# Лабораторна робота №7

## Тема: Шифрування з відкритим ключем на основі алгоритму RSA

## Мета: Ознайомитись з використанням криптопровайдерів .Net для побудови асиметричної криптосистеми

### Базові відомості

Шифр RSA отримав назву на честь його розробників Ріверса (Ron Rivers), Шаміра (Adi Shamir) і Адлемана (Leonard Adleman). В RSA системі використовуються наступні факти з теорії чисел:

1. Задача перевірки числа на простоту є порівняно простою.
2. Задача розкладання числа n=p*q, де p і q – прості числа, на множники є дуже складною задачею, якщо ми знаємо тільки n, а p і q – великі числа (задача факторизації).

Основні повідомлення між сторонами B і A 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: $e=m^d$ mod N |
| B: y; B→A: m'=$e^c$ mod N; |

Алгоритм гарантує, що m'=m. Пара чисел (c,N) є секретним, а (d,N) –публічним ключем сторони B.

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)**.

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

1. Створюється контейнер для зберігання ключів:

```
CspParameters cp = new CspParameters();
```

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);

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.

## Хід виконання роботи

1. Відшукайте в Інтернет-ресурсах чисельний приклад з використання алгоритму RSA (наприклад, в Вікіпедії ) та опрацюйте його.
2. Розробіть інтерфейс криптографічної системи для шифрування з використанням RSA, передбачивши окремий діалог для формування відкритого ключа.
3. Розробіть методи, які б забезпечували:
    a. Генерацію пари «відкритий –закритий» ключі.
    b. Шифрування з використанням відкритого ключа.

c. Розшифрування з використанням закритого ключа.

Додаткові завдання

1. Ознайомтесь з можливостями онлайн калькулятор для розкладання числа на прості множники і скористайтесь ним для проведення атаки на шифр RSA. Оцініть область значень параметрів шифру RSA, за яких така атака є реальною.