ОБЕСПЕЧЕНИЕ БЕЗОПАСНОЙ МАРШРУТИЗАЦИИ В БЕСПРОВОДНЫХ САМООРГАНИЗУЮЩИХСЯ СЕТЯХ

***Волков М.С.А.*[1] *, Гордеев Э.Н.*[2]**

Цель статьи: *разработка алгоритма распределенной маршрутизации для обеспечения безопасности в беспроводных самоорганизующихся сетях в условиях атак злоумышленника на структуру сети.*

Методы: *применение теории алгоритмов, теории графов, дискретной оптимизации и эвристических подходов на основе результатов численных экспериментов.*

Полученный результат: *в работе предложен модифицированный распределенный алгоритм маршрутизации для самоорганизующихся сетей. Представленный алгоритм основан на распределенной версии алгоритма Дейкстры, предназначенной для обнаружения кратчайших путей, не содержащих петель, на графе в условиях изменения веса его ребер. Свобода от петель при этом достигается за счет хранения каждым узлом дополнительной таблицы, содержащей предпоследние узлы на кратчайших маршрутах ко всем узлам, что позволяет узлу выстроить дерево кратчайших маршрутов с корнем в нем самом. В модификации алгоритма эти таблицы используются узлами для проверки соответствия заявленного маршрута и обратного к нему, что позволяет распознавать и исключать из сети злоумышленника, осуществляющего атаки с целью разрушения правильного механизма маршрутизации. Эффективность предложенного алгоритма для защиты от атак на маршрутизацию, в частности, атаки типа «черная дыра», подтверждается результатами тестирования на программной модели.*

Ключевые слова: *распределенная маршрутизация, проактивные протоколы, адаптивный алгоритм, маршрутная петля, DOS-атака, «черная дыра», NP-полнота.*

Работа поддержана грантом РФФИ 20-01-00645

DOI:10.21681/2311-3456-2022-2-52-62

Введение

Самоорганизующиеся беспроводные сети – это тип сетей, не имеющий централизованного управления узлами, в котором каждый узел помимо действия в качестве пользователя сети, участвует в поиске и установлении маршрутов, а также пересылает пакеты узлов, находящихся вне зоны прямой беспроводной передачи друг друга. Узлы в данном типе сетей являются мобильными и могут изменять свое местоположение, подключаться и отключаться от сети в любое время, из-за чего поиск и обеспечение надежности маршрута от источника к получателю представляется важной проблемой.

Рассмотрим задачу обеспечения безопасной маршрутизации в беспроводных самоорганизующихся сетях. Управление в таких сетях реализуется без использования дополнительной сетевой инфраструктуры или какого-либо централизованного администрирования, а передача данных между узлами осуществляется с помощью ретрансляции через промежуточные узлы [1].

Интерес к проблеме маршрутизации в самоорганизующихся сетях привел к разработке ряда протоколов динамической маршрутизации [2, 3, 4]. Наибольшее распространение среди них получили проактивные (табличные) протоколы маршрутизации, в которых каждый узел сети строит и поддерживает таблицу маршрутизации ко всем остальным узлам [5]. Основной отличительной особенностью проактивных протоколов маршрутизации является выбор алгоритмов, которые лежат в их основе. Так большинство проактивных протоколов используют распределенные алгоритмы Беллмана-Форда [6] или Дейкстры [7]. Эти алгоритмы схожи с точки зрения способа обновления таблиц маршрутизации, в обоих случаях узлы периодически рассылают соседям информацию о своей наименьшей стоимости маршрутов к адресатам, а за-

1 Волков Мария Сабина Александровна, студентка кафедры ИУ-8 «Информационная безопасность» МГТУ им. Н.Э. Баумана, Москва, Россия. E-mail: sabina-volkoff@yandex.ru

2 Гордеев Эдуард Николаевич, доктор физико-математических наук, профессор кафедры ИУ-8 «Информационная безопасность» МГТУ им. Н.Э. Баумана, Москва, Россия. E-mail: werhorn@yandex.ru

тем используют информацию, полученную от соседей, для обновления стоимостей своих маршрутов.

Надежность и быстродействие данного типа алгоритмов послужили поводом к возникновению ряда работ, целью которых была разработка распределенных алгоритмов, не образующих петель маршрутизации при изменениях в топологии сети. В одной из ранних работ возникновение петель маршрутизации избегалось за счет установления строгих ограничений на порядок обновления маршрутной информации[3]. Для этого порядок обновления узлами маршрутов к узлу назначения устанавливался в соответствии с положением узлов в дереве с корнем в узле назначения, индуцированном кратчайшими маршрутами к нему. Развитием этого метода послужил алгоритм, основанный на том факте, что возникновение петель маршрутизации может быть вызвано только увеличением стоимости соединения между узлами или его отказом[4]. Этот алгоритм «замораживал» часть сети, во время распространения маршрутной информации. Также было предложено множество других решений [8, 9], таких как широковещательная передача последовательности узлов в кратчайших маршрутах[5] или отслеживание узлов с обоих концов маршрута[6]. Однако ни одно из вышеприведенных решений не обладает защитой от атак на маршрутизацию [10], поскольку все они основываются на доверии ко всем узлам сети.

Целью данной работы является организация защиты от атак на маршрутизацию в беспроводных самоорганизующихся сетях. Для достижения этой цели в работе предложена модификация одного из распределенных алгоритмов[7], свободного от петель маршрутизации, в основе которого лежит алгоритм Дейкстры. Предложенная здесь модификация использует особенности данного алгоритма для того, чтобы выявить изменения в маршрутной информации сети, которые невозможны при ее нормальной работе.

Статья организована следующим образом. В части 2 определена модель функционирования сети, а также представлено описание исходного алгоритма маршрутизации. В части 3 рассматриваются возможности воздействия злоумышленника на самоорганизующуюся сеть, использующую исходный алгоритм маршрутизации, а также приводится модификация исходного алгоритма, позволяющая обнаружить и устранить последствия атаки. Часть 4 посвящена результатам моделирования работы модифицированного алгоритма при различных воздействиях злоумышленника на сеть.

Результатом работы является модифицированный распределенный алгоритм маршрутизации, способный распознавать и исключать из сети злоумышленника, осуществляющего атаки на маршрутизацию.

Модель функционирования сети

Представим сеть в виде неориентированного графа $G=(V,E)$ с конечным множеством вершин $V=\{v_1 \dots v_n\}$, соответствующим узлам, и множеством ребер $E \subseteq V \times V$, соответствующим соединениям между узлами и обозначающимся $(v_i,v_j) \in E$. Вершинам графа приписаны координаты, из значений которых рассчитывается расстояние между ними по формуле: $d(v_i,v_j)=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$, где x_i, y_i и x_j, y_j – координаты вершин v_i и v_j соответственно. Ребра графа зададим функцией от расстояния между вершинами следующим образом:

$$E=\bigcup_{\substack{v_i,v_j \in V \\ v_i \neq v_j}} f(v_i,v_j), \quad (1)$$

$$f(v_i,v_j)=\begin{cases} \{(v_i,v_j)\}, & \text{если } (v_i,v_j) \le p \\ \varnothing, & \text{если } (v_i,v_j) > p \end{cases}, \quad (2)$$

где p – максимальное значение дальности прямой беспроводной передачи узла. То есть ребро на графе откладывается, если расстояние между вершинами, не превышает некоторого заранее заданного значения p, которое примем одинаковым для всех узлов. Каждому отложенному ребру (v_i,v_j) графа G присвоим значение стоимости $C(v_i,v_j)$, соответствующее некоторому показателю качества связи между узлами и зависящее от расстояния между узлами. Таким образом стоимость ребра может изменяться со временем, но всегда положительна.

Соседями узла A будем считать те узлы, с которыми у данного узла есть прямая связь, в модели это означает наличие общего ребра между вершинами.

3 P. Merlin, A. Segall. A failsafe distributed routing protocol // IEEE Transactions on Communications, 1979, vol. 27, no. 9, pp. 1280–1287.

4 J. Jaffe, F. Moss. A Responsive Distributed Routing Algorithm for Computer Networks // IEEE Transactions on Communications, 1982, vol. 30, no. 7, pp. 1758-1762, doi: 10.1109/TCOM.1982.1095632.

5 K.G. Shin, M.S. Chen. Performance analysis of distributed routing strategies free of ping pong-type looping // IEEE Transactions on Computers, 1987, vol. C- 36, pp. 129-137, doi: 10.1109/TC.1987.1676875.

6 S. W. Park, W. K. Tsai. Distributed routing algorithm for loop detection and resolution // MILCOM 91 - Conference record, 1991, pp. 6-10 vol.1, doi: 10.1109/MILCOM.1991.258191.

7 P. A. Humblet. Another adaptive distributed shortest path algorithm // IEEE Transactions on Communications, vol. 39, no. 6, pp. 995-1003, June 1991, doi: 10.1109/26.87189.

Стоимость маршрута между двумя вершинами примем равной сумме стоимостей всех ребер, входящих в этот маршрут. Расстояние между двумя вершинами определим как наименьшую стоимость маршрута между ними.

Кроме того, предположим, что протокол канального уровня гарантирует, что каждый узел знает состояние соединений со своими соседями и что все пакеты, передаваемые в сети, принимаются правильно и в надлежащей последовательности. Предполагается также, что к моменту времени начала работы алгоритма сеть уже существует некоторое время, все таблицы маршрутизации содержат актуальные данные, а новых изменений пока не происходило.

Исходный алгоритм представляет собой адаптивную распределенную версию алгоритма Дейкстры и основывается на построении каждым узлом своего дерева кратчайших маршрутов. Построение дерева осуществляется при помощи значений предпоследних узлов в маршрутах ко всем узлам назначения.

Рассмотрим узел A с соседями $B_1,\ldots,B_k$. Пусть DES – некоторый узел, доступный из A. На каждом таком узле A хранятся таблицы маршрутизации, содержащие пять видов записей:

- Запись $NT(A, DES, B_i)$ содержит минимальную стоимость маршрута от соседнего с A узла B_i до DES.
- Запись $NN(A,DES)$ хранит идентификатор соседнего с A узла, через который проходит маршрут наименьшей стоимости из A в DES.
- Запись $RT(A,DES)$ содержит стоимость маршрута из A в DES через узел $NN(A,DES)$, т.е. минимальную стоимость маршрута из A в DES.
- Запись $RTn(A,DES)$ хранит идентификатор предпоследнего узла, лежащего на кратчайшем пути от A в DES.
- Запись $NNn(A,DES,B_i)$ хранит идентификатор предпоследнего узла, лежащего на кратчайшем пути от соседнего с A узла B_i в DES.

Сообщения маршрутизации, отправляемые узлом A, состоят из одной или нескольких записей, каждая из которых представляет собой тройку формы $(A,RT(A,DES),RTn(A,DES))$.

Алгоритм состоит из двух основных частей: в первой части узел наблюдает за изменениями стоимостей соединений со своими соседями или получает сообщения об обновлении от соседей и сохраняет эти значения в свои таблицы маршрутизации. Во второй части каждый узел A в функции COMPUTE() фактически строит из всех своих записей $NT(A,DES,B_i)$ большое дерево со взвешенными ребрами, в котором один идентификатор узла может появляться много раз. Для этого узел A устанавливается качестве корня и соединяется ребрами со всеми своими соседями, от которых затем откладываются полученные от них деревья кратчайших маршрутов. Затем узлы этого большого дерева просматриваются по принципу поиска в ширину начиная с A в порядке неубывания стоимостей маршрутов от A. При этом узел DES добавляется в новое дерево RT, только если он еще не был включен и при этом является соседом A, или если его сосед K по направлению к A в большом дереве, $K = NNn(A,DES,B_i)$, уже был обработан. Дерево RT узел A принимает как свое новое дерево кратчайших маршрутов, и рассылает соседним узлам сообщения, содержащее записи вида $(A,RT(A,DES),RTn(A,DES))$ с изменениями, произошедшими в этом дереве по сравнению с его предыдущей версией.

Доказательство правильности алгоритма, а также оценка времени его выполнения приведены в исходной работе. Основанием выбора данного алгоритма послужила способность узлов при помощи своих таблиц маршрутизации выяснять полную последовательность узлов на кратчайших маршрутах своих соседей, что впоследствии используется при его модификации для обнаружения подделывания маршрутов.

Модификация исходного алгоритма при атаке на сеть

По сравнению с централизованными алгоритмами, задача обеспечения безопасности маршрутизации распределенных алгоритмов сталкивается с дополнительными трудностями, вызванными их особенностями [11]. Поскольку в распределенных алгоритмах маршрутизации данные о состоянии сети собираются со всех узлов, эти алгоритмы основываются на предположении о добросовестной работе всех узлов. Таким образом если узел заявляет, что он обладает кратчайшим маршрутом к узлу назначения, утверждение считается достоверным. Это создает уязвимость для различных атак типа «отказ в обслуживании» [12, 13], в частности, для атак типа «черная дыра» [14], когда злонамеренный узел рассылает поддельные кратчайшие маршруты наименьшей стоимости к узлам назначения, а затем, получая пакеты с данными для пересылки адресатам, отбрасывает их. Аналогично, если узел сообщает об изменениях в состоянии соединений, остальные узлы вынуждены обновить

свои таблицы маршрутизации. Достоверность таких изменений крайне сложно проверить, поскольку беспроводные мобильные узлы могут независимо перемещаться и в любой момент времени подключаться к сети или покидать ее. Из-за отсутствия физической защиты в самоорганизующихся сетях [15], подобные атаки на маршрутизацию представляют собой серьезную угрозу безопасности, поскольку злоумышленник может легко присоединиться к сети и начать рассылать ложную информацию о маршрутах, чтобы нарушить обычную связь.

Рассмотрим возможные действия злонамеренного узла BH, желающего распространить по сети проходящие через него ложные маршруты к как можно большему числу узлов сети. Чтобы составить представление о существующих в сети узлах и маршрутах к ним, этот узел должен некоторое время принимать участие в маршрутизации, поэтому предполагается, что у его соседей имеется достоверная на текущий момент информация о кратчайших маршрутах этого узла. В силу ограничения дальности передачи разослать поддельную маршрутную информацию по сети узел может, только отправив ее своим соседям.

Пусть A - некоторый сосед BH, DES – узел назначения поддельного маршрута. Чтобы A принял маршрут через BH в свою таблицу $RT(A,DES)$ и разослал этот маршрут другим узлам, должно выполняться условие: $RT(A,DES) > NT(A,DES,BH) + C(A,BH)$, где $NT(A,DES,BH)$ – заявляемая узлом BH стоимость маршрута в DES. В процедуре COMPUTE() узел добавляется в таблицу маршрутизации $RT(A)$ либо если он сосед A, либо если предыдущий узел на маршруте к нему уже был добавлен в таблицу маршрутизации A. Поэтому узлу A из записей $NNn(A,DES,BH)$ и $NTn(A,DES,BH)$ должны быть известны все узлы в кратчайших маршрутах BH, в том числе узел A может определить следующий после BH узел в данном маршруте. Таким образом у злоумышленника есть два варианта действий:

1. Объявить следующим узлом ложного маршрута некоторого подлинного соседа.
2. Объявить следующим узлом ложного маршрута узел, не являющийся в действительности его соседом.

Рассмотрим два этих случая подробнее и предложим алгоритмы обнаружения злоумышленника для каждого из них.

Пусть злонамеренный узел BH рассылает сообщения с поддельным маршрутом к узлу DES, объявляя следующим узлом этого маршрута своего соседа A. Тогда, поскольку сообщения отправляются по беспроводной сети, являющейся разделяемой средой, его получат все соседи BH в том числе и узел A. Обнаружив в записях этого сообщения маршрут к узлу DES стоимости W такой, что $W < RT(A,DES) + C(A,BH)$, узел A по ним обновляет значения $NNn(A,DES,BH)$ и $NTn(A,DES,BH)$, при помощи которых может определить следующий после BH узел в заявляемом им маршруте к DES. Если этим узлом оказывается сам A, то маршрут объявляется поддельным, поскольку при нормальной работе сети узел BH не может узнать об уменьшении стоимости маршрута, проходящего через A раньше самого A. Таким образом, при получении подобных сообщений, узел A отбрасывает их, блокирует BH и рассылает по сети сообщения об обнаружении злоумышленника. Остальные соседи BH, получив это сообщение, определяют, что оно действительно пришло от соседа BH, через которого был заявлен кратчайший маршрут, и вслед за A блокируют BH.

Дополнительно в функцию COMPUTE() вводится ограничение, при котором узел добавляет маршрут к соседу в свои таблицы маршрутизации только если он не содержит промежуточных узлов. Данное ограничение кажется разумным, поскольку маршрут через промежуточный узел не может быть короче прямого маршрута. Это позволяет сократить время восстановления сети, поскольку если все соседи BH являются соседями друг друга, то BH будет сразу заблокирован.

Представим теперь, что злоумышленник BH рассылает сообщения с поддельным маршрутом к узлу DES, объявляя следующим узлом маршрута некоторый узел P, не являющийся в действительности его соседом. Пусть W – объявленная узлом BH стоимость несуществующего ребра (BH,P). Тогда в отправленном BH сообщении должна содержаться запись (P,W,BH). Чтобы обнаружить подобную запись узел A, проверяет каждую запись вида (I,D,J) в полученном сообщении на выполнение условия $(J \neq NNn(A,DES,BH)) \wedge (J = BH) \wedge (D < p)$, где p – максимальное значение дальности прямой беспроводной передачи узла. Это условие, по сути, требует появления нового ребра, инцидентного отправителю сообщения в его дереве маршрутизации. Очевидно, что оно выполняется для записи (P,W,BH).

В действительности данное условие может выполняться также и в отсутствие атаки, например, при сближении двух узлов и появлении новой связи, поэтому узел A при обнаружении подобной записи выполня-

ет процедуру обработки сообщения из оригинального алгоритма, с той разницей, что в отправляемые узлом A сообщения с обновлениями добавляется новое поле подозрительных соединений NE, в которое он помещает запись (BH,P). Кроме того, A добавляет идентификатор BH в дополнительный список $SUSP$, хранящийся на этом узле.

Таким образом сообщения, содержащие маршрут через ребро (BH,P) распространяются до тех пор, пока не попадают в некоторый узел T, лежащий к узлу P ближе, чем к BH. Этот узел должен находиться примерно на середине настоящего кратчайшего маршрута между BH и P. Поскольку стоимость маршрута из T в P меньше, чем стоимость маршрута из него в BH, то и стоимость маршрута из T в P через BH будет больше, чем стоимость уже имеющегося у T маршрута в P. Тогда при обработке узлом T этого сообщения, маршрут в P через BH не будет записан в $RT(T,P)$, $NN(T,P)$ и $RTn(T,P)$, но останется в $NT(T,P,M)$ и $NTn(T,P,M)$, где M — сосед T, который прислал ему это сообщение. Из-за этого обновления с записью (BH,P) из узла T отправляться не будут. Таким образом узел T находится примерно на середине маршрута из P в BH и хранит в своих таблицах оба маршрута в P, объединение которых образует цикл на графе сети. Воспользуемся этим свойством чтобы установить достоверность существования ребра (BH,P). Если ребро (BH,P) действительно существует, то узлы I, для которых до появления ребра (P,BH) выполнялось условие:

$$RT(I,BH)-RT(I,P)>W, \quad (3)$$

должны будут изменить свои кратчайшие маршруты в BH, проложив их через ребро (P,BH). И высока вероятность, что, одним из этих узлов окажется следующий за T узел на маршруте из T в P, $S=NN(T,P)$. Тогда чтобы определить существование ребра (P,BH) узел T проверяет выполнение условия:

$$NN(S,BH)\neq P\wedge RT(S,BH)-RT(S,P)>W \quad (4)$$

Если оно выполняется, значит $NN(T,P)$ не изменил свой кратчайший маршрут в BH на более короткий маршрут, проходящий через P, следовательно, узел P не заявил о появлении ребра (P,BH) и этого ребра на самом деле нет. Чтобы реализовать данную проверку в алгоритме, в функцию COMPUTE() добавляются дополнительные условия. При выполнении COMPUTE() на любом узле I всякий раз, когда некоторый узел J, из поддерева соседнего с I узла B, не добавляется в дерево кратчайших маршрутов RT, происходит проверка наличия записи $(NNn(I,J,B),J)$ в списке NE сообщения, вызвавшего выполнение функции COMPUTE(). Если такая запись существует, узел I проверяет выполнение условия:

$$\begin{aligned} &NNn(I,J,B)\neq RTn(I,J)\wedge(RT(I,J)<NT(I,J,B)+\\ &+C(I,B))\wedge\Big(NT(I,J,NN(I,J))+\Big(NT(I,J,B)-\\ &-NT(I\ NNn(I,J,B)),B)\Big)<\\ &<NT(I,NNn(I,J,B),NN(I,J))) \end{aligned} \quad (5)$$

Это условие эквивалентно условию (4) того, что ребро $(NNn(I,J,B),J)$ не используется узлом $NN(I,J)$ в маршруте к $NNn(I,J,B)$, при том, что предполагаемый маршрут через это ребро имеет меньшую стоимость, чем используемый узлом $NN(I,J)$. Это, по существу, говорит о подделывании узлом $NNn(I,J,B)$ ребра $(NNn(I,J,B),J)$.

При выполнении проверки (5) может оказаться, что ребро $(NNn(I,J,B),J)$ действительно существует, но узел $NN(I,J)$ еще не обновил маршрут к $NNn(I,J,B)$ или эти обновления еще не были обработаны узлом I. Поэтому при выполнении приведенного выше условия, узел I не сразу объявляет о подделывании маршрута, а добавляет запись $(NNn(I,J,B),J,B,K)$ в новый список $VERIF$. K в этой записи обозначает счетчик, по истечении которого узел I заново выполняет проверку условия подделывания маршрута. Если к тому моменту условие до сих пор выполняется, этот узел рассылает по сети сообщения о том, что узел $NNn(I,J,B)$ является нарушителем. Когда это сообщение доходит до соседей злоумышленника, они проверяют свои списки $SUSP$ и, обнаружив там идентификатор этого узла, блокируют его.

На рисунках 1-6 приведены основные процедуры модифицированного распределенного алгоритма маршрутизации, красным цветом выделены блоки, введенные в модификации.

Рис. 1. Обработка узлом I сведений об изменении ребра (I,M).

Результаты работы модифицированного алгоритма

Приведенный в статье алгоритм с добавленными модификациями был реализован на языке Python 3. На тестовом программном макете была смоделирована работа сети из 25 узлов, распределенных по площади 100 x 100. Начальное расположение узлов задавалось случайно. Дальность прямой беспроводной передачи узлов одинакова и равна $p = 35$.

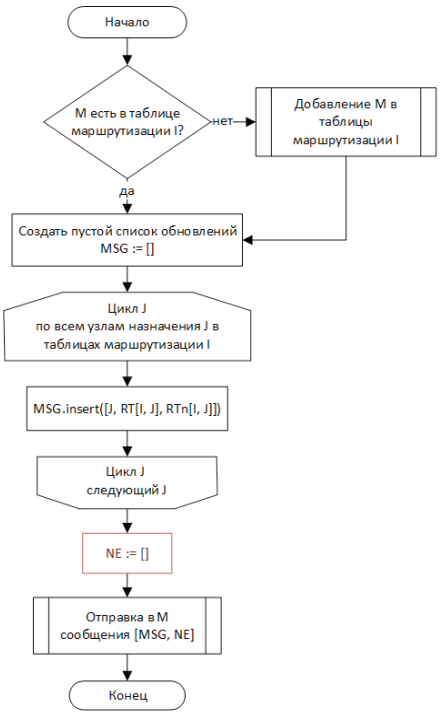

Рис. 2. Обработка узлом I сведений об установлении соединения с узлом M

Для определения распознавания атак с подделкой маршрутов было проведено по 100 испытаний при отправке злоумышленником поддельных маршрутов двух типов:

1. Злоумышленник выбирает следующим узлом ложного маршрута некоторого действительного соседа.
2. Злоумышленник выбирает следующим узлом ложного маршрута другой случайный узел, не являющийся его соседом.

Каждый тип испытаний проводился при случайном выборе одного узла в качестве злоумышленника и нескольких типах его стратегий. В зависимости от заданной конфигурации, злонамеренный узел мог рассылать поддельные маршруты как ко всем узлам сети, так и лишь к определенному заранее заданному кругу узлов, например только к тем узлам, стоимость кратчайших маршрутов до которых находилась в заданных пределах.

Испытание считалось успешно пройденным, если узлам удавалось обнаружить искажения в маршрутной информации и правильно определить злонамеренный узел, и, если при этом ни один из других узлов не оказывался ложно обвиненным. Если злоумышленник не был определен, атака считалась нераспознанной. Если же помимо злоумышленника был обвинен другой узел, срабатывание алгоритма считалось ложным. Результаты испытаний по обнаружению атак с подделыванием маршрутов приведены в таблице 1.

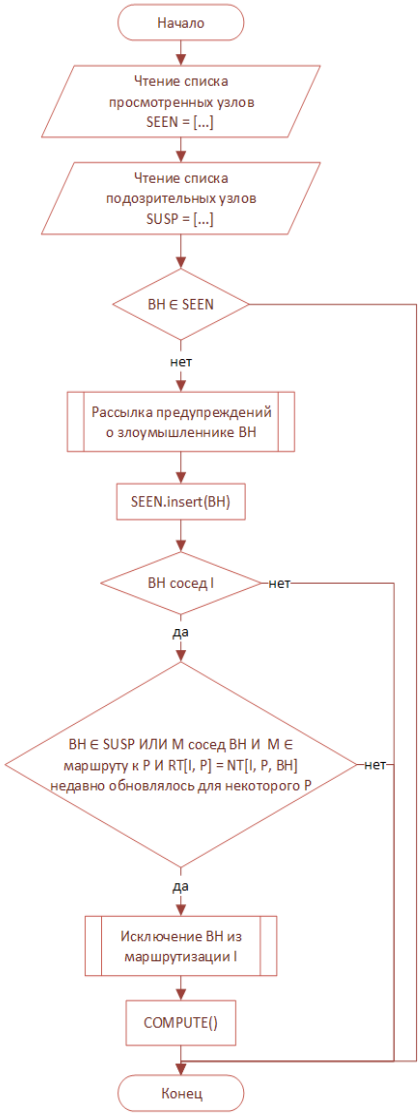

Рис. 3. Получение узлом I сообщения маршрутизации $[MSG, NE]$ от узла M

Кроме этого, для проверки ложных срабатываний алгоритма в случае изменения топологии сети при отсутствии атаки, были обработаны случаи добавления,

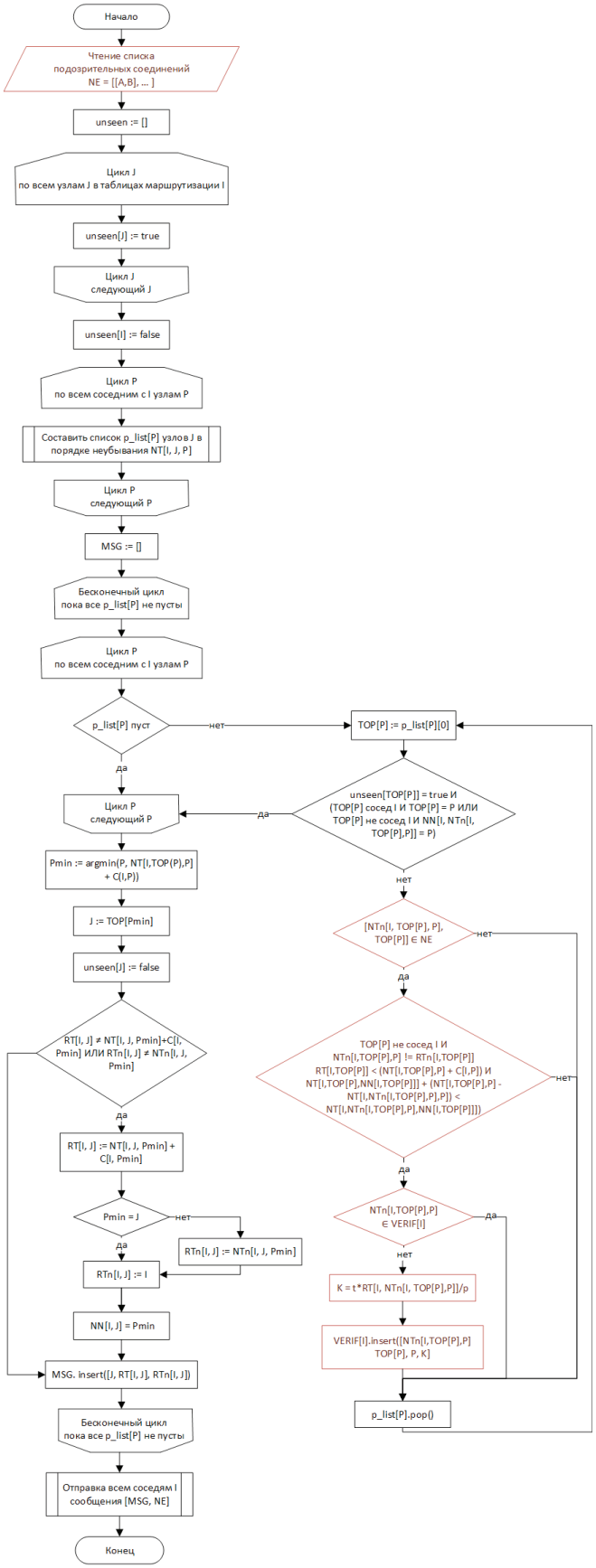

Рис. 4. Выполнение процедуры COMPUTE() на узле I

Таблица 1

Результаты испытаний по обнаружению атак

Тип испытаний	Число правильных определений злоумышленника	Число нераспознанных атак	Число ложных срабатываний
Отправка поддельных маршрутов первого типа	97	3	0
Отправка поддельных маршрутов второго типа	100	0	0

Таблица 2

Результаты проверки алгоритма при нормальной работе сети

Тип испытаний	Число успешных испытаний	Число ложных срабатываний
Уменьшение стоимостей ребер	25	0
Увеличение стоимостей ребер	25	0
Добавление ребер	25	0
Удаление ребер	25	0

удаления, уменьшения и увеличения стоимостей ребер, в том числе множественные. Для каждого типа изменений было проведено 25 испытаний на том же тестовом стенде. Результаты приведены в таблице 2.

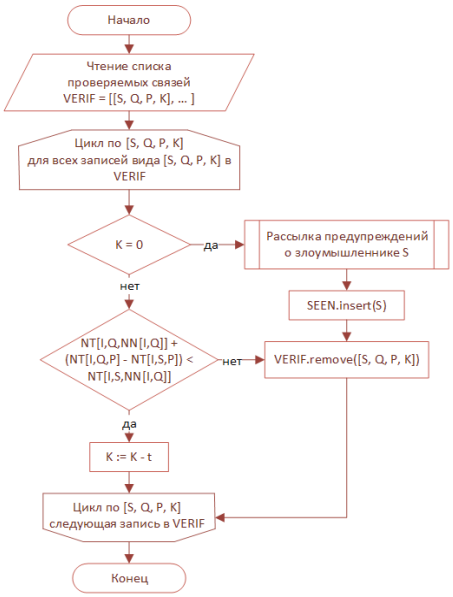

Рис. 5. Выполнение проверки состояния подозрительных соединений на узле I

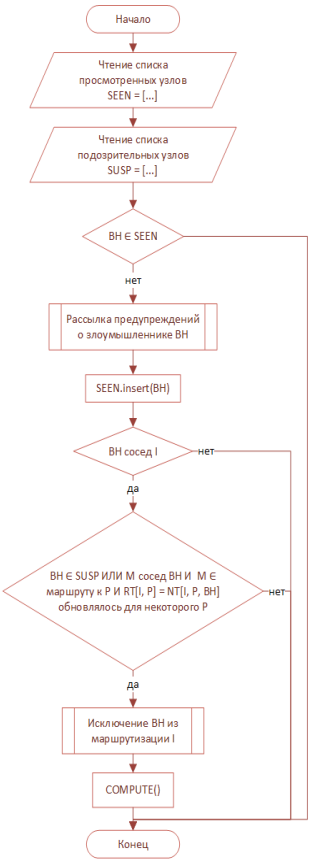

Рис. 6. Обработка узлом I сообщения с объявлением злоумышленника BH от узла M

Определим временную сложность алгоритма так же, как и в исходной работе [15] – как максимальное число шагов выполнения алгоритма между отправкой первого сообщения с обновлением маршрутизации и моментом, когда все узлы содержат окончательные кратчайшие маршруты до всех других узлов и сообщения с этим обновлением далее не рассылаются. За шаг выполнения алгоритма берется верхняя граница промежутка времени между созданием сообщения и окончанием его обработки, включая выполнение на узле функции COMPUTE(). Предполагается, что в каждый такой промежуток времени узлы синхронно выполняют распределенный алгоритм, обрабатывая все обновления маршрутизации, полученные к этому моменту от соседей. В оригинальной работе [15] доказано, что сложность исходного алгоритма в худшем случае составляет $O(h)$ где h – наибольшая глубина дерева кратчайших маршрутов. Величина h зависит от структуры сети и может изменяться со временем, но всегда $h < n$.

В условиях нормальной работы сети при обработке сообщений маршрутизации добавленные в модифицированном алгоритме проверки лишь увеличивают время выполнения обработки сообщений, при этом состав записей в сообщениях не затрагивается, а сами сообщения рассылаются тому же кругу узлов что и в исходном алгоритме. Таким образом для установления кратчайших маршрутов во всех узлах выполняется такое же число шагов что и в оригинальном алгоритме, то есть $O(h)$ в худшем случае.

Число шагов, необходимых для обнаружения злоумышленника и его блокировки различается для двух описанных ранее случаев. Если злоумышленник BH рассылает ложные маршруты, проходящие через своего соседа A, они будут обнаружены узлом A, который сразу же выполнит рассылку остальным соседям BH сообщений обнаружения злоумышленника. Рассылка этих сообщений будет выполняться до достижения ими всех соседей BH и потребует не более чем за $O(h)$ шагов. Если же BH рассылает маршруты, проходящие через узел P, не являющийся его соседом, его обнаружение потребует больше времени, поскольку сообщения должны будут дойти до узла T, находящегося на середине настоящего маршрута между BH и P. В худшем случае распространение сообщений до узла T займет $O(h)$ шагов, если BH объявит своим соседом P самый отдаленный узел сети. Затем узел T будет ожидать истечения счетчика K. Этот счетчик должен содержать число, не превосходящее максимальную глубину дерева маршрутизации $O(h)$. Если T убедится в недостоверности маршрута, отправленного BH, он должен оповестить остальные узлы рассылкой сообщений, которая затронет узлы, лежащие на обратном пути от T к BH и также займет $O(h)$ шагов.

В обоих вышеприведенных случаях после этого злоумышленник исключается из маршрутизации своими соседями. Процедура восстановления маршрутов после исключения злоумышленника из маршрутизации аналогична выполнению исходного алгоритма при отказе узла сети и требует также $O(h)$ шагов.

Таким образом общая вычислительная сложность модифицированного алгоритма аналогична исходной с увеличением на множитель, равный константе, и в худшем случае составляет $O(h)$.

Заключение

В работе предложена модификация одного из распределенных алгоритмов маршрутизации. Разработанная модификация позволяет распознавать ложную маршрутную информацию, распространяемую по сети, а также обнаруживать и блокировать ее отправителя. Для случаев подделывания маршрутной информации были рассмотрены возможные действия злоумышленника и разработаны методы обнаружения этих действий. Аналогично исходному алгоритму, модификация является адаптивной и распределенной и не создает петель маршрутизации. Она не требует больших затрат на поддержание дополнительной информации, а также не увеличивает алгоритмическую сложность оригинального алгоритма. Результаты тестирования на программном макете показали, что модифицированный алгоритм с высокой точностью обнаруживает злоумышленника и не вызывает ложных срабатываний.

Литература

1. B. Baron, P. Spathis, M. Dias de Amorim, Y. Viniotis, M. H. Ammar. Mobility as an Alternative Communication Channel: A Survey // IEEE Communications Surveys & Tutorials, vol. 21, No. 1, pp. 289-314, Firstquarter 2019. DOI: 10.1109/COMST.2018.2841192.
2. X. Fan, W. Cai, J. Lin. A survey of routing protocols for highly dynamic mobile ad hoc networks // 2017 IEEE 17th International Conference on Communication Technology (ICCT), 2017, pp. 1412-1417. DOI: 10.1109/ICCT.2017.8359865.

3. R. Skaggs-Schellenberg, N. Wang, D. Wright. Performance Evaluation and Analysis of Proactive and Reactive MANET Protocols at Varied Speeds // 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0981-0985. DOI: 10.1109/CCWC47524.2020.9031233
4. Y. Jahir, M. Atiquzzaman, H. Refai, A. Paranjothi, P. LoPresti. Routing protocols and architecture for disaster area network: A survey // Ad Hoc Networks, 2019, vol. 82, pp. 1-14, ISSN 1570-8705. DOI: 10.1016/j.adhoc.2018.08.005.
5. S. Shruthi. Proactive routing protocols for a MANET — A review // 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 821-827. DOI: 10.1109/I-SMAC.2017.8058294
6. B. R. Devi, K. K. Rao, M. A. Rani. Application of Modified Bellman-Ford Algorithm for Cooperative Communication // Wireless Personal Communications, 2019, No. 109, pp. 2025–2049. DOI: 10.1007/s11277-019-06666-7
7. T. Batista da Silveira, E. Mendes Duque, S. J. Ferzoli Guimarães, H. Torres Marques-Neto, H. Cota de Freitas. Proposal of Fibonacci Heap in the Dijkstra Algorithm for Low-power Ad-hoc Mobile Transmissions // IEEE Latin America Transactions, vol. 18, No. 03, pp. 623-630, March 2020. DOI: 10.1109/TLA.2020.9082735.
8. J. L. Wijekoon, P. K. W. Abeygunawardhana. Effective use of network device state information for network path selection// 2017 6th National Conference on Technology and Management (NCTM), 2017, pp. 205-210. DOI: 10.1109/NCTM.2017.7872855.
9. S.J. Gudakahriz, S. Jamali, M.V. Khiavi, A. Soleimany. A Stable TORA Based for Routing in Mobile Ad Hoc Networks // Engineering, Technology & Applied Science Research Vol. 8, No. 1, 2018, pp.2532-2536.
10. Z. Kartit, O. Diouri. Security Extension for Routing Protocols in Ad hoc Mobile Networks: A comparative Study // 2nd International Conference on Networking, Information Systems & Security (NISS19), 2019, vol. 69, pp. 1–7. DOI: 10.1145/3320326.3320403.
11. S. Kumar, M. Goyal, D. Goyal, R. C. Poonia. Routing protocols and security issues in MANET // 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), 2017, pp. 818-824. DOI: 10.1109/ICTUS.2017.8286119
12. S. Aluvala, R.S. Krovi, D. Vodnala. An Empirical Study of Routing Attacks in Mobile Ad-hoc Networks // Procedia Computer Science, 2016, No. 92, pp. 554-561. DOI: 10.1016/j.procs.2016.07.382.
13. S. Abbas, M. Haqdad, M. Z. Khan, H. U. Rehman, A. Khan A. u. R. Khan. Survivability Analysis of MANET Routing Protocols under DOS Attacks // KSII Transactions on Internet and Information Systems, 2020, vol. 14, No. 9, pp. 3639-3662. DOI: 10.3837/tiis.2020.09.004.
14. J. Vinayagam, CH. Balaswamy, K. Soundararajan. Certain Investigation on MANET Security with Routing and Blackhole Attacks Detection // Procedia Computer Science, 2019, vol. 165, pp. 196-208. DOI: 10.1016/j.procs.2020.01.091.
15. O. Sbai, M. Elboukhari. Simulation of MANET's Single and Multiple Blackhole Attack with NS-3 // 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), 2018, pp. 612-617. DOI: 10.1109/CIST.2018.8596606.

PROVIDING SAFE ROUTING IN WIRELESS SELF-ORGANIZING NETWORKS

Volkov M.S.A.[8], Gordeev E.N.[9]

Purpose of the study: *development of a distributed routing algorithm to ensure security in wireless self-organizing networks from intruder attacks on the network.*

Methods: *application of algorithms, graph theory, discrete optimization and heuristic approaches based on the results of numerical experiments.*

Results: *the paper proposes a modified distributed routing algorithm for self-organizing networks. The algorithm, presented in this paper, is based on a distributed version of Dijkstra's algorithm, designed to detect the shortest paths without loops on a graph under conditions of changing the weight of its edges. The loop freedom in this case is achieved by storing at each node an additional table containing the penultimate nodes on the shortest routes to all*

8 Sabina Volkov, student of the «Information Security» department, Bauman Moscow State Technical University, Moscow, Russia. E-mail: sabina-volkoff@yandex.ru

9 Eduard N. Gordeev, Dr.Sc. (Math.), professor of the «Information Security» department, Bauman Moscow State Technical University, Moscow, Russia. E-mail: werhorn@yandex.ru.

nodes, which allows the node to build a tree of shortest routes with itself as a root. In the modification of the algorithm, these tables are used by the nodes to check the correspondence of the declared route and the return route, which makes it possible to recognize and exclude from the network an intruder who carries out attacks to disrupt the correct routing mechanism. The effectiveness of the proposed algorithm for protection against routing attacks, in particular, black hole attacks, is confirmed by the results of testing on a program model.

Keywords: *distributed routing, proactive protocols, adaptive algorithm, routing loop, DOS-attack, "black hole", NP-completeness.*

References

1. B. Baron, P. Spathis, M. Dias de Amorim, Y. Viniotis, M. H. Ammar. Mobility as an Alternative Communication Channel: A Survey // IEEE Communications Surveys & Tutorials, vol. 21, No. 1, pp. 289-314, Firstquarter 2019. DOI: 10.1109/COMST.2018.2841192.
2. X. Fan, W. Cai, J. Lin. A survey of routing protocols for highly dynamic mobile ad hoc networks // 2017 IEEE 17th International Conference on Communication Technology (ICCT), 2017, pp. 1412-1417. DOI: 10.1109/ICCT.2017.8359865.
3. R. Skaggs-Schellenberg, N. Wang, D. Wright. Performance Evaluation and Analysis of Proactive and Reactive MANET Protocols at Varied Speeds // 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0981-0985. DOI: 10.1109/CCWC47524.2020.9031233
4. Y. Jahir, M. Atiquzzaman, H. Refai, A. Paranjothi, P. LoPresti. Routing protocols and architecture for disaster area network: A survey // Ad Hoc Networks, 2019, vol. 82, pp. 1-14, ISSN 1570-8705. DOI: 10.1016/j.adhoc.2018.08.005.
5. S. Shruthi. Proactive routing protocols for a MANET — A review // 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2017, pp. 821-827. DOI: 10.1109/I-SMAC.2017.8058294
6. B. R. Devi, K. K. Rao, M. A. Rani. Application of Modified Bellman-Ford Algorithm for Cooperative Communication // Wireless Personal Communications, 2019, No. 109, pp. 2025–2049. DOI: 10.1007/s11277-019-06666-7
7. T. Batista da Silveira, E. Mendes Duque, S. J. Ferzoli Guimarães, H. Torres Marques-Neto, H. Cota de Freitas. Proposal of Fibonacci Heap in the Dijkstra Algorithm for Low-power Ad-hoc Mobile Transmissions // IEEE Latin America Transactions, vol. 18, No. 03, pp. 623-630, March 2020. DOI: 10.1109/TLA.2020.9082735.
8. J. L. Wijekoon, P. K. W. Abeygunawardhana. Effective use of network device state information for network path selection// 2017 6th National Conference on Technology and Management (NCTM), 2017, pp. 205-210. DOI: 10.1109/NCTM.2017.7872855.
9. S.J. Gudakahriz, S. Jamali, M.V. Khiavi, A. Soleimany. A Stable TORA Based for Routing in Mobile Ad Hoc Networks // Engineering, Technology & Applied Science Research Vol. 8, No. 1, 2018, pp.2532-2536.
10. Z. Kartit, O. Diouri. Security Extension for Routing Protocols in Ad hoc Mobile Networks: A comparative Study // 2nd International Conference on Networking, Information Systems & Security (NISS19), 2019, vol. 69, pp. 1–7. DOI: 10.1145/3320326.3320403.
11. S. Kumar, M. Goyal, D. Goyal, R. C. Poonia. Routing protocols and security issues in MANET // 2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS), 2017, pp. 818-824. DOI: 10.1109/ICTUS.2017.8286119
12. S. Aluvala, R.S. Krovi, D. Vodnala. An Empirical Study of Routing Attacks in Mobile Ad-hoc Networks // Procedia Computer Science, 2016, No. 92, pp. 554-561. DOI: 10.1016/j.procs.2016.07.382.
13. S. Abbas, M. Haqdad, M. Z. Khan, H. U. Rehman, A. Khan A. u. R. Khan. Survivability Analysis of MANET Routing Protocols under DOS Attacks // KSII Transactions on Internet and Information Systems, 2020, vol. 14, No. 9, pp. 3639-3662. DOI: 10.3837/tiis.2020.09.004.
14. J. Vinayagam, CH. Balaswamy, K. Soundararajan. Certain Investigation on MANET Security with Routing and Blackhole Attacks Detection // Procedia Computer Science, 2019, vol. 165, pp. 196-208. DOI: 10.1016/j.procs.2020.01.091.
15. O. Sbai, M. Elboukhari. Simulation of MANET's Single and Multiple Blackhole Attack with NS-3 // 2018 IEEE 5th International Congress on Information Science and Technology (CiSt), 2018, pp. 612-617. DOI: 10.1109/CIST.2018.8596606.