НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

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

**КРИПТОГРАФИЧЕСКИЕ МЕТОДЫ ЗАЩИТЫ ИНФОРМАЦИИ**

**Практическая работа 4**

**Криптосистемы с открытым ключом**

**Евсютин О.О.**

Москва 2025

**1 ЦЕЛЬ РАБОТЫ**

Целью данной работы является приобретение навыков программной реализации криптосистем с открытым ключом.

**2 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ**

**2.1 Криптография с открытым ключом**

Основной проблемой симметричной криптографии является обеспечение конфиденциальности ключей шифрования при их распределении между пользователями. С данной проблемой сталкиваются все симметричные криптосистемы. Стороны–участники защищенного информационного обмена по открытому каналу связи должны каким-то образом предварительно получить общий секретный ключ, который нельзя передавать в открытом виде. Кроме того, необходимо обеспечить эффективное управление сеансовыми ключами, безопасное хранение долговременных ключей и т. д. Все перечисленные проблемы настолько важны, что в современной криптографии управление ключами выделяют в отдельный раздел.

Для решения проблемы обеспечения конфиденциальности ключей симметричного шифрования в 1976 году была предложена концепция криптографии с открытым ключом.

Криптосистемы с открытым ключом, также называемые асимметричными криптосистемами, используют два разных ключа: открытый ключ зашифрования и закрытый ключ расшифрования. Открытый ключ в общем случае доступен всем желающим, а закрытый ключ известен только законному владельцу. Оба ключа связаны между собой некоторой зависимостью. При этом данная зависимость такова, что, зная один ключ, вычислить другой практически невозможно.

Решение указанной проблемы заключается в том, что отсутствует необходимость в передаче секретной ключевой информации между пользователями. Закрытый ключ должен быть известен лишь его владельцу и не требует передачи.

Базовым понятием в криптографии с открытым ключом является понятие однонаправленной или односторонней функции. Значение такой функции для заданного аргумента достаточно легко вычислить, но практически невозможно вычислить значение аргумента для заданного значения функции. Непосредственно для шифрования однонаправленные функции не используются. Для шифрования применяются однонаправленные функции с лазейкой (ловушкой). Для такой функции вычислить обратную функцию достаточно просто, если известна некоторая секретная информация, и практически невозможно, если эта информация неизвестна.

В асимметричных криптосистемах, построенных на однонаправленных функциях с лазейкой, открытый ключ определяет конкретную реализацию функции, а закрытый ключ дает информацию о лазейке. Основными задачами, приводящими к однонаправленным функциям, являются задачи разложения на множители и дискретного логарифмирования в конечном поле.

**2.2 Криптосистемы с открытым ключом**

**2.2.1 Криптосистема RSA**

Данная криптосистема является первой криптосистемой с открытым ключом. Она основывается на сложности проблемы факторизации целых чисел, то есть разложения целых чисел на простые множители.

*Алгоритм генерации ключей.*

1. Пользователь *A* генерирует два больших простых числа и , отличных друг от друга. При этом – большое число, хотя и имеют приблизительно одинаковый битовый размер.

2. Держа и в секрете, Пользователь *A* вычисляет их произведение , которое называют модулем алгоритма.

3. Пользователь *A* вычисляет значение функции Эйлера для по формуле  
.

4. Пользователь *A* выбирает целое число , взаимно простое со значением функции . Это число называется экспонентой зашифрования.

5. Пользователь *A* применяет расширенный алгоритм Евклида к паре чисел и и вычисляет значение , удовлетворяющее соотношению. Это значение называется экспонентой расшифрования.

6. Пара публикуется в качестве открытого ключа пользователя *A*, является закрытым ключом и держится в секрете.

*Алгоритм зашифрования.*

1. Пользователь *B* получает аутентичную копию открытого ключа пользователя *A* – пару .

2. Пользователь *B* представляет сообщение в виде числа , меньшего модуля алгоритма. В общем случае сообщение может быть разбито на блоки, каждый из которых представляется своим числом.

3. Пользователь *B* вычисляет .

4. Зашифрованное сообщение отправляется пользователю *A*.

*Алгоритм расшифрования.*

1. Пользователь *A* получает криптограмму от пользователя *B*.

2. Пользователь *A* вычисляет .

**2.2.2 Криптосистема Рабина**

Криптосистема Рабина, как и криптосистема RSA, основывается на трудности задачи извлечения квадратного корня по модулю составного числа, которая, в свою очередь, сводится к задаче факторизации целых чисел.

*Алгоритм генерации ключей.*

1. *A* генерирует два больших простых числа и таких, что . Такой специальный вид простых чисел ускоряет процедуру извлечения квадратных корней по модулю и .

2. *A* вычисляет .

3. Открытый ключ *A* есть .

4. Закрытый ключ *A* есть пара .

*Алгоритм зашифрования.*

1. *B* получает аутентичную копию ключа *A*, то есть число .

2. *B* представляет сообщение в виде числа , меньшего . В общем случае сообщение может быть разбито на блоки, каждый из которых представляется своим числом.

3. *B* вычисляет .

4. Зашифрованное сообщение *B* отправляет *A*.

Алгоритм расшифрования.

1. *A* получает шифртекст .

2. *A* извлекает из четыре квадратных корня по модулю .

3. *A* определяет нужное значение из четырех корней.

Задача извлечения квадратного корня по модулю простого числа решается достаточно легко, а задача извлечения квадратного корня по модулю составного числа может быть легко решена только при известной факторизации составного модуля. В этом случае необходимо извлечь квадратные корни по модулям простых множителей составного числа и воспользоваться китайской теоремой об остатках. Поэтому законный пользователь *A* с легкостью может выполнить извлечение квадратных корней из шифртекста по модулю , в то время как злоумышленник, не знающий факторизацию , не может это сделать.

**2.2.3 Криптосистема Эль-Гамаля**

Данная криптосистема в настоящее время является одной из основных криптосистем с открытым ключом. В отличие от криптосистем RSA и Рабина, она основана на задаче дискретного логарифмирования в конечной абелевой группе. Кроме того, в криптосистеме Эль-Гамаля существуют некоторые открытые параметры, которые могут использоваться определенными группами пользователей. Они называются параметрами домена: — большое простое число, такое, что делится на другое простое число меньшей битовой длины; — элемент мультипликативной группы поля , степени которого по модулю порождают большое число элементов .

*Алгоритм генерации ключей.*

1. Пользователь *A* задает параметры домена и .

2. Пользователь *A* выбирает случайное число *x* в интервале .

3. Пользователь *A* вычисляет

4. Открытый ключ пользователя *A* есть .

5. Закрытый ключ пользователя *A* есть .

*Алгоритм зашифрования.*

1. Пользователь *B* получает аутентичную копию открытого ключа пользователя A — число .

2. Пользователь *B* представляет сообщение в виде числа в интервале  
. В общем случае сообщение может быть разбито на блоки, каждый из которых представляется своим числом.

3. Пользователь *B* вычисляет сеансовый ключ в интервале .

4. Пользователь *B* вычисляет и .

5. Пользователь *B* отправляет пару пользователю *A*.

*Алгоритм расшифрования.*

1. Пользователь *A* получает шифртекст — пару .

2. Пользователь *A*, используя свой секретный ключ, вычисляет открытый текст по следующей формуле:

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

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

Для нахождения обратного элемента 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 .

**2.3.2 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 .

**2.3.3 Тесты целых чисел на простоту**

Еще одним важным аспектом криптографии с открытым ключом является использование простых чисел, которые используются во всех рассмотренных выше криптосистемах.

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 одной из перечисленных ниже асимметричных криптосистем (по выбору студента) с использованием больших чисел. Программная реализация должна быть выполнена студентом самостоятельно без использования готовых библиотечных функций, напрямую реализующих алгоритм шифрования. Варианты криптосистем для реализации:

- RSA;

- Рабина;

- Эль-Гамаля;

1. изучить методы криптоанализа выбранной криптосистемы;
2. реализовать (вручную или программно) не менее одной атаки на выбранную криптосистему, исключая наивную переборную атаку, для случая, когда параметры криптосистемы не являются большими числами;
3. подготовить отчет о выполнении работы.

Программа должна обладать следующей функциональностью:

1. принимать на вход файл, содержащий открытый текст, подлежащий зашифрованию, или шифртекст, подлежащий расшифрованию;
2. принимать на вход ключевую пару (открытый ключ, закрытый ключ);
3. давать пользователю возможность сгенерировать ключевую пару;
4. осуществлять зашифрование или расшифрование введенного текста по выбору пользователя.

Отчет должен содержать следующие составные части:

1. раздел с заданием;
2. раздел с краткой теоретической частью;
3. раздел с примером «ручного» (без использования программных средств) зашифрования и расшифрования для произвольной последовательности символов и малого размера ключа;
4. раздел с описанием программного кода и результатами работы программы для той же последовательности символов, что и в предыдущем разделе, а также с использованием больших чисел. Допускается привести код программной реализации в отчёте, приложить отдельным файлом или привести в отчёте ссылку на программный код;
5. раздел с подробным описанием реализованной атаки на криптосистему с приведением численных результатов;
6. раздел с выводами о проделанной работе.