УДК 681.325.3

С. Б. Макась, ассистент

ОРГАНИЗАЦИЯ ЗАЩИТЫ ЛИЦЕНЗИОННОЙ ИНФОРМАЦИИ ИНТЕРНЕТ-ПРИЛОЖЕНИЙ НА ОСНОВЕ АЛГОРИТМОВ КОМПЬЮТЕРНОЙ СТЕГАНОГРАФИИ

Method of protecting the Internet-appendices component on the basis of technology of licensing the Internet-based applications on .Net Framework. In particular, the way of the organization of structure of the confidential certificate – licenses of a program component the Internet-appendix on a platform .Net Framework is described. The method of protection of the license information from opening and the falsifications, based on algorithms computer steganography and cryptography with use of elements of noiseproof coding is described. The mathematical description of the offered way of protection of the license information is presented. Results of the statistical and complex analysis of results of concealment of the confidential license information are resulted. On the basis of results of the analysis the optimum size of the block of the protected information in structure of the confidential certificate of a program component from the point of view of their use in the Internet-appendices is offered.

Введение. Повсеместная интеграция информационных систем с телекоммуникационными и сетевыми технологиями приводит к созданию новой информационно-вычислительной среды. В связи с этим существенно изменился принцип организации прикладного, сетевого, клиент-серверного и другого программного обеспечения. Это заключается в смещении от настольных (автономных) приложений в сторону распределенных приложений, активно использующих глобальные информационные сети в качестве сервиса обмена информацией и распределение нагрузки при выполнении задач. Наиболее яркий представитель такой интеграции – всемирная паутина Интернет.

Такая интеграция программных продуктов и приложений во всемирную сеть открывает новые аспекты в вопросах безопасности Интернет-приложений и их защиты. Обобщенно, все задачи обеспечения безопасности, рассматриваемые в контексте Интернет-приложений, можно разделить на следующие категории [1]:

1) доступ к конфиденциальным данным приложения через Интернет;

2) доступ к внутренним ресурсам платформы исполнения посредством ошибок как в самом приложении, так и в среде его исполнения;

3) обеспечение защиты программных компонент самих приложений от незаконного использования.

Первые две категории вопросов решаются на уровне проектирования операционных систем и платформ исполнения приложений (Интернет-сервера) и зависят друг от друга. Тогда как особенности функционирования Интернет-приложений накладывают на решение третьей категории вопросов ряд ограничений и дополнительных требований по сравнению с традиционными методами защиты программных продуктов. Это делает актуальной проблему разработки специализированных методов защиты от взлома, модификации и незаконного использования, посредством которого злоумышленник может обойти два первых критерия безопасности Интернет-приложений.

1. Особенности организации защиты компонент Интернет-приложений. В среде глобальной информационной сети Web-приложения, Web-сервисы и основанные на их использовании Интернет-приложения на платформе исполнения .Net Framework являются наиболее распространенным и часто используемым ресурсом, и, как следствие, их компоненты в первую очередь нуждаются в защите. Принцип построения, развертывания и функционирования этих программных решений достаточно сильно отличается от настольных (автономных) приложений, что заключается в следующем:

1. Приложение платформы .Net Framework храниться в виде IL-кода, транслируемого средой исполнения при загрузке в аппаратно-зависимый (*native*) байт-код *host*-сервера.

2. Большинство Web-приложений располагаются на публичных серверах общего доступа, что накладывает ограничения на использование и доступ к системным и аппаратным ресурсам *host*-сервера.

3. Алгоритмы защиты и верификации компонент должны быть достаточно высокопроизводительны и не требовательны к ресурсам системы.

4. Наличие в жизненном цикле приложения *round trip* циклов [2] для почти всех компонент экземпляра приложения при обработке каждого последующего запроса клиента накладывает достаточно строгое ограничение на организацию вычислительного процесса обращения к ресурсам системы.

5. Одновременный запуск на сервере множества сессий одного приложения. К Web-серверу одновременно обращаются множество клиентов (от 10 до 1500 в зависимости от времени суток).

Ввиду этих особенностей известные методы защиты, такие как серийный номер пакета инсталляции, машинный ключ, шифрование исполнимых файлов, защита сегментов оператив-

ной памяти, имеют ряд недостатков, которые обусловлены требованием обеспечения достаточно больших полномочий для доступа к операционной системе *host*-сервера.

2. Защита компонент Web-приложений методом лицензирования. Учитывая вышеперечисленное, в данной работе предлагается использовать технологию лицензирования как наиболее оптимальный способ защиты компонент Web-приложений. Главной идеей лицензирования программных продуктов можно назвать процесс создания специальных защищенных удостоверений (*ticket's*), специфичных для каждого экземпляра приложения или компонента [1]. Все это ставит актуальной задачу защиты удостоверений (лицензий) от взлома, изменения и фальсификации злоумышленником.

Техническая реализация лицензионной защиты заключается в использовании семейства открытых интерфейсов *ILicense*. В этом случае для решения задачи необходимо реализовать методы из этих интерфейсов, которые среда исполнения (.Net Framework) будет вызывать автоматически при загрузке классов защищаемого компонента.

Этот метод по сравнению с остальными имеет ряд преимуществ [3]:

1) позволяет защищать не все приложения сразу, а наиболее важные объекты приложения;

2) дает возможность достаточно просто внести элементы защиты приложения во множество классов программы, другими словами, организовать множество контрольных точек проверки лицензии;

3) способ достаточно автономный от аппаратных и человеческих ресурсов.

Для реализации лицензионной защиты компонент необходимо решить ряд задач:

– организация структуры лицензионной информации;

– хранение лицензионной информации;

– издание распространяемых компонент и генерация лицензионной информации;

– проверка лицензий, методы активации.

3. Организация лицензионной информации. Для оптимизации доступа к лицензионной информации ее структуру удобно представить в виде XML-документа. Пример лицензионной информации показан на рис. 1. Основные элементы лицензии: *assemblyId* – ключ идентификации экземпляра сборки, которая представляет собой вычисляемый хэш H на основании следующих данных:

$$assemblyId = H(asmHash, SerialKey, ServerKey), \quad (1)$$

где *asmHash* – хэш, вычисляемый средой исполнения для каждой сборки; *SerialKey* – серийный номер сборки; *ServerKey* – уникальный ключ, известный только серверу лицензирования; *serialId* – серийный номер сборки; *clientId* – идентификатор клиента; *power.swicher* – бинарный контейнер, в котором храниться ключевая информация для активации лицензии, извлекаемая из него при помощи ключей активации.

```
<licensing>
<clienInfo>
  <clientId
value="Y6lpxlkLjBz6oPdUI5BVnmKBzTM=" />
  <firstName value="Peter" />
  <lastName value="Smith" />
  <organisation value="Big company" />
  <email value="Peter@company.com" />
  </clienInfo>
<assemblyInfo>
  <serialId value="UrJa-b4X0-sseY-Ja1W" />
  <assemblyId
value="VLBH82pmGaclg5XPDt65pp5ZfoE=" />
  </assemblyInfo>
<power>
<switcher>cATxtLFTs4WAeNyN7JqJnONSPK8v2mC9PuXLX
lt4Q1aDkiFzRTv4kuu53XIgtJRYukTZhahGrsnNLc7AjqQv
4+0+xjwHTAWMqV+rbkCeyqOVu9p=XOWYT5616bYhzdZ=G=O
StNv0MEU=ofBwl4Z8DAto7clIc1r4pVMP1ZaxAN+XjpuKfd
zs5QHjnDIOcCxAMEEY55JSGJ0PnPjuZWBofloNRIZGMeQ3R
TOaGu5BaoGTbE0COF1xxfe9U+SlG+L8</switcher>
</power>
</licensing>
```

Рис. 1. Организация лицензионной информации

4. Хранение и защита лицензии и информации активации. Информация проверки лицензии (*power.switcher)* представляет собой контейнер со следующей информацией: первая пара ключей K_1 для извлечения контрольного *hash*-массива из файла активации, половина ключа Sa_1, используемого при вычислении проверочного *hash*-массива активации лицензии.

В процессе активации клиент на запрос получает в ответ вторую половину ключей K_2 и ключ активации, который представляет собой контейнер с осажденной информацией следующего содержания: тип лицензии, срок действия, периодичность проверки, серверная подпись клиента, вторая половина проверочного ключа Sa_2, контрольная сумма проверочного *hash*-массива, вычисляемого на основании ключей Sa_1, Sa_2, лицензионного и серверного идентификатора клиента.

В основу защиты ключа активации и информации проверки лицензии от вскрытия и фальсификации положена гипотеза о сохранении статистических параметров контейнера после процесса осаждения в него ключевой информации. Гипотеза заключается в следующем: в контейнер (байт-массив), сформированный с заданными статистическими характеристиками, осадить определенным методом байт-массив ключевой информации так, чтобы статистические характеристики контейнера существенно не изменились, что позволит защитить ключевую информацию от незаконного извлечения

методом анализа последовательности контейнера с сообщением. В качестве реализации метода предлагается модифицировать стеганографический метод осаждения информации, предложенный Луби и Рекоффом [4], в котором преследовалась цель осаждения информации методом псевдослучайной перестановки при условии минимизации показателей визуального искажения графической информации.

Процесс осаждения лицензионной информации в контейнер осуществляется по следующему алгоритму [5]:

1. Генерация матрицы-контейнера $M[X, Y]$ с четко заданными статистическими характеристиками распределения.

2. Сворачивание ключевой информации в бинарную матрицу $K[N, N]$, где N – кратно степени двойки $(N = 2^n)$.

3. Осаждение упакованной информации в контейнер методом псевдослучайной перестановки байт осаждаемой информации в пределах контейнера.

Осаждение информации в контейнер выполняется методом псевдослучайной перестановки байт, где позиция осаждения вычисляется посредством *hash*-фукций f_K по секретному ключу K:

$$f_K(i) = H(K \circ i). \tag{2}$$

Для увеличения секретности секретный ключ K разделяется на 4 части и для каждой части используется отдельный *hash*-метод объединения текущей позиции байта контейнера с частью ключа K_i:

$$\begin{aligned} y &= y \oplus f_{K_1}(x) = y \oplus H(K_1 \circ x), \\ x &= x \oplus f_{K_2}(y) = x \oplus H(K_2 \circ y), \\ y &= y \oplus f_{K_3}(x) = y \oplus H(K_3 \circ x), \\ x &= x \oplus f_{K_4}(y) = x \oplus H(K_4 \circ y). \end{aligned} \tag{3}$$

Практически функции объединения позиции осаждаемого байта x и y с ключами реализованы следующим образом:

$$\begin{aligned} y &= \mathrm{div}(i, Y) + 1, \\ y &= \mathrm{mod}(i, Y) - 1, \\ x &= \mathrm{mod}(x + f_{K_1}(y), X) + 1, \\ y &= \mathrm{mod}(y + f_{K_2}(x), Y) + 1, \end{aligned} \tag{4}$$

где функции объединения $f_{K_1}(x)$ и $f_{K_2}(x)$ представляют собой следующие преобразования, вычисляемые для всех пар ключей K:

$$\begin{aligned} f_{K_1}(x) &= (x + K_1 \oplus y) \bmod X, \\ f_{K_2}(y) &= (y + K_2 \oplus x) \bmod Y, \end{aligned} \tag{5}$$

где X, Y – размерность контейнера; x, y – позиция байта для осаждения информации; K_1, K_2 – пара секретных ключей K.

Извлечение упакованной информации осуществляется методом воспроизведения последовательности перестановки на основании ключей K_1 и K_2.

5. Анализ контейнера с осажденной информацией. В качестве подтверждения гипотезы о сохранении контейнером первоначальных статистических характеристик было проведено несколько исследований, направленных на определение изменений статистических характеристик и качества осаждения. В частности, проведен ряд тестов для определения, осталась ли последовательность по-прежнему равномерно распределенной и насколько изменились ее статистические характеристики.

Для анализа были вычислены следующие статистические характеристики для осаждения одного и того же массива в контейнеры разного размера. Результат приведен в табл. 1.

Таблица 1

Статистические показатели контейнера с осажденной информацией

N	m_1	m_2	m_3	m_4	*chi*	K_λ
16	126.1	$5.4 \cdot 10^3$	$-2.8 \cdot 10^3$	$5.2 \cdot 10^7$	3.10	16.81
24	126.3	$5.3 \cdot 10^3$	$5.3 \cdot 10^3$	$5.3 \cdot 10^7$	4.58	18.47
32	127.8	$5.5 \cdot 10^3$	$-9.3 \cdot 10^3$	$5.4 \cdot 10^7$	4.90	18.47
48	128.6	$5.3 \cdot 10^3$	$5.3 \cdot 10^3$	$5.1 \cdot 10^7$	2.45	20.09
64	128.1	$5.4 \cdot 10^3$	$-7.0 \cdot 10^3$	$5.3 \cdot 10^7$	2.79	21.66

Примечание. N – размерность контейнера; m_1, m_2, m_3, m_4 – статистические моменты последовательности распределения контейнера с осажденной информацией; *chi* – вычисленное значение критерия χ^2 для равномерно распределенной последовательности; K_λ – табличное значение χ^2 для заданного количества степеней свободы и уровня значимости.

Для сравнения в табл. 2 приведены статистические характеристики исходного контейнера.

Таблица 2

Статистические показатели контейнера

m_1	m_2	m_3	m_4	*chi*	K_λ
128.3	$5.0 \cdot 10^3$	$-7.6 \cdot 10^3$	$5.3 \cdot 10^7$	3.53	128.3

Как видно из результатов анализа, для всех размеров контейнера наблюдается выполнение критерия χ^2 для равномерно распределенной последовательности. Более наглядную картину качества осаждения информации можно получить, используя *комплексный анализ* объектов по многим признакам и вычисление значений *автокорреляционной функции*.

Для применения комплексного анализа необходимо настроить функцию желательности, по которой будут вычисляться приведенные комплексные коэффициенты. В анализе используется функция желательности с двусто-

ронним ограничением по причине того, что отклонения характеристики контейнера с информацией как в большую, так и в меньшую сторону от идеала (контейнера без информации) считаются ухудшением качества последовательности. Функция желательности для каждого из признаков объекта (столбцы табл. 1) вычисляется по следующей формуле:

$$\mu(x,a,b) = e^{-\frac{(x-a)^2}{2b^2}}, \qquad (6)$$

где x – значение признака; a – значение идеала; b – допустимое отклонение, при котором качество считается удовлетворительным.

Для оценки качества установим значение отклонения b, равным 5% от значения оригинала, что соответствует уровню значимости 0.95, используемому в анализе χ^2.

Значение функции желательности для первого признака (m_1) показано на рис. 2.

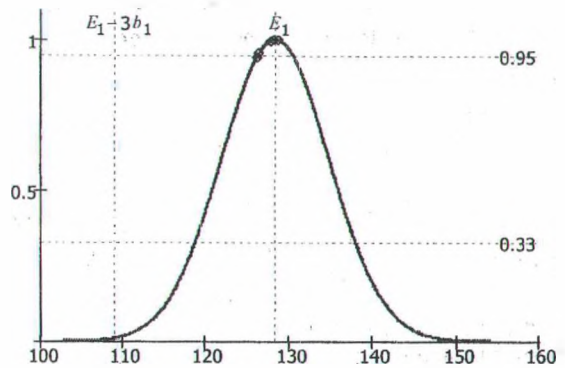

Рис. 2. Функция желательности для первого статистического момента

Результат вычисления обобщенного комплексного коэффициента качества представлен в табл. 3.

Таблица 3

Значение обобщенного показателя качества *D* осаждения для различных размеров контейнера

Размер контейнера, байт	16	24	32	48	64
D	0.907	0.601	0.414	0.571	0.785

Анализируя результаты исследования контейнера методом χ^2, можно сделать следующие выводы, что наиболее качественное осаждение информации наблюдается в контейнерах размером 16×16 и 64×64, что соответствует размерности байт-массивов 256 и 4096 байт. Как следствие, наиболее оптимальным будет использование при защите лицензионной информации контейнера 16×16 (256 байт), так как он обеспечивает высокое качество осаждения и при этом его малый размер не будет создавать существенный избыточный трафик между *host*-сервером Интернет-приложения и сервером лицензирования при периодических проверках и активации.

Еще один способ проверки качества распределения байт в контейнере после осаждения это вычислить коэффициент автокорреляции и получить набор дискретных значений автокорреляционной функции для k точек. Коэффициент автокорреляции рассчитывается по формуле

$$r_k = \frac{\sum_{i=1}^{N-k}(x_i - m_1)(x_{i+k} - m_1)}{\frac{N-k}{m_2}}, \qquad (7)$$

где m_1, m_2 – соответственно первый и второй статистический момент; k – окно (лаг) автокорреляционной функции.

Для вычисления автокорреляционной функции значение k выбиралось от 1 до 25, где 25 – это примерно 10% от всей длины исследуемой последовательности. Были получены следующие данные:

$r =$

	1
1	0.00917
2	0.03001
3	0.0201
4	0.01572
5	0.00697
6	0.02723
7	0.01073
8	0.02857
9	0.02676
10	0.00184

$ac =$

	1
1	0.03071
2	0.05174
3	0.03646
4	0.02118
5	0.02544
6	0.04814
7	0.01093
8	-0.04993
9	0.04401
10	-0.01128

Рис. 3. Результаты вычисления автокорреляционной функции:
r – значения автокорреляционного коэффициента для набора из *k* лагов;
ac – дискретные значения автокорреляционной функции, вычисленной по алгоритму

Алгоритм определения дискретного значения автокорреляционной функции представлен на рис. 4. Алгоритм реализован в математическом пакете MathCAD 13.

В качестве оценки значения вычисленных коэффициентов автокорреляции воспользуемся тестом «портмоне» [6]. Критерий этого теста находится по формуле

$$Q = T(T+2)\sum_{k=1}^{S}\frac{r_k^2}{T-k}, \qquad (8)$$

где T – объем выборки; S – количество вычисленных коэффициентов автокорреляции.

$$
\begin{aligned}
&\text{dev} \leftarrow 0\\
&\text{for } i \in 1..N\\
&\quad rv_i \leftarrow Values_{i+FromIndex}\\
&\text{avr} \leftarrow \text{mean}(rv)\\
&\text{for } i \in 1..N\\
&\quad \text{dev} \leftarrow \text{dev} + (rv_i - \text{avr})^2\\
&\text{dev} \leftarrow \frac{\text{dev}}{N}\\
&\text{for } j \in 1..Npoint\\
&\quad Xsum \leftarrow 0\\
&\quad \text{for } i \in 1..N-j\\
&\quad\quad Xsum \leftarrow Xsum + (rv_i - \text{avr})\cdot(rv_{i+j} - \text{avr})\\
&\quad AC_j \leftarrow \frac{Xsum}{\text{dev}\cdot(N-j)}
\end{aligned}
$$

Рис. 4. Алгоритм вычисления дискретного значения автокорреляционной функции в k точках

Для значений r и ac были получены следующие значения теста «портмоне»:

– для набора коэффициентов автокорреляции r значение теста $Q = 1.075$;

– для набора дискретных значений автокорреляционной функции ac значение теста $Q = 3.345$.

Согласно определению теста, если значение Q не превосходит критического значения критерия χ^2 для данного количества степеней свободы, то гипотеза о незначительном отклонении всех коэффициентов автокорреляции от нуля выполняется, что соответствует отсутствию периодичности в исследуемой последовательности. Как было показано в табл. 1, критическое значение коэффициента χ^2 для количества степеней свободы 6–9 лежит в пределах 16–21. Исходя из этого, можно сделать вывод, что анализ контейнера с осажденной информацией методом автокорреляции показал, что распределение байт по-прежнему равномерно распределенное, т. е. статистические характеристики существенно не изменились.

Заключение. По результатам статистического анализа байт-массива с лицензионной информацией можно сделать вывод, что предложенный метод защиты достаточно эффективный и устойчивый к статистическим атакам на предмет вычленения ключевой информации из контейнера. Комплексный анализ качества осаждения показал, что оптимальным для хранения массива ключевой информации размером 64 байта является контейнер размером 256 и 4092 байта. Извлечение ключевой информации можно осуществить в один проход контейнера, зная пару ключей K_1 и K_2, что соответствует заявленному требованию к большой вычислительной скорости алгоритма защиты лицензионной информации в Интернет-приложениях.

Литература

1. Freeman, A. Programming .NET Security / A. Free-man, A. Jones // O'Reilly Press. – June 2003. – P. 714.

2. Meier, J. D. The Lifetime of a Secure Application / J. D. Meier, A. Mackman, M. Dunner // MSDN Magazine. – March 2004.

3. Купцевич, Ю. Е. Альманах программиста / Ю. Е. Купцевич. – Т. 4: Безопасность в Microsoft .NET. – 2004. – 304 с.

4. Luby, M. How to Construct Pseudorandom Permutations from Pseudorandom Functions / M. Luby, C. Rackoff // SIAM Journal on Computing-1998.

5. Коханович, Г. Ф. Компьютерная стеганография / Г. Ф. Коханович, А. Ю. Пузыренко. – Киев: МК-PRESS, 2006. – 282 с.

6. Ljung, G. On a measure of lack of fit in time series models / G. Ljung, G. E. P. Box // "Biometrika 65", 1978.