МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждения образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

по дисциплине «Основы информационной безопасности »

Выполнил студентка 2 курса 3 группы специальность ИСиТ Лыкова Ю.Д.

Преподаватель Ржеутская Н.В

**Практическое задание № 3**

**Тема «**Настройка Брандмауэра Windows**»**

Цель: Овладение навыками настройки и использования Брандмауэра Windows.

**Теоретическое введение**

Брандмауэр (Межсетевой экран) - это аппаратный или программный комплекс, позволяющий проверять (фильтровать) входные и выходные потоки данных, проходящие через интернет или сеть. В случаи нарушения политики безопасности компьютера, брандмауэр блокирует эти данные (рис. 1).

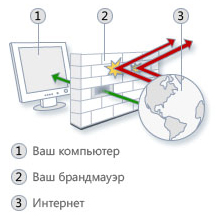


Рисунок 1 – Принцип действия Брандмауэра

Межсетевой экран является одним из основных компонентов защиты сетей. Наряду с Internet-протоколом межсетевого обмена (Internet Security Protocol — IPSec). Межсетевой экран является одним из важнейших средств защиты, осуществляя надежную аутентификацию пользователей и защиту от несанкционированного доступа. Отметим, что большая часть проблем с информационной безопасностью сетей связана с "прародительской" зависимостью коммуникационных решений от ОС UNIX – особенности открытой платформы и среды программирования UNIX сказались на реализации протоколов обмена данными и политики информационной безопасности. Вследствие этого ряд Internet-служб и совокупность сетевых протоколов (Transmission Control Protocol/Internet Protocol — TCP/IP) имеет "бреши" в защите.

К числу таких служб и протоколов относятся:

* служба сетевых имен (Domain Name Server — DNS);
* доступ к всемирной паутине WWW;
* программа электронной почты Send Mail;
* служба эмуляции удаленного терминала Telnet;
* простой протокол передачи электронной почты (Simple Mail Transfer Protocol — SMTP);
* протокол передачи файлов (File Transfer Protocol);
* графическая оконная система X Windows.

Настройки межсетевого экрана, т.е. решение пропускать или отсеивать пакеты информации, зависят от топологии распределенной сети и принятой политики информационной безопасности. В связи с этим политика реализации межсетевых экранов определяет правила доступа к ресурсам внутренней сети. Эти правила базируются на двух общих принципах – *запрещать всё, что не разрешено в явной форме, и разрешать всё, что не запрещено в явной форме*. Использование первого принципа дает меньше возможностей пользователям и охватывает жёстко очерченную область сетевого взаимодействия. Политика, основанная на втором принципе, является более мягкой, но во многих случаях она менее желательна, так как она предоставляет пользователям больше возможностей "обойти" межсетевой экран и использовать запрещенные сервисы через нестандартные порты (User Data Protocol – UDP), которые не запрещены политикой безопасности.

**Создание правила по ограничению доступа программ к сети**

Создание правила для входящих подключений:

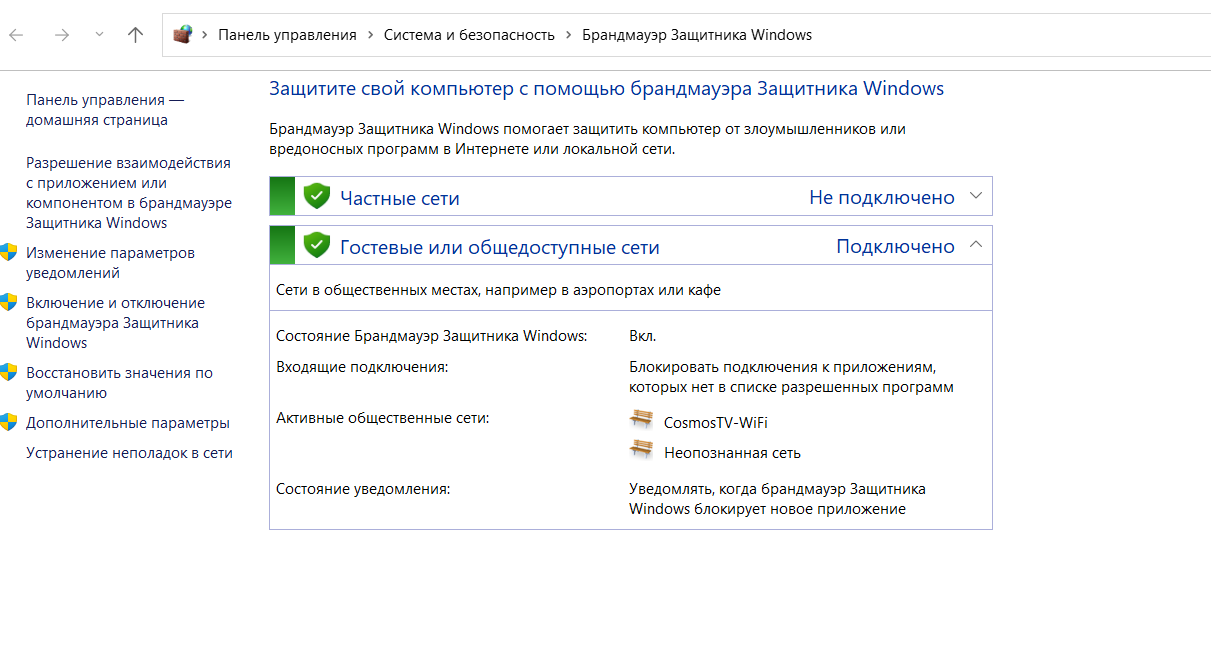


Рисунок 1­­- Брандмауэр



Рисунок2 - Включен ли Брэндмауэр

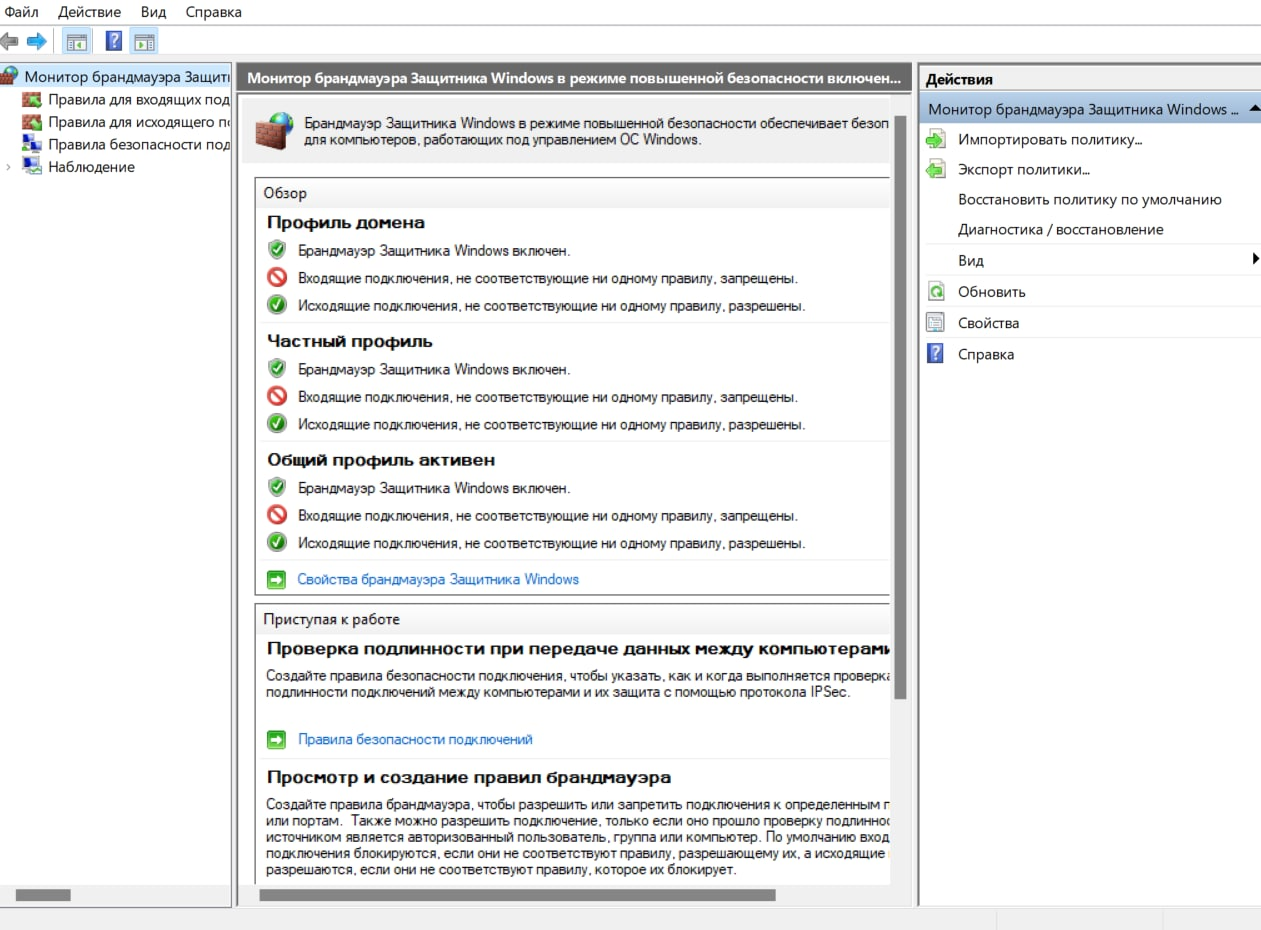


Рисунок 3-Монитор Брэндмауэра

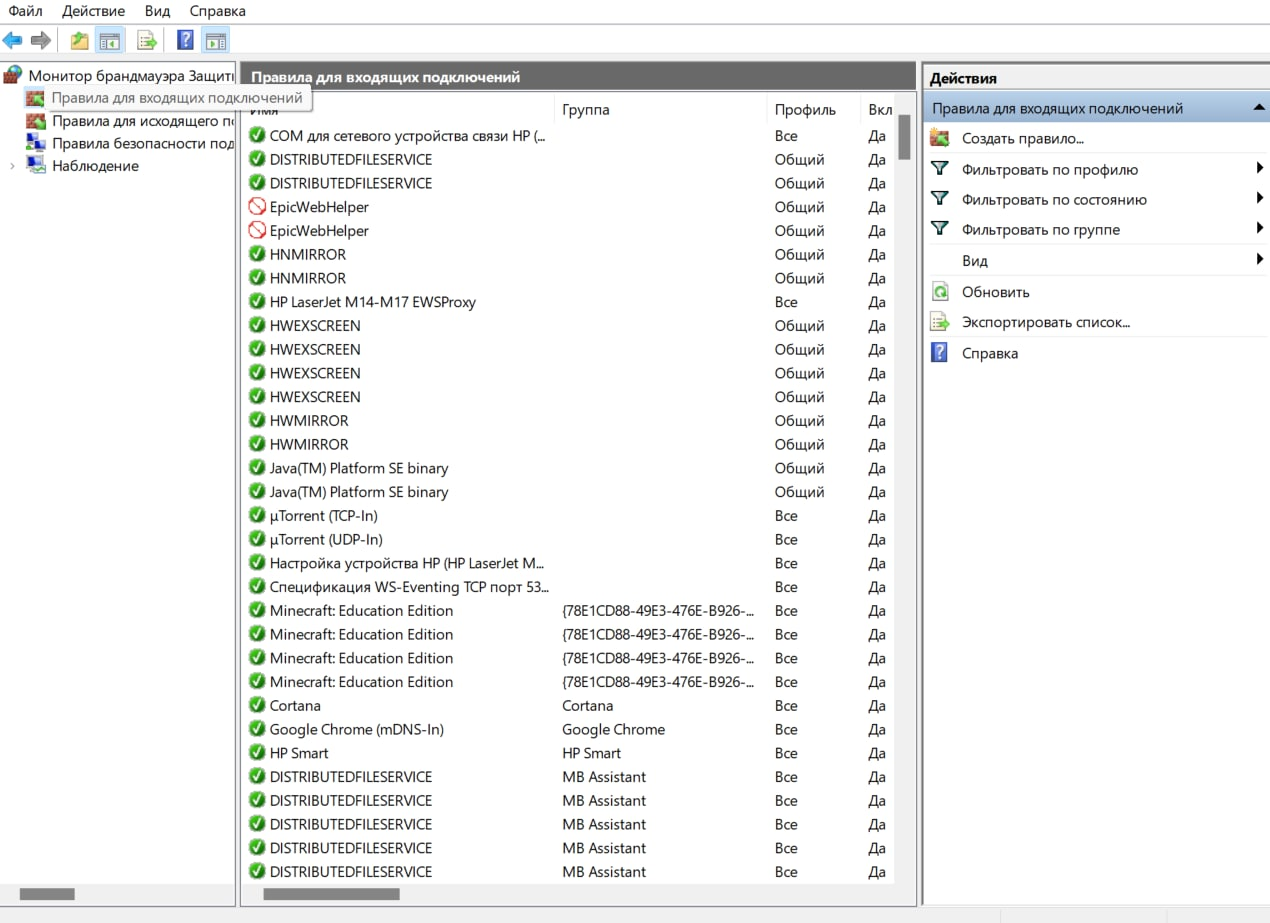


Рисунок 4-Панель правил для входящих подключений

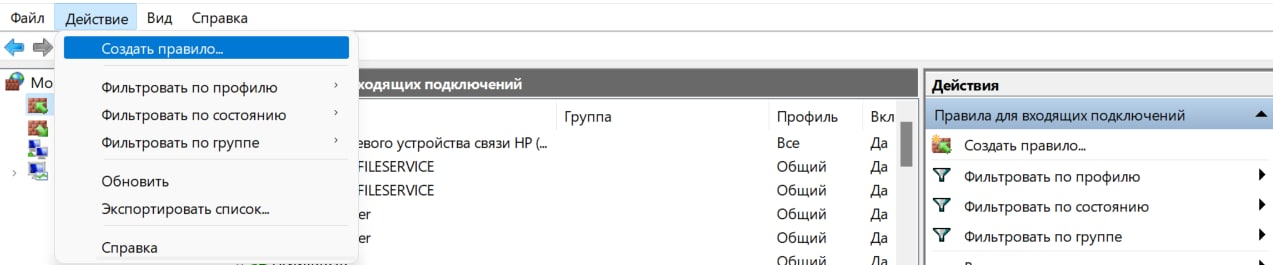


Рисунок 5-Создание правила для входящих подключений

Создать правило для входящих подключений можно либо справа ,либо слева Действие -> Создать правило

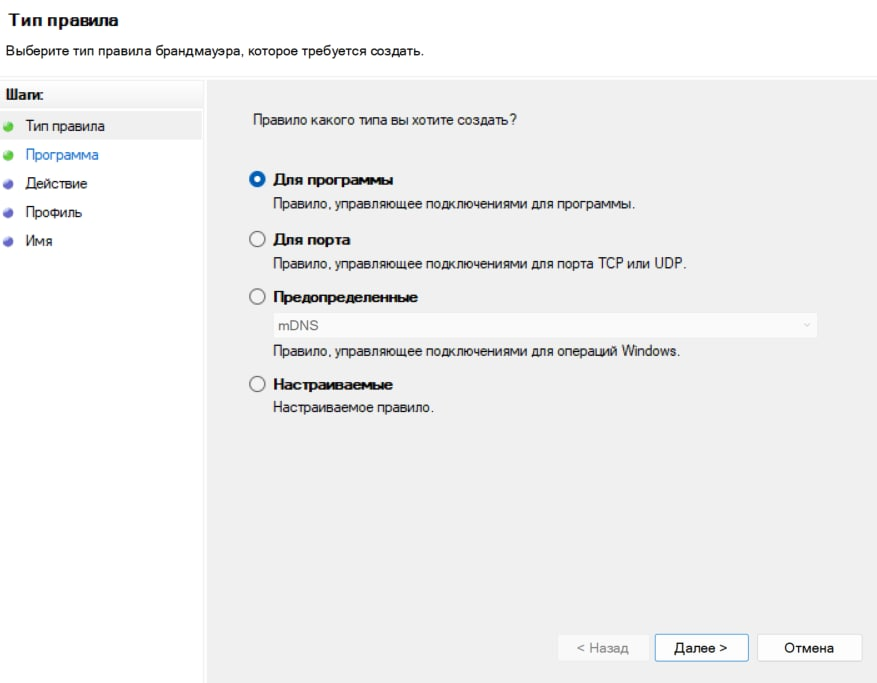


Рисунок 6- Выбор типа правила (в нашем случае для программ)

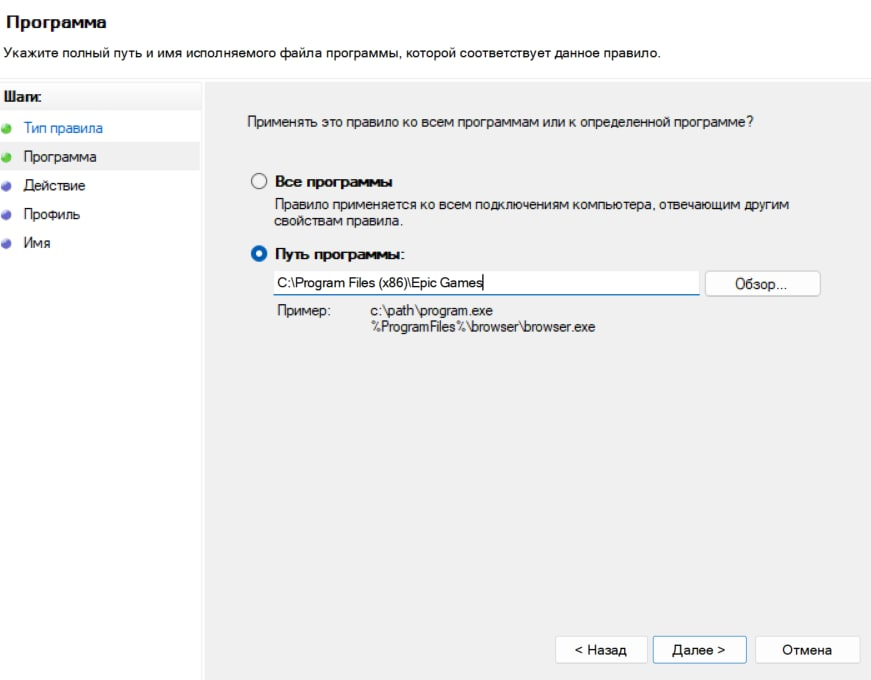


Рисунок 7- Прописываем путь программы (ни в коем случае не ярлык, а именно путь)

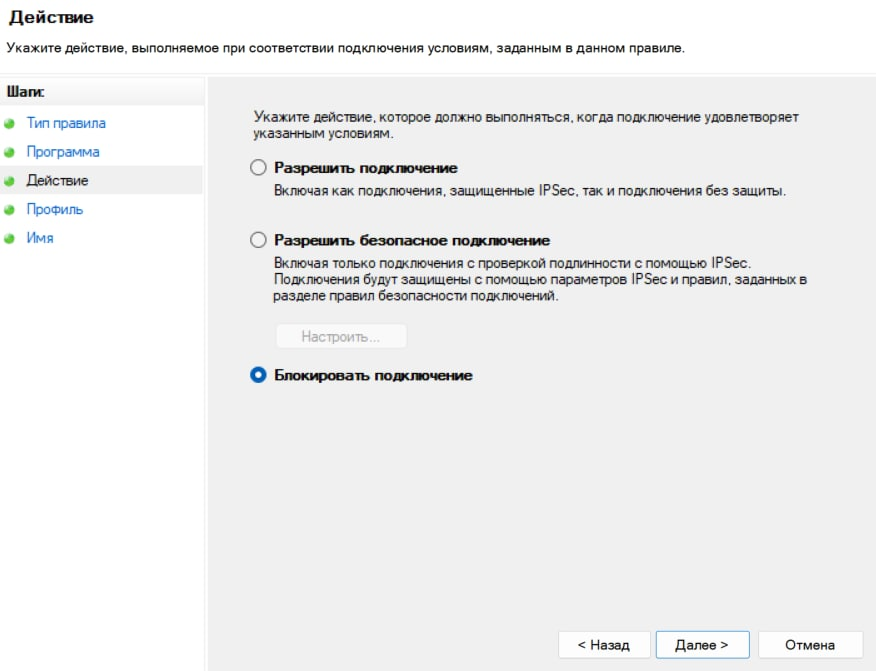


Рисунок 8- Выбор действия правила



Рисунок 9- Было создано правило Epic Games(с блокировкой подключения)

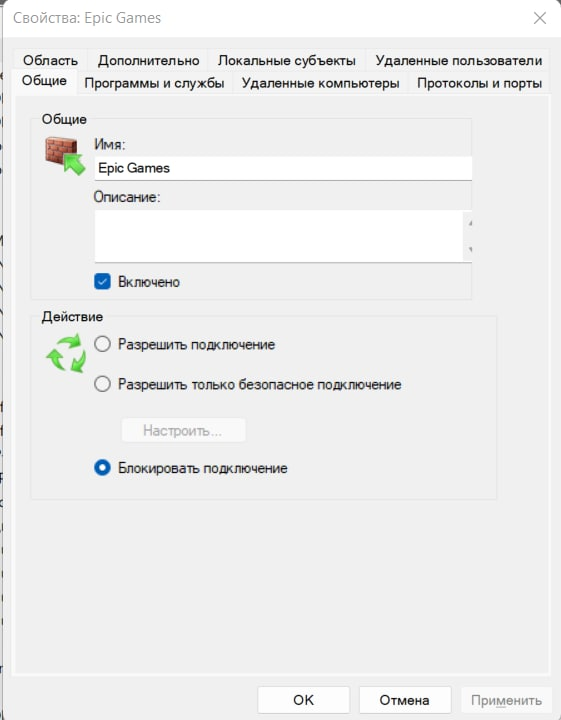


Рисунок 10-Свойства правила Epic Games

Свойство правила можно посмотреть при нажатии на него 2 раза.

Создание правила для исходящих подключений:

Правила для исходящих подключений создаются также как и для входящих.

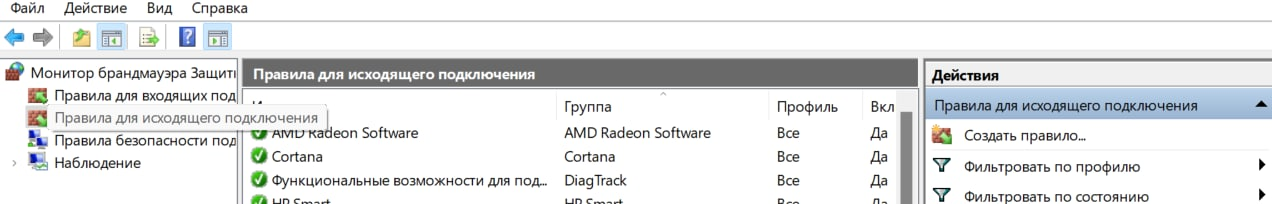


Рисунок 11-Выбор правила для исходящего подключения

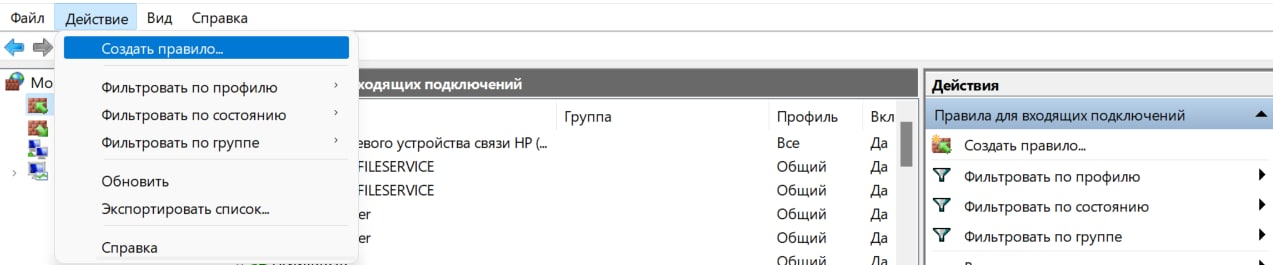


Рисунок 12-Создание правила для входящих подключений

Создать правило для входящих подключений можно либо справа ,либо слева Действие -> Создать правило

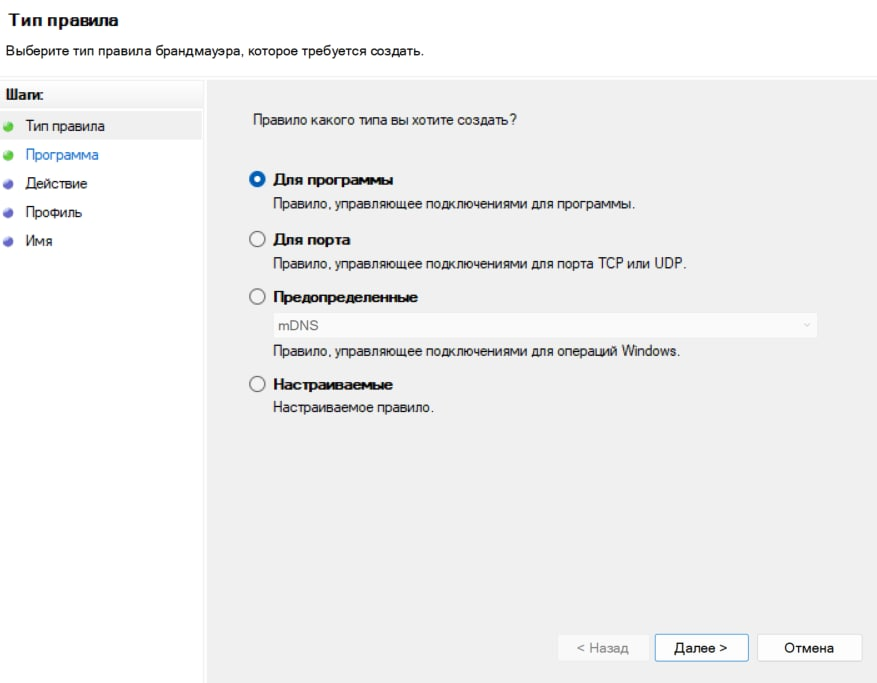


Рисунок 13- Выбор типа правила (в нашем случае для программ)

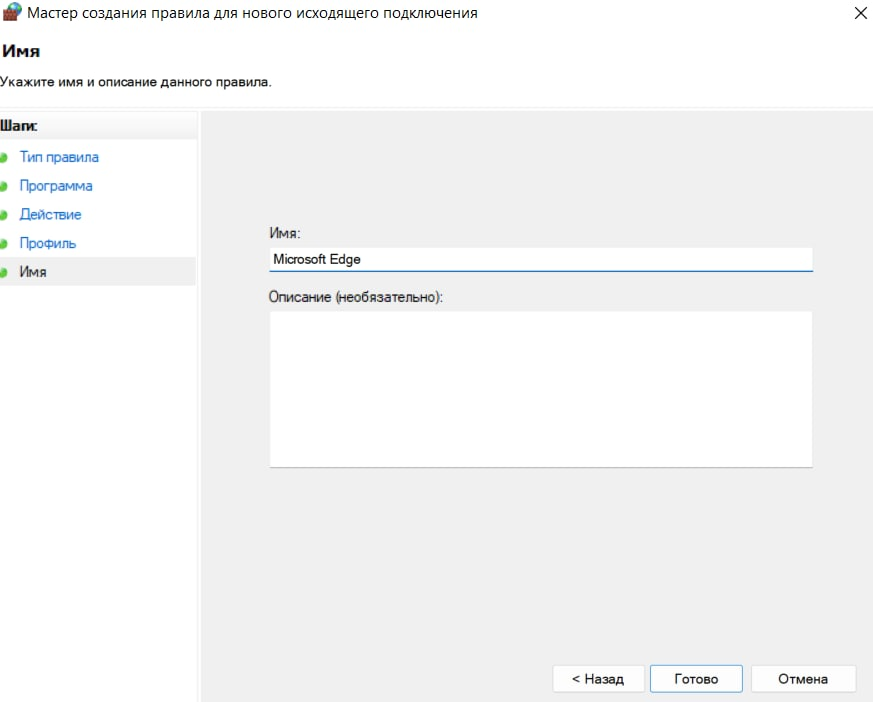


Рисунок 14-Название правила

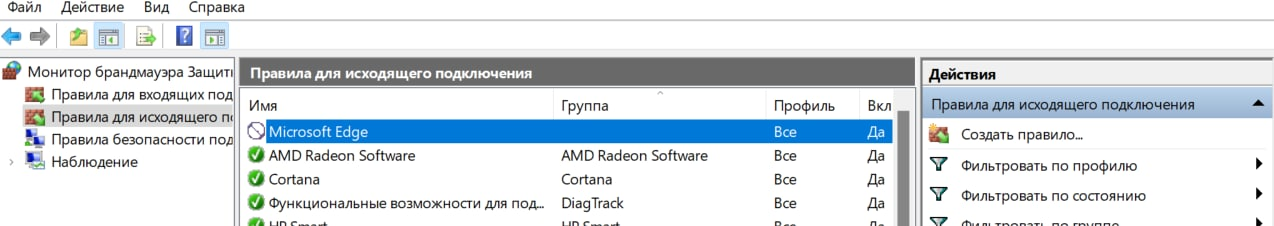


Рисунок 15-Правило Microsoft Edge было создано

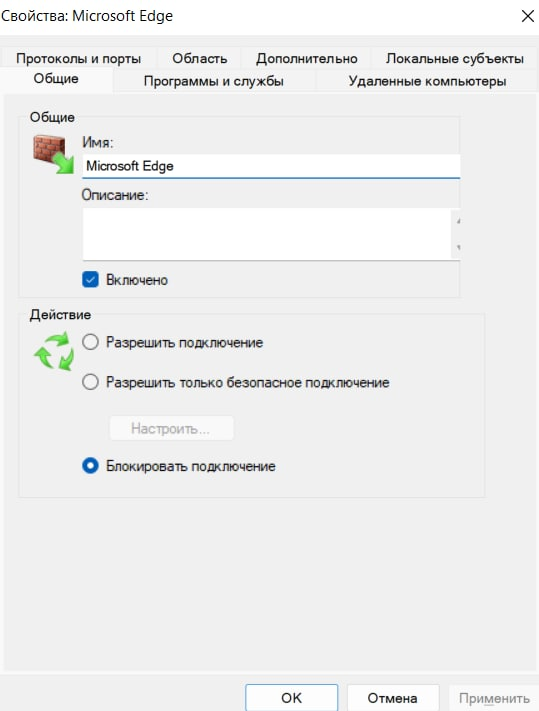


Рисунок 16-Свойства правила Epic Games

Свойство правила можно посмотреть при нажатии на него 2 раза.

Существуют большое количество команд для перехода в определенные настройки или ввод для подключения программ. Команды мы вводили в PowerShell (расширяемое средство автоматизации для выполнения обычных команд)

Были введены и показаны не все команды. Некоторые из них:

* Администрирование-control admintools;
* Диспетчер проверки драйверов-verifier;
* Диспетчер служебных программ-utilman.



Рисунок 1-Ввод команды control admintools

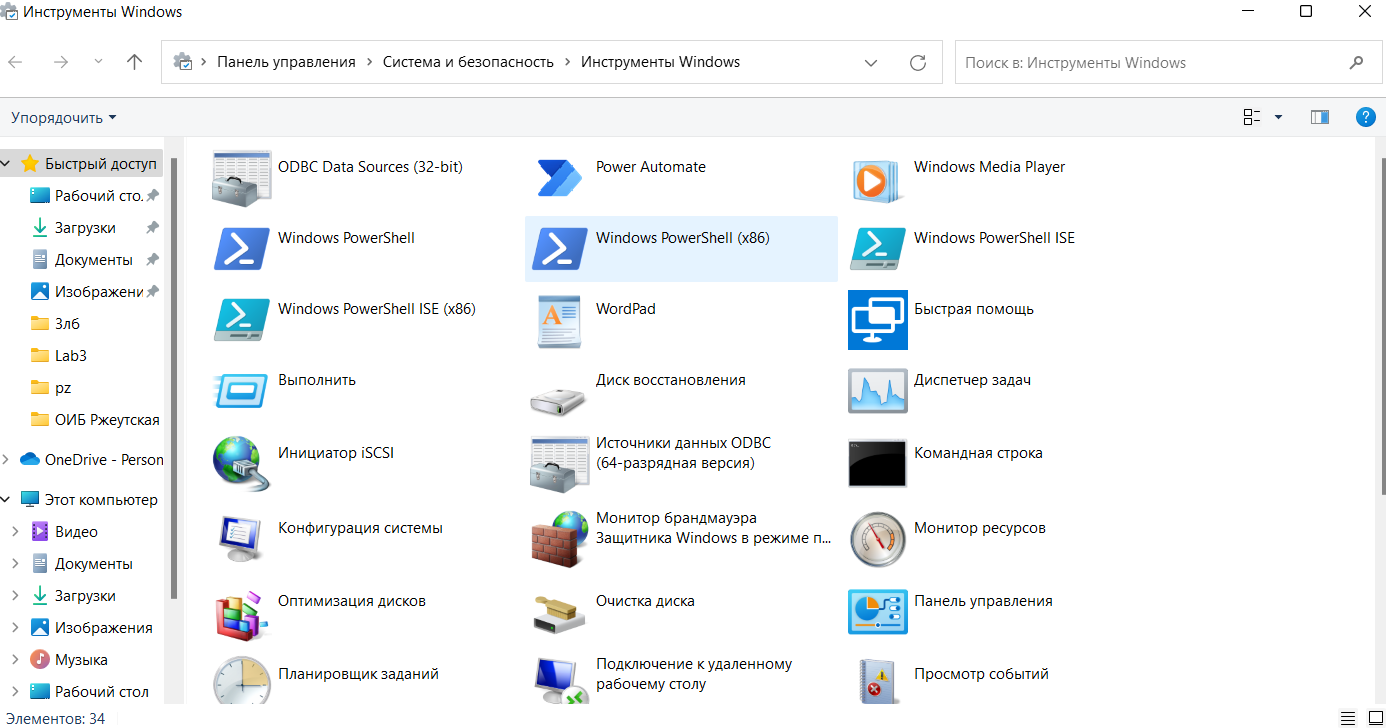


Рисунок 2-Переход на инструменты Windows

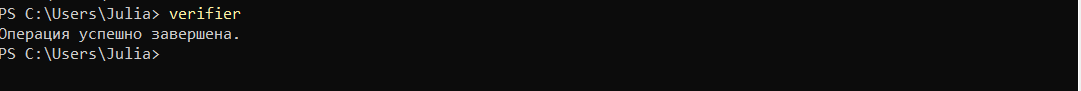


Рисунок 3-Ввод команды verifier

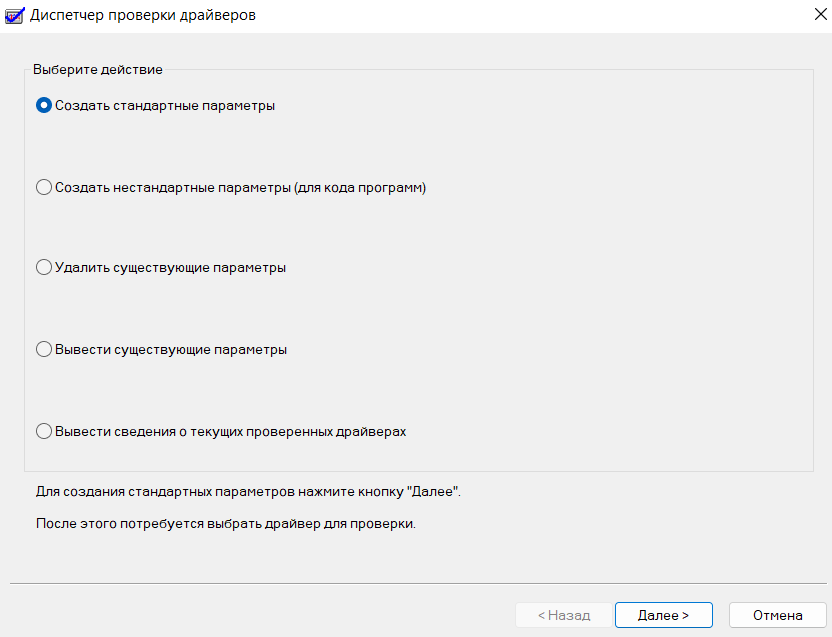


Рисунок 4- Диспетчер проверки драйверов



Рисунок 5 - Ввод команды utilman



Рисунок 6- Дисптчер служебных программ

**Вывод**

Брандмауэр - это программа, которая позволяет проверять (фильтровать) входные и выходные потоки данных, проходящие через интернет или сеть. В случаи нарушения политики безопасности компьютера, брандмауэр блокирует эти данные. Он используется для защиты компьютера от угроз и работает по 2 основным принципам – Разрешено всё, что не запрещено в явном виде и запрещено всё, что не разрешено в явном виде.

**Практическое задание №4.1**

**Тема «Криптографическая защита информации»**

Цель: Овладение основными криптографическими алгоритмами симметричного шифрования .

**Теоретическое введение**

Криптография - наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним) и аутентичности (целостности и подлинности авторства) информации.

Изначально криптография изучала методы шифрования информации – обратимого преобразования открытого (исходного) текста на основе секретного алгоритма и/или ключа в шифрованный текст (шифротекст). Традиционная криптография образует раздел симметричных криптосистем, в которых зашифрование и расшифрование проводится с использованием одного и того же секретного ключа.

Помимо этого современная криптография включает в себя асимметричные криптосистемы, системы электронной цифровой подписи, хеш-функции, управление ключами, получение скрытой информации, квантовую криптографию.

Шифрованием (encryption) называют процесс преобразования открытых данных (plaintext) в зашифрованные (шифртекст, ciphertext) или зашифрованных данных в открытые по определенным правилам с применением ключей.

В англоязычной литературе зашифрование / расшифрование – enciphering / deciphering.

Классификация алгоритмов шифрования

1. Симметричные (с секретным, единым ключом, одноключевые, single-key).

1.1. Потоковые:

· с одноразовым или бесконечным ключом (infinite-key cipher);

· с конечным ключом;

· на основе генератора псевдослучайных чисел.

1.2. Блочные:

1.2.1. Шифры перестановки (permutation, P-блоки);

1.2.2. Шифры замены (substitution, S-блоки):

· моноалфавитные;

· полиалфавитные;

2. Асимметричные (с открытым ключом, public-key):

· Диффи-Хеллман DH (Diffie, Hellman);

· Райвест-Шамир-Адлeман RSA (Rivest, Shamir, Adleman);

· Эль-Гамаль (ElGamal).

Симметричные алгоритмы шифрования (или криптография с секретными ключами) основаны на том, что отправитель и получатель информации используют один и тот же ключ. Этот ключ должен храниться в тайне и передаваться способом, исключающим его перехват.

Обмен информацией осуществляется в 3 этапа:

* отправитель передает получателю ключ (в случае сети с несколькими абонентами у каждой пары абонентов должен быть свой ключ, отличный от ключей других пар);
* отправитель, используя ключ, зашифровывает сообщение, которое пересылается получателю;
* получатель получает сообщение и расшифровывает его.

Если для каждого дня и для каждого сеанса связи будет использоваться уникальный ключ, это повысит защищенность системы.

При блочном шифровании информация разбивается на блоки фиксированной длины и шифруется поблочно. Блочные шифры бывают двух основных видов:

· шифры перестановки (transposition, permutation, P-блоки);

· шифры замены (подстановки, substitution, S-блоки).

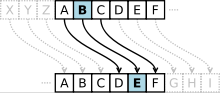
Шифры перестановок переставляют элементы открытых данных (биты, буквы, символы) в некотором новом порядке. Различают шифры горизонтальной, вертикальной, двойной перестановки, решетки, лабиринты, лозунговые и др.

Шифры замены заменяют элементы открытых данных на другие элементы по определенному правилу. Paзличают шифры простой, сложной, парной замены, буквенно-слоговое шифрование и шифры колонной замены. Шифры замены делятся на две группы:

· моноалфавитные (код Цезаря);

·полиалфавитные (шифр Видженера, цилиндр Джефферсона, диск Уэтстоуна, Enigma).

В моноалфавитных шифрах замены буква исходного текста заменяется на другую, заранее определенную букву. Например в коде Цезаря буква заменяется на букву, отстоящую от нее в латинском алфавите на некоторое число позиций.



Очевидно, что такой шифр взламывается совсем просто. Нужно подсчитать, как часто встречаются буквы в зашифрованном тексте, и сопоставить результат с известной для каждого языка частотой встречаемости букв.

В полиалфавитных подстановках для замены некоторого символа исходного сообщения в каждом случае его появления последовательно используются различные символы из некоторого набора. Понятно, что этот набор не бесконечен, через какое-то количество символов его нужно использовать снова. В этом слабость чисто полиалфавитных шифров.

В современных криптографических системах, как правило, используют оба способа шифрования (замены и перестановки). Такой шифратор называют составным (product cipher). Oн более стойкий, чем шифратор, использующий только замены или перестановки.

В асимметричных алгоритмах шифрования (или криптографии с открытым ключом) для зашифровывания информации используют один ключ (открытый), а для расшифровывания - другой (секретный). Эти ключи различны и не могут быть получены один из другого.

Схема обмена информацией такова:

· получатель вычисляет открытый и секретный ключи, секретный ключ хранит в тайне, открытый же делает доступным (сообщает отправителю, группе пользователей сети, публикует);

· отправитель, используя открытый ключ получателя, зашифровывает сообщение, которое пересылается получателю;

· получатель получает сообщение и расшифровывает его, используя свой секретный ключ.

**Задание 2**

**сообщение Лыкова Юлия**

***Шифр Цезаря***

Шифр Цезаря — один из древнейших шифров. При шифровании каждый символ заменяется другим, отстоящим от него в алфавите на фиксированное число позиций. Шифр Цезаря можно классифицировать как шифр подстановки, при более узкой классификации — шифр простой замены.

Шифр назван в честь римского императора Гая Юлия Цезаря, использовавшего его для секретной переписки.

Математическая модель

Если сопоставить каждому символу алфавита его порядковый номер (нумеруя с 0), то шифрование и дешифрование можно выразить формулами:

http://kriptografea.narod.ru/13.png

http://kriptografea.narod.ru/14.png

где x — символ открытого текста

y — символ шифрованного текста

n — мощность алфавита (кол-во символов)

k — ключ.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Сообщение | Л | ы | к | о | в | а | Ю | л | и | я |
| 1(номер) | 13 | 29 | 12 | 16 | 3 | 1 | 32 | 13 | 10 | 33 |
| 1+12 | 25 | 8 | 24 | 28 | 15 | 13 | 11 | 25 | 21 | 12 |
| шифр | Ч | ж | ц | ь | н | л | Й | ч | ф | к |

***Лыкова Юлия****-(сдвиг12)****Чжцънл\_Йчфк***

***Шифрование с использованием системы Трисемуса:***

В 1508 г. аббат из Германии Иоганн Трисемус написал печатную работу по криптологии под названием «Полиграфия». В этой книге он впервые систематически описал применение шифрующих таблиц, заполненных алфавитом в случайном порядке. Для получения такого шифра замены обычно использовались таблица для записи букв алфавита и ключевое слово (или фраза). В таблицу сначала вписывалось по строкам ключевое слово, причем повторяющиеся буквы отбрасывались. Затем эта таблица дополнялась не вошедшими в нее буквами алфавита по порядку. На рис.5.2 изображена таблица с ключевым словом «ЗАЩИТА».



Каждая буква открытого сообщения заменяется буквой, расположенной под ней в том же столбце. Если буква находится в последней строке таблицы, то для ее шифрования берут самую верхнюю букву столбца.

Исходное сообщение Лыкова Юлия Дмитриевна, зашифрованное – ***Пяоужб\_Щпги\_Кргдхглжсб***.

***Шифрование с использованием системы Виженера:***

В [шифре Цезаря](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%A6%D0%B5%D0%B7%D0%B0%D1%80%D1%8F) каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 32 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 32 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова. Например, предположим, что исходный текст имеет вид:

Лыкова Юлия

Человек, посылающий сообщение, записывает ключевое слово («ЗАЩИТА») циклически до тех пор, пока его длина не будет соответствовать длине исходного текста:

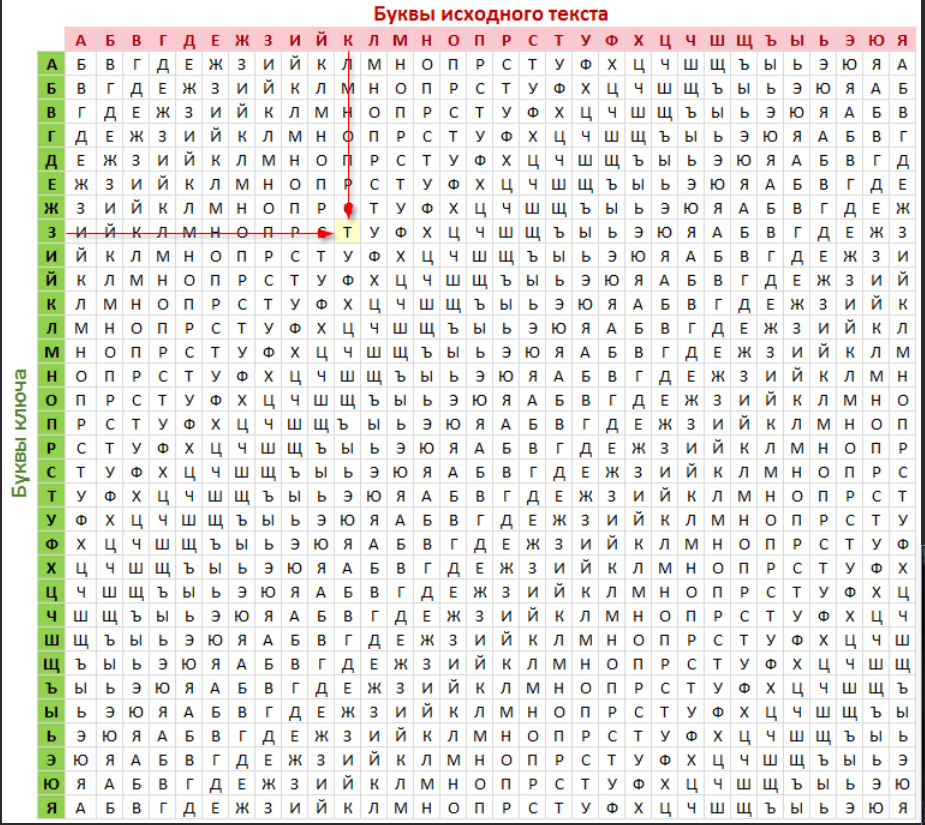


Таблица Виженера для русского алфавита

*Шифрование с использованием системы Виженера:*

Исходный текст: ЛыковаЮлия

Ключ: Защитазащи

Зашифрованный текст: ***уыдчфаёлвз***

**Шифр Плейфера** использует матрицу 5х5 (для латинского алфавита, для кириллического алфавита необходимо увеличить размер матрицы до 4х8), содержащую ключевое слово или фразу. Для создания матрицы и использования шифра достаточно запомнить ключевое слово и четыре простых правила. Чтобы составить ключевую матрицу, в первую очередь нужно заполнить пустые ячейки матрицы буквами ключевого слова (не записывая повторяющиеся символы), потом заполнить оставшиеся ячейки матрицы символами алфавита, не встречающимися в ключевом слове, по порядку (в русских текстах обычно «Е» и «Ё» объединяются в одну ячейку). Ключевое слово может быть записано в верхней строке матрицы слева направо, либо по спирали из левого верхнего угла к центру. Ключевое слово, дополненное алфавитом, составляет матрицу 4х8 и является ключом шифра.

Для того чтобы зашифровать сообщение, необходимо разбить его на биграммы (группы из двух символов), например «Лыкова\_Юлия» становится «Лы ко ва \_Юл ия », и отыскать эти биграммы в таблице. Два символа биграммы соответствуют углам прямоугольника в ключевой матрице. Определяем положения углов этого прямоугольника относительно друг друга. Затем, руководствуясь следующими 4 правилами, зашифровываем пары символов исходного текста:

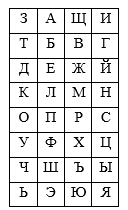
1. Если два символа биграммы совпадают (или если остался один символ), добавляем после первого символа «Х», зашифровываем новую пару символов и продолжаем. В некоторых вариантах шифра Плейфера вместо «Х» используется «Q».

2. Если символы биграммы исходного текста встречаются в одной строке, то эти символы замещаются на символы, расположенные в ближайших столбцах справа от соответствующих символов. Если символ является последним в строке, то он заменяется на первый символ этой же строки.

3. Если символы биграммы исходного текста встречаются в одном столбце, то они преобразуются в символы того же столбца, находящиеся непосредственно под ними. Если символ является нижним в столбце, то он заменяется на первый символ этого же столбца.

4. Если символы биграммы исходного текста находятся в разных столбцах и разных строках, то они заменяются на символы, находящиеся в тех же строках, но соответствующие другим углам прямоугольника.

Для расшифровки необходимо использовать инверсию этих четырёх правил, откидывая символы «Х» (или «Q»), если они не несут смысла в исходном сообщении.



Лы ­-Мъ(биграмма форм.прямоугольник)

ко-уч(бигрмма расп. в одном столбце)

ва-бщ(биграмма форм.прямоугольник)

Юл-Эм(биграмма форм.прямоугольник)

ия-ги(биграмма расп. в одном столбце одна из букв в конце столбца)

***Мъучбщ\_Эмги***

|  |  |  |
| --- | --- | --- |
|  | **Ие михежцчжшйч сшихуцчб** | Расшифровать с помощью шифра Цезаря. Ключ 5 |

Расшифровка: да здравствует мудрость

|  |  |  |
| --- | --- | --- |
| 2. | **съчпщг окхчхиге ичлкпщг** | Расшифровать с помощью шифра Цезаря. Ключ 7 |

Расшифровка: курить здоровью вредить

|  |  |  |
| --- | --- | --- |
| 3. | **ьоъщочто т лечшыхтлшыьё** | Расшифровать с помощью шифра Цезаря. Ключ 10 |

Расшифровка: терпение и выносливость

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 4. | **Лжбзеииеизиачзнр** | Расшифровать с помощью **Магического квадрата**   |  |  |  |  | | --- | --- | --- | --- | | 7 | 12 | 1 | 14 | | 2 | 13 | 8 | 11 | | 16 | 3 | 10 | 5 | | 9 | 6 | 15 | 4 | |

Расшифровка:

|  |  |  |  |
| --- | --- | --- | --- |
| л | ж | б | З |
| е | и | и | Е |
| и | з | и | А |
| ч | з | н | р |

без различие жизни

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5. | **тяеонаыбиьсрль\_т** | Расшифровать с помощью **Магического квадрата**   |  |  |  |  | | --- | --- | --- | --- | | 9 | 16 | 2 | 7 | | 6 | 3 | 13 | 12 | | 15 | 10 | 8 | 1 | | 4 | 5 | 11 | 14 | |

Расшифровка:

|  |  |  |  |
| --- | --- | --- | --- |
| т | я | е | О |
| н | а | ы | б |
| и | ь | с | р |
| л | ь | \_ | т |

реальность \_бытия

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 6. | **иоарткдпвл\_натоа** | Расшифровать с помощью **Магического квадрата**   |  |  |  |  | | --- | --- | --- | --- | | 7 | 12 | 1 | 14 | | 2 | 13 | 8 | 11 | | 16 | 3 | 10 | 5 | | 9 | 6 | 15 | 4 | |

Расшифровка:

|  |  |  |  |
| --- | --- | --- | --- |
| и | о | а | р |
| т | к | д | п |
| в | л | \_ | н |
| а | т | о | а |

атлантида\_покров

|  |  |  |
| --- | --- | --- |
| 7. | **гэ ишн зижшпэг оюжи** | Расшифровать с помощью шифра Цезаря.  Ключ 8.  Ключевое слово ВЕСНА |

Расшифровка: не так страшен чёрт

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Ш | Щ | Ъ | Ы | Ь | Э | Ю | Я | **В** | **Е** | **С** | **Н** | **А** | Б | Г | Д | Ё | Ж | З | И | Й | К | Л | М | О | П | Р | Т | У | Ф | Х | Ц | У |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **8** | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |

|  |  |  |
| --- | --- | --- |
| 8. | **дороюё дъ нозбъвё жодщйц** | Расшифровать с помощью шифра Цезаря.  Ключ 5.  Ключевое слово ОСЕНЬ |

Расшифровка: нечего на зеркало пенять

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Ъ | Ы | Э | Ю | Я | **О** | **С** | **Е** | **Н** | **Ь** | А | Б | В | Г | Д | Ё | Ж | З | И | Й | К | Л | М | П | Р | Т | У | Ф | Х | Ц | Ч | Ш | Щ |
| 0 | 1 | 2 | 3 | 4 | **5** | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |
| Ъ | Ы | Ь | Э | Ю | Я | **З** | **И** | **М** | **А** | Б | В | Г | Д | Е | Ё | Ж | Й | К | Л | Н | О | П | Р | С | Т | У | Ф | Х | Ц | Ч | Ш | Щ |
| 0 | 1 | 2 | 3 | 4 | 5 | **6** | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |

|  |  |  |
| --- | --- | --- |
| 9. | **жъйтаьъщ ёьръ клъюё жёйлал** | Расшифровать с помощью шифра Цезаря.  Ключ 6.  Ключевое слово ЗИМА |

Расшифровка: паршивая овца стадо портит

|  |  |  |
| --- | --- | --- |
| 10. | **Боитдиултоьтдгь-псеоснояшмяил\_ьу-бу\_дччуч\_** | Расшифровать с помощью **Метода простой перестановки Таблица6х7** |

Расшифровка: будешь долго мучаться \_что-нибудь получится \_ \_ \_

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| б | у | д | е | ш | ь | д |
| о | л | г | о | м | у | ч |
| и | т | ь | с | я | \_ | ч |
| т | о | - | н | и | б | у |
| д | ь | п | о | л | у | ч |
| и | т | с | я | \_ | \_ | \_ |

|  |  |  |
| --- | --- | --- |
| 11. | **гг\_ \_ ип-оомитрдр\_ос,яубогп\_**  **мбадираоитнла\_гноаавуоа** | Расшифровать с помощью **Метода простой перестановки Таблица7х7** |

Расшифровка: горбатого \_одна \_ могила \_ исправит, \_ а \_ у прямого-дубина

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| г | о | р | б | а | т | о |
| г | о | \_ | о | д | н | а |
| \_ | м | о | г | и | л | а |
| \_ | и | с | п | р | а | в |
| и | т | , | \_ | а | \_ | у |
| п | р | я | м | о | г | о |
| - | д | у | б | и | н | а |

|  |  |  |
| --- | --- | --- |
| 12. | **нотеч\_е\_ \_кем\_кчалемре,осеал\_втссоаеотив\_к\_** | Расшифровать с помощью **Метода простой перестановки Таблица6х7** |

***Расшифровка: не \_ место\_красит \_человека , \_ а \_ человек \_ место\_***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| н | е | \_ | м | е | с | т |
| о | \_ | к | р | а | с | и |
| т | \_ | ч | е | л | о | в |
| е | к | а | , | \_ | а | \_ |
| ч | е | л | о | в | е | к |
| \_ | м | е | с | т | о | \_ |

|  |  |  |
| --- | --- | --- |
| 13. | **\_яетож\_нксучнтуотдеыужбьатйдны\_,**  **с\_ао\_о\_яс,\_ндк,е\_иаиу\_ккддндаруто\_**  **а\_еди.в\_нта\_ \_ахе:\_** | Расшифровать с помощью **Одиночной перестановки по ключу.**  **Ключ – СЧАСТЬЕ Таблица 12х7** |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| ­\_ | н | а | \_ | \_ | р | \_ |
| я | т | т | о | и | у | н |
| е | у | й | \_ | а | т | т |
| т | о | д | я | и | о | а |
| о | т | н | с | у | \_ | \_ |
| ж | д | ы | , | \_ | а | \_ |
| \_ | е | \_ | \_ | к | \_ | а |
| н | ы | , | н | к | е | х |
| к | у | с | д | д | д | е |
| с | ж | \_ | к | д | и | ; |
| у | б | а | , | н | . | \_ |
| ч | ь | о | е | а | в |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| с | ч | а | с | т | ь | е |
| 19 | 25 | 1 | 19 | 20 | 30 | 6 |

**Вывод**: Овладение основными криптографическими алгоритмами симметричного шифрования, применение на практике

**Практическое задание №4.2**

**Тема «Криптографическая защита информации»**

Цель: Овладение основными криптографическими алгоритмами асимметричного шифрования .

**Теоретические сведения**

· Диффи-Хеллман DH (Diffie, Hellman);

· Райвест-Шамир-Адлeман RSA (Rivest, Shamir, Adleman);

· Эль-Гамаль (ElGamal).

Реализация элементов криптосистемы RSA

RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел.

Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи. Алгоритм используется в большом числе криптографических приложений, включая PGP, S/MIME, TLS/SSL, IPSEC/IKE и других.

Весь алгоритм расписан в таблице:

|  |  |  |
| --- | --- | --- |
| Этап | Описание операции | Результат операции |
| Генерация ключей | Выбрать два простых различных числа | p=3557,  q=2579 |
| Вычислить модуль (произведение) | n = p \cdot q = 3557 \cdot 2579 = 9173503 |
| Вычислить функцию Эйлера | \varphi(n) = (p-1) (q-1) = 9167368 |
| Выбрать открытую экспоненту | e = 3 |
| Вычислить секретную экспоненту | d = e^{-1} \mod \varphi(n)  d = 6111579 |
| Опубликовать открытый ключ | \{e, n\} = \{3,9173503 \} |
| Сохранить закрытый ключ | \{d, n\} = \{6111579, 9173503 \} |
| Шифрование | Выбрать текст для зашифровки | m = 111111 |
| Вычислить шифротекст | \begin{align} c &= E(m) \\  &= m^e \mod n \\  &= 111111^3   \mod 9173503 \\  &= 4051753 \end{align} |
| Расшифрование | Вычислить исходное сообщение | \begin{align} m &= D(c) = \\   &= c^d \mod n \\   &= 4051753^{6111579} \mod 9173503 \\   &= 111111 \end{align} |

# Реализация элементов схемы шифрования Эль-Гамаля

## **Генерация ключей**

1. Генерируется случайное простое число ~p длины ~n [битов](https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D1%82).
2. Выбирается случайный примитивный элемент ~g.
3. Выбирается случайное целое число ~x такое, что ~1 < x < p-1.
4. Вычисляется ~y = g^x\,\bmod\,p.
5. Открытым ключом является тройка \left( p,g,y \right), закрытым ключом — число ~x.

## **Шифрование**

Сообщение ~M шифруется следующим образом:

1. Выбирается сессионный [ключ](https://ru.wikipedia.org/wiki/%D0%9A%D0%BB%D1%8E%D1%87_(%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%BE%D0%B3%D1%80%D0%B0%D1%84%D0%B8%D1%8F)) — случайное целое число ~k такое, что ~1 < k < p - 1
2. Вычисляются числа a = g^k\,\bmod\,p и b = y^k M\,\bmod\,p.
3. Пара чисел \left( a, b \right) является [шифротекстом](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80%D0%BE%D1%82%D0%B5%D0%BA%D1%81%D1%82).

Нетрудно видеть, что длина шифротекста в схеме Эль-Гамаля длиннее исходного сообщения M вдвое.

## **Расшифрование**

Зная закрытый ключ ~x, исходное сообщение можно вычислить из шифротекста \left( a, b \right) по формуле:

M = b(a^x)^{-1}\,\bmod\,p.

При этом нетрудно проверить, что

~(a^x)^{-1}\equiv g^{-kx}\pmod{p}

и поэтому

~b(a^x)^{-1}\equiv (y^kM)g^{-xk}\equiv (g^{xk}M) g^{-xk}\equiv M \pmod{p}.

Для практических вычислений больше подходит следующая формула:

M = b(a^x)^{-1}\,\bmod\,p = b \cdot a^{(p-1-x)}\,\bmod\,p 

## **Пример**

**Шифрование**

Допустим, что нужно зашифровать сообщение ~M=5.

Произведем генерацию ключей :

пусть ~p=11, g=2. Выберем ~x=8 - случайное целое число ~x такое, что ~1 < x < p.

Вычислим ~y= g^x\bmod{p}=2^8\bmod{11}=3.

Итак, открытым является тройка ~(p,g,y)=(11,2,3), а закрытым ключом является число ~x=8.

Выбираем случайное целое число ~k такое, что 1 < k < (p − 1). Пусть ~k=9.

Вычисляем число ~a=g^k\bmod{p}=2^9 \bmod{11}=512 \bmod{11}=6.

Вычисляем число ~b=y^k M\bmod{p}=3^9 5 \bmod{11}=19683 \cdot 5 \bmod{11}=9.

Полученная пара ~(a,b)=(6,9) является шифротекстом.

**Расшифрование**

Необходимо получить сообщение ~M=5 по известному шифротексту ~(a,b)=(6,9) и закрытому ключу ~x=8.

Вычисляем M по формуле : ~M=b(a^x)^{-1}\bmod{p}=9(6^8)^{-1}\mod{11}=5

Получили исходное сообщение ~M=5.

# Реализация элементов схемы шифрования Дифи-Хеллмана

## **Генерация ключей**

В 1976 году после публичной критики алгоритма DES и указания на сложность обработки секретных ключей Уитфилд Диффи (Whitfield Diffie) и Мартин Хеллман (Martin Hellman) опубликовали свой алгоритм обмена ключами. Это была первая публикация на тему криптографии с открытым ключом и, возможно, самый большой шаг вперед в области криптографии, сделанный когда‑либо.

Из‑за невысокого быстродействия, свойственного асимметричным алгоритмам, алгоритм Диффи‑Хеллмана не предназначен для шифрования данных. Он был ориентирован на передачу секретных ключей DES, ARS или других подобных алгоритмов через небезопасную среду. В большинстве случаев алгоритм Диффи‑Хеллмана не используется для шифрования сообщений, потому что он, в зависимости от реализации, от 10 до 1000 раз медленнее алгоритма DES.

До алгоритма Диффи‑Хеллмана было сложно совместно использовать зашифрованные данные из‑за проблем хранения ключей и передачи информации. В большинстве случаев передача информации по каналам связи небезопасна, потому что сообщение может пройти десятки систем, прежде чем оно достигнет потенциального адресата, и нет никаких гарантий, что по пути никто не сможет взломать секретный ключ. Уитфилд Диффи и Мартин Хеллман предложили зашифровывать секретный ключ DES по алгоритму Диффи‑Хеллмана на передающей стороне и пересылать его вместе с сообщением, зашифрованным с использованием DES. Тогда на другом конце его сможет расшифровать только получатель сообщения.

На практике **обмен ключами** по алгоритму Диффи‑Хеллмана происходит по следующей схеме.

1. Два участника обмена договариваются о двух числах. Один выбирает большое простое число, а другой – целое число, меньшее числа первого участника. Переговоры они могут вести открыто, и это никак не отразится на безопасности.
2. Каждый из двух участников, независимо друг от друга, генерирует другое число, которое они будут хранить в тайне. Эти числа выполняют роль секретного ключа. Далее в вычислениях используются секретный ключ и два предыдущих целых числа. Результат вычислений посылается участнику обмена, и он играет роль открытого ключа.
3. Участники обмена обмениваются открытыми ключами. Далее они, используя собственный секретный ключ и открытый ключ партнера, конфиденциально вычисляют ключ сессии. Каждый партер вычисляет один и тот же ключ сессии.
4. Ключ сессии может использоваться как секретный ключ для другого алгоритма шифрования, например DES. Никакое третье лицо, контролирующее обмен, не сможет вычислить ключ сессии, не зная один из секретных ключей.

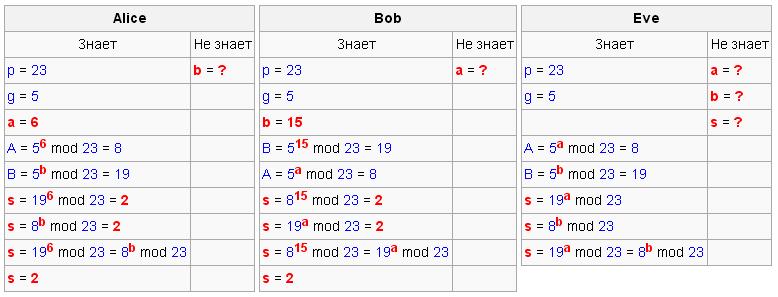
**Самое сложное в алгоритме** Диффи‑Хеллмана обмена ключами – это понять, что в нем фактически два различных независимых цикла шифрования. Алгоритм Диффи‑Хеллмана применяется для обработки небольших сообщений от отправителя получателю. Но в этом маленьком сообщении передается секретный ключ для расшифровки большого сообщения.

**Сильная сторона алгоритма** - никто не сможет скомпрометировать секретное сообщение, зная один или даже два открытых ключа получателя и отправителя. В качестве секретных и открытых ключей используются очень большие целые числа. Алгоритм Диффи‑Хеллмана основан на полезных для криптографии свойствах дискретных логарифмов.

**Пример**

Ева — криптоаналитик. Она читает пересылку Боба и Алисы, но не изменяет содержимого их сообщений.

* s = секретный ключ. s = 2
* g = простое число меньшее p. g = 5
* p = открытое простое число. p = 23
* a = секретный ключ Алисы. a = 6
* A = открытый ключ Алисы. A = ga mod p = 8
* b = секретный ключ Боба. b = 15
* B = открытый ключ Боба. B = gb mod p = 19



**Задание №1**

*Рассказать процесс работы алгоритма RSA.*

RSA (аббревиатура от фамилий Rivest, Shamir и Adleman) — криптографический алгоритм с открытым ключом, основывающийся на вычислительной сложности задачи факторизации больших целых чисел. Криптосистема RSA стала первой системой, пригодной и для шифрования, и для цифровой подписи.

Для начала нужно сгенерировать публичный и приватный ключ.

* Выбираю два простых числа. Пусть это будет p=3 и q=7.
* Вычисляю модуль — произведение p и q: n=p\*q=3\*7=21.
* Вычисляю функциюЭйлера: φ=(p-1)\*(q-1)=2\*6=12.
* Выбираю число e, отвечающее следующим критериям:

1. Оно должно быть простое
2. Оно должно быть взаимно простое с φ; остаются варианты 5, 7, 11.
3. Оно должно быть взаимно простое с φ; остаются варианты 5, 7, 11. Пусть e=5. Это открытая экспонента.

Теперь пара чисел {e, n} — это мой открытый ключ. Я отправляю его кому-то, чтобы этот кто-то зашифровал своё сообщение. Но мне еще нужно получить закрытый ключ.

Мне нужно вычислить число d, обратное е по модулю φ. То есть остаток от деления по модулю φ произведения d\*e должен быть равен 1. d может мыть равен 17. Пара {d, n} — это секретный ключ, его я оставляю у себя. Его нельзя сообщать никому. Только обладатель секретного ключа может расшифровать то, что было зашифровано открытым ключом.

Теперь зашифрую какое-нибудь сообщение(число).

Пусть это будет число 19. Обозначу его P=19. Также имеется открытый ключ: {e, n} = {5, 21}. Шифрование выполняется по следующему алгоритму:

* Выбранное число возводится в степень e по модулю n. То есть, вычисляется 19 в степени 5 (2476099) и берётся остаток от деления на 21. Получается 10 — это закодированные данные.

Полученные данные E=10 отправляются назад отправителю.

Однако сообщение P=19 не должно быть больше n=21, иначе ничего не получится.

Теперь расшифрую то, что было зашифровано.

Я получил данные (E=10), и у меня имеется закрытый ключ  
{d, n} = {17, 21}.

Суть в том, что открытый ключ не может расшифровать сообщение, а закрытый ключ я никому не сообщал.

*Начинаю расшифровывать:*

Я делаю операцию, похожую на шифровку, но вместо e использую d. Возвожу E в степень d: получаю 10 в степени 17. Вычисляю остаток от деления на 21 и получаю 19 — исходное сообщение.

Никто, кроме обладателя закрытого ключа не сможет расшифровать сообщение (ну почти никто).

# **Преимущество** — при помощи открытого ключа алгоритма шифрования невозможно прочитать видоизмененное сообщение. Для этого требуется закрытый ключ, который есть только у адресата.

**Недостаток** — медленное шифрование из-за громоздкости вычислительных операций. Невозможность получить доступ ко всем сообщениям, которые шифруются одним ключом.

**Задание №2**

*Рассказать процесс работы алгоритма Диффи-Хеллмана.*

Пусть есть 2 числа – g и p. Для того, чтобы создать неизвестный более никому секретный ключ, оба абонента генерируют большие случайные числа: первый абонент — число a, второй абонент — число b. Затем первый абонент вычисляет значение A = g^a mod p и пересылает его второму, а второй вычисляет B = g^b mod p и передаёт первому. Предполагается, что злоумышленник может получить оба этих значения, но не модифицировать их (т.е. у него нет возможности вмешаться в процесс передачи). На втором этапе первый абонент на основе имеющегося у него a и полученного по сети B вычисляет значение, B^a mod p = g^(a\*b) mod p, а второй абонент на основе имеющегося у него b и полученного по сети A вычисляет значение A^b mod p = g^(a\*b) mod p. Как нетрудно видеть, у обоих абонентов получилось одно и то же число: K = g^(a\*b) mod p. Его они и могут использовать в качестве секретного ключа, поскольку здесь злоумышленник встретится с практически неразрешимой (за разумное время) проблемой вычисления g^(a\*b) mod p по перехваченным g^a mod p и g^b mod p, если числа p, a, b выбраны достаточно большими.

**Преимущество** — никто не сможет скомпрометировать секретное сообщение. Полученный ключ может быть использован для шифрования по любым доступным сторонам алгоритма.

**Недостаток** — два различных независимых цикла шифрования. Отсутствие взаимной аутентификации сторон.

**Задание №3**

*Рассказать процесс работы алгоритма Эль-Гамаля.*

Схема была предложена Тахером Эль-Гамалем в 1984 году.Эль-Гамаль разработал один из вариантов алгоритма Диффи-Хеллмана. Он усовершенствовал систему Диффи-Хеллмана и получил два алгоритма, которые использовались для шифрования и для обеспечения аутентификации. В отличие от RSA алгоритм Эль-Гамаля не был запатентован и, поэтому, стал более дешевой альтернативой, так как не требовалась оплата взносов за лицензию. Считается, что алгоритм попадает под действие патента Диффи-Хеллмана.

Пусть имеются абоненты А, В, С, ..., которые хотят передавать друг другу зашифрованные сообщения, не имея никаких защищенных каналов связи. Фактически здесь используется схема Диффи-Хеллмана, чтобы сформировать общий секретный ключ для двух абонентов, передающих друг другу сообщение, и затем сообщение шифруется путем умножения его на этот ключ. Для каждого следующего сообщения секретный ключ вычисляется заново.

*Шифрование*

Допустим, что нужно зашифровать сообщение М = 5.  
  
1. Произведем генерацию ключей: пусть p = 11, g = 2. Выберем x = 8 - случайное целое число x такое, что 1 < х < (р - 1).

Вычислим y = g^x mod p = 2^8 mod 11 = 3.

Итак, открытым является тройка (p, g, y) = (11, 2, 3), а закрытым ключом является число x = 8.

2. Выбираем случайное целое число k такое, что 1 < k < (p - 1). Пусть k = 9.

3.Вычисляем число A = g^k mod p = 2^9 mod 11 = 512 mod 11 = 6.

4.Вычисляем число B = y^k\*M mod p = 3^9\*5 mod 11 = 19683\*5 mod 11 = 9

Полученная пара (a, b) = (6, 9) является шифротекстом.

*Расшифрование*

Необходимо получить сообщение M = 5 по известному шифротексту

(a, b) = (6, 9) и закрытому ключу x = 8.

Вычисляем M по формуле M = b\*(a^x)^-1 mod p = 9\*(6^8)^-1 mod 11 = 5.

Получили исходное сообщение M = 5.

**Преимущество** — вероятностный характер шифрования — большая стойкость.

**Недостаток** — удвоение длины зашифрованного текста по сравнению с начальным текстом. Само сообщение и ключ не определяют шифртекст однозначно.

**Вывод**:

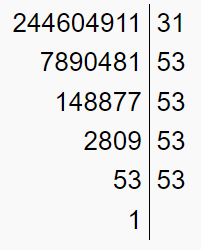
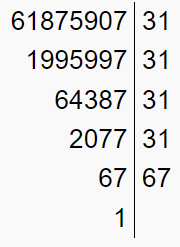
Овладение основными криптографическими алгоритмами асимметричного шифрования (RSA, Диффи-Хеллмана, Эль-Гамаля).Они являются более громоздкими и затратными по ресурсам .

**Практическое задание 6**

**Вариант №13**

**Задание №1**

Найти канонические разложения чисел а и b.

*а* = 244604911, *b* = 61875907, тогда  
  

Итого, 627 = 534 ∙ 31; 399 = 67 ∙ 314

**Задание №2**

1 задание + Вычисляем последовательно разности  до получения последней ненулевой разности, которая и совпадает с НОД *(a, b).*

Найти НОД (a, b) пользуясь:

a) алгоритмом Евклида,

б) разложением чисел на простые множители.

Алгоритм Евклида

*а* = 244604911, *b* = 61875907, тогда

244604911 = 61875907 ∙ 3 + 58977190  
61875907 = 58977190 ∙ 1 + 2898717  
58977190 = 2898717 ∙ 20 + 1002850  
2898717 = 1002850 ∙ 2 + 893017  
1002850 = 893017 ∙ 1 + 109833  
893017 = 109833 ∙ 8 + 14353  
109833 = 14353 ∙ 7 + 9362  
14353 = 9362 ∙ 1 + 4991  
9362 = 4991 ∙ 1 + 4371  
4991 = 4371 ∙ 1 + 620  
4371 = 620 ∙ 7 + 31  
620 = 31 ∙ 20 + 0

Разложение чисел на простые множители

Наибольшим общим делителем будет произведение одинаковых множителей, входящих, как в одно, так и в другое разложения чисел: НОД (244604911; 61875907) = 31

**Задание №3**

Собственно алгоритма Евклида - прогонки вниз и прогонки вверх – последовательного выражения остатков в каждом из шагов предыдущего этапа

С помощью расширенного алгоритма Евклида найти целые u, v, удовлетворяющие соотношению Безу: au + bv = НОД (a, b)

31 = 4371 – 620∙7 = 4371 – (4991 – 4371)∙7 = 4371∙8 – 4991∙7 = (9362 – 4991)∙8 – 4991∙7 = 9362∙8 – 4991∙15 = 9362∙8 – (14353 – 9362)∙15 = 9362∙23 – 14353∙15 = (109833 – 14353∙7)∙23 – 14353∙15 = 109833∙23 - 14353∙176 = 109833∙23 – (893017 – 109833∙8)∙176 = 109833∙1431 – 893017∙176 = (1002850 – 893017)∙1431 – 893017∙176 = 1002850∙1431 – 893017∙1607 = 1002850∙1431 – (2898717 – 1002850∙2)∙1607 = 1002850∙4645 – 2898717∙1607 = (58977190 – 2898717∙20)∙4645 – 2898717∙1607 = 58977190∙4645 – 2898717∙94507 = 58977190∙4645 – (61875907 – 58977190)∙94507 = 58977190∙99152 – 61875907∙94507 = (244604911 – 61875907∙3)∙99152 – 61875907∙94507 = 244604911∙99152 – 61875907∙391963 = 244604911∙(99152) + 61875907∙(-391963) = 31

a = 244604911, u = 99152  
b = 61875907, v = -391963

**Задание №4**

(Малая теорема Ферма). Пусть *p –* простое число и целое число *a* не делится на . Тогд*а .*

Найти остаток от деления данного числа на простое.

Найти остаток от деления  на 17.

**2001 = 17∙117 + 6; 2001 ≡ 6 mod 17**

**φ (17) = 17(1 - 1/17) = 16; 616 ≡ 1 mod 17; 1995 = 16∙124 + 11 => 61995 = 616∙124 + 11 = (616)124 ∙ 611 ≡ 362797056 mod 17; 362797056 = 21341003∙17 + 5 => 362797056 ≡ 5 mod 17**

**Ответ: r(20011995)17 = 5**

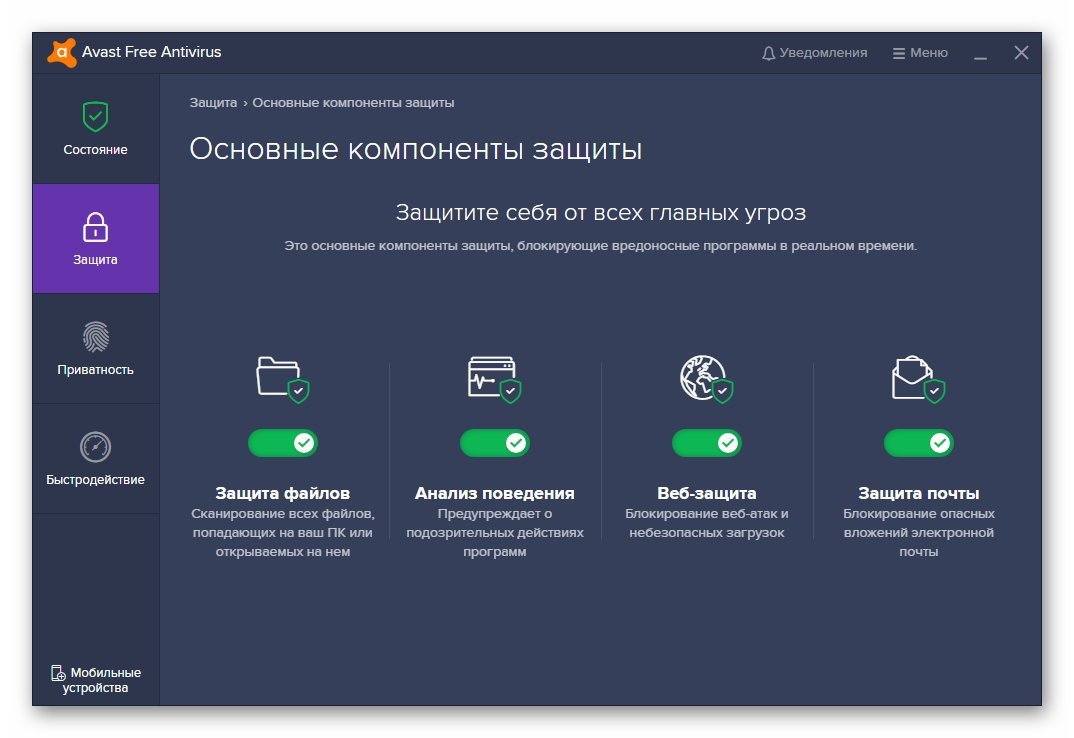
**Вывод: Были повторены определенные задачи из предмета математика. А также алгоритмы Эйлера и Евклида здорово помогают найти НОД чисел. А с помощью малой теоремы Ферма и формулы Эйлера можно найти остаток от деления больших чисел.**

**Практическая работа 7**

**AVAST**

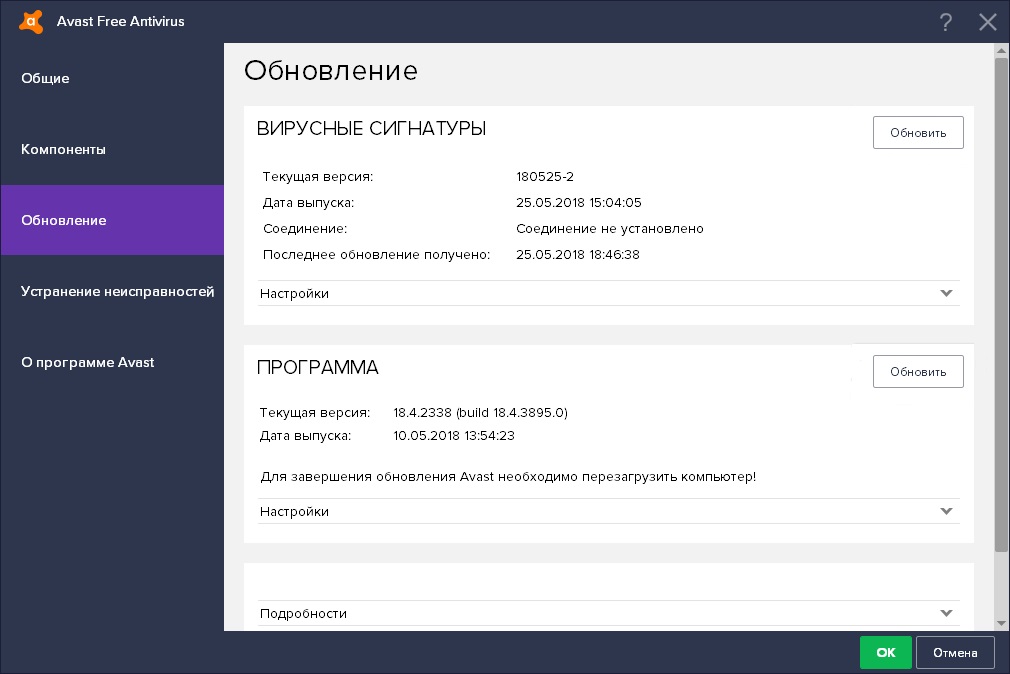
**Задание №1**

Установить и настроить антивирусную программу



**Задание №2**

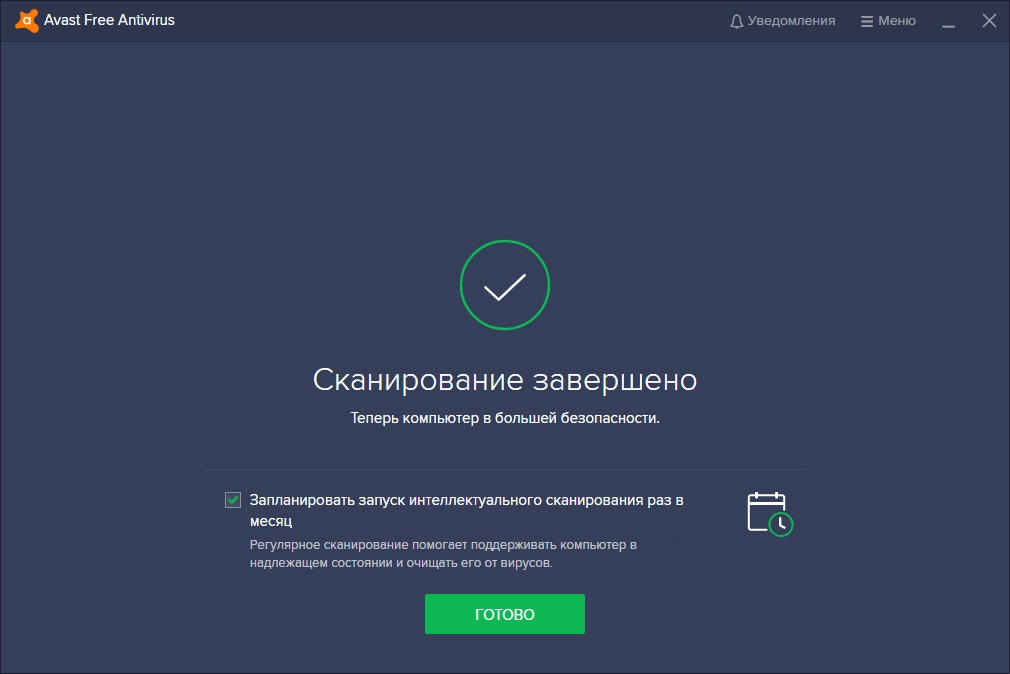
Обновить базу данных сигнатур вирусов.



Каждые 4 часа он сам их обновляет

**Задание №3**

Выполнить сканирование дисков



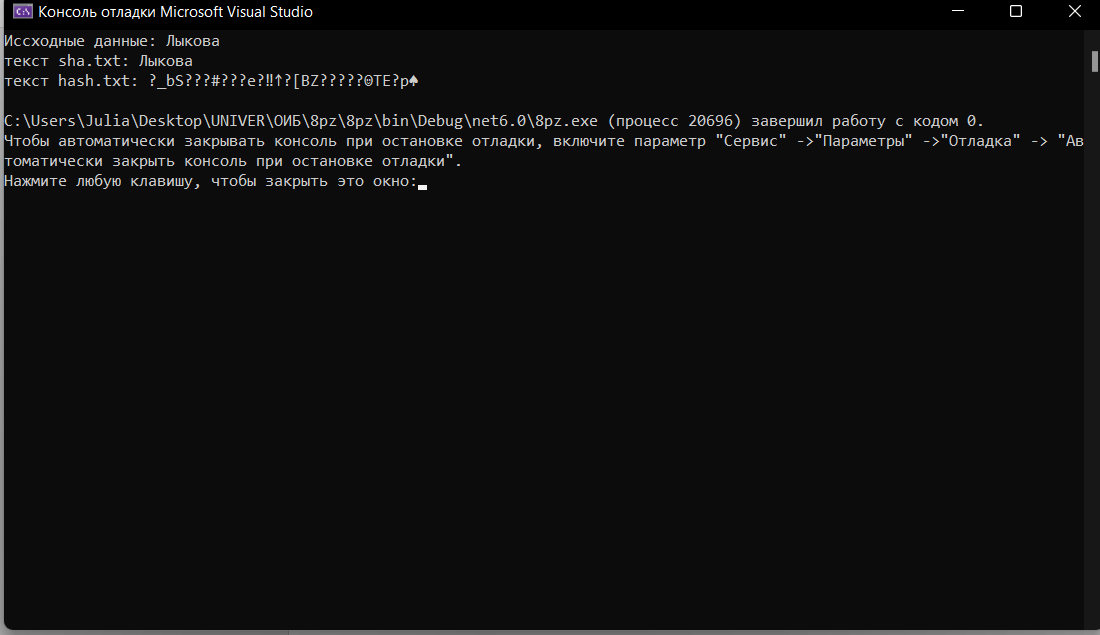
**Практическое задние 8**

Цель: Изучить модель криптографии .NET Framework, основные классы и структуры данных, разработать приложение для шифрования файлов использующих симметричные и ассиметричные алгоритмы шифрования

System.Security.Cryptography(программные интерфейсы CryptoApi (CAPI) и Cryptography Next Generation API)/CryptoStream/тип доступа.Функций Encrypt и Decrypt.Инициализации экземпляра-ComputeHash

**Задание №1**

Выполнить шифрование, дешифрование и хеширование(отсутствия необходимости в обратном преобразовании). своей фамилии по указанным алгоритмам. Используя функции работы с файлами сохранить ключи шифрования, результаты шифрования и хеширования.





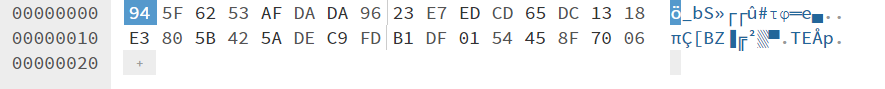
**Задание №2**

Для выше указанных алгоритмов используя Hex-редактор продемонстрировать ключи шифрования, зашифрованные и захешированные данные.

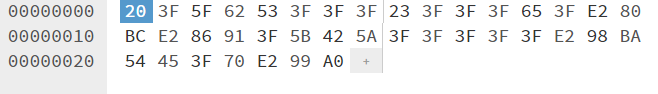
Лыкова



Hash.txt

****

sha.txt



?\_bS???#???e?‼↑?[BZ?????☺TE?p♠

**Практическая работа №9**

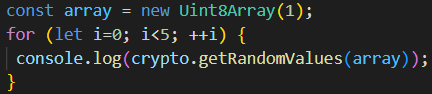
Цель: Изучить интерфейс **SubtleCrypto**, Основные классы и структуры данных, разработать приложение для шифрования файлов использующих симметричные и ассиметричные алгоритмы шифрования

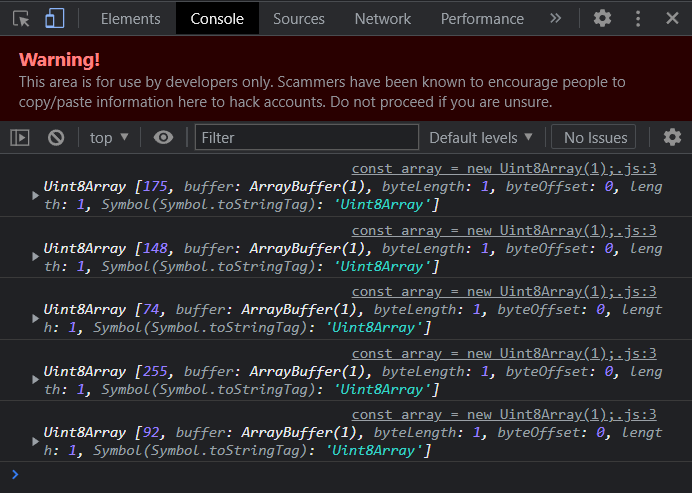
Интерфейс SubtleCrypto Web Crypto API предоставляет ряд низкоуровневых криптографических функций

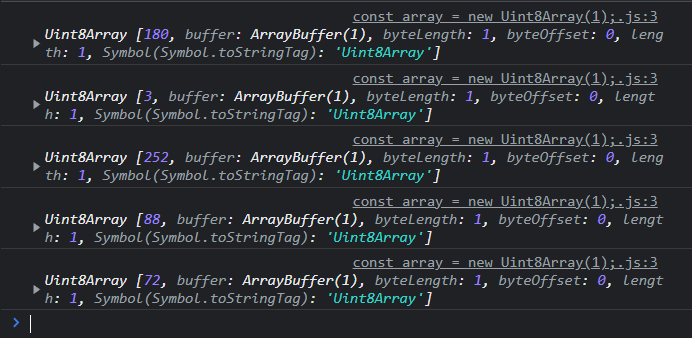
**Задание №1**

Доступ к функциям SubtleCrypto осуществляется через объёкт Crypto.subtle.Math.random()-для криптографии неприемлемо. Для решения данной задачи решить эту проблему, криптографически безопасный генератор псевдослучайных чисел (cryptographically secure pseudorandom number generator, CSPRNG) дополнительно включает в себя источник энтропии качестве входных данных, такие системных свойств, которые проявляют непредсказуемое поведение.

Выполнить генерацию и вывод в консоль случайный чисел.





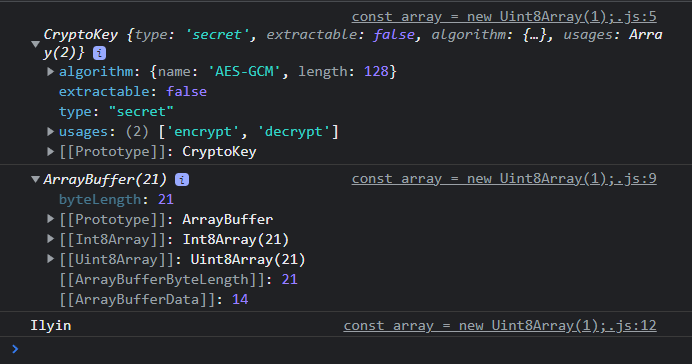


**Задание №2**

Выполнить шифрование(SubtleCrypto.encrypt()), дешифрование (SubtleCrypto.decrypt()) и хеширование(SubtleCrypto.digest()) своей фамилии по указанным алгоритмам.

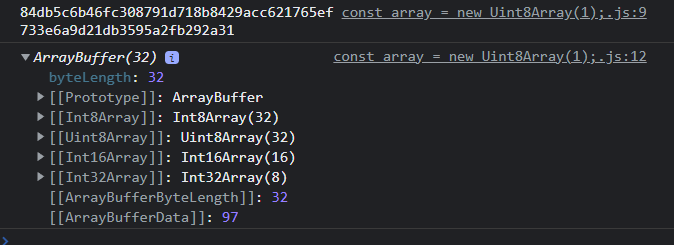


Результат хеширования обычно используется в 16-ой форме.Объект SubtleCrypto позволяет использовать как открытый ключ, так и симметричные алгоритмы для шифрования и дешифрования сообщений



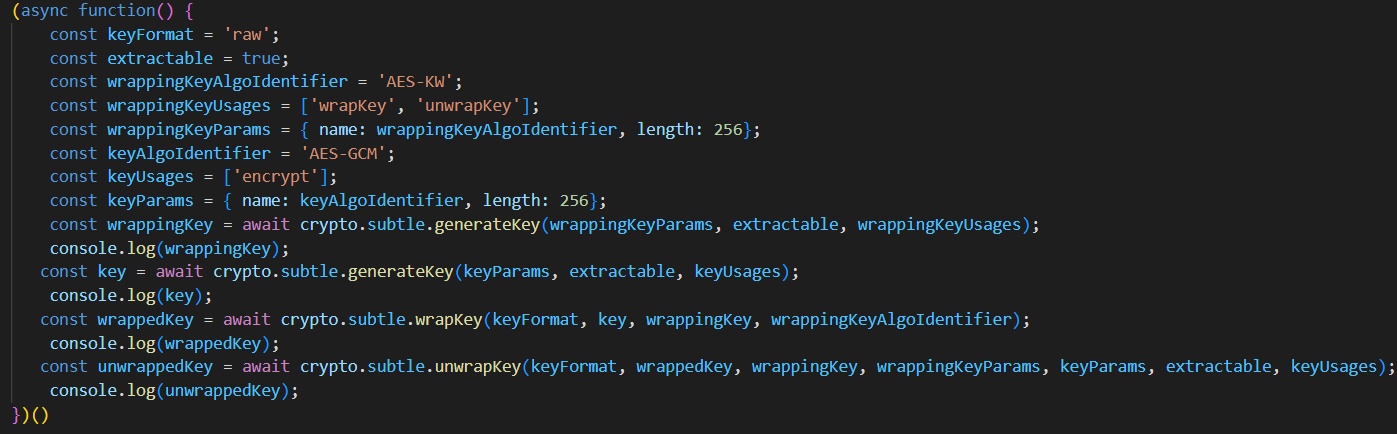






**Задание №3**

Продемонстрировать упаковку(SubtleCrypto.wrapKey()) и распаковку(SubtleCrypto.unwrapKey()) ключа, полученного в предыдущем задании используя алгоритм AES-KW



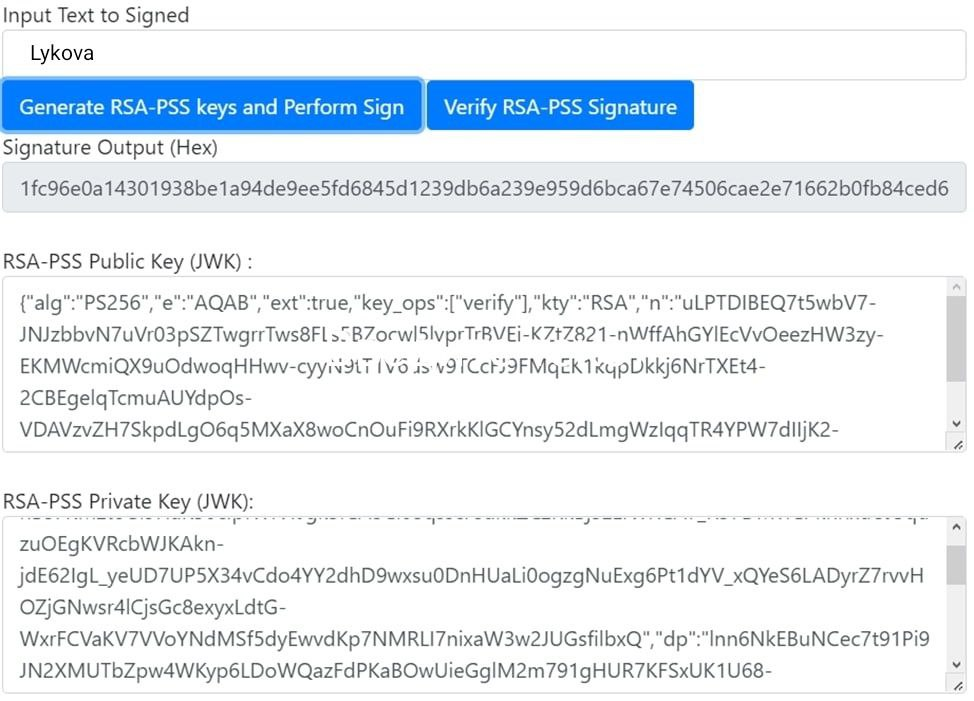
Для переноса ключа требуется строка форматирования, экземпляр CryptoKey для переноса, и объект params для указания алгоритма.

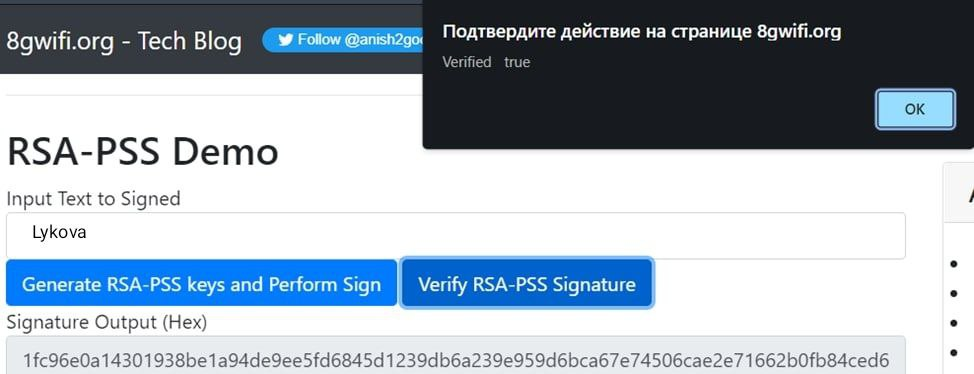


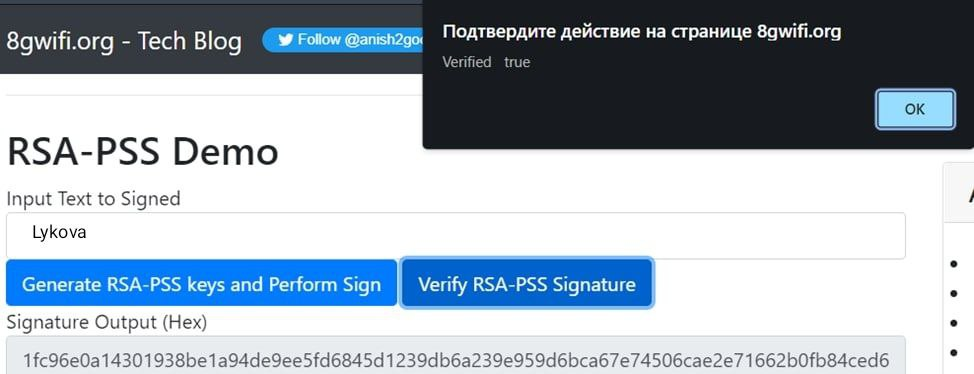
**Задание №4**

Объект SubtleCrypto позволяет использовать алгоритмы с открытым ключом для генерации подписей с использованием закрытого ключа или для проверки подписей с использованием открытого ключа. SubtleCrypto.sign() и SubtleCrypto.verify().Для подписания сообщения требуется объект params, частный CryptoKey, ArrayBuffer или ArrayBufferView для подписи.

Выполнить процедуру подписи сообщения и проверку подлинности с использованием RSA-PSS или ECDSA на выбор.







**Вывод:** Web Crypto API – это интерфейс, позволяющий использовать криптографические примитивы для построения систем с использованием криптографии. Данный интерфейс включает в себя возможность генерировать, использовать и применять пары криптографических ключей; шифровать и дешифровать сообщения; надежно генерировать случайные числа. Существует множество алгоритмов шифрования, однако каждый из них отличается своей надёжностью. Если безопасность данных жизненно необходима, можно воспользоваться алгоритмом SHA-256. Упаковка и распаковка ключей используется для обеспечения безопасной и надёжной передачи данных по незащищённому каналу.