УДК 681.3

Защита программного обеспечения от нелегального использования с помощью сетей Петри

И.А. Кочаровский, А.Н. Гамова
Саратовский национальный исследовательский государственный университет имени Н.Г. Чернышевского, г. Саратов

С использованием нелегального программного обеспечения можно бороться различными методами: юридическими, экономическими и, наконец, некоторые производители прибегают к защите программного обеспечения от взлома и нелегального копирования, что выдвигает на первый план задачу создания эффективной системы защиты программного продукта. Математический аппарат сетей Петри дает возможность построения ключей для защиты программного продукта.

Сеть Петри представляет собой двудольный ориентированный граф, состоящий из позиций, переходов, а также дуг, связывающих позиции с переходами и наоборот. В позициях могут быть размещены метки, которые перемещаются по сети. Сами же метки нужны для работы сети Петри, которая происходит посредством запуска переходов.

Сеть Петри, имеющая в своих позициях метки, называется маркированной сетью Петри. Переход разрешен тогда и только тогда, когда каждая его входная позиция имеет число меток, по крайней мере, равное числу исходящих дуг из данной позиции в переход. Запуск перехода заключается в удалении меток из его входных позиций и образовании новых меток в его выходных позициях.

В работе будут рассматриваться маркированные сети Петри с приоритетами. В таких сетях переходам будут задаваться приоритеты, означающие, что если могут одновременно выполниться несколько переходов, то запуститься переход с наивысшим приоритетом.

Для реализации системы защиты программного обеспечения от нелегального использования с помощью сетей Петри разработаем пример сети Петри с 4-битовым входом. В приведенной ниже сети Петри решающий переход t_2 будет выполнен тогда и только тогда, когда начальная маркировка помечает позиции $p_0, \dots, p_3$ следующим образом: $p_0 = 1,\ p_1 = 1,\ p_2 = 0,\ p_3 = 1$. Нарушение начальной маркировки хотя бы в одной из позиций приведет к недостижимости решающего перехода. Рассмотрим работу сети Петри, представленную на рисунке 1.

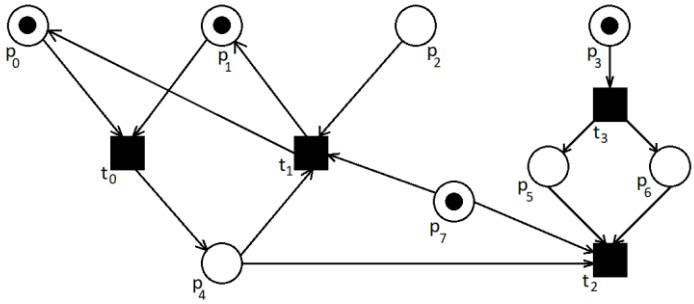

Рисунок 1 – Сеть Петри, кодирующая двоичное число 1101

Данная сеть Петри может быть использована тогда, когда, например, пользователю необходимо зарегистрировать программный продукт, посредством ввода 4-битного ключа. В этом случае позициями для начальной маркировки, куда будет записываться ключ, являются $p_0, \dots, p_3$. Отметим, что позиция p_7 всегда имеет одну метку, а остальные позиции при начальной маркировке не имеют меток. Решающим является переход t_2, то есть программа будет считаться зарегистрированной, если переход t_2 сработает. Зададим решающему переходу t_2 приоритет ниже, чем переходу t_1. Это означает, что если одновременно разрешены переходы t_1 и t_2, то сработает переход t_1.

Исследовав сеть Петри, представленную на рисунке 1, легко заметить, что для того, чтобы найти правильный ключ необходимо перебрать всего лишь 16 значений. Это можно сделать даже вручную.

На сегодняшний день, использование ключей, длина которых меньше 256 бит, является недостаточным для обеспечения защиты. Следовательно, предложенная на рисунке 1 сеть Петри не годится для использования на практике. Чтобы увеличить длину ключа при использовании сетей Петри, приведенную на рисунке 1 сеть Петри можно наращивать. Под этим понимается следующее: к каждой позиции, которую можно пометить при начальной маркировке, присоединяется новая сеть. При этом позиция теряет возможность быть помеченной при начальной маркировке, однако, появляются новые позиции, которые доступны для начальной маркировки. Например, нарастим сеть над позицией p_0. Для этого достаточно просто копировать уже имеющуюся сеть. На рисунке 2 представлена сеть Петри, полученная после наращивания уже имеющейся сети.

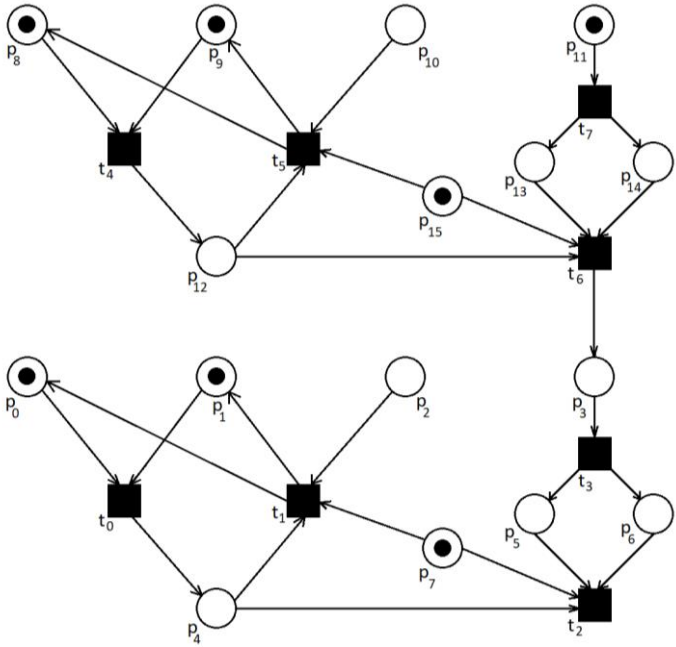

Рисунок 2 – Пример наращивания сети Петри над позицией p_0

При этом переход t_2 остается решающим, и появляются новые позиции $p_8, \dots, p_{11}$, которые можно пометить при начальной маркировке. Легко видеть, что переход t_6 фактически выполняет ту же роль, что и переход t_2 в нижней части сети. Очевидно, что для срабатывания перехода t_2 необходимо срабатывание перехода t_6, а для этого начальной маркировкой необходимо пометить позиции следующим образом: $p_0 =$ зом: $p_0 = 1,\ p_1 = 1,\ p_2 = 0, p_8 = 1,\ p_9 = 1,\ p_{10} = 0,\ p_{11} = 1.$ Длина ключа увеличилась до 7 бит. Теперь для полного перебора необходимо осуществить уже 128 проверок. Надстроив сеть аналогичным образом над позициями $p_0,\ p_1$ и p_2, легко получить ключ длиной 16 бит, для полного перебора необходимо рассмотреть 2^{16} комбинаций, чего также не достаточно. В общем случае размер ключа легко нарастить до 128 бит, в этом случае вопрос о переборе 2^{128} значений потеряет свою актуальность. При дальнейшем наращивании длина ключа достигнет 256 бит, что можно считать достаточным для надежной защиты.

Логично выяснить, будут ли сети, надстроенные над позициями $p_0, \dots, p_3$ одинаковыми. Проще всего, ответить на этот вопрос, исследуя сеть для 16-битного ключа. В этом случае единственная позиция,

которая не поддается наращиванию по аналогии с позицией p_0 – это позиция p_2. Это очевидно, так как для достижимости решающего перехода t_2 необходимо, чтобы позиция p_0 была пуста. Если нарастить над ней копию уже имеющейся сети, то ее нулевое значение достижимо при 15 различных комбинациях. Это означает, что диапазон возможных ключей сужается. Поэтому для позиции p_2 следует изменить надстроенную сеть. Одним из вариантов может явиться наращивание, представленное на рисунке 3.

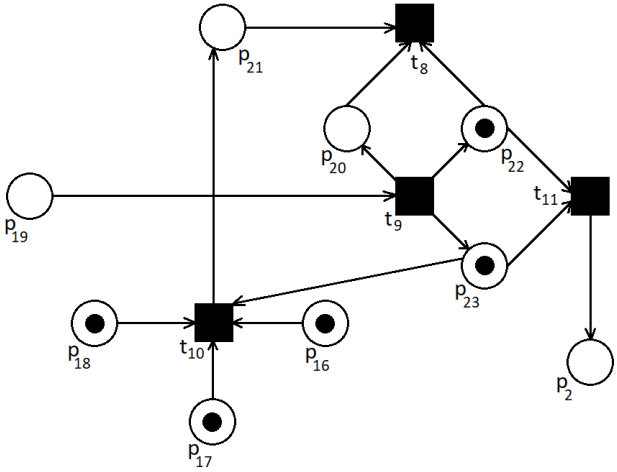

Рисунок 3 – Наращивание сети Петри над позицией p_2

Здесь для того, чтобы переход t_{11} не сработал, а это именно то, чего нам нужно добиться, необходимо назначить ему наименьший приоритет, а начальной маркировкой задать значения: $p_{16} = 1,\ p_{17} = 1,\ p_{18} = 1,\ p_{19} = 0$.

Итак, в общем случае для рассмотренной сети Петри исходная маркировка содержит ключ в позициях $p_0, \dots, p_{k-1}$, где k – длина ключа. Одна или несколько входных позиций гарантированно содержат метки при начальной маркировке. После того, как будет введен правильный ключ, сеть Петри выполнится несколько раз до достижения ключевого перехода. При этом маркировка сети Петри изменится. Если введен неверный ключ, то решающий переход не будет достигнут.