Всероссийский конкурс учебно-исследовательских работ старшеклассников по политехническим, естественным, математическим дисциплинам для учащихся 9-11 классов.

Информатика и информационные технологии

# Использование мессенджеров, основанных на одноранговых сетях

Митрополит Иван Юрьевич,
11 класс, МБОУ Лицей №1,
Пермь

Батин Сергей Евгеньевич,
Учитель информатики

Пермь – 2018

# Введение

В данное время все большую популярность обретают сервисы, при помощи которых люди могут обмениваться файлами или сообщениями на расстоянии. Существует бесчисленное количество данных сервисов, называемых мессенджерами, и подавляющее большинство из них ничем не отличаются друг от друга. Каждый разработчик пытается заработать на популярных нуждах людей, но далеко не каждый может похвастаться чем-то уникальным в своей программе. Но существует несколько программ, содержащих новаторские функции и способных удивить даже опытных пользователей данных программ.

Сейчас одним из главных требований, помимо прочих, к мессенджерам является безопасность и конфиденциальность. Все знаменитые сервисы соответствуют данным требованиям, но соответствуют до того момента, как уполномоченные органы не потребуют предоставить переписки того или иного лица. Существуют так же мессенджеры, пользователям которых такое точно не грозит. Данные сервисы работают на основе одноранговых сетей, то есть все сообщения хранятся только на устройствах двух или нескольких пользователей, которые обмениваются сообщениями друг с другом. Для защиты сообщений от сторонних лиц, на чьих компьютерах так же хранятся эти сообщения, используется асимметричное шифрование или, другими словами, шифрование с открытым ключом.

# Теоретическая часть

## Bitmessage

Сеть Bitmessage работает по принципу шифрования всех входящих и исходящих сообщений каждого пользователя, используя алгоритмы асимметричного шифрования; таким образом, только получатель сообщения способен его расшифровать. [3]

Чтобы обеспечить анонимность:

1. Система рассылает все сообщения всем узлам этой сети, перемешивая тем самым эти сообщения.
2. Система использует длинные адреса вида BM-BcbRqcFFSQUUmXFKsPJgVQPSiFA3Xash, которые могут создаваться пользователем локально практически в неограниченном количестве.
3. Система использует алгоритмы шифрования с открытым ключом, тем самым только получатель может расшифровать сообщение. Особенности алгоритма таковы, что даже исходный отправитель сообщения не может расшифровать своё собственное сообщение обратно, потому что для шифрования и расшифровывания используются разные ключи.

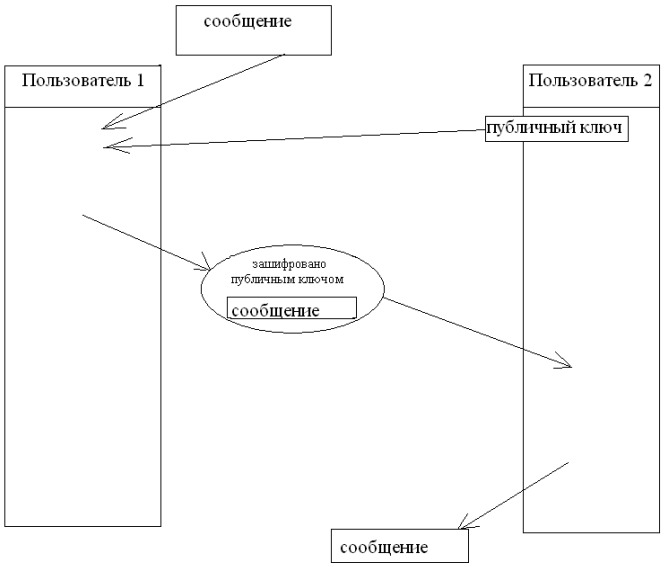


*Рис. 1 Схема работы асимметричного шифрования при передаче сообщения*

4.Отправляемое сообщение не содержит адрес получателя, поэтому каждый участник сети пытается расшифровать абсолютно все сообщения, даже не предназначенные для него. Поскольку участник сети способен расшифровать только сообщения, предназначенные ему, то участник знает, что сообщения, которые он не смог расшифровать, были предназначены не ему. [1]

Сеть являлась и до сих пор является популярной в достаточно узких кругах пользователей, которые особенно беспокоятся о безопасности своих переписок. С каждым годом количество пользователей растет, а вместе с ним растет желание людей скрывать свои переписки от лишних глаз. Содержание переписок абсолютно разное: от безобидных разговоров о садоводстве, до обсуждения остро политических тем.[4][7]

| дни/показатели | Пользователи | Сообщения | Широковещательные сообщения |
|---|---|---|---|
| 9 | ≈400 | ≈1600 | ≈200 |
| 10 | ≈300 | ≈1850 | ≈200 |
| 11 | ≈325 | ≈1500 | ≈210 |
| 12 | ≈350 | ≈1800 | ≈400 |
| 13 | ≈325 | ≈1550 | ≈450 |
| 14 | ≈275 | ≈1200 | ≈370 |
| 15 | ≈250 | ≈1400 | ≈1400 |
| 16 | ≈400 | ≈1600 | ≈250 |
| 17 | ≈275 | ≈1200 | ≈220 |
| 18 | ≈300 | ≈1350 | ≈250 |
| 19 | ≈225 | ≈1000 | ≈400 |
| 20 | ≈175 | ≈1000 | ≈3250 |
| 21 | ≈400 | ≈1400 | ≈3100 |
| 22 | ≈350 | ≈900 | ≈3600 |
| 23 | ≈375 | ≈1200 | ≈3300 |
| 24 | ≈275 | ≈1250 | ≈600 |
| 25 | ≈300 | ≈1200 | ≈200 |
| 26 | ≈275 | ≈1300 | ≈210 |
| 27 | ≈325 | ≈1250 | ≈230 |
| 28 | ≈300 | ≈1150 | ≈200 |
| 29 | ≈330 | ≈1300 | ≈210 |
| 30 | ≈340 | ≈1400 | ≈220 |
| 1 | ≈450 | ≈1430 | ≈220 |
| 2 | ≈400 | ≈1250 | ≈150 |
| 3 | ≈725 | ≈1600 | ≈170 |
| 4 | ≈600 | ≈1400 | ≈170 |
| 5 | ≈475 | ≈1200 | ≈170 |
| 6 | ≈490 | ≈1220 | ≈175 |
| 7 | ≈460 | ≈1500 | ≈170 |

*Таб.1 данные об использовании Bitmessage на октябрь 2013, спустя год после запуска протокола*

## Шифрование с открытым ключом

Идея криптографии с открытым ключом тесно связана с идеей односторонних функций, таких функций f(x), в которых по известному x довольно просто найти значение f(x), но определение x из f(x) невозможно за разумный срок. Односторонней функцией можно зашифровать сообщение, но расшифровать нельзя. Поэтому криптография с открытым ключом использует односторонние функции с лазейкой. Лазейка — это некий секрет, который помогает расшифровать. То есть существует такой y, что зная f(x) и y, можно вычислить x.[8]

Пример: каждый пользователь в сети имеет свой пароль. При входе он указывает имя и вводит секретный пароль. Но если хранить пароль на диске компьютера, то его может считать кто-то, кто имеет доступ к файлам, например, администратор, и получить доступ к секретной информации. Для решения задачи используется односторонняя функция. При создании секретного пароля в компьютере сохраняется не сам пароль, а результат вычисления функции от этого пароля и имени пользователя. Например, пользователь 1 придумала пароль «АБВ». При сохранении этих данных вычисляется результат функции (ПОЛЬЗОВАТЕЛЬ1_АБВ), пусть результатом будет строка ПРО, которая и будет сохранена в системе. В результате файл паролей примет следующий вид:

| **Имя** | **(имя_пароль)** |
|---|---|
| **ПОЛЬЗОВАТЕЛЬ1** | **ПРО** |
| **ПОЛЬЗОВАТЕЛЬ2** | **МИТ** |

*Таб.2 вид файла, содержащего пароли*

| **Имя:** | **ПОЛЬЗОВАТЕЛЬ1** |
|---|---|
| **Пароль:** | **АБВ** |

*Таб.3 вид  входа в систему*

Когда Алиса вводит «секретный» пароль, компьютер проверяет, даёт или нет функция, применяемая к ПОЛЬЗОВАТЕЛЬ1_АБВ, правильный результат ПРО, хранящийся на диске компьютера. Ели изменить хотя бы одну букву в имени или в пароле, то результат функции будет отличаться. Пароль не хранится в компьютере ни в каком виде. [2][10]

## Применение

Алгоритмы криптосистемы с открытым ключом можно использовать:

1. Как самостоятельное средство для защиты передаваемой и хранимой информации,
2. Как средство распределения ключей (обычно с помощью алгоритмов криптосистем с открытым ключом распределяют ключи, малые по объёму, а саму передачу больших информационных потоков осуществляют с помощью других алгоритмов),
3. Как средство аутентификации пользователей.

4. Электронная цифровая подпись

5. Смарт-карты

Преимущества асимметричных шифров перед симметричными:

1. Не нужно предварительно передавать секретный ключ по надёжному каналу; только одной стороне известен ключ дешифрования, который нужно держать в секрете (в симметричной криптографии такой ключ известен обеим сторонам и должен держаться в секрете обеими);
2. В больших сетях число ключей в асимметричной криптосистеме значительно меньше, чем в симметричной.

Недостатки алгоритма асимметричного шифрования в сравнении с симметричным:

1. В алгоритм сложно внести какие-либо изменения;
2. Более длинные ключи — ниже приведена таблица, сопоставляющая длину ключа симметричного алгоритма с длиной ключа RSA с аналогичной криптостойкостью:

| **Длина симметричного ключа, бит** | **Длина ключа RSA, бит** |
|---|---|
| 56 | 384 |
| 64 | 512 |
| 80 | 768 |
| 112 | 1792 |
| 128 | 2304 |

*Таб.4 соотношение размеров симметричных и асимметричных ключей*

3. Шифровка-расшифровка с использованием пары ключей проходит на два-три порядка медленнее, чем шифровка-расшифровка того же текста симметричным алгоритмом. [5][9]
4. При симметричном шифровании ключ нужно держать в секрете обеим сторонам и менять его после каждой передачи, при асимметричном же шифровании только один ключ держится в секрете, а оба ключа можно длогое время не менять.[6]

# Исследование

Объектом исследования будет являться протокол Bitmessage ввиду его популярности и новаторства среди данного сегмента мессенджеров

Одним из основных факторов, влияющих на работу группового чата, а точнее на скорость его работы, является количество узлов, из которых состоит система. Помимо этого, на скорость работы системы влияет размер сообщений, скорость интернет-соединения.

Данная работа ставит целью оценить эффективность работы системы группового чата в зависимости от количества узлов в этой системе. Для этого нужно оценить время, за которое один узел закончит свою работу.

Общее время работы узла будет складываться из времени проверки всех сообщений узлом на возможность расшифровки и времени расшифровки сообщений, которые узел может расшифровать. Существует так же время на передачу сообщения по сети, но этим временем можно пренебречь ввиду его независимости от количества узлов.

Рассмотрим систему из n равноправных узлов, которые генерируют одинаковое количество сообщений в промежуток времени:

Промежутком времени примем сутки.

1)Допустим, что количество сообщений, адресуемых одному узлу, равно количеству сообщений, генерируемых этим узлом.

2)Чем больше количество узлов, тем меньше от него зависит количество сообщений, отправляемых узлом; рассмотрим на примере:

Изначально сеть состоит из двух узлов, общающихся друг с другом. Со временем сеть начинает увеличиваться, и среди прибавляющихся узлов все реже встречаются собеседники у произвольного пользователя, например, знакомые, потому что количество узлов увеличивается вне зависимости от количества знакомых того или иного пользователя. Помимо того, физически человек не способен общаться со слишком большим количеством узлов, даже если у него есть желание.

n – количество узлов в сети

N – среднее количество сообщений, генерируемых узлом в сутки

Таким образом, для описания зависимости N от n возьмем логарифмическую функцию.

Получаем: $N(n)=k*\log_2(n/n0)$, где k и n0 – постоянные, связывающие количество узлов и количество сообщений.

Тогда общая формула для вычисления времени работы узла выглядит так:

$T= k*\log_2(n/n0)*(t1+t2*n)$

t1 – время на расшифровку одного сообщения узлом

t2 – время на проверку узлом одного сообщения на возможность расшифровки

Для вычисления констант n0 и k подставим в функцию средние значения:

$30=k*\log_2(100/n0)$

$60=k*\log_2(10000/n0)$

При решении системы уравнений получились значения:

k=4,28; n0=0,78

3)Вычислим величину t1:

Средняя скорость шифровки/расшифровки сообщения при 512-битном ключе и процессоре 2ГГц составляет примерно 30кбит/с. При среднем размере сообщения в 15кбайт расшифровка занимает 4 секунды – t1=4с.

4)Поскольку величина t2 неизвестна, сделаем расчет для разных значений этой величины.

Таким образом, объединив все значения в формулу, получаем:

$T(n)=4,28*\log_2(n/0,78)*(4+t2*n)$.

Вычислим зависимость времени работы узла от количества участников в его сети при t2=0,4с; t2=0,04с; t2=0,004с; t2=0,0004с; t2=0,00004с:

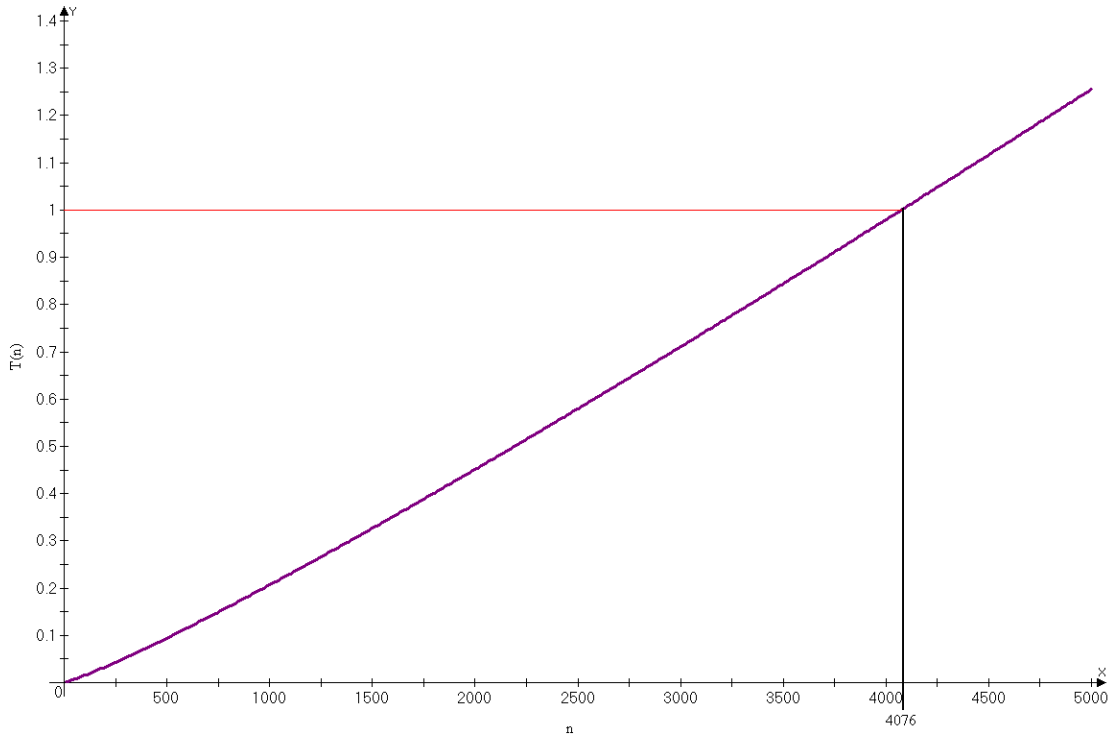


*Рис. 2 Зависимость времени работы узла от количества узлов при t2=0,4с.*

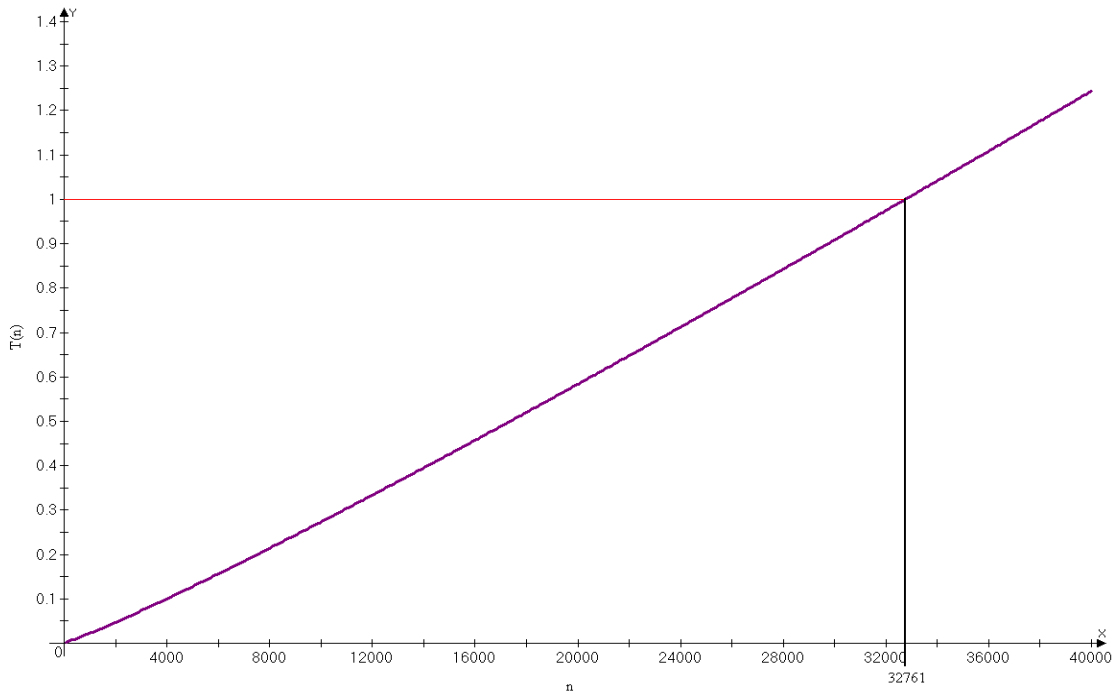


*Рис. 3 Зависимость времени работы узла от количества узлов при t3=0,04с*

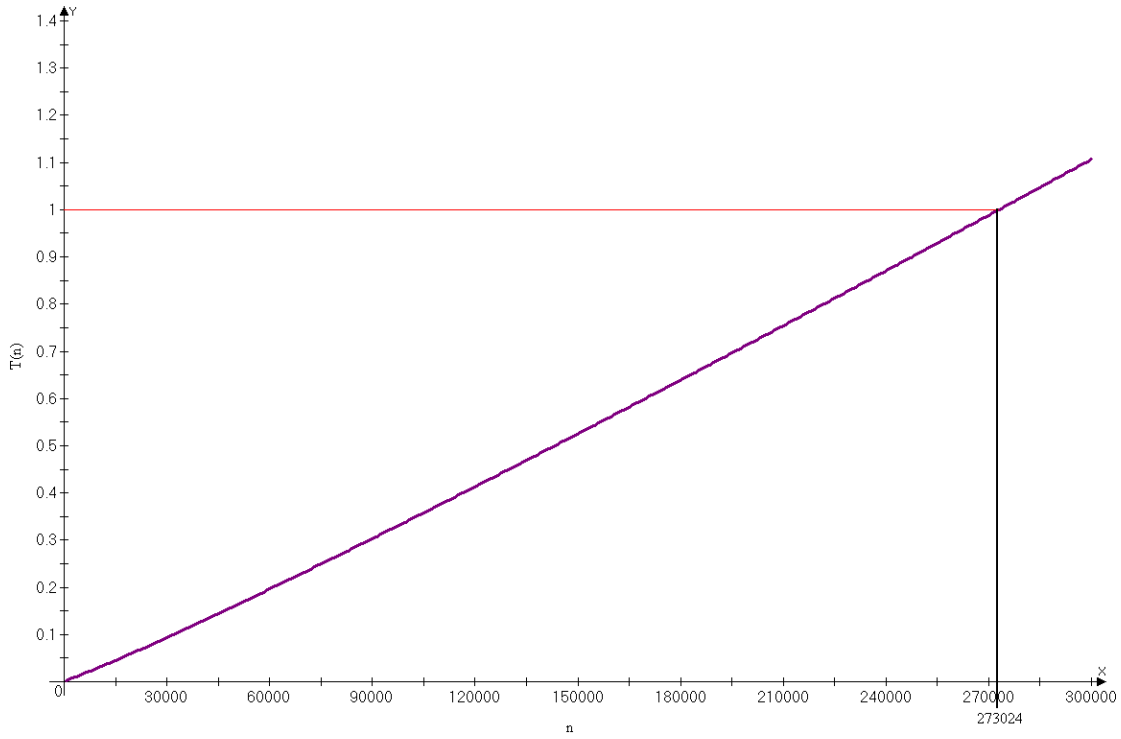


*Рис. 4 Зависимость времени работы узла от количества узлов при t2=0,004с*

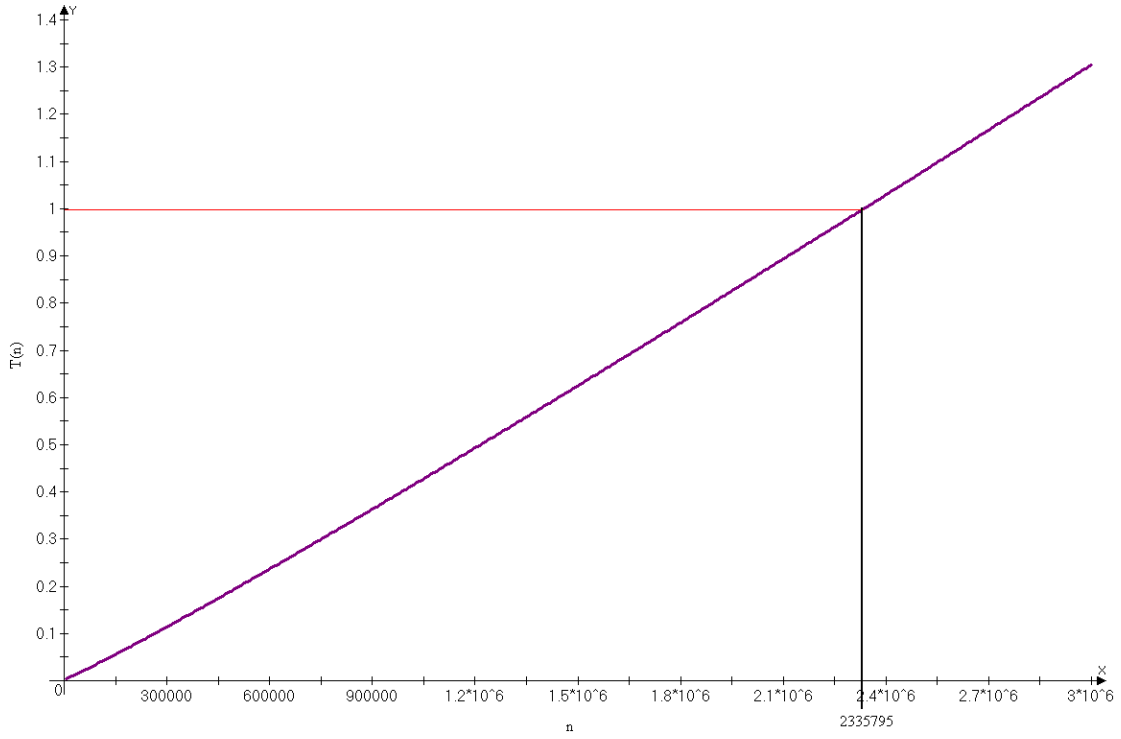


*Рис. 5 Зависимость времени работы узла от количества узлов при t2=0,0004с*

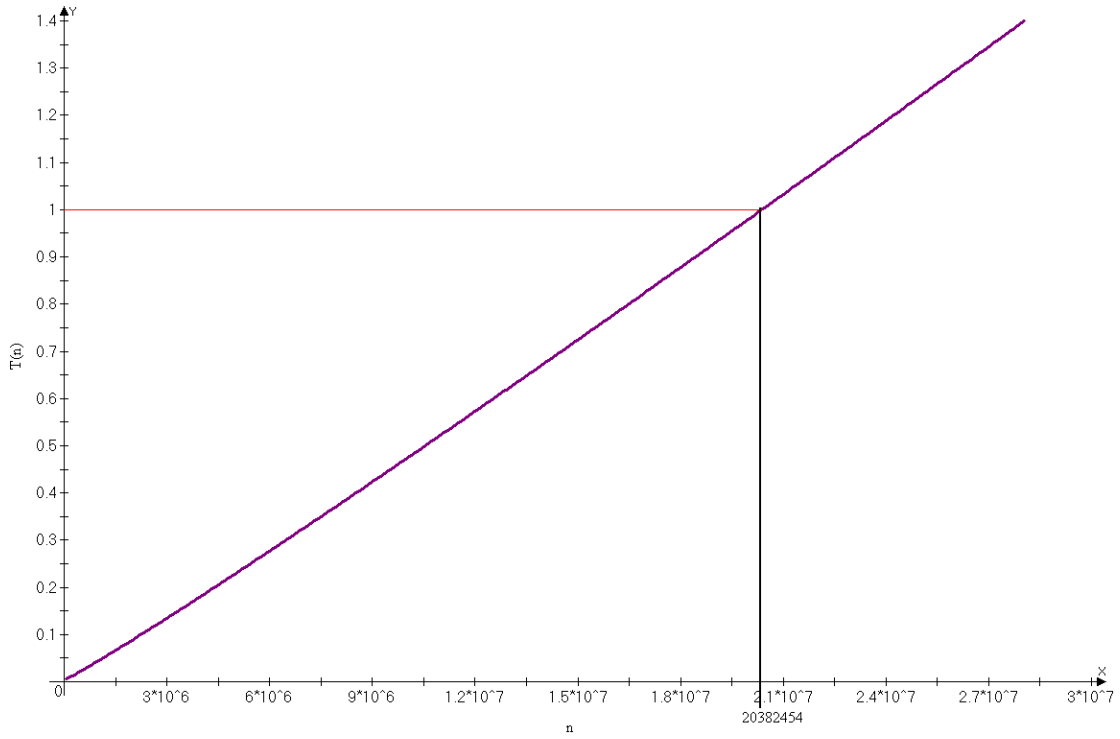


*Рис. 6 Зависимость времени работы узла от количества узлов при t2=0,00004с*

Точка 1 по оси Y на каждом из графиков обозначает критическую точку, при которой время работы узла составляет одни сутки - время, за которое в сети образовалось данное количество сообщений. При дальнейшем увеличении количества узлов исследуемый узел будет обрабатывать сообщения за время большее, чем время образования этих сообщений в сети, и вся система уже не будет работать эффективно.

На разных графиках критическая точка соответствует разному количеству узлов, это связано со временем на проверку сообщений на возможность расшифровки; чем время на проверку меньше, тем большее количество узлов может находиться в системе.

Ниже приведено соответствие времени на проверку одного сообщения (t2) и количества узлов (n max), при котором система работает эффективно:

При t2=0,4с n max=4076 узлов

При t2=0,04с n max=32761 узлов

При t2=0,004с n max=273024 узлов

При t2=0,0004с n max=2335795 узлов

При t2=0,00004с n max=20382454 узлов

# Заключение

В ходе проделанной работы был изучен материал по протоколу для обмена сообщениями Bitmessage, по принципу работы этого протокола, а именно по одноранговым, или децентрализованным сетям, шифрованию с открытым ключом, односторонним функциям. Так же были получены данные о работе протокола и зависимости времени работы системы, состоящей из n равноправных узлов. Для этого была выведена формула зависимости полного времени работы одного узла от количества пользователей в сети, были построены графики этой зависимости при разных значениях времени на проверку одного сообщения, а также были получены данные о критическом количестве пользователей, при котором система перестает работать эффективно.

## Источники


1. Freedom Belarus [Электронный ресурс]. – Режим доступа: https://freedombelarus.github.io/about-bitmessage.html , свободный.

2. Handbook of Applied Cryptography, Menezes A.J., Oorschot P.C., Vanstone S.A. C. 25—26

3. Linux – обзоры, Linux для начинающих [Электронный ресурс]. – Режим доступа: http://zenway.ru/page/pybitmessage , свободный.

4. Technology box – портал, посвященный вопросам информационной безопасности [Электронный ресурс]. – Режим доступа: http://teh-box.ru/informationsecurity/algoritm-shifrovaniya-rsa-na-palcax.html, свободный.

5. www.razlib.ru Библиотека обучающей и информационной литературы [Электронный ресурс]. – Режим доступа: http://www.razlib.ru/kompyutery_i_internet/zashiti_svoi_kompyuter_na_100_ot_virusov_i_hakerov/p4.php , свободный.

6. Асимметричные шифры [Электронный ресурс]. – Режим доступа: http://kryptography.narod.ru/assimmetr.html, свободный.

7. Личный архив Евгения Золотова, 1999-2017 [Электронный ресурс]. – Режим доступа: http://knoppix.ru/sentinel/081013.html, свободный.

8. Саломаа А. Криптография с открытым ключом. — М.: Мир, 1995. — 318 с.

9. Студопедия - лекционный материал для студентов [Электронный ресурс]. – Режим доступа: https://studopedia.su/12_81487_asimmetrichnoe-shifrovanie.html, свободный.

10. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си = Applied Cryptography. Protocols, Algorithms and Source Code in C. — М.: Триумф, 2002. — 816 с.