СХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ім. В.ДАЛЯ
ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА ЕЛЕКТРОНІКИ

КАФЕДРА ПРОГРАМУВАННЯ та МАТЕМАТИКИ

Пояснювальна записка

до дипломної роботи
магістра
(освітньо-кваліфікаційний рівень)

на тему «Розробка системи контролю авторизації програмного забезпечення користувача»

Виконав: студент 6 курсу, групи ІНФ-16ДМ
напряму підготовки 6.040302 „Інформатика”
спеціальності 8.04030201 „Інформатика”
_____ Задорожній Є.О.
(підпис)

Керівник,
Д.е.н., проф_____ Даніч В.М.
(підпис)

Рецензент,
доцент, к.т.н._____ Іванов В.Г.
(підпис)

Сєвєродонецьк - 2018

ЗМІСТ

ВСТУП

Актуальність досліджень. Незважаючи на всі зусилля різних організацій на чолі з Business Software Alliance (BSA) – торгової асоціації, що представляє інтереси ряду найбільших у світі розробників програмного забезпечення, в останні роки триває ріст комп'ютерного піратства. У середньому частка піратського програмного забезпечення в глобальному масштабі становить 40%, тобто кожні чотири з десяти копій програми виявляються украденими у розроблювача-виробника й позбавляють його прибутку. По розрахунках BSA в 2014 році збитки софтверної галузі від піратства склали порядку 13 мільярдів доларів [21]. Для рішення задачі забезпечення захисту розроблювального програмного забезпечення необхідна сукупність обладнань і дій, призначених для збереження конфіденційності даних від несанкціонованого доступу. Найбільш перспективним для захисту недорогих умовно-безкоштовних програм є метод авторизації через Інтернет [9]. Він припускає первісну активацію продукту на обчислювальній машині користувача, а також авторизацію користувача на сервері розроблювачів при кожному запуску програми. Така авторизація користувача при кожному запуску програми дозволяє розроблювачеві стежити за статистикою використання програми, виявляти випадки порушення ліцензій, позбавляти ліцензій несумлінних користувачів, а також гнучко змінювати ліцензійну політику у відповідності зі своїми потребами. Тому одержання засобу технічного захисту при розробці програмного забезпечення є актуальною задачею.

Об'єкт досліджень: мережна модель контролю авторизації розроблювального програмного забезпечення.

Предмет досліджень: мережний протокол авторизації розроблювального програмного забезпечення.

Мета дослідження: розробка програмного комплексу, що дозволяє контролювати авторизацію розробленого програмного забезпечення користувача.

Завдання дослідження:

a) розробити об'єктну модель ідентифікації й аутентифікації й запропонувати протокол авторизації програмного забезпечення;
b) розробити динамічну бібліотеку, що підключається до користувацького програмного забезпечення й контролює його авторизацію;
c) для препроцесора гіпертексту PHP, що працює на http-сервері Apache розробити сценарії взаємодії динамічної бібліотеки й бази даних MYSQL, написати сценарій створення web-форми адміністрування.

Методологічна й теоретична основа дослідження: засоби захисту програмного забезпечення, метод авторизації через Інтернет.

Практичне значення отриманих результатів. Отримані результату роботи й засіб технічного захисту для розроблювального програмного забезпечення можуть бути практично використані розроблювачами програмного забезпечення при реалізації його захисту від несанкціонованого використання й контролю авторизації програмних продуктів.

РОЗДІЛ 1 АНАЛІТИЧНИЙ ОГЛЯД ЗАСОБІВ ЗАХИСТУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

У першому розділі будуть розглянуті рівні програмного захисту, методи й програмні засоби забезпечення захисту інформації.

1.1 Види й рівні засобів програмного захисту

Виробники програмного забезпечення зазнають більших збитків через нелегальне використання їх продукції, так званого “піратства”. Найчастіше юридичні методи боротьби із правопорушниками не є ефективними, тому для захисту своїх інтересів розроблювачам доцільно прибігати до технічних засобів захисту програмного продукту від нелегального використання. Під захистом в інформаційних технологіях розуміють сукупність дій, спрямованих на протидію злому, нелегальному копіюванню й несанкціонованому доступу. До технічних методів захисту відносять програмні й програмно-апаратні засоби, а також використання програм як онлайн-сервісів [1, 2].

До програмних відносяться методи [2], реалізовані софтверним і алгоритмічним шляхом, що припускають використання прив'язки програми до конфігурації обладнання, на якім її встановили. У них не зачіпаються фізичні характеристики носіїв інформації, спеціальне устаткування. Прив'язка відбувається в момент установки програмного забезпечення й вимагає або введення ліцензійного ключа, або проходження процедури активації – одержання одноразового коду (ключа), що залежить від конфігурації обладнання користувача й придатного для використання тільки на цьому комп'ютері. У першому випадку нічого не заважає користувачеві незаконно поширювати продукт разом з ліцензійним ключем. У другому випадку сумлінному користувачеві необхідно погодитися з незручностями,

які виникають при кожній зміні обладнання, на якім він використовує програмне забезпечення, а несумлінний користувач має можливість незаконно використовувати й поширювати програмне забезпечення разом з так званою віртуальною машиною [3], що програмно імітує необхідне конфігураційне обладнання.

Програмно-апаратні засоби захисту [1, 2] припускають перевірку наявності деяких апаратних засобів, які поставляються разом із програмою й для яких неможливо, або дуже важко виготовити копію. Наприклад, широко використовують спеціально виготовлені CD, DVD і апаратні електронні ключі, що підключаються до Usb-портам комп'ютера, щоб ідентифікувати оригінальну версію програми й захистити продукт від нелегального використання. Такі методи вважаються більш надійними, ніж програмні, але вони мають істотні недоліки. По-перше, таке програмне забезпечення можна поставляти тільки в «коробковій» версії, по-друге, такий засіб захисту може суттєво збільшити вартість програмного продукту, по-третє, користувачеві також потрібно погодитися з деякими незручностями. У зв'язку із цим даний метод захисту не придатний для більшості недорогих і використовуваних у повсякденнім житті програмних продуктах.

Крім того, відомі [2] кількаразові випадки злому захисту обох типів і подальшого нелегального й безперешкодного поширення програмного забезпечення.

Самим надійним засобом від копіювання програм можна вважати виконання їх як онлайн-сервісів, так званих Software as a service (SaaS) («Програмне забезпечення як послуга») [4], які припускають виконання програмного забезпечення на стороні виробника, не доступної для користувача. Користувач же може тільки ввести вхідні й одержати вихідні дані через веб-інтерфейс. Такий вид захисту мало придатний для програмного забезпечення, яке вимагає більших об'ємів вхідних або вихідних

даних, і в кожному разі він не зможе замінити традиційні способи поширення програм.

Останнім часом для розроблювачів перспективним є надання так званої умовно-безкоштовної (Shareware) версії програми, коли користувач може використовувати повністю робочу версію програми обмежений час. Однак така можливість найчастіше є слабким місцем у захисті програмного забезпечення, оскільки обмеження на використання традиційно виконується шляхом перевірки проміжку між поточним часом і часом першого запуску програми або підрахунку кількості запусків програми, а результат зберігається в пам'яті комп'ютера. У такому випадку з'являються так звані «кряки» для скидання лічильника й забезпечується можливість користуватися програмою необмежений час.

Розглянемо сфери й рівні захисту (рис. 1.1.1), які використовуються в сучасних програмних продуктах.

Рисунок 1.1.1 - Сфери програмного захисту

Існують наступні рівні захисту програмного забезпечення:

- технічні: використання серійних номерів для активації продукту, сканування мережі на предмет пошуку двох копій одного продукту, що працюють із застосуванням однакового ключа активації, використання електронних ключів і т.і.;

- ❖ юридичні: залучення порушників до відповідальності, передбаченої чинних законодавством;
- ❖ організаційні: необхідність внесення передоплати для використання пробної версії програмного продукту, неможливість його використання без звернення до служби технічної підтримки і т.і.

Захист від шкідливого програмного забезпечення дозволяє не тільки забезпечити цілістність конфіденційної інформації, але й зберегти працездатність програмних продуктів, установлених на комп'ютері (рис. 1.1.2).

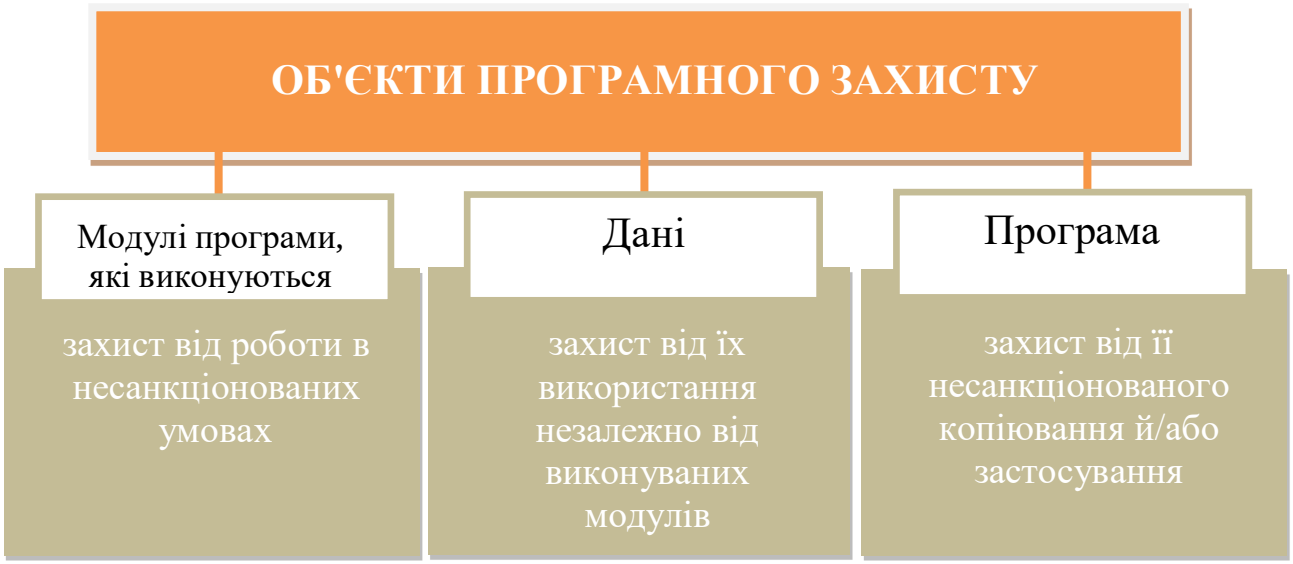

Рисунок 1.1.2 - Об'єкти програмного захисту

Програмні засоби забезпечення захисту інформації являють собою сукупність обладнань і дій, призначених для збереження конфіденційності даних.

Існують наступні категорії засобів програмного захисту:

1) власний захист: засоби, вбудовані безпосередньо в програмне забезпечення для захисту інформації на всіх стадіях його використання;
2) засоби, що є частиною операційної системи: брандмауери;
3) засоби захисту із запитом інформації: для здійснення повноцінної роботи із програмним продуктом потрібно ввести код доступу;

4) активний захист: використовуються при спробах несанкціонованого впровадження в програмне середовище, наприклад, неправильнім уведенні пароля або порушенні порядку входу;
5) захист від шкідливого програмного забезпечення за допомогою спеціалізованих антивірусних програм;
6) пасивний захист: пошук відомостей, що підтверджують факт спроби вторгнення в програму й розкрадання даних.

1.2 Засоби захисту програмного забезпечення

Тому що методи захисту програмного забезпечення можна розділити на програмні й апаратні, то засобами захисту програмного забезпечення є фізичні обладнання й програмні рішення, що дозволяють позбавити програмне забезпечення від несанкціонованого впливу ззовні.

Найбільш ефективними й часто використовуваними є технічні засоби захисту програмного забезпечення (рис. 1.1.3):

1) Локальний програмний захист. Застосування серійних номерів (ключів), які можуть бути використані тільки для однієї копії програми. Цей метод ефективно використовувався тоді, коли програми поширювалися тільки на фізичних носіях (приміром, компакт-дисках). На коробці з диском був надрукований серійний номер, що підходить тільки до даної копії програми. З поширенням мереж очевидним недоліком стала проблема поширення образів дисків і серійних номерів по мережі. Тому в даний момент метод використовується тільки в сукупності одним або більш інших методів (приміром, організаційних). На цей час використовується в комплексі з іншими методами.

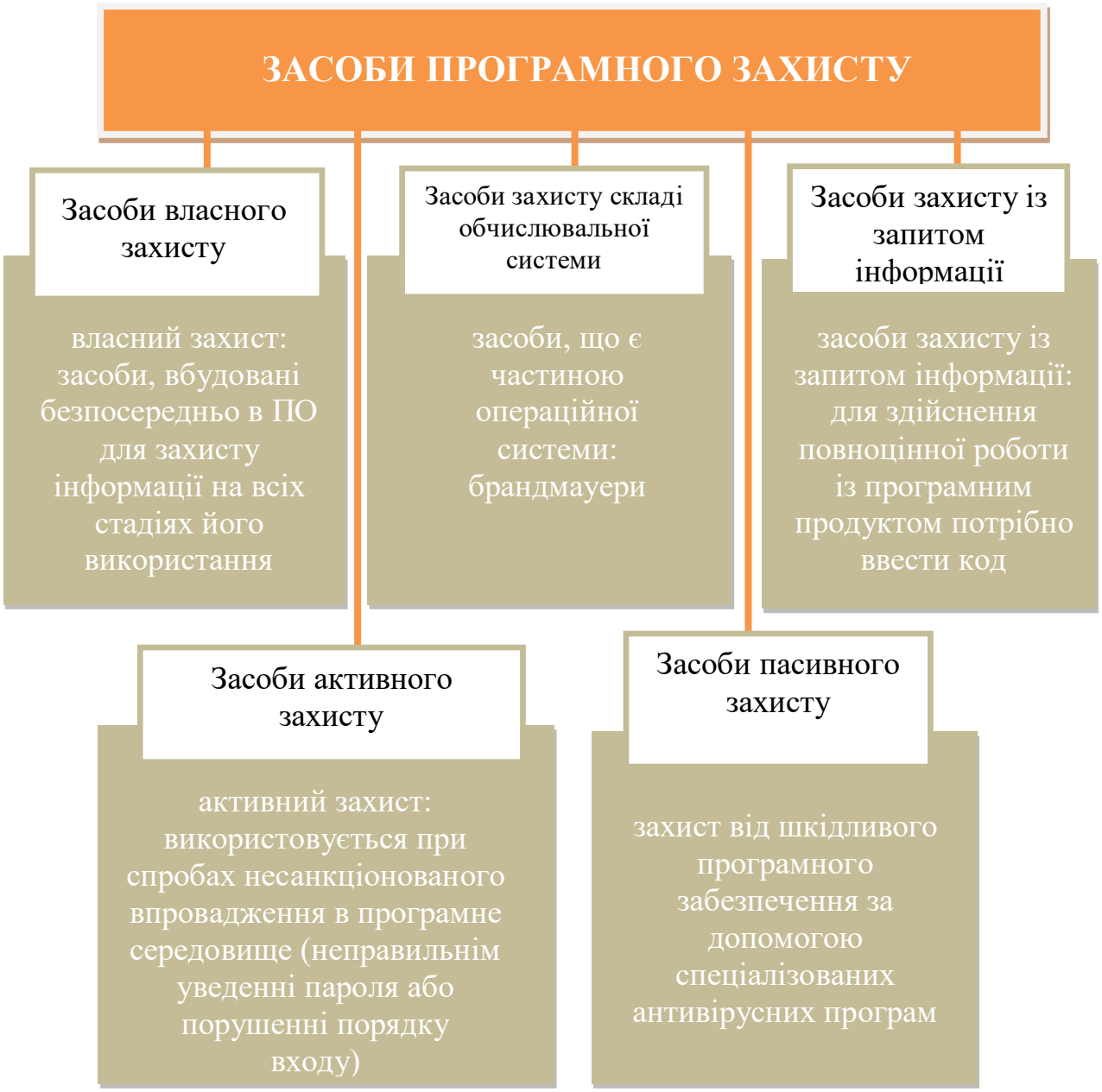

Рисунок 1.1.3 - Засоби програмного захисту

2) Мережний програмний захист:

а) локальна, здійснювана шляхом сканування мережі на предмет наявності двох однакові реєстраційні ключів у різних користувачів, і їх блокування.

Недолік захисту полягає в тому, що брандмауер можна настроїти так, щоб він не пропускав пакети, що належать захищеній програмі. Крім того, програма може взаємодіяти по мережі (приміром, при організації

мережної гри). У цьому випадку брандмауер повинен пропускати такий трафік.

b) глобальна, здійснювана за допомогою використання серверів, з якими повинен зв'язатися користувач копії програмного продукту для забезпечення його нормальної працездатності. У тому випадку, якщо використовуваний ключ є не унікальним, можливість роботи із програмою буде заблокована.

Крім технічних способів, системи захисту програмного забезпечення можуть використовувати апаратні засоби захисту програмного забезпечення, представлені різними електронними обладнаннями, наприклад, пристосуваннями для шифрування інформації.

1.2.1 Проста перевірка ушкодженого носія

Даний механізм дозволяє захистити програму від несанкціонованого копіювання й застосування, а також захистити виконувані модулі програми — від роботи в несанкціонованих умовах. Певна ділянка заданого файлу, що не беруть участь в роботі програми, технологічно ушкоджується (наприклад, поверхня компакт-диска — на заводі лазерним променем). Таким чином, певний фрагмент файлу стає таким, що не читається. Далі у виконуваний код прошиваються інструкції з перевірки його наявності, читання, порівняння його розміру, відповідності кількості байтів у неушкодженій його області з еталоном і т.д. Залежно від того, чи коректно пройшла перевірка, можна ухвалювати рішення щодо подальшого розвитку подій у коді, що виконується, програми.[17]

Позитивна якість: простота й швидкість реалізації.

Недоліки: потрібне апаратна участь із боку заводу виробника, досить хитка й у принципі, що легко зламується.

Деякі сучасні засоби дозволяють переписати файл захисту без ушкодженого фрагмента, що згодом дозволить відтворити його, замінивши ушкоджені ділянки еквівалентною кількістю байтів «баласту».

Перераховані вище методи й засобу захисту програмного забезпечення дозволяють забезпечити його цілісність, невразливість для впровадження сторонніх шкідливих програм, а також захистити його правовласника від несанкціонованого використання ліцензійних продуктів.

1.2.2 Обмеження числа установок програми

Дозволяє обмежити число установок системи на комп'ютер. Застосовується в основному при установці програмного продукту з дискет, при цьому при кожній установці з дискети «вичитується» значення кількості установок, що залишилися. Застосовується, як правило, у комбінації з перевіркою ушкодженого носія, роблячи тим самим процес копіювання один в один неможливим.

Позитивна якість: такий захист вигідний як для розроблювача, так і для користувача програмного продукту. Розроблювач може поширювати програмний продукт за ціною, що залежить від кількості дозволених інсталяцій.

Недоліки: потрібне апаратна участь із боку заводу виробника; захист не може бути застосована самостійно (без перевірки носія на предмет наявності дефекту поверхні).

1.2.3 Обмеження числа запусків програми

Дозволяє обмежити число запусків програми. При кожному запуску програми віднімається значення кількості запусків, що залишилися.

Позитивна якість: має гарну рекламну властивість.

Недоліки: для злому досить визначити місце в системі (файл або системний реєстр), куди заноситься інформація про кількість запусків, що залишилися.

1.2.4 Контрольні питання

Захищає виконувані модулі програми від роботи в несанкціонованих умовах. Як правило, спрацьовує після закінчення певного строку від дня застосування. Користувачеві задається контрольне питання (питання), що вимагає наявності в нього ліцензійної версії програмного продукту, наприклад про написи на поверхні компакт-диска, або слова із прикладеної до програмного забезпечення інструкції. Таким чином, може бути забезпечена прив'язка до поверхні компакт-диска, коробці продукту або посібнику користувача. Основний зміст тут полягає в наступному: припустимо, що потенційному зломщикові вдалося запустити програму не санкціоновано, тираж піратських копій їм уже підготовлений і майже розпроданий, і отут виявляється, що програма вимагає введення третього слова з п'ятнадцятого рядка на останній сторінці довідкового керівництва.

Позитивна якість: механізм уповільненого спрацьовування, при вдалім розосередженні запитів по виконуваному коду захист, що важко зламується.

Недоліки: механізм не може застосовуватися як основний засіб захисту програми або навчального курсу.

1.2.5 Тимчасові версії

Захищає виконувані модулі програми від роботи в несанкціонованих умовах. Спрацьовує по даті або по кількості запусків. Програма попросту перестає працювати починаючи, наприклад, з 2000 року. Застосовується в основному для створення тимчасових, демонстраційних, так званих

«умираючих» (trial) версій. Можливе застосування іншого варіанта, наприклад програма перестає працювати під керуванням операційних систем, чиї системні файли обновлялися пізніше 2000 року (наприклад, system.ini, win.ini, control.ini ...).

Позитивна якість: забезпечує захист у випадках, коли створення повноцінного захисту в силу певних обставин неможливо. Один з найдешевших з погляду виробництва тиражу варіантів захисту (немає необхідності в складних технологічних процесах ушкодження поверхні, ключових дискетах і т.д.).

Недоліки: перший варіант попросту можна обійти, змінивши системну дату, хоча це й трохи незручно для користувача.

1.2.6 Версії, що працюють із обмеженнями

Захищає виконувані модулі програми від роботи в несанкціонованих умовах. Змушує користувача обмежуватися лише деякими функціями або часом роботи програми в одному сеансі запуску. У той же час може мати механізм перекладу в повноцінний режим. Наприклад, при запуску генерується унікальний для даного комп'ютера й сукупності програмного забезпечення номер по певному алгоритму, рознесеному у виконуваному коді. Повідомивши це число компанії розроблювачеві програмного забезпечення, користувач одержує число-відповідь, уведення якого розкриває всі можливості програми або знімає всі тимчасові або інші обмеження при її роботі.

Позитивна якість: має гарну рекламну властивість. Програма в такому випадку може поширюватися безкоштовно, наприклад у якості додатка до журналу або по мережі Internet. У такий спосіб з користувача знімається необхідність купувати «кота в мішку», а з розроблювача — витрати на рекламу, доставку, телефонні розмови або поштові перекази.

Недоліки: при переустановці програмного забезпечення комп'ютера буде потрібно повторна перереєстрація копії програми. Крім того, у випадку з обмеженими функціями проте програма в принципі працює, і це може влаштувати деяких користувачів.

1.2.7 Аналіз поверхні ушкодженого носія

Захищає від несанкціонованого копіювання й застосування, а також виконувані модулі програми від роботи в несанкціонованих умовах. Такий захист складний у реалізації. Повторює ті ж дії, що й проста перевірка ушкодженого носія, однак крім усього іншого ще й перевіряє обов'язкова наявність фізичного дефекту поверхні носія в певному місці, роблячи, таким чином, механізм «злому», описаний раніше, непридатним.

Позитивна якість: захист досить надійний і має найменший ступінь зламу.

Недоліки: захист складний у реалізації через те, що різні обладнання носіїв (зокрема, CD-ROM) поводяться зовсім по-різному при спробі читання ушкодженої області. Наприклад, переважна більшість обладнань CD-ROM (приблизно 85%) попросту повертають помилку читання. Однак деякі обладнання зависають геть-чисто, що створює необхідність у реалізації обхідного варіанта для цих випадків. Доводиться штучно відловлювати завислий процес, наприклад по таймеру, і «убивати» вручну, наприклад програмним відкриттям або закриттям лотка обладнання (тому що всі інші методи, як правило, ні до чого не приводять) або перезавантаженням комп'ютера. В останньому випадку, зрозуміло, слід подбати про механізм реєстрації подій, що фіксує реалізацію обхідного варіанта, що й запускає для таких обладнань допоміжний захист, наприклад «контрольні питання». Може створювати певний дискомфорт при запуску програми (простої, перезавантаження, зависання).

1.2.8 Апаратні ключі (dongles)

Захищають від несанкціонованого копіювання й застосування, а також виконувані модулі програми — від роботи в несанкціонованих умовах. програмне забезпечення, Що захищається, вимагає для своєї роботи наявності апаратного ключа, установлюваного на порт принтера комп'ютера. Такий ключ виготовляється й поширюється разом із програмним забезпеченням. Кожний ключ має свій унікальний серійний номер, який деблокується при введенні унікального значення для кожної копії програми. Якщо такий ключ програмою не виявлений, то вона або не запускається, або запускається з обмеженнями.

Позитивна якість: висока надійність механізму. Усі ідентифікаційні коди, паролі, а також формули їх одержання перебувають не в оперативній пам'яті комп'ютера й не на жорсткому диску або дискеті, а в мікросхемах апаратного ключа, будучи тим самим практично недоступними.

Недоліки: складність і дорожнеча реалізації. Дискомфорт при роботі (можливі проблеми з портом і принтером). Механізм непридатний для продажів або поширення програмного забезпечення засобами Internet.

1.2.9 Мережний захист

Забезпечує роботу обмеженого числа зареєстрованих копій програмного продукту в локальній обчислювальній мережі. Захист може бути реалізована як контроль над числом одночасно працюючих копій програми із сервера додатків або в автономному (локальному) режимі. Механізм вимагає наявності системи спостереження за числом одночасно працюючих копій і повинен забезпечувати постановку користувачів у чергу з автоматичним запуском або без автоматичного запуску копії, що звільнився. Крім того, розроблювач додатка, що бажає захистити його в локальній обчислювальній

мережі, повинен подумати й про механізм «дозакупки» робочих місць такої системи.

Позитивна якість: захист корисний при розробці корпоративного продукту.

Недоліки: складність реалізації (необхідність підтримки різних протоколів локальних мереж).

1.3 Огляд сучасних комплексів побудови захисту

Розглянемо сучасні комплекси побудови захисту програмного забезпечення. Інструментальний пакет *Crypto++ SDK **0*** **(**Copyright © 1997-1999 Sampson Multimedia.) складається з динамічних бібліотек OCX, DLL, що контролюють увесь процес захисту, а також допоміжного засобу Key Calculator Wizard, що дозволяє генерувати знакові послідовності для захисту додатків.

Crypto++ SDK застосовується для захисту програмних продуктів методом “умираючих” версій і призначений для їхнього поширення за допомогою Internet. При цьому використовуються механізми обмежень кількості запусків, інсталяцій і обмеження по даті роботи програми. Крім цього в арсеналі Crypto++ є також засоби по підтримці механізму прив'язки програмного продукту до конкретного комп'ютера апаратно. Пакет містить бібліотеки, що динамічно підключаються, підтримують багато розповсюджених мов програмування, такі як: Visual Basic, Visual C++, Delphi, C++ Builder і Borland C++ . Підтримується як Win32 так і Win16.

Крім того, в Crypto ++ вбудована підтримка Activex і Director Xtras, остання дозволяє захищати додатки, створені за допомогою Macromedia Director так звані прожектори (projectors).

Тимчасові (1 рік) версії засобів розробки механізмів захисту програмних продуктів за допомогою Crypto++ SDK 3.0 можна скачати із сайта компании Sampson Multimedia. При цьому Вам поштою надішлють список паролів для

розпакування завантажених Вами модулів (за умови, що Ви вказали свою електронну адресу під час реєстрації).

Інструментальний пакет *Gus* **(**Gus Communications, Inc.)також являє собою бібліотеки OCX для автоматизації розробки захисту програмного забезпечення. Механізм дії пакета схожий на механізм дії Crypto++. Захист, реалізована за допомогою Gus Software Protection OCX повністю програмна й перетворює зареєстровані копії, перенесені на інші комп'ютери в тимчасові.

Системні вимоги: Windows 95/98 або Windows NT; Будь-який OCX (Activex) сумісна мова програмування (Delphi, Visual Basic, Microsoft Visual C++ і т.д.).

Розроблювачі пропонують використовувати два способи захисту додатків – за допомогою *Cripkey Software Developer Kit* **(**Kenonic Controls Ltd.)або програмної реалізації тих же функцій Crypkey Instant. Програма автоматично встановлює Crypkey у готовий EXE (як Win 16 так і Win32). Далі майстер забезпечить вибір необхідних опцій захисту (в основному це опції обмеженої версії).Очевидно, що використання такого простого інструмента для створення надійного захисту досить сумнівно, це набагато реальніше зробити за допомогою SDK надаваного розроблювачем.

Основною відмінною рисою Crypkey є те, що він дозволяє передавати зареєстровану версію від комп'ютера до комп'ютера, деінсталюючи при цьому первинну версію.

Крім цього, Crypkey може контролювати використання програмного продукту в локальній обчислювальній мережі. Це означає, що якщо користувач заплатив за 5 копій програмного продукту, те 6-я копія із сервера додатків у локальній мережі не запуститися. Реалізована система не залежить від використовуваного протоколу (IPX, Netbios або Winsockets). Примітно ще те, що при захисті програмного продукту засобами Criptkey немає необхідності в розробці множини виконуваних модулів для підтримки різних платформ і мережних протоколів. Система постачена автоматичним

менеджером черги додатків, який повідомляє користувачів черги про копію, що звільнився.

Із усього спектра програмних рішень по захисту, розроблювальних компанією *Microcosm Copy Protection*, розглянемо два наступні:

Unlock-it - програма дозволяє реалізувати механізм обмежених версій і не вимагає наявності додаткових апаратних засобів (дискет або апаратних ключів). Має можливість мережного захисту й антивірусного захисту виконуваних модулів. Unlock-it підтримує режим так званого наймання додатків, при якому користувач може «замовити» певна кількість запусків або певний час роботи додатка. Unlock-it сполучимо з операційними системами: старше MS-DOS 2.0, PC DOS або Novell DOS, Concurrent DOS, Windows 3.x, NT і з мережами DOS або Windows (включаючи Novell).

Dinkey Dongles - реалізує механізм апаратного ключа, використовуючи для цього порт принтера. Підтримує чотири режими Dinkey Dongle: Dinkey 1, Dinkey 1S, Dinkey 2 і Dinkey Net.

Dinkey 1: унікальний серійний номер, привласнений апаратному ключу, дозволяє захистити лише один додаток;

Dinkey 1S: серії апаратних ключів привласнюється той самий серійний номер. Таким чином, розроблювач може тиражувати програмне забезпечення без його індивідуалізації, для використання з певним апаратним ключем;

Dinkey 2: дозволяє реалізувати тимчасові або обмежені версії на базі апаратного ключа, а також одночасний захист до 17 додатків. Має можливість дистанційно змінювати параметри захисту.

Dinkey Net: являє собою мережну версію апаратного ключа Dinkey 2. При установці в локальній обчислювальній мережі реалізує механізм мережного захисту. Має можливість дистанційно змінювати параметри захисту.

Інструментальний пакет *PC Guard* **(**Copyright © by Blagoje Ceklic)являє собою систему захисту програмних продуктів, що дозволяє повністю автоматизувати цей процес. PC Guard досить простий в обігу. Принцип його

роботи полягає в тому, що навколо програми, що захищається, створюється конверт, що захищає її від аналізу потенційним зломщиком або копіювання з одного комп'ютера на інші. Чудово те, що PC Guard легко комбінується з будь-яким іншим механізмом захисту. Крім того, в PC Guard для Win32 вбудована функція захисту додатка паролем. Захистити можна безпосередньо модулі EXE або DLL. Цей механізм не вимагає вбудовування бібліотек у код програмного продукту, що захищається, забезпечуючи, таким чином, підтримку практично будь-якої мови програмування. Захисний конверт складається з декількох захисних шарів, кожний з яких реалізує відповідну задачу (захист від аналізу виконуваних модулів, захист із використанням механізмів обмежень, захист паролем).

PC Guard дозволяє забезпечити прив'язку програмного продукту, що захищається, до апаратури комп'ютера, а саме: до жорсткого диска, до дати BIOS і до встановленої операційної системи. PC Guard працює під керуванням операційних систем DOS,Windows 3.x (Win16) і Windows 95/98/NT (Win32).

Ще один інструмент для розроблювачів *Safeserial OCX для Windows 95, 98, NT, 2000* **(**Sikander Soft Inc.),що дозволяє автоматизувати процес захисту програмного продукту.

Підтримує створення ключових дискет, тимчасових і обмежених версій продукту, настроюється для контролю над мережними версіями, а також постачений механізмом «переміщення ліцензії» з одного жорсткого диска на іншій, що таки зручно.

Слід зазначити, що відмінною рисою бібліотек OCX у порівнянні зі стандартними DLL є максимальна простота вбудовування в код розроблювальної програми. Необхідно всього лише «перетягнути» компонент керування на основну «форму» додатка, що захищається. Наочним прикладом тому служить Safeserial OCX. Додайте до цього ще й набір усіляких стандартних повідомлень і діалогових вікон на 8 різних мовах.

Бібліотека підтримує мови програмування Delphi (тільки Win32), Visual Basic, C++ Builder і інш, що підтримують механізм Activex. Для одержання більш детальних відомостей про роботу системи розроблювачі радять прочитати посібник з використання Safeserial OCX.

Компанія ©1999 Copy-Protection.com розробила й реалізувала механізм апаратного ключа у своєму продукті *WIBU-KEY Copy Protection System*. Основною областю застосування розробленої технології обраний захист програм, що працюють в обчислювальних мережах.

Апаратні ключі розробляються на базі чипів ASIC (Application Specific Integrated Circuit) і сумісні з персональними комп'ютерами як PC так і Macintosh.

Lock & Key **(**Timeless Technologies, Inc.)розроблений для платформ Windows 3.x, 95/98, NT і повністю сполучимо з Delphi версій 1, 2, 3, 4. Являє собою досить просту реалізацію стандартних алгоритмів по створенню тимчасових або обмежених версій програмних продуктів.

Інструментальний пакет *Windows Protection Tool-Kit*™ (Advanced Software Technologies)працює під керуванням Windows 3.x, 95/98, NT. Оснащений механізмом прив'язки до ключової дискети, що працюють, щоправда, тільки під керуванням Windows NT. Може захищати програми для роботи в локальних мережах на базі Windows NT, Lantastic або Novell. додатки, що захищаються, можуть прив’язуватися до жорсткого диска або до локальної мережі. За бажанням можна створювати версії з обмеженням кількості інсталяцій, що й навіть уміють повертати кількість інсталяцій при деінсталяції захищеного програмного продукту.

Однак система захищає програмний продукт лише на стадії його установки (setup).

Пакет Windows Protection Tool-Kit™ сполучимо з такими відомими пакетами автоматизації створення інсталяцій, як Install Shield і Visual Basic's Setup Wizard, а розроблявся він для WISE Installation System.

Інструментальний комплекс *Sheriff* (Acudata Limited)пропонує ще один засіб розробки механізму програмного захисту додатків (тимчасових і обмежених версій) – Sheriff Software Development Kit (SDK) для Win32.

Sheriff представлений користувачеві досить гнучким набором бібліотек: як стандартної Windows DLL, так і компонентом Activex (OCX control).

На відміну від рішень, розглянутих раніше, API системи Sheriff пропонує самостійно настроїти всі характеристики створюваної версії програмного продукту, а не заповнити поля в стандартнім діалоговім вікні. Пропонований SDK досить гнучкий і включає класи й демонстраційні приклади на різних мовах з використанням різноманітних технологій: Visual C++, Visual Basic, Visual Foxpro і Delphi, а також Activex Control. Слід зазначити, що Sheriff має досить більший набір механізмів захисту від резидентного моніторингу, трасування в відладчиках або перестановки системної дати або дат зміни різних системних файлів.

Компанія Aladdin [6] розробила *Hardlock Bistro* **(**Aladdin Knowledge Systems GmbH & Co.) – цілу серію засобів для захисту додатків від копіювання й злому, засновану на використанні апаратних Usb-ключів. Захист орієнтований у першу чергу на ринок корпоративних користувачів, однак мало придатною для невеликих недорогих програм. Aladdin пропонує найбільш повний набір засобів автоматизації захисту додатків, що входять до складу Hardlock Bistro: Espresso Wizard, Espresso, Cappuccino і Lattecino, і дев'ять різних апаратних прив'язок для забезпечення функціонування різних механізмів апаратних залежностей. Являє собою інтерфейс найвищого рівня між властиво модулем Hardlock програмою, що й захищається. Ледве «нижче» розташовується наступний модуль – програма Espresso, що дозволяє захищати програми в автоматизованому режимі, а для аматорів програмувати Aladdin пропонує реалізацію API – Lattecino.

1.4 Постановка задач магістерської роботи.

Тож після розгляду сучасних комплексів побудови захисту програмного забезпечення ми зауважимо на поставлених завданнях даної роботи та умовно розділимо подальшу роботу на три етапи:

1. Спочатку потрібно розробити об'єктну модель ідентифікації й аутентифікації й побудувати протокол авторизації програмного забезпечення;
2. Потім розробити динамічну бібліотеку, що підключається до користувацького програмного забезпечення й контролює його авторизацію;
3. для препроцесора гіпертексту PHP, що працює на http-сервері Apache розробити сценарії взаємодії динамічної бібліотеки й бази даних MYSQL, написати сценарій створення web-форми адміністрування.
4. Та провести тестування основних режимів роботи та сценаріїв протоколу. Зробити порівняльну характеристику представленої системи з іншими системами захисту програмного забезпечення.

РОЗДІЛ 2 ІНФОРМАЦІЙНА МОДЕЛЬ АВТОРИЗАЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ЧЕРЕЗ ГЛОБАЛЬНУ МЕРЕЖУ

2.1 Метод авторизації через Internet

В інформаційних системах під ідентифікацією розуміють процедуру, у результаті виконання якої для суб'єкта ідентифікації виявляється його ідентифікатор, однозначно ідентифікуючий цього суб'єкта у інформаційній системі [18]. Для виконання процедури ідентифікації в інформаційній системі суб'єктові попередньо повинен бути призначений відповідний ідентифікатор, тобто проведена реєстрація суб'єкта в інформаційній системі.

Процедура ідентифікації прямо пов'язана з аутентифікацією (процедурою перевірки дійсності): суб'єкт проходить процедуру аутентифікації, і якщо аутентифікація успішна, то інформаційна система, на основі факторів аутентифікації визначає ідентифікатор суб'єкта [19]. При цьому вірогідність ідентифікації повністю визначається рівнем вірогідності виконаної процедури аутентифікації (таб. 2.1.1).

Таблиця 2.1.1 - Приклади ідентифікації [18]

Варіант ідентифікації	Фактори аутентифікації	Результат ідентифікації (ідентифікатор)
Ідентифікація користувача	1) Логін/пароль («я знаю»)	Логін
Ідентифікація по банківській карті	1) мікропроцесорна банківська карта («я маю»), 2) ПІН-Код («я знаю»)	Обліковий номер карти (PAN) - зчитується з банківської карти
Ідентифікація по банківській карті з біоверифікацією	1) мікропроцесорна банківська карта («я маю»), 2) біометричний фактор (відбиток пальця) («я є»)	Обліковий номер карти (PAN) - зчитується з банківської карти
Ідентифікація товару	1) штрих-код («я маю»)	Обліковий номер

по штрих-коду		товару
Ідентифікація файла по контрольной сумме	1) контрольная сумма («я є»)	Ім'я файлу
Ідентифікація громадянина за електроним підписом	1)носій електронного підпису («я маю»), 2) пароль доступу до носія («я знаю»)	Ідентифікатор сертифіката (СНИЛС— для сертифіката ключа перевірки кваліфікованому електронному підпису)

Найбільш перспективним для захисту недорогих безкоштовних й умовно-безкоштовних програм є метод авторизації через Інтернет [9, 21]. В інформаційних системах під авторизацією (англ. authorization — дозвіл, уповноважування) розуміють надання певній особі або групі осіб прав на виконання певних дій, а також процес перевірки (підтвердження) даних прав при спробі виконання цих дій [19, 20].

Метод авторизації через Інтернет припускає первісну активацію продукту на обчислювальній машині користувача, а також авторизацію користувача на сервері розроблювачів при кожному запуску програми. Даний метод у цей час має дуже мале поширення й готових рішень у відкритому доступі не існує [21]. Авторизація користувача при кожному запуску програми дозволяє розроблювачеві стежити за статистикою використання програми, виявляти випадки порушення ліцензій, позбавляти ліцензій несумлінних користувачів, а також гнучко змінювати ліцензійну політику у відповідності зі своїми потребами.

У якості обмежень на використання безплатних-умовно-безкоштовних програм може служити введення обмеження на легальне використання програмного продукту, наприклад обмеження на кількість запусків програми на місяць, або обмеження на кількість переглянутих і створених документів за деякий проміжок часу або недопущення одночасного запуску програми на декількох комп'ютерах [21]. Такі

обмеження повинні бути прописані в ліцензійній угоді разом із санкціями за їхні порушення. Під санкцією може матися на увазі як повне, так і тимчасове позбавлення ліцензії. Перевірка порушення лімітів повинна проводитися в недоступному для користувача модулі контролю ліцензій, наприклад, віддаленому сервері, куди програма повинна посилати дані й перевіряти наявність дозволу на запуск. Якщо раптом нелегальна копія програми виявиться опублікованої в публічнім місці, то ліміти досить швидко будуть перевищені й ліцензія буде заблокована. Для того щоб організувати такий захист, необхідно організувати безпечний обмін по відкритому каналу зв'язку між програмою й модулем, у якім несумлінний користувач може читати й змінювати будь-які дані, що пересилаються. У зв'язку з повсюдним поширенням Інтернету такий обмін можна організувати досить просто, не викликаючи значних незручностей у користувачів. Крім того, програма може контролювати спробу зміни коду й посилати повідомлення серверу.

Для реалізації такої схеми розробимо безпечний протокол передачі даних між динамічною бібліотекою, що підключається до програми й віддаленим сервером.

2.2 Протокол авторизації програмного забезпечення

У даній роботі запропонований протокол контролю авторизації програмного забезпечення, який заснований на використанні прив'язки до апаратного забезпечення обчислювальної машини й шифрування 128-бітним алгоритмом хешуванням MD5 відкритого ключа [10, 22]. Суть його полягає в наступному.

Розглянемо додаток Report.exe, установлений на комп'ютері користувача User, для якого необхідно організувати контроль авторизації, і

віддалений сервер Server, що належить розроблювачам додатка або їх довіреній особі.

Розроблювач програмного забезпечення генерує відкритий ключ key_client, який у результаті буде зашифрований 128-бітним алгоритмом хешуванням MD5. Закритий ключ key_server повинен зберігатися на сервері Server. При першому запуску програми користувач одержує ідентифікатор (логін) Identification з боку сервера Server, у випадку якщо адміністратор дав загальний дозвіл на роботу в системі й додавання нових користувачів. Щораз, коли користувач User намагається виконати одну з дій установлених розроблювачем, наприклад запуск програми, створення, відкриття або збереження документа, програма Report.exe, з підключеною динамічною бібліотекою control.dll, повинна надсилати запит серверу Server про можливість продовжити роботу, зробивши наступні кроки передачі даних (рис. 2.2.1):

1. Програма Report.exe, за допомоги динамічної бібліотеки control.dll, перевіряє підключення до інтернет і у випадку його відсутності приховує робочий функціонал програми й видає відповідне повідомлення.
2. У програмі Report.exe формується параметр Comp – ім'я локального комп'ютера й параметр IP – IP адреса локального комп'ютера.
3. У програмі Report.exe обчислюється параметр SerNum – прив'язка до апаратного забезпечення обчислювальної машини, де вона встановлена (4-х байтний серійний номер системного диска C).
4. Програма Report.exe, за допомоги динамічної бібліотеки control.dll, передає пакет з параметрами key_client, Mode, IP, Comp, SerNum серверу Server.
5. Сервер Server перевіряє можливість використання програми користувача з ідентифікатором Identification, який визначається іменем локального комп'ютера - Comp, IP адресою локального комп'ютера – IP, серійним

номером системного диска SerNum, і порівнює зашифрований ключ key_client із ключ key_server.

6. Сервер Server відправляє пакет з параметром f_status програмному забезпеченню Report.exe.

7. Програма Report.exe, за допомоги динамічної бібліотеки control.dll, одержує пакет від серверу Server. Якщо в параметрі f_status підтверджена можливість запуску, то бібліотека control.dll активує головну форму програми Report.exe й відображає робочу панель, а якщо ні, то видає повідомлення й завершує роботу програми Report.exe.

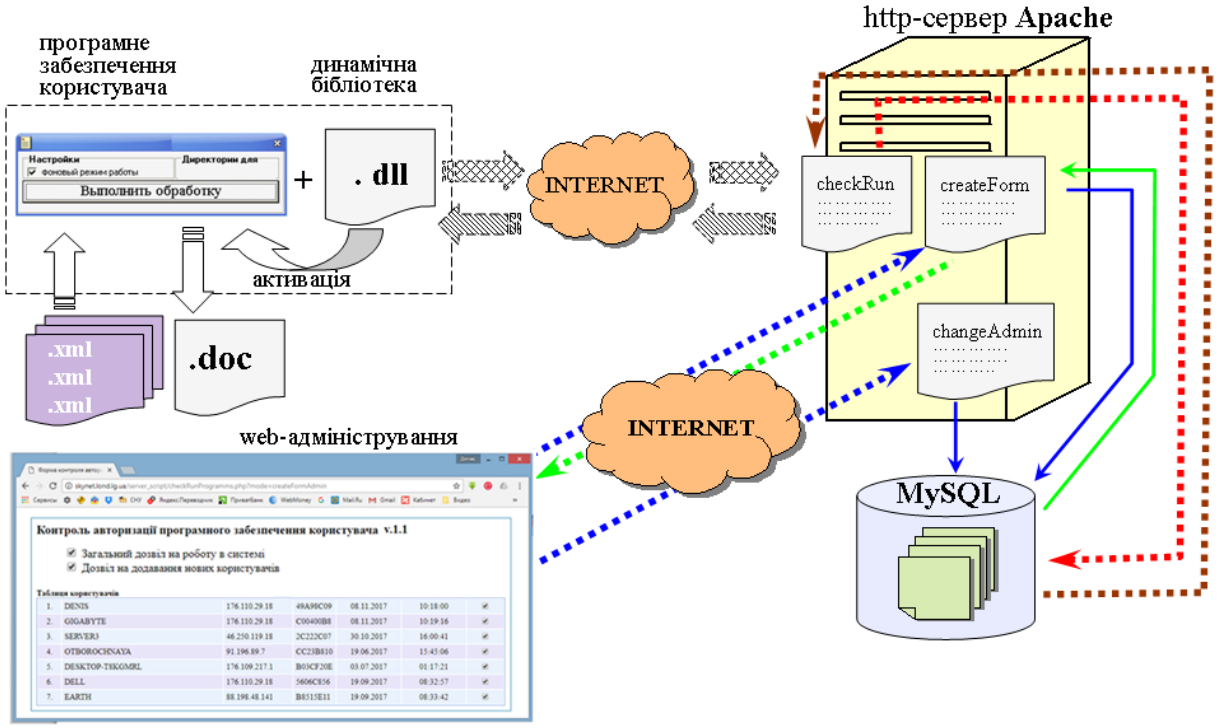

Рисунок 2.2.1 - Схема обміну пакетами між програмним забезпеченням, для якого організується контроль авторизації й віддаленим сервером

2.3 Аналіз розробленого протоколу

Зв'язок із сервером може здійснюватися через публічну глобальну мережу Інтернет по його доменному імені або IP-адресі.

Дозвіл на запуск програми дається віддаленим сервером і може грунтуватися на наступних даних: кількість зроблених успішних запусків

програми, час використання програми, оплачуваний користувачем баланс, кількість створених документів при використанні програми й ін.

Перед першим використанням програми користувач повинен одержати ідентифікатор програми, що запускається, у випадку, якщо адміністратор дав загальний дозвіл на роботу в системі й додавання нових користувачів. Через web-форму адміністрування авторизації ПО адміністратор системи контролю може також поставити ознаку заборони активації ПО в окремих користувачів (рис. 2.3.1).

Рисунок 2.3.1 – Web-форма адміністрування авторизації ПО

Для унікальності реєстраційних даних для даного ідентифікатора в базі даних MySQL зберігаються ім'я локального комп'ютера, IP-адреса локального комп'ютера й 4-х байтний серійний номер системного диска.

Проаналізуємо розроблений протокол від наступних видів атак [21]:

1) атака добору пароля. Для зменшення ймовірності добору пароля користувача при реалізації протоколу необхідно ввести лічильник неуспішних входів у систему й блокувати на якийсь час можливість входу користувача після деякого числа невдалих входів;

2) атака на електронний цифровий підпис. Для підтвердження справжності віддаленого сервера використовується електронний цифровий підпис. Спроба підміни віддаленого сервера заснована на можливості злому алгоритму електронному цифровому підпису й виходить за межі розробки

даного алгоритму. Можна лише рекомендувати використовувати при реалізації протоколу будь-яку сучасну схему електронному цифровому підпису [10];

3) колізія. Для неможливості використовувати раніше отримані від сервера відповіді використовуються унікальні числа, сгенеровані з часу та дати. Можна рекомендувати після генерації такого числа виконувати деяку затримку;

4) атака підміни відкритого ключа. Для неможливості підміни відкритого ключа віддаленого сервера в коді програми використовуються засоби заплутування програмного коду. Суть таких перетворень – усунути можливість простого інжинірингу програми [11 – 13], тобто спроби проаналізувати й змінити вихідний код.

2.4 Програмна реалізація об'єктної моделі

У процесі реалізації розробленого протоколу виявилося, що практично для будь-яких мов програмування є готові бібліотеки, що полегшують написання коду.

В інтегрованім середовищі розробки Embarcadero RAD Studio XE8 розроблена динамічна бібліотека, що підключається до користувацького програмного забезпечення й контролює його авторизацію. Об'єктну модель класу TAdmin, що реалізує протокол авторизації програмного забезпечення, наведено на рис. 2.4.1. У якості механізму шифрування обраний 128-бітний алгоритм хешированія MD5 [10, 22]. Інтерфейсна частина із класом TAdmin наведена в лістингу 2.4.1.

Лістинг 2.4.1 - Інтерфейсна частина заголовного файлу із класом TAdmin:

```
INTERFACE  type
 TAnswerServer = record
  f_status  : byte;
  f_message : string;
 end;
 TAdmin = CLASS
   // метод пінгування (перевірка наявності) ІНТЕРНЕТУ
      function IsInternetConnected : Boolean; virtual; abstract;
   // метод одержання імені комп'ютера
      function getComputerNetName : string; virtual; abstract;
   // метод запиту до сервера
      function getQuery(const URL : String):TAnswerServer; virtual; abstract;
  // властивість - хост для пінга інтернет
       function  getHost : string;              virtual; abstract;
       procedure setHost(host : string);    virtual; abstract;
       property  Host : string  read getHost write setHost;
   // властивість - серійний номер диска
       function  getSerNum:string;              virtual; abstract;
       procedure setSerNum(disk : string);  virtual; abstract;
       property  SerNum : string read getSerNum write setSerNum;
   // властивість - відповідний пакет сервера
      function  getAnsServer : TAnswerServer;              virtual; abstract;
      procedure setAnsServer (par : TAnswerServer);       virtual; abstract;
      property  ansserver : TAnswerServer read getAnsServer write setAnsServer;
 END;
```

Рисунок 2.4.1 - Клас TAdmin

РОЗДІЛ 3 РЕАЛІЗАЦІЯ СИСТЕМИ КОНТРОЛЮ АВТОРИЗАЦІЄЮ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Підключення динамічної бібліотеки авторизації до розроблювального програмного забезпечення

Розглянемо підключення динамічної бібліотеки авторизації до програми розроблювача Report.exe. Сама програма Report.exe установлюється на комп'ютері користувача User і для неї організується контроль авторизації. Програма обробляє xml-файли, структура яких наведена на рис. 3.1.1. Файли перебувають у відповідній директорію XML. Результатом роботи програми є створення файлу-звіту (рис. 3.1.2) у форматі doc (рис. 3.1.5).

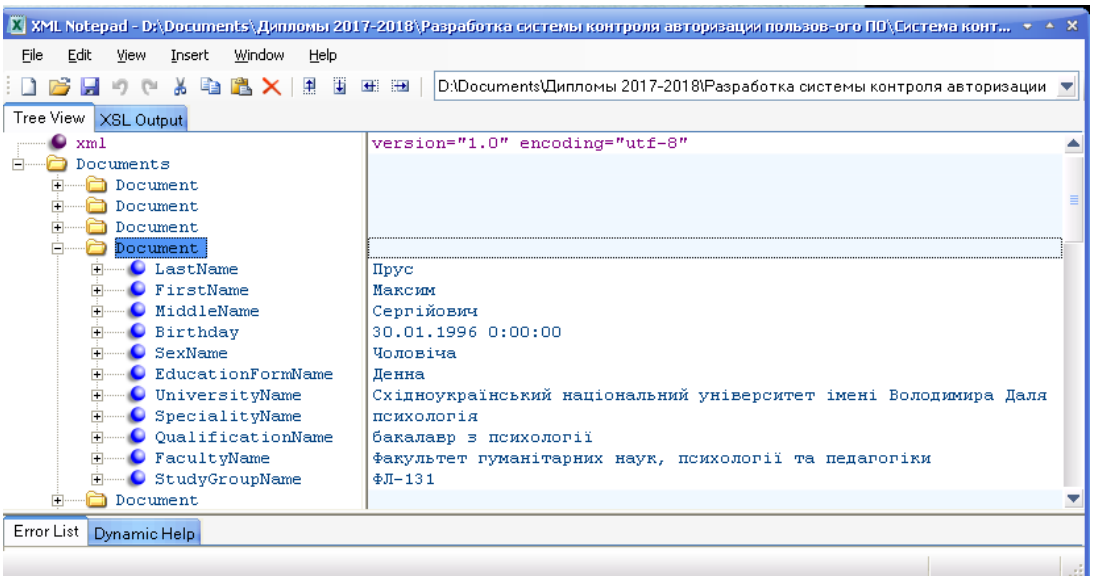

Рисунок 3.1.1 - Формат XML-файлу – джерела даних для програми Report.exe

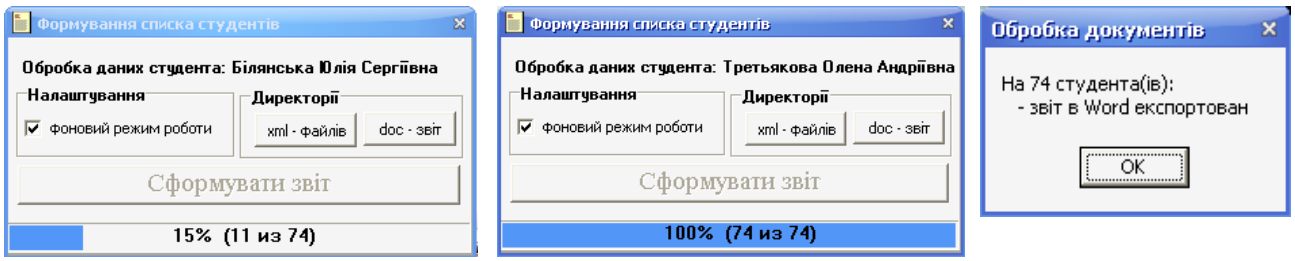

Рисунок 3.1.2 - Процес роботи програми Report.exe

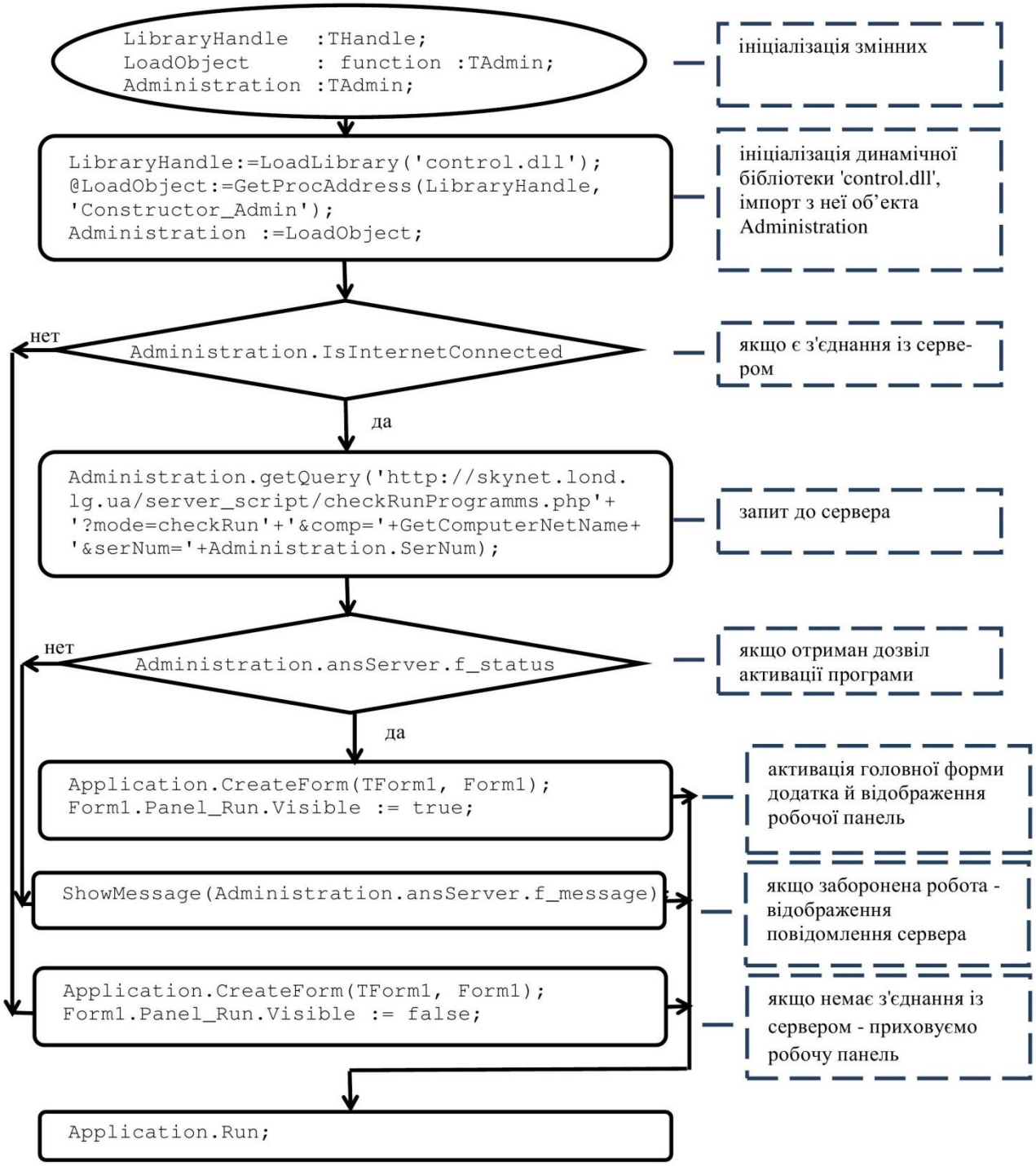

Рисунок 3.1.3 - Блок-схема алгоритму підключення й використання класу TAdmin на клієнтській стороні

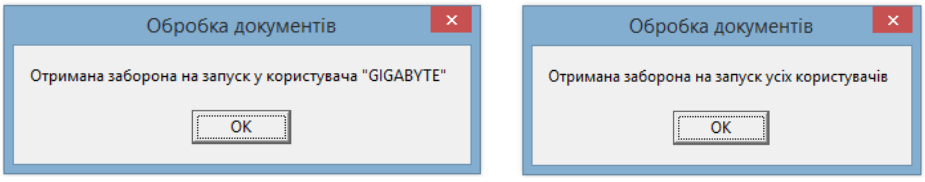

Рисунок 3.1.4 - Реакція бібліотеки на заборонний пакет активації програми Report.exe

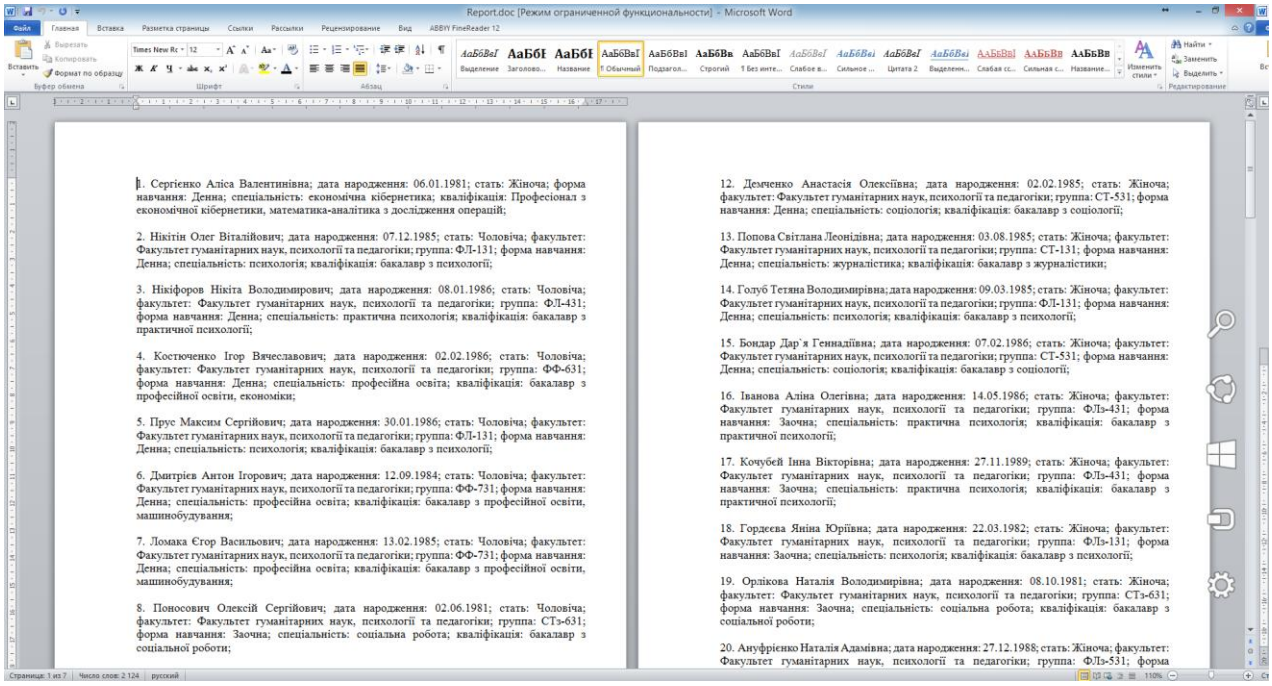

Рисунок 3.1.5 - Результат роботи програми Report.exe – створення файлу-звіту у форматі doc

Згідно з розробленим протоколом авторизації програмного забезпечення, розроблювач у своєй програмі повинен провести ініціалізацію динамічної бібліотеки «control.dll» і імпортувати з неї об'єкт Administration, що належить класу TAdmin. Інтерфейс і реалізація класу TAdmin описані в додатку Б. Далі, користуючись методами наведеними в інтерфейсній частині Administration, програма розроблювача повинна провести перевірку з'єднання із сервером, зробити запит до сервера й, одержавши відповідь, активувати або деактивувати роботу всієї програми. Блок-схема алгоритму підключення й використання класу TAdmin на клієнтській стороні наведено на рис 3.1.3. У випадку одержання пакета деактивації динамічна бібліотека control.dll видасть повідомлення сервера й закриє програму Report.exe (рис. 3.1.4).

3.2 Серверні скрипти, взаємодіючі з динамічною бібліотекою

Серверна частина програмного комплексу реалізована мовою програмування PHP. Фрагмент коду програми наведено в лістингу 3.2.1. Реалізація взаємодіючого класу php із класом TAdmin з динамічної бібліотеки «control.dll» описана в додатку В. На рис. 3.2.1 показана база

даних bd_test на сервері MySQL, з таблицями для збереження ідентифікатора, IP комп'ютера, на якому інстальована програма.

Лістинг 3.2.1 – PHP-фрагмент програмної реалізації серверної частини

```
//------------------- РЕЖИМ [checkRun] (проверка работы)----------------------------------
  if ($_GET['mode']=='checkRun'){ //режима [checkRun]
    $comp_name = $_GET['comp'];
    $serNum    = $_GET['serNum'];
    $connect = mysql_connect("localhost", "user_test", "test8311022802") or DIE("Не могу создать соединение");
    mysql_select_db("bd_test", $connect) or DIE("Не могу выбрать базу данных");

    $query = mysql_query("SELECT * FROM tab_ProgramSetting", $connect);
    $myrow = mysql_fetch_array($query);
    $allRun = $myrow['col_All_Run'];
    $allAdd = $myrow['col_Add'];
    if($allRun==1){    // есть разрешение на работу
        $query = mysql_query("SELECT * FROM tab_ProgramUser WHERE col_nameComp='".$comp_name."' AND col_serNum='".$serNum."'"
        $count_rec = mysql_num_rows($query);
        $user = mysql_fetch_array($query);
        if ($count_rec==0){ // не зарегистрирован в базе
            if ($allAdd==1){// добавляем
                $text_query = sprintf("INSERT INTO tab_ProgramUser (col_nameComp, col_IP, col_serNum, col_Date, col_Time, co
                                       VALUES ('%s', '%s', '%s', FROM_UNIXTIME(%d), FROM_UNIXTIME(%d), %d)",
                                       $comp_name, $_SERVER['REMOTE_ADDR'], $serNum, strtotime(date('Y-n-d')), strtotime(date
                $query = mysql_query($text_query, $connect);
                echo '1:Разрешено';
            }
            else { // стоит запрет на добавление новых
            echo '0:Отримана заборона на підключення нового користувача';
            }
        }
        else { // уже зарегистрирован в базе
        // вносим изменения по пользователю
        $text_query = sprintf("UPDATE tab_ProgramUser SET col_IP='%s',  col_date=FROM_UNIXTIME(%d), col_time=FROM_UNIXTIME(%
                              WHERE col_id=%d",
                              $_SERVER['REMOTE_ADDR'],
                              strtotime(date('Y-n-d')), strtotime(date('H:i:s')), $user['col_id']);
        $query = mysql_query($text_query, $connect);
        // проверяем разрешение для данного пользователя
        if ($user['col_Run'] == 1){ // разрешено для данного пользователя
            echo '1:Дозволено';
        }
```

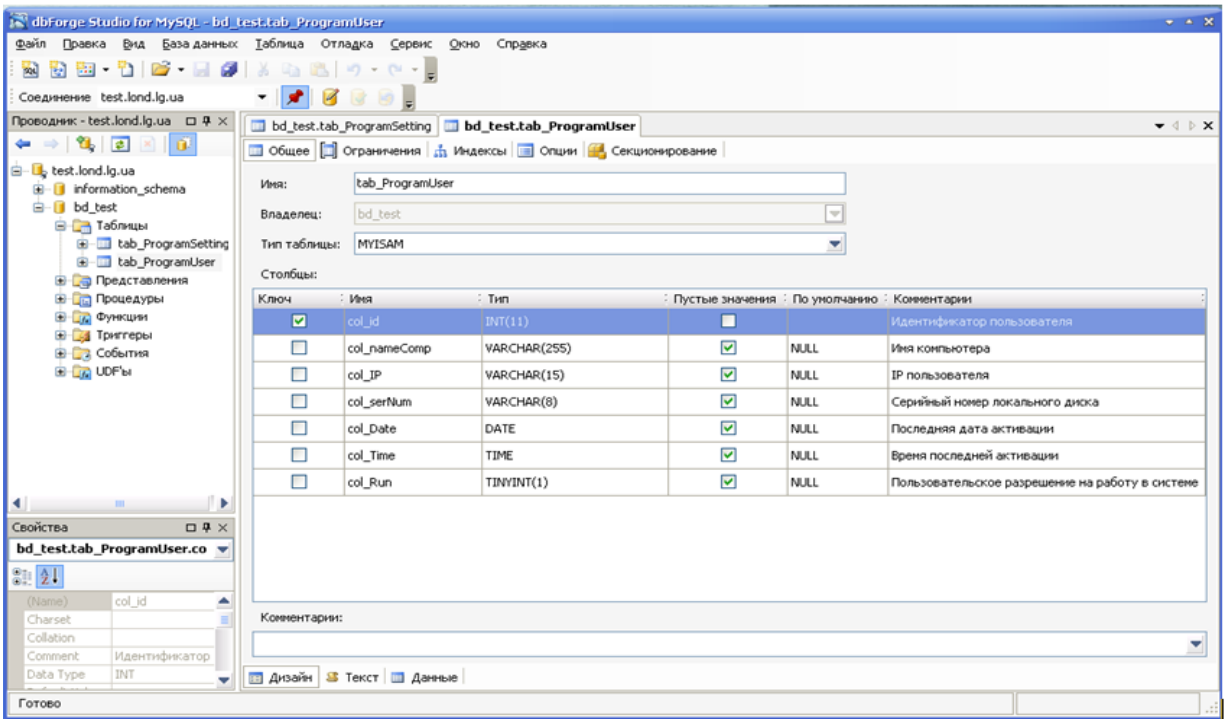

Рисунок 3.2.1 - База даних bd_test на сервері MySQL, з таблицями для збереження даних комп'ютера й IP користувача

3.3 Порівняльний аналіз методів контролю авторизацією

Приведемо порівняння запропонованого способу захисту з методами StarForce, Аладдін, SAS, CrypLine (табл. 3.3.1).

Позначення в таблиці, відповідно:

K_1 – використання апаратних засобів при використанні програми;

K_2 – активація програми перед першим використанням;

K_3 – відсутність прив'язки до Інтернет при роботі програми або її активації;

K_4 – немає можливості нелегально використовувати програму шляхом переносу її на віртуальну машину;

K_5 – є можливість використовувати умовно безкоштовну версію (Shareware).

Таблиця 3.3.1 Порівняння запропонованого методу з відомими методами

Метод	K_1	K_2	K_3	K_4	K_5
StarForce	-	+	+	+	-
Алладін	-	+	+	+	-
SAS	+	-	-	+	-
CrypLine	+	-	-	-	+
Запропонований у роботі	+	+	-	+	+

З таблиці видно, що необхідність підключення до Інтернету при активації або використанні програми заміняє необхідність використовувати апаратний ключ. Запропонований у роботі метод дозволяє захистити умовно-безкоштовну програму, не втративши при цьому ефективність захисту. Метод дозволяє забезпечити більш надійний захист у порівнянні з існуючими програмними методами, оскільки виключає можливість установки програмного продукту на віртуальну машину. Цей метод дозволяє відмовитися від використання апаратних обладнань при запуску програми. Простота реалізації клієнтської й серверної частини програмного забезпечення дозволяє використовувати його як у

багатомодульних, так і невеликих програмах, не здорожуючи вартість розробки.

Подальше поліпшення протоколу може полягати в удосконаленні методів протоколу для здійснення можливості більш гнучкого керування ліцензійною політикою розроблювачем.

ВИСНОВКИ

1. У ході виконання дипломної роботи була розроблена об'єктна модель ідентифікації й аутентифікації й запропонований протокол авторизації програмного забезпечення.
2. В інтегрованім середовищі розробки Embarcadero RAD Studio XE8 розроблена динамічна бібліотека, що підключається до користувацького програмного забезпечення й контролює його авторизацію.
3. Для препроцесора гіпертексту PHP, що працює на http-сервері Apache розроблені сценарії взаємодії динамічної бібліотеки й бази даних MySQL, написаний сценарій створення форми адміністрування.
4. Отримані результату роботи й засіб технічного захисту для розроблювального програмного забезпечення можуть бути практично використані розроблювачами програмного забезпечення при реалізації його захисту від несанкціонованого використання й контролю авторизації програмних продуктів. Авторизація користувача при кожному запуску програми дозволяє розроблювачеві стежити за статистикою використання його програми, виявляти випадки порушення ліцензій, позбавляти ліцензій несумлінних користувачів, а також гнучко змінювати ліцензійну політику у відповідності зі своїми потребами.

СПИСОК ЛІТЕРАТУРИ

1. Варлатая С.К. Програмно-апаратний захист інформації: учб. посібник / С.К. Варлатая, М.В. Шаханова. – Владивосток: ИздДУ ДВГТУ, 2007. – 318 с.

2. Скляров Д.В. Мистецтво захисту й злому інформації / Д.В. Скляров. – Спб.: Бхв-Петербург, 2004. – 288 с.

3. Віртуальна машина [Електронний ресурс]. – Режим доступу до ресурсу: http://ru.wikipedia.org/wiki/ Віртуальна_машина.

4. Software as a service [Електронний ресурс]. – Режим доступу до ресурсу: http://ru.wikipedia.org/wiki/Software_as_a_service.

5. Захист від піратства й ліцензування ПО Starforce [Електронний ресурс]. – Режим доступу до ресурсу: http://www.star-force.ru/.

6. Захист програм Hasp [Електронний ресурс]. – Режим доступу до ресурсу: http://www.aladdin-rd.ru/.

7. Сервіс Software Activation Service [Електронний ресурс]. – Режим доступу до ресурсу: http://www.softactivation.com/asp/getstarted.asp.

8. Захист Crypline [Електронний ресурс]. – Режим доступу до ресурсу: http://crypline.ru.

9. Аналіз ринку засобів захисту від копіювання й злому програмних засобів [Електроний ресурс]. – Режим доступу до ресурсу: http://citforum.ru/security/articles/analis/.

10. Молдовян Н.А. Введення в криптосистеми з відкритим ключем / Н.А. Молдовян, А.А. Молдовян. – Спб.: Бхв-Петербург, 2005. – 288 с.

11. Буінцев Д.Н. Метод захисту програмних засобів на основі перетворень, що заплутують: дис. .канд. техн. наук : 05.13.19 / Д.Н. Буинцев. – Томськ, 2006. – 121 с.

12. Зворотна розробка [Електронний ресурс]. – Режим доступу до ресурусу: http://ru.wikipedia.org/wiki/Зворотна_розробка.

13. Система програмного захисту додатків від несанкціонованого копіювання Asprotect [Електронний ресурс]. – Режим доступу до ресурусу: http://www.aspack.com/asprotect.aspx.

14. Дшхунян В.Л. Електронна ідентифікація. Безконтактні електронні ідентифікатори й смарткарти / В.Л. Дшхунян, В.Ф. Шаньгин. – ИздДВ «НТ Прес», 2004. – 695 с.

15. GMP Install Instruction for Windows Platform [Електронний ресурс]. – Режим доступу до ресурусу: http://cs.nyu.edu/exact/core/gmp/.

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

17. Студопедія. [Електронний ресурс] . – Режим доступу до ресурсу: https://studopedia.su/15_3511_zashchita-po.html.

18. Wikipedia. [Електронний ресурс]. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Идентификация.

19. Wikipedia. [Електронний ресурс]. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Авторизация.

20. Авторизация пользователя. [Електронний ресурс]. – Режим доступу до ресурсу: https://dic.academic.ru/dic.nsf/business/17457.

21. Метод авторизации через интернет для защиты shareware программ [Електронний ресурс]. – Режим доступу до ресурсу: http://docplayer.ru/51866929-Metod-avtorizacii-cherez-internet-dlya-zashchity-shareware-programm.html.

22. MD5. [Електронний ресурс]. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/MD5.

Додаток А

Сучасні засоби захисту програмного забезпечення

Таблиця А.1 - Порівняльна характеристика захисту Cdr-cops.

Тип захисту:	Вимір фізичних характеристик без нанесення особливих міток на носій
Спосіб подолання захисту:	"Кряк"
Апаратна сумісність (cd/ dvdвприводи різних виробників):	Середня (сумісна тільки з популярними приводами)
Наявність особливої апаратури для захисту серії:	НЕ потрібно
Надання SDK для виробників:	ТАК
Захист дрібних партій (CD/R/RW):	НЕМАЄ
Фірма - виробник:	Link Data Security (http://www.linkdata.com/)
Комерційні продукти, що використовують даний вид захисту:	Interactive English / De Agostini National encyklopedia Agostini Atlas 99 Agostini Basetera BMM DK Kort Lademanns'99
Особливості захисту:	Дана програма користується найефективнішою системою захистом від копіювання, заснованої не на нанесенні фізичних міток, а на способі виміру специфічних характеристик CD\DVD-ROM. За словами виробника, система аналізує фізичний кут між минулим і поточними логічними блоками на компакт-диску. Слабке місце захисту - це сам код, що аналізує кути. Фірмі поки не вдалося знайти ефективного способу протидії дизассемблерам і відлачикам.

Таблиця А.2 - Порівняльна характеристика захисту Star Force.

Тип захисту:	Вимір фізичних характеристик без нанесення особливих міток на носій
Спосіб подолання захисту:	Спосіб не знайдений (теоретично злом можливий шляхом покрокового налагодження нестандартним відладчиком у середовищі емуляції операційної системи)
Апаратна сумісність (cd/ dvdвприводи різних виробників):	100% (за рахунок можливості введення спеціального ключа)
Наявність особливої апаратури для захисту серії:	НЕ потрібно
Надання SDK для виробників:	ТАК
Захист дрібних партій	ТАК

(CD/R/RW):	
Фірма - виробник:	Protection Technology (http://www.star-force.com/)
Комерційні продукти, що використовують даний вид захисту:	1С (гри), Нивал, Softmax Co, Q-puncture Inc, Scholastic, Hypnosys World, руссобитум (гри) Російські клієнти компанії: 1С, Диасофт, Руссобит, Лукойл, Бука, Акела
Особливості захисту:	Дана програма так само, як CD-COPS і TAGES, користується найефективнішою системою захистом від копіювання, заснованої на способі виміру специфічних характеристик CD\DVD-ROM. Оскільки захист не розкритий, те не відомий і спосіб, яким виробникам вдається ідентифікувати різні диски. Заявлена 100% сумісність із будь-якою апаратурою й 100% ідентифікація дисків підтверджується незалежними сайтами. Захист ефективно протидіє відладчикам і дизассемблерам. Ефективність захисту підтверджує той факт, що за рік існування не найшовся спосіб її нейтралізації. Додатковий лиск захисту надає можливість захисту дрібних партій дисків (CD-R\RW) і наявність SDK, за допомогою якого розроблювачі ігор можуть шифрувати окремі фрагменти коду й даних. Окремо можна скористатися функцією шифрування ігрових даних від несанкціонованого "перекладу" або "адаптації".

Таблиця А.3 - Порівняльна характеристика захисту Laserlock.

Тип захисту:	Фізичне нанесення мітки на носій
Спосіб подолання захисту:	Копіювання (Blindread), "кряк", емуляція (D-Tools)
Апаратна сумісність (cd/ dvdвприводи різних виробників):	Низька
Наявність особливої апаратури для захисту серії:	ТАК, потрібно
Надання SDK для виробників:	ТАК
Захист дрібних партій (CD/R/RW):	НЕМАЄ
Фірма - виробник:	MLS Laserlock International (http://www.laserlock.com/)
Комерційні продукти, що використовують даний вид захисту:	Asghard, Fallout 2, Icewind Dale, Jagged Alliance 2, Messiah, Metro Police, Outcast, Shogo, Specops
Особливості захисту:	Використовує унікальне маркування. Кожний додаток на CD має унікальне блокування параметра, який відповідає кінцевий захист від копіювання. Багато слів, а мало справи. Продукт уже розкритий усіма можливими методами. Розроблювачі системи явно не встигають вносити зміни в код захисту для протидії вивертам піратів.

Таблиця А.4 - Порівняльна характеристика захисту Safedisk.

Тип захисту:	Фізичне нанесення мітки на носій
Спосіб подолання захисту:	Копіювання (Clonecd), "кряк", емуляція (D-Tools)
Апаратна сумісність (cd/ dvdвприводи різних виробників):	Гарна
Наявність особливої апаратури для захисту серії:	ТАК, потрібно
Надання SDK для виробників:	ТАК
Захист дрібних партій (CD/R/RW):	НЕМАЄ
Фірма - виробник:	Macrovision Corporation (http://www.macrovision.com/)
Комерційні продукти, що використовують даний вид захисту:	Практично всі ігри після 01-01-2001
Особливості захисту:	Тут також застосовується метод нанесення фізичних міток, для чого потрібне додаткове встаткування. Виробник затверджує про велику ефективність системи, не розкриваючи методів, які застосовуються в захисті. Тільки піратов це не зупинило. Усі програми, захищені даної системою вже розкриті. Способів протидії захисту також знайдене трохи. Незважаючи на надання SDK компаніямзвиробникам ігор не вдалося забезпечити надійної протидії

Таблиця А.5 - Порівняльна характеристика захисту Securom.

Тип захисту:	Фізичне нанесення мітки на носій
Спосіб подолання захисту:	"Кряк", емуляція (D-Tools)
Апаратна сумісність (cd/ dvdвприводи різних виробників):	Низька
Наявність особливої апаратури для захисту серії:	ТАК, потрібно
Надання SDK для виробників:	НЕМАЄ
Захист дрібних партій (CD/R/RW):	НЕМАЄ
Фірма - виробник:	Sony (http://www.securom.com/)
Комерційні продукти, що використовують даний вид захисту:	Diablo 2, Simcity 3000, Decent Freespace, FIFA 99, Panzer Commander, S.A.G.A: Rage of the Vikings
Особливості захисту:	Використовуються всі ті ж мітки. Досить цікавою відмінністю даного захисту є те, що вони дійсно не копіруються. Принаймні дотепер не знайдене програм, здатних побитово копіювати захищені диски. В іншому ж так само як і з багатьма системами - є методи обходу захисту, і це незважаючи на славетне ім'я корпорації, що робить захист. Справедливості заради варто відзначити, що над удосконалюванням захисту ведуться постійні роботи. Хто знає, може їм і вдасться стати абсолютно непроникними. мітка, яка не копірується уже є.

Таблиця А.6 - Порівняльна характеристика захисту TAGES

Тип захисту:	Вимір фізичних характеристик без нанесення особливих міток на носій
Спосіб подолання захисту:	Емуляція, "Кряк"
Апаратна сумісність (cd/ dvdвприводи різних виробників):	Висока
Наявність особливої апаратури для захисту серії:	НЕ потрібно
Надання SDK для виробників:	НЕМАЄ
Захист дрібних партій (CD/R/RW):	НЕМАЄ
Фірма - виробник:	Thomson & MPO (http://www.licensing.thomson-csf.com/buy/cdcopy.html)
Комерційні продукти, що використовують даний захист:	Moto Racer 3
Особливості захисту:	Захист заснований на досить оригінальному методі багаторазового читання того самого сектору з наступним порівнянням результатів. Цілком можливо, що тут відбувається аналіз фізичних характеристик носія. Слабке місце захисту її програмний модуль, який уже розкритий

Таблиця А.7 - Порівняльна характеристика захисту Aladdin

Тип захисту:	Захист від несанкціонованого використання. Метод реалізації - електронний ключ
Спосіб подолання захисту:	Нова версія ще не розкрита (теоретичний спосіб злому - емуляція HASP і аналіз коду, що виконується)
Апаратна сумісність (cd/ dvdвприводи різних виробників):	-
Наявність особливої апаратури для захисту серії:	Потрібен електронний ключ
Надання SDK для виробників:	ТАК
Захист дрібних партій (CD/R/RW):	ТАК
Фірма - виробник:	"Аладдин" (http://www.aladdin.ru)
Комерційні продукти, що використовують даний захист:	1С, АСКОН
Особливості захисту:	Захист базується на використанні зовнішніх електронних ключів, що підключаються до портів комп'ютерів. Заявлені різні моделі HASP для портів USB,, COM. Плати ISA, PCI. Використовуються потужні унікальні ключі. Захист забезпечується SDK. Слабке місце захисту - звертання до портів, які можна емулювати. З особливостей SDK можна відзначити легкість освоєння. При використанні можливостей HASP4 можливо створити захист високого рівня. Єдина із представлених захистів має в наявності мережні ліцензії.

Додаток Б

Інтерфейс і код реалізації динамічної бібліотеки control.dll, яка підключається до користувацького програмного забезпечення й контролює його авторизацію.

```
LIBRARY Control;
uses   SysUtils, Classes, WinSock2, WinInet, Windows;
{$R *.res}
const
     HTTP_PORT = 80;
     CLRF = #13#10;
     Header = 'Content-Type: application/ www-form-urlencoded' + CLRF;
type    TAnswerServer = record    f_status : byte;    f_message : string;  end;

  TLAdmin = CLASS
    private
          FHost : string;     FSerNum : string;     FAnsServer : TAnswerServer;
    public
     constructor Create;
     function isInternetConnected: Boolean; virtual;
     function getComputerNetName: string; virtual;
     function getQuery(const URL: String):TAnswerServer; virtual;
     function getHost : string; virtual;      procedure setHost(host : string); virtual;
     property Host : string read getHost write setHost;
     function readSerNum (disk : char) : string ;      function getSerNum : string; virtual;
     procedure setSerNum (disk : string); virtual;    //  властивість - серійний номер диска
     property Sernum : string read getsernum write setsernum;
     function    getAnsServer : TAnswerServer; virtual;
     procedure setAnsServer (par : TAnswerServer); virtual;
     property   ansserver : TAnswerServer  read getAnsServer write setAnsServer;
  end;
  constructor TLAdmin.Create;              begin    Host :='www.microsoft.com';    SerNum :='C';  end;
  function TLAdmin.getHost:string;                  begin result := FHost; end;
  procedure TLAdmin.setHost (Host : string);    begin Fhost:= Host; end;
  function TLAdmin.getSerNum:string;             begin result := FSerNum; end;
```

```
 procedure TLAdmin.setSerNum(disk : string);      begin FSerNum:=readSerNum(disk[1]);  end;
 function    TLAdmin.getAnsServer : TAnswerServer;          begin result := FAnsServer; end;
 procedure TLAdmin.setAnsServer(par : TAnswerServer);   begin  FAnsServer:=par;  end;

//__ перевірка наявності ІНТЕРНЕТУ__
function TLAdmin.IsInternetConnected: Boolean;
 var WSDATA: TWSADATA;
 begin  try
   Result := False;
   if  WSAstartUp(MAKEWORD(2, 0), WSDATA) <> 0 then Exit;
   Result := GetHostByName(Pchar(FHost)) <> nil;
 finally WSAcleanUp;  end;
 end;

//__одержання імені комп'ютера___
function TLAdmin.GetComputerNetName: string;
 var  buffer: array[0..255] of char;    size: dword;
 begin
    size := 256;
    if GetComputerName(buffer, size) then  Result := buffer    else Result := ''
 end;

 //__читання серійного номера диска___
function TLAdmin.readSerNum(disk : char) : string;
 var
    Volumename, FileSystemName : array [0..MAX_PATH-1] of Char;
    MaxComponentLength, FileSystemFlags : Cardinal;     VolumeSerialNo : DWord;
 begin
   GetVolumeInformation(Pchar(disk+':/'),    Volumename,      MAX_PATH,
                              @Volumeserialno,  MaxComponentLength,  FileSystemFlags,
                               FileSystemName, MAX_PATH);
   Result:=IntToHex(VolumeSerialNo, 0);
  end;
```

```
//__запит до сервера__
function TLAdmin.getQuery(const URL: String):TAnswerServer;
 var
  FSession, FConnect, FRequest: HINTERNET;
  FHost, FScript, buffer: String;    Ansi: PAnsiChar;
  Buff: array [0..1024] of Char;    BytesRead : Cardinal;

 function Host(URL: String): String;
  begin
    if Pos('http://', URL) > 0 then Delete(Url, 1, 7);
    Result := Copy(Url, 1, Pos('/', Url) - 1);
    if Result = '' then Result := URL;
  end;

begin
  // Парсинг: витягаємо ім'я хоста й параметри звертання до скрипту
  FHost := Host(Url);    FScript := Url;
  Delete(Fscript, 1, Pos(Fhost, Fscript) + Length(Fhost));
  //  Ініциалізуємо WinInet
  if not Assigned(FSession) then Exit;
  try   // Спроба з'єднання із сервером
   FConnect := InternetConnect(FSession, Pchar(FHost), HTTP_PORT, nil,'HTTP/1.0',
                                    INTERNET_SERVICE_HTTP, 0, 0);
   if not Assigned(FConnect) then Exit;
   try      // Підготовляємо запит сторінки
    Ansi := 'text/*';
    FRequest := HttpOpenRequest(FConnect, 'GET', Pchar(Fscript), 'HTTP/1.0', '',
                                      @Ansi, INTERNET_FLAG_RELOAD, 0);
    try    // Додаємо заголовки
     {if not (Httpaddrequestheaders(Frequest, Header, Length(Header),
                                        HTTP_ADDREQ_FLAG_REPLACE or
                                        HTTP_ADDREQ_FLAG_ADD)) then Exit;}
     // Відправляємо запит
     if not (Httpsendrequest(Frequest, nil, 0, nil, 0)) then Exit;
```

```
        // Одержуємо відповідь
        repeat  //збираємо всі пакети в 1 буфер
          FillChar(Buff, SizeOf(Buff), 0);
          InternetReadFile(Frequest, @Buff, Sizeof(Buff), Bytesread);
          buffer := buffer + Buff;
        until ((Bytesread = 0) or (length(buffer)>25000));

        // обробка відповіді
        Result.f_status      :=  StrToInt(Copy(buffer, 1, 1));
        Result.f_message  :=  Copy(buffer, 3, length(buffer)-2);
        FAnsServer := Result;
      finally InternetCloseHandle(FRequest);   end;
    finally   InternetCloseHandle(FConnect);  end;
  finally    InternetCloseHandle(FSession);    end;
end;

function CreateAdmin:TLAdmin; stdcall;
   begin  Result:=TLAdmin.Create; end;

 / / експорт об'єкта
 EXPORTS CreateAdmin NAME 'Constructor_Admin';
begin end.
```

Додаток В

Сценарій взаємодії динамічної бібліотеки й бази даних MYSQL

для препроцесора гіпертексту PHP, що працює на http-сервері

```
<?php
//------ РЕЖИМ [checkrun] {перевірка роботи}----------------------------------
 if ($_GET['mode']=='checkrun'){ //режиму [checkrun]
        $comp_name = $_GET['comp'];
        $sernum    = $_GET['sernum'];
        $connect = mysql_connect("localhost", "user_test", "test8311022802") or
                                        DIE("Не можу створити з'єднання");
        mysql_select_db("bd_test", $connect) or DIE("Не можу вибрати базу даних");
        $query = mysql_query("SELECT * FROM tab_ProgramSetting", $connect);
        $myrow = mysql_fetch_array($query);
        $allrun = $myrow['col_All_Run'];
        $alladd = $myrow['col_Add'];
        if($allrun==1){   // є дозвіл на роботу
        $query = mysql_query("SELECT * FROM tab_Programuser
                                    WHERE col_namecomp='".$comp_name."' AND
                                    col_sernum='".$sernum."'", $connect);
         $count_rec = mysql_num_rows($query);
         $user = mysql_fetch_array($query);
        if ($count_rec==0){ // не зареєстрований у базі
           if ($allAdd==1){// додаємо
            $text_query = sprintf("INSERT INTO tab_Programuser (col_namecomp, col_IP,
                                        col_sernum, col_Date, col_Time, col_Run)
           VALUES ('%s', '%s', '%s', FROM_UNIXTIME(%d), FROM_UNIXTIME(%d), %d)",
            $comp_name, $_SERVER['REMOTE_ADDR'],
            $sernum, strtotime(date('Y-n-d')), strtotime(date('H:i:s')), 1);
            $query = mysql_query($text_query, $connect);
            echo '1:Дозволене';
        }
        else { // стоит заборона на додавання нових
                echo '0:Отримана заборона на підключення нового користувача';
        }
```

```
        else { // уже зареєстрований у базі
          // вносимо зміни по користувачеві
           $text_query = sprintf("UPDATE tab_Programuser SET col_IP='%s',
                    col_date=FROM_UNIXTIME(%d), col_time=FROM_UNIXTIME(%d)
                     WHERE col_id=%d",   $_SERVER['REMOTE_ADDR'],
                     strtotime(date('Y-n-d')), strtotime(date('H:i:s')), $user['col_id']);
           $query = mysql_query($text_query, $connect);
           // перевіряємо дозвіл для даного користувача
           if ($user['col_Run'] == 1){ // дозволене для даного користувача
             echo '1:Дозволене';
           else{ // заборона даного користувача
           echo '0:Отримана заборона на запуск у користувача "'.$comp_name.'"';
        }}
        else echo '0:Отримана заборона на запуск усіх користувачів';
        mysql_close($connect);
}; // закінчення режиму [checkrun]

//--- РЕЖИМ [createformadmin] {формування форми адміністрування}-
         if ($_GET['mode']=='createformadmin'){
           header('Content-type: text/html; charset=windows-1251');
           $res = '';   $i = 0;
           $connect = mysql_connect("localhost", "user_test", "test8311022802") or DIE("Не
можу створити з'єднання");
          mysql_select_db("bd_test", $connect) or DIE("Не можу вибрати базу даних");
          // запит до таблиці глобальних настроювань
        $query = mysql_query("SELECT * FROM tab_ProgramSetting", $connect);
        $setting = mysql_fetch_array($query);
        // запит до таблиці користувачів
        $query = mysql_query("SELECT * FROM tab_ProgramUser", $connect);
        while($users = mysql_fetch_array($query)){
             $res.='<tr col_id="'.$users['col_id'].'" class="tr'.(fmod($i,2)+1).'">
                    <td style="width:5%; text-align:center">'.($i+1).'.</td>
                    <td style="width:35%; text-align:left; padding-left:5px">'.
        $users['col_nameComp']. '</td>
```

```
          <td style="width:15%; text-align:left; padding:5px;">'.$users['col_IP'].'</td>
        <td style="width:9%; text-align:left; padding:5px;">'.$users['col_sernum'].'</td>
        <td style="width:14%; text-align:center">'.Convertion_Date($users['col_Date'],1).'</td>
        <td style="width:14%; text-align:center">'.$users['col_Time'].'</td>
        <td style="width:8%; text-align:center; padding:0px;" title="Відкрити користувачеві
'.$users['col_namecomp'].' доступ">
          <input type="checkbox" '.($users['col_Run']==1 ? ' checked ' : '').'
onchange="onchangesetuse(this)">
        </td></tr>';
        $i++;   }
        mysql_close($connect);
        echo "<!DOCTYPE html PUBLIC '-//W3C//DTD XHTML 1.0 Transitional//EN'
        'http://www.w3.org/TR/xhtml1/DTD/xhtml 1atransitional.dtd'>
        <html xmlns='http://www.w3.org/1999/xhtml' xml:lang='ru' lang='ru'>
        <head>
        <meta http-equiv='Content-Type' content='text/html; charset=windows-1251'></meta>
        <title>Форма контролю авторизації</title>
        <style>
         #content{margin-left:50px; margin-top:20px; padding:10px; border: solid 1px #0076a3;
width:90%}
        .tr1{ text-align:center; background-color : rgba(218,233,251,.6);  }
                .tr2{ text-align:center; background-color : rgba(219,212,245,.6);  }
                .tr1:hover, .tr2:hover{ background-color:rgba(177,177,177,0.6);    }
        .labelAll{width:700px; margin-bottom:0px; padding:0;}
        </style>
        <script type='text/javascript'>
        xmlhttp  = new Xmlhttprequest();
        xmlhttp.sendserver = function (param) {
        this.open('POST', 'checkrunprogramms.php', true);
        this.setRequestHeader('Content-Type', 'application/ wwweformeurlencoded');
        this.send(param);
        };
        function onChangeSetting(){
           xmlhttp.sendserver('mode='+encodeuricomponent('changeadmin')+'&'+
```

```
'pchangeallrun='+encodeuricomponent(document.getelementbyid('CB1').checked?1:0)+ '&' +
'pchangeadd='+encodeuricomponent(document.getelementbyid('CB2').checked?1:0));
};
function
onChangeSetUse(obj){xmlhttp.sendserver('mode='+encodeURIComponent('changeAdmin')+'&'
+
'pcb='+encodeURIComponent(obj.checked?1:0)+'&'+
        'pid='+encodeURIComponent(obj.parentNode.parentNode.getAttribute('col_id')));
}
</script>
</head>
 <body>
        <div id='content'>
        <div style='font-weight:bold; font-size:18pt;' class='labelAll'>
            Контроль авторизації програмного забезпечення користувача v.1.1
        </div><BR>
        <input type='checkbox' id='CB1'".($setting['col_All_Run']==1 ? ' checked ' : '')."
onchange='onchangesetting()'>
        <label for='CB1' class='labelAll'>Загальний дозвіл на роботові в
системі</label><BR>
        <input type='checkbox' id='CB2'".($setting['col_Add']==1 ? ' checked ' : '')."
onchange='onChangeSetting()'>
        <label for='CB2' class='labelall'>Дозвіл на додавання нових користувачів
        </label><BR><BR>
        <div style='font-weight:bold'>Таблиця користувачів</div>
        <table style='border: solid 1px #759cdd; cellpadding:0; cellspacing:1; width:100%;
class='tab_class'>
        <tbody>".$res.
        "</tbody>
        </table>
        </div>
</body>
</html>";
}
```

```
// закінчення режиму [createformadmin]

//--- РЕЖИМ [changeadmin] {зміна параметрів}---
 if ($_POST['mode']=='changeadmin'){
        $connect = mysql_connect("localhost", "user_test", "test8311022802") or
                    DIE("Не можу створити з'єднання");
         mysql_select_db("bd_test", $connect) or DIE("Не можу вибрати базу даних");
         if (isset($_POST['pchangeallrun'])){ // загальні зміна настроювань
             $text_query = sprintf("UPDATE tab_Programsetting
                                    SET col_All_Run=%d, col_Add=%d",
                                    $_POST['pChangeAllRun'], $_POST['pChangeAdd']);
         }
         else{ // зміна настроювань користувача
              $text_query = sprintf("UPDATE tab_Programuser
                                    SET col_Run=%d WHERE col_id=%d",
                                     $_POST['pcb'], $_POST['pid']);
          }
           $query = mysql_query($text_query, $connect);
     } // закінчення режиму [changeadmin]

?>
```