$\leqslant_f$, которое представляет рост (или убывание) субъективной оценки, т. е. основанной на качестве самого сообщения, между элементами $P(I)$. Это новое частичное упорядочение также дает полную решетку, и таким образом можно ввести понятие тройной решетки.

Определение тройной решетки находится в согласии с определением двойной решетки как множества с двумя частичными порядками, каждый из которых формирует решетку на этом множестве, т. е. генерируя собственные операторы пересечения и объединения, а также унарные операторы инверсии.

Грани, относящиеся к трем частичным порядкам, показаны в табл. 1.

Таблица 1

Грани, относящиеся к трем частичным порядкам

Относительный порядок	Грани	Наибольшие и наименьшие элементы в $P(I)$
$\leqslant_k$	A, N	Информационные
$\leqslant_s$	Tt, Ff	Конфиденциальные
$\leqslant_f$	TF, tf	Объективности

Полученная решетка имеет пять информационных уровней, пять уровней конфиденциальности и пять уровней объективной оценки, показанных в табл. 2.

Таблица 2

Уровни информированности, секретности и объективности

	Информированность	Секретность	Объективная оценка
1	N	Ff	TF
2	T, F, t, f	F, f, TFf, Ftf	T, F, TFt, TFf
3	TF, Tf, Tt, Ff, Ft, tf	A, TF, Tf, Ft, tf, N	A, Tt, Tf, Ft, Ff, N
4	TFf, TFt, Ttf, Ftf	T, t, TFt, Ttf	t, f, Ttf, Ftf
5	A	Tt	tf

На практике это позволяет разместить важные системные файлы в нижней части иерархии модели Белла и Лападулы. За счет правила «нет записи вниз» осуществляется защита целостности от троянских коней. Указанный подход можно использовать при разграничении доступа для составных субъектов и объектов, а также для разработки общей политики безопасности системы, составленной из отдельных политик безопасности для ее подмножеств.

УДК 519.8

АНАЛИЗ ЗАЩИЩЕННОСТИ СЕТИ С ИСПОЛЬЗОВАНИЕМ ЦЕПЕЙ МАРКОВА

В. П. Кушнир, И. Н. Кирко

Входными параметрами информационных систем являются информационные потоки, контролируемые техническими, программными и организационными средствами. Результаты анализа прохождения информационных потоков по телекоммуника-

ционным и вычислительны сетям, например программно-аппаратные сбои, несанкционированные операции, внешние и внутренние атаки, нерегламентируемые действия персонала сети, преднамеренное искажение данных и т. д., фиксируются в специальных файлах (журналах).

Основные задачи анализа методов, систем и механизмов обеспечения заданной защищенности данных при их обработке в информационных системах электронного документооборота — построение моделей возникновения ошибок и угроз, их взаимодействия; получение вариантов структуры системы для обеспечения требуемого уровня зашиты; определение вероятностей получения необнаруженных ошибок и угроз.

Задача разработки, внедрения и эксплуатации систем обеспечения защиты информации состоит в том, чтобы создать технологическую структуру, позволяющую достичь заданного уровня (класса) защищенности, либо обеспечить минимум суммарных потерь системы с учетом затрат на разработку и функционирование механизмов контроля, на исправление ошибок и на потери от угроз в системе при использовании недостоверных данных либо утечки информации. В [1] отмечено, что система защищенного документооборота должна соответствовать определенному классу защищенности, защиты от несанкционированного доступа, защиты от угроз и т. д. Для поддержания системы в заданном диапазоне (классе) защищенности при постоянно меняющихся внутренних и внешних условиях (атаки, модернизация технических и программных средств, организационные и структурные мероприятия, изменения и дополнения в законодательных и правовых актах и т. д.) необходимо оперативно адаптировать функциональные возможности технических и организационных средств защиты информации в зависимости от результатов анализа информационных потоков.

Сложность вычислительных сетей и комплексов, наличие множества факторов, влияющих на их работу, действие неконтролируемых воздействий и помех создают большие трудности при определении и поддержании оптимальных режимов. Эти трудности преодолеваются применением инвариантных и адаптивных методов оптимизации, повышением их работоспособности в обстановке помех.

Для поддержания сети в оптимальном режиме функционирования при заданном наборе целевых функций и ограничений, существующих возмущающих воздействиях и случайных помехах можно применить методы поисковой оптимизации. Из всего разнообразия методов поисковой оптимизации, получивших наибольшее распространение в промышленной оптимизации, достойное место занимает последовательный симплексный метод поиска с распознаванием состояний, так как метод прост в алгоритмизации, учитывает априорную информацию, аппроксимирует результаты поиска.

Сущность симплексных методов состоит в том, что в k-мерном пространстве управляемых переменных x_i движение к оптимуму осуществляется последовательным отражением вершин симплекса. Симплекс представляет собой фигуру с $k+1$ вершинами, не принадлежащими ни одному пространству меньшей размерности. В случае $k=1$ это прямая, при $k=2$ — треугольник, $k=3$ — тетраэдр и т. д. Целевая функция вычисляется в каждой из вершин симплекса. При поиске максимума вершина с наименьшим значением целевой функции отбрасывается и строится новый симплекс. Направление последнего перемещения симплекса в факторном пространстве достаточно близко к направлению градиента линейного приближения целевой функции.

Использование теории цепей Маркова позволяет адекватно описать процесс поисковой оптимизации с использованием алгоритмов симплексного поиска в обстановке неопределенности. Синтез алгоритмов осуществляется путем оптимизации структуры и параметров многосвязной марковской цепи, описывающей статистические свойства

симплексного поиска. Структура синтезированной трехсвязной марковской цепи, соответствующая процессу поиска на этапе восхождения, представлена стохастическим графом и описывает поиск с запретом возврата. Данный подход к синтезу структуры многосвязной цепи Маркова позволяет получить оптимальные алгоритмы на этапах восхождения и доводки при различных критериях эффективности поиска.

ЛИТЕРАТУРА

1. *Кузнецов Н. А., Кульба В. В., Микрин Е. А.* Информационная безопасность систем организационного управления. Теоретические основы. М.: Наука, 2006. 430 с.

УДК 004.056

РЕАЛИЗАЦИЯ МЕТОДА ЗАЩИТЫ АУТЕНТИФИКАЦИОННЫХ ДАННЫХ В МНОГОУРОВНЕВЫХ ПРИЛОЖЕНИЯХ

П. А. Паутов

Многоуровневое (или N-уровневое) приложение — приложение, разделенное на N самостоятельных уровней, каждый из которых может выполняться на отдельной платформе. Типичным примером является трехуровневая архитектура, используемая в веб-приложениях, где уровни распределены следующим образом:

1) уровень представления — браузер;
2) уровень приложения — программа, исполняемая на веб-сервере;
3) уровень данных — СУБД.

Применение трехуровневой архитектуры приводит к появлению нескольких звеньев аутентификации. В классическом двухуровневом приложении клиент аутентифицируется перед сервером. В трехуровневом приложении появляется второе звено аутентификации — клиент аутентифицируется перед прикладным уровнем, а прикладной уровень аутентифицируется перед СУБД. В современных веб-приложениях прикладной уровень работает с СУБД от имени фиксированного числа пользователей (часто от имени одного пользователя). При использовании парольной аутентификации возникает необходимость хранения аутентификационных данных (имен и паролей) пользователей СУБД на прикладном уровне. Данная проблема была рассмотрена в работе [1], в которой было предложено несколько схем защиты аутентификационных данных СУБД, хранимых на прикладном уровне. Предпочтительной для использования является схема с использованием асимметричного шифра.

Для реализации данной схемы был выбран язык программирования PHP. Программная библиотека, реализующая схему с использованием асимметричного шифра, решает следующие задачи:

1) реализацию рассмотренных в [1] алгоритмов управления пользователями;
2) управление сеансами работы пользователей и подключениями к СУБД.

Первая задача была решена с использованием библиотеки OpenSSL и базы данных SQLite. OpenSSL предоставляет необходимые криптографические функции, а модуль SQLite позволяет работать с локальным файлом как с SQL-совместимой базой данных. Так было организовано описанное в работе [1] хранилище учетных данных.

Рассмотрим вторую задачу подробнее. Для этого опишем работу типичного веб-приложения, разработанного с использованием языка программирования PHP.

Браузер посылает веб-серверу HTTP-запрос. Веб-сервер запускает соответствующую PHP-программу. Программа *устанавливает соединение с СУБД*, получает необ-