Задания СРС:

**Цель: у**глубление теоретических знаний и развитие практических навыков по обеспечению кибербезопасности, формирование компетенций в области защиты информации, освоение принципов функционирования систем защиты, а также анализ современных киберугроз и методов противодействия им.

**Лекция 3: Введение в криптографию с открытым ключом**

**Цель:**  
Познакомиться с основными принципами криптографии с открытым ключом, понять её отличия от симметричной криптографии и области применения.

**Задание:**

1. Изучить понятие криптографии с открытым ключом:  
   – Что такое открытый и закрытый ключ  
   – Принцип асимметричного шифрования  
   – Роль доверия и центров сертификации
2. Рассмотреть основные алгоритмы:  
   – RSA  
   – Диффи — Хеллман (обмен ключами)  
   – Эллиптические кривые (ECC)  
   – ElGamal
3. Выполнить практическую часть:  
   – Построить простую схему работы RSA (на небольших числах вручную или в Excel/Python)  
   – Привести пример, как происходит шифрование и дешифрование
4. Подготовить:  
   – Презентацию (8–10 слайдов) с примерами применения (например, HTTPS, цифровые подписи)  
   – Краткий письменный доклад (1 страница):  
   • Преимущества и недостатки криптографии с открытым ключом  
   • Где используется в реальной жизни

**Цель лекции**: первоначальное знакомство с принципами шифрования с открытым ключом.

### Предпосылки создания методов шифрования с открытым ключом и основные определения

При использовании шифрования с закрытым ключом возникают две достаточно серьезные проблемы. Первая проблема заключается в изготовлении секретных ключей и доставке их участникам информационного обмена. При большом количестве и территориальной распределенности участников информационного обмена, использующих каналы связи общего назначения, например, обычную или электронную почту, часто бывает сложно гарантировать *безопасность* доставки такого ключа и его подлинность. В ["Поточные шифры и генераторы псевдослучайных чисел. Часть 2"](https://www.intuit.ru/studies/courses/691/547/lecture/12385)учебного пособия проблема распределения ключей для симметричного шифрования была подробно рассмотрена.

Второй проблемой является обеспечение подлинности партнеров при электронном общении. Развитие деловой переписки и электронной коммерции требует наличия методов, при использовании которых невозможно было бы подменить кого-либо из участников обмена. Получатель корреспонденции должен иметь возможность удостовериться в подлинности документа, а создатель электронного послания должен быть в состоянии доказать свое авторство получателю или третьей стороне. Следовательно, электронные документы должны иметь аналог обычной подписи.

Многие криптографы работали над решением этих проблем, в результате чего во второй половине семидесятых годов ХХ века были разработаны принципиально новые подходы, позволяющие решить перечисленные выше (и некоторые другие) задачи. Основой послужило открытие так называемых **асимметричных криптоалгоритмов**, или методов, в которых процедуры прямого и обратного криптопреобразования выполняются на различных ключах и не имеют между собой очевидных и легко прослеживаемых связей, которые позволили бы по одному ключу определить другой. *Асимметричные алгоритмы* гораздо больше основаны на свойствах математических функций, чем алгоритмы симметричного шифрования, использующие в основном только *операции* перестановки и замены. Большой вклад в эти исследования внесли американские ученые У. Диффи (W. Diffie), Э. Хеллман (M. Hellman), Р. Меркль (R. Merkle). Они первыми предложили способы решения обеих задач, которые радикально отличаются от всех предыдущих подходов к шифрованию.

*Асимметричные алгоритмы* шифрования называются также **алгоритмами с открытым ключом**. В отличие от *алгоритмов симметричного шифрования* (алгоритмов шифрования с закрытым ключом), в которых для шифрования и расшифрования используется один и тот же *ключ*, в ассиметричных алгоритмах один *ключ* используется для шифрования, а другой, отличный от первого, – для расшифрования. Алгоритмы называются асимметричными, так как ключи шифрования и расшифрования разные, следовательно, отсутствует *симметрия* основных криптографических процессов. Один из двух ключей является **открытым** (*public key*) и может быть объявлен всем, а второй – **закрытым** (*private key*) и должен держаться в секрете. Какой из ключей, открытый или закрытый, используется для шифрования, а какой для расшифрования, определяется назначением криптографической системы.

В настоящее время *асимметричные алгоритмы* широко применяются на практике, например, для обеспечения информационной безопасности телекоммуникационных сетей, в том числе сетей, имеющих сложную топологию; для обеспечения информационной безопасности в глобальной сети *Internet*; в различных банковских и платежных системах (в том числе использующих *интеллектуальные карты*).

*Алгоритмы шифрования* с открытым ключом можно использовать для решения, как *минимум*, трех задач:

1. Для шифрования передаваемых и хранимых данных в целях их защиты от несанкционированного доступа.
2. Для формирования цифровой подписи под электронными документами.
3. Для распределения секретных ключей, используемых потом при шифровании документов симметричными методами.

### Односторонние функции

Все *алгоритмы шифрования* с открытым ключом основаны на использовании так называемых *односторонних функций*. **Односторонней функцией** (*one-way* *function*) называется *математическая функция*, которую относительно легко вычислить, но трудно найти по значению функции соответствующее *значение* аргумента. То есть, зная х легко вычислить f(x), но по известному f(x) трудно найти подходящее *значение* x. Под словом "трудно вычислить" понимают, что для этого потребуется не один год расчетов с использованием ЭВМ. *Односторонние функции* применяются в криптографии также в качестве хеш-функций (см. ["Криптографические хеш-функции"](https://www.intuit.ru/studies/courses/691/547/lecture/12381)). Использовать *односторонние функции* для шифрования сообщений с целью их защиты не имеет смысла, так как обратно расшифровать зашифрованное сообщение уже не получится. Для целей шифрования используются специальные *односторонние функции* – **односторонние функции с люком** (или с секретом) – это особый вид *односторонних функций*, имеющих некоторый секрет (*люк*), позволяющий относительно быстро вычислить обратное *значение* функции.

Для *односторонней функции* с люком f справедливы следующие утверждения:

1. зная х, легко вычислить f(x),
2. по известному значению f(x) трудно найти x,
3. зная дополнительно некоторую секретную информацию, можно легко вычислить x.

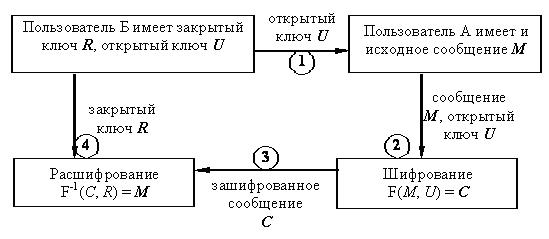
### Использование асимметричных алгоритмов для шифрования

В 70-х годах ХХ века Диффи и Хеллман предложили принцип шифрования, основанный на использовании двух разных ключей, хотя и связанных между собой, но устроенных так, что вычислить по одному из них (открытому) другой (закрытый) практически невозможно. Этот принцип может быть использован для решения проблемы снабжения пользователей ключами шифрования/расшифрования, а точнее – для устранения этой проблемы. Согласно Диффи и Хеллману предварительно распределяемые закрытые ключи вообще не должны использоваться для шифрования данных (так как секрет, который известен более чем одному человеку, – уже не секрет). *Закрытый ключ* должен быть известен только одному лицу – его владельцу. Такой принцип использования *асимметричных алгоритмов* получил название *открытого шифрования* или *шифрованием с открытым ключом*.

Согласно этому принципу, любой желающий может зашифровать сообщение открытым ключом. Расшифровать сообщение сможет только владелец закрытого ключа. Пусть, например, пользователи А и Б, имеющие возможность обмениваться электронными сообщениями, используют схему открытого шифрования. Предположим, *пользователь* А должен передать секретное сообщение пользователю Б так, чтобы никто другой не смог его прочитать. Для этого необходимо выполнить следующие действия:

1. Пользователь Б посылает пользователю А свой открытый ключ U по любому каналу связи, например, по электронной почте.
2. Пользователь А шифрует свое сообщение М полученным открытым ключом U и получает зашифрованное сообщение С.
3. Зашифрованное сообщение С пересылается пользователю Б.
4. Пользователь Б расшифровывает полученное сообщение С своим закрытым ключом R.

Если операцию шифрования обозначить как F, а операцию расшифрования как F-1, то схему протокола обмена информацией между пользователями можно изобразить, как на [рис. 9.1](https://www.intuit.ru/studies/courses/691/547/lecture/12387?page=1" \l "image.9.1).



**Рис. 9.1.**Схема открытого шифрования

Использование открытого шифрования снимает проблему распределения ключей. Раньше пользователи перед обменом зашифрованными данными должны были каким-либо образом по закрытому каналу связи согласовывать используемый *секретный ключ*. Для этого они могли встретиться лично или использовать курьера. Если один из пользователей считал нужный изменить *ключ*, он должен был передать новый *ключ* своему абоненту. *Криптография* с открытыми ключами все упрощает. Теперь абоненты не должны заботиться о возможности компрометации секретного ключа. Пользователи системы связи могут совершенно свободно обмениваться открытыми ключами и зашифрованными ими сообщениями. Если *пользователь* надежно хранит свой *закрытый ключ*, никто не сможет прочитать передаваемые сообщения.

Для упрощения процедуры обмена в сети передачи сообщений обычно используется *база данных* (подробнее об этом см. в ["Криптографические алгоритмы с открытым ключом и их использование"](https://www.intuit.ru/studies/courses/691/547/lecture/12391)), в которой хранятся открытые ключи всех пользователей. При необходимости любой *пользователь* системы может запросить из базы *открытый ключ* другого человека и использовать полученный *ключ* для шифрования сообщений.

### Цифровая подпись на основе алгоритмов с открытым ключом

Как и все люди, абоненты *сети передачи данных* могут не доверять друг другу или вести себя нечестно. Они могут подделывать чужие сообщения, отрицать свое авторство или выдавать себя за другое лицо. Особенно актуальными становятся эти проблемы в связи с развитием электронной коммерции и возможностью оплаты услуг через *Интернет*. Поэтому во многих системах связи получатель корреспонденции должен иметь возможность удостовериться в подлинности документа, а создатель электронного послания должен быть в состоянии доказать свое авторство получателю или третьей стороне. Следовательно, электронные документы должны иметь аналог обычной физической подписи. При этом подпись должна обладать следующими свойствами:

1. подпись воспроизводится только одним лицом, а подлинность ее может быть удостоверена многими;
2. подпись неразрывно связывается с данным сообщением и не может быть перенесена на другой документ;
3. после того, как документ подписан, его невозможно изменить;
4. от поставленной подписи невозможно отказаться, то есть лицо, подписавшее документ, не сможет потом утверждать, что не ставило подпись.

*Асимметричные алгоритмы* шифрования могут быть использованы для формирования **цифровой (электронной) подписи** (*digital signature*) – уникального числового дополнения к передаваемой информации, позволяющего проверить ее авторство. *Электронная (цифровая) подпись* (*ЭЦП*) представляет собой последовательность *бит* фиксированной длины, которая вычисляется определенным образом с помощью содержимого подписываемой информации и секретного ключа.

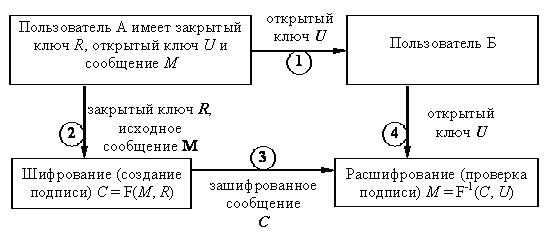
При формировании цифровой подписи специальным образом шифруется или все сообщение целиком, или результат вычисления хеш-функции от сообщения. Последний способ обычно оказывается предпочтительнее, так как подписываемое сообщение может иметь разный размер, иногда довольно большой, а хеш-код всегда имеет постоянную не очень большую длину. Рассмотрим подробнее оба варианта формирования *ЭЦП*.

Самый простой способ основывается, так же как и при открытом шифровании, на использовании пары связанных между собой ключей (открытого и закрытого). Однако роли закрытого и открытого ключей меняются – *ключ* подписывания становится секретным, а *ключ* проверки – открытым. Если при этом сохраняется свойство, что по открытому ключу нельзя практически найти *закрытый ключ*, то в качестве подписи может выступать само сообщение, зашифрованное секретным ключом. Таким образом подписать сообщение может только владелец закрытого ключа, но каждый, кто имеет его *открытый ключ*, может проверить подпись.

Пусть, например, *пользователь* А хочет отправить пользователю Б подписанное сообщение. Процедура создания и проверки подписи состоит из следующих шагов:

1. Пользователь А посылает пользователю Б свой открытый ключ U по любому каналу связи, например, по электронной почте.
2. Пользователь А шифрует сообщение М своим закрытым ключом R и получает зашифрованное сообщение С.
3. Зашифрованное сообщение пересылается пользователю Б.
4. Пользователь Б расшифровывает полученное сообщение С, используя открытый ключ пользователя А. Если сообщение расшифровалось, значит, оно подписано пользователем А.

Этот протокол можно изобразить в виде схемы, как на [рис. 9.2](https://www.intuit.ru/studies/courses/691/547/lecture/12387?page=2" \l "image.9.2).



**Рис. 9.2.**Первый вариант схемы создания и проверки цифровой подписи

До тех пор, пока *пользователь* А надежно хранит свой *закрытый ключ*, его подписи достоверны. Кроме того, невозможно изменить сообщение, не имея доступа к закрытому ключу абонента А; тем самым обеспечивается *аутентичность* и *целостность* данных.

Физическое *представление* пары ключей зависит от конкретной системы, поддерживающей использование *ЭЦП*. Чаще всего *ключ* записывается в *файл*, который, в *дополнение* к самому ключу, может содержать, например, информацию о пользователе - владельце ключа, о сроке действия ключа, а также некий набор данных, необходимых для работы конкретной системы (подробнее об этом см. ["Электронная цифровая подпись"](https://www.intuit.ru/studies/courses/691/547/lecture/12393)). Данные о владельце ключа позволяют реализовать другую важную функцию *ЭЦП* - установление авторства, поскольку при проверке подписи сразу же становится ясно, кто подписал то или иное сообщение. Обычно программные продукты, осуществляющие *проверку ЭЦП*, настраиваются так, чтобы результат исполнения появлялся на экране в удобном для восприятия виде с указанием поставившего подпись пользователя, например, так:

"Подпись файла приказ.doc верна (Автор: Соколов А.И.)"

На [рис. 9.2](https://www.intuit.ru/studies/courses/691/547/lecture/12387?page=2" \l "image.9.2)представлена схема формирования так называемой **цифровой подписи с восстановлением документа**. Цифровые подписи с восстановлением документа как бы содержат в себе подписываемый документ: в процессе проверки подписи автоматически вычисляется и *тело документа*. Если при расшифровывании сообщение восстановилось правильно, значит, подпись была верной. *Цифровая подпись* с восстановлением документа может быть реализована, например, с помощью одного из самых популярных алгоритмов формирования *ЭЦП* – *RSA*.

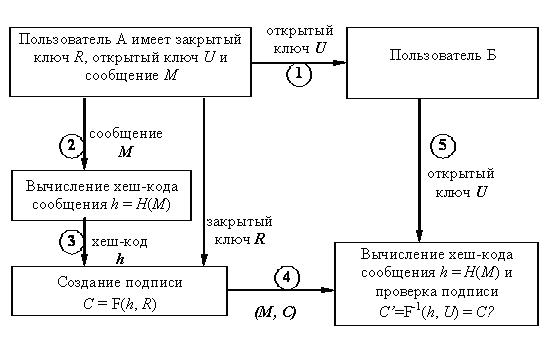
В случае использования цифровой подписи с восстановлением документа все сообщение целиком подписывается, то есть шифруется. В настоящее время на практике так обычно не делается. *Алгоритмы шифрования* с открытым ключом достаточно медленные, кроме того, для подтверждения целостности сообщения требуется много памяти. К тому же практически все применяемые алгоритмы вычисления *ЭЦП* используют для расчета сообщения заранее заданной стандартной длины. Например, в российском алгоритме формирования цифровой подписи ГОСТ Р34.10-94 этот размер определен равным 32 байтам. Поэтому для экономии времени и вычислительных ресурсов, а также для удобства работы асимметричный *алгоритм* обычно используется вместе с какой-нибудь однонаправленной хеш-функцией. В этом случае вначале с помощью хеш-функции из сообщения произвольной длины вычисляется хеш-код нужного размера, а затем для вычисления *ЭЦП* производится *шифрование* полученного на предыдущем этапе хеш-кода от сообщения.

*ЭЦП*, вычисленные по хеш-коду документа, называют **присоединяемыми цифровыми подписями**. Такие цифровые подписи представляют собой некоторый числовой код, который необходимо пристыковывать к подписываемому документу. Само сообщение при этом не шифруется и передается в открытом виде вместе с цифровой подписью отправителя.

Если *пользователь* А хочет отправить пользователю Б сообщение М, дополненное присоединенной цифровой подписью, то процедура создания и проверки подписи должна состоять из следующих шагов:

1. Пользователь А посылает пользователю Б свой открытый ключ U по любому каналу связи, например, по электронной почте.
2. Пользователь А с помощью некоторой надежной хеш-функции Н вычисляет хеш-код своего сообщения h = H(M).
3. Затем пользователь А шифрует хеш-код сообщения h своим закрытым ключом R и получает цифровую подпись С.
4. Исходное сообщение М вместе с цифровой подписью С пересылаются пользователю Б.
5. Пользователь Б вычисляет хеш-код h полученного сообщения М, а затем проверяет цифровую подпись С, используя открытый ключ пользователя А.

Этот протокол можно изобразить в виде схемы, как на [рис. 9.3](https://www.intuit.ru/studies/courses/691/547/lecture/12387?page=2" \l "image.9.3).



**Рис. 9.3.**Второй вариант схемы создания и проверки цифровой подписи

Хеш-*функция* не являются частью алгоритма *ЭЦП*, поэтому в схеме может быть использована любая надёжная хеш-*функция*.

Описанный процесс создания подписи не обеспечивает *конфиденциальность*. То есть сообщение, посланное таким способом, невозможно изменить, но можно прочитать. Даже если не использовать хеш-функцию, а шифровать все сообщение целиком, *конфиденциальность* не обеспечивается, так как любой может расшифровать сообщение, используя *открытый ключ* отправителя.

Во многих ситуациях приведенной схемы создания и использования цифровой подписи оказывается вполне достаточно. Однако бывают случаи, когда *пользователь* Б может смошенничать. Предположим, что пересылаемым документом был чек от пользователя А, например, за оказанные услуги. *Пользователь* Б удостоверился, что *цифровая подпись* на нем верная и использовал его для получения денег. Никто не может помешать пользователю Б снять одну или несколько копий с подписанного документа (тем более, если документ электронный) и периодически с небольшим интервалом предъявлять их в банк для получения денег.

Для устранения возможности такого жульничества в цифровые подписи часто включают *метки времени*. Дата и время подписания документа добавляются к сообщению и подписываются вместе со всем документом. При оплате чека *метка* времени может быть зафиксирована банком и занесена в базу данных. При попытке повторного предъявления чека банк это обнаружит и примет соответствующие меры.

Разновидностью цифровой подписи является *неотрицаемая цифровая подпись*. Как и обычная *цифровая подпись*, неотрицаемая *цифровая подпись* зависит от подписываемого документа и закрытого ключа автора. В отличие от обычной *ЭЦП* неотрицаемая подпись не может быть проверена без разрешения подписавшего. Таким образом, получатель корреспонденции не сможет показать подпись (или не сможет доказать правильность подписи) без согласия лица, подписавшего сообщение.

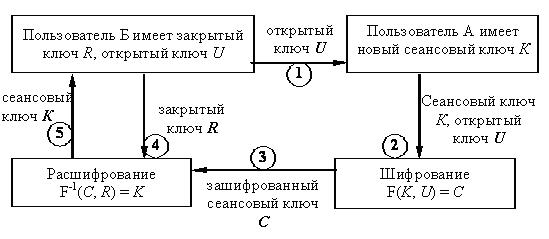
### Формирование секретных ключей с использованием асимметричных алгоритмов

На практике алгоритмы с открытым ключом редко используются для непосредственного шифрования сообщений. Этому препятствует относительная невысокая скорость *асимметричных алгоритмов* при шифровании и расшифровании больших объемов данных. Данный фактор связан с тем, что основной операцией в системах с открытым ключом является возведение в степень по большому модулю 500-1000 битовых чисел, что при программной реализации производится намного медленнее, чем *шифрование* того же объема данных классическими симметричными способами. Однако при обработке коротких блоков данных, например, ключей определенной длины, *алгоритмы шифрования* с открытым ключом могут использоваться достаточно эффективно. Поэтому часто используют следующую комбинированную схему: асимметричный *алгоритм* применяется для согласования *ключа сессии*, а затем этот *ключ* выступает в роли секретного ключа для шифрования сообщений *симметричным алгоритмом*.

Простейший протокол формирования секретного *ключа сессии* может выглядеть следующим образом (если пользователи некоторой системы связи имеют *доступ* к базе данных открытых ключей абонентов системы, предоставляемой центром распределения ключей, то они могут получать из нее открытые ключи друг друга):

1. Пользователь А получает открытый ключ пользователя Б из *центра распределения ключей* или непосредственно от пользователя Б.
2. Пользователь А генерирует случайный сеансовый ключ и зашифровывает его полученным открытым ключом.
3. Зашифрованный сеансовый ключ пересылается пользователю Б.
4. Пользователь Б расшифровывает полученный пакет своим закрытым ключом.
5. Пользователи А и Б используют согласованный сеансовый ключ для обмена шифрованными сообщениями.

Схему формирования парой пользователей А и В общего секретного ключа К для шифрования - расшифрования можно изобразить следующим образом ([рис. 9.4](https://www.intuit.ru/studies/courses/691/547/lecture/12387?page=3" \l "image.9.4)).



**Рис. 9.4.**Схема формирования общего секретного ключа

Эта схема напоминает [рис. 9.1](https://www.intuit.ru/studies/courses/691/547/lecture/12387?page=1" \l "image.9.1), и это неудивительно, ведь в ней используется тот же самый режим шифрования открытым ключом. Отличие заключается в том, *что* шифруется. В схеме на [рис. 9.4](https://www.intuit.ru/studies/courses/691/547/lecture/12387?page=3" \l "image.9.4) производится *шифрование* небольшого по размеру сеансового ключа, который в дальнейшем будет использован в качестве секретного ключа при шифровании *симметричным алгоритмом*. *Шифрование* небольшого по размеру блока данных выполняется достаточно быстро и не замедляет телекоммуникационные процессы даже в системе с многими тысячами пользователей.

Существуют и более сложные протоколы распределения ключей, обеспечивающие взаимное подтверждение подлинности участников сеанса связи, подтверждение достоверности сеанса *механизмом запроса*-ответа или другие требования.

### Требования к алгоритмам шифрования с открытым ключом

Рассмотрев основные способы применения алгоритмов шифрования с открытым ключом, изучим требования, которым должен, по мнению основоположников теории шифрования с открытым ключом Диффи и Хеллмана, удовлетворять *алгоритм* шифрования с открытым ключом. Эти требования следующие:

1. Вычислительно легко создавать пару (открытый ключ, закрытый ключ).
2. Вычислительно легко зашифровать сообщение открытым ключом.
3. Вычислительно легко расшифровать сообщение, используя закрытый ключ.
4. Вычислительно невозможно, зная открытый ключ, определить соответствующий закрытый ключ.
5. Вычислительно невозможно, зная только открытый ключ и зашифрованное сообщение, восстановить исходное сообщение.

Из этих общих требований видно, что реализация конкретного алгоритма с открытым ключом зависит от соответствующей *односторонней функции*.

Математиками и криптографами предложено большое количество алгоритмов шифрования с открытым ключом, основанных на различных односторонних функциях. Некоторые алгоритмы можно задействовать тремя, рассмотренными ранее в данной лекции способами, в то время как другие могут использоваться только одним или двумя способами. Мы рассмотрим четыре алгоритма с открытым ключом, три из которых достаточно давно применяются на практике, а четвертый вид алгоритмов совсем недавно начал применяться в системах защиты информации. Эти алгоритмы используются обычно для различных целей, что отражено в следующей таблице.

|  |  |  |  |
| --- | --- | --- | --- |
| **Название алгоритма** | **Возможность использования** | | |
| Шифрование / расшифрование данных | Цифровая подпись | Согласование или формирование ключа |
| RSA | Да | Да | Да |
| *Алгоритм Диффи-Хеллмана* | Нет | Нет | Да |
| Алгоритм Эль-Гамаля | Да | Да | Да |
| Алгоритмы с использованием эллиптических кривых | Да | Да | Да |

Перед изучением наиболее известных алгоритмов шифрования с открытым ключом, необходимо напомнить, что все *асимметричные алгоритмы* основаны на свойствах тех или иных математических функций. Доказательства правильности работы рассматриваемых алгоритмов могут быть достаточно сложными, поэтому мы ограничимся изучением основных принципов их работы. Многие криптографические алгоритмы базируются на результатах классической теории чисел. Основные факты и положения этой теории, необходимые для понимания алгоритмов и выполнения упражнений, сформулированы в ["Основные положения теории чисел, используемые в криптографии с открытым ключом"](https://www.intuit.ru/studies/courses/691/547/lecture/12389).

### Ключевые термины

**Алгоритм шифрования с открытым ключом** (или **асимметричные криптоалгоритмы** ) – криптографический *алгоритм*, в котором для шифрования и расшифрования используются разные ключи.

**Закрытый ключ** – *ключ*, используемый в асимметричных криптографических алгоритмах, который должен храниться в секрете.

**Односторонняя функция** – *математическая функция*, которую относительно легко вычислить, но трудно найти по значению функции соответствующее *значение* аргумента. То есть, зная х легко вычислить f(x), но по известному f(x) трудно найти подходящее *значение* x.

**Односторонняя функция с люком** (или **с секретом** ) – это особый вид *односторонних функций*, имеющих некоторый секрет (*люк*), позволяющий относительно быстро вычислить обратное *значение* функции.

**Открытый ключ** – *ключ*, используемый в асимметричных криптографических алгоритмах, который может не храниться в секрете.

**Присоединяемые цифровые подписи** – подписи, вычисленные по хеш-коду документа. Такие цифровые подписи представляют собой некоторый числовой код, который необходимо пристыковывать к подписываемому документу. Само сообщение при этом не шифруется и передается в открытом виде вместе с цифровой подписью отправителя.

**Цифровая (электронная) подпись** (*digital signature*) – уникальное числовое *дополнение* к передаваемой информации, позволяющее проверить ее авторство. *Электронная (цифровая) подпись* (*ЭЦП*) представляет собой последовательность *бит* фиксированной длины, которая вычисляется определенным образом с помощью содержимого подписываемой информации и секретного ключа.

**Цифровые подписи с восстановлением документа** – подписи, которые как бы содержат в себе подписываемый документ: в процессе проверки подписи автоматически вычисляется и *тело документа*. Если при расшифровывании сообщение восстановилось правильно, значит, подпись была верной.

### Краткие итоги

*Асимметричные алгоритмы* шифрования (или алгоритмы с открытым ключом) – криптографические алгоритмы, в которых один *ключ* используется для шифрования, а другой, отличный от первого, – для расшифрования. Алгоритмы называются асимметричными, так как ключи шифрования и расшифрования разные, следовательно, отсутствует *симметрия* основных криптографических процессов. Один из двух ключей является открытым (*public key*) и может быть объявлен всем, а второй – закрытым (*private key*) и должен держаться в секрете. Какой из ключей, открытый или закрытый, используется для шифрования, а какой для расшифрования, определяется назначением криптографической системы.

Все *алгоритмы шифрования* с открытым ключом основаны на использовании *односторонних функций*. Односторонней функцией называется *математическая функция*, которую относительно легко вычислить, но трудно найти по значению функции соответствующее *значение* аргумента. Использовать *односторонние функции* для шифрования сообщений с целью их защиты не имеет смысла, так как обратно расшифровать зашифрованное сообщение уже не получится. Для целей шифрования используются *односторонние функции с люком* (или с секретом) – особый вид *односторонних функций*, имеющих некоторый секрет (*люк*), позволяющий относительно быстро вычислить обратное *значение* функции.

*Алгоритмы шифрования* с открытым ключом можно использовать для решения следующих задач:

1. Для шифрования передаваемых и хранимых данных в целях их защиты от несанкционированного доступа.
2. Для формирования цифровой подписи под электронными документами.
3. Для распределения секретных ключей, используемых потом при шифровании документов симметричными методами.

Цифровая (электронная) подпись – уникальное числовое *дополнение* к передаваемой информации, позволяющее проверить ее авторство. *Электронная (цифровая) подпись* (*ЭЦП*) представляет собой последовательность *бит* фиксированной длины, которая вычисляется определенным образом с помощью содержимого подписываемой информации и секретного ключа. Различают присоединяемые цифровые подписи и цифровые подписи с восстановлением документа. Присоединяемые цифровые подписи – подписи, вычисленные по хеш-коду документа. Такие цифровые подписи представляют собой некоторый числовой код, который необходимо пристыковывать к подписываемому документу. Само сообщение при этом не шифруется и передается в открытом виде вместе с цифровой подписью отправителя. Цифровые подписи с восстановлением документа – подписи, которые как бы содержат в себе подписываемый документ: в процессе проверки подписи автоматически вычисляется и *тело документа*. Если при расшифровывании сообщение восстановилось правильно, значит, подпись была верной.

#### Вопросы для самопроверки

1. Чем *асимметричные алгоритмы* шифрования отличаются от симметричных?
2. Для решения каких задач могут на практике применяться алгоритмы шифрования с открытым ключом?
3. Какие математические функции называются односторонними? Для чего они могут применяться в криптографии?
4. Что такое цифровая подпись?
5. Каков алгоритм формирования цифровой подписи при использовании алгоритмов шифрования с открытым ключом?
6. Каким образом алгоритмы шифрования с открытым ключом могут использоваться для формирования общего секретного ключа у группы пользователей?
7. Какие требования предъявляются к асимметричным алгоритмам?