Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Основы защиты информации

Студент: Валдайцев А. Д.

ФИТ 2 курс 5 группа

Преподаватель: Берников В. О.

Минск 2022

**Практическое занятие №7**

**Тема «****Криптографическая защита информации»**

**Цель:** изучить и закрепить умение реализации ЭЦП на примере RSA**.**

**Теоретическое введение**

**Создание ЭЦП с применением RSA**

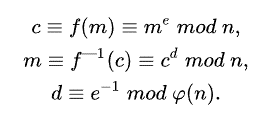
Электронная подпись - информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связана с такой информацией и которая используется для определения лица, подписывающего информацию.

 То есть, электронная подпись (ЭП) - определение лица, подписавшего некоторый документ. Помимо этого, ЭП может быть использована для определения искажений переданного сообщения.

Алгоритм RSA, помимо шифрования сообщений, позволяет также создать электронную подпись.

Асимметричные криптографические системы (которой является RSA) основаны на так называемых односторонних функциях с секретом. Под односторонней понимается такая функция y=f(x), которая легко вычисляется при имеющемся x, но аргумент x при заданном значении функции вычислить сложно. Аналогично, односторонней функцией с секретом называется функция y=f(x, k), которая легко вычисляется при заданном x, причём при заданном секрете k аргумент x по заданному y восстановить просто, а при неизвестном k – сложно.

Подобным свойством обладает операция возведения числа в степень по модулю:



Здесь *φ(n)* – функция Эйлера числа *n*. Теперь нужно понять, что из это является ключами Боба, а что сообщением. В нашем распоряжении имеются числа *c, m, n, e, d*.

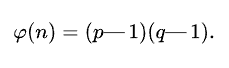
Давайте посмотрим на первое выражение. Здесь число c получено в результате возведения в степень по модулю числа m. Назовём это действие шифрованием. Тогда становится очевидно, что m выступает в роли открытого текста, а c – шифртекста. Результат c зависит от степени e, в которую мы возводим m, и от модуля n, по которому мы получаем результат шифрования. Эту пару чисел (e, n) мы будем называть открытым ключом. Им Алиса будет шифровать сообщение.

Смотрим на второе действие. Здесь d является параметром, с помощью которого мы получаем исходный текст m из шифртекста c. Этот параметр мы назовём закрытым ключом и выдадим его Бобу, чтобы он смог расшифровать сообщение Алисы.

Что есть что разобрались, теперь перейдём к конкретике, а именно – генерации ключей Боба. Давайте выберем число n такое, что:

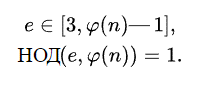


где p и q – некоторые разные простые числа. Для такого n функция Эйлера имеет вид:

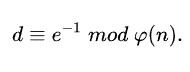


Такой выбор n обусловлен следующим. Как вы могли заметить ранее, закрытый ключ d можно получить, зная открытый e. Зная числа p и q, вычислить функцию Эйлера не является вычислительно сложной задачей, ровно как и нахождение обратного элемента по модулю. Однако в открытом ключе указано именно число n. Таким образом, чтобы вычислить значение функции Эйлера от n (а затем получить закрытый ключ), необходимо решить задачу факторизации, которая является вычислительно сложной задачей для больших n (в современных системах, основанных на RSA, n имеет длину 2048 бит).

Возвращаемся к генерации ключей. Выберем целое число **e**:



Для него вычислим число **d**:



Для отыскания числа, обратного по модулю, можно воспользоваться алгоритмом Евклида.

Мы завершили с этапом генерации ключей. Теперь Боб публикует свой открытый ключ *(e, n)*, прячет закрытый d, а мы переходим к Алисе.

Возьмём в качестве сообщения число *m* (*m* ∈ [1, *n* − 1]). Чтобы Алисе зашифровать его, необходимо возвести его в степень *e* по модулю *n*. Эти числа идут вместе с открытым ключом Боба:

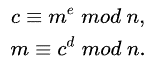


Здесь за с обозначен шифртекст, который Алиса будет должна передать Бобу. Отметим также, что *c* ∈ [1, *n* − 1], как и m. Расшифруем шифртекст, возведя его в степень закрытого ключа Боба *d*:

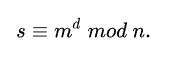


Причём, благодаря математическим соотношениям, *m* ≡ *m*′ и получатель идентифицирует подпись отправителя.

Ещё раз напишем две ключевые формулы шифрования и расшифрования соответственно:



Теперь давайте предположим, что Боб хочет отправить Алисе открытку *m* от своего имени. У Боба в распоряжении уже имеются два ключа *(e, n)* и *d*, которые он сгенерировал по алгоритму, описанному ранее. Поскольку *d* является закрытым ключом, то можно им воспользоваться как уникальным идентификатором Боба. Давайте "зашифруем" *m* с помощью *d*:



Результат данной операции и есть подпись сообщения Боба. Заметим, что подпись напрямую зависит от подписываемого сообщения, а не только от того, что его подписывает Боб. Далее, Алиса получает сообщение *m*, подпись *s* и открытый ключ *(e, n)*. По аналогии с расшифрованием, проверка подписи осуществляется возведением подписи *s* в степень открытой экспоненты *e*:

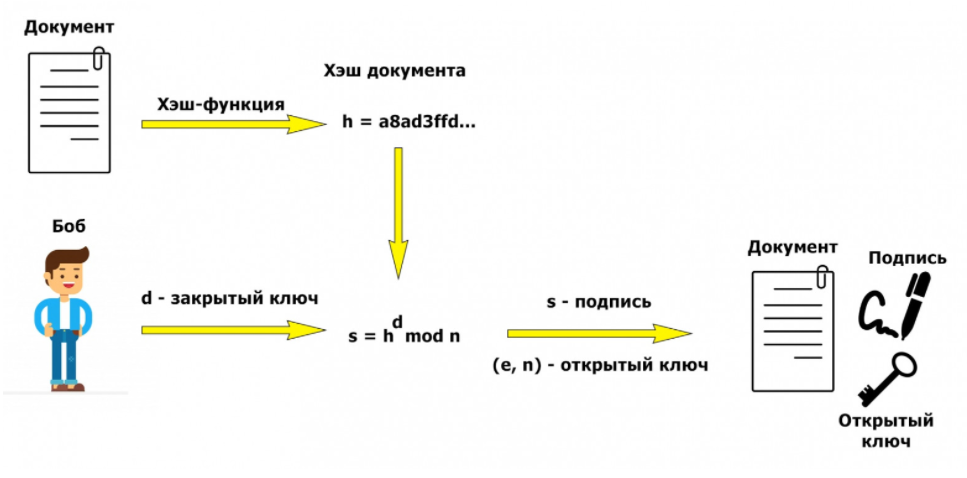


Если Алиса получила, что *m* ≡ *m*′, то подпись считается правильной.

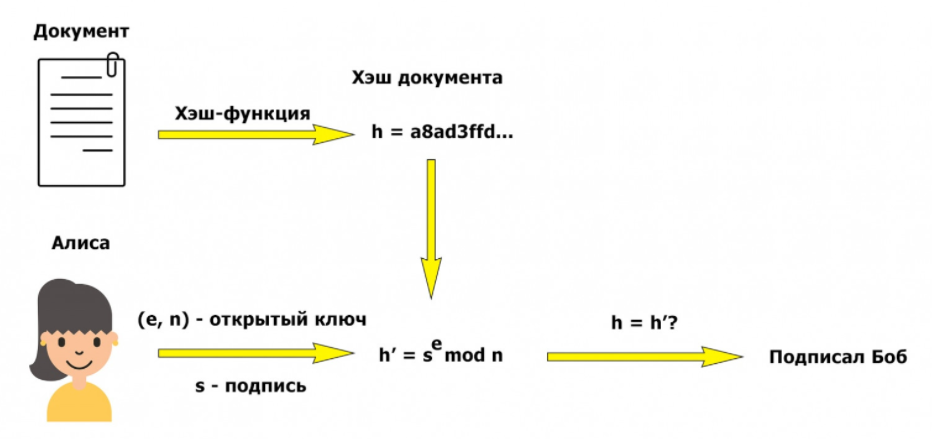
Рассмотренный алгоритм получения подписи изящен и прост в осознании, однако операция возведения в степень несколько "мешается". Наша текущая задача – подписать объёмный документ. Чтобы сэкономить время, мы не будем подписывать содержимое документа, а прибегнем к помощи *хэш-функций*. Выходная последовательность хэш-функции имеет небольшую (по сравнению с размером ключей) длину, а также по имеющемуся хэшу нельзя однозначно восстановить исходные данные.

На картинках наглядно показано, в какой момент мы используем хэширование.

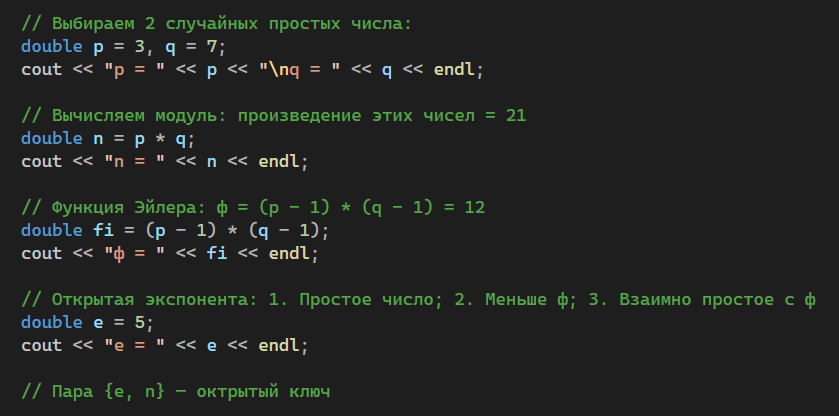
Создание подписи:

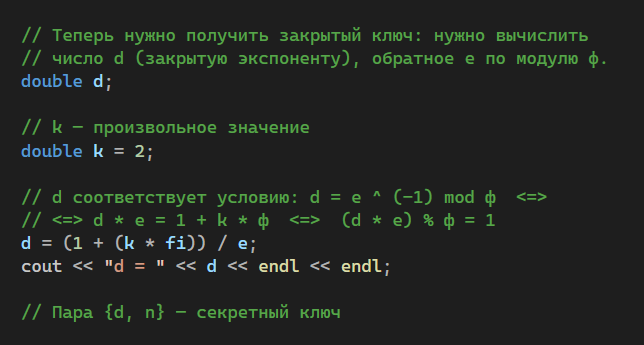


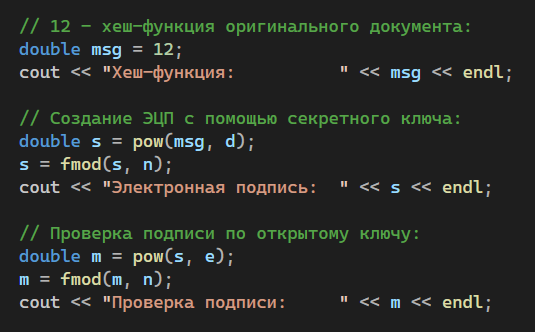
Проверка подписи:



**Создание ЭЦП с применением RSA на C++**







**Ответы на контрольные вопросы**

1. Дайте определение понятию "электронная цифровая подпись".

Электронная цифровая подпись — информация в электронной форме, которая присоединена к другой информации в электронной форме (подписываемой информации) или иным образом связана с такой информацией и которая используется для определения лица, подписывающего информацию.

2. Опишите последовательность действий участников протокола при отправке и проверке ЭЦП.

1. Отправитель формирует хеш-функцию на основе оригинального документа по алгоритму, например, SHA-256;
2. Отправитель формирует ЭЦП с использованием своего закрытого ключа по формуле s = h ^ d mod n, где h – хеш-функция;
3. Получатель получает документ и также вычисляет на его основании его хеш-функцию;
4. Получатель проверяет ЭЦП по формуле h’ = s ^ e mod n, где s – полученная ЭЦП, e и n – открытый ключ;

3. Какой порядок использования ключей (открытый; закрытый) при отправке и проверке ЭЦП?

Закрытый ключ применяется при отправке ЭЦП, открытый – при проверке ЭЦП.

4. Опишите схему протокола ЭЦП на основе алгоритма RSA.

1. Отправитель формирует хеш-функцию на основе оригинального документа по алгоритму, например, SHA-256;
2. Отправитель формирует ЭЦП с использованием своего закрытого ключа по формуле s = h ^ d mod n, где h – хеш-функция;
3. Получатель получает документ и также вычисляет на его основании его хеш-функцию;
4. Получатель проверяет ЭЦП по формуле h’ = s ^ e mod n, где s – полученная ЭЦП, e и n – открытый ключ;

5. Перечислите специальные схемы ЭЦП.

RSA, DSA, Эль-Гамаля, Рабина, Шнорра, Диффи-Лампорта.