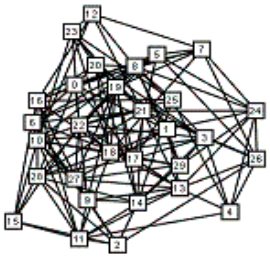
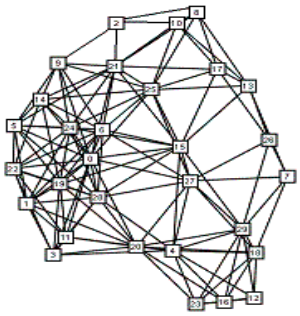


а) Граф типа RRG, p=0.33, $\bar{k} \approx 9.13$



а) Граф типа RGG, r_0=0.39, $\bar{k} \approx 9.46$

Рис. 3. Графы разных типов с одинаковой средней степенью вершин

Граф типа RGG обладает более регулярной структурой, граф RGG характеризуется наличием «малых миров» (сгущений). Эти особенности существенны при моделировании конкретных эпидемий сетевых червей.

Литература

1. Климентьев К.Е. Моделирование распространения и взаимодействия самовоспроизводящихся объектов [Текст] / К.Е. Климентьев // Известия Самарского научного центра РАН. - Самара: изд-во СНЦ РАН, 2014. – т.16 №4(2) - С. 313-317.

2. Райгородский А.М. Модели случайных графов и их применения [Текст]/ А.М. Райгородский // М.: Труды МФТИ. – 2010. – Том 2, №4. – С. 130-140.

3. Вентцель Е.С., Овчаров Л.А. Прикладные задачи теории вероятностей [Текст]/ Е.С. Вентцель [и др.]. – М.: Радио и связь, 1983. – 416 с.

4. Philip J. The probability distribution of the distance between two random points in a box [Текст]/ J. Philip // TRITA MAT 7(10), 2007. – 13 pp.

А.В. Козачок, Л.М. Туан

ОБОСНОВАНИЕ ВОЗМОЖНОСТИ ПРИМЕНЕНИЯ НЕРАЗЛИЧИМОЙ ОБФУСКАЦИИ ДЛЯ ЗАЩИТЫ ИСПОЛНЯЕМЫХ ФАЙЛОВ

(Академия ФСО России)

В настоящее время угрозы информационной безопасности, связанные с утечкой конфиденциальных данных, являются одними из наиболее опасных для любой организации, так как приводят к прямым материальным убыткам и ущербу, потере интеллектуальной собственности. По мере развития компьютерных технологий и услуг связи растет число возможных каналов утечки информации.

Целью проводимого исследования является построение системы защиты исполняемых файлов от несанкционированного доступа. Для достижения поставленной цели был выбран подход на основе обфускации программного кода.

В процессе решения задачи исследования необходимо подробно рассмотреть понятие обфускации, классификацию и стойкость существующих методов, а также провести их анализ.

Обфускацией программы называется преобразование ее кода, которое сохраняет функциональность, но изменяет ее содержимое таким образом, что извлечение из программного кода определенной информации об алгоритме и структурах данных, реализованных в программе, становится трудоемкой задачей [1].

Цель обфускации программного кода заключается в том, чтобы затруднить понимание и анализ программного кода и воспрепятствовать целенаправленной их модификации. В настоящее время исследование в области обфускации программного кода проводится по двум направлениям:

– с позиции системного программирования;

– с позиции криптографии.

Со стороны системного программирования обфускация программного кода может использоваться для защиты авторских прав на программное обеспечение, для предотвращения реинженерии программ, для создания и защиты водяных знаков, обеспечения безопасности мобильных агентов в информационных сетях, для проведения безопасного поиска в потоках данных и защиты баз данных. Однако существенным недостатком данного подхода является отсутствие гарантированного обоснования стойкости.

Со стороны математической криптографии разработка эффективных алгоритмов позволит решить целый ряд серьезных задач, например, с их помощью можно преобразовать криптосистемы с секретным ключом в криптосистемы с открытым ключом, проводить вычисления над зашифрованными данными, реализовать функциональное шифрование, доверенные схемы перешифрования и электронно-цифровой подписи, создать верифицируемые системы тайного голосования и схемы двусмысленного шифрования. При этом перечисленные приложения можно считать криптографически стойкими при условии, что стойкость используемых методов обфускации программ также будет доказана.

Большую роль в успешном внедрении указанных подходов к обфускации играют непроницаемые предикаты (opaque predicates), значения которых известны в процессе обфускации программы, но трудно вычисляются на этапе ее анализа. В ряде работ было проведено рассмотрение методов построения и оценки стойкости непроницаемых предикатов. Дополнительные возможности, усиливающие скрытность непроницаемых предикатов, открываются в случае обфускации распределенных программ [2].

Процедура обфускации может осуществляться:

– на уровне исходных текстов;

– на уровне промежуточного кода;

– на уровне машинных команд.

Реализация обфускации на уровне исходных кодов и на уровне промежуточного кода не предоставляет возможность контроля целостности обфусцированного программного кода и имеет привязку к какому-либо определенному

языку программирования. Обфускация, осуществляющаяся на уровне машинного кода позволяет осуществлять проверку целостности программного кода, но обладает существенным недостатком – низкой скоростью работы защищенного кода.

К обфускации программ предъявляются три главных требования:

– сохранение функциональности программы;

– полиномиальное замедление;

– требование стойкости.

Вероятностная машина Тьюринга *O* является обфускатором машины Тьюринга *(TM* обфускатором), стойким в модели «черного ящика», если он удовлетворяет следующим трем условиям [1].

1. Функциональная эквивалентность. Для всякой машины Тьюринга *(TM) M* любое выходное слово *O(M)* машины *O* на входе *M* вычисляет ту же функцию, что и *M.*

2. Полиномиальное замедление. Длина и время выполнения *O(M)* должна быть полиноминально больше аналогичных показателей для *M*. То есть существует полином *poly* такой, что $|O(M) < poly(|M|)|$ и, если *M* заканчивает свою работу через *t* шагов при определенном входе *x,* то *O(M)* оканчивает свою работу за *poly(t)* шагов при том же входе *x.*

3. Свойство виртуального черного ящика. Для любой полиномиальной вероятностной машины Тьюринга (PPT) *A* существует PPT *S* (симулятор) и пренебрежимо малая функция α такая, удовлетворяющие для любой машины Тьюринга *M* соотношению:

$$\left|\Pr\left[A(O(M))=1\right]-\Pr\left[S^{M}(1^{|M|})=1\right]\right|\le a(|M|)$$

где $\Pr$ – вероятность наступления некоторого события;

A – вероятностная машина Тьюринга;

$O(M)$ – запутанная машина Тьюринга *M;*

S^{M} – вероятностная машина Тьюринга, анализирующая входные и выходные данные машины Тьюринга *M* и не имеющая непосредственного доступа к $O(M)$.

Вероятностная машина Тьюринга *O* называется неразличимым обфускатором, если он удовлетворяет требованиям функциональности, эффективности, а также следующему требованию стойкости: для любой пары эквивалентных МТ *M,* M_0, имеющих одинаковый размер, распределения вероятностей случайных величин $O(M)$ и $O(M_0)$ вычислительно неотличимы, т.е. для любой PPT *T* справедливо соотношение:

$$\left|\Pr\left[T(O(M))=1\right]-\Pr\left[T(O(M_D))=1\right]\right|\le a(|M|)$$

Обфускация программ считается стойкой в модели виртуального «черного ящика», если всякий противник, имеющий неограниченный доступ к тексту обфусцированной программы, может извлечь из этого текста не больше информации, чем можно было бы получить, проводя одни лишь тестовые эксперименты с программой без доступа к ее тексту.

Применение неразличимой обфускации позволяет обеспечить выполнение свойства виртуального «черного ящика», таким образом становится возможным применение данного подхода для защиты исполняемых файлов от несанкционированного доступа.

Литература

1. Варновский Н.П., Захаров В.А., Кузюрин Н.Н., Шокуров А.В. Современное состояние исследований в области обфускации программ: определения стойкости обфускации // Труды Института системного программирования РАН (электронный журнал), том 26, № 3, с. 167-198.

2. Majumdar A., Thomborson C. Manufacturing opaque predicates in distributed systems for code obfuscation // *Proceedings of the 4th International Conference on Information Security*. Hobart, Tasmania, Australia, 2006, p. 187-196.

А.В. Линьков, М.Е. Гордеева

АНАЛИЗ ПРОБЛЕМЫ АНОНИМНОСТИ В ИНТЕРНЕТЕ: ПОЛЬЗОВАТЕЛЬ, ГОСУДАРСТВО, ЗЛОУМЫШЛЕННИК

(Самарский государственный университет)

Современная жизнь невозможна без использования информационных технологий. Одной из таких технологий является Интернет, который прочно вошел в современную жизнь. В некоторых ситуациях, пользователь желает остаться анонимным, с целью защиты личных данных, например, о своем местоположении, паспортных данных, состоянии банковского счета и тому подобном. Под *анонимностью* можно понимать недоступность вышеперечисленных данных для обеспечения личного комфорта и безопасности пользователя сети Интернет. Анонимность пользователя в сети в общем случае можно рассматривать через призму понятий информационной безопасности. В частности, проблемы анонимности имеют отношение к безопасности в сети. В последнее время информационной безопасности личности и ее взаимосвязи с информационной безопасностью государства в постиндустриальном обществе уделяется большое внимание [1-4]. Проблема анонимности тесно связана с проблемами свободы слова и печати [5], ограничения распространения информации ограниченного доступа, защиты детей от информации, причиняющей вред их здоровью и развитию, распространением экстремисткой информации, защиты информации и обеспечения безопасности информационных процессов в глобальных телекоммуникационных сетях, которые сейчас имеют не только теоретическое, но и большое практическое значение.

Проблемы анонимности в сети можно и нужно подвергать анализу с нескольких направлений. Во-первых, можно выделить правовую и техническую составляющие. Во-вторых, при анализе проблемы с выделенных точек зрения, анонимность в интернете требует учёта непосредственных интересов пользова-