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

Факультет кібернетики
Кафедра математичного моделювання

Українець Ірина Василівна

АНАЛІЗ І ДОСЛІДЖЕННЯ КРИПТОГРАФІЧНИХ ЗАСОБІВ ЗАХИСТУ ІНФОРМАЦІЇ НА БАЗІ «УКРГАЗБАНК»

8.080201 – „Інформатика"

А В Т О Р Е Ф Е Р А Т

магістерської дисертації на здобуття академічного ступеня магістра з інформатики

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

Рівне – 2011

Ірина Василівна Українець,

магістрант інформаційних технологій

УДК 614.2

Українець І.В. Аналіз і дослідження криптографічних засобів захисту інформації на базі «Укргазбанк». Автореферат магістерської дисертації на здобуття академічного ступеня магістра з інформатики. Науковий керівник Р.М.Літнарович. МЕГУ, Рівне, 2011.- 32 с.

Робота виконана на кафедрі математичного моделювання Міжнародного економіко-гуманітарного університету імені академіка Степана Дем'янчука

Рецензенти:В.Г.Бурачек, доктор технічних наук, професор
В.О.Боровий, доктор технічних наук, професор
…….Є.С.Парняков, доктор технічних наук, професор
Відповідальний за випуск: Й.В.Джунь, доктор фіз.-мат. наук, професор

Важливість і актуальність питань захисту інформації вже давно вийшли на одне з перших місць серед інших завдань, що вирішуються в процесі проектування, створення та використання сучасних інформаційних систем. Причини такої підвищеної уваги до цієї проблеми цілком очевидні - від якості заходів захисту інформації безпосередньо залежить економічна безпека організації.
Ключові слова – інформація, захист, криптографія, безпека.

Важность и актуальность вопросов защиты информации уже давно вышли на одно из первых мест среди других заданий, которые решаются в процессе проектирования, создания и использования современных информационных систем. Причины такого повышенного внимания к этой проблеме полностью очевидны - от качества мероприятий защиты информации непосредственно зависит экономическая безопасность организации.
Ключевые слова - информация, защита, криптография, безопасность.

Importance and actuality of questions of priv already a long ago went out on one of the first places among other tasks, which decide in the process of planning, creation and use of the modern informative systems. Reasons of such enhanceable attention to this problem are fully obvious - on quality of measures of priv economic security of organization depends directly.
Keywords - information, defence, cryptography, safety.

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Організація магістерської роботи спрямована на дослідження та аналіз криптографічних засобів захисту інформації на базі УКРГАЗ Банку. Широке використання обчислювальних мереж, призводить до того, що з'являється велика можливість для несанкціонованого доступу до переданої інформації.

Останнім часом зріс інтерес до питань захисту інформації. це пов'язують з тим, що стали більш широко використовуватися обчислювальні мережі, що призводить до того, що з'являються великі можливості для не-
санкціонованого доступу до переданої інформації.

Важливість і актуальність питань захисту інформації вже давно вийшли на одне з перших місць серед інших завдань, що вирішуються в процесі проектування, створення та використання сучасних інформаційних систем. Причини такої підвищеної уваги до цієї проблеми цілком очевидні - від якості заходів захисту інформації безпосередньо залежить економічна безпека організації.

Банківська інформація завжди була об'єктом пильної уваги всякого роду зловмисників. У системі забезпечення безпеки банку захист інформації відіграє найбільш важливу роль. Сучасні технології дають банкам переваги в організації систем дос-тавки товарів і послуг. Використання електронних засобів зв'язку дозволяє реалізувати:

- ✓ електронні платежі і розрахунки в точці продажу;
- ✓ клієнтські термінали, які здійснюють прямий зв'язок з банком;
- ✓ домашнє банківське обслуговування за допомогою персонального комп'ютера або телефону;
- ✓ обмін електронними даними в мережі з розширеним набором послуг;
- ✓ технології електронних банківських карт, включаючи магнітні та електронні пластикові карти.

Актуальність роботи викликана потребою створення програмної системи захисту інформації у комп'ютерних мережах на основі криптографічних засобів захисту інформації, які забезпечують найбільший ступінь захисту інформації. Проектування та розробка системи захисту інформації створена на основі новітньої платформи для розробки Visual Studio 2010, що є лідером серед засобів для розробки складних програмно-інженерних систем.

В магістерській роботі за основу взято симетричну та асиметричну криптографію. Особливу увагу приділено алгоритмам захисту DES, ГОСТ, Blowfish та RSA. Велику увагу у роботі приділено сліпому цифровому підпису. Сліпий цифровий підпис реалізовано за допомогою алгоритму RSA. Найбільш широке застосування протокол сліпих підписів знайшов у сфері цифрових грошей.

Мета роботи — розробити криптографічно-стійку систему захисту інформації, що представляється у вигляді програмної системи, що дозволяє проводити кодування та передачу інформаційних ресурсів у мережі. Основними завданнями для досягнення мети стали:

- Дослідження проблеми захисту інформації на основі криптографічних методів;
- Розробити програмну систему захисту інформації для зберігання обробки та використання закодованої інформації;
- Аналіз симетричної та асиметричної криптографії, дослідження роботи шифрів блочного типу. Виділення найбільш надійних шифрів захисту інформації;
- Організувати базу даних користувачів для обміну повідомлень із відповідним цифровим підписом кожного із користувачів системи;
- Реалізувати програмну систему у середовищі Visual Studio 2010 на мові програмування C#.

Наукова новизна полягає у реалізації системи захисту інформації, що дозволяє проводити обмін, зберігання обробку інформаційних ресурсів із врахуванням крипто-стійкості та

відповідної надійності для банківських систем. Систему захисту інформації розроблено за допомогою новітньої мови програмування C#, що дозволило врахувати можливості передачі та ідентифікації електронних повідомлень між користувачами системи.

Практична значимість і реалізація роботи полягає в розробці програмного продукту, який є перевірений, протестований та впроваджений на базі УКРГАЗ Банку. Розроблена криптографічна система захисту інформації відповідає усім вимогам, які були поставлені до даного програмного забезпечення. Перевагою системи є забезпечення можливості її роботи на будь-якому ПК. Головною вимогою ставиться наявність NET Framework 4.0.

Апробація роботи. Окремі розділи дисертації були докладені і отримали одобрення на наукових конференціях студентів і аспірантів у 2010 і 2011 роках, а також на науковому семінарі кафедри математичного моделювання.

Публікації. Основні положення дисертації опубліковані в монографії автора : Українець І.В. Аналіз і дослідження криптографічних засобів захисту інформації на базі «Укргазбанк». Науковий керівник Р.М.Літнарович. МЕГУ, Рівне, 2011.- 150 с.
http://elartu.tntu.edu.ua/handle/123456789/1568

Основні положення дисертації, що виносяться на захист:

- повний опис практичного застосування криптографії у сфері інформаційної безпеки;
- огляд криптографічних алгоритмів захисту інформації;
- розробка програмного продукту;
- опис алгоритму реалізації системи;
- середовище розробки;
- інструкція користувачеві по використанню;
- реалізація подій та методів головної форми вікна;
- реалізація об'єктів шифрування даних;
- програмування додаткових інтерфейсів.

Структура і об'єм роботи:

Магістерська дисертація складається із вступу, трьох розділів, розбитих на підрозділи, висновків і списку використаних джерел. Обсяг дисертації 111 сторінок,31 рисунка, 13 таблиць. Список використаної літератури із 67 найменувань, в тому числі 3 на іноземній мові.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі обгрунтовується актуальність теми, дається короткий огляд результатів, що мають безпосереднє відношення до теми роботи, та загальна характеристика магістерської дисертації.

В першому розділі описується практичне застосування криптографії у сфері інформаційної безпеки, приводяться криптографічні засоби захисту інформації, математичні основи і дається порівняльний аналіз симетричних та асиметричних алгоритмів.

У другому розділі розглядається огляд криптографічних алгоритмів захисту інформації – шифри DES, ГОСТ, Blowfish, RSA.

Розробці програмного продукту посвячується третій розділ. Дається програмна реалізація алгоритму, описується алгоритм реалізації системи, акцентується увага на середовищі розробки, приводиться інструкція користувачеві по використанню розробленого автолром програмного продукту.

В додатку 1 дається реалізація подій та методів головної форми вікна.

В додатку 2 відображається реалізація об’єктів шифрування даних.

В додатку 3 приводиться програмування додаткових інтерфейсів.

3.1. Опис алгоритму реалізації системи

Алгоритм реалізації системи захисту інформації можна умовно розділити на декілька етапів:

- ✓ Вибір сервера баз даних (БД) і організація даних про користувачів системи;
- ✓ Налаштування програмного середовища для роботи із БД;
- ✓ Організація відправки електронних листів користувачам системи, що включають шифрований файл (ключ);
- ✓ Організація технології обміну ключами, перевірка цифрового підпису ключа на основі шифру RSA;
- ✓ Організація методів потокової взаємодії із файлами довільного типу;
- ✓ Організація процесу кодування даних шифром DES;
- ✓ Організація процесу кодування даних шифром ГОСТ;
- ✓ Організація процесу кодування даних шифром Blowfish;

На початковому етапі розробки системи було сформовано БД. В якості сервера БД обрано Microsoft Access 2003, який є найбільш поширеним і зручним при використанні для користувачів.

У базі даних зберігається інформація про користувачів системи захисту інформації. У БД зберігається поля із наступною інформацією:

- ➢ прізвище;
- ➢ ім’я;
- ➢ по батькові;
- ➢ email;
- ➢ відкритий ключ n;
- ➢ відкритий ключ e;

Заповнена таблиця зберігає (рис.3.1) необхідну інформацію про користувачів у системі захисту інформації. Поле у якому міститься електронна адреса кожного із користувачів призначене для автоматизованої відправки електронних листів на їх адреси. Ключі, що зберігаються у БД призначені для кодування файлу, що додається до листа.

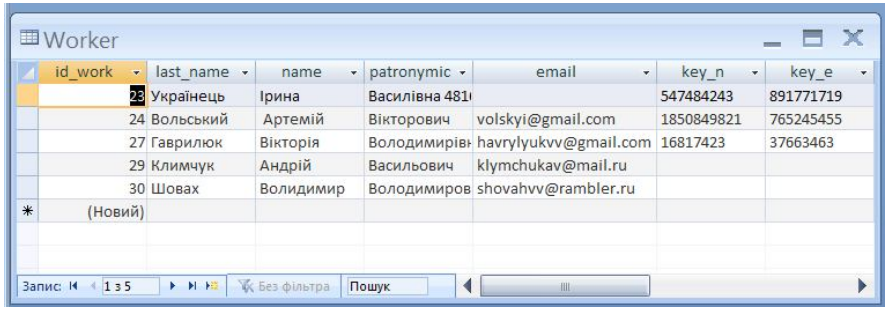

Рис. 3.1. Користувачі у базі даних

Загалом файл, що прикріпляється до листа — це зашифрований файл за допомогою шифру RSA. У файл можна записувати довільну текстову інформацію, що реалізується у системі захисту інформації. Даний файл, що надсилається можна дешифрувати тільки закритим ключом кожного із користувачів системи, для якого цей файл відправлено. Ключ дешифрується закритим ключом кожного користувача.

Така технологія організації відправки листів дозволяє організувати обмін ключами для шифрів DES, ГОСТ та Blowfish. Для даних шифрів ключ повинен знаходитися в таємниці. Для даних шифрів ключ, що використовується для шифрування та дешифрування інформації. У шифрі RSA є відкриті ключі для шифрування та закритий для дешифрування. Надійність шифру вважається найбільш криптостійкою, тому обмін ключами реалізується на основі даного шифру захисту інформації.

Відправка листа реалізована за допомогою smtp сервера електронної пошти. Для розсилки повідомлень користувачам БД у системі потрібно вказати наступні налаштування електронного ящика:

- ✓ smtp сервер;
- ✓ smtp порт;
- ✓ email;
- ✓ пароль.

Тобто потрібно вказати налаштування власної електронної пошти кожного із користувачів системи, щоб можна було проводити відправку листів. Відправка листа на мові програмування C# має наступний вигляд:


```
private void SendMail(string toAdd, string temaEmail, string textEmail)
{
    string smtpServer = "";
    string smtpPort = "";
    string email = "";
    string password = "";
    //Читаємо параметри із файлу налаштувань електронноїй пошти
    try
    {
        FileStream myFileStream = new FileStream("email.ini",
FileMode.Open,
            FileAccess.Read);
        BinaryReader binReade = new BinaryReader(myFileStream);
        smtpServer = CryptionClass.Decrypt(binReade.ReadString(),
"34msrti98iew");
        smtpPort = CryptionClass.Decrypt(binReade.ReadString(),
"34msrti98iew");
        email = CryptionClass.Decrypt(binReade.ReadString(),
"34msrti98iew");
        password = CryptionClass.Decrypt(binReade.ReadString(),
"34msrti98iew");
        binReade.Close();
        myFileStream.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show("Помилка при відкритті файлу, що містить
параметри підключення до email " + ex.ToString(), "Помилка!");
    }

    SmtpClient Smtpotpravka = new SmtpClient();
    try
    {
        Smtpotpravka = new SmtpClient(Convert.ToString(smtpServer),
int.Parse(smtpPort));
        Smtpotpravka.Credentials = new NetworkCredential(email,
password);
    }
    catch (Exception ex)
    {
        MessageBox.Show("Проблема при підключенні до електронної пошти.
" + ex.Message.ToString());
    }
    if (toAdd == "")
        return;
    //Відпрвка пошти із електронного адресу
    try
    {
```

```
        MailMessage Email = new MailMessage();
        Email.From = new MailAddress(email);
        Email.To.Add(new MailAddress(toAdd));
        Email.Subject = temaEmail;
        Email.Body = textEmail;
        Attachment attachData = new Attachment("Ключ.key");
        Email.Attachments.Add(attachData);
        Smtpotpravka.Send(Email); // Отправляем сообщения
    }
    catch (Exception ex)
    {
        MessageBox.Show("Проблема при відправці листа. " +
ex.Message.ToString(), temaEmail);
    }

}
```

Наступною задачою алгоритму є організація кодування інформації довільного типу за допомогою симетричних шифрів DES, ГОСТ та Blowfish. Для читання і запису файлів довільного типу використовуються потоки для роботи із двійковими даними. При цьому розуміється читання і запис послідовності бітів, де файл не має значення. Це реалізовано наступним програмним кодом на мові програмування C#:

```
myFileCode = dlgCode.FileCode;
myFileDecode = dlgCode.FileDecode;
myKey = dlgCode.Key;
int temp = 0;
int leng = 8;
byte[] key = new byte[leng];

FileStream readKey = null;
BinaryReader binKey = null;
try
{
    readKey = new FileStream(myKey, FileMode.Open, FileAccess.Read);
    binKey = new BinaryReader(readKey);
    for (temp = 0; temp < leng; temp++)
    {
        key[temp] = binKey.ReadByte();
    }
    binKey.Close();
}
catch (IOException errorKey)
{
    MessageBox.Show("Помилка при відкриті файлу, який містить
ключ!!!\n" + errorKey.Message, "Помикла!");
    is_good = false;
```



```
}
finally
{
    if (readKey != null)
        readKey.Close();
}
if (temp != leng && is_good)
{
    MessageBox.Show("Розмір файлу, який містить ключ є надто малим!!!",
"Помилка!");
    is_good = false;
}
FileStream stream = null;         //Файл, який потрібно читати
FileStream streamWrite = null;    //Файл в який потрібно записувати
BinaryReader reader = null;
BinaryWriter writer = null;
if (is_good)
{
    try
    {
        stream = new FileStream(myFileCode, FileMode.Open,
FileAccess.Read);
        streamWrite = new FileStream(myFileDecode, FileMode.Create,
FileAccess.Write);
        reader = new BinaryReader(stream);
        writer = new BinaryWriter(streamWrite);
        byte[] data = new byte[8];
        DES my_des = new DES();
        my_des.des_key(ref key);
        long processed = 0;
        long fileSize = stream.Length;
        long lushok = fileSize % 8;
        fileSize -= lushok;
        int i = 0;
        while (processed < fileSize)
        {
            for (i = 0; i < 8; i++)
                data[i] = reader.ReadByte();
            my_des.des_enc(ref data);
            for (i = 0; i < 8; i++)
                writer.Write(data[i]);
                            processed += 8;    //Переміщаємося на 8 біт
        }
        for (i = 0; i < lushok; i++)
            writer.Write(reader.ReadByte());
    }
    catch (IOException expt)
    {
        MessageBox.Show("Помилка при відкриті файлу, який потрібно
кодувати!!!\n" + expt.Message, "Помикла!");
```

```
            is_good = false;
        }
        finally
        {
            if (stream != null)
            {
                stream.Close();
            }
            if (streamWrite != null)
                streamWrite.Close();
            if (is_good)
            {
                //Фуксуємо, що кодування проведено шифром DES
                action = 1;           //Кодування файлу проведенно успішно
            }
            timeEnd = DateTime.Now;
            this.Invalidate();
        }
}
```

Показано приклад реалізації для шифру DES. Для шифрування використовується спеціальний об'єкт `DES my_des = new DES()`. У якому реалізовано основний механізм шифрування та дешифрування інформації даним шифром.
Аналогічно обробку файлів реалізовано за допомогою шифрів ГОСТ та Blowfish. У даному випадку різницю має спосіб шифрування даних та розмір відповідних ключів для даних шифрів.
Тами чином, на основі організації такого алгоритму вдалося досягти великої гнучкості системи захисту інформації, що передбачає мережеву взаємодію із надійністю та зручністю обміну інформацією із забезпеченням відповідної потрібної надійності усієї системи.

3.2. Середовище розробки

Розробка програм проводилася у середовищі Visual Studio 2010, що дозволило розробити ефективну систему програмного захисту інформації. Представимо короткий опис основних можливостей та переваг середовища розробки Visual Studio [48]. Вкажемо також процес створення та розгортки системи захисту інформації.
Visual Studio - це набір інструментів розробки, заснованих на використанні компонентів, і інших технологій для створення

потужних, продуктивних додатків. Крім того, середа Visual Studio оптимізована для спільного проектування, розробки та розгортання корпоративних рішень.
Середовище розробки Visual Studio представляє собою повний набір засобів розробки для створення веб-додатків ASP.NET, XML (веб-служби), настільних та мобільних додатків [46]. Visual Basic, Visual C# і Visual C++ використовують єдине інтегроване середовище розробки (IDE), яка дозволяє спільно використовувати засоби і спрощує створення рішень на базі декількох мов. Крім того, в цих мовах використовуються функціональні можливості платформи. NET Framework [29], яка дозволяє отримати доступ до ключових технологій, що спрощує розробку веб-додатків ASP і XML (веб-служби).
Програма розроблена на мові програмування C#. Синтаксис C # дуже виразний, але простий у вивченні. Усі, хто знайомий з мовами C, C + або Java з легкістю впізнають синтаксис з фігурними дужками, характерний для мови C#. Розробники, які знають будь-який з цих мов, як правило, зможуть добитися ефективної роботи з мовою C# за дуже короткий час. Синтаксис C# робить простіше те, що було складно в C++, і забезпечує потужні можливості, такі як типи значень Nullable, перерахування, делегати, лямбда-виразу і прямий доступ до пам'яті, чого немає в Java.
Програма на мові C# виконується в середовищі. NET Framework - інтегрованому компоненті Windows, що містить віртуальну систему виконання (середовище CLR) і уніфікований набір бібліотек класів [41]. Середовище CLR представляє собою комерційну реалізацію Майкрософт інфраструктури CLI (common language infrastructure), міжнародного стандарту, основи середовищ виконання і розробки з тісною взаємодією мов і бібліотек.
Опишемо процес створення програми за допомогою середовища Visual Studio 2010 на основі Windows Forms, оскільки наша програма представлена у вигляді стандартної програми Windows.

Проект програми Windows Forms є основою більшості рішень, що включають Windows Forms. Такий проект просто створити в інтегрованому середовищі розробки (IDE) [46].

Створення проекту програми Windows Forms:

- ✓ Запустіть Visual Studio 2010.
- ✓ У меню Файл виберіть команду **Создать** та виберіть **Проект**.
- ✓ Відкриється діалогове вікно **Новый проект**.
- ✓ На панелі Установленные шаблоны розгорніть Visual C#, оберіть **Windows**.
- ✓ Над середньою областю в списку потрібну версію NET Framework.
- ✓ У середній області (рис. 3.2) виберіть шаблон **Приложение Windows Forms**.
- ✓ У текстовому полі **Имя** задайте ім'я проекту.
- ✓ У текстовому полі **Расположение** вкажіть каталог, в якому потрібно зберегти проект. Натисніть кнопку **ОК**.

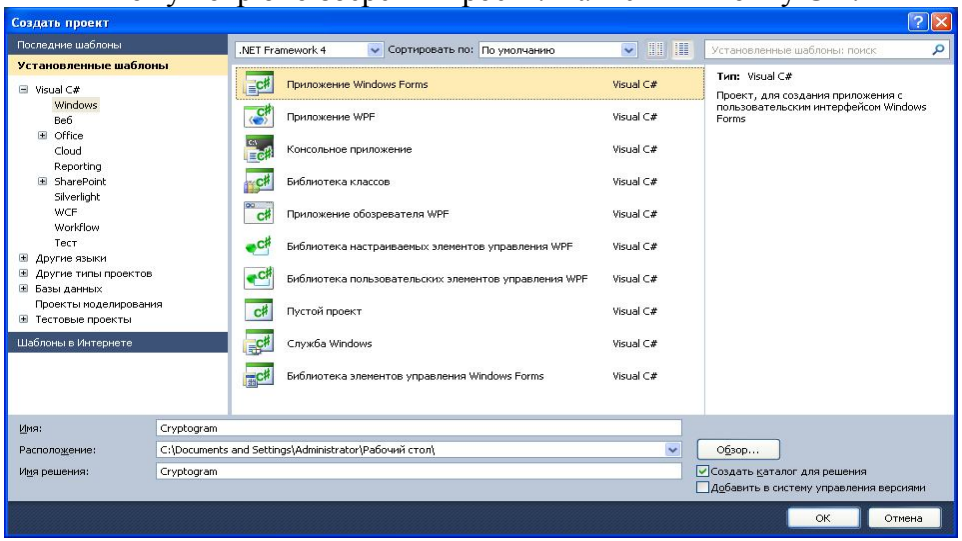

Рис. 3.2. Створення програми Windows Forms

Таким чином створюється пустий проект, який містить у собі форму, яка запускається автоматично, тобто є головним вікном програми (рис 3.3).

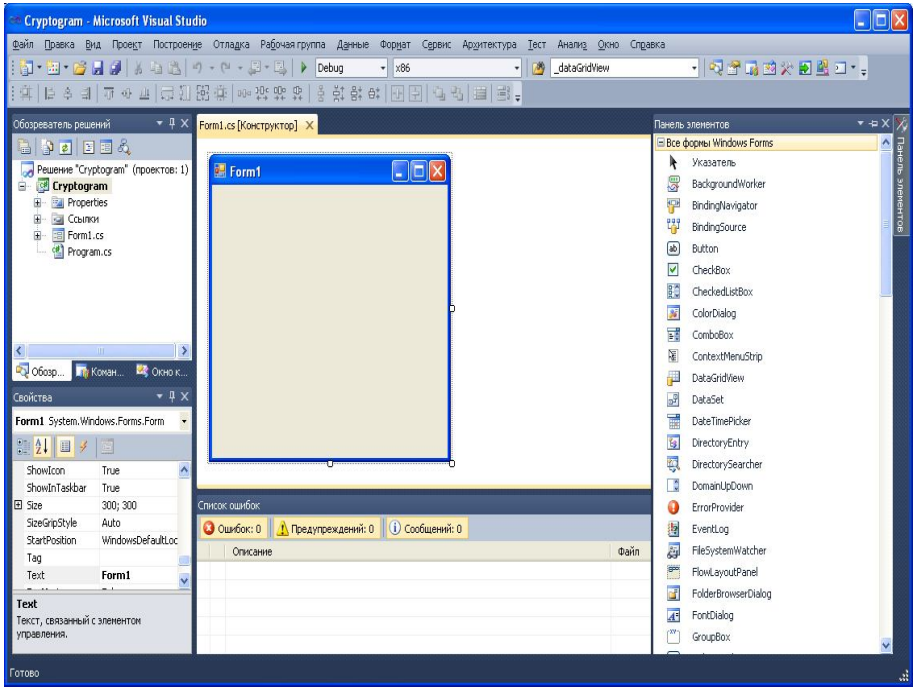

Рис. 3.3. Новостворений проект рішення

Після того, як створено проект варто звернути увагу на можливості роботи і організації інтерфейсу користувача, який має великий набір властивостей та інтерактивних компонентів. Windows Forms є технологією інтелектуальних клієнтів для NET Framework; це набір керованих бібліотек, що забезпечують поширені завдання програм, наприклад читання і запис у файлову систему. За допомогою середовища розробки типу Visual Studio можна створювати додатки Windows Forms, які відображають інформацію, запитують введення від користувачів і обмінюються даними з віддаленими комп'ютерами по мережі. У Windows Forms форма є видимою поверхнею, на якій відображається інформація для користувача. Зазвичай додаток Windows Forms будується шляхом поміщення елементів керування на форму і написанням коду для реагування на дії користувача, такі як клацання миші або натиснення клавіш.

Елемент управління - це окремий елемент користувацького інтерфейсу, призначений для відображення або вводу даних. При виконанні користувачем будь-якої дії з формою або одним з її елементів управління, створюється подія. Додаток реагує на ці події за допомогою коду і обробляє події при їх виникненні. Windows Forms включає широкий набір елементів керування, які можна додавати на форми: текстові поля, кнопки, розкриваємі списки, перемикачі і навіть веб-сторінки. Список всіх елементів керування, які можна використовувати у формі. Якщо існуючий елемент управління не задовольняє потребам, в Windows Forms можна створити власні настроювані елементи керування за допомогою класу UserControl.

До складу Windows Forms входять елементи призначеного для користувача інтерфейсу з розширеними функціями, відповідними можливостями потужних додатків, таких як Microsoft Office. Використовуючи елементи управління ToolStrip і MenuStrip, можна створювати панелі інструментів і меню, що містять текст і малюнки, що відображають підменю та містять в собі інші елементи керування, такі як текстові поля і поля з випадним списком.

За допомогою конструктора Windows Forms Visual Studio, що підтримує перетягування, можна легко створювати додатки Windows Forms: Досить виділити елемент керування курсором і помістити його на потрібне місце на формі. Конструктор надає такі засоби, як лінії сітки і "прив'язка ліній" для подолання труднощів вирівнювання елементів управління. І в разі використання Visual Studio або компіляції з командного рядка можна використовувати елементи керування FlowLayoutPanel, TableLayoutPanel і SplitContainer для створення просунутих розміток форми за мінімальний час і з мінімальними зусиллями [48].

Після розробки усієї системи проект був наповнений (рис. 3.4) відповідними вікнами, класами, елементами керування та ін..

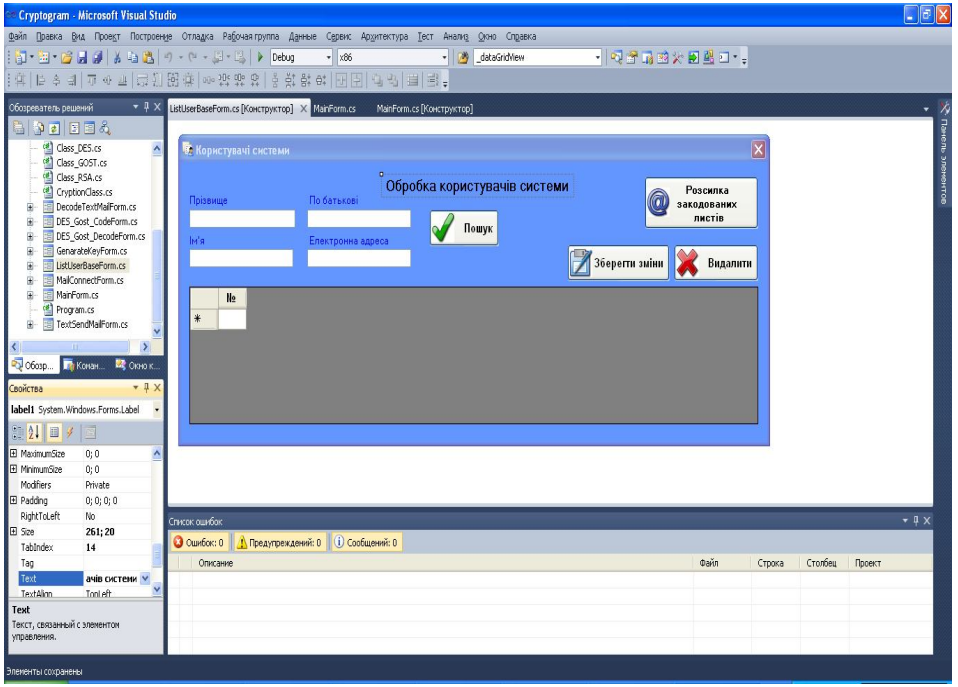

Рис. 3.4. Розробка системи захисту інформації

Виходячи з вище зазначеного, розроблено програмну систему захисту інформації у середовищі Visual Studio NET. Ефективна система інформування та допомоги візуального представлення елементів надзвичайно сильно допомагає розробляти та реалізувати програмні системи різної складності та гнучкості функціонування.

3.3. Інструкція користувачеві по використанню

Програмну систему захисту інформації розроблено на базі додатків Windows Forms. Середовищем розробки є Visual Studio 2010. Мова програмування C#.

Головне вікно програми (рис. 3.5) представляє собою стандартний додаток Windows. Головне вікно складається із заголовка, верхнього меню, робочої області та стрічки стану. Верхнє меню містить у собі основні команди роботи програми. Верхнє меню включає в себе меню: Файл, Кодування, Налаштування.

Для повної функціональності програми потрібно провести налаштування системи захисту інформації. Для цього призначене (Рис. 3.6) меню **Налаштування**.

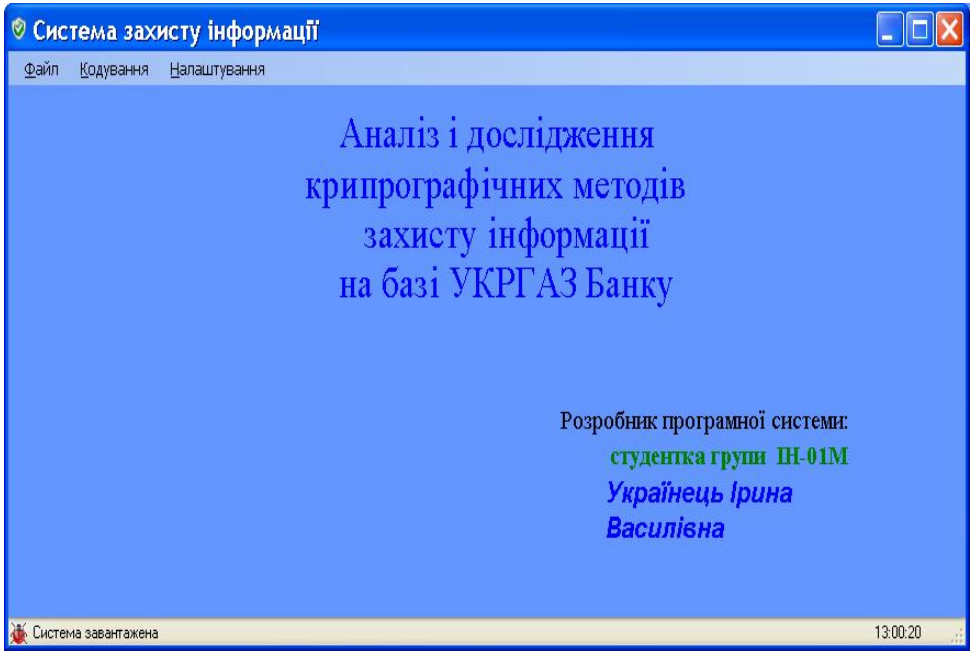

Рис. 3.5. Головне вікно програми

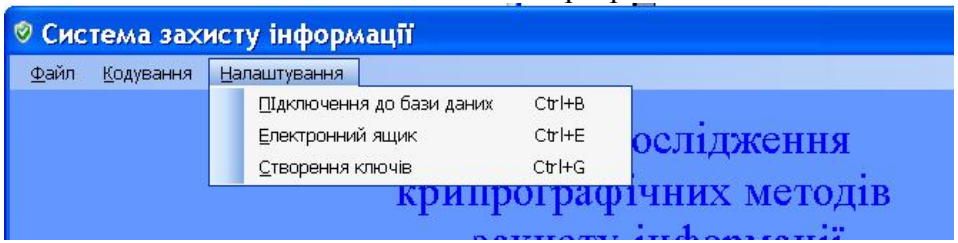

Рис. 3.6. Вміст меню **Налаштування**

На початковій стадії роботи із програмою потрібно вказати розміщення бази даних. Для цього потрібно виконати команду **Налаштування → Підключення до бази даних**. Після цього буде відображено (рис. 3.7) вікно **Підключення до бази даних**.

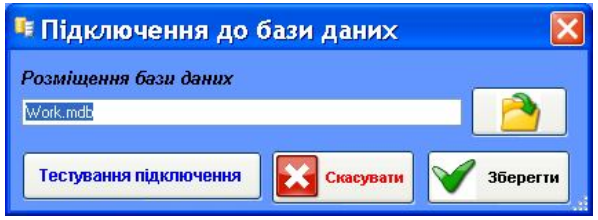

Рис. 3.7. Налаштування підключення до БД

У полі *Розміщення бази даних* потрібно вказати файл бази даних, якщо шлях до файлу не вказано, то база даних буде шукатися у каталозі програми. Кнопка  дозволяє поводити пошук файлу (рис. 3.8.) у операційні системи за допомогою стандартного вікна для Windows відкриття файлу.

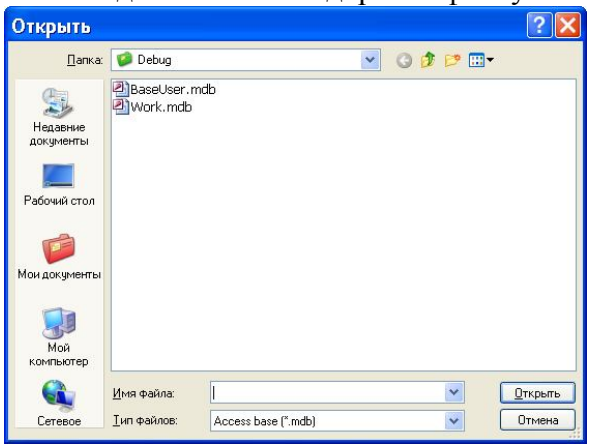

Рис. 3.8. Пошук бази даних для системи захисту інформації

У вікні **Підключення до бази даних** кнопки мають наступне призначення:

- *Тестування підключення* — перевірка можливості підключення.
- *Скасувати* — закрити вікно і не зберігати зроблені зміни.
- *Зберегти* — закрити вікно і зберегти зміни.

Налаштування підключення до БД зберігаються у файлі ***sql.ini***. Для налаштування підключення до електронної пошти із якої буде відправлятися лист потрібно виконати команду **Налаштування → Електронний ящик**. Після цього буде

відображено (рис. 3.9) вікно **Підключення до електронного ящика**.

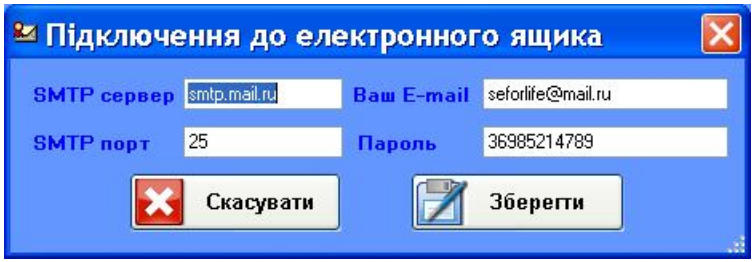

Рис. 3.9. Параметри підключення до БД

Створення ключів відкритих ключів та закритого проводиться за допомогою їх створення на основі правили алгоритму RSA. Для створення ключів потрібно викликати команду **Налаштування → Створення ключів.** Перед користувачем з'явиться (рис. 3.10) діалогове вікно **Створення ключів для шифру RSA.**

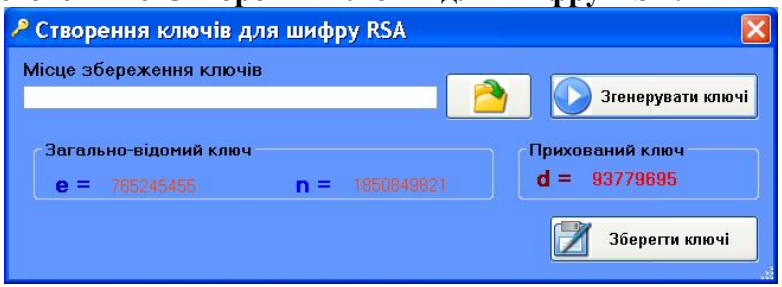

Рис. 3.10. Створення та збереження ключів

По-замовчувані програма проводить читання ключів із файлу, де було попередньо їх збережено. При натиску на кнопку ***Генерування ключів*** програма випадковим чином створює ключі і користувачеві їх відображає. При натиску на кнопку ***Зберегти ключі*** програма проводить збереження ключів і відображає відповідне повідомлення (рис. 3.11) про успішність виконання операції.

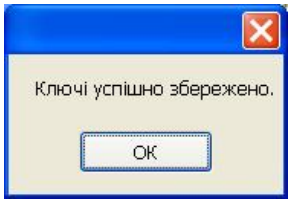

Рис. 3.11. Успішне збереження ключів

Після проведення усіх необхідних налаштувань програми можна приступати до заповнення бази даних користувачів програми. Для цього потрібно виконати команду **Кодування → Розсилка повідомлень**. Після цього (рис 3.12) відображається вікно **Користувачі системи**.

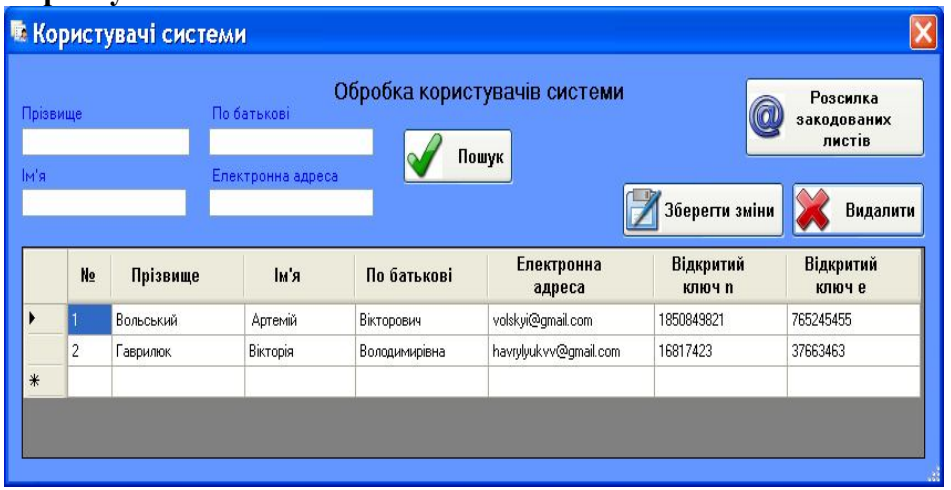

Рис. 3.12. Роботи із користувачами системи

У даному вікні є усі можливості для добавлення нових користувачів до системи, редагування користувачів та видалення потрібних користувачів. Після зроблених змін потрібно натиснути кнопку ***Зберегти зміни***. Також є можливість пошуку користувачів, якщо потрібно відібрати окремих користувачів. Для цього потрібно заповнити поля для пошуку і натиснути кнопку ***Пошук***. Якщо в параметрах пошуку нічого не вказувати, то буде відображено увесь список користувачів.

Необхідними полями для заповнення є електронна пошта користувача, відкритий ключ **n** та **e**. Після відбору користувачів є можливість відправляти їх повідомлення, при цьому необхідно натиснути кнопку ***Розсилка закодованих лисів***. Після цього (рис. 3.13) відображається вікно **Відправка листа**. На малюнку показано приклад відправки повідомлення із вкладеним ключом для кодування даних шифрами DES, ГОСТ та BlowFish. Оскільки у шифрі Blowfiish використовується ключ найбільшої

довжини. Відповідний даний ключ підходить для шифру DES і ГОСТ.

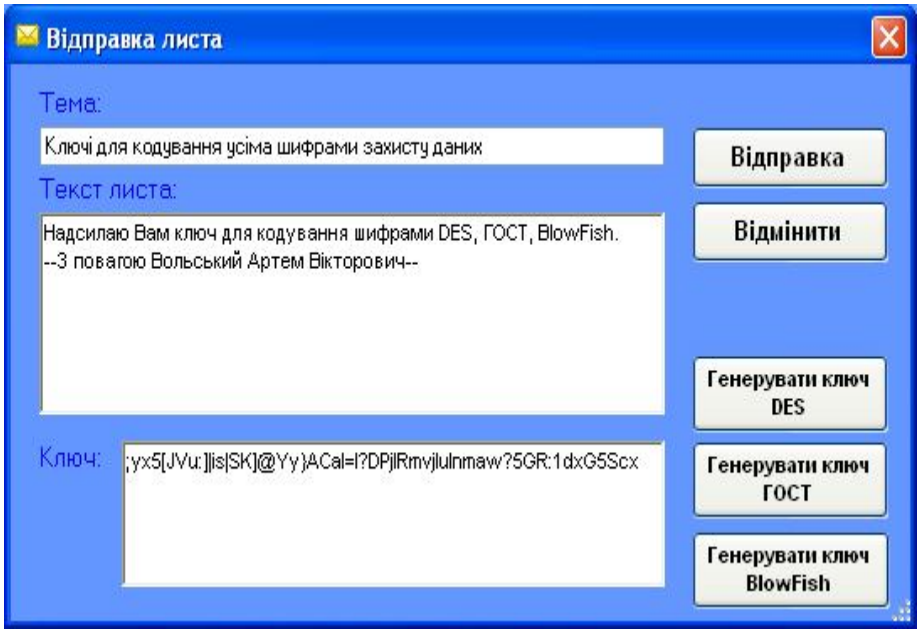

Рис. 3.13. Відправка повідомлень

Після натиску на кнопку **Відправити** лист із прикріпленим файлом ключа надсилається на електронну пошту користувачів системи. Ключ при цьому кодується відкритими ключами користувачів, що вказані для них. Користувачі отримують лист читається його вміст і проводять загрузку файлу із ключем на ПК.

Для розшифрування ключа потрібно виконати команду **Кодування → Дешифрування ключів**. При цьому відображається вікно (рис. 3.14) **Декодування повідомлень**.

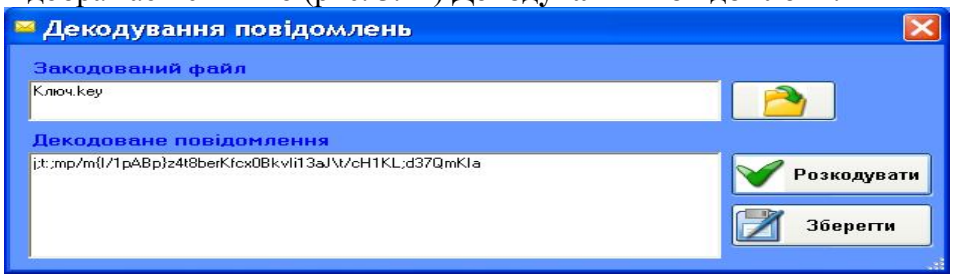

Рис. 3.14. Процес декодування повідомлень

У полі З*акодований файл* необхідно вказати файл, що потрібно розкодувати. У полі *Декодоване повідомлення* відображається результат дешифрування повідомлення. Для розшифрування повідомлення потрібно натиснути кнопку **Розкодувати**. Для збереження результатів необхідно натиснути кнопку **Зберегти**. Після цього відображається вікно (рис. 3.15) збереження результату, що є стандартним для даної операційної системи.

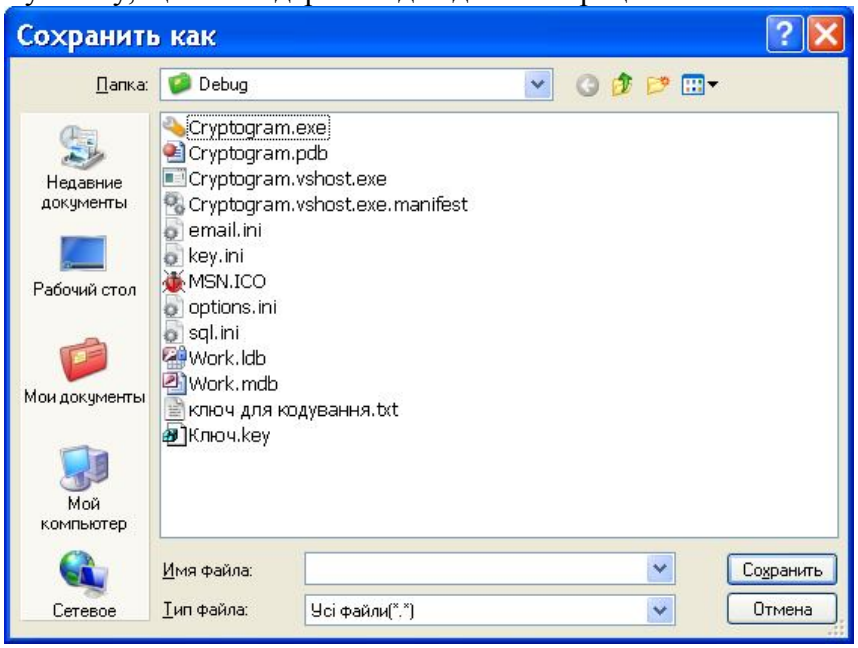

Рис. 3.15. Збереження результатів у файл

Після збереження результатів можна проводити шифрування файлів симетричними шифрами. Нами було виокремлено шифри DES, ГОСТ та BlowFish. Для шифрування інформації шифром DES потрібно виконати команду **Кодування → Des → Шифрування**. Після чого з'являється діалогове вікно (рис. 3.16) кодування інформації шифром DES.

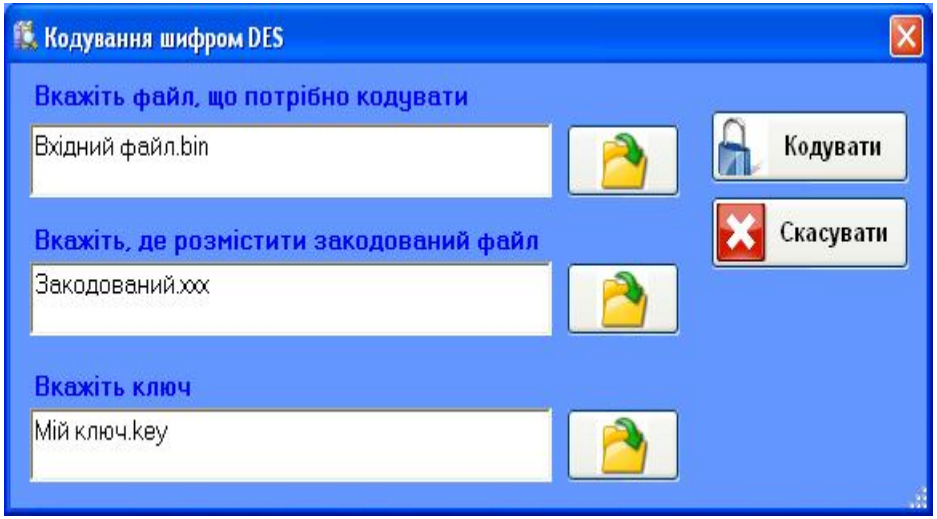

Рис. 3.16. Кодування інформації шифром DES

При вказані закодованого (рис. 3.17) файлу, можна вживати довільне ім'я і вказувати його місце розташування.

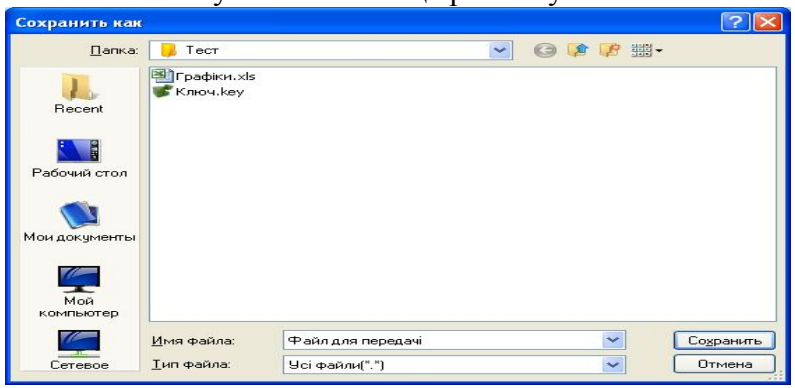

Рис. 3.17. Збереження закодованого файлу

Приклад налаштування параметрів кодування інформації шифром DES може мати вигляд наведений на рис. 3.18. Варто зазначити, що шифрування інформації можна проводити довільного типу. Розмір вхідного файлу при цьому значення не має. Швидкість кодування інформації має лінійну закономірність.

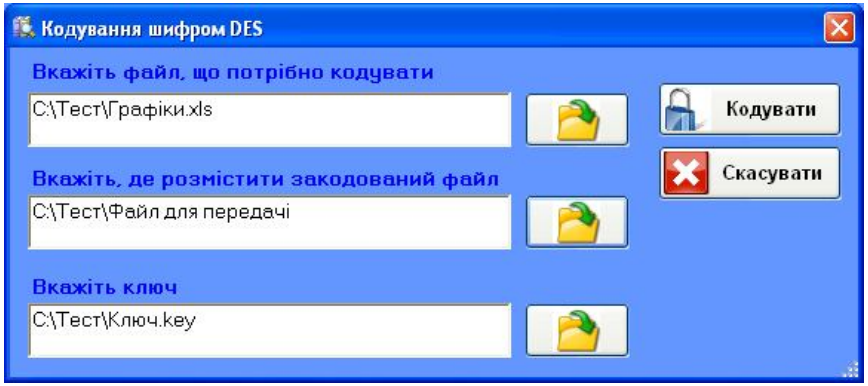

Рис. 3.18. Налаштування параметрів кодування

Після натиску на кнопку **Кодувати** виконується шифрування інформації згідно налаштувань користувача. При успішному результаті виконання операції шифрування (рис. 3.19) виводиться результат у робочу область.

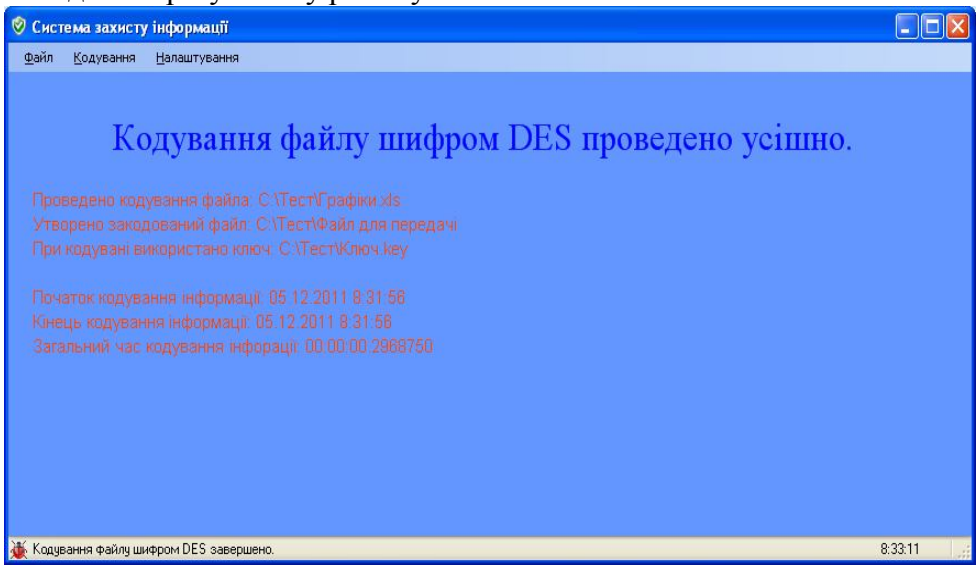

Рис. 3.19. Звіт по процесу кодування інформації шифром DES

Для виконання зворотного процесу потрібно виконати команду **Кодування → Des → Дешифрування**. Після цього відображається (рис. 3.20) діалогове вікно, де потрібно вказати параметри налаштування процесу декодування. Детально описувати процес дешифрування не будемо, оскільки параметри

налаштування є схожими до процесу шифрування. Після налаштування параметрів дешифрування потрібно натиснути кнопку **Декодувати.**

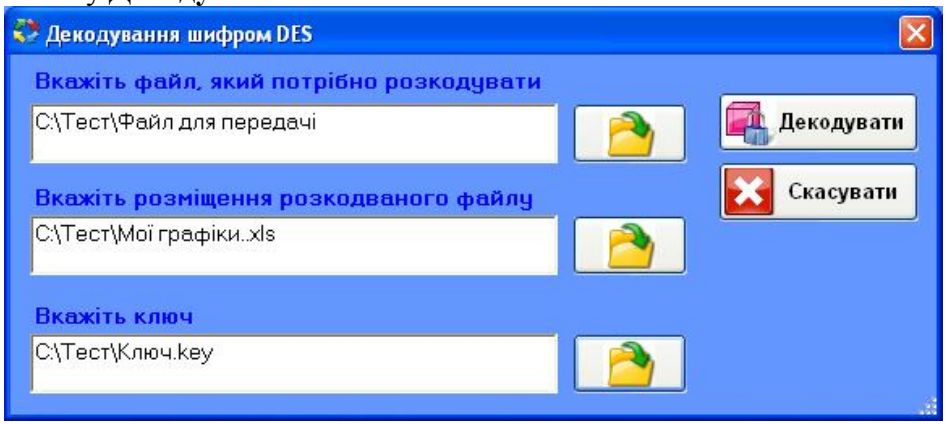

Рис. 3.20. Декодування інформації шифром DES

Після дешифрування вказаного файлу результат виводиться (рис. 3.21) у робочу область вікна.

Рис. 3.21. Результат успішного декодування інформації

Для шифрування інформації шифром ГОСТ потрібно виконати команду **Кодування → ГОСТ → Шифрування**. У результаті відображається діалогове вікно **Кодування шифром ГОСТ**. Дане вікно є аналогічним рис. 3.16, тому актуалізувати увагу на шифруванні інформації шифром ГОСТ не будемо, аналогічно виконується процес дешифрування. При цьому потрібно виконати команду **Кодування → ГОСТ → Дешифрування**.

Зазначимо лише, про шифруванні інформації шифром ГОСТ ключ повинен бути розміром не меншим 256 біт, а при шифруванні шифром DES не меншим 64 біт.

Для шифрування BlowFish ключ має змінну довжину, яка може досягати довжини 448 біт. Для кодування інформації шифром BlowFish потрібно виконати команду **Кодування → BlowFish → Шифрування**. При цьому відображається (рис. 3.22) вікно **Кодування шифром BlowFish**.

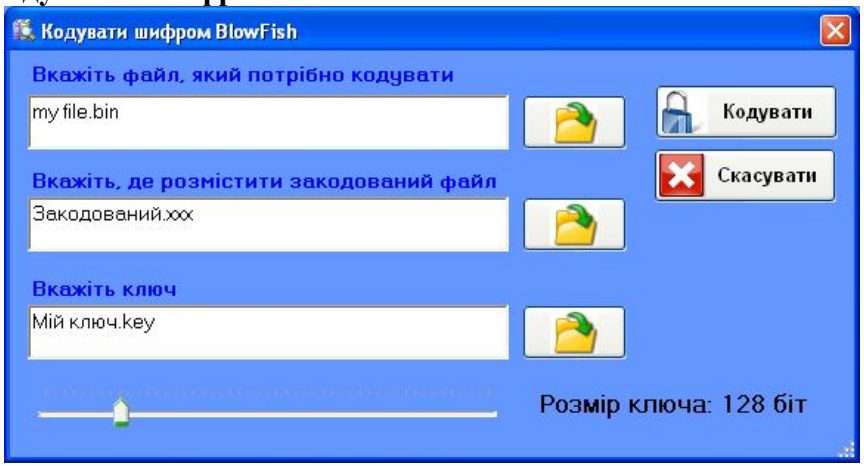

Рис. 3.22. Кодування інформації шифром BlowFish

Важливим параметром при шифруванні інформації шифром BlowFish є розмір ключа. Для зміни розміру ключа використовується повзунок у низу діалогового вікна. Розмір ключа безпосередньо впливає на надійність шифруванні інформації. Після налаштування параметрів шифрування потрібно натиснути кнопку **Кодувати**. Програма виконає шифрування вказаної інформації і результат (Рис. 3.23) буде відображено у робочій області вікна.

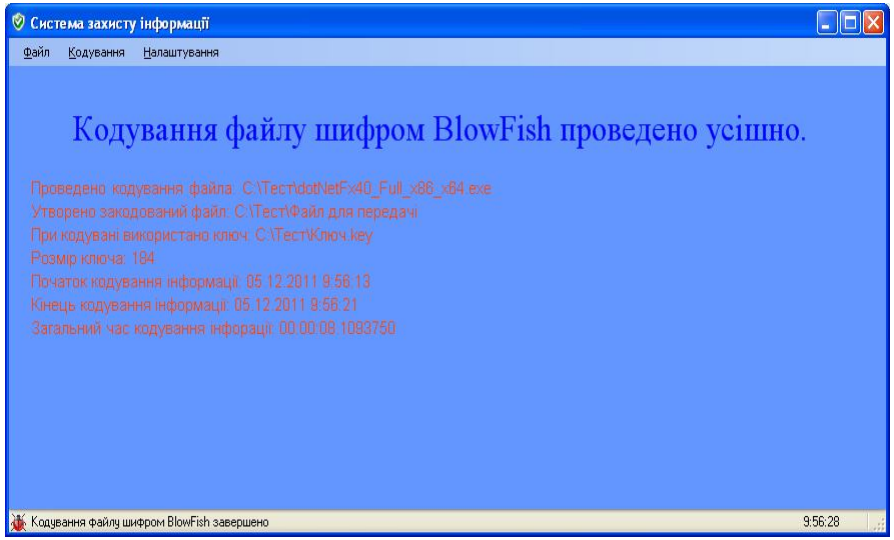

Рис. 3.23. Результат шифрування даних шифром BlowFish

У даному прикладі ми використали ключ довжиною 184 біти і провели кодування файлу розміром 48.1 Мб. Для виконання зворотного процесу потрібно виконати команду **Кодування → BlowFish → Дешифрування**. Після цього буде відображено (рис. 3.24) вікно **Декодування шифром BlowFish**.

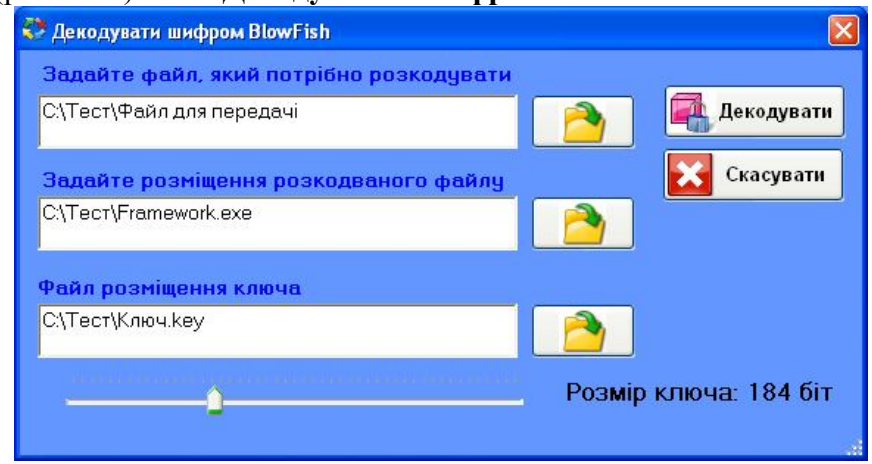

Рис. 3.24. Налаштування процесу дешифрування шифром BlowFish

Після проведення усіх необхідних налаштувань потрібно натиснути кнопку **Декодувати**. Результат буде виведено (рис. 3.25) у робочій області головного вінка.

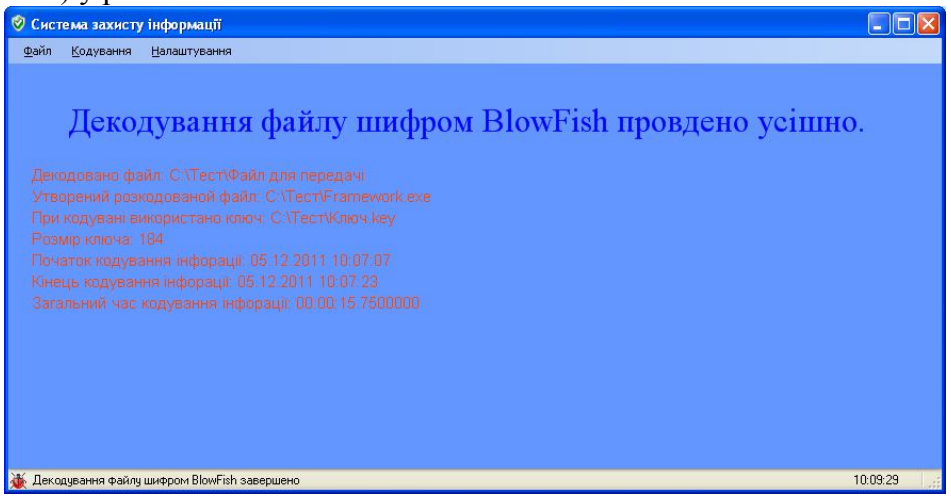

Рис. 3.25. Результат дешифрування шифром BlowFish

Таким чином нами розроблено ефективну системи захисту інформації, що представлена у вигляді додатку Windows. Програма включає в себе усі можливості для шифрування та дешифрування інформації відомими світовими стандартами захисту інформації. Програма була протестована у операційній системи Windows XP Profissional із пакетом поновлення Service Pack 3.

Висновки

У магістерській дисертації проаналізовано та досліджено криптографічні методи захисту інформації. Криптографічні методи діляться на симетричні та асиметричні. У роботі практично реалізовано як симетричні так і асиметричні алгоритми захисту інформації. Серед симетричних алгоритмів виокремлено алгоритми блочного типу. Оскільки вони є зручними у реалізації та мають великий спектр для застосу-вання. Симетричні шифри блочного типу характеризу-ються високою швидкістю роботи.

Загалом серед асиметричних алгоритмів захисту інформації основну увагу приділено шифру RSA. Даний шифр використовується у банківських системах, оскільки він дає найбільшу криптостійкість інформації, що захищається. Одними із недоліків шифру RSA є низька швидкість кодування інформації і складна апаратна реалізація.

Нами було реалізовано та простестовано наступні шифри захисту інформації:

- ✓ DES;
- ✓ ГОСТ;
- ✓ Blowfish;
- ✓ RSA.

Кожен із досліджених та реалізованих шифрів є досить надійним, що визначає його популярность у всьому світі. Проектування та реалізація даних шифрів була проведена із урахуванням їх практичного застосування у цифрових системах. Велику увагу у роботі приділено сліпому цифровому підпису. Сліпий цифровий підпис реалізовано за допомогою алгоритму RSA. Найбільш широке застосування протокол сліпих підписів знайшов у сфері цифрових грошей.

На основі розглянутих шифрів було розроблено систему захисту інформації. При виконанні операцій шифрування та дешифрування кожним із шифрів показується швидкість та ефективність їх роботи, проте із збільшенням об'єму інформації час роботи алгоритмів відповідно збільшується. Найбільш швидко працюють шифри блочного типу.

Програмна система захисту інформації включає в себе можливості відправки електронних повідомлень, кодування довільного типу інформації симетричними шифрами, організація роботи із базою даних користувачів системи.

Система розроблена на мові програмування C#. Середовищем розробки обрано Visual Studio 2010.

Таким чином, нам вдалося організувати систему захисту інформації, що дозволяє забезпечити необхідний ступінь захищеності для користувачів, які використовують дану систему. Система ефективно впроваджена у використання на

базі УКРГАЗ Банку. Головною вимогою до розробленої системи є правильність її налаштування для досягнення використання усіх її можливостей.

Основний зміст дисертації опублікований в монографії автора:

1. Українець І.В. Аналіз і дослідження криптографічних засобів захисту інформації на базі «Укргазбанк». Науковий керівник Р.М.Літнарович. МЕГУ, Рівне, 2011.- 150 с. http://elartu.tntu.edu.ua/handle/123456789/1568

Українець Ірина Василівна
спеціаліст системотехнік, магістрант інформаційних технологій

АНАЛІЗ І ДОСЛІДЖЕННЯ КРИПТОГРАФІЧНИХ ЗАСОБІВ ЗАХИСТУ ІНФОРМАЦІЇ НА БАЗІ «УКРГАЗБАНК»

8.080201 – „Інформатика"

А В Т О Р Е Ф Е Р А Т

магістерської дисертації на здобуття академічного ступеня магістра з інформатики

Комп'ютерний набір в редакторі Microsoft®Office® Word 2007 І.В.Українець
Редагування, верстка, макетування та дизайн 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