Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Исследование алгоритмов генерации и верификации

электронной цифровой подписи

Студент: Шулаков А.А.

ФИТ 3 курс 5 группа

Преподаватель: Савельева М.Г.

Минск 2023

## 1. Описание приложения

Приложение написано на языке программирования C# и реализиует следующие операции:

* генерация и верификация ЭЦП на основе алгоритмов RSA, Эль-Гамаля и Шнорра;
* оценку времени выполнения указанных процедур при реальных ключевых параметрах.

## 2. Методика выполнения поставленных задач

## 2.1. Алгоритм RSA

Алгоритм генерации подписи заключается в следующих операциях:

* выбор простых чисел *p, q*;
* выбор случайного числа e, взаимно простого с функцией Эйлера *ȹ*(*n*)=(*p*-1)(*q*-1);
* нахождение числа *d*, такого что *ed* = 1 *mod* (*p*-1)(*q*-1);
* вычисление хеш-образа сообщения *h=H(M)*;
* вычисление ЭЦП: *S = hd mod n*.

Далее полученный открытый ключ {*e, n*} и письмо с ЭЦП {*m, sign*} будут отправлены получателю. Программная реализация генерации ЭЦП представлена на рисунке 2.1.

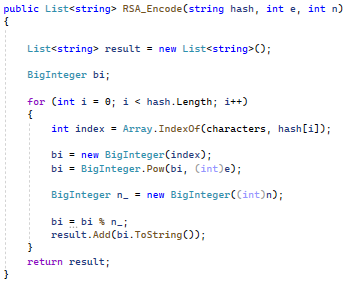


Рис. 2.1 – Генерация ЭЦП

Для того, чтобы получатель смог провести операцию верификации сообщения, необходимо выполнить следующие действия:

* вычислить *h = Se mod n*;
* сравнить значение выше с полученным *h=H(M*);

Если полученные значения совпали, подпись верифицирована. Программная реализация генерации хеша на стороне получателя для дальнейшего сравнения с полученным хешем представлена на рисунке 2.2.

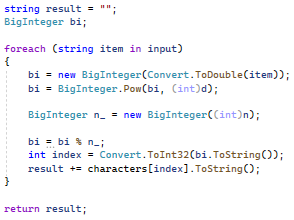


Рис. 2.2 – Генерация хеша получателя

Запустим наше приложение. Результат работы представлен на рисунке 2.3. На консоль выводятся все значения, которые были заданы изначально или вычислены в ходе работы программы. Далее был автоматически открыт входной файл, текст которого считывался и хешировался. При нажатии любой клавиши на консоли на экран выводится результат верификации. В первом случае файл остался прежним, верификация прошла. Далее мы модифицировали его с целью проверить правильно работы и на консоль вывелось сообщение о том, что файл не подлинный.

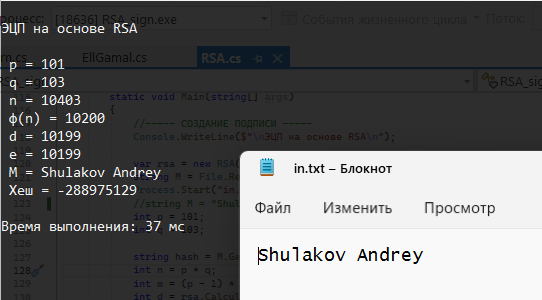


Рис. 2.3 – Результат работы алгоритма RSA

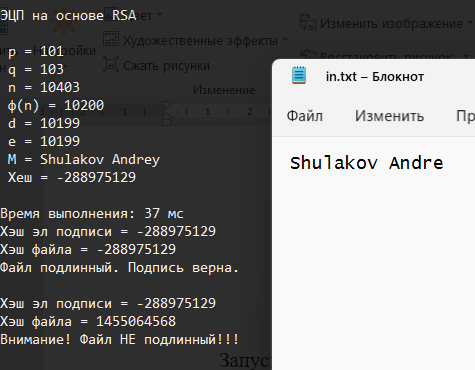


Рис. 2.4 – Результат работы алгоритма RSA

## 2.2. Алгоритм Эль-Гамаля

Алгоритм генерации подписи заключается в следующих операциях:

* выбор простого числа *p*;
* вычисление *g*, причем *g<p* – первообразный корень по модулю p;
* выбор *x*, меньшего *p*;
* вычисление *y = gx mod p*.

В результате зашифрования сообщения с ЭЦП на выходе будет лишь одна пара чисел, не для каждого блока сообщения.

Далее необходимо проделать следующие операции:

* выбрать *k* – взаимно простое число с (*p*-1);
* вычислить *a = gk mod p*;
* вычислить *b = k-1 (H(Mo) – xa) mod (p-*1*)*;

Пара чисел *S = {a,b}* и будет являться цифровой подписью. Далее получателю будет отправлено сообщение *M’=Mo||S*, которое является конкатенацией исходного сообщения и ЭЦП.

Для верификации подлинности полученного сообщения необходимо проверить равенство *ya × ab (mod p)* = *gh (mod p*), в которое подставляются все вычисленные ранее значения, *h=H(Mп)* – хеш-образ полученного сообщения.

Если данное равенство выполняется, подпись верифицирована и подлинна.

Протестируем приложение, запустив его. Результат в консоли представлен на рисунке 2.4. Изначально на консоль выводятся все значения, которые мы задавали самостоятельно или которые были вычислены в ходе работы приложения. Далее выводится значения ЭЦП и результат проверки.

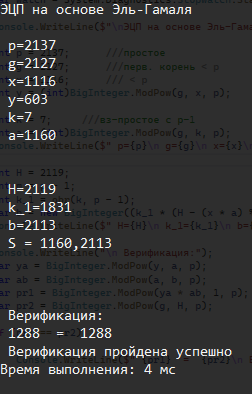


Рис. 2.4 – Результат работы алгоритма Эль-Гамаля

## 2.3. Алгоритм Шнорра

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.

Алгоритм генерации ключевой информации заключается в следующих шагах:

* генерация простых чисел *p, q*;
* вычисление (*p-*1) – делителя;
* выбор любого *g*≠1, такого что *gq =* 1 *mod p*;
* выбор любого числа *х<q* – это и будет закрытый ключ;
* вычисление *y = g–X* *mod p*;

Для собственно подписи сообщения необходимо реализовать следующий алгоритм:

* выбор случайного числа *k < q*;
* вычислить *a = gk mod p*;
* создать хеш-образ сообщения: *h = H(Mo||a)*;
* вычислить *b = (k+xh) mod q*;

Получателю будет отправлено сообщение *M’ = Mo||S*, являющееся конкатенацией исходного сообщения и ЭЦП *S={h,b*}.

Для проверки подписи на подлинность необходимо вычислить *X = gb × yh (mod p),* после чего проверить выполняется ли равенство вычисленного ранее *h* хеш-образа и *H(Мп||X)* хеш-образа конкатенации полученного сообщения с вычисленным значением *Х*. Если равенство выполняется, подпись верифицирована.

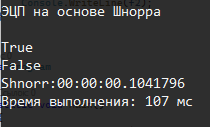


Рис. 2.5 – Результат работы алгоритма Шнорра

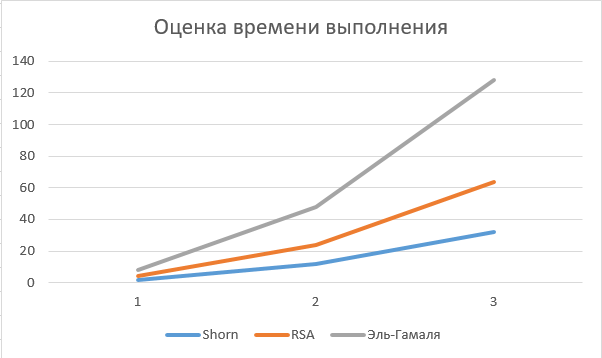


Рис. 2.6 – Оценка времени выполнения ЭЦП в алгоритмах Shorn, RSA, Эль-Гамаля

## Вывод

В ходе лабораторной работы были изучены алгоритмы генерации и верификации электронной цифровой подписи, а также приобретены практические навыки их реализации.

Также была оценена скорость генерации и верификации ЭЦП.