МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Лабораторная работа № 12**

**ИССЛЕДОВАНИЕ АЛГОРИТМОВ ГЕНЕРАЦИИ И ВЕРИФИКАЦИИ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ**

Разработала: Некрасова А.П.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М.Г.

Минск 2023

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

**Задачи:**

1. Закрепить теоретические знания по алгебраическому описанию и алгоритмам реализации операций генерации и верификации электронной цифровой подписи (ЭЦП).
2. Получить навыки практической реализации методов генерации и верификации ЭЦП на основе хеширования подписываемых сообщений и алгоритмов RSA, Эль-Гамаля и Шнорра, а также DSA.
3. Разработать приложение для реализации заданных алгоритмов генерации и верификации ЭЦП.
4. Оценить скорость генерации и верификации ЭЦП.
5. Результаты выполнения лабораторной работы оформить в виде описания разработанного приложения, методики выполнения экспериментов с использованием приложения и результатов эксперимента.
6. **Теоретические сведения**

Электронная цифровая подпись (ЭЦП) является важным элементом современных информационных систем, использующих методы и технологии криптографического преобразования информации.

Понятие «электронная цифровая подпись» было введено в 1976 году У. Диффи и М. Хеллманом.

После создания RSA разработаны алгоритмы цифровой подписи И. Рабина и Р. Меркле. В 1984 году Ш. Гольдвассер, С. Микали и Р. Ривест сформулировали требования безопасности к алгоритмам ЭЦП, описали атаки на ЭЦП.

*Определение 1.* 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.

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.

Как следует из опр. 1, основным компонентом в технологии ЭЦП является ключ. Принадлежность ключа, в предположении, что он известен только законным пользователям, позволяет решать все «возложенные на ЭЦП», сформированную на основе этого ключа, задачи. В соответствии с этим обстоятельством перечисленные выше функции ЭЦП могут быть реализованы на основе классических методов зашифрования/расшифрования:

* на основе симметричных систем (с тайным ключом),
* на основе симметричных систем и посредника,
* на основе асимметричных систем (с открытым ключом).

Первый из перечисленных методов ничем не отличается, например, от DES.

Во втором случае создаются, по существу, две симметричные системы: между отправителем и посредником и между посредником и получателем. Причем посредник выдает двум сторонам различный тайный (для иных субъектов системы) ключ.

Во последнем случае сообщение, отправляемое получателю, шифруется тайным ключом отправителя. Отправитель же верифицирует подпись (в данном случае – устанавливает авторство, используя для расшифрования публичный ключ отправителя, и получает гарантию в защищенности переданного сообщения от подделок, если после расшифрования формат и содержание документа имеют логическую стройность) с помощью открытого ключа отправителя.

Таким образом, в этом случае, как и в первых двух случаях, ЭЦП, как отдельный, самостоятельный, присоединенный к исходному документу элемент получаемого сообщения, отсутствует. Кроме, того, в отличие от классической асимметричной криптографии, где используется ключевая информация получателя, в нашем случае используется ключевая информация отправителя: открытый ключ – для зашифрования, тайный – для расшифрования.

С учетом изложенного можем сформулировать определение ЭЦП в несколько ином виде.

*Определение 2.* Электронная цифровая подпись – бинарная (или в ином виде) последовательность символов, являющаяся реквизитом электронного документа, зависящая от содержания этого документа и предназначенная для подтверждения целостности и подлинности электронного документа.

**ЭЦП на основе алгоритма RSA:**

Здесь можно рассматривать две ситуации:

* сообщение *М0* подписывается и передается в открытом (незашифрованном) виде;
* сообщение *М0* подписывается и передается в зашифрованном.

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

*S* = (*H*(*M0*))*d0* *mod* *n0* (1.1)

при указанном выше реверсе в отношении ключевой информации; в (1.1) *d0* и *n0* – элементы тайного ключа отправителя. Передаваемое сообщение *М'* = *М0||S*.

Соответственно, операция расшифрования на приемной стороне (получатель анализирует *Мn||S*) будет производиться в соответствии с формулой с известной модификацией ключей:

*H*(*M0*) = (*S*)*е0* *mod n0* (1.2)

Далее вычисляется *Н*(*Mn*). Если *H*(*M0*) = *H*(*Mn*), 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 (*еn* и *nn*), который перед основным процессом верификации подписи расшифровывает послание своим тайным ключом: *dn* и *nn*. Далее осуществляются вычисления и анализ, как и в первом случае.

**ЭЦП на основе алгоритма 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*(*M0*).

Итак, ключевая информация отправителя: открытый ключ: *y*, *g* и *р*; тайный ключ: *х*. Чтобы подписать сообщение *М0*, обладатель используемых для ЭЦП ключей должен выбрать, как и в предыдущей схеме, случайное число *k*, взаимно простое с *р*-1. Затем вычисляются числа *а* и *b*, являющиеся цифровой подписью (*S* = {*a, b*}):

*a* = *gk mod p* (1.3)

для вычисления *b* с помощью расширенного алгоритма Евклида решается уравнение:

*Н*(*M0*) = (*xa* + *kb*) *mod* (*p*-1) (1.4)

Получателю отправляется сообщение *М'* = *М0||S*.

Для верификации подписи вычисляется хеш полученного сообщения, *Н*(*Мn*) = *h*. Далее нужно убедиться, что выполняется равенство:

*ya* × *ab* = *gh* (*mod p*) (1.5)

Если равенство выполняется, подпись верифицируется.

**ЭЦП на основе алгоритма 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* (1.6)

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

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

*y* = *g-х* *mod p* (1.7)

Секретный ключ имеет длину не менее 160 бит. Для подписи сообщения М0 выбирается случайное число k (1 < *k* < *q*) и вычисляется параметр *а*:

*а* = *gk* *mod p* (1.8)

Далее вычисляется хеш от конкатенации сообщения *М0* и число *а*: *h* = *H*(*M0||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* (1.9)

Получателю отправляются *М'* = *М0||S*; *S* = *{h, b}*.

Для проверки подписи получатель вычисляет:

*Х* = *gb × yh* *mod p* (1.10)

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

Основные вычисления для генерации подписи могут производиться предварительно. Порядок величин *х* и *h* – около 140 двоичных разрядов, порядок числа *k* – около 70-72 разрядов. С учетом этого сложность операций умножения можно считать ничтожно малой по сравнению с модульным умножением в схеме RSA.

1. **Практическая часть**

В данной лабораторной работе необходимо разработать пользовательское приложение, которое должно реализовывать генерацию и верификацию ЭЦП на основе алгоритмов RSA, Эль-Гамаля и Шнорра.

При выполнении лабораторной работы были использованы алгоритмы шифрования, разработанные в предыдущих лабораторных работах (RSA и Эль-Гамаля). Для генерации и верификации ЭЦП на основе алгоритма **RSA** была написана функция, которая хеширует введенный текст при помощи алгоритма MD-5, после чего полученный хеш зашифровывается алгоритмом RSA.



Рисунок 2.1 – Верификация сообщения при помощи RSA

В итоге получаем зашифрованное сообщение (последовательность чисел). ЭЦП в данном случае служит хеш, полученный на предыдущем этапе.

Для верификации сообщения необходимо расшифровать полученный шифротекст. В результате должен получится хеш исходного сообщения. Если они равны – сообщение не изменялось.

Для генерации и верификации ЭЦП на основе алгоритма **Эль-Гамаля** была разработана функция, которая хеширует сообщение при помощи алгоритма MD-5, после чего шифрует сообщение алгоритмом Эль-Гамаля. По итогу функция возвращает числовые пары и ЭЦП – полученный хеш. Аналогично с предыдущим заданием, если хеш расшифрованного сообщения совпадает с хешем на предыдущем этапе, то ЭЦП верна.

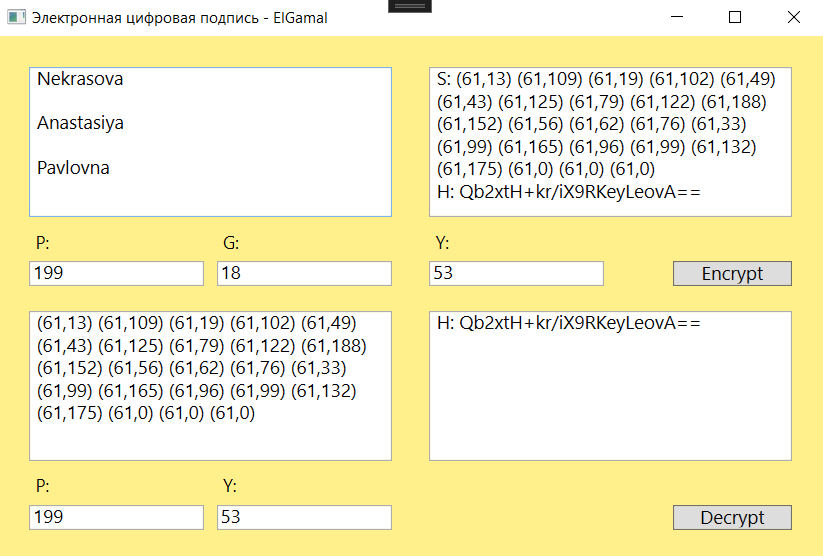
****

Рисунок 2.3 – Верификация сообщения с помощью алгоритма Эль-Гамаля

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*, любое число *g* (*g* ≠ 1) такое, что *gq* ≡ 1 mod *p*. Это так же один из наших открытых ключей.

*T* – случайное число в диапазоне от *q*- 1 до 100000 – наше сообщение.

Следующая функция позволяет задать секретный ключ x и вычисляет последний элемент открытого ключа: *y* ≡ *g–х* mod *p*.

Следующая функция позволяет задать случайное число *k* (1<*k*<*q*) и рассчитывает тайный ключ *a*, который находится по формуле *gk* mod *p*.

Большое значение *h* – это хеш, полученный при помощи конкатенации нашего исходного сообщения (число *T*) и тайного ключа *a*.

Далее вычисляется значение *b*, которое можно найти по формуле: *b* = (*k*(случайное число, участвующее в генерации) + *h* \* *x*(тайный ключ)) mod *q*.

Получателю отправляется сообщение, являющееся конкатенацией входного сообщения *T* и пары (*h*, *b*).

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

*Z* = (*gb* \* *yh*) mod *P*.

Если *Z==h*, то ЭЦП подлинна и сообщение не изменялось.

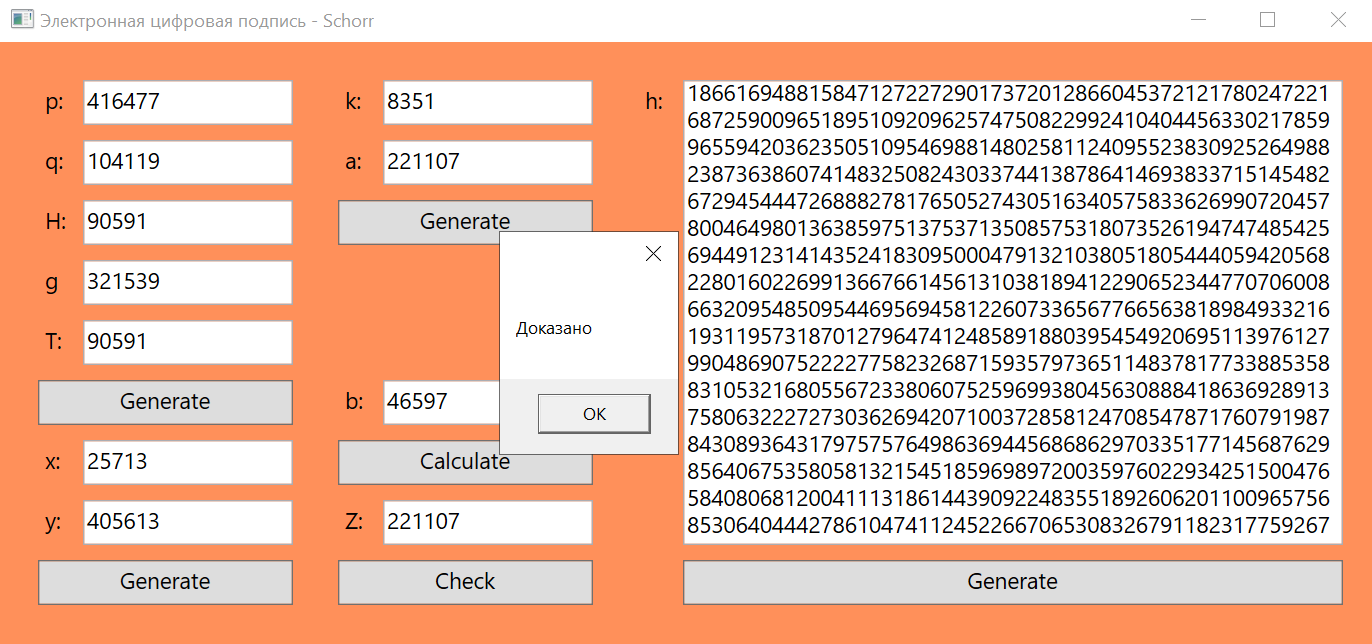
****

Рисунок 2.5 – Проверка ЭЦП (Шнорр)

График времени, затраченного на генерацию/верификацию сообщения алгоритмами RSA и Эль-Гамаля представлены на рис. 2.6. Необходимо отметить, что здесь отсутствую тайминги при использовании алгоритма Шнорра, потому что они несоизмеримо велики в сравнении с остальными алгоритмами.

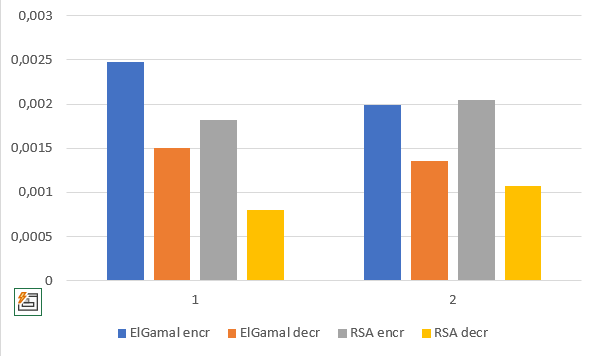


Рис. 2.6 – Графики затраченного времени

**Вывод:** в данной лабораторной работе, были разобраны принципы алгоритмов шифрования RSA и Эль-Гамаль, а также алгоритм Шнорра. Закреплены теоретические знания по алгебраическому описанию и алгоритмам реализации операций генерации и верификации электронной цифровой подписи (ЭЦП). И как результат лабораторной работы: разработано приложение для реализации заданных алгоритмов генерации и верификации ЭЦП.