УДК 519.710.73

ЗАЩИТА ПРОГРАММНЫХ СРЕДСТВ С ИСПОЛЬЗОВАНИЕМ СЕТЕЙ ПЕТРИ

канд. техн. наук, доц. О.Е. ШЕСТОПАЛОВА, С.А. СКРИПЛЁНОК
(Полоцкий государственный университет)

Статья посвящена разработке и анализу алгоритмов сетей Петри, используемых для защиты программных средств. Приведена характеристика алгоритмов простых временных, приоритетных и ингибиторных сетей Петри, используемых для реализации лицензионной защиты программных средств. Выполнен анализ методики построения защиты программных средств с использованием алгоритмов данных разновидностей сетей Петри. Рассмотрен аналог и предложена альтернативная реализация элемента защиты программных средств в виде трех вариантов сети Петри для ввода и проверки 4-х битной части ключа защиты, отличающихся от аналога дополнительной функциональностью, препятствующей попыткам подбора ключа защиты случайным перебором, и использованием структурной избыточности, усложняющей аналитический подбор ключа. Осуществлено моделирование и анализ вариантов реализации элементов защиты, результаты которого подтверждают перспективность использования алгоритмов сетей Петри для реализации технологической защиты программных средств. Задачи разработки и исследования алгоритмов защиты решались с использованием авторского программного обеспечения PetriNets InProject.

К технологической защите своей интеллектуальной собственности разработчики лицензионных программных средств вынуждены прибегать ввиду масштабного распространения пиратского использования программных продуктов, с одной стороны, и низкой эффективностью юридических и экономических методов борьбы с этим явлением – с другой. Технологии защиты программных систем постоянно эволюционируют. Компании-разработчики программных средств используют целый ряд эмпирических подходов для создания систем лицензионной безопасности: аппаратные ключи, активацию через Интернет, серийные номера, методы борьбы с отладкой и декомпилированием и проч. Одним из таких современных подходов является использование моделей, методов и алгоритмов обеспечения лицензионной безопасности на основе сетей Петри. Перспективность этой методологии подтверждается ее использованием в таких программных продуктах, как Антивирус Касперского (разработчик – ЗАО «Лаборатория Касперского») и InfoWatch Enterprise Solution (разработчик – ЗАО «Инфовотч») [1, 2].

Сеть Петри представляет собой ориентированный граф, содержащий позиции, определяющие условия, имеющиеся в системе, и переходы, отображающие связанные с этими условиями действия. В позициях проставляются метки, если соответствующее условие выполнено. Позиции соединяют дугой с переходом, если выполнение заданного условия является необходимым для запуска связанного с данным переходом действия. Переход соединяют дугой с позицией, если связанное с ним действие порождает выполнение условия, представленного данной позицией. Последовательность изменения состояний сети определяется продвижением меток и сменой маркировок позиций, что в свою очередь определяется правилами срабатывания переходов. В простой временной сети Петри:

- срабатывает только активный переход, т.е. такой, во всех входных позициях которого имеются метки;
- срабатывание перехода наступает через заданный конечный промежуток времени после его активизации, причем если возникает конфликт (одновременная активизация двух и более переходов, имеющих общие входные позиции), то срабатывает равновероятно только один из конфликтных переходов;
- в результате срабатывания перехода число меток в каждой входной позиции уменьшаются на единицу, а число меток во всех выходных позициях увеличивается на единицу [3].

Аппарат сетей Петри как инструмент описания и исследования динамики функционирования асинхронных, распределенных, параллельных, недетерминированных и/или стохастических систем широко применяется для исследования систем защиты информации. Управление условиями срабатывания переходов позволяет моделировать процессы преодоления защиты. Кроме того, аппарат сетей Петри позволяет формализовать процесс исследования эффективности систем защиты информации [4]. Однако широкий спектр средств в аппарате сетей Петри, позволяющих моделировать условные действия со сколь угодно сложными условиями, открывает достаточно широкие перспективы использования сетей Петри и для реализации средств защиты программных средств.

Известные алгоритмы реализации такой защиты строятся на построении сложной структуры сети Петри путем наращивания элементарных фрагментов [1, 5]. Наращивание структуры эквивалентно увеличению длины ключа. В каждом элементарном фрагменте единственный, так называемый ключевой, переход срабатывает только при условии истинности вводимой части ключа защиты. Ввод части ключа

задается начальной маркировкой доступных для маркирования позиций. Понятно, что к срабатыванию ключевого перехода должна приводить одна единственная правильная маркировка элементарного фрагмента. Так, автор [5] предлагает в качестве примера реализации 4-х битной части ключа сеть Петри (рис. 1).

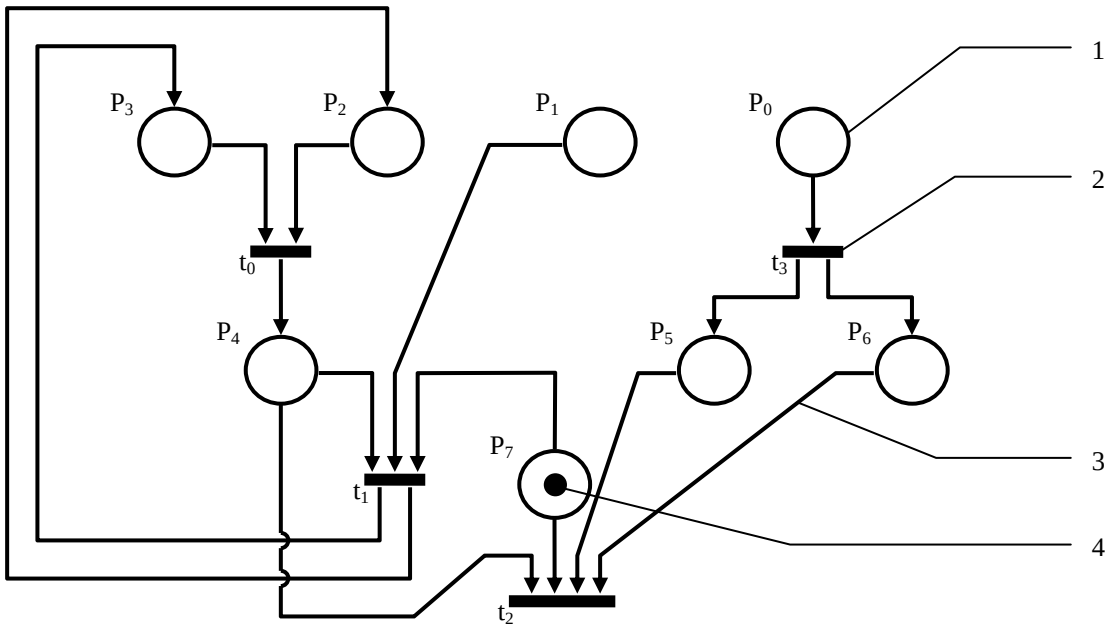


Рис. 1. Элементарный фрагмент сети Петри для ввода и проверки 4-х битной части ключа защиты:
1 – позиция; 2 – переход; 3 – дуга; 4 – метка

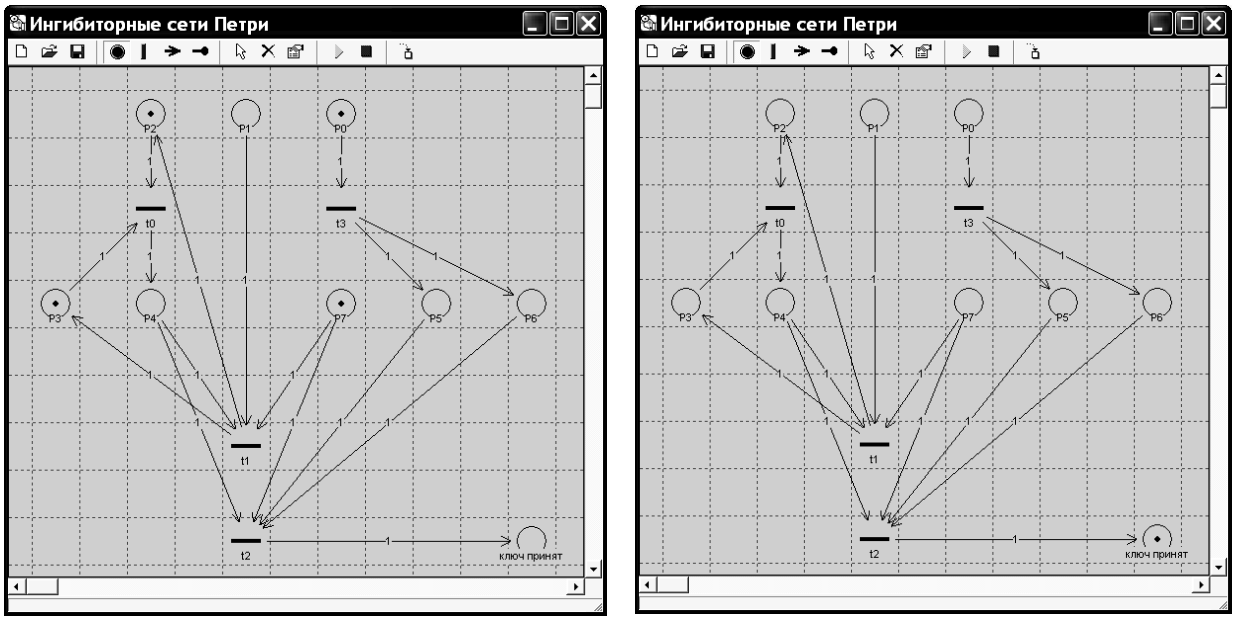
В сети Петри, представленной на рисунке 1, для маркировки внешнему пользователю доступны только позиции $P_0 \ldots P_3$. Установленная в позиции метка эквивалентна установке соответствующего бита части ключа защиты в значение 1, в противном случае значение бита равно 0. Ключевым переходом, срабатывание которого эквивалентно принятию программой защиты введенной части ключа, в данной сети Петри является переход t_2, имеющий наименьший приоритет в условиях конфликта. Это означает, что при одновременной активизации переходов t_1 и t_2, имеющих общую входную позицию P_7, сработает переход t_1, а не t_2. По утверждению разработчика данного фрагмента, единственной допустимой (ключевой) маркировкой доступных пользователю позиций является следующая маркировка: $P_0 = 1$, $P_1 = 0$, $P_2 = 1$, $P_3 = 1$. При всех остальных допустимых начальных маркировках переход t_2 не сработает [5].

Следует отметить, что сеть Петри (см. рис. 1) решает поставленную задачу избыточными средствами. Простую проверку правильности 4-х битной части кода можно было бы реализовать одним (единственным) ключевым переходом, срабатывающим при правильной маркировке позиций $P_0 \ldots P_3$. Однако повышение избыточности элементарного фрагмента сети Петри путем введения дополнительных позиций и неключевых переходов повышает степень защиты от взлома, так как в отличие от программ, защищенных последовательными методами, идентификация ключевого перехода в сети Петри, обладающей достаточной избыточностью, невозможна, потому что при неправильной маркировке он просто недостижим [5]. В программной реализации такой защиты каждый переход сети Петри будет являться потоком, который проверяет набор бит на входе и в зависимости от результата устанавливает биты на выходе. Так, поток t_3 и проверяет метки (биты) в позиции P_0, и в зависимости от наличия меток (бит установлен) или их отсутствия (бит сброшен) устанавливает биты в выходных позициях P_5 и P_6. То есть для анализа поведения сети необходимо отслеживать N потоков, где N – количество переходов. Если при этом код потоков равномерно распределен в коде защищаемой программы, а для усложнения идентификации неключевых переходов последние реализованы не бесконечными циклами, а с использованием дополнительных выходов по условию, то взлом существенно осложняется. Подбор верной комбинации перебором в рассматриваемом фрагменте сложностей не представляет, так как возможных вариантов всего $2^4 = 16$. Однако, как уже отмечалось, приведенная сеть является только элементарным фрагментом. Полная структура защиты будет включать требуемое для заданной длины ключа количество подобных фрагментов. Уже при использовании четырех таких фрагментов количество вариантов становится равным $2^{16} = 65536$, а современные системы защиты требуют, как правило, реализации ключей не менее 128 бит.

Проверим предлагаемое решение путем реализации и тестирования фрагмента сети Петри, приведенного на рисунке 1, в программном обеспечении PetriNets InProject. Программное обеспечение моделирования сетей Петри разрабатывалось в УО «Полоцкий государственный университет» на кафедре технической кибернетики (кафедра вычислительных систем и сетей) с 2003 года. Последняя авторская версия этого программного обеспечения PetriNets InProject позволяет:

- создание и редактирование моделей простых временных и ингибиторных сетей Петри;
- исследование динамики изменения состояний (маркировок) в автоматическом режиме или режиме пошагового прогона [3, 6].

Реализация исследуемого фрагмента сети Петри в PetriNets InProject и результат прогона модели для проверки ключевой маркировки приведены на рисунке 2. Дополнительная позиция «ключ принят» добавлена в исследуемый фрагмент для наглядности демонстрации срабатывания ключевого перехода t_2: появление метки в этой позиции возможно только после срабатывания t_2.



а) б)

Рис. 2. Реализация сети Петри для ввода и проверки 4-х битной части ключа защиты в PetriNets InProject: а – ввод корректного значения части ключа защите; б – конечная маркировка сети

Для всех переходов реализации сети Петри, представленной на рисунке 2, задано время срабатывания, равное 1 секунде. Поэтому после запуска моделирования с начальной маркировкой, показанной на рисунке 2, а, активизированные этой маркировкой переходы t_0 и t_3 сработают синхронно, в результате чего метки появятся в позициях $P_4 \ldots P_7$. Это приводит к активизации ключевого перехода t_2, срабатыванию этого перехода и достижению конечной маркировки, показанной на рисунке 2, б.

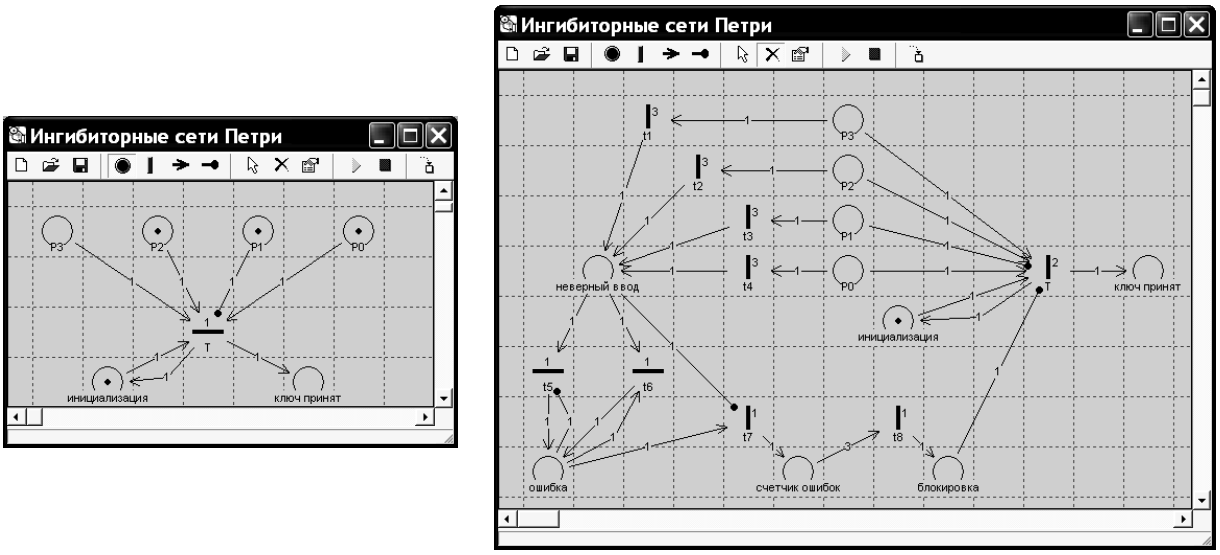
Исследование рассматриваемой реализации сети Петри для ввода и проверки 4-х битной части ключа защиты позволило установить следующее:

- из возможных 16 комбинаций значений 4-х битной части ключа защиты к срабатыванию ключевого перехода приводит только ключевая комбинация;
- из всех возможных неверных комбинаций к возникновению конфликта – одновременной активизации переходов t_1 и t_2 – приводит единственная комбинация, соответствующая маркировке ($P_0 = 1$, $P_1 = 1$, $P_2 = 1$, $P_3 = 1$).

Таким образом, конфликт переходов t_1 и t_2, для разрешения которого автор [5] предлагал введение различных приоритетов, действительно имеет место. При реализации в PetriNets InProject использование авторского способа разрешения конфликта невозможно, так как сегодняшняя версия программного обеспечения не позволяет назначать приоритеты срабатывания переходов, и конфликты разрешаются равновероятным срабатыванием любого из конфликтующих переходов. То есть при моделировании исследуемого варианта элементарного фрагмента сети Петри 4-х битной части ключа защиты в PetriNets InProject без учета приоритетов конфликтных переходов результаты будут недостоверны, так как при достаточном

объеме экспериментальной выборки модель будет демонстрировать принятие 4-х битной части ключа защиты в 50 % случаев ввода неверной комбинации вида (1, 1, 1, 1).

В арсенале средств PetriNets InProject имеется возможность моделировать ингибиторные сети Петри. Использование этой разновидности аппарата сетей Петри позволяет решить рассматриваемую задачу другим способом. В ингибиторных сетях Петри в дополнение к обычным дугам используются запрещающие, так называемые ингибиторные дуги. Такая дуга запрещает активацию и срабатывание перехода при наличии достаточного количества меток во входных позициях обычных дуг до тех пор, пока в ее входной позиции имеются метки в количестве, равном или большем веса дуги. На графах сетей Петри ингибиторные дуги, в отличие от обычных, отображаются линией с кружком на конце. Варианты реализации элементарного фрагмента сети Петри для ввода и проверки 4-х битной части ключа защиты, разработанные с использованием аппарата ингибиторных сетей Петри, приведены на рисунке 3.



а) б)

Рис. 3. Реализации ингибиторных сетей Петри для ввода и проверки 4-х битной части ключа защиты в PetriNets InProject: а – ключевой переход; б – сеть с функциональной избыточностью

Предлагаемая реализация элементарного фрагмента сети Петри для ввода и проверки 4-х битной части ключа защиты без введения избыточности, реализующая только ключевой переход (рис. 3, а), отличается от аналога, рассмотренного выше, следующими особенностями:

- для проверки истинности введенной части ключа защиты используется не пара конфликтных переходов, а один ключевой переход *T*, запрещаемый к активизации при попытке ввода неверного кода либо ингибиторной дугой, либо недостаточностью неверной маркировки для активизации;

- реализовано возобновление (инициализация) начальной маркировки сети Петри после срабатывания ключевого перехода для обеспечения возможности неоднократного использования элементарного фрагмента, например, при наличии повторяющихся последовательностей в коде ключа.

Маркировка, приведенная на рисунке 3, а, иллюстрирует попытку ввода неверного кода 4-х битной части ключа защиты. Видно, что ключевой переход *T* не активен, так наличие метки в позиции P_1 запрещает активизацию исходящей ингибиторной дугой. Единственной маркировкой, приводящей к активизации и срабатыванию ключевого перехода *T*, является правильное значение 4-х битной части ключа защиты: $P_0 = 1$, $P_1 = 0$, $P_2 = 1$, $P_3 = 1$, и наличии метки в позиции инициализации.

Вариант того же элементарного фрагмента, но реализованный с функциональной избыточностью, в частности с блокированием ключевого перехода при трехкратном введении неверной части кода (попытки подбора), приведен на рисунке, 3, б. Для разрешения конфликта одновременной активизации переходов t_1, t_2, t_4 с ключевым переходом *T* в случае введения правильной части ключа в этом варианте использованы разные времена срабатывания конфликтующих переходов, что допускается в последней версии PetriNets InProject, но в программной реализации защиты можно решить эту задачу и назначение ключевому переходу бóльшего приоритета. Введение неверного значения 4-х битной части ключа защиты вида (0, 0, 0, 0) данная сеть Петри игнорирует. Любая другая неверная комбинация приводит к срабатыванию от одного до четырех переходов $t_1 \ldots t_4$ и к появлению соответствующего количества меток в

позиции «неверный ввод». Поскольку любое количество меток в этой позиции означает попытку подбора части кода, такая маркировка должна в конечном итоге приводить к появлению одной метки в позиции «счетчик ошибок». Это реализовано следующим образом: любая ненулевая маркировка позиции «неверный ввод» приводит к активизации единственного перехода – t_5, и после его срабатывания одна метка изымается из позиции «неверный ввод» и появляется в позиции «ошибка». Если исходная метка была единственной, то это приводит к разрешению и срабатыванию перехода t_7, т.е. результат счета ошибочных попыток ввода увеличивается на 1: метка добавляется в позицию «счетчик ошибок». Если исходных меток в позиции «неверный ввод» было несколько, то переход t_7 запрещен к срабатыванию исходящей из этой позиции ингибиторной ветвью до удаления лишних для счета ошибок меток. Удаление организовано через открытый переход t_6. Чтобы срабатывание этого перехода стало возможным не ранее использования первой из меток неверного ввода для счета ошибок, входной позицией для этого перехода является также позиция «ошибка». Каждое срабатывание этого перехода удаляет из позиции «неверный ввод» одну метку с возвращением нужной для счета ошибок метки в позицию «ошибка». Удаление всех лишних для счета меток из позиции «неверный ввод» разрешает срабатывание перехода t_7, и метка счета перемещается в позицию «счетчик ошибок». Накопление в этой позиции трех меток счета активизирует переход t_8, так как выходная дуга позиции имеет вес, равный 3. В результате срабатывания t_8 метка, появившаяся в позиции «блокировка», запрещает к срабатыванию ключевой переход *T* вне зависимости от верности или неверности вводимой 4-х битной части кода, т.е. блокирует подбор кода перебором вариантов. Исследование данного варианта в режиме пошагового прогона в PetriNets InProject подтвердило соответствие функциональности приведенному описанию.

Заметим, что вариант реализации сети Петри для ввода и проверки 4-х битной части ключа защиты, показанный на рисунке, 3, отличает от аналога не только дополнительная функциональность, но и повышенная структурная сложность ввиду наличия бóльшего числа вспомогательных переходов с различными сложными условиями активизации, срабатыванием в цикле и проч. Однако структурную сложность, а значит, степень защиты от взлома можно дополнительно увеличить введением дополнительного избыточного числа позиций и переходов без нарушения уже реализованной функциональности. Пример реализации такой структуры в PetriNets InProject с неинформативными обозначениями позиций и переходов приведен на рисунке 4.

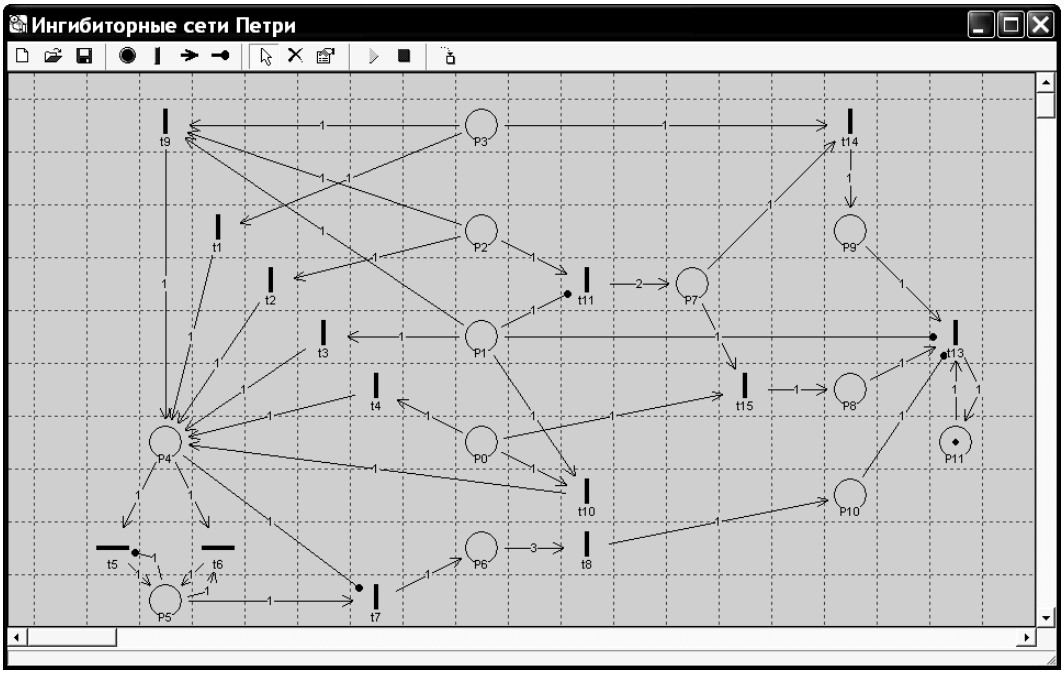


Рис. 4. Реализация ингибиторной сети Петри для ввода и проверки 4-х битной части ключа защиты в PetriNets InProject с повышением степени защиты путем введения структурной избыточности

Даже при заданной структуре (см. рис. 4), в отсутствие информации о доступности позиций для маркирования и в условиях, связанных с ними, о действиях, моделируемых переходами, и временах их срабатывания, анализ функционирования этой сети Петри для определения верной ключевой 4-х битной комбинации представляет существенные сложности. А с учетом возможности программной реализации

каждого узла сети отдельным потоком с собственным типом защиты степень эффективности предлагаемого решения можно оценить как достаточно высокую.

Таким образом, анализ разработанных сетей Петри для ввода и проверки части ключа защиты программного средства, а также их аналога позволяет утверждать, что аппарат сетей Петри предоставляет возможность реализации сколь угодно сложных средств защиты благодаря вариативности структур элементарных фрагментов, возможность их автоматического наращивания для увеличения размера ключа и введения в защиту информационной избыточности. Следует также добавить, что в данной статье рассматривалось использование конкретных видов сетей Петри – ингибиторных и приоритетных. За рамками приведенного материала остались прочие разновидности: вероятностные и цветные сети Петри, возможность использование которых для решения задач создания эффективной защиты программных средств также не вызывает сомнения.

ЛИТЕРАТУРА

1. Доля, А.В. Алгоритмы безопасного перехода в сетях Петри для лицензионной защиты программных систем: автореф. дис. … канд. техн. наук: 05.13.11 / А.В. Доля; Южный федер. ун-т. – Ростов н/Д, 2007. – 20 с.
2. Касперски, К. Приемы защиты исходных текстов и двоичного кода / К. Касперски // Открытые системы. – 2001. – № 7 – 8.
3. Скрипленок, С.А. Модификация программного обеспечения моделирования систем массового обслуживания ингибиторными сетями Петри / С.А. Скрипленок // Труды молодых специалистов Полоц. гос. ун-та. Вып. 31. Промышленность. – 2008. – С. 31 – 34.
4. Завгородний, В.И. Комплексная защита информации в компьютерных системах / В.И. Завгородний. – М.: Логос, 2001. – 263 с.
5. Schneider, T. Hardening Registration Number Protection Schemes against Reverse Code Engineering with Multithreaded Petri Nets. Talk at RECON2005 / T. Schneider // The Reverse Code Engineering Community: Scientific Board for Software Protection & Reverse Code Engineering & Damn Vulnerable Linux [Electronic resource]. – The talk at RECON2005. Mode of access: http://www.reverse-engineering.net. – Date of access: 07.02.2009.
6. Shestopalova, O.E. Ingproject – Petri Nets Software Tool for Queuing Systems Simulation / O.E. Shestopalova, S.A. Skriplenok // Проблемы проектирования и производства РЭС: сб. материалов междунар. науч.-техн. конф., Новополоцк, 29 – 30 мая. – Новополоцк: ПГУ, 2008. – С. 91 – 94.

Поступила 13.04.2009