протоколов, серверов приложений и операционных систем (ОС). На наш взгляд, основными угрозами информационной безопасности облачных вычислений являются [1]:

Несанкционированный доступ (НСД) к учетным записям или сервисам клиентов. Во время подключения пользователя к облачной среде между ним и облаком (возможно, еще и виртуальным сервером приложений) устанавливаются «доверительные отношения» посредством аутентификации пользователя и организации защищенного соединения с использованием криптографических средств.

Использование небезопасных программных интерфейсов (API), которые иногда предоставляют некоторые провайдеры конечным пользователям для реализации различных услуг облака.

Утечка данных. Поскольку доступ к облаку осуществляется исключительно по сети, необходимо обеспечивать сохранность и конфиденциальность хранимых данных (включая резервные копии), в том числе в процессе их передачи по сети.

Парирование угроз. Основными методами парирования выявленных угроз информационной безопасности облачных вычислений являются, на наш взгляд, криптозащита и технология DLP (технология зашиты от утечки данных) .

Для защиты от НСД должны использоваться криптографические средства обеспечения сохранности данных. Все данные, с которыми клиент работает в рамках сервиса, должны надежно шифроваться. Должна быть предусмотрена трёхуровневая криптозащита:

на уровне массивов данных;

на уровне приложений;

на уровне файлов.

Очень важную роль для криптозащиты в облаке играет жизненный цикл информации в нём. Этот цикл можно разделить на ряд основных этапов, каждый из которых должен иметь свои средства криптозащиты.

Для защиты от небезопасных программных интерфейсов целесообразен детальный аудит облачных провайдеров перед их выбором.

Для предотвращения утечки данных в облаке целесообразно использовать технологию DLP, которая является популярным инструментом защиты облачной информации. Следует отметить, что для применения технологии DLP в облаке или виртуальной среде может потребоваться специальная адаптация. Применение технологии DLP желательно совместить с новейшими продуктами для облачной безопасности, такими, например, как решения Security as a Service (безопасность как услуга) [2].

Список использованных источников:

1. Прузан А.Н., Сечко Г.В., Таболич Т.Г. Анализ угроз информационной безопасности облачных вычислений // Современные средства связи: материалы XVII Междунар. науч.-техн. конф., 16–18 сент. 2012 года, Минск, Респ. Беларусь / редкол.: А. О. Зеневич [и др.]. – Минск: УО ВГКС, 2012. – 332 с. – С. 236-237.

2. Граннеман Дж. Security as a Service: преимущества и риски на базе облака // Безопасность ИТ-инфраструктуры. – 2012. – № 11. – С. 9-14.

ПОДХОДЫ К СОЗДАНИЮ ПРОГРАММНОЙ МОДЕЛИ СИСТЕМЫ ЗАЩИТЫ РЕЧЕВОЙ ИНФОРМАЦИИ ОТ УТЕЧКИ ПО АКУСТИЧЕСКОМУ КАНАЛУ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Савченко И.В.

Давыдов Г.В. – канд. техн. наук, доцент

Для компьютерного моделирования системы защиты речевой информации от утечки по акустическому каналу целесообразно применять программную систему конечно-элементного анализа ANSYS с использованием оптимального количества параметров и показателей, необходимых для построения модели.

Современные достижения в области средств и методов защиты речевой информации, широкое применение информационных технологий и разнообразных средств связи, используемых для передачи информации, требуют разработки новых подходов и методов к решению данной проблемы, основанных на использовании компьютерных программных средств.

Речь, являясь универсальным инструментом человеческого общения, следует рассматривать как особый вид информационного сообщения, обладающего специфическими характеристиками, требующими применения специальных мер защиты. В этой связи существует необходимость компьютерного моделирования различных условий и параметров и их взаимосочетаний, определяющих степень защиты речевой информации, передаваемой по акустическому каналу, способствующих предупреждению ее утечки и обеспечивающих безопасность защищаемой информации.

Компьютерное моделирование различных ситуаций выполняется с использованием широко распространенных программных комплексов, которые представляют собой пакет виртуальных расчетных программ, позволяющих спроектировать конкретную ситуацию с использованием большого числа параметров

и показателей. При создании программной модели для решения задач защиты речевой информации целесообразно использовать программную среду ANSYS. ANSYS является универсальной программной системой конечно-элементного анализа, работающей на основе геометрического ядра Parasolid, которая может быть использована для решения широкого спектра задач, в том числе и задач акустики [1].

Для создания программной модели защиты речевой информации в среде ANSYS необходимо определение требуемого количества параметров и показателей, используемых для построения модели. К таким параметрам можно отнести размеры помещения, местоположение и размеры источника звука, плотность воздуха, тип среды и тип колебаний, скорость звука, частота звука и др. [2, 3]. Выбор параметров должен позволять модели определять оптимальные условия для предотвращения или минимизации утечки речевой информации за счет несанкционированного прослушивания или изменения речевого сообщения посредством модификации информации или индивидуальных особенностей говорящего. Разрабатываемая программная модель должна иметь возможность исследовать зависимость распространения речевой информации от различных физических характеристик акустической среды и их сочетаний, и уязвимости речевой информации при изменении какой-либо характеристики. Такие возможности программной модели обеспечиваются за счет определения общих принципов разбиения объектов на конечные элементы, поиска системы с оптимальными параметрами за счет моделирования с использованием различных вариантов разбиения объектов на конечные элементы, оценки результатов данного моделирования и выявления оптимального разбиения объектов на конечные элементы.

Возможности данной программы значительно упрощают процедуры создания требуемой модели и оценку результатов, позволяют использовать интерактивную графику для проверки геометрии модели. Вывод графической информации на экран способствует проведению контроля результатов проектных решений непосредственно в процессе работы. Применение данной программы позволяет избежать дорогостоящих и длительных циклов разработки и получить требуемые результаты в минимально короткие сроки.

Таким образом, разработка программной модели защиты речевой информации с помощью средств многоаспектного моделирования, реализованных в ANSYS, и оценка результатов моделирования на основании построенной модели позволит определить оптимальных методы защиты речевой информации и разработать рекомендации по защите речевой информации от утечки по акустическим каналам.

Список использованных источников:

1. Басов, К.А. ANSYS: справочник пользователя / К. А. Басов. – М.: ДМК Пресс, 2005. – 640 с.
2. Хорев, А. А. Методы защиты речевой информации и оценки их эффективности / А. А. Хорев, Ю. К. Макаров // Защита информации. Конфидент, 2001. – № 4. – С. 22-33.
3. Хорев, А. А. Оценка эффективности защиты информации от утечки по техническим каналам / А. А. Хорев // Специальная техника, 2006. – № 6. – С. 53-61.

ФУНКЦИОНАЛЬНОЕ ПРОГРАММИРОВАНИЕ В JVM

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Селивошко В. И., Шинкевич А. А.

Сечко Г. В. – канд. техн. наук, доцент

Рассматриваются возможности функционального программирования в рамках наиболее распространённых языков для JVM

1 Что такое JVM? Java Virtual Machine (*JVM*) — *виртуальная машина Java* — основная часть исполняющей системы Java, так называемой *Java Runtime Environment* (*JRE*). *JVM* интерпретирует *байт-код Java*, предварительно созданный из исходного текста *Java-программы компилятором Java* (*javac*). *JVM* может также использоваться для выполнения программ, написанных на других языках программирования (ЯП). Например, исходный код на языке *Ada* может быть откомпилирован в *байт-код Java*, который затем может выполниться с помощью *JVM*. *JVM* доступны для многих аппаратных и программных платформ, что позволяет описать *Java* как «скомпилировано однажды, запускается везде» (compile once, run anywhere) [1].

2 Краткий обзор доступных JVM ЯП. JVM, кроме непосредственно программ, написанных на Java, позволяет интерпретировать *байт-код Java*, полученный в процессе компиляции исходного кода одного из множества допустимых ЯП. Список наиболее часто используемых ЯП, доступных для запуска на *JVM* [2, 3]:

Clojure, функциональный диалект языка программирования Lisp;

Processing, объектно-ориентированный язык для создания изображений и анимации;

Groovy, объектно-ориентированный, скриптовый ЯП;

Scala, мультипарадигмальный ЯП, сочетающий в себе возможности как функционального, так и объектно-ориентированного программирования;

JRuby, реализации языка *Ruby*;

Jython, реализация языка *Python*;

Rhino, реализация языка *JavaScript*;

Kotlin, статически типизированный объектно-ориентированный ЯП компании JetBrains.