УДК 681.3.067

Ю. Є. Яремчук
Вінницький національний технічний університет
вул. Хмельницьке шосе, 95, 21021 Вінниця, Україна

Розробка алгоритмів прискореного обчислення елементів рекурентних послідовностей для криптографічних застосувань

Представлено алгоритми прискореного обчислення елементів рекурентної V_k*-послідовності для додатних і від'ємних значень індексу n цієї послідовності. Для кожних із цих значень розглянуто по два можливих варіанти алгоритмів — на основі бінарного методу та на основі методу з розкладанням індексу елемента послідовності. Отримано оцінки складності представлених алгоритмів, які показали, що складність обчислення елемента* V_k*-послідовності за модулем є приблизно на тому ж рівні як і відповідне піднесення до степеня, що забезпечує можливість ефективного використання рекурентних* V_k^-*- та* U_k*-послідовностей для різних криптографічних застосувань.*

Ключові слова: *рекурентні послідовності, інформація, захист інформації, криптографія.*

Вступ

На сьогодні криптографічні методи [1, 2] застосовуються в системах захисту і додатках різного призначення. При цьому актуальною є задача спрощення обчислень криптографічних методів, особливо тих, що базуються на технології відкритого ключа, які використовують великі ключі та числа великої розрядності. Виходячи з цього, актуальним є побудова цих криптографічних методів на основі таких математичних апаратів, які б могли забезпечувати спрощення обчислень.

З цієї точки зору певний інтерес викликає апарат на основі рекурентних послідовностей [3], який дозволяє за певних умов спрощувати обчислення криптографічних методів, що базуються на його основі. Так з метою спрощення обчислень у роботі [4] запропоновано використовувати рекурентні послідовності Люка за модулем простого числа замість традиційного піднесення до степеня. Однак, у роботі [5] було вказано на певну слабкість такого підходу щодо криптографічної стійкості.

Рекурентні послідовності в загальному вигляді породжуються таким співвідношенням [3]

$$u_n = a_1 \cdot u_{n-1} + a_2 \cdot u_{n-2} + \ldots + a_k \cdot u_{n-k},$$

де a_1, a_2, ..., a_k — коефіцієнти; k — порядок послідовності, виходячи з початкових елементів u_0, u_1, ..., u_k.

У роботі [6] розглянуто рекурентну V_k-послідовність, в якій коефіцієнти утворюючого рекурентного співвідношення пов'язані зі значеннями початкових елементів. Ця послідовність складається з двох видів послідовності V_k^+ та V_k^-.

V_k^+-послідовністю називається послідовність чисел, що обчислюються за формулою

$$v_{n,k} = g_k v_{n-1,k} + g_1 v_{n-k,k} \tag{1}$$

для початкових значень $v_{0,k} = 1$, $v_{1,k} = g_2$ для $k = 2$; $v_{0,k} = v_{1,k}$ = ... = $v_{k-3,k} = 0$, $v_{k-2,k} = 1$, $v_{k-1,k} = g_k$ для $k > 2$, де g_1, g_k — цілі числа; n і k — цілі додатні.

V_k^--послідовністю називається послідовність чисел, що обчислюються за формулою

$$v_{n,k} = \frac{v_{n+k,k} - g_k \cdot v_{n+k-1,k}}{g_1} \tag{2}$$

для n-від'ємних при початкових значеннях $v_{-1,k} = 0$, $v_{-2,k} = g_1^{-1}$ для $k = 2$; $v_{-1,k} = 0$, $v_{-2,k} = g_1^{-1}$, $v_{-3,k} = v_{-4,k} = ... = v_{-k,k} = 0$ для $k > 2$.

Формула (1) дозволяє отримувати значення для зростаючих n, починаючи з $n = 0$. За формулою (2) здійснюється зворотна процедура, коли елементи послідовності обчислюються для спадних n, починаючи з деякого значення $n = l$. Причому обчислення за формулою (2) може продовжуватись і для $n < 0$, формуючи таким чином елементи V_k^--послідовності.

У роботах [6, 7] показано можливість використання рекурентних V_k^+- та U_k-послідовностей для побудови криптографічних методів, що базуються на технології відкритого ключа, зокрема методів розподілу ключів та автентифікації. Запропонований підхід дозволяє спрощувати обчислення криптографічних перетворень порівняно з відомими аналогами. При цьому відбувається заміна модулярного піднесення до степеня обчисленням за модулем елемента U_k-послідовності з певним індексом. Однак, запропоновані методи забезпечують спрощення обчислень відносно відомих методів лише в разі, якщо складність обчислення елемента рекурентної V_k-послідовності буде принаймні не меншою, ніж піднесення до степеня.

Проблема обчислення елемента $v_{n,k}$ полягає в тому, що для великих значень n, а саме такі значення повинні використовуватись у криптографічних застосуваннях, безпосереднє обчислення $v_{n,k}$ за формулою (1) для додатних значень n або за формулою (2) для від'ємних значень n є неприйнятним, оскільки є дуже повільним з-за послідовного характеру його обчислення за цими формулами. Потрібен більш швидкий метод обчислення елемента $v_{n,k}$, який би міг використовувати санкціонований користувач криптографічної системи.

У даній роботі пропонуються алгоритми прискореного обчислення елементів V_k-послідовності, які дозволять значно прискорювати криптографічні перетворення в криптосистемах.

Розробка алгоритмів прискореного обчислення елементів V_k -послідовності для додатних n

Для будь-яких цілих додатних n, m та k отримано таку аналітичну залежність [6]:

$$v_{n+m,k} = v_{m+(k-2),k} \cdot v_{n,k} + g_1 \cdot \sum_{i=1}^{k-1} v_{m+(k-2)-i,k} \cdot v_{n-k+i,k} \,. \tag{3}$$

В окремому випадку, коли $m = n$ залежність (3) буде мати такий вигляд:

$$v_{2n,k} = v_{n+(k-2),k} \cdot v_{n,k} + g_1 \cdot \sum_{i=1}^{k-1} v_{n+(k-2)-i,k} \cdot v_{n-k+i,k} \,. \tag{4}$$

Далі пропонується спосіб прискореного обчислення елементів $v_{n,k}$ для додатних n, який базується на тій же ідеї, що і бінарний метод [8] піднесення до степеня. Скористаємося даним методом для отримання адитивного ланцюжка

$$1 = c_0,\ c_1,\ c_2, \ldots,\ c_t = n\,.$$

Якщо записати n у двійковій системі числення як

$$n = \sum_{i=0}^{t} \alpha_{t-i} 2^{t-i}\,, \tag{5}$$

то для кожного $i = \overline{1,t}$ правило отримання адитивного ланцюжка, починаючи з c_1, буде таким:

— якщо значення α_{t-i} дорівнює 0, то $c_i = 2c_{i-1}$;

— якщо значення розряду α_{t-i} дорівнює 1, то $c_i = 2c_{i-1} + 1$.

Як наслідок, дійшовши до крайнього правого розряду n, отримаємо $c_t = n$.

Звідси, обчислення $v_{n,k}$ буде зводитися до послідовного обчислення $v_{c_i,k} = v_{2c_{i-1}+1,k}$ або $v_{c_i,k} = v_{2c_{i-1},k}$.

Обчислення $v_{c_i,k} = v_{2c_{i-1},k}$ будемо здійснювати згідно аналітичної залежності (4), а $v_{c_i,k} = v_{2c_{i-1}+1,k}$ будемо отримувати, обчислюючи спочатку $v_{2c_{i-1},k}$, а потім $v_{2c_{i-1}+1,k}$ за формулою (1).

З (4) видно, що для отримання елемента $v_{2n,k}$ використовуються елементи $v_{n+k-2,k},\ldots,\ v_{n-(k-2),k}$, $v_{n-(k-1),k}$. Тобто на кожному кроці необхідно визначати та зберігати набір з $2k-2$ елементів. Розглянемо обчислення цих елементів.

Елементи $v_{2n,k}$, $v_{2n-1,k},\ldots,\ v_{2n-(k-3),k}$, $v_{2n-(k-2),k}$ можуть бути обчислені згідно залежності (3) відповідно як $v_{n+n,k}$, $v_{n+(n-1),k},\ldots,\ v_{n+(n-(k-3)),k}$, $v_{n+(n-(k-2)),k}$.

Елемент $v_{2n-(k-1),k}$ не може бути обчислений згідно залежності (3), оскільки для його обчислення, окрім елементів, які є в наведеному вище наборі, потрібен елемент $v_{n-k,k}$. Розширення цього набору елементів елементом $v_{n-k,k}$ не бажано, тому що для обчислення $v_{2n-k,k}$ буде потрібен елемент $v_{2n-(k+1),k}$. Щоб усунути цей недолік будемо обчислювати елемент $v_{2n-(k-1),k}$ за формулою (2).

У такому випадку необхідним є елемент $v_{2n+1,k}$. Цей елемент може бути обчислений згідно залежності (3). При цьому набір необхідних елементів буде розширений елементом $v_{n+k-1,k}$.

Елементи $v_{2n+k-1,k},\ldots,\ v_{2n+3,k}$, $v_{2n+2,k}$ можуть бути отримані на основі вже обчислених елементів $v_{2n+1,k}$, $v_{2n,k},\ldots,\ v_{2n-(k-3),k}$, $v_{2n-(k-2),k}$ за формулою (1).

Таким чином, для обчислення елемента $v_{2n,k}$ на кожному кроці необхідно визначати та зберігати набір з $2k-1$ елементів.

Слід також відзначити, що в прискореному алгоритмі, який розробляється, індекс n елемента V_k-послідовністі буде приймати великі значення, тому доцільно одразу всі операції виконувати за модулем, тим більше, що і в криптографічних застосуваннях, зокрема в методах на основі технології відкритого ключа, обчислення виконуються над числами великої розрядності.

Позначивши l як поточне значення індексу елемента V_k-послідовності, маємо такий алгоритм прискореного обчислення елементів цієї послідовності для додатних n.

Алгоритм 1.

П. 1. Провести початкову ініціалізацію: $i \leftarrow t$; $l \leftarrow 1$; присвоїти елементам $v_{l+k-1,k},\ldots,\ v_{l-(k-2),k}$, $v_{l-(k-1),k}$ відповідні значення V_k-послідовності.

П. 2. $i \leftarrow i-1$.

П. 3. $l \leftarrow 2l$.

П. 4. Обчислити нові значення $v_{l+1,k}$, $v_{l,k},\ldots,\ v_{l-(k-3),k}$, $v_{l-(k-2),k}$ за модулем p, використовуючи (3).

П. 5. Обчислити елемент $v_{l-(k-1),k}$ за модулем p, використовуючи (2).

П. 6. Якщо $k > 2$, то обчислити елементи $v_{l+k-1,k}$, $v_{l+k-2,k},\ldots,\ v_{l+3,k}$, $v_{l+2,k}$ за модулем p, використовуючи (1).

П. 7. Якщо $\alpha_i = 0$, то перейти до п. 10.

П. 8. $l \leftarrow l+1$.

П. 9. Обчислити нові значення $v_{l+k-1,k},\ldots,\ v_{l-(k-2),k},\ v_{l-(k-1),k}$ шляхом присвоювання кожному попередньому елементу значення наступного за ним елемента та обчислення за модулем p останнього елемента $v_{l+k-1,k}$ за формулою (1), використовуючи тільки-но обчислені елементи.

П. 10. Якщо $i-1\neq 0$, то перейти до п. 3, інакше завершити роботу алгоритму.

Якщо порівнювати складність обчислень за послідовною формулою (1), починаючи з елементів $v_{k-1,k},\ldots,\ v_{1,k},\ v_{0,k}$, то потрібно виконати $(k+3)n$ операцій. При цьому за алгоритмом 1 потрібно виконати приблизно $\log_2 n(2k^2+6k-1)$ операцій. Це означає, що наприклад, при $k=2$ та $n=2^{256}$ за формулою (1) потрібно виконати приблизно 2^{514} операцій, у той час, як за алгоритмом 1 потрібно виконати всього 2^{11} операцій.

Як вже зазначалось, у криптографічних застосуваннях важливим є пошук будь-яких можливостей зменшення обчислювальної складності криптографічних перетворень.

У цьому зв'язку, звертаємо увагу на відомий підхід до реалізації алгоритмів, коли кількість операцій, що виконуються, зменшується за рахунок збільшення кількості проміжних результатів, котрі зберігаються в процесі обчислень. Виходячи з цього, пропонується ще один алгоритм, який реалізує такий підхід.

Запишемо n, як і в попередньому способі, у вигляді (5) і для кожного $h=\overline{1,r}$, де $r=\sum_{i=0}^{t}\alpha_{t-i}$, сформуємо послідовність чисел x_h за таким правилом: якщо значення α_{t-i} дорівнює 1, то $x_h=x_{h-1}+2^{t-i}$.

Тоді обчислення $v_{n,k}$ зводиться до послідовного обчислення $v_{x_h,k}=v_{x_{h-1}+2^{t-i},k}$, яке будемо здійснювати згідно залежності (3).

Для обчислення елемента $v_{n+m,k}$ згідно залежності (3) потрібні елементи $v_{n,k}$, $v_{n-1,k},\ldots,\ v_{n-(k-2),k},\ v_{n-(k-1),k}$ та елементи $v_{m+k-2,k},\ v_{m+k-3,k},\ldots,\ v_{m,k},\ v_{m-1,k}$.

На кожному кроці обчислень $v_{x_h,k}$, елементи $v_{n+m-1,k},\ldots,\ v_{n+m-(k-2),k},\ v_{n+m-(k-1),k}$, як і елемент $v_{n+m,k}$, будемо обчислювати згідно залежності (3). При цьому набір необхідних елементів $v_{m+k-2,k},\ v_{m+k-3,k},\ldots,\ v_{m,k},\ v_{m-1,k}$ буде розширений елементами $v_{m-2,k},\ldots,\ v_{m-(k-1),k},\ v_{m-k,k}$.

Оскільки в нашому випадку $m=2^{t-i}$, то пропонується набір елементів $v_{2^{t-i}+j,k}$, $i=\overline{0,t}$, $j=\overline{-k,k-2}$, обчислити попередньо за алгоритмом 1 і зберігати в пам'яті для використання при безпосередній роботі алгоритму.

Таким чином, на кожному кроці обчислень $v_{x_h,k}$ необхідно визначати та зберігати набір з k елементів.

Враховуючи вищесказане, маємо наступний алгоритм прискореного обчислення елемента $v_{n,k}$ для додатних n.

Алгоритм 2.

П. 1. Обчислити елементи $v_{2^{t-i}+j,k}$, $i=\overline{0,t}$, $j=\overline{-k,k-2}$, за модулем p, використовуючи алгоритм 1 для $j=\overline{-(k-1),k-2}$, а потім формулу (2) для $j=-k$.

П. 2. Провести початкову ініціалізацію: $i \leftarrow t$, $l \leftarrow 0$; присвоїти елементам $v_{l+j,k}$, $j=\overline{-(k-1),0}$, відповідні значення V_k-послідовності.

П. 3. Якщо $\alpha_i = 0$, то перейти до п. 6.

П. 4. Обчислити $v_{l+2^i+j,k}$, $j=\overline{-(k-1),0}$, за модулем p, використовуючи (3).

П. 5. $l \leftarrow l+2^i$.

П. 6. Якщо $i \neq 0$, то $i \leftarrow i-1$ та перейти до п. 3, інакше завершити роботу алгоритму.

Зазначимо, що п. 1 алгоритму виконується попередньо перед безпосереднім обчисленням елементів $v_{n,k}$.

Аналіз алгоритмів 1 та 2 показує, що останній алгоритм порівняно з першим є менш складним за кількістю арифметичних операцій, оскільки потребує на кожному кроці обчислення лише k елементів згідно залежності (3). Однак, перший алгоритм порівняно з другим не потребує попередньої процедури обчислень елементів V_k -послідовності та додаткового використання пам'яті для зберігання цих елементів у процесі роботи алгоритму.

Розробка алгоритмів прискореного обчислення елементів V_k -послідовності для від'ємних *n*

Для будь-яких цілих додатних n і m, таких що $1 \le m < n$, та будь-якого цілого додатного k отримано таку аналітичну залежність [6]:

$$v_{n-m,k} = v_{-m+(k-2),k} \cdot v_{n,k} + g_1 \cdot \sum_{i=1}^{k-1} v_{-m+(k-2)-i,k} \cdot v_{n-k+i,k} \,. \qquad (6)$$

З наведених формул та отриманих залежностей для V_k-послідовності видно, що обчислення елементів $v_{n,k}$ для від'ємних значень n безпосередньо за формулою (2) є неприйнятним для криптографічних застосувань з тієї ж причини, що і для додатних n, коли вони приймають великі значення. Тому розглянемо алгоритм прискореного обчислення елемента $v_{n,k}$ для від'ємних n. Розробку алгоритму проведено аналогічно, як і для додатних n, включаючи використання бінарного методу.

Для того, щоб обчислити за бінарним методом елемент $v_{n,k}$ необхідно, в першу чергу, мати формулу для обчислення елемента $v_{2n,k}$. Якщо для будь-якого додатного r прийняти $n=-r$, $m=r$, то з аналітичної залежності (6) отримаємо таку залежність для обчислення елемента $v_{-2r,k}$:

$$v_{-2r,k} = v_{-r+(k-2),k} \cdot v_{-r,k} + g_1 \cdot \sum_{i=1}^{k-1} v_{-r+(k-2)-i,k} \cdot v_{-r-k+i,k} \ . \qquad (7)$$

Аналіз аналітичної залежності (7) показує, що для отримання елемента $v_{-2r,k}$ використовуються елементи $v_{-r+k-2,k}, \ldots, v_{-r-(k-2),k}$, $v_{-r-(k-1),k}$. Обчислення елемента $v_{n,k}$ для великих від'ємних значень n потребує багатократного використання залежності (7). Тому на кожному кроці обчислень елемента $v_{n,k}$, як мінімум потрібно мати всі елементи, що використовуються в залежності (7).

Елементи $v_{-2r+k-2,k}, \ldots, v_{-2r,k}$, $v_{-2r-1,k}$ обчислимо згідно залежності (6) відповідно як $v_{(-r+k-2)-r,k}, \ldots, v_{-r-r,k}$, $v_{(-r-1)-r,k}$. При цьому необхідно мати елементи $v_{-r+k-2,k}, \ldots, v_{-r-(k-1),k}$, $v_{-r-k,k}$, з яких визначеними вже, з точки зору обчислень, є лише елементи $v_{-r+k-2,k}, \ldots, v_{-r,k}$, $v_{-r-1,k}$. Обчислення ж елементів $v_{-2r-2,k}, \ldots, v_{-2r-(k-1),k}$, $v_{-2r-k,k}$ здійснимо за формулою (2), використовуючи вже обчислені елементи $v_{-2r+k-2,k}, \ldots, v_{-2r,k}$, $v_{-2r-1,k}$.

Таким чином, визначено всі елементи, які потрібні для обчислення елемента $v_{n,k}$ для від'ємних значень n за бінарним методом. Це елементи $v_{-r+k-2,k}, \ldots, v_{-r-(k-1),k}$, $v_{-r-k,k}$.

Використовуючи l в тому ж значенні, що і в алгоритмі 1, маємо такий алгоритм прискореного обчислення елемента $v_{n,k}$ для від'ємних n за модулем p.

Алгоритм 3.

П. 1. Провести початкову ініціалізацію: $i \leftarrow t$, $l \leftarrow 1$; присвоїти елементам $v_{-l+k-2,k}, \ldots, v_{-l-(k-1),k}$, $v_{-l-k,k}$ відповідні значення V_k-послідовності.

П. 2. $i \leftarrow i-1$.

П. 3. $l \leftarrow 2l$.

П. 4. Обчислити нові значення елементів $v_{-l+k-2,k}, \ldots, v_{-l,k}$, $v_{-l-1,k}$ за модулем p, використовуючи (6).

П. 5. Обчислити елементи $v_{-l-2,k}, \ldots, v_{-l-(k-1),k}$, $v_{-l-k,k}$ за модулем p, використовуючи (2).

П. 6. Якщо $\alpha_i = 0$, то перейти до п. 9.

П. 7. $l \leftarrow l+1$.

П. 8. Обчислити нові значення $v_{-l+k-2,k}, \ldots, v_{-l-(k-1),k}$, $v_{-l-k,k}$ шляхом присвоювання кожному елементу значення попереднього елемента та обчислення за модулем p першого з цього набору елемента $v_{-l-k,k}$ за формулою (2), використовуючи тільки-но обчислені елементи.

П. 9. Якщо $i-1 \neq 0$, то перейти до п. 3, інакше завершити роботу алгоритму.

За послідовною формулою (2), починаючи з елементів $v_{0,k}, \ldots, v_{-k+2,k}$, $v_{-k+1,k}$, потрібно виконати $(k+3)n$ операцій. За алгоритмом 3 потрібно виконати приблизно $\log_2 n(2k^2+6k-1)$ операцій. Тобто, як і алгоритм 1, так і алгоритм 3 забезпечують однакову кількість операцій для обчислення елементів $v_{n,k}$.

Далі представимо ще один алгоритм прискореного обчислення елемента $v_{n,k}$ для від'ємних значень n, який розроблено за аналогією з алгоритмом 2. Відмінність полягає в тому, що на кожному кроці обчислення набору елементів проводяться за формулою (6), замість (3), використовуючи при цьому елементи $v_{-2^{t-i}+j,k}$, $i=\overline{0,t}$, $j=\overline{-k,k-2}$, які обчислюються попередньо за алгоритмом 3.

Алгоритм 4.

П. 1. Обчислити елементи $v_{-2^{t-i}+j,k}$, $i=\overline{0,t}$, $j=\overline{-k,k-2}$, за модулем p, використовуючи алгоритм 3.

П. 2. Провести початкову ініціалізацію: $i \leftarrow t$, $l \leftarrow 0$; присвоїти елементам $v_{l+j,k}$, $j=\overline{-(k-1),0}$, відповідні значення V_k-послідовності.

П. 3. Якщо $\alpha_i = 0$, то перейти до п. 6.

П. 4. Обчислити $v_{l-2^i+j,k}$, $j=\overline{-(k-1),0}$, за модулем p, використовуючи (6).

П. 5. $l \leftarrow l-2^i$.

П. 6. Якщо $i \neq 0$, то $i \leftarrow i-1$ та перейти до п. 3, інакше завершити роботу алгоритму.

Алгоритм 4 порівняно з алгоритмом 3 є менш складним, оскільки потребує на кожному кроці обчислення лише k елементів згідно залежності (6). Однак, алгоритм 3 порівняно з алгоритмом 4 не потребує попередньої процедури обчислень елементів V_k-послідовності та додаткового використання пам'яті для зберігання цих елементів у процесі роботи алгоритму.

Проведено оцінювання складності розроблених алгоритмів прискореного обчислення елементів V_k-послідовності. Визначено, що складність обчислень за алгоритмами прискореного обчислення елементів $v_{n,k}$ для додатних і для від'ємних значень n на рівні машинних одиниць інформації будуть однаковими, причому як для випадку використання бінарного методу, так і для випадку методу з використанням розкладання індексу. Визначено, що максимальна оцінка складності кожного з них не буде перевищувати $H^2 q \cdot [6H(k^2+3k)+9(k^2+2k)]$ операцій над машинними одиницями інформації, де H — кількість машинних одиниць інформації для зберігання великого числа, q — кількість розрядів машинної одиниці інформації. Аналіз отриманих оцінок складності обчислень показав, що обчислення за модулем певного елемента V_k-послідовності має приблизно той же порядок, що і складність модулярного піднесення до заданого степеня.

Висновки

Запропоновано по два варіанти алгоритмів прискореного обчислення елементів V_k-послідовності окремо для додатних та окремо для від'ємних значень індексу n елемента цієї послідовності. Перший варіант базується на основі відомого бінарного методу, а другий — на основі методу з використанням розкладання індексу. Показано, що алгоритм на основі методу з розкладанням індексу є менш складним, ніж алгоритм на основі бінарного методу, але останній не потребує попередніх обчислень елементів V_k-послідовності та додаткової пам'яті для зберігання цих елементів у процесі роботи алгоритму.

Проведено оцінювання обчислювальної складності усіх запропонованих алгоритмів прискореного обчислення елементів V_k-послідовності. Визначено, що складність виконання алгоритмів на основі бінарного методу для додатних і від'ємних значень n є однаковою, так само і складність виконання алгоритмів на основі методу з розкладанням індексу як для додатних, так і для від'ємних значень індексу n також є однаковою.

Отримано оцінки складності алгоритмів, які показали, що в цілому алгоритми потребують приблизно $O(\log_2 n)$ операцій, у той час як безпосереднє обчислення елементів V_k-послідовності вимагає $O(n)$ операцій.

Це означає, що завдяки розробленим алгоритмам вдалося забезпечити складність обчислення елемента рекурентної V_k-послідовності за модулем приблизно на тому ж рівні, що і складність модулярного піднесення до степеня. А це, в свою чергу, дозволяє ефективно використовувати математичний апарат на основі рекурентних V_k- та U_k-послідовностей для різних криптографічних застосувань.

1. *Menezes A.J.* Handbook of Applied Cryptography / A.J. Menezes, P.C. van Oorschot, S.A. Vanstone. — CRC Press, 2001. — 816 p.

2. *Шнайер Б.* Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. — М.: Триумф, 2002. — 816 с.

3. *Маркушевич А.И.* Возвратные последовательности / А.И. Маркушевич. — М.: Наука, 1975. — 48 с.

4. *Smith P.* A Public-Key Cryptosystem and a Digital Signature System Based on the Lucas Function Analogue to Discrete Logarithms / P. Smith, C. Skinner // In Advances in Cryptology Asiacrypt'94, Springer-Verlag. — 1995. — P. 357–364.

5. *Bleichenbacher D.* Some Remarks on Lucas-Based Cryptosystems / D. Bleichenbacher, W. Bosma, A. Lenstra // In Advances in Cryptology Crypto'95, Springer-Verlag. — 1995. — P. 386–396.

6. *Яремчук Ю.Є.* Використання рекурентних послідовностей для побудови криптографічних методів з відкритим ключем / Ю.Є. Яремчук // Захист інформації. — 2012. — № 4. — С. 120–127.

7. *Яремчук Ю.Є.* Метод автентифікації сторін взаємодії на основі рекурентних послідовностей / Ю.Є. Яремчук // Сучасний захист інформації. — 2013. — № 1. — С. 4–10.

8. *Кнут Д.* Искусство программирования для ЭВМ / Д. Кнут. — М.: Вильямс. — 2004. — Т. 2. Получисленные алгоритмы. — 832 с.

Надійшла до редакції 05.02.2013