = Г_ш^{(1)} \oplus T_ш^{(1)}$

Блок зашифрованих даних $T_ш^{(1)}$ є початковим заповненням накопичувачів N1 і N2 для вироблення другого блоку гамми шифру $Г_ш^{(2)}$: $Г_ш^{(2)} = A(T_Ш^{(1)})$. Отримане заповнення накопичувачів N1 і N2 зашифровується в режимі простої заміни. Утворений в результаті зашифрування блок $Г_ш^{(2)}$ підсумовується порозрядний по модулю 2 а суматорі $СМ_5$ з другим блоком зашифрованих даних $T_ш^{(2)}$. В результаті отримують другий блок відкритих даних. Аналогічно в N1, N2 послідовно записують блоки зашифрованих даних $T_ш^{(2)}$, $T_ш^{(3)}$ $T_ш^{(m)}$, з яких в режимі простої заміни виробляються блоки гамми шифру $Г_ш^{(3)}$. $Г_ш^{(4)}$ $Г_ш^{(m)}$.

Блоки гамми шифру підсумовуються порозрядний по модулю 2 в суматорі См5 з блоками зашифрованих даних $T_ш^{(3)}$, $T_ш^{(4)}$ $T_ш^{(m)}$

В результаті отримують блоки відкритих даних

$T_0^{(3)}, T_0^{(4)}, \ldots, T_0^{(m)}$,

при цьому останній блок відкритих даних $T_0^{(m)}$ може містити менше 64 розрядів.

Питання щодо самоконтролю:

1) В чому полягає суть режиму гамування із зворотним зв'язком?
2) Яким чином реалізується процедура зашифрування даних?
3) Яким чином реалізується процедура розшифрування даних?

Тема 17 (4 години)
Криптосистема з депонуванням ключа. Загальні відомості. Процедура генерації ключів.

Мета: Ознайомитись з криптосистемою депонуванням ключа

План

1. Основні поняття ключів
2. правила депонування ключа
3. Кріптоалгоритм Skipjack

Література

1. Ємець В., Мельник А., Попович Р. Сучасна криптографія. Основні поняття. - Львів: БАК, 2005 - 144 с.
2. Корольов Інформаційна безпека. - Харків :ХФН, 2000

Криптосистема з депонуванням ключа призначена для шифрування призначеного для користувача трафіку (наприклад, мовного або передачі даних) так, щоб сеансові ключі, що використовуються для зашифрування і розшифрування трафіку, були доступні при певних надзвичайних обставинах авторизованій третій стороні.

По суті, криптосистема з депонуванням ключа реалізує новий метод криптографічного захисту інформації, що забезпечує високий рівень інформаційної безпеки при передачі по відкритих каналах зв'язку й що відповідає вимогам національної безпеки. Цей метод засновано на застосуванні спеціальної мікросхеми, що шифрує/дешифрує, типу Clipper і процедури депонування ключа, що визначає дисципліну розкриття унікального ключа цієї мікросхеми. Мікросхема Clipper розроблена за технологією TEMPEST, що перешкоджає прочитуванню інформації за допомогою зовнішніх дій.

Генерація й запис унікального ключа в мікросхему виконується до вбудовування мікросхеми в кінцевий пристрій. Треба відзначити, що не існує способу, що дозволяє безпосередньо прочитувати цей ключ як підчас, так і після закінчення технологічного процесу виробництва й програмування даної мікросхеми.

Ключ розділяється на два компоненти, кожен з яких шифрується і потім передається на зберігання довіреним Агентам Депозитної Служби, які є урядовими організаціями, що забезпечують надійне зберігання компонентів ключа протягом терміну його дії. Агенти Депозитної Служби видають ці компоненти ключа тільки по відповідному запиту, підтвердженому вирішенням Федерального Суду. Отримані компоненти ключа дозволяють службам, що відповідають за національну безпеку, відновити унікальний ключ і виконати розшифрування призначеного для користувача трафіку.

В 1994 р. у США був введений новий стандарт шифрування з депонуванням

ключа EES (Escrowed Encryption Standard). Стандарт EES призначений для захисту інформації, що передається по комутованих телефонних лініях зв'язку ISDN (Integrated Services Digital Network) і радіоканалам, включаючи голосову інформацію, факс і передачу даних з швидкостями стандартних комерційних модемів. Стандарт EES специфікує алгоритм криптографічного перетворення Skipjack з 80-бітовим ключем і метод обчислення спеціального поля доступу LEAF (Law Enforcement Access Field), що дозволяє згодом розкрити секретний ключ у цілях контролю трафіку за умови законності. Алгоритм Skipjack і метод обчислення поля LEAF повинні бути реалізовані на базі мікросхеми типу Clipper. Цей стандарт специфікує унікальний ідентифікатор (серійний номер) мікросхеми UI (Device Unique Identifier), унікальний ключ мікросхеми KU (Device Unique Key) і загальний для сімейства мікросхем ключ KF (Family Key). Вся ця інформація записується в мікросхему після її виробництва, але до вбудування в конкретний пристрій. Хоча ключ KU не використовується безпосередньо для шифрування інформаційних потоків між відправником й одержувачем, об'єднання його з полем доступу LEAF і ключем KF дозволяє відновити сеансовий ключ KS і виконати дешифрування.

Кріптоалгорітм Skipjack.

Кріптоалгорітм Skipjack перетворює 64-бітовий вхідний блок в 64-бітовий вихідний блок за допомогою 80-бітового секретного ключа. Оскільки один і той же ключ використовується для шифрування і розшифрування, цей алгоритм відноситься до класу симетричних криптосистем. Відзначимо, що розмір блоку в алгоритмі Skipjack такий же, як й в DES, Алі при цьому використовується довший ключ.

Алгоритм Skipjack може функціонувати в одному з чотирьох режимів, введених для стандарту DES:

- електронної кодової книги (ECB);
- зчеплення блоків шифротекста (CBC);
- 64-бітовому зворотному зв'язку по виходу (OFB);
- зворотному зв'язку по шифротексту (CFB) для 1-, 8-, 16-, 32- або 64-бітових блоків.

Алгоритм Spikjack був розроблений безпосередньо Агентством Національної Безпеки (АНБ) США й засекречений, щоб зробити неможливою розробку програмної або апаратної реалізації процедур шифрування і розшифрування окремо від процедури депонування ключа. Біractична кріптостійкість алгоритму Skipjack була підтверджена групою незалежних експертів-криптографів.

Метод обчислення поля LEAF. Поле доступу LEAF передається одержувачеві на початку кожного сеансу зв'язку й містить секретний сеансовий ключ шифрування/розшифрування KS (Session Key). Тільки офіційні особи, що мають законний дозвіл, можуть отримати сеансовий ключ KS і дешифрувати всі раніше зашифровані на ньому повідомлення. Хоча ключ KS передається в поле LEAF, останнє використовується виключно для контролю над дотриманням законності й не

призначене для розподілу ключів абонентам. Мікросхема Clipper, що встановлена в приймаючому пристрої, не дозволяє витягувати ключ KS з інформації в полі LEAF. Поле доступу LEAF обчислюється як функція від сеансового ключа KS. вектора ініціалізації IV (Initialization Vector), ідентифікатора мікросхеми UI й унікального ключа KU. Поле LEAF складається з ключа KS. зашифрованого на ключі KU (80 біт), ідентифікатора UI (32 біта) і 16- бітового аутентифікатора EA (Escrow Authenticator). Формування вмісту поля LEAF завершается шифруванням вказаної інформації на ключі сімейства мікросхем KF, тобто

$LEAF = E_{KF}(E_{KU}(KS), UID, EA)$.

Таким чином, сеансовий ключ KS закритий подвійним шифруванням (рис. 3.14) і може бути розкритий у результаті послідовного розшифрування на ключах KF і KU.

Деталі алгоритму обчислення LEAF засекречені, включаючи режим шифрування алгоритму Skipjack і метод обчислення аутентифікатора EA. Аутентифікатор EA дозволяє контролювати цілісність і захищає поле LEAF від нав'язування помилкової інформації.

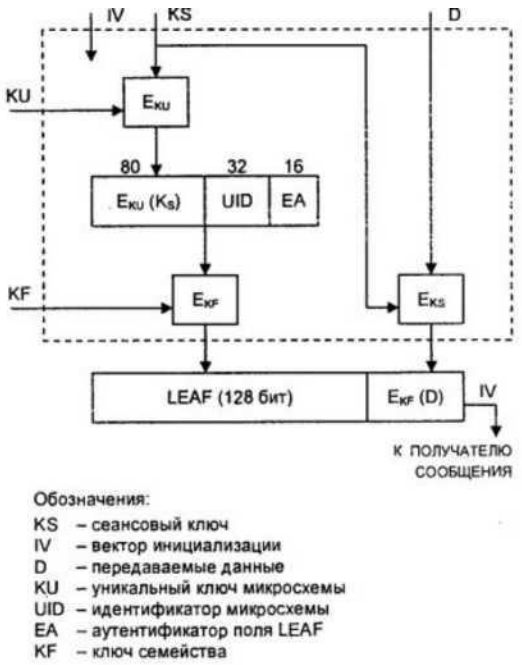

Рис. 3.14. Вычисление LEAF и формирование зашифрованного сообщения

Спосіб застосування. Для захисту телефонних переговорів кожен з абонентів повинен мати спеціальний криптографічний пристрій, Clipper, що містить, Capstone або іншу аналогічну мікросхему (рис. 3.15). У цьому пристрої повинен бути реалізований протокол, що дозволяє абонентам обмінюватися секретним сеансовим ключем KS, наприклад, за допомогою відомого методу "цифрового конверта" (digital envelope).

Даний метод застосовується в пристрої захисту телефонних переговорів TSD (3600 Telephone Security Device) компанії At&t. Воно підключається в стик між

телефонною трубкою й основним блоком й активізується натисненням кнопки. Після встановлення ключового синхронізму ключ KS разом з вектором ініціалізації IV подається на вхід мікросхеми для обчислення LEAF. Потім LEAF разом з IV передається приймаючій стороні для перевірки й синхронізації мікросхем на кінцях, що передають і приймальному. Після синхронізації мікросхем сеансовий ключ KS використовується для шифрування й розшифрування даних (мова заздалегідь оцифровується) в обох напрямах.

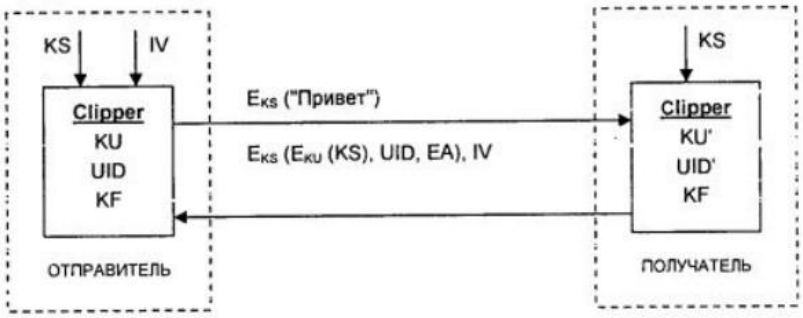

Рис. 3.15. Схема защиты телефонных переговоров с использованием микросхем Clipper

У дуплексному й напівдуплексному режимах зв'язку кожен криптографічний пристрій передає свою унікальну пару IV і LEAF. При цьому обидва пристрої використовують один і той же сеансовий ключ KS для шифрування і розшифрування.

Перша партія мікросхем для криптографічних пристроїв з депонуванням ключа була виготовлена компанією VLSI Technology Inc. і запрограмована фірмою Mykotronx. У пристрої TSD використовується мікросхема MYK-78T (Mykotronx) із швидкодією 21 Мбіт/с.

Розглянемо докладніше функціонування криптосистеми з депонуванням ключа й необхідну для її роботи інфраструктуру.

Процедура генерації ключів

У процедурі генерації ключів беруть участь Національний Менеджер Програми; два Агенти Депозитної Служби; два Агенти KF-служби, що відповідають за формування і зберігання ключа сімейства KF; представник Служби Програмування мікросхем (Programming Facility).

До генерації унікального ключа мікросхеми KU і відповідних йому ключових компонентів KC_1 і KC_2 необхідно згенерувати допоміжні ключі KN_1 і KN_2 і випадкові числа RS_1 і RS_2 (Random Seeds). У процедурі генерації використовується спеціальна смарт-карта, на якій відповідно до стандартом X9.17 (FIPS 171) реалізований генератор псевдовипадкових чисел. Початкове значення генератора формується як результат обчислення хеш-функції від послідовності випадкових символів, введених з клавіатури, тимчасових інтервалів між натисками при введенні символів з клавіатури і поточного часу доби.

Описана вище процедура використовується Агентами Депозитної Служби для

генерації ключових чисел KN), KN_2 і випадкових чисел RS_1 RS_2.

Компоненти ключа сімейства KF. Агенти KF-служби, що відповідають за формування ключа сімейства KF, генерують компоненти KFC_1 і KFC_2 на окремих операційних пунктах.

По дві копії кожного компоненту записуються на магнітні носії. Кожен магнітний носій поміщають в спеціальний пронумерований контейнер. Контейнер з копією кожного компонента на магнітному носієві поміщається в окремий сейф.

Ключові і випадкові числа. Кожен Агент Депозитної Служби генерує і записує на магнітні носії чотири копії ключових чисел KN_1 і KN_2. Кожен магнітний носій поміщається в спеціальний пронумерований контейнер. Контейнер з копією кожного компоненту на магнітному носієві помещают в окремий сейф, встановлений в операційному пункті Агента. Крім того, кожен Агент генерує і записує на магнітний носій одне випадкове число RS. Магнітний носій в контейнері поміщається в сейф.

Кожен Офіцер Депозитної Служби (представник Агента) доставляє в Службу Програмування мікросхем дві копії ключового числа (KN_1 або KN_2 і одна копія випадкового числа (RS_1 або RS_2).

Питання щодо самоконтролю:

1) Для чого призначена криптосистема з депонуванням ключа?
2) Що ви знаєте про стандарт EES?
3) Ким був розроблений алгоритм шифрування Skipjack?
4) Що ви знаєте про процедуру генерації ключів?
5) Розкажіть про ключові і випадкові числа.
6) Що ви знаєте про режим вироблення імітовставки?