**Практическое занятие №7**

**Тема «Криптографическая защита информации»**

**Цель: изучить и закрепить умение реализации ЭЦП на примере RSA.**

**Теоретические сведения**

**Реализация элементов ЭЦП RSA**

Протоколы ЭЦП с одной стороны относят к протоколам аутентификации, т.к. гарантируют, что сообщение поступило от достоверного отправителя, а с другой стороны к протоколам контроля целостности, т.к. гарантируют, что сообщение пришло в неискаженном виде. Более того, получатель в дальнейшем может использовать ЭЦП как доказательство достоверности сообщения третьим лицам (арбитру) в том случае, если отправитель впоследствии попытается отказаться от него.

Говоря о схеме цифровой подписи, обычно имеют в виду следующую классическую ситуацию:

* отправитель знает содержание сообщения, которое он подписывает;
* получатель, зная открытый ключ проверки подписи, может проверить правильность подписи полученного сообщения в любое время без какого-либо разрешения и участия отправителя;
* безопасность схемы подписи гарантируется.

Электронная цифровая подпись – реквизит электронного документа, предназначенный для защиты данного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе (Федеральный закон "Об электронной цифровой подписи").

При создании цифровой подписи по классической схеме отправитель:

* применяет к исходному сообщению T хеш-функцию h(T) и получает хеш-образ r сообщения;
* вычисляет цифровую подпись **s** по хеш-образу r с использованием своего закрытого ключа;
* посылает сообщение T вместе с цифровой подписью s получателю.

Получатель, отделив цифровую подпись от сообщения, выполняет следующие действия:

* применяет к полученному сообщению T хеш-функцию h(T)и получает хеш-образ r сообщения;
* расшифровывает хеш-образ r’ из цифровой подписи s с использованием открытого ключа отправителя;
* проверяет соответствие хеш-образов r и r’ и если они совпадают, то отправитель действительно является тем, за кого себя выдает, и сообщение при передаче не подверглось искажению.

Как видно из этой схемы, порядок использования ключей обратный тому, который используется при передаче секретных сообщений. Вначале отправитель использует свой закрытый ключ, а затем получатель применяет открытый ключ отправителя.

**Разновидности ЭЦП**

Кроме классической схемы ЭЦП различают еще несколько специальных:

* схема "конфиденциальной" (неотвергаемой) подписи – подпись не может быть проверена без участия сгенерировавшего ее лица;
* схема подписи "вслепую" ("затемненной" подписи) - отправитель не знает подписанного им сообщения;
* схема "мультиподписи" - вместо одного отправителя сообщение подписывает группа из нескольких участников;
* схема "групповой" подписи - получатель может проверить, что подписанное сообщение пришло от члена некоторой группы отправителей, но не знает, кем именно из членов группы оно подписано. В тоже время, в случае необходимости, отправитель может быть определен;
* и др.

**Этапы генерации и проверки ЭЦП**

Этап 1. Выработка ключей (выполняет отправитель А) - см. практическую работу 6  "Шифрование методом RSA".

Этап 2. Отправка сообщения и электронной подписи (выполняет отправитель А).

Отправка сообщения и ЭЦП на базе алгоритма RSA

Таблица 7.1 – Ключи для алгоритма RSA



Этап 3. Получение сообщения и проверка электронной подписи (выполняет получатель B).

Получение сообщения и проверка ЭЦП на базе алгоритма RSA

Таблица 7.2 – Ключи для алгоритма RSA



**Задание для выполнения**

На базе алгоритма RSA получить ЭЦП. Удостовериться, что ЭЦП принадлежит именно этому сообщению.

Ключи для алгоритма RSA используем исходя из практической работы 6.

|  |  |  |  |
| --- | --- | --- | --- |
| # | e | n | d |
| 1 | 13 | 1073 | 853 |
| 2 | 7 | 533 | 343 |
| 3 | 17 | 391 | 145 |
| 4 | 11 | 1121 | 95 |
| 5 | 17 | 497 | 173 |
| 6 | 17 | 989 | 761 |
| 7 | 5 | 559 | 101 |
| 8 | 13 | 403 | 277 |
| 9 | 17 | 493 | 369 |
| 10 | 11 | 779 | 131 |

Таблица 7.3 – Ключи для алгоритма RSA

|  |  |  |  |
| --- | --- | --- | --- |
| # | e | n | d |
| 3 | 17 | 391 | 145 |

Таким образом, открытый ключ равен паре {17, 391}, закрытый ключ равен паре {145, 391}.

Используя исходный код на C#, зашифруем электронную подпись и проверим ее.

Приложение 7.1 – Код рабочей программы на С#

namespace Lab\_7

{

public class Signature

{

public long e;

public long n;

public long d;

public BigInteger h;

public BigInteger h\_2;

public BigInteger h\_3;

public BigInteger s;

public string input;

public Signature()

{

Console.WriteLine("Введите е ");

e = Convert.ToInt64(Console.ReadLine());

Console.WriteLine("Введите n ");

n = Convert.ToInt64(Console.ReadLine());

Console.WriteLine("Введите d ");

d = Convert.ToInt64(Console.ReadLine());

Console.WriteLine("Введите текст для отправки ");

input = Console.ReadLine();

h = GetHashCode\_ForInput(input);

int res = 0;

for (int i = 0; i < input.Length; i++)

{

res += input[i];

}

Console.WriteLine(res);

Console.WriteLine(h);

s = GetHashImage(h, d, n);

h\_2 = GetHashImage\_Reciver(input);

Console.WriteLine(h == h\_2);

h\_3 = GetHashImage\_ReciverSigniture(s, e, n);

Console.WriteLine(h\_2 == h\_3);

}

static public BigInteger GetHashImage(BigInteger h, long d, long n)

{

BigInteger \_n = new BigInteger((int)n);

BigInteger s = BigInteger.Pow(h, (int)d) % \_n;

Console.WriteLine($"s: {s}");

return s;

}

public BigInteger GetHashImage\_Reciver(string input)

{

BigInteger h\_2 = new BigInteger(GetHashCode\_ForInput(input));

Console.WriteLine($"h\_2: {h\_2}");

return h\_2;

}

public BigInteger GetHashImage\_ReciverSigniture(BigInteger s, long e, long n)

{

BigInteger \_n = new BigInteger((int)n);

BigInteger h\_3 = BigInteger.Pow(s, (int)e) % \_n;

Console.WriteLine($"h\_3: {h\_3}");

return h\_3;

}

public int GetHashCode\_ForInput(string input)

{

int res = 0;

for (int i = 0; i < input.Length; i++)

{

res += input[i];

}

return res;

}

}

class Program

{

static void Main(string[] args)

{

Signature a = new Signature();

}

}

}

**Контрольные вопросы**

1. Дайте определение понятию "электронная цифровая подпись".

Электронная цифровая подпись – реквизит электронного документа, предназначенный для защиты данного документа от подделки, полученный в результате криптографического преобразования информации с использованием закрытого ключа ЭЦП и позволяющий идентифицировать владельца сертификата ключа подписи, а также установить отсутствие искажения информации в электронном документе (Федеральный закон "Об электронной цифровой подписи").

1. Опишите последовательность действий участников протокола при отправке и проверке ЭЦП.

При создании цифровой подписи по классической схеме отправитель:

* применяет к исходному сообщению T хеш-функцию h(T) и получает хеш-образ r сообщения;
* вычисляет цифровую подпись s по хеш-образу r с использованием своего закрытого ключа;
* посылает сообщение T вместе с цифровой подписью s получателю.

Получатель, отделив цифровую подпись от сообщения, выполняет следующие действия:

* применяет к полученному сообщению T хеш-функцию h(T)и получает хеш-образ r сообщения;
* расшифровывает хеш-образ r’ из цифровой подписи s с использованием открытого ключа отправителя;
* проверяет соответствие хеш-образов r и r’ и если они совпадают, то отправитель действительно является тем, за кого себя выдает, и сообщение при передаче не подверглось искажению.

1. Какой порядок использования ключей (открытый; закрытый) при отправке и проверке ЭЦП?

Закрытый, открытый.

1. Опишите схему протокола ЭЦП на основе алгоритма RSA.

Пусть открытый текст – m. Используя секретный ключ {d, n} создадим цифровую подпись s по формуле:



1. Перечислите специальные схемы ЭЦП.

Кроме классической схемы ЭЦП различают еще несколько специальных:

* схема "конфиденциальной" (неотвергаемой) подписи – подпись не может быть проверена без участия сгенерировавшего ее лица;
* схема подписи "вслепую" ("затемненной" подписи) - отправитель не знает подписанного им сообщения;
* схема "мультиподписи" - вместо одного отправителя сообщение подписывает группа из нескольких участников;
* схема "групповой" подписи - получатель может проверить, что подписанное сообщение пришло от члена некоторой группы отправителей, но не знает, кем именно из членов группы оно подписано. В тоже время, в случае необходимости, отправитель может быть определен;
* и др.

**Вывод:** Был изучен способ **реализации ЭЦП на примере RSA.**