

УДК 004.056+519.832

Алгоритмы распределения ресурсов для защиты информации между объектами информационной системы на основе игровой модели и принципа равной защищенности объектов

Быков А. Ю.^{1,*}, Шматова Е. С.¹

[*abykov@bmstu.ru](mailto:abykov@bmstu.ru)

¹МГТУ им. Н.Э. Баумана, Москва, Россия

В статье рассмотрены алгоритмы для решения задачи распределения ресурсов, выделенных на защиту информации, между объектами информационной системы. Задача распределения ресурсов сформулирована как игра двух игроков (защитника и нападающего) с нулевой суммой. Каждый игрок решает задачу линейного программирования при фиксированном решении другого игрока. Предложены три алгоритма, которые могут применяться последовательно, если предыдущий алгоритм не получил результат: оптимизационный алгоритм поиска седловой точки; алгоритм поиска седловой точки на основе обеспечения равной защищенности объектов; алгоритм поиска решения защитника на основе принципа гарантированного результата. Сформулированы достаточные условия того, что решение является седловой точкой при равномерной защищенности объектов. Представлены результаты решения задачи и тестирования алгоритмов.

Ключевые слова: информационная безопасность, игра с нулевой суммой, линейное программирование, седловая точка, критерий Вальда (максиминный критерий)

Введение

При решении различных практических задач, связанных с защитой информации, достаточно часто используется математический аппарат теории игр.

В [1] рассматривается применение теории игр для обеспечения безопасности в мобильных сетях MANET (Mobile Ad hoc Network – беспроводные децентрализованные самоорганизующиеся сети, состоящие из мобильных устройств). При применении математического аппарата теории игр к проблемам безопасности обычно рассматриваются два игрока: сторона защиты и сторона нападения. В сетях типа MANET использование двух игроков не эффективно, так как централизованное администрирование не доступно. Предложено использовать теорию игр среднего поля, которая ориентирована на множество игроков (в пределе бесконечное число игроков), каждый из которых стремится

к оптимизации некоторого функционала. Значение функционала зависит от поведения игрока и динамики распределения положений всех игроков. Динамика каждого игрока описывается дифференциальным уравнением, также зависящим от распределения игроков. Распределение положений игроков описывается вероятностной мерой. Предложенная схема может позволить отдельному узлу в MANETs принять решения по безопасности без централизованного администрирования.

В [2] рассмотрен обзор существующих решений теории игр для проблем сетевой безопасности, в статье приведен обзор. Представлена классификация игр по различным признакам: по числу шагов игры (статические, динамические, стохастические); по полноте информации о предыдущих ходах игроков; по полноте информации о функциях выигрыша других игроков. Рассмотрены различные игровые модели: модели кооперативных игр; модели некооперативных игр. Описаны различные комбинации признаков классов игр.

В [3] предложено использовать модели теории игр для моделирования общедоступных сетей, в которых могут быть как законные, так и незаконные пользователи. Незаконные пользователи имеют цель нанесения ущерба законным пользователям. В свою очередь, законные пользователи изначально не знают, какие из пользователей являются законными, а какие незаконными. Разработана модель взаимодействия законных и незаконных пользователей. Модель базируется на многошаговых графических играх с неполной информацией. Основное преимущество использования модели теории игр заключается в том, что проблема безопасности определяется через верхнюю границу ущерба, вызванного незаконными пользователями.

В [4] рассматриваются бортовые сети (Vehicular networks – VANETs), элементами которых являются устройства, установленные на автомобилях, и ретрансляторы. Аспекты безопасности VANETs исследуются на игровой модели, в которой защитные меры оптимизированы относительно угрозы, представленной возможными атаками на сеть, например, с помощью помех. Предложенная математическая модель является достаточно абстрактной и ориентирована на применение в сетях, которые возможно будут созданы в будущем. В игровой модели одним из показателей является мера централизованности (centrality) сети, вычисляемая через отображение величин централизованности автомобильных сетей к базовой дорожной топологии. Результирующие стратегии помогают определить места наиболее важных и уязвимых точек (например, против глушения). Предложены различные игровые постановки: постановка задачи игры двух игроков с нулевой суммой, с показателем качества и ограничениями в виде равенств и неравенств; нечеткая игра, некоторые параметры описаны нечетко, предложены треугольные и трапециевидные функции принадлежности; стохастическая фиктивная игра. Проведены оценки эффективности игровых решений на компьютерных моделях, в которых были использованы реальные данные, полученные из систем организации трафика.

В [5] рассматривается проблема аутентификации в мобильных сетях и анализируется независимое поведение мобильных узлов при помощи теоретической игровой модели, в которой каждый игрок стремится максимизировать конфиденциальность места своего расположения при минимальной стоимости. В теории рассмотрен алгоритм получения равновесия по Нэшу в статических играх n игроков с полной информацией. Так как на практике мобильные узлы не знают выигрыша своих противников, предлагается использовать статические игры с неполной информацией. Установлено, что симметричное байесовское равновесие по Нэшу существует в простых предельных стратегиях (threshold strategies). С помощью численных экспериментов предсказано поведение независимых мобильных узлов. Также рассмотрены динамические игры, в которых игроки выполняют ходы последовательно, и показано, как это поведение влияет на стратегии в равновесии. В заключение предложены протоколы на основе теории игр.

В [6] игровая модель применяется для оценки возможностей применения средств защиты от вредоносных программ. Некоторые пользователи не используют средства защиты из-за дополнительных расходов, в статье рассматриваются побудительные мотивы к использованию средств защиты пользователем большой сети. В модели учитывается уязвимость пользователя к угрозам и возможные потери от реализации этих угроз. Функции потери заданы как непрерывные. Сформулированы условия, которые гарантируют, что сумма, потраченная на защиту, является возрастающей функцией уязвимости пользователя и возможных потерь от реализации угроз. Показано, что для предотвращения большего числа угроз на защиту требуется потратить незначительную часть возможных потерь от непредотвращения этих угроз. Также рассмотрено множество пользователей в сети, подверженных заражению вредоносными программами. Получены условия, которые стимулируют пользователей к разумным тратам на безопасность. Рассмотрены условия для получения состояния равновесия при координации пользователей с высокой степенью их независимости.

В [7] модель некооперативной игры с нулевой суммой используется при моделировании двух абонентов и передатчика помех в канале аддитивного белого нормально-распределенного шума. В качестве целевой функции используется количество информации каналов ввода-вывода (channel's input/output mutual information). Считается, что источник помех может слушать канал и использовать полученную информацию для формирования коррелированной с сигналом помехи. В источнике помех также различается информация между сигналом и помехой. Исследованы условия существования равновесных стратегий при различных предположениях о характеристиках канала связи и информации, доступной о сигналах пользователей в источнике помех.

В [8] для борьбы с атаками в беспроводных сетях с множеством абонентов используются модели игры со многими игроками и косвенным взаимодействием между ними, так как прямое взаимодействие между многими игроками затруднено. В модели происходит «наказание» атакующих узлов с помощью остановки их сетевых служб. Цель состоит в том, чтобы игрок нес расходы из-за потери доступа к сети, даже в том случае,

если он реализует антагонистические стратегии. Таким образом, узлы сети мотивированы на отказ от антагонистического поведения в их собственных интересах. Таким образом, получены оптимальные стратегии и условие устойчивости оптимальных стратегий. Предложенная модель устойчива против сговора некоторых игроков и значительно снижает возможности проведения атак. Представлены результаты моделирования, которые показывают, что предложенная модель значительно выигрывает у моделей, основанных на прямом взаимодействии игроков, особенно в крупномасштабных сетях с мобильными узлами.

В [9] рассмотрена модель Smart-Grid сети («Умные сети электроснабжения») в виде игры Маркова для моделирования защиты сети от различных атак. Игровая модель задает взаимоотношения между провайдером и атакующей стороной, получены оптимальные стратегии сторон. Однако вычислительная сложность алгоритмов получения решений является экспоненциальной в зависимости от размера инфраструктуры сети. Это является неприемлемым для использования в реальных сетях. В статье рассматриваются приближенные алгоритмы получения оптимальных стратегий, так как в этом случае мощность множества состояний игры Маркова может быть сокращена. Разработан алгоритм сокращения множества состояний, в котором используется пороговое значение специального параметра для того, чтобы качественно управлять компромиссом между точностью решения и временем вычисления. Кроме того, для уменьшения времени вычисления при использовании многоядерных процессоров алгоритм также может быть реализован в параллельной форме, приведены результаты экспериментов.

В [10] моделируется радиосеть, в которой присутствуют основные и дополнительные пользователи, совместно использующие спектр радиосигнала. Предложена новая модель сотрудничества в интеллектуальных радиосетях, в которой основные пользователи повышают конфиденциальность своих данных, одновременно повышая защищенность дополнительных пользователей, в присутствии интеллектуального пассивного наблюдателя, пытающегося декодировать сообщения основных пользователей. Для получения достижимой пары, состоящей из уровня конфиденциальности основных пользователей и скорости передачи дополнительных пользователей, при различных условиях используется игра Stackelberg, в которой моделируется взаимодействие между основными и дополнительными пользователями. В этой игре уровни мощности передачи являются ключом для максимизации скорости передачи данных. На основе 2D-представления показано, как достижимые уровни скорости передачи зависят от областей уровня мощности. Сформулированы условия состояния равновесия для получения оптимальных стратегий основных и дополнительных пользователей. Представлены результаты моделирования.

В [11] игровая постановка задачи используется для моделирования системы обнаружения вторжений в неоднородной сети, в которой для разных узлов сети могут применяться разные средства защиты. В игре используются две стороны: сторона нападения и сторона защиты (средства обнаружения вторжений). В поставленной задаче

выполнен поиск равновесия по Нэшу. На основе сформулированной задачи получены оптимальные стратегии стороны нападения и оптимальные стратегии стороны защиты, требующие минимум ресурсов. Проведено моделирование полученных результатов.

В [12] для выбора средств защиты от DDoS-атак предложено использовать матричную игру двух игроков с нулевой суммой. Решение по различным критериям оптимальности искалось на платежной матрице игры.

В [13] представлен обзор применения игровых моделей в задачах защиты информации, рассмотрено применение игровой модели для систем обнаружения вторжений.

В [14] рассматривается игра, в которой защитник выбирает средства защиты информации, а нападающий – средства проведения атак на информационные ресурсы. Каждая из сторон при выборе решает задачу дискретного программирования. При решении задачи защитником используется минимаксный критерий для снижения возможного ущерба, предложены точные и приближенные алгоритмы решения задачи.

В данной статье рассмотрим постановку задачи распределения ресурсов между объектами защиты в распределенной информационной системе. Похожая постановка задачи была представлена в [11] применительно к выбору контролируемых узлов в системе обнаружения вторжений, в основе была использована биматричная игра и предложен алгоритм поиска ситуации равновесия по Нэшу, в случае существования данной ситуации, на основе упорядочивания узлов сети по степени их важности. Рассмотрим более общую постановку задачи распределения ресурсов между объектами защиты информационной системы.

1. Постановка задачи распределения ресурсов между объектами защиты

Исходные данные

1. $S = \{s_1, s_2, \dots, s_n\}$ – множество объектов системы, на которых могут находиться защищаемые информационные ресурсы, $N = \{1, 2, \dots, n\}$ – множество индексов этих объектов.
2. $w_i > 0, \forall i \in N$ – стоимость защищаемых ресурсов объекта (возможный ущерб при нарушении требований безопасности).
3. $c_{zi} > 0, \forall i \in N$ – стоимость полной защиты объекта.
4. $c_{ni} > 0, \forall i \in N$ – стоимость проведения полной атаки на объект со стороны нападения.
5. $p_{ni} > 0, \forall i \in N$ – вероятность предотвращения атаки на объекте при использовании средств защиты.

Показатели игроков

Для стороны защиты введем переменную $p_i \in [0, 1], \forall i \in N$, имеющую содержательный смысл вероятности контроля объекта (можно провести другую

интерпретацию значения этой переменной как доли выделенных ресурсов на защиту объекта: значение 0 – ресурсы на защиту не выделяются, 1 – на защиту выделяются максимум возможных ресурсов для объекта), переменные образуют вектор \vec{P} . Для стороны нападения введем подобную переменную $q_i \in [0, 1], \forall i \in N$, имеющую содержательный смысл вероятности атаки на объект (или доли выделенных ресурсов для проведения атаки на объект), переменные образуют вектор \vec{Q} .

В отношении каждого объекта для сторон защиты и нападения возможны две стратегии: сторона защиты проводит защиту объекта или не проводит, сторона нападения проводит атаку на объект или нет. Для каждой стороны можно составить свою матрицу игры.

Матрицы игр сторон защиты и нападения для отдельного i -го объекта представлены в табл. 1 и 2, соответственно.

Таблица 1. Матрица игры стороны защиты

	Защищает	Не защищает
Нападает	$-(1 - p_{np\ i}) w_i - c_{zi}$	$-w_i$
Не нападает	$-c_{zi}$	0

Таблица 2. Матрица игры стороны нападения

	Защищает	Не защищает
Нападает	$(1 - p_{np\ i}) w_i - c_{ni}$	$w_i - c_{ni}$
Не нападает	0	0

Эти таблицы задают биматричную игру для отдельного объекта, похожая задача на основе биматричной игры была рассмотрена в [11].

Выигрыш стороны защиты в целом, который можно рассматривать как результат кооперативной игры «защитников» отдельных объектов, определяется возможным ущербом и затратами с противоположным знаком:

$$F_3(\vec{P}, \vec{Q}) = \sum_{i \in N} [p_i q_i (-(1 - p_{np\ i}) w_i - c_{zi}) - (1 - p_i) q_i w_i - p_i (1 - q_i) c_{zi}] = \sum_{i \in N} [p_i q_i p_{pr\ i} w_i - q_i w_i - p_i c_{zi}]. \quad (1)$$

Выигрыш стороны нападения, который можно рассматривать как результат кооперативной игры «атакующих» отдельные объекты, определяется возможным ущербом для стороны защиты минус затраты на проведение атак:

$$F_H(\vec{P}, \vec{Q}) = \sum_{i \in N} [p_i q_i ((1 - p_{np\ i}) w_i - c_{ni}) + (1 - p_i) q_i (w_i - c_{ni})] = \sum_{i \in N} [q_i w_i - p_i q_i p_{pr\ i} w_i - q_i c_{ni}]. \quad (2)$$

Представленные показатели запишем в другой форме. Показатель (1) принимает вид:

$$F_3(\vec{P}, \vec{Q}) = U_{np}(\vec{P}, \vec{Q}) - U_{max}(\vec{Q}) - Z_3(\vec{P}). \quad (3)$$

Показатель (2) принимает вид:

$$F_H(\vec{P}, \vec{Q}) = U_{max}(\vec{Q}) - U_{np}(\vec{P}, \vec{Q}) - Z_H(\vec{Q}), \quad (4)$$

где $U_{np}(\vec{P}, \vec{Q}) = \sum_{i \in N} p_i q_i p_{pr\ i} w_i$ – предотвращенный ущерб стороной защиты;

$U_{max}(\vec{Q}) = \sum_{i \in N} q_i w_i$ – максимальный ущерб, который может быть нанесен стороной нападения при отсутствии защиты;

$Z_3(\vec{P}) = \sum_{i \in N} c_{zi} p_i$ – затраты стороны защиты;

$Z_H(\vec{Q}) = \sum_{i \in N} c_{hi} q_i$ – затраты стороны нападения.

Преобразуем предложенную модель игры с показателями (1) и (2) или (3) и (4) в модель игры с нулевой суммой. Для этого затраты игроков вынесем из показателей ограничения, т.е. компоненты показателей (3) и (4), определяющие затраты защитника и нападающего.

Ограничения

Для стороны защиты ограничение на стоимость ресурсов, выделяемых на защиту:

$$\sum_{i \in N} c_{zi} p_i \leq C_3^{(max)}, \quad (5)$$

где $C_3^{(max)}$ – максимальная стоимость ресурсов, выделенных на защиту.

Для стороны нападения ограничение на стоимость ресурсов, выделяемых на проведение атак:

$$\sum_{i \in N} c_{hi} q_i \leq C_H^{(max)}, \quad (6)$$

где $C_H^{(max)}$ – максимальная стоимость ресурсов, выделенных на проведение атак.

Как правило, на практике размер затрат нападающего для проведения атаки на любой из объектов (стоимость проведения атаки) – c_{hi} ниже, чем стоимость средств, выделяемых на защиту объекта – c_{zi} , и существенно ниже, чем стоимость защищаемых ресурсов (ущерб защитника) – w_i ($c_{hi} \ll w_i, \forall i \in N$). В этом случае сторона нападения расходует все ресурсы, выделенные на проведение атак, т.е. неравенство (6) превращается в равенство:

$$\sum_{i \in N} c_{hi} q_i = C_H^{(max)}, \quad (7)$$

при условии, что $\sum_{i \in N} c_{hi} > C_H^{(max)}$ (т.е. ресурсов нападающего не хватает на проведение полных атак на все объекты). Ограничение (5) также представим в виде равенства:

$$\sum_{i \in N} c_{zi} p_i = C_3^{(max)}, \quad (8)$$

при условии, что $\sum_{i \in N} c_{zi} > C_3^{(max)}$ (т.е. ресурсов защитника не хватает на полную защиту всех объектов). Кроме того, обычно стоимость защиты объекта меньше, чем предотвращенный ущерб $c_{zi} < p_{pr} i w_i$ (иначе использовать защиту объекта не целесообразно).

Тогда слагаемые с затратами исключим из показателей (3) и (4), показатели игроков примут вид:

Для стороны защиты:

$$F_3(\vec{P}, \vec{Q}) = U_{пр}(\vec{P}, \vec{Q}) - U_{max}(\vec{Q}). \quad (9)$$

Для стороны нападения:

$$F_H(\vec{P}, \vec{Q}) = U_{max}(\vec{Q}) - U_{пр}(\vec{P}, \vec{Q}). \quad (10)$$

Ограничение для стороны защиты задано равенством (8), ограничение для стороны нападения – равенством (7) (следует также учитывать ограничения $p_i \in [0, 1], \forall i \in N$, и $q_i \in [0, 1], \forall i \in N$).

Получаем игру с нулевой суммой, выигрыш защитника – его ущерб с противоположным знаком, выигрыш нападающего – ущерб защитника. Множества возможных решений для защитника и нападающего (вектора \vec{P} и \vec{Q}) представляют собой континуальные множества, поэтому матричное представление для такой игры не подходит. Рассмотрим возможные подходы к решению данной игры.

2. Алгоритмы решения задачи

Будем рассматривать стратегию решения игры со стороны защитника. Остановимся на двух подходах:

1. Поиск седловой точки, если она существует. Седловая точка – это решение игры $\langle \vec{P}^*, \vec{Q}^* \rangle$, которое удовлетворяет условиям: $F_3(\vec{P}, \vec{Q}^*) \leq F_3(\vec{P}^*, \vec{Q}^*)$, $F_H(\vec{P}^*, \vec{Q}) \leq F_H(\vec{P}^*, \vec{Q}^*)$ для любых допустимых векторов \vec{P} и \vec{Q} .
2. Критерий Вальда. Это максиминный критерий, который гарантирует определенный выигрыш при наихудших условиях. Его можно использовать, если седловой точки не существует или алгоритм не позволяет ее получить, или защитник не обладает полной информацией о ресурсах нападающего.

Проведем преобразование исходной задачи. Учитывая то обстоятельство, что для стороны защиты в показателе (9) слагаемое $U_{max}(\vec{Q})$ не зависит от решения защитника, для стороны защитника будем использовать показатель:

$$F_3(\vec{P}, \vec{Q}) = U_{пр}(\vec{P}, \vec{Q}) = \sum_{i \in N} w_i p_{pr i} q_i p_i. \quad (11)$$

Данный показатель определяет предотвращенный ущерб стороны защиты. Защитник стремится максимизировать показатель (11).

Для стороны нападения показатель можно записать в следующем виде:

$$F_H(\vec{P}, \vec{Q}) = \sum_{i \in N} (q_i w_i - p_i q_i p_{pr i} w_i) = \sum_{i \in N} (w_i - p_i p_{pr i} w_i) q_i. \quad (12)$$

Данный показатель определяет общий ущерб стороны защиты.

Также будем осуществлять поиск решения при условиях: $\sum_{i \in N} c_{zi} > C_3^{(max)}$ и $\sum_{i \in N} c_{hi} > C_H^{(max)}$ (ресурсов защитнику и нападающему не хватает для использования всех возможных средств защиты или нападения). Так как, если $\sum_{i \in N} c_{hi} \leq C_H^{(max)}$, решение очевидное, нападающий использует все средства нападения (все компоненты вектора \vec{Q} равны 1), а защитник при заданном \vec{Q} решает задачу линейного программирования (ЗЛП) максимизации показателя (11) с ограничением (8). Аналогично, если $\sum_{i \in N} c_{zi} \leq C_3^{(max)}$, то защитник использует все средства защиты (все компоненты вектора \vec{P} равны 1), а нападающий при заданном \vec{P} решает ЗЛП максимизации показателя (12) с ограничением (7).

2.1 Алгоритм решения задачи линейного программирования

Для начала приведем простой алгоритм решения задачи линейного программирования с одним ограничением (подобная задача решается одной стороной игры при фиксированном решении противоположной стороны). Данную задачу можно интерпретировать как распределение ограниченного ресурса между объектами. Постановка задачи имеет вид:

Максимизировать целевую функцию:

$$F(\vec{X}) = \sum_{i=1}^n c_i x_i, \quad (13)$$

при ограничениях:

$$\begin{cases} \sum_{i=1}^n a_i x_i \leq b, \\ x_i \in [0, 1], i = \overline{1, n}, \end{cases} \quad (14)$$

где $x_i, i = \overline{1, n}$ – искомые переменные, задающие распределение ресурса между объектами;

$c_i \geq 0, i = \overline{1, n}$ – заданные коэффициенты при целевой функции, можно интерпретировать как полезность (получаемая выгода) использования единицы ресурса на i -ом объекте;

$a_i > 0, i = \overline{1, n}$ – заданные коэффициенты при ограничении, можно интерпретировать как стоимость единицы ресурса при использовании на i -ом объекте;

$b > 0$ – заданное значение, можно интерпретировать, как суммарную доступную стоимость ресурса.

Будем также полагать, что $\sum_{i=1}^n a_i > b$, т.е. ресурса на все объекты не хватает, так как в противном случае, оптимальным является допустимое решение $\vec{X} = \|1, 1, \dots, 1\|^T$, состоящее из всех 1. В связи с этим неравенство в (14) можно заменить равенством (ресурс будет израсходован полностью):

$$\sum_{i=1}^n a_i x_i = b. \quad (15)$$

Не ограничивая общность постановки задачи, считаем, что переменные пронумерованы в порядке невозрастания отношений: $\frac{c_1}{a_1} \geq \frac{c_2}{a_2} \geq \frac{c_3}{a_3} \geq \dots \geq \frac{c_n}{a_n}$. Отношения $k_i = \frac{c_i}{a_i}, i = \overline{1, n}$ назовем коэффициентами полезности использования ресурса на i -м объекте, содержательный смысл коэффициента – полезность (получаемая выгода) при использовании ресурса единичной стоимости на объекте. При работе алгоритма переменные делятся на два непересекающихся класса: фиксированные, которым уже присвоены значения, и свободные, которым еще значения не присвоены. Алгоритм можно описать следующим образом.

В начале – все переменные свободные.

Шаг 0 (подготовительный). Значение стоимости свободного ресурса полагаем равным: $b_0 = b$.

Шаг i ($i=1, 2, \dots$, максимум n). Полагаем $x_i = x_i^{(max)}$, где $x_i^{(max)}$ – максимальное значение переменной с учетом выполнения ограничения (15),

$$x_i^{(max)} = \begin{cases} 1, & \text{если } a_i \leq b_0; \\ b_0/a_i, & \text{в противном случае.} \end{cases}$$

Если $a_i \geq b_0$, работа алгоритма завершена (свободный ресурс закончился), оставшиеся свободные переменные x_{i+1}, \dots, x_n полагаем равными 0, в противном случае, $b_0 = b_0 - a_i$ и переходим к шагу $i+1$.

Таким образом, вектор-решение имеет вид: $\vec{X} = \|1, 1, \dots, r, 0, 0, \dots, 0\|^T$, где $r \in (0, 1]$. В зависимости от исходных данных множество компонент, равных 1, или множество компонент, равных 0, могут быть пустыми (но не оба множества одновременно, если размерность вектора \vec{X} больше единицы).

Докажем, что только представленное решение может быть оптимальным. Рассмотрим три случая:

- 1) Возьмем любую компоненту из решения, неравную 0, и приравняем ее к 0, а компоненте, равной нулю, присвоим ненулевое значение, таким образом, чтобы выполнялось равенство (15). Не ограничивая общности, возьмем из решения i -ю компоненту, неравную нулю. Пусть $a_i x_i = b'$ ($x_i \neq 0$), где b' - стоимость ресурса, выделенного на i -ю компоненту решения, $x_i = \frac{b'}{a_i}$. j -й компоненте ($x_j = 0$) присвоим ненулевое значение. Стоимость ресурса b' будет направлена на j -ю компоненту, $a_j x_j = b' \Rightarrow x_j = \frac{b'}{a_j}$ (в этом случае, возможно, что $x_j > 1$). Тогда в целевой функции (13) вместо слагаемого $c_i x_i = c_i \frac{b'}{a_i} = k_i b'$ будет слагаемое $c_j x_j = c_j \frac{b'}{a_j} = k_j b'$. Так как $k_i \geq k_j$ и $b' > 0$, то $c_i x_i \geq c_j x_j$. Таким образом, целевая функция не улучшит своего значения.
- 2) Одну компоненту из решения, неравную нулю, заменим на несколько компонент, равных нулю. Рассмотрим случай замены одной компоненты на две (при замене на большее число компонент ход рассуждений такой же, но соотношения будут более громоздкими). Пусть $x_i \neq 0$, а $x_j = x_\gamma = 0$. Ресурс стоимости $a_i x_i = b'$ будет распределен между j -й и γ -й компонентами: $a_j x_j + a_\gamma x_\gamma = b'$. Следовательно, $x_j = \frac{b' - a_\gamma x_\gamma}{a_j}$, $x_\gamma = \frac{b' - a_j x_j}{a_\gamma}$. В целевой функции (13) вместо слагаемого $c_i x_i = c_i \frac{b'}{a_i} = k_i b'$ будет пара слагаемых $c_j x_j + c_\gamma x_\gamma = c_j \frac{b' - a_\gamma x_\gamma}{a_j} + c_\gamma \frac{b' - a_j x_j}{a_\gamma} = k_j (b' - a_\gamma x_\gamma) + k_\gamma (b' - a_j x_j)$. Пусть $k_j \geq k_\gamma$, тогда можно записать $k_\gamma = k_j g$, где $0 < g \leq 1$ - некоторый множитель. Тогда $c_j x_j + c_\gamma x_\gamma = k_j (b' - a_\gamma x_\gamma) + g k_j (b' - a_j x_j) = k_j [(b' - a_\gamma x_\gamma) + g (b' - a_j x_j)]$. Сумма $(b' - a_\gamma x_\gamma) + (b' - a_j x_j) = b'$, так как $0 < g \leq 1$ и слагаемые в сумме неотрицательные, то $(b' - a_\gamma x_\gamma) + g (b' - a_j x_j) \leq (b' - a_\gamma x_\gamma) + (b' - a_j x_j) = b'$. Поскольку $k_j \leq$

k_i , то $k_j [(b' - a_\gamma x_\gamma) + g(b' - a_j x_j)] \leq k_i b'$ или $c_j x_j + c_\gamma x_\gamma \leq c_i x_i$. Таким образом, значение целевой функции не может быть увеличено.

- 3) Несколько компонент из решения, неравных нулю (например, две), заменим на одну компоненту, равную нулю. Пусть $x_i \neq 0$ и $x_j \neq 0$, а $x_\gamma = 0$. То, рассуждая аналогично, ограничение $a_i x_i + a_j x_j = b'$ будет заменено на $a_\gamma x_\gamma = b'$. При этом $x_i = \frac{b' - a_j x_j}{a_i}$, $x_j = \frac{b' - a_i x_i}{a_j}$. В целевой функции пара слагаемых $c_i x_i + c_j x_j$ будет заменена на $c_\gamma x_\gamma = c_\gamma \frac{b'}{a_\gamma} = k_\gamma b'$. Преобразуем пару слагаемых: $c_i x_i + c_j x_j = c_i \frac{b' - a_j x_j}{a_i} + c_j \frac{b' - a_i x_i}{a_j} = k_i (b' - a_j x_j) + k_j (b' - a_i x_i)$. Пусть $k_i \geq k_j$, тогда можно записать $k_i = k_j g$, где $g \geq 1$ - некоторый множитель, $c_i x_i + c_j x_j = k_j g (b' - a_j x_j) + k_j (b' - a_i x_i) = k_j [g(b' - a_j x_j) + (b' - a_i x_i)]$. Так как $k_j \geq k_\gamma$ и $g(b' - a_j x_j) + (b' - a_i x_i) \geq b'$, то $k_j [g(b' - a_j x_j) + (b' - a_i x_i)] \geq k_\gamma b'$, следовательно, $c_i x_i + c_j x_j \geq c_\gamma x_\gamma$. Т.е. значение целевой функции не может быть увеличено.

Случай замены нескольких компонент на несколько можно свести к комбинации приведенных трех случаев.

Следствие. Если предположить, что все коэффициенты полезности равны между собой, $k_1 = k_2 = \dots = k_n$, то тогда назначать можно любые значения компонентам вектора \vec{X} , чтобы выполнялись ограничения $\begin{cases} \sum_{i=1}^n a_i x_i = b, \\ x_i \in [0, 1], i = \overline{1, n}, \end{cases}$ т.е. безразлично, как распределять ресурс между существующими объектами.

2.2 Оптимизационный подход к поиску седловой точки

В соответствии с приведенным алгоритмом решения ЗЛП для защитника исходя из показателя (11) и ограничения (8) коэффициенты полезности имеют вид $k_{3i}(q_i) = \frac{q_i p_{pr} i w_i}{c_{3i}}$, $i = \overline{1, n}$, т.е. зависят от решения стороны нападения – \vec{Q} . Для нападающего исходя из показателя (12) и ограничения (7) коэффициенты полезности имеют вид $k_{ni}(p_i) = \frac{w_i - p_i p_{pr} i w_i}{c_{ni}}$, $i = \overline{1, n}$, зависят от решения стороны защиты – \vec{P} . В соответствии с алгоритмом решения ЗЛП защитник в первую очередь будет выделять ресурсы на объекты с большими k_{3i} , нападающий – на объекты с большими k_{ni} .

Рассмотрим следующий эвристический алгоритм поиска седловой точки в рамках оптимизационного подхода или определения того, что алгоритм не позволяет получить седловую точку.

Шаг 0 (предварительный). Полагаем, что решение для стороны нападения равно $\vec{Q} = \|1, 1, \dots, 1\|^T$, т.е. решение состоит из всех единиц, оно недопустимо с точки зрения выполнения ограничений (7). Решаем ЗЛП для стороны защиты, т.е. оптимизирует

показатель (11) при ограничении (8), получаем значение вектора \vec{P} . Полученное решение помещаем в список решений.

Шаг 1, 3 (нечетные шаги). Решаем ЗЛП для стороны нападения, т.е. оптимизирует показатель (12) при ограничении (7), получаем значение вектора \vec{Q} .

Шаг 2, 4 (четные шаги). Решаем ЗЛП для стороны защиты, т.е. оптимизирует показатель (11) при ограничении (8), получаем значение вектора \vec{P} . Проверяем список решений. Если полученное решение находится в списке решений, то переходим к заключительному шагу. Если подобного решения нет в списке решений, то помещаем решение в список и переходим к следующему шагу.

Шаг заключительный (нечетный). Если решение \vec{P} , полученное на предыдущем шаге, находится последним в списке, то решаем ЗЛП для стороны нападения, т.е. оптимизирует показатель (12) при ограничении (7), получаем значение вектора \vec{Q} . Вектора \vec{P} и \vec{Q} задают седловую точку, если решать оптимизационные задачи дальше, то эти два решения будут повторяться. Если решение \vec{P} , полученное на предыдущем шаге, находится не последним в списке, то данный алгоритм не позволяет получить седловой точки, решения также будут повторяться, но не на соседних шагах, а через несколько шагов.

2.3 Подход к поиску седловой точки, основанный на равной защищенности объектов

Рассмотрим подход к возможному решению задачи распределения ресурсов для стороны защиты, т.е. к выбору значений вектора \vec{P} . Если удастся подобрать компоненты вектора \vec{P} , таким образом, чтобы коэффициенты полезности для стороны нападения были равны между собой:

$$k_{n1} = k_{n2} = \dots = k_{nn}. \quad (15)$$

(тем самым защитник обеспечивает одинаковую защищенность своих объектов), то любое допустимое по ограничению (8) решение стороны нападения, будет давать нападающему одинаковый выигрыш. Для этого необходимо решить систему уравнений относительно компонент вектора \vec{P} следующего вида:

$$\begin{cases} \frac{w_1 - p_1 p_{pr1} w_1}{c_{n1}} = \frac{w_2 - p_2 p_{pr2} w_2}{c_{n2}}, \\ \frac{w_1 - p_1 p_{pr1} w_1}{c_{n1}} = \frac{w_3 - p_3 p_{pr3} w_3}{c_{n3}}, \\ \dots \\ \frac{w_1 - p_1 p_{pr1} w_1}{c_{n1}} = \frac{w_n - p_n p_{prn} w_n}{c_{nn}}, \\ \sum_{i \in N} c_{zi} p_i = C_3^{(max)}. \end{cases} \quad (16)$$

Данная система является системой n линейных уравнений с n неизвестными переменными. Если представить ее в векторно-матричной форме вида $A \vec{P} = \vec{B}$, то матрица A имеет вид:

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & \dots & 0 \\ a_{21} & 0 & a_{23} & 0 & \dots & 0 \\ a_{31} & 0 & 0 & a_{34} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ a_{n-1 1} & 0 & 0 & 0 & \dots & a_{n-1 n} \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & \dots & a_{nn} \end{pmatrix},$$

где $a_{11} = a_{21} = a_{31} = \dots = a_{n-1 1} = \frac{p_{pr 1} w_1}{c_{H 1}}$, $a_{12} = -\frac{p_{pr 2} w_2}{c_{H 2}}$, $a_{23} = -\frac{p_{pr 3} w_3}{c_{H 3}}$, \dots ,
 $a_{n-1 n} = -\frac{p_{pr n} w_n}{c_{H n}}$, $a_{n1} = c_{31}$, $a_{n2} = c_{32}$, \dots , $a_{nn} = c_{3n}$.

Вектор \vec{B} имеет вид:

$$\vec{B} = \begin{pmatrix} \frac{w_1}{c_{H 1}} - \frac{w_2}{c_{H 2}} \\ \frac{w_1}{c_{H 1}} - \frac{w_3}{c_{H 3}} \\ \dots \\ \frac{w_1}{c_{H 1}} - \frac{w_n}{c_{H n}} \\ C_3^{(max)} \end{pmatrix}.$$

Если представленные элементы матрицы A , которые могут быть ненулевыми, действительно не равны нулю, то определитель A не равен 0, т.к. отсутствуют нулевые строки (столбцы) и нет линейно-зависимых строк (столбцов). В этом случае в соответствии с теоремой Кронекера-Капелли система (16) имеет единственное решение.

Решив систему уравнений (16) любым известным методом, получим вектор \vec{P} , который обеспечит то, что нарушитель при выборе любого допустимого решения всегда будет получать одно и то же значение показателя качества.

Подобную систему уравнений можно составить для нарушителя. Чтобы защитнику было все равно, какое решение принимать, необходимо обеспечить равенство коэффициентов полезности защитника:

$$k_{3 1} = k_{3 2} = \dots = k_{3 n}.$$

Тем самым нарушитель обеспечивает одинаковую угрозу или незащищенность для всех объектов. Для этого необходимо решить систему уравнений относительно компонент вектора \vec{Q} следующего вида:

$$\begin{cases} \frac{q_1 p_{pr 1} w_1}{c_{3 1}} = \frac{q_2 p_{pr 2} w_2}{c_{3 2}}, \\ \frac{q_1 p_{pr 1} w_1}{c_{3 1}} = \frac{q_3 p_{pr 3} w_3}{c_{3 3}}, \\ \dots \\ \frac{q_1 p_{pr 1} w_1}{c_{3 1}} = \frac{q_n p_{pr n} w_n}{c_{3 n}}, \\ \sum_{i \in N} c_{ni} q_i = C_H^{(max)}. \end{cases} \quad (17)$$

Если все коэффициенты при $q_i, \forall i \in N$, в системе (17) не равны 0, то система (17) также имеет единственное решение.

Если решить системы (16) и (17), то только при некоторых исходных данных все компоненты являются допустимыми, т.е. $p_i \in [0, 1], q_i \in [0, 1], \forall i \in N$. Вероятнее будет

ситуация, когда $\exists i \in N, p_i < 0 \vee p_i > 1 \vee q_i > 1$, т.е. полученное решение будет недопустимым. Рассмотрим, как можно разрешить эту ситуацию, если использовать равную защищенность не всех объектов, а только некоторых из них. Аналогично нападающий будет обеспечивать одинаковую незащищенность только для некоторых объектов. Защитника будем называть первым игроком, а нападающего – вторым игроком.

Лемма 1. Если первый (второй) игрок обеспечил равную защищенность (незащищенность) для некоторых объектов, т.е. принял решение на основе решения системы уравнений вида (16) ((17) для второго игрока) с учетом оставшегося ресурса (за вычетом ресурса, направленного на другие объекты), и второй (первый) обеспечил равную незащищенность (защищенность) для этих же объектов на основе решения своей системы уравнений, то эти решения игроков будут оптимальными по показателям вида (11) или (12), содержащим только компоненты, соответствующие этим объектам. Т.е. если один игрок будет менять свое решение при фиксированном решении второго игрока, т.е. по-другому распределять ресурсы между этими объектами, то он не улучшит значение своего показателя качества.

Лемма 2. Если игрок для некоторых объектов обеспечивает равную защищенность (незащищенность), то его решение имеет вид $\|1, 1, \dots, r_1, r_2, \dots, r_k, 0, \dots, 0\|$, где компоненты решения $r_1, r_2, \dots, r_k \in [0, 1]$ соответствуют объектам, для которых обеспечена равная защищенность (незащищенность). Решение имеет такой вид, если элементы вектора пронумерованы и порядке невозрастания коэффициентов полезности, т.е. при решении ЗЛП единицы вначале назначаются компонентам с большими коэффициентами полезности, далее оставшийся ресурс распределяется между объектами для обеспечения равной защищенности или незащищенности. Объекты, соответствующие компонентам r_1, r_2, \dots, r_k , имеют одинаковые коэффициенты полезности.

Таким образом, компоненты вектора защитника можно разбить на три непересекающихся множества: $N = N_3^{(1)} \cup N_3^{(r)} \cup N_3^{(0)}$, где $N_3^{(1)}$ – множество индексов компонент, которые равны 1 (может быть пустым), $N_3^{(0)}$ – множество индексов компонент, которые равны 0 (множество может быть пустым), $N_3^{(r)}$ – множество, состоящее как минимум из двух элементов, данные значения компонент подбираются для обеспечения равномерной защиты объектов (если множество $N_3^{(r)}$ состоит из 1-го элемента, то решение соответствует оптимизационному подходу).

Аналогично для нападающего компоненты решения также можно разбить на 3 непересекающихся множества: $N = N_n^{(1)} \cup N_n^{(r)} \cup N_n^{(0)}$. При этом $N_3^{(r)} = N_n^{(r)} = N^{(r)}$.

Теорема 1 (достаточные условия того, что решение является седловой точкой).

Если элементы решения защитника делятся на три непересекающихся множества $N = N_3^{(1)} \cup N_3^{(r)} \cup N_3^{(0)}$, таким образом, чтобы выполнялось:

- $k_{3i}(q_i) = k_{3j}(q_j), \forall i, j \in N_3^{(r)}$;
- если множество $N_3^{(1)}$ (компоненты решения равные 1) не является пустым, то $k_{3i}(q_i) \geq k_{3j}(q_j), \forall i \in N_3^{(1)}, \forall j \in N_3^{(r)}$;

- если множество $N_3^{(0)}$ (компоненты решения равные 0) не является пустым, то $k_{3i}(q_i) \geq k_{3j}(q_j), \forall i \in N_3^{(r)}, \forall j \in N_3^{(0)}$.

Аналогично, если элементы решения нападающего делятся на три непересекающихся множества $N = N_H^{(1)} \cup N_H^{(r)} \cup N_H^{(0)}$ ($N_3^{(r)} = N_H^{(r)}$), таким образом, чтобы выполнялось:

- $k_{Hi}(p_i) = k_{Hj}(p_j), \forall i, j \in N_H^{(r)}$;
- если множество $N_H^{(1)}$ (компоненты решения равные 1) не является пустым, то $k_{Hi}(p_i) \geq k_{Hj}(p_j), \forall i \in N_H^{(1)}, \forall j \in N_H^{(r)}$;
- если множество $N_H^{(0)}$ (компоненты решения равные 0) не является пустым, то $k_{Hi}(p_i) \geq k_{Hj}(p_j), \forall i \in N_H^{(r)}, \forall j \in N_H^{(0)}$.

Тогда это решение является седловой точкой игры.

Доказательство. Чтобы решение являлось седловой точкой необходимо, чтобы при фиксированном решении стороны нападения \vec{Q} вектор \vec{P} доставлял максимум показателю (11), а при фиксированном решении стороны защиты \vec{P} вектор \vec{Q} доставлял максимум показателю (12). Из алгоритма решения ЗЛП следует, что защитник будет присваивать значение единицы компонентам с наибольшими коэффициентами полезности, поэтому коэффициенты полезности защитника, соответствующие элементам множества $N_3^{(1)}$, будут максимальными. Среди компонент множества $N_3^{(r)}$ будет распределен оставшийся ресурс (после назначения 1 элементам множества $N_3^{(1)}$). Так как для элементов множества $N_3^{(r)}$ определен одинаковый коэффициент полезности, то защитнику все равно как распределить между этими объектами свой ресурс (он распределяет для равной защищенности этих объектов). Для компонент множества $N_3^{(0)}$ ресурса не хватает, их коэффициенты полезности принимают нулевые значения. Таким образом, решение защитника будет оптимальным. Аналогично, можно показать, что решение нападающего, удовлетворяющее условию теоремы, также будет оптимальным.

Используя результаты теоремы 1, сформулируем алгоритм поиска седловой точки или определения того факта, что в рамках данного подхода седловая точка не будет получена. Алгоритм базируется на неполном переборе вариантов, за счет исключения вариантов, которые не могут приводить к седловой точке или являются недопустимыми из-за ограничений. Алгоритм является рекурсивным. На каждом шаге алгоритма решаются системы уравнений вида (16) и (17) для обеспечения равной защищенности и (или) незащищенности некоторых объектов, индексы этих объектов принадлежат множеству $N^{(r)}$. Если решение является недопустимым, то индекс некоторого i -го объекта исключается из множества $N^{(r)}$. При этом варианты могут быть следующие:

- защитник включает индекс i в множество $N_3^{(1)}$, нападающий – в множество $N_H^{(1)}$, т.е. $p_i=1, q_i=1$;

- защитник включает индекс i в множество $N_3^{(0)}$, нападающий – в множество $N_H^{(0)}$, т.е. $p_i=0, q_i=0$;
- защитник включает индекс i в множество $N_3^{(0)}$, нападающий – в множество $N_H^{(1)}$, т.е. $p_i=0, q_i=1$;
- защитник включает индекс i в множество $N_3^{(1)}$, нападающий – в множество $N_H^{(0)}$, т.е. $p_i=1, q_i=0$, этот случай можно не рассматривать, т.к. при $q_i=0, k_{3i}(q_i) = 0$, защитник в этом случае не имеет стимула защищать данный объект.

При описании алгоритма переменные $p_i, q_i, \forall i \in N^{(r)}$, соответствующие объектам, индексы которых входят в множество $N^{(r)}$, будем называть свободными. Алгоритм содержит следующие шаги:

Шаг 0. Подготовка старта рекурсивного алгоритма. Все переменные в векторах \vec{P}, \vec{Q} считаются свободными, $N^{(r)} = N$, множества $N_3^{(1)} = N_3^{(0)} = N_H^{(0)} = N_H^{(1)} = \emptyset$.

Шаг 1. Старт рекурсивного алгоритма. На входе алгоритма: множество индексов свободных переменных $N^{(r)}$ в векторах \vec{P}, \vec{Q} , множества индексов элементов $N_3^{(1)}, N_3^{(0)}, N_H^{(0)}, N_H^{(1)}$.

Шаг 2. Решаем для свободных переменных в \vec{P}, \vec{Q} системы уравнений вида (16) и (17). Если полученное решение является допустимым, т.е. $p_i \in [0, 1], q_i \in [0, 1], \forall i \in N^{(r)}$, и выполняются условия теоремы 1, то завершение работы алгоритма, полученное решение является седловой точкой.

Шаг 3. Если число свободных переменных больше 2, то старт цикла просмотра свободных переменных. На каждой итерации цикла выполняются действия с i -й переменной из списка свободных. Шаги 4, 5, 6, 7 выполняются в этом цикле. Если число свободных переменных не больше 2, то завершение работы алгоритма.

Шаг 4. Проверяем можно ли индекс i -го объекта включить в множества $N_3^{(1)}, N_3^{(0)}, N_H^{(0)}, N_H^{(1)}$ так, чтобы потенциально не нарушались ограничения (7) и (8). Индекс i -го объекта можно включить в множество $N_3^{(0)}$, т.е. можно присвоить $p_i = 0$, если выполняется неравенство $\sum_{j \in N^{(r)} \cup N_3^{(1)}, j \neq i} c_{3j} \geq C_3^{(max)}$ (если неравенство не выполняется, то переменной нельзя присваивать 0, т.к. ресурсы не будут израсходованы, даже если все свободные переменные будут равны 1). Аналогично, индекс i -го объекта можно включить в множество $N_H^{(0)}$, т.е. можно присвоить $q_i = 0$, если выполняется неравенство $\sum_{j \in N^{(r)} \cup N_H^{(1)}, j \neq i} c_{Hj} \geq C_H^{(max)}$.

Индекс i -го объекта можно включить в множество $N_3^{(1)}$, т.е. можно присвоить $p_i = 1$, если выполняется неравенство $\sum_{j \in N_3^{(1)} \cup \{i\}} c_{3j} \leq C_3^{(max)}$ (если неравенство не выполняется, то переменной нельзя присваивать 1, т.к. ресурсов не хватит). Аналогично, индекс i -го объекта можно включить в множество $N_H^{(1)}$, т.е. можно присвоить $q_i = 1$, если выполняется неравенство $\sum_{j \in N_H^{(1)} \cup \{i\}} c_{Hj} \leq C_H^{(max)}$. Потенциальные возможности или

невозможности включения индекса в множества фиксируем с помощью переменных-флагов. Переходим к следующему шагу.

Шаг 5. Если индекс можно включить в множества $N_3^{(1)}$ и $N_H^{(1)}$ по ограничениям, то проверяем потенциальную возможность (или невозможность) включения индекса в эти множества так, чтобы не нарушались условия теоремы 1. Индекс i можно включить в множества $N_3^{(1)}$ и $N_H^{(1)}$, если при значениях $p_i = 1$ и $q_i = 1$ выполняются неравенства $k_{H_i}(p_i) \geq k_{H_j}(p_j), \forall j \in N_H^{(0)}$, (коэффициент полезности нападающего для i -го объекта должен быть не меньше коэффициентов полезности для объектов, для которых $q_j = 0$) и $k_{3_i}(q_i) \geq k_{3_j}(q_j), \forall j \in N_3^{(0)}$ (коэффициент полезности защитника для i -го объекта должен быть не меньше коэффициентов полезности для объектов, для которых $p_j = 0$). Если все неравенства выполняются, то индекс i исключаем из множества $N^{(r)}$, включаем его в множества $N_3^{(1)}$ и $N_H^{(1)}$ и вызываем рекурсивный алгоритм (см. шаг 1), на входе которого будут новые измененные множества. Если рекурсивный алгоритм не получил седловую точку, то индекс i исключаем из $N_3^{(1)}$ и $N_H^{(1)}$, включаем его в $N^{(r)}$ и переходим к следующему шагу. Если рекурсивный алгоритм получил седловую точку, то завершение работы алгоритма. Если индекс нельзя включить в указанные множества по ограничениям или по условиям теоремы 1, то переходим к следующему шагу.

Шаг 6. Если индекс можно включить в множества $N_3^{(0)}$ и $N_H^{(1)}$ по ограничениям, то проверяем потенциальную возможность (или невозможность) включения индекса в эти множества так, чтобы не нарушались условия теоремы 1. Индекс i можно включить в множества $N_3^{(0)}$ и $N_H^{(1)}$, если при значениях $p_i = 0$ и $q_i = 1$ выполняются неравенства $k_{H_i}(p_i) \geq k_{H_j}(p_j), \forall j \in N_H^{(0)}$, и $k_{3_i}(q_i) \leq k_{3_j}(q_j), \forall j \in N_3^{(1)}$ (коэффициент полезности защитника для i -го объекта должен быть не больше коэффициентов полезности для объектов, для которых $p_j = 1$). Если все неравенства выполняются, то индекс i исключаем из множества $N^{(r)}$, включаем его в множества $N_3^{(0)}$ и $N_H^{(1)}$ и вызываем рекурсивный алгоритм (см. шаг 1), на входе которого будут новые измененные множества. Если рекурсивный алгоритм не получил седловую точку, то индекс i исключаем из $N_3^{(0)}$ и $N_H^{(1)}$, включаем его в $N^{(r)}$ и переходим к следующему шагу. Если рекурсивный алгоритм получил седловую точку, то завершение работы алгоритма. Если индекс нельзя включить в указанные множества по ограничениям или по условиям теоремы 1, то переходим к следующему шагу.

Шаг 7. Если индекс можно включить в множества $N_3^{(0)}$ и $N_H^{(0)}$ по ограничениям, то проверяем потенциальную возможность (или невозможность) включения индекса в эти множества так, чтобы не нарушались условия теоремы 1. Индекс i можно включить в множества $N_3^{(0)}$ и $N_H^{(0)}$, если при значениях $p_i = 0$ и $q_i = 0$ выполняются неравенства $k_{H_i}(p_i) \leq k_{H_j}(p_j), \forall j \in N_H^{(1)}$, (коэффициент полезности нападающего для i -го объекта должен быть не больше коэффициентов полезности для объектов, для которых $q_j = 1$) и

$k_{3i}(q_i) \leq k_{3j}(q_j), \forall j \in N_3^{(1)}$. Если все неравенства выполняются, то индекс i исключаем из множества $N^{(r)}$, включаем его в множества $N_3^{(0)}$ и $N_H^{(0)}$ и вызываем рекурсивный алгоритм (см. шаг 1), на входе которого будут новые измененные множества. Если рекурсивный алгоритм не получил седловую точку, то индекс i исключаем из $N_3^{(0)}$ и $N_H^{(0)}$, включаем его в $N^{(r)}$ и переходим к следующему шагу. Если рекурсивный алгоритм получил седловую точку, то завершение работы алгоритма. Если индекс нельзя включить в указанные множества по ограничениям или по условиям теоремы 1, то переходим к следующему шагу.

Шаг 8. Завершение цикла просмотра свободных переменных, завершение работы алгоритма.

При этом существует возможность, что седловой точки получено не будет, так как она может не существовать или достаточные условия, сформулированные в теореме 1, не обязательно являются необходимыми. В этом случае при принятии решения можно воспользоваться принципом гарантированного результата со стороны защиты.

2.4 Подход к поиску решения стороной защиты, основанный на принципе гарантированного результата

Если седловой точки не существует (или она не найдена в рамках подхода, обеспечивающего равную защищенность объектов), то возможен подход к поиску решения стороной защиты, основанный на принципе гарантированного результата. Также этот подход можно использовать в эвристической форме и в том случае, если размер ресурсов нападающего неизвестен стороне защиты. При этом можно за основу взять рекурсивный алгоритм, предложенный в пп. 2.3.

Защитник может использовать различные критерии принятия решений в зависимости от информации, которая ему известна. В качестве критериев принятия решений защитником будем использовать оптимизацию некоторых показателей. Рассмотрим следующие критерии:

1. Минимизация показателя нападающего (12) при ограничении (7) в зависимости от полученного вектора \vec{P} , для каждого полученного допустимого \vec{P} необходимо решать ЗЛП для нападающего. Этот подход можно использовать, если известен ресурс нападающего, собственно, использование этого показателя и определяет гарантированный результат в рамках поставленной задачи. Если ресурс нападающего неизвестен, то защитник стремится минимизировать коэффициенты полезности объектов для нападающего. Следующие критерии подходят для этого случая и имеют эвристическую природу.
2. Минимизация максимального коэффициента полезности нападающего среди всех объектов:

$$F_1(\vec{P}) = \max_{i \in N} k_{Hi}(p_i) = \max_{i \in N} \frac{w_i - p_i p_{pr} w_i}{c_{Hi}} \rightarrow \min_{\vec{P}}. \quad (19)$$

3. Минимизация суммы коэффициентов полезности нападающего для всех объектов:

$$F_2(\vec{P}) = \sum_{i \in N} k_{hi}(p_i) = \sum_{i \in N} \frac{w_i - p_i p_{pr} w_i}{c_{hi}} \rightarrow \min_{\vec{P}}. \quad (20)$$

4. Максимизация суммы разностей коэффициентов полезности по сравнению со случаем, когда защитник не будет применять защиту объектов:

$$F_3(\vec{P}) = \sum_{i \in N} [k_{hi}(0) - k_{hi}(p_i)] = \sum_{i \in N} \frac{p_i p_{pr} w_i}{c_{hi}} \rightarrow \max_{\vec{P}}, \quad (21)$$

где $k_{hi}(0) = \frac{w_i}{c_{hi}}$ – значение коэффициента полезности нападающего в случае, если $p_i = 0$.

Можно использовать и другие критерии. Алгоритм построен на основе изложенного в пп. 2.3 алгоритма, основанного на принципе равной защищенности объектов, но из рассмотрения исключен вектор \vec{Q} , так же просматриваются все возможные варианты, в пп. 2.3. алгоритм завершал работу при нахождении седловой точки. Алгоритм включает в себя следующие шаги:

Шаг 0. Подготовка старта рекурсивного алгоритма. Все переменные в векторе \vec{P} считаются свободными, $N^{(r)} = N$, множества $N_3^{(1)} = N_3^{(0)} = \emptyset$.

Шаг 1. Старт рекурсивного алгоритма. На входе алгоритма: множество индексов свободных переменных $N^{(r)}$ в векторе \vec{P} , множества индексов $N_3^{(1)}$ и $N_3^{(0)}$.

Шаг 2. Решаем для свободных переменных в \vec{P} систему уравнений вида (16), если полученное решение является допустимым, т.е. $p_i \in [0, 1]$, то вычисляем значения показателей (12), решая ЗЛП для нападающего с ограничением (7), или вычисляем значения показателей вида (19), (20), (21). Если это первое допустимое решение, то решение запоминаем как рекордное, значение показателя как рекорд. Если решение не первое, то сравниваем значение показателя со значением рекорда. Если полученное значение показателя лучше рекорда, то новое решение запомнить как рекорд, переходим к следующему шагу.

Шаг 3. Если число свободных переменных больше 2, то старт цикла просмотра свободных переменных. Проверяем потенциальную возможность присвоения переменной значения $p_i = 0$, чтобы выполнялось ограничение (8), по аналогии с алгоритмом, описанном в пп. 2.3. Если такая возможность существует, то переходим к шагу 4, в противном случае, переходим к шагу 5. Если число свободных переменных не больше двух, то завершение работы алгоритма, рекордным решением является искомое решение.

Шаг 4. Переносим индекс i в множество $N_3^{(0)}$ из множества $N^{(r)}$. Вызываем рекурсивный алгоритм (см. шаг 1). После завершения алгоритма, переносим индекс i обратно в множество $N^{(r)}$ из $N_3^{(0)}$. Переходим к следующему шагу.

Шаг 5. Проверяем потенциальную возможность присвоения переменной значения $p_i = 1$, чтобы выполнялось ограничение (8). Если такая возможность существует, то переходим к шагу 6, в противном случае, переходим к шагу 7.

Шаг 6. Переносим индекс i в множество $N_3^{(1)}$ из множества $N^{(r)}$. Вызываем рекурсивный алгоритм (см. шаг 1). После завершения алгоритма, переносим индекс i обратно в множество $N^{(r)}$ из $N_3^{(1)}$. Переходим к следующему шагу.

Шаг 7. Завершение цикла просмотра свободных переменных, завершение работы алгоритма.

Когда алгоритм завершает работы, то искомое решение будет сохранено в качестве рекордного решения.

3. Пример решения задачи

Рассмотрим практический пример решения задачи для информационной системы с 10 объектами: 3 сервера, на которых хранятся данные разных типов, 7 автоматизированных рабочих мест (АРМ). Исходные данные для системы представлены в табл. 3.

Таблица 3. Исходные данные для задачи

№ п/п	Обозначение объекта защиты	Название параметров			
		Стоимость защищаемых ресурсов, $w_i, \forall i \in N$, в у.е.	Стоимость полной защиты объекта, $c_{zi}, \forall i \in N$, в у.е.	Стоимость полной атаки на объект, $c_{ni}, \forall i \in N$, в у.е.	Вероятность предотвращения атаки при полной защите $p_{пр i}, \forall i \in$ N
1.	Сервер1	5 000 000	500 000	50 000	0,95
2.	Сервер2	4 000 000	450 000	40 000	0,90
3.	Сервер3	3 000 000	400 000	40 000	0,90
4.	АРМ1	1 000 000	100 000	10 000	0,90
5.	АРМ2	1 000 000	100 000	10 000	0,90
6.	АРМ3	1 000 000	100 000	10 000	0,90
7.	АРМ4	800 000	100 000	10 000	0,85
8.	АРМ5	700 000	90 000	10 000	0,85
9.	АРМ6	600 000	90 000	10 000	0,85
10.	АРМ7	500 000	80 000	10 000	0,85

Из табл. 3 следует, что на полную защиту всех объектов необходимо 2 010 000, будем считать, что защитник имеет ресурсов на 1 000 000 у.е. На проведение полных атак нарушителю требуется 200 000 у.е., будем считать, что нарушитель имеет 100 000 у.е.

Алгоритмы реализованы на языке Си++ в среде Microsoft Visual Studio 2013. При заданных исходных данных поиск по оптимизационному алгоритму не привел к получению седловой точки. Алгоритм, основанный на равномерной защите объектов, привел к получению седловой точки, полученное решение представлено в табл. 4. Также в этой таблице представлены результаты проверки полученной седловой точки $\langle \vec{P}^*, \vec{Q}^* \rangle$, т.е. решение первым игроком своей ЗЛП при фиксированном векторе \vec{Q}^* и решение вторым игроком своей ЗЛП при фиксированном векторе \vec{P}^* (использовался алгоритм, описанный в пп. 2.1). Результаты проверки показали, что полученное решение действительно является седловой точкой (каждый игрок в отдельности не может улучшить значение своего показателя качества). Далее в табл. 4 представлено решение, гарантирующее результат 1-му игроку, в качестве показателя использовался показатель второго игрока, первый игрок стремился его минимизировать, решение совпало с седловой точкой. Рядом представлено оптимальное решение для второго игрока и значение его показателя качества.

Таблица 4. Результаты решения задачи

№ п/п	Решение стороны защиты, вектор \vec{P}	Значение показателя стороны защиты	Решение стороны нападения, вектор \vec{Q}	Значение показателя стороны нападения
Седловая точка:				
	0,508 0,536 0,651 0,536 0,536 0,536 0,415 0,307 0,162 0,000	4566221,317	0,481 0,571 0,406 0,507 0,507 0,507 0,672 0,691 0,806 0,000	5175836,209
Проверка седловой точки (каждый игрок решает свою оптимизационную задачу):				
	1,000 0,244 0,000 0,000 1,000 1,000 1,000 1,000 0,000 0,000	4566221,317	1,000 0,000 0,250 1,000 1,000 1,000 0,000 0,000 1,000 0,000	5175836,209
Решение, гарантирующее результат, стороне защите и оптимальное решение стороны нападения:				
	0,508 0,536 0,651 0,536 0,536 0,536 0,415 0,307 0,162 0,000	—————	1,000 0,000 0,250 1,000 1,000 1,000 0,000 0,000 1,000 0,000	5175836,209

Определим некоторые параметры предложенных алгоритмов на основе тестов, в которых исходные данные формируются с помощью генератора псевдослучайных чисел. Рассмотрим сколько раз оптимизационный алгоритм, описанный в пп. 2.2, приводил к получению седловой точки в результате 1000 испытаний, и сколько требовалось шагов алгоритма (на каждом шаге решается ЗЛП), прежде чем было получено повторное решение, и как эти параметры зависят от размерности задачи. Зависимость параметров оптимизационного алгоритма от размерности задачи представлена в табл. 5.

Таблица 5. Зависимость параметров оптимизационного алгоритма от размерности задачи (число испытаний 1000)

Параметры алгоритма	Размерность задачи, n (число объектов)										
	10	11	12	13	14	15	16	17	18	19	20
Получена седловая точка, число раз	122	91	88	86	72	74	68	55	52	48	53
Получена седловая точка, в процентах	12,2	9,1	8,8	8,6	7,2	7,4	6,8	5,5	5,2	4,8	5,3
Минимальное число шагов	4	4	4	4	4	4	4	4	4	4	4
Среднее число шагов	8,358	8,418	8,692	8,808	8,982	8,984	9,258	9,276	9,388	9,628	9,556
Максимальное число шагов	30	26	26	38	32	30	50	40	40	42	36
Минимальное число шагов в случае, когда была получена седловая точка	4	4	4	4	4	4	4	4	4	4	4
Среднее число шагов в случае, когда была получена седловая точка	4,475	4,549	4,568	4,512	4,528	4,459	4,647	4,400	4,615	4,458	4,642
Максимальное число шагов в случае, когда была получена седловая точка	8	6	6	6	6	8	6	6	6	6	6
Минимальное число шагов в случае, когда не была получена седловая точка	6	6	6	6	6	6	6	6	6	6	6
Среднее число шагов в случае, когда не была получена седловая точка	8,897	8,805	9,090	9,212	9,328	9,346	9,594	9,560	9,650	9,889	9,831
Максимальное число шагов в случае, когда не была получена седловая точка	30	26	26	38	32	30	50	40	40	42	36

Рассмотрим теперь параметры алгоритма, основанного на принципе равной защищенности объектов и их зависимость от исходных данных. В табл. 6 представлены следующие параметры: число раз, когда алгоритм получал седловую точку в результате 100 испытаний, а также время работы алгоритма (минимальное, среднее и максимальное), полученное в результате 100 испытаний (когда $n=12$, число испытаний равно 10). Времена работы алгоритма получены на ноутбуке с процессором *Intel*® *Core*™ i5-2410, тактовая частота 2,3 ГГц.

Таблица 6. Зависимость параметров алгоритма равномерной защиты от размерности задачи

Параметры алгоритма	Размерность задачи, n (число объектов)							
	5	6	7	8	9	10	11	12
Получена седловая точка, количество раз среди 100 испытаний (при размерности 12 среди 10 испытаний)	65	52	47	50	47	51	48	4
Получена седловая точка, в процентах	65,0	52,0	47,0	50,0	47,0	51,0	48,0	40,0
Минимальное время работы при 100 испытаниях (при размерности 12 среди 10 испытаний), с.	0,00	0,00	0,00	0,00	0,01	0,17	0,83	0,296
Среднее время работы при 100 испытаниях (при размерности 12 среди 10 испытаний), с.	0,01	0,01	0,02	0,06	0,45	3,16	39,63	829,00
Максимальное время работы при 100 испытаниях (при размерности 12 среди 10 испытаний), с.	0,04	0,04	0,06	0,23	1,98	11,20	166,02	2461,74

Определим параметры алгоритма, работающего по принципу гарантированного результата. Рассмотрим случай, когда седловая точка найдена алгоритмом, описанным в пп. 2.3. В табл. 7 представлены следующие параметры: сколько раз алгоритм получил решение, совпадающее с седловой точкой; сколько раз было найдено худшее решение для стороны защиты (по критерию минимизации показателя (12)); сколько раз было найдено лучшее решение (по критерию минимизации показателя (12)). В табл. 7 приведена зависимость перечисленных параметров от использования разных показателей вида (8), (19)-(21), проведена серия из 1000 тестов при размерности задачи $n=7$.

Таблица 7. Зависимость параметров алгоритма, основанного на принципе гарантированного результата, от используемых показателей

Параметры алгоритма				
	Показатель (12)	Показатель (19)	Показатель (20)	Показатель (21)
Число раз, когда решение совпало с седловой точкой	1000	397	0	0
Число раз, когда решение было лучше, чем седловая точка	0	0	0	0
Число раз, когда решение было хуже, чем седловая точка	0	603	1000	1000

Заключение

В статье рассмотрены следующие алгоритмы: алгоритм поиска седловой точки на основе оптимизационного подхода; алгоритм поиска седловой точки на основе принципа обеспечения равной защищенности некоторых объектов; алгоритм поиска решения на основе принципа гарантированного результата. Практическое тестирование алгоритмов на исходных данных, полученных с помощью генераторов псевдослучайных чисел, показало следующее:

- алгоритм поиска седловой точки на основе оптимизационного подхода позволил получить седловую точку в 5-12 % случаях, при увеличении размерности задачи это значение имеет тенденцию к уменьшению, алгоритм имеет полиномиальную трудоемкость, число шагов алгоритма практически не увеличивалось при увеличении размерности задачи;
- алгоритм поиска седловой точки на основе принципа обеспечения равной защищенности некоторых объектов позволил получить седловую точку примерно в 50 % случаях, алгоритм имеет экспоненциальную трудоемкость, причем разброс минимальной и максимальной длительностей решения задач при одинаковой размерности задачи является весьма существенным (2-3 порядка);
- алгоритм поиска решения на основе принципа гарантированного результата при использовании исходного показателя, задающего ущерб защитника, на примерах, когда седловая точка найдена предыдущим алгоритмом, позволил получить решение, совпадающее с седловой точкой, в 100 % случаев. В случае неполной информации о ресурсах нападающего и при использовании эвристических показателей лучший результат, в 39 % случаях совпадающий с седловой точкой, был получен в случае минимизации максимального коэффициента полезности для нападающего среди всех объектов. Если использовать такие показатели, как сумма коэффициента полезности нападающего для всех объектов и сумма разностей коэффициентов полезности нападающего по сравнению случаем, когда защита не применяется, полученные результаты были существенно хуже.

Список литературы

1. Yanwei Wang, Yu F.R., Tang H., Minyi Huang. A Mean Field Game Theoretic Approach for Security Enhancements in Mobile Ad hoc Networks // IEEE Transactions on Wireless Communications. 2014. Vol. 13, no. 3. P. 1616-1627. DOI: [10.1109/TWC.2013.122313.131118](https://doi.org/10.1109/TWC.2013.122313.131118)
2. Xiannuan Liang, Yang Xiao. Game Theory for Network Security // IEEE Communications Surveys & Tutorials. 2013. Vol. 15, iss. 1. P. 472-486. DOI: [10.1109/SURV.2012.062612.00056](https://doi.org/10.1109/SURV.2012.062612.00056)

3. Theodorakopoulos G., Baras J.S. Game Theoretic Modeling of Malicious Users in Collaborative Networks // IEEE Journal on Selected Areas in Communications. 2008. Vol. 26, iss. 7. P. 1317-1327. DOI: [10.1109/JSAC.2008.080928](https://doi.org/10.1109/JSAC.2008.080928)
4. Alpcan Tansu, Buchegger S. Security Games for Vehicular Networks // IEEE Transactions on Mobile Computing. 2011. Vol. 10, iss. 2. P. 280-290. DOI: [10.1109/TMC.2010.146](https://doi.org/10.1109/TMC.2010.146)
5. Freudiger J., Manshaei M.H., Hubaux J. P., Parkes D.C. Non-Cooperative Location Privacy // IEEE Transactions on Dependable and Secure Computing. 2013. Vol. 10, iss. 2. P. 84-98. DOI: [10.1109/TDSC.2012.85](https://doi.org/10.1109/TDSC.2012.85)
6. Lelarge M. Coordination in Network Security Games: A Monotone Comparative Statics Approach // IEEE Journal on Selected Areas in Communications. 2012. Vol. 30, iss. 11. P. 2210-2219. DOI: [10.1109/JSAC.2012.121213](https://doi.org/10.1109/JSAC.2012.121213)
7. Shafiee S., Ulukus Sennur. Mutual Information Games in Multiuser Channels With Correlated Jamming // IEEE Transaction on Information Theory. 2009. Vol. 55, iss. 10. P. 4598-4607. DOI: [10.1109/TIT.2009.2027577](https://doi.org/10.1109/TIT.2009.2027577)
8. Liang Xiao, Yan Chen, Lin W.S., Liu K.J.R. Indirect Reciprocity Security Game for Large-Scale Wireless Networks // IEEE Transactions on Information Forensics and Security. 2012. Vol. 7, iss. 4. P. 1368-1380. DOI: [10.1109/TIFS.2012.2202228](https://doi.org/10.1109/TIFS.2012.2202228)
9. Ma C.Y.T., Yau D.K.Y., Rao N.S.V. Scalable Solutions of Markov Games for Smart-Grid Infrastructure Protection // IEEE Transactions on Smart Grid. 2013. Vol. 4, iss. 1. P. 47-55. DOI: [10.1109/TSG.2012.2223243](https://doi.org/10.1109/TSG.2012.2223243)
10. Yongle Wu, Liu K.J.R. An Information Secrecy Game in Cognitive Radio Networks // IEEE Transactions on Information Forensics and Security. 2011. Vol. 6, iss. 3. P. 831-842. DOI: [10.1109/TIFS.2011.2144585](https://doi.org/10.1109/TIFS.2011.2144585)
11. Lin Chen, Leneutre J. A Game Theoretical Framework on Intrusion Detection in Heterogeneous Networks // IEEE Transactions on Information Forensics and Security. 2009. Vol. 4, iss. 2. P. 165-178. DOI: [10.1109/TIFS.2009.2019154](https://doi.org/10.1109/TIFS.2009.2019154)
12. Абденов А.Ж., Заркумова Р.Н. Выбор средства эффективной защиты с помощью методов теории игр // Вопросы защиты информации. 2010. № 2. С. 26-31.
13. Лаврентьев А.В., Зязин В.П. О применении методов теории игр для решения задач компьютерной безопасности // Безопасность информационных технологий. 2013. № 3. С. 19-24.
14. Быков А.Ю., Алтухов Н.О., Сосенко А.С. Задача выбора средств защиты информации в автоматизированных системах на основе модели антагонистической игры // Инженерный вестник МГТУ им. Н.Э. Баумана. Электрон. журн. 2014. № 4. С. 525-542. Режим доступа: <http://engbul.bmstu.ru/doc/708106.html> (дата обращения 01.06.2015).

The Algorithms of Resource Distribution for Information Security Between Objects of an Information System Based on the Game Model and Principle of Equal Security of Objects

A.Yu. Bykov^{1,*}, E.S. Shmatova¹

* abykov@bmstu.ru

¹Bauman Moscow State Technical University, Moscow, Russia

Keywords: information security, zero-sum game, linear programming, saddle point, Wald's maximin model

The modern methods to solve the practical tasks related to designing the information security systems in automated systems of different purpose rather often use models of the theory of games. The work, for resource distribution between securable objects, formulates a task of the two players' game (a defender and a forward) with the zero-sum. The possible loss of a defender is used as a score. The statement of problem for each player, when holding the other player's solution, is the task of linear programming with one restriction. For the solution of similar tasks the paper offers a linear programming method.

For a defender to search a solution, it is offered to use a saddle-point criterion. If the saddle point is not found, a criterion of the guaranteed result is used. For search of the saddle point, two algorithms are offered. Originally, an optimization algorithm, based on the sequential solution of the optimization tasks by each player, seeks the saddle point. If the first algorithm does not find the saddle point, there is a developed search algorithm of the saddle point based on the support of equal security of some objects of a securable system. For the support of equal security of some objects an offer is to solve a system of linear equations. For algorithm are formulated and proved sufficient conditions that the received solution is the saddle point.

If both algorithms did not find the saddle point, for a defender to obtain the solution, using a principle of guaranteed result is offered. The search algorithm of solution has been developed to provide the guaranteed result based on the algorithm of support of equal security in which it is possible to use the initial quality score or, in case there is no information about forward's resources, the heuristic scores.

Results of the task solution of resource distribution with the given basic data, as well as test results of the developed algorithms are provided.

References

1. Yanwei Wang, Yu F.R., Tang H., Minyi Huang. A Mean Field Game Theoretic Approach for Security Enhancements in Mobile Ad hoc Networks. *IEEE Transactions on Wireless Communications*, 2014, vol. 13, no. 3, pp. 1616-1627. DOI: [10.1109/TWC.2013.122313.131118](https://doi.org/10.1109/TWC.2013.122313.131118)
2. Xiannuan Liang, Yang Xiao. Game Theory for Network Security. *IEEE Communications Surveys and Tutorials*, 2013, vol. 15, iss. 1, pp. 472-486. DOI: [10.1109/SURV.2012.062612.00056](https://doi.org/10.1109/SURV.2012.062612.00056)
3. Theodorakopoulos G., Baras J.S. Game Theoretic Modeling of Malicious Users in Collaborative Networks. *IEEE Journal on Selected Areas in Communications*, 2008, vol. 26, iss. 7, pp. 1317-1327. DOI: [10.1109/JSAC.2008.080928](https://doi.org/10.1109/JSAC.2008.080928)
4. Alpcan Tansu, Buchegger S. Security Games for Vehicular Networks. *IEEE Transactions on Mobile Computing*, 2011, vol. 10, iss. 2, pp. 280-290. DOI: [10.1109/TMC.2010.146](https://doi.org/10.1109/TMC.2010.146)
5. Freudiger J., Manshaei M.H., Hubaux J, pp., Parkes D.C. Non-Cooperative Location Privacy. *IEEE Transactions on Dependable and Secure Computing*, 2013, vol. 10, iss. 2, pp. 84-98. DOI: [10.1109/TDSC.2012.85](https://doi.org/10.1109/TDSC.2012.85)
6. Lelarge M. Coordination in Network Security Games: A Monotone Comparative Statics Approach. *IEEE Journal on Selected Areas in Communications*, 2012, vol. 30, iss. 11, pp. 2210-2219. DOI: [10.1109/JSAC.2012.121213](https://doi.org/10.1109/JSAC.2012.121213)
7. Shafiee S., Ulukus Sennur. Mutual Information Games in Multiuser Channels With Correlated Jamming. *IEEE Transaction on Information Theory*, 2009, vol. 55, iss. 10, pp. 4598-4607. DOI: [10.1109/TIT.2009.2027577](https://doi.org/10.1109/TIT.2009.2027577)
8. Liang Xiao, Yan Chen, Lin W.S., Liu K.J.R. Indirect Reciprocity Security Game for Large-Scale Wireless Networks. *IEEE Transactions on Information Forensics and Security*, 2012, vol. 7, iss. 4, pp. 1368-1380. DOI: [10.1109/TIFS.2012.2202228](https://doi.org/10.1109/TIFS.2012.2202228)
9. Ma C.Y.T., Yau D.K.Y., Rao N.S.V. Scalable Solutions of Markov Games for Smart-Grid Infrastructure Protection. *IEEE Transactions on Smart Grid*, 2013, vol. 4, iss. 1, pp. 47-55. DOI: [10.1109/TSG.2012.2223243](https://doi.org/10.1109/TSG.2012.2223243)
10. Yongle Wu, Liu K.J.R. An Information Secrecy Game in Cognitive Radio Networks. *IEEE Transactions on Information Forensics and Security*, 2011, vol. 6, iss. 3, pp. 831-842. DOI: [10.1109/TIFS.2011.2144585](https://doi.org/10.1109/TIFS.2011.2144585)
11. Lin Chen, Leneutre J. A Game Theoretical Framework on Intrusion Detection in Heterogeneous Networks. *IEEE Transactions on Information Forensics and Security*, 2009, vol. 4, iss. 2, pp. 165-178. DOI: [10.1109/TIFS.2009.2019154](https://doi.org/10.1109/TIFS.2009.2019154)
12. Abdenov A.Zh., Zarkumova R.N. Choice of means of effective protection by the instrumentality of game theory methods. *Voprosy zashchity informacii = Information security questions*, 2010, no. 2, pp. 26-31. (in Russian).

13. Lavrent'ev A. V., Zjazin V. P. Applying the Game Theory Methods for Solving Problems of Computer Security. *Bezopasnost informacionnyh tehnologij*, 2013, no. 3, pp. 19-24. (in Russian).
14. Bykov A. Ju., Altuhov N. O., Sosenko A. S. The problem of choosing means of information protection in automated information systems based on the model of an antagonistic game. *Inzhenernyi vestnik MGTU im. N.E. Baumana = Engineering Herald of the Bauman MSTU*, 2014, no. 4, pp. 525-542. Available at: <http://engbul.bmstu.ru/doc/708106.html>, accessed 01.06.2015. (in Russian).