УДК 004.05 **DOI: 10.18413/2518-1092-2024-9-1-0-2**

Кузьминых Е.С.
Ильина С.П.
Маслова М.А.

АНАЛИЗ НЕПРОБИВАЕМЫХ АЛГОРИТМОВ ШИФРОВАНИЯ

Севастопольский государственный университет,
ул. Университетская, 33, г. Севастополь, 299053, Россия

e-mail: egor2014ru@mail.ru, sofi.ilina@mail.ru, mashechka-81@mail.ru

Аннотация
В данной статье исследуются непробиваемые алгоритмы шифрования, включая DES, RSA и AES, с установленной целью определения наиболее надежного и безопасного алгоритма шифрования. Статья анализирует характеристики каждого из шифров, такие как длина ключа, стойкость и актуальность, а также оценивает их популярность и применение в современных криптографических реалиях. В результате исследования сделан вывод, что AES является самым превосходным вариантом. Это объясняется тем, что DES и RSA уже были взломаны и имеют известные уязвимости, тогда как AES продолжает быть безопасным и лучшим из представленных шифров. Статья заключается в том, что AES рекомендуется для использования в системах, где требуется надежное и непроницаемое шифрование.
Ключевые слова: информационная безопасность; безопасность; кибербезопасность; непробиваемые алгоритмы шифрования; программирование; криптография; шифрование; RSA; DES; AES; ECB; CBC; взлом шифра; безопасность шифра; квантовые компьютеры; квантовая криптография
Для цитирования: Кузьминых Е.С., Ильина С.П., Маслова М.А. Анализ непробиваемых алгоритмов шифрования // Научный результат. Информационные технологии. – Т.9, №1, 2024. – С. 10-18. DOI: 10.18413/2518-1092-2024-9-1-0-2

Kuzminykh E.S.
Ilina S.P.
Maslova M.A.

ANALYSIS OF IMPENETRABLE ENCRYPTION ALGORITHMS

Sevastopol State University,
33 Universitetskaya St., Sevastopol, 299053, Russia

e-mail: egor2014ru@mail.ru, sofi.ilina@mail.ru, mashechka-81@mail.ru

Abstract
This article explores impenetrable ciphers, including DES, RSA and AES, with the stated goal of determining the most reliable and secure encryption algorithm. The article analyzes the characteristics of each of the ciphers, such as key length, durability and relevance, and also evaluates their popularity and application in modern cryptographic realities. As a result of the study, it was concluded that AES is the most excellent option. This is because DES and RSA have already been hacked and have known vulnerabilities, whereas AES continues to be secure and the best of the ciphers presented. The article is that AES is recommended for use in systems where reliable and impenetrable encryption is required.
Keywords: information security; security; cybersecurity; code; programming; cryptography; encryption; RSA; DES; AES; ECB; CBC; cipher hacking; cipher security; quantum computers; quantum cryptography
For citation: Kuzminykh E.S., Ilina S.P., Maslova M.A. Analysis of impenetrable encryption algorithms // Research result. Information technologies. – T. 9, №1, 2024. – P. 10-18. DOI: 10.18413/2518-1092-2024-9-1-0-2

ВВЕДЕНИЕ

Шифрование информации является основополагающим принципом для обеспечения конфиденциальности и безопасности данных в цифровой эпохе. Это процесс преобразования информации в непонятную форму, непроходимую для несанкционированного доступа. На протяжении многих лет, криптографы разработали разнообразные алгоритмы шифрования, чтобы защитить данные от злоумышленников и обеспечить безопасность коммуникации. Однако, современные технологии и вычислительная мощность постоянно развиваются, и то, что ранее считалось непробиваемым, может стать уязвимым. Существуют алгоритмы шифрования, которые не возможно взломать математически, а только с помощью брутфорса – это и есть непробиваемые алгоритмы шифрования. В свете этого возникает важный вопрос: насколько надежны и безопасны существующие алгоритмы шифрования?

Целью данной статьи является анализ непробиваемых алгоритмов шифрования, с целью оценки их надежности и выявления возможных уязвимостей. Будут рассмотрены различные алгоритмы шифрования, такие как DES, RSA и AES, а также проанализированы их фундаментальные принципы и математическая основа.

ОСНОВНАЯ ЧАСТЬ

Рассмотрим один из наиболее известных алгоритмов — DES (Data Encryption Standard). DES был разработан в 1970-х годах и был стандартом шифрования США в течение долгого времени. Хотя с тех пор он был заменен более сильными алгоритмами, изучение DES позволит лучше понять принципы и методы работы современных шифров.

DES представляет собой блочный алгоритм шифрования, который оперирует блоками данных фиксированного размера (обычно 64 бита). Основная идея DES заключается в использовании модифицированной формы перестановки и замены, известной как шифрование Фейстеля. Рассмотрим шаги DES внимательнее.

1. Генерация ключей:
   – Начинается с исходного ключа длиной 64 бита;
   – применяется операция перестановки и выбора для создания двух 28-битных подключей (субключей) — левого и правого;
   – затем выполняется раундовая функция шифрования, в результате которой создается новый набор подключей. Этот процесс повторяется 16 раз для получения 16 подключей.
2. Шифрование:
   – Исходный блок данных размером 64 бита проходит через начальную перестановку;
   – он разделяется на две половины — левую и правую — каждая по 32 бита;
   – затем последовательно выполняются 16 раундовых операций шифрования, каждая из которых включает в себя сеть Фейстеля;
   – правая половина расширяется до 48 битов и применяется побитовая операция XOR с соответствующим раундовым подключом;
   – результат подается на вход подстановочных блоков (S-блоков), которые заменяют каждый 6-битный блок на 4 бита с использованием заранее определенных таблиц замен;
   – результаты S-блоков объединяются в блок размером 32 бита;
   – этот блок проходит через операцию перестановки и выбора;
   – левая половина обменивается с правой, а правая половина дополняется побитовой операцией XOR с результатом раундовой функции;
   – после завершения всех раундовых операций, левая и правая половины обмениваются местами и проходят через конечную перестановку, в результате которой получается зашифрованный блок данных.
3. Расшифрование. Процесс расшифрования DES аналогичен процессу шифрования, но раундовые подключи используются в обратном порядке. Таким образом, зашифрованный блок

данных пропускается через алгоритм DES, начиная с конечной перестановки и заканчивая начальной перестановкой, чтобы получить исходные данные.

Пример использования DES:

Допустим, у нас есть исходный блок данных размером 64 бита: 01101011 11000101 01010100 11110000 00110011 10100110 11001101 00101100.

Применяя алгоритм DES с выбранным ключом и раундовыми подключами, мы получаем зашифрованный блок данных: 11010110 00110111 01101100 11100111 00101101 01010001 11000101 01011000.

Существуют несколько методов взлома DES, которые были разработаны исследователями и специалистами в области криптографии. Однако стоит отметить, что эти методы взлома были разработаны в контролируемых условиях и предназначены для демонстрации уязвимостей DES, но не для реальных атак.

Одним из наиболее известных методов взлома DES является атака методом полного перебора, которая известна как «атака на основе исчисления времени». В этом методе злоумышленник перебирает все возможные ключи DES (2^56 комбинаций) и для каждого ключа выполняет шифрование входных данных. Затем злоумышленник сравнивает время, необходимое для шифрования, со средним временем шифрования при правильном ключе. Ключ, который приводит к наиболее близкому времени шифрования, считается правильным ключом. Однако, этот метод требует огромных вычислительных ресурсов и времени.

Другой вариант атаки на DES — это дифференциальный криптоанализ. Этот метод основан на изучении различий в выходных данных при изменении входных данных и ключа шифрования. Атакующий сравнивает различия в выходных данных из разных пар входных данных и пытается извлечь информацию о ключе шифрования. Этот метод также требует больших объемов данных и вычислительных ресурсов.

Однако, стоит отметить, что DES, разработанный в 1970-х годах, сейчас считается устаревшим и ненадежным с точки зрения безопасности. В 1999 году DES был официально заменен стандартом AES (Advanced Encryption Standard), который обеспечивает значительно более высокий уровень безопасности [1-4].

Следующий метод является одним из крупнейших алгоритмов шифрования RSA, который применяется на смарт-картах, в защищенных телефонах, на сетевых платах Ethernet, активно используется в криптографическом оборудовании Thales. Данный алгоритм является одним из составов основных протоколов для защищенных коммуникаций Internet, в том числе S/MIME, SSL и S/WAN. Так же он применяется в учреждениях, корпорациях, университетах, правительственных службах, государственных органах и лабораториях [5].

Алгоритм RSA (Rivest-Shamir-Adleman) — это криптографический алгоритм. Он используется для шифрования и подписи данных, основан на математической проблеме факторизации больших чисел. Данный алгоритм обеспечивает высокий уровень безопасности.

1. Генерация ключей:

– Шаг 1: Выбор двух простых чисел p и q, которые должны быть достаточно большими и случайными.

– Шаг 2: Вычисление модуля N, рассчитывается следующим образом: N = p * q.

– Шаг 3: Вычисление функции Эйлера от N (формула 1):

$$\varphi(N) = (p - 1) * (q - 1) \quad (1)$$

– Шаг 4: Выбор открытой экспоненты e. На которое накладывается условие: $e < \varphi(N)$ и взаимно простым с $\varphi(N)$.

– Шаг 5: Вычисление закрытой экспоненты d с помощью расширенного алгоритма Евклида. Закрытая экспонента должна удовлетворять условию $(e * d) \bmod \varphi(N) = 1$.

– Шаг 6: Пара ключей сформирована: открытый ключ (N, e) и закрытый ключ (N, d).

2. Шифрование данных:

– Шаг 1: Преобразование сообщения M в числовое представление m.

– Шаг 2: Шифрование числа m с использованием открытого ключа (N, e). Зашифрованное сообщение вычисляется по формуле 2:

$$c = m^e \cdot mod\ N \quad (2)$$

– Шаг 3: Шифрованный текст c — это зашифрованная версия сообщения M.

3. Дешифрование данных:

– Шаг 1: Расшифрование зашифрованного текста c с использованием закрытого ключа (N, d). Расшифрованное сообщение вычисляется по формуле 3:

$$m = c^d \cdot mod\ N \quad (3)$$

– Шаг 2: Число m преобразуется обратно в исходное сообщение M.

Реализация на языке Python:

```
# Функция для проверки простоты числа
def is_prime(num):
    if num < 2:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True
# Функция для вычисления наибольшего общего делителя
def gcd(a, b):
    while b:
        a, b = b, a % b
    return a
# Функция для вычисления расширенного алгоритма Евклида
def extended_gcd(a, b):
    if b == 0:
        return a, 1, 0
    gcd, x, y = extended_gcd(b, a % b)
    return gcd, y, x - (a // b) * y
# Функция для генерации ключей RSA
def generate_rsa_keys():
    p = 17
    q = 11
    N = p * q
    phi = (p - 1) * (q - 1)
    e = 7

    # Проверка, что e и phi взаимно просты
    if gcd(e, phi) != 1:
        raise ValueError("Ошибка: e и phi не взаимно просты.")
    _, d, _ = extended_gcd(e, phi)
    # Положительное значение d
    d = d % phi
    public_key = (N, e)
    private_key = (N, d)
    return public_key, private_key
# Функция для шифрования сообщения с помощью открытого ключа
def encrypt(message, public_key):
    N, e = public_key
    ciphertext = pow(message, e, N)
    return ciphertext
# Функция для расшифрования сообщения с помощью закрытого ключа
```

```
        def decrypt(ciphertext, private_key):
            N, d = private_key
            message = pow(ciphertext, d, N)
            return message
        # Пример использования
        message = 721782079
        public_key, private_key = generate_rsa_keys()
        encrypted_message = encrypt(message, public_key)
        decrypted_message = decrypt(encrypted_message, private_key)
        print("Открытый ключ:", public_key)
        print("Закрытый ключ:", private_key)
        print("Зашифрованное сообщение:", encrypted_message)
print("Расшифрованное сообщение:", decrypted_message)
```

Данный пример является базовой реализацией алгоритма RSA, который предназначен для понимания работы алгоритма и не обладает высокой производительностью и степенью защиты. Для улучшения алгоритма используются библиотеки такие как, cryptography или pycryptodome, которые обеспечивают больший функционал.

Учёные считали, что для взлома данного алгоритма потребуется квантовый компьютер с вычислительной мощностью в сотни тысяч кубитов, который появится минимум через 10 лет. В начале 2023 года эксперты из Китая опубликовали статью «Factoring integers with sublinear resources on a superconducting quantum processor» [9], где описали методику взлома RSA-48 при помощи квантового компьютера мощностью 372 кубита, что ставит под сомнение безопасность интернета, банков и других сфер, где используется RSA. В настоящее время существует квантовый компьютер мощностью в 433 кубита и компания IBM обещала уже в этом году сделать доступным для клиентов данный сверхмощный компьютер, что насторожило ИБ-специалистов, ведь RSA попросту не устоит. Способ взлома довольно прост, ученые использовали методику Клауса-Питера Шнорра и оптимизировали алгоритм таким образом, что для дешифровки ключа RSA длиной 48 бит хватило 10-кубитного компьютера. Также учёные рассказали, что по их методике для взлома 2048-битного ключа понадобится всего 372 кубита, а не сотни тысяч, как предполагалось ранее [6, 7, 8]. Чтобы предотвратить возможность данной атаки, необходимо воспользоваться тем же средством, квантовым компьютером и использовать более длинные ключи, основанные на квантовой криптографии, или же другие алгоритмы.

Третьим рассматриваемым алгоритмом шифрования будет AES, данный шифр в настоящее время не был взломан. AES является симметричным блочным алгоритмом шифрования, выбранным в качестве стандарта правительством США. AES обладает высокой стойкостью и безопасностью, и широко применяется в различных приложениях и протоколах. Он поддерживает ключи длиной 128, 192 и 256 битов, что обеспечивает большую стойкость. AES остается крепким и безопасным алгоритмом шифрования на сегодняшний день.

Основные этапы алгоритма AES:

1. Инициализация ключа: вначале определяется ключ шифрования. Длина ключа может быть 128 бит, 192 бита или 256 бит, в зависимости от необходимого уровня безопасности.

2. Добавление паддинга: если длина данных не кратна размеру блока (128 бит), то выполняется добавление паддинга до нужной длины.

3. Шифрование раундами: алгоритм AES состоит из нескольких раундов шифрования, которые применяются последовательно к данным.

– К каждому блоку данных применяется операция подстановки байтов (SubBytes), заменяющая каждый байт на соответствующий байт из заранее заданной таблицы подстановки (S-Box).

– Затем происходит сдвиг строк (ShiftRows), при котором байты в каждой строке блока сдвигаются циклически влево на определенное количество позиций.

– Происходит перемешивание столбцов (MixColumns), при котором каждый столбец блока умножается на определенную матрицу, что вносит нелинейность в шифрование.

– В конце каждого раунда применяется операция добавления ключа (AddRoundKey), при которой блок данных побитово складывается с соответствующим раундовым ключом.

4. Финальный раунд: последний раунд отличается от предыдущих тем, что он не выполняет операцию перемешивания столбцов (MixColumns).

5. Результат: после завершения всех раундов, полученные данные являются зашифрованным сообщением.

Пример реализации алгоритма на Python:

```
        from Crypto.Cipher import AES
        from Crypto.Random import get_random_bytes
        # Генерация случайного 128-битного ключа
        key = get_random_bytes(16)
        # Инициализация шифратора AES с выбранным ключом
        cipher = AES.new(key, AES.MODE_ECB)
        # Зашифрование сообщения
        message = b"Hello, AES!"
        ciphertext = cipher.encrypt(message)
        # Дешифрование зашифрованного сообщения
        decipher = AES.new(key, AES.MODE_ECB)
        decrypted_message = decipher.decrypt(ciphertext)
        print("Зашифрованное сообщение:", ciphertext)
print("Дешифрованное сообщение:", decrypted_message)
```

В этом примере используется режим шифрования ECB (Electronic Codebook), который применяет шифратор к каждому блоку данных в отдельности. Однако, для повышения безопасности обычно рекомендуется использовать режимы, такие как CBC (Cipher Block Chaining), который добавляет вектор инициализации для каждого блока данных. Данный пример является базовым примером работы алгоритма, для его улучшения необходимо использовать те же библиотеки, что и для прошлого алгоритма RSA [10 - 13].

Режимы шифрования ECB (Electronic Codebook) и CBC (Cipher Block Chaining) являются двумя различными методами использования алгоритмов блочного шифрования, таких как AES. Они отличаются в том, как они обрабатывают и шифруют блоки данных, и имеют свои особенности и преимущества.

1. Режим шифрования ECB:

В режиме ECB каждый блок данных одинакового размера шифруется независимо друг от друга.

Каждый блок данных передается в шифратор, который независимо применяет операции шифрования к каждому блоку.

Простой пример: представим, что у нас есть изображение, и мы применяем ECB для его шифрования. Каждый блок пикселей (обычно 8x8 или 16x16) будет обрабатываться независимо от других блоков.

2. Режим шифрования CBC:

В режиме CBC каждый блок данных перед шифрованием комбинируется с предыдущим зашифрованным блоком данных.

Используется дополнительный начальный вектор инициализации (IV), который служит для инициализации первого блока шифрования.

Каждый следующий блок данных перед шифрованием проходит операцию XOR с предыдущим зашифрованным блоком данных.

Простой пример: предположим, что у нас есть текстовое сообщение, разделенное на блоки. Первый блок передается в шифратор, а затем каждый следующий блок передается через операцию XOR с предыдущим зашифрованным блоком.

В отличие от ECB, где каждый блок данных обрабатывается независимо, CBC включает предыдущий зашифрованный блок данных в процесс шифрования следующего блока. Это делает

режим CBC более безопасным, поскольку вносит дополнительные изменения в шифрованные данные, усложняя обнаружение и анализ повторяющихся паттернов. Однако он более чувствителен к ошибкам передачи данных, поскольку изменение одного бита повлияет на расшифровку остальных блоков [14, 15].

На сегодняшний день AES (Advanced Encryption Standard) считается крайне надежным алгоритмом шифрования. Он широко применяется и рекомендован для использования правительством США и другими организациями по всему миру.

На данный момент нет известных методов взлома AES с использованием общедоступных вычислительных ресурсов. Он протестирован и подвергался компьютерным атакам на протяжении многих лет, и до сих пор не было документированных случаев успешного взлома. Такие атаки на AES требуют огромные вычислительные мощности и времени, что делает их практически невозможными для большинства злоумышленников. Конечно, в течение времени могут появиться новые методы или уязвимости, которые могут повлиять на безопасность AES. Поэтому постоянное исследование и анализ алгоритмов шифрования, включая AES, является важной задачей для обеспечения безопасности информации.

ЗАКЛЮЧЕНИЕ

Таким образом, в статье были рассмотрены шифры DES, RSA и AES, оценены их стойкость, безопасность и популярность. Шифр DES, несмотря на свою историческую значимость, потерял актуальность из-за ограничений в длине ключа и слабостей алгоритма. RSA, в свою очередь, является асимметричным алгоритмом, широко используемым для шифрования и подписи, но уязвим к будущим атакам с использованием квантовых компьютеров.

С другой стороны, AES продолжает оставаться самым лучшим алгоритмом шифрования в настоящее время. Он обладает высокой стойкостью, безопасностью и применяется во множестве приложений и протоколов, поддерживает различные длины ключей, что позволяет выбирать уровень безопасности в зависимости от потребностей.

На основании обзора этих алгоритмов, можно сделать вывод, что AES является наиболее предпочтительным выбором для надежного шифрования данных. Он обеспечивает современные стандарты безопасности и безопасное применение в широком спектре сценариев.

Важно отметить, что безопасность шифрования также зависит от правильной реализации, корректного использования криптографических протоколов и хранения ключей секретными. Регулярное обновление системы и следование рекомендациям по безопасности являются также важными факторами для обеспечения надежного шифрования данных [16, 17].

Список литературы

1. Стандарт шифрования данных (DES) [Электронный ресурс]. URL: https://intuit.ru/studies/courses/552/408/lecture/9362?page=3
2. Алгоритм шифрования DES [Электронный ресурс]. URL: https://studme.org/239561/informatika/algoritm_shifrovaniya
3. Создание подключей в алгоритме des [Электронный ресурс]. URL: https://studfile.net/preview/2204125/page:4/
4. Стандарт шифрования данных Data Encryption Standard [Электронный ресурс]. URL: https://protect.htmlweb.ru/des.htm
5. Использование криптосистемы RSA в настоящее время [Электронный ресурс]. URL: https://studfile.net/preview/299352/page:6/
6. Китайские программисты взломали алгоритм RSA. Это угрожает всему интернету [Электронный ресурс]. URL: https://4pda.to/2023/01/08/408266/kitajskie_programmisty_vzlomali_algoritm_rsa_eto_ugrozhaet_vsemu_internetu/
7. Эксперты из Китая взломали RSA-шифрование с помощью квантовых компьютеров [Электронный ресурс]. URL: https://www.anti-malware.ru/news/2023-01-06-1447/40255

8. Китайские исследователи заявили об успешном взломе шифрования RSA [Электронный ресурс]. URL: https://cryptonews.net/ru/news/blockchain/18936323/
9. Factoring integers with sublinear resources on a superconducting quantum processor [Электронный ресурс]. URL: https://arxiv.org/pdf/2212.12372.pdf
10. Как работает AES (Advanced Encryption Standard) [Электронный ресурс]. URL: https://vc.ru/dev/656195-kak-rabotaet-aes-advanced-encryption-standard-obyasnenie-dlya-gumanitariev-tipa-menya
11. Объяснение шифрования AES [Электронный ресурс]. URL: https://blog.kraden.com/ru/aes-256-encryption
12. Advanced Encryption Standard (AES) [Электронный ресурс]. URL: https://www.geeksforgeeks.org/advanced-encryption-standard-aes/
13. What is the Advanced Encryption Standard (AES)? [Электронный ресурс]. URL: https://www.zenarmor.com/docs/network-security-tutorials/what-is-advanced-encryption-standard-aes
14. Несколько режимов работы шифрования AES [Электронный ресурс]. URL: https://russianblogs.com/article/2418837814/
15. How to choose an AES encryption mode (CBC ECB CTR OCB CFB)? [Электронный ресурс]. URL: https://stackoverflow.com/questions/1220751/how-to-choose-an-aes-encryption-mode-cbc-ecb-ctr-ocb-cfb
16. Костиков В.А. Необходимость сжатия зашифрованных данных с помощью алгоритмов кодирования LZW и Хаффмана / В.А. Костиков, М.А. Маслова // Теория и практика проектного образования. – 2021. – № 3(19). – С. 62-64.
17. Реализация ESG-принципов в стратегии устойчивого развития экономики России / Н.Г. Вовченко, Н.Г. Кузнецов, Е.Н. Макаренко [и др.]. – Ростов-на-Дону: Ростовский государственный экономический университет "РИНХ", 2022. – 508 с.

References

1. Data Encryption Standard (DES) [Electronic resource]. URL: https://intuit.ru/studies/courses/552/408/lecture/9362?page=3
2. DES encryption algorithm [Electronic resource]. URL: https://studme.org/239561/informatika/algoritm_shifrovaniya
3. Creation of a plug-in in the des algorithm [Electronic resource]. URL: https://studfile.net/preview/2204125/page:4/
4. Data Encryption Standard Data Encryption Standard [Electronic resource]. URL: https://protect.htmlweb.ru/des.htm
5. The use of the RSA cryptosystem at the present time [Electronic resource]. URL: https://studfile.net/preview/299352/page:6/
6. Chinese programmers hacked the RSA algorithm. This threatens the entire Internet [Electronic resource]. URL: https://4pda.to/2023/01/08/408266/kitajskie_programmisty_vzlomali_algoritm_rsa_eto_ugrozhaet_vsemu_internetu/
7. Experts from China cracked RSA encryption using quantum computers [Electronic resource]. URL: https://www.anti-malware.ru/news/2023-01-06-1447/40255
8. Chinese researchers have announced the successful cracking of RSA encryption [Electronic resource]. URL: https://cryptonews.net/ru/news/blockchain/18936323/
9. Factoring integers with sublinear resources on a superconducting quantum processor [Electronic resource]. URL: https://arxiv.org/pdf/2212.12372.pdf
10. How AES (Advanced Encryption Standard) works [Electronic resource]. URL: https://vc.ru/dev/656195-kak-rabotaet-aes-advanced-encryption-standard-obyasnenie-dlya-gumanitariev-tipa-menya
11. Explanation of AES encryption [Electronic resource]. URL: https://blog.kraden.com/ru/aes-256-encryption
12. Advanced Encryption Standard (AES) [Electronic resource]. URL: https://www.geeksforgeeks.org/advanced-encryption-standard-aes/
13. What is the Advanced Encryption Standard (AES)? [Electronic resource]. URL: https://www.zenarmor.com/docs/network-security-tutorials/what-is-advanced-encryption-standard-aes
14. Multiple AES encryption modes [Electronic resource]. URL: https://russianblogs.com/article/2418837814/

15. How to choose an AES encryption mode (CBC ECB CTR OCB CFB)? [Electronic resource]. URL: https://stackoverflow.com/questions/1220751/how-to-choose-an-aes-encryption-mode-cbc-ecb-ctr-ocb-cfb

16. Kostikov V.A. The need to compress encrypted data using LZW and Huffman coding algorithms / V.A. Kostikov, M.A. Maslova // Theory and practice of project education. – 2021. – No. 3(19). – pp. 62-64.

17. Implementation of ESG principles in the strategy for sustainable development of the Russian economy / N.G. Vovchenko, N.G. Kuznetsov, E.N. Makarenko [at al.]. – Rostov-on-Don: Rostov State Economic University "RINH", 2022. – 508 p.

Кузьминых Егор Сергеевич, студент четвертого курса кафедры «Информационная безопасность»
Ильина София Павловна, студент четвертого курса кафедры «Информационная безопасность»
Маслова Мария Александровна, доцент кафедры «Информационная безопасность»

Kuzminykh Egor Sergeevich, 4th year student of the Department of Information Security
Ilina Sofia Pavlovna, 4th year student of the Department of Information Security
Maslova Maria Aleksandrovna, Associate Professor of the Department of Information Security