Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный технологический университет»

Факультет информационных технологий

Кафедра информационных систем и технологий

**Отчет к лабораторной работе №10**

**«Изучение криптографических хеш-функций»**

Выполнил:

студентка 3 курса 2 группы

Черноок Ю. С.

Проверил:

ассистент

Копыток Д. В.

Минск 2021

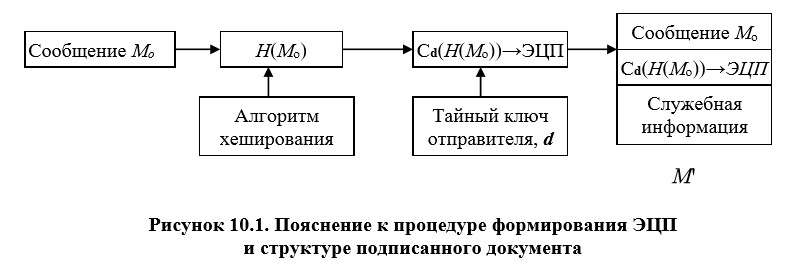
Цель: изучение алгоритмов генерации и верификации электронной цифровой подписи и приобретение практических навыков их реализации.

**Теоретическая часть**

Электронная цифровая подпись – 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.



**ЭЦП на основе RSA**

Здесь можно рассматривать две ситуации:

* сообщение Мo подписывается и передается в открытом (незашифрованном) виде;
* сообщение Мo подписывается и передается в зашифрованном.

При этом подпись S вычисляется на основе известного из лабораторной работы №8:

S = (H(Mo))dо mod no (1)

при указанном выше реверсе в отношении ключевой информации; в (1) dо и no – элементы тайного ключа отправителя. Передаваемое сообщение М' = Мo||S. Соответственно, операция расшифрования на приемной стороне (получатель анализирует Мп||S) будет производиться следующим образом:

H(Mo) = (S)ео mod no. (2)

Далее вычисляется Н(Mп). Если H(Mo) = H(Mп), подпись верифицирована. Если подписываемое сообщение М (М') также должно передаваться в зашифрованном виде, то обычно М' шифруется на стороне отправителя стандартным образом: с помощью открытого ключа получателя (еп и nп), который перед основным процессом верификации подписи расшифровывает послание своим тайным ключом: dп и nп. Далее осуществляются вычисления и анализ, как и в первом случае.

**ЭЦП Эль-Гамаля**

Ключевая информация отправителя для ЭЦП создается точно так же, как это описано в материалах к ЛР №8. Она состоит из тех же элементов, что и ключи в DSA. Основное отличие в применении расчетов состоит в том, что результатом зашифрования является только одна пара чисел, а не пара – для каждого блока исходного сообщения. Причем в рассматриваемом случае таким сообщением является хеш подписываемого документа: Н(Mо).

Итак, ключевая информация отправителя: открытый ключ: y, g и р; тайный ключ: х. Чтобы подписать сообщение Мо, обладатель используемых для ЭЦП ключей должен выбрать, как и в предыдущей схеме, случайное число k, взаимно простое с р-1. Затем вычисляется числа а и b, являющиеся цифровой подписью (S = {a, b}):

a = gkmod p; (3)

для вычисления b с помощью расширенного алгоритма Евклида решается уравнение

Н(Mо) = (xa + kb) mod (p-1). (4)

Получателю отправляется сообщение М' = Мо||S. Для верификации подписи вычисляется хеш полученного сообщения, Н(Мп) = h. Далее нужно убедиться, что выполняется равенство

ya \* ab = gh (mod p). (5)

Если равенство выполняется, подпись верифицируется.

**ЭЦП Шнорра**

Рассматриваемая схема является основой стандарта ЭЦП в Беларуси. Алгоритм ЭЦП К. Шнорра (К. Schnorr ) является вариантом алгоритма ЭЦП Эль-Гамаля.

Одной из особенностей ЭЦП Эль-Гамаля является то, что число p должно быть очень большим, чтобы сделать действительно трудной проблему дискретного логарифма. Рекомендуется длина p, по крайней мере, должна составлять 1024 бита. Чтобы уменьшить размер подписи, Шнорр предложил новую схему, но с уменьшенным размером подписи.

Ключевая информация: p – простое число в диапазоне от 512 до 1024 бит; q –160-битное простое число, делитель p-1; любое число g (g≠1) такое, что

gq=1 mod p. (6)

Числа p, g, q являются открытыми и могут применяться группой пользователей.

Выбирается число х < q (х является тайным ключом) и вычисляется последний элемент открытого ключа:

y = g-х mod p. (7)

Секретный ключ имеет длину не менее 160 бит.

Для подписи сообщения Мо выбирается случайное число k (1<k<q) и вычисляет параметр а:

а = gkmod p. (8)

Далее вычисляется хеш от канкатенации сообщения Мо и числа а: h = H(Mo||a). Обратим внимание, что хэш-функция непосредственно не применяется к сообщению. Создается хеш-образ подписываемого сообщения, спереди присоединенного к числу а. Далее вычисляется значение b:

b = (k + xh ) mod q. (9)

Получателю отправляются М' = Мо||S; S = {h, b}.

Для проверки подписи получатель вычисляет

Х = gbyh (mod p). (10)

Затем он проверяет выполнение равенства: h = Н(Mп||Х). Подпись достоверна, если равенство выполняется.

**Практическая часть**

Разработать авторское оконное приложение в соответствии с целью лабораторной работы. Приложение должно реализовывать генерацию и верификацию ЭЦП на основе алгоритмов RSA, Эль-Гамаля и Шнорра.

**ЭЦП на основе RSA**

На рисунке 1 показана функция Create(), в которую мы передаем числа р и q, пути к текстовым файлам. Производим необходимые операции нахождения n, m, d, e. Шифрование текста из файла Source.txt в RSA.txt.

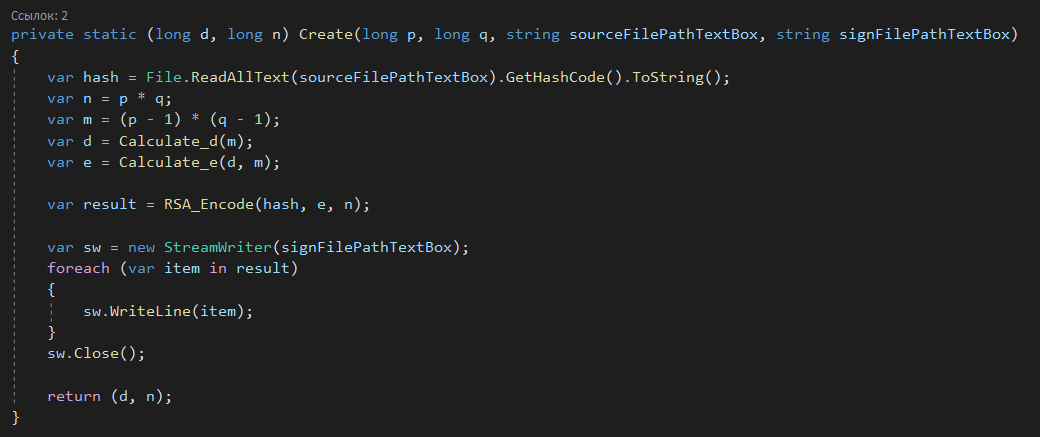


Рисунок 1 – Функция Create

Весь процесс шифрования и расшифрования показан на рисунке 2.

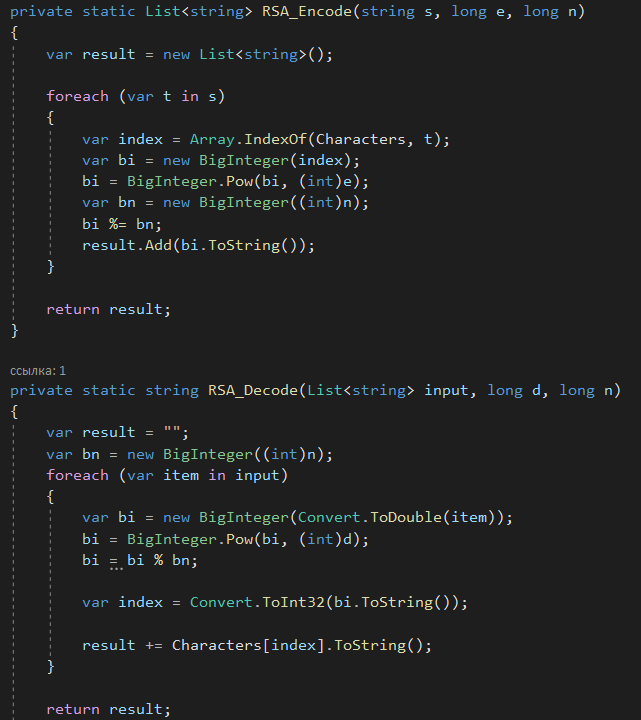


Рисунок 2 – Шифрование и дешифрование

Чтобы проверить ЭЦП действительна или нет была разработана специальная функция Verify. В ней мы передаем файлы и сравниваем содержимое файлов (их хеш) если файлы идентичны, то ЭЦП верифицирована, если нет, то программа напишет False (рисунок 3).

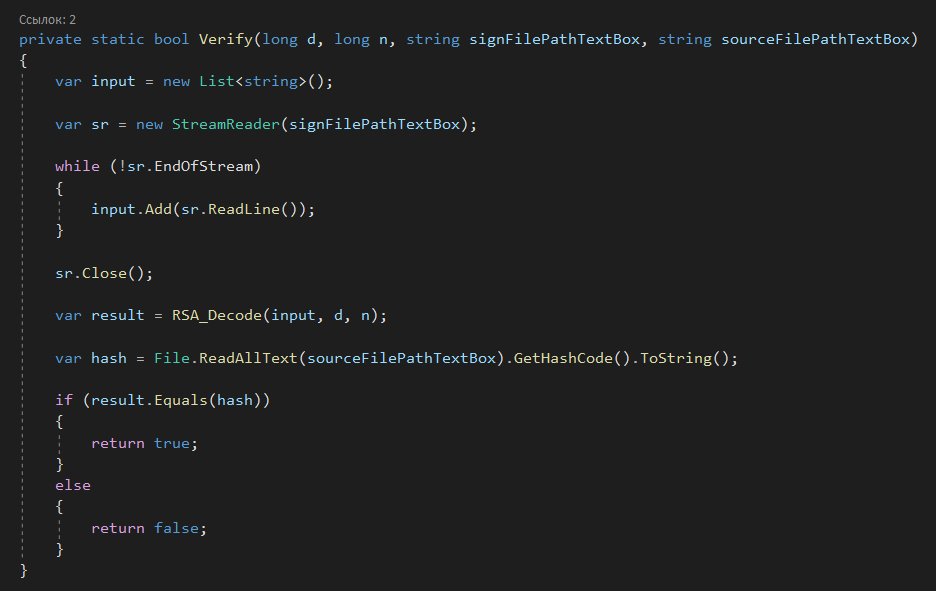


Рисунок 3 – Функция Verify

Давайте проверим работоспособность приложения. Запишем некий текст в файлы и сравним их (рисунок 4). Мы видим, что файлы различны значит программа скажет нам что ЭЦП не действительна.

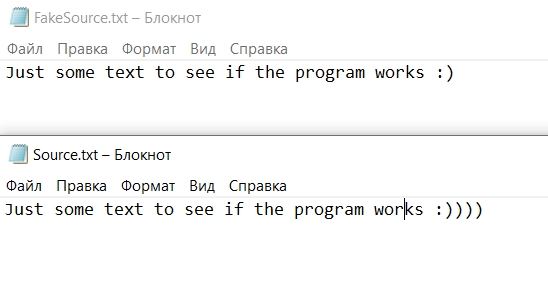


Рисунок 4 – Содержимое текстовых файлов

На рисунке 5 мы видим вывод работы программы и можем сделать вывод о том, что программа работает правильно.

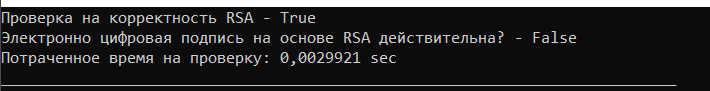


Рисунок 5 – Итог выполнения программы

**ЭЦП на основе Эль-Гамаля**

В случае этого алгоритма мы используем заранее указанные строки "Hello world". Для получения хеша строки я использовала стандартную функцию MD5. Эта функция продемонстрирована на рисунке 6.

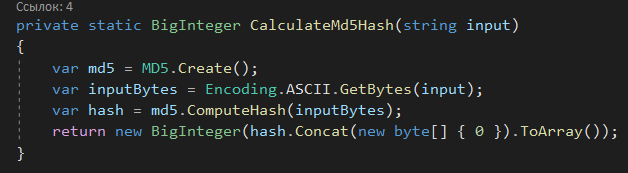


Рисунок 6. – Функция CalculateMd5Hash

Шифрование и расшифрование продемонстрировано на рисунке 7.

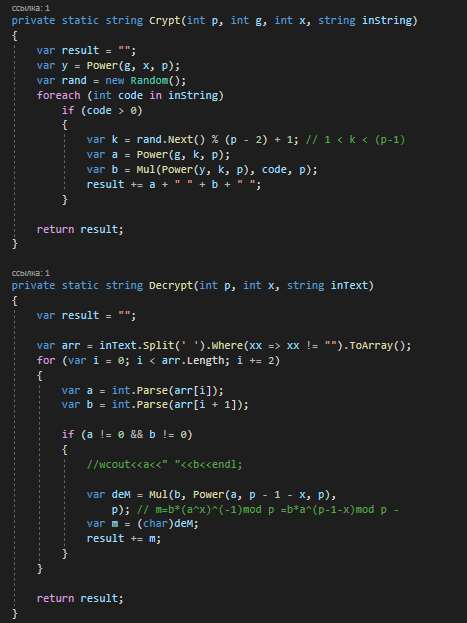


Рисунок 7 – Шифрование и дешифрование

Пример как сравниваются строки (их хеш) приведен на рисунке 8.

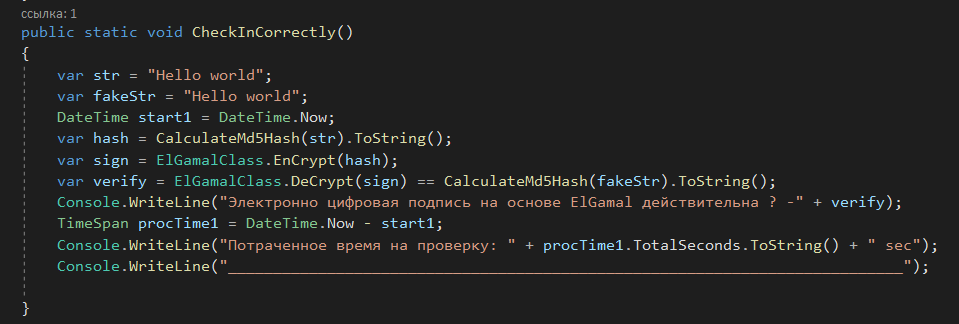


Рисунок 8 – Функция CheckInCorrectly

Мы можем заметить, что строки (из рисунка 8) абсолютно равны, а это значит, что программа вернет нам True. Итог выполнения программы продемонстрирован на рисунке 9.

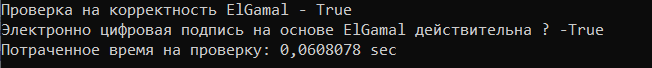


Рисунок 9 – Выполнение программы

**ЭЦП на основе Шнорра**

Для реализации алгоритма Шнорра были написаны:

Функция, которая подбирает простое число P в диапазоне от 1 до 999999, простое число Q, которое является делителем P – 1. Также от 1 до P – 1 выбирается случайное число H (для полинома).

Число G равно H^(P – 1 / G) modP. Это один из открытых ключей.

T – случайное число в диапазоне от Q – 1 до 100000 – сообщение.

Следующая функция позволяет задать секретный ключ S (можно сгенерировать случайный) от 1 до Q – 1.

Число A является публичным ключом и находится по формуле A = Ginv^SmodP, где Ginv–число, обратное числу G по модулю P.

Следующая функция позволяет задать случайное число R, необходимое для разового шифрования (с ним будет происходить конкатенация), а также позволяет рассчитать тайный ключ X, который находится по формуле G^RmodP.

Большое значение E – это хеш, полученный при помощи конкатенации исходного сообщения (число T) и тайного ключа X.

Далее вычисляется значение Y, которое можно найти по формуле: Y = (R(случайное число, участвующее в генерации X) + E(хеш) \* X(тайный ключ)) modP.

Получателю отправляется сообщение, являющееся конкатенацией входного сообщения Tи пары (E(хеш), Y).

Для того, чтобы узнать тайный ключ, нужно подставить соответствующие элементы в формулу:

Z = (G^Y \* A^E) modP.

Если Z==X, то ЭЦП подлинна и сообщение не изменялось (рисунок 10).

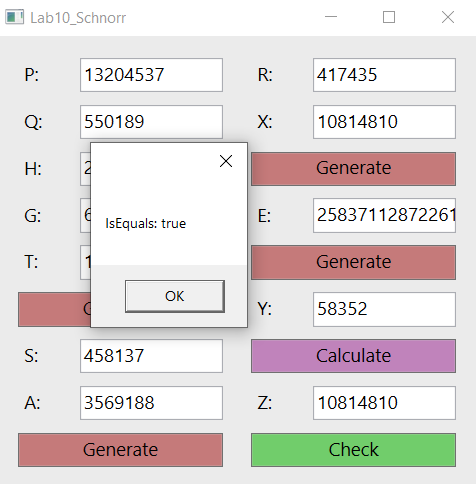


Рисунок 10 – Проверка ЭЦП

Вывод: я изучила алгоритмы генерации и верификации электронной цифровой подписи и приобрела практические навыки их реализации.

