Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Исследование алгоритмов генерации и верификации**

**электронной цифровой подписи**

Студент: Каспер Н.В.

ФИТ 3 курс 5 группа

Преподаватель: Берников В.О.

Минск 2020

1. **Описание приложения**

Приложение написано на языке программирования C# и позволяет:

* генерировать ЭЦП на основании алгоритма RSA;
* генерировать ЭЦП на основании алгоритма Эль-Гамаля;
* генерировать ЭЦП на основании алгоритма Шнорра;
* оценивать время выполнения генерации ЭЦП.

1. **Методика выполнения поставленных задач**

ЭЦП – контрольная характеристика сообщения, которая вырабатывается с использованием личного ключа, проверяется с использованием открытого ключа, служит для контроля целостности и подлинности сообщения и обеспечивает невозможность отказа от авторства.

Таким образом, ЭЦП выполняет те же функции, что и собственноручная (поставленная «от руки») подпись:

* аутентифицировать лицо, подписавшее сообщение;
* контролировать целостность подписанного сообщения;
* защищать сообщение от подделок;
* доказать авторство лица, подписавшего сообщение, если это лицо отрицает свое авторство.

Важнейшими отличительными особенностям ЭЦП являются:

* ЭЦП представляет собой бинарную последовательность (в отличие от графического образа, каковым является подпись от руки);
* указанная бинарная последовательность зависит от содержания подписываемого сообщения.

Результат работы приложения представлен на рисунке 2.1.

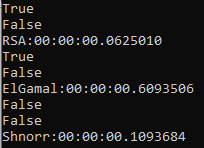


Рисунок 2.1 – Результат работы приложения

Для создания ЭЦП на основе алгоритма RSA и верификации документа, приложение высчитывает ключевую информацию, хеш текста из файла и создает подпись на основе закрытого ключа, а верификацию хеша производит на основании открытого ключа.

Результат работы ЭЦП на основе алгоритма RSA представлен на рисунке 2.2.

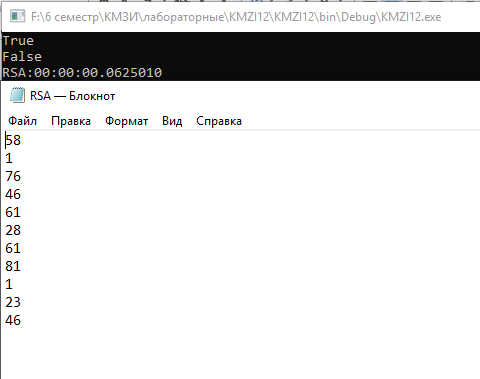


Рисунок 2.2 – Результат работы ЭЦП RSA

Если подписываемое сообщение М также должно передаваться в зашифрованном виде, то обычно М шифруется на стороне отправителя стандартным образом: с помощью открытого ключа получателя (е и n), который перед основным процессом верификации подписи расшифровывает послание своим тайным ключом: d и n.

Создание ЭЦП на основе RSA происходит в функции Create класса RSA. Реализация данной функции представлена на рисунке 2.3.

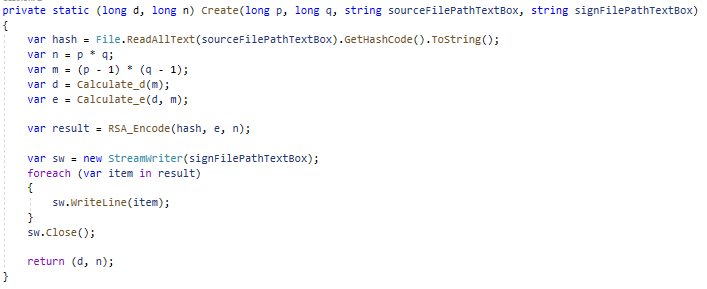


Рисунок 2.3 – Реализация функции Create

Ключевая информация отправителя: открытый ключ: y, g и р; тайный ключ: х. Чтобы подписать сообщение М, обладатель используемых для ЭЦП ключей должен выбрать случайное число k, взаимно простое с р-1.

Затем вычисляется числа а и b, являющиеся цифровой подписью (S = {a, b}) путем возвещения числа g в степень k и деления по модулю p. Для вычисления b используется расширенный алгоритм Евклида.

Создание ЭЦП на основе Эль-Гамаля происходит в функции Crypt класса ElGamal. Реализация функции представлена на рисунке 2.4.

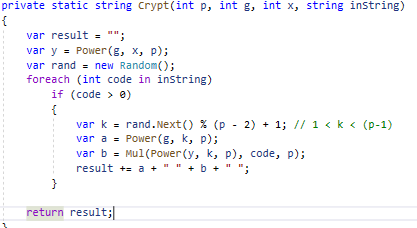


Рисунок 2.4 – Реализация функции Crypt

Создание ЭЦП на основе Шнорра происходит в функции Do класса Shnorr. Реализация функции представлена на рисуке 2.5.

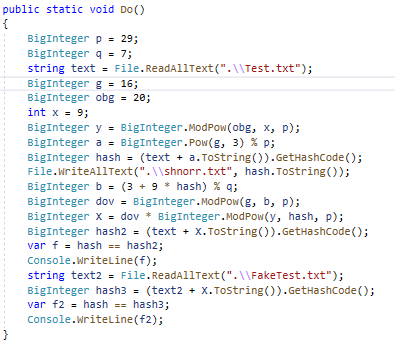


Рисунок 2.5 – Реализация функции Do

Для верификации подписи вычисляется хеш полученного сообщения, Н(М) = h. Далее нужно убедиться, что выполняется равенство ya \* ab = gh (mod p. Если равенство выполняется, подпись верифицируется.

Проверка ЭЦП на основе алгоритма RSA выполняется в функции Verify, реализация которой представлена на рисунке 2.6.

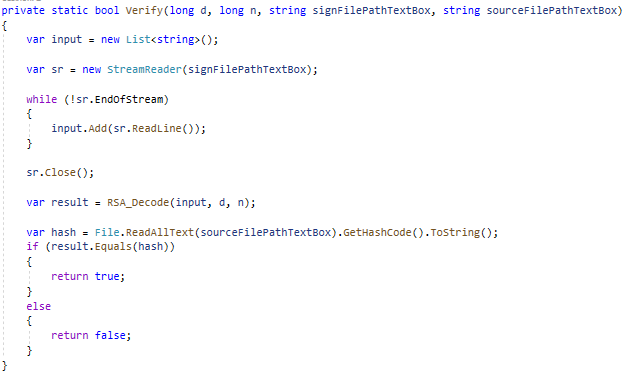


Рисунок 2.6 – Реализация функции Verify

Таким образом, были реализованы все поставленные задачи. Были исследованы асимметричные шифры.

**Вывод**

В ходе лабораторной работы были приобретены навыки подписи ЭЦП на основе алгоритмов RSA, Эль-Гамаля, Шнорра.

Также было разработано приложение, на языке программирования C#, для реализации задач, связанных подписью ЭЦП.