**IT-Колледж “Сириус”**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ДОКЛАД**

по дисциплине “Введение в специальность”

на тему “Криптография”

Выполнил:  
Студент группы

Группа 1.9.7.3  
Зайцев Алексей Вадимович

Принял:

Старший преподаватель  
Тенигин Альберт Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT-Колледж “Сириус”  
2022

Оглавление

[Введение 3](#__RefHeading___Toc641_4227535575)

[Симметричная криптография 4](#__RefHeading___Toc643_4227535575)

[Асимметричная криптография 6](#__RefHeading___Toc645_4227535575)

[HASH-функции 9](#__RefHeading___Toc846_4227535575)

[Смарт контракты 11](#__RefHeading___Toc1057_4227535575)

[Solidity 13](#__RefHeading___Toc446_778917031)

[Выводы 16](#__RefHeading___Toc1059_4227535575)

[Список литературы 17](#__RefHeading___Toc1061_4227535575)

# Введение

Криптогра́фия (от др.-греч. κρυπτός «скрытый» + γράφω «пишу») — наука о методах обеспечения конфиденциальности (невозможности прочтения информации посторонним), целостности данных (невозможности незаметного изменения информации), аутентификации (проверки подлинности авторства или иных свойств объекта), шифрования.[1]

В нашей жизни очень многое связано с криптографией, хотя мы этого даже не замечаем. Потребность в криптографии появилась довольно давно с появлением информации, которую надо скрыть от посторонних глаз. Криптографию впервые начали использовать в древнем Египте около 4000 лет назад.

Традиционная криптография состоит в основном из криптосистем с закрытым ключом, в них шифрование и дешифровка сообщения производится одним и тем же ключом. Современная криптография также включает в себя криптосистемы с открытым ключом, в которой открытый ключ может передаваться по открытому каналу и используется для шифрования и проверки электронной подписи, а для дешифровки сообщения и создания электронной подписи закрытый ключ. Далее в докладе будут рассмотрены оба этих принципа.

# Симметричная криптография

Суть симметричной криптографии заключается в том, что для шифровки и дешифровки сообщения используется один и тот же ключ. То есть сторонам надо заранее договорится об используемом ключе. В случае использования надежного ключа можно добиться стойкости шифра. Но возникает одна проблема, если стороны не договорились о ключе заранее, то его надо передать другой стороне, а делать по открытому каналу это нельзя иначе весь процесс потеряет смысл. Эту проблему решил алгоритм Диффи-Хэлмана. В алгоритме Диффи-Хеллмана участники разглашают некоторую информацию о ключе, но сохраняют в тайне данные, позволяющие воспроизвести этот ключ

Симметричные шифры:

* Блочные шифры — шифруют информацию блоками определенной длины, применяя к блоку ключ в установленном порядке, при помощи нескольких циклов подстановки и перемешивания (За счет этого достигается лавинный эффект).
* Поточные шифры — шифры в которых шифруется каждый бит или байт исходных данных с применением гаммирования.

Популярные алгоритмы симметричного шифрования:

AES (Advanced Encryption Standard) – алгоритм симметричного блочного шифрования, принят правительство США по результатам конкурса AES. Один из самых распространенных алгоритмов блочного шифрования в мире.

DES (Data Encryption Standard) – алгоритм симметричного шифрования разработанный IBM. Является предшественником AES.

Традиционные алгоритмы симметричного шифрования:

* Простая перестановка

Текст записывается в таблицу по столбцам, после считывается по строкам. Отправителю и получателю нужно договорится о размере таблицы

* Одиночная перестановка

отличается от простой перестановки тем, что колонки таблицы переставляются по ключевому слову длиной в строку таблицы.

* Двойная перестановка

Для большей стойкости шифра сообщение шифруется повторно. Размер(длина строк и столбцов) второй таблицы должен отличаться от первой.

* Магический квадрат

Магический квадрат — это квадратная таблица со вписанными натуральными числами, сумма которых в каждой строке, столбце или диагонали должна равняться одному и тому же числу. Шифруемый текст вписывался по приведенной нумерации, потом содержимое таблицы выписывается по строкам, то получается шифровка с перестановкой букв.

# Асимметричная криптография

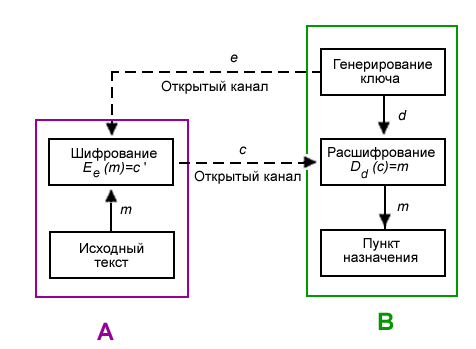
Асимметричная криптография — это система электронной подписи или шифрования, в которой публичный (открытый) ключ можно передавать по открытому каналу. Он нужен для шифровки сообщения или проверки подписи. Для создания электронной подписи и расшифровки сообщения используется закрытый ключ.

На рисунке 1 показана схема передачи информации от Алисы (А) к Бобу (В)

1. Боб выбирает пару ключей (е,d) и шлёт ключ шифрования e (открытый ключ) Алисе по открытому каналу, а ключ дешифровки d (закрытый ключ) Боб держит в секрете
2. Для того, чтобы отправить Бобу сообщение m, Алиса применяет функцию шифрования, определённую открытым ключом e:

Ee (m)=c; c — полученный шифротекст.

1. Боб дешифрует сообщение с, при помощи обратного преобразования Dd, однозначно определённое значением d.

Рисунок 1 [1]

Асимметричная криптография (криптография с открытым ключом) использует односторонние функции - f(x), где при известном х должно быть просто найти f(x), но зная f(x) должно быть невозможно найти х. [2]

Основные алгоритмы:

* RSA (аббревиатура от имени создателей: Rivest, Shamir, Adleman) – криптографический алгоритм, основанный на разложении натуральных чисел на простые множители (факторизация).
* ECC (от английского Elliptic Curve Cryptography) – криптография на эллиптических кривых) – направление криптографии с открытым ключом, которая основана на алгебраической структуре эллиптических кривых над конечными полями. Практическое отличие от других направлений криптографии с открытым ключом — может обеспечить тот же уровень безопасности при меньшей длине закрытого ключа. На ЕСС базируется 2 алгоритма - ****ECDH и ECDSA.****
* ECDH (Протокол Ди́ффи-Хе́ллмана на эллиптических кривых) — протокол, который позволяет двум сторонам, имеющим пары ключей на эллиптических кривых, получить общий секретный ключ, используя открытый канал связи. Этот секретный ключ может быть использован как для шифрования дальнейшего обмена, так и для формирования нового ключа, который затем может использоваться для последующего обмена информацией с помощью алгоритмов симметричного шифрования. Это вариация протокола Диффи-Хеллмана с использованием эллиптической криптографии.[3]
* ECDSA (Elliptic Curve Digital Signature Algorithm) — алгоритм с открытым ключом, который используется для создания и проверки электронной подписи при помощи криптографии на эллиптических кривых.[4]

# ****HASH-функции****

Хеш-функция(от английского hash – превращать в фарш) — это функция, которая преобразует входной массив данных любой длинны в выходную строку установленной длины, выполняемое детерминированным алгоритмом.

Выходное сообщение называют — хеш-суммой или хешем.

Входные данные называют — входным массивом.

Детерминированный алгоритм — алгоритм, который выдает предопределенный и уникальный результат для заданных входных данных.

Требования к хеш-функции:

* Лавинный эффект(даже при малейшем изменении входных данных функция должна сильно изменится)
* Однонаправленность(невозможность провернуть функцию обратно и вернуть входные данные)
* Устойчивость к коллизиям(Вероятность нахождения 2 разных входных данных с одинаковой хеш-суммой, у хороших хеш-функций она должна стремится к теоретическому минимуму)

популярные хеш-алгоритмы:

* MD5 – Разработан Роном Райвестом. Ранее был довольно популярен, но сейчас от него постепенно отказываются из-за случаев коллизий.
* Алгоритмы семейства SHA – семейство хеш-функций, представленных Национальным институтом стандартов и технологий для федерального стандарта обработки информации в США.
* CRC 16/32 (циклический избыточный код) — алгоритм находящий контрольную сумму для проверки целостности данных.

Применение хеш-функций:

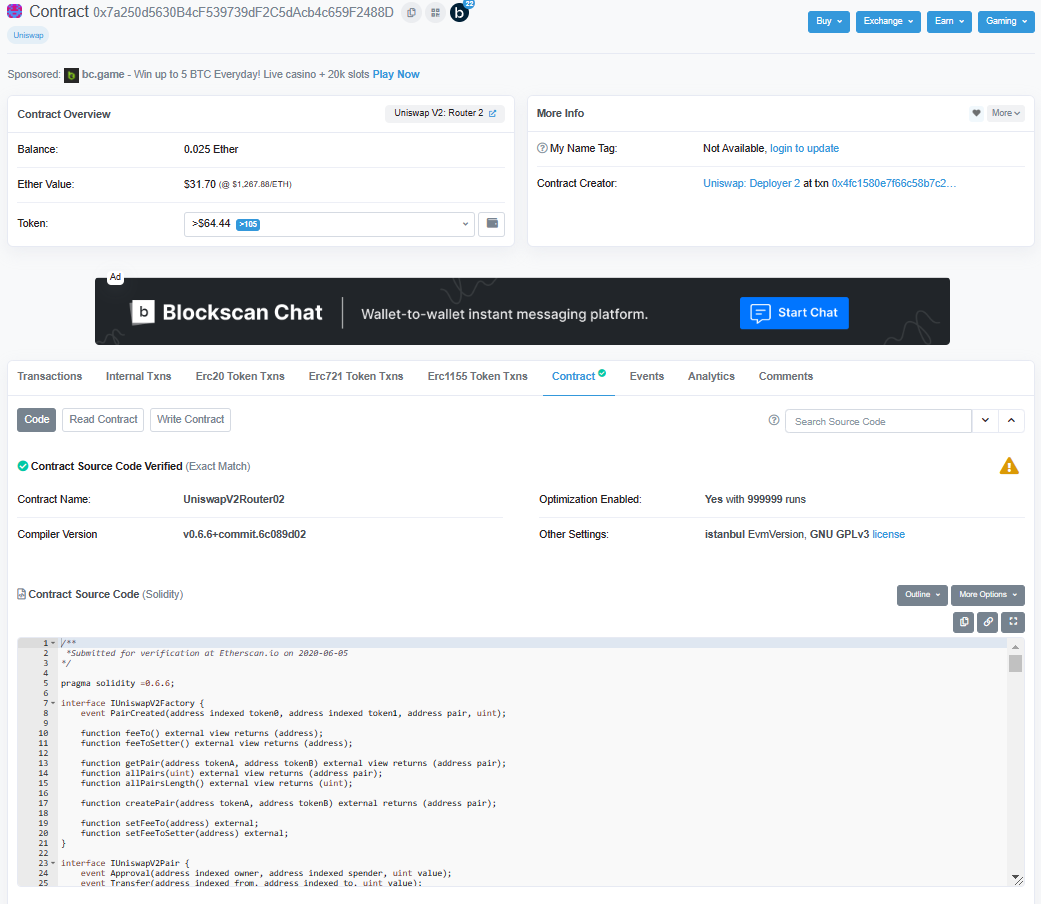
* проверка целостности файлов
* **хранение паролей на серверах(чтобы в случае взлома сервера к злоумышленникам попал только хеш пароля, а не он в первоначальной форме)**

# ****Смарт контракты****

Смарт контракт — это алгоритм, который формирует, управляет и предоставляет информацию о владении чем либо. Он оцифровывает договор, превращая его в код, который находится на децентрализованной платформе и автоматически выполнятся если условия соблюдены.

Первое упоминание о смарт контрактах появилось в 1994 году. Этот термин ввел Ник Сабо. Но в те времена эту задумку сложно было реализовать. Практическая реализация смарт контрактов стала возможна в 2008 году с появлением блокчейна. Именно появление технологии блокчейна сделало смарт контракты полностью автономными и децентрализованными. Но использование в сети биткойна смарт контрактов было сильно ограничено, они не были тьюринг полными из соображений безопасности.

**Смарт контракты стали более популярными лишь с появлением ETHEREUM. Они используются для создания DeFi (Децентрализованные финансы) и Dapps (Децентрализованные приложения).На рисунке 2 представлен пример смарт контракта децентрализованного приложения Uniswap V2 на блокчейне эфириума, открытый в обозревателе блокчейна Etherscan. На этом скриншоте можно увидеть адрес контракта, его баланс, версию компилятора. А во вкладке «code» можно увидеть сам код контракта.**

Рисунок 2 [5]

# ****Solidity****

**Solidity – это объектно ориентированный язык программирования, используемый для написания смарт контрактов ethereum, исполняющихся в виртуальной машине ethereum (EVM).**

**Solidity статически типизированный, JavaScript подобный язык. Он считается тьюринг полным, что позволяет полностью раскрыть возможности технологии смарт контрактов. Программы транслируются в байткод EVM​. Позволяет разработчикам создавать приложения содержащие бизнес-логику, результирующие в неотменяемые транзакции в блокчейне.**

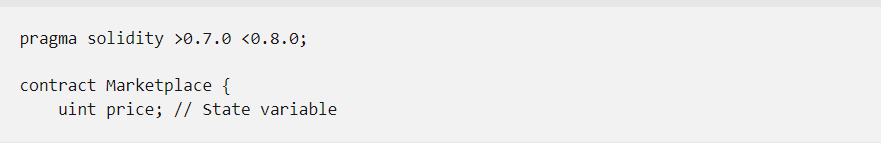
**Pragma – это ключевое слово, которое помогает компилятору проверить соответствие версии solidity требуемой.**

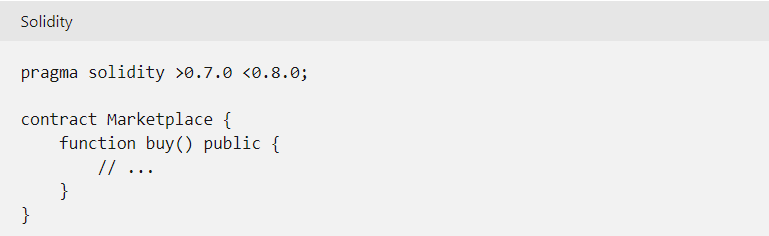
**Для кода на рисунке 3 будет использован компилятор версии от 0.7.0 до 0.7.9. В примере используется целочисленная переменная price типа unit. Переменные типа unit могут хранить 256 битное число без знака. Также можно задавать видимость переменной**

* **public – элемент интерфейса контрактов, который доступен из других контрактов**
* **internal – элемент, который доступен только из текущего контракта.**
* **Private – элемент, который доступен только в том контракте, где он определен**

**В контракте исполняемые единицы кода называются функциями. При помощи функций описываются действия, необходимые для решения общей задачи. Функции можно вызывать из других файлов(библиотек). Поведения функций в solidity примерно такое же, как и в других языках программирования. Пример определения функции в solidity представлен на рисунке 4. В этом примере можно увидеть функцию buy c уровнем видимости Public. Функции в solidity могут принимать параметры и возвращать переменные, с помощью которых осуществляется передача параметров и знаний между ними. На 5 рисунке приведена функция, принимающая целочисленный параметр price и возвращающая число.**

**Также в языке solidity есть события, которые описывают действия выполняемые в контракте. При вызове события необходимо указать параметры. Для вызова события необходимо использовать ключевое слово emit. Пример вызова события можно увидеть на рисунке 6.**

Рисунок 3 [6]

Рисунок 4 [6]

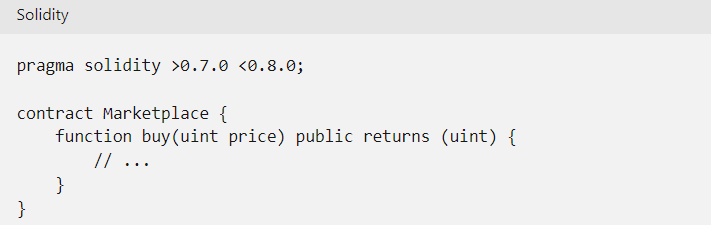
Рисунок 5 [6]

Рисунок 6 [6]

# ****Выводы****

**Криптография — это значима часть электронных систем она используется повсюду: от сетевых протоколов до криптовалют. Она обеспечивает безопасность, точность и конфиденциальность. Подтверждает юридическую силу документов, подписанных ЭЦП, удостоверяет вашу личность и т. д.**

**Возможно в будущем с появлением квантовых компьютеров достаточной мощности (более 1000 кубит) или более эффективного алгоритма факторизации натуральных чисел нынешняя криптография ставит неактуальна. А ее место займет квантовая криптография, которая основана на поляризации квантов. Но пока криптография с открытым ключом обеспечивает надежную защиту наших данных.**

**Также криптография позволяет избавиться от банков и сделать финансовую систему более независимой и децентрализованной, перевести условия договора в код, а с недавнего времени ее применяют и на выборах(электронное голосование). Поэтому так важно разбираться в принципах криптографии.**

# ****Список литературы****

1. **wikipedia : сайт. – URL: https://ru.wikipedia.org/wiki/Криптография (дата обращения: 19.12.2022)**
2. **wikipedia : сайт. – URL: https://ru.wikipedia.org/wiki/Криптосистема\_с\_открытым\_ключом (дата обращения: 19.12.2022)**
3. **wikipedia : сайт. – URL:** [**https://ru.wikipedia.org/wiki**/Протокол\_Диффи](https://ru.wikipedia.org/wiki/Протокол_Диффи)**\_—\_Хеллмана\_на\_эллиптических\_кривых (дата обращения: 19.12.2022)**
4. **wikipedia : сайт. – URL: https://ru.wikipedia.org/wiki/ECDSA (дата обращения: 19.12.2022)**
5. **Etherscan: сайт. – URL: https://etherscan.io/address/0x7a250d5630B4cF539739dF2C5dAcb4c659F2488D (дата обращения: 19.12.2022)**
6. **Microsoft learn : сайт. – URL: https://learn.microsoft.com/ru-ru/training/modules/blockchain-learning-solidity/3-language-basics (дата обращения: 19.12.2022)**