

# Зберігання, аналіз та захист даних

УДК 004.056

Д.М. Андрущенко, Г.Л. Козина

*Запорозьский национальный технический университет, Запорожье*

## МЕТОД ЗАЩИТЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ОТ НЕЗАКОННОГО ИСПОЛЬЗОВАНИЯ

*Предлагается метод защиты недорогих программ путем авторизации через Интернет. Описываются разработанный протокол, основанный на использовании электронной цифровой подписи, и его программная реализация. Проведен анализ стойкости метода к известным видам атак. Рассмотрены известные методы защиты программного обеспечения от незаконного использования и проведено сравнение предложенного метода с ними. Показано, что предложенный метод позволяет защитить условно-бесплатную программу, не потеряв при этом эффективность защиты.*

**Ключевые слова:** протокол, защита от копирования, программное обеспечение, shareware.

### Введение

Производители программного обеспечения несут большие убытки из-за нелегального использования их продукции, так называемого “пиратства”. Чаще всего юридические методы борьбы с правонарушителями не являются эффективными, поэтому для защиты своих интересов разработчикам целесообразно прибегать к техническим средствам защиты программного продукта от нелегального использования. К техническим методам защиты относят программные и программно-аппаратные средства, а также использование программ как онлайн-сервисов [1, 2].

Программные методы защиты обычно [2] подразумевают использование привязки программы к конфигурации оборудования, на котором ее установили. Привязка происходит в момент установки программного обеспечения и требует либо ввода лицензионного ключа, либо прохождения процедуры активации – получения одноразового кода (ключа), зависящего от конфигурации оборудования пользователя и пригодного для использования только на этом компьютере. В первом случае ничего не мешает пользователю незаконно распространять продукт вместе с лицензионным ключом. Во втором случае добросовестному пользователю необходимо согласиться с неудобствами, которые возникают при каждой смене оборудования, на котором он использует программное обеспечение, а недобросовестный пользователь имеет возможность незаконно использовать и распространять программное обеспечение вместе с так называемой виртуальной машиной [3], программно имитирующей необходимое конфигурационное оборудование.

Программно-аппаратные средства защиты [1, 2] подразумевают проверку наличия некоторых аппаратных средств, которые поставляются вместе с программой и для которых невозможно, либо очень трудно изготовить копию. Например, широко используют специально изготовленные CD, DVD и аппаратные ключи, подключаемые через USB-порт. Такие методы считаются более надежными, чем программные, но они имеют существенные недостатки. Во-первых, такое программное обеспечение можно поставлять только в «коробочной» версии, во-вторых, такое средство защиты может существенно увеличить стоимость программного продукта, в-третьих, пользователю также требуется согласиться с некоторыми неудобствами. В связи с этим данный метод защиты не пригоден для большинства недорогих и используемых в повседневной жизни программных продуктах.

Кроме того, известны [2] неоднократные случаи взлома защиты обоих типов и дальнейшего нелегального и беспрепятственного распространения программного обеспечения.

Самым надежным средством от копирования программ можно считать исполнение их как онлайн-сервисов, так называемых Software as a service (SaaS) («Программное обеспечение как услуга») [4], которые подразумевают исполнение программного обеспечения на стороне производителя, не доступной для пользователя. Пользователь же может только ввести входные и получить выходные данные через веб-интерфейс. Такой вид защиты малоприменим для программного обеспечения, которое требует больших объемов входных либо выходных данных, и в любом случае он не сможет заменить традиционные способы распространения программ.

В последнее время для разработчиков перспективным является предоставление так называемой условно бесплатной (Shareware) версии программы, когда пользователь может использовать полностью рабочую версию программы ограниченное время. Однако такая возможность чаще всего является слабым местом в защите программного обеспечения, поскольку ограничение на использование традиционно выполняется путем проверки промежутка между текущим временем и временем первого запуска программы либо подсчета количества запусков программы, а результат сохраняется в памяти компьютера. В таком случае появляются так называемые «кряки» для сброса счетчика и обеспечивается возможность пользоваться программой неограниченное время.

Таким образом, проблема обеспечения технической защиты при разработке программного продукта является актуальной.

### Анализ существующих методов

Технология StarForce [5] защищает от копирования и основана на анализе физических характеристик оптического носителя («привязка к диску»), являющемся носителем программы. Недостатком метода является необходимость использования оптического носителя при запуске программы. Защита хорошо подходит для компьютерных игр, однако малоприменима для небольших служебных программ.

Компания Алaddin [6] предлагает защиту от копирования и взлома, основанную на использовании аппаратных USB-ключей. Защита ориентирована в первую очередь на рынок корпоративных пользователей, однако также малоприменима для небольших недорогих программ.

Технология Software Activation Service (SAS) [7] защищает от копирования и основана на использовании привязки приобретенного программного обеспечения к электронному кошельку Webmoney, с которого были перечислены средства при покупке программы. Недостатком этого метода защиты является необходимость авторизоваться в webmoney, прежде чем использовать защищенное приложение. Кроме того, способы оплаты защищенного приложения ограничиваются лишь электронными деньгами webmoney.

Защита программного обеспечения CrypLine [8] основана на использовании привязки программы к аппаратному обеспечению компьютера. Недостатком метода является краткость его описания и малое распространение.

Наиболее перспективным для защиты недорогих условно-бесплатных программ, по мнению авторов, является метод авторизации через Интернет [9]. Он подразумевает первоначальную активацию продукта на вычислительной машине пользователя,

а также авторизацию пользователя на сервере разработчиков при каждом запуске программы. Однако, данный метод в настоящее время имеет очень малое распространение и готовых решений в открытом доступе авторами найдено не было.

### Метод авторизации через Интернет

Авторизация пользователя при каждом запуске программы позволяет разработчику следить за статистикой использования программы, выявлять случаи нарушения лицензий, лишать лицензий недобросовестных пользователей, а также гибко изменять лицензионную политику в соответствии со своими нуждами.

В качестве ограничений на использование условно-бесплатных программ может служить введение ограничения на легальное использование программного продукта, например ограничение на количество запусков программы в месяц, либо ограничение на количество просмотренных и созданных документов за некоторый промежуток времени либо недопущение одновременного запуска программы на нескольких компьютерах. Такие ограничения должны быть прописаны в лицензионном соглашении вместе с санкциями за их нарушения. Под санкцией может подразумеваться как полное, так и временное лишение лицензии. Проверка нарушения лимитов должна производиться в недоступном для пользователя модуле контроля лицензий, например, удаленном сервере, куда программа должна посылать данные и проверять наличие разрешения на запуск. Если вдруг нелегальная копия программы окажется опубликованной в публичном месте, то лимиты достаточно быстро будут превышены и лицензия будет заблокирована. Для того чтобы организовать такую защиту, необходимо организовать безопасный обмен по открытому каналу связи между программой и модулем, в котором недобросовестный пользователь может читать и изменять любые пересылаемые данные. В связи с повсеместным распространением Интернета такой обмен можно организовать достаточно просто, не вызывая значительных неудобств у пользователей. Кроме того, программа может контролировать попытку изменения кода и посылать сообщение серверу.

Однако, для реализации такой схемы необходима разработка безопасного протокола передачи данных между программой и удаленным сервером.

### Протокол защиты программного обеспечения

Авторами разработан протокол защиты программного обеспечения, который основан на использовании механизма электронной цифровой подписи (ЭЦП) [10]. Суть его состоит в следующем.

Пусть имеется защищаемое приложение *Prog*,

установленне на комп'ютері користувача  $U$ , і віддалений сервер  $S$ , що належить розробникам програми або їх довірчому лицю.

Розробчик повинен вибрати систему електронної цифрової підписи і сгенерувати пару ключів – відкритий ключ  $e$  і закритий ключ  $d$ . Закритий ключ  $d$  повинен зберігатися на сервері  $S$ , а відкритий ключ  $e$  – в програмі  $Prog$ . Перед першим запуском програми користувач повинен отримати ідентифікатор (логін)  $I$  і пароль  $P$ . Кожен раз, коли користувач  $U$  намагається виконати одне з дій установлених розробником, наприклад запуск програми, створення, відкриття або збереження документа, програма  $Prog$  повинна надіслати запит серверу  $S$  про можливість продовжити роботу, виконавши наступні кроки передачі даних (рис. 1):

1. В програмі  $Prog$  генерується випадкове число  $RND$ .
2. В програмі  $Prog$  обчислюється деяке число  $F$  – прив'язка до програмно-апаратного забезпечення обчислювальної машини, де вона встановлена.
3. Програма  $Prog$  передає дані  $I, P, RND$ ,  $F$  серверу  $S$ .
4. Сервер  $S$  перевіряє можливість використання програми користувачу з ідентифікатором  $I$ , паролем  $P$  і прив'язкою  $F$ .
5. В разі підтвердження можливості запуску програми  $Prog$ ,  $S$  обчислює електронну цифрову підпис  $C(RND)$ , використовуючи закритий ключ  $d$ .
6. Сервер  $S$  надсилає значення  $C(RND)$  програмному забезпеченню  $Prog$ .
7. В програмі  $Prog$  здійснюється перевірка автентичності підпису сервера  $C(RND)$  за відомим відкритим ключем  $e$ . Якщо підпис автентичний, то програма продовжує виконуватися, в протилежному разі завершує роботу.

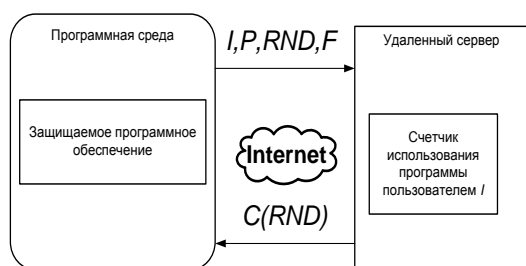


Рис. 1. Схема обміну даними між захищеним програмним забезпеченням і віддаленим сервером

**Аналіз розробленого протоколу.** Зв'язок з сервером може здійснюватися через публічну глобальну мережу Інтернет за його доменним іменем або IP-адресою.

Разрешення на запуск програми дається уда-

леним сервером і може ґрунтуватися на наступних даних: кількість завершених успішних запусків програми, час використання програми, оплачуваний користувачем баланс, кількість створених документів при використанні програми і др.

Перед першим використанням програми користувач повинен отримати ідентифікатор і пароль для запуску програми.

Це може здійснюватися правообладателем програми або третьою довірчою стороною. Для унікальності реєстраційних даних правообладатель може вимагати паспортні дані користувача.

Проаналізуємо розроблений протокол з наступних видів атак:

1) *атака підбору пароля.* Для зменшення ймовірності підбору пароля користувача при реалізації протоколу необхідно ввести лічильник неуспешних входів в систему і заблокувати на деякий час можливість входу користувача після деякого числа невдалих входів;

2) *атака на електронну цифрову підпис.* Для підтвердження автентичності віддаленого сервера використовується електронна цифрова підпис. Спроба підміни віддаленого сервера ґрунтується на можливості взлому алгоритму електронної цифрової підписи і виходить за межі розробки даного алгоритму. Можливо лише рекомендувати використовувати при реалізації протоколу будь-яку сучасну схему електронної цифрової підписи [10];

3) *коллізія.* Для неможливості використовувати раніше отримані від сервера відповіді використовуються випадкові числа. Можливо рекомендувати після генерації випадкового числа виконувати деяку затримку;

4) *атака підміни відкритого ключа.* Для неможливості підміни відкритого ключа віддаленого сервера в код програми використовуються засоби запобігання коду програми. Суть таких перетворень – усунути можливість простого інженірингу програми [11 – 13], т.е. спроби проаналізувати і змінити вихідний код.

**Програмна реалізація протоколу.** В процесі реалізації розробленого протоколу виявилось, що практично для будь-яких мов програмування існують готові бібліотеки, дуже спрощують написання коду. Клієнтська частина розроблена на мові програмування С#. Блок-схема алгоритму наведена на рис. 2. В якості механізму електронної цифрової підписи обрано алгоритм RSA [14], який реалізовано на основі застосування бібліотеки для роботи з довгими числами libgmp-3.dll [15]. Фрагмент коду програми наведено в листинзі 1.

Серверная часть была реализована на языке программирования PHP. Блок-схема алгоритма приведена на рис. 3. Для реализации алгоритма RSA использован класс `rsa.class.php` [16]. Фрагмент кода программы приведен в листинге 2.

Листинг 1

```

Фрагмент программной реализации
клиентской части на языке C#:
//Открытый ключ сервера
string N =
"93606078985615463936324467936728205462489080
2622109359669985073430011139453";
string E = "65537";
//Генерируем случайное число
Random r = new Random();
RND = r.Next(100000002, 999999997).ToString();
// Вычисляем ЭЦП
C = server(I,P,F,RND);
    
```



Рис. 2. Блок-схема алгоритма клиентской части

```

// Если подпись сервера не подлинная, то завершаем
работу программы;
If (!RsaEncrypt(C) == E) exit();
    
```

Листинг 2

Фрагмент программной реализации серверной части на языке PHP:

```

//Закрытый ключ сервера
$N =
"93606078985615463936324467936728205462489080
2622109359669985073430011139453";
$D =
"62070733412752273713609483457214224525497466
7653791289861110526810920668639";
//Вычисление ЭЦП
$C = $RSA->mdecrypt($RND, $D, $N);
//Если программу запускать можно, то возвращаем
ЭЦП от числа RND
If (allow(I,P,F)) echo $C else echo "false";
    
```

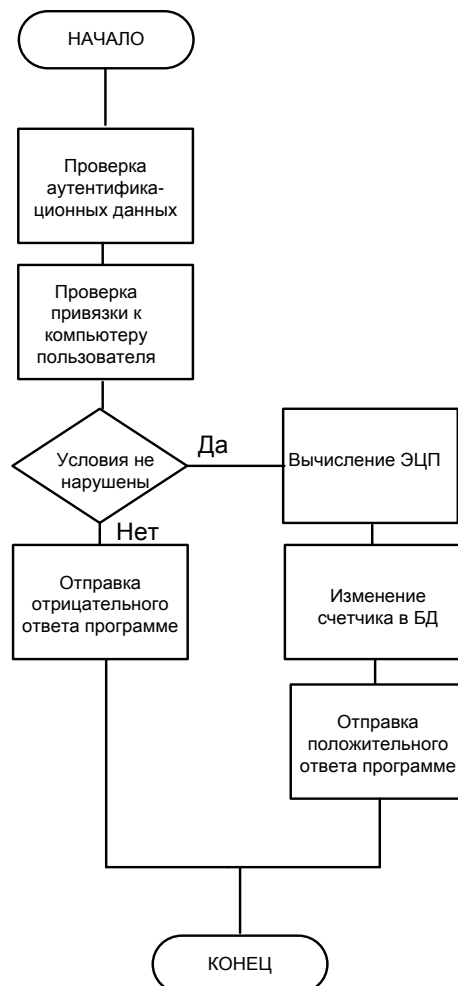


Рис. 3. Блок-схема алгоритма серверной части

### Сравнение методов

Приведем сравнение предложенного способа защиты с методами StarForce, Аладдин, SAS, Crypline (табл. 1).

Обозначения в таблице, соответственно:

C<sub>1</sub> – нет необходимости использовать аппаратные средства при использовании программы;

C<sub>2</sub> – нет необходимости активировать программу перед первым использованием;

C<sub>3</sub> – нет необходимости использовать Интернет при использовании программы либо ее активации;

C<sub>4</sub> – нет возможности нелегально использовать программу путем переноса ее на виртуальную машину;

C<sub>5</sub> – есть возможность использовать условно бесплатную версию (Shareware).

Таблица 1

Сравнение предложенного метода с известными

Метод	C <sub>1</sub>	C <sub>2</sub>	C <sub>3</sub>	C <sub>4</sub>	C <sub>5</sub>
StarForce	–	+	+	+	–
Аладдин	–	+	+	+	–
SAS	+	–	–	+	–
Crypline	+	–	–	–	+
предложенный	+	+	–	+	+

Из таблицы видно, что необходимость подключения к Интернету при активации либо использовании программы заменяет необходимость использовать аппаратный ключ. Защитить условно-бесплатную программу, не потеряв при этом эффективность защиты, позволяет лишь предложенный метод.

### Выводы

Предложенный метод позволяет обеспечить более надежную защиту по сравнению с существующими программными методами, поскольку исключает возможность установки программного продукта на виртуальную машину. Этот метод позволяет отказаться от использования аппаратных устройств при запуске программы. Простота реализации клиентской и серверной части программного обеспечения позволяет использовать его как в больших, так и небольших программах, не удорожая стоимость разработки. В дальнейшем авторами планируется усовершенствование протокола для осуществления возможности более гибкого управления лицензионной политикой разработчиком.

### Список литературы

1. Варлатая С.К. Программно-аппаратная защита информации: учебн. пособие / С.К. Варлатая, М.В. Шаханова. – Владивосток: Изд-во ДВГТУ, 2007. – 318 с.
2. Складов Д.В. Искусство защиты и взлома информации / Д.В. Складов. – СПб.: БХВ-Петербург, 2004. – 288 с.

3. Виртуальная машина [Электронный ресурс]. – Режим доступа до ресурсу: [http://ru.wikipedia.org/wiki/Виртуальная\\_машина](http://ru.wikipedia.org/wiki/Виртуальная_машина).

4. Software as a service [Электронный ресурс]. – Режим доступа до ресурсу: [http://ru.wikipedia.org/wiki/Software\\_as\\_a\\_service](http://ru.wikipedia.org/wiki/Software_as_a_service).

5. Защита от пиратства и лицензирование ПО StarForce [Электронный ресурс]. – Режим доступа до ресурсу: <http://www.star-force.ru/>.

6. Защита программ Насп [Электронный ресурс]. – Режим доступа до ресурсу: <http://www.aladdin-rd.ru/>.

7. Сервис Software Activation Service [Электронный ресурс]. – Режим доступа до ресурсу: <http://www.softactivation.com/asp/getstarted.asp>.

8. Защита Crypline [Электронный ресурс]. – Режим доступа до ресурсу: <http://crypline.ru>.

9. Анализ рынка средств защиты от копирования и взлома программных средств [Электронный ресурс]. – Режим доступа до ресурсу: <http://citforum.ru/security/articles/analisis/>.

10. Молдовян Н.А. Введение в криптосистемы с открытым ключом / Н.А. Молдовян, А.А. Молдовян. – СПб.: БХВ-Петербург, 2005. – 288 с.

11. Буинцев Д.Н. Метод защиты программных средств на основе запутывающих преобразований: дис. ... канд. техн. наук : 05.13.19 / Д.Н. Буинцев. – Томск, 2006. – 121 с.

12. Обратная разработка [Электронный ресурс]. – Режим доступа до ресурсу: [http://ru.wikipedia.org/wiki/Обратная\\_разработка](http://ru.wikipedia.org/wiki/Обратная_разработка).

13. Система программной защиты приложений от несанкционированного копирования ASProtect [Электронный ресурс]. – Режим доступа до ресурсу: <http://www.aspack.com/asprotect.aspx>.

14. Дихунян В.Л. Электронная идентификация. Бесконтактные электронные идентификаторы и смарт-карты / В.Л. Дихунян, В.Ф. Шаньгин. – Изд-во «ИТ Пресс», 2004. – 695 с.

15. GMP Install Instruction for Windows Platform [Электронный ресурс]. – Режим доступа до ресурсу: <http://cs.nyu.edu/exact/core/gmp/>.

16. RSA class [Электронный ресурс]. – Режим доступа до ресурсу: <http://code.google.com/p/phpjsrsa/source/browse/trunk/rsa.class.php?r=6>.

Поступила в редколлегию 7.09.2010

**Рецензент:** д-р техн. наук, проф. Д.М. Пиза, Запорожский национальный технический университет, Запорожье.

### МЕТОД ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ВІД НЕЗАКОННОГО ВИКОРИСТАННЯ

Д.М. Андрущенко, Г.Л. Козіна

Пропонується метод захисту недорогих програм шляхом авторизації через інтернет. Описуються розроблений протокол, заснований на використанні електронного цифрового підпису, і його програмна реалізація. Проведено аналіз стійкості методу до відомих видів атак. Розглянуті відомі методи захисту програмного забезпечення від незаконного використання і проведено порівняння запропонованого методу з ними. Показано, що запропонований метод дозволяє захистити умовно-безкоштовну програму, не втративши при цьому ефективність захисту.

**Ключові слова:** протокол, захист від копіювання, програмне забезпечення, shareware.

### METHOD OF DEFENCE OF SOFTWARE OT THE ILLEGAL USE

D.M. Andruschenko, G.L. Kozina

The method of defence of the inexpensive programs is offered by authorizing over the internet. Described the developed protocol, based on the use of electronic digital signature, and his programmatic realization. The analysis of firmness of method is conducted to the known types of attacks. The known methods of defence of software are considered on the illegal use and comparing of the offered method is conducted to them. It is shown that the offered method allows to protect a shareware, not losing efficiency of defence here.

**Keywords:** protocol, protecting from copying, software, shareware.