неправильно розпізнаних бітів, а при масштабуванні зображення на 3 % і 100 % помилка розпізнавання становить 0 % та 55 % бітів відповідно.

Аналіз стійкості запропонованого методу до атаки шляхом внесення додаткового шуму показав, що при внесенні шуму такого рівня, при якому сумарна похибка відхилень координат є більшою від початкової у 5,7 разів, що призводить до повної деградації зображення, помилка розпізнавання становить усього 33,46 % неправильно розпізнаних бітів.

Проведено оцінювання стійкості запропонованого методу до пасивних атак, спрямованих на визначення місця розташування ЦВЗ. Результати оцінювання показали забезпечення достатнього рівня стійкості, наприклад, для типової векторної географічної карти, яка складається з 64 тис. точок розміром близько 1,5 Мб, кількість комбінацій розміщення бітів ЦВЗ становить приблизно 2^{1010}.

Література: ***1.*** *Zheng L. Research on Vector Map Digital Watermarking Technology / L. Zheng, Y. Jia, Q. Wang. // First International Workshop on Education Technology and Computer Science – 2009. – P. 303 – 307.* ***2.*** *Карпінець В. В. Зменшення відхилень координат точок внаслідок вбудовування цифрових водяних знаків у векторні зображення / В. В. Карпінець, Ю. Є. Яремчук // Правове, нормативне, та метрологічне забезпечення системи захисту інформації в Україні – 2010. – № 2(21). – С.101 – 109.* ***3.*** *Карпінець В. В. Аналіз впливу цифрових водяних знаків на якість векторних зображень / В. В. Карпінець, Ю. Є. Яремчук // Сучасний захист інформації. – 2011. – №1. – С.72 – 82 .* ***4.*** *Карпінець В. В. Дослідження стеганографічної стійкості методу вбудовування цифрових водяних знаків у векторні зображення / В. В. Карпінець, Ю. Є. Яремчук // ВІСНИК Вінницького політехнічного інституту. – 2011. - №3. – С. 200-205.* ***5.*** *Карпінець В. В. Аналіз стійкості методу вбудовування цифрових водяних знаків у векторні зображення до зловмисних атак / В. В. Карпінець, Ю. Є. Яремчук // ВІСНИК Вінницького політехнічного інституту. – 2011. - №4. – С. 154-159.* ***6.*** *Холл М. Комбинаторика / М. Холл // Издательство «МИР». Москва. — 1970. 424 с.*

УДК 512

РЕКУРЕНТНІ АЛГОРИТМИ ОБЧИСЛЕННЯ КОРЕНЯ ДОВІЛЬНОГО СТЕПЕНЮ У КІЛЬЦІ ЛИШКІВ

Людмила Ковальчук, Олексій Беспалов, Павло Огнєв
Физико-технічний інститут НТУУ "КПІ"

Аннотация: **Побудовано поліноміальні рекурентні алгоритми добування кореню довільного степеню у кільці лишків.**
Annotation: **Polynomial algorithms are constructed for obtaining the roots of arbitrary powers in a residue ring.**
Ключові слова: **Кільця лишків, добування кореня, еліптичні криві.**

I Вступ

У даній роботі будуть побудовані алгоритми добування кореня кубічного та коренів вищих степенів у кільцях лишків.

Питання розв'язку степеневих рівнянь у кільцях лишків є цікавим як з точки зору власне теорії чисел, так і з точки зору її застосувань у криптології. В першу чергу у криптології застосовуються алгоритми розпізнавання квадратичності та добування квадратного кореня як за простим модулем, так і за складеним, що є добутком двох простих чисел. Слід зазначити, що такі алгоритми є повністю описаними (наприклад, у [1, 2]). Вони поліноміальні (у окремих випадках – імовірнісні). Ці алгоритми мають застосування як при побудові криптосистем (наприклад, криптосистема Блюма), так і при криптоаналізі (поліноміальна еквівалентність задачі добування кореня за складеним модулем та задачі факторизації), а також при побудові криптографічно сильних генераторів (наприклад, BBS).

Побудова зручних поліноміальних алгоритмів добування кубічного кореня та кореня більш високого степеню також є цікавою задачею теорії чисел, хоча й не такою прикладною, як добування квадратного кореня. Проте деякі застосування все ж таки можна навести. По-перше, як і у випадку з квадратним коренем, задача добування кореня (будь-якого степеню) за складеним модулем поліноміально еквівалентна задачі факторизації, тобто може використовуватись у криптоаналізі алгоритмів, що базуються на складності задачі факторизації. По-друге, задача добування саме кубічного кореня може мати застосування при побудові криптосистем на еліптичних кривих, що задані над кільцями лишків (за складеним модулем). На таких еліптичних кривих можна побудувати як RSA-подібні криптосистеми, так і Ель-Гамале-подібні.

Криптосистеми на еліптичних кривих є невід'ємною і необхідною частиною сучасної криптології (див., наприклад, [1, 3]). Задачі, які виникають при побудові та аналізі таких криптосистем, є цікавими як з прикладної, так і з математичної точок зору. Зокрема, зараз можна помітити зростання популярності алгоритмів та протоколів з використанням еліптичних кривих, які визначені не над скінченими полями, а над кільцями лишків певного виду. Хоча такі криптосистеми не мають особливих переваг, вони активно досліджуються, про що свідчить, наприклад, значна кількість доповідей та дисертаційних (в більшості, російських) робіт [4 – 6], посилання на які можна знайти в Internet та які, на жаль, не завжди є повністю доступними. Однією з причин підвищення інтересу до таких криптосистем можна вважати цікаві математичні задачі, що виникають при їх дослідженні.

При побудові криптосистем на еліптичних кривих зазвичай використовують принаймні один з наступних імовірнісних алгоритмів: "вкладання" відкритого тексту у точку кривої та знаходження базової точки кривої. Обидва з них використовують алгоритм перевірки квадратичності та алгоритм розв'язування квадратного рівняння у скінченому полі. У випадку, коли крива задана рівнянням $y^2 = x^3 + a$ (а саме такі криві розглядають над кільцями лишків Z_n, $n = pq$), то зазначені алгоритми можна замінити детермінованими, при певних обмеженнях на кільце лишків, над яким задана крива. Але при цьому замість обчислення $y = y(x)$ за заданим x ми будемо обчислювати $x = x(y)$, тобто замість розв'язку квадратного рівняння потрібно буде обчислювати кубічний корінь (достатньо вміти його обчислювати за простим модулем, внаслідок теореми про ізоморфізм кілець лишків). Виходячи з вищесказаного, ми можемо сформулювати цікаву й актуальну задачу: дати критерій кубічності у простому скінченому полі та навести зручні поліноміальні алгоритми обчислення $\sqrt[3]{a} \bmod p$, де $a \in Z_p^*$ і є кубічним лишком. Не менш цікавим, принаймні з математичної точки зору, є й узагальнення цієї задачі на випадок кореня довільного степеню. Така задача буде повністю розв'язана у даній роботі.

II Огляд відомих алгоритмів розв'язку степеневих рівнянь у кільці лишків

Як було зазначено, задача добування кореня квадратного у скінчених простих полях та кільцях лишків певного виду повністю розв'язана у [1, 2]. У цьому розділі ми наведемо декілька основних алгоритмів, що розв'язують більш загальну задачу, та коротко їх проаналізуємо.

Задачі розв'язку степенивих рівнянь у кільцях лишків досить детально розглянуті у [7], де наведено два (основних) типи таких алгоритмів (взагалі кажучи, не поліноміальних).

Алгоритм 1 ([7], стор. 43-44, алгоритм 2.1), який розв'язує рівняння m-го степеню за простим модулем p, має складність

$$O\left(m^2 \log m \log^3 p\right).$$

Цей алгоритм є імовірнісним, і дана оцінка складності є оцінкою в середньому. Він використовує декілька звертань до алгоритму Евкліда для поліномів над Z_p, степінь яких не перевищує p, та багато операцій ділення таких поліномів. Хоча для малих значень m його можна вважати поліноміальним (у середньому), він є досить складним і громіздким. Крім того, негативним є той факт, що "імовірнісну частину" алгоритму не можна виділити у попередні обчислення, тому що вона суттєво залежить від виду рівняння.

Алгоритм 2 ([7], стор. 66, алгоритм 2.7) розв'язує рівняння виду $a^m \equiv b\left(\bmod\ p^\alpha\right)$, де m є дільником p-1, використовуючи імовірнісний Лас-Вегас алгоритм знаходження утворюючого елемента мультиплікативної групи простого скінченого поля (цей алгоритм можна виконати заздалегідь у попередніх обчисленнях), алгоритм Евкліда, багато піднесень до степеню (не більшого за $\varphi\left(p^\alpha\right)$) та множень за модулем p^α, а також алгоритм розв'язку задачі DLP у підгрупі мультиплікативної групи кільця лишків Z_{p^α}, порядок якої дорівнює m^k, де $\varphi\left(p^\alpha\right) = m^{k+1}h, \quad (m,h) = 1$. Хоча для невеликих значень m та k алгоритм можна вважати поліноміальним, він є досить громіздким, а у випадку, коли k=1 – дуже надлишковим, особливо порівняно з запропонованими нами у наступній частині алгоритмами.

Далі наведемо вказані алгоритми більш детально.

Алгоритм 1

Ймовірнісний алгоритм знаходження коренів многочлена $f(x)$ над полем Z_p

Дано: просте число p, многочлен $f(x) \in Z_p[x]$, $\deg f(x) = m$.

Результат: корінь a многочлена $f(x)$ або інформація, що многочлен $f(x)$ не має коренів в Z_p.

Алгоритм:

1. Обчислити $d(x) = (x^p - x, f(x))$.

   Якщо $d(x) = 1$, то многочлен $f(x)$ не має коренів в Z_p і алгоритм закінчує роботу.

   Якщо $deg\, d(x) = 1$, то $d(x) = x - a$ і a -шуканий корінь $f(x)$. Алгоритм закінчує роботу.

   Якщо $deg\, d(x) > 1$, то перейти до наступного кроку.

2. Обрати випадковий елемент $b \in Z_p$.

3. Обчислити $g(x) = (d(x), (x+b)^{\frac{p-1}{2}} - 1)$.

   Якщо $g(x) = 1$ або $g(x) = d(x)$, то перейти до кроку 2.

   Якщо $deg\, g(x) = 1$, то $g(x) = x - a$ і a -шуканий корінь $f(x)$ в Z_p. Алгоритм закінчує роботу.

   Якщо $deg\, g(x) = deg\, d(x) - 1$, то $\frac{d(x)}{g(x)} = x - a$ і a -шуканий корінь $f(x)$ в Z_p. Алгоритм закінчує роботу.

   Якщо $2 \le deg\, g(x) < deg\, d(x) - 1$, то перейти до кроку 4.

4. Зробити заміну $d(x) = g(x)$, якщо $deg\, g(x) \le \frac{d(x)}{g(x)}$ і $d(x) = \frac{d(x)}{g(x)}$, якщо $deg\, g(x) > \frac{d(x)}{g(x)}$.

   Перейти до кроку 2.

Оцінка складності: очевидно, що $(x^p - x, f(x)) = (x^p - x \bmod f(x), f(x))$ де, для знаходження $x^p - x \bmod f(x)$ необхідно обчислити $x^p \bmod f(x)$ за допомогою повторюваного піднесення до квадрату. Це потребує не більше $O(m^2 \log p)$ операцій у полі Z_p. Далі обчислюємо $(x^p - x \bmod f(x), f(x))$ за $O(m^2)$ операцій у полі Z_p.

Алгоритм 2

Дано: непарне просте число p, $\alpha \ge 1$, елемент $b \in Z_{p^\alpha}$;

просте число m, таке що $m \mid (p-1)$;

$T = \varphi(p^\alpha) = p^{\alpha-1}(p-1) = m^k h$, де $(m,h) = 1$.

Результат: елемент $a \in Z_p$ такий, що $a^m \equiv b (mod\ p^\alpha)$.

Алгоритм:

1. Знайти будь-який первісний корінь c за модулем p^α.
2. Обчислити $\xi = c^h \bmod p^\alpha$.
3. За допомогою розширеного алгоритму Евкліду знайти цілі числа u, v, такі що $um + vh = 1$.
4. Знайти лишок $r \bmod m^{k-1}$, такий що $b^{vh} \equiv \xi^{mr} (mod\ p^\alpha)$.
5. Покласти $a = b^u \xi^r \bmod p^\alpha$. Алгоритм закінчує роботу.

Перевірка: $a^m \equiv b^{um} \xi^{rm} \equiv b^{1-vh} \xi^{rm} \equiv b (mod\ p^\alpha)$.

Зауваження. Наведені вище алгоритми обчислюють корінь або розв'язують степеневе рівняння у примарному кільці лишків та у простому полі. За Китайською теоремою про лишки та теоремою про ізоморфізм кілець лишків [1, 2], цього достатньо, щоб розв'язати аналогічну задачу у довільному кільці лишків з використанням поліноміального алгоритму зведення.

Як було зазначено, наведені у цьому розділі алгоритми можуть вважатись поліноміальними лише при певних обмеженнях на їх параметри. Крім того, вони є досить громіздкими, і в окремих випадках, що мають

реальний практичний інтерес – занадто надлишковими (наприклад, коли степінь невелика, або коли степеневе рівняння має вигляд $x^k = a$). Тому актуальною задачею на сьогоднішній день залишається пошук поліноміальних та більш оптимальних (порівняно з наведеними) алгоритмів розв'язку степеневих рівнянь у поодиноких випадках, що мають найбільший практичний інтерес: наприклад, добування коренів невеликих степенів у скінчених полях та у кільціх лишків, коли відповідний модуль є добутком двох простих. Саме розв'язку цієї задачі присвячені наступні розділи.

III Алгоритми обчислення кубічного кореня у простому скінченому полі

У цьому розділі ми будуємо алгоритми розпізнавання кубічності та добування кубічного кореня за простим модулем, причому намагаємось максимально зберегти аналогію з алгоритмами добування квадратного кореня, описаними в [1, 2].

Алгоритми обчислення кубічного кореня, що будуть наведені у цьому розділі, дуже подібні до алгоритмів обчислення квадратного кореня, наведених у першій частині. У випадку k=1 вони є сутєво простішими за описані вище алгоритми 1 і 2 (наприклад, всього одне піднесення до степеню, який не перевищує p, або навіть суттєво менший за p). Якщо k>1, то алгоритм 2 та новий запропонований алгоритм мають приблизно однакову складність, але новий алгоритм не потребує розв'язку задачі DLP та є більш простим.

Позначення та основні результати

Нехай p – просте число, g – утворюючий елемент Z_p^*, $a \in Z_p^*$. Введемо позначення:

$$T_p = \left\{ a \in Z_p^* \mid \exists y \in Z_p^* : a \equiv y^3 \left(mod\ p\right) \right\}.$$

Теорема 4: якщо НСД (3, p-1)=1, то $T_p = Z_p^*$. При цьому для будь-якого $a \in Z_p^*$ існує єдине $y \in Z_p^*$, таке, що $a \equiv y^3 \left(mod\ p\right)$.

Якщо ж p − 1=3t для деякого $t \in Z$, то справедливі наступні твердження:

1) $\left|T_p\right| = t$;

2) $a \in T_p \Leftrightarrow a^t\ mod\ p = 1$;

3) $a \in T_p \Leftrightarrow \exists\, 1 \le s \le t : a \equiv g^{3t} \left(mod\ p\right)$.

Доведення даної теореми спирається на той факт, що група Z_p^* є циклічною. Воно аналогічне до доведення відповідної теореми про квадратичні лишки в [1, 2].

Нехай p – 1=3t для деякого $t \in Z$. Тоді, за теоремою про циклічну групу, у групі Z_p^* існує два елементи третього порядку. Будемо їх називати коренями третього степеню з одиниці і позначати $\varepsilon_1, \varepsilon_2$, причому $\varepsilon_2 = \left(\varepsilon_1\right)^2$. Тобто $\left\{1, \varepsilon_1, \varepsilon_2\right\}$ – множина усіх кубічних коренів з одиниці. Зазначимо, що, якщо g – утворюючий елемент Z_p^*, то $\varepsilon_1 = g^t\ mod\ p, \varepsilon_2 = g^{2t}\ mod\ p$, тобто $\varepsilon_1, \varepsilon_2$ легко обчислюються.

Теорема 5: нехай p −1=3t, $a \in T_p$. Тоді:

1) існує рівно три корені кубічних з a за модулем p;

2) якщо $y_1 = \sqrt[3]{a}\ mod\ p$, то два інших корені кубічних будуть мати вигляд

$$y_2 = y_1 \varepsilon_1\ mod\ p \text{ та } y_3 = y_1 \varepsilon_2\ mod\ p.$$

Доведення: оскільки $\varepsilon_i^{\,3}\ mod\ p = 1$, то

$$y_i^{\,3}\ mod\ p = \varepsilon_i^{\,3} y_1^{\,3}\ mod\ p = y_1^{\,3}\ mod\ p = a,$$

тобто y_1, y_2, y_3 є коренями кубічними з елемента a. За теоремою про кількість коренів поліному у полі [8], кількість коренів даного рівняння не може бути більшою за 3. Теорему доведено.

Алгоритми обчислення кубічного кореня у простому скінченому полі

У цьому пункті буде наведено чотири алгоритми обчислення кубічного кореня $\sqrt[3]{a} \bmod p$ для $a \in T_p$, залежно від вигляду числа *p* та/або параметра *t*.

Випадок 1: НСД (3, *p*−1)=1.

Тоді, за розширеним алгоритмом Евкліда, існують такі цілі числа *u*, *v*, що $3u + (p-1)v = 1$. Отже, $a^{3u}\left(a^{p-1}\right)^v \equiv a(mod\ p)$, звідки $\left(a^u\right)^3 \equiv a(mod\ p)$, тобто $y_1 = a^u \bmod p$.

Випадок 2: $p - 1 = 3t$. Цей випадок розбивається на три можливі підпункти: $t \equiv 0, 1, 2\ (mod\ 3)$. Для кожного підпункту буде наведено свій алгоритм обчислення кубічного кореня.

У випадках $t \equiv 1, 2\ (mod 3)$ алгоритм обчислення кубічних коренів є детермінованим.

У випадку $t \equiv 0(mod\ 3)$ алгоритм є імовірнісним. Він передбачає або знання утворюючого елемента мультиплікативної групи відповідного поля, або знання кубічного нелишку певного виду. Хоча алгоритми знаходження утворюючого елемента та кубічного нелишка є імовірнісними Лас-Вегас алгоритмами, математичне сподівання кількості кроків до першого успіху є невеликим, тому алгоритм все одно залишається поліноміальним.

Випадок 2а: *p*−1=3*t*, де $t \equiv 2(mod\ 3)$, $a \in T_p$.

Тоді $a^t \equiv 1(mod\ p)$. Оскільки $t = 3k + 2$ для деякого $k \in Z$, то

$$a^{3k+2} \equiv 1(mod\ p).$$

Домноживши ліву і праву частини на *a*, отримаємо:

$$a^{3(k+1)} \equiv a(mod\ p),$$

звідки

$$y_1 = \sqrt[3]{a} \bmod p = a^{k+1} \bmod p,\ y_2 = y_1\varepsilon_1 \bmod p,\ y_3 = y_1\varepsilon_2 \bmod p.$$

Випадок 2б: *p*−1=3*t*, де $t \equiv 1(mod\ 3)$, $a \in T_p$.

Тоді $a^t \equiv 1(mod\ p)$. Оскільки $t = 3k + 1$ для деякого $k \in Z$, то

$$a^{3k+1} \equiv 1(mod\ p).$$

Домноживши ліву і праву частини на a^{-1}, отримаємо:

$$a^{3k} \equiv a^{-1}(mod\ p),$$

отже,

$$\left(a^{-k}\right)^3 \equiv a(mod\ p)$$

звідки

$$y_1 = a^{-k} \bmod p,\ y_2 = y_1\varepsilon_1 \bmod p,\ y_3 = y_1\varepsilon_2 \bmod p.$$

Випадок 2в: *p* −1=3*t*, де $t \equiv 0(mod\ 3)$, $a \in T_p$.

Нехай $t = 3^k s$, де *s* не ділиться на 3.

Тоді $a^t \equiv 1(mod\ p)$, тобто

$$a^{3^k s} \equiv 1(mod\ p).$$

Тоді можливі три випадки:

$$a^{3^{k-1}s} \equiv \begin{bmatrix} \varepsilon_1 (mod\ p), \\ \varepsilon_2 (mod\ p), \\ 1\ (mod\ p). \end{bmatrix}$$

Подальший розв'язок залежить від того, яке саме порівняння виконується. Нам потрібно знайти два кубічні нелишки u_1, u_2 за модулем p, такі, що

$$u_1^{\frac{p-1}{3}} \equiv \varepsilon_1 (mod\ p), \quad u_2^{\frac{p-1}{3}} \equiv \varepsilon_2 (mod\ p).$$

Ці кубічні нелишки легко обчислити, якщо відомий утворюючий елемент g для групи Z_p^* (який, як відомо з [1, 2], обчислюється поліноміальним імовірнісним алгоритмом). Тоді, наприклад, покладемо

$$u_1 = g, \quad u_2 = g^2\ mod\ p\ .$$

Нехай $a^{3^{k-1}s} \equiv \varepsilon_1 (mod\ p)$. Тоді, оскільки $u_2^{\frac{p-1}{3}} \equiv g^{\frac{2(p-1)}{3}} \equiv \varepsilon_2 (mod\ p)$, то, перемноживши обидві конгруенції, отримаємо:

$$u_2^{3^k s} a^{3^{k-1}s} \equiv 1 (mod\ p),$$

звідки

$$u_2^{3^{k-1}s} a^{3^{k-2}s} \equiv \begin{bmatrix} \varepsilon_1 (mod\ p), \\ \varepsilon_2 (mod\ p), \\ 1\ (mod\ p). \end{bmatrix}$$

Якщо $a^{3^{k-1}s} \equiv \varepsilon_2 (mod\ p)$, то виконуючи аналогічну процедуру, але з використанням конгруенції

$$u_1^{\frac{p-1}{3}} \equiv g^{\frac{p-1}{3}} \equiv \varepsilon_1 (mod\ p),$$

отримаємо

$$u_1^{3^k s} a^{3^{k-1}s} \equiv 1 (mod\ p),$$

і, аналогічно,

$$u_1^{3^{k-1}s} a^{3^{k-2}s} \equiv \begin{bmatrix} \varepsilon_1 (mod\ p), \\ \varepsilon_2 (mod\ p), \\ 1\ (mod\ p). \end{bmatrix}$$

Якщо ж $a^{3^{k-1}s} \equiv 1 (mod\ p)$, то

$$a^{3^{k-2}s} \equiv \begin{bmatrix} \varepsilon_1 (mod\ p), \\ \varepsilon_2 (mod\ p), \\ 1\ (mod\ p). \end{bmatrix}$$

Описану процедуру повторюємо доти, доки степінь l у виразі $a^{3^l s}$ стане рівним нулю (аналогічно до знаходження квадратного кореня імовірнісним алгоритмом у випадку $p \equiv 1 (mod\ 8)$, див. [1, 2]). Тоді на k-му кроці для деякого $b \in Z_p^*$ (як саме виглядає елемент b, залежить від усіх попередніх кроків; зазначимо лише, що $b = b(u_1, u_2)$) отримаємо:

$$b^3 a^s \equiv 1 (mod\ p).$$

Далі, якщо $s = 3m + 2$, то

$$b^3 a^{3m+2} \equiv 1 (mod\ p),\ b^3 a^{3(m+1)} \equiv a (mod\ p),$$

звідки

$$y_1 = ba^{m+1}\ mod\ p\ ,\ y_2 = y_1 \varepsilon_1 mod\ p\ ,\ y_3 = y_1 \varepsilon_2 mod\ p\ .$$

Аналогічно, якщо $s = 3m+1$, то

$$b^3 a^{3m+1} \equiv 1(mod\ p),\ b^3 a^{3m} \equiv a^{-1}(mod\ p),\ \left(b^{-1}a^{-m}\right)^3 \equiv a(mod\ p),$$

звідки

$$y_1 = \left(ba^m\right)^{-1} mod\ p,\ y_2 = y_1\varepsilon_1 mod\ p,\ y_3 = y_1\varepsilon_2 mod\ p.$$

У даному розділі отримано нові алгоритми розпізнавання кубічності та обчислення кубічного кореня у простому скінченому полі. Як було зазначено у розділі 1, дані алгоритми можна узагальнити на випадок кільця лишків, використовуючи Китайську теорему та теорему про ізоморфізм кілець лишків. Отримані алгоритми є поліноміальними; у одному з описаних випадків алгоритм буде імовірнісним Лас-Вегас алгоритмом. Слід зазначити, що алгоритми розпізнавання кубічності та добування кубічного кореня виявились дуже подібними до відповідних «квадратичних» алгоритмів, описаних у [1, 2]. Це можна пояснити тим, що принципи побудови алгоритмів були дуже схожими, хоча є й окремі відмінності. Наприклад, при розпізнаванні квадратичності та добуванні квадратного кореня є суттєвим той факт, що для простого p число $p-1$ завжди ділиться на 2, а от аналогічного твердження для кубічного кореня вже немає. Також слід зазначити, що саме випадок кубічного рівняння є найбільш цікавим з практичної точки зору, наприклад, при побудові еліптичних кривих над кільцями лишків та обчисленні їх параметрів (зокрема, базової точки).

IV Алгоритми обчислення кореня довільного степеню у простому скінченому полі

У даному розділі ми узагальнимо результати розділу 3 на випадок довільного степеню. Природньо, що алгоритми при цьому стануть більш громіздкими, але все одно залишаться більш зручними та швидкими порівняно з алгоритмами 1 і 2.

Позначення та основні результати

Нехай p – просте число, g – утворюючий елемент Z_p^*, $a \in Z_p^*$; k – деяке просте від 3 до $p-1$. Введемо позначення:

$$T_p^{(k)} = \left\{a \in Z_p^* \mid \exists y \in Z_p^* : a \equiv y^k (mod\ p)\right\}.$$

Теорема 6: якщо НСД (k, $p-1$)=1, то $T_p^{(k)} = Z_p^*$. При цьому для будь-якого $a \in Z_p^*$ існує єдине $y \in Z_p^*$, таке, що $a \equiv y^k (mod\ p)$.

Якщо ж $p-1=kt$ для деякого $t \in Z$, то справедливі наступні твердження:

1) $\left|T_p^{(k)}\right| = t$;

2) $a \in T_p^{(k)} \Leftrightarrow a^t\ mod\ p = 1$;

3) $a \in T_p^{(k)} \Leftrightarrow \exists 1 \le s \le t : a \equiv g^{kt}(mod\ p)$.

Нехай $p-1=kt$ для деякого $t \in Z$. Тоді, за теоремою про циклічну групу, у групі Z_p^* існує $\varphi(k) = k-1$ елементів k-го порядку. Будемо їх називати коренями k-го степеню з одиниці і позначати $\varepsilon_1, ..., \varepsilon_{k-1}$, причому $\varepsilon_2 = (\varepsilon_1)^2$, і т.д. Тобто $\{1, \varepsilon_1, ..., \varepsilon_{k-1}\}$ – множина усіх коренів k-го степеню з одиниці. Зазначимо, що, якщо g – утворюючий елемент Z_p^*, то $\varepsilon_1 = g^t\ mod\ p, ..., \varepsilon_{k-1} = g^{(k-1)t}\ mod\ p$, тобто $\varepsilon_1, ..., \varepsilon_{k-1}$ легко обчислюються.

Теорема 7: нехай $p-1=kt$, $a \in T_p$. Тоді:

1) існує рівно k коренів k-го степеню з a за модулем p;

2) якщо $y_1 = \sqrt[k]{a}\ mod\ p$, то всі інші корені k-го степеню будуть мати вигляд $y_2 = y_1\varepsilon_1\ mod\ p, ..., y_k = y_1\varepsilon_{k-1}\ mod\ p$.

Доведення теорем 6 та 7 є аналогічними до доведень теорем 4 та 5.

Алгоритми обчислення кореня *k*-го степеню у простому скінченому полі

У цьому пункті буде наведено алгоритми обчислення кубічного кореня для $a \in T_p^{(k)}$, залежно від вигляду числа *p* та/або параметра *t*. Тут ми їх наведемо лише частково.

Випадок 1: НСД (*k*, *p* −1)=1.

Тоді, за розширеним алгоритмом Евкліда, існують такі цілі числа *u*, *v*, що

$$3k + (p-1)v = 1.$$

Отже,

$$a^{ku}\left(a^{p-1}\right)^v \equiv a(mod\ p),$$

звідки

$$\left(a^u\right)^k \equiv a(mod\ p),$$

тобто

$$y_1 = a^u mod\ p.$$

Випадок 2: p −1=kt, де $(k,t)=1$. Тобто p −1 ділиться на k, але не ділиться на k^2. Цей випадок розбивається на два підвипадки. У обох підвипадках алгоритми є детермінованими. Вони зводять задачу обчислення кореня *k*-го степеню до задачі обчислення кореня степеню $s \le [\frac{k}{2}]$.

У випадку $t \equiv 0(mod\ 3)$ алгоритм є імовірнісним. Він передбачає або знання утворюючого елемента мультиплікативної групи відповідного поля, або знання кубічного нелишку певного виду. Хоча алгоритми знаходження утворюючого елемента та кубічного нелишка є імовірнісними Лас-Вегас алгоритмами, математичне сподівання кількості кроків до першого успіху є невеликим, тому алгоритм все одно залишається поліноміальним.

Випадок 2а: $\frac{p-1}{k} = lk + r,$, де $\frac{k}{2} < r < k-1$, $a \in T_p^{(k)}$. Тоді $a^{lk+r} \equiv 1(mod\ p)$.

Домноживши ліву і праву частини на a^{k-r}, отримаємо:

$$a^{lk+r+k-r} \equiv a^{k-r}(mod\ p),$$
$$a^{(l+1)k} \equiv a^{k-r}(mod\ p),$$
$$\left(a^{l+1}\right)^k \equiv a^{k-r}(mod\ p),$$
$$\left(\sqrt[k-r]{a^{l+1}}\right)^k \equiv a(mod\ p),$$

звідки

$$y_1 = \sqrt[k-r]{a^{l+1}}\ mod\ p,\ \ldots,\ y_k = y_1\varepsilon_{k-1} mod\ p.$$

Випадок 2б: $\frac{p-1}{k} = lk + r,$, де $1 \le r \le \frac{k}{2}$, $a \in T_p^{(k)}$.

Тоді $a^{lk+r} \equiv 1(mod\ p)$.

Домноживши ліву і праву частини на a^{-r}, отримаємо:

$$a^{lk} \equiv a^{-r}(mod\ p),$$
$$\left(\left(\sqrt[r]{a^l}\right)^{-1}\right)^k \equiv a(mod\ p),$$

звідки

$$y_1 = \left(\sqrt[r]{a^l}\right)^{-1}\ mod\ p,\ \ldots,\ y_k = y_1\varepsilon_{k-1} mod\ p.$$

Випадок 2в: $p-1=kt$, де $t \equiv 0(mod\ k)$, $a \in T_p^{(k)}$.

Нехай $t = k^u s$, де s не ділиться на k.

Тоді $a^t \equiv 1(mod\ p)$, тобто

$$a^{k^u s} \equiv 1(mod\ p). \quad (1)$$

Позначимо $\sum_p^{(k)} = \{\varepsilon_1, \varepsilon_2, ..., \varepsilon_k\}$ - множину усіх коренів k - го степеню з одиниці (вважаємо, що $\varepsilon_k = 1$) у полі Z_p. Зазначимо, що задача знаходження множини $\sum_p^{(k)}$ зводиться до задачі знаходження генератора Z_p^*. Дійсно, якщо g – генератор Z_p^*, то $\varepsilon_1 = g^{\frac{p-1}{k}} mod\ p = g^t mod\ p$, $\varepsilon_2 = g^{2t} mod\ p$, …, $\varepsilon_{k-1} = g^{(k-1)t} mod\ p$, $\varepsilon_k = g^{kt} mod\ p = 1$. Тобто $\varepsilon_i = \varepsilon_{i-1} g^t mod\ p = (\varepsilon_1)^i mod\ p = g^{ti} mod\ p$. Рівняння (1), за теоремою про кількість коренів степеневого рівняння у полі (див., наприклад, [7]), еквівалентно наступній сукупності рівнянь:

$$a^{k^{n-1}s} \equiv \begin{cases} \varepsilon_1 mod\ p; \\ \varepsilon_2 mod\ p; \\ \vdots \\ \varepsilon_{k-1} mod\ p; \\ 1. \end{cases}$$

Подальший розв'язок залежить від того, яке саме порівняння виконується. Нехай виконується порівняння

$$a^{k^{u-1}s} \equiv \varepsilon_i (mod\ p). \quad (2)$$

Зазначимо, що $\varepsilon_i \cdot \varepsilon_{k-i} mod\ p = g^{ti} \cdot g^{t(k-i)} mod\ p = g^{kt} mod\ p = 1$, тобто

$$\varepsilon_i \cdot \varepsilon_{k-i} \equiv 1(mod\ p). \quad (3)$$

Оскільки $\varepsilon_{k-i} = g^{t(k-i)} mod\ p$, то

$$g^{t(k-i)} \equiv \varepsilon_{k-i} (mod\ p). \quad (4)$$

Перемноживши (2) і (4) та враховуючи (3) отримаємо:

$$a^{k^{u-1}s} g^{t(k-i)} \equiv 1(mod\ p), \text{ або } g^{k^u s(k-i)} a^{k^{u-1}s} \equiv 1(mod\ p),$$

звідки

$$a^{k^{u-1}s(k-i)} a^{k^{u-2}s} \equiv \begin{cases} \varepsilon_1 mod\ p; \\ \vdots \\ \varepsilon_{k-1} mod\ p; \\ 1(mod\ p). \end{cases}$$

Описану процедуру повторюємо доти, доки степінь l у виразі $a^{k^l s}$ не стане рівним нулю (аналогічно до знаходження квадратного кореня імовірнісним алгоритмом у випадку $p \equiv 1(mod\ 8)$). Тоді на деякому кроці для деякого $b \in Z_p^*$ (як саме виглядає елемент b, залежить від усіх попередніх кроків) отримаємо конгруенцію $b^k a^s \equiv 1(\mathrm{mod}\ p)$, що розв'язується відповідно до випадку 2б.

У даному розділі побудовано алгоритми обчислення кореня довільного степеню у простому скінченому полі. Зазначимо, що внаслідок китайської теореми про лишки та теореми про ізоморфізм кілець лишків на базі даних алгоритмів можна побудувати аналогічні алгоритми у кільці лишків, але за умови, що порядок кільця вільний від квадратів. Наведені алгоритми є рекурентними; вони зводяться до алгоритмів добування кореня, степінь якого менша як мінімум у два рази. Тобто кількість ітерацій не буде перевищувати величину двійкового логарифму від степеню рівняння. Також зручним є той факт, що деякі попередні обчислення, що не залежать від входу (тобто від числа, з якого добувається корінь), можуть бути зроблені заздалегідь, що суттєво зменшить час роботи алгоритму. Це стосується, в першу чергу, алгоритму знаходження

примітивного елементу мультиплікативної групи відповідного скінченого поля та інших обчислень, в яких примітивний елемент використовується.

V Висновки

У даній роботі отримано критерій степеневості елемента скінченого поля та наведено прості рекурентні алгоритми обчислення кубічного кореня та коренів більш високого степеню з елементу поля, який є відповідним лишком. Алгоритми розпізнавання степеневості та добування кореня за складеним модулем (наприклад, за модулем $n=pq$) повністю визначаються відповідними алгоритмами за простим модулем, за умови, що відомий розклад числа n на прості множники та це число є вільним від квадратів.

Цікаво також зазначити, що задача розкладу на прості множники та задача добування кореня є поліноміально еквівалентними відносно імовірнісного алгоритму.

Дані алгоритми є, перш за все, цікавими з математичної точки зору, а їх поодинокий випадок – алгоритми добування кубічного кореня – можуть бути використані при побудові криптосистем на еліптичних кривих над кільцями лишків. Так, з використанням вказаних алгоритмів, замість імовірнісних алгоритмів обчислення базової точки кривої та алгоритмів "вкладання" відкритого тексту у точку кривої можна побудувати детерміновані алгоритми.

*Література: **1.** Коблиц Н. Курс теории чисел и криптографии. // Пер. с англ. – М.: Научное изд-во ТВП, 2001. – 254 с. **2.** О. Вербицький, "Вступ до криптології" // Л.: Вид-во науково-технічної літератури, 1998р., 247с. **3.** А. Бессалов, А. Тележенко, "Криптосистеммы на эллиптических кривых", Киев, 2004, 223 с. **4.** Н. Бабенко, "Методы и алгоритмы вычисления структур на ЭК с параллелизмом машинных операций" // Автореф. на соискание степени к.ф.-м.н., Ставрополь, 2011, 19с. **5.** А. Нестеренко, "Об одном варианте метода Ленстры факторизации целых чисел" // Материалы 3-ей международной конференции "Математика и безопасность информации" (МаБИТ-07), МГУ, 25-27 октября 2007 года, М.: МЦМНО, 2008, с. 234-240. **6.** М. Самохина. "Эллиптические кривые" // Доклад на семинаре кафедры радиотехники МФТИ, radio.fiztex.ru//infsec/f_3kdhla/f_3erfbr/seminar_6.pdf. **7.** М. Глухов. И. Круглов. А. Пичкур, А. Черёмушкин, "Введение в теоретико-числовые методы криптографии" // СПб.: Изд-во "Лань", 2011, 400с. **8**. Лидл Р., Нидеррайтер Г. Конечные поля: В 2-х т. / Пер. с англ. – М.: Мир, 1988. – 822 с.*

УДК 004.056.5

РАЗРАБОТКА НОВОГО ПОДХОДА К ВЫЯВЛЕНИЮ ЗАМЕЩАЮЩЕЙ ОБЛАСТИ В ЦИФРОВОМ ИЗОБРАЖЕНИИ

Алла Кобозева, Елена Лебедева

Одесский национальный политехнический университет

Анотація: ***Стаття присвячена подальшій розробці нового підходу до рішення задачі виявлення і локалізації заміщуючої області в цифровому зображенні, що був запропонований одним з авторів раніше, реалізацією якого став новий метод, який є інваріантним відносно формату збереження зображення.***

Summary: ***The article is devoted to the further development of a new approach to solving the problem of detection and localization of substitution region in a digital image that was proposed by one of the authors earlier, the implementation of which was a new method that is invariant relative to the storage format of the image.***

Ключові слова: ***Цифрове зображення, заміщуюча область, фальсифікація, сингулярне число, сингулярний вектор, матриця.***

I Введение

Неотъемлемой составной частью функционирования и жизнеобеспечения современного общества является анализ и обработка цифровых сигналов, преследующая разные цели. Несанкционированные изменения информационных контентов могут привести к невосполнимым пагубным последствиям как для отдельных лиц, так и для организаций и даже государства.

Последние успехи в создании, развитии и общедоступности редактирующего цифровые сигналы программного обеспечения требуют обязательного включения в комплексную систему защиты информации методов проверки целостности цифровых изображений (ЦИ), видео, аудио, отнесенных в [1] к пассивным методам информационной безопасности.