УДК 004.056.53

В.В. Давыдов, Д.С. Гребенюк

Национальный технический университет «ХПИ», Харьков

КОМПЛЕКС ПРОЦЕДУР ГЕНЕРАЦИИ ЛИЦЕНЗИОННОГО КЛЮЧА ДЛЯ ЗАЩИТЫ АВТОРСКИХ ПРАВ ИНТЕЛЛЕКТУАЛЬНОЙ СОБСТВЕННОСТИ НА ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

В статье описан процесс разработки программного комплекса генерации лицензионного ключа защиты авторских прав интеллектуальной собственности на программное обеспечение, учитывающего индивидуальные данные конечного пользователя. Верификация лицензионного ключа выполняется единожды при каждом запуске программного обеспечения в режиме «offline», т.е. без выполнения запросов на сервер, так как вся необходимая информация уже хранится локально.

Ключевые слова: *защита авторских прав, лицензионный ключ, клиент-серверная архитектура, защита от тиражирования, кроссплатформенное программное обеспечение, REST-сервисы.*

Введение

Постановка проблемы. В условиях повсеместного использования компьютерных, телекоммуникационных и других компьютеризированных средств, а также постоянного совершенствования и обновления их программного обеспечения (ПО), достаточно актуальной задачей является защита интеллектуальной собственности и авторских прав на программные продукты (обеспечение).

Особенно острой эта проблема выглядит в Украине, где компании-разработчики ПО несут финансовые потери из-за несанкционированного (незаконного) использования авторских прав на созданные программные продукты.

Анализ последних исследований и публикаций. Анализ литературы [1, 2, 4, 6] показал, что одним из наиболее эффективных средств защиты интеллектуальной собственности на ПО является лицензионный ключ защиты приложений (ЛКЗП). Обычно ключ применяется во время установки. Программа-установщик применяет алгебраические вычисления к вводимому ключу для проверки его на подлинность. Например, алгоритму необходимо определить, что вводимый ключ должен содержать 5 чисел, сумма которых равна 25, и что ключ также должен содержать 3-5 литер так, что после перевода их в числовые эквиваленты получим сумму 42 [3, 11].

Проведенные исследования [2, 4, 6, 7] показали, что в настоящее время существует ряд подходов к формированию ЛКЗП. Их основой являются известные криптографические алгоритмы, позволяющие формировать последовательности различного уровня сложности и стойкости. Следует заметить, что у большинства фирм-разработчиков ПО эта информация является конфиденциальной. В то же время, анализ открытых интернет-ресурсов [3] показал массовое предложение на программное обеспечение, позволяющее формировать так называемые keygen, которые пишутся как отдельными программистами, так и хакерскими группами, например, C.O.R.E., ORiON, Z.W.T, REVOLUTiON, XNTeam, Fight For Fun и др., специализирующимися на взломе программного обеспечения. Иногда такие группы заявляют о себе также тем, что включают своё название в сгенерированный ключ в открытом либо зашифрованном виде [3].

Поэтому актуальной является разработка генератора ЛКЗП, реализующего современные принципы контроля разрешений исполнения прикладного кода, который бы позволил минимизировать риск хакерской подделки, и тем самым повысил уровень защиты авторских прав на интеллектуальную собственность. Решение поставленной задачи невозможно без разработки соответствующего программного комплекса.

Цель статьи. Таким образом, целью статьи является разработка комплекса процедур генерации ЛКЗП для защиты авторских прав на ПО. Данные процедуры легли в основу программного комплекса генерации ЛКЗП.

Основные результаты исследований

В ходе разработки программного комплекса генерации ЛКЗП, с целью защиты программного комплекса от тиражирования, была разработана клиент-серверная архитектура, позволяющая продемонстрировать работу разработанного метода генерации лицензионного ключа.

Клиентское ПО, которое имеет защиту от тиражирования, использующую разработанный комплекс, имеет в своей структуре:

– полезный код, т.е. код самого программного продукта;

– client-processor.jar – разработанная библиотека, подтверждающая легальность лицензионного ключа и реализующая обмен сообщениями с серве-

ром. Данная библиотека обфусцирована, что уменьшает вероятность анализа алгоритма злоумышленниками. Библиотека состоит из:

1) сервисов доступа к базе данных на стороне клиента;

2) client-systeminfo.jar - модуля, отвечающего за получение информации о комплектующих клиентской компьютерной системы;

3) client-decoder.jar - модуля, декодирующего переданный лицензионный ключ. Выявляет в лицензионном ключе закодированный программный код, запускает его;

4) common-license.jar - модуля, отвечающего за генерацию цифровой подписи к сообщению, а также проверку подписи сообщения на основе тела сообщения, хэш-суммы и имеющегося публичного ключа. Его дубликат находится также на сервере;

5) common-api.jar - модуля, представляющего собой интерфейс доступа к сервисам сервера. Содержит интерфейсы функций реализованных возможностей для использования путем обмена REST-запросами. Его дубликат находится также на сервере;

Серверная часть состоит из:

– сервисов доступа к базе данных сервера;

– server-encoder.jar, - модуля, который на основе входного java-файла, содержащего код, выполняющийся на стороне клиента, а также дополнительных настроек – информации о конкретном программном продукте, создает class-файл (скомпилированный java-файл), который в последствии кодируется описанным в статье [5] алгоритмом;

– common-license.jar;

– common-api.jar.

Все нижеописанные диаграммы последовательностей состоят из четырех структур:

1. База данных клиента. В связи с тем, что предполагается хранение данных в формате «ключ-значение» и отсутствием связанных объектов, была выбрана база данных MapDB [9], предназначенная специально для хранения пар «ключ-значение». Данная база данных позволяет ввести систему аутентификации для защиты от несанкционированного доступа. При этом, в качестве пароля для базы данных используется хэш-сумма строки, содержащей информацию о комплектующих данной компьютерной системы, которая описана работе [5], и является практически уникальной для каждой компьютерной системы. Это уменьшает вероятность использование злоумышленником базы данных другого пользователя с зарегистрированным программным продуктом и позволяет избежать «тиражирования» лицензии на данный программный продукт.

2. Клиентское ПО. Состоит из программного обеспечения, которое имеет лицензионный ключ, и надстройки сервисов программного обеспечения по обработке лицензий, предназначенных для регистрации пользователя, регистрации программного продукта, верификации лицензионного ключа. При запуске данного ПО:

– происходит обращение к клиентской базе данных с использованием пароля, который вычисляется «на лету» (т.е. пароль от базы данных нигде не хранится. В случае, если поменялась конфигурация системы, то пароль уже подходить не будет);

– из базы данных берется закодированный лицензионный ключ для данного программного продукта и происходит его верификация, в частности выполнение закодированного в нем кода. Код представлен в виде блока программы на языке программирования Java, что дает как возможность кроссплатформенности использования данного подхода лицензирования, так и возможность перехватывания выполнения недопустимых команд, приводящих к аварийному завершению программы или доступа к «чужой» памяти.

3. Сервер. Находится в центре сертификации компании, поставляющей данное программное обеспечение. Представлен в виде REST API компонентов, которые работают под управлением сервера приложений Jboss Wildfly 8.2.0. При текущей реализации системы генерации лицензионных ключей используется Java 1.7. На сервере приложений настроен SSL доступ для повышения защиты механизма обмена сообщениями между сервером и клиентом от злоумышленного воздействия.

4. База данных сервера. Согласно бизнес требованиям была выбрана база данных MongoDB [10], имеющая более высокие показатели производительности чтения данных по сравнению с другими базами данных. Содержит информацию о всех зарегистрированных клиентах, приобретенных ими лицензиях, а также информацию о каждой зарегистрированной пользовательской компьютерной системе для возможности восстановления пароля или организации возможности обновления пароля/лицензионного ключа, если конечный пользователь согласованно меняет комплектующие системы.

Сервер предоставляет следующие возможности:

– регистрация нового пользователя;

– авторизация пользователя при утере/повреждении клиентской базы данных;

– регистрация новой компьютерной системы.

Любая покупка программного обеспечения начинается с того, что пользователь вносит свою клиентскую информацию в базу данных сервера, который владеет лицензией на данное программное обеспечение. Диаграмма последовательностей данного процесса представлена на рис. 1. При помощи надстроек сервисов программного обеспечения по обработке лицензий пользователь передает свои клиентские данные, которые сохранятся на сервере и будут отражать его в клиентской базе.

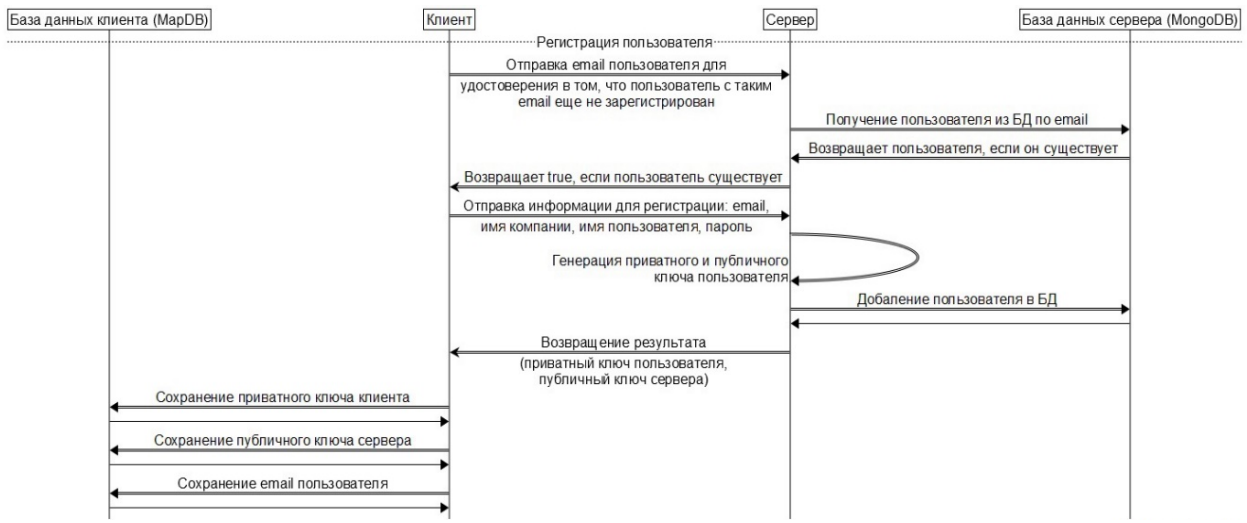

Рис. 1. Диаграмма последовательностей процесса регистрации пользователя в системе

К таким данным на текущий момент относятся:

– E-mail пользователя;

– имя компании, которую представляет данный пользователь. В случае, если пользователь – физическое лицо, это поле может оставаться пустым;

– ФИО пользователя, на имя которого происходит регистрация;

– пароль пользователя.

При этом, e-mail и пароль пользователя могут выступать в процессе аутентификации пользователя, в связи с этим, e-mail должен быть уникален в системе, а пароль должен быть безопасным согласно политике паролей [11].

После того, как система-сервер получила информацию о пользователе, активизируются следующие действия:

– проверяется наличие данного пользователя в своей базе данных. В случае, если пользователь с таким идентификатором-email уже существует, возвращается ответ с соответствующей ошибкой;

– генерируется пара ключей, с учетом собственного центра сертификации, конечного пользователя для возможности дальнейшего безопасного обмена сообщения с ним.

– сохраняется в базе данных переданная информация о пользователе, а также пара ключей. При этом, в целях защиты пользовательских данных, пароль пользователя не хранится в открытом виде, а хранится только его хэш-сумма, созданная при помощи утилиты BCrypt [8], основанная на шифре Blowfish.

– в качестве ответа, сервер возвращает клиенту публичный ключ сервера и приватный ключ клиента для возможности безопасного обмена сообщениями. Данный ответ не подлежит дополнительным средствам защиты от злоумышленника.

Получив успешный ответ от сервера, клиент сохраняет полученные ключи, а также свой e-mail в клиентской базе данных. Пользователь зарегистрирован в системе, и имеет возможность регистрировать конкретную компьютерную систему для данного программного обеспечения.

Проведенные исследования показали, что очень часто, в случае воздействия злоумышленного программного обеспечения или переустановки операционной системы, возникают ситуации утери или повреждения базы данных клиентского программного обеспечения. Пренебрежение этим существенно снижает практическую ценность разработки. Поэтому для учета данного фактора был разработан механизм восстановления указанной информации при наличии сохраненного e-mail и пароля зарегистрированного пользователя. Механизм аутентификации пользователя с восстановлением его данных представлен на рис. 2.

Процесс аутентификации происходит следующим образом:

1. Пользователь отправляет на сервер свои e-mail и пароль. В целях безопасности пароль отправляется в виде хэш-суммы, построенной утилитой BCrypt.

Сервер сверяет пользовательские e-mail и пароль. В случае ошибки авторизации – возвращается соответствующий ответ с ошибкой. Если пользовательские данные корректные, то из базы извлекается уже имеющаяся информация о конечном пользователе – его приватный ключ, и возвращается ответом с сервера вместе с публичным ключом сервера.

2. Получив успешный ответ от сервера, клиент сохраняет полученные ключи, а также свой e-mail в клиентской базе данных.

После того, как пользователь аутентифицировался на сервере, он имеет возможность добавлять/получать лицензии на конкретную компьютерную систему. Процесс добавления клиентской компьютерной системы описан на рис. 3, и состоит из описанных ниже этапов.

Рис. 2. Диаграмма последовательностей
процесса аутентификации пользователя в системе

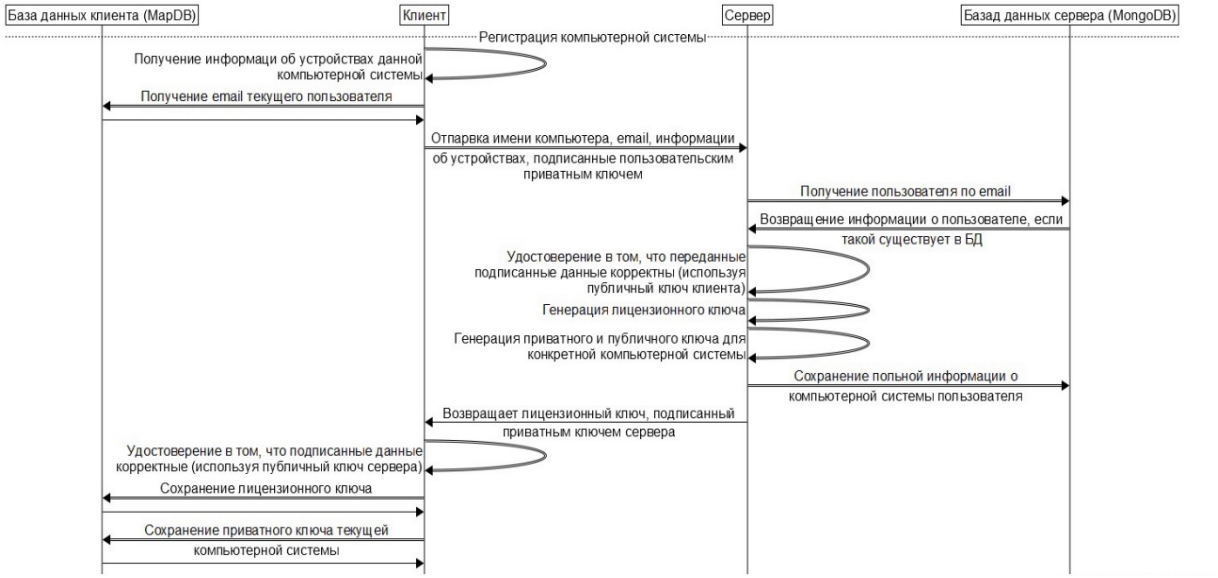

Рис. 3. Диаграмма последовательностей
процесса регистрации клиентской КС на сервере

1. Лицензионный модуль программного обеспечения получает информацию о компонентах компьютерной системы.

2. На основе хранимой в базе данных информации о пользователе, происходит отсылка запроса на сервер, содержащий следующую информацию: имя компьютерной системы (должно быть уникальным для пользователя, например, «Мой ПК 1»), e-mail пользователя, сгенерированная информация об устройстве. Вся эта информация подписывается приватным ключом клиента.

3. В связи с тем, что любая лицензия требует оплаты, пользователь отправляется на страницу оплаты лицензии через платежную систему, например, на paypal.com. Дальнейшая регистрация клиентской компьютерной системы осуществляется только при подтверждении транзакции от платежной системы.

4. Сервер получает информацию о зарегистрированном пользователе на основе полученного e-mail, проверяет достоверность переданного сообщения на основе имеющегося публичного ключа пользователя.

5. Сервер генерирует лицензионный ключ, кодирует его.

6. Копия лицензионного ключа хранится на сервере в закодированном виде с целью возможности его восстановления.

7. Сервер отправляет сгенерированный лицензионный ключ. Сообщение подписывается приватным ключом сервера.

8. Лицензионный модуль клиентского программного обеспечения проверяет достоверность переданного сообщения на основе имеющегося публичного ключа сервера и сохраняет лицензионный ключ в закодированном виде в базе данных, находящейся на стороне клиента.

9. Лицензионный ключ декодируется и запускается, что, в случае успешной работы, приводит к тому, что программное обеспечение бу-

дет зарегистрированным. Данный процесс выполняется каждый раз при запуске программного обеспечения.

Выводы

Таким образом, разработан программный комплекс генерации лицензионного ключа защиты приложений для защиты авторских прав интеллектуальной собственности на программное обеспечение. Отличительной особенностью данного комплекса является учет индивидуальных данных конечного пользователя, что предотвращает возможность тиражирования лицензионного ключа злоумышленниками.

В результате выполнения функций генерации сформированный лицензионный ключ представляет собой программный код, исполняемый на стороне конечного пользователя, что дает дополнительную защиту программного продукта от злоумышленного воздействия.

Для защиты баз данных от воздействия вредоносного программного обеспечения или переустановки операционной системы, разработан механизм аутентификации пользователя с восстановлением его данных.

Кроме этого, для защиты баз данных предусмотрена процедура обфускации.

Список литературы

1. Закон України «Про розповсюдження примірників аудіовізуальних творів, фонограм, відеограм, комп'ютерних програм, баз даних» від 13.01.2016 [Електронний ресурс] / - Режим доступу: http://zakon3.rada.gov.ua/laws/show/1587-14.

2. Болтенков В.А. Практическое исследование современных систем электронной цифровой подписи / Болтенков В.А., Еникеев Р.И. – Одесса: ОНПУ, 2014. – Том 4. - № 3. – 201-209 с.

3. Генератор ключей [Электронный ресурс] : - Режим доступа : ru.wikipedia.org.

4. Семенов С.Г. Исследования технологий динамического анализа бинарного кода программного обеспечения / С.Г. Семенов, С.Ю. Гавриленко, А.В. Мовчан // Компьютерные системы и проектирование технологических процессов и оборудования: Мат-лы Всеукр. научтехн. конф. – Черновцы: ЧФ НТУ «ХПИ», 2016. – С. 152-154.

5. Семенов С.Г. Система формирования цифрового идентификатора программного обеспечения для защиты авторских прав / С.Г. Семенов, В.В. Давыдов, А.В. Мовчан // Современные проблемы информатики в управлении, экономике, образовании и преодолении последствий Чернобыльской катастрофы: Мат-лы XV Междунар. науч. сем. - К.: Национальная академия управления, 2016. – С. 110-116.

6. Цифровые подписи в исполняемых файлах и обход этой защиты во вредоносных программах [Электронный ресурс] : - Режим доступа : https://habrahabr.ru/post/112289/

7. A.M. Bahaa-Eldin A comprehensive Software Copy Protection and Digital Rights Management platform / A.M. Bahaa-Eldin, M.A.A. Sobh // Ain Shams Engineering Journal, 2014 – Volume 5. – Issue 3. – P. 703-720.

8. BCrypt [Электронный ресурс] : - Режим доступа : ru.wikipedia.org.

9. Introduction to MapDB [Электронный ресурс] : - Режим доступа : https://www.gitbook.com/book/jankotek/mapdb/details.

10. Introduction to MongoDB [Электронный ресурс] : Режим доступа : https://docs.mongodb.com/manual/introduction/

11. Password Policy [Электронный ресурс] : Режим доступа : en.wikipedia.org

Надійшла до редколегії 24.01.2017

Рецензент: д-р техн. наук, с.н.с. С.Г. Семенов, Національний технічний університет «Харківський політехнічний інститут», Харків.

КОМПЛЕКС ПРОЦЕДУР ГЕНЕРАЦІЇ ЛІЦЕНЗІЙНОГО КЛЮЧА ДЛЯ ЗАХИСТУ АВТОРСЬКИХ ПРАВ ІНТЕЛЕКТУАЛЬНОЇ ВЛАСНОСТІ НА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

В.В. Давидов, Д.С. Гребенюк

В статті описано процес розробки програмного комплексу генерації ліцензійного ключа захисту авторських прав інтелектуальної власності на програмне забезпечення, що враховує індивідуальні дані кінцевого користувача. Верифікація ліцензійного ключа виконується один раз при кожному запуску програмного забезпечення в режимі «offline», тобто без виконання запитів на сервер, так як вся необхідна інформація вже зберігається локально.

Ключові слова: *захист авторських прав, ліцензійний ключ, клієнт-серверна архітектура, захист від тиражування, кросплатформенність, REST-сервіси.*

LICENSE KEY GENERATION PRODUCT FOR SOFTWARE INTELLECTUAL PROPERTY COPYRIGHT PROTECTION

V.V. Davydov, D.S. Hrebeniuk

The article describes the process of developing a software system generating license key copyright protection of intellectual property rights of software, taking into account the individual end-user data. License key verification is performed only once each time you start the software in «offline» mode, that is without executing queries to server, since all the necessary information is already stored locally.

Keywords: *copyright protection, license key, the client-server architecture, protection from replication, cross-platform software, REST-services.*