НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Московский институт электроники и математики им. А.Н. Тихонова

**КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ**

**Практическая работа 8**

**Схемы электронной подписи**

**Евсютин О.О.**

Москва 2024

**1 ЦЕЛЬ РАБОТЫ**

Целью данной работы является приобретение навыков программной реализации схем электронной подписи.

**2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**2.1 Общие сведения об электронной подписи**

Электронная цифровая подпись (электронная подпись, цифровая подпись) сообщения – это строка бит, присоединяемая к сообщению и зависящая от сообщения и секретного элемента данных – ключа подписи, известного только его владельцу. Электронная цифровая подпись предназначена для аутентификации лица, подписавшего электронное сообщение. Кроме того, использование электронной цифровой подписи предоставляет возможность обеспечить следующие свойства при передаче в системе подписанного сообщения:

* осуществить контроль целостности передаваемого подписанного сообщения;
* доказательно подтвердить авторство лица, подписавшего сообщение;
* защитить сообщение от возможной подделки.

Существуют два основных подхода к построению схем электронной цифровой подписи: на основе симметричных шифров и на основе алгоритмов с открытым ключом.

Первый подход предусматривает наличие посредника или арбитра, обладающего доверием всех участников информационного обмена. Каждый пользователь имеет с посредником общий ключ шифрования, отличный от аналогичных ключей других пользователей. Эти ключи выдаются пользователям посредником и могут использоваться многократно. Такая схема достаточно хорошо работает, но проблемой является нахождение непогрешимого посредника.

Схемы электронной цифровой подписи на основе алгоритмов с открытым ключом являются наиболее распространенными на сегодняшний день, и именно такие схемы положены в основу действующих государственных стандартов.

В данном случае каждый участник электронного взаимодействия обладает двумя ключами: ключом подписи и ключом проверки. Ключ подписи представляет собой закрытый ключ, используемый для формирования электронной цифровой подписи и известный только его владельцу. Ключ проверки представляет собой открытый ключ, математически связанный с ключом подписи, известный всем участникам электронного взаимодействия и используемый ими для проверки электронной цифровой подписи данного участника.

**2.2 Стандарты электронной подписи**

**2.2.1 ГОСТ Р 34.10-94**

Первый отечественный стандарт электронной цифровой подписи описан в нормативном документе «ГОСТ Р 34.10-94. Информационная технология. Криптографическая защита информации. Процедуры выработки и проверки электронной цифровой подписи на базе асимметричного криптографического алгоритма». Данный стандарт был введен 1 января 1994 года и действовал до середины 2002 года.

Важной особенностью стандартов электронной цифровой подписи является использование хэш-функций. Подписываемое сообщение предварительно сворачивается в строку фиксированной длины с помощью хэш-функции, и процедура выработки электронной цифровой подписи применяется к полученному хэш-коду вместо самого сообщения. С аналогичного действия начинается процедура проверки электронной цифровой подписи для данного сообщения.

В ГОСТ Р 34.10-94 для этой цели используется хэш-функция, определенная в стандарте ГОСТ Р 34.11-94.

Все вычисления в ГОСТ Р 34.10-94 производятся в конечном поле по модулю простого числа. Стойкость соответствующей схемы электронной цифровой подписи основывается на сложности задачи дискретного логарифмирования в мультипликативной группе такого конечного поля, а также на стойкости используемой хэш-функции. Собственно электронная цифровая подпись представляет собой двоичный вектор длиной 512 бит.

**2.2.2 ГОСТ Р 34.10-2001**

Второй отечественный стандарт электронной цифровой подписи описан в нормативном документе «ГОСТ Р 34.10-2001. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи». Данный стандарт был введен 1 июля 2002 года и действовал до 31 декабря 2012 года.

Отличие между ГОСТ Р 34.10-94 и ГОСТ Р 34.10-2001 заключается в том, что в стандарте 2001 года все вычисления производились в группе точек эллиптической кривой над конечным полем, в том время как в стандарте 1994 года все вычисления производились в конечном поле по модулю простого числа. Соответственно стойкость стандарта ГОСТ Р 34.10-2001 основывается на сложности задачи дискретного логарифмирования в группе точек эллиптической кривой, а также на стойкости используемой хэш-функции

ГОСТ Р 34.10-2001 определяет схему электронной цифровой подписи, процессы формирования и проверки цифровой подписи под заданным сообщением (документом), передаваемым по незащищенным телекоммуникационным каналам общего назначения в системах обработки информации различного назначения.

Для хэширования сообщения в данном стандарте используется хэш-функция ГОСТ Р 34.11-94.

Электронная цифровая подпись, вырабатываемая по ГОСТ Р 34.10-2001, представляет собой двоичный вектор длиной 512 бит.

**2.2.3 ГОСТ Р 34.10-2012**

Действующим отечественным стандартом электронной цифровой подписи является стандарт, описанный в нормативном документе «ГОСТ Р 34.10-2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи».

Данный стандарт введен 1 января 2013 года на смену ГОСТ Р 34.10-2001. Замена стандарта связана с тем, что внедрение цифровой подписи на основе ГОСТ Р 34.10-2012 повышает, по сравнению с ранее действовавшей схемой электронной цифровой подписи, уровень защищенности передаваемых сообщений от подделок и искажений. Однако алгоритмы формирования и проверки электронной цифровой подписи ГОСТ Р 34.10-2012 полностью повторяют аналогичные алгоритмы ГОСТ Р 34.10-2001. Фактически между двумя данными стандартами есть только два существенных отличия:

* ГОСТ Р 34.10-2001 формирует электронную цифровую подпись длиной 512 бит, в то время как в ГОСТ Р 34.10-2012 длина электронной цифровой подписи варьируется и может составлять 512 бит и 1024 бита;
* для хэширования сообщения в ГОСТ Р 34.10-2012 используется хэш-функция ГОСТ Р 34.11-2012 вместо ГОСТ Р 34.11-94, используемой ранее в ГОСТ Р 34.10-2001.

Схема электронной цифровой подписи, представленная в ГОСТ Р 34.10-2012, оперирует следующими параметрами:

* Простое число – модуль эллиптической кривой.
* Эллиптическая кривая , задаваемая своим инвариантом или коэффициентами , .
* Целое число – порядок группы точек эллиптической кривой , такое, что  
  .
* Простое число – порядок циклической подгруппы группы точек эллиптической кривой , для которого выполнены следующие условия:
* Точка эллиптической кривой с координатами , удовлетворяющая равенству .
* Хэш-функция , .
* Ключ подписи – целое число *.*
* Ключ проверки – точка эллиптической кривой .

*Алгоритм формирования подписи*

* 1. Вычислить хэш-код сообщения : .
  2. Вычислить целое число , двоичным представлением которого является вектор , и определить . Если , то определить .
  3. Сгенерировать случайное целое число , удовлетворяющее неравенству .
  4. Вычислить точку эллиптической кривой и определить . Если , то вернуться к шагу 3
  5. Вычислить значение . Если , то вернуться к шагу 3.
  6. Вычислить двоичные векторы, соответствующие числам и , и определить цифровую подпись как конкатенацию данных двоичных векторов.

*Алгоритм проверки подписи*

* 1. По полученной подписи вычислить целые числа и. Если выполнены неравенства , , то перейти к следующему шагу. В противном случае подпись неверна.
  2. Вычислить хэш-код сообщения .
  3. Вычислить целое число , двоичным представлением которого является вектор , и определить . Если , то определить .
  4. Вычислить значение .
  5. Вычислить значения , .
  6. Вычислить точку эллиптической кривой и определить значение .
  7. Если выполнено равенство , то подпись принимается, в противном случае, подпись неверна.

**2.2.4 DSS**

Стандарт электронной цифровой подписи, действующий в США, носит название DSS (Digital Signature Standard). Он имеет следующую историю. Исходный стандарт опубликован в 1994 году в документе FIPS PUB 186. Данный документ несколько раз пересматривался и публиковался в следующих редакциях:

* 1998 год, FIPS PUB 186-1;
* 2000 год, FIPS PUB 186-2;
* 2009 год, FIPS PUB 186-3;
* 2013 год, FIPS PUB 186-4;
* 2023 год, FIPS PUB 186-5.

Актуальный стандарт DSS включает три алгоритма, предназначенных для формирования и проверки электронной цифровой подписи:

* алгоритм цифровой подписи на основе криптосистемы RSA;
* алгоритм цифровой подписи на основе эллиптических кривых ECDSA (Elliptic Curve Digital Signature Algorithm);
* алгоритм цифровой подписи на основе кривых Эвардса EdDSA (Edwards Curve Digital Signature Algorithm).

Алгоритм ECDSA по своему устройству схож с аналогичным алгоритмом, представленным в российском стандарте электронной цифровой подписи. Алгоритм EdDSA основан на семействе эллиптических кривых Эдвардса, обладающих некоторыми преимуществами перед эллиптическими кривыми в форме Вейерштрасса.

**2.3 Сложение точек эллиптической кривой**

Эллиптической кривой над конечным полем вычетов по модулю простого числа  
 называется множество точек , удовлетворяющих уравнению  
, где *a*, *b*   и , дополненное бесконечно удаленной точкой 0, не имеющей численного выражения. Данное множество точек, обозначаемое , представляет собой абелеву группу относительно операции сложения точек.

Операция сложения точек эллиптической кривой задается следующим образом. Чтобы сложить точки и , необходимо провести через них прямую, которая в общем случае будет проходить еще через одну точку эллиптической кривой. Эту третью точку необходимо симметрично отразить относительно оси абсцисс, полученный результат и будет представлять собой сумму .

Зная координаты двух исходных точек и , достаточно легко вывести формулы для нахождения координат третьей точки . При этом необходимо учесть три случая.

Первый случай. Складываются две одинаковые точки и . При выводе координат результирующей точки необходимо воспользоваться уравнением касательной к эллиптической кривой. Формулы для нахождения координат точки  
 имеют вид

(1)

Второй случай. Складываются две разные точки, и , причем  
. При выводе координат результирующей точки необходимо воспользоваться уравнением секущей к эллиптической кривой. Формулы для нахождения координат точки имеют вид

(2)

Третий случай. Складываются две разные точки, и , причем  
, . Такие точки являются взаимно обратными элементами группы , то есть , поэтому их сумма дает нейтральный элемент группы, то есть бесконечно удаленную точку 0.

**2.4 Теоретико-числовые алгоритмы для реализации криптографических преобразований**

**2.4.1 Нахождение обратного элемента по модулю простого числа**

Формулы (1) и (2) используют операцию деления, под которой в арифметике остатков подразумевается умножение на обратное по модулю значение. Для нахождения обратного значения по модулю натурального числа применяется расширенный алгоритм Евклида.

Вход: целые числа .

Выход: и целые , , такие, что .

1. Полагаем , , , .

2. Пока , выполнять следующее:

2.1. , , , ;

2.2. , , , , , .

3. , , и возврат .

Чтобы найти , необходимо подать на вход алгоритма Евклида пару , и если , вернуть в качестве значение .

**2.4.2 Возведение в степень по модулю**

Криптографические алгоритмы с открытым ключом при их использовании на практике оперируют числами большой битовой длины (или просто большими числами), когда речь идет о сотнях и тысячах бит. Для некоторых операцией над такими числами созданы специальные алгоритмы. В случае криптосистем RSA и Эль-Гамаля необходимо иметь алгоритм, который позволит осуществлять быстрое возведение в степень по модулю. Данный алгоритм представлен ниже.

*Алгоритм возведения в степень по модулю.*

Вход: , .

Выход: .

1. . Если , то переход к шагу 5.

2. .

3. Если , то .

4. Для выполняем следующее:

4.1. .

4.2. Если , то .

5. Возврат .

Сложение точек эллиптической кривой осуществляется по аналогичному алгоритму. Если для данной точки необходимо вычислить точку , то искомая точка представляется в виде или в зависимости от четности числа . Далее происходит удвоение точки по формулам (1), после чего процесс повторяется, пока не будет вычислена искомая точка.

**2.4.3 Тесты целых чисел на простоту**

Еще одним важным аспектом криптографии с открытым ключом является использование простых чисел.

Наиболее развитые вероятностные алгоритмы проверки чисел на простоту основаны на малой теореме Ферма.

*Малая теорема Ферма.*

Пусть — простое число, и . Тогда .

Соотношение, приведенное в теореме, используется в тесте, проверяющем, является ли заданное число составным. Этот тест называют тестом Ферма.

*Тест Ферма.*

Вход: нечетное число .

Выход: ответ на вопрос «является ли простым».

1. Для выполняем следующее:

1.1. Выбираем случайное целое число .

1.2. Вычисляем с помощью алгоритма возведения в степень по модулю.

1.3. Если , то возврат «*n* — составное».

Тест Ферма по основанию определяет простоту с вероятностью , после итераций вероятность ошибки составляет .

**3 ЗАДАНИЕ**

1. написать программную реализацию схемы электронной цифровой подписи, представленной в ГОСТ Р 34.10-2012. Программная реализация должна быть выполнена студентом самостоятельно без использования готовых библиотечных решений (допускается использование готовой реализации хэш-функции ГОСТ Р 34.11-2012);
2. подготовить отчет о выполнении работы.

Программа должна обладать следующей функциональностью:

1. принимать на вход файл, для которого необходимо сформировать или проверить электронную цифровую подпись;
2. принимать на вход файл, содержащий электронную цифровую подпись;
3. принимать на вход ключ подписи или ключ проверки подписи;
4. давать пользователю возможность сгенерировать ключевую пару;
5. осуществлять формирование или проверку электронной цифровой подписи по выбору пользователя.

Отчет должен содержать следующие составные части:

1. раздел с заданием;
2. раздел с краткой теоретической частью;
3. раздел с описанием программной реализации с учетом особенностей выбранной среды разработки и языка программирования и кодом. Допускается привести код программной реализации в отчёте, приложить отдельным файлом или привести в отчёте ссылку на программный код;
4. раздел с результатами работы программы;
5. раздел с выводами о проделанной работе.