УДК 004.056.53

КЛІЄНТ-СЕРВЕРНА СИСТЕМА ДЛЯ БЕЗПЕЧНОГО ОБМІНУ ПРИВАТНИМИ ПОВІДОМЛЕННЯМИ ІЗ ЗАСТОСУВАННЯМ КРИПТОГРАФІЇ З ВІДКРИТИМ КЛЮЧЕМ

ПЕТРЕНКО О. М. [1*] *д.т.н., професор,*
КЛИМЕНКО С. В. [2*] *к.т.н., доцент,*
ПОЛЯКОВ Г. О. [3*] *к.т.н., доцент.*

[1*] Петренко Олександр Миколайович Дніпровський національний університет імені Олеся Гончара, декан фізико-технічного факультету, д.т.н., професор кафедри радіоелектронної автоматики, dnu.petrenko@gmail.com
[2*] Клименко Світлана Володимирівна Дніпровський національний університет імені Олеся Гончара, фізико-технічний факультет, к.т.н., доцент кафедри радіоелектронної автоматики, klimenko_s_v@ua.fm
[3*] Поляков Гліб Олександрович Дніпровський національний університет імені Олеся Гончара, фізико-технічний факультет, к.т.н., доцент кафедри радіоелектронної автоматики, gleb.a.polyakov@gmail.com

Анотація. ***Мета.*** Метою роботи є розробка кроссплатформенної системи, яка забезпечує високий рівень інформаційного захисту передачі повідомлень, і при цьому може експлуатуватися в середовищі операційних систем Windows та Linux. ***Методика.*** Розробка програмного забезпечення для шифрування приватних повідомлень за допомогою клієнт-серверної системи обміну, що використовує криптографічні перетворення з відкритим ключем. Високий рівень інформаційної безпеки досягається за рахунок використання криптографічного алгоритму RSA. ***Результати.*** Розроблений програмний продукт є актуальним, оскільки спілкування та передача інформації – невід'ємна частина діяльності людини, як і захист особистої інформації. Проведено аналіз існуючих клієнт-серверних систем обміну приватними повідомленнями; розроблений програмний продукт, який дозволяє передавати зашифровану послугу між користувачами локальної мережі. ***Практична значимість.*** Результати роботи можуть представляти інтерес для всіх, хто хоче захистити передачу інформації (повідомлення) від несанкціонованого доступу.

Ключові слова: криптографія; шифрування; rsa; windows; linux; java; клієнт-сервер

КЛИЕНТ-СЕРВЕРНАЯ СИСТЕМА ДЛЯ БЕЗОПАСНОГО ОБМЕНА ЛИЧНЫМИ СООБЩЕНИЯМИ С ИСПОЛЬЗОВАНИЕМ КРИПТОГРАФИИ С ОТКРЫТЫМ КЛЮЧОМ

ПЕТРЕНКО А. Н. [1*] *, д.т.н., профессор,*
КЛИМЕНКО С. В. [2*] *, к.т.н., доцент,*
ПОЛЯКОВ Г. А. [3*] *, к.т.н., доцент.*

[1*] Петренко Александр Николаевич Днепровский национальный университет имени Олеся Гончара, декан физико-технический факультет, д.т.н., профессор кафедры радиоэлектронной автоматики. dnu.petrenko@gmail.com
[2*] Клименко Светлана Владимировна Днепровский национальный университет имени Олеся Гончара, физико-технический факультет, к.т.н., доцент кафедры радиоэлектронной автоматики. klimenko_s_v@ua.fm
[3*] Поляков Глеб Александрович, Днепровский национальный университет имени Олеся Гончара, физико-технический факультет, кафедра радиоэлектронной автоматики. gleb.a.polyakov@gmail.com

Аннотация. ***Цель***. Целью работы является разработка кроссплатформенной системы, которая обеспечивает высокий уровень информационной защиты передаваемых сообщений, и при этом может эксплуатироваться в среде операционных систем Windows и Linux. ***Методика.*** Разработка программного обеспечения для шифрованной передачи частных сообщений с помощью клиент-серверной системы обмена, использующей криптографические преобразования с открытым ключом. Высокий уровень информационной безопасности достигается за счет использования криптографического алгоритма RSA. ***Результаты.*** Разработанный программный продукт является актуальным, поскольку общение и передача информации – неотъемлемая часть деятельности человека, также как и защита личной информации. Проведен анализ существующих клиент-серверных систем обмена частными сообщениями; разработан программный продукт, который позволяет передавать зашифрованные сообщение между пользователями локальной сети. ***Практическая значимость.*** Результаты работы могут представлять интерес для всех, кто хочет защитить передаваемую информацию (сообщения) от несанкционированного доступа.

Ключевые слова: криптография; шифрование; rsa; windows; linux; java; клиент-сервер

CLIENT-SERVER SYSTEM FOR SECURE PRIVATE MESSAGES EXCHANGE USING PUBLIC KEY CRYPTOGRAPHY

PETRENKO O. M. [1*], *doctor of sciences, professor,*
KLYMENKO S. V. [2*], *PhD, associate professor,*
POLIAKOV H. O. [3*], *PhD, associate professor.*

[1*] Petrenko O. M. Oles Honchar Dniprovsky National University, Faculty of Physics and Technology, Department of Radioelectronic Automation. dnu.petrenko@gmail.com
[2*] Klymenko S. V. Oles Honchar Dniprovsky National University, Faculty of Physics and Technology, Department of Radioelectronic Automation. klimenko_s_v@ua.fm
[3*] Poliakov H. O., Oles Honchar Dniprovsky National University, Faculty of Physics and Technology, Department of Radioelectronic Automation. gleb.a.polyakov@gmail.com

Abstract. ***Purpose.*** The purpose of this application is to develop a portable system product that provides a high level of information security, while capable to run in Windows and Linux environments. ***Methodology.*** Development of software for encrypted transmission of private messages using a client-server exchange system uses public key cryptographic.High level of information security is achieved through the use of the RSA encryption algorithm. ***Results.*** The developed system is actual, since communication and transmission of information – an integral part of human activity, as well as protection of personal information. An analysis of existing client-server systems for the exchange of private messages is performed, capable software product for encrypted transmission of messages between network users is developed. ***Practical significance***. Research results may be interesting to anyone who wants to protect information (messages) from unauthorized access during transmission.

Keywords*:* cryptography; rsa; encryption; windows; linux; java; client-server

Вступ

На сьогоднішній день інформація грає велику роль серед людей. Ми обмінюємося інформацією з часів початку людства у вигляді повідомлень (жести, слова, письмо). Часто виникає потреба у передачі конфіденційного повідомлення. Але використання складних криптографічних систем для збереження конфіденційності і забезпечення передачі самого повідомлення ускладнене. Однією із таких причин є неможливість використання якісних надійних продуктів криптографічного захисту, які, в основному, являються комерційними і для простого користувача персонального комп'ютера (ПК), а також для навчання студентів є недоступними.

Вирішенням цієї проблеми э розробка системи, а саме програмного продукту, з використання існуючих алгоритмів шифрування даних, що реалізовує можливість безпечного обміну інформацією (текстовими повідомленнями). А також розроблення програмного забезпечення дозволить студентам, які навчаються за спеціальністю 125 «Кібербезпека» розглянути на практиці алгоритм шифрування та передачі даних в мережі.

Щодо визначення терміну криптографії, криптографія (від грецького kryptós – прихований і gráphein – писати) – наука про математичні методи забезпечення конфіденційності, цілісності і автентичності інформації з допомогою криптографічних перетворень [4, 7]. З точки зору практичної потреби криптографія надає можливість передавати важливі відомості найнадійнішим чином.

Опис криптографічної клієнт-серверної системи

При розробці системи використовувався один із криптографічних алгоритмів, а саме RSA (абревіатура від прізвищ авторів алгоритму – Rivest, Shamir та Adleman). Розроблена клієнт-серверна система забезпечує передачу шифрованих повідомлень (файлів від відправника до сервера, та від сервера отримувачу), тобто, шифрувати повідомлення відкритим ключем адресата, отриманим зі захищеного від втручання сховища відкритих ключів абонентів системи, розташованого на сервері, та передавати повідомлення на сервер. Після цього повідомлення передається зі сервера адресату (отримувачу), де розшифровується приватним ключем отримувача.

Криптографічна клієнт-серверна система обміну приватними повідомленнями з відкритим ключем являє собою програмний продукт, функціональність якої заключається в наданні можливості спілкування (за допомогою текстових повідомлень), а також обміну невеликими файлами у локальній мережі підприємства чи через глобальну мережу Інтернет. При цьому виключається можливість порушення конфіденційності спілкування (ознайомлення з інформацією інших осіб окрім відправника та адресата).

Клієнт-серверна система – обчислювальна або мережева система, в якій завдання або мережеве навантаження розподілені між постачальниками послуг, званими серверами, і замовниками послуг, званими клієнтами. Клієнт і сервер – це програмне забезпечення (ПЗ), де клієнтська частина (модуль) і серверна частина (модуль) розташовані на різних

електронних обчислювальних машинах (ЕОМ) і взаємодіють між собою через обчислювальну мережу за допомогою мережевих протоколів. Також модулі можуть бути розташовані і на одній машині. Програма-сервер, очікує від програми-клієнта запити і надають їм свої ресурси у вигляді даних. Наприклад, завантаження файлів, потокового мультимедіа, або ж робота з базами даних.

Оскільки одна програма-сервер може виконувати запити від безлічі програм-клієнтів, її розміщують на спеціально виділеній ЕОМ, налаштованій особливим чином, як правило, спільно з іншими програмами-серверами, тому продуктивність цієї машини повинна бути високою. Через особливу роль такої машини в мережі, специфіки її обладнання та програмного забезпечення, її також називають сервером, а машини, які виконують клієнтські програми відповідно – клієнтами, в основному, такими машинами виступають персональні комп'ютери (ПК) користувачів-клієнтів.

Розроблена програма застосовує криптографічний алгоритм шифрування даних RSA (несиметричний алгоритм шифрування даних з відкритим ключем). Цей алгоритм використовується для процесу автентифікації клієнта на сервері, а також для захищеної передачі повідомлень і файлів від відправника до серверу, і від сервера до отримувача. Програма, складається з двох складових (частин-модулів).

Модуль програми – функціонально завершений фрагмент програми, оформлений у вигляді окремого файлу з кодом, призначений для використання в інших програмах. Ці модулі дозволяють розбивати складні задачі на менші відповідно до принципу модульності. Зазвичай, модуль проектується таким чином, щоб надати програмістам зручну для багаторазового використання функціональність (інтерфейс).

Програмне забезпечення призначене для використання у локальній мережі підприємства або у мережі Інтернет. Зв'язок між сервером і клієнтом відбувається за допомогою стеку протоколів TCP/IP.

Стек протоколів TCP/IP, (TCP/IP-модель) – набір протоколів мережі Інтернет. Назва походить від назви протоколів мережі Інтернет – IP (англ. Internet Protocol – «міжмережевий протокол») і TCP (англ. Transmission Control Protocol – «протокол керування передаванням»), це систематизований стек транспортних протоколів, який служить для передачі інформації [6].

Архітектура програмного забезпечення

Архітектура програмного забезпечення (від англ. software architecture) – спосіб структурування програмної або обчислювальної системи, упорядкування елементів системи на певній фазі її роботи [1]. Система може складатись з кількох рівнів обробки даних, і мати багато фаз роботи, кожна з яких може мати окрему архітектуру. Дослідження архітектури програмного забезпечення намагається визначити як найкраще розбити систему на частини (модулі), як ці частини визначають та взаємодіють одна з одною, як між ними передається інформація, як ці частини розвиваються поодинці, а також записати використовуючи формальний або неформальний опис.

Архітектура розробленого програмного забезпечення має наступний вигляд (рис. 1).

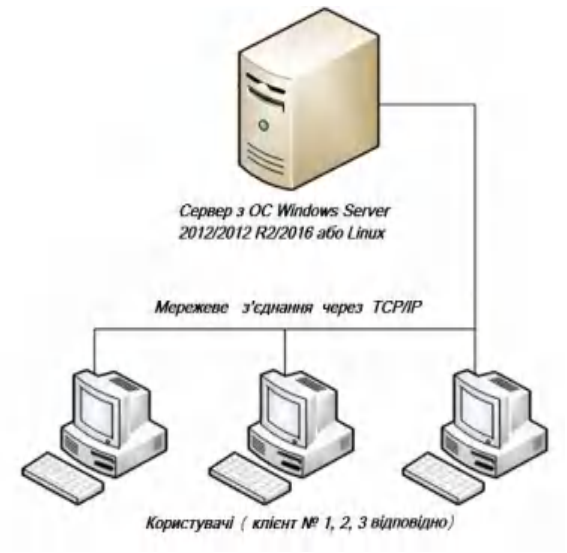

Рис. 1. Архітектура комутації криптографічної клієнт-серверної системи обміну приватними повідомленнями з відкритим ключем / Switching architecture of the client-server system for private massages exchange using public key cryptography

На рис. 1 наведена архітектура комутації криптографічної клієнт-серверної системи, яка призначена для обміну повідомленнями між користувачами всередині локальної мережі, використовуючи стек протоколів TCP/IP.

Процедура обміну інформацією між клієнтом і сервером відбувається наступним чином (рис. 2):

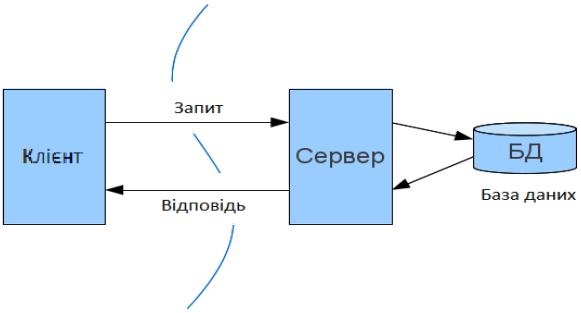

Рис. 2. Процедура обміну інформацією між клієнтом і сервером / Procedure for information exchange between client and server

Процес обміну повідомленнями між клієнтами представлено на рис. 3.

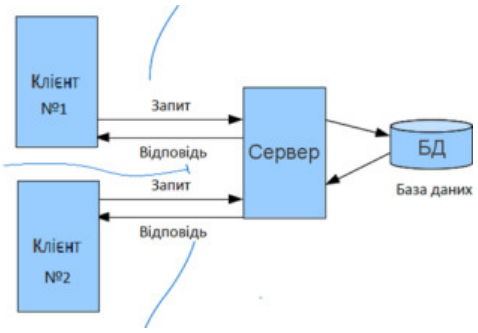

Рис. 3 процедура обміну повідомленнями між клієнтами використовуючи сервер / The procedure for exchanging messages between clients using a server

Опис алгоритму шифрування

Криптографічну систему з відкритим ключем RSA використовують більшість продуктів на ринку інформаційної безпеки. Його криптостійкість ґрунтується на складності розкладання великих чисел на множники. Задача обчислення приватного ключа RSA на підставі відповідного йому відкритого є складною, так як для цього буде потрібно вирішити задачу знаходження дільників дуже великого цілого числа. Найбільш крипостійкі системи на основі RSA використовують числа довжиною 2048–3072 біт (приблизно 600–900 десяткових знаків).

Розглянемо алгоритм RSA з практичної точки зору. Для початку необхідно згенерувати відкритий і приватний ключі, для цього:

– візьмемо два великих простих числа p і q;

– визначимо число n, як результат множення p на q ($n= pq$);

– визначимо $Ф(n)$ як $(p-1)(q-1)$;

– виберемо невелике випадкове число, яке назвемо e. Це число повинно бути менше за $Ф(n)$ та взаємно простим з ним (не мати жодного спільного дільника, крім одиниці);

– визначимо (наприклад, з допомогою узагальненого алгоритму Евкліда) таке число d, для якого є істинним співвідношення $(ed) \bmod Ф(n) = 1$;

– назвемо відкритим ключем числа e і n, а приватним – числа d і n.

Для того, щоб зашифрувати дані з допомогою відкритого ключа $\{e, n\}$, необхідно виконати наступне:

– розбити текст, що зашифровується на блоки, кожен з яких може бути представлений у вигляді числа $M(I)=0,1,2...,n-1$.

– зашифрувати текст, що розглядається як послідовність чисел $M(I)$ за формулою

$$C(I)=(M(I)^e) \bmod n.$$

Щоб розшифрувати ці дані, використовуючи приватний ключ $\{d, n\}$, необхідно виконати наступні обчислення: $M(I)= (C(I)^d) \bmod n$. В результаті чого і буде отримано послідовність чисел $M(I)$, які представляють собою вихідний текст.

Наприклад, зашифруємо і розшифруємо повідомлення *«CAB»* за алгоритмом RSA. Для простоти візьмемо невеликі числа – це скоротить наші розрахунки:

– виберемо $p = 3$ і $q = 11$.

– визначимо $n = 3 \cdot 11 = 33$.

– знайдемо $Ф(n)=(p-1)(q-1)=20$. Оберемо e, наприклад, 7: $(e=7)$.

– знайдемо число d таке, щоб:

$$(d \cdot 7) \bmod 20 = 1,$$

отже d у нашому випадку дорівнюватиме 3: $(d=3)$.

Уявімо повідомлення, яке шифруємо, як послідовність чисел у діапазоні від 0 до 31. Буква $A=1$, $B=2$, $C=3$. Зашифруємо повідомлення, використовуючи відкритий ключ $\{7,33\}$

$C1=(3^7) \bmod 33 = 2187 \bmod 33 = 9$;

$C2 = (1^7) \bmod 33 = 1 \bmod 33 = 1$;

$C3=(2^7) \bmod 33 = 128 \bmod 33 = 29$;

Тепер розшифруємо дані, використовуючи приватний ключ $\{3,33\}$.

$M1 = (9^3) \bmod 33 = 729 \bmod 33 = 3 (C)$;

$M2 = (1^3) \bmod 33 = 1 \bmod 33 = 1 (A)$;

$M3 = (29^3) \bmod 33 = 24389 \bmod 33 = 2 (B)$;

$M= «CAB»$.

Дані розшифровані.

Саме цей алгоритм використовується у якості алгоритму шифрування з відкритим ключем у розроблюваній клієнт-серверній системі обміну приватними повідомленнями.

Аналіз клієнт-серверної системи обміну приватними повідомленнями

Функціональність клієнт-серверної системи обміну приватними повідомленнями з відкритим ключем полягає в зашифрованій передачі повідомлень між двома та більше користувачами використовуючи посередником сервер. Так як розроблена система призначена для обміну повідомленнями між користувачами в одній локальній мережі, то доцільним було використати обмін даними між клієнтом і сервером, використовуючи сокети.

Сокети (англ. socket – заглиблення, гніздо, роз'єм) – назва програмного інтерфейсу для забезпечення обміну даними між процесами . Процеси при такому обміні можуть виконуватися як на одній ЕОМ, так і на різних ЕОМ, пов’язаних між собою мережею. Сокет – абстрактний об’єкт, що представляє кінцеву точку з'єднання. Якщо ПК підключений до обчислювальної мережі у якій декілька комп’ютерів поєднані між собою, розроблена система буде використовувати IP-адрес комп’ютера, на якому запущена, у якості ідентифікатора ПК.

Таким чином, якщо сервер запущений на ПК локальної мережі, IP-адрес якого у даній мережі: 192.168.0.101, то і доступом для серверу являється даний IP-адрес для всіх клієнтів розробленої мережі. Модуль клієнта розроблено аналогічним чином, але в цьому випадку уже сервер звертатиметься до клієнта.

Для розробки клієнт-серверної системи обрано високорівневу мову програмування Java, яка має

велику кількості готових бібліотек, що вже мають свій певний функціонал [3,5].

У якості основної бібліотеки для введення системі функціоналу встановлення зв'язку між клієнтами і сервером обрано стандартні бібліотеки Java – com.socket і com.ui.ChatFrame, що уже вміщають у собі функції передачі даних через мережу, використовуючи сокети, а також систему передачі даних по типу чатів (мережевих засобів для швидкого обміну текстовими повідомленнями між користувачами комп'ютерних мереж в режимі реального часу).

Також використані наступні стандартні бібліотеки:

java.io – бібліотека для зчитування і запису даних;

java.net – бібліотека для роботи з протоколами мережі (TCP), а також з числовими IP-адресами;

java.util.logging (JUL) – бібліотека для виводу даних в файл протоколу;

java.io.serializable – бібліотека, що дозволяє зберігати стан продукту для більш коректної його роботи (оптимізація);

java.xml – бібліотека для роботи с файлами типу .xml, що потрібні для процесу збереження даних, що передаються;

java.org.w3c.dom – бібліотека для зручнішої роботи і маніпуляцій з файлами типу .xml;

Для роботи з графічним інтерфейсом (для зручності) використовувались бібліотеки javax.swing і java.awt.

Структурні схеми передачі і прийому повідомлень клієнт-серверної системи

Криптографічна клієнт-серверна система обміну приватними повідомленнями з відкритим ключем складається з наступних модулів:

– серверний модуль;

– клієнтський модуль (може бути декілька);

Загальний вигляд зв'язку клієнтського модуля і серверного модуля можна зобразити наступною схемою (рис. 4).

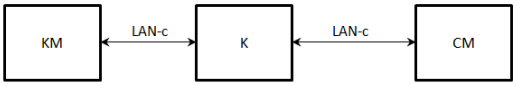

Рис. 4. Комутаційна схема клієнт-серверної системи / Switching scheme of the client-server system

де

КМ – клієнтський модуль (що складається з користувача, ПК користувача і встановленої програми);

К – комутаційний пристрій локальної мережі;

СМ – серверний модуль (що складається з сервера та встановленого програмного забезпечення сервера)

LAN-c – обчислювальна мережа.

Структурні схеми програми (клієнт-серверної системи обміну приватними повідомленнями з відкритим ключем) для сторони відправника і отримувача представлені на рис.5 та рис. 6.

Виділимо три основні частини на стороні відправника:

– отримання відкритого ключа адресата з розташованого сховища відкритих ключів абонентів;

– стиснення повідомлення (за допомогою методу LZ-Huff, опис якого наведено у навчальному посібнику [2]);

– криптографічне перетворення з допомогою відкритого ключа адресата.

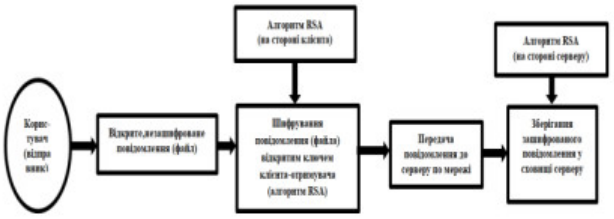

Рис 5. Структурна схема передачі повідомлення через сервер (на стороні відправника) / Block diagram for sending messages through the server (on the sender's side)

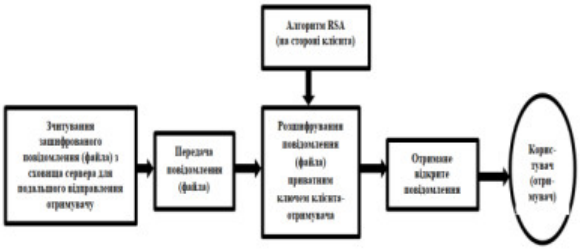

Рис 6. Структурна схема прийому повідомлення від серверу (на стороні отримувача) при передачі повідомлення програмою / The block diagram of receiving messages from the server (on the recipient's side) when sending messages to the program

Принцип роботи клієнт–серверної системи

Підключення клієнта до сервера і принцип передачі даних (повідомлення):

– підключення клієнта через програму-клієнт до сервера (указуючи IP-адрес або ім'я сервера у мережі, і номер порта підключення (стандартний номер порта для розробленої системи: 13000));

– авторизація клієнта на сервері (ідентифікація і автентифікація), надання доступу. Якщо користувач не зареєстрований – слідує процес реєстрації, який включає генерацію клієнтом пари ключів для даного користувача та передачу на сервер відкритого ключа;

– отримання даних у вигляді списку підключених користувачів (програм-клієнтів користувачів);

– вибір отримувача серед списку доступних;

– передача сервером відкритого ключа отримувача, завіреного електронним цифровим підписом (ЕЦП) сервера (для запобігання підробці відкритого ключа при передачі);

– введення повідомлення або вибір файлу для передачі видправником;

– шифрування повідомлення програмою-клієнтом відправника відкритим ключем отримувача, отриманим від сервера (використовуючи алгоритм RSA);

– передача повідомлення через локальну мережу серверу.

Передача повідомлення від сервера отримувачу (адресату):

– отримання шифрованого повідомлення сервером через локальну мережу від відправника;

– якщо отримувач повідомлення зараз авторизован у системі;

– передача повідомлення через мережу програмі-клієнту отримувача (якщо не авторизован – зберігання шифрованого повідомлення на сервері до моменту авторизації отримувача;

– передача повідомлення програмі-клієнту отримувача (сервер виконує функції поштової скриньки);

– розшифрування отриманого повідомлення приватним ключем отримувача (що був згенерований програмою-клієнтом отримувача під час реєстрації отримувача на сервері; цей ключ зберігається отримувачем та ніколи не передається у мережі);

– вивід на екран розшифрованого повідомлення.

Недоліком системи являється не дуже велика швидкість шифрування / дешифрування. Вдосконалення програми планується у наступній версії шляхом використання гібридної криптографії.

Висновки

В статті розглянуто клієнт-серверну систему для обміну приватними повідомленнями із застосуванням криптографічного перетворення з відкритим ключем RSA. Система надає змогу клієнтам (користувачам) у локальній мережі або мережі Інтернет захищено обмінюватися приватними повідомленнями, звуковими файлами, схемами, або картинками. Розроблена система є кросплатформенною – має можливість роботи у середовищах операційних систем Windows та Linux. Система являється простою у користуванні і водночас надійною в плані захищеності передачі інформації (повідомлень), може використовуватися у будь-яких локальних мережах (або у глобальній мережі Інтернет) і експлуатуватись на всіх ОС, які підтримують JVM, а також використовуватися на практичних заняттях студентами, що вивчають роботу алгоритму RSA.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Архітектура комп'ютера: навчальний посібник / Б. І. Мороз, В. П. Дюбко, С. В. Клименко, В. О. Яковенко, В. В. Поліщук. – Дніпропетровськ: «Академія митної служби України», 2012. – 169 с.

2. Поляков Г. О. Основи теорії стиснення повідомлень без втрат інформації: навчальний посібник / Г. О. Поляков. – Дніпропетровськ: РВВ ДНУ, 2013. – 148 с.

3. Орфали Роберт. Java и CORBA в приложениях клиент-сервер / Роберт Орфали, Дан Харки – М.: Лори, 2009. – 716 с

4. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – Москва: Триумф, 2002. – 816 с.

5. Comer Douglas E. Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture /– English, 4/e, 2013. – 755 p.

6. Oleshchuk V. A. On Public-Key Cryptosystem Based on Church-Rosser String-Rewriting Systems. / Oleshchuk V. A. // Computing and Combinatorics: First Annual International Conference, COCOON '95. – Springer, 1995. – С. 264-269.

7. Saternos Casimir. Client-Server Web Apps with JavaScript and Java / Casimir Saternos. – Gardners Books, 2014. – 350 p.

REFERENCES

1. Moroz B.I., Diubko V.P., Klymenko S.V., Yakovenko V.O. and Polishchuk V.V.. *Arkhitektura kompiutera* [Computer architecture]. *Navchalnyi posibnyk* [schoolbook]. Dnipropetrovsk: AMSU, 2012, 169 p. (in Ukrainian).

2. Poliakov H.O.. *Osnovy teorii stysnennia povidomlen bez vtrat informatsii* [Fundamentals of lossless data compression theory]. *Navchalnyi posibnyk* [schoolbook]. Dnipropetrovsk: RVV DNU, 2013,148 p. (in Ukrainian).

3. Orfali Robert and Kharki Dan. *Java i CORBA v prilozheniiach kliyent-server* [Java and CORBA in client-server applications]. Moscow: Lori, 2009, 716 p. (in Russian).

4. Shnaier B. *Prikladnaia kriptografiia. Protokoly, algoritmy, ischodnyye teksty na yazyke Si* [Applied Cryptography. Protocols, Algorithms and Source Code in C]. Moscow: Triumf , 2002, 816 p. (in Russian).

5. Douglas E. Comer. *Internetworking with TCP/IP Volume 1: Principles, Protocols, and Architecture*. English, 4/e. 2013, 755p.

6. Oleshchuk V.A.. *On Public-Key Cryptosystem Based on Church-Rosser String-Rewriting Systems.* Computing and Combinatorics: First Annual International Conference, COCOON '95. – Springer Publ., 1995, pp. 264-269.

7. Saternos Casimir. *Client-Server Web Apps with JavaScript and Java*. Gardners Books, 2014, 350 p.