УДК 004.72 Научная статья

https://doi.org/10.32603/2071-8985-2024-17-1-68-77

Моделирование алгоритма аутентификации трафика на стороне контроллера в программно-конфигурируемой сети

К. И. Никишин✉, М. А. Митрохин

Пензенский государственный университет,
кафедра «Вычислительная техника», Пенза, Россия

✉ nkipnz@mail.ru

Аннотация. Для обеспечения эффективной и оптимальной передачи разнородного пользовательского трафика в сети необходимо учитывать такие условия, как задержка кадра, быстродействие, диапазон частот и переполнение очереди в коммутаторе Ethernet. В программно-конфигурируемых сетях (ПКС) эти условия могут быть управляемыми, что отличает ПКС от традиционной компьютерной сети. ПКС – новая парадигма в области сетевых телекоммуникаций. Использование сетей Петри при моделировании ПКС позволяет создавать формальные модели, которые могут быть проверены на корректность и эффективность до реализации в реальной сети. Также сети Петри позволяют представлять архитектуру ПКС в виде графа, что улучшает понимание и визуализацию работы ПКС. В данной статье представлено решение проблемы уязвимости ПКС с использованием инструментов CPN Tools. Авторами был предложен алгоритм аутентификации трафика на стороне контроллера в ПКС для обеспечения безопасности передачи трафика и устранения атак типа «человек посередине» в ПКС. Цель исследования состоит в моделировании предложенного алгоритма и его верификации на основе аппарата сетей Петри. В статье приведены и описаны все компоненты модели на сетях Петри, проведен сравнительный анализ алгоритмов по скорости шифрования и выявлена максимальная скорость шифрования трафика предложенным алгоритмом.

Ключевые слова: программно-конфигурируемые сети, контроллер, коммутатор, OpenFlow, сети Петри, CPN Tools, аутентификация трафика, алгоритм Диффи–Хеллмана

Для цитирования: Никишин К. И., Митрохин М. А. Моделирование алгоритма аутентификации трафика на стороне контроллера в программно-конфигурируемой сети // Изв. СПбГЭТУ «ЛЭТИ». 2024. Т. 17, № 1. С. 68–77. doi: 10.32603/2071-8985-2024-17-1-68-77.

Конфликт интересов. Авторы заявляют об отсутствии конфликта интересов.

Original article

Modelling of the Traffic Authentication Algorithm on a Controller Side in a Software Defined Network

K. I. Nikishin✉, M. A. Mitrokhin

Penza State University, Department of Computer Science, Penza, Russia

✉ nkipnz@mail.ru

Abstract. To ensure efficient and optimal transmission of heterogeneous user traffic in a network, it is necessary to take into account such conditions as frame delay, speed, frequency range, and queue overflow in the Ethernet switch. In software-defined networks (SDN), these conditions can be controlled, which distinguishes SDN from a conventional computer network. SDN is a new paradigm in the field of network telecommunications. Petri nets in SDN modeling can be used to create formal models, which can be checked for correctness and efficiency before being implemented in a real network. Petri nets are also helpful in representing the architecture of SDN in the form of a graph, which improves the understanding and visualization of the SDN work. In this paper, we present a solution to the problem of SDN vulnerability using CPN Tools. We develop an algorithm

for traffic authentication on the controller side in SDN to ensure the security of traffic transmission and eliminate man-in-the-middle attacks in SDN. The research aim consists in simulating the proposed algorithm and its verification based on the apparatus of Petri nets. All the components of the model on Petri nets are described. A comparative analysis of algorithms for encryption speed is carried out, and the maximum speed of traffic encryption by the proposed algorithm is revealed.

Keywords: software defined network, controller, switch, OpenFlow, Petri nets, CPN Tools, traffic authentication, Diffie–Hellman algorithm

For citation: Nikishin K. I., Mitrokhin M. A. Modelling of the Traffic Authentication Algorithm on a Controller Side in a Software Defined Network // LETI Transactions on Electrical Engineering & Computer Science. 2024. Vol. 17, no. 1. P. 68–77. doi: 10.32603/2071-8985-2024-17-1-68-77.

Conflict of interest. The authors declare no conflicts of interest.

Введение. Для обеспечения эффективной и оптимальной передачи разнородного пользовательского трафика в сети необходимо учитывать такие условия, как задержка кадра, быстродействие, диапазон частот и переполнение очереди в коммутаторе Ethernet. В программно-конфигурируемых сетях (ПКС) эти условия могут быть управляемыми, что отличает ПКС от традиционной компьютерной сети [1], [2].

Сегодня ПКС широко используются и продолжают распространяться по всему миру благодаря облачным сервисам и центрам обработки данных. Использование ПКС в таких сетях позволяет сократить издержки на сопровождение сети за счет централизации управления на программном контроллере. Это повышает динамическое и интеллектуальное управление сетевыми ресурсами в сети.

Основной протокол управления в ПКС – это OpenFlow. Протокол OpenFlow всегда отличался отсутствием шагов аутентификации, он не требует, чтобы контроллер аутентифицировал коммутаторы, и контроллеру не требуется разрешать коммутаторам доступ к себе. Поскольку это – явная уязвимость протокола, она может затронуть любую реализацию OpenFlow.

Существует и крайне важный недостаток OpenFlow – то, что конфигурация сети хранится в одном месте, а именно на персональном компьютере администратора.

Авторами был предложен алгоритм аутентификации и побитного шифрования трафика на стороне контроллера в ПКС для обеспечения безопасности передачи трафика и устранения атак типа «человек посередине» в ПКС [3], [4].

Цель исследования заключается в моделировании предложенного алгоритма и его верификации. Моделирование ПКС – это процесс описания сетевых функций, требуемых для работы ПКС, – перехвата и изменения трафика, управление пропускной способностью, маршрутизацией и др. В качестве математического аппарата для разработки имитационной модели был выбран аппарат сетей Петри и программный пакет CPN Tools для построения и исследования компьютерных сетей на основе сетей Петри.

Преимущество использования сетей Петри при моделировании ПКС состоит в возможности создания формальных моделей, которые могут быть проверены на корректность и эффективность до создания в реальной сети [5]. Это позволяет ускорить процесс разработки ПКС и снизить ошибки и затраты на реализацию. Кроме того, использование сетей Петри позволяет представлять архитектуру ПКС в виде графа [6], что улучшает понимание и визуализацию работы ПКС.

Моделирование алгоритма на основе сетей Петри. Для того чтобы описать работу модели на сетях Петри в CPN Tools, необходимо прежде всего описать цвета, используемые в [7]. Цвет KEY – это ключ, который вычисляется с помощью алгоритма Диффи–Хеллмана. Он принимает значение от 0 до 65536:

colset KEY = int with 0..65536;

Цвет myNum, который используется для генерации числа, полученного случайным образом, как на станции, так и на контроллере, представлен следующим образом:

colset myNum = int with 1..100;

Цвет JOB является кортежем, состоящим из полей типа myNum и IP-, MAC-адресов, представлен следующим образом:

colset JOB=product myNum*IP*MAC;

Цвет PC – это цвет, описывающий компьютер, состоящий из IP-, MAC-адресов и ключа. На начальном этапе ключ не определен. После авторизации заполняется вычисленным ключом:

colset PC = product IP*MAC*KEY;

Цвет PCList – список рабочих станций в ПКС, представлен следующим образом:

colset PCList = list PC;

Цвет textf – это набор из IP-адреса и самого текстового сообщения. Используется для информации, полученной из файла:

colset textf= product IP*STRING declare input_ms;

Цвет textfs – кортеж, состоящий из ключа и сообщения, используется в подсети маршрутизации, представлен следующим образом:

colset textfs= product KEY*STRING declare input_ms;

Цвет Kdata – кортеж, состоящий из ключа и сообщения, зашифрованного этим ключом, представлен следующим образом:

colset Kdata = product KEY*STRING;

Цвет msg – один из главных цветов, тоже кортеж, и содержит в себе IP- и MAC-адреса получателя, IP-адрес отправителя и сообщение, зашифрованное ключом:

colset msg = product IP*MAC*IP*Kdata;

Цвет TextList – список значений цвета textf, цвет предназначен для обработки очереди сообщений из файла, представлен следующим образом:

colset TextList = list textf;

Числа p и g – это числа, которые можно передавать в открытом доступе для алгоритма Диффи–Хеллмана:

val g = 13; g – это первообразный корень по модулю p, небольшое целое число.

val p = 65536; p – большое простое число, минимум 1024 бит.

Эти числа используются в алгоритме Диффи-Хеллмана.

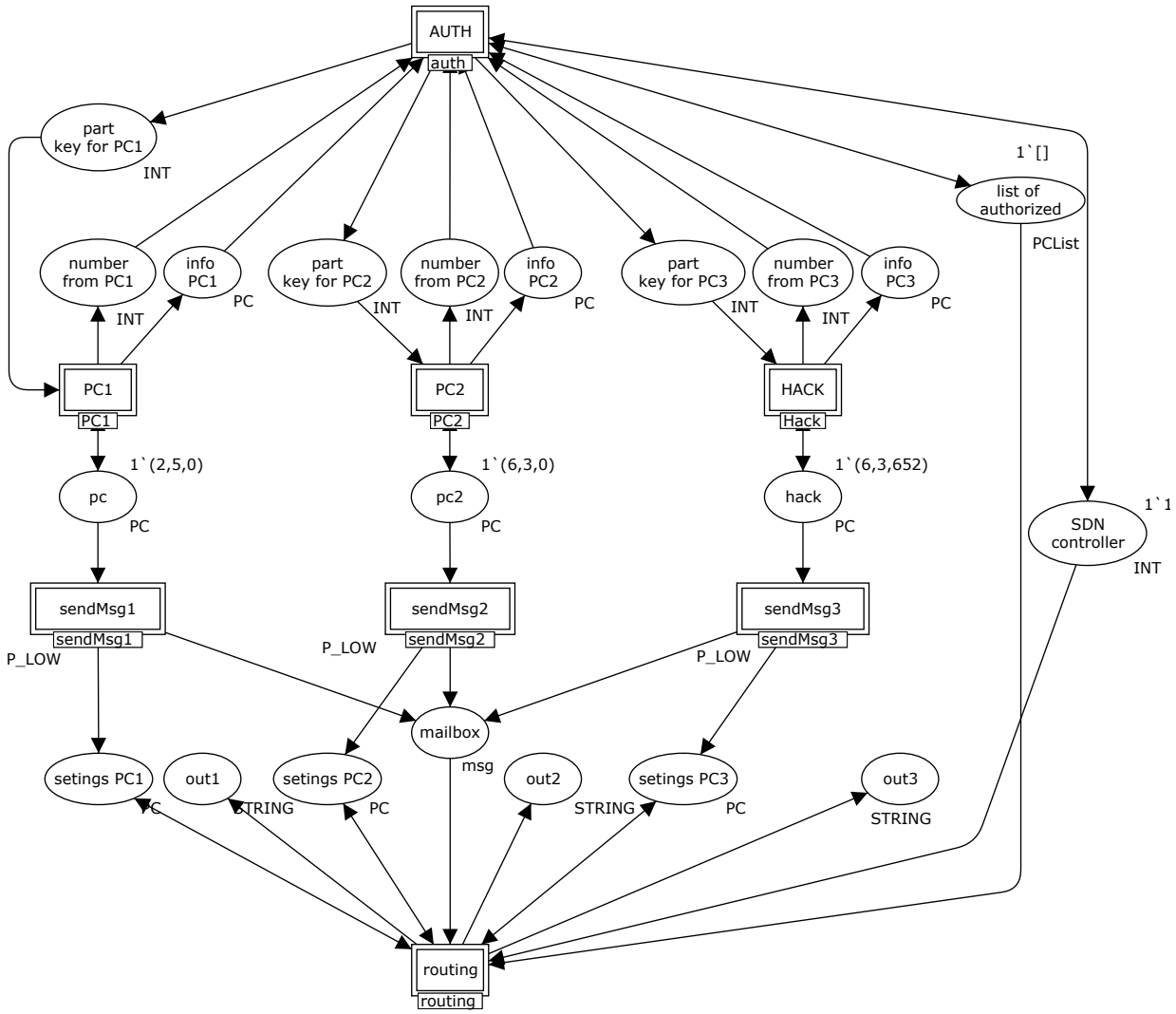

Рис. 1. Верхний уровень модели в программе CPN Tools
Fig. 1. Top level of the model in CPN Tools

На рис. 1 представлен общий вид сети. На верхнем уровне коммутатора и машин можно выделить следующие подсети:

• 3 машины (PC1, pc2, Hack);

• подсети отправки сообщения (sendMsg1, sendMsg2, sendMsg3);

• подсеть авторизации (AUTH);

• подсеть маршрутизации (routing).

У контроллера есть один маркер для выполнения любой операции (авторизация одной станции или обработка одного сообщения). Как только действие закончится, контроллер освободится и получит новый маркер.

Подсеть компьютера представлена на рис. 2. Входная позиция pc принимает состояние компьютера, хранит IP и MAC. На выходе подсеть хранит данные с полученным ключом. Выходная позиция (send part key) – расчет части ключа по алгоритму Диффи–Хеллмана и отправка этой части на контроллер. Выходная позиция (send info to Controller) нужна для отправки информации о рабочей станции (IP и MAC) и дальнейшего использования в подсети авторизации. Входная позиция (get number from controller) принимает вычисленную часть ключа от контроллера в компьютер.

Подсеть компьютера предназначена для вычисления ключа компьютера. В начальной позиции PC ключ компьютера равен нулю, позиция принимает IP- и MAC-адреса. Затем происходит срабатывание перехода Start getKey. Вначале подготавливается часть ключа для отправки ее на контроллер. Компьютер генерирует случайное число x, переходя в позицию get rand с помощью функции getNum().

Далее срабатывает переход start calc. Отправляется на контроллер информация о рабочей станции в виде маркера (src_IP, src_MAC,0). Маркер записывается в позицию Send info to Controller. По алгоритму Диффи–Хеллмана число x возводится в степень g и берется остаток от деления от числа p. Полученное число – это часть ключа, которая отправляется на контроллер. Вычисление происходит при записи маркера в позицию Send part key. Позиция info pc необходима для того, чтобы в дальнейшем вернуть на рабочую станцию ключ. Позиция send rand передает сгенерированное случайным образом число x.

При получении части ключа от контроллера через позицию get number from controller происходит дальнейшее вычисление ключа на переходе rand and numcont. В позицию calc KEY записывается ключ, который получается следующим образом: передается часть ключа от контроллера и сгенерированное случайным образом число x, затем часть ключа возводится в степень x и берется остаток от деления от числа p. На переходе return key реализован монитор, записывающий ключ в файл. Аналогично работают другие подсети рабочих станций.

На рис. 3 представлена подсеть авторизации. Данная подсеть имеет входные и выходные позиции: сам контроллер (SDN controller) и список авторизованных рабочих станций (list of authorized). У каждого сокета на входе имеется две позиции: часть ключа (get number from PC) и информация от рабочей станции (get info PC). Выходная позиция – часть ключа, вычисляемая контроллером (send part key for PC). Эти входные и выходные позиции дублируются для каждого сокета. Позиции включают в себя информацию о станции в виде маркера (src_IP, src_MAC,0), часть ключа от станции и контроллера.

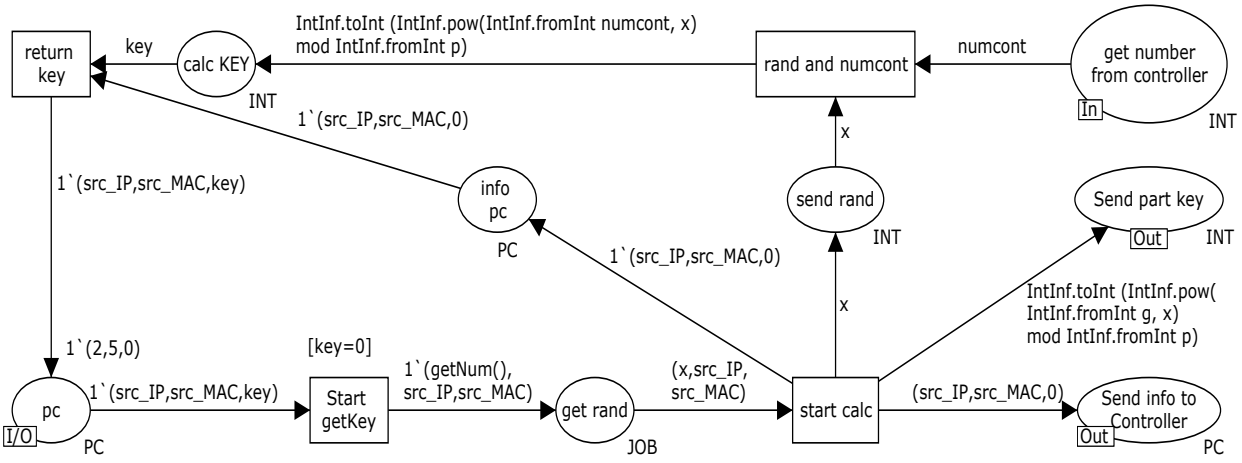

Рис. 2. Подсеть Петри рабочей станции (компьютера) в программе CPN Tools
Fig. 2. Petri subnet of the work station (computer) in CPN Tools

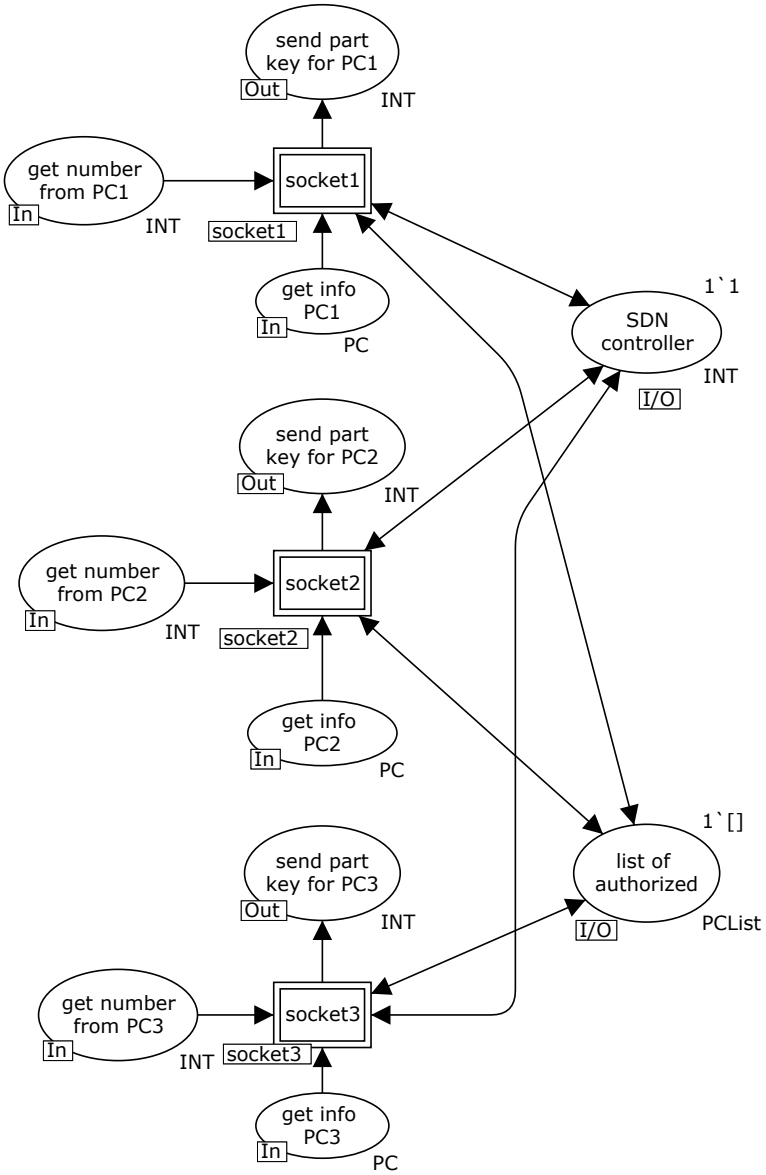

Рис. 3. Подсеть Петри авторизации ПКС в программе CPN Tools
Fig. 3. Petri subnet of SDN authorization in CPN Tools

Работа сокета представлена на рис. 4. У подсети сокета входной-выходной позицией служит контроллер (SDN controller) и список авторизованных рабочих станций (list of authorized). Входные позиции – часть ключа от рабочей станции (get number from PC) и информация о компьютере (get info PC). Выходная позиция – это часть ключа, вычисляемая в этом сокете (send part key for PC).

Работа сокета выглядит следующим образом: как только сокет начинает свою работу, он забирает маркер контроллера из позиции SDNcontroller. Происходит это на переходе start и передается маркер разрешения работы из позиции permission one auth. Сокет отрабатывает единожды для каждой станции.

Вначале вычисляется случайным образом число x при переходе в позицию get rand. Далее происходит срабатывание перехода send rand. С помощью сгенерированного случайным образом числа вычисляется часть ключа при переходе в позицию send part key for PC1 по алгоритму Диффи–Хеллмана.

Далее сокет получает часть ключа от рабочей станции через позицию get number from PC1. Срабатывает переход start calc и вычисляется ключ, переходя в позицию calc key.

В конце сокет получает информацию от станции в виде маркера (src_IP, src_MAC,0), далее срабатывает переход packaging. При переходе в позицию authorized PC к маркеру добавляется информация о вычисленном ключе. Затем срабатывает переход save PC, возвращая маркер в контроллер (позиция SDN controller). При переходе в позицию list of authorized выбирается список авторизованных машин, выполняется операция добавления

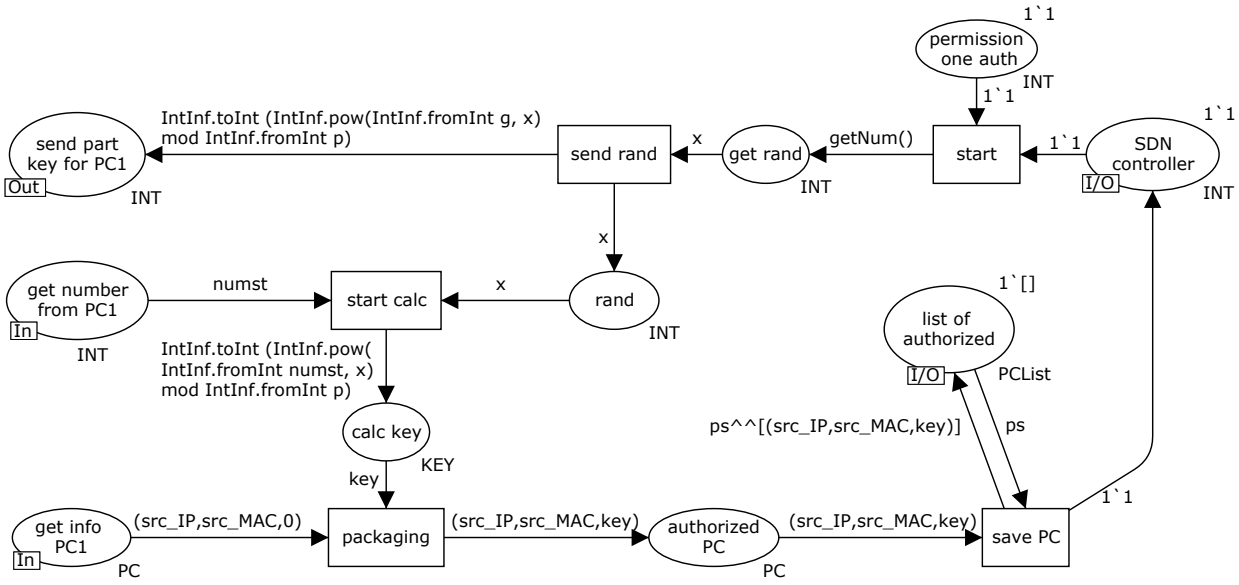

Рис. 4. Подсеть Петри работы сокета в программе CPN Tools
Fig. 4. Petri subnet of socket operation in CPN Tools

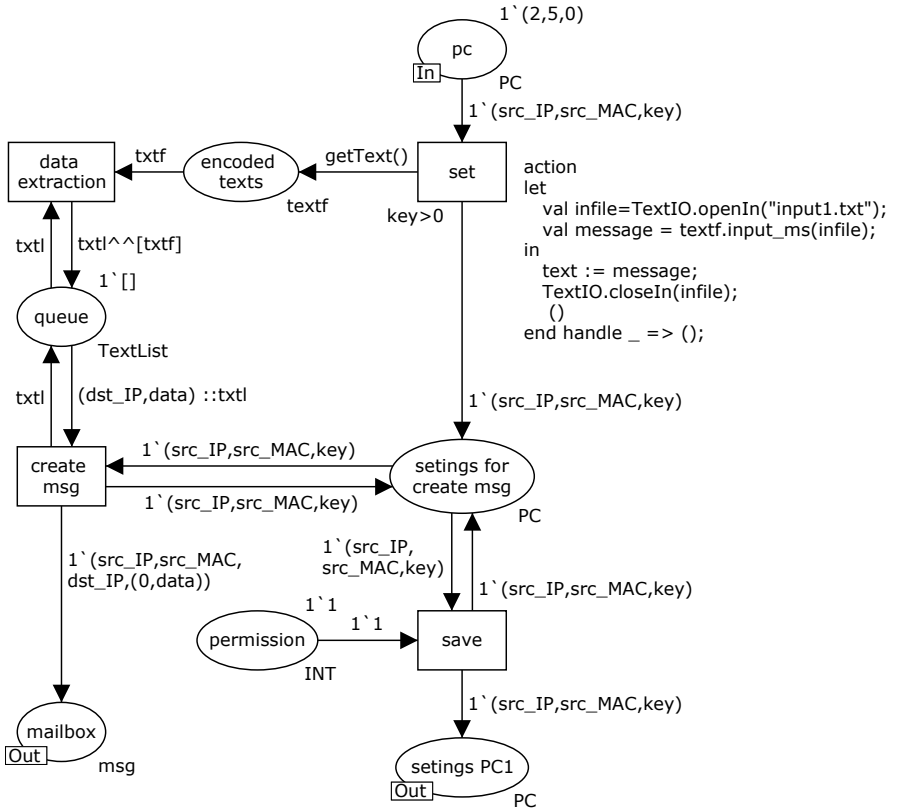

Рис. 5. Подсеть Петри отправки сообщений в программе CPN Tools
Fig. 5. Petri subnet of sending messages in CPN Tools

маркера в этот список ps^^[(src_IP, src_MAC,key)]. Аналогично происходит работа всех сокетов.

После того как компьютер авторизован, он может отправлять сообщения. Подсеть отправки сообщений представлена на рис. 5. У данной подсети входной позицией служит сам компьютер (pc), выходными позициями – почтовый ящик (mailbox) и настройки каждого компьютера (settings PC1).

Отправка сообщения происходит при условии, что ключ больше нуля. На переходе set читается файл закодированных сообщений. В позиции encoded texts считываются все сообщения. Переход data extraction, который берет список из позиции queue, дополняя его строкой из текста txtl^^ [txtf]. Таким образом формируется очередь.

При переходе из set в позицию settings for create msg сохранялись настройки станции, которые через переход save записывались в позицию settings PC1. Переход create msg работает при условии не пустой очереди, который считывает список из позиции queue, извлекая IP-адрес получателя и само сообщение (dst_IP,data)::txtl. В позицию mailbox записывается сообщение в виде маркера 1`(src_IP, src_MAC,dst_IP,(0,data)), IP и MAC отправителя, IP получателя и зашифрованное сообщение.

В подсети маршрутизации (рис. 6) входные позиции – это почтовый ящик (mailbox), список авторизованных машин (list of authorized) и контроллер (SDN controller); входные-выходные позиции – настойки рабочих станций (settings PC1, settings PC2, settings PC3). Выходной позицией служит полученное зашифрованное сообщение от каждой станции (out1, out2, out3).

Подсеть маршрутизации включает в себя процесс перекодирования сообщения и доставку получателю. Работа начинается с позиции mailbox и перехода Start transcoding. Если почтовый ящик не пуст и контроллер свободен, передается маркер старта и алгоритм начинается.

Передается одно сообщение из почтового ящика и перехода Start transcoding в позицию one msg. В позиции list of authorized получается список авторизованных рабочих станций благодаря переходу get List. Данный переход активируется

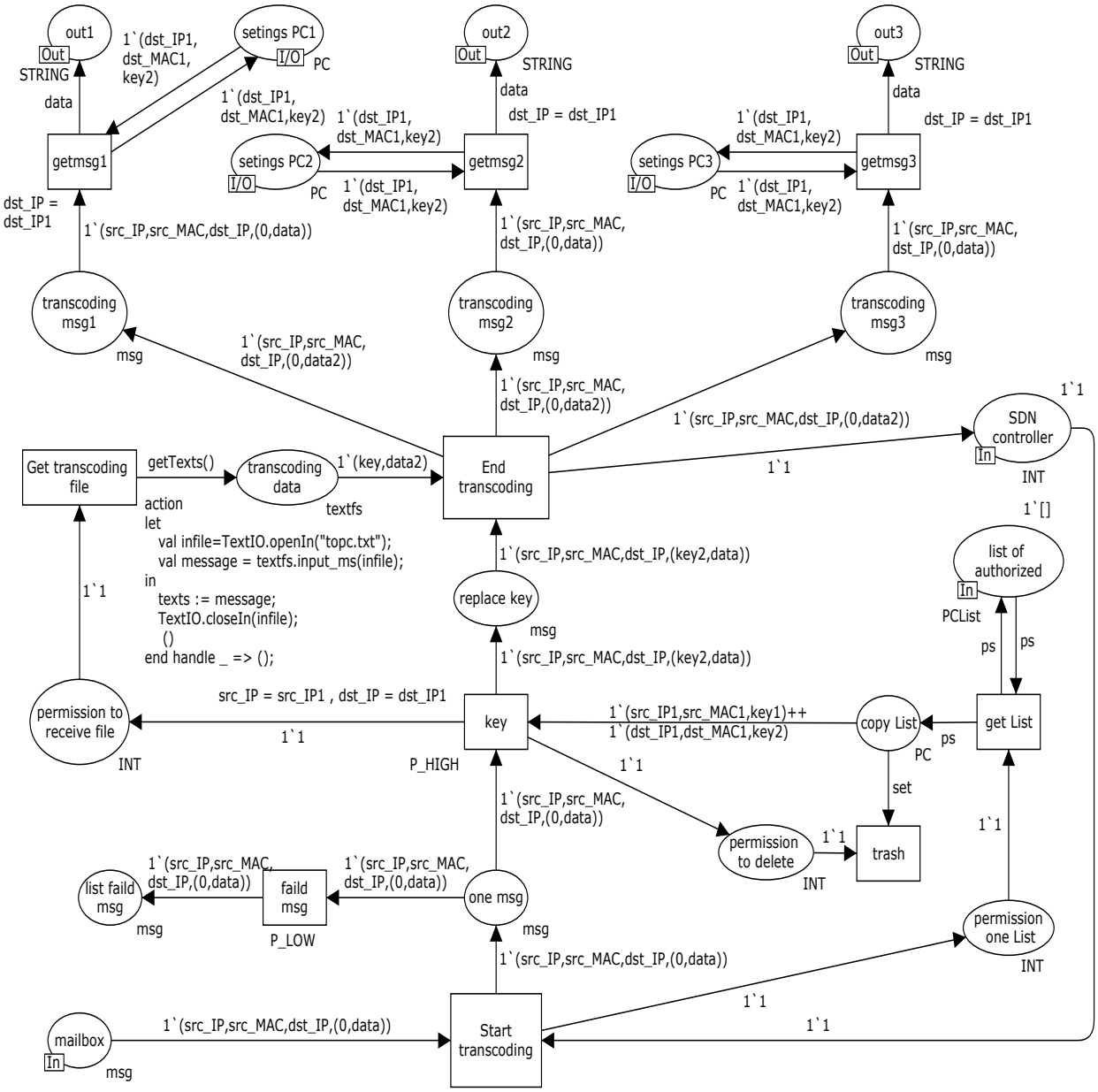

Рис. 6. Подсеть Петри маршрутизации в программе CPN Tools
Fig. 6. Petri subnet of routing in CPN Tools

по метке, полученной от позиции permission one List и перехода Start transcoding.

Из списка авторизованных рабочих станций будет взята связка, которая удовлетворяет условию для перехода key src_IP = src_IP1, dst_IP = dst_IP1, где src_IP и dst_IP берутся из сообщения позиции one msg. Таким образом выполняется проверка авторизации в сети отправителя и получателя и получения их ключей. Если проверка не проходит, то такие сообщения из позиции one msg через переход faild msg попадают в список удаленных сообщений (позиция list faild msg).

Если проверка прошла, срабатывает монитор на переходе key, в файл сохраняется ключ отправителя для дешифрации, ключ получателя для шифрации и само закодированное сообщение. Затем скриптом на высокоуровневом языке выполняется дешифрация ключом отправителя и шифрация ключом получателя, результат сохраняется в файл.

В позицию replace key передаются IP- и MAC-адреса отправителя, IP-адреса получателя. Сигнал из позиции permission to receive file разрешает чтение зашифрованного сообщения из файла. Сообщение сохраняется в позиции transcoding data, сформированный маркер направляется в каждую позицию transcoding msg N.

На переходе getmsg N проверяется условие совпадения IP-адреса получателя в сообщении и рабочей станции (IP-адрес получателя для рабочей станции берется из ее настройки позиции settings PC N). В позиции out сохраняется зашифрованное сообщение для рабочей станции.

На переходах getmsg1, getmsg2, getmsg3 реализованы мониторы, которые сохраняют полученное сообщение в файл. На этом работа подсети заканчивается. В итоге подсеть рабочей станции расшифровывает полученное сообщение своим ключом и скриптом, написанным на высокоуровневом языке программирования.

Результаты моделирования. В модели используется алгоритм Диффи–Хеллмана для получения ключа [8] и проводится побитовое шифрование AMDD (Advanced Method Data Division). Для создания ключа алгоритм Диффи–Хеллмана использует возведение в степень случайных и промежуточных значений. Подбор ключа требует бесконечного количества времени, потому что злоумышленнику приходится перебирать все возможные произведения, что может потребовать до сотен перемножений. Число возможных произведений достигает порядка 2^{128} вариантов при использовании современных эллиптических кривых. Однако быстрое возведение в степень требует только порядка log(N) операций умножения.

Алгоритм XOR-шифрования, рассмотренный в статье, отличается высокой скоростью шифрования по сравнению с известными алгоритмами блочного шифрования, использующими XOR-шифрование (длина блока и ключа равна 128). Сравнительная диаграмма представлена на рис. 7.

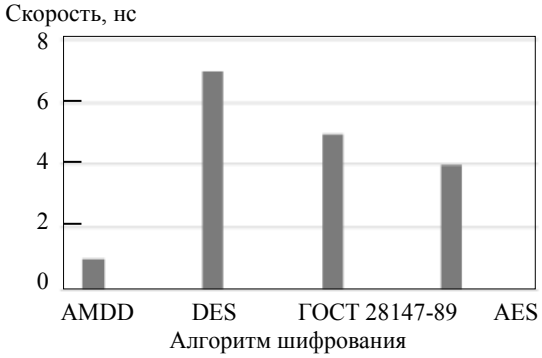

Рис. 7. Сравнительная диаграмма алгоритмов по скорости шифрования
Fig. 7. Comparative diagram of algorithms by encryption speed

Таким образом, за счет своей простоты алгоритм, представленный в статье и в модели, достигает наименьшего времени, необходимого для шифрования данных. XOR-шифр взламывается злоумышленником путем частотного анализа, однако вычисление ключа происходит по алгоритму Диффи–Хеллмана, устойчивого к атакам [8], [9].

Заключение. Приведены и описаны все компоненты модели на сетях Петри, проведен сравнительный анализ алгоритмов по скорости шифрования и выявлена максимальная скорость шифрования трафика предложенным алгоритмом.

Предложен алгоритм аутентификации трафика на стороне контроллера в ПКС для обеспечения безопасности передачи трафика и устранения атак типа «человек посередине» в ПКС, разработана модель на основе сетей Петри.

Список литературы

1. Advanced study of SDN/OpenFlow controllers / A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, R. Smeliansky // Proc. of the 9th Central & Eastern European Software Engin. Conf. in Russia. ACM. Moscow, 2013. P. 1–6. doi: 10.1145/2556610.2556621.

2. Никульчев Е. В., Паяин С. В., Плужник Е. В. Динамическое управление трафиком программно-

конфигурируемых сетей в облачной инфраструктуре // Вестн. Рязанского гос. радиотехнического ун-та. 2013. № 3 (45). С. 54–57.

3. Пименова А. А., Никитин Д. Д., Никишин К. И. Моделирование сценариев безопасности в программно-конфигурируемых сетях // Вестн. Рязанского гос. радиотехнического ун-та. 2022. № 82. С. 60–72. doi: 10.21667/1995-4565-2022-82-60-72.

4. Коляденко Ю. Ю., Лукинов И. Г. Модель распределенных атак в программно-конфигурируемых сетях связи // Вестн. Южно-Уральского гос. ун-та. Сер.: Компьютерные технологии, управление, радиоэлектроника. 2017. Т. 17. № 3. С. 34–43.

5. Никишин К. И. Моделирование процесса передачи трафика реального времени с использованием планировщика и функцией контроля доставки в программно-конфигурируемых сетях // Изв. СПбГЭТУ «ЛЭТИ». 2023. Т. 16. № 1. С. 53–65. doi: 10.32603/2071-8985-2023-16-1-53-65.

6. Никишин К. И. Метод комплексного контроля передачи трафика в программно-конфигурируемых сетях // Изв. СПбГЭТУ «ЛЭТИ». 2023. Т. 16, № 5. С. 49–58. doi: 10.32603/2071-8985-2023-16-5-49-58.

7. Multilevel modelling of coloured Petri nets // Intern. Workshop on Multi-Level Modelling MoDELS (Workshops) / A. Rodriguez, A. Rutle, F. Duran, L. M. Kristensen, F. Macias // CEUR Workshop Proc. Copenhagen, Denmark, 2018. P. 663–672.

8. Обзор А. А. Версия протокола Диффи–Хеллмана, использующая дополнительные скрытые множители // Прикладная дискретная математика. Приложение. 2017. № 10. С. 89–91.

9. Шурховецкий Г. Н. Криптостойкость алгоритмов шифрования // Молодая наука Сибири. 2018. № 2. С. 84–91.

Информация об авторах

Никишин Кирилл Игоревич – канд. техн. наук, доцент кафедры «Вычислительная техника». Пензенский государственный университет, ул. Красная, д. 40, Пенза, 440026, Россия.
E-mail: nkipnz@mail.ru
http://orcid.org/0000-0001-7966-7833

Митрохин Максим Александрович – д-р. техн. наук, доцент, зав. кафедрой «Вычислительная техника». Пензенский государственный университет, ул. Красная, д. 40, Пенза, 440026, Россия.
E-mail: vt@pnzgu.ru
http://orcid.org/0000-0001-6719-4610

References

1. Advanced study of SDN/OpenFlow controllers / A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, R. Smeliansky // Proc. of the 9th Central & Eastern European Software Engin. Conf. in Russia. ACM. Moscow, 2013. P. 1–6. doi: 10.1145/2556610.2556621.

2. Nikul'chev E. V., Pajain S. V., Pluzhnik E. V. Dinamicheskoe upravlenie trafikom programmno-konfiguriruemyh setej v oblachnoj infrastrukture // Vestn. Rjazanskogo gos. radiotehnicheskogo un-ta. 2013. № 3 (45). S. 54–57. (In Russ.).

3. Pimenova A. A., Nikitin D. D., Nikishin K. I. Modelirovanie scenariev bezopasnosti v programmno-konfiguriruemyh setjah // Vestn. Rjazanskogo gos. radiotehnicheskogo un-ta. 2022. № 82. S. 60–72. doi: 10.21667/1995-4565-2022-82-60-72. (In Russ.).

4. Koljadenko Ju. Ju., Lukinov I. G. Model' raspredelennyh atak v programmno-konfiguriruemyh setjah svjazi // Vestn. Juzhno-Ural'skogo gos. un-ta. Ser.: Komp'juternye tehnologii, upravlenie, radiojelektronika. 2017. T. 17, № 3. S. 34–43. (In Russ.).

5. Nikishin K. I. Modelirovanie processa peredachi trafika real'nogo vremeni s ispol'zovaniem planirovshhika i funkciej kontrolja dostavki v programmno-konfiguriruemyh setjah // Izv. SPbGJeTU «LJeTI». 2023. T. 16. № 1. S. 53–65. doi: 10.32603/2071-8985-2023-16-1-53-65. (In Russ.).

6. Nikishin K. I. Metod kompleksnogo kontrolja peredachi trafika v programmno-konfiguriruemyh setjah // Izv. SPbGJeTU «LJeTI». 2023. T. 16. № 5. S. 49–58. doi: 10.32603/2071-8985-2023-16-5-49-58. (In Russ.).

7. Multilevel modelling of coloured Petri nets // Intern. Workshop on Multi-Level Modelling MoDELS (Workshops) / A. Rodriguez, A. Rutle, F. Duran, L. M. Kristensen, F. Macias // CEUR Workshop Proc. Copenhagen, Denmark, 2018. P. 663–672.

8. Obzor A. A. Versija protokola Diffi–Hellmana, ispol'zujushhaja dopolnitel'nye skrytye mnozhiteli // Prikladnaja diskretnaja matematika. Prilozhenie. 2017. № 10. S. 89–91. (In Russ.).

9. Shurhoveckij G. N. Kriptostojkost' algoritmov shifrovanija // Molodaja nauka Sibiri. 2018. № 2. S. 84–91. (In Russ.).

Information about the authors

Kirill I. Nikishin – Cand. Sci. (Eng.), Associate Professor of the Department of Computer Science, Penza State University, Krasnaya St., 40, Penza, 440026, Russia.
E-mail: nkipnz@mail.ru
http://orcid.org/0000-0001-7966-7833

Maxim A. Mitrokhin – Dr Sci. (Eng.), Associate Professor, Head of the Department Computer Science, Penza State University, Krasnaya St., 40, Penza, 440026, Russia.
E-mail: vt@pnzgu.ru
http://orcid.org/0000-0001-6719-4610

Статья поступила в редакцию 12.09.2023; принята к публикации после рецензирования 21.11.2023; опубликована онлайн 30.01.2024.

Submitted 12.09.2023; accepted 21.11.2023; published online 30.01.2024.