XXXI Неделя науки СПбГПУ. Материалы межвузовской научной конференции. Ч. VI: С.194-195, 2003.

УДК 004.451.83.056.53

М.К. Заднепрянец (4 курс, каф. РВКС), Е.А. Крук, д.т.н., проф.

О РАСПРЕДЕЛЕННОЙ АТАКОУСТОЙЧИВОЙ СИСТЕМЕ

В настоящее время одной из наиболее. актуальных проблем построения приложений в архитектуре «клиент-сервер» является обеспечение их безопасности. В данной работе предлагается подход к повышению безопасности трехуровневых систем, состоящих из хранилищ конфиденциальной информации (СУБД), веб-серверов и «тонких» клиентов пользователей. Как правило, в подобных системах реализуется следующая система безопасности. При обращении клиента к веб-серверу последний формирует запросы к СУБД. При этом для выполнения всех действий приложение, исполняемое на веб-сервере, должно быть аутентифицировано СУБД. Большинство современных систем используют для этого парольный механизм, т.е. серверному приложению сопоставляется пользователь СУБД с некоторым секретным паролем. Этот пароль должен быть доступен серверному приложению во время работы, для чего он, как правило, записывается в его конфигурационный файл. Это приводит к тому, что в случае выполнения успешной атаки на веб-сервер эта информация становится доступной взломщику, который может далее использовать ее для доступа к СУБД и извлечения конфиденциальной информации. Данный доклад представляет описание работы по построению распределенной атакоустойчивой системы, предназначенной для решения описанной проблемы.

Секретные объекты (пароли доступа к СУБД и т.п.) предлагается хранить в зашифрованном виде и расшифровывать их только по мере необходимости, после чего быстро уничтожать. При этом возникает проблема хранения ключа шифрования секретного пароля.

В [1] был описан протокол, позволяющий производить распределенное хранение секретного ключа системы шифрования RSA. Для дешифрования блока информации M клиент должен передать его серверам хранения компонентов секретного ключа (СХК). Каждый из серверов возвращает клиенту значение

$$S_i = M^{d_i} \bmod N$$

где d_i –часть секретного ключа, хранящегося на СХК. Клиент вычисляет произведение

$$S = \prod_{i=1}^{t} S_i \bmod N,$$

где t – количество серверов распределенного хранения секрета в системе, что дает дешифрованный текст. Здесь

$$d = \prod_{i=1}^{t} d_i,$$

где d - секретный ключ RSA. Таким образом, дешифрование сообщения выполняется без сборки секретного ключа в одном месте.

При шифровании сравнительно больших блоков данных, как правило, используется «цифровой конверт». «Цифровой конверт» состоит из сообщения, зашифрованного симметричным секретным ключом, и самого симметричного секретного ключа, обычно зашифрованного асимметричным ключом (открытый RSA-ключ). Предполагается, что каждый из участников обмена знает секретный RSA-ключ, следовательно, при необходимости получатель может расшифровать симметричный секретный ключ, а затем, применив его, расшифровать и само сообщение.

В рассматриваемом приложении реализован алгоритм распределенной генерации пары RSA-ключей, который описан в работе [2]. Все соединения между элементами приложения защищаются при помощи протокола SSL версии v2. Для защиты передачи конфиденциальной информации внутри системы применяется технология «цифрового конверта»: сообще-

ние шифруется при помощи симметричного секретного ключа системы шифрования DES, после чего DES-ключ шифруется открытым RSA-ключом.

Приложение состоит из приложений-клиентов, приложения-администратора и серверов хранения распределенного секрета. Приложение-администратор предоставляет возможность передачи секретных объектов в зашифрованном виде приложениям-клиентам, а также осуществляет управление СХК. При необходимости использования какого-либо секретного объекта приложения-клиенты взаимодействуют с СХК для его дешифрования, а после окончания его использования (например, после успешной аутентификации) – уничтожают.

Выводы: Разработана атакоустройчивая система, выполняющая основные криптографические действия (шифрования, дешифрования, распределенной генерации ключа на основе RSA). В дальнейшем предполагается улучшить и реализовать алгоритм обранужения вторжений [1].

ЛИТЕРАТУРА

1. Thomas Wu, Michael Malkin, and Dan Boneh. Building intrusion-tolerant applications. In *Proceedings of 8th USENIZ Security Symposium,* pp 79-92, 1999.

2. D. Boneh and M. Franklin. Efficient generation of shared RSA keys. *Lecture Notes in Computer Science,* 1294:425-439, 1997.