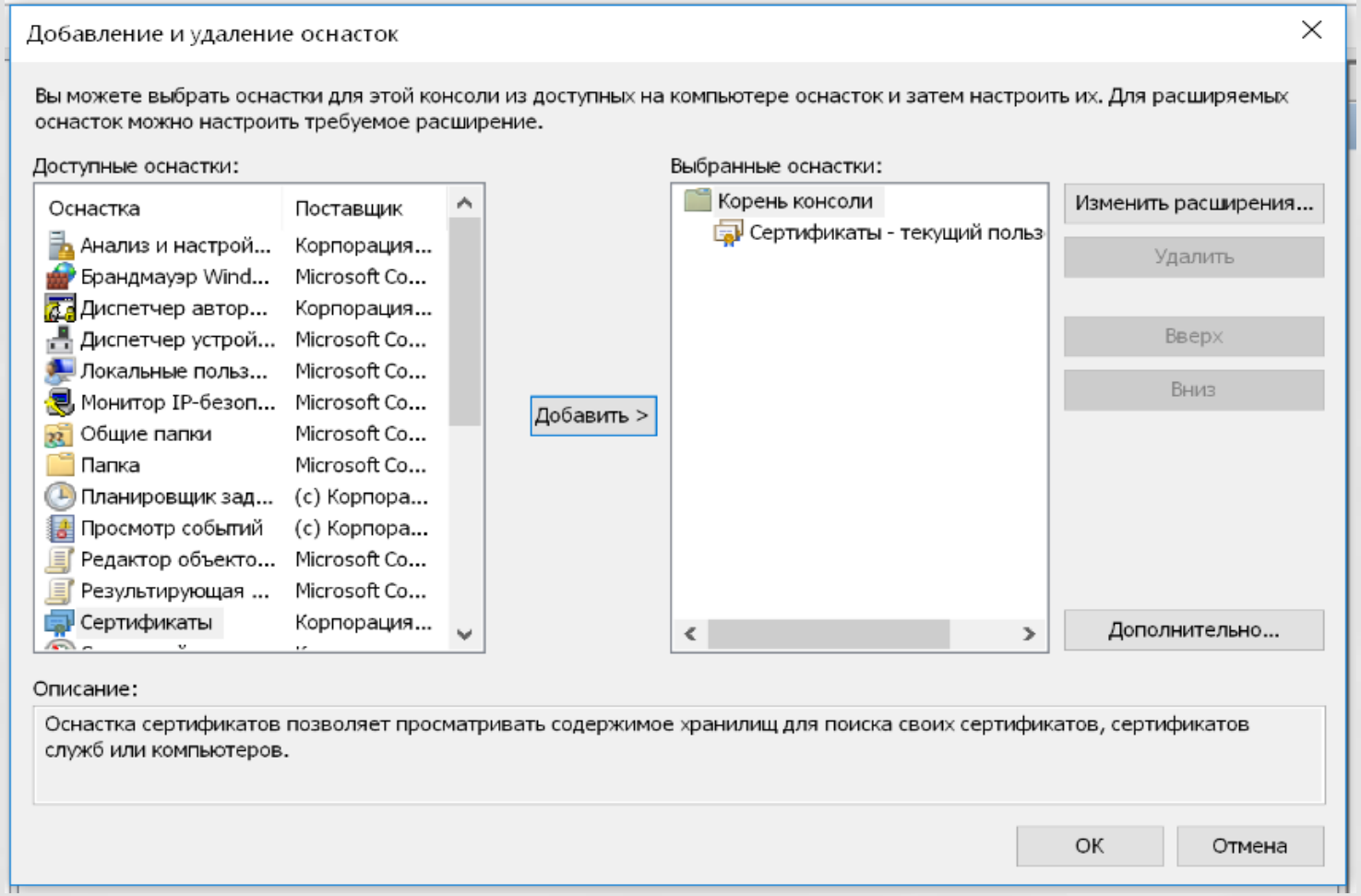
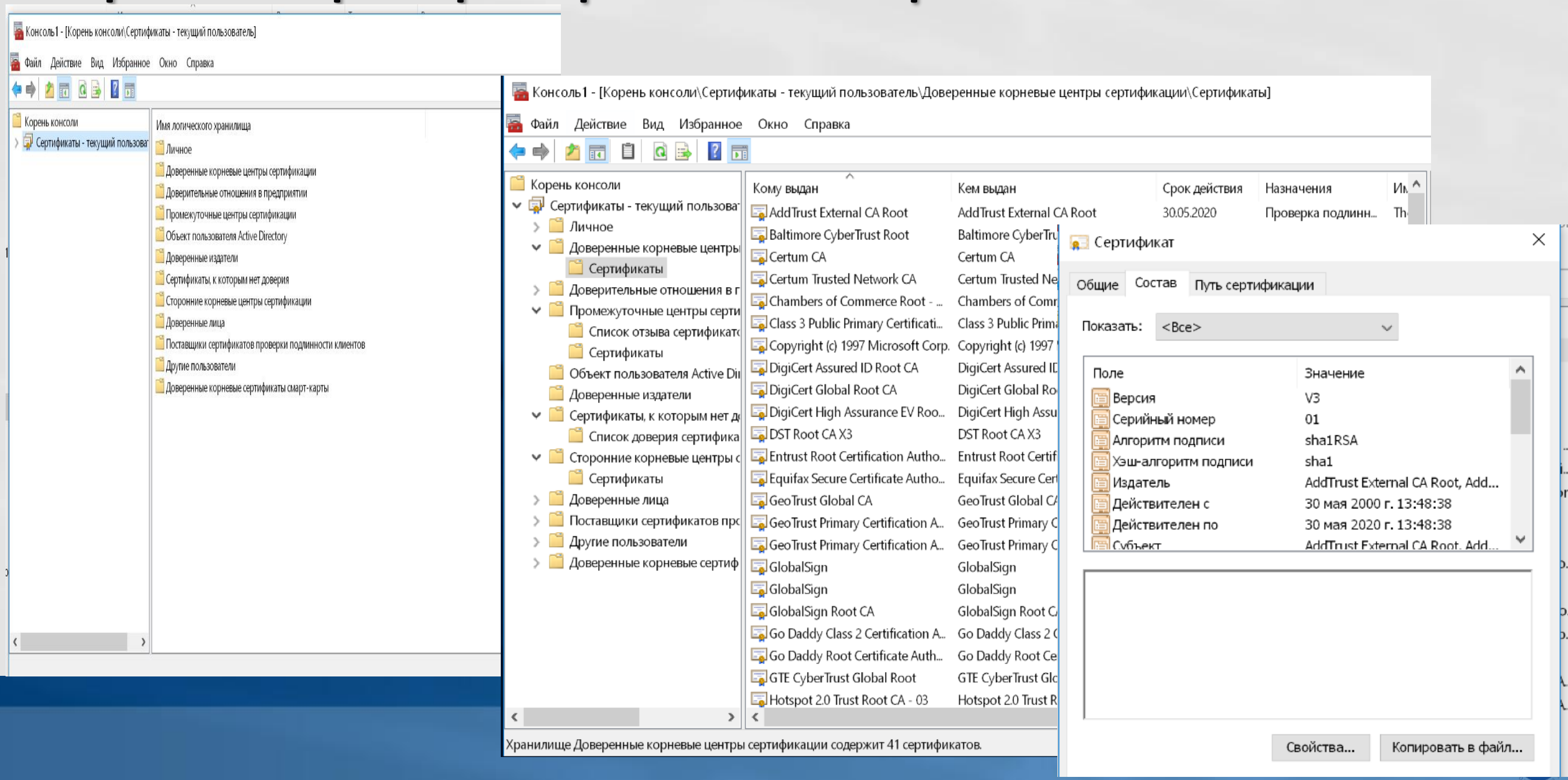
Примеры использования криптографических методов защиты информации

Поиск сертификатов через Microsoft Management Console

- Вызов WIN+R, затем вводим MMC без параметров
- Добавляем оснастку «Сертификаты»



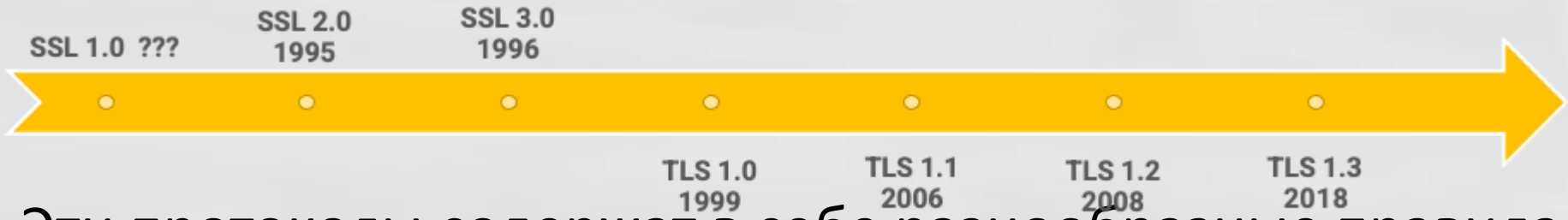
Просмотр сертификатов через MMC



Криптографические методы в протоколе TLS/SSL

Общие сведения

- SSL и TLS - это протокол для защищенной передачи данных между двумя прикладными процессами клиента (**браузер**) и сервера (**web-сервер**) . TLS является прямым наследником SSL.



- Эти протоколы содержат в себе разнообразные правила коммуникации и опираются на различные крипографические преобразования - шифронаборы (Cipher Suites)
- От версии к версии протоколы трансформировались путем добавления новых шифронаборов, удаления устаревших, исправления уязвимостей и минорных изменений правил коммуникации.

Цели протокола

- Обеспечение конфиденциальности данных: шифрование используется для всех сообщений после простого диалога, который служит для определения секретного ключа.
- Обеспечение аутентификации сторон: серверная сторона диалога всегда аутентифицируется, в то время как клиентская - аутентифицируется опционально.
- Обеспечение целостности данных: транспортировка сообщений включает в себя проверку целостности с привлечением кодов аутентификации сообщений (MAC)

Шифронаборы используемой криптографии

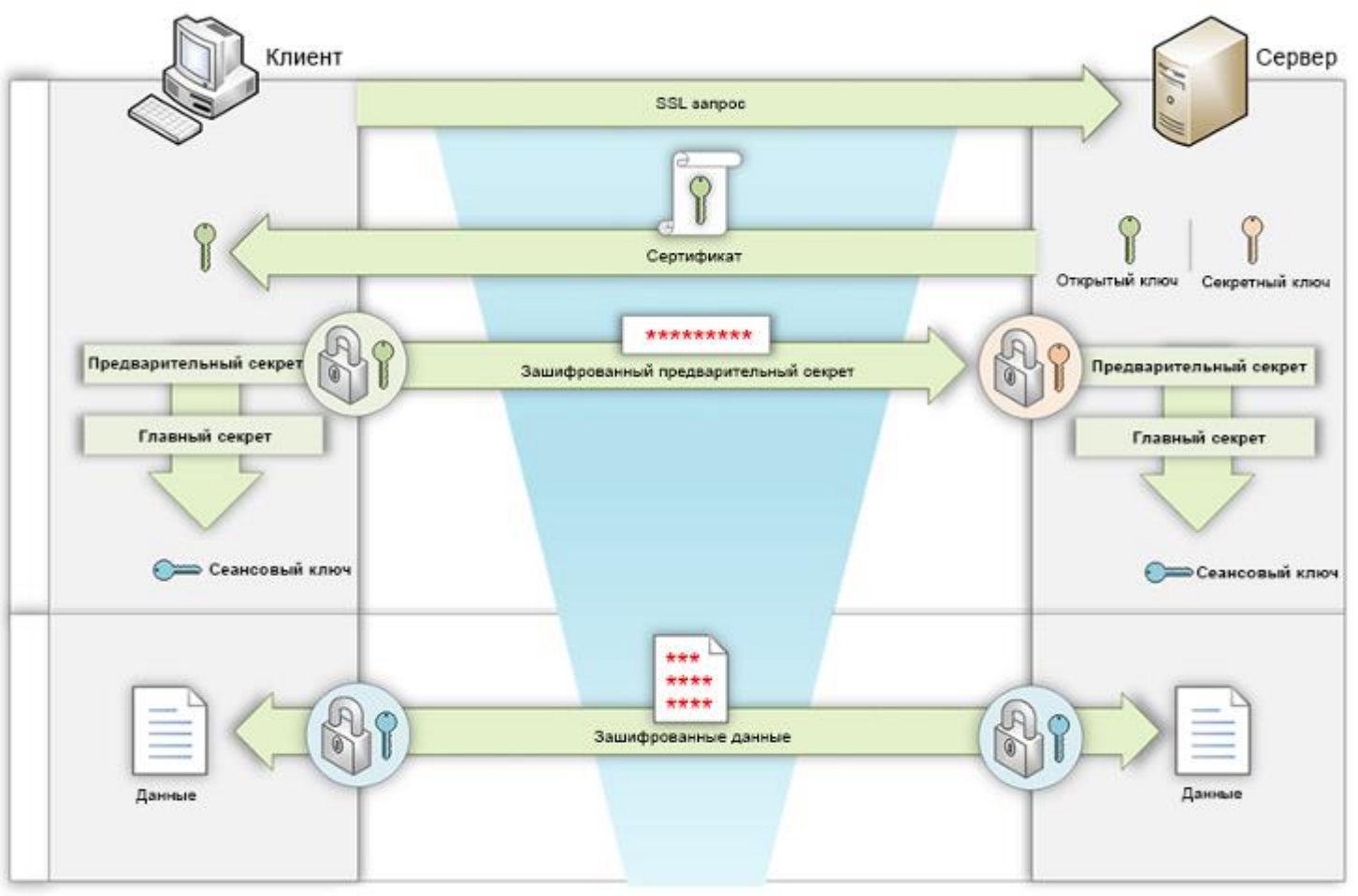
“TLS_DH_RSA_WITH_AES_256_CBC_SHA256 “

- Protocol **(TLS)** – Протокол, по которому осуществляется соединение, а именно TLS или SSL.
- Key Exchange **(DH)** – протокол обмена ключами. Это может быть RSA / DH / DHE (эфемерный DH) / ECDH / ECDHE (эфемерный ECDH) и другие.
- Authentication **(RSA)** – определяет, по какому алгоритму будет проходить аутентификация сторон (RSA / DSA / ECDSA / ...)
- Stream encryption **(AES_256_CBC)** - определяет, по каким протоколам будет осуществляться шифрование потока. При этом указывается длина ключа и режим (RC4_128 / AES_256_CBC / 3DES_EDE_CBC / ...)
- Message Authentication **(SHA256)-** определяет, каким алгоритмом будет осуществляться контроль целостности сообщений (MD5 / SHA / SHA256 / ...)

Фаза протокола «Рукопожатие» (Handshake)

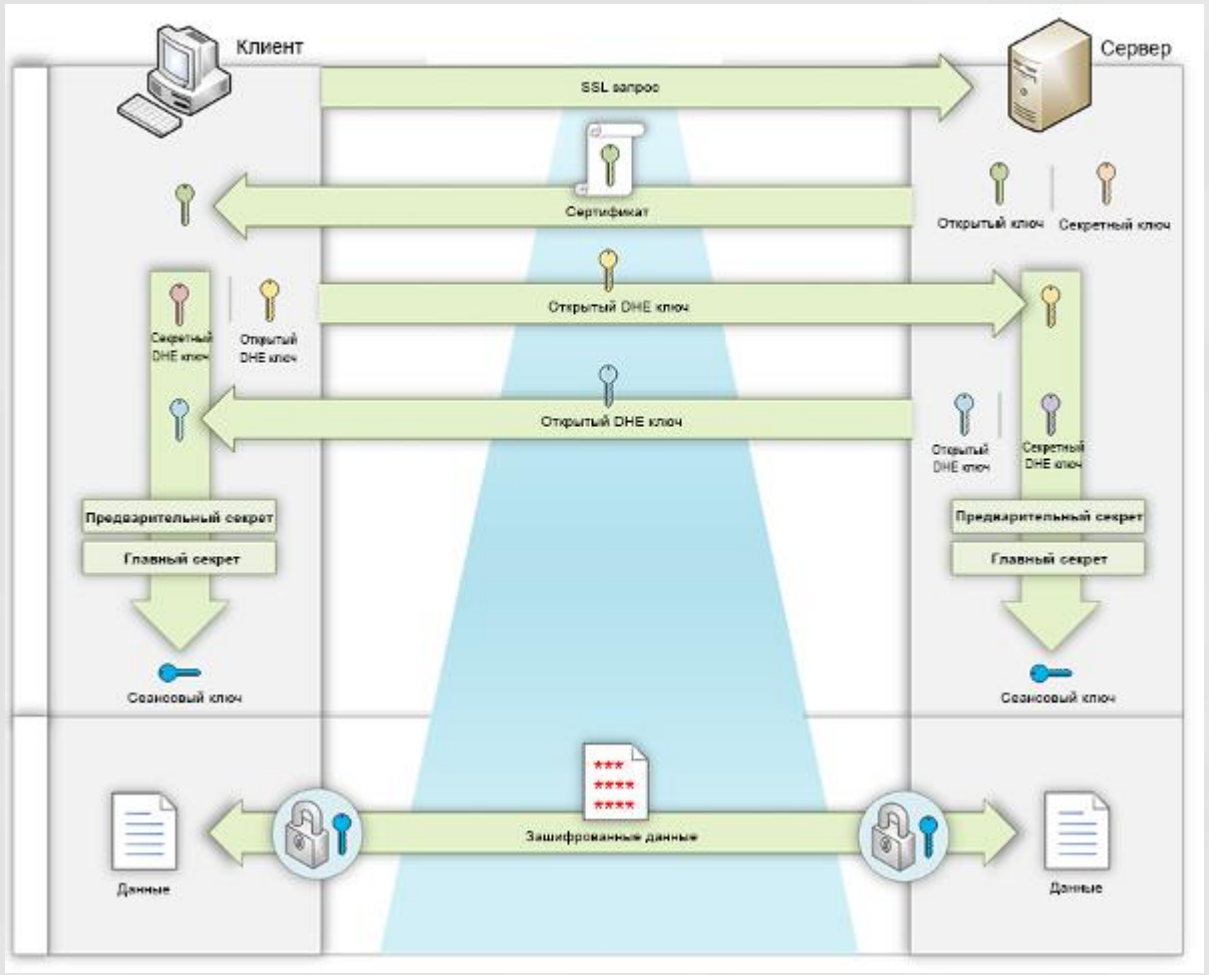
- Этап инициализации соединения:
  - ClientHello – это первое сообщение с клиентской версией протокола, списком шифров и номером сессии
  - ServerHello – сертификат сервера, список базовых шифров
  - ClientKeyExchange - материал для генерации сессионных ключей
- Этап аутентификации клиента:
  - ClientFinished– зашифрованный номер сессии
  - ServerFinished – зашифрованный номер сессии

Установка защищенного соединения TLS/SSL handshake (RSA)



- При таком подходе предварительный секрет шифруется на публичном ключе сервера и передается по сети
- Злоумышленник, он может годами записывать такой трафик в надежде на компрометацию приватного ключа сервера и, если это произойдёт и злоумышленник получит приватный ключ, то он сможет расшифровать весь трафик

Установка защищенного соединения TLS handshake (DHE/ECDHE)



- Предварительный и Главный секреты генерируются независимо друг от друга на клиенте и сервере и не передаются по сети ни в каком виде.
- В случае, если все ключи будут скомпрометированы, единственное, что сможет сделать нарушитель - это расшифровать последний сеанс связи, поскольку ключи, необходимые для генерации секретов, живут лишь в рамках одной сессии.

Фаза протокола «Запись » (Record)

Виды SSL сертификатов

- **Domain Validation (DV)** - сертификаты с проверкой только доменного имени. Подходят для некоммерческих сайтов, так как они подтверждают тот факт, что посетитель находится на сайте в указанном домене, а не перенаправлен на какой-то иной сайт. Этот вид сертификата самый дешевый и популярный.
- **Organization Validation (OV)** - сертификаты с проверкой организации. Более надежны, так как подтверждают еще регистрационные данные компании-владельца. Эту информацию юридическое лицо обязано предоставить при покупке сертификата, а удостоверяющий центр может проверить эту информацию. При этом данные о владельце в сертификате не отображаются.
- **Extended Validation (EV)** - сертификаты с расширенной проверкой. обычно имеют очень крупные организации, так как процедура его получения довольно сложная, дорогая и трудоёмкая, поскольку связана детальным аудитом компании. При посещении веб-сайта, защищённого сертификатом этого типа , в адресной строке отображается название организации, на которую зарегистрирован домен.

ПРИМЕЧАНИЕ: По ссылке https://1cloud.ru/blog/otobrazhenie-https-v-raznyh-brauzerah можно узнать, как разные браузеры информируют пользователей о наличии сертификата или возникающих ошибках

Рекомендации по стандартизации: Использование криптографических алгоритмов в протоколе TLS 1.3

ТЕХНИЧЕСКИЙ КОМИТЕТ ПО СТАНДАРТИЗАЦИИ «КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ»

Новости Документы Проекты документов 1 Активности О нас Форум EN

Поправка к Р 50.1.113–2016 «Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования»

Р 1323565.1.033–2020 «Информационная технология. Криптографическая защита информации. Использование российских алгоритмов электронной подписи в протоколах и форматах сообщений на основе XML»

Утвержден Приказом № 1112-ст от 17.11.2020 г. Федерального агентства по техническому регулированию и метрологии с датой введения в действие 1 апреля 2021 г.

Р 1323565.1.032-2020 «Информационная технология. Криптографическая защита информации. Использование российских криптографических механизмов для реализации обмена данными по протоколу DLMS»

Утвержден Приказом №941-ст от 27.10.2020 Федерального агентства по техническому регулированию и метрологии с датой введения в действие 1 апреля 2021 года.

Р 1323565.1.030-2020 «Информационная технология. Криптографическая защита информации. Использование криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.3)»

Утвержден Приказом №84-ст от 27.02.2020 Федерального агентства по техническому регулированию и метрологии с датой введения в действие 1 июня 2020 года.

Р 1323565.1.029–2019 «Информационная технология. Криптографическая защита информации. Протокол защищенного обмена для индустриальных систем»

Утвержден Приказом № 1504-ст от 30.12.2019 г. Федерального агентства по техническому регулированию и метрологии с датой введения в действие 1 сентября 2020 г.

Р 1323565.1.028–2019 «Информационная технология. Криптографическая защита информации.

https://tc26.ru/standarts/rekomendatsii-po-standartizatsii/

ФЕДЕРАЛЬНОЕ АГЕНТСТВО

ПО ТЕХНИЧЕСКОМУ РЕГУЛИРОВАНИЮ И МЕТРОЛОГИИ

РЕКОМЕНДАЦИИ ПО СТАНДАРТИЗАЦИИ

Р 1323565.1.030— 2020

Информационная технология

КРИПТОГРАФИЧЕСКАЯ ЗАЩИТА ИНФОРМАЦИИ

Использование российских криптографических алгоритмов в протоколе безопасности транспортного уровня (TLS 1.3)

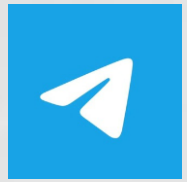
Издание официальное

Настоящие рекомендации содержат описание протокола безопасности транспортного уровня версии 1.3 (TLS 1.3) с криптонаборами на основе алгоритмов блочного шифрования «Магма» и «Кузнечик», описанных в ГОСТ Р 34.12.

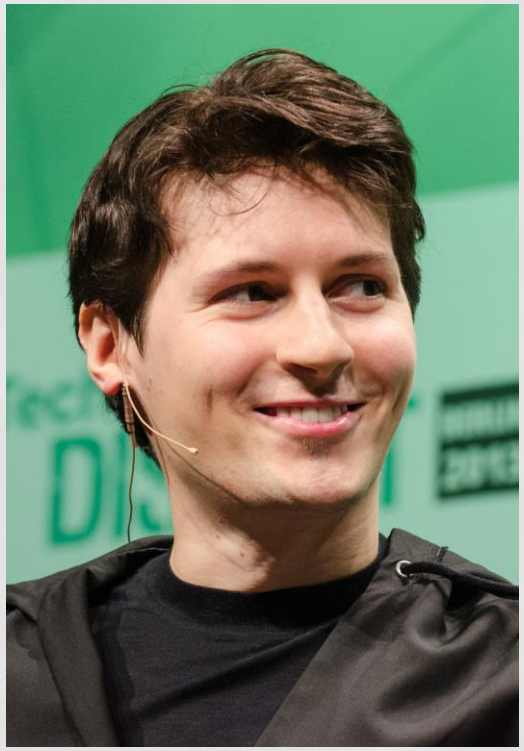
Необходимость разработки настоящего документа вызвана потребностью в обеспечении совместимости различных реализаций протокола TLS 1.3 с использованием российских государственных криптографических стандартов.

Москва
Стандартинформ
2020

Криптографические методы в технологии протоколе MTProto (мессенджер Telegram)



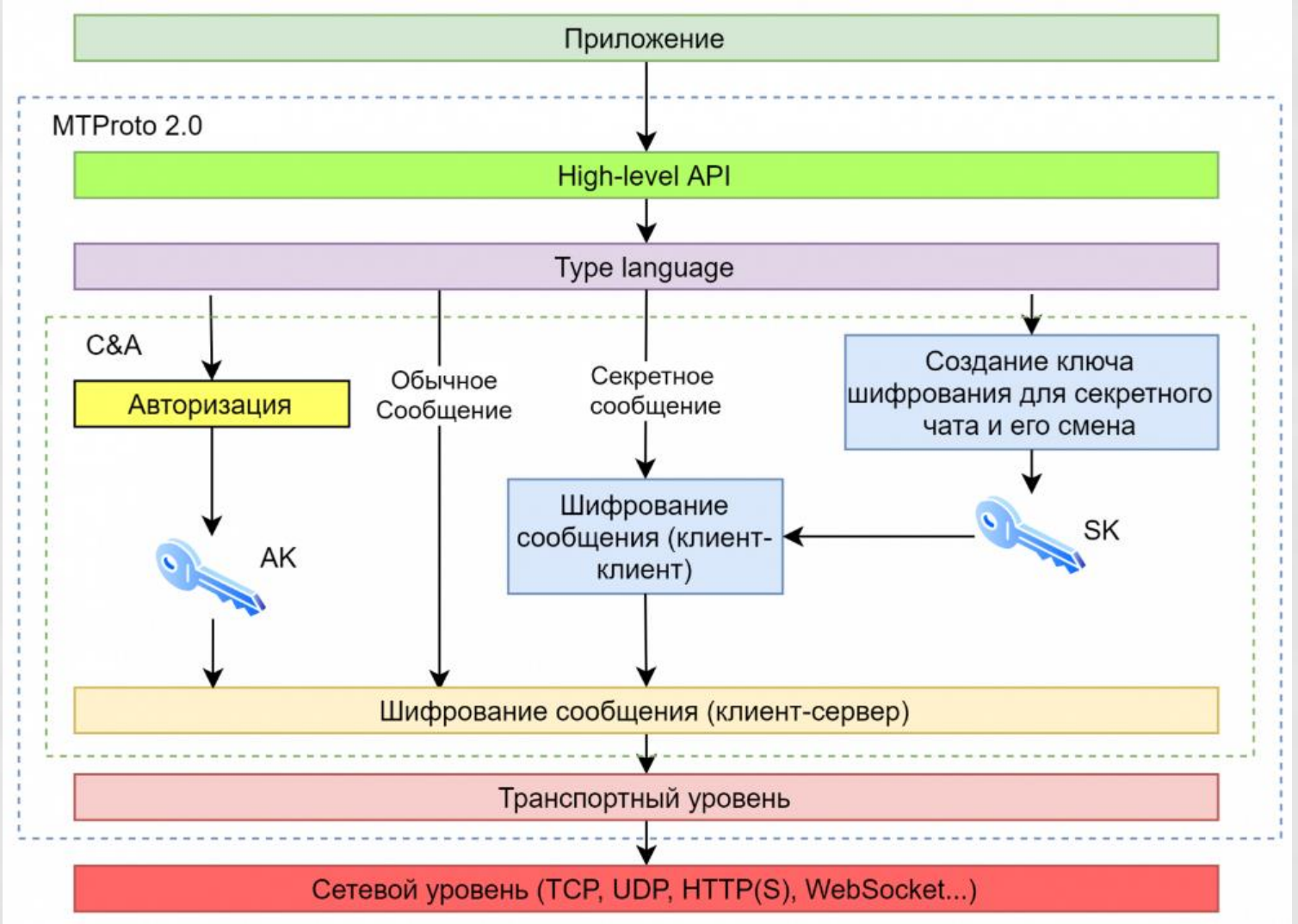
Telegram, CEO



- Па́вел Вале́рьевич Ду́ров (уроженец Ленинграда, 1984) — российский предприниматель и программист,
- Один из создателей социальной сети «ВКонтакте» и одноимённой компании, кроссплатформенного мессенджера Telegram и других проектов.
- В 2001 году с отличием окончил Академическую гимназию, в 2006 — филологический факультет Санкт-Петербургского государственного университета по специальности «Английская филология и перевод» с красным дипломом
- В студенческие годы — лауреат стипендий Президента РФ и Правительства РФ, трёхкратный лауреат Потанинской стипендии.

Общая характеристика протокола

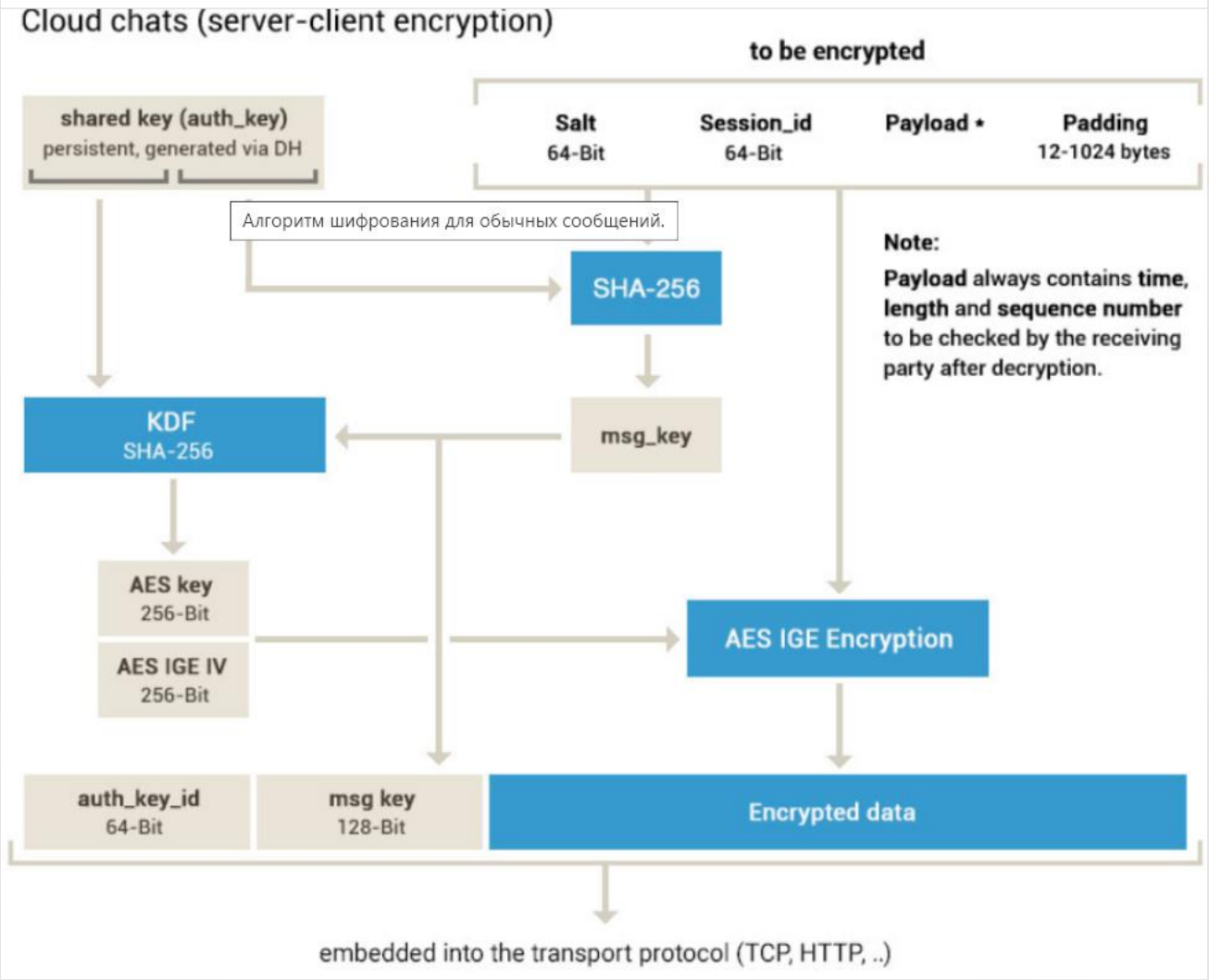
- MTProto – протокол, основанный на и использовании всевозможных известных криптографических алгоритмов с добавлением множества оригинальных «фишек» повышающих секретность передачи данных и снижающих вероятность взлома протокола
- В общем случае MTProto – это клиент-серверный набор протоколов, служащий для доступа к серверу из клиентского приложения через незащищенное соединение.
- Вся криптография сосредоточена на уровне ***Cryptographic and authorization components (C&A)***



Основные термины

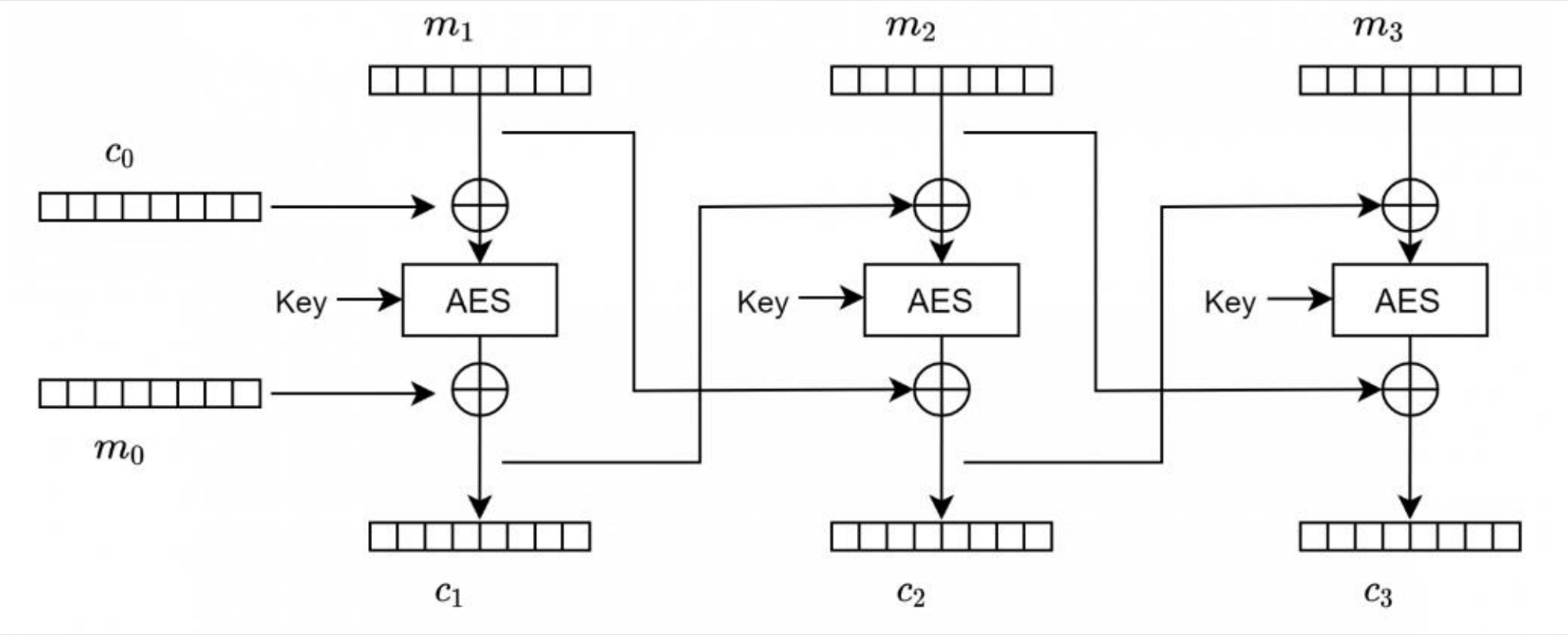
- Authorization key (AK, auth_key) – 2048-битный ключ, который создается на этапе пользовательской регистрации посредством алгоритма DH, а доступ к нему имеет только клиент и сервер
- Key identifier (auth_key _id) – 64 младших бита хеша (SHA-1) auth_key, которые используются для идентификации конкретного ключа, используемого для шифрования сообщения
- Key derivative function (KDF) – функция, формирующая один или несколько секретных ключей на основе секретного значения с помощью псевдослучайной функции (используется SHA-256).
- Session_id – случайное 64-битное число, генерируемое клиентом с целью различить отдельные сеансы одного пользователя (на телефоне, на ПК несколько окон)
- Message key (msg_key) – средние 128 бит хеша (SHA-256) сообщения, которое надо зашифровать (учитывает при расчёте internal header и padding).
- Server salt – случайное 64-битное число, меняющееся каждые 30 минут (отдельно для каждой сессии) по запросу сервера. Сообщения должны приниматься лишь с новой солью , но старые валидны в течение 1800 секунд. Требуется для защиты от атак повторного воспроизведения.
- Padding (12-1024 бит) – добавление ничего не значащих данных к информации, нацеленное на повышение криптостойкости

Клиент-серверное шифрование



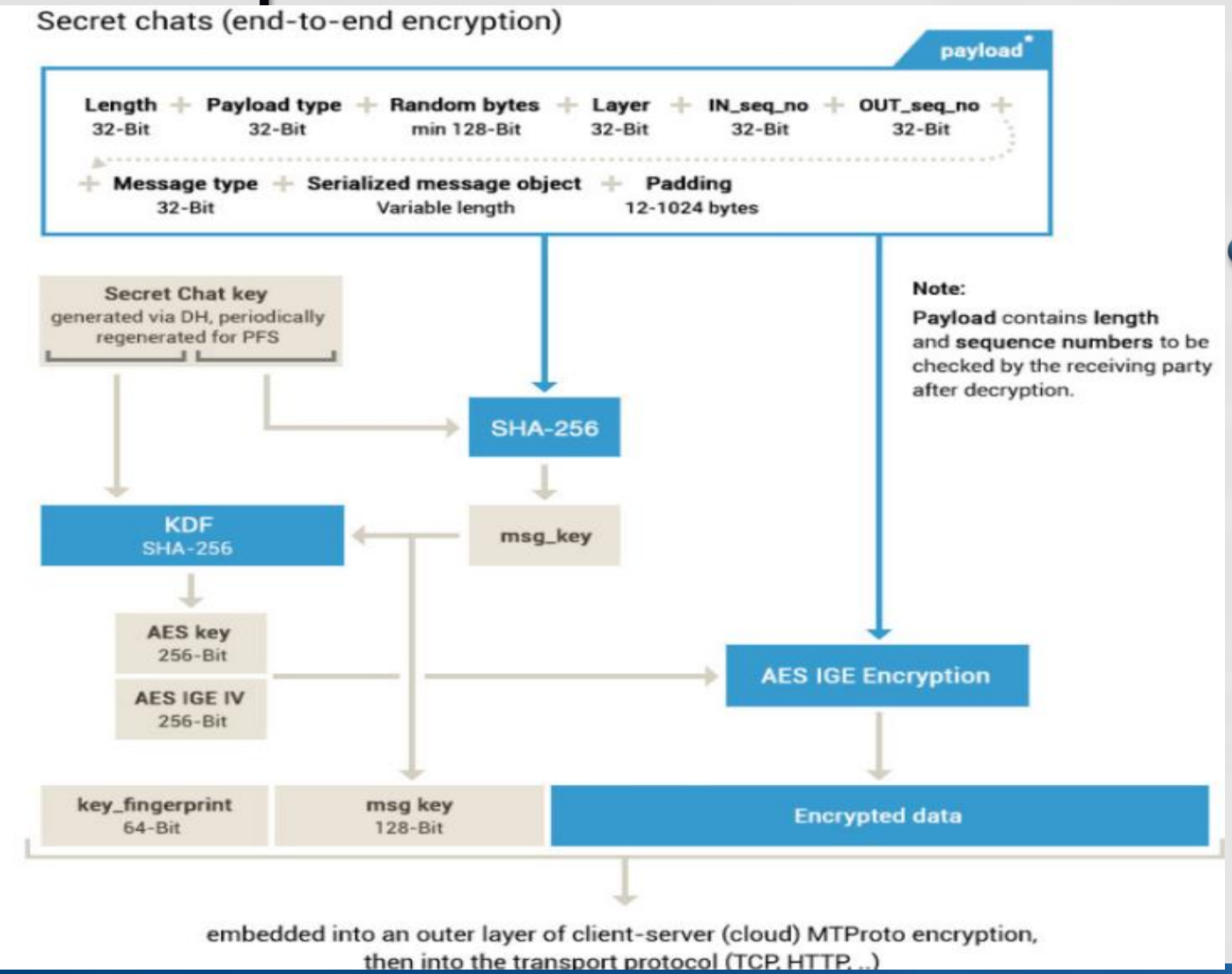
- Собирается пакет для шифрования, состоящий из server salt, session_id, самого сообщения (в него включены время, длина и порядковый номер, которые проверяются на стороне получателя) и padding
- Определяется msg_key, 128 средних бита хэша (SHA-256) от сообщения с добавлением 32-байтового фрагмента auth_key
- Auth_key в комбинации с новонайденным msg_key определяет при помощи KDF 256-битный aes_key и 256-битный вектор инициализации aes_iv
- Далее сформированные значения aes_key и aes_iv используются в алгоритме AES в режиме IGE для шифрования сообщения

Шифрование AES в режиме CBC-IGE



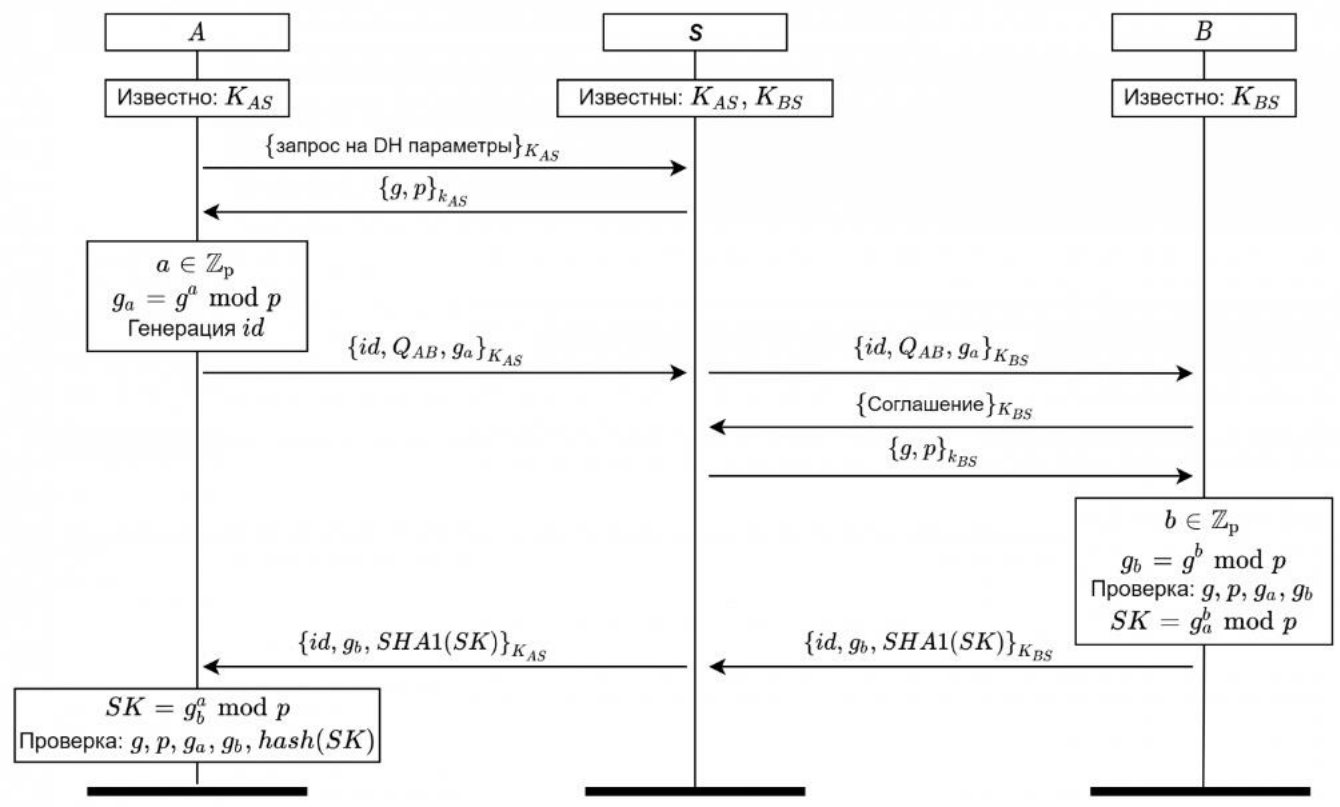
- CBC-IGE (Infinite Garble Extension) – это режим расширения с бесконечными искажениями (разновидность CBC), которая гарантирует, что при изменении блока зашифрованного текста этот блок и каждый блок после него НЕ будут расшифрованы правильно из-за быстрого распространения ошибок

Секретные чаты и сквозное шифрование



- ***SK (Secret chart Key) -*** симметричный секретный ключ, формируемый по протоколц согласования ключей DH

Протокол согласования секретного ключа SK



- Реализован также алгоритм смены секретного ключа SK на основе протокола DH со сменными параметрами
- Секретные ключи, используемые в сквозном шифровании сторон, сменяются каждые 100 сообщений или же каждую неделю.

1. ***A:*** получает параметры g, p от ***S***, генерирует id (для определения текущей сессии), рассчитывает $g_a = g^a mod p$. Отправляет $\{id, Q_{AB}, g_a\}_{K_{AS}}$, где Q_{AB} – запрос на инициализацию секретного чата от клиента ***A*** клиенту ***B.***

2. ***B:*** при согласии на инициализацию секретного чата, получает параметры g, p от ***S,*** рассчитывает $g_b = g^b mod p$, а также $SK = g_a^b mod p$ и его 64-битный хеш (SHA1). Отправляет $\{id, g_a, SHA1(SK)\}_{K_{BS}}$.

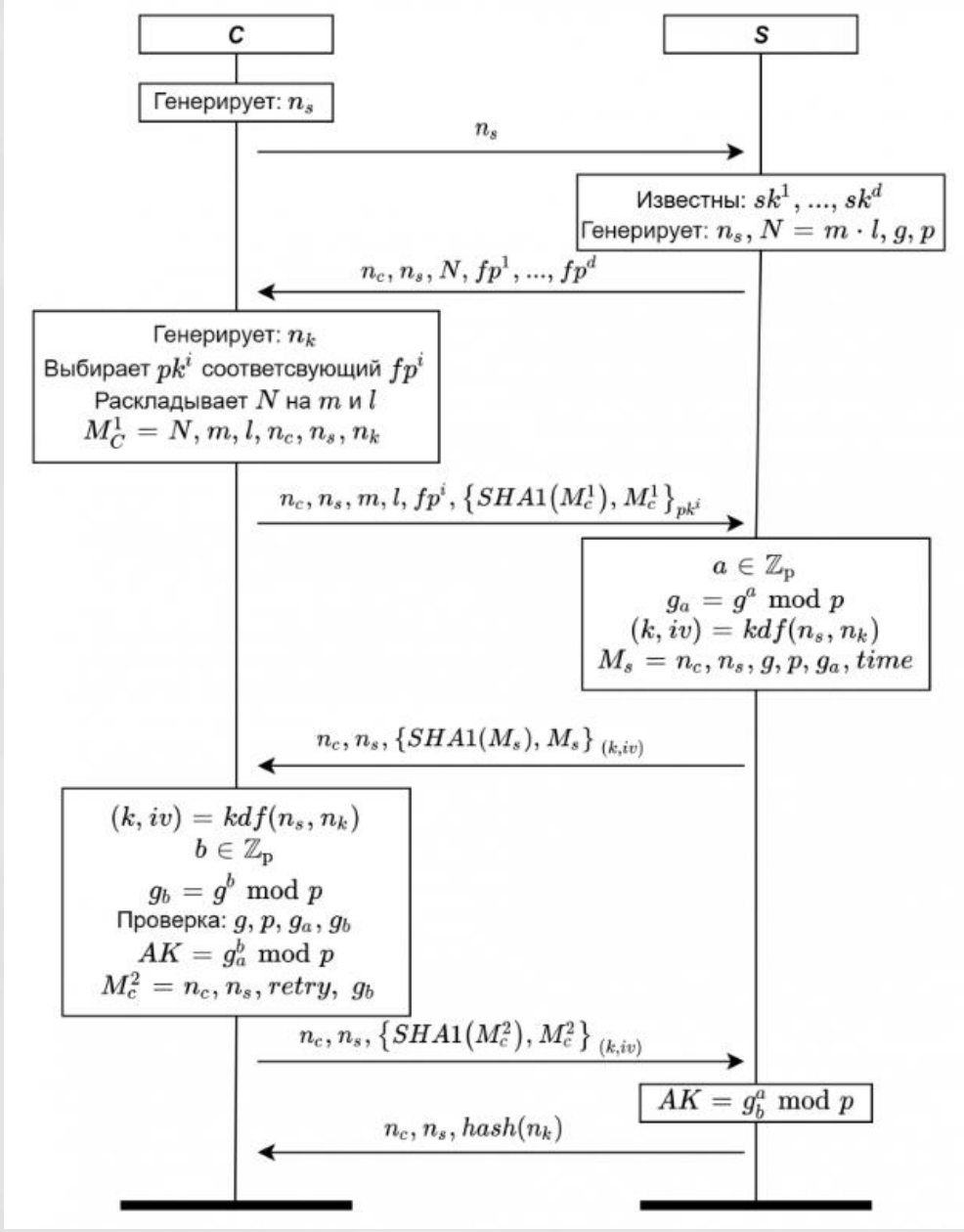
3. ***A:*** рассчитывает $SK = g_b^a mod p$ и сверяет хеш (для удостоверения правильной работы клиентского приложения).

Протокол согласования ключа авторизации (АК)

- В основе гибрид из протокола шифрования RSA и протокола согласования ключей DH:

RSA: sk^i – 2048-битный закрытый ключ RSA, pk^i – 2048-битный открытый ключ, используемые на этапе регистрации и генерации ключа авторизации.

- Детали протокола см. по ссылке https://habr.com/ru/articles/590667/



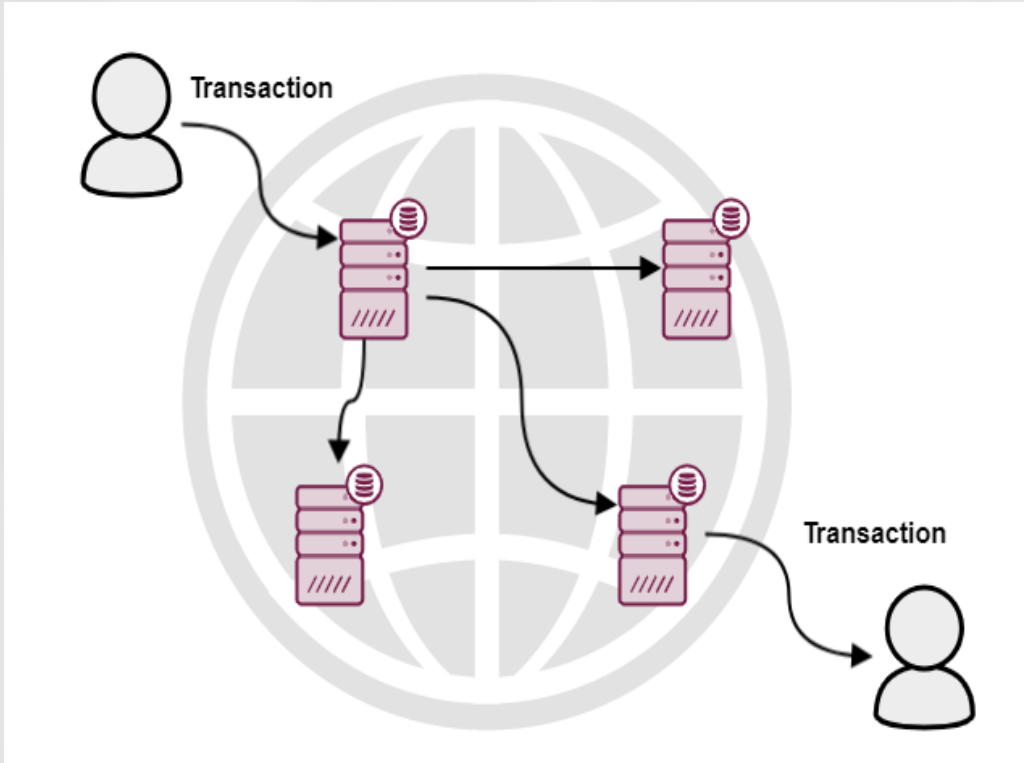
Криптографические методы в технологии Блокчейн (Blockchain)

Задача блокчейна

- Совершение доверительной передачи собственности на цифровые активы (assets) в *недоверительной среде и без посредников*
- Примеры:
  - В сети Bitcoin цифровой актив — это цифровые монеты Bitcoin
  - В сети Ethereum цифровой актив – это программные коды Smart-Contracts

Централизованная сеть

- Доверительный Центр имеет сервера с базами данных, расположенных в разных дата-центрах
- При переводе актива отправителя *Центром проверяется* и регистрируется транзакция
- Транзакция реплицируется на все сервера
- Активы доходят до получателя



Проблемы централизованной сети

- Необходимость идентификации (персонализации) участников со стороны Центра и желание анонимности транзакций участниками
- Корректность выполнения транзакций базируется на доверии к Центру.
- Возможность мошенничества, называемое двойной тратой (double-spending) – потенциальная возможность потратить свой баланс несколько раз, пока транзакция не реплицировалась на все сервера.

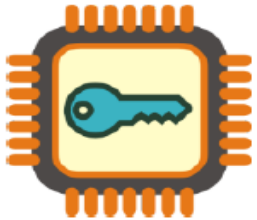
Проблемы централизованной сети (продолжение)

- Возможность атаки на конечное число серверов Центра, которые станут недоступными непредумышленно или по злому умыслу
- Центр обязательно возьмет свою ощутимую комиссию
- Ограниченность управления транзакциями: желание участников не только переводить активы друг другу, но и проверять различные условия прохождения транзакции, программировать сценарии работы, автоматически выполнять действия в зависимости от условий и т.д.

Принципы технологии блокчейн

Децентрализованная - отсутствует единый центр контроля и эмиссии.

Распределенная – данные и их обработка распределены по разным вычислительным узлам системы



Доверие – участники доверяют алгоритмам и проверяют ими информацию других участников, неизменность информации

Публичность – доступность всей информации всем участникам сети

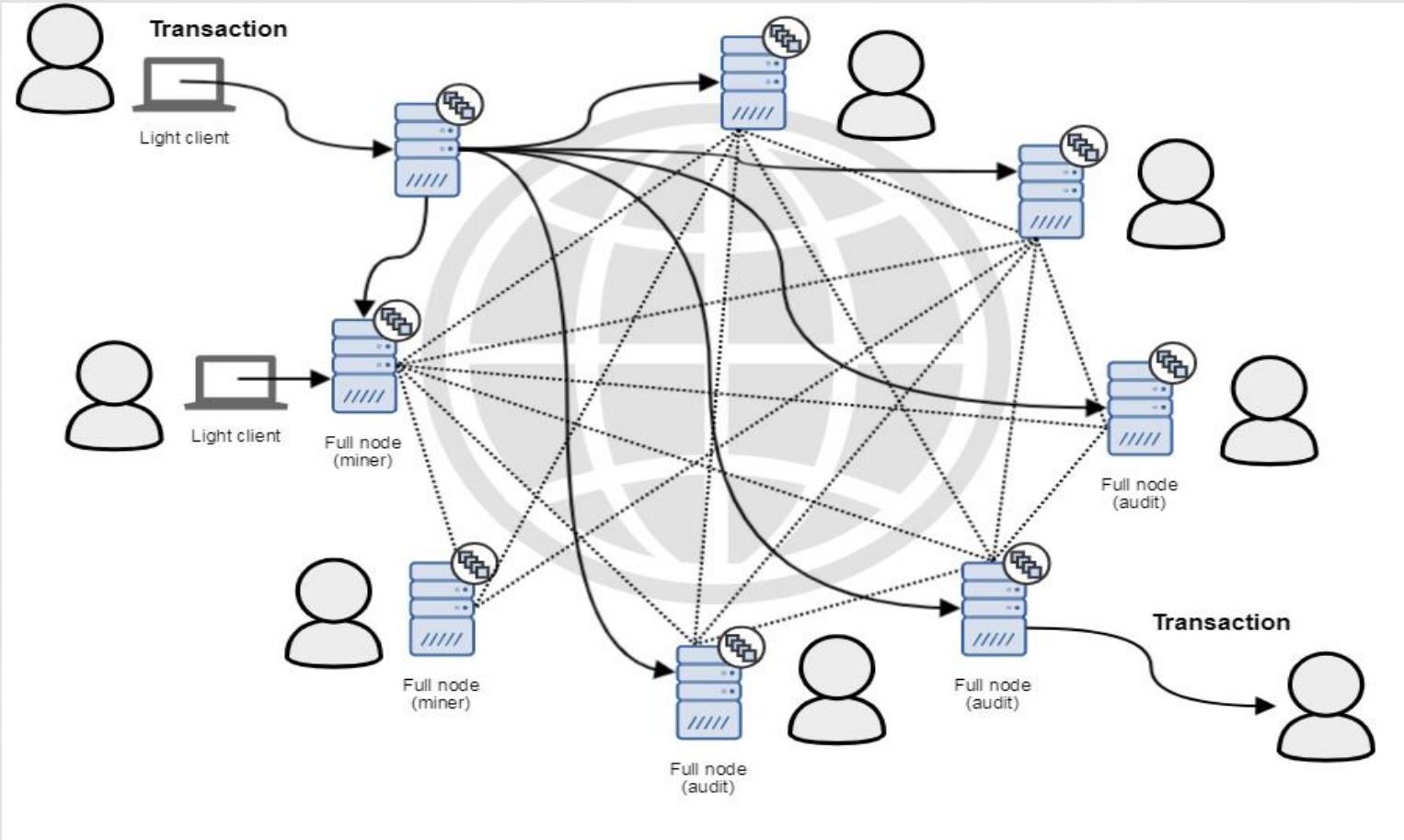
Технологическое решение проблем

- Идентификация участников осуществляется с помощью пары ключей: закрытого и открытого, а алгоритм цифровой подписи однозначно идентифицирует отправителя и получателя, оставляя их личности анонимными
- Транзакции собираются в блоки, вычисляется хэш блока, который записывается в следующий блок. Это делает невозможным незаметное изменение / удаление блоков или отдельных транзакций из блоков
- Мошенничество double-spending предотвращается путем достижения консенсуса в сети, какие данные считать верными, а какие отбрасывать

Технологическое решение проблем

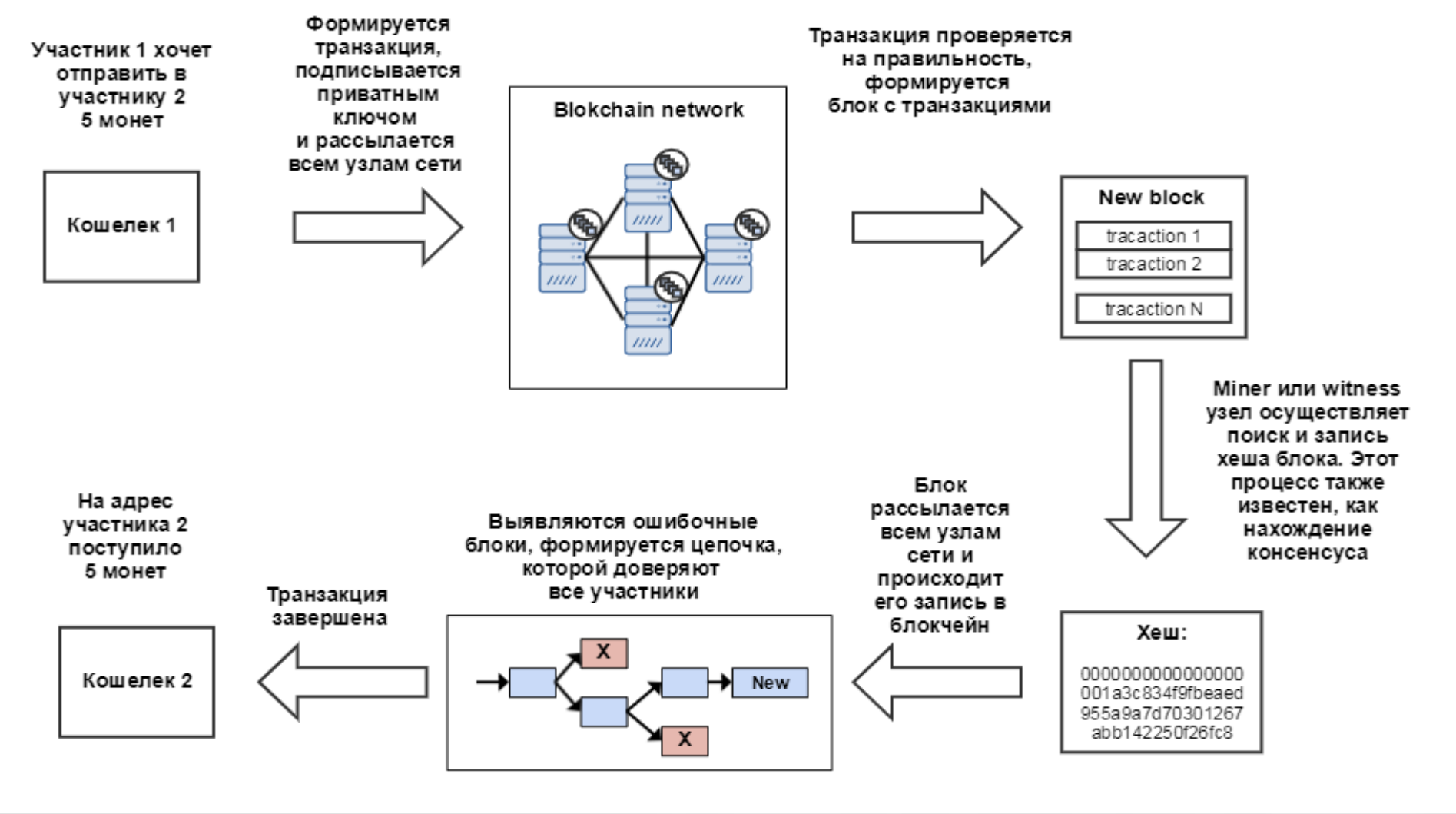
- Надежность функционирования сети достигается тем, что блокчейн является публичным, где каждый участник может получить полную копию блокчейна и, более того, самостоятельно начать проверять транзакции на правильность
- Полностью от комиссии в блокчейне не избавится, т.к. надо платить людям поддерживающим сеть, но в блокчейне необходимость комиссии убедительно доказывается
- Современные блокчейны имеют возможность реализовывать бизнес логику, которая в блокчейне называется Smart Contracts. Логика смарт контрактов реализуются на различных языках высокого уровня.

Архитектура сети блокчейн



- Нода (node) — это ПО, позволяющее взаимодействовать с сетью, подтверждать транзакций и блоки, проверять блоки, таким образом, обеспечивать безопасность и безотказную работу сети.

Жизненный цикл транзакции



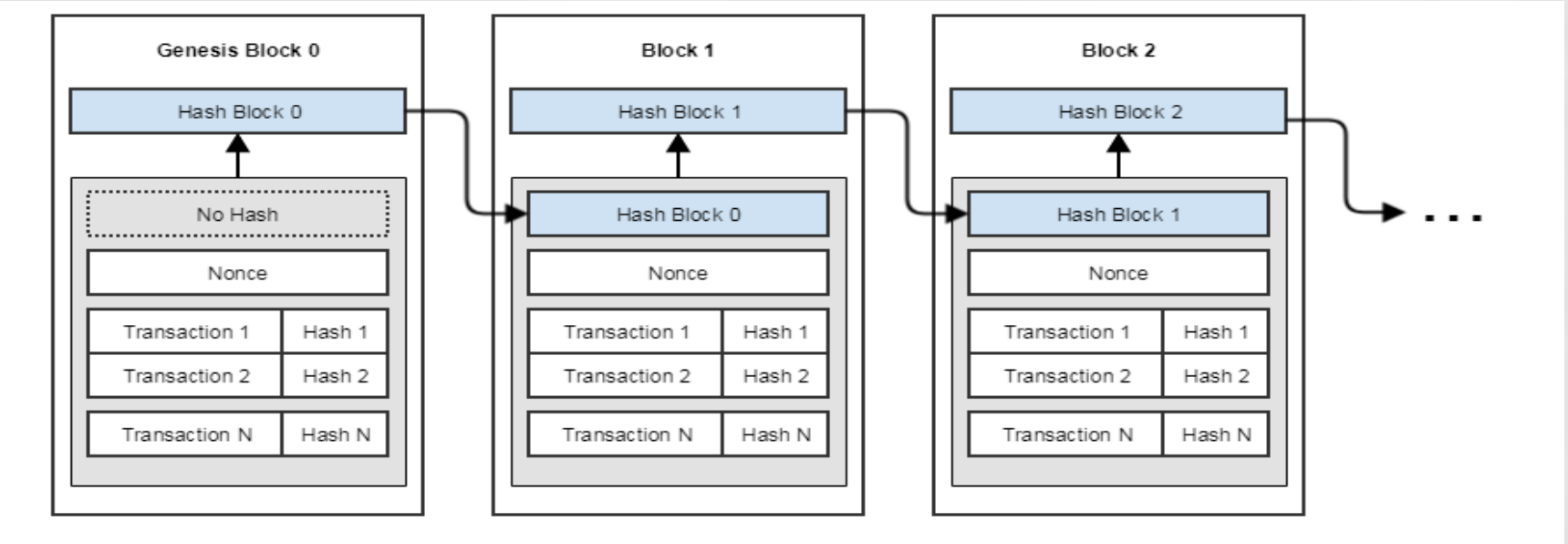
Идентификация и аутентификация участников

- Каждая блокчейн транзакция должна быть подписана цифровой подписью, например, алгоритмом на эллиптических кривых
- Для совершения транзакции каждый участник должен иметь пару ключей: private / public.
- Эту пару ключей называют кошелек (wallet), т.к. ключи однозначно связаны с уникальным цифровым адресом и балансом участника
- Закрытый ключ должен быть строго секретен и храниться в безопасности. При его утери доступ к цифровому активу восстановить невозможно

Транзакции

- Каждая транзакция имеет по крайней мере следующие данные:
  - *From: 0x48C89c341C5960Ca2Bf3732D6D8a0F4f89Cc4368 - цифровой адрес отправителя*
  - *To: 0x367adb7894334678b90afe7882a5b06f7fbc783a - цифровой адрес получателя*
  - *Value: 0.0001 - сумма транзакции T*
  - *Transaction Hash: 0x617ede331e8a99f46a363b32b239542bb4006e4fa9a2727a6636ffe3eb095cef - хэш транзакции*
- Транзакция подписывается секретным ключом и рассылается всем узлам (нодам) в блокчейне для проверки на валидность.
- Алгоритм проверки транзакции включает два десятка шагов, например, передаваемый актив не превышает запаса этого актива

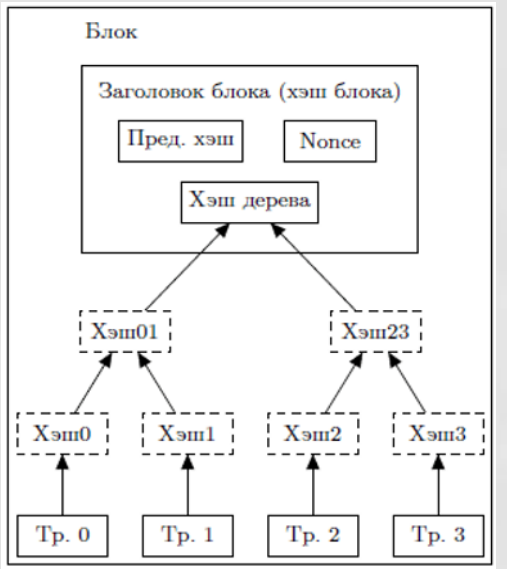
Блоки транзакций



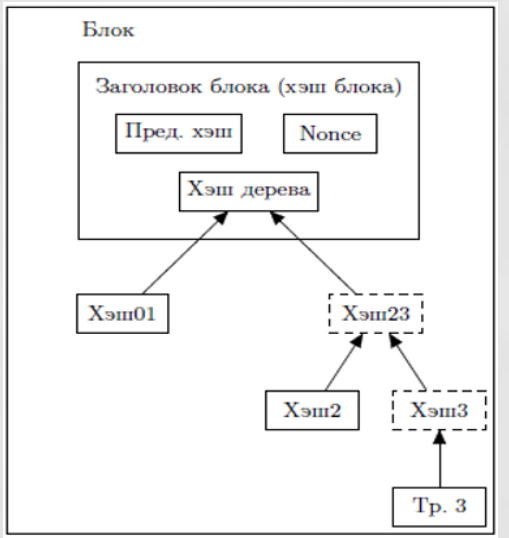
- Хэш блока *должен обладать установленным условиям сложности* – не превосходить заданное число (т.е. иметь определенное число нулей в начале)
- Для подбора хэша используется поле Nonce - это единственные данные в блоке, которые можно изменить
- Успешное нахождение хэша и является доказательством проделанной работы (Proof-of-Work, PoW) для сетей Bitcoin или Ethereum.
- Процесс нахождения хэшей называется майнингом (mining)

Экономия дискового пространства

- Как только последняя транзакция в цепочке, связанной с активом, окажется внутри достаточно старого блока, все предшествующие ей транзакции в цепочке могут быть удалены в целях очистки дискового пространства.
- Чтобы хэш блока остался неизменным, все транзакции в блоке хранятся в виде хэш-дерева Меркла и лишь его корень включается в хэш блока
- Размер старых блоков может быть уменьшен за счет удаления ненужных ветвей этого дерева, а хранить промежуточные хэши необязательно



Хэш-дерево транзакций



После удаления Тр. 0–2 из блока

Сферы применения blockchain вне финансовых сервисов

- Авторство и право владения
- Операции с товарами и сырьем
- Управление данными
- Бриллианты
- Цифровая идентичность, проверка подлинности и подтверждение прав доступа
- Энергетика
- Средства электронного голосования
- Азартные и видеоигры
- Организация частного и государственного управления
- Интернет вещей
- Биржи труда
- Прогнозирование рынка
- Распространение мультимедиа и другого контента
- Сетевая инфраструктура
- «Прозрачная» благотворительность и общественно полезная деятельность
- Недвижимость
- Репутационные рейтинги
- Сервисы райдшеринга
- Социальные сети
- Сертификация цепочек поставки в пищевой промышленности

Спасибо за внимание !