**Міністерство освіти і науки, молоді та спорту України**
**Вищий приватний навчальний заклад**
**Міжнародний економіко-гуманітарний університет імені академіка Степана Дем'янчука**

# Д.Б. Охота

# Методологія розробки сучасних криптографічних систем


**Науковий керівник:**
**Р.М.Літнарович, доцент,к.т.н.**

**Рівне – 2012 р.**

**УДК 004.353.4.**

Охота Д.Б. Методологія розробки сучасних криптографічних систем. Монографія. Науковий керівник Р.М. Літнарович. МЕГУ, Рівне, 2011.-76 с. Okhota D.B. Methodology of development of the modern cryptographic systems. Scientific leader R.M. Litnarovich. IEGU, Rivne, 2011.-76 p.

Рецензенти: В.Г.Бурачек, доктор технічних наук, професор
Є.С. Парняков, доктор технічних наук, професор
В.О.Боровий , доктор технічних наук, професор

**Відповідальний за випуск: Й.В. Джунь, доктор фізико-математичних наук, професор**

Розроблена система ефективно виконує поставлені перед нею завдання, тобто забезпечує криптографічний захист файлів різноманітних форматів які використовуються при інформаційній взаємодії - від звичайних текстових файлів до баз даних і навіть мультимедійних файлів.

**Ключові слова**: криптографія, кодування, методологія , система

Разработанная система эффективно выполняет поставленные перед ней задания, то есть обеспечивает криптографическую защиту файлов разнообразных форматов которые используются при информационном взаимодействии - от обычных текстовых файлов к базам данных и даже мультимедийным файлам.

Ключевые слова: криптография, кодировка, методология, система

The system is developed effectiv executes the tasks put before it, that provides cryptographic defence of files of various formats which are used for informative co-operation - from ordinary text files to the bases given and even multimedia files.

Keywords: cryptography, code, methodology, system

**Дмитро Борисович Охота,**
**спеціаліст системотехнік, магістрант**
**інформаційних технологій**



# ЗМІСТ

# Вступ

Людина - істота соціальна, от уже багато тисячоліть вона живе в суспільстві собі подібних. І цілком природно, що однією з найважливіших її здібностей є уміння спілкуватися з іншими людьми - передавати їм відомості про те, що відбувається в навколишньому світі, і про факти своєї суб'єктивної реальності. Друга сигнальна система - мова - по праву вважається однією з найважливіших відмінних ознак, що якісно відрізняють людство від тваринного світу. Характер інформаційного обміну між членами навіть самого дикого племені багаторазово перевищує по своїй складності все те, що можна побачити у тварин. Комунікація в людському суспільстві має ще одну відмінну ознаку - вона вузько виборча. Ми розмовляємо з різними людьми зовсім по-різному, і те, що повідомляємо одним, намагаємося сховати від інших.

Таким чином, із зародженням людської цивілізації виникло уміння передавати інформацію одним людям так, щоб вона не ставала відомою іншим. Поки люди використовували для передачі повідомлень винятково голос і жести, зробити це звичайно не складало особливої праці - потрібно було лише виключити присутність у

безпосередній близькості від тих людей, що розмовляють, тих для яких інформація, що повідомляються, не призначена. Однак іноді зовнішні фактори накладали на поводження співрозмовників обмеження, що не дозволяли їм укритися від сторонніх ушей і очей для проведення конфіденційної бесіди. Для дії в подібних обставинах була створена, а точніше склалася сама собою, система кодованих повідомлень. У різних ситуаціях вона носила зовсім різний характер - від окремого таємного знака, що говорить про настання визначеної події, до розвитих секретних мов, що дозволяли виражати думки практично будь-якої складності. Навіть у самому найпростішому випадку це була по своїй суті друга сигнальна система в мініатюрі, призначена для передачі обмеженого набору відомостей і відома як правило лише невеликій групі присвячених, фрагмент - великої чи маленької - альтернативної мови, і саме вона поклала початок розвиненому пізніше мистецтву таємно передавати повідомлення.

Звичайно, використання розвитої "секретної" мови для захисту переданих даних забезпечує набагато більшу можливість в комунікації, чим кілька таємних знаків, про які учасники домовилися напередодні, однак цей шлях має



і набагато більші витрати. За всіма присвяченими устежити важко, і рано чи пізно така мова стане зрозумілою тим, від кого намагаються приховати розмову. У цьому випадку виникне необхідність її замінити іншою, розробити ж досить могутню мову і навчити їй необхідну кількість людей дуже важко, а зробити це оперативно - неможливо. Тому подібний підхід до проблеми проходить тільки в особливих випадках, коли є сприятливі обставини. Так, він використовувався американцями під час другої світової війни: кораблі ВМФ США здійснювали зв'язок мовою нечисленного і компактно проживаючого індійського племені. На кожнім кораблі було кілька індіанців - "шифрувальників", у супротивника не було практично ніяких шансів роздобути собі такого "криптографа".

Треба зауважити, що тема передачі повідомлення за допомогою таємних знаків активно використовувалася письменниками і сценаристами пригодницького жанру. Згадайте фільм "Варіант омега", у якому радянський розвідник передавав сигнал про те, що він працює під контролем німецької контррозвідки, ставлячи чи пропускаючи крапку після визначеної фрази повідомлення. Але німці цей фокус розкусили і щораз акуратно

вирізували шматок магнітної стрічки з крапкою. Це, однак, їм не допомогло - доблесний радянський розвідник передав свій сигнал вибором квітів під час покупки їх у квіткаря-зв'язкового. До речі, про квіти – згадайте горщик з квіткою на підвіконні проваленої явочної квартири, що так легковажно проігнорував... Мова таємної передачі повідомлень жестами активно використовується й у наші дні представниками деяких кримінальних "спеціальностей", наприклад, шулерами - під час "роботи" пара шулерів веде дуже оживлену бесіду, непомітну для ока непрофесіонала.

З виникненням писемності задача забезпечення таємності і дійсності переданих повідомлень стала особливо актуальною. Дійсно, повідомлення, передане словесно чи показане жестами, доступне для стороннього тільки в той короткий проміжок часу, поки воно "в дорозі", а в його авторстві і дійсності в одержувача ніяких сумнівів бути не може, тому що він бачить свого співрозмовника. Інша справа, коли повідомлення записане - воно вже живе окремим життям і має свій шлях, що найчастіше далеко розходиться зі шляхом людини що, його створила. Записане на папері повідомлення існує в матеріальному світі набагато більш тривалий проміжок часу, і в людей, що



бажають ознайомитися з його змістом проти волі відправника й одержувача, з'являється набагато більше шансів зробити це. Тому саме після виникнення писемності з'явилося мистецтво тайнопису, мистецтво "таємно писати" - набір методів, призначених для секретної передачі записаних повідомлень від однієї людини іншій.

Дані про перші способи тайнопису дуже обривкові. Передбачається, що вона була відома в древньому Єгипті і Вавилоні. До нашого часу дійшли вказівки на те, що мистецтво секретного листа використовувалося в древній Греції. Перші дійсно достовірні зведення з описом методу шифрування відносяться до періоду зміни старої і нової ери й описують шифр Цезаря - спосіб, яким Юлій Цезар ховав свої записи від зайво зацікавлених очей. З висоти досягнень сучасної криптографії шифр Цезаря примітивний: у ньому кожна буква повідомлення замінялася на наступну за нею за алфавітом. Однак для того часу, коли уміння читати і писати було рідкісним винятком, його криптостійкості цілком вистачало. Використання шифру вирішувало проблему таємності переданого повідомлення, а проблема його дійсності вирішувалася практично сама собою:

- по-перше, для людини, що не знала шифр, було неможливо внести осмислені зміни в зашифровані повідомлення, що носили винятково текстовий характер, а зміни, внесені навмання приводили до того, що після розшифровки виходив безглуздий набір букв;
- по-друге, практично до зовсім ще недавніх по історичним міркам часів повідомлення, що відправляються, записувалися від руки, а кожна людина має свій індивідуальний, властивий тільки їй почерк, що дуже важко відтворити іншій людині; запам'ятати почерк навіть декількох десятків найбільш важливих своїх кореспондентів, не складає особливої праці.

Людство винайшло велике число способів секретного листа, багато з них були відомі ще в стародавності. В деяких способах використовуються фізичні особливості носіїв інформації. Так *симпатичне чорнило* зникає незабаром після написання ним тексту чи невидиме із самого початку. Але його можна знову зробити видимим, обробивши документ спеціальним хімічним реактивом чи освітивши променями визначеної частини спектра, звичайно - ультрафіолетом.



*Стеганографія* припускає, що переданий текст "розчиняється" у повідомленні більшого розміру з зовсім "стороннім" змістом. Але якщо взяти і витягти з нього деякі символи по визначеному закону, наприклад - кожен другий, чи третій, і т.д., одержимо цілком конкретне таємне повідомлення. *Шифрування* є перетворенням повідомлення за визначеними правилами, що робить його безглуздим набором знаків для непосвяченої в таємницю шифру людини.

Класифікувати способи засекречування переданих повідомлень можна по-різному, однак визначальних факторів лише два:

- використовуються для засекречування властивості матеріальних носіїв і матеріального середовища передачі інформації чи воно здійснюється незалежно від них;
- секретне повідомлення ховається чи воно просто робиться недоступним для усіх, крім одержувача.

Як відзначено вище, велике число способів засекретити повідомлення пов'язане з впливом на його носій як на матеріальний об'єкт. Це дуже цікава тема, однак вона є предметом вивчення фізики і хімії, і ніякого відношення до теорії інформації не має. Для масового

практичного застосування набагато більший інтерес представляють методи захисту даних, що спираються винятково на властивості самих даних і ніяк не зв'язані з особливостями їхнього фізичного представлення. Образно кажучи, при використанні методів даного типу бар'єр між власне повідомленням і зловмисником, що бажає його прочитати чи спотворити, зводиться винятково із самої інформації. Мова надалі піде тільки про такі способи захисту.

У залежності від відповіді на друге з приведених вище питань виходять різні класи способів засекречування даних - *стеганографія* і *шифрування*. Якщо розглядати інформацію окремо від її матеріального представлення, то де ж її тоді можна сховати? Відповідь однозначна: тільки в ще більшому масиві інформації - як голку в стозі сіна. У цьому і полягає принцип дії стеганографії. Наприклад, ми відправляємо нашому кореспонденту по електронній пошті файл із растровою чорно-білою картинкою, у якому найменш значущий біт у коді яскравості кожної крапки зображення буде елементом нашого таємного повідомлення. Одержувач листа витягне всі такі біти і складе з них "щире" повідомлення. Картинка присутня тут тільки для відводу очей, так і залишиться для непосвячених



простою картинкою. Стеганографія буває корисна, коли необхідно не просто передати *секретне повідомлення*, а *секретно передати секретне повідомлення*, тобто сховати сам факт передачі секретного повідомлення. Такий спосіб ведення таємної комунікації, однак, має ряд недоліків:

- по-перше, важко обґрунтувати його стійкість - раптом зловмисникам стане відомий спосіб "підмішування" секретних даних до "болванки" - масиву відкритих даних;
- по-друге, при його використанні обсяг переданих чи збережених даних різко збільшується, що негативно позначається на продуктивності систем їхньої обробки;

Інший підхід - не ховати факт передачі повідомлення, але зробити його недоступним сторонньому. Для цього повідомлення повинне бути записане так, щоб з його вмістом не міг ознайомитися ніхто за винятком самих кореспондентів - у цьому і полягає суть *шифрування*. І криптографія виникла саме як практична дисципліна, що вивчає і розробляє способи шифрування повідомлень.

Повернемося до історії. Чим активніше листування в суспільстві, тим більша відчувалася потреба в засобах його засекречування. Відповідно, виникали усе більш

хитромудрі шифри. Спочатку при зацікавлених особах з'явилися шифрувальники, потім групи з декількох шифрувальників, а потім і цілі шифрувальні відділи. Коли обсяги підлягаючої закриттю інформації стали критичними, у допомогу людям були створені механічні пристрої для шифрування. Треба сказати, що основними споживачами криптографічних послуг були дипломатичні і шпигунські місії, таємні канцелярії правителів і штаби військових з'єднань. Для цього етапу розвитку криптографії характерно наступне:

- захисту підлягали винятково текстові повідомлення, написані на природних мовах - інших типів дискретно представлених даних просто не існувало;
- шифрування спочатку здійснювалося вручну, пізніше винайдені порівняно нескладні механічні пристосування, тому використовувані тоді шифри були досить простими і нескладними;
- криптографія і криптоаналіз були скоріше мистецтвом, чим наукою, науковий підхід до побудови шифрів і їх розкриттю був відсутній;
- криптографія використовувалася в дуже вузьких сферах - тільки для обслуговування вищих правлячих шарів і військової верхівки держав;



- основною задачею криптографії був захист переданих повідомлень від несанкціонованого ознайомлення з ними, оскільки шифрувалися винятково текстові повідомлення, ніяких додаткових методів захисту від нав'язування помилкових даних не застосовувалося - імовірність одержати щось осмислене після розшифровування перекрученого зашифрованого тексту була мізерно мала в силу величезної надмірності, характерної для природних мов.

Поява в середині минулого сторіччя перших електронно-обчислювальних машин кардинально змінила ситуацію. З проникненням комп'ютерів у різні сфери життя виникла принципово нова галузь господарства - інформаційна індустрія. Обсяг циркулюючої в суспільстві інформації з тих пір стабільно зростає по експонентному закону - він приблизно подвоюється кожні п'ять років. Фактично, на порозі нового тисячоріччя людство створило інформаційну цивілізацію, у якій від успішної роботи засобів обробки інформації залежить саме благополуччя і навіть виживання людства в його нинішній якості. Зміни, що відбулися за цей період можна охарактеризувати в такий спосіб:

- обсяги оброблюваної інформації зросли за піввіку на кілька порядків;
- склалося таке положення речей, що доступ до визначених даних дозволяє контролювати значні матеріальні і фінансові цінності; інформація придбала вартість, що у багатьох випадках навіть можна підрахувати;
- характер оброблюваних даних став надзвичайно різноманітним і більш не зводиться до винятково текстових даних;
- інформація цілком "знеособилася", тобто особливості її матеріального представлення утратили своє значення - порівняйте лист минулого століття і сучасне послання по електронній пошті;
- характер інформаційних взаємодій надзвичайно ускладнився, і поряд із класичною задачею захисту переданих текстових повідомлень від несанкціонованого прочитання і перекручування виникли нові задачі сфери захисту інформації, що раніше стояли і зважувалися в рамках використовуваних "паперових" технологій - наприклад, підпис під електронним документом і вручення електронного документа "під розписку" -



мова про подібні "нові" задачі криптографії ще попереду;
- суб'єктами інформаційних процесів тепер є не тільки люди, але і створені ними автоматичні системи, що діють по закладеній у них програмі;
- обчислювальні "здібності" сучасних комп'ютерів підняли на зовсім новий рівень як можливості по реалізації шифрів, раніше немислимих через свою високу складність, так і можливості аналітиків по їх злому.

Перераховані вище зміни привели до того, що дуже швидко після поширення комп'ютерів у діловій сфері практична криптографія зробила у своєму розвитку величезний стрибок, причому відразу в декількох напрямках:

- по-перше, були розроблені стійкі блокові шифри із секретним ключем, призначені для рішення класичної задачі - забезпечення таємності і цілісності переданих чи збережених даних, вони дотепер залишаються "робочою конячкою" криптографії, найбільше часто використовуваними засобами криптографічного захисту;

- по-друге, були створені методи рішення нових, нетрадиційних задач сфери захисту інформації, найбільш відомими з який є задача підпису цифрового документа і відкритого розподілу ключів.

Як бачимо, термін "криптографія" далеко пішов від свого первісного значення - "тайнопис", "таємний лист". Сьогодні ця дисципліна поєднує методи захисту інформаційних взаємодій зовсім різного характеру, що спираються на перетворення даних по секретних алгоритмах, включаючи алгоритми, що використовують секретні параметри. Термін "інформаційна взаємодія" чи "процес інформаційної взаємодії" тут позначає такий процес взаємодії двох і більш суб'єктів, основним змістом якого є передача і/чи обробка інформації.



# Розділ 1 Сучасний стан методології побудови криптографічних систем

## 1.1 Особливості методології побудови систем кодування

Представлена криптосистема працює по визначеній методології (процедурі). Вона складається з двох алгоритмів шифрування (математичних формул), ключів, використовуваних цими алгоритмами шифрування, незашифрованого тексту і зашифрованого тексту (шифртексту).

Відповідно до методології спочатку до тексту застосовуються алгоритм шифрування (роздруківки процедур представлені в додатках) і ключ для одержання з нього шифртексту. Потім шифртекст передається до місця призначення, де той же самий алгоритм використовується для його розшифровки, щоб одержати знову текст.

### 1.1.1 Методологія з використанням ключа

У цій методології [1] алгоритм шифрування поєднує ключ з текстом для створення шифртексту. Безпека систем

шифрування такого типу залежить від конфіденційності ключа, використовуваного в алгоритмі шифрування, а не від збереження в таємниці самого алгоритму. Багато алгоритмів шифрування загальнодоступні і були добре перевірені завдяки цьому .

Але основна проблема, зв'язана з цією методологією, полягає в тому, як генерувати і безпечно передати ключі учасникам взаємодії. Як установити безпечний канал передачі інформації між учасниками взаємодії до передачі ключів?

Іншою проблемою є аутентифікація. При цьому існують дві серйозних проблеми:

Повідомлення шифрується кимсь, хто володіє ключем у даний момент. Це може бути власник ключа; але якщо система скомпрометована, це може бути інша людина.

- Коли учасники взаємодії одержують ключі, відкіля вони можуть довідатися, що ці ключі насправді були створені і послані уповноваженим на це обличчям?

Існують дві методології з використанням ключів - симетрична (із секретним ключем) і асиметрична (з відкритим ключем). Кожна методологія використовує свої власні процедури, свої способи розподілу ключів, типи



ключів і алгоритми шифрування і розшифровки ключів. Тому що термінологія, використовувана цими методологіями, може показатися незрозумілою, дамо визначення основним термінам:

| Термін | Значення | **Зауваження** |
|---|---|---|
| Симетрична методологія | Використовується один ключ, за допомогою якого проводиться як кодування, так і розшифровка з використанням того самого алгоритму симетричного кодування.<br><br>Цей ключ передається двом учасникам взаємодії безпечним образом до передачі зашифрованих | Часто називається методологією із секретним ключем. |

| | | |
|---|---|---|
| | даних. | |
| Асиметрична методологія | Використовує алгоритми симетричного кодування і симетричні ключі для кодування даних<br><br>Використовує алгоритми асиметричного кодування й асиметричні ключі для кодування симетричного ключа. Створюються два взаємозалежних аси-метричних ключі. Симетричний ключ, зашифрований з використанням одного асиметрич. | Часто називається методологією з відкритим ключем |



| | | |
|---|---|---|
| | ключа й алгоритму асиметричного кодування, повинний розшифровуватися з використанням іншого ключа і того ж алгоритму кодування.<br><br>Створюються два взаємозалежних асиметричних ключі. Один повинний бути безпечно переданий його власнику, а інший - тій особі, що відповідає за збереження цих ключів (СА – сертифікаційному центру ключів), до | |

| | | |
|---|---|---|
| | початку їхнього використання. | |
| Секретний ключ(1) | Симетрична методологія | Використовується один ключ, за допомогою якого відбувається як кодування, так і розшифровка. Див. вище |
| Секретний ключ(2) | Секретний ключ симетричного кодування | Симетричний секретний ключ |
| Секретний ключ(3) | Секретний ключ асиметричного кодування | Асиметричний секретний ключ<br><br>Асиметричні ключі створюються парами, тому що зв'язані один з одним. Вираз "секретний ключ" |



| | | |
|---|---|---|
| | | часто використовують для одного з пари аси-метричних ключів, що повинний триматися в секреті.<br><br>Асиметричний секретний ключ не має нічого загального із си-метричним секретним ключем. |
| Відкритий ключ (1) | Асиметрична методологія | Використовує пари ключів, що спільно створюються і зв'язані один з одним. Усе, що зашифровано одним ключем, |

| | | |
|---|---|---|
| | | може бути розшифровано тільки іншим ключем цієї пари. |
| Відкритий ключ (2) | Відкритий ключ асиметричного кодування | Асиметричні ключі створюються парами, кожний із двох ключів зв'язаний з іншим.<br><br>Вираз "відкритий ключ" часто використовують для одного з пари асиметричних ключів, що повинний бути усім відомий. |
| Сеансів ключ | | Використовується в асиметричній методології для кодування самих |



| | | |
|---|---|---|
| | | даних за допомогою симетричних методологій.<br><br>Це просто симетричний секретний ключ (див. вище) |
| Алгоритм кодування | Математична формула | Для симетричних алгоритмів вимагаються симетричні ключі.<br><br>Для асиметричних алгоритмів вимагаються асиметричні ключі.<br><br>Ви не можете використовувати симетричні ключі для асиметричних |

| | | |
|---|---|---|
| | | алгоритмів і навпаки. |
| Секретні криптосистеми | Використовують симетричні алгоритми і симетричні (секретні) ключі для кодування даних. | |
| Відкриті криптосистеми | Використовує Асиметричні алгоритми й асиметричні ключі для кодування сеансових ключів.<br><br>Використовують симетричні алгоритми і симетричні (секретні) ключі для кодування даних. | |



### 1.1.2 Симетрична (секретна) методологія

У цій методології і для шифрування, і для розшифровки відправником і одержувачем застосовується той самий ключ, про використання якого вони домовилися до початку взаємодії. Якщо ключ не був скомпрометований, то при розшифровці автоматично виконується аутентифікація відправника, тому що тільки відправник має ключ, за допомогою якого можна зашифрувати інформацію, і тільки одержувач має ключ, за допомогою якого можна розшифрувати інформацію. Тому що відправник і одержувач - єдині люди, що знають цей симетричний ключ, при компрометації ключа буде скомпрометована тільки взаємодія цих двох користувачів. Проблемою, що буде актуальна і для інших криптосистем, є питання про те, як безпечно поширювати симетричні (секретні) ключі.

Алгоритми симетричного шифрування використовують ключі не дуже великої довжини і можуть швидко шифрувати великі обсяги даних.

Порядок використання систем із симетричними ключами:

1. Безпечно створюється, поширюється і зберігається симетричний секретний ключ.
2. Відправник використовує швидкий симетричний алгоритм шифрування-розшифровки разом із секретним симетричним ключем до отриманого пакета для одержання зашифрованого тексту. Неявно в такий спосіб виробляється аутентифікація, тому що тільки відправник знає симетричний секретний ключ і може зашифрувати цей пакет. Тільки одержувач знає симетричний секретний ключ і може розшифрувати цей пакет.
3. Відправник передає зашифрований текст. Симетричний секретний ключ ніколи не передається по незахищених каналах зв'язку.
4. Одержувач використовує той же самий симетричний алгоритм шифрування-розшифровки разом з тим же самим симетричним ключем (який уже є в одержувача) до зашифрованого тексту для відновлення вихідного тексту й електронного підпису. Його успішне відновлення аутентифікує когось, хто знає секретний ключ.



## 1.2 Основні алгоритми кодування

Метод шифрування/дешифрування називають **шифром** (**cipher**). Деякі алгоритми шифрування засновані на тім, що сам метод шифрування (алгоритм) є секретним. Нині такі методи представляють лише історичний інтерес і не мають практичного значення. Усі сучасні алгоритми використовують **ключ** для керування шифровкою і дешифруванням; повідомлення може бути успішно дешифровано тільки якщо відомий ключ. Ключ, використовуваний для дешифрування може не збігатися з ключем, використовуваним для шифрування, однак у більшості алгоритмів ключі збігаються. Алгоритми з використанням ключа поділяються на два класи: симетричні (чи алгоритми з секретним ключем ) і асиметричні (чи алгоритми з відкритим ключем). Різниця в тім, що симетричні алгоритми використовують один ключ для шифрування і для дешифрування (чи ж ключ для дешифрування просто обчислюється по ключу шифровки). У той час як асиметричні алгоритми використовують різні ключі, ключ для дешифрування не може бути обчислений по ключу шифровки.

Симетричні алгоритми підрозділяють на **потокові шифри** і **блокові шифри**. Потокові дозволяють шифрувати інформацію побітово, у той час як блокові працюють з деяким набором біт даних (звичайно розмір блоку складає 64 біта) і шифрують цей набір як єдине ціле.

Асиметричні шифри (також іменовані алгоритмами з відкритим ключем, чи --- у більш загальному плані --- криптографією з відкритим ключем) допускають, щоб відкритий ключ був доступним усім (скажемо, опублікований у газеті). Це дозволяє кожному зашифрувати повідомлення. Однак розшифрувати це повідомлення зможе тільки потрібна людина (той, хто володіє ключем дешифрування). Ключ для шифрування називають **відкритим ключем**, а ключ для дешифрування --- **закритим** чи **секретним ключем**.

Сучасні алгоритми шифровки/дешифрування досить складні і їх неможливо реалізувати вручну. Дійсні криптографічні алгоритми розроблені для використання комп'ютерами чи спеціальними апаратними пристроями. У більшості випадків криптографія виконується програмним забезпеченням і мається безліч доступних криптографічних пакетів.



Взагалі говорячи, симетричні алгоритми працюють швидше, ніж асиметричні. На практиці обидва типи алгоритмів часто використовуються разом: алгоритм із відкритим ключем використовується для того, щоб передати випадковим чином генерований секретний ключ, що потім використовується для дешифрування повідомлення.

## 1.3 Криптоаналіз і атаки на криптосистеми

Криптоаналіз - це наука про дешифрування закодованих повідомлень не знаючи ключів. Мається багато криптоаналітичних підходів. Деякі з найбільш важливих для розроблювачів приведені нижче.

- **Атака зі знанням лише шифрованого тексту** (**ciphertext-only attack**): Це ситуація, коли той хто атакує не знає нічого про зміст повідомлення, і йому приходиться працювати лише із самим шифрованим текстом. На практиці, часто можна зробити правдоподібні припущення про структуру тексту, оскільки багато повідомлень мають стандартні

заголовки. Навіть звичайні листи і документи починаються з легко передбачуваної інформації. Також часто можна припустити, що деякий блок інформації містить задане слово.

- **Атака зі знанням змісту шифровки** (**known-plaintext attack**): Той хто атакує знає чи може угадати зміст усього чи частини зашифрованого тексту. Задача полягає в розшифровці іншого повідомлення. Це можна зробити шляхом обчислення ключа шифровки, або минаючи це.
- **Атака з заданим текстом** (**chosen-plaintext attack**): Той хто атакує має можливість одержати шифрований документ для будь-якого потрібного йому тексту, але не знає ключа. Задачею є підбір ключа. Деякі методи шифрування дуже уразливі для атак цього типу. При використанні таких алгоритмів треба ретельно стежити, щоб атакуючий не міг зашифрувати заданий їм текст.
- **Атака з підставкою** (**Man-in-the-middle attack**): Атака спрямована на обмін шифрованими повідомленнями і, особливо, на протокол обміну ключами. Ідея полягає в тім, що коли дві сторони обмінюються ключами для секретної комунікації ,



супротивник впроваджується між ними на лінії обміну повідомленнями. Далі супротивник видає кожній стороні свої ключі. В результаті, кожна зі сторін буде мати різні ключі, кожен з яких відомий супротивнику. Тепер супротивник буде розшифровувати кожне повідомлення своїм ключем і потім зашифровувати його за допомогою іншого ключа перед відправленням адресату. Сторони будуть мати ілюзію секретного листування, у той час як насправді супротивник читає всі повідомлення.

- **Атака за допомогою таймера** (**timing attack**): Цей новий тип атак заснований на послідовному вимірі часу, затрачуваного на виконання операції зведення в степінь по модулю цілого числа.

Приведене вище є, очевидно, найбільш важливим для практичної розробки систем. Мається безліч інших криптографічних атак і криптоаналітичних підходів. Якщо хто-небудь збирається створювати свій алгоритм шифрування, йому необхідно розуміти дані питання значно глибше, але для розуміння принципів роботи даної криптографічної системи досить і вищевикладеного.

# Розділ 2 Методологія побудови криптографічної системи

## 2.1 Задачі, які підлягають автоматизації розробкою криптографічної системи

Тепер перейдемо до характеристик задач, розв'язуваних криптографічними методами в наші дні. Інформаційні взаємодії між різними суб'єктами можуть носити різний, часом дуже заплутаний характер, відповідно існують різні погрози їх нормальному здійсненню. Для того, щоб поставити задачу захисту інформації, необхідно докладно відповісти на питання трьох наступних груп:

### 2.1.1 Що саме ми маємо намір захищати?

Ця група питань відноситься до інформаційного процесу, нормальний плин якого ми маємо намір забезпечити:

- хто є учасником інформаційного процесу;
- які задачі учасників інформаційного процесу;
- яким саме образом учасники процесу виконують задачі, що стоять перед ними;



### 2.1.2 Від чого ми збираємося захищати нашу систему?

Ця група питань охоплює можливі відхилення від нормального плину процесу інформаційної взаємодії:

- який критерій "нормального" проходження процесу інформаційної взаємодії;
- які можливі відхилення від "норми";

### 2.1.3 Від кого ми збираємося захищати нашу систему?

Ця група питань відноситься до тих суб'єктів, що починають ті чи інші дії для того, щоб відхилити процес від норми:

- хто може виступати як зловмисник, тобто починати зусилля для відхилення процесу інформаційної взаємодії від нормального плину;
- яких цілей домагаються зловмисники;
- якими ресурсами можуть скористатися зловмисники для досягнення своїх цілей;
- які дії можуть почати зловмисники для досягнення своїх цілей.

Розгорнута відповідь на перше питання є моделлю інформаційного процесу. Докладна відповідь на друге питання повинна включати критерій "нормальності" процесу і список можливих відхилень від цієї "нормальності", названих у криптографії *погрозами*, - ситуацій, що ми б хотіли зробити неможливими. Суб'єкт, що перешкоджає нормальному протіканню процесу інформаційної взаємодії, у криптографічній традиції називається "зловмисником", у якості його може виступати в тому числі і законний учасник інформаційного обміну, що бажає домогтися переваг для себе. Розгорнута відповідь на третє питання називається в криптографії *моделлю зловмисника*. Зловмисник - це не конкретне обличчя, а деяка персоніфікована сума цілей і можливостей, для якої справедливий принцип Паулі з фізики елементарних часток: два суб'єкти, що мають ідентичні цілі і можливості по їхньому досягненню, у криптографії розглядаються як той самий зловмисник.

Відповівши на всі перераховані вище питання, ми одержимо постановку задачі захисту свого інформаційного процесу. От приклад - задача захисту програмного забезпечення (п/з) від незаконного копіювання:



1. Суб'єкти цього процесу - постачальник п/з і користувач, у рамках процесу вони виконують наступні дії:

- постачальник передає користувачу дистрибутивний носій п/з;
- користувач інсталює п/з на своєму комп'ютері, одержуючи при цьому робочу копію п/з;
- користувач використовує робочу копію п/з, запускаючи на виконання програми, що входять у її склад;
- користувач може знищити робочу копію п/з на одному комп'ютері і інсталювати її на іншому.

2. Нормальним плином інформаційного процесу є ситуація, при якій робоча копія п/з встановлена і використовується тільки на одному комп'ютері. Відхиленням є інсталяція і використання робочих копій п/з більш ніж на одному комп'ютері.

3. Використовувати більш однієї копії програмного забезпечення можуть спробувати зробити наступні суб'єкти:

- законний власник п/з, що володіє дистрибутивним комплектом, може установити ще одну робочу

копію п/з, при цьому він може використовувати як вже інстальовану робочу копію п/з, так і дистрибутивний комплект;

- людина, що має доступ до комп'ютера з інстальованим п/з, може спробувати скопіювати його на інший комп'ютер і використовувати там.

Очевидно, що обоє зловмисника мають різні можливості по досягненню своїх цілей, і задача захисту в першому випадку (зловмисник - власник п/з) набагато складніша, чим у другому (у зловмисника немає дистрибутива п/з).

Різні задачі зі сфери захисту інформації відрізняються одна від другої саме різними відповідями на приведені вище питання. Тільки після детальної і вичерпної відповіді на всі питання задача захисту інформаційного процесу може вважатися поставленою і ми можемо приступити до проектування захисту.

## 2.2 Структурна схема роботи інформаційної системи

По великому рахунку, криптографічною може вважатися будь-яка функція перетворення даних, секретна сама по собі чи залежна від секретного параметра $\boldsymbol{S}$:



$\boldsymbol{T'} = \boldsymbol{f}(\boldsymbol{T})$, чи

$\boldsymbol{T'} = \boldsymbol{f}(\boldsymbol{T},\boldsymbol{S})$.

Нехай потрібно зашифрувати наступне повідомлення (відкритий текст):

DEAR DAD
SEND MORE MONEY AS SOON AS POSSIBLE TOM
«Дорогий тато. Якомога швидше пришли ще грошей.
Том.»

Один зі способів шифрування – проста заміна, при якій кожна буква відкритого тексту заміняється на якусь букву алфавіту (можливо, на ту ж саму). Для цього відправник повідомлення повинний знати, на яку букву в шифротексті варто замінити кожну букву відкритого тексту. Часто це робиться шляхом зведення потрібних відповідностей букв у вигляді двох алфавітів, наприклад так, як показано нижче в таблиці:

| Алфавіт | |
|---|---|
| Відкритий | ABCDEFGHIJKLMNOPQRSTUVWXYZ |
| Шифрувальний | BLUESTOCKINGADFHJMPQRVWXYZ |

Шифрограма виходить шляхом заміни кожної букви відкритого тексту на записану безпосередньо під нею букву шифрувального алфавіту.

ESBM EBE
HGSBPS PSDE AFMS AFDSY BP PFFD BP HFPPKLGS QFA

Дві алфавітні послідовності, використовувані в процесі шифрування, називаються, відповідно, відкритим і шифрувальним компонентом системи. Щоб одержувач шифрограми міг відновити відкритий текст і прочитати повідомлення, йому необхідно мати копію вищенаведеної таблиці. Дешифровщик повторює в зворотному порядку всі дії шифрувальника, розкриваючи тим самим зміст повідомлення.

У вищенаведеному прикладі використовувався алгоритм по буквеної заміни. Цей метод називається простою, чи моноалфавітною заміною. Ключ до даного шифру складається з таблиці, що містить відкритий і шифрувальний алфавіти, у якій указується, на яку букву в шифротексті варто замінити букву відкритого тексту. У такій криптографічній системі передбачається, що



алгоритм шифрування загальновідомий, тоді як ключ доступний тільки відправнику й одержувачу відповідних повідомлень.

У відкритому алфавіті букви розташовані в їхній звичайній послідовності; такий алфавіт називається прямою стандартною послідовністю. Шифрувальний же алфавіт у нашому прикладі складається з 26 букв латинського алфавіту, певним чином переставлених з використанням ключового слова BLUESTOCKING (букв. «синя панчоха»). Після ключового слова (ключової фрази) ключ далі записується з опущенням усіх тих букв, що вже з'явилися в цьому слові (чи в першому слові цієї фрази), а потім вписуються букви алфавіту, що залишаються, у звичайному порядку, знову ж з опущенням усіх, що раніше з'явилися букв. Так, якщо в якості ключової ми використовуємо фразу UNITED STATES OF AMERICA, то шифрувальний алфавіт буде виглядати в такий спосіб: U N I T E D S A O F M R C B G H J K L P Q V W X Y Z

За допомогою ключового слова (фрази), при шифруванні можна перемішати будь-яку алфавітну послідовність. Використання ключових слів полегшує відновлення відкритого і шифрувального компонента

системи, оскільки при цьому необхідно запам'ятати тільки відповідне ключове слово (фразу). Немає необхідності записувати (чи розгадувати) які б то не було таблиці: якщо пам'ятати ключове слово, то алфавітну послідовність завжди можна відновити по пам'яті. У вищенаведеній шифрограмі між словами збережені пробіли, однак шифровку можна зробити більш захищеною (чи, як говорять криптографи, стійкою до злому; шифр вважається тим більше стійким, чим довше він не піддається розкриттю) шляхом видалення міжслівних пробілів з остаточного шифротекста. Відповідно до сталої практики, шифротекст прийнято поділяти на групи з п'яти букв кожна. (Колись телеграфні компанії при стягуванні плати кожну групу з п'яти букв вважали як одне слово відкритого тексту.) Якщо забрати пробіли між словами, то нашу шифрограму можна було б записати так: ESBME BEMGS BPSPS DEAFM SAFDS YBPPF FDBPH FPPKL GSQFA. Заміна – одне з основних криптографічних перетворень. Іншим найважливішим шифрувальним алгоритмом є перестановка. У шифрі перестановки всі букви відкритого тексту залишаються без змін, але переставляються відповідно до заздалегідь обговореного правила. Тут також може використовуватися ключ, керуючий процедурою



шифрування. Так, використовуючи як ключ слово PANAMA, ми могли б зашифрувати вищезгаданий відкритий текст у такий спосіб:

| | |
|---|---|
| Ключ | P A N A M A |
| Числова послідовність | 6 1 5 2 4 3 |
| Блок перестановки | D E A R D A |
| | D P L E A S |
| | E S E N D M |
| | O N E Y A S |
| | S O O N A S |
| | P O S S I B |
| | L E T O M |
| Шифрограма | EPSNO OEREN YNSOA SMSSB |
| | DADAA IMALE EOSTD DEOSP L |

У цьому прикладі ключове слово використане для одержання шифрувальної числової послідовності шляхом нумерації букв ключового слова (відносно один одного) у порядку їхнього проходження зліва на право у стандартному алфавіті. Далі під числовою послідовністю в рядках, рівних по довжині ключовому слову, записаний відкритий текст. У процесі шифрування текст виписується вже по окремих стовпцях у порядку, обумовленому даною числовою послідовністю. Цей метод перестановки називається перестановкою стовпців, але можна обрати й інші «маршрути» перестановки, наприклад виписувати шифротекст випливаючи по діагоналі (ліворуч чи праворуч , чи ж чергуючи лівий і правий напрямки) чи по спіралі і т.п. Крім того, букви шифротексту можуть записуватися у вигляді різних геометричних фігур чи будь-якими іншими способами. Один з них складається в подвійному шифруванні шляхом повторної перестановки стовпців. При цьому й у першому, і в другому блоках перестановки може бути використане те саме ключове слово, хоча краще використовувати різні ключові слова. Такий шифр, що називається подвійною перестановкою, одержав широке поширення в 20 ст.



| | |
|---|---|
| Третім основним алгоритмом шифрування є дроблення. При цьому кожній букві відкритого тексту зіставляється більш одного символу шифротекста, після чого символи перемішуються (переставляютьс я) у визначеному порядку. Нижче приведена система, що демонструє процедуру дроблення з використанням | UNITED STATES OF AMERICA |

| | |
|---|---|
| знаменитого шифру Bifid, авторство якого приписується французькому криптографу Феліксові Марі Деластеллю. Ключова фраза | |
| | 1 2 3 4 5 |
| | 1 U N I T E |
| Полибіанский квадрат | 2 D S A O F |
| | 3 M R C B G |
| | 4 H K L P Q |
| | 5 V W X Y Z |
| Відкритий текст | DEARDADPLEASESENDMOREMONEY |
| Відповідний рядок | 21232224412212112323132115 |



| | |
|---|---|
| Відповідний стовпець | 15321314353252521142514254 |
| Шифротекст | DASOHSNUAAIDEERITGRWWUKVKY |

Спочатку складається шифрувальна таблиця розміром 5/5 (т.зв. полібІанский квадрат), куди порядково вписується шифрувальний алфавіт із ключовою фразою UNITED STATES OF AMERICA; причому заради того, щоб загальне число букв алфавіту не перевищувало 25, буква J опускається (оскільки ця буква, з одного боку, маловживана в англійських текстах, а з іншого боку – цілком може бути замінена буквою I, без якої-небудь утрати для змісту). У даній таблиці букві A, наприклад, відповідають координати 23, букві B – 34 і т.д.

Далі, у процесі шифрування під кожною буквою відкритого тексту в стовпчик записуються її табличні координати – номер рядка і, нижче, номер стовпця, а потім така цифрова послідовність переводиться за допомогою тієї ж таблиці назад у буквену форму, але цього разу вона читається вже в рядок, тобто 21 – буква D, 23 – буква A, 22 – буква S і т.д. При такому шифруванні координата рядка і

координата стовпця кожної букви виявляються роз'єднаними, що характерно саме для шифру, що роздрібнює.

Заміна, перестановка і дроблення являють собою основні криптографічні алгоритми. Ці три базових перетворення, найчастіше в сполученні один з одним, використовуються в більшості систем шифрування для створення дуже складних шифрувальних алгоритмів, особливо коли шифрування виробляється комп'ютером.

Код відрізняється від шифру тим, що підходить для перетворення відкритого тексту скоріше з лінгвістичної точки зору, ніж з чисто формальної, як у вищеописаних прикладах. (У повсякденному вживанні термін «код» звичайно плутають з терміном «шифр».) Кодування, як правило, містить у собі застосування великої таблиці чи кодового словника, де перераховані числові відповідності (еквіваленти) не тільки для окремих букв, але і для цілих слів і найбільш уживаних фраз і речень. Наприклад, така кодова група, як AABRT, цілком може відповідати відкритому тексту «якнайшвидше». Раніше у дипломатичній і військовій криптографічній системах часто використовувалися дуже великі коди. Однак з



появою комп'ютерів найбільше поширення одержали шифри.

# Розділ 3 Розробка програми

## 3.1 *Засоби створення системи*

Сучасні технології потребують досконалих засобів для їх реалізації. Для вибору необхідних засобів проектування і створення системи необхідно визначити основні технічні характеристики майбутньої системи. В залежності від цих характеристик можливо обґрунтувати вимоги до програмних засобів проектування та реалізації проекту системи. Тож розроблена система повинна володіти такими основними властивостями:

- кінцева програмна системи має працювати в графічному середовищі операційних систем Windows 98\ME\NT\2000\XP, оптимально використовувати ресурси комп'ютерної системи, такі як оперативна пам'ять, дисковий простір та ресурси центрального процесора;
- зручний віконний графічний користувацький інтерфейс з усіма можливостями та перевагами , які дає використання системи GUI (Graphic User Interface – графічний інтерфейс користувача ) ОС Windows;

- забезпечення якісної та непомітної для користувача системи обробки та виправлення помилок, що можуть виникнути при роботі системи і викликані такими причинами, як збої в роботі операційної системи, невірні дії користувача або внутрішні помилки в програмі при роботі з файлами та дисковими носіями.

Окрім цих основних вимог потрібно визначити такий спосіб зберігання оперативної інформації, який би забезпечив надійність та ефективність роботи програми з нею та зручність для користувача. Крім того необхідно відгородити користувача від технічних подробиць реалізації роботи з файлами, а тому створити такі умови, щоб йому не потрібно було турбуватися про правильність і оптимальність збереження інформації.

Звичайно вся інформація, яка має значний обсяг та потребує збереження після закінчення роботи програми зберігається на комп'ютері у файлах на дискових носіях. Тож необхідно вибрати тип файлів для збереження матеріалів, оптимальний метод роботи системи з цими файлами. Збереження даних в текстових файлах не забезпечить тих можливостей, які потрібні користувачам програми. Інший розповсюджений спосіб роботи з файловими потоками – створення файлів, які будуть



складатися з таких логічних одиниць як записи. Запис або структура – це складний тип даних, присутній в багатьох мовах програмування високого рівня. Але лише деякі системи програмування мають можливості представляти файли у вигляді таких записів, тобто набору логічних одиниць, які можуть містити дані різних мовних типів, а робота з такими файлами як правило відносно повільна і тому неефективна. Виконання таких простих операцій як сортування або пошук певних значень може потребувати досить значних обчислювальних ресурсів.

Іншим способом організації зберігання даних може бути створення нового типу файла і оригінальна програмна реалізація роботи з ним. Але ця задача потребує великого обсягу кропіткої роботи і це як правило невиправдані витрати в порівнянні з таким способом збереження інформації, як використання табличних файлів форматів сучасних систем управління базами даних. Звичайно для створення і модифікації таких файлів необхідно або використовувати відповідні СКБД, або спеціальні програмні засоби (драйвери чи програмні бібліотеки). Але цей недолік повністю нівелюється перевагами роботи з файлами баз даних. Тому в системі

використовуються технології баз даних для збереження інформації та її використання.

Всі ці вимоги і привели до того, що в якості засобу розробки системи було обране інтегроване середовище програмування Delphi. 6.0 виробництва корпорації Borland Software Corp., цей програмний пакет являє собою середовище для візуальної розробки інтерфейсної частини програмної системи, редактор програмного коду, засоби для роботи з програмними компонентами, утиліти для роботи з базами даних та інші додаткові можливості для ефективної розробки програмних засобів.

Основними перевагами вибраного середовища є:

- система Delphi об'єднує в єдиному графічному середовищі візуальну систему проектування інтерфейсу програми, компілятор мови Object Pascal, додаткові засоби для управління проектом програми та засоби для роботи з найбільш розповсюдженими локальними та клієнт-серверними СКБД; всі перераховані елементи об'єднані в одному програмному комплексі, який відноситься до так званих засобів швидкої розробки програмного забезпечення RAD (Rapid Application Development);



- високопродуктивний компілятор в машинний код забезпечує компіляцію вихідних текстів на мові програмування Object Pascal;

- розвинута структура об'єктно-орієнтованого програмування і бібліотека візуальних компонентів VCL (Visual Component Library), яка інкапсулює візуальні та не візуальні елементи керування, за допомогою яких будується графічний інтерфейс і логічна структура системи;

- глибока інтеграція системи з іншим продуктом – Borland C++ Builder, який за часи свого існування став потужним інструментом для розробки прикладних систем різного рівня складності, особливо популярним на теренах бувшого Радянського Союзу та в Європі, розділення з цим продуктом бібліотеки візуальних компонентів;

- включення бібліотеки стандартних шаблонів STL (Standard Template Library), яка інкапсулює в собі багато алгоритмів та контейнерних класів-шаблонів (вектори, списки, стеки та інше);

- наслідувана із стандартного C++ гнучка система роботи з виключеннями із доповненнями в діалекті Borland C++;

- можливість встановлення та використання компонентів сторонніх виробників, як для системи C++ Builder, так і для Delphi;

- можливість створення нових класів та компонентів, що забезпечує використання всіх переваг ООП;

- орієнтація програмного комплексу на роботу з базами даних в локальній, файл-серверній та клієнт-серверній архітектурі за допомогою програмної підсистеми BDE (Borland Database Engine), яка забезпечує доступ та роботу з найбільш розповсюдженими системами управління базами даних;

- повне використання таких сучасних технологій як COM, ADO, SOAP, CORBA, MIDAS, наявність засобів для локалізації програмних продуктів, використання всіх можливостей ОС Windows (DDE, OLE, DLL);

- можливість створення програмного коду для розробки міжплатформенних програмних систем за допомогою альтернативної бібліотеки CLX (Component Library for Cross-Platforming Applications), при цьому можна створювати програми, які можуть були скомпільовані в виконувані файли для ОС Windows або Linux; це особливо актуально в останній час , коли значно



зросла популярність Unix-подібних систем, зокрема вільно-розповсюджених систем сімейства Linux;

- можливість інтеграції системи з іншими допоміжними сумісними засобами для розробки програмного забезпечення.

Звичайно, як і кожна інша, дана програмна система має свої недоліки, основними з яких є досить високі вимоги до апаратної та програмної складових комп'ютерної системи, порівняно невелика швидкість компіляції вихідного тексту, великі розміри кінцевих виконуваних файлів. Більшість цих недоліків можна вирішити застосуванням додаткових програмних засобів, а за своїми якісними характеристиками система Borland Delphi займає одне із провідних місць на ринку систем розробки програмного забезпечення.

## 3.2 Методологія роботи із криптографічною системою

Переді мною стояла задача дослідити методологію захисту інформаційних потоків на підприємстві від несанкціонованого доступу і розробити систему здатну забезпечити конфіденційність руху документів між суб’єктами інформаційної взаємодії.

Проаналізувавши документообіг на підприємстві, я визначив, що значна його частина ( більше 90 % ) відбувається в електронному вигляді, тобто моя задача зводиться до розробки криптографічної системи, спроможної реалізувати криптографічний захист файлів різноманітних форматів які використовуються при інформаційній взаємодії - від звичайних текстових файлів до баз даних і навіть мультимедійних файлів.

Представлена програма – пакет криптографічних програм реалізованих на основі симетричного алгоритму, призначений для криптозахисту інформаційних потоків на підприємстві.

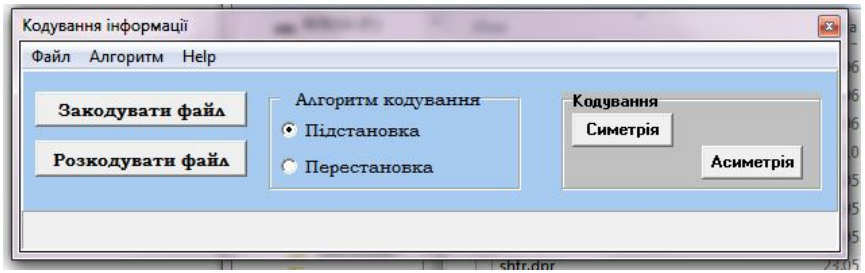


Рис.1

Зручний інтерфейс забезпечує безтурботну роботу користувача, всі функції пакету здійснюються через кнопки на передній панелі.



Для вибору алгоритму необхідно скористатись пунктом “Алгоритми кодування” на передній панелі.

Вибір файла для кодування – кнопка “Закодувати файл”.

Вибір файла для розкодування – кнопка “Розкодувати файл”.

Після відкриття стандартного діалогового вікна пропонується вибрати файл для роботи.

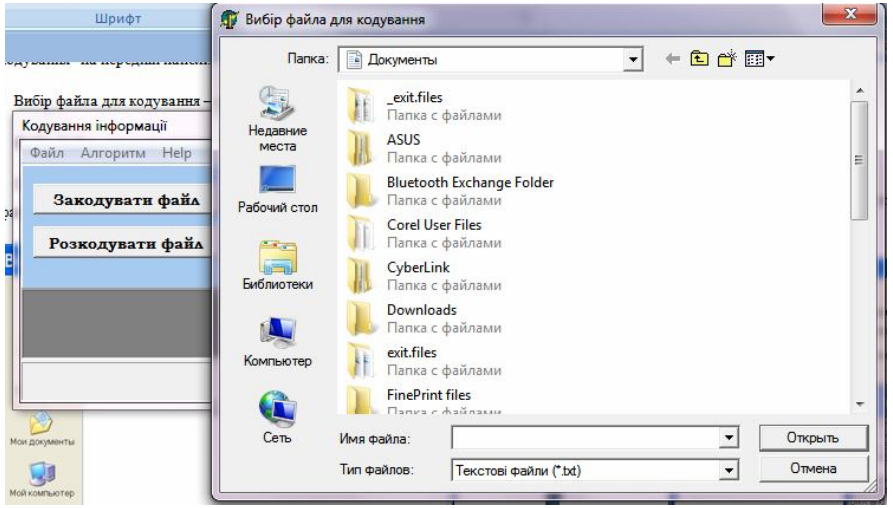


Рис.2

Після вибору файла з’являється вікно введення ключа. Ключ - комбінація з шести, або більше символів,

яку потрібно запам’ятати, або зафіксувати любим іншим способом, так як в подальшому він (ключ) знадобиться для розшифрування.

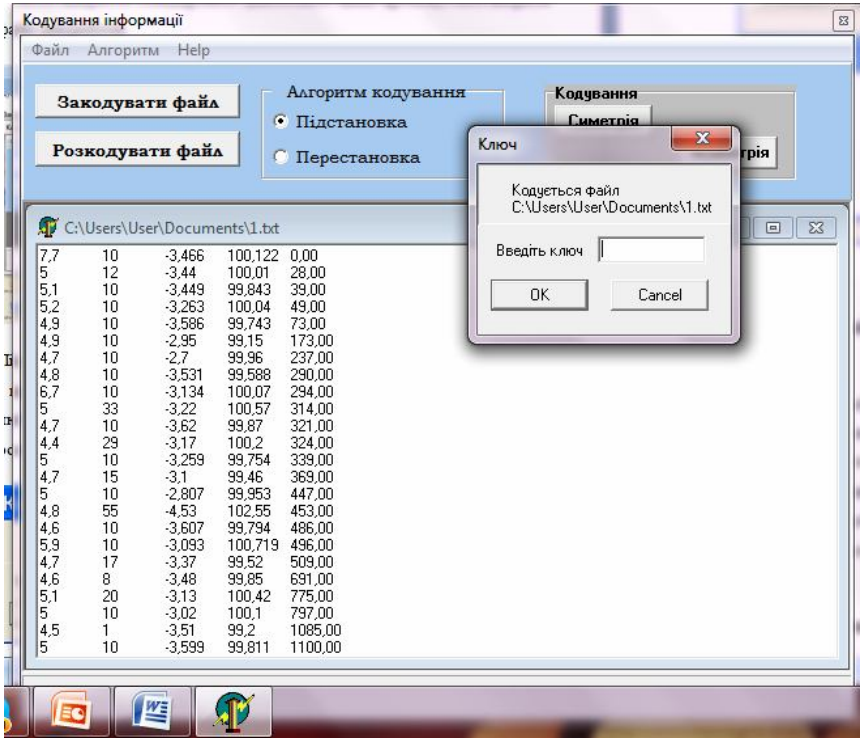


Рис.3

Після кодування програма створює закодований файл з розширенням sh1 або sh2. Аналогічно здійснюємо розкодування.



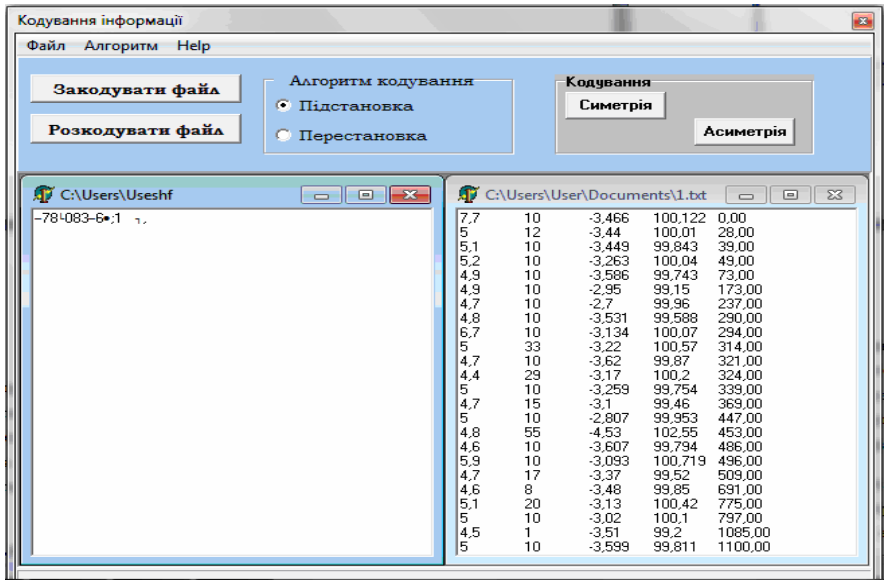


Рис.4

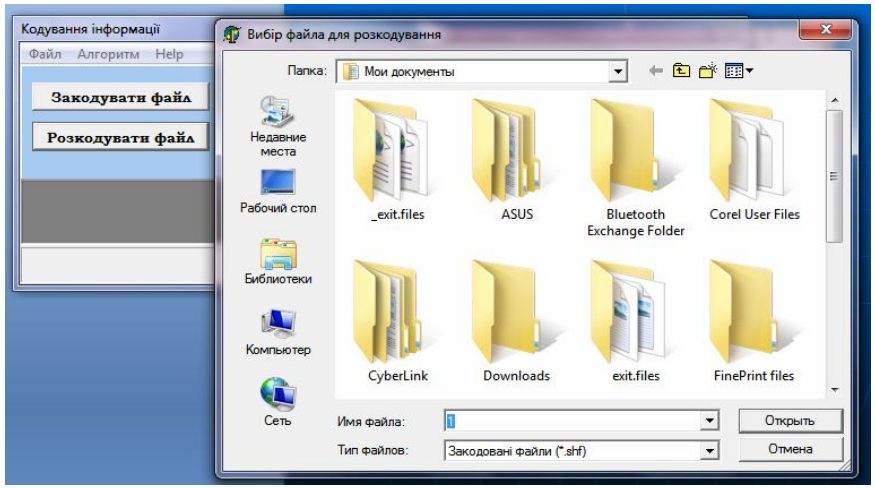


Рис.5

## 3.3 Аналіз технічного забезпечення необхідного для впровадження системи

Система абсолютно невибаглива до апаратних ресурсів. Тобто для впровадження системи необхідна мінімальна конфігурація, здатна винести операційну систему з графічним інтерфейсом починаючи з Windows 95. Дефіцит оперативної пам'яті або швидкості процесора буде відображатись на швидкості роботи програми таким ж чином як і на швидкості роботи операційної системи.

## 3.4 Аналіз соціально-психологічних умов роботи колективу після впровадження системи

Впровадження системи зустрічається з тими ж проблемами які мають місце завжди при впровадженні нового програмного забезпечення. Інертність, консервативність головним чином членів колективу перед-пенсійного віку викликають деяку настороженість та презавзяте відношення до нового програмного продукту. Але завдячуючи тому, що система проста в користуванні, має дружній сучасний інтерфейс, а головним чином тому, що система справляється з поставленими перед нею



задачами і не переобтяжена різного роду надмірностями, строк її впровадження та адаптації мінімально короткий.

## Висновки

Після проведення тестування можна стверджувати, що в загальному розроблена система ефективно виконує поставлені перед нею завдання, тобто забезпечує криптографічний захист файлів різноманітних форматів які використовуються при інформаційній взаємодії - від звичайних текстових файлів до баз даних і навіть мультимедійних файлів.

В результаті проведених нами досліджень була реалізована прикладна програма зі зручним інтуїтивно зрозумілим інтерфейсом, розрахованим на роботу користувачів без особливої попередньої підготовки. Програма реалізує всі зазначені при постановці задачі функції і разом з тим, завдяки відкритій технології, є можливість доповнення її функціональних можливостей та вдосконалення програми в цілому. Одними із основних напрямків розвитку програми можна вважати її переведення на платформу Unix-подібних систем Linux, особливо беручи до уваги те, що в сучасних економічних умовах ці операційні системи набули великої популярності по причинам своєї дешевизни та надійності, а також те що

система створена в середовищі Delphi, випуск версії якого для ОС Linux корпорацією Borland Software вже відбувся. Ще один перспективний напрямок вдосконалення системи – вдосконалення її аналітичних функцій.

Підведемо підсумки:

- криптографія - це набір методів захисту інформаційних взаємодій від відхилень від їх нормального, штатного протікання, викликаних злочинними діями різних суб'єктів, методів, що базуються на секретних алгоритмах перетворення інформації, включаючи алгоритми, що не є власне секретними, але секретні параметри, що вони використовують;
- історично першою задачею криптографії був захист переданих текстових повідомлень від несанкціонованого ознайомлення з їх змістом, що знайшло відображення в самій назві цієї дисципліни, цей захист базується на використанні "секретної мови", відомої тільки відправнику й одержувачу, усі методи шифрування є лише розвитком цієї філософської ідеї;
- з ускладненням інформаційних взаємодій у людському суспільстві виникли і продовжують



виникати нові задачі по їх захисту, деякі з них були вирішені в рамках криптографії, що зажадало розвитку принципово нових підходів і методів.

В міру утворення інформаційного суспільства, великим державам стають доступні технологічні засоби тотального нагляду за мільйонами людей. Ви можете планувати політичну кампанію, обговорювати ваші податки, чи займатися різними незаконними справами. Чи ви можете робити щось, відчуваючи, що це не повинно бути заборонено, однак є таким. Що б це не було, ви не бажаєте, щоб ваше особисте електронне повідомлення, чи конфіденційні документи були прочитані кимсь ще, крім адресата. Немає нічого некоректного в тім, що ви хочете зберегти в таємниці свою інформацію. Тому криптографія стає одним з основних інструментів гарантуючих конфіденційність, довіру, авторизацію, електронні платежі, корпоративну безпеку і незліченну безліч інших важливих речей. Широке поширення криптографії є одним з способів захисту від ситуації, коли людина раптом виявляє, що живе в тоталітарній державі, яка може контролювати кожен її (людини) крок. Таким чином представлена робота являє собою не тільки і не стільки найпростішу

криптографічну систему, а й в першу чергу потужний інструмент боротьби проти тоталітаризму держави.

## Список використаних джерел


1. А.В.Аграновский, А.В.Балакин, Р.А.Хади, "Классические шифры и методы их криптоанализа", М: Машиностроение, Информационные технологии, №10, 2001.
2. А.А.Молдовян, Н.А.Молдовян, Советов Б.Я., "Криптография": СПб.: Издательство "Лань", 2000.
3. Чмора А.Л. , "Современная прикладная криптография" , М.: Гелиос АРВ, 2001.
4. Устинов Г.Н. , "Основы информационной безопасности" , М: Синтег, 2000.
5. Анин Б. , "Защита компьютерной информации" , СПб: БХВ, 2000.
6. Романец Ю.В., Тимофеев П.А. , "Защита информации в компьютерных системах и сетях" , М: Радио и связь, 2001.
7. Саломаа А.: "Криптография с открытым ключом", Москва: "Мир", 1995. - 318с.
8. Конспект з дисципліни “Основи захисту інформації”, викладач Ольшанський П. В.





9. Жельников В., Криптография от папируса до компьютера. М.: ABF, 1996.
10. Форсайт Дж., Малькольм М., Моулер К. Машинные методы математических вычислений /Пер. с англ. Х. Д. Икрамова. М.: Мир, 1980.
11. Криптозащита текстовых файлов Василий Текин 15.05.2001 Мир ПК, #05/2001

## Додаток

Лістінги деяких функцій та процедур.

*Реалізація на мові Pascal в системі Delphi*

**Процедура яка реалізує перестановочний алгоритм кодування**

```
procedure TfmMain.shifr_comb(Sender: TObject);
var f1,f2 : file of char;
   c : char;
   i,j,l : byte;
   s,s1 : string;
   keynum : array[1..255] of byte;
   last : boolean;

procedure cod(s:string;var ss:string);
var i1 : byte;
begin
ss:=s;
if OD1.Title='Вибір файла для кодування'
 then for i1:=1 to l do ss[keynum[i1]]:=s[i1]
```



```
 else for i1:=1 to l do ss[i1]:=s[keynum[i1]]
end;

begin
l:=length(key);
for i:=1 to l do begin
  keynum[i]:=1;
  for j:=1 to l do if key[i]>key[j] then inc(keynum[i]);
end;
 assignfile(f1,OD1.FileName);reset(f1);
  fn:=OD1.FileName;
  delete(fn,length(fn)-2,3);
if OD1.Title='Вибір файла для кодування'
 then fn:=fn+'shf'
 else  fn:=fn+'fhs';
 assignfile(f2,fn);rewrite(f2);
last:=false;
repeat s:='';
 for i:=1 to l do begin
   if eof(f1) then begin last:=true; break end;
   read(f1,c); s:=s+c
 end;
 if last then s1:=s else cod(s,s1);
```

```
 for i:=1 to length(s) do write(f2,s1[i])
until last; closefile(f1); closefile(f2);
end;
```



## Процедура яка реалізує підстановочний алгоритм кодування

```
procedure TfmMain.sh_any_f(Sender: TObject);
var f1,f2 : file of byte;
    b,b1 : byte;
      k : longint;

procedure cod(b:byte;var bb:byte);
var i1,a : byte;
begin
i1:=k mod length(key);
a:=ord(key[i1]);
bb:=a xor b
end;

begin
assignfile(f1,OD1.FileName);reset(f1);
 fn:=OD1.FileName;
 delete(fn,length(fn)-2,3);
if OD1.Title='Вибір файла для кодування'
 then fn:=fn+'shf'
 else  fn:=fn+'fhs';
```

```
assignfile(f2,fn);rewrite(f2);
k:=0;
while not eof(f1) do begin
  read(f1,b); inc(k);
  cod(b,b1);
  write(f2,b1)
end;
closefile(f1); closefile(f2);
end;
```



*Процедура яка реалізує частотний аналіз символів в текстовому файлі*

```
procedure TfmMain.frequency(Sender: TObject);
var f:file of char;
   k:longint;
   c:char;
   fr:array[#0..#255] of real;
begin
assignfile(f,OD1.FileName);reset(f);
for c:=#0 to #255 do fr[c]:=0;
k:=0;
while not eof(f) do begin
  read (f,c);
  k:=k+1;
  fr[c]:=fr[c]+1
end;
closefile(f);
with DM2.frequency do begin
 active:=false;
 EmptyTable;
 active:=true
```

```
end;


for c:=#0 to #255 do  begin
  fr[c]:=fr[c]/k;
  if fr[c]>1E-10 then with DM2.frequency do begin
     append;
     FieldByName('ascii').AsInteger:=ord(c);
     FieldByName('symbol').AsString:=c;
     if ord(c)>159 then
       FieldByName('symbol').AsString:=chr(ord(c)+64);
     FieldByName('freq').AsFloat:=fr[c];
     post;
   end
   end
end;
```



**Дмитро Борисович Охота,**
**спеціаліст системотехнік, магістрант**
**інформаційних технологій**

# Методологія розробки сучасних криптографічних систем

ІН 11М

**Комп'ютерний набір, верстка і макетування та дизайн в редакторі Microsoft®Office® Word 2003 Д.Б. Охота.**

**Редагування Р.М.Літнарович.**

**Науковий керівник Р. М. Літнарович, доцент, кандидат технічних наук**

**Міжнародний Економіко-Гуманітарний Університет ім. акад. Степана Дем'янчука**

**Кафедра математичного моделювання**
**33027,м.Рівне,Україна**
**Вул.акад. С.Дем'янчука,4, корпус 1**
**Телефон:(+00380) 362 23-73-09**
**Факс:(+00380) 362 23-01-86**
**E-mail:mail@regi.rovno.ua**
**E-mail:__dima__90__@mail.ru**
**E-mail:litnarovich@windowslive.com**