Муниципальное автономное общеобразовательное учреждение
Полевского городского округа
«Средняя общеобразовательная школа-лицей № 4 «Интеллект»

Математические основы криптографии

Исследовательский проект

Работу выполнил ученик 8Г класса
Кайль Лев Владимирович
Руководитель: учитель математики
Наталья Михайловна Бажова

Полевской, 2022 год

ОГЛАВЛЕНИЕ:

ВВЕДЕНИЕ

В данном проекте я расскажу о математических основах криптографии.

Введём некоторые слова: **исходное сообщение** – то, что подлежит шифровке. **Шифротекст** (шифрованное сообщение) – сообщение, прошедшее процесс шифровки. **Шифр** — это сам алгоритм, по которому мы преобразовываем сообщение. **Ключ** — это компонент, на основе которого можно произвести шифрование или дешифрование.

Криптография – наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним), целостности данных (невозможности незаметного изменения информации), аутентификации (проверки подлинности авторства или иных свойств объекта), шифрования (кодировка данных).

Сейчас жизнь устроена так, что между людьми происходит интенсивный обмен информацией, причем часто на громадные расстояния. Люди давно поняли, что информация может быть настоящим сокровищем, и, поэтому, много усилий тратится на обеспечение её безопасности.

Актуальность проекта: в настоящее время методы криптографии используются для обеспечения информационной безопасности не только государства, но и частных лиц, и организаций.

Задачи проекта:

- Изучить историю криптографии;
- Изучить значение теоремы криптографии в жизни;
- Реализовать современный способ криптографии;
- Показать гибкость криптографии при решении современных проблем.

Цели проекта: показать значимость и гибкость современной криптографии в повседневной жизни.

Гипотеза: современная криптография гарантирует безопасность и сохранность данных при передаче.

ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1. ИЗ ИСТОРИИ

Криптография — одна из самых древних научных дисциплин. Первые методы шифрования (шифр Атбаша, шифр Цезаря и т. д.) были довольно просты и «взламывались» (т. е. по шифротексту можно получить исходное сообщения, не зная ключа). Именно эти алгоритмы являются первыми, которые использовались для шифрования.

Основная проблема данных шифров – полная некриптостойкость (абсолютная криптостойкость – криптосистема не может быть раскрыта ни теоретически, ни практически даже при наличии у атакующего бесконечно больших вычислительных ресурсов; достаточно криптостойкая система – потенциальная возможность вскрыть шифр существует, но при выбранных параметрах и ключах шифрования. На практике атакующий на современном этапе развития технологий не может обладать достаточными вычислительными ресурсами для вскрытия шифра за приемлемое время).

Далее появляются более сложные алгоритмы: шифр Вернама (XOR), являющийся одним из первых методом шифрования одноразовым блокнотом (т. е. обладает абсолютной криптостойкостью), Шифр Энигмы.

После появления компьютера появляются более сложные алгоритмы: AES, SEAL, RC (системы симметричных ключей); DSS, ElGamal и RSA (системы ассиметричных ключей).

1.2. ЗНАЧИМОСТЬ КРИПТОГРАФИИ В ПОВСЕДНЕВНОЙ ЖИЗНИ

В современной жизни почти у каждого есть социальная сеть, будь то WhatsApp, Telegram или ВКонтакте. Именно там и происходит общение людей со всего мира. Но если бы оно не шифровалось, то любое сообщение можно было бы перехватить и прочитать. Также при проведении транзакций в банке данные ваших карт шифруются, чтобы мошенники не смогли их узнать.

1.3. ТЕОРЕМА И ФУНКЦИЯ ЭЙЛЕРА.

Теорема Эйлера: если a и m взаимнопросты, то $a^{\varphi(m)} \equiv 1(mod\ m)$, где $\varphi(m) -$ функция Эйлера.

Отсюда следует $a^{\varphi(m)-1} * a \equiv 1\ (mod\ m) \Rightarrow a^{\varphi(m)-1} \equiv a^{-1}(mod\ m)$. Т. е. данная теорема позволяет найти число, обратное данному числу по модулю.

Функция Эйлера $\varphi(n)$ – мультипликативная арифметическая функция, значение которой равно количеству натуральных чисел, меньших n и взаимно простых с ним.

Рассмотрим число $a = q * w$, где $q, w \in P$. Требуется найти $\varphi(a)$. Если числа q и w будут большими (например, больше 2^{256}), то нахождение функции Эйлера от такого числа займёт бесконечно много времени (под этим понятием понимается настолько большое время, что его нельзя выполнить за разумное время). В данном случае, нахождение функции Эйлера от числа a на обычном компьютере на языке C++ (примерно 10^8 операций в секунду) выполнится за время, превышающее время существования Вселенной в миллионы раз.

Поэтому можно применить свойство мультипликативности функции Эйлера: $\varphi(a) = \varphi(q * w) = \varphi(q) * \varphi(w)$. Также заметим, что q и w – простые числа (по условию). Тогда функция Эйлера равна данному числу минус 1 (т. к. само число не является взаимно простым с самим собой).

Т.е. $\varphi(a) = (q - 1) * (w - 1)$, что легко можно быстро вычислить на компьютере (за единицу времени).

На практике, теорема Эйлера не используется для нахождения обратного числа по модулю из-за огромных затрат по времени.

1.4. РАСШИРЕННЫЙ АЛГОРИТМ ЕВКЛИДА

Существует два основных метода нахождения НОД для двух чисел: рекуррентный и итеративный.

Рассмотрим второй метод. Пусть есть два числа: a, b $(a \geq b)$. Выразим число $a = b * q_0 + r_0$. Тогда следующим шагом «сдвинем» значения a, b, r влево. Т.е. получим следующее равенство: $b = r_0 * q_1 + r_1$ и т. д. Базовый случай (т. е. крайняя итерация) – $r_n = 0$. Тогда НОД двух чисел является r_{n-1}.

Рассмотрим формулу: $ax + by = \text{НОД}(a, b)$

Если a и $b -$ взаимно простые, то формула упрощается до следующего вида: $ax + by = 1$.

Представим, что b является пределом конечного кольца (или поля), т.е. является его порядком. Тогда данная формула принимает следующий вид: $ax + + by \equiv 1(mod\ b)$. Т.к. by является нулевым элементом, то формула принимает следующий вид: $ax \equiv 1(mod\ b) \Rightarrow x \equiv a^{-1}(mod\ b)$ (формула 1). Т.е. зная коэффициент x можно получить обратное число по модулю. А расширенный алгоритм Евклида даёт возможность быстро вычислить данный коэффициент.

Расширенный алгоритм Евклида предполагает наличие 4 констант: $x_0 = 1, x_1 = 0, y_0 = 0, y_1 = 0$.

Рассмотрим нахождение НОД(88; 36) с помощью итеративного метода:

$88 = 36 * 2 + 16$

$36 = 16 * 2 + 4$

$16 = 4 * 4 + 0$. Т.к. $r = 0$, то НОД равен 4.

Формула расширенного алгоритма Евклида: $x_i = x_{i-2} - q_i * x_{i-1}$. Тогда $x_2 = 1 - 2 * 0 = 1. x_3 = 0 - 2 * 1 = -2$

Аналогично для y: $y_2 = 0 - 2 * 1 = -2. y_3 = 1 - 2 * (-2) = 5.$

Проверим данные числа: $88 * (-2) + 36 * 5 = 4 = \text{НОД}(88; 36)$.

Заметим, что к данным числам нельзя подобрать обратное число по модулю, т.к. они не взаимно просты.

Рассмотрим последние коэффициенты для взаимно простых чисел 101 и 36. $x_4 = 5, y_4 = -14$. Тогда по формуле 1: $36 * (-14) \equiv 1(mod\ 101)$. Т.к. -14 в данной группе не существует, то надо преобразовать -14 в противоположный элемент по аддитивной группе. Т.е. 87 – обратный элемент к 36 по модулю 101.

1.5. МЕТОД ШИФРОВАНИЯ RSA

Начнём с генерации публичного и приватного ключей (т.к. RSA – метод ассиметричного шифрования).

Возьмём два таких числа p и q, что $p, q \in \mathbb{P}, p \neq q$. Далее выявим функцию произведения простых чисел: $n = pq$.

Далее выявим функцию Эйлера от числа n: $\varphi(n)$. Как было сказано выше, то простой перебор чисел до $n - 1$ является крайне неэффективен, поэтому применим свойство мультипликативности функции Эйлера: $\varphi(n) = (p - 1)(q - -1)$.

Создадим открытую экспоненту. Открытая экспонента является непосредственно частью открытого ключа. Для открытой экспоненты должны выполнятся следующие условия: $e \in \mathbb{P}$, $\text{НОД}\big(e, \varphi(n)\big) = 1, e < \varphi(n)$.

Чаще всего в качестве числа e выступает множество простых чисел Ферма. $e \in \{F_0, F_1, F_2, F_3, F_4\}$. Число Ферма вычисляется следующим образом: $F_n = 2^{2^n} + + 1$. Тогда $F_0 = 3, F_1 = 5, F_2 = 17, F_3 = 257, F_4 = 65537$. Все пять чисел являются простыми. Ферма предполагал, что и дальше будут идти простые числа, но Эйлер доказал то, что шестое и далее числа будут составными. Для открытой экспоненты используются числа Ферма из-за их «хорошего» представления в двоичном виде: $F_0 = 11_2, F_1 = 101_2, F_2 = 10001_2, F_3 = 1\ k(7)\ 1_2, F_4 = 1\ k(15)1_2,$ где $k -$ кол − во нулей в числе. Т.к. в данных числах меньше всего единиц, то алгоритм можно применить алгоритм быстрого возведения в степень для увеличения быстродействия.

Далее необходимо вычислить закрытую экспоненту d такую, что она удовлетворяет следующему равенству: $ed \equiv 1(mod\ \varphi(n))$. Т.е. требуется найти обратное число к открытой экспоненте по модулю $\varphi(n)$. Для этого можно применить расширенный алгоритм Евклида. Если же разрабатывается только теоретическая реализация, то можно использовать теорему Эйлера (на практике её использование крайне неэффективно, рассматривается в практической части).

Пара чисел $\{e, d\} - public\ key$ (публичный ключ), $\{d, n\} - private\ key$ (приватный ключ). После создания ключей рекомендуется удалять простые числа p и q, а также результат функции Эйлера для экономии памяти. Также если злоумышленники узнают хотя бы одно из значений, то они смогут быстро вычислить закрытую экспоненту, т.е. алгоритм будет взломан.

Следующий этап: шифрования/расшифрование.

Есть сообщение m, которое требуется зашифровать. Тогда шифротекст c получается следующим образом: $m^e \equiv c\ (mod\ n)$. Для расшифрования применяем следующую формулу: $c^d \equiv m\ (mod\ n)$. Т.е. мы получили исходное сообщение.

Рассмотрим доказательство работы данного метода.

Как было сказано ранее, $ed \equiv 1(mod\ \varphi(n))$. Тогда существует k такое, что $ed = 1 + k\varphi(n)$ (ф. 1). Представим, что $\varphi(n)$ является порядком некого кольца (поля). Тогда $ed \equiv 1(mod\ \varphi(n))$, что является верным.

Рассмотрим процесс шифрования и расшифрования: $m = (m^e)^d \equiv m^{ed}\ m^{ed}$(mod n). Тогда вместо ed подставим правую часть ф.1. Тогда $m^{ed} \equiv \equiv m^{1+k\varphi(n)}(mod\ n)$.

Из теоремы Эйлера следует следующее: $a^{\varphi(z)+1} \equiv a(mod\ z)$. Отсюда следует $m^{1+k\varphi(n)} \equiv m^{1+\varphi(n)} \equiv m(mod\ n)$ ч.т.д.

1.6. СЛОЖНОСТЬ КРИПТОСИСТЕМЫ RSA

Операция шифрования: $m^e \equiv c\ (mod\ n)$. Если попытаться выполнить обратную операцию, то получим следующее: $m \equiv \sqrt[e]{c}\ (mod\ n)$, но для данного вычисления нет общего алгоритма, следовательно таким способом нельзя вычислить исходный текст.

Данная проблема решается при вычислении закрытой экспоненты. А это можно сделать только одним путём – разложить число n на множители. Задачи вычисления корня по модулю и факторизации больших чисел являются сложными вычислительными задачами. В настоящий момент никто не придумал

эффективных алгоритмов, выполняющихся на машине Тьюринга, которые позволяли бы решить такую задачу.

Чтобы найти закрытую экспоненту требуется найти обратное число к открытой экспоненте по модулю $\varphi(n)$: $d = e^{-1}\ (mod\ \varphi(n))$. Для этого требуется возвести открытую экспоненту в функцию Эйлера от результата функции Эйлера от числа n, а для этого надо знать, как число n раскладывается на простые множители. Перебор всех чисел от 0 до $\mathrm{n} - 1$ и проверка, являются ли они взаимно простыми или нет, будет крайне неэффективна (число n может занимать до 4096 бит).

Расширенный алгоритм Евклида также неприменим из-за проблемы факторизации числа n.

Следовательно, при использовании достаточно больших простых чисел p и q и большой открытой экспоненте (пятое число Ферма), алгоритм является достаточно криптостойким.

2. ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1. ОСНОВНЫЕ ПОНЯТИЯ

В практической части я реализую метод шифрования RSA, применение различных алгоритмов для оптимизации шифрования на интерпретируемом языке Python в среде разработки PyCharm.

Основные понятия:

- Компилятор — программа, переводящая написанный на языке программирования текст в набор машинных кодов.
- Процессорное время — время, затраченное процессором компьютера на обработку задачи (программы).
- Код — текст компьютерной программы на каком-либо языке программирования.
- Входные данные – некоторые значения, которые получает программа в начале.
- Выходные данные – результат работы программы.

2.2. РЕАЛИЗАЦИЯ

В первой версии кода я реализовал полноценное шифрование RSA.

Простые числа p и q фиксированные, создаются с помощью онлайн-генератора. В данной версии закрытая экспонента вычисляется с помощью теоремы Эйлера.

Входные данные: $p = 3001, q = 3083, m = 123987$. Открытая экспонента $e = 65537$, закрытая экспонента $d = 39834113$, функция Эйлера $\varphi(n) = = 51164320$.

Выходные данные: зашифрованное сообщение $c = 26245995$. Вывод компилятора на проверку корректности зашифровки: True. Т.е. сообщение зашифровано верно, следовательно алгоритм составлен верно.

Но процессорное время выполнения программы: 5302 секунды, а зашифрованное сообщение должно быть меньше 24 бит, что мало даже для одного слова в двоичном представлении. Т.е. для увеличения максимального размера

шифруемого сообщения требуется увеличивать простые числа, но тогда это приведёт к огромным затратам времени.

Во второй версии кода был реализован расширенный алгоритм Евклида, что ускорило вычисление закрытой экспоненты до минимально возможного времени. Далее были взяты простые числа длиной 300 цифр (997 бит). И вычисление закрытой экспоненты затратило очень мало времени (программа не успела замерить время).

Входные данные: $p, q -$ простые числа длиной 300 цифр, исходное сообщение длиной 300 цифр.

Вывод компилятора на проверку корректности зашифровки: True. Т.е. сообщение зашифровано верно, следовательно алгоритм составлен верно.

Суммарное время выполнения программы: 0.031 секунда, что является хорошим результатом по сравнению с первой версией.

В третьей версии кода была реализована функция кодировки самого сообщения. Т.е. каждый символ кодировался в бинарный код, и уже полученный текст шифровался. Но из-за этого размер шифруемого сообщения снизился в 7 раз (т.к. каждый символ кодируется 7 битами).

Входные данные: $p, q -$ простые числа длиной 300 цифр, исходное сообщение: "Это тестовый пример для проекта по математике".

Вывод компилятора на проверку корректности зашифровки: True. Т.е. сообщение зашифровано верно, следовательно алгоритм составлен верно.

Данная система имеет ключ почти в 1000 бит, что является достаточным уровнем защиты информации.

3. УЯЗВИМОСТЬ КРИПТОСИСТЕМЫ RSA

Система RSA используется не только для шифрования, но и для цифровых подписей. Цифровая подпись осуществляется следующим образом: текст шифруется с помощью приватного ключа, а расшифровка – с помощью публичного ключа. Данный алгоритм является верным, т.к. при доказательстве корректности шифрования системой RSA не учитывается порядок применения ключей.

Обозначим шифрование следующим образом: $E_a(x)$ – сообщение x зашифровано (проверка подписи) пользователем a.

Обозначим расшифровку следующим образом: $D_a(x)$ – сообщение x расшифровано (подписано) пользователем a.

Пусть в некой системе есть Алиса (А) и Боб (Б). Опишем их систему общения.

Пусть Алиса хочет передать сообщение m Бобу, которое будет зашифровано и подписано алгоритмом RSA. Также у каждого из участников есть публичный и приватный ключи, где публичный известен всем, приватный – только его обладателю.

Действия Алисы при отправке: А | $E_{\text{Б}}(D_A(m)) \longrightarrow$ Б. Алиса сперва подписывает сообщение своим приватным ключом, потом шифрует публичным ключом Боба.

Действия Боба при получении: Б | $E_{\text{А}}(D_{\text{Б}}(c)) = E_{\text{А}}\left(D_{\text{Б}}(E_{\text{Б}}(D_A(m)))\right) = m$. Т.е. сперва Боб расшифровывает своё сообщение своим приватным ключом, потом проверяет подпись публичным ключом Алисы.

Также есть вторая функция протокола – подписание сообщения (т.е. основная цель – подпись сообщения).

$A \mid E_{\text{Б}}(D_A(m)) \longrightarrow$ Б .

Б | $E_{\text{А}}(D_{\text{Б}}(m)) \rightarrow$ А.

Пусть есть третий пользователь-злоумышленник Ева. Рассмотрим ситуацию: Алиса отправляет сообщение Бобу. А | $E_{\text{Б}}(D_A(m)) \longrightarrow$ Б

Ева перехватывает сообщение и без модификаций отправляет Бобу на подпись. Тогда Боб подписывает своей подписью и шифрует публичным ключом Евы (т.к. сообщение пришло именно от неё).

Б | $E_E\left(D_Б\left(E_Б(D_A(m))\right)\right) \longrightarrow E$. Тогда Ева расшифровывает полученное сообщение своим приватным ключом. Тогда остаётся следующая часть сообщения: $D_Б\left(E_Б(D_A(m))\right)$. Заметим, что последние два действия являются противоположными (шифрование – расшифровка), тогда эти действия просто удаляются. Оставшаяся часть сообщения: $D_A(m)$. Остаётся лишь проверить подпись Алисы (расшифровать сообщение приватным ключом) и получим исходное сообщение m, т.е. алгоритм был взломан.

Возможные решения данной проблемы:

1. Использование разных алгоритмов (например, для шифрования – RSA, для подписи – схему ElGamal). Тогда требуется хранить два ключа.
2. Использовать две разные пары ключей RSA. Также требует больше места.
3. Для одного и того же значения n создать две разные открытые экспоненты (например, для шифрования – пятое число Ферма, для подписи – третье число Ферма). Тогда для каждой открытой экспоненты будет своя закрытая экспонента. Данный способ лучше остальных, т.к. требует меньше всего места (т.к. для обеих экспонент одно и тоже число n).

4. ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

4.1. ШИФР ЦЕЗАРЯ

Условие:

Известно, что сообщение «Са фуьнчейс жцйъ тшрдсн, е йинтныесн – цйёд» является результатом с помощью шифра Цезаря с величиной сдвига 5 (русский алфавит). Расшифруйте сообщение.

Решение:

Т.к. при шифровке сдвиг равнялся 5, то при расшифровке он будет таким же, но в обратную сторону (т.е. влево).

Составим таблицу сдвига для расшифровки.

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
Ы	Ь	Э	Ю	Я	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

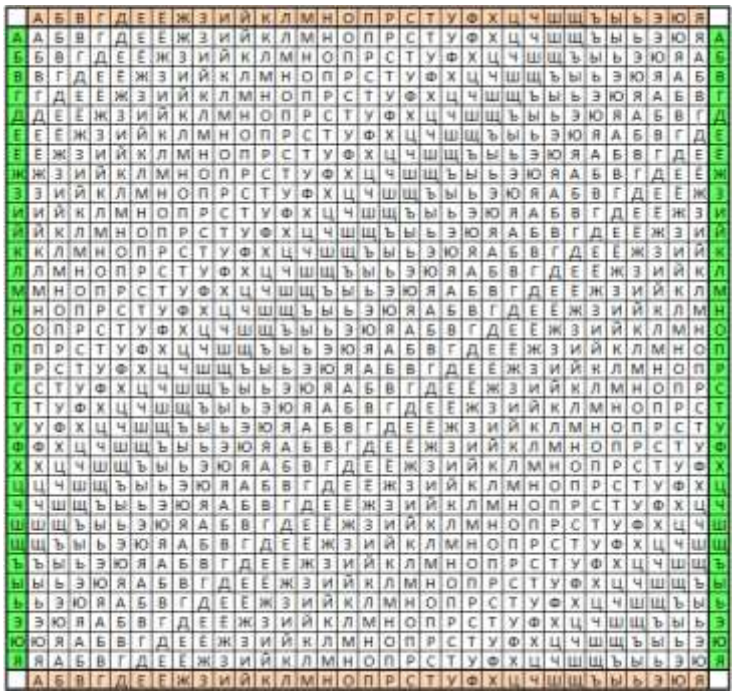
Расшифрованное сообщение: «Мы почитаем всех нулями, а единицами – себя».

Ответ. «Мы почитаем всех нулями, а единицами – себя».

4.2. ШИФР ВИЖЕНЕРА

Условие:

Дан текст, зашифрованный шифром Виженера: «ыуннтцпь ыьгбрв щб ооньтё», ключ – «нуль». Расшифруйте текст.



Решение:

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 русскому алфавиту таблица Виженера составляется из строк по

33 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 33 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

Например, предположим, что исходный текст имеет такой вид: «шифровка». Человек, посылающий сообщение, записывает ключевое слово «лимон» циклически до тех пор, пока его длина не будет соответствовать длине исходного текста: «лимонлим».

Первый символ исходного текста ("ш") зашифрован последовательностью л, которая является первым символом ключа. Первый символ зашифрованного текста ("д") находится на пересечении строки ш и столбца л в таблице Виженера. Точно так же для второго символа исходного текста используется второй символ ключа и т.д. Зашифрованный текст: «дсбяьнум».

Для расшифровки требуется выполнить обратную операцию – найти строку с первым символом ключа, в этой строке найти первый символ шифротекста. Тогда столбец, на пересечении которого со строкой образуется первый символ шифротекста, и есть исходный символ текст и т.д.

Пользуясь этим правилом, получаем первоначальный текст: «Навсегда ничего не бывает».

Ответ. «Навсегда ничего не бывает»

4.3. ШИФР ПЛЕЙФЕРА

Условие:

Дан зашифрованный текст «M HHYA KKPLHMZMALR» с ключом «HELLO». Известно, что шифрование выполнялось с помощью шифра Плейфера. Требуется расшифровать текст.

Решение:

Для данной задачи используется английский алфавит для упрощения вычислений.

Рассмотрим данный метод шифрования. 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 (буквы I и J записываются в одной ячейке). Ключевое слово записывается в верхней строке матрицы слева направо. Ключевое слово, дополненное алфавитом, составляет матрицу 5x5 и является ключом шифра.

Для того, чтобы зашифровать сообщение необходимо разбить его на биграммы (группы из двух символов), например «Hello World» становится «HE LL OW OR LD», 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:

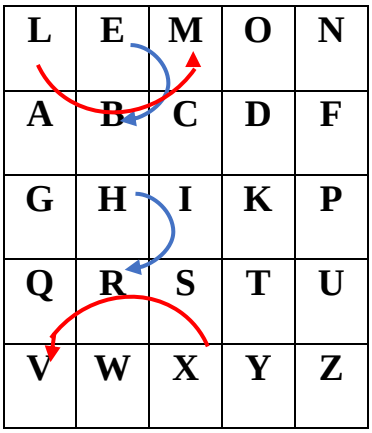
1.Если два символа биграммы совпадают, добавляем после первого символа «X», зашифровываем новую пару символов и продолжаем.

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, находящимися 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 (соседним по строке). Для расшифровки необходимо использовать инверсию этих четырёх правил, откидывая символы «X», если они не несут смысла в исходном сообщении.

Рассмотрим пример шифрования. Изначальный текст: «HELLO WORLD», кодовое слово: «LEMON». Составим таблицу 5x5.



Первая пара – «HE». Т.к. буквы в одной строке, то руководствуясь правилом 3 получаем шифротекст: «RB». Данный шаг указан в таблице синими стрелками.

Вторая пара – «LL». Т.к. буквы повторяются, то данную пару заменяем на «LX» и шифруем. Т.к. данные буквы стоят в разных строках и столбцах, то по правилу 4 (обозначено красным цветом). Таким образом вторая пара букв шифротекста: «MV».

Заметим, что следующая пара будет «LO». Такие действия повторяются, пока не закончится исходное сообщение. Итоговый шифротекст: «RBMVEN YEQECY».

Перейдём к расшифровке изначального текста: «M HHYA KKPLHMZMALR». Составим таблицу.

H	**E**	**L**	**O**	**A**
B	**C**	**D**	**F**	**G**
I	**K**	**M**	**N**	**P**
Q	**W**	**S**	**T**	**U**
V	**W**	**X**	**Y**	**Z**

Т.к. выполняется процесс обратный шифрованию, то все действия выполняются также в обратном направлении.

Первым действием руководствуемся правилом 4 (показано голубыми стрелками). И так для каждой пары.

В итоге мы получили текст: «I LOVE PINEAPXPLES». Заметим, что в данном текст присутствует символ «X», без которого слово имеет смысл. Также данный символ находится между двумя одинаковыми буквами. Отсюда можно сделать вывод, что символ является вспомогательным.

Ответ. «I LOVE PINEAPPLES»

4.4. ЗАДАЧИ ДЛЯ САМОСТОЯТЕЛЬНОГО РЕШЕНИЯ

1) Известно, что S является результатом шифрования с помощью шифра Цезаря со сдвигом n. Расшифруйте исходное сообщение при:

a) S = ”Йё, с дюнкё вёкчэ хгбя”, n = 19;
b) S = “Кэбл дёпщ, кэбл иыюёпщ, кэбл явнёпщ”, n = 30;
c) S = “Ъыкмуц ю шкь чшщнщ, к ъыкмоё шпэ”, n = 11

Ответ. a)Чу, я слышу пушек гром; b) Надо жить, надо любить, надо верить; c) Правил у нас много, а правды нет

2) Известно, что S является результатом шифрования с помощью шифра Виженера с ключом a. Расшифруйте исходное сообщение при:

a) S = “ъъъ аисйяьсж тжкехез шьч ыгуж, ужя лхщхк пшрке”,
a = “шифр”
b) S = “шчэ пяк юьчсуа, ауб пяк ч эюуаацч”, a = “неон”
c) S = “фсяак щит ыб фьоык учуьаххы ебтчйтжъ, узш кзвмяоич ыъйвю отйасх”, a = “жизнь”

Ответ. a)Всё приходит вовремя для того, кто умеет ждать: b)Кто всё поймет, тот всё и простит; c) ничто так не нужно молодому человеку, как общество умных женщин.

3) Известно, что S является результатом шифрования с помощью шифра Плейфера с ключом a. Расшифруйте исходное сообщение при:

a) S = “ Tpmlof gtlnont, k lhstal mq corl ra hfy gyox gwel xgcu gl rbwfr”, a = “book”
b) S = “Dumgzgrmnx, smmomiv epssy ekah ekztkmcmev”, a = “time”
c) S = “Ml uiv iiayq ilz qv ovev miayq ilz qc oonrz”, a = “love”

Ответ. A)Unlike animals, a person is able to get away from what he loves (В отличие от животных, человек уйти способен от того, что любит.); b)Gradually, reality turns into invalidity (Постепенно действительность превращается в недействительность); c)He who knows how to love knows how to wait (Умеющий любить умеет ждать).

5. ВЫВОД

Криптография применяется как в повседневной жизни (общение в соцсетях), так и в крупном бизнесе (безопасная передача данных банковских карт). Без криптографии невозможно было бы использование таких вещей как: криптовалюты, онлайн-магазины, мессенджеры.

Данное направление в течение продолжительного времени будет развиваться, т.к. увеличивается вычислительная мощность компьютеров, тем самым различные алгоритмы становятся всё менее криптостойкими.

Помимо математики и информатики в современной криптографии всё чаще используется физика. Её алгоритмы используют принципы квантовой суперпозиции, делая данные алгоритмы абсолютно криптостойкими.

Изначальная гипотеза подтвердилась при выполнении проекта.

6. СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

- Общедоступная многоязычная универсальная интернет-энциклопедия Википедия [Электронный ресурс]. – Режим доступа: https://www.Wikipedia.org/ – свободный – (08.11.2022).
- Научная электронная библиотека РусАрх [Электронный ресурс]. – Режим доступа: http://rusarch.ru/ – свободный – (08.11.2022).
- Информатика - подготовка к ЕГЭ [Электронный ресурс]. – Режим доступа: https://informatika-ege.ru/ – свободный – (08.11.2022).
- Образовательный интернет-проект [Электронный ресурс]. – Режим доступа: https://infourok.ru/ – свободный – (08.11.2022).
- Бесплатная поисковая система по полным текстам научных публикаций [Электронный ресурс]. – Режим доступа: https://scholar.google.com/ – свободный – (08.11.2022).
- Онлайн калькуляторы простых чисел [Электронный ресурс]. – Режим доступа: https://ilovecalc.com/ – свободный – (08.11.2022).