УДК 004.056+511.176

Ю.Є. Яремчук

ОСОБЛИВОСТІ РОЗРОБЛЕННЯ ПРОГРАМНИХ ЗАСОБІВ ДЛЯ АСИМЕТРИЧНОГО ШИФРУВАННЯ НА ОСНОВІ РЕКУРЕНТНИХ Vk-ПОСЛІДОВНОСТЕЙ

У роботі представлено розробку алгоритмів та програм реалізації асиметричного шифрування інформації на основі V_k –послідовностей. Розглянуто основні особливості реалізації модулів виконання криптографічних операцій, а також модуля реалізації математичного апарату рекурентних V_k –послідовностей, для цього у програмі виділено спеціальний клас, описано параметри та операції цього класу.

Ключові слова: захист інформації, криптографія, асиметричне шифрування, рекурентні послідовності, програмні засоби.

Вступ

В асиметричних криптографічних системах шифрування інформації [1] використовуються різні ключі: відкритий ключ – для зашифрування даних, а секретний – для дешифрування. Відкритий ключ публікується для використання усіма користувачами системи, що бажають зашифрувати і відправити дані одержувачу. Секретний ключ використовується одержувачем для дешифрування даних, він не може бути визначений з відкритого ключа зашифрування, принаймні на це необхідний неприйнятно великий час.

Найбільш відомими асиметричними криптоалгоритмами є RSA [2] та Ель-Гамаля [3]. Стійкість методів асиметричного шифрування базується на складності вирішення математичних задач для великих чисел, однак необхідність виконувати обчислення з великими числами створює певні обчислювальні проблеми і санкціонованому користувачу, що використовує метод.

У цьому зв'язку в роботі [4] було запропоновано метод асиметричного шифрування інформації, що базується на математичному апараті рекурентних U_k –послідовностей, який у порівнянні з відомими аналогами забезпечує спрощення обчислень при достатньому рівні криптографічної стійкості. У роботі [5] представлено алгоритми та особливості програмної реалізації цього методу.

Однак представлений у роботі [4] метод асиметричного шифрування має потенційні можливості для підвищення крипостійкості, тому в роботі [6] запропоновано метод асиметричного шифрування на основі математичного апарату рекурентних V_k^+ –послідовностей, який, у порівнянні з методом шифрування на основі U_k –послідовностей забезпечує вищу криптографічну стійкість, оскільки під час шифрування/дешифрування відкрите/зашифроване повідомлення поєднується з результатом обчислень елементу V_k^+ –послідовностей, обчисленого за мультиплікативним, а не адитивним способом зміни індексу.

При цьому актуальним залишається розроблення алгоритмів та визначення особливостей програмної реалізації представленого у [6] методу асиметричного шифрування на основі V_k –послідовностей.

Розроблення алгоритмів та програм реалізації методу асиметричного шифрування на основі V_k –послідовностей.

Розглянемо особливості розробки програмної реалізації процедур шифрування та дешифрування інформації згідно представленого у [6] методу асиметричного шифрування інформації. Алгоритми реалізації цих процедур представлено на рисунках 1 і 2.

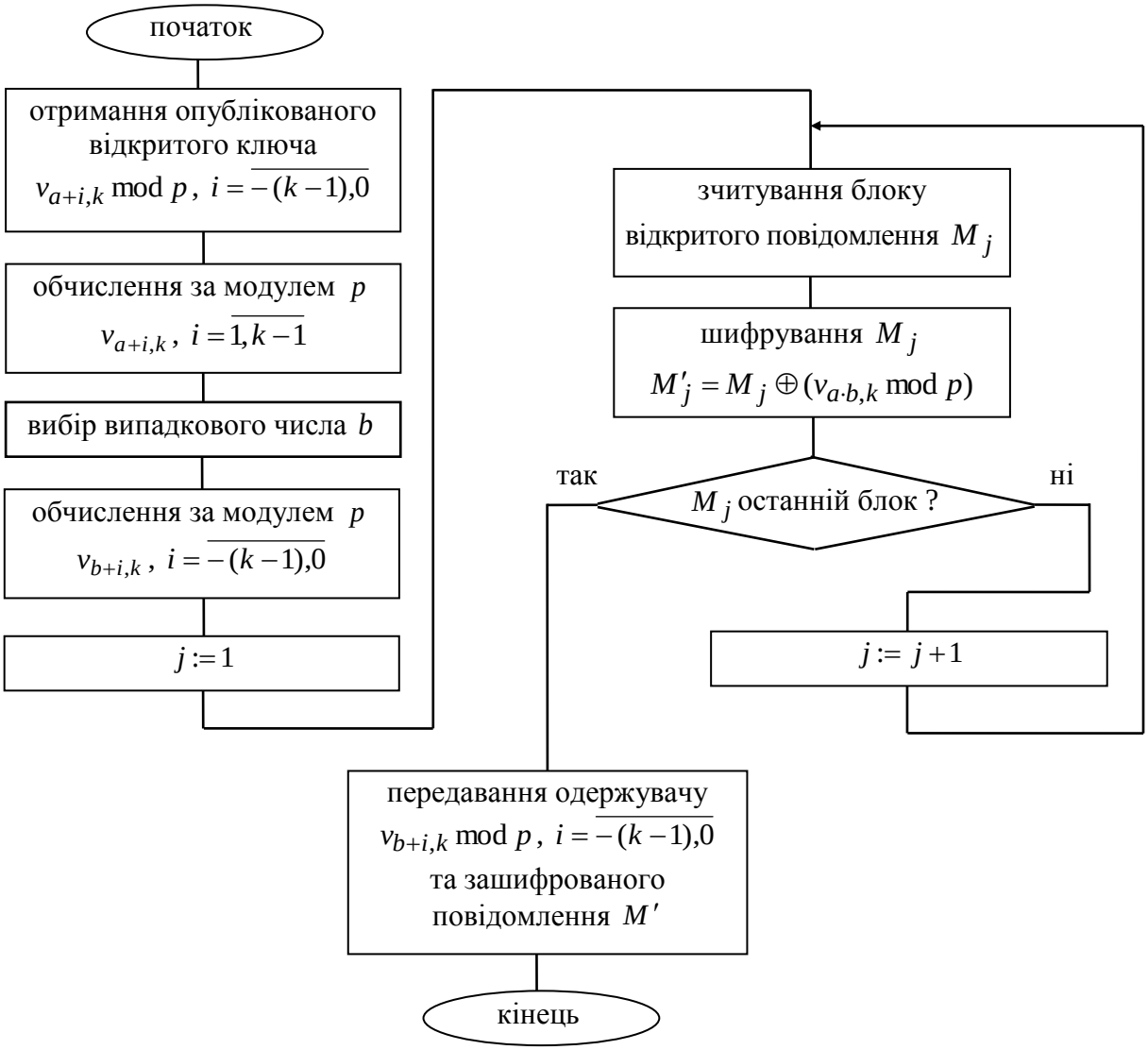

Рис. 1. Структура програми шифрування на основі елементів
V_k – послідовностей з боку відправника

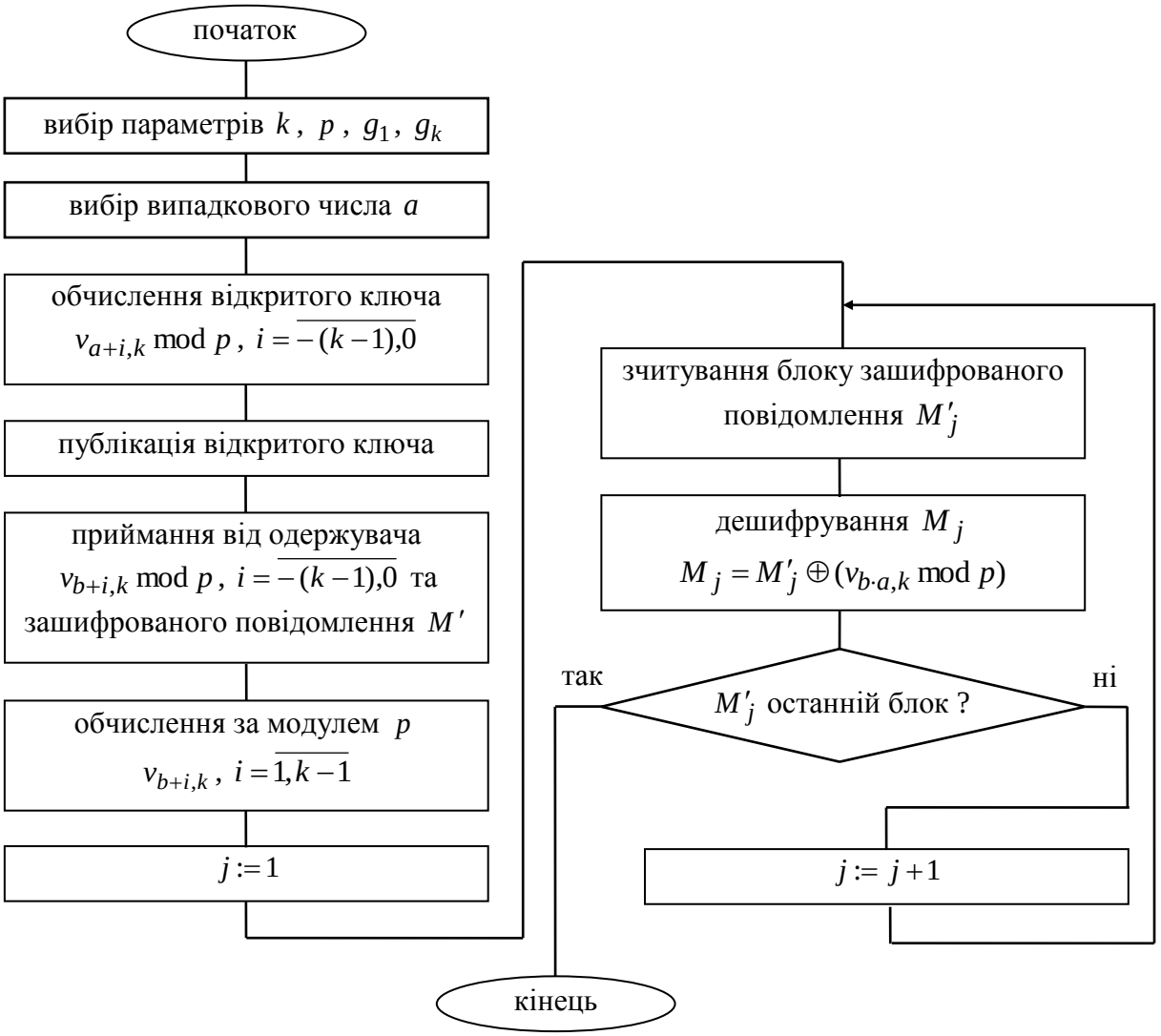

Рис. 2. Структура програми дешифрування на основі елементів
V_k – послідовностей з боку одержувача

Шифрування інформації в наведених алгоритмах відбувається над блоками відкритого повідомлення M, яке розбивається на окремі частини M_j, що визначається розміром модуля – параметру p.

Параметр p вибирається як випадкове число, розрядність якого кратна розрядності машинної одиниці інформації і залежить від можливостей комп'ютера, на якому реалізується програма шифрування інформації. Для сучасних комп'ютерів цю розрядність слід вибирати 1024, 2048 або 4096 розряди.

Вибір параметру p здійснюється в програмному модулі завдання та вибору параметрів, де окрім нього задається параметр k та вибираються коефіцієнти рекурентного співвідношення g_i, $i = \overline{1,k}$.

Коефіцієнти g_i, $i = \overline{1,k}$, вибираються як випадкові числа. Оскільки параметр p є модулем при обчисленнях та визначає верхню границю усіх чисел, що використовуються в алгоритмах шифрування, вибір параметрів g_i, $i = \overline{1,k}$, здійснюється в діапазоні $[1, p]$.

Для вибору параметрів алгоритмів шифрування/дешифрування використаємо звичайні генератори випадкових чисел. Для генерування параметрів g_i, $i=\overline{1,k}$, може використовуватись один з відомих генераторів випадкових чисел, наприклад, лінійний конгруентний генератор.

В алгоритмах шифрування/дешифрування інформації генератор випадкових чисел потрібен і для вибору секретних ключів a та b. Для вибору секретних ключів рекомендується використовувати більш випадкові генератори. Наприклад, генератор оснований на затримках між натисненнями клавіш клавіатури.

Програмна реалізація представленого у [6] методу асиметричного шифрування проводилася мовою програмування Microsoft .Net Framework 4.5 з використанням мови C#, та Microsoft SQL Server 2012, так як в них є базова реалізація роботи з числами багатократної точності, генерація ПВП, зручні способи роботи з СУБД, та широкі можливості створення і розробки клієнт-серверних додатків та графічних інтерфейсів. Реалізація проводилась у інтегрованому середовищі розробки Visual Studio 2013.

Одним з найбільших під час програмної реалізації є модуль реалізації математичного апарату рекурентних V_k –послідовностей, для якого виділено спеціальний клас його реалізації, що складається з таких параметрів та операцій:

– параметр Module, який являє собою модуль, за яким будуть виконуватись усі необхідні операції під час розрахунків; він задається один раз при створенні нового об’єкту послідовності;

– параметр K, що визначає порядок послідовності; цей параметр є публічний на читання і задається лише при створенні нового об’єкту послідовності;

– масив елементів G з K великих чисел, які є базовими для генерації V_k – послідовностей; ці елементи можна задати або при створенні нового об’єкту послідовності, або викликати їх генерацію на основі модуля P;

– параметр ModInverse, який являє собою результат інверсії за модулем P нульового елементу масиву G; цей параметр доступний тільки для читання і вираховується одразу після отримання елементів масиву G;

– параметр vkList, який являє собою сортований список пар типу ключ-значення, де ключем виступає номер елементу в послідовності, а значенням відповідний до цього номеру елемент послідовності; цей параметр не доступний ззовні об’єкту і використовується як кеш вже підрахованих елементів послідовності, що дозволяє збільшити швидкість при виконанні операцій над елементами послідовностей;

– конструктор об’єкту, в якому виконується задавання початкових параметрів, необхідних для генерації послідовності;

– метод InitParams, який виконує ініціалізацію, задавання початкових значень нульовим елементам послідовності та попередній обрахунок мінімального набору елементів, що необхідні для подальших розрахунків та елементів послідовності, які необхідні для бінарного методу обрахунку n-го елементу послідовності;

– метод CalcVk використовується для обрахунку n-го елементу послідовності за прямими формулами обчислення елементів V_k –послідовності;

– метод CalcVkReverse використовується для обрахунку n-го елемента послідовності за зворотними формулами обчислення елементів V_k –послідовності;

– метод CalcSet обраховує мінімальний набір елементів, що необхідні для роботи криптографічних алгоритмів;

– метод CalcQuickSetPlus обраховує мінімальний набір елементів, необхідний для операцій над додатними елементами послідовності;

– метод VeryQuickCalcPlus, який використовує бінарний метод обрахунку n-го елементу додатної частини послідовності;

– оператор індексації, який дозволяє працювати з об'єктом послідовності неначе зі звичайним масивом, автоматично повертаючи необхідний елемент, або обраховуючи його за допомогою методу CalcQuickSetPlus;

– метод CalcNPlusM використовується для обрахунку елементу V_k –послідовності з індексом n+m;

– метод QuickCalcPlus – для обрахунку елементу V_k –послідовності з індексом n*m;

– метод ToSerizable для зберігання інформації про рекурентну послідовність у зовнішні джерела.

Для кожної з вищеперерахованих операцій для класу Vk розроблено модульні тести, що перевіряють правильність реалізованих операцій. Загалом тести містять у собі порівняння елементів послідовності отримані у результаті використання певної операції та у результаті використання прямої формули.

Розглянемо основні особливості щодо реалізації модулів виконання криптографічних операцій та відповідних їм модульних тестів, які будуть підтверджувати правильність реалізації алгоритмів.

В основу усіх криптографічних алгоритмів покладено власно розроблений програмний інтерфейс IProtocol, який зобов'язує методи реалізовувати певні параметри та методи, а саме:

– параметр Side власного типу Sides, який може приймати тільки значення A, B, TrustedCenter, що відповідають відповідним сторонам криптографічного процесу;

– параметр TrustedCenter типу поточного об'єкта, який відповідає за довірчий центр; для об'єкта, який, власне, є довірчим центром, це значення записується як null;

– метод Initialize, у якому виконуються усі необхідні підготовчі операції без самих криптографічних процесів для поточної сторони криптографічного процесу.

Модуль реалізації алгоритмів асиметричного шифрування складається з трьох основних класів: класу реалізації асиметричного шифрування за методом Ель-Гамаля, класу реалізації асиметричного шифрування за запропонованим методом на основі V_k – послідовностей та класу реалізації асиметричного шифрування на основі U_k – послідовностей. У цьому модулі реалізація не зобов'язувала реалізовувати інтерфейс IProtocol, хоча у загальному класи повністю підлягають його вимогам.

У реалізації методу Ель-Гамаля маємо такі параметри для програмних методів у частині, що виступає у якості довірчого серверу:

– параметр P – модуль, за яким будуть виконуватись обрахунки;

– параметр G – велике випадкове число, яке не більше модуля P;

– метод Initialize, який виконує ініціалізацію перерахованих вище параметрів.

Сторона A використовує такі параметри і методи:

– параметр X та K, де перше – це випадкове число, яке менше за P–1, а друге – це спряжене до модуля число;

– метод Initialize, який виконує ініціалізацію перерахованих вище параметрів;

– метод Encrypt, який виконує по-блокове шифрування переданої йому інформації на основі відкритих параметрів відправника, у результаті якого отримується кортеж з 2-х елементів: великого цілого числа (відкритий ключ) та самого блоку зашифрованого повідомлення.

Одержувач не має ніяких особистих параметрів, а має лише метод Decrypt, який, отримуючи на вхід створений вище кортеж, виконує дешифрування інформації, повертаючи дешифроване повідомлення.

Модульний тест цього алгоритму просто генерує велике повідомлення, створює об'єкти довірчого центру та відправника і одержувача, виконує початкову ініціалізацію цих об'єктів, шифрує повідомлення одержувачем при відомих відкритих параметрах відправника, потім дешифрує отриманий шифр відправником за допомогою своїх секретних параметрів та перевіряє рівність початкового і дешифрованого повідомлення.

Реалізація алгоритмів методів асиметричного шифрування на основі V_k та U_k – послідовностей [4, 6] є схожою, тому опишемо їх разом. Кожна сторона генерує велике просте число A та B, яке зберігається у відповідних параметрах кожного модуля. Також є параметри, у яких зберігаються послідовності _vka та _vkb відповідно для кожної сторони. Вираховується A-й та B-й елемент послідовності. Процес шифрування складається з того, що формується відкритий ключ внаслідок отримання елементу з індексом A+B, і послідовність додається за модулем 2, внаслідок чого отримується шифртекст. Процес дешифрування виглядає ідентично, тільки виконується додавання елементів послідовності B та A, після чого шифротекст додається за модулем 2 з отриманим ключем. Модульні тести для методів на основі V_k та U_k –послідовностей аналогічні.

Висновки

Таким чином здійснено розробку програм реалізації представленого у [6] методу асиметричного шифрування інформації на основі V_k –послідовностей. Розроблено алгоритми програмної реалізації процедур шифрування та дешифрування інформації для цього методу. Розглянуто основні особливості щодо реалізації модулів виконання криптографічних операцій та відповідних їм модульних тестів. Найбільш докладно розглянуто програмний модуль реалізації математичного апарату рекурентних V_k – послідовностей, для якого виділено спеціальний клас його реалізації, описано параметри та операції цього класу.

Література

1. Menezes, A.J. Handbook of Applied Cryptography [Текст] / A.J. Menezes, P.C. van Oorschot, S.A. Vanstone. – CRC Press, 2001. – 816 p.

2. Rivest, R.L. A method for obtaining digital signatures and public-key cryptosystems [Текст] / R.L. Rivest, A. Shamir, and L.M. Adleman // Communications of the ACM. – 1978. – Volume 21, Issue 2. – P. 120–126.

3. ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms [Текст] / T. ElGamal // IEEE Intern. Symp. Informat. Theory. – 1985. – V.IT–31. №4. – P. 469–472.

4. Яремчук, Ю.Є. Метод асиметричного шифрування інформації на основі рекурентних послідовностей [Текст] / Ю.Є. Яремчук // Сучасна спеціальна техніка. – №4, 2012. – С. 79–87.

5. Яремчук, Ю.Є. Особливості розроблення програмних засобів реалізації протоколів асиметричного шифрування інформації на основі рекурентних послідовностей [Текст] / Ю.Є. Яремчук // Вісник Тернопільського національного технічного університету. – №1, 2013. – С. 174–182.

6. Яремчук, Ю.Є. Метод шифрування інформації з відкритим ключем на основі рекурентних послідовностей [Текст] / Ю.Є. Яремчук // Інформаційна безпека. – №3, 2013. – С. 123–129.

Надійшла 13.11.2014 р. Рецензент: д.т.н., проф. Бурячок В.Л.