**Лабораторная работа №5**

*Тема: Схема* [*Эль-Гамаля*](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%AD%D0%BB%D1%8C-%D0%93%D0%B0%D0%BC%D0%B0%D0%BB%D1%8F) *и хэш-функция* [*SHA HYPERLINK "https://ru.wikipedia.org/wiki/SHA-2"-2*](https://ru.wikipedia.org/wiki/SHA-2)*.*

*Введение.*

Формально, как и для любой схемы [электронно-цифровой подписи](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D0%BF%D0%BE%D0%B4%D0%BF%D0%B8%D1%81%D1%8C) (ЭЦП), для работы схемы Эль-Гамаля необходимо реализовать 3 основных алгоритма: генерацию ключей, функцию подписи и функцию проверки подписи. Подробности можно посмотреть в [лекции](https://drive.google.com/open?id=1HwaVAWHAakhzNjurTQyZUn4sp5b1rHI0). С учетом того, что в лабораторной не требуется реализовывать генерацию простых чисел, алгоритмы ЭЦП имеют следующий вид:

*Генерация ключей (Gen)*

*Вход: q –* простое число.

*Шаги:*

* Выбрать такое четное число *R,* что *R < 4 (q + 1).*
* Вычислить число *p = qR + 1*.
* Если *2qR ≠ 1 (mod p)* или *2R = 1 (mod p),* то возвратиться к шагу 1, иначе *p* – простое.
* Случайным образом выбрать *x* из *Zp*и вычислить *g = xR (mod p).*
* Если *g = 1*, то возвратиться к шагу 4, иначе искомое *g* найдено.
* Случайным образом выбрать личный ключ *d* из *Zq.*
* Вычислить открытый ключ *e = gd (mod p).*

*Выход: (p, q, g)* – параметры ЭЦП; e – открытый ключ; *d* – личный ключ.

*Функция подписи (Sign)*

*Вход: (p, q, g)* – параметры ЭЦП; *d* – личный ключ; *M* – подписываемое сообщение (в виде строки текста произвольной длины).

*Шаги:*

* Вычислить хэш-значение *m* от сообщения *M*: *m = h(M)* (*h()* – хэш-функция).
* Случайным образом выбрать одноразовый личный ключ *k* из *Zq / {0}.*
* Вычислить *r = gk (mod p)*.
* Вычислить *s = k-1(m – dr) (mod q).*

*Выход:* *(r, s)* – подпись.

*Функция проверки подписи (Verify)*

*Вход: (p, q, g)* – параметры ЭЦП; *e* – открытый ключ; *M* – подписываемое сообщение (в виде строки текста произвольной длины; *(r, s)* – подпись.

*Шаги:*

* Если *r* не лежит в *Zp / {0}* или *s* не лежит в *Zq*, то вернуть *FALSE.*
* Вычислить хэш-значение *m* от сообщения *M*: *m = h(M)* (*h()* – хэш-функция).
* Если *errs = gm (mod p),* то вернуть *TRUE*, иначе вернуть *FALSE.*

*Выход: TRUE,* если подпись корректна; *FALSE,* если подпись некорректна.

*Замечания:*

* Для возведения в степень рекомендуется использовать [алгоритм быстрого возведения в степень](http://e-maxx.ru/algo/binary_pow). Нельзя использовать готовую реализацию этого алгоритма, если она есть в языке программирования, на котором вы выполняете лабораторную работу.
* Найти *k-1* можно с помощью [расширенного алгоритма Евклида](http://e-maxx.ru/algo/export_extended_euclid_algorithm). Нельзя использовать готовую реализацию этого алгоритма, если она есть в языке программирования, на котором вы выполняете лабораторную работу.
* В качестве хэш-функции используйте SHA-256. Ее реализация, написанная на С11, [находится](https://drive.google.com/open?id=1p0VUsuCfF33UYHIqtNWi8LKc0t0H18sB) в папке с условием [в файлах sha256.h и sha256.c осуществлена реализация хэш-функции SHA-256, а в файле main.c приводится пример вычисления хэш-значения]. Вы также можете найти готовую / написать свою реализацию SHA-256 на любом другом допустимом языке программирования.

*Условие лабораторной работы:*

* Для выполнения всех заданий необходимо использовать "длинную" арифметику. Разрешается использовать любую готовую библиотеку или написать свою. При этом учитывайте, что допустимые языки программирования {С/С++, C#, Java, Python, R}.
* Ваша программа должна принимать на вход указание, какую операцию необходимо выполнить: *Gen*, *Sign*, *Verify или Exit*. В зависимости от операции программа должна считывать (из консоли, файла и т.п.) необходимые данные, выполнять шаги выбранного алгоритма и возвращать (в консоль, файл и т.п.) результат выполнения соответствующей операции*.* При вызове операции *Exit* программа не считывает входных данных, а просто завершает работу.
* Программа должна иметь проверку корректности входных данных (*d* лежит в *Zq* для *Sign*; *e* лежит в *Zp* для *Verify*).
* Программа должна предлагать пользователю выполнить очередную операцию до тех пор, пока пользователь не выберет вариант выхода из программы.

*Отчет (*условия по вариантам смотрите на последней странице)*:*

1) В условиях своего варианта для заданного числа *q* необходимо выполнить алгоритм *Gen*.

2) Для параметров, полученных с помощью *Gen,* и сообщения вида «I, Ivan Ivanov, love MiKOZI», где «Ivan Ivanov» – ваше имя и фамилия соответственно, выполните алгоритм *Sign*.

3) Проверьте результаты подписи из 2) пункта, выполнив алгоритм *Verify.*

4) Результаты шагов 1 – 3 необходимо записать в текстовый файл, назвав его «Report.txt».

*Бонусное задание.*

Реализовать один из следующих методов нахождения дискретного логарифма:

* [Алгоритм HYPERLINK "https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%93%D0%B5%D0%BB%D1%8C%D1%84%D0%BE%D0%BD%D0%B4%D0%B0\_%E2%80%94\_%D0%A8%D0%B5%D0%BD%D0%BA%D1%81%D0%B0"Гельфонда HYPERLINK "https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%93%D0%B5%D0%BB%D1%8C%D1%84%D0%BE%D0%BD%D0%B4%D0%B0\_%E2%80%94\_%D0%A8%D0%B5%D0%BD%D0%BA%D1%81%D0%B0" — HYPERLINK "https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%93%D0%B5%D0%BB%D1%8C%D1%84%D0%BE%D0%BD%D0%B4%D0%B0\_%E2%80%94\_%D0%A8%D0%B5%D0%BD%D0%BA%D1%81%D0%B0"Шенкса](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%93%D0%B5%D0%BB%D1%8C%D1%84%D0%BE%D0%BD%D0%B4%D0%B0_%E2%80%94_%D0%A8%D0%B5%D0%BD%D0%BA%D1%81%D0%B0);
* [Алгоритм HYPERLINK "https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%9F%D0%BE%D0%BB%D0%B8%D0%B3%D0%B0\_%E2%80%94\_%D0%A5%D0%B5%D0%BB%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0"Полига HYPERLINK "https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%9F%D0%BE%D0%BB%D0%B8%D0%B3%D0%B0\_%E2%80%94\_%D0%A5%D0%B5%D0%BB%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0" — Хеллмана](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%9F%D0%BE%D0%BB%D0%B8%D0%B3%D0%B0_%E2%80%94_%D0%A5%D0%B5%D0%BB%D0%BB%D0%BC%D0%B0%D0%BD%D0%B0);
* [ро HYPERLINK "https://ru.wikipedia.org/wiki/%D0%A0%D0%BE-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4\_%D0%9F%D0%BE%D0%BB%D0%BB%D0%B0%D1%80%D0%B4%D0%B0\_%D0%B4%D0%BB%D1%8F\_%D0%B4%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B3%D0%BE\_%D0%BB%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F"-метод HYPERLINK "https://ru.wikipedia.org/wiki/%D0%A0%D0%BE-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4\_%D0%9F%D0%BE%D0%BB%D0%BB%D0%B0%D1%80%D0%B4%D0%B0\_%D0%B4%D0%BB%D1%8F\_%D0%B4%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B3%D0%BE\_%D0%BB%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F"Полларда](https://ru.wikipedia.org/wiki/%D0%A0%D0%BE-%D0%BC%D0%B5%D1%82%D0%BE%D0%B4_%D0%9F%D0%BE%D0%BB%D0%BB%D0%B0%D1%80%D0%B4%D0%B0_%D0%B4%D0%BB%D1%8F_%D0%B4%D0%B8%D1%81%D0%BA%D1%80%D0%B5%D1%82%D0%BD%D0%BE%D0%B3%D0%BE_%D0%BB%D0%BE%D0%B3%D0%B0%D1%80%D0%B8%D1%84%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F);
* [Алгоритм HYPERLINK "https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC\_%D0%90%D0%B4%D0%BB%D0%B5%D0%BC%D0%B0%D0%BD%D0%B0"Адлемана](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%90%D0%B4%D0%BB%D0%B5%D0%BC%D0%B0%D0%BD%D0%B0);
* [Алгоритм COS](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_COS).

Алгоритм должен использовать "длинную" арифметику. Успешная реализация любого из алгоритмов дает возможность НЕ выполнять 0.5 (половину) любой из 5 (пяти) основных лабораторных работ. Можно реализовать несколько алгоритмов: все бонусы суммируются (т.е., если вы реализуете 2 (два) алгоритма, то сможете НЕ выполнять 1 (одну) из 5 (пяти) основных лабораторных работ).

*Порядок сдачи лабораторной работы:*

Вам необходимо создать архив формата «.zip», название которого должно иметь вид «5\_5\_Ivanov.zip», где «5» – № вашей группы, а «Ivanov» – ваша фамилия латинскими буквами. В архиве должен быть исходный код вашей реализации двух классических шифров (НЕ весь проект целиком, а только файлы с исходным кодом (.c, .cpp, .java и т.п.)) и файл «Report.txt»;

Таблица №1: Простое число q для алгоритма *Gen*

|  |  |
| --- | --- |
| **№ вар.** | **q** |
| 1 | 140990220132661942094353836270106675983065715626747592141413983992936990059339 |
| 2 | 139917935451086595095698262488180793987376018819267640047704411792656425400927 |
| 3 | 172653929034048084950674429899193619418057473778703331247504102312524336679143 |
| 4 | 210697455032337684943121194039863591186004713463570796268765689709223108292419 |
| 5 | 225523118922100465769758148413025421807220956799975071969675156327098020673011 |
| 6 | 184473195695736471971416350642657205911659384914907526521111574974012585936539 |
| 7 | 185497337331032352746789861834054230815003783507015530555482386637095968522879 |
| 8 | 204549505434169694705359613953840507945808782458714026455216935639206129681467 |
| 9 | 123604951293454262968546700045396147510790229186492084779809948917693394204003 |
| 10 | 178706295845602514261898501512678379830100933245565184651730520471740478551107 |
| 11 | 191009199543846311909430551847309684248048071280580287138396185530444394820287 |
| 12 | 169394527852051056333893830578611667478723590794747008027304972047381407838783 |
| 13 | 183389827575056336316166878451739830722567621993753249490764758988976692094991 |
| 14 | 228620023921267193730928153886743793396324452340577138987972760236418208443847 |
| 15 | 229450218509507942931245813212393379294488209726688075663898430612977467498699 |