УДК 621.396

ПЕРЕДАЧА ТРАФИКА В ПРОГРАММНО-КОНФИГУРИРУЕМЫХ СЕТЯХ С ИСПОЛЬЗОВАНИЕМ АУТЕНТИФИКАЦИИ И ШИФРОВАНИЯ

К. И. Никишин, к.т.н., доцент кафедры ВТ ПГУ, Пенза, Россия;
orcid.org/0000-0001-7966-7833, e-mail: nkipnz@mail.ru
М. А. Митрохин, д.т.н., доцент, заведующий кафедрой ВТ ПГУ, Пенза, Россия;
orcid.org/0000-0001-6719-4610, e-mail: vt@pnzgu.ru
К. А. Артемов, студент кафедры ВТ ПГУ, Пенза, Россия;
orcid.org/0009-0008-9482-6271, e-mail: artk_007@mail.ru
Д. Д. Григорьева, студент кафедры ВТ ПГУ, Пенза, Россия;
orcid.org/0009-0006-9211-7012, e-mail: g.darya.work@gmail.com
Е. А. Мереняшева, студент кафедры ВТ ПГУ, Пенза, Россия;
orcid.org/0000-0003-3744-4521, e-mail: lizamerenyasheva@gmail.ru

Управление трафиком, критериями в программно-конфигурируемых сетях (ПКС) выполняется контроллером в связке с коммутатором, который работает по открытому протоколу OpenFlow. Протокол OpenFlow является основной частью ПКС. Таким образом, открытый протокол OpenFlow успешно позволяет справляться с возросшими требованиями к пользовательскому трафику, приоритету. Актуальность исследования состоит в том, что существует 9 типов атак на ПКС, атаки на топологию, такие как Fake Topology и ARP (Address Resolution Protocol) Positioning, которые связаны с безопасностью и решаются добавлением этапов аутентификации и шифрования передаваемых данных. ***Цель исследования*** *– осуществить передачу трафика в ПКС с использованием аутентификации и побитного шифрования на стороне контроллера для снижения в сети атак «человек посередине». В алгоритме используется побитное симметричное шифрование ключом, полученным по алгоритму Диффи-Хеллмана. Для исследования алгоритма была разработана имитационная модель, включающая использование алгоритма на аппарате сетей Петри. Результаты моделирования показали, что предложенная передача трафика в ПКС использованием аутентификации и побитного шифрования на стороне контроллера позволила обеспечить безопасную передачу для атак типа «человек посередине» в ПКС, вычислять ключи, не позволяя передавать по сети в открытом виде, а также повысить скорость шифрования трафика в ПКС.*

Ключевые слова: *программно-конфигурируемые сети, контроллер, коммутатор, OpenFlow, сети Петри, CPN Tools, симметричное шифрование, алгоритм Диффи-Хеллмана.*

DOI: 10.21667/1995-4565-2023-86-63-72

Введение

При приеме, обработке, а также передаче разнородного пользовательского трафика важными условиями в сети являются задержка кадра, быстродействие, диапазон частот, переполнение очередей в коммутаторе Ethernet. Многие эти условия становятся управляемыми в программно-конфигурируемых сетях (ПКС) [1-3] в отличие от привычных компьютерных сетей. Основным протоколом управления в ПКС является протокол OpenFlow [2, 4].

На сегодняшний день ПКС находят широкое применение и продолжают распространяться по всему миру благодаря набирающим популярность облачным сервисам [5]. Кроме этого, ПКС используются в центрах обработки данных [6]. ПКС позволяют сократить издержки на сопровождение сети за счёт централизации управления на программном контроллере. Таким образом, повышается динамическое и интеллектуальное управление сетевыми ресурсами в сети.

OpenFlow – это протокол, являющийся неотъемлемой частью ПКС, который организует взаимодействие между элементами сети, такими как маршрутизаторы и коммутаторы, посредством центрального контроллера [7]. Центральный контроллер - устройство, управляющее остальными элементами сети, имеющее в своём составе сетевую операционную систему, которая установлена на выделенном физическом сервере в ПКС.

OpenFlow всегда отличался отсутствием шагов аутентификации, он не требует, чтобы контроллер аутентифицировал коммутаторы, и контроллеру не требуется разрешать коммутаторам доступ к себе. Поскольку это обстоятельство является уязвимостью протокола, оно может затронуть любую реализацию OpenFlow.

Цель исследования заключается в передаче трафика в ПКС с использованием аутентификации и побитного шифрования на стороне контроллера для снижения в сети атак «человек посередине». Задачами исследования являются разработка имитационной модели на основе аппарата сетей Петри согласно передаче трафика, верификация разработанной имитационной модели, оценка эффективности модели.

Теоретическая часть

Исследование передачи трафика в ПКС с использованием аутентификации и побитного шифрования на стороне контроллера проводится на примере атаки «человек посередине». С помощью данной атаки злоумышленник тайно передает и, возможно, изменяет сообщения между двумя сторонами, которые считают, что они напрямую общаются только друг с другом, и не распознают, что между ними добавился злоумышленник [8,9]. Поэтому была выбрана именно эта атака, так как она лучше всего показывает уязвимость протокола OpenFlow [10] в виде отсутствия аутентификации и шифрования данных. Схема атаки «человек посередине» представлена на рисунке 1.

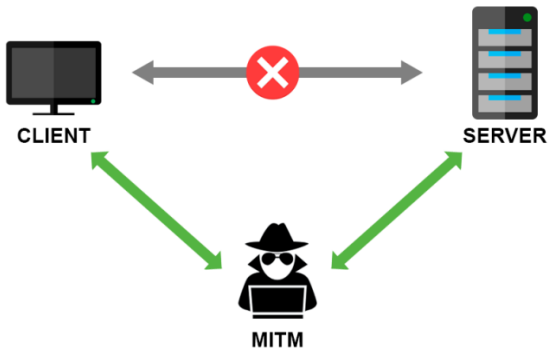


Рисунок 1 – Схема атаки «человек посередине»
Figure 1 – Scheme of «Man in the middle» MITM attack

Для вычисления ключа используем алгоритм Диффи-Хеллмана. Алгоритм Диффи-Хеллмана при использовании этапа аутентификации наиболее устойчив к атакам типа «человек посередине», так как не передаёт ключи по сети [3].

Ключ необходим для реализации алгоритма блочного битового шифрования AMDD (Advanced Method Data Division). Преимущества данного алгоритма над другими алгоритмами симметричного шифрования (простых перестановок, Цезаря, транспонирования):

1) битовое шифрование повышает криптостойкость по сравнению с алгоритмами симметричного шифрования, которые описаны выше, позволяет получить более высокие результаты при частотном анализе (один из методов криптоанализа);

2) подходит для реализации на любых языках программирования, так как не задействует алфавит;

3) возможность выбора ключа любого размера, так как блоки не имеют фиксированного размера, а подбираются по размеру ключа.

Алгоритм AMDD не требует наличия блоков одинаковой длины для выявления потерь и лишних символов. Работа данного алгоритма заключается в разбиении сообщения на фиксированные блоки. Если текст сообщения не делится нацело, то длина последнего блока полу-

чается операцией остатка от деления по модулю ключа. Затем каждый блок шифруется с использованием ключа, полученного на основе алгоритма Диффи-Хеллмана, операцией «Побитовое или». Блок-схема предложенного алгоритма шифрования и дешифрования сообщения представлена на рисунке 2, *а*, блок-схема подпрограммы представлена на рисунке 2, *б*. В блок-схемах используется два ключа: один – на стороне отправителя, другой – на стороне получателя.

Алгоритм передачи сообщений заключается в следующем. Функции для выполнения операций шифрования и дешифрования сообщений выполняются на каждой рабочей станции и контроллере. Рабочая станция зашифровывает сообщение ключом, полученным в ходе авторизации, и отправляет на контроллер свои IP- и MAC-адреса источника, IP-адрес получателя, ключ и зашифрованное сообщение. Контроллер, получая сообщение, расшифровывает сообщение ключом отправителя, снова зашифровывает ключом получателя и отправляет на IP-адрес получателя. Получатель уже расшифровывает сообщение своим ключом.

UML-диаграмма активностей алгоритма вычисления ключа и передачи сообщения представлена на рисунке 3. На этой диаграмме описано взаимодействие двух рабочих станций и контроллера. Рабочая станция 1 отправляет контроллеру информацию о себе (IP-, MAC-адреса). Далее на рабочей станции 1 и на контроллере параллельно генерируется случайное число [x = get rand ()] и с его помощью вычисляется часть ключа по алгоритму Диффи-Хеллмана [key = Calc part Key(p, g, x)].

Контроллер с рабочей станцией 1 обмениваются частями ключей. На рабочей станции 1 и контроллере вычисляется ключ по алгоритму Диффи-Хеллмана, используя полученную часть ключа [Calc key (key)]. В результате на рабочей станции 1 и на контроллере получается полный ключ. Далее на контроллере происходит сохранение ключа вместе с информацией о рабочей станции (IP-, MAC-адреса) в список авторизованных устройств [packing and save PC (IP, MAC, key)].

Рабочая станция 1 генерирует сообщение с помощью функции gen() и сразу зашифровывает его своим ключом encrypt_code(key, text). Далее происходит отправка сообщения с рабочей станции 1 на синхронизирующее устройство типа «почтовый ящик», где сообщения хранятся и обрабатываются по очереди. Сообщение содержит в себе следующую информацию: IP-, MAC-адреса получателя, IP-адрес отправителя и зашифрованное сообщение.

Контроллер обрабатывает очередь из почтового ящика: берет сообщение и расшифровывает его ключом отправителя и заново зашифровывает ключом получателя transcoding[decrypt_code(key, code), encypt_code(key, text)].

Рабочая станция 2 отправляет запрос на получение сообщений контроллеру, по IP-адресу запроса на соответствующую рабочую станцию передается сообщение. При получении сообщения на рабочей станции 2 происходит расшифровывание этого сообщения decrypt_code(key, code).

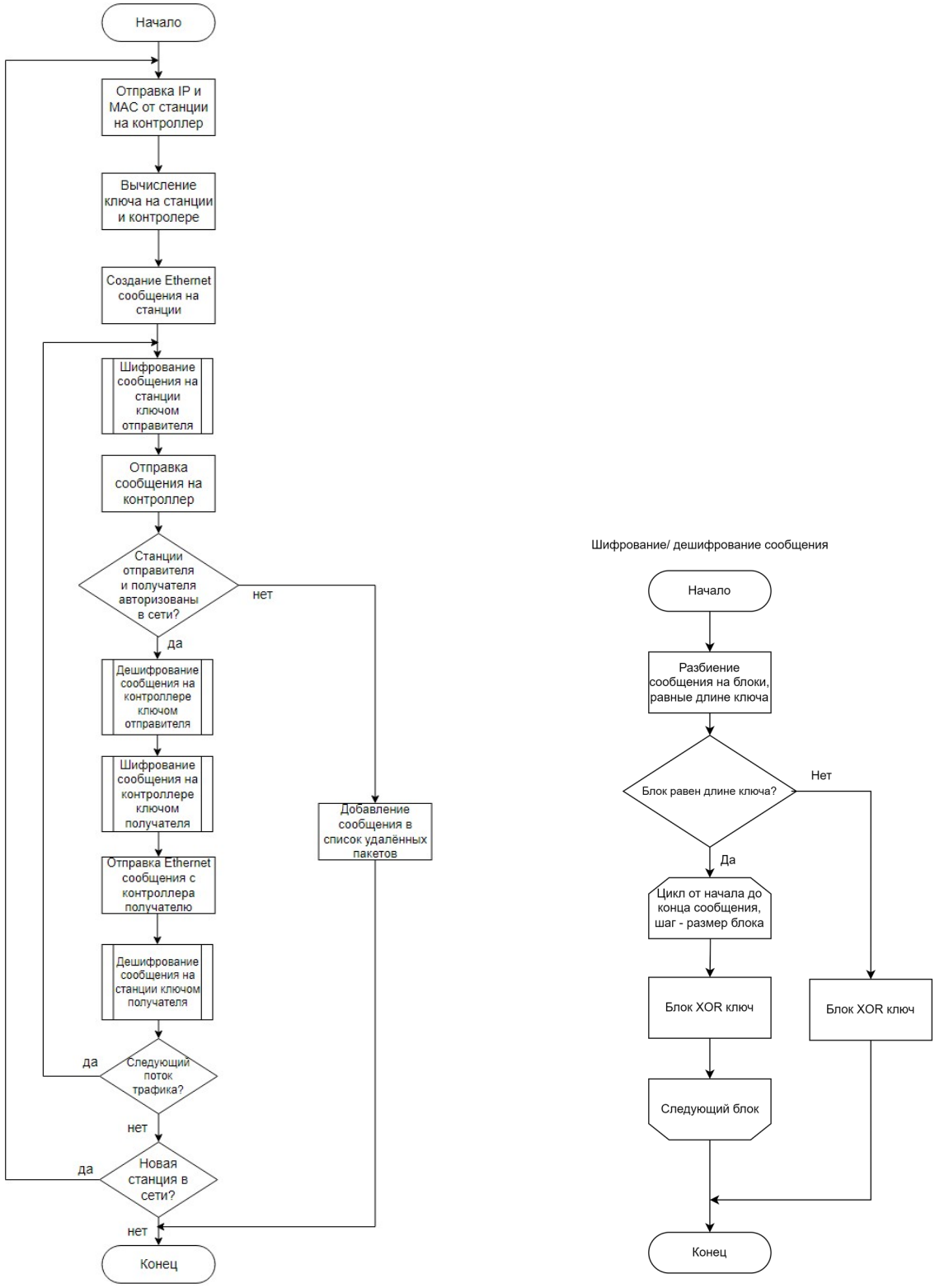
В настройках контроллера и рабочих станциях хранятся константы p (первообразный корень по модулю p) и g (случайное простое число). При запуске рабочая станция и контроллер генерируют случайные числа a и b.

Авторизация проходит следующим образом. Сначала станция отправляет на контроллер свои MAC-, IP-адреса и сгенерированное число, преобразованное алгоритмом Диффи-Хеллмана:

$$A = g^a \bmod p.$$

Далее станция ждёт от контроллера его число, преобразованное по такому же алгоритму:

$$B = g^b \bmod p.$$



а (a) **б (b)**

Рисунок 2: а – Блок-схема алгоритма; б – Блок-схема подпрограммы шифрования/дешифрования сообщения
Figure 2: a –Block diagram of the algorithm;
Figure 2b – Subroutine block diagram for encrypting and decrypting a message

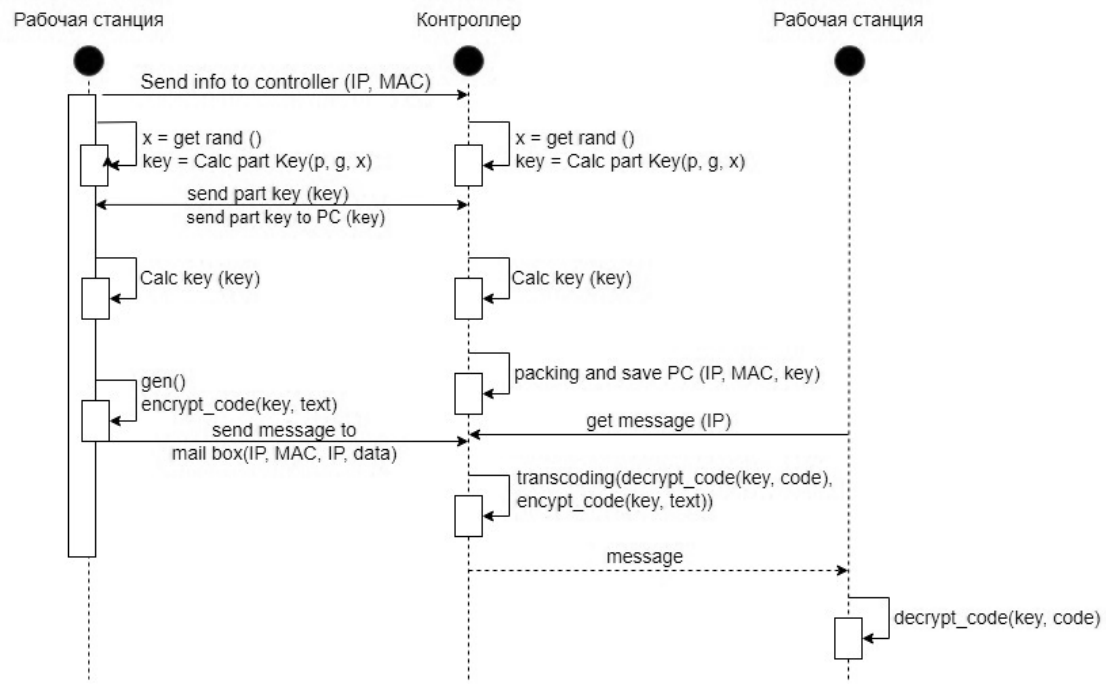


Рисунок 3 – UML-диаграмма алгоритма
Figure 3 – UML diagram of the algorithm

На следующем этапе контроллер и рабочая станция обмениваются преобразованиями, которые будут использованы для вычисления общего ключа:

$$B^a \bmod p = g^{ab} \bmod p;$$

$$A^b \bmod p = g^{ab} \bmod p.$$

В результате того на контроллере и рабочей станции вычисляются одинаковые ключи по формуле:

$$K = g^{ab} \bmod p.$$

После обмена ключами контроллер добавляет машину в список авторизованных пользователей, сохраняя её ключ, IP-, MAC-адреса.

Экспериментальные исследования

В качестве первоначального алгоритма шифрования был предложен метод перестановок. Однако метод имеет существенный недостаток, который не позволяет провести полноценную оценку и верификацию модели алгоритма на сетях Петри, а также учесть специфику форматов передачи кадров в компьютерных сетях. Алгоритм начинает работать верно только на сообщениях с размерностью от 1000 бит и более. Но в реальности в сети Ethernet могут поступать пакеты сообщений с размерностью от 512 до 12144 бит.

Сообщение должно разбиваться на блоки, длина которых равна ключу, затем символы в каждом блоке записываются в обратном порядке. Из-за размера ключа блоком являлось всё сообщение, соответственно расшифровать его можно любым ключом большего размера, полученным только по алгоритму Диффи-Хеллмана.

С учетом уязвимости и непрактичности предыдущего алгоритма был использован другой алгоритм для реализации блочного битового шифрования. Суть алгоритма шифрования заключается в следующем: сообщение разбивается на блоки, равные длине ключа, в каждом блоке выполняется шифрование. Шифрование заключается в операции «Побитовое или» текущего блока и ключа. Результат шифрования переводится в строковый формат UTF-8 и записывается в файл для модели. Результатом работы алгоритма является файл с зашифрованным сообщением, подходящим под формат аппарата сетей Петри. Файл содержит маркеры позиций на сетях Петри, в которых хранятся IP-адрес получателя и зашифрованные сообще-

ния (рисунок 4). Листинг программы, реализующей алгоритм блочного битового шифрования, представлен ниже.

```
import numpy as np
# Перевод строк в nparray и выполнение «Побитового или»
def xor_numpy(block, key):
  np_block = np.frombuffer(block, dtype='uint8')
  np_key = np.frombuffer(key, dtype='uint8')
  return (np_block ^ np_key).tobytes()

# Шифрование сообщения (ключ, сообщение)
def encrypt_code(key, text):
  blockSize = len(str(key))
  textSize = len(text)
  string = ''

  # Текст разбивается на блоки, равные ключу, каждый блок кодируется и добавляется в строку
  result = [text[i:i + blockSize] for i in range(0, textSize, blockSize)]
  for element in result:
    string += (xor_numpy(element.encode('utf-8'), str(key).encode('utf-8'))).decode("utf-8")
  return string
```

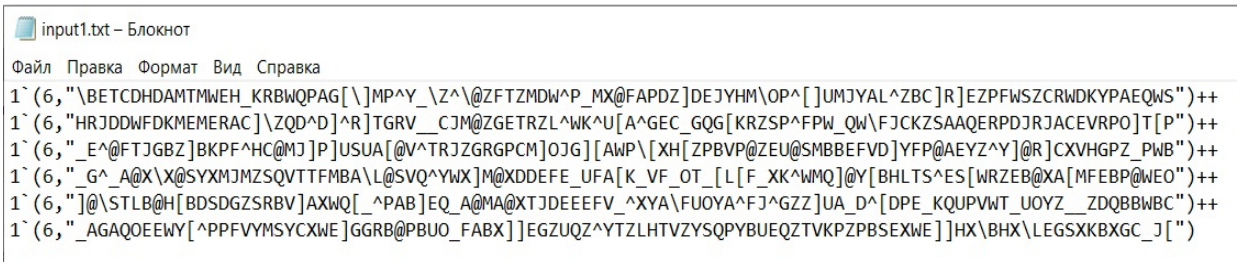

input1.txt – Блокнот
Файл Правка Формат Вид Справка
1`(6,"\BETCDHDAMTMWEH_KRBWQPAG[\]MP^Y_\Z^\@ZFTZMDW^P_MX@FAPDZ]DEJYHM\OP^[]UMJYAL^ZBC]R]EZPFWSZCRWDKYPAEQWS")++
1`(6,"HRJDDWFDKMEMERAC]\ZQD^D]^R]TGRV__CJM@ZGETRZL^WK^U[A^GEC_GQG[KRZSP^FPW_QW\FJCKZSAAQERPDJRJACEVRPO]T[P")++
1`(6,"_E^@FTJGBZ]BKPF^HC@MJ]P]USUA[@V^TRJZGRGPCM]OJG][AWP\[XH[ZPBVP@ZEU@SMBBEFVD]YFP@AEYZ^Y]@R]CXVHGPZ_PWB")++
1`(6,"_G^_A@X\X@SYXMJMZSQVTTFMBA\L@SVQ^YWX]M@XDDEFE_UFA[K_VF_OT_[L[F_XK^WMQ]@Y[BHLTS^ES[WRZEB@XA[MFEBP@WEO")++
1`(6,"]@\STLB@H[BDSDGZSRBV]AXWQ[_^PAB]EQ_A@MA@XTJDEEEFV_^XYA\FUOYA^FJ^GZZ]UA_D^[DPE_KQUPVWT_UOYZ__ZDQBBWBC")++
1`(6,"_AGAQOEEWY[^PPFVYMSYCXWE]GGRB@PBUO_FABX]]EGZUQZ^YTZLHTVZYSQPYBUEQZTVKPZPBSEXWE]]HX\BHX\LEGSXKBXGC_J[")
```

**Рисунок 4 – Пример зашифрованных сообщений**
**Figure 4 – Example of crypt message**

Для оценки быстродействия шифрования данных алгоритма было выбрано имитационное моделирование. Была разработана модель на основе сетей Петри в свободно распространяемом пакете CPN Tools, который наилучшим образом подходит для верификации алгоритмов, протоколов в компьютерных сетях [11]. Верхний уровень модели представлен на рисунке 5.

Подсеть «AUTH» выполняет аутентификацию. Справа на рисунке 5 находятся позиции контроллера и списка авторизованных машин, SDN controller и list of authorized соответственно. В сеть включены три рабочие станции (подсети PC1, PC2 и атакующая станция HACK).

В сети представлены подсеть авторизации, подсеть маршрутизации («out» и «routing»), подсеть для отправки сообщений «sendMsg», подсеть каждого компьютера. Подсеть авторизации нужна для того, чтобы рабочая станция могла получить ключ от контроллера. Подсеть компьютера нужна для того, чтобы отправить контроллеру свои данные и высчитать часть ключа. В подсети маршрутизации происходят дешифрование сообщения ключом отправителя и заново шифрование ключом получателя. Рабочая станция отправляет свой IP-адрес для получения всех сообщений и получает сообщения по IP-адресу.

Пример подсети отправки сообщений приведен на рисунке 6. Из полученного файла разработанная программа считывает набор сообщений и IP-адреса и через очередь обрабатывает данные сообщения.
```

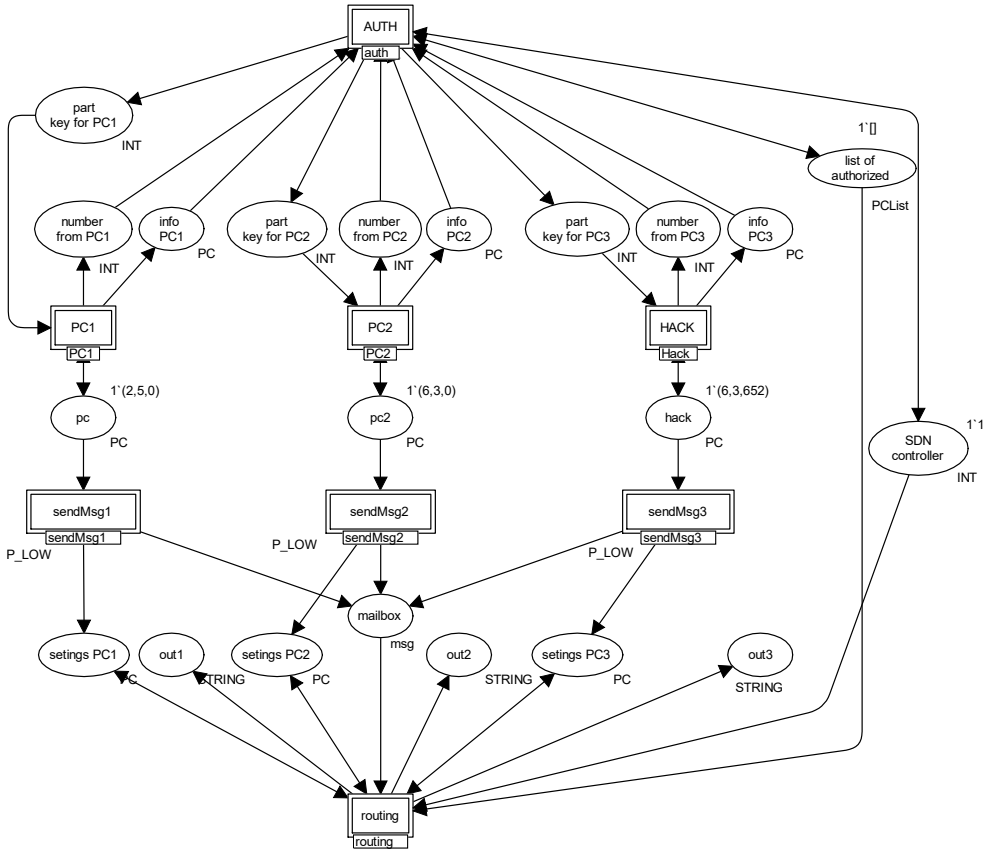


Рисунок 5 – Верхний уровень модели
Figure 5 – Top level of model

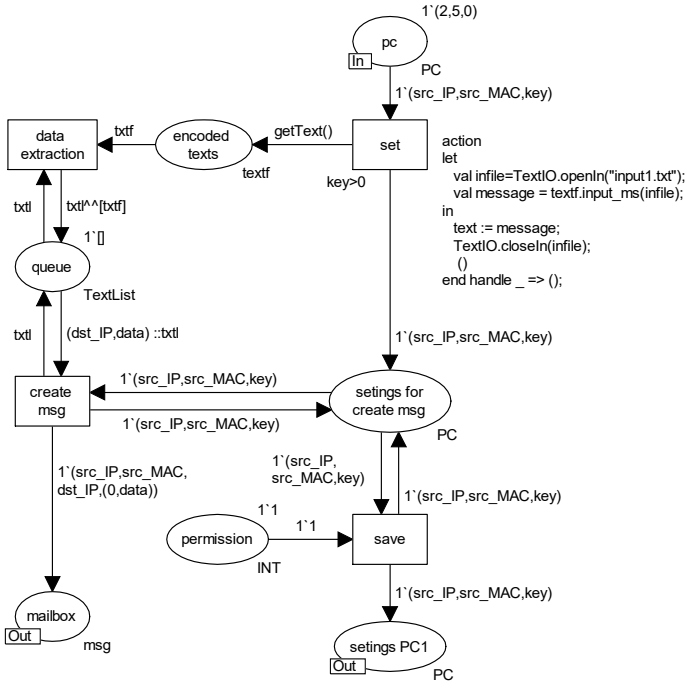


Рисунок 6 – Отправка сообщений
Figure 6 – Sending messages

Далее сообщение передается в виде кортежа, состоящего из полей IP-адреса отправителя, MAC-адреса отправителя, IP-адреса получателя, самого сообщения. Данный кортеж направляется в почтовый ящик.

Кроме проведения этапа верификации исследуемого алгоритма производилась оценка его эффективности на примере скорости шифрования данных разными симметричными алгоритмами. Для вычисления скорости шифрования данных экспериментальным путем производились замеры времени шифрования в программе и модели. В качестве ближайших аналогов алгоритма блочного битового шифрования AMDD были выбраны наиболее известные алгоритмы, такие как DES (Data Encryption Standard), ГОСТ 28147-89, AES (Advanced Encryption Standard) (рисунок 7). В имитационную модель специально был встроен узел злоумышленника, который не являлся авторизированным узлом в ПКС за счет аутентификации, не смог получать расшифрованную передачу трафика по сети. Авторизированные узлы успешно зашифровывали и расшифровали сообщения в модели с использованием аутентификации и побитного шифрования.

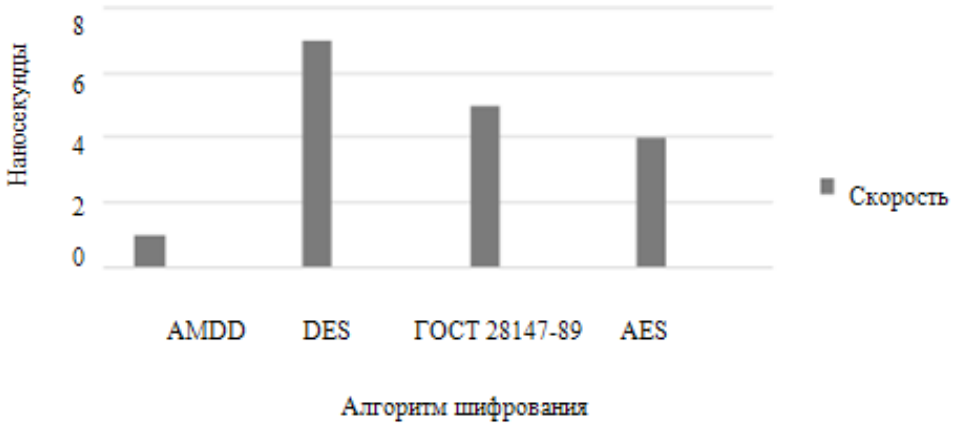


Рисунок 7 – Сравнительная диаграмма алгоритмов по скорости шифрования
Figure 7 – Comparative diagram of algorithms by encryption speed

Заключение

При исследовании темы аутентификации и шифрования данных на стороне контроллера ПКС были выявлены проблема, важность и актуальность данного направления. Была осуществлена передача трафика в ПКС с использованием аутентификации и побитного шифрования на стороне контроллера для снижения атак «человек посередине». Добавление аутентификации и побитного симметричного шифрования ключом, полученным по алгоритму Диффи-Хеллмана, позволило обеспечить безопасную передачу для атак типа «человек посередине» в ПКС, вычислять ключи, не позволяя передавать по сети в открытом виде, а также повысить скорость шифрования трафика в ПКС.

Библиографический список

1. **Shalimov A., Zuikov D., Zimarina D., Pashkov V., Smeliansky R.** Advanced study of SDN/OpenFlow controllers // Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia. ACM. Moscow. 2013. pp. 1–6. DOI: 10.1145/2556610.2556621.

2. **Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N.** Maturing of OpenFlow and Software-Defined Networking Through Deployments // Computer Networks. 2014. vol. 61. pp. 151-175.

3. **Jawad T.A., Mahmood A.N., Hameed A.N.** Detecting man-in-the-middle attacks via hybrid quantum classical protocol in software-defined networks // Indonesian Journal of Electrical Engineering and Computer Science. 2023. Vol. 31. No. 1. pp. 205-211.

4. **Лихачев В.А.** Программно-конфигурируемые сети на основе протокола OpenFlow // Приволжский научный вестник. 2014. №. 3-1 (31). С. 18-21.

5. **Никульчев Е.В., Паяин С.В., Плужник Е.В.** Динамическое управление трафиком программно-конфигурируемых сетей в облачной инфраструктуре // Вестник Рязанского государственного радиотехнического университета. 2013. № 45. С. 54-57.

6. **Ушакова М.В., Ушаков Ю.А.** Исследование сети виртуальной инфраструктуры центра обработки данных с гибридной программно-конфигурируемой коммутацией // Вестник Рязанского госу-

дарственного радиотехнического университета. 2021. № 75. С. 34-43. DOI: 10.21667/1995-4565-2021-75-34-43.

7. **Никишин К.И.** Моделирование контроллера и верификация процесса передачи данных в программно-конфигурируемых сетях // Вестник Рязанского государственного радиотехнического университета. 2022. №. 80. С. 75 - 83. DOI - 10.21667/1995-4565-2022-80-75-83.

8. **Коляденко Ю.Ю., Лукинов И.Г.** Модель распределенных атак в программно-конфигурируемых сетях связи // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. 2017. Т. 17. №. 3. С. 34-43.

9. **Курочкин И.** Безопасность сетей SDN. Классификация атак // Современные информационные технологии и ИТ-образование. 2015. Т. 2. №. 11. С. 381-383.

10. **Пименова А.А., Никитин Д.Д., Никишин К.И.** Моделирование сценариев безопасности в программно-конфигурируемых сетях // Вестник Рязанского государственного радиотехнического университета. 2022. № 82. С. 60-72. DOI: 10.21667/1995-4565-2022-82-60-72.

11. **Никишин К.И.** Моделирование процесса передачи трафика в программно-конфигурируемых сетях // Вестник Рязанского государственного радиотехнического университета. 2022. № 81. С. 32-41. DOI: 10.21667/1995-4565-2022-81-32-41.UDC 621.396

UDC 004.72

TRAFFIC TRANSMISSION IN SOFTWARE-DEFINED NETWORKS USING AUTHENTICATION AND BITWISE ENCRYPTION

K. I. Nikishin, Ph.D. (in technical sciences), associate professor, department of computer science, PSU, Penza, Russia;
orcid.org/0000-0001-7966-7833, e-mail: nkipnz@mail.ru
M. A. Mitrokhin, Dr. in technical sciences, associate professor, department of computer science, Head of the department, PSU, Penza, Russia;
orcid.org/0000-0001-6719-4610, e-mail: vt@pnzgu.ru
K. A. Artemov, student, department of computer science, PSU, Penza, Russia;
orcid.org/ 0009-0008-9482-6271, e-mail: artk_007@mail.ru
D. D. Grigorieva, student, department of computer science, PSU, Penza, Russia;
orcid.org/ 0009-0006-9211-7012, e-mail: g.darya.work@gmail.com
E. A. Merenyasheva, student, department of computer science, PSU, Penza, Russia;
orcid.org/0000-0003-3744-4521, e-mail: lizamerenyasheva@gmail.ru

Traffic and criteria management in software-defined networks (SDN) is performed by a controller in conjunction with a switch that operates using the OpenFlow protocol. The OpenFlow protocol is the main part of SDN. Thus, the OpenFlow protocol successfully allows you to cope with the increased requirements for user traffic, priority. The relevance of the research is that there are 9 types of attacks on SDN, most of them are security-related and solved by encryption, the rest are traffic problems. ***The aim of the research*** *is to transfer traffic in SDN using authentication and bitwise encryption on controller side to reduce man-in-the-middle attacks. The algorithm uses bitwise symmetric encryption with a key obtained by Diffie-Hellman algorithm. To research the algorithm, a simulation model was developed, including the use of the algorithm on the apparatus of Petri nets. Simulation results showed that the proposed traffic transfer in SDN using authentication and bitwise encryption on controller side made it possible to ensure secure transmission for man-in-the-middle attacks, calculate keys without allowing transmission over the network in open form, and also increase the speed of traffic encryption in SDN.*

Keywords*: software-defined networks, controller, switch, OpenFlow, Petri nets, CPN Tools, symmetric encryption, Diffie-Hellman algorithm.*

DOI: 10.21667/1995-4565-2023-86-63-72

References

1. **Shalimov A., Zuikov D., Zimarina D., Pashkov V., Smeliansky R.** Advanced study of SDN/ OpenFlow controllers. *Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia*. ACM. Moscow. 2013. pp. 1-6. DOI: 10.1145/2556610.2556621.

2. **Kobayashi M., Seetharaman S., Parulkar G., Appenzeller G., Little J., Van Reijendam J., McKeown N.** Maturing of OpenFlow and Software-Defined Networking Through Deployments. *Computer Networks*. 2014, vol. 61. pp. 151-175.

3. **Jawad T.A., Mahmood A.N., Hameed A.N.** Detecting man-in-the-middle attacks via hybrid quantum classical protocol in software-defined networks. *Indonesian Journal of Electrical Engineering and Computer Science*. 2023, vol. 31, no. 1, pp. 205-211.

4. **Lihachev V.A.** Programmno-konfiguriruemye seti na osnove protokola OpenFLOW. *Privolzhskij nauchnyj vestnik*. 2014, no. 3-1 (31), pp. 18-21 (in Russian).

5. **Nikul'chev E.V., Pajain S.V., Pluzhnik E.V.** Dinamicheskoe upravlenie trafikom programmno-konfiguriruemyh setej v oblachnoj infrastructure. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2013, no. 3 (45), pp. 54-57, (in Russian).

6. **Ushakova M.V., Ushakov Ju. A.** Issledovanie seti virtual'noj infrastruktury centra obrabotki dannyh s gibridnoj programmno-konfiguriruemoj kommutaciej. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta.* 2021, no. 75, pp. 34-43. (in Russian). DOI: 10.21667/1995-4565-2021-75-34-43.

7. **Nikishin K.I.** Modelirovanie kontrollera i verifikacija processa peredachi dannyh v programmno-konfiguriruemyh setjah . *Vestnik Rjazanskogo gosudarstvennogo radiotehnicheskogo universiteta*. 2022, no. 80, pp. 75-83. (in Russian). DOI: 10.21667/1995-4565-2022-80-75-83.

8. **Kolyadenko YU. YU., Lukinov I. G.** Model' raspredelennyh atak v programmno-konfiguriruemyh setyah svyazi. *Vestnik YUzhno-Ural'skogo gosudarstvennogo universiteta. Seriya: Komp'yuternye tekhnologii, upravlenie, radioelektronika*. 2017. vol. 17, no. 3, pp. 34-43 (in Russian).

9. **Kurochkin I.** Bezopasnost' setej SDN. Klassifikaciya atak. *Sovremennye informacionnye tekhnologii i IT-obrazovanie*. 2015, vol. 2, no. 11, pp. 381-383 (in Russian).

10. **Pimenova A.A., Nikitin D.D., Nikishin K.I.** Modelirovanie scenariev bezopasnosti v programmno-konfiguriruemyh setjah. *Vestnik Ryazanskogo gosudarstvennogo radiotekhnicheskogo universiteta*. 2022, no. 82, pp. 60-72. (in Russian). DOI: 10.21667/1995-4565-2022-82-60-72.

11. **Nikishin K.I.** Modelirovanie processa peredachi trafika v programmno-konfiguriruemyh setjah. *Vestnik Ryazanskogo gosudarstvennogo radiotekh-nicheskogo universiteta*. 2022, no. 81, pp. 32-41. (in Russian). DOI: 10.21667/1995-4565-2022-81-32-41.