МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

Кафедра комп’ютерних систем та мереж

Методичні вказівки

до виконання лабораторних робіт з дисципліни

Захист інформації у комп’ютерних системах

для студентів денної та заочної форми навчання
спеціальності
123 "Комп'ютерна інженерія"

Тернопіль -2019

Методичні вказівки для лабораторних робіт з дисципліни «Захист інформації у комп’ютерних системах» розроблені у відповідності з навчальним планом за спеціальністю 123 “Комп’ютерна інженерія”

УКЛАДАЧ: ст. викл. каф. КС Жаровський Р.О.

Методичні вказівки розглянуті і затвердженні на засіданні кафедри комп’ютерних систем та мереж, протокол №7 від 07.02.2019 року.

Зміст

Вступ

Методичні вказівки до лабораторних робіт з курсу “ Захист інформації у комп'ютерних системах ” покликані допомогти студентам денної та заочної форм навчання засвоїти використання систем захисту інформаційного обміну в комп’ютерних мережах.

Зміст та структура методичних вказівок відповідає освітньо-професійній програмі підготовки фахівців з спеціальності 123 "Комп'ютерна інженерія". Лабораторний практикум охоплює основний зміст матеріалу дисципліни “ Захист інформації у комп'ютерних системах ”.

Мета курсу. Метою викладання дисципліни є ознайомлення студентів зі способами захисту інформації у комп'ютерних системах. У курсі розглядаються основні підходи до розробки системи інформаційної безпеки комп'ютерних систем.

У ході виконання лабораторних робіт студенти отримують наступні навики і уміння:

- Основний склад і принципи функціонування систем захисту інформації;
- Принципи і методи криптографічного захисту інформації;
- Механізми і методи контролю КС;
- Принципи роботи основних типів шкідливих комп’ютерних програм і методи боротьби з ними:
- Здійснювати захист інформації в КС або мережах;
- Розробляти та використовувати сучасні засоби та методи криптографічного захисту інформації;
- Виконувати розрахунки характеристик безпечного використання паролів та ключів шифрування;
- Виконувати проектування системи захисту інформації в КС та мережах, використовуючи сучасні засоби розмежування доступу користувача до критичної інформації з елементами аутентифакції суб’єктів та повідомлень;
- Адекватно обирати методи і засоби шифрування інформації;
- Виявляти і усувати потенційно небезпечні місця у системі безпеки комп'ютерних систем;
- Працювати з системним журналом і відслідковувати модифікацію системи.

Кожна лабораторна робота містить наступні розділи:

- тема,мета роботи;
- теоретичні відомості, необхідні для виконання роботи;
- порядок виконання роботи;
- контрольні запитання.

Лабораторна робота №1.

Тема: ”Комп’ютерні віруси: знайомство з принципами роботи. Захист від вірусів. Огляд основних антивірусних програм”

Мета роботи: ознайомитись з основними видами комп’ютерних вірусів, принципами їх роботи поширення і знищення. Розглянути програми для захисту від вірусів, принцип дії, ефективність, можливості.

Теоретичні відомості

1. Комп'ютерні віруси, їх властивості і класифікація

1.1. Властивості комп'ютерних вірусів

Зараз застосовуються персональні комп'ютери, в яких користувач має вільний доступ до всіх ресурсів машини. Саме це відкрило можливість для небезпеки, яка отримала назву комп'ютерного вірусу.

Що таке комп'ютерний вірус? Формальне визначення цього поняття до цих пір не придумане, і є серйозні сумніви, що його взагалі може бути дано. Численні спроби дати «сучасне» визначення вірусу не привели до успіху. Тому ми обмежимося розглядом деяких властивостей комп'ютерних вірусів, які дозволяють говорити про них як про деякий певний клас програм.

Перш за все, вірус - це програма. Таке просте твердження саме по собі здатне розвіяти безліч легенд про незвичайні можливості комп'ютерних вірусів. Вірус може перевернути зображення на вашому моніторі, але не може перевернути сам монітор. До легенд про віруси-вбивці, «що знищують операторів за допомогою виведення на екран смертельної колірної гамми 25-м кадром» також не варто відноситися серйозно. На жаль, деякі авторитетні видання час від часу публікують «найсвіжіші новини з комп'ютерних фронтів», які при найближчому розгляді виявляються наслідком не цілком ясного розуміння предмету.

Вірус - програма, що володіє здібністю до самовідтворення. Така здатність є єдиним засобом, властивим всім типам вірусів. Але не тільки віруси здібні до самовідтворення. Будь-яка операційна система і ще безліч програм здатні створювати власні копії. Копії ж вірусу не тільки не зобов'язані повністю співпадати з оригіналом, але, і можуть взагалі з ним не співпадати!

Вірус не може існувати в «цілковитій ізоляції»: сьогодні не можна уявити собі вірус, який не використовує код інших програм, інформацію про файлову структуру або навіть просто імена інших програм. Причина зрозуміла: вірус повинен яким-небудь способом забезпечити передачу собі управління.

1.2. Класифікація вірусів

На сьогодні відомі десятки тисяч вірусів, які в цілому мають конкретну класифікацію. Спробуємо детальніше розглянути основні групи, на які поділяються комп'ютерні віруси.

I. Поділ вірусів за середовищем їх розповсюдження:

- Завантажувальні віруси - це найбільш небезпечна група вірусів, що заражають Boot Record та Master Boot Record логічних та фізичних дисків. Про ці віруси ми вже говорили попередньо.
- Файлові віруси. Ці віруси поширюються, заражаючи файли різних типів, як вже було сказано, - найчастіше це виконуючі файли та файли оверлеїв. До цієї групи слід також віднести макровіруси, хоч інколи їх виділяють як окремий клас вірусів.
- Завантажувально-файлові віруси здатні вразити як код завантажувальних секторів, так і код файлів, як правило системних.
- Віруси сімейства Dir використовують інформацію про файлову структуру та вміст каталогів (див. попередній матеріал).
- Multipartition - віруси можуть вражати одночасно виконуючі файли, boot - сектор, MBR, FAT і каталоги і є найбільш небезпечними, особливо, якщо вони ще й володіють поліаморфними властивостями і елементами невидимості.
- Мережеві віруси - це віруси, що поширюються як сукупність машинного коду в комп'ютерних мережах.
- Поштові віруси - на сьогодні досить нова але надзвичайно поширена група вірусів, що розповсюджуються разом із поштовими повідовленнями у вигляді прикріплених до них файлів (Attachment) із програмним кодом. Як правило, такі віруси досить швидко розмножуються і час від часу викликають вірусні епідемії (згадати, хоча б, такі резонансні в останні роки віруси як "I Love You" (LoveLetter), Melissa або "Anna Kournicova").

II. Класифікація комп'ютерних вірусів за алгоритмом роботи:

- Віруси "паразити" найпростіші віруси що використовують "тіло" інших файлів (виконуючих), записуючи туди себе. Вони можуть бути досить легко виявлені і знешкоджені.
- Віруси супутники створюють копію exe-файлу з розширенням com і записують туди себе. Коли з командного рядка DOS завантажують такий файл, то як правило розширення не вказують, а за правилами DOS, першим завантажується com файл, тобто вірус.
- Віруси "черв'яки" (віруси-реплікатори) не створюють собі файлу, а поширюються лише в комп'ютерних мережах та в оперативній пам'яті у вигляді певного машинного коду. Вони ніби черв'яки проникають в оперативну пам'ять ПК через комп'ютерну мережу, пронизуючи системи захисту. Найбільш грізними представниками цього типу вірусів є Nimda (неодноразовий переможець рейтингів найнебезпечніших вірусів), Gigger та Redesi (здатні відформатувати диск C), Bumerang (здатний знищити FlashBIOS та таблиці файлової системи вінчестера), SirCam (найдзвичано швидкий у розповсюдженні та знищує інформацію на диску C), Кігау та Paukor (знищують всі файли із системних папок Windows).
- Студентські віруси - це віруси, які мають в собі багато помилок і написані, як правило, початківцями.
- Віруси "невидимки" (Stealth - віруси) фальсифікують інформацію, перехоплюючи звертання антивірусної програми, до заражених ділянок диску і

направляючи її на незаражені. Вірус перехоплює вектор переривання int 13h. Ця технологія використовується, як у файлових, так і в завантажувальних вірусах.

- Віруси "мутанти" ("привиди") або поліаморфні (polimorphic) - не мають постійної сигнатури (машинного коду), за якою можна було б виявити вірус. Вони міняють сигнатуру з кожною копією і тому з ними важко боротись. Виявляють такі віруси лише за допомогою еврістичного аналізу, коли антивірусна програма "прокручує" алгоритм роботи виконуючих файлів і в разі підозрілих операцій приймає це за вірус. Таким же чином антивірусні програми шукають невідомі ще їм віруси.
- Ретровіруси - це звичайні файлові віруси, які прагнуть заразити антивірусні програми, знищуючи їх або роблячи непрацездатними. Тому практично всі антивіруси в першу чергу перевіряють свої власні розміри і контрольну суму.
- "Троянські" віруси (Trojans) здійснюють шкідливі дії замість оголошених легальних функцій або разом з ними. Вони переважно не здатні на саморозповсюдження і передаються тільки при копіюванні користувачем. Часто ці віруси використовують в якості "шпигунів". Проникаючи по мережі на ПК, вони стараються "затаїтись" і "вкрасти" паролі користувача (особливо виходу в Internet) і передати їх господарю. Деякі троянські віруси готують грунт на зараженому ними ПК для проникнення без перешкод інших вірусів, що слідують за ними. Боротись з такими вірусами (особливо новими) досить важко, адже в їх коді немає ніякої дистуктивної дії (не міняється розмір інших файлів, не форматуються диски), а навпаки вони стараються ніяк себе не проявити. Для боротьби з такими вірусами використовуються спеціальні програми FireWoll (файрволл) - мережеві екрани, які під час підключення до мережі слідкують чи не пробує якась програма на ПК вийти в Internet. Якщо така спроба відбулась, то вона блокується і виводиться повідомлення, із запитом дозволу на таку операцію. Корисною функцією файрволл є те, що він може захистити не лише від троянських вірусів, але й від хакерських атак із зовні (з Internet). Потрібно відмітити, що існує багато відомих троянських вірусів, які не лише виступають в ролі "шпигунів" але й самі несуть досить високу деструктивну дію (наприклад TROJ_ZERAF знищує EXE і SYS файли та робить помилки в системному реєстрі).
- Віруси таймери очікують лише певного часу (певної години, дня і т.д.), і лише тоді спрацьовують.

III. Поділ вірусів за деструктивною дією:

- Нешкідливі віруси - це віруси, які не приносять жодної шкоди, а просто себе копіюють багато разів, заповнюючи диски, або загромаджуючи оперативну пам'ять.
- Не небезпечні віруси схожі до попередніх, але крім цього їх дія супроводжується різними спецефектами (відеота звуковими).
- Небезпечні віруси - це віруси дія яких призводить до серйозних збоїв в роботі ПК, таких як зависання комп'ютера іт.д.
- Дуже небезпечні віруси - це віруси, дія яких супроводжується знищенням інформації (файлів, каталогів, форматування цілих дисків). В січні

1998 року, завдяки журналу "Virus Bulletin", з'явився термін WildList (список "диких вірусів"). Він регулярно поновлюється і публікується цим авторитетним міжнародним виданням.

IV. Класифікація вірусів за принципом дії:

- Резидентні - це віруси, що завантажуються в оперативну пам'ять і постійно там знаходяться, аж до виключення живлення чи перезавантаження ПК.
- Нерезидентні - це віруси, які короткочасно завантажуються в пам'ять, виконують потрібні їм дії і вивантажуються з пам'яті.

V. Поділ вірусів за місцем втілення у файли:

- **На початку файлу.**
- **Всередині файлу.**
- **В кінці файлу.**

2. Види файлів, які можуть бути заражені вірусом.

Як правило, кожна конкретна різновидність вірусу може заразити тільки один або два типи файлів. На даний час частіше всього зустрічаються макровіруси, тоді як в 90-ті роки найпоширенішими були віруси, що заражали COM-файли, а на другому місці - EXE-файли.

Види файлів, які можуть бути заражені вірусом:

- **виконуючі файли**, тобто файли з розширеним ім'ям COM і EXE, а також оверлейні файли, завантажені іншими програмами. Вірус в заражених виконуючих файлах починає свою роботу при завантаженні тієї програми, в якій він знаходиться;
- **файли документів та шаблонів**, створених програмами Word, Excel, Access та іншими офісними програмами, а точніше макроси, що використовуються там. Цей тип вірусів порівняно молодий і називають його макровірусами. Деякі з вірусів цього типу є надзвичайно шкідливими. Наприклад вірус W97M.Thus при активізації 13 грудня здатний знищити всі файли на диску C, зберігаючи при цьому структуру каталогів (папок).
- **блок початкового завантаження** операційної системи і головний завантажувальний запис жорсткого диску. Вірус, який заразив ці ділянки, як правило, складається з 2-х частин, оскільки на цих ділянках диску, важко розмістити програму вірусу в цілому. Частина вірусу, що не поміщається в них, розташована на іншій ділянці диску, який оголошується дефектним. Такий вірус починає свою роботу при початковому завантажуванні операційної системи і є резидентним, тобто постійно знаходиться в пам'яті комп'ютера. Відомі випадки, коли вірус форматує додаткову доріжку диску, куди і записує основну частину програми;
- **таблиці файлової системи та каталоги**. Як відомо, в кожен каталог записуються імена файлів, дата та час створення, номер першого кластера файлу, а також резервні байти, що ОС не використовуються. Віруси цього типу, записавшись в кластери, помічають їх як пошкоджені, а тоді реорганізовують файлову систему. При цьому інформація про перші кластери деяких виконуючих файлів записується у резервні біти, а на її місце поміщається посилання на тіло вірусa. Тому, при спробі користувача завантажити відповідну програму - вірус

отримує керівництво. Цей тип вірусів, з'явившись в 1991 році, викликав в Росії справжню епідемію, яку можна порівняти із чумою.

- **драйвери пристроїв**, тобто файли, які здійснюють програмне керування зовнішнім пристроєм. Вірус, який знаходиться в цих файлах, починає свою роботу при кожному звертанні до відповідного пристрою;
- **системні файли**, тобто файли IO.SYS і MSDOS.SYS. Це досить небезпечно, оскільки вони, як і у випадку зараження блоків початкового завантаження дисків, починають діяти при кожному завантаженні ПК.

3. Шляхи проникнення вірусів в комп'ютер

Розглянемо, тепер, яким чином комп'ютерний вірус може потрапити на ПК звичайного користувача. На початку їх існування основним середовищем розповсюдження вірусів були переносні диски, переважно дискети. Пізніше, із набуттям популярності CD-дисків вони також стали зручним середовищем поширення вірусів (перш за все це не ліцензійні програмні продукти). В другій половині 2001 року комп'ютерні віруси проникли і на DVD (вірус Funvole).

В другій половині 90-х років основним середовищем розповсюдження комп'ютерних вірусів стали комп'ютерні мережі та електронна пошта. Це викликано надзвичайно бурхливим розвитком Internet, що дозволило фактично миттєво поширюватись новим вірусам на дуже великі території.

Прикладом може бути 2001 рік, визнаний відомими міжнародними антивірусними виданнями роком троянських вірусів та поштових черв'яків, а лідером серед вірусів став поштовий черв'як Nimda.

Варто пам'ятати проте, що розробники комп'ютерних вірусів не зупиняються на досягнутому і шукатимуть нові шляхи розповсюдження комп'ютерних вірусів. Так, наприклад, відомі вже випадки поширення вірусів через файли в форматі RTF, PDF та анімаційні файли, створені в Macromedia Flash (перший відомий вірус SWF/LFM-926).

4. Захист від комп’ютерних вірусів.

4.1. Класифікація антивірусних програм.

Комп'ютерний вірус - це дуже неприємне шкідливе явище, побачити яке на своєму ПК не хотів би, напевне, жоден користувач. Застрахуватись від вірусів на сьогодні повністю неможливо, хіба що зовсім ізолювати ПК від обміну інформацією із навколишнім світом. Але робити це, напевне ніхто не буде, адже тоді ПК втратить багато своїх переваг.

Необхідно застосовувати спеціалізовані програми для захисту від вірусів. Ці програми можна поділити на декілька видів.

- **детектори** - дозволяють знайти файли, заражені яким-небудь одним, наперед відомим нам вірусом, або одним з багатьох відомих вірусів;
- **вакцини** (іммунізатори) - модифікують (інфікують) програми і диски таким чином, що це не відображається на роботі програм. Після цього вірус, від якого виконується вакцинація, вважає ці програми або диски вже інфікованими і повторно їх не заражає;

- **лікарі** (фаги) - лікують заражені програми або диски "викусуючи" із заражених програм тіло віруса, тобто відновлюючи програму в тому стані, в якому вона була до зараження вірусом;
- **ревізори** - спочатку запам'ятовують стан інформації (розмір, дату і час створення) і системних ділянок дисків, а потім порівнюють його з поточним. При виявленні невідповідностей про це повідомляється користувачу;
- **лікарі-ревізори** - це гібриди ревізорів і лікарів, тобто програми, які не тільки помічають зміни в файлах і системних ділянках дисків, але й можуть у випадку виявлення змін вилікувати заражені файли;
- **фільтри** (монітори) - резидентні програми для захисту від вірусів, які поміщаються резидентно в оперативній пам'яті комп'ютера і перехоплюють звернення вірусів до системних ділянок і файлів. Користувач може дозволити або заборонити виконання відповідних операцій;
- **поліфаги** - це найбільш ефективніша група програм, що поєднують в собі декілька вище приведених типів антивірусів, наприклад, фільтрів, детекторів та лікарів.

4.2. Огляд антивірусних засобів.

Темі боротьби з комп’ютерними вірусами в світі приділяється багато уваги. Багато великих та малих компаній займаються розробкою нових та ефективних програм для захисту ПК від вірусів. Найбільш впливовим і авторитетним показником ефективності антивірусних програм є рейтинг, який щомісяця проводить міжнародний комп'ютерний журнал Virus Bulettin (Англія).

Проводяться тестування, при яких антивіруси встановлюються в однакових умовах на заражені різними типами вірусів комп'ютери і визначається відсоток виявлених та знешкоджених ними вірусів. Тестування проводиться по таких основних категоріях, як wild ("дикі"), макровіруси, поліморфні та стандартні. При тестуванні враховуються також такі параметри як швидкість роботи програми, її вартість та зручність інтерфейсу. Сама участь антивірусної програми в тестуванні вже є великим визнанням для неї.

Порядок виконання роботи

1. Ознайомитись з основними типами вірусів.
2. Вибрати один з типів вірусів:
    a. Розглянути принцип роботи даного типу вірусів;
    b. Навести назви вірусів даного типу;
    c. Методи поширення даного типу вірусів;
    d. Які існують програми для знищення даного типу вірусів;
    e. Які методи використовують для знищення даного типу вірусів.
3. Оформити звіт по роботі
    a. Звіт повинен включати:
        - тему, мету роботи;
        - короткий виклад основних теоретичних положень;

- тип, параметри досліджуваного у роботі класу вірусів;
- алгоритм роботи вірусу;
- методи знищення вірусу;
- висновки.

Контрольні запитання

1. Що таке комп’ютерний вірус?
2. Властивості вірусів?
3. Як працюють віруси?
4. Як розповсюджуються комп’ютерні віруси?
5. Поняття зараженої програми?
6. Які файли можуть бути заражені вірусом?
7. Що таке Інтернет – хробаки?
8. Які принципи роботи троянських програм?
9. Що таке антивіруси?
10. Як функціонують антивірусні програми?
11. Класи антивірусних програм.

Лабораторна робота №2.

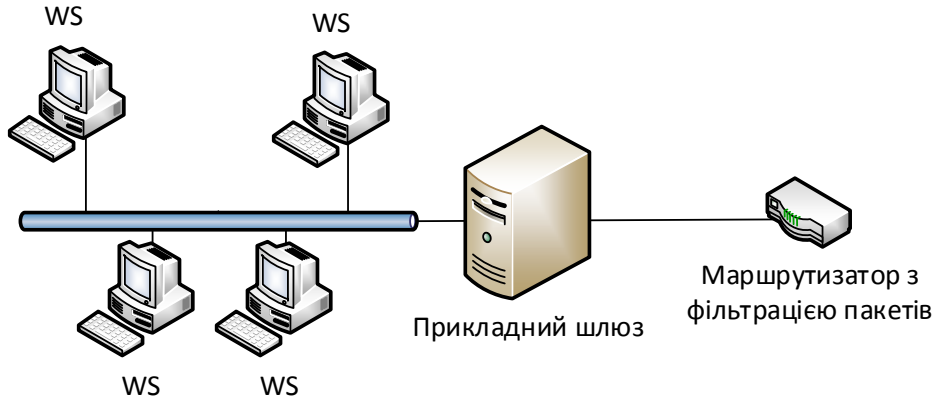
Тема: ” Інструментальні засоби захисту, firewall’и ”

Мета роботи: ознайомитись з основними видами firewall’ів, принципами їх роботи, ефективність, можливості.

Теоретичні відомості.

1 Поняття брандмауера

Брандмауер - це не просто маршрутизатор, хост або група систем, які забезпечують безпеку в мережі, швидше, брандмауер - це підхід до безпеки; він допомагає реалізувати політику безпеки, яка визначає дозволені служби і типи доступу до них, і є реалізацією цієї політики в термінах мережевої конфігурації, декількох хостов і маршрутизаторів, і інших заходів захисту, таких як посилена аутентифікація замість статичних паролів. Основна мета системи брандмауера - управління доступом До або З мережі, що захищається. Він реалізує політику мережевого доступу, примушуючи проходити усі з'єднання з мережею через брандмауер, де вони можуть бути проаналізовані і дозволені або знехтувані.



Малюнок 2.1 Приклад брандмауера з маршрутизатором і прикладним шлюзом

Система брандмауера може бути маршрутизатором, персональним комп'ютером, хостом, або групою хостів, створеною спеціально для захисту мережі або підмережі від неправильного використання протоколів і служб хостами, що знаходяться поза цією підмережею. Зазвичай система брандмауера створюється на основі маршрутизаторів верхнього рівня, зазвичай на тих, які сполучають мережу з Інтернетом, хоча може бути створена і на інших маршрутизаторах, для захисту тільки частини хостов або підмереж.

Компоненти брандмауера

- основними компонентами брандмауера є:
- політика мережевого доступу
- механізми посиленої аутентифікації
- фільтрація пакетів
- прикладні шлюзи

Наступні розділи описують детальніше кожну з цих компонент.

Політика мережевого доступу

Є два види політики мережевого доступу, які впливають на проектування, установку і використання системи брандмауера. Політика верхнього рівня є проблемною концептуальною політикою, яка визначає, доступ до яких сервісів буде дозволений або явно заборонений з мережі, що захищається, як ці сервіси використовуватимуться, і за яких умов робитиметься виключення і політика не дотримуватиметься. Політика нижнього рівня описує, як брандмауер повинен насправді обмежувати доступ і фільтрувати сервіси, які вказані в політиці верхнього рівня. Наступні розділи коротко описують ці політики.

Політика доступу до сервісів

Політика доступу до сервісів повинна фокусуватися на проблемах використання Інтернету, описаних вище, і, судячи з усього, на усьому доступі до мережі ззовні (тобто політика доступу по модемах, з'єднань SLIP і PPP). Ця політика має бути уточненням загальної політики організації відносно захисту інформаційних ресурсів в організації. Щоб брандмауер успішно захищав їх, політика доступу до сервісів має бути реалістичною і узгоджуватися із зацікавленими особами перед установкою брандмауера. Реалістична політика - це така політика, в якій знайдений баланс між захистом мережі від відомих ризиків, але в той же час забезпечений доступ користувачів до мережевих ресурсів. Якщо система брандмауера забороняє або обмежує використання деяких сервісів, то в політиці має бути явно описана суворість, з якою це робиться, щоб запобігти зміні параметрів засобів управління доступом миттєвим чином. Тільки підтримувана керівництвом реалістична політика може забезпечити це.

Брандмауер може реалізовувати ряд політик доступу до сервісів, але типова політика може забороняти доступ до мережі з Інтернету, і дозволяти тільки доступ до Інтернету з мережі. Іншою типовою політикою може бути дозвіл деякого доступу з Інтернету, але тільки до обраних систем, таким як інформаційні сервера і поштові сервера. Брандмауери часто реалізують політик доступу до сервісів, які дозволяють користувачам мережі працювати з Інтернету з деякими обраними хостами, але цей доступ надається, тільки якщо він поєднується з посиленою аутентифікацією.

Політика брандмауера

Вона специфічна для конкретного брандмауера. Вона визначає правила, використовувані для реалізації політики доступу до сервісів. Не можна розробляти цю політику, не розуміючи такі питання, як можливості і обмеження брандмауера, і погрози і узявимые місця, пов'язані з TCP/IP. Як правило, реалізується одна з двох базових політик проекту:

- дозволити доступ для сервісу, якщо він явно не заборонений
- заборонити доступ для сервісу, якщо він явно не дозволений

Брандмауер, який реалізує першу політику, пропускає усі сервіси в мережу за умовчанням, нсли тільки цей сервіс не був явно вказаний в політиці управління доступом як заборонений. Брандмауер, який реалізує другу політику, за умовчанням забороняє усі сервіси, але пропускає ті, які вказані в списку

дозволених сервісів. Друга політика наслідує класичну модель доступу, використовувану в усіх областях інформаційну безпеку.

Перша політика менш бажана, оскільки вона надає більше способів обійти брандмауер, наприклад, користувачі можуть дістати доступ до нових сервісів, що не забороняються політикою (або навіть не вказаних в політиці), або запустити заборонені сервіси на нестандартних портах TCP/UDP, які не заборонені політикою. Певні сервіси, такі як X Windows, FTP, ARCHIE і RPC, складно фільтрувати, і для них краще підходить брандмауер, що реалізовує першу політику. Друга політика більш суворо і безпечніше, але її важче реалізувати і вона може вплинути на роботу користувачів в тому відношенні, що ряд сервісів, такі, як описані вище, можуть виявитися блокованими або використання їх буде обмежено.

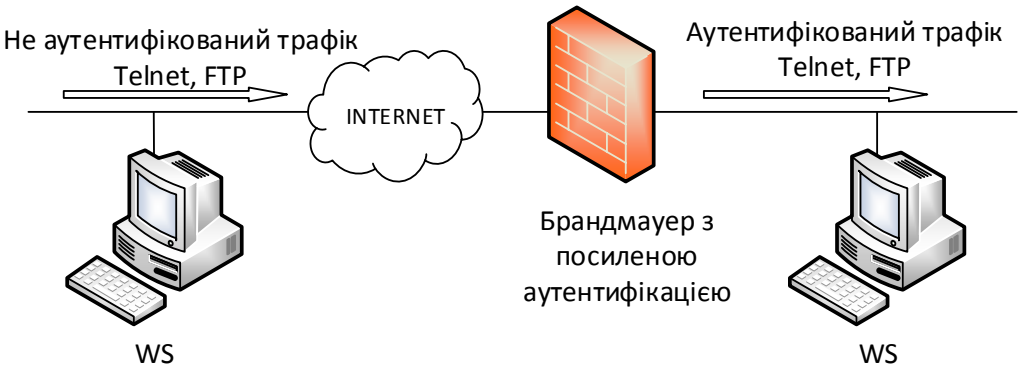
Взаємозв'язок між концептуальною політикою доступу до сервісів і відповідною їй другою частиною описаний вище. Цей взаємозв'язок існує через те, що реалізація політики доступу до сервісів сильно залежить від можливостей і обмежень системи брандмауера, а також вразливих місць, наявних в дозволених інтернетівських сервісах. Наприклад, може виявитися необхідним заборонити сервіси, дозволені політикою доступу до сервісів, якщо вразливі місця в них не можуть ефективно контролюватися політикою нижнього рівня і, якщо безпека мережі найважливіша. З іншого боку, організація, яка сильно залежить від цих сервісів при рішенні своїх завдань, може прийняти це вищий ризик і дозволити доступ до цих сервісів. Цей взаємозв'язок призводить до того, що формулювання обох політик стає ітеративним процесом.

Політика доступу до сервісів - найважливіший компонент з чотирьох, описаних вище. Інші три компоненти використовуються для реалізації політики. (політика доступу до сервісів повинна відбивати загальну політику безпеки організації). Ефективність системи брандмауера при захисті мережі залежить від типу використовуваної реалізації його, від правильності процедур роботи з ним, і від політики доступу до сервісів.

1.1 Посилена аутентифікація

Розроблений ряд заходів посиленої аутентифікації, таких як смарт-карти, біометричні механізми, і програмні механізми, для захисту від уразливості звичайних паролів. Хоча вони і відрізняються один від одного, усі вони однакові в тому відношенні, що паролі, генеровані пристроєм посиленої аутентифікації, не можуть бути повторно використані таким, що атакує, який перехоплює трафік з'єднання. Оскільки проблема з паролями в Інтернеті є постійною, брандмауер для з'єднання з Інтернетом, який не має засобів посиленої аутентифікації або не використовує їх, бессмысленен.

Ряд найбільш популярних пристроїв посиленої аутентифікації, використовуваних сьогодні, називаються системами з одноразовими паролями. Смарт-карта, наприклад, генерує відповідь, яку хост використовує замість традиційного пароля. Оскільки смарт-карта працює спільно з програмою або устаткуванням на хості, генеровані відповіді унікальні для кожного встановлення сеансу. Результатом є одноразовий пароль, який, якщо перехоплюється, не може бути використаний зловмисником для встановлення сеансу з хостом під виглядом користувача.



Малюнок 2.2 Використання посиленої аутентифікації в брандмауері для попередньої аутентифікації трафіку TELNET, FTP

Оскільки брандмауери можуть централізувати управління доступом в мережі, вони є логічним місцем для установки програм або пристроїв посиленої аутентифікації. Хоча заходи посиленої аутентифікації можуть використовуватися на кожному хосте, більше практичним є їх розміщення на брандмауері. Малюнок 2.2 показує, що в мережі без брандмауера, що використовує заходи посиленої аутентифікації, неаутентифицированный трафік таких застосувань як TELNET або FTP, може безпосередньо проходити до систем в мережі. Якщо хосты не використовують заходів посиленої аутентифікації, зловмисник може спробувати зламати паролі або перехоплювати мережевий трафік знайти в нім сеанси, в ході яких передаються паролі. Малюнок 2.2 також показує мережа з брандмауером, що використовує посилену аутентифікацію, при якій сеанси TELNET або FTP, що встановлюються з боку Інтернету з системами мережі, повинні проходити перевірку за допомогою посиленої аутентифікації перед початком роботи. Самі системи мережі можуть продовжувати вимагати статичні паролі перед доступом до себе, але ці паролі не можна буде використовувати, навіть якщо їх перехопити, оскільки заходи посиленої аутентифікації і інші компоненти брандмауера не дозволять зловмисникові проникнути або обійти брандмауер.

1.2 Фільтрація пакетів

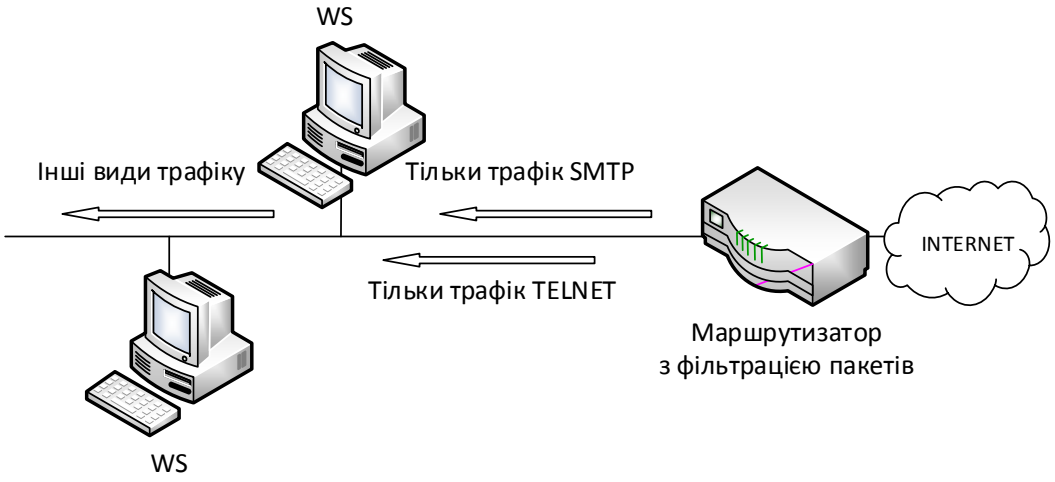
Фільтрація IP -пакетов зазвичай виконується за допомогою маршрутизатора з фільтрацією пакетів, що здійснює її, коли пакети передаються між інтерфейсами маршрутизатора. Маршрутизатор, що фільтрує, зазвичай може фільтрувати IP -пакеты на основі групи полів з наступних полів пакету:

- IP -адрес відправника
- IP -адрес одержувача
- TCP/UDP -порт відправника
- TCP/UDP -порт одержувача

Не усі маршрутизатори, що фільтрують, зараз фільтрують по TCP/UDP -порту відправника, але багато виробників почали включати таку можливість. Деякі маршрутизатори перевіряють, з якого мережевого інтерфейсу маршрутизатора прийшов пакет, і потім використовують цю інформацію як додатковий критерій фільтрації. Деякі версії Unix мають можливість фільтрації пакетів, але далеко не усе.

Фільтрація може бути використана по-різному для блокування з'єднань від або до окремих хостам або мереж, і для блокування з'єднань до різних портів. Організації може знадобитися блокувати з'єднання від специфічних адрес, таких як хосты або мережі, які вважаються ворожими або ненадійними. Або ж організація може захотіти блокувати з'єднання від усіх адрес, зовнішніх по відношенню до організації (з невеликими виключеннями, такими як SMTP для отримання пошти).

Додавання фільтрації по портах TCP і UDP до фільтрації по IP -адресам дає велику гнучкість. Нагадаємо главу 1, в якій говорилося, що сервера, такі як демон TELNET, пов'язані зазвичай з конкретними портами, такими як порт 23 для TELNET. Якщо брандмауер може блокувати з'єднання TCP або UDP або від певних портів, то можна реалізувати політику, при якій певні види з'єднань можуть бути здійснені тільки з конкретними хостами, але не з іншими. Наприклад, організація може захотіти блокувати усі з'єднання, що входять, для усіх хостов, окрім декількох систем, що входять до складу брандмауера. Для цих систем можуть бути дозволені тільки певні сервіси, такі як SMTP для однієї системи, і TELNET або FTP для іншої. При фільтрації по портах TCP і UDP ця політика може бути легко реалізована маршрутизатором з фільтрацією пакетів або хостом з можливістю фільтрації пакетів.



Малюнок 2.3 Приклад фільтрації пакетів для TELNET і SMTP

Для прикладу розглянемо політику, в якій дозволяються тільки певні з'єднання з мережею з адресою 123.4.*.* З'єднань TELNET дозволяються тільки з одним хостом, 123.4.5.6, який може бути прикладним TELNET -шлюзом мережі, а SMTP -соединения дозволяються тільки з двома хостами, 123.4.5.7 і 123.4.5.8, які можуть бути двома поштовими шлюзами мережі. NNTP (Network News Transfer Protocol) звільняється тільки від сервера новин, що взаємодіє з мережею, 129.6.48.254, і тільки з NNTP -сервером мережі, 123.4.5.9, а протокол NTP(мережевого часу) дозволений для усіх хостов. Усі інші сервіси і пакети блокуються. Приклад набору правил наведений ниже:

Тип	Адреса відправника	Адреса одержувача	Порт джерела	Порт одержувача	Дія
tcp	*	123.4.5.6	>1023	23	дозволити
tcp	*	123.4.5.7	>1023	25	дозволити
tcp	*	123.4.5.8	>1023	25	дозволити
tcp	129.6.48.254	123.4.5.9	>1023	119	дозволити

udp	*	123.4.*.*	>1023	123	дозволити
*	*	*	*	*	заборонити

Перше правило дозволяє пропускати пакети TCP з Інтернету від будь-якого джерела, що мають порт відправника більше ніж 1023, до адреси 123.4.5.6, якщо з'єднання встановлюється з портом 23. Порт 23 - це порт, пов'язаний з сервером TELNETa, а усі клієнти TELNETa повинні використовувати непривілейовані порти більше, ніж 1024. Друге і третє правило працюють аналогічно, крім того, що дозволяються адреси призначення 123.4.5.7 і 123.4.5.8 і порт 25 - SMTP. Четверте правило пропускає пакети до NNTP -серверу мережі, але тільки від адреси 129.6.48.254 до адреси 123.4.5.9 з портом призначення 119 (129.6.48.254 - єдиний NNTP -сервер, від якого мережа отримує новини, тому доступ до мережі відносно NNTP обмежений тільки цією системою). П'яте правило дозволяє трафік NTP, який використовує UDP, а не TCP, від будь-якого джерела до будь-якої системи в мережі. Нарешті, шосте правило блокує усі інші пакети - якщо цього правила не було б, маршрутизатор міг блокувати, а міг і не блокувати інші тиы пакетів. Це дуже простий приклад фільтрації пакетів. Справжні правила дозволяють здійснити складнішу фільтрацію і є гнучкішими.

Які протоколи фільтрувати

Рішення про те, які протоколи або групи портів фільтрувати, залежить від політики мережевого доступу, тобто від того, які системи повинні мати доступ до Інтернету і які типи доступу дозволені. Описані нижче сервіси потенційно уразливі до атак і зазвичай блокуються на брандмауері при вході в мережу або виході з неї.

Tftp, порт 69, спрощений FTP, використовуваний для завантаження ОС на бездисковых робочих станціях, термінальних серверах і маршрутизаторах, може також бути використаний для читання будь-якого файлу в системі при його неправильній установці.

RPC, порт 111, служби виклику видалених процедур, включаючи NIS і NFS, які можуть використовуватися для крадіжки системної інформації, включаючи паролі, а також читання і записи файлів

rlogin, rsh, rexec, порти 513, 514, 512, служби, які можуть при їх неправильній конфігурації привести до неавторизованого доступу в систему

Ряд інших засобів також зазвичай фільтрується або їх використання дозволяється тільки для тих систем, яким вони насправді потрібні. У це список входять:

TELNET, порт 23, часто дозволяється тільки для окремих систем

FTP, порти 20 і 21, аналогічно TELNET його використання дозволене тільки для окремих систем

SMTP, порт 25, часто дозволяється тільки для центрального поштового сервера

RIP, порт 520, протокол передачі інформації про маршрутизацію пакетів, може бути фальсифікований для перенаправлення пакетів

DNS, порт 53

UUCP, порт 540, UNIX - to - UNIX CoPy, при неправильній конфігурації може бути використаний для діставання неавторизованого доступу

NNTP, порт 119, протокол передачі мережевих новин, для доступу і читання мережевих новин

Gopher, http, порти 70 і 80,

Хоча деякі з цих служб, такі як TELNET і FTP, є небезпечними за своєю суттю, повне блокування доступу до інших може виявитися неприйнятним для багатьох організацій. Проте, не усі системи вимагають доступу до усіх служб. Наприклад, дозвіл доступу по TELNET і FTP з Інтернету тільки до тих систем, яким потрібний цей вид доступу, може поліпшити безпеку, не заподіюючи незручності користувачам. Такі служби, як NNTP, на перший погляд не представляють особливої небезпеки, але дозвіл цих служб тільки для тих систем, яким вони потрібні, допоможе створити більше впорядковане мережеве середовище і зменшить вірогідність їх використання такими, що атакують із-за ще невідомих вразливих місць.

Проблеми з маршрутизаторами з фільтрацією пакетів

Маршрутизатори з фільтрацією пакетів мають ряд недоліків. Правила фільтрації пакетів складно формулюються і зазвичай немає засобів для тестування їх коректності (окрім як ручне тестування). У деяких маршрутизаторів немає засобів протоколювання, тому якщо правила фільтрації пакетів все-таки дозволять небезпечним пакетам пройти маршрутизатора, такі пакети не зможуть бути виявлені до виявлення проникнення.

Часто вимагається зробити виключення з правил, щоб дозволити певні види доступу, які зазвичай блокуються. Але виключення з правил фільтрації іноді можуть зробити правила фільтрації такими складними, що вони стануть неконтрольованими. Наприклад, досить просто написати правило для блокування усіх з'єднань, що входять, до порту 23 (серверу TELNETа). Якщо ж робляться виключення, тобто якщо з деякими системами мережі дозволяється мати прямі з'єднання по TELNET, то має бути додане правило для кожної такої системи. Іноді додавання певних правил може ускладнити усю схему фільтрації. Як було вже сказано, тестування складного набору правил на їх коректність може виявитися дуже важким.

Деякі маршрутизатори з фільтрацією пакетів не фільтрують по порту TCP/UDP відправника, що може зробити набір правил фільтрації дуже складним і створити "діри" в схемі фільтрації. Якщо система ініціює SMTP -з’єднання з сервером, портом джерела буде випадково вибраний порт з номером більше 1024, а портом одержувача буде буде порт з номером 25, порт, який слухає сервер SMTP. Сервер повертатиме пакети з номером порту відправника 25, і номером порту одержувача, рівним випадково вибраному клієнтом номеру порту. Якщо в мережі дозволені ті, що входять і витікаючі SMTP -соединения, то маршрутизатор повинен дозволяти з'єднання з портами відправника і одержувача, великими 1023, в обох напрямах. Якщо маршрутизатор може фільтрувати по порту відправника, він може блокувати усі пакети, що входять в мережу організації, у яких порт одержувача більше 1023, а порт відправника не дорівнює 25. Якщо він не може фільтрувати пакети по порту відправника, маршрутизатор повинен дозволити з'єднання, які використовують порти відправника і одержувача більше 1024. Користувачі іноді можуть спеціально запустити сервера на портах, великих 1023, і обходити таким чином політику

фільтрації (тобто зазвичай сервер telnet в системі слухає порт 23, але може бути конфігурований так, що слухатиме замість цього порт 9876; і користувачі в Інтернеті зможуть організувати telnet -сеанс з цим сервером навіть, якщо маршрутизатор блокує з'єднання з портом призначення 23).

Іншою проблемою є те, що ряд служб RPC дуже важко заблокувати через те, що сервера для цих служб слухають порти, випадково вибирані в процесі завантаження системи. Служба, відома під назвою portmapper відображує первинні виклики служб RPC в призначені їм номери служб, але її еквіваленту не існує для маршрутизатора з фільтрацією пакетів. Оскільки маршрутизатору не можна повідомити, з яким портом працює служба, не можна повністю заблокувати ці служби, хіба що заблокувати повністю усі пакети UDP (RPC - службы в-основном використовують UDP). Блокування усіх пакетів UDP приведе до блокування ряду інших корисних служб, таких як DNS. Тому блокування RPC призводить до дилеми.

Маршрутизатори з фільтрацією пакетів з більш ніж двома інтерфейсами іноді не мають можливостей по фільтрації пакетів залежно від того, з якого інтерфейсу прийняті пакети, і куди мають бути спрямовані. Фільтрація пакетів, що входять і витікаючих, спрощує правила фільтрації пакетів і дозволяє маршрутизатору легко визначити, який IP -адрес справжній, а який - фальшивий. Маршрутизатори без такої можливості утрудняють реалізацію стратегій фільтрації.

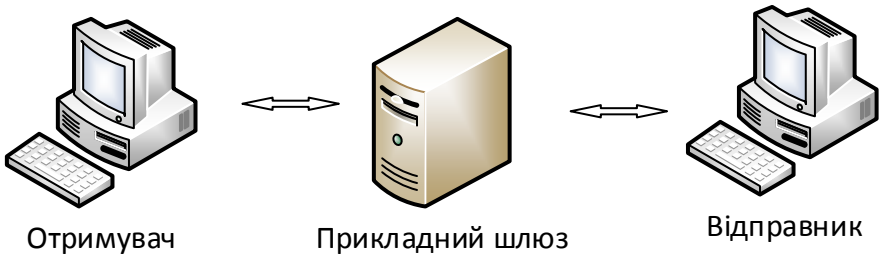
1.3 Прикладні шлюзи

Щоб захиститися від ряду вразливих місць, пов'язаних з маршрутизаторами з фільтрацією пакетів, в брандмауерах треба використовувати прикладні програми для перенаправлення і фільтрації з'єднань з такими службами, як TELNET і FTP. Таке застосування називається прокси-службой, а хост, на якому працює прокси-служба, - прикладним шлюзом. Прикладні шлюзи і маршрутизатори з фільтрацією пакетів можуть бути об'єднані для досягнення вищої безпеки і гнучкості, чим була б досягнута, якби вони використовувалися окремо.

Наприклад, розглянемо мережу, в якій блокуються з'єднання TELNET і FTP, що входять, за допомогою маршрутизатора з фільтрацією пакетів. Цей маршрутизатор дозволяє пропускати пакети TELNET або FTP тільки до однієї машини, прикладного шлюзу TELNET/FTP. Користувач, який хоче з'єднатися зовні з системою в мережі, повинен спочатку з'єднатися з прикладним шлюзом, а потім вже з потрібним хостом:

- спочатку користувач встановлює telnet -з’єднання з прикладним шлюзом і вводить ім'я внутрішнього хоста
- шлюз перевіряє IP -адрес користувача і дозволяє або забороняє з'єднання відповідно до того або іншого критерію доступу
- може знадобитися аутентифікація користувача(можливо за допомогою одноразових паролів)
- проксі-сервер створює telnet -з’єднання між шлюзом і внутрішнім хостом

- проксі-сервер передає дані між цими двома з'єднаннями
- прикладний шлюз протоколює з'єднання



Малюнок 2.4 Віртуальні з'єднання, що реалізовуються за допомогою прикладного шлюзу і проксі-служб

Цей приклад демонструє декілька переваг використання проксі-служб. По-перше, проксі- служби дозволяють тільки ті служби, для яких є проксі. Іншими словами, якщо прикладний шлюз містить проксі для FTP і TELNET, то в підмережі, що захищається, будуть дозволені тільки FTP і TELNET, а інші служби будуть повністю блоковані. Для деяких організацій такий вид безпеки важливий, оскільки гарантує, що тільки ті служби, які вважаються безпечними, пропускатимуться через брандмауер. Цей підхід також оберігає від можливості розробки нових небезпечних служб без повідомлення адміністраторів брандмауера.

Іншою перевагою використання проксі-служб є те, що може бути здійснена фільтрація протоколів. Наприклад, деякі брандмауери, можуть фільтрувати ftp - соединения і забороняти використання команди FTP put, що було б корисно для отримання гарантій того, що користувачі не можуть, наприклад, писати на анонімний FTP -сервер.

Прикладні шлюзи мають ряд серйозних переваг в порівнянні із звичайним режимом, при якому прикладний трафік пропускається безпосередньо до внутрішніх хостам. Вони включають:

- приховання інформації, при якому імена внутрішніх систем необов'язково будуть відомі зовнішнім системам за допомогою DNS, оскільки прикладний шлюз може бути єдиним хостом, чиє ім'я має бути відоме зовнішнім системам.
- надійна аутентифікація і протоколювання, при якому прикладний трафік може бути попередній аутентифицирован до того, як він досягне внутрішніх хостов, і може бути запротокольований ефективніше, ніж стандартні засоби протоколювання хоста.
- оптимальне співвідношення між ціною і ефективністю через те, що додаткові програми або устаткування для аутентифікації або протоколювання треба встановлювати тільки на прикладному шлюзі.
- прості правила фільтрації, оскільки правила на маршрутизаторі з фільтрацією пакетів будуть менш складними, чим вони були б, якби

маршрутизатор сам фільтрував прикладний трафік і відправляв його великому числу внутрішніх систем. Маршрутизатор повинен тільки пропускати прикладний трафік до прикладного шлюзу і блокувати увесь інший трафік.

Недолік прикладного шлюзу полягає в тому, що при використанні клієнт-серверних протоколів, таких як TELNET, потрібно двокрокову процедуру для входження всередину або виходу назовні. Деякі прикладні шлюзи вимагають модифікованих клієнтів, що може розглядатися або як недолік, або як перевагу, залежно від того, чи роблять модифіковані клієнти легшим використанням брандмауера. Прикладний шлюз TELNET необов'язково вимагає модифікованого клієнта TELNET, проте він вимагає іншої логіки дій від користувача: користувач повинен встановити з'єднання (але не сеанс) з брандмауером, а не безпосередньо встановити сеанс з хостом. Але модифікований клієнт TELNET робить брандмауер прозорим, дозволяючи користувачеві вказати кінцеву систему (а не брандмауер) в команді TELNET. Брандмауер є як би дорогим до кінцевої системи і тому перехоплює з'єднання, а потім виконує додаткові кроки, такі як запит одноразового пароля. Користувачеві не треба в цьому випадку нічого робити, але на кожній системі має бути встановлений модифікований клієнт.

Окрім TELNET, зазвичай прикладні шлюзи використовуються для FTP і електронної пошти, а також X Windows і ряду інших служб. Деякі прикладні шлюзи FTP мають можливості блокування команд get і put для деяких хостов. Наприклад, зовнішній користувач, FTP, що встановив, сеанс (через прикладний шлюз FTP) з внутрішньою системою, такою, як анонімний FTP - сервер, може спробувати скопіювати файли на сервер. Прикладний шлюз може фільтрувати FTP -протокол і блокувати усі команди put для анонімного FTP - сервера; це дозволить гарантувати, що ніхто не зможе завантажити на сервер чого-небудь, і дасть великі гарантії, чим проста упевненість в тому, що права доступу до файлів на анонімному FTP -сервере встановлені коректно(деякі організації ввели політики, в яких забороняються команди get і put для певних директорій; наявність брандмауера, FTP, що фільтрує, -команды, була б особлива корисно в цій ситуації. Деякі місця заборонили команди get для зовнішніх хостов, щоб користувачі не могли рахувати інформацію або програми із зовнішніх хостов. У інших же мережах заборонена команда put для зовнішніх хостов, щоб користувачі не могли зберегти локальну інформацію на зовнішніх FTP -серверах. Але типовим є варіант. Коли забороняються команди put, що входять, щоб зовнішні користувачі не могли писати на FTP -сервера в мережі)

Прикладний шлюз для електронної пошти служить для централізованого збору електронної пошти і поширення її по внутрішніх хостам і користувачах. Для зовнішніх користувачів усі внутрішні користувачі матимуть адресу виду пользователь@почтовый хост, де поштовий хост - ім'я шлюзу для пошти. Шлюз повинен приймати пошту від зовнішніх користувачів, а потім переправляти її на інші внутрішні системи. Користувачі, що посилають електронні листи з внутрішніх систем, можуть посилати їх безпосередньо з внутрішніх систем, або,

якщо внутрішні імена систем не відомі зовні мережі, лист має бути посланий на прикладний шлюз, який потім переправить його до хосту призначення. Деякі поштові шлюзи використовують безпечнішу версію програми sendmail для прийому пошти.

2. Приклад налаштування брандмауера Windows 7

1. Два комп'ютери, підключені один до одного безпосередньо або через концентратор або комутатор.
2. ОС Windows 7, встановлена на обох комп'ютерах.
3. Комп'ютери повинні знаходитися в одній робочій групі і мати загальну маску підмережі.

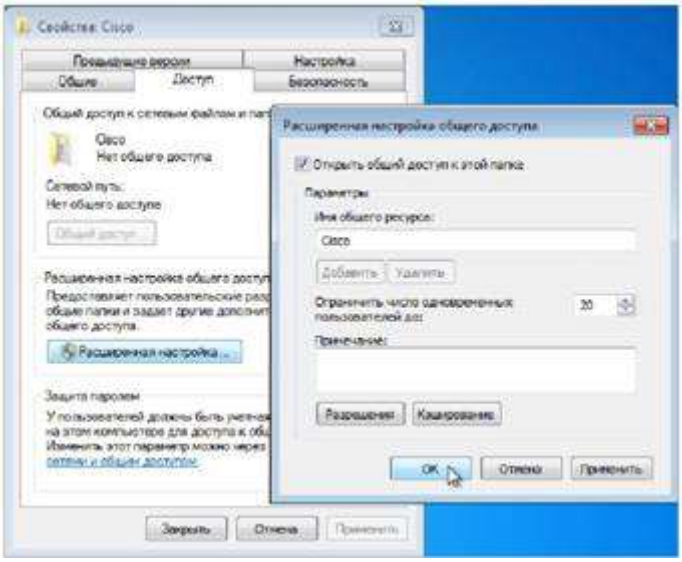
Порядок виконання роботи:

1. На комп'ютері 1 клацніть правою кнопкою миші на робочому столі і виберіть **Создать > Папку**.

Задайте ім'я новій теці - Test1.

Клацніть правою кнопкою миші на теці Test1 і виберіть **Предоставить общий доступ > Расширенная настройка общего доступа > Расширенная настройка общего доступа**.

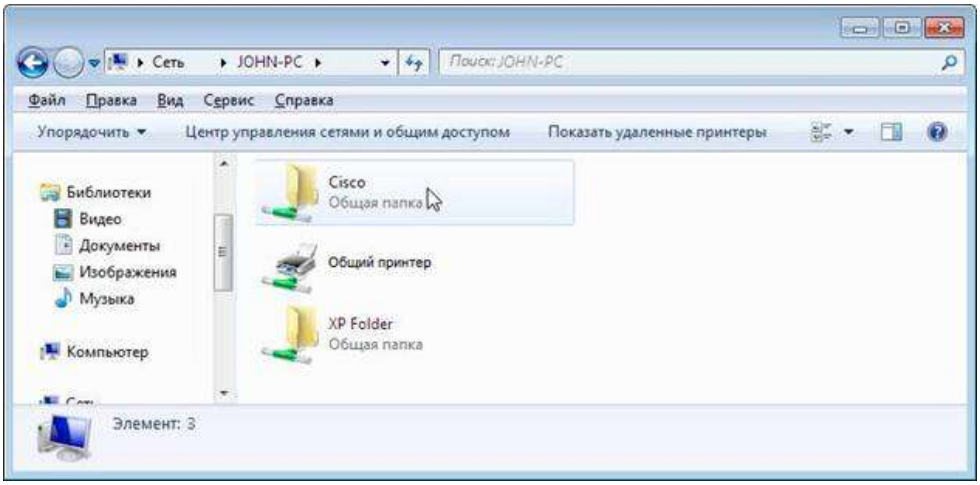
Відкриється вікно «Расширенная настройка общего доступа».



Надайте загальний доступ до цієї теки, використовуючи ім'я за умовчанням Test1.

На комп'ютері 2 натисніть кнопку **Пуск**, потім виберіть **Панель управления > Центр управления сетями и общим доступом > Сеть**.

Двічі клацніть компьютер 1.



Чи видно тепер загальну теку Test1?

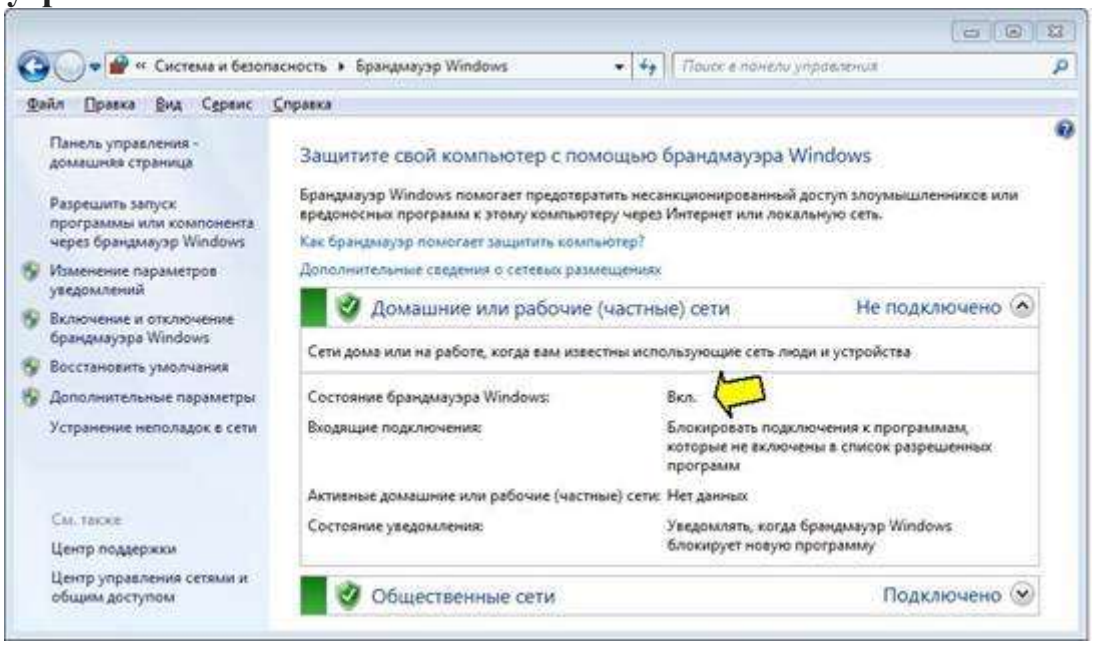
Примітка. Якщо відповідь негативна, звернетеся по допомогу до інструктора.

Закрийте **Сеть.**

Примітка. При виконанні частини лабораторної роботи, що залишилася, використайте комп'ютер 1, якщо не вказано інакше.

2. Перейдіть до брандмауера Windows 7.

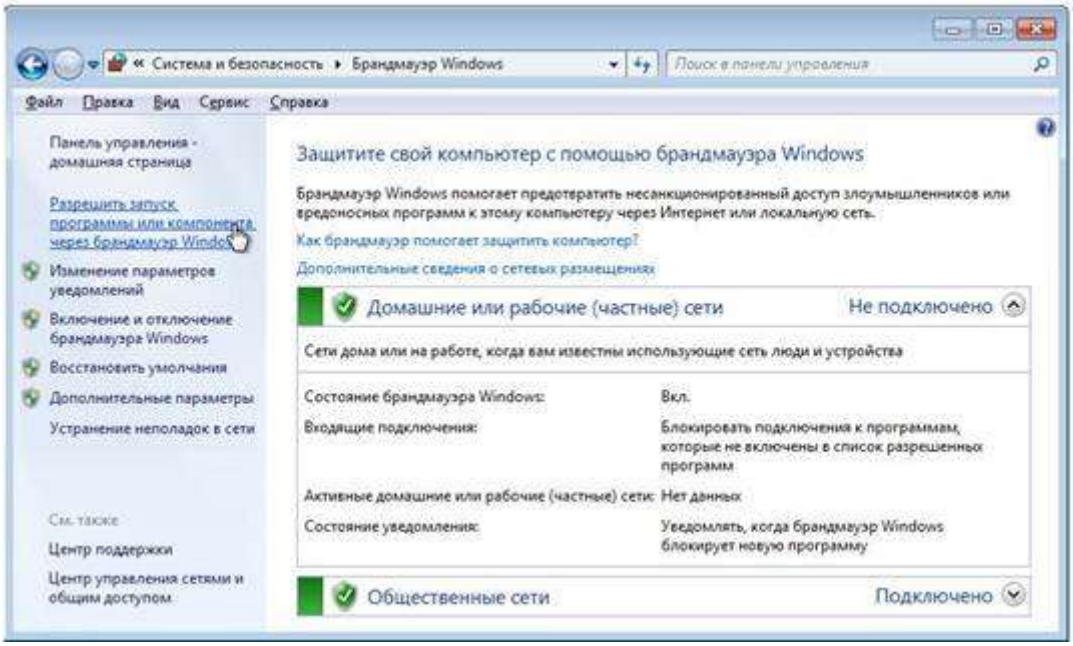
Виберіть **Пуск > Панель управления > Система и безопасность > Брандмауэр Windows**.



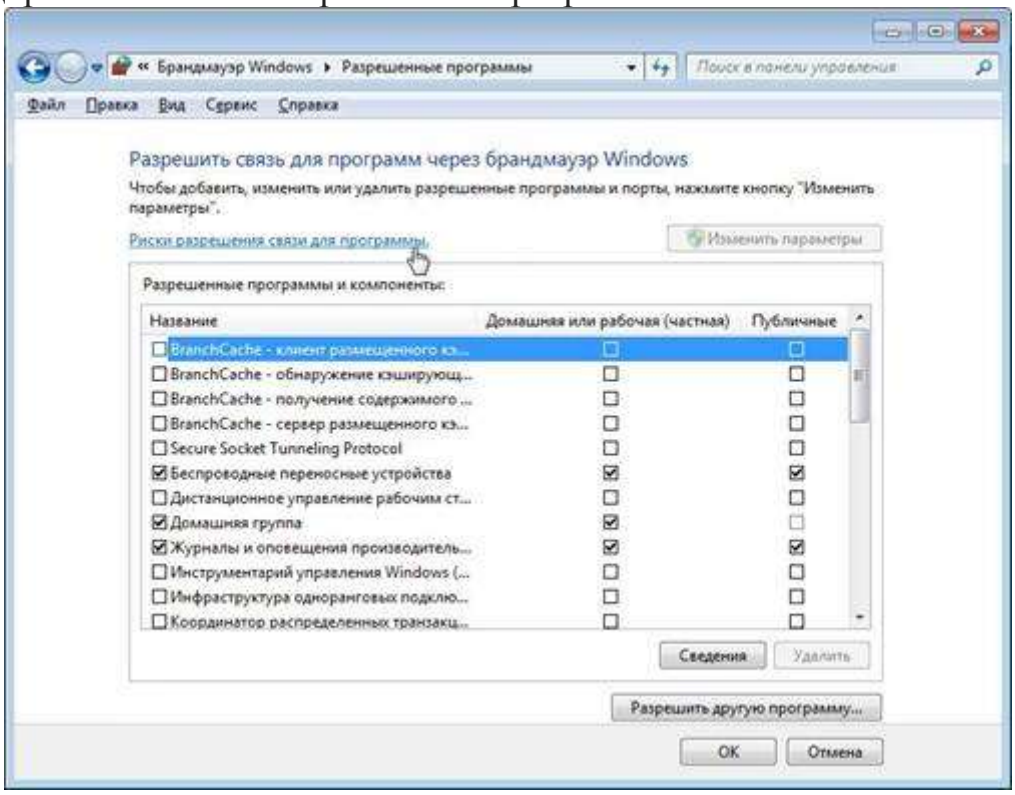
Індикатор брандмауера показує стан брандмауера. Стандартная настройка – «**Включен**».

Вкажіть переваги брандмауера Windows

3. Перейдіть за посилкою **Разрешить запуск программы или компонента через брандмауэр Windows**.



4. Відкриється вікно «Разрешенные программы».



Програми і служби, що не блокуються брандмауером Windows, будуть помічені прапорцями.

Можна додавати додатка до цього списку. Це може бути необхідно, якщо у клієнта є додаток, що вимагає зв'язку із зовнішньою мережею, але по какой- те причині брандмауер Windows не може виконати налаштування автоматично. Для завершення цієї процедури необхідно увійти до системи на цьому комп'ютері в якості адміністратора.

Перейдіть по посиланню **Риски разрешения связи для программы.**

Відкриється вікно «Справка и поддержка».

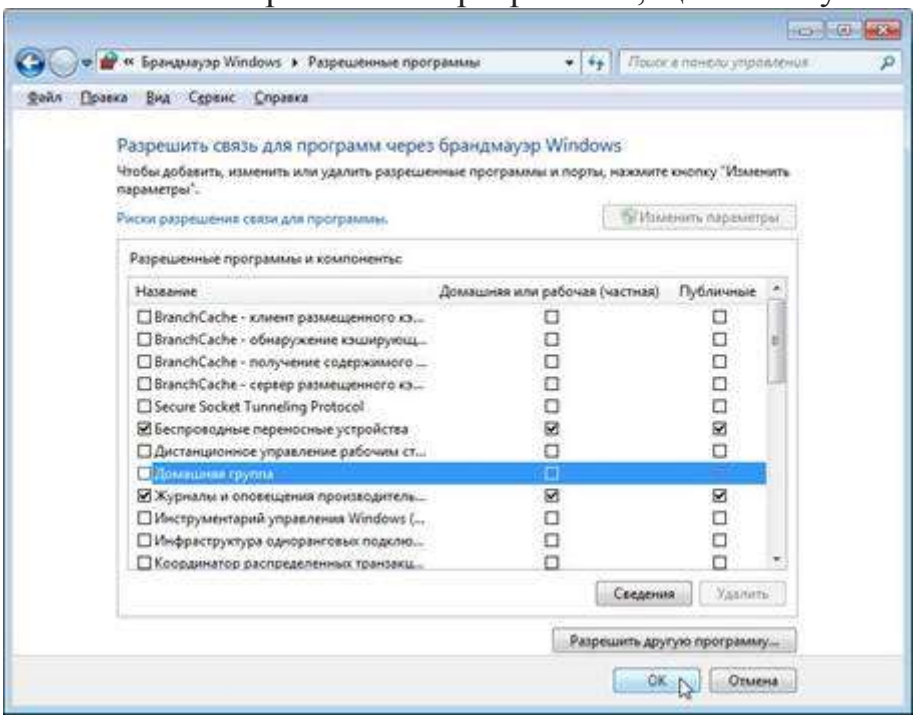
Створення занадто великого числа виключень у файлі "Програми і служби" може спричинити негативні наслідки.

Опишіть негативні наслідки великої кількості виключень.

Закрийте вікно «Справка и поддержка».

5. З комп'ютера 1 виконаєте наступні дії:

Клацніть вікно окно «Разрешенные программы», щоб активувати його.

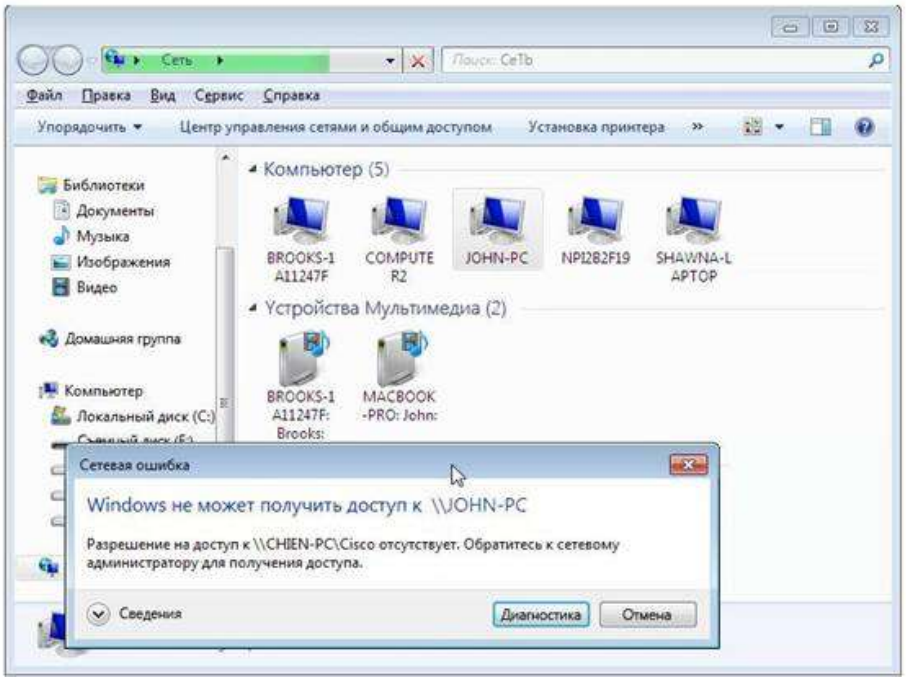


Для відключення виключення зніміть прапорець **Общий доступ к файлам и принтерам > нажмите кнопку «ОК».**

З комп'ютера 2 виконаєте наступні дії:

Відкрийте мережеве підключення до комп'ютера 1.

Послідовно виберіть **Пуск > Панель управления > Центр управления сетями и общим**



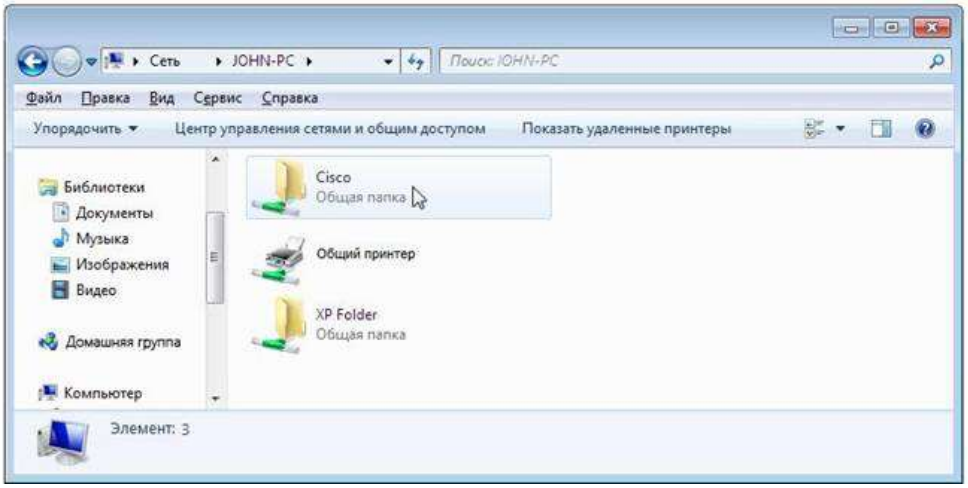
Чи можна підключитися до комп'ютера 1?

З комп'ютера 1 виконаєте наступні дії:

Для включення - виключення встановите прапорець **Общий доступ к файлам и принтерам > нажмите кнопку «ОК».**

З комп'ютера 2 виконаєте наступні дії:

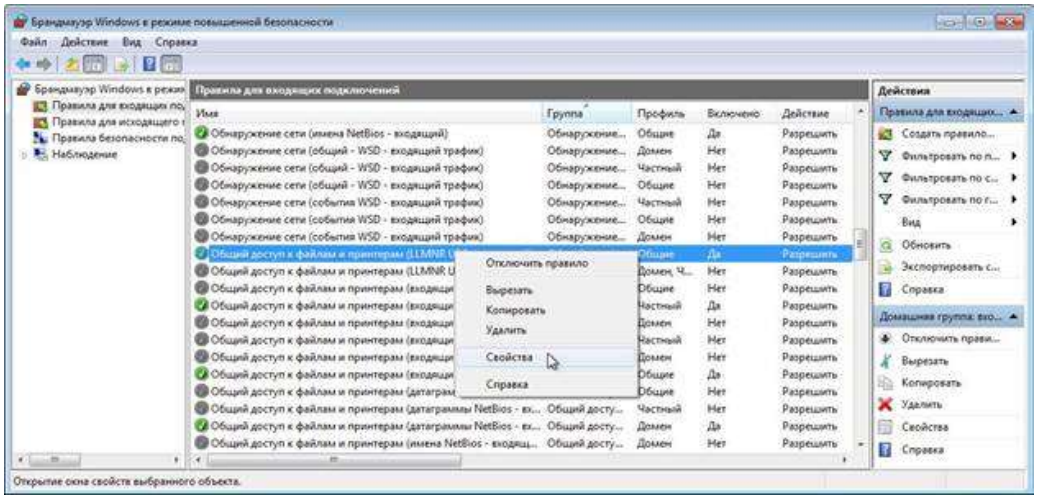
Оновіть вікно **Сеть** і підключіться до комп'ютера 1.



Чи можна підключитися до комп'ютера 1?

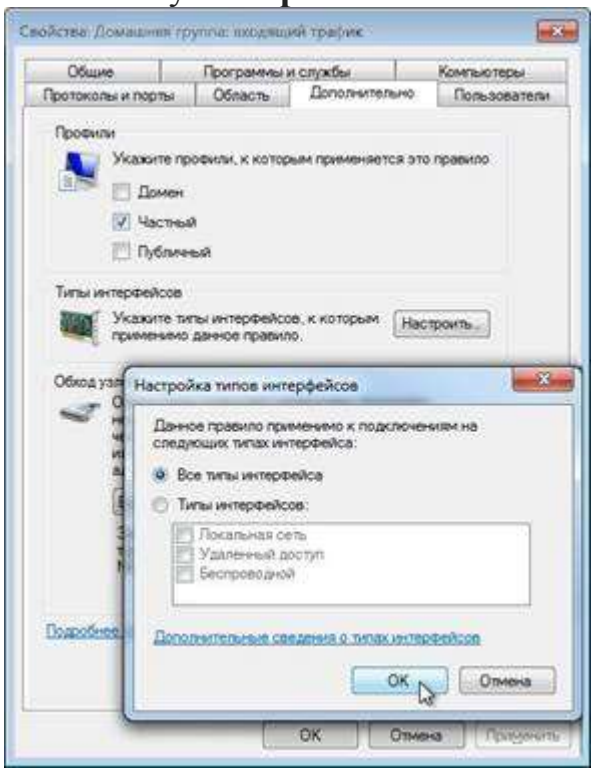
Завершіть сеанс на комп'ютері 2. Використайте комп'ютер 1 в тій частині лабораторної роботи, що залишилася.

6. Виберіть **Пуск > Панель управления > Система и безопасность > Администрирование > Брандмауэр Windows в режиме повышенной безопасности > Правила для входящих подключений.**



Розгорніть вікно, щоб можна було побачити повне ім'я правил для підключень, що входять. Знайдіть «Общий доступ к файлам и принтерам (эхо-запрос – входящий трафик ICMPv4)».

Клацніть правило правою кнопкою миші, виберіть **Свойства > вкладка Дополнительно >** натисніть кнопку **Настроить**.

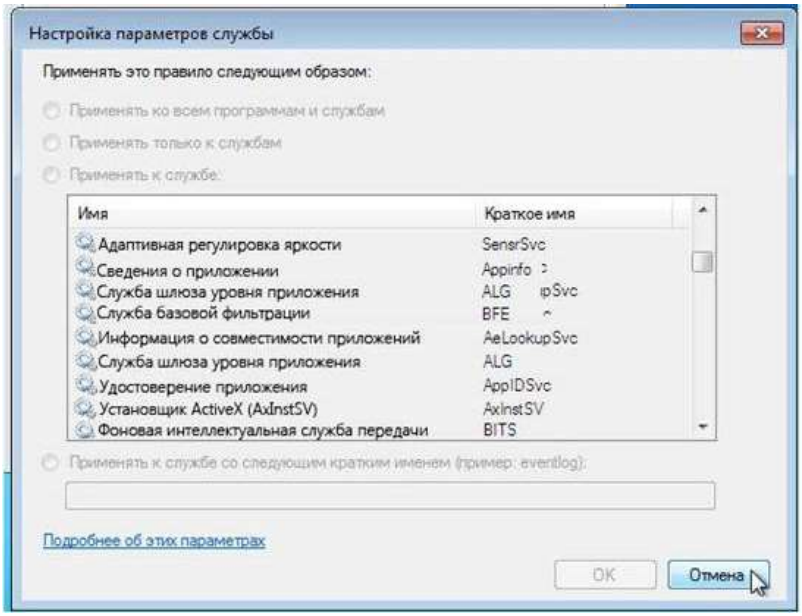


На вкладці «Дополнительно " відображається профіль (-і), використовуваний комп'ютером, а у вікні «Настройка типов интерфейсов» відображаються різні підключення, налаштовані на комп'ютері.

Натисніть кнопку **ОК**.

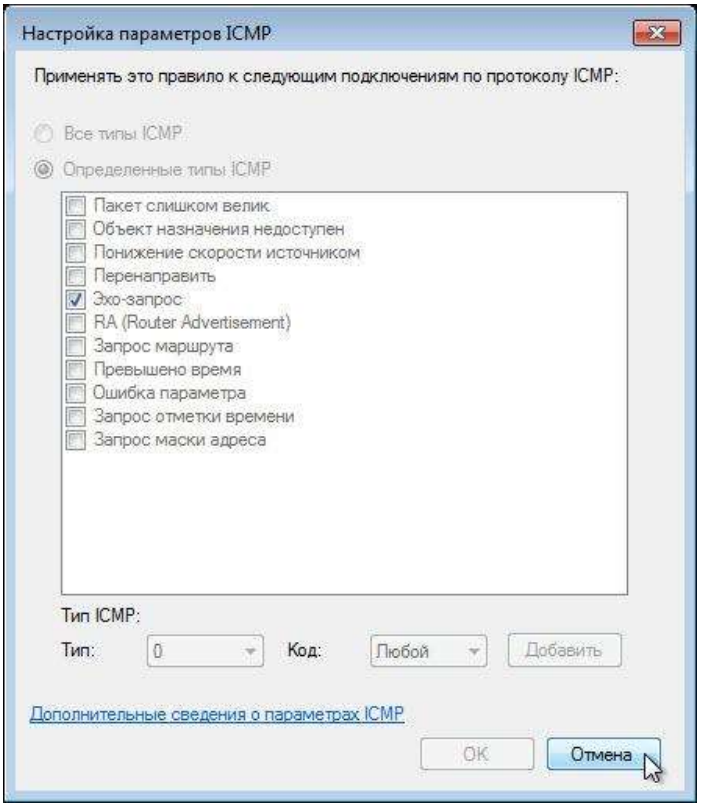
Перейдіть на вкладку **Программы и службы**.

Відкриється вікно «Настройка параметров службы».



Перерахуйте короткі імена чотирьох доступних служб.
Натисніть кнопку **Отмена**.

7. Існує безліч додатків, зазвичай непомітних для користувача, яким необхідно мати доступ до комп'ютера через брандмауер Windows. Це команди рівня мережі, що направляють трафік в мережі і Інтернеті.Перейдіть на вкладку **Протоколы и порты**. Для налаштування ICMP натисніть кнопку **Настроить**. Можна буде побачити меню, в якому настроюються виключення ICMP.



В даному прикладі дозвіл вхідних ехо-запитів дозволяє користувачам мережі перевірити, чи є присутнім в ній комп'ютер. Він також дозволяє бачити, наскільки швидко передається інформація до комп'ютера і від нього.

Перерахуйте кілька типів ICMP.

Закрийте усі вікна.

Порядок виконання роботи

1. Розглянути які існують програмні засоби захисту.
2. Визачити для чого використовуються firewall’и.
3. Здійснити дії наведені в пункті 2.
4. Дослідити стан захисту операційної системи для трьох випадків.
    a. без увімкнутого брандмауера,
    b. використовуючи вбудований в ОС брандмауер,
    c. Встановити інший вільно обраний брандмауер)
5. Описати принцип роботи і політики безпеки обраного брандмауера
6. Навести приклади журналу фільтрації мережевого трафіку
7. Для визначення ефективності кожного брандмауера скористатись сканером безпеки
8. Визначити можливості і недоліки використовуваного брандмауера.
9. Дані по результатах дослідження оформити у вигляді таблиці
10. Оформити звіт по роботі

Лабораторна робота №3.

Тема: ”Розробка і дослідження засобів ідентифікації користувачів в комп’ютерних системах ”

Мета роботи: засвоїти методику та отримати практичні навики побудови засобів ідентифікації користувачів.

1. Теоретичні відомості

Ідентифікація об’єкта – це одна з функцій підсистеми захисту. Перед тим, як отримати доступ до комп’ютерної системи (КС), користувач повинен ідентифікувати себе, після чого засоби захисту повинні підтвердити, чи даний користувач є насправді тим, за кого себе видає. Програма ідентифікації призначена для одноразового встановлення особи користувача та надання йому прав доступу в систему. Одним з найпростіших способів ідентифікації є *парольна ідентифікація*, яка здійснюється шляхом порівняння введеного імені та паролю, із тими які зберігаються у файлі паролів (рис.3.1).

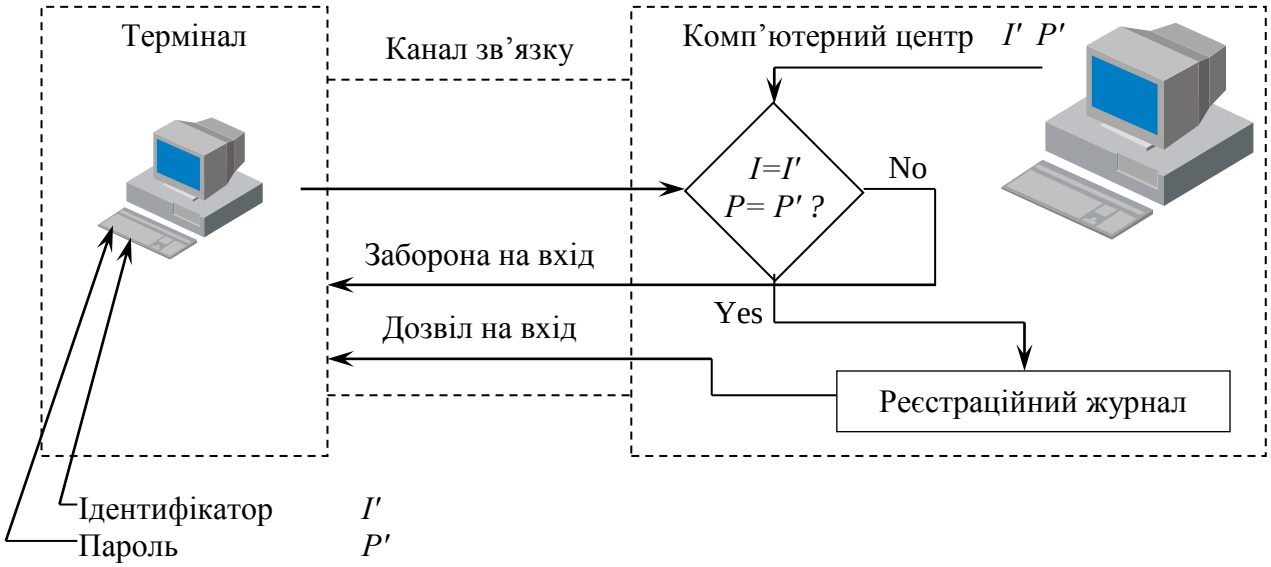


Рисунок 3 1. Класичний спосіб парольної ідентифікації

Проте такий підхід не захищає КС від програмних та апаратних засобів сканування клавіатури та ліній передачі, а отже може призвести до витоку конфіденційної інформації. Тому, як правило, в сучасних КС пароль не передається в явній формі по лініях передачі, а натомість в якості паролю використовують якесь його відображення використовуються важко оборотні однонапрямлені функції, застосування яких гарантує неможливість розкриття пароля за його відображенням за розумний час.

В такому випадку процедура ідентифікації описується таким алгоритмом (рис. 2):

1. Користувач вводить свій ідентифікатор
2. Засоби ідентифікації переглядають список зареєстрованих ідентифікаторів. Якщо ідентифікатор не зареєстрований, - то виводиться повідомлення, що такий користувач в системі не зареєстрований і далі перехід на крок 1, або ж завершення роботи

програми входу в систему. Якщо ж ідентифікатор зареєстрований, - то перехід на крок 3.

3. КС генерує випадкове число x, та обчислює значення важкооборотної однонапрямленої функції y, яка використовується в системі для відображення паролю користувача.
4. Число x передається користувачу.
5. Користувач обчислює значення важкооборотної функції y' та передає його в КС.
6. КС порівнює значення y і y'. Якщо вони співпадають то КС дозволяє вхід користувача в систему. В інакшому випадку видається повідомлення про помилку вводу паролю, перехід на крок 3, або ж завершення роботи програми входу в систему.

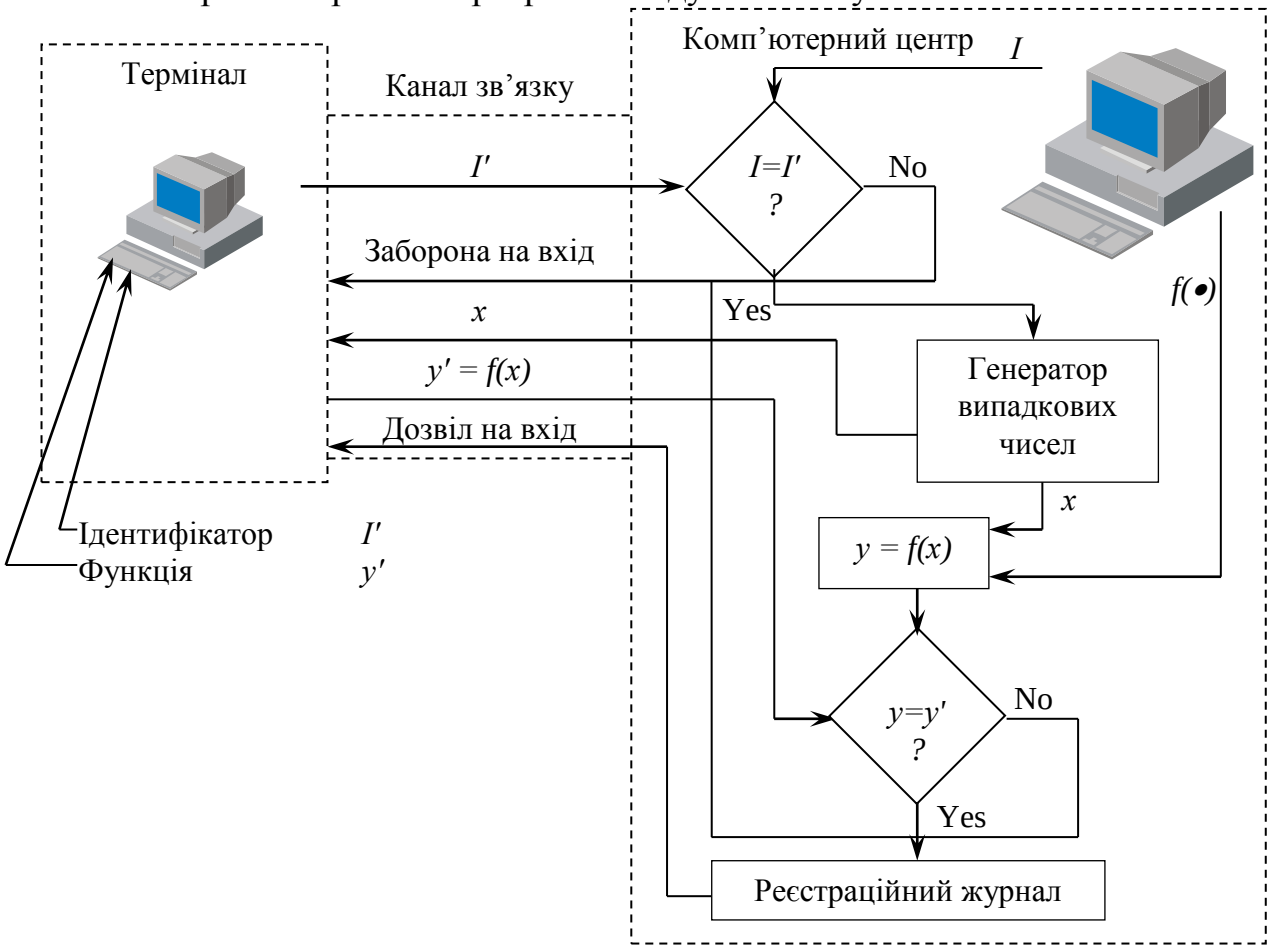


Рисунок 3.2. Ідентифікація користувача за допомогою важкооборотної функції відображення паролю

Зрозуміло, що стійкість такої КС до інтерполяції використовуваної функції визначається важкооборотністю функції y та частотою генерації і розподілу ключів в системі.

Іншою важливою складовою частиною КС є програма реєстрації. *Програма реєстрації* призначена для реєстрації або видалення користувачів в Реєстраційному журналі системи із наданням їм певних правил доступу. Право реєстрації або видалення належить лише одному користувачу – адміністратору системи. Імена користувачів, їх паролі та права доступу зберігаються в явному вигляді в файлі. Розробник повинен оцінити розмір реєстраційного журналу, виходячи із заданих параметрів. Така інформація буде корисною для оцінки

трудомісткості процедур сортування та фільтрування. Крім того, на основі отриманих даних розробник може дати рекомендації адміністратору КС стосовно регулярності процедур генерування та розподілу ключів.

2. Порядок виконання роботи

1. Ознайомитись з викладеним вище матеріалом.
2. Отримати індивідуальне завдання**.**
3. Відповідно до завдання розробити структуру заданої КС.
4. Провести розрахунки параметрів системи.
5. Розробити необхідне алгоритмічне забезпечення.
6. Реалізувати задану КС.
7. Скласти звіт з виконання лабораторної роботи та захистити його.

2. Зміст звіту

1. Назва та зміст лабораторної роботи.
2. Структурна схема та параметри розробленої КС.
3. Текст програми основних модулів КС.
4. Приклади результатів роботи програми.
5. Реєстраційний журнал.
6. Висновки.

3. Індивідуальні завдання.

Розробити програму ідентифікації користувачів, котра б фіксувала вхід (вихід) користувачів в Реєстраційному журналі. Необхідні параметри програми наведені в таблиці.

Варіант	Кількість користувачів	Кількість реалізованих функцій	Функція криптування пароля
1.	7	>10	*ln(a*x)*
2.	8	>10	*exp(-a*x)*
3.	9	>10	*a*sin(x)*
4.	10	>10	*a*ln(x)*
5.	8	>10	*a/x*
6.	7	>10	*ln(a/x)*
7.	10	>10	*x/sin(a)*
8.	9	>10	*a* sin(1/x)*
9.	8	>10	*tg(a*x)*
10.	7	>10	*a*ln(2+x)*

4. Контрольні запитання.

1. Що таке ідентифікація користувачів в КС?
2. Які види ідентифікації вам відомі?
3. Що таке парольна ідентифікація?
4. Які класичні недоліки класичного способу парольної ідентифікації?
5. Що таке реєстраційний журнал? Для їх використовують?

Лабораторна робота №4.

Тема: ” Розробка і дослідження засобів аутентифікації користувачів в комп’ютерних системах ”

Мета роботи: засвоїти методику та отримати практичні навики побудови засобів аутентифікації користувачів.

1. Теоретичні відомості

Аутентифікація - полягає в періодичній (стохастичній) перевірці достовірності ідентифікації користувача. Така процедура проводиться для повторної перевірки користувача. Аутентифікація може здійснюватись як апаратними, так і програмними методами за якимись особистими ознаками, чи персональними відомостями користувача.

При апаратній реалізації користувач може бути аутентифікованим за певними фізичними ознаками: вага тіла, колір очей, відбитки пальців, геометрія долоні, код ДНК і т.п. Окрім того, можуть використовуватись додаткові особисті пристрої: наручні браслети, ключі і т.д. Даний вид аутентифікації характеризується вищим рівнем надійності, проте є складнішим та дорожчим у використанні, тому він використовується на підприємствах, де необхідно забезпечити високий рівень захисту інформації.

Дешевший варіант аутентифікації користувачів полягає у створенні програмних засобів. Резидентна програма періодично з певним кроком часу задає випадковим чином запитання із заздалегідь створеного файлу, або ж випадкові три- , чотири- розрядні десяткові числа. КС порівнює відповіді з наперед зареєстрованими, або ж обчисленими відповідями, і на основі цього надає, або забороняє роботу користувача. У випадку правильної відповіді за користувачем залишаються його права, а у випадку неправильної відповіді - користувач втрачає права доступу і повинен заново ввійти в систему. Стійкість даного виду аутентифікації забезпечується конфіденційністю інформації

Наведемо основні способи аутентифікації користувачів:

- наперед визначена інформація, якою може користуватися користувач: пароль, персональний ідентифікаційний номер, домовленість про використання спеціальних закодованих фраз;
- елементи апаратного забезпечення, якими може користуватися користувач: ключі, магнітні картонки, мікросхеми і т.п.;
- характерні особисті ознаки користувача: відбитки пальців, рисунок ставки ока, тембр голосу і т.п.;
- характерні навики та риси поведінки користувача в режимі реального часу: особливості динаміки та стиль роботи на клавіатурі, прийоми роботи з маніпулятором і т.п.;
- навики та знання користувачів, обумовлені освітою, культурою, навчанням, вихованням, звичками і т.п.

Процедура аутентифікації користувачів може бути реалізована як з постійним, так і з адаптивним періодом повтору. Постійний період повтору використовується в тих КС, в яких частота появи користувачів в системі та інтенсивність їх роботи є приблизно рівномірною. При виборі періоду процедури

аутентифікації слід керуватися такими міркуваннями: при досить великому періоду збільшується імовірність НСД, а при досить малому - зменшується ефективність роботи користувачів, оскільки вони постійно відволікаються від виконання основних своїх обов'язків. В системах, до яких ставляться вимоги підвищеної захищеності можуть застосовуватися засоби аутентифікації з адаптивним періодом повтору. Період повтору в таких КС визначається як інтенсивністю роботи користувачів, так і спробами НСД.

Окрім того, після встановлення достовірності ідентифікації користувача виконується реєстрація в часі всіх дій користувача в Операційному журналі системи. В такому журналі окрім записів санкціонованого використання тих, чи інших ресурсів системи, можуть накопичуватися дані про спроби несанкціонованого доступу користувачів з автоматичною сигналізацією адміністратору системи для прийняття організаційних заходів з метою виявлення порушників. При створенні операційного журналу слід пам'ятати, що різні типи користувачів мають доступ до різних типів ресурсів.

Для періодичної аутентифікації зручно використовувати переривання системного таймера INT 1Ch. Слід пам'ятати, що програма аутентифікації повинна заборонити усі інші види переривань.

INT 10h використовується для тимчасового очищення екрану шляхом використання функцій прокрутки, чи перемиканням на іншу відео-сторінку

INT 1Ah використовується для отримання точних значень системної дати та часу.

Загальна структура КС аутентифікації приведена на рис.3.

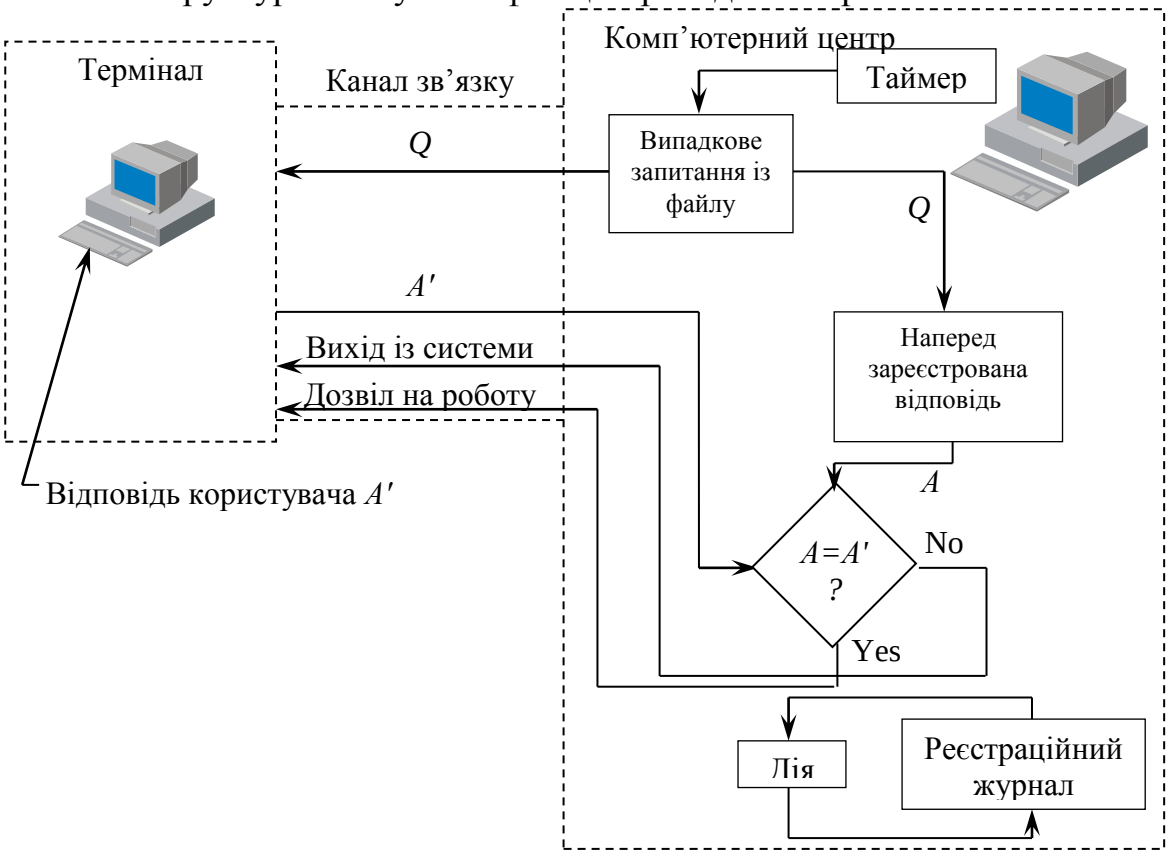


Рис. 3. Загальна структура КС аутентифікації

2. Порядок виконання роботи

1. Ознайомитись з викладеним вище матеріалом.
2. Отримати індивідуальне завдання.
3. Відповідно до завдання розробити структуру заданої КС.
4. Провести розрахунки параметрів системи.
5. Розробити необхідне алгоритмічне забезпечення.
6. Реалізувати задану КС.
7. Скласти звіт з виконання лабораторної роботи та захистити його.

3. Зміст звіту

1. Назва та зміст лабораторної роботи.
2. Структурна схема та параметри розробленої КС.
3. Текст програми основних модулів КС.
4. Приклади результатів роботи програми.
5. Операційний журнал.
6. Висновки.

4. Індивідуальні завдання.

Розробити програму аутентифікації користувачів, котра б дії користувачів в Операційному журналі. Необхідні параметри програми наведені в таблиці.

Варіант	Кількість запитань	Період повтору процедури аутентифікації, хв	Кількість запитань в одній ітерації процедури аутентифікації
1	15	2	3
2	14	2,5	4
3	7	3	2
4	8	3,5	3
5	9	4	4
6	10	4,5	2
7	4	1,5	3
8	12	5	4
9	6	1	2
10	11	5,5	3

5. Контрольні запитання.

1. Що таке аутентифікація користувачів в КС?
2. Які види аутентифікація вам відомі?
3. Назвіть основні способи аутентифікації користувачів в КС за допомогою програмних засобів.
4. Назвіть основні способи аутентифікації користувачів в КС за допомогою апаратних засобів.
5. Що таке операційний журнал?
6. Для чого використовують операційний журнал?

Лабораторна робота №5.

Тема: ”Дослідження алгоритмів криптографічного захисту на основі підстановок та перестановок. Блочні шифри”

Мета роботи: Розглянути принцип роботи і алгоритми різних блочних шифрів.

1. Теоретичні відомості

Класифікація

Шифрсистеми класифікуються за різними ознаками:

- по видах інформації (текст, мова, відеоінформація), що захищається,
- по криптографічній стійкості,
- за принципами забезпечення захисту інформації (симетричні, асиметричні, гібридні),
- за конструктивними принципами (блокові і потокові) і ін.

При побудові шифру використовуються з математичної точки зору два види відображень:

- перестановки елементів відкритого тексту
- заміни елементів відкритого тексту на елементи деякої множини.

У зв'язку з цим безліч шифрів ділиться на 3 види:

- шифри перестановки,
- шифри заміни
- композиційні шифри, що використовують поєднання перестановок і замін.

Блочні шифри.

Алгоритм Lucifer

В кінці шестидесятих років корпорація IBM запустила дослідницьку програму по комп'ютерній криптографії, названу Lucifer (Люцифер) і керовану спочатку Хорстом Файстелем (Horst Feistel), а потім Уолтом Тачменом (Walt Tuchman). Таке ж ім'я - Lucifer - одержав блоковий алгоритм, що з'явився в результаті цієї програми на початку сімдесятих років. Насправді існує, щонайменше, два різні алгоритми з таким ім'ям. Один з них містить ряд пропусків в специфікації алгоритму. Все це привело до помітної плутанини.

Алгоритм Lucifer є мережею перестановок і підстановок, його основні блоки нагадують блоки алгоритму DES. У DES результат функції f складається операцією XOR з входом попереднього раунду, утворюючи вхід наступного раунду. У S-блоків алгоритму Lucifer 4-бітові входи і виходи, вхід S-блоків є перетасованим виходом S-блоків попереднього раунду, входом S-блоків першого раунду служить відкритий текст. Для вибору використовуваного S-блоку з двох можливих використовується біт ключа. (Lucifer реалізує все це в єдиному T-блоці з 9 бітами на вході і 8 бітами на виході). На відміну від алгоритму DES, половини блоку між раундами не переставляються, та і саме поняття половини блоку в алгоритмі Lucifer не використовується. У цього

алгоритму 16 раундів, 128-бітові блоки і простіша, ніж в DES, схемі розгортки ключа.

Деякі вважають, що Lucifer надійніше за DES через більшу довжину ключа і нечисленності опублікованих відомостей. Але очевидно, що це не так.

Алгоритм Madryga

У. Е. Мадріга (W. E. Madryga) запропонував цей блоковий алгоритм в 1984 році. Його можна ефективно реалізувати програмним шляхом: у алгоритмі немає дратівливих перестановок, і всі операції виконуються над байтами.

Варто перерахувати завдання, які вирішував автор при проектуванні алгоритму:

- Без допомоги ключа відкритий текст неможливо одержати з шифртекста. (Це означає тільки те, що алгоритм стійок).
- Число операцій, необхідних для відновлення ключа за зразком шифртекста і відкритого тексту, повинно бути статистично рівно твору числа операцій при шифруванні на число можливих ключів. (Це означає, що ніякий розтин з відкритим текстом не може бути ефективніше лобового розтину).
- Публікація алгоритму не впливає на стійкість шифру. (Безпека повністю визначається ключем).
- Зміна одного біта ключа повинна радикально змінювати шифртекст одного і того ж відкритого тексту, а зміна одного біта відкритого тексту повинна радикально змінювати шифртекст для того ж ключа (лавинний ефект).
- Алгоритм повинен містити некомутативну комбінацію підстановок і перестановок.
- Підстановки і перестановки, використовувані в алгоритмі, повинні визначатися як вхідними даними, так і ключем.
- Надмірні групи бітів відкритого тексту повинні бути повністю замасковані в шифртексте.
- Довжина шифртекста повинна співпадати з довжиною відкритого тексту.
- Між будь-якими можливими ключами і особливостями шифртекста недопустимі прості взаємозв'язки.
- Всі можливі ключі повинні забезпечувати стійкість шифру. (Не повинно бути слабких ключів).
- Довжина ключа і текст повинні мати можливість варіювання для реалізації різних вимог до безпеки.
- Алгоритм повинен допускати ефективну програмну реалізацію на мейнфреймах, міні- і микрокомпыотерах і за допомогою дискретної логіки. (По суті, число функцій, використовуваних в алгоритмі, обмежено операціями XOR і бітовим зрушенням).

Опис алгоритму Madryga

Алгоритм Madryga складається з двох вкладених циклів. Зовнішній цикл повторюється вісім разів (для гарантії надійності число циклів можна збільшити)

і полягає в застосуванні внутрішнього циклу до відкритого тексту. Внутрішній цикл перетворює відкритий текст в шифртекст і виконується одноразово над кожним 8-бітовим блоком (байтом) відкритого тексту. Таким чином, весь відкритий текст послідовно вісім разів обробляється алгоритмом.

Алгоритм REDOC

REDOC II є ще одним блоковим алгоритмом, розробленим Майклом Вудом (Michael Wood) для корпорації Cryptech. У ньому використовуються 20-байтовий (160-бітовий) ключ і 80-бітовий блок.

Алгоритм REDOC II виконує всі маніпуляції - перестановки, підстановки і операції XOR з ключем - з байтами. Цей алгоритм зручний для програмної реалізації. У REDOC II використані змінні таблиці функцій. На відміну від алгоритму DES, що має фіксований (хоч і оптимізований з погляду стійкості) набір таблиць підстановок і перестановок, в REDOC II використовуються залежні від ключа і відкритого тексту набори таблиць (по суті, S-блоки). У REDOC II 10 раундів, кожен раунд складається з складної послідовності маніпуляцій з блоком.

Інша унікальна особливість REDOC II — використання масок. Це числа - похідні таблиці ключів, які використовуються для вибору таблиць даної функції для даного раунду. Для вибору таблиць функцій використовуються як значення даних, так і маски.

За умови, що найефективніший засіб злому цього алгоритму - лобовий розтин, REDOC II вельми надійний: для відновлення ключа потрібно 2160 операцій. Томас Кузік (Thomas Cusick) виконав криптоаналіз одного раунду REDOC II, але розширити розтин на декілька раундів йому не вдалося. Використовуючи диференціальний криптоаналіз, Біхам і Шамір успішно виконали криптоаналіз одного раунду REDOC II за допомогою 2300 підібраних відкритих текстів. Вони не зуміли розширити цю атаку на декілька раундів, але їм вдалося набути трьох значення маски після чотирьох раундів. Були і інші спроби криптоаналізу.

Алгоритм REDOC III

Алгоритм REDOC III є спрощеною версією REDOC II, теж розробленою Майклом Вудом. Він оперує з 80-бітовим блоком. Довжина ключа може змінюватися і досягати 2560 байт (204800 біт). Алгоритм складається тільки з операцій XOR над байтами ключа і відкритого тексту, перестановки і підстановки не використовуються.

1. Створюють таблицю ключів з 256 10-байтових ключів, використовуючи секретний ключ.
2. Створюють два 10-байтові блоки масок M1 і M2. M1 є результатом операції XOR перших 128 10-байтових ключів, а M2 - результат операції XOR других 128 10-байтових ключів.
3. Для шифрування 10-байтового блоку:

Виконують операцію XOR з першим байтом блоку даних і першим байтом M1. Вибирають ключ в таблиці ключів, розрахованій в раунді 1. Використовують обчислене значення XOR як індекс таблиці. Виконують операцію XOR з кожним, окрім першого, байтом блоку даних і відповідним байтом вибраного ключа.

Виконують операцію XOR з другим байтом блоку даних і другим байтом M1. Вибирають ключ в таблиці ключів, розрахованій в раунді 1. Використовують обчислене значення XOR як індекс таблиці. Виконайте операцію XOR з кожним, окрім другого, байтом блоку даних і відповідним байтом вибраного ключа.

Продовжують ці дії зі всім блоком даних (з 3-10 байтами), поки не буде використаний кожен байт для вибору ключа з таблиці після виконання операції XOR з ним і відповідним значенням M1. Потім виконують операцію XOR з кожним, окрім використаного для вибору ключа, байтом, і ключем.

Повторюють етапи a-c для M2.

Це нескладний і швидкісний алгоритм. На процесорі 80386 з тактовою частотою 33МГц він шифрує дані із швидкістю 2.75 Мбит/сек. За оцінкою Вуда, конвейеризированный процесор з трактом даних 64 біт і тактовою частотою 20 Мгц може шифрувати дані з швидкістю понад 1.28 Гбиг/сек.

Алгоритм REDOC III нестійок. Він уразимо до диференціального криптоаналізу. Для відновлення обох масок достатньо близько 223 підібраних відкритих текстів.

Алгоритм LOKI

Алгоритм LOKI розроблений в Австралії і вперше представлений в 1990 році як можлива заміна DES. У ньому використовуються 64-бітовий блок і 64-бітовий ключ.

Використовуючи диференціальний криптоаналіз, Біхам і Шамір зламували алгоритм LOKI з 11 і менш раундами швидше, ніж лобовим розтином [170]. Більш того, алгоритм характеризується 8-бітовою комплементарностью, що спрощує лобовий розтин в 256 разів.

Як показав Скрині Кнудсен (Lars Knudsen), алгоритм LOKI з 14 і менш раундами уразливий диференціальному криптоаналізу. Крім того, якщо в LOKI використовуються альтернативні S-блоки, то одержаний шифр, ймовірно, теж уразливий диференціальному криптоаналізу.

Алгоритм LOKI91

У відповідь на описані вище розтини розробники LOKI повернулися за креслярську дошку і переглянули свій алгоритм. В результаті з'явився алгоритм LOKI91. (Попередня версія LOKI була перейменована LOKI89).

Щоб підвищити стійкість алгоритму до диференціального криптоаналізу і позбавитися комплементарности, в початковий проект були внесені наступні зміни:

Алгоритм генерації подключей модифікований з тим, щоб половини переставлялися не після кожного, а після кожного другого раунду.

Алгоритм генерації подключей модифікований так, що число позицій циклічного зрушення лівого подключа складало то 12, то 13 бітів.

Виключені початкова і завершальна операції XOR з блоком і ключем.

Змінена функція S-блоку з метою згладити профілі XOR S-блоків (щоб підвищити їх стійкість до диференціального криптоаналізу), і виключити всі значення *x,* для яких *f(x)*= 0, де *f* - комбінація *E*-, *S*- і P-блоків.

Алгоритм LOKI не запатентований - реалізувати і використати LOKI може хто завгодно.

Алгоритми Khufu і Khafre

У 1990 році Ральф Меркл (Ralph Merkle) запропонував два алгоритми. У основу конструкції закладені наступні принципи:

56-бітовий розмір ключа DES дуже малий. Оскільки вартість збільшення розміру ключа нехтує мала (комп'ютерна пам'ять недорога і доступна), довжину ключа слід збільшити.

Широке використання в DES перестановок, хоч і зручно для апаратних реалізацій, надзвичайно утрудняє програмні реалізації. Найшвидкісніші реалізації DES виконують перестановки за допомогою таблиць підстановок. Таблиці підстановок можуть забезпечити ті ж характеристики «розсіювання», що і власне перестановки, і набагато підвищити гнучкість реалізації.

S-блоки DES, що містять всього 64 4-бітових елементів, дуже малі. Тепер, із збільшенням об'єму пам'яті, повинні зрости і S-блоки. Більш того, всі вісім S-блоків в DES використовуються одночасно. Хоча це і зручніше для апаратури, для програмної реалізації це представляється непотрібним обмеженням. Повинні бути реалізовані більший розмір S-блоків і послідовне (а не паралельне) їх використання.

Загальновизнано, що початкова і завершальна перестановки криптографічний безглузді, а тому повинні бути виключені.

Всі швидкісні реалізації DES наперед обчислюють ключі для кожного раунду. Звідси, немає причин не зробити ці обчислення складнішими.

На відміну від DES, критерії проектування S-блоків повинні бути загальнодоступні.

В даний час до цього переліку Меркл, можливо, додав би «стійкість до диференціального і лінійного криптоаналізу, адже у той час ці методи розтину не були відомі.

Алгоритми Khufu і Khafre запатентовані. Початковий код цих алгоритмів приведений в патенті.

Алгоритм MMB

Незадоволеність використанням в одному з криптоалгоритмов 64-бітового блоку шифрування привела до створення Джоаной Деймен алгоритму під назвою MMB (Modular Multiplication-based Block cipher - модулярный мультиплікативний блоковий шифр). У основі MMB лежить змішування операцій різних груп алгебри. MMB - ітеративний алгоритм, що головним чином складається з лінійних дій (XOR і використання ключа) і паралельного застосування чотирьох крупних оборотних нелінійних підстановок. Ці підстановки визначаються за допомогою множення по модулю 232-1 з постійними множниками. У результаті з'являється алгоритм, що використовує 128-бітовий ключ і 128-бітовий блок.

Алгоритм Blowfish

Blowfish - це алгоритм, розроблений Брюсом Шнайером спеціально для реалізації на великих мікропроцесорах. Алгоритм Blowfish не запатентований. При проектуванні алгоритму Blowfish Шнайер намагався задовольнити наступним критеріям:

Швидкість. Програма, що реалізовує алгоритм Blowfish на 32-бітових мікропроцесорах, шифрує дані із швидкістю 26 тактів на байт.

Компактність. Для виконання програмної реалізації алгоритму Blowfish достатньо 5 Кбайт пам'яті.

Простота. У алгоритмі Blowfish використовуються тільки прості операції: складання, XOR і підстановка з таблиці по 32-бітовому операнду. Аналіз його схеми нескладний, що знижує ризик помилок реалізації алгоритму.

Стійкість, що настроюється. Довжина ключа Blowfish змінна і може досягати 448 біт.

Алгоритм Blowfish оптимізований для застосування в системах, що не практикують частої зміни ключів, наприклад, в лініях зв'язку і програмах автоматичного шифрування файлів. При реалізації на 32-бітових мікропроцесорах з великим розміром кеша даних, наприклад, процесорах Pentium і PowerPC, алгоритм Blowfish помітно швидше за DES. Алгоритм Blowfish не годиться для застосування у випадках, де потрібна часта зміна ключів, наприклад, в комутаторах пакетів, або як однонаправлена хэш-функція. Великі вимоги до пам'яті не дозволяють використовувати цей алгоритм в смарт-картах.

Алгоритм RC5

RC5 є блоковим шифром з безліччю параметрів: розміром блоку, розміром ключа і числом раундів. Він винайдений Роном Рівестом і проаналізований в RSA Laboratories.

У алгоритмі RC5 передбачені три операції: XOR, складання і циклічні зрушення. На більшості процесорів операції циклічного зрушення виконуються за постійний час, змінні циклічні зрушення є нелінійною функцією. Ці циклічні зрушення, залежні як від ключа, так і від даних, - цікава операція.

Об'єднання блокових шифрів

Відома безліч шляхів об'єднання блокових алгоритмів для отримання нових алгоритмів. Створення подібних схем стимулюється бажанням підвищити безпеку, уникнувши трудності проектування нового алгоритму. Так, алгоритм DES відноситься до надійних алгоритмів, він піддавався криптоаналізу добрих 20 років і, проте, якнайкращим способом злому залишається лобовий розтин. Проте ключ DES дуже короткий. Хіба не погано було б використовувати DES як компоненту іншого алгоритму з довшим ключем? Це дозволило б скористатися перевагами обох систем: стійкістю, гарантованою двома десятиліттям криптоаналізу, і довгим ключем.

Один з методів об'єднання - багатократне шифрування. В цьому випадку для шифрування одного і того ж блоку відкритого тексту алгоритм шифрування використовується кілька разів з декількома ключами. Каскадне шифрування подібно до багатократного шифрування, але використовує різні алгоритми. Відомі і інші методи.

Повторне шифрування блоку відкритого тексту одним і тим же ключем за допомогою того ж або іншого алгоритму неефективно. Повторне використання того ж алгоритму не підвищує складність лобового розтину. (Ми припускаємо, що криптоаналітику відомі алгоритм і число операцій шифрування). При використанні різних алгоритмів складність лобового розтину може, як зростати,

так і залишатися незмінною. При цьому потрібно переконатися в тому, що ключі для послідовних шифрувань різні і незалежні.

Алгоритм DES

Стандарт шифрування даних DES (Data Encryption Standard) опублікований в 1977 р. Національним бюро стандартів США.

Стандарт DES призначений для захисту від несанкціонованого доступу до важливої, але несекретної інформації в державних і комерційних організаціях США. Алгоритм закладений в основу стандарту, розповсюджувався досить швидко, і вже в 1980 р. був схвалений Національним інститутом стандартів і технологій США (НІСТ). До теперішнього часу DES є найбільш поширеним алгоритмом, що використовується в системах захисту комерційної інформації.

Основні переваги алгоритму DES:

- використовується тільки один ключ довжиною 56 біт;
- зашифрувавши повідомлення за допомогою одного тексту програм, для дешифрування можна використати будь-який інший пакет програм, що відповідає стандарту DES;
- відносна простота алгоритму забезпечує високу швидкість обробки;
- достатньо висока стійкість алгоритму.

Алгоритм DES використовує комбінацію підстановок і перестановок. DES здійснює шифрування 64-бітових блоків даних за допомогою 64-бітового ключа, в якому значущими є 56 біт (інші 8 біт - перевірочні біти для контролю на парність). Дешифрування в DES є операцією, оберненою шифруванню, і виконується шляхом повторення операцій шифрування в оберненій послідовності. Узагальнена схема процесу шифрування в алгоритмі DES показана на рис.4. Процес шифрування полягає в початковій перестановці бітів 64-бітового блоку, шістнадцяти циклах шифрування і, нарешті, в кінцевій перестановці бітів.

Всі перестановки і коди в таблицях підібрані розробниками таким чином, щоб максимально ускладнити процес дешифрування шляхом підбору ключа. При описі алгоритму DES (рис.5) застосовані наступні позначення:

L і *R* - послідовності бітів (ліва (left) і права (right));

LR - конкатенація послідовностей *L* і *R*, тобто така послідовність бітів, довжина якої рівна сумі довжин *L* і *R*, в послідовності *LR* біти послідовності *R* слідують за бітами послідовності *L*;

⊕ - операція побітового додавання по модулю 2.

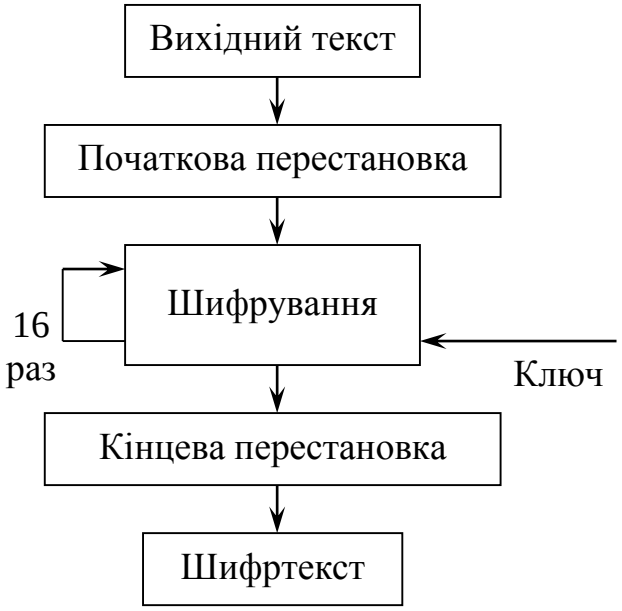


Рис. 4. Узагальнена схема шифрування в алгоритмі DES

Розглянемо роботу алгоритму DES. Нехай з файлу вихідного тексту прочитаний черговий 64-бітовий (8-байтовий) блок *T*. Цей блок *T* перетворюється за допомогою матриці початкової перестановки *IP* (додаток 1).

Біти вхідного блоку *T* (64 біта) переставляються відповідно до матриці *IP*. біт 58 вхідного блоку *T* стає бітом 1, біт 50 - бітом 2 і т.д. (див. додаток 1). Цю перестановку можна описати виразом $T_0 = IP(T)$. Отримана послідовність бітів T_0 розділяється на дві послідовності: L_0 - ліві або старші біти, R_0, - праві або молодші біти, кожна з яких містить 32 біти.

Потім виконується ітеративний процес шифрування, що складається з 16 кроків (циклів). Нехай T_i - результат *i* - ої ітерації

$$T_i = L_i R_i$$

де $L_i = t_1 t_2 ... t_{32}$ (перші 32 біти); $R_i = t_{33} t_{34} ... t_{64}$ (останні 32 біти). Тоді результат *i* - ої ітерації описується наступними формулами:

$$L_i = R_{i-1}, i = 1,2,...,16$$

$$R_i = L_{i-1} \oplus f(R_{i-1}, K_i), i = 1,2,...,16$$

Функція f називається функцією шифрування. її аргументами є послідовність R_{i-1}, що отримується на попередньому кроці ітерації, і 48-бітовий ключ K_i, який є результатом перетворення 64-бітового ключа шифру *K*. (Детальніше функція шифрування f і алгоритм отримання ключа K_i описані нижче.)

На останньому кроці ітерації отримують послідовності R_{16} і L_{16} (без перестановки місцями), які конкатенуються в, 64-бітову послідовність $R_{16}L_{16}$.

По закінченні шифрування здійснюється відновлення позицій бітів за допомогою матриці зворотної перестановки IP^{-1}(додаток 2).

Процес дешифрування даних є обернений по відношенню до процесу шифрування. Всі дії повинні бути виконані в оберненому порядку. Це означає, що дані, які дешифруються спочатку переставляються відповідно до матриці IP'1, а потім над послідовністю бітів R16L,6 виконуються ті ж дії, що і в процесі шифрування, але в оберненому порядку.

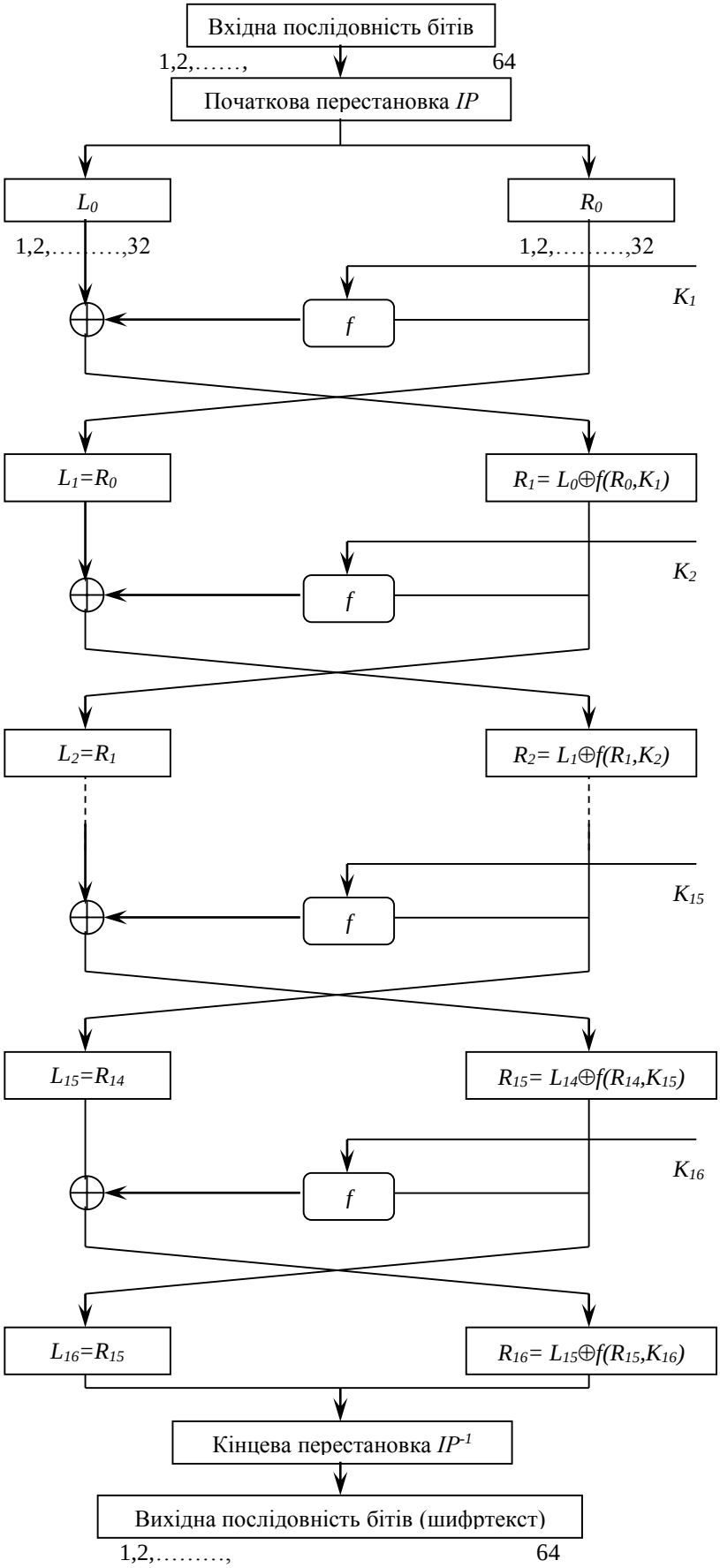


Рис. 5. Структура алгоритму DES

Ітеративний процес дешифрування може бути описаний наступними формулами:

$$R_{i-1} = L_i, i = 1,2,...,16$$

$$L_{i-1} = R_i \oplus f(L_i, K_i), i = 1,2,...,16$$

Таким чином, для процесу дешифрування з переставленим вхідним блоком $R_{16}L_{16}$ на першій ітерації використовується ключ K_{16}, на другій ітерації – K_{15} і т.д. На 16-й ітерації використовується ключ K. На останньому кроці ітерації будуть отримані послідовності L_0 *і* R_0, які конкатенуються в 64-бітову послідовність L_0R_0. Потім в цій послідовності 64 біти переставляються відповідно до матриці *IP*. Результат такого перетворення - початкова послідовність бітів (дешифроване 64-бітове значення).

Розглянемо реалізацію функції для перетворення *f*. Схема обчислення функції шифрування показана на рис.6.

Для обчислення значення функції f використовуються:

- функція £ (розширення 32 біт до 48);
- функція S,,S2.....S8 (перетворення 6-бітовогочисла в 4-бітове);
- функція Р (перестановка бітів в 32-бітовій послідовності).

Визначення цих функцій наведено в додатках 3-5. Одержаний результат (позначимо його £(/?,_,)) додається за модулем 2 (операція XOR) з поточним значенням ключа Кі і потім розбивається на вісім 6-бітових блоків В1,В2,...,S8: E(R,_jeKl=B02...Ba.

Далі кожний з цих блоків використовується як номер елемента в функціях-матрицях S1,S2,...,S8, що містять 4-бітові значення (додаток 4).

В результаті отримуємо S,(BJS2(B2)S3(BZ)...SB(BS), тобто 32-бітовий блок (оскільки матриці Sy- містять 4-бітові елементи). Цей 32-бітовий блок перетворюється за допомогою функції перестановки бітів Р (додаток 5).

Таким чином, функція шифрування "

*(«І.,Л)=Р<ЗД).....s6(ВВ)).

На кожній ітерації використовується нове значення ключа К, (довжиною 48 біт). Нове значення ключа К(обчислюється з початкового ключа К (рис. 7).

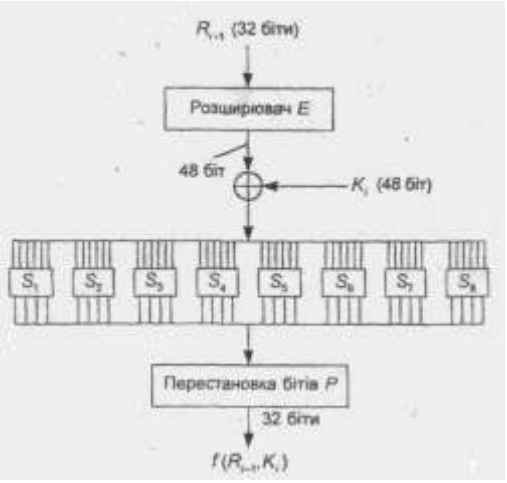


Рис.6. Схема обчислень функції шифрування f

Ключ К являє собою 64-бітовий блок з 8 бітами контролю по парності, розташованими в позиціях 8, 16, 24, 32, 40, 48, 56, 64. Для видалення контрольних бітів і підготовки ключа до роботи використовується функція G первинної підготовки ключа (додаток 6).

Таблиця додатку 6 розділена на дві частини. Результат перетворення G(K) розбивається на дві половини Со і Ц, по 28 біт кожна. Перші чотири рядки матриці G визначають, як вибираються біти послідовності Со (першим бітом Со буде біт 57 ключа шифру, потім біт 49 і т.д., а останніми бітами - біти 44 і 36 ключа).

Наступні чотири рядки матриці G визначають, як вибираються біти послідовності Do (тобто послідовність О0 буде складатися з бітів 63, 55, 47,..., 12,4 ключа шифру).

Як видно з додатку 6, для генерації послідовностей Со і Do не використовуються біти 8,16,24,32,40, 48, 56 і 64 ключа шифру. Ці біти не впливають на шифрування і можуть служити для інших цілей (наприклад, для контролю по парності). Таким чином, насправді ключ шифру є 56-бітовим.

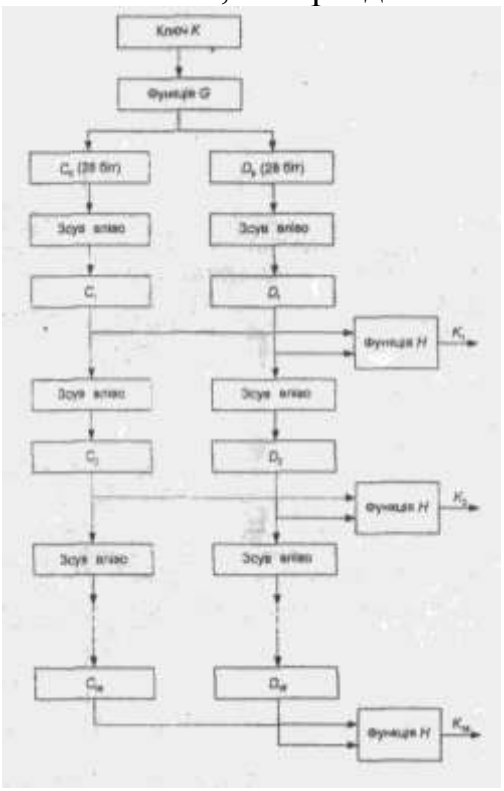


Рис.7. Схема алгоритму обчислення ключів Кі 22

Після визначення Со і Do рекурсивно визначаються С, і D,, I т 1,2,...,16 . Для цього застосовуються операції циклічного зсуву вліво на один або два біти в залежності від номера кроку ітерації, як показано в додатку 7.

Операції зсуву виконуються для послідовностей С, і D, незалежно. Наприклад, послідовність С3 виходить за допомогою циклічного зсуву вліво на дві позиції послідовності С2, а послідовність D3 • за допомогою зсуву вліво на дві позиції послідовності D2, С16 і О16 виходять з С15 і D15 за допомогою зсуву вліво на одну позицію.

Ключ К,, що визначається на кожному кроці ітерації, є результат вибору конкретних бітів з 56-бітової послідовності С, D, і їх перестановки. Іншими словами, ключ К, =H(£iD^, де функція Н визначається матрицею завершальної обробки ключа (додаток 8).

Як випливає з додатку 8, першим бітом ключа К; буде 14-й біт послідовності CjDj, другим - 17-й бгт, 47-м бітом ключа К, буде 29-й бгт С, D,, а 48-м бітом - 32-й біт С, D,.

2. Порядок виконання роботи

1. Ознайомитись з викладеним вище матеріалом.
2. Опишіть особливості шифру у відповідності до варанту, наведіть алгоритм роботи шифру.
3. Зашифруйте журнал в якому зберігаються ідентифікатори користувачів з лабораторної роботи №3. Передбачити що при перевірці існування користувача система автоматично виконувала дешифрування журналу з переліком ідентифікаторів.
4. Дати відповідь на контрольні запитання.
5. Скласти звіт з виконання лабораторної роботи та захистити його.

3. Зміст звіту

1. Назва та зміст лабораторної роботи.
2. Відповідь на контрольні запитання.
3. Структура алгоритму.
4. Аналіз швидкісних характеристик.
5. Висновки.

4. Індивідуальні завдання.

Розглянути загальні характеристики блочних шифрів. Детально описати обраний блочний шифр. Вибирати згідно номеру в журналі.

1. LUCIFER
2. MADRYGA
3. NewDES
4. FEAL
5. REDOC
6. LOKI
7. KHUFU і KHAFRE
8. RC2
9. IDEA
10. MMB
11. CA-1.1
12. SKIPJACK
13. ГОСТ
14. CAST
15. BLOWFISH
16. SAFER
17. 3-WAY
18. CRAB
19. SXAL8/MBAL
20. RC5

5. Контрольні запитання.

1. Які шифри називають симетричними?
2. Наведіть приклади симетричних шифрів.
3. Назвіть основні переваги та недоліки симетричних шифрів?
4. Що таке блочні шифри?
5. Дайте характеристику операціям підстановки та перестановки.
6. Чим забезпечується криптостійкість алгоритму?
7. Назвіть основні області застосування алгоритму.

Лабораторна робота №6.

Тема: ”Дослідження процедури шифрування та дешифрування в криптосистемі RSA ”

Мета роботи: Засвоїти методику та отримати практичні навики побудови засобів захисту інформації на основі криптосистеми RSA.

1. Теоретичні відомості

Алгоритм RSA запропонували в 1978 р. троє авторів: Р.Райвест (Rivest), А.Шамір (Shamir) і А.Адлеман (Adleman). Алгоритм одержав свою назву за першими буквами прізвищ його авторів. Алгоритм RSA став першим повноцінним алгоритмом :: відкритим ключем, що може працювати як у режимі шифрування даних, так і в режимі електронного цифрового підпису.

Надійність алгоритму ґрунтується на складності задач факторизації великих чисел та обчислення дискретних логарифмів.

У криптосистемі RSA відкритий ключ K_B, секретний ключ k_B, повідомлення M і криптограма C належать множині цілих чисел

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

P і Q - випадкові великі прості числа.

Для забезпечення максимальної безпеки вибирають P і Q рівної довжини і зберігають у секреті

Множина Z_N з операціями додавання і множення за модулем N утворює арифметику за модулем N.

Відкритий ключ K_B вибирають випадковим чином так, щоб виконувалися умови:

$$1 < K_B \le \varphi(N), НСД(K_B, \varphi(N)) = 1, \varphi(N) = (P-1)(Q-1)$$

$\varphi(N)$ - функція Ейлера,

НСД - найбільший спільний дільник.

Функція Ейлера $\varphi(N)$ вказує кількість додатних цілих чисел в інтервалі від 1 до N, що взаємно прості з N. Друга із зазначених вище умов означає, що відкритий ключ K_B і функція Ейлера $\varphi(N)$ повинні бути взаємно простими.

Дані, використовуючи розширений алгоритм Евкліда (див. додаток 1), обчислюють секретний ключ k_B, такий, що

$$k_B * K_B \equiv 1 (\mathrm{mod}\, \varphi(N))$$

або

$$k_B = K_B^{-1} (\mathrm{mod}(P-1)(Q-1)).$$

Це нескладно здійснити, оскільки відома пара простих чисел (P, Q) і можна легко знайти $\varphi(N)$. Слід відмітити, що k_B і N повинні бути взаємно простими.

Відкритий ключ K_B використовують для шифрування даних, а секретний ключ k_B - для дешифрування.

Процедура шифрування визначає криптограму C через пару (відкритий ключ K_B, повідомлення M) у відповідності з наступною формулою:

$$C = E_{K_B}(M) = E_B(M) = M^{K_B} (\mathrm{mod}\, N)$$

Як алгоритм швидкого обчислення значення C використовують ряд послідовних піднесень до квадрату цілого M множень на M з приведенням за модулем N.

Зворотна операція, тобто визначення значення M за відомим значенням C, K_B і N, практично не здійсненна при $N \approx 2^{512}$.

Однак обернену задачу, тобто задачу дешифрування криптограми C, можна вирішити, використовуючи пари (секретний ключ k_B , криптограма C) за наступною формулою:

$$M = D_{k_B}(C) = D_B(C) = C^{k_B} (\mathrm{mod}\, N)$$

Таким чином, якщо криптограму

$$C = M^{K_B} (\mathrm{mod}\, N)$$

піднести до степеня k_B, то в результаті відновлюється вихідний відкритий текст M. тому що

$$(M^{K_B})^{k_B} = M^{K_B k_B} = M^{n\cdot\varphi(N)+1} \equiv M (\mathrm{mod}\, N).$$

Таким чином, одержувач В, що створює криптосистему, захищає два параметри:

1. секретний ключ k_B
2. пару чисел (P, Q), добуток яких дає значення модуля N.

З іншого боку, одержувач В відкриває значення модуля N і відкритий ключ K_B .

Супротивнику відомі лише значення K_B і N. Якби він зміг розкласти число N на множники P і Q (задача факторизації), то він довідався б "таємний хід" - трійку чисел $\{P, Q, K_B\}$, обчислив значення функції Ейлера

$\varphi(N)=(P-1)(Q-1)$ і визначив значення секретного ключа k_B.

Однак, як уже відзначалося, розкладання дуже великого N на множники не здійснено за реальний час (за умови, що довжини обраних P і Q складають не менш 100 десяткових знаків).

Процедура шифрування і дешифрування в криптосистемі RSA

Припустимо, що користувач А хоче передати користувачу В повідомлення в зашифрованому виді, використовуючи криптосистему RSA.

У такому випадку користувач А виступає в ролі відправника повідомлення, а користувач В - у ролі одержувача. Як відзначалося вище, криптосистему RSA повинен сформувати одержувач повідомлення, тобто користувач В. Розглянемо послідовність дій користувача В і користувача А.

1. Користувач В вибирає два довільних великих простих числа P і Q.
2. Користувач В обчислює значення модуля $N = P * Q$.
3. Користувач В обчислює функцію Ейлера

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

і вибирає випадковим чином значення відкритого ключа K_B з урахуванням виконання умов:

$$1 < K_B \le \varphi(N), НСД(K_B, \varphi(N)) = 1$$

4. Користувач В обчислює значення секретного ключа k_B, використовуючи розширений алгоритм Евкліда

$$k_B \equiv K_B^{-1} (\mathrm{mod}\, \varphi(N))$$

5. Користувач В пересилає користувачу А пари чисел (N, K_B) по незахищеному каналі.

Якщо користувач А хоче передати користувачу В повідомлення M, він виконує наступні кроки.

6. Користувач А розбиває вихідний відкритий текст M на блоки, кожний з яких може бути представлений у вигляді числа

$$M_i = 1,2,...N-1.$$

7. Користувач А шифрує текст, представлений у вигляді послідовності чисел M, за формулою

$$C_i = M_i^{K_B} (\mathrm{mod}\, N)$$

і відправляє криптограму

$$C_1, C_2, C_3, ..., C_i, ...,$$

користувачу В.

8. Користувач В розшифровує прийняту криптограму

$$C_1, C_2, C_3, ..., C_i, ...,$$

використовуючи секретний ключ k_B, за формулою

$$M_i = C_i^{k_B} (\mathrm{mod}\, N)$$

У результаті буде отримана послідовність чисел M_i, що являють собою вихідне повідомлення M. Щоб алгоритм RSA мав практичну цінність, необхідно мати можливість без істотних витрат генерувати великі прості числа, вміти оперативно обчислювати значення ключів K_B і k_B.

Приклад.

Шифрування повідомлення С А В.

Для простоти обчислень будуть використовуватися невеликі числа. На практиці застосовуються дуже великі числа.

Дії користувача В.

1. Вибирає P=3 і Q=11.
2. Обчислює модуль $N = P*Q = 3*11 = 33$.
3. Обчислює значення функції Ейлера для N=33:

$$\varphi(N) = \varphi(33) = (P-1)(Q-1) = 2*10 = 20.$$

Вибирає як відкритий ключ K_B, довільне число з урахуванням виконання умов:

$$1 < K_B \le 20, НСД(K_B, 20) = 1$$

Нехай K_B=7.

4. Обчислює значення секретного ключа k_B, використовуючи розширений алгоритм Евкліда для розв'язку конгруенції

$$k_B \equiv 7^{-1} (\mathrm{mod}\, 20)$$

Рішення дає k_B = 3.

5. Пересилає користувачу А пари чисел (N=33, K_B=7).

Дії користувача А.

6. Представляє шифроване повідомлення як послідовність цілих чисел у діапазоні 0...32.

Нехай буква А представляється як число 1, буква В – як число 2, буква С - як число 3.

Тоді повідомлення С А В можна представити як послідовність чисел 3 1 2, тобто $M_1 = 3$, $M_2 = 1$, $M_3 = 2$.

7. Шифрує, текст, представлений у виді послідовності чисел M_1, M_2 і M_3, використовуючи ключ $K_B = 7$, за формулою

$$C_i = M_i^{K_B} (\mathrm{mod}\, N) = M_i^7 (\mathrm{mod}\, 33)$$

Одержує

$$C_1 = 3^7 (\mathrm{mod}\, 33) = 2187 (\mathrm{mod}\, 33) = 9$$

$$C_2 = 1^7 (\mathrm{mod}\, 33) = 1 (\mathrm{mod}\, 33) = 1$$

$$C_3 = 2^7 (\mathrm{mod}\, 33) = 128 (\mathrm{mod}\, 33) = 29$$

Відправляє користувачу В криптограму

$$C_1, C_2, C_3 = 9, 1, 29.$$

Дії користувача В.

8. Розшифровує прийняту криптограму C_1, C_2, C_3, використовуючи секретний ключ $k_B = 3$, за формулою

$$M_i = C_i^{k_B} (\mathrm{mod}\, N) = C_i^3 (\mathrm{mod}\, 33)$$

Одержує

$$M_1 = 9^3 (\mathrm{mod}\, 33) = 729 (\mathrm{mod}\, 33) = 3$$

$$M_2 = 1^3 (\mathrm{mod}\, 33) = 1 (\mathrm{mod}\, 33) = 1$$

$$M_3 = 29^3 (\mathrm{mod}\, 33) = 24389 (\mathrm{mod}\, 33) = 2$$

Таким чином, відновлене вихідне повідомлення:

С	А	В
3	1	2

Безпека і швидкодія криптосистеми RSA

Як було зазначено раніше безпека алгоритму RSA базується на складності задачі факторизації великих чисел, що є добутками двох великих простих чисел. Справді, криптостійкість алгоритму RSA визначається тим, що після формування секретного ключа k_B і відкритого ключа K_B "стираються" значення простих чисел P і Q, і тоді надзвичайно важко визначити секретний ключ k_B за відкритим ключем K_B .оскільки для цього необхідно знайти дільники P і Q модуля N.

Розкладання величини N на прості множники P і Q дозволяє обчислити функцію $\varphi(N)=(P-1)(Q-1)$, а відтак значення секретного ключа k_B, використовуючи рівняння.

$$K_{B^*}\, k_B \equiv 1 (\mathrm{mod}\ \varphi(N))$$

Іншим можливим способом криптоаналізу алгоритму RSA є безпосереднє обчислення чи підбір значення функції $\varphi(N)=(P-1)(Q-1)$. Якщо встановлене значення $\varphi(N)$, то співмножники P й Q обчислюються досить просто. Справді, нехай

$$x = P + Q = N + 1 - \varphi(N),$$

$$y = (P - Q)^2 = (P + Q)^2 - 4 * N$$

Знаючи $\varphi(N)$, можна визначити x і потім y знаючи x і y, можна визначити числа P і Q з наступних співвідношень:

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

Однак ця атака не простіша задачі факторизації модуля N.

Задача факторизації є важко розв'язною задачею для великих значень модуля N.

В табл.2 наведено оцінки довжин ключів для асиметричних криптосистем. Ці оцінки дані для трьох груп користувачів (індивідуальних користувачів, корпорацій і державних організацій), відповідно до вимог щодо їхньої інформаційної безпеки. Звичайно, дані оцінки варто розглядати як приблизні, внаслідок можливої тенденції змін безпечних довжин ключів асиметричних криптосистем згодом.

Криптосистеми RSA реалізуються як апаратним, так і програмним шляхом.

Для апаратної реалізації операцій шифрування і дешифрування RSA розроблені спеціальні процесори. Ці процесори, реалізовані на надвеликих інтегральних схемах (НВІС), дозволяють виконувати операції RSA, пов'язані зі зведенням великих чисел у колосально великий степінь за модулем N, за відносно короткий час. І все-таки апаратна реалізація RSA приблизно в 1000 разів повільніше апаратної реалізації симетричного криптоалгоритма DES.

Таблиця 2 Оцінки довжин ключів для асиметричних криптосистем, біт

Рік	Окремі користувачі	Корпорації	Державні організації
1995	768	1280	1536
2000	1024	1280	1536
2005	1280	1536	2048
2010	1280	1536	2048
2015	1536	2048	2048

Одна з найшвидших апаратних реалізацій RSA з модулем 512 біт на надвеликій інтегральній схемі має швидкодію 64 Кбіт/с. Кращими з серійного випуску НВІС є процесори фірми CYLINK, що виконують 1024-бітове шифрування RSA.

Програмна реалізація RSA приблизно в 100 разів повільніша за програмну реалізацію DES. З розвитком технології ці оцінки можуть трохи змінюватися, але асиметрична криптосистема RSA ніколи не досягне швидкодії симетричних криптосистем.

Слід зазначити, що мала швидкодія криптосистем RSA обмежує область їхнього застосування, але не перекреслює їхню цінність.

2. Порядок виконання роботи

1. Ознайомитись з викладеним вище матеріалом.
2. Отримати індивідуальне завдання у викладача.

3. Розробити структуру алгоритму RSA.
4. Реалізувати алгоритм RSA.
5. Провести не менше 2 експериментів шифрування за алгоритмом RSA згідно варіанту.
6. Дослідити залежність швидкості роботи програми від об'єму вхідного тексту та оцінити стійкість до криптоаналізу для кожного експерименту.
7. Порівняти отримані результати із алгоритмом шифрування DES.
8. Забезпечити автоматизоване шифрування запитань і відповідей, які передаються по відкритому каналу зв’язку в лабораторній роботі №4.
9. Дати відповідь на контрольні запитання.
10. Скласти звіт з використання лабораторної роботи та захистити його до початку виконання наступної лабораторної роботи.

3. Зміст звіту

1. Назва та зміст лабораторної роботи.
2. Відповіді, на контрольні запитання.
3. Структура алгоритму RSA.
4. Текст програми основних модулів.
5. Аналіз швидкісних характеристик реалізованого алгоритму, відповідно до варіанту і представлених у вигляді графіків та таблиць.
6. Висновки.

4. Індивідуальні завдання.

Розробити програму шифрування та дешифрування даних на основі алгоритму RSA. Дослідити залежність часу шифрування, дешифрування від розміру ключів та об'єму тексту. Отримані результати представити у вигляді таблиці та графіків. Порівняти отримані результати з оцінками швидкості криптування за алгоритмом DES.

Варіант	Експеримент 1			Експеримент 2		
	Об’єм, Mb	P	Q	Об’єм, Mb	P	Q
1.	~0,1	$\sim 10^3$	$\sim 10^4$	~10	$\sim 10^2$	$\sim 10^3$
2.	~0,2	$\sim 10^2$	$\sim 10^3$	~9	$\sim 10^4$	$\sim 10^3$
3.	~0,3	$\sim 10^4$	$\sim 10^5$	~8	$\sim 10^3$	$\sim 10^4$
4.	~0,4	$\sim 10^6$	$\sim 10^4$	~7	$\sim 10^5$	$\sim 10^4$
5.	~0,5	$\sim 10^5$	$\sim 10^3$	~6	$\sim 10^2$	$\sim 10^3$
6.	~0,6	$\sim 10^5$	$\sim 10^5$	~5	$\sim 10^4$	$\sim 10^3$
7.	~0,7	$\sim 10^6$	$\sim 10^4$	~4	$\sim 10^3$	$\sim 10^4$
8.	~0,8	$\sim 10^4$	$\sim 10^3$	~3	$\sim 10^5$	$\sim 10^4$
9.	~0,9	$\sim 10^2$	$\sim 10^5$	~2	$\sim 10^3$	$\sim 10^3$
10.	~1	$\sim 10^3$	$\sim 10^4$	~1	$\sim 10^4$	$\sim 10^3$

5. Контрольні запитання.

1. Які шифри називаються асиметричними?
2. Наведіть приклади асиметричних шифрів.

3. Наведіть основні переваги та недоліки асиметричних шифрів?
4. Чим забезпечується криптостійкість алгоритму RSA?
5. Назвіть основні області застосування алгоритму RSA.
6. Як визначити НСД двох чисел?
7. Що таке функція Ейлера?

Лабораторна робота №7

Тема: "Розробка політики інформаційної безпеки за стандартом ISO/IEC 17799"

Мета: Ознайомитись із основними положеннями стандарту ISO/IEC 17799 «Управління інформаційною безпекою. Практичні правила». Навчитись розробляти політику безпеки організації.

Теоретичні відомості

1. Стандарт ISO17799

Стандарт ISO17799 визначає загальну організацію, класифікацію даних, системи доступу, напрями планування, відповідальність співробітників, використання оцінки ризиків і т.д. в контексті інформаційної безпеки. В процесі впровадження стандарту створюється так звана система менеджменту інформаційної безпеки, мета якої – скорочення матеріальних втрат, пов'язаних з порушенням інформаційної безпеки. Стандарт покликаний заощадити підприємству засоби, а в деяких випадках навіть врятувати від банкрутства, і не є якоюсь зовнішньою обов'язковою вимогою, що приводить до появи додаткової статті витрат.

ISO17799 - це модель системи менеджменту, і в цьому сенсі не є технічним стандартом. Цей підхід до інформаційної безпеки на основі цілей менеджменту, а не фіксованих технічних специфікацій є принциповим для ISO17799 як стандарту системи управління.

У стандарті ISO17799 приводяться рекомендації по управлінню інформаційною безпекою. Він складає загальну основу для різних організацій при розробці, реалізації і оцінці ефективності процедур управління безпекою, а також дає можливість встановити довірчі відносини між організаціями.

Даний документ можна використовувати як загальноприйнятий стандарт при встановленні ділових відносин між організаціями і при висновку контрактів з субпідрядниками або придбанні інформаційних систем або продуктів.

2. Зміст інформаційної безпеки

Мета інформаційної безпеки — забезпечити безперебійну роботу організації і звести до мінімуму збиток від подій, що таять загрозу безпеці, за допомогою їх запобігання і зведення наслідків до мінімуму.

Управління інформаційною безпекою дає можливість колективно використовувати інформацію, забезпечуючи при цьому її захист і захист обчислювальних ресурсів.

Інформаційна безпека складається з трьох основних компонентів:

а) конфіденційність: захист конфіденційної інформації від несанкціонованого розкриття або перехоплення;

б) цілісність: забезпечення точності і повноти інформації і комп'ютерних програм;

в) доступність: забезпечення доступності інформації і життєво важливих сервісів для користувачів, коли це потрібно.

Інформація існує в різних формах. Її можна зберігати на комп'ютерах, передавати по обчислювальних мережах, роздруковувати або записувати на папері, а також озвучувати в розмовах. З погляду безпеки всі види інформації, включаючи паперову документацію, бази даних, плівки, мікрофільми, моделі, магнітні стрічки, дискети, розмови і інші способи, використовувані для передачі знань і ідей, вимагають належного захисту.

2.1 Необхідність захисту

Інформація і що підтримують її інформаційні системи і мережі є цінними виробничими ресурсами організації. Їх доступність, цілісність і конфіденційність можуть мати особливе значення для забезпечення конкурентоспроможності, руху грошової готівки, рентабельності, відповідності правовим нормам і іміджу організації. Сучасні організації можуть зіткнутися із зростаючою загрозою порушення режиму безпеки, що йде від цілого ряду джерел. Інформаційним системам і мережам можуть загрожувати такі небезпеки, як комп'ютерне шахрайство, шпигунство, саботаж, вандалізм, а також інші джерела відмов і аварій. З'являються все нові загрози, здатні завдати збитку організації, такі, як, широко відомі комп'ютерні віруси або хакери. Передбачається, що такі загрози інформаційній безпеці з часом стануть поширенішими, небезпечнішими і витонченішими. В той же час із-за зростаючої залежності організацій від інформаційних систем і сервісів, вони можуть стати уразливішими по відношенню до загроз порушення захисту. Розповсюдження обчислювальних мереж надає нові можливості для несанкціонованого доступу до комп'ютерних систем, а тенденція до переходу на розподілені обчислювальні системи зменшує можливості централізованого контролю інформаційних систем фахівцями.

Захисні заходи виявляються значно дешевшими і ефективнішими, якщо вони вбудовані в інформаційні системи і сервіси на стадіях завдання вимог і проектування. Чим швидше організація прикмет міри по захисту своїх інформаційних систем, тим більше дешевими і ефективними вони будуть для неї згодом.

2.2. Структура стандарту ISO/IEC 17799

Пропоновані практичні правила розбиті на наступних 10 розділів:

Розділ 1. Політика безпеки
Розділ 2. Організація захисту
Розділ 3. Класифікація ресурсів і їх контроль
Розділ 4. Безпека персоналу
Розділ 5. Фізична безпека і безпека навколишнього середовища
Розділ 6. Адміністрування комп'ютерних систем і обчислювальних мереж
Розділ 7. Управління доступом до систем
Розділ 8. Розробка і супровід інформаційних систем
Розділ 9. Планування безперебійної роботи організації
Розділ 10. Виконання вимог

У цих розділах представлений вичерпний набір засобів управління безпекою, заснованих на реальних заходах по захисту інформації, таких, що реалізовуються зараз в британських і міжнародних організаціях.

2.3 Застосовність засобів управління безпекою

Не всі засоби контролю застосовні до кожного інформаційного середовища; їх треба використовувати вибірково з урахуванням місцевих умов. Це ставати ясно з опису. Проте більшість засобів контролю, описаних в даному документі, широко застосовуються крупними організаціями з великим досвідом роботи, і їх використання рекомендується для всіх ситуацій, зрозуміло, з урахуванням обмежень, що накладаються технологією і навколишнім середовищем. Ці загальноприйняті засоби контролю називають базовими засобами управління безпекою, оскільки всі вони в сукупності визначають базовий промисловий стандарт на підтримку режиму безпеки.

Десять ключових засобів контролю, пропоновані цим стандартом, є особливо важливими. Ці ключові засоби є хорошою відправною точкою для управління інформаційною безпекою.

При використанні деяких із засобів контролю, наприклад, шифрування даних, можуть знадобитись поради фахівців з безпеки і оцінки ризиків, щоб визначити, чи потрібні вони і яким чином їх реалізувати. Для забезпечення вищого рівня захисту особливо цінних ресурсів або надання протидії виключно високим рівням загроз порушення режиму безпеки, в ряду випадках можуть потрібно сильніші засоби контролю, які виходять за рамки цих правил.

2.4 Ключові засоби контролю

Десять ключових засобів контролю є або обов'язкові вимоги, наприклад, вимоги чинного законодавства, або вважаються основними структурними елементами інформаційної безпеки, наприклад, навчання правилам безпеки. Ці засоби контролю застосовні до всіх організаціях і середовищам і відмічені символом ключа. Вони служать як основа для організацій, що приступають до реалізації засобів управління інформаційною безпекою.

Ключовими є наступні засоби контролю:

- документ про політику інформаційної безпеки (див. Документ про політику інформаційної безпеки);
- розподіл обов'язків по забезпеченню інформаційної безпеки (див. Розподіл обов'язків по забезпеченню інформаційної безпеки);
- навчання і підготовка персоналу до підтримки режиму інформаційної безпеки (див. Навчання правилам інформаційної безпеки);
- повідомлення про випадки порушення захисту (див. Повідомлення про інциденти в системі безпеки);
- засоби захисту від вірусів (див. Засоби захисту від вірусів);
- процес планування безперебійної роботи організації (див. Процес планування безперебійної роботи організації);

- контроль за копіюванням програмного забезпечення, захищеного законом про авторське право (див. Контроль за копіюванням ПЗ, захищеного законом про авторське право);
- захист документації організації (див. Захист документації організації);
- захист даних (див. Захист даних);
- відповідність політиці безпеки (див. Відповідність політиці безпеки).

2.5 Задання вимог до інформаційної безпеки організації

Існують три основні групи вимог до системи безпеки в будь-якій організації:

1) — це унікальний набір ризиків порушення безпеки, що складається із загроз, яким піддаються інформаційні ресурси, та їх вразливостей і можлива дія цих ризиків на роботу організації. Більшість з цих ризиків описані в справжніх правилах і їм можна успішно протистояти, якщо скористатися наведеними тут рекомендаціями. Проте існують ризики, що вимагають спеціального звернення, і їх необхідно розглядати з урахуванням їх оцінки в кожній конкретній організації або для кожного конкретного компоненту системи.

2) — це набір правових і договірних вимог, яким повинні задовольняти організація, її торгові партнери, підрядчики і постачальники послуг; при цьому зростає необхідність стандартизації у міру розповсюдження електронного обміну інформацією по мережах між організаціями. Дані практичні правила можуть служити надійною основою для завдання загальних вимог цього типу.

3) — це унікальний набір принципів, цілей і вимог до обробки інформації, який розроблений організацією для виробничих цілей. Важливо (наприклад, для забезпечення конкурентоспроможності), щоб в політиці безпеки були відображені ці вимоги, і життєво важливо, щоб реалізація або відсутність засобів управління безпекою в інформаційній інфраструктурі не заважали виробничої діяльності організації.

Залучення належних засобів контролю і необхідна гнучкість з самого початку процесу планування інформаційних систем є необхідними умовами для успішного завершення роботи.

2.6 Оцінка ризиків порушення безпеки

Витрати на систему захисту інформації необхідно зіставити і привести у відповідність з цінністю інформації, що захищається, і інших інформаційних ресурсів, піддаються ризику, а також із збитком, який може бути нанесений організації із-за збоїв в системі захисту.

Зазвичай методики аналізу ризиків (див. Аналіз ризиків) застосовуються до повних інформаційних систем і сервісів, але цими ж методиками можна скористатися і для окремих компонентів системи або сервісів, якщо це доцільно і практично. Для оцінки ризиків необхідно систематично розглядати наступні аспекти:

а) збиток, який може нанести діяльності організації серйозне порушення інформаційної безпеки, з урахуванням можливих наслідків порушення конфіденційності, цілісності і доступності інформації;

б) реальна вірогідність такого порушення захисту в світлі превалюючих загроз і засобів контролю.

Результати цієї оцінки необхідні для розробки основної лінії і визначення належних дій і пріоритетів для управління ризиками порушення інформаційної безпеки, а також для реалізації засобів контролю, що рекомендуються в справжніх практичних правилах. Оцінка цих двох аспектів ризики залежить від наступних чинників:

- характеру виробничої інформації і систем;
- виробничій меті, для якої інформація використовується;
- середовища, в якому система використовується і управляється;
- захисту, що забезпечується існуючими засобами контролю.

Оцінка ризиків може виявити виключно високий ризик порушення інформаційної безпеки організації, що вимагає реалізації додаткових, сильніших засобів контролю, ніж ті, які рекомендуються в справжніх правилах. Використання таких засобів контролю необхідно обґрунтувати виходячи з виводів, отриманих в результаті оцінки ризиків.

2.7 Умови успішної реалізації системи інформаційної безпеки

Перераховані нижче чинники є визначальний для успішної реалізації системи інформаційної безпеки в організації:

а) цілі безпеки і її забезпечення повинні ґрунтуватися на виробничих цілях і вимогах; функції управління безпекою повинно узяти на себе керівництво організації;

б) явна підтримка і прихильність до підтримки режиму безпеки вищого керівництва;

в) хороше розуміння ризиків порушення безпеки (як загроз, так і вразливостей), яким піддаються ресурси організації, і рівня їх захищеності в організації, який повинен ґрунтуватися на цінності і важливості цих ресурсів;

г) ознайомлення з системою безпеки всіх керівників і рядових співробітників організації;

д) надання вичерпного посібника з політики і стандартів інформаційної безпеки всім співробітникам і підрядчикам.

2.8 Розробка власних рекомендацій

Не існує єдиної оптимальної структури захисту інформації. Кожна категорія користувачів (див. Спеціальний привілей) або фахівців з інформаційних технологій, що працюють в конкретному середовищі, може мати свій власний, такий, що відрізняється від інших, набір вимог, проблем і пріоритетів, залежно від функцій конкретної організації і виробничого або обчислювального середовища.

Багато організацій вирішують цю проблему, розробляючи набір окремих керівних принципів для відповідних груп співробітників, щоб забезпечити ефективніше розповсюдження знань в області захисту інформації. Організаціям, що вирішили прийняти іншу структуру (або навіть розробити свої рекомендації), бажано ввести перехресні посилання на текст діючих правил, щоб їх майбутні ділові партнери або аудитори могли встановити прямі зв'язки між цим стандартом і прийнятими в даній організації принципами системи захисту інформації.

2. Політика інформаційної безпеки

Метою політики інформаційної безпеки сформулювати мету і забезпечити підтримку інформаційної безпеки керівництвом організації.

Вище керівництво повинне поставити чітку мету і всесторонньо подавати свою підтримку інформаційної безпеки за допомогою розповсюдження політики безпеки серед співробітників організації.

4. Документ про політику інформаційної безпеки

Письмовий документ про політику безпеки повинен бути доступний всім співробітникам, що відповідають за забезпечення режиму інформаційної безпеки.

Вище керівництво повинне надати задокументовану політику інформаційної безпеки всім підрозділам організації. Цей документ повинен містити принаймні наступне:

1) визначення інформаційної безпеки, її основні цілі і область її застосування, а також її значення як механізму, що дає можливість колективно використовувати інформацію (див. Введення);

2) виклад позиції керівництва по питаннях реалізації цілей і принципів інформаційної безпеки;

3) роз'яснення конкретних варіантів політики безпеки, принципів, стандартів і вимог до її дотримання, включаючи:

- виконання правових і договірних вимог;
- вимоги до навчання персоналу правилам безпеки;
- політика попередження і виявлення вірусів;
- політика забезпечення безперебійної роботи організації.

4) визначення загальних і конкретних обов'язків по забезпеченню режиму інформаційної безпеки;

5) роз'яснення процесу повідомлення про події, що таять загрозу безпеці.

Необхідно розробити процес перевірки, визначити обов'язку і задати дати перевірок для дотримання вимог документа про політику безпеки.

3. Приклад політики безпеки

Нижче представлений витяг з еталонної політики безпеки Підприємства, яка включає наступні розділи:

1. Загальні положення;
2. Політика управління паролями;
3. Ідентифікація користувачів;
4. Повноваження користувачів;
5. Захист інформаційних ресурсів ІС від комп'ютерних вірусів;
6. Правила установки і контролю мережних з'єднань;

7. Правила політики безпеки по роботі з системою електронної пошти;

8. Правила забезпечення безпеки інформаційних ресурсів.

У прикладі представлено перші два розділи Політики безпеки.

Загальні положення

Забезпечення інформаційної безпеки є необхідною умовою для здійснення діяльності Підприємства. Порушення інформаційної безпеки може привести до серйозних наслідків, включаючи втрату довіри з боку клієнтів і зниження конкурентоспроможності.

Основою заходів по забезпеченню режиму інформаційної безпеки адміністративного рівня, тобто заходів, що робляться керівництвом організації, є політика безпеки. Під політикою безпеки розуміється сукупність документованих управлінських рішень, направлених на захист інформації і асоційованих з нею ресурсів. Політика безпеки Підприємства визначає основні напрями і вимоги по забезпеченню інформаційної безпеки Підприємства.

Забезпечення безпеки інформації включає будь-яку діяльність, направлену на захист інформації і/або підтримуючої інфраструктури. Справжня політика інформаційної безпеки охоплює всі автоматизовані і телекомунікаційні системи, власником і користувачем яких є Підприємство. Положення цього документа відносяться до всього штатного персоналу, тимчасових службовців і інших співробітників Підприємства, а також клієнтів Підприємства і третіх осіб, що мають доступ до автоматизованих і телекомунікаційних систем Підприємства.

Політика управління паролями

Користувачі повинні вибирати нестандартні паролі. Це означає, що паролі не повинні бути пов'язані із заняттями або особистим життям користувачів. Наприклад, не можна використовувати як пароль номер власного автомобіля, ім'я дружини або частину адреси. Це також означає, що пароль не повинен бути просто словом із словника. Так, не повинні використовуватися як паролі імена власні, географічні назви, технічні терміни і сленг. Якщо є відповідні системні програмні засоби для здійснення контролю надійності що призначаються користувачам паролів, то необхідно використовувати ці засоби для того, щоб заборонити користувачам вибір легко вгадуваних паролів.

Користувачі можуть вибрати паролі, що легко запам'ятовуються, які в теж час є важко вгадуваними для третіх осіб, якщо буде виконано хоч би одна з наступних умов:

- Декілька слів написано злито (такі паролі відомі під назвою «passphrases»);
- При наборі слова на клавіатурі використані клавіші, зміщені щодо потрібних, на один ряд вгору, вниз, управо або вліво;
- Слово набране із зсувом на певну кількість букв вгору або вниз за абеткою;
- Комбінація цифр і звичайного слова;
- Навмисно неправильне написання слова (але не звичайна в даному слові орфографічна помилка).

Рекомендується, щоб в паролі були не тільки букви, але і інші символи, тобто цифри (0-9) і знаки пунктуації. Використання символів, що управляють, і інших знаків, що не відображаються, не рекомендується, оскільки через це

можуть виникнути проблеми при передачі даних по мережі, несподівано активізуватися певні системні утиліти або виникнути інші побічні ефекти.

Всі паролі повинні полягати не менше чим з шести символів. Довжина паролів винна завжди автоматично перевірятися в той момент, коли користувачі створюють або вибирають паролі.

Користувачі не повинні створювати паролі, які ідентичні або в значній мірі повторюють раніше використовувані ними паролі. Якщо є відповідні системні програмні засоби, необхідно заборонити користувачам, повторно використовувати свої попередні паролі.

Паролі не повинні зберігатися в доступній для читання формі в командних файлах, сценаріях автоматичної реєстрації, програмних макросах, функціональних клавішах терміналу, на комп'ютерах з неконтрольованим доступом, а також в інших місцях, де не уповноважені особи можуть дістати до них доступ. Наприклад, ні в яких застосуваннях користувачі не повинні вибирати таку опцію конфігурації, як автоматичне збереження пароля.

Не можна записувати паролі і залишати ці записи в місцях, де до них можуть дістати доступ не уповноважені особи. Пароль повинен бути негайно змінений, якщо є підстави вважати, що цей пароль став відомий кому-небудь ще, окрім самого користувача.

Практичне завдання

1. Розробити політику безпеки організації у відповідності з рекомендаціями стандарту ISO/IEC 17799.
2. Обґрунтувати запропоновану політику безпеки.
3. Оформити звіт за результатами виконання роботи.

Контрольні запитання

1. Охарактеризувати спрямованість стандарту ISO/IEC 17799.
2. В чому полягає зміст інформаційної безпеки?
3. Описати структуру стандарту ISO/IEC 17799.
4. Які засоби контролю використовуються при управлінні безпекою?
5. Які групи вимог до системи безпеки організації визначаються стандартом ISO/IEC 17799?
6. Які аспекти розглядаються при оцінці ризиків безпеки?
7. Які умови необхідно виконати для успішної реалізації системи інформаційної безпеки?
8. Яку інформацію повинен містити документ про політику інформаційної безпеки?
9. Описати структуру документу про політику інформаційної безпеки організації?

Вимоги до оформлення звітів по лабораторних роботах.

Звіт з лабораторної роботи оформлюється на аркушах формату А4, які заповнюються з однієї сторони. Текст повинен бути рукописним або друкованим на принтері. Використання кольорових чорнил дозволяється лише для ілюстративних матеріалів.

Звіт до лабораторної роботи формується відповідно до змісту і повинен містити такі розділи:

- титульна сторінка;
- мета роботи;
- короткі теоретичні відомості;
- порядок виконання роботи;
- опис усіх етапів виконання роботи;
- опис отриманих результатів;
- висновки за результатами роботи.

Звіт зшивається з лівої сторони листів формату А4. Титульна сторінка звіту обов’язково друкується на принтері.

Організація, контроль виконання та захист лабораторних робіт.

Лабораторні роботи виконуються кожним студентом згідно з графіком, який встановлений робочою програмою курсу. Графік виконання роботи студентом контролюється викладачем. Перед виконанням лабораторної роботи викладач опитує студентів, щоб визначити їх підготовленість до виконання роботи.

До виконання лабораторної роботи допускаються студенти, які мають теоретичні знання, що необхідні для виконання цієї роботи.

Захист лабораторної роботи відбувається тільки за наявності належно оформленого звіту з цієї роботи. Лабораторна робота подається і захищається безпосередньо після її виконання згідно з графіком, який встановлений робочою програмою курсу. Роботи, які захищені із запізненням, зараховуються з мінімальною кількістю балів. При захисті роботи студент демонструє результати виконаної роботи та відповідає на контрольні запитання за темою лабораторної роботи.

Кожна лабораторна робота, що виконана і захищена за графіком, оцінюється за бальною системою, яка встановлена робочою програмою курсу.

Список використаної літератури

Базова

1. Зегжда Д. П. Основы безопасности информационных систем. / Д. П. Зегжда, А. М. Ивашко. – М.: Горячая линия —Телеком, 2000.
2. Лукацкий А. Обнаружение атак / А. Лукацкий. – СПб.: БХВ-Петербург, 2003.
3. Максим М. Безопасность беспроводных сетей / Пер. с англ. А. В. Семенова / М. Максим, Д. Полино. – М.: ДМ К Пресс, 2004.
4. Петров А. Компьютерная безопасность: криптографические методы защиты / А. А. Петров. – Москва: ДМК Пресс, 2000.

Допоміжна

1. Галицкий А. В. Защита информации в сети — анализ технологий и синтез решений / А. В. Галицкий, С.Д. Рябко, В. Ф. Шаньгин - М.: ДМК Пресс, 2004.
2. Дшхунян В. Л. Электронная идентификация. Бесконтактные электронные идентификаторы и смарт-карты / В. Л. Дшхунян, В. Ф. Шаньгин - М.: АСТ: НТ Пресс, 2004.
3. ИСО/МЭК 14888-1—98. Информационная технология. Методы защиты. Цифровые подписи с приложением. Часть 1. Общие положения.
4. ИСО/МЭК 14888-2—99. Информационная технология. Методы защиты. Цифровые подписи с приложением. Часть 2. Механизмы на основе подтверждения подлинности.
5. ИСО/МЭК 10118-1—94. Информационная технология. Методы защиты. Хэш-функции. Часть 1. Общие положения.
6. ИСО/МЭК 10118-2—94. Информационная технология. Методы защиты. Хэш-функции. Часть 2. Хэш-функции с использованием я-битного блочного алгоритма шифрации.
7. ISO 17799 — Международный стандарт безопасности информационных систем. 2002.
8. ISO/IEC 14443-1 Identification Cards — Contactless integrated circuit(s) cards Proximity Cards Part 1: Physical characteristics International Standard. 2000.
9. ISO/IEC 14443-2 Identification Cards — Contactless integrated circuit(s) cards Proximity Cards Part 2: Radio frequency power and signal interface International Standard. 2001.

Інформаційні ресурси

1. Базовый стандарт организации беспроводных локальных сетей IEEE 802.11 [Електронний ресурс] / IEEE. – 1999. – Режим доступу до ресурсу: http://standards.ieee.org/reading/ieee/std/lanman/ 802 1 l-1999.pdf.

2. Беляев А. В. Методы и средства защиты информации [Електронний ресурс] / А. В. Беляев. – 2000. – Режим доступу до ресурсу: http:// www.citforum.ru/internet/infsecure/its2000_01.shtml.

3. Коротыгин С. Развитие технологии беспроводных сетей: стандарт IEEE 802.11 [Електронний ресурс] / С. Коротыгин. – 2000. – Режим доступу до ресурсу: http://www.ixbt.com/comm/wlan.shtml.

4. Скородумов Б. И. Стандарты для безопасности электронной коммерции в сети Интернет [Електронний ресурс] / Б. И. Скородумов. – 2000. – Режим доступу до ресурсу: http://www.stcarb.comcor.ru.

5. Daemen J. Rijndael. Document version 2 [Електронний ресурс] / J. Daemen, V. Rijmen. – 1999. – Режим доступу до ресурсу: www.esat.kuleuven.ac.be/~rijmen/rijndael.

Додатки

Додаток А

Найбільш небезпечні віруси 2008 року.

1. I-Worm.Klez
2. I-Worm.Lentin
3. I-Worm.Ganda
4. Trojan.PSW.M2
5. Win95.CIH
6. Macro.Word97.Thus
7. Troyan.PSW.MiniLD
8. PS-MPS-based
9. Backdoor.IRC.Zcrew
10. Macro.Word97.VMPC-based
11. Mnemonix.Oracle
12. I-Worm.LovGate
13. BW-based
14. Win95.Spaces -
15. Win32.HLLP.Hantaner
16. I-Worm.Tanatos
17. BackDoor.SubSeven
18. Backdoor.SdBot.gen
19. I-Worm.Sobig
20. VBS.Redlof
21. Petik.N

Додаток В

Основний модуль програми для знаходження НСД за допомогою алгоритму Евкліда.

```
long NSD (long a, long b)
{
        long r0, r1, q, t;
        if (a>b) {r0=a; r1=b;}
        else { r1=a; r0=b; }
        do {
                t=r0;
                q=ceil (t/r1);
                r1=t%r1;
                r0=(long) (t-r1)/q;
                } while (r1>0);
        return r0;
}
```