УДК 004.4+004.056.5

Погромська Г.С.
Миколаївський національний університет імені В.О. Сухомлинського

Христодоров О.В.
Миколаївський національний університет імені В.О. Сухомлинського

ШИФРУВАННЯ ДАНИХ ЯК СКЛАДНИК СТВОРЕННЯ ВІЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ В ГАЛУЗІ НАВЧАЛЬНОГО ПРОЦЕСУ

У статті надані особливості реалізації алгоритму криптографічного методу шифрування / дешифрування даних RSA й описані характеристики та технологія криптографічного протоколу SSL. Розроблена захищена процедура автентифікації користувачів засобами мови Qt для модернізації програмного продукту "StudBD" v 1.0 із відкритим кодом для обліку відомостей про студентів вищого навчального закладу та їхню успішність із навчальних дисциплін на посилення криптографічного складника. Надано опис логіки шифрування даних, опрацьовуваних програмним продуктом.

Ключові слова: *база даних, вільне програмне забезпечення, шифрування даних, Qt, C++, RSA, SSL.*

Постановка проблеми. Оброблення та зберігання інформації – це невід'ємний складник функціонування будь-якого підприємства, організації, університети не є винятком. Метою захисту даних, що зберігаються в інформаційній системі, є перешкоджання неконтрольованому поширенню інформації, запобігання втраті її або неможливості доступу до неї, і, як результат, – забезпечення безперебійної роботи організації і зведення до мінімуму шкоди від подій, що загрожують безпеці.

З погляду захисту інформації комп'ютерна система розглядається як набір функціональних послуг. Кожна послуга являє собою набір функцій, які дозволяють протистояти певним загрозам. Під час розроблення програмного забезпечення (далі – ПЗ) варто визначитися, від чого треба захищати дані в програмній системі. Це можуть бути: загрози конфіденційності (пов'язані з несанкціонованим ознайомленням з інформацією), загрози цілісності (що стосуються несанкціонованої модифікації або знищення інформації), загрози доступності (порушення можливості використання комп'ютерних систем або оброблюваної інформації) і загрози спостережливості (пов'язані з порушенням ідентифікації і контролем над діями користувачів, керованістю комп'ютерної системи).

Вибір мови програмування та середовища розроблення під час створення вільного програмного забезпечення важливі з погляду ефективності та вартості роботи програмістів, а також кінцевої вартості продукту. Це зумовлює актуальність дослідження питань реалізації вільного ПЗ на підтримку діяльності навчального процесу із забезпеченням належного рівня захисту конфіденційних даних. Адже в умовах сучасної жорсткої конкуренції розроблень програмного забезпечення питання коштів, які доведеться витратити на розроблення, придбання програмного продукту, є досить вагомим.

Аналіз останніх досліджень і публікацій. Теоретичний доробок у галузі створення та впровадження вільного програмного забезпечення, зокрема в системі освіти, є значним. Багато закордонних дослідників, серед яких Х. Браінінк, Е. Верхалст, К. Лауверс, Р. Столмен [7; 11], та вітчизняних науковців, як-от М. Карпенко, М. Кияк, О. Кравчина, Р. Селіверстов, І. Саврас [2; 4], розглядали це питання. Аналіз досліджень дозволив дійти висновку, що сьогодні багатьма країнами визнана доцільність використання вільного програмного забезпечення в державному секторі та сфері освіти. Вільне програмне забезпечення має чимало важливих, зокрема й стратегічних, переваг, чи не найголовнішою серед яких є можливість суттєвого заощадження бюджетних коштів. Проте нормативна неврегульованість,

недостатня обізнаність населення щодо зазначеного, а також спорадичні та несистемні згадки про нього у вітчизняних медіа є найголовнішими причинами інерційності користувачів у цьому питанні.

Сучасний стан розвитку мов і середовищ програмування відображений у роботах багатьох практиків, а саме: Г. Буча, Б. Вагнера, В. Кауфмана, К. Лармана, Б. Мейєра, М. Фаулера, Б. Еккеля [1; 3; 5; 8; 10; 16] тощо. Автори розглянули фундаментальні концепції та принципи, втілені в сучасних та перспективних мовах програмування. Практиками представлені різні стилі програмування: операційний, ситуаційний, функціональний, реляційний, паралельний, об'єктно-орієнтований.

Питання захисту інформації, зокрема методи захисту даних, розробляли багато всесвітньо відомих криптографів, як-от: М. Дік, К. Гентрі, Ш. Гелеві, Р. Рівест [9]. Такими науковцями, як: М. Баварський, Б. Газі, М. Судан, Р. Ривест та ін. [12; 14], запропоновано гомоморфні методи шифрування, оптимізації коляративних вибірок, оборотність виключної диз'юнкції для двійкових кортежів тощо. Зазначені методи застосовуються для модифікації способів захисту й унеможливлення несанкціонованого доступу до даних.

Спираючись на сучасні розробки провідних криптографів і логічний аналіз потенційних вразливостей даних ПЗ, дійшли висновку, що питання захисту інформації в сучасних ІС висвітлено достатньо, але для вирішення проблеми вибору методу та способів захисту даних у розробленому програмному забезпеченні здебільшого вирішальним чинником є власні знання та досвід програміста.

Постановка завдання. Метою статті є модернізація програмного продукту з відкритим кодом для обліку відомостей про студентів вищого навчального закладу та їхньої успішності з навчальних дисциплін "StudBD" v 1.0 на посилення криптографічного складника. Останнє зумовило завдання даної статті, а саме:

1. Провести аналіз криптографічних методів шифрування та дешифрування даних.

2. Розробити захищену процедуру автентифікації користувачів засобами мови Qt для ПЗ "StudBD" v 1.0.

3. Надати опис логіки шифрування даних опрацьовуваних програмним продуктом.

Виклад основного матеріалу дослідження. Проведений у [6] аналіз наявних систем, спрямованих на підтримку роботи навчального закладу, як-от АС «Деканат» Науково-дослідного інституту прикладних програмних інформаційних технологій, м. Київ, та пакету програм «Деканат» фірми «Політек-софт» дозволив дійти висновку, що зазначені програмні засоби забезпечують автоматизацію обліку роботи навчального закладу, але кожна з них є платною, що знижує їхню конкурентоспроможність для навчальних закладів у сучасних умовах. У зв'язку із цим спробували створити замінник цим платним аналогам, беручі до уваги переваги вільного програмного забезпечення.

2014 р. нами розроблено ПЗ "StudBD" v 1.0, що є клієнт-серверним програмним продуктом, написаним на діалекті мови програмування C++ комплексного середовища Qt. Основними перевагами програмної системи "StudBD" v 1.0, на нашу думку, є її відкритість, багатоплатформеність та використання бібліотеки Qt. У своїй основі пропонована програмна система має інтерфейс для роботи користувачів (розроблений у середовищі Qt), базу даних (MySQL). До складу програмного продукту входить модифікований стандартний компонент QTableWidget (додана можливість виділення рядків та реалізований сигнал реагування на подвійний клік лівої кнопки миші), який можна використовувати в подальшому масштабуванні інших відкритих програмних продуктів.

Основні характеристики ПЗ "StudBD" v 1.0 надані в [6]. Пропонований програмний продукт "StudBD" v 1.0 підпадає під відкриту ліцензію GPL, яка передбачає повний доступ до сирцевих кодів розроблення програмного забезпечення, база даних теж має права GPL, що забезпечує вільний доступ до модифікації структури бази даних.

Варто взяти до уваги такий вагомий складник створення програмного продукту, як забезпечення захисту інформації, особливо коли це стосується персональних даних. У процесі реалізації даної програмної системи ми вважаємо це за важливе, тому розширили можливості першої версії ПЗ на ускладнення несанкціонованого доступу до оброблювальної інформації додатка.

Розглянемо криптографічні методи шифрування та дешифрування даних. Криптографія – це сукупність методів, стандартів і протоколів кодування, покладених на посилення та зберігання даних. Розглянемо найбільш поширені алгоритми шифрування та дешифрування даних. RSA [15] побудований на принципі складності факторизації, криптографічна система, яка використовує два ключі – відкритий і секретний, які разом утворюють пари ключів. Відкритий ключ можна зберігати у відкритому вигляді – він використовується для шифрування даних. Якщо повідомлення було

зашифроване відкритим ключем, то розшифрувати його можна тільки відповідним секретним ключем.

Генерація ключів. Для генерування пари ключів треба:

a) обрати два великих простих числа p та q;

b) обчислити їхній добуток $n = pq$;

c) обчислити функцію Ейлера $\varphi(n) = (p - 1)(q - 1)$;

d) обрати ціле e, таке, що $1 < e < \varphi(n)$ та e взаємно просте із $\varphi(n)$;

e) за допомогою розширеного алгоритма Евкліда знайти число d, таке, що $ed \equiv 1 \bmod (\varphi(n))$.

Число n називається модулем, а числа e і d – відкритою й секретною експонентами відповідно. Пари чисел (n, e) є відкритою частиною ключа, а (n, d) – секретною. Числа p і q після генерації пари ключів можуть бути знищені, але в жодному разі не повинні бути розкриті.

Шифрування й дешифрування. Щоб зашифрувати повідомлення $m < n$, обчислюється $c = m^e \bmod n$.

Число c і використовується як шифр тексту. Для розшифрування потрібно обчислити $m = c^d \bmod n$.

Неважко переконатися, що під час розшифрування ми відновимо вихідне повідомлення: $c^d \equiv (m^e)^d \equiv m^{ed} \pmod n$.

З умови $ed \equiv 1 \bmod (\varphi(n))$ виходить, що $ed = k\varphi(n) + 1$ для деякого цілого k, отже, $m^{ed} \equiv m^{k\varphi(n)+1} \pmod n$.

Згідно з теоремою Ейлера:

$m^{\varphi(n)} \equiv 1 \pmod n$,

тому

$m^{k\varphi(n)+1} \equiv m \pmod n$;

$c^d \equiv m \pmod n$.

Цифровий підпис. RSA може використовуватися не тільки для шифрування, але й для цифрового підпису. Підпис s повідомлення m обчислюється з використанням секретного ключа за формулою:

$s = m^d \bmod n$.

Для перевірки правильності підпису потрібно переконатися, що виконується рівність:

$m = s^e \bmod n$.

SSL [13] – криптографічний протокол, який забезпечує встановлення безпечного з'єднання між клієнтом і сервером. SSL спочатку розроблений компанією Netscape Communications . Згодом, на підставі протоколу SSL 3.0, був розроблений і ухвалений стандарт RFC – TLS.

Протокол забезпечує конфіденційність обміну даними між клієнтом і сервером, що використовують TCP / IP, причому для шифрування використовується асиметричний алгоритм із відкритим ключем. Для шифрування з відкритим ключем використовується два ключі, будь-яким із них можна шифрувати повідомлення. Якщо для шифрування використовується один ключ, то для розшифровки потрібен інший. У такій ситуації можна отримувати захищені повідомлення, тобто публікувати відкритий ключ, а секретний зберігаючи в таємниці.

Протокол SSL складається із двох підпротоколів: SSL запису і стиснення. Протокол SSL запису визначає формат, який застосовується для передачі даних. Протокол SSL включає стиснення з використанням протоколу SSL запису для обміну серіями повідомлень між сервером і клієнтом під час встановлення першого з'єднання. Для роботи SSL потрібно, щоб на сервері був SSL-сертифікат.

SSL надає канал, що має три основні властивості [13], як-от:

Аутентифікація. Сервер завжди автентифікований, тоді як клієнт автентифікований залежно від алгоритму.

Цілісність. Обмін повідомленнями передбачає перевірку цілісності.

Конфіденційність каналу. Шифрування використовується після встановлення з'єднання і для всіх наступних повідомлень.

У протоколі SSL всі дані передаються у вигляді записів-об'єктів, що складаються із заголовка і переданих даних. Передача починається із заголовка. Заголовок містить або два, або три байти коду довжини. Якщо старший біт у першому байті коду дорівнює одиниці, то запис не має заповнювача, повна довжина заголовка дорівнює двом байтам, інакше запис містить заповнювач, а повна довжина заголовка дорівнює трьом байтам. Код довжини запису не включає в себе число байтів заголовка. Довжина запису двобайтового заголовка (зразок подано мовою Java):

RecLength = (byte [0] & 0x7F <<8) | byte [1].

Тут *byte [0]* і *byte [1]* – перший і другий отримані байти.

Довжина запису трибайтового заголовка:

RecLength = (byte [0] & 0x3F <<8) | byte [1];

Escape = (byte [0] & 0x40)! = 0;

Padding = byte [2];

Тут *Padding* визначає число байтів, доданих відправником до початкового тексту для того, щоб зробити довжину запису кратною розміру блока шифру за використання блокового шифру.

Тепер відправник «заповненого» запису додає заповнювач до наявних даних і шифрує все це. Причому вміст заповнювача жодної ролі не відіграє.

Оскільки обсяг переданих даних відомий, заголовок може бути сформований з урахуванням *Padding*.

У свою чергу, одержувач запису дешифрує все поле даних і отримує повну вихідну інформацію. Потім обчислюється значення *RecLength* за відомим *Padding*, і заповнювач із поля даних видаляється. Дані запису SSL складаються із трьох компонент:

1. MAC_Data [Mac_Size] – (Message Authentication Code) – код аутентифікації повідомлення.

2. Padding_Data [Padding] – дані заповнювача.

3. Actual_Data [N] – реальні дані.

Коли записи надсилаються відкритим текстом, очевидно, що жодні шифри не використовуються. Тоді довжини *Padding_Data* і *MAC_Data* дорівнюють нулю. За шифрування *Padding_Data* залежить від розміру блока шифру, а *MAC_Data* залежить від вибору шифру. Приклад обчислення *MAC_Data*:

MacData = Hash (Secret, Actual_Data, Padding_Data, Sequence_Number).

Значення *Secret* залежить від того, хто (клієнт або сервер) посилає повідомлення. *Sequence_Number* – лічильник, який інкрементується як сервером, так і клієнтом. Тут *Sequence_Number* є 32-хбітовим кодом, який передається хеш-функції у вигляді 4-х байтів, причому першим передається старший байт. Для *MD2, MD5 MAC_Size* дорівнює 16 байтам (128 бітам). Для двобайтового заголовка максимальна довжина запису дорівнює 32 767 байтам, а для трибайтового заголовка – 16 383 байти.

Виконаний аналіз дозволив розробити *захищену процедуру автентифікації користувачів.*

Програмна система “StudBD” v 2.0, якщо порівнювати з версією “StudBD” v 1.0, модифікована нами додаванням функцій шифрування оброблювальних даних додатка на всіх етапах роботи з даними користувачів.

У програмному продукті “StudBD” v 2.0 розроблена система автентифікації без явного збереження пароля на пристрої користувача. Початковий вхід у програмне середовище “StudBD” здійснюється заповненням полів початкової форми (рис. 1) – «Логін» і «Пароль». За відсутності облікового запису в БД системи надається можливість створення нового облікового запису (рис. 2).

Під час автентифікації програмний продукт генерує зашифрований ключ, який є як ключем доступу до програмного продукту, так і ключем відображення інформації, до якої має доступ окремий користувач. Програмну реалізацію зазначених дій представлено на рис. 3.

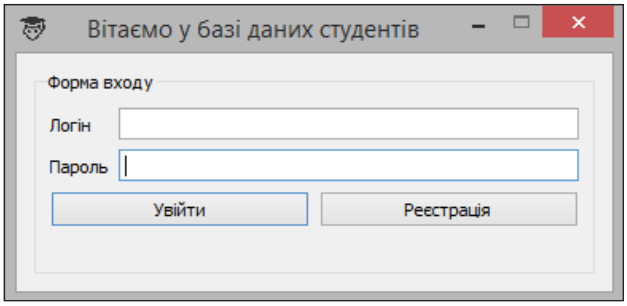

Рис. 1. Початкова форма входу

Ім’ям користувача є зашифрований ключ, згенерований на етапі реєстрації (див. рис. 2).

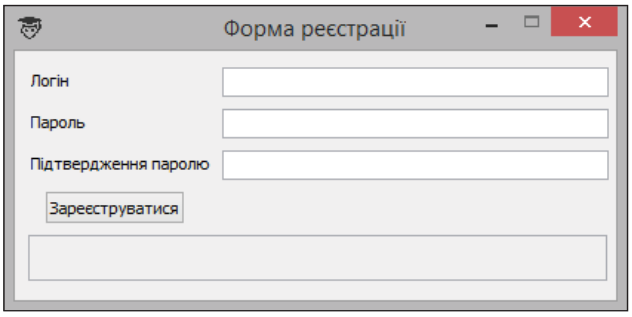

Рис. 2. Вікно створення облікового запису

Завдяки відсутності збереженої локальної копії пароля та шифруванню самих токінів доступу до всієї інформації досягається високий рівень захисту даних, згенерованих програмним продуктом.

Опишемо розроблену логіку шифрування даних, опрацьовуваних додатком. Для шифрування даних у “StudBD” v 2.0 застосовується криптографічний алгоритм RSA. Безпека алгоритму RSA зумовлена принципом складності факторизації цілих чисел. Алгоритм використовує два ключі – відкритий (public) і секретний (private), відкритий і відповідний йому секретний ключі разом утворюють пари ключів (key pair). Відкритий ключ не потрібно зберігати в таємниці, він використовується для шифрування даних. Якщо повідомлення було зашифровано відкритим ключем, то розшифрувати його можна тільки відповідним секретним ключем.

Висновки. Проведений аналіз криптографічних методів шифрування та дешифрування даних дозволив виконати модернізацію програмного продукту “StudBD” v 1.0 на підтримку обліку відомостей студентів навчальних закладів та їхньої успішності з навчальних дисциплін. Порівняно з версією 1.0 програмного продукту “StudBD” (2015 р.) до версії 2.0 були додані функції, що унеможливлюють несанкціонований доступ до даних, оброблюваних у додатку. Завдяки цьому

```
boolean isLogin() {
    String ConcatPassword = "";
    ConcatPassword = Md5Util.md5(first.getText().toString() +
second.getText().toString() + thirt.getText().toString() + fourth.getText().toString());
    sPref = getSharedPreferences("users", MODE_PRIVATE);
    SharedPreferences.Editor ed = sPref.edit();
    String PasswordFromPreferences = sPref.getString(USER_CREATE, "");
    Toast.makeText(this, ConcatPassword.equals(PasswordFromPreferences) ?
"Вітаннячка!" : "Введіть валідний пароль", Toast.LENGTH_LONG).show();
    if (ConcatPassword.equals(PasswordFromPreferences)) {
      return true;
    } else {
      return false;
    }
  }
```

Рис. 3. Програмна реалізація створення нового облікового запису

рігання й оброблення інформації за студентами; додавання інформації про студентів; зберігання інформації щодо сесій. Під час модернізації програмного продукту особливу увагу приділено забезпеченню захисту інформації, обробленню та зберіганню персональних даних.

У процесі модернізації у версії 2.0. розроблено систему доступу за токіном, без необхідності збереження паролів на пристрої користувача. Додаток генерує унікальні ключі для кожного користувача, що додає стійкості перед несанкціонованим доступом. Вся внесена конкретним користувачем інформація шифрується, доступ до неї уможливлюється тільки за знання пароля доступу до програмного продукту конкретного користувача програмної системи.

програмний продукт забезпечує: адміністрування (управління програмною системою користувачами; зміну, редагування, створення сесій; додавання та редагування напрямів підготовки; редагування списку факультетів навчального закладу; редагування, зміну та створення дисциплін); збе-

До перспективного напряму подальшого розвитку пропонованого програмного продукту “StudBD” v 2.0 відносимо можливість реалізації функціоналf програмної системи із застосуванням хмарних технологій.

Список літератури:

1. Буч Г. Объектно-ориентированный анализ и проектирование. М: Вильямс, 2008. 720 с.
2. Карпенко М. Перспективи та можливості впровадження вільного програмного забезпечення в навчальних закладах та державних установах України. URL: http://old.niss.gov.ua/Monitor/june2009/15.htm/.
3. Кауфман В. Языки программирования. Концепции и принципы. М.: ДМК Пресс, 2010. 464 с.
4. Кравчина О. Основні напрями використання вільного програмного забезпечення в закладах освіти зарубіжжя. URL: http://archive.nbuv.gov.ua/e-journals/ITZN/em20/content/10kojaie.htm.
5. Ларман К. Применение UML 2.0 и шаблонов проектирования. Введение в объектно-ориентированный анализ, проектирование и итеративную разработку. СПб.: Вильямс, 2013. 736 с.
6. Погромська Г., Христодоров О. Реалізація вільного програмного забезпечення засобами комплексного середовища розробки міжпатформових додатків Qt // Вимірювальна та обчислювальна техніка в технологічних процессах. 2015. № 1 (50). С. 224–232.
7. Столмен Р. Свободные программы в учебных заведениях. Операционная система GNU. URL: http://www.gnu.org/education/edu-schools.html.
8. Страуструп Б. Язык программирования С++. М: Бином; Невский диалект, 2008. 1136 с.
9. Dijk M. Fully Homomorphic Encryption over the Integers. URL: https://www.researchgate.net/publication/220335489_V_Fully_Homomorphic_Encryption_over_the_Integers.
10. Eckel B. Thinking in Java. Prentice Hall Ptr, 2006. 1079 p.
11. Free Soft ware in education Advise, vision and propose daction plan / Herman Bruyninckx, Mark de Quid, Wilfried Feijens, Kim Lauwers, Eric Verhulst. URL: http://www.ond.vlaanderen.be/ict/english/free_software_in_ed_Flemish_Community_advise.pdf.
12. Mohammad B. Ghazi B. Haramaty E. The Optimality of Correlated Sampling. Electronic Colloquium on Computational Complexity. 2016. № 194. URL: https://eccc.weizmann.ac.il/report/2016/194/.
13. Open SSL Cryptography and SSL / TLS Toolkit. URL: https://www.openssl.org/docs/faq.html.
14. Rivest R. The invertibility of the XOR of rotations of a binary word. International Journal of Computer Mathematics. 2009. № 21. P. 281–284. URL: http://www.tandfonline.com/doi/full/10.1080/00207161003596708.
15. RSA 21st century enlightenment. URL: https://www.thersa.org/.
16. Wagner B. Effective C# (Covers C# 4.0): 5.0 Specific Ways to Improve Your C#. Addison-Wesley Professional, 2010. 352 p.

ШИФРОВАНИЕ ДАННЫХ КАК СОСТАВЛЯЮЩАЯ СОЗДАНИЯ СВОБОДНОГО ПРОГРАМНОГО ОБЕСПЕЧЕНИЯ В ОБЛАСТИ УЧЕБНОГО ПРОЦЕССА

В статье представлены особенности реализации алгоритма криптографического метода шифрования / дешифрования данных RSA, описаны характеристики и технология криптографического протокола SSL. Разработана защищенная процедура аутентификации пользователей средствами языка Qt для модернизация программного продукта "StudBD" v 1.0 с открытым кодом для учета сведений о студентах вуза и их успеваемости по учебным дисциплинам для усиления криптографической составляющей. Представлено описание логики шифрования данных, обрабатываемых программным продуктом.

Ключевые слова: *база данных, свободное программное обеспечение, шифрование данных, Qt, C ++, RSA, SSL.*

THE DATA ENCRYPTION AS OPERATION COMPONENT OF THE CREATION OF THE FREE SOFTWARE IN EDUCATIONAL PROCESS

The article deals with the features of implementation of the cryptographic technique algorithm of encrypting/ decrypting RSA data and describes the characteristics and technology of SSL cryptographic protocol. It was designed the users authentication procedure by means of the Qt language for upgrading of software product "StudBD" v 1.0 with the open code for the accounting of information about university students and their achievement of academic disciplines to strengthen the cryptographic component. It is provided the description of data encryption logic processed by a software product.

Key words: *database, the free software, data encryption, Qt, C ++, RSA, SSL.*