УДК 004.056.5

ЗАСОБИ ЗАХИСТУ ІНФОРМАЦІЇ В АВТОМАТИЗОВАНИХ СИСТЕМАХ ДИСТАНЦІЙНОГО НАВЧАННЯ

В.Ю. ШАДХІН, Д.Г. ДЕЛЬ, М.О. КАРАЗІЯ

Київський національний університет технологій та дизайну

Статтю присвячено питанню захисту інформації в автоматизованих системах дистанційного навчання. Розглянуті головні проблеми та запропоновані рішення для кожної з них. Використовуючи технологію шифрування відкритим ключем та поліморфні алгоритми розроблено програмне забезпечення для комплексного захисту системи дистанційної освіти

Останнім часом багато проблем розробникам програмного забезпечення створюють незаконне копіювання і поширення програм, віруси, які постійно удосконалюються, спроби зламу хакерами різних мереж і систем. На боротьбу зі шкідливим програмним забезпеченням витрачаються величезні матеріальні ресурси. Тому захист інформації зараз є однією з найбільш важливих проблем розвитку інформаційних технологій.

Об’єктами даного дослідження є процес захисту інформації в системах дистанційного навчання. Для них характерні такі завдання з інформаційної безпеки:

– захист від несанкціонованого копіювання;

– захист від модифікації програмного коду на користь користувача;

– приховування від користувача частини інформації, тощо.

Багато з цих завдань дуже актуальні для систем дистанційного навчання і тестування.

Дуже важливою проблемою в області організації самостійної роботи і, особливо, комп’ютерного зовнішнього контролю є слабка захищеність освітнього програмного забезпечення від зламу з метою доступу до відповідей і підробки результатів контролю [1]. Ця проблема виходить з того, що в основному сучасні контролюючі системи побудовані на антропоморфному принципі, сутність якого відносно автоматизації навчання полягає у використанні пам’яті комп’ютера для зберігання еталонних відповідей разом із завданнями.

Існує також проблема захисту навчального програмного забезпечення від модифікації коду з метою зміни алгоритму оцінювання результатів тестування. Слабка захищеність будь-яких антропоморфних контролюючих систем створює труднощі при проведенні контролю в системах дистанційної освіти.. У зв’язку з цим, іспит можливий лише за умови виїзду викладача до місця зустрічі зі студентами. Але і в цьому випадку об’єктивність не гарантується, оскільки завдяки наявності відповідей в контролюючій програмі, викладач може не лише користуватися інструкціями по проведенню іспиту, але і проявляти власну ініціативу.

Таким чином, дослідження методів створення системи захисту програм дистанційного навчання мають велике практичне значення.

Основні проблеми захисту систем дистанційного навчання

Сформулюємо основні проблеми, пов’язані з захистом інформації, а також низку інших питань, які відносяться до автоматизованих систем дистанційного навчання. На жаль, перші дві проблеми лежать поза сферою можливостей програмних засобів без застосування додаткового апаратного забезпечення:

1. Відсутність можливості достовірно визначити, чи пройшов студент тестування самостійно. Для цього завдання він міг використовувати іншу людину (наприклад, більш підготовленого студента).

Це найскладніше завдання. Неможливо перешкодити студентові запропонувати комусь виконати певну лабораторну роботу або пройти тестування. Без застосування спеціальної апаратури це практично нереально. Але застосування апаратних засобів неможливе в силу хоча б своєї ціни [2]. Отже, такий варіант розглядатися не буде. Принаймні він нереальний на цьому етапі стану освіти в нашій країні.

Рішенням цієї проблеми може бути тільки правильна побудова курсу. Процес контролю знань слід будувати так, щоб ускладнити процес підміни дублером. Знайти дублера на один тест набагато простіше, ніж на весь період навчання.

2. Невідомо, скільки разів студент зробив спробу пройти тестування. Студент має можливість встановлювати систему дистанційного навчання в декількох екземплярах і/або копіювати її, тим самим зберігаючи її поточний стан. Так студент дістає можливість необмеженої кількості спроб проходження тестування і можливість вибрати з них спробу з найкращим результатом.

Це завдання аналогічне попередньому і також, на жаль, не має програмного рішення.

Розв’язати цю проблему не просто. У будь-якому випадку неможливо дізнатися, чи встановив студент пакет програм дистанційного навчання на двох комп’ютерах, а потім використовує один для тренування і підбору правильних відповідей, а другий вже для тестування. При чому він може поступити простіше і скористатися програмою для створення безлічі віртуальних машин на одній фізичній, наприклад, VMWare [3].

Ця проблема може бути вирішена правильною побудовою системи дистанційного навчання. Наприклад, при тестуванні доцільно запропонувати досить велику кількість питань, найбільш оптимальним виходом є їх автоматична генерація.

3. Існує можливість створення універсального редактора файлів результатів тестування. Він може використовуватися студентом для коригування оцінок виставлених програмою тестування.

Тут можна було б скористатися ідеєю ключової дискети/диску або флеш-драйву для збереження результату. Модифікація результату стала б дуже важкою, але цей метод має одне обмеження – необхідність використання для передачі результату дискети. Тобто замість того, щоб просто відіслати результат по мережі, доведеться доставляти його на дискеті.

Але є й інший метод – використання шифрування з відкритим ключем. Які б не були складні і надійні криптографічні системи, їх слабке місце при практичній реалізації – проблема розподілу ключів. Для того, щоб був можливий обмін конфіденційною інформацією між двома суб’єктами ІС, ключ має бути згенерований одним з них, а потім якимось чином знову ж таки в конфіденційному порядку переданий іншому. Тобто в загальному випадку для передачі ключа знову вимагається використання деякої криптосистеми. Для вирішення цієї проблеми на основі результатів, отриманих класичною і сучасною алгеброю, були запропоновані системи з відкритим ключем. Один ключ оголошується відкритим, а інший закритим. Відкритий ключ публікується і доступний будь-кому, хто бажає послати повідомлення адресату, секретний ключ зберігається в таємниці. Початковий текст шифрують відкритим ключем адресата і передають йому, зашифрований текст в принципі не може бути розшифрований тим же відкритим ключем. Розшифрування повідомлення можливе тільки з використанням закритого ключа,

який відомий тільки самому адресатові [1]. Саме такий механізм необхідно реалізувати в системі захисту. Зазначимо, що використовуватиметься шифрування за допомогою відкритого ключа не в класичному розумінні. Метод полягає в генерації поліморфних алгоритмів шифрування/розшифрування. При цьому, одному алгоритму шифрування відповідатиме один алгоритм розшифрування. Модуль повинен буде забезпечити побудову складного для аналізу поліморфного коду, що повинно перешкоджати побудові зворотного алгоритму.

4. Існує загроза створення універсальної програми перегляду файлів із завданнями і відповідями. Тим самим, студент має можливість знайти вірні відповіді на питання в тестах.

Виходом з цієї ситуації є застосування шифрування даних. Але принципово цю проблему вирішити неможливо. Студентові необхідно поставити питання і звірити з відповіддю, а для цього необхідно розшифрувати дані з еталонними відповідями, для чого потрібен ключ, який у будь-якому випадку необхідно десь зберігати. Отже, за бажанням, інформацію можна отримати у відкритому виді. Побічною проблемою є можливість внесення зацікавленою особою несанкціонованої зміни баз учбових систем. Зберігання даних у навчальній системі передбачає можливість їх перегляду, а, отже, наявність способу доступу до цих даних.

Додаткову складність повинен внести генератор алгоритмів шифрування/розшифрування. Шляхом шифрування різними алгоритмами даних, які віддаються студентові, досягатиметься додаткова складність створення універсальної програми перегляду.

5. Можливість модифікації програмного коду системи тестування з метою зміни алгоритму виставляння оцінок.

Для систем побудованих з використанням мережі Internet, цієї проблеми практично не існує, оскільки контролююча частина знаходиться на стороні сервера. Для систем дистанційного навчання, призначених для локального режиму використання, ця проблема практично зводиться до широко відомої проблеми захисту ПЗ від зламу. Очевидно, що це і є причиною того, що це питання не отримує розкриття в різних роботах, але не робить його менш важливим.

Майже кожен студент, що уперше стикається з новою для нього системою комп'ютерного тестування, прагне відшукати лазівки, що дозволяють отримувати завищені оцінки. Рано чи пізно більшість лазівок виявляються і стають надбанням усіх студентів. Боротися з цим не треба, більш того, корисно заохочувати цей процес, тому що він допомагає відшукувати недоліки захисту системи комп'ютерного тестування. Для цього, правда, в неї доводиться вбудовувати спеціальні механізми для спостереження за діями студентів, оскільки не в їх інтересах ділитися з викладачем своїми знахідками.

Як не дивно, підказати рішення можуть такі програми, як віруси. Точніше, поліморфні віруси. Поліморфною називається програма, кожен штам (копія) якої відрізняється від іншого. Два екземпляри такої програми можуть не співпадати жодною послідовністю байт, але при цьому функціонально вони є копіями [4]. Віруси використовують поліморфні генератори для ускладнення їх виявлення. Для нас поліморфний код цікавий з іншої причини – в нього дуже складно внести зміни. Точніше, внести виправлення до конкретного екземпляру програми не складає великої проблеми, а ось застосувати цей метод модифікації до іншого екземпляру неможливо.

В результаті виникає ідея побудови підсистеми за наступним описом. Система представляє з себе файл, який зберігається у зашифрованому вигляді. Програма-завантажувач розшифровує його

безпосередньо в пам'яті і потім запускає. Кожен файл зашифрований своїм методом, а отже, і проста модифікація неможлива. Можливе створення програми зламу систему тестування, яка базується на методах динамічної модифікації пам'яті програми або на створенні і завантаження копії даних в пам'ять. Але створення подібної програми вже само по собі дуже складне і вимагає високої кваліфікації.

6. Потрібна легка адаптація існуючих систем дистанційного навчання і тестування. Це, в першу чергу, пов'язано з тим, що до цих систем вже існують бази лекцій, тестових завдань і так далі.

Важливим чинником є те, що існуючі на даний момент різні системи автоматизації процесу навчання написані на різних мовах. Отже, для взаємодії з ними треба зручний і, головне, підтримуваний усіма цими мовами механізм взаємодії.

Останнім часом широке застосування знайшла технологія COM. Багато АСДН розроблено з її використанням або з використанням таких її різновидів як OLE і ActiveX. Це робить систему гнучкою, дозволяє легко її модифікувати. Взаємодія модуля захисту з використанням технології COM дуже гнучко і широко використовується для побудови модульних програм. Це дуже важливо, оскільки потрібна найлегша інтеграція у вже існуючі системи. Прикладом простоти роботи з COM-модулями може служити Visual Basic.

Постановка завдання

Метою дослідження є аналіз засобів захисту інформації АСДН без використання допоміжних апаратних засобів. Для того, щоб відмовитися від використання дорогих апаратних засобів, необхідно розробити програмне забезпечення захисту дистанційної навчальної системи шляхом шифрування виконуваних exe-файлів, яке б ґрунтувалося на використанні множини унікальних поліморфних алгоритмів, та організувати захист інформації та обмін нею з використанням ідеології відкритого ключа, яка також ґрунтується на поліморфних алгоритмах.

Результати та їх обговорення

Поліморфні алгоритми шифрування та розшифрування завжди генеруються парами, механізм їх генерації схожий і здійснюється одним кодом. Різниця тільки у тому, що використовуються блоки, які виконують зворотні перетворення [5].

Алгоритми шифрування/розшифрування складаються з восьми обов'язкових функціональних блоків, деякі з них можуть повторюватися. Для їх виконання використовується віртуальна машина, яка у свою чергу використовує віртуальні регістри та пам'ять.

Хоча кожен з блоків виконує чітко визначену функцію, він може бути реалізований багатьма способами з використанням різних віртуальних регістрів та різних комірок пам'яті. Початковий зміст віртуальної пам'яті, як і сам згенерований алгоритм, міститься у файлі. Наприклад, у віртуальній пам'яті може бути записана кількість байт, які необхідно розшифрувати. Поліморфний генератор довільним шляхом обирає, який саме регістр або комірка пам'яті буде приймати участь у кожному конкретному алгоритмі шифрування/розшифрування. У даному випадку поліморфізм – не тільки вибір і поєднання довільного набору блоків, але й їх розміщення в пам'яті. Розглянемо функції кожного блоку.

На рисунку 1 наведено загальний вигляд алгоритму шифрування/розшифрування.

Блок *1* заносить у віртуальний регістр або змінну (позначимо її як *A*1) адресу блоку даних для шифрування/розшифрування. Для віртуальної машини ця адреса насправді завжди є нулем. Річ у тому, що коли відбувається виконання віртуальної інструкції модифікації даних, то віртуальна машина додає

до цієї адреси справжню адресу в пам'яті і вже з ним проводить операції. Можна представити *A*1 як індекс в масиві даних для шифрування/розшифрування, який адресується з нуля.

Блок *2* заносить у віртуальний регістр або змінну (позначимо її як *A*2) розмір блоку даних. *A*2 виконує роль лічильника в циклі перетворення даних. Зазначимо, що її значення завжди в 4 рази менше, ніж справжній розмір даних для шифрування/розшифрування. Це пов'язано з тим, що поліморфні алгоритми завжди працюють з блоками даних, кратними за розміром 4 байтам. При чому, операції перетворення виконуються над блоками, кратними 4 байтам.

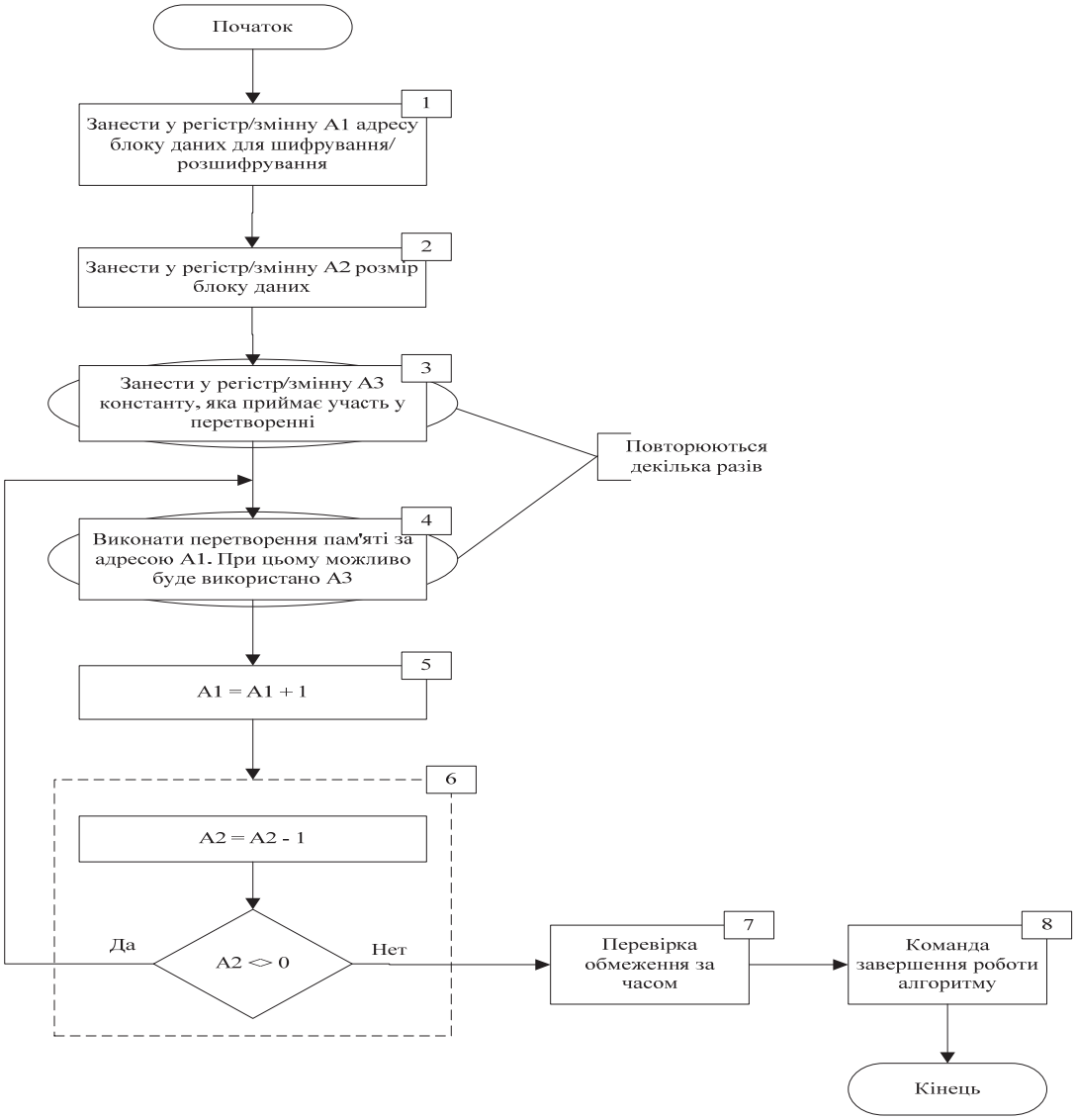

Рис. 1. **Алгоритм шифрування/розшифрування у загальному виді**

Блок *3* заносить у віртуальний регістр або змінну (позначимо її як A3) константу, яка бере участь в перетворенні. Ця константа, можливо, потім і не буде використана для перетворення даних, усе залежить від того, який код буде згенерований. Блок 3 може бути повторений кілька разів. Над даними здійснюється цілий набір різних перетворень, і в кожному з них беруть участь різні регістри/змінні, які було ініціалізовано у блоці 3.

Блок *4* можна назвати основним. Саме він, а, точніше, набір цих блоків проводить шифрування/розшифрування даних. Кількість цих блоків випадкова і дорівнює кількості блоків номер 3.

При перетвореннях не обов'язково буде використано значення з A3. Наприклад, замість A3 може використовуватися константа або значення з лічильника. На даний момент поліморфний генератор підтримує 3 види перетворень: побітове XOR, складання і віднімання. Набор цих перетворення можна легко розширити, головне, щоб таке перетворення мало зворотну операцію.

Блок *5* служить для збільшення A1 на одиницю. Як і в усіх інших блоках ця операція може бути виконана по-різному, тобто з використанням різних елементарних інструкцій віртуальної машини.

Блок *6* організовує цикл. Він зменшує значення A2 на одиницю, і якщо результат не дорівнює 0, то віртуальна машина переходить до виконання блоку 4. Насправді управління може бути передане на один з порожніх блоків між блоком 3 та 4, але з функціональної точки зору це значення не має.

Блок *7* проводить перевірку обмеження за часом використання алгоритму. Код по перевірці на обмеження за часом відноситься до порожніх команд і, насправді, може бути присутнім і виконуватися в коді велику кількість разів. Саме тому він і винесений як один з функціональних блоків. Якщо ж при генерації алгоритму від генератора не вимагається обмеження за часом, то як аргумент до віртуальної команди перевірки часу використовується спеціальне число.

Блок 8 завершує роботу алгоритму.

Висновки

У результаті дослідження було розроблено засоби захисту дистанційної навчальної системи шляхом шифрування виконуваних exe-файлів без використання додаткових апаратних засобів, яке ґрунтується на використанні множини унікальних поліморфних алгоритмів, та організовано захист інформації та обмін нею з використанням ідеології відкритого ключа, в основі якої також лежать поліморфні алгоритми. У розробленому ПЗ використано новітні технології шифрування даних, розроблено генератор поліморфних алгоритмів шифрування та розшифрування, повністю виключена необхідність використання апаратних засобів.

Завдяки використанню даного ПЗ у майбутньому з'явиться можливість зменшити на 80 відсотків несанкціоноване використання копій навчальних матеріалів та на 90 відсотків зменшити ймовірність використання "дублерів" під час здачі контрольних тестів.

ЛІТЕРАТУРА

1. Шаньгин В.Ф. Информационная безопасность компьютерных систем и сетей: Уч. Пособ. – М.: Форум, – 2008. – 416 с.
2. Ложников П.С. Распознавание пользователей в системах дистанционного образования: обзор // Educational Technology & Society. – 2001. – № 4, http://ifets.ieee.org/russian/depository/v4_i2/html/4.html
3. Ерижоков А.А. Использование VMWare 2.0 // Публікація у мережі □абичев□ на сервері http://www.citforum.ru/operating_systems/vmware/index.shtml
4. Касперский Е.В. Компьютерные вирусы: что это такое и как с ними бороться. – М.: СК Пресс, – 1998. – 288 с.
5. Сабичев С.Г. Основы современной криптографии. – М.: Горячая линия – Телеком, – 2001. – 120 с.

Надійшла