щих, где это возможно, правила преобразования *take_right* или *grant_right* без явного построения множеств R^{tgo} и F^{tgo}.

Аналогично реализуется собственная часть алгоритма 3 функцией access-lock() с применением в каждом цикле встречающегося правила преобразования *rename_entity*, *access_read*, *access_write* или *access_append*.

Применяемые правила преобразования графа доступов реализованы функциями, изменяющими биты матрицы смежности и значения функции иерархии H в соответствии с определениями этих правил в [1].

С целью проверки корректности работы программы замыкания (ею является функция access-lock()) были проведены тесты на графах размером до 10 вершин. Данная реализация может обработать граф до 30000 вершин на обычных пользовательских компьютерах без дополнительных модификаций исходного кода, однако время ожидания может быть слишком большим.

В продолжение работы планируется разработать средства задания графов доступов, реализовать графическую подсистему представления графа доступов и увеличить эффективность работы программы.

ЛИТЕРАТУРА

1. *Девянин П. Н.* Анализ безопасности управления доступом и информационными потоками в компьютерных системах. М.: Радио и связь, 2006.
2. *Колегов Д. Н.* Применение ДП-моделей для анализа защищенности сетей // Прикладная дискретная математика. 2008. №1. С. 71–88.

УДК 004.432.2

ТЕХНОЛОГИЯ И ИНСТРУМЕНТАЛЬНАЯ СРЕДА СОЗДАНИЯ ЗАЩИЩЁННЫХ СИСТЕМ ОБРАБОТКИ ИНФОРМАЦИИ

Д. А. Стефанцов

Сообщается о разработке и реализации технологии и инструментальной среды создания систем обработки информации (СОИ), защищённых политиками безопасности (ПБ) от угроз нарушения целостности, конфиденциальности и доступности обрабатываемых данных. Результатом применения разработанной технологии является программная система, состоящая из двух подсистем — защищаемой СОИ и модуля, реализующего заданную ПБ. Разработанные инструментальные средства позволяют создавать СОИ и ПБ независимо друг от друга с последующим объединением их с помощью соединительных модулей, обновлять ПБ и вносить в неё изменения и дополнения без изменения защищаемой СОИ.

В основу разработанной технологии положено аспектно-ориентированное программирование (АОП) — способ объединения независимых программных подсистем как аспектов, написанных на базовом языке программирования (C, Pascal, Smalltalk и т. п.), соединённых модулями, написанными на метаязыке программирования [1]. Инструментальная среда состоит из трёх основных частей:

1) языка программирования AspectTalk, обладающего конструкциями базового языка программирования Smalltalk и метаязыковыми конструкциями, основанными на протоколах метаобъектов [2];
2) виртуальной машины, выполняющей операции над данными, описанными на языке AspectTalk;
3) транслятора с языка AspectTalk в язык виртуальной машины.

За основу языка программирования AspectTalk взят диалект Little Smalltalk [3] языка Smalltalk. В него введён специальный тип данных — *метакласс*, элементы которого являются классами классов и позволяют программисту давать виртуальной машине дополнительные указания о функционировании объектной системы. Эти указания определяются в виде обработчиков операции посылки сообщения от объекта к объекту и наследуются метаклассами в иерархии наследования метаклассов. Граф иерархии наследования метаклассов является подграфом графа иерархии наследования классов. Данный подход позволяет избавиться от недостатков АОП, описанных в [4], а также упростить семантическую модель языка. Подробнее об AspectTalk можно прочитать в [5].

Виртуальная машина имеет стековую архитектуру с разделением стека данных и стека вызова процедур. Набор команд виртуальной машины включает операции управления стеком данных, а также операции над примитивными типами данных. Язык программ для виртуальной машины представляет собой польскую инверсную запись последовательности команд. Для реализации транслятора с AspectTalk в язык виртуальной машины построена грамматика класса LL(2).

Использование предложенных методов и средств позволит снизить затраты на внесение изменений в политики безопасности, а также повторно использовать реализации политик безопасности при разработке новых программных систем. Научная новизна работы состоит в модификации аспектно-ориентированного подхода в программировании, состоящей в специализации средств метаязыка, а именно: в классическом подходе они используются при написании аспекта совместно с операциями объединения, в предложенном подходе — при написании только соединительных модулей. Последнее упрощает процедуру повторного использования аспектов: в классическом подходе это требует переписывания заново самого аспекта, в предложенном подходе — переписывания только соединительного модуля, который, как правило, много проще присоединяемого аспекта.

ЛИТЕРАТУРА

1. *Elrad T., Aksit M. M., Kiczales G., et al.* Discussing aspects of AOP // Communications of ACM. 2001. October. V. 44. No. 10. P. 33–38.
2. *Kiczales G.* The Art of Meta-Object Protocol. The MIT Press, 1991. 345 p.
3. *Budd T.* A Little Smalltalk. Addison-Wesley, 1987. 280 pp.
4. *Bouraqadi N., Seriai A., Leblanc G.* Towards unified aspect-oriented programming // ESUG 2005 Research Conference. 2005. 22 p.
5. *Стефанцов Д. А.* Реализация политик безопасности в компьютерных системах с помощью аспектно-ориентированного программирования // Прикладная дискретная математика. 2008. №1(1). С. 94–100.

УДК 681.511:3

ЦИФРОВЫЕ ВОДЯНЫЕ ЗНАКИ, УСТОЙЧИВЫЕ К АТАКЕ СГОВОРОМ

Р. С. Стружков, Т. М. Соловьёв, Р. И. Черняк

В настоящее время основным способом борьбы с пиратством в сфере цифровых технологий является внедрение в медиа-данные цифровых водяных знаков. В большинстве случаев водяной знак выбирается как произвольный идентификатор пользователя