Теорема 3. Пусть бесконечное множество точек $S \subseteq \mathbb{R}^k$, $k \geqslant 2$, лежащих в общем положении, обладает следующим свойством: существует непрерывная замкнутая плоская кривая, содержащая S и ограничивающая выпуклое плоское множество. Тогда сужение любого протокола распределения укрытий между $n \geqslant 2$ участниками на множество S не может быть ограниченным. Кроме того, если множество S более чем счётно, то сужение любого протокола распределения укрытий между $n \geqslant 2$ участниками на множество S не может быть ограниченным по сообщениям.

В рамках исследования криптографического аспекта задачи RinS будем считать, что входные данные (координаты роботов и укрытий) берутся из некоторого конечного множества точек в пространстве $\mathbb{R}^k$, $k \geqslant 2$, все координаты которых — числа, заданные конечным числом разрядов в какой-либо (фиксированной) позиционной системе счисления. Заметим, что в этом случае смысл протокола распределения укрытий состоит в вычислении некоторой функции координат роботов и укрытий, значением которой является искомое распределение. В силу теоремы о конфиденциальных вычислениях (Oblivious Transfersecure multi-party computation) [5, 6] существует способ вычисления этой функции, при котором участники не получат никаких дополнительных сведений о входных данных друг друга.

Теорема 4. Пусть $S \subset \mathbb{R}^k$ — произвольное конечное множество точек, все координаты которых — числа с фиксированным числом разрядов в некоторой фиксированной позиционной системе счисления. Тогда

1) существует сужение на множество S основанного на простом щелчке протокола распределения укрытий, в котором агенты не сообщают друг другу свои координаты;
2) существует сужение на множество S основанного на щелчке со сравнениями протокола распределения укрытий, в котором агенты не сообщают друг другу свои расстояния до укрытий.

ЛИТЕРАТУРА

1. *Таненбаум Э., ван Стеен М.* Распределённые системы. Принципы и парадигмы. СПб.: Питер, 2003.
2. *Тель Ж.* Введение в распределённые алгоритмы. М.: МЦНМО, 2009.
3. *Wooldridge M.* An Introduction to Multiagent Systems. John Willey & Sons Ltd, 2002.
4. *Бодин Е. В., Гаранина Н. О., Шилов Н. В.* Задача о роботах на Марсе (мультиагентный подход к задаче Дейкстры) // Моделирование и анализ информационных систем. 2011. Т. 18. № 2. С. 113–128.
5. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си. М.: Триумф, 2002.
6. *Goldreich O.* Foundations of Cryptography — A Primer // Foundations and Trends in Theoretical Computer Science. 2005. V. 1. No. 1. P. 1–116.

УДК 004.056.53

ПОИСК МОСТОВ В МОДЕЛИ БЕЗОПАСНОСТИ TAKE-GRANT

Д. М. Бречка

В настоящее время для разработки и исследования защищённых компьютерных систем широко используются дискреционные модели безопасности. Такие модели основаны на субъектно-объектном подходе, согласно которому система представляется

в виде активных сущностей (субъектов) и пассивных сущностей (объектов), при этом множество субъектов является подмножеством множества объектов ($S \subseteq O$). Дискреционные модели безопасности в явном виде задают разрешённые (запрещённые) правоотношения для каждой пары субъект — объект компьютерной системы.

Разновидностью дискреционных моделей безопасности является модель Take-Grant [1 – 3]. В этой модели система правоотношений содержит два особых права: *take* (t) — брать права на объект (субъект) у другого объекта (субъекта) и *grant* (g) — давать права на объект (субъект) другому объекту (субъекту). Сама компьютерная система представляется в виде ориентированного графа (графа доступов), в котором вершинами являются субъекты и объекты, а дугами — права доступов, установленные между соответствующими субъектами и объектами. Анализ информационной безопасности в модели Take-Grant проводится путем исследования графа доступов и поиска в нем определенных структур. Примеры таких структур — острова и мосты.

Определение 1. Островом в графе доступов называется подграф, состоящий из вершин-субъектов, причём эти вершины соединены между собой дугами, содержащими права *take* либо *grant* (направление дуг не учитывается).

Определение 2. Мостом в графе доступов называется проходящий по вершинам-объектам путь, каждая дуга которого содержит право *take* или *grant*, при этом словарная запись пути имеет вид $\overrightarrow{t}^*$, $\overleftarrow{t}^*$, $\overrightarrow{t}^*\overrightarrow{g}\overleftarrow{t}^*$ либо $\overrightarrow{t}^*\overleftarrow{g}\overleftarrow{t}^*$.

Согласно [1], отыскание данных структур в графе доступов необходимо для анализа информационной безопасности, однако в классической модели Take-Grant методы поиска не описаны. Ниже приводится алгоритм поиска мостов, если острова в графе уже найдены.

Пусть в графе доступов $G = (V, E)$ (где $V = O$ — множество вершин, E — множество дуг) уже известны острова и необходимо найти мост между островами $I_1 \subset S$ и $I_2 \subset S$. Будем искать мост между вершинами $s \in I_1$ и $f \in I_2$. Опишем алгоритм поиска моста, основанный на поиске в глубину [4].

Введем шесть цветов для раскраски вершин: *красный*, *зелёный*, *синий*, *белый*, *серый* и *чёрный*. Пусть в начале работы алгоритма все вершины графа G окрашены в *белый* цвет. Окраску вершины будем обозначать $c(x)$.

В ходе работы алгоритма требуется временно запоминать некоторые посещённые вершины. Для этих целей удобнее всего использовать память, организованную как стек: первым пришёл — последним вышел.

Утверждение 1. Алгоритм 1 корректно распознает наличие моста в графе доступов и заканчивает свою работу за конечное число шагов, независимо от того, существует мост в графе или нет.

Доказательство. Покажем, что алгоритм закончит свою работу, если в графе не существует моста. Число вершин и дуг в графе не меняется в ходе работы алгоритма. Алгоритм просматривает каждую вершину не более одного раза, это гарантируется проверкой условия на шагах 5, 6 и 7 (просматриваются только белые вершины). Если на некотором шаге рекурсии алгоритм не обнаружит для текущей вершины смежной с ней вершины u, которую еще можно посетить (шаг 8), то алгоритм окрашивает текущую вершину в *чёрный* и возвращается к предыдущему шагу рекурсии. Таким образом, если в графе не существует моста, то за некоторое число шагов вершина s будет окрашена в *чёрный* цвет и алгоритм остановит свою работу. При этом будет просмотрено не более $|O|$ вершин.

Алгоритм 1. Поиск моста

1: Сделать текущей вершину s.
2: Окрасить текущую вершину в *серый*.
3: **Если** вершина f не *белая*, **то**
завершить алгоритм — мост существует.
4: **Для всех** вершин u, смежных с текущей, выполнить:
5: **Если** $c(u) =$ *белый* и текущая вершина *серая* или *красная*, и существует дуга $\overrightarrow{t}$ от текущей вершины до u, **то**
окрасить u в *красный* и перейти на шаг 11.
6: **Если** $c(u) =$ *белый* и текущая вершина *серая* или *красная*, и существует дуга $\overleftarrow{g}$ или дуга $\overrightarrow{g}$ от текущей вершины до u, **то**
окрасить u в *зелёный* и перейти на шаг 11.
7: **Если** $c(u) =$ *белый* и текущая вершина *серая* или *зелёная*, или *синяя*, и существует дуга $\overleftarrow{t}$ от текущей вершины до u, **то**
окрасить u в *синий* и перейти на шаг 11.
8: В остальных случаях окрасить текущую вершину в *чёрный*.
9: **Если** $c(s) =$ *чёрный*, **то**
завершить алгоритм — моста не существует,
10: **иначе**
сделать текущей первую вершину из стека, перейти на шаг 3.
11: Положить текущую вершину в стек, сделать текущей вершину u, перейти на шаг 3.

Предположим теперь, что в графе существует мост типа $\overrightarrow{t}^*$. Такой мост будет распознан путём последовательного выполнения шага 5. Все вершины, входящие в этот мост, будут окрашены в *красный* цвет.

Аналогично, если в графе существует мост типа $\overleftarrow{t}^*$, то он распознаётся путём последовательного выполнения шага 7, и все вершины, входящие в этот мост, будут окрашены в *синий* цвет. Мосты $\overrightarrow{t}^*\overrightarrow{g}\overleftarrow{t}^*$ и $\overrightarrow{t}^*\overleftarrow{g}\overleftarrow{t}^*$ распознаются путем последовательного выполнения шагов 5, 6 и 7. ■

Разработка алгоритма поиска мостов является важным вкладом в развитие классической модели Take-Grant. Представленный алгоритм позволяет решать задачу поиска моста между известными островами графа доступов за полиномиальное время и может быть применен для автоматического анализа безопасности компьютерных систем.

ЛИТЕРАТУРА

1. *Lipton R. J. and Richard J.* A Linear Time Algorithm for Deciding Subject Security // J. ACM. 1977. No. 3. P. 455–464.
2. *Девянин П. Н.* Модели безопасности компьютерных систем: учеб. пособие для студентов высших учебных заведений. М.: Академия, 2005. 144 с.
3. *Гайдамакин Н. А.* Разграничения доступа к информации в компьютерных системах. Екатеринбург: Изд-во Урал. ун-та, 2003. 328 с.
4. *Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К.* Алгоритмы: построение и анализ. М.: МЦНМО, 2000. 960 с.