

Лабораторная работа №5

Тема: Схема Эль-Гамала и хэш-функция SHA-2.

Введение.

Формально, как и для любой схемы электронно-цифровой подписи (ЭЦП), для работы схемы Эль-Гамала необходимо реализовать 3 основных алгоритма: генерацию ключей, функцию подписи и функцию проверки подписи. Подробности можно посмотреть в лекции. 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:

1. Выбрать такое четное число R , что $R < 4(q + 1)$.
2. Вычислить число $p = qR + 1$.
3. Если $2^{qR} \not\equiv 1 \pmod{p}$ или $2^R \equiv 1 \pmod{p}$, то возвратиться к шагу 1, иначе p – простое.
4. Случайным образом выбрать x из Z_p и вычислить $g = x^R \pmod{p}$.
5. Если $g = 1$, то возвратиться к шагу 4, иначе искомое g найдено.
6. Случайным образом выбрать личный ключ d из Z_q .
7. Вычислить открытый ключ $e = g^d \pmod{p}$.

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:

1. Вычислить хэш-значение m от сообщения M : $m = h(M)$ ($h()$ – хэш-функция).
2. Случайным образом выбрать одноразовый личный ключ k из $Z_q \setminus \{0\}$.
3. Вычислить $r = g^k \pmod{p}$.
4. Вычислить $s = k^{-1}(m - dr) \pmod{q}$.

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:

1. Если r не лежит в $Z_p \setminus \{0\}$ или s не лежит в Z_q , то вернуть *FALSE*.
2. Вычислить хэш-значение m от сообщения M : $m = h(M)$ ($h()$ – хэш-функция).
3. Если $e^r r^s = g^m \pmod{p}$, то вернуть *TRUE*, иначе вернуть *FALSE*.

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

Замечания:

- 1) Для возведения в степень рекомендуется использовать [алгоритм быстрого возведения в степень](#). Нельзя использовать готовую реализацию этого алгоритма, если она есть в языке программирования, на котором вы выполняете лабораторную работу.
- 2) Найти k^{-1} можно с помощью [расширенного алгоритма Евклида](#). Нельзя использовать готовую реализацию этого алгоритма, если она есть в языке программирования, на котором вы выполняете лабораторную работу.
- 3) В качестве хэш-функции используйте SHA-256. Ее реализация, написанная на C11, [находится](#) 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:

- 1) Для выполнения всех заданий необходимо использовать "длинную" арифметику. Разрешается использовать любую готовую библиотеку или написать свою. При этом учитывайте, что допустимые языки программирования {C/C++, C#, Java, Python, R}.
- 2) 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.
- 3) Программа должна иметь проверку корректности входных данных (d лежит в Z_q для *Sign*; e лежит в Z_p для *Verify*).
- 4) 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:

- 1) [Алгоритм Гельфонда — Шенкса](#);
- 2) [Алгоритм Полига — Хеллмана](#);
- 3) [ро-метод Полларда](#);
- 4) [Алгоритм Адлемана](#);
- 5) [Алгоритм 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:

- 1) 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»;
- 2) Не позже, чем за 24 часа до сдачи лабораторной работы преподавателю в университете, вы должны отправить архив по одному из контактов, указанных в ответе на вопрос №5 в файле «[FAQ.pdf](#)».

Н. В. При возникновении затруднений, связанных с выполнением лабораторной работы, вы можете проконсультироваться с преподавателем, написав ему на один из контактов, указанных в ответе на вопрос №5 в файле «[FAQ.pdf](#)».

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