ЛАБОРАТОРНА РОБОТА № 11

ТЕМА: 1. СИСТЕМА ШИФРУВАННЯ RSA

МЕТА: НАДАТИ ПРОГРАМНУ РЕАЛІЗАЦІЮ КРИПТОСИСТЕМИ НА ОСНОВІ ПРОТОКОЛУ RSA

ТЕОРЕТИЧНІ ВІДОМОСТІ

Шифр RSA отримав назву на честь його розробників Ріверса (Ron Rivers), Шаміра (Adi Shamir) і Адлемана (Leonard Adleman).

Основні повідомлення в протоколі RSA представляються наступною діаграмою :

A↔B: N=PQ, P,Q -прості;

B: f=(P-1)(Q-1); d<f, взаємно просте з f ; cd mod f=1;

B→A: d;

A:m; A→B: $e=m^d \bmod N$

B: y; B→A: $m'=e^c \bmod N$;

Алгоритм гарантує, що m'=m.

На платформі .NET алгоритм RSA реалізується за допомогою об'єктів класу **RSACryptoServiceProvider** з простору імен **System.Security.Cryptography**. Генерація відкритого та закритого ключів здійснюється при створенні нового екземпляра класу.

Після створення нового екземпляра класу можна отримати інформацію про ключ одним із двох способів:

1. Метод **ToXMLString** – повертає інформацію про ключ в форматі XML.
2. Метод **ExportParameters** – повертає структуру **RSAParameters**, що містить ключові відомості.

Обидва методи приймають як параметр логічне значення, яке показує: **false** – слід повертати відомості тільки про відкритий ключ; **true** – слід повертати відомості і про відкритий, і про закритий ключі.

Ініціалізація класу RSACryptoServiceProvider може бути здійснена також двома шляхами:

1. Метод **FromXmlString** – використовує дані ключа з рядка XML.
2. Метод **ImportParameters** – використовує дані структури RSAParameters.

Асиметричні закриті ключі ніколи не повинні зберігатися в роздрукованому вигляді або у вигляді простого тексту на локальному комп'ютері. Якщо необхідно зберігати закритий ключ, слід використовувати для цього *контейнер ключа*.

Контейнер ключа представляє собою екземпляр класу **CspParameters** (з простору імен **System.Security.Cryptography)**. В полі **CspParameters.KeyContainerName** задається ім'я контейнера.

Порядок розшифрування за допомогою об'єктів класу **RSACryptoServiceProvider** такий:

1. Створюється контейнер для збереження ключів:

```
CspParameters cp = new CspParameters();

cp.KeyContainerName = "Key Name";
```

2. Створюється екземпляр криптопровайдера з розміщенням ключів у контейнері:

```
RSACryptoServiceProvider rsa = new RSACryptoServiceProvider(cp)
```

3. Публічний ключ експортується для передачі іншій стороні:

```
string pubKey = rsa.ToXmlString(false);

Console.WriteLine("Public Key: \n {0}", pubKey);
```

4. Після отримання байтових даних *byte[] EncryptBytes*, зашифрованих за допомогою публічного ключа, здійснюється їх розшифрування за допомогою закритого ключа:

```
byte[] DecryptBytes = rsa.Decrypt(EncryptBytes, false);

string decryptStr = Encoding.Unicode.GetString(DecryptBytes);

//string decryptStr =BitConverter.ToString(DecryptBytes);

Console.WriteLine("Decripted string: \n {0}", decryptStr);
```

Порядок шифрування полягає у такому:

1. Створюється екземпляр криптопровайдера :

```
RSACryptoServiceProvider rsa1 = new RSACryptoServiceProvider()
```

2. Імпортується публічний ключ:

```
rsa1.FromXmlString(pubKey);
```

3. Текст повідомлення перетворюється у байтову послідовність і зашифровується публічним ключем:

```
string dataToEncrypt = "Data to encrypt";

byte[] byteToEncript = Encoding.Unicode.GetBytes(dataToEncrypt);

byte[] EncryptBytes = rsa1.Encrypt(byteToEncript, false);
```

4. Зашифрована байтова послідовність відправляється стороні, яка має для розшифрування відповідний закритий ключ.

ПОРЯДОК ВИКОНАННЯ РОБОТИ

1. Ознайомитись з алгоритмом RSA.
2. Написати програму для генерації публічного і секретного ключів і їх вткористання для шифрування та розшифрування повідомлень з використанням алгоритму RSA.

3. Підготувати звіт про виконання роботи. Звіт оформлюється у вигляді документу Word з такою структурою:
    a. титульний лист,
    b. тема і мета роботи,
    c. опис алгоритму роботи програми у вигляді блок-схеми або UML- діаграм (класів, діяльності тощо),
    d. функціональні можливості програми (основні і додаткові),
    e. фрагмент програмного коду, що реалізує базову функціональність,
    f. особливості програмної реалізації окремих функцій.
4. Електронну копію звіту відправити за адресою: George@aprodos.kpi.ua.