**Шифрование** – это технология кодирования и раскодирования данных.

**Шифрование** — это процесс кодирования информации с целью предотвращения несанкционированного доступа. В случае кражи или утечки зашифрованные данные будут недоступны для прочтения без соответствующего **ключа.**

Зашифрованные данные — это результат применения алгоритма для кодирования данных с целью сделать их недоступными для чтения. Данные могут быть раскодированы в исходную форму только путем применения специальный ключа. Шифрование является важной частью обеспечения безопасности данных, поскольку оно защищает конфиденциальную информацию от угроз, в числе которых использование вредоносного ПО и несанкционированный доступ третьих сторон. Шифрование данных — это универсальное защитное решение: оно может применяться к части данных, например, к паролю, к информации в файле или даже ко всем данным, содержащимся на носителе.

*Как шифрование данных защищает пользователей компьютеров*

Обычно сталкиваетесь с шифрованием данных, когда просят ввести личную идентификационную информацию в веб-форму. Сайты финансовых, правительственных, образовательных и торговых организаций обычно шифруют ваши данные, чтобы защитить их от краж и мошенничества. На то, что веб-формы защищены и что ваши данные будут зашифрованы, вам укажет следующее:

***- Адрес веб-страницы начинается с "https": это означает, что ваши данные будут зашифрованы и переданы с использованием защищенного протокола.***

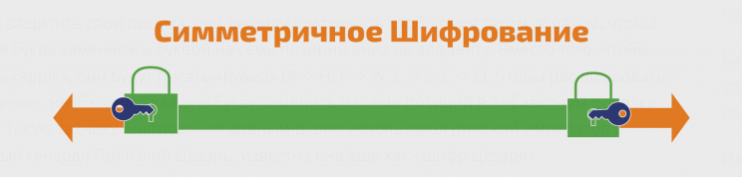
**GDPR - Общий регламент по защите данных.**

Согласно **GDPR** к персональной информации относятся имена и фамилии, фотографии, электронные адреса, телефонные номера, номера банковских счетов, отпечатки пальцев и голоса. Общий регламент о защите данных (GDPR), который действует во всех странах-членах ЕС с 25 мая 2018 года, описывает шифрование как защиту от репутационных рисков.

Представьте, что один из сотрудников теряет USB-ключ со списком клиентов предприятия. Согласно GDPR, компания должна сообщить возможным жертвам из списка об инциденте. Для некоторых пользователей это может стать причиной смены поставщика.

Два основных типа шифрования — **симметричное и асимметричное.**

Метод симметричного шифрования, как и следует из названия, использует один криптографический ключ для шифрования и дешифрования данных. Использование одного ключа для обеих операций делает процесс простым. Давайте разберемся с процессом симметричного шифрования на простом примере:



В Киеве живут два близких друга Антон и Алиса. По какой-то причине Алиса вынуждена покинуть город. Единственный способ общаться друг с другом — по почте. Но есть одна проблема: Антон и Алиса боятся, что кто-то сможет прочитать их письма.

Чтобы защитить свои письма, они решили шифровать сообщения таким образом, чтобы каждая буква заменялась буквой на семь позиций вниз по алфавиту. Вместо того, чтобы писать «Apple», они будут писать «hwwsl» (A -> H, P -> W, L -> S, E -> L). Чтобы расшифровать сообщение, необходимо каждую букву заменить на семь позиций в алфавитном порядке назад. Такую технику шифрования давным-давно использовал римский император и военный генерал Гай Юлий Цезарь, известна она еще как «шифр Цезаря».

«+»:

* Значительно быстрее, чем их аналоги асимметричного шифрования (о чем мы вскоре поговорим);
* Требуется меньше вычислительной мощности;
* Не снижается скорость интернета.

Существуют сотни алгоритмов симметричного типа! Наиболее распространенные из них — **AES, RC4, DES, 3DES, RC5, RC6** и т. д. Давайте рассмотрим три самых популярных.

**Асимметричное шифрование**

Асимметричное шифрование, в отличие от симметричного, включает в себя несколько ключей для шифрования и дешифрования данных, которые математически связаны друг с другом. Один из этих ключей известен как «открытый ключ», а другой — как «закрытый ключ». Асимметричный метод шифрования также известен как «криптография с открытым ключом».

Симметричное шифрование прекрасно работает, когда наши Алиса и Антон хотели обмениваться информацией между собой. Но что, если Антон хочет безопасно общаться с сотнями людей? Непрактично и неудобно использовать разные ключи для каждого собеседника.

Чтобы решить эту проблему, Антон использует шифрование с открытым ключом, т.е. он дает открытый ключ каждому, кто отправляет ему информацию, а секретный ключ он хранит при себе. Он поручает им зашифровать информацию с помощью открытого ключа, чтобы данные можно было расшифровать только с помощью его личного ключа. Это исключает риск компрометации ключа, поскольку данные могут быть расшифрованы только с использованием закрытого ключа, который есть у Антона.

**Что делает асимметричное шифрование отличной техникой**

* Первое (и наиболее очевидное) преимущество этого типа шифрования — безопасность, которую он обеспечивает. В этом методе открытый ключ — который является общедоступным — используется для шифрования данных, в то время как расшифровка данных выполняется с использованием закрытого ключа, который необходимо надежно хранить. Это гарантирует, что данные остаются защищенными от атак «человек посередине» (MiTM). Для веб-серверов и серверов электронной почты, которые постоянно подключаются к сотням тысяч клиентов нужно управлять только одним ключом и защищать его. Другой ключевой момент заключается в том, что криптография с открытым ключом позволяет создавать зашифрованное соединение без необходимости встречаться в автономном режиме, чтобы сначала обменяться ключами.
* Вторая важная особенность, которую предлагает асимметричное шифрование, — это аутентификация. Как мы видели, данные, зашифрованные с помощью открытого ключа, могут быть расшифрованы только с помощью закрытого ключа, связанного с ним. Следовательно, он гарантирует, что данные видит и дешифрует только тот объект, который должен их получить. Проще говоря, это подтверждает, что вы разговариваете или обмениваетесь информацией с реальным человеком или организацией.

**2 основных типа алгоритмов асимметричного шифрования**

**1. Алгоритм асимметричного шифрования RSA**

В 1977 году алгоритм изобрели трое ученых из Массачусетского технологического института Рон Ривест, Ади Шамир и Леонард Адлеман (Ron Rivest, Adi Shamir, and Leonard Adleman отсюда «RSA»). ***На сегодняшний день является наиболее используемым алгоритмом асимметричного шифрования. Его эффективность заключается в методе «первичной факторизации». По сути, выбираются два различных случайных простых числа заданного размера (например, 1024 бита каждое) и умножаются, чтобы создать еще одно гигантское число. Задача состоит в том, чтобы определить исходные простые числа из умноженного гигантского. Оказывается, эта головоломка практически невозможна для современных суперкомпьютеров, не говоря уже о людях.***

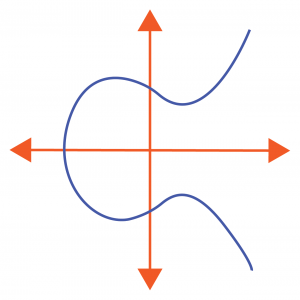
В 2010 году группа добровольцев провела исследование, и им потребовалось более 1500 лет вычислительного времени (распределенного по сотням компьютеров), чтобы взломать 768-битный ключ RSA, что намного ниже стандартного 2048-битного, который используется сегодня.

Преимущество использования алгоритма шифрования RSA

**2. Алгоритм асимметричного шифрования ECC**

В 1985 году два математика по имени Нил Коблиц и Виктор Миллер предложили использовать эллиптические кривые в криптографии. Спустя почти два десятилетия их идея воплотилась в реальность, алгоритм ECC (Elliptic Curve Cryptography) начали использовать в 2004-2005 годах.

В процессе шифрования ECC эллиптическая кривая представляет набор точек, которые удовлетворяют математическое уравнение (y 2 = x 3 + ax + b).



Эллиптическая кривая

Как и RSA, ECC также работает по принципу необратимости. Проще говоря, в ECC число, символизирующее точку на кривой, умножается на другое число и дает другую точку на кривой. Теперь, чтобы взломать эту головоломку, вы должны выяснить новую точку на кривой. Математика ECC построена таким образом, что найти новую точку практически невозможно, даже если вы знаете исходную точку.

Под катом:

* Шифр Цезаря
* Шифр пар
* Шифр четырех квадратов
* Матричный шифр
* Шифр ADFGX
* Шифр Виженера

#### **Шифр Цезаря**

Каждую букву заменяют на третью (или N-ю) по алфавиту после нее.

а б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я  
б в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а  
в г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б  
г д е ё ж з и й к л м н о п р с т у ф х ц ч ш щ ъ ы ь э ю я а б в  
...

**Сообщение**: хабрахабр

**Шифр**: шгдугшгду

#### **Шифр пар**

Алфавит случайным образом записывают в 2 строки, и шифрования текста происходит заменой буквы на соседнюю ей по вертикали.

Пример таблицы:

в ы э ё у ж и ф м н о ъ к п а р я  
е т х й ш ю д ь б щ з с ц г ч л -

**Сообщение**: хабрахабр

**Шифр**: эчмлчэчмл

#### **Шифр четырех квадратов**

Строят 4 квадрата размера 5х5 или 4х4 (в зависимости от объема алфавита). В каждый из них случайным образом записывают алфавит.



Шифрование: Выбирая по 2 буквы сообщения, находим их соответственно в левом верхнем и правом нижнем квадрате.

Для шифра используем 2 буквы, формирующих с выбранными вершины прямоугольника (слева направо)

**Сообщение**: хабрахабр (ха бр ах аб рр)

**Шифр**: днедлягозн (дн ед ля го зн)

#### **Матричный шифр**

Подготавливаем матрицу *MxN*.

Подготавливаем два ключа *KM* и *KN*.

1) Сообщение по строкам записываем в матрицу *MxN*.

2) По краям матрицы записываем ключи.

3) Строки переставляем так, чтобы буквы ключа стали упорядоченными по алфавиту.

То же делаем со столбцами.

4) Списываем буквы из матрицы по столбцам.

Размеры матрицы: *M*=4, *N*=5

Ключ *KM*: пуля, *KN*: дурак.



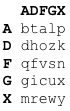
**Сообщение**: заголовок должен быть наполнен смыслом

**Шифр**: еокьозобнлднжготлавымлмосаеооныооосолпно

#### **Шифр ADFGX**

В матрицу 5х5 случайным образом записывают (определенным образом) буквы латинского алфавита

Используем слово-ключ (например, *formula*).



**Сообщение**: one two three

1) каждой букве сообщения ставим в соответствие 2 буквы (по таблице).

https://habrastorage.org/r/w1560/storage/3b093b16/d92ba6f3/7458e11a/77979112.png

2) полученный текст записываем под ключом:



3) переставляем столбцы так, чтобы буквы ключа упорядочились по алфавиту:



4) записываем буквы из полученной матрицы (по столбцам) в шифр.

**Сообщение**: one two three

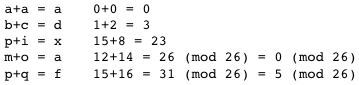
**Шифр**: ADXF DDDF FAFX XDDX FXDX FGXF XFXF

#### **Шифр Виженера**

На алфавите длиной *N* вводят операцию добавления (циклического сдвига) букв.

https://habrastorage.org/r/w1560/storage/84da6809/907a012b/7e716624/1cedb899.png

Пронумеровав буквы, добавляем их по модулю *N* (для англ. алфавита *N*=26).



Где mod , это математическая операция

Выбираем слово-ключ (пускай *pass*) и подписываем его под сообщением сколько нужно раз

https://habrastorage.org/r/w1560/storage/3b88a722/377e3435/54aa30a6/0cbc3ecb.png

**Сообщение**: search twitter  
**Шифр**: hesjrhloxtlwg

**Задание**

Сообщения для шифрования различными методами. Вы можете использовать их для практики шифрования и дешифрования:

1. Оригинальное сообщение: "HELLO, WORLD!" Ключ шифра: 3
2. Оригинальное сообщение: "GOOD MORNING" Ключ шифра: 7
3. Оригинальное сообщение: "ENCRYPT ME" Ключ шифра: 5
4. Оригинальное сообщение: "SECRET CODE" Ключ шифра: 8
5. Оригинальное сообщение: "SUNSHINE" Ключ шифра: 1
6. Оригинальное сообщение: "COMPUTER SCIENCE" Ключ шифра: 10
7. Оригинальное сообщение: "CONFIDENTIAL" Ключ шифра: 4
8. Оригинальное сообщение: "WELCOME BACK" Ключ шифра: 6
9. Оригинальное сообщение: "SECURITY" Ключ шифра: 13
10. Оригинальное сообщение: "PRIVACY MATTERS" Ключ шифра: 9