1. Всі найпростіші КМ можна розділити на 5 груп за ознакою наявності та кількості їх типів: модулі, що не мають жодного типу (такі модулі не можуть бути використані для реалізації повних підстановок), модулі, що мають 1, 2 або 3 типи (такі модулі можуть бути використані для реалізації повних підстановок за певних умов) і модулі, що мають всі 4 типи (такі модулі можуть бути використані для реалізації повних підстановок).
2. Кількість комбінацій чотирьох типів дорівнює 75. При цьому, якщо один з типів модуля є 2в, то всі інші його типи також 2в.
3. Модулі, сусідні типи яких є сумісними, реалізують повну підстановку.
4. Якщо пара модулів реалізує повну підстановку, то її об'єднаний тип завжди належить до класифікації, що наведена вище.
5. Шляхом повного перебору доведено, що об'єднаний тип двох модулів не залежить від типів модулів, а тільки від комбінації бокових та основних функцій. При цьому, якщо каскад модулів реалізує повну підстановку, то його відповідний об'єднаний тип 2а, 2б або 3а. Якщо каскад не реалізує повну підстановку, то його об'єднаний тип або не входить в наведену вище класифікацію, або 2в (останній випадок дає можливість нарощувати каскад). Модулі типу 3б та 3в не можуть бути отримані шляхом об'єднання двох найпростіших модулів модулів.

Висновки

В статті розглянуто методику реалізації повних підстановок за допомогою найпростішого каскаду конструктивних модулів при довільній розрядності підстановок. Кожний такий модуль має прості логічні функції, каскад легко нарощується, тому реалізація як окремого модуля, так і каскада в цілому за допомогою сучасних технологій не викликає ускладнень. Тому метод є перспективним для реалізації масових підстановок довільної розрядності. Необхідно відмітити, що найпростіший ОККМ може реалізувати до двох різних підстановок на найпростіших ОККМ першого варіанту і до чотирьох різних підстановок для другого варіанту найпростіших ОККМ; для цього необхідно лише змінити значення на бокових входах.

*Література:**1.** Опанасенко В. Н., Сахарин В. Г. ПЛИС типа FPGA фирмы Xilinx: возможности, проектирование и применение. Электронные системы и компоненты. – 2003, № 4, с.7-11. **2.** Кузелин М. О., Кнышев Д. А., Зотов В. Ю. Современные семейства ПЛИС фирмы Xilinx. – М: «Горячая линия-Телеком», 2004, 440 с. **3.** Кострикин А. И. Введение в алгебру. М. Наука, 1994, 232 с **4.** Тарасенко В. П., Тесленко О. К., Яновська О. Ю. Проблеми апаратної реалізації підстановок. Наукові записки УНДІЗ, №2, 2007, с 52-58 **5.** Михайлюк А. Ю., Тарасенко В. П., Тесленко А. К. "Анализ эффективности применения нетрадиционных элементарных функций шифрования". Матеріали Третьої міжнародної науково-практичної конференції «Безпека інформації в інформаційно-телекомунікацій-них системах» К.2000. с. 161-168 **6.** Тарасенко В. П., Тесленко О. К, Яновська О. Ю. Реалізація повних підстановок на простому двомодульному каскаді конструктивних модулів, Інформаційні технології та комп'ютерна інженерія, № 1(11), 2008, с.88-97. **7.** Яновська О. Ю. Тесленко О. К., Тарасенко В. П., Комп'ютерна програма дослідження реалізації повних підстановок на найпростіших нерегулярних структурах (SuperModuli), Свідоцтво про реєстрацію авторського права на твір № 25782 від 24.09.2008.*

УДК 681.3

АЛГОРИТМИ КОДУВАННЯ-ДЕКОДУВАННЯ УЗАГАЛЬНЕНИХ ЗАВАДОСТІЙКИХ КОДІВ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ В УМОВАХ ПРИРОДНИХ ВПЛИВІВ

Вячеслав Василенко
Національний авіаційний університет

Анотація: **Для використання в задачах забезпечення цілісності інформаційних об'єктів в умовах природних впливів пропонуються узагальнені завадостійкі коди з удосконаленими алгоритмами кодування – декодування.**

Summary: **For the use in the tasks of providing of integrity of information's holding object in the conditions of influence of natural factors the generalized antigambling codes are offered.**
Ключові слова: **Виявлення викривлень, виправлення викривлень, контроль цілісності, завадостійкі корегуючі коди.**

І Задачі захисту цілісності інформаційних об'єктів телекомунікаційних мереж

Відповідно до термінології нормативних документів Департаменту спеціальних телекомунікаційних систем і захисту інформації Служби безпеки України [1] під цілісністю інформації розуміється відсутність в ній будь-яких викривлень (модифікацій), які не були санкціоновані її власником, незалежно від причин або джерел виникнення таких викривлень.

Порушення цілісності можливі на будь-якому етапі її циркуляції в обчислювальних мережах, в першу чергу на етапі передачі. Однією з причин таких викривлень можуть бути випадкові природні викривлення, пов'язані з дією природних чинників (атмосферні електромагнітні розряди, іскріння контактів в автомобілях, електротранспорті, недостатня надійність електронних елементів і елементів електричних ланцюгів, порушення реєструючого шару магнітних або оптичних носіїв і багато що інше). Наслідком таких впливів є викривлення того або іншого числа символів в цифровому представленні інформації, незалежно від системи числення або форми представлення інформації.

Використання викривленої інформації тягне за собою наслідки (часто надзвичайно важкі) для власників або користувачів цієї інформації. Тому задача забезпечення цілісності інформаційних ресурсів є однією з найактуальніших при розробці і експлуатації АС і їх елементів.

Для забезпечення контролю та поновлення цілісності інформаційних об'єктів, включаючи і відновлення зруйнованої інформації, до складу інформації, яка захищається, включають надмірну інформацію – ознаку цілісності або контрольну ознаку (залежно від прийнятої в задачах контролю цілісності або завадостійкого кодування термінології) − своєрідний образ, відображення цієї інформації, процедура формування якого відома, і який з дуже високою вірогідністю відповідає інформації, що захищається.

При цьому між інформацією, що захищається, і ознаками цілісності або контрольними ознаками встановлюється регулярний (функціональний) односторонній зв'язок (алгоритми розрахунку контрольної ознаки за початковою інформацією, що захищається, відомі, а алгоритму розрахунку початкової інформації по контрольних ознаках найчастіше не існує). Контроль цілісності (на відсутність викривлень) зводиться при цьому до тих або інших алгоритмів перевірки наявності вказаного регулярного (функціонального) одностороннього зв'язку між ознаками цілісності і прийнятою з каналу зв'язку інформацією.

Характерною особливістю випадкових викривлень є те, що вони, через їх хаотичність, відсутність навмисності, порушують регулярний (функціональний) односторонній зв'язок між прийнятою (або зчитаною із ЗП) інформацією і ознаками цілісності, сформованими перед передачею (перед записом в ЗП). Тому при виявленні порушення вказаного зв'язку встановлюється факт наявності таких викривлень, а за певних умов, і їх місця та величини (характер). За відсутності порушення цього зв'язку встановлюється факт відсутності викривлень.

Серед основних способів (механізмів) забезпечення цілісності інформації в умовах природних дій (проблема завадостійкості) для каналів ТКМ (взагалі для мереж передачі даних) слід виділяти застосування різного роду завадостійких корегуючих кодів (ЗКК), які дозволяють реалізувати програмні, апаратурні або програмно-апаратурні засоби виявлення і усунення викривлень. Одним із видів таких кодів є узагальнені завадостійкі корегуючі коди [2].

Під узагальненими розуміються коди, призначені для виявлення (виявлення і виправлення) пакетних викривлень з кратністю b, в яких використовуються алгоритми кодування і декодування стосовно узагальнених b – розрядних символів.

В цих кодах початкова двійкова кодова послідовність − базове кодове слово (БКС) I1 I2 … Ik розбивається на n = k/b груп двійкових розрядів з розрядністю b, в яких передбачається виявлення та виправлення викривлень:

$$\underbrace{I_1 \ldots\ldots I_b}_{1\text{ – а група}},\ \underbrace{I_{b+1} \ldots\ldots I_{2b}}_{2\text{ – а група}} \ldots\ldots \underbrace{I_{к-b+1} \ldots\ldots I_k}_{n\text{ – а група}};$$

Двійкові символи, що входять в одну b − розрядну групу, розглядаються як b − значний узагальнений символ, який може приймати будь-яке із s значень від 0 до (s – 1), де

$$s = 2^b.$$

При кодуванні та декодуванні операції над узагальненими символами в [2] запропоновано виконувати операції по деякому модулю, тобто розшукувати лишок від розподілу результату операції на деякий модуль. Це дало автору можливість, в разі застосування алгоритмів, які можуть бути аналогічними відповідним

алгоритмам двійкових кодів (але відносно узагальнених символів), для відмінності відповідних узагальнених кодів від двійкових ввести в їх назву слово "лишок", тобто говорити про лишково – Хеммінгові (ЛХ), лишково – матричні (ЛМ), лишково – згорточні (ЛЗ) чи лишково – ланцюгові (ЛЛ) та інші коди.

Принципи побудови та застосування таких кодів розглянемо на прикладі лише деяких із таких кодів. При потребі читач самостійно може застосувати викладені підходи і щодо інших кодів цього класу.

II Узагальнений завадостійкий код умовних лишків

Одним із прикладів узагальнених кодів є код умовних лишків (лишків умовних код, ЛУ – код). Теоретичною основою ЛУ – коду є теорія лишкових класів [3]. З цієї теорії відомо, що будь-яке число можна представити у вигляді набору лишків від розподілу цього числа на набір взаємно простих чисел, які мають назву основ системи числення – pi, де i = 1, 2, …, n, n – кількість таких основ. Вибір величини n здійснюється з умови, яка викладена нижче. Тоді

$$A = \alpha_1, \alpha_2, \ldots, \alpha_n, \quad (1)$$

де $\alpha = A - [A/p_i]\cdot p_i$, а позначка $[A/p_i]$ означає операцію розрахунку цілої частини від дробового числа A/p_i. При цьому між числом A і його уявленням (1) існує взаємна однозначна відповідність, якщо

$$A \le P = \prod_{i=1}^{n} p_i .$$

У цьому виразі величина P – діапазон представлення або робочий діапазон чисел. Звернемо увагу на те, що величина αi представляє собою групу двійкових розрядів, кількість яких не перевищує розрядності відповідної основи pi.

Чудовою властивістю системи лишкових класів (СЛК) є те, що в неї легко вводяться властивості виявлення і виправлення викривлень. Відомо, що якщо ввести ще одну, надлишкову, основу pk, то уявлення A в розширеному діапазоні R = P·pk, у вигляді

$$A = \alpha_1, \alpha_2, \ldots, \alpha_n, \alpha_k, \quad (2)$$

де $\alpha_к$ – лишок по основі $p_к$, має чудову для побудови корегуючих кодів властивість: при $p_k > p_n$ будь-яке викривлення в одному з лишків α_i може бути знайденим, а при $p_к > 2\cdot p_n\cdot p_{n-1}$, де p_n, p_{n-1} – найбільші із основ, може бути і виправленим. Це означає, що при представленні чисел у вигляді (2) створюється завадостійкий код з можливостями або виявлення викривлень, або і їх корекції.

Такий код має принаймні два недоліки. Перший з них пов'язаний з тим, що можливі викривлення знаходяться і виправляються (викривлений символ поновлюється) тільки в тому випадку, якщо викривлений лише один із символів αi, тобто викривлення повинні бути фіксованими в межах однієї із груп розрядів. Такий недолік є притаманним і будь-якому іншому коду і тому усувається відомими способами − застосуванням перемежування з глибиною не меншою ніж два. Другий недолік пов'язаний з необхідністю роботи з числами в системі числення в залишкових класах. Цей недолік достатньо просто усувається в коді умовних лишків, який вводиться таким чином.

Хай є код деякого числа A (БКС), представленого в будь-якій системі числення, зокрема позиційної, наприклад двійкової. Для визначеності, хай це число A представлено послідовність з нулів і одиниць. Розіб'ємо цю послідовності певним (у загальному випадку довільним) чином на n узагальнених символів, як і для решти узагальнених кодів.

Як і раніше код кожного i – го узагальненого символу розглядатимемо як s – значний розряд αi, який може приймати будь-яке з s значень від 0 до (s – 1), де s = 2b. Важатимемо цей код лишком деякого умовного числа A по основі pi. Оскільки величина αi, як елемент початкового числа

$$0 \le \alpha_i \le s - 1,$$

а як лишок від ділення A на p_i

$$0 \le \alpha_i \le p_i,$$

то для представлення коду будь-якої групи у вигляді лишку по основі pi необхідно, щоб виконувалася умова

$$p_i > s - 1,$$

інакше в групу із b розрядів може бути записаним код $\alpha_i \ge p_i$, що в лишкових класах не допустимо.

Приклад. Хай b = 3, s = 7, тоді αi може приймати значення 000, 001, 010,..., 110. При pi = 5 максимальне значення αi обмежується кодом 100, тобто коди 101, 110, 111 є "неправильними". Якщо ж взяти pi > 7, наприклад pi = 9, то максимальне значення αi обмежується не величиною pi, а розрядністю групи b, тобто αmax = 111.

При такому підході будь-які комбінації початкового коду числа A "вписуються" в систему числення з основами pi (i = 1, 2,...). Якщо розширити систему основ на надлишкову (контрольну) pk і для одержаного

набору умовних лишків αi (i = 1, 2,...) розрахувати надлишковий умовний лишок αk, то на одержане умовне число

$$A' = \alpha_1, \alpha_2, \ldots, \alpha_{n1}, \alpha_к \quad (3)$$

поширюються усі можливості СЛК з виявлення і виправлення викривлення, тобто одержаний код (3) має всі властивості коду (2), але останній код може бути отриманим для будь-якої двійкової послідовності, а не тільки для чисел в лишкових класах. Відзначимо, що таким чином усунено другий недолік коду (2).

Оскільки для отримання контрольної ознаки, тобто для кодування будь-якої послідовності двійкових цифр завадостійким кодом, умовно, не реально, не фізично, групи розрядів початкового числа розглядаються як деякі лишки, то такий код одержав найменування коду умовних лишків.

Слід звернути увагу на те, що при кодуванні ЛУ-кодом початкова послідовність не змінюється, до неї тільки приформовуються додаткові, обчислені за окремими правилами, контрольні символи.

Таким чином ЛУ-код дозволяє знаходити і виправляти b − розрядні пакети викривлень, згруповані в межах будь-якого з n узагальнених символів і потребує при цьому надмірність біля

$$r \approx 2b + 1$$

двійкових розрядів (оскільки $p_k \approx 2p_n p_{n+1}$, $r = [\log 2p_k] + 1$). В конкретних випадках ця надмірність може відхилятися в ту або іншу сторону, що залежить також від алгоритмів кодування-декодування.

III Алгоритми кодування-декодування ЛУ-коду

Оскільки в основі ЛУ-коду лежать властивості СЛК, то в цьому коді принципово можуть бути використані відомі алгоритми кодування-декодування. До таких алгоритмів відносяться алгоритм нулізації і, так званий z – алгоритм [3, 4].

В основі цих алгоритмів лежить той факт, що будь-яке викривлення в одній з груп розрядів α_i переводить початкове число з робочого діапазону $[0, P = \prod_{i=1}^{k} p_i)$ в діапазон $[P, R = p_k \cdot P)$, тобто призводить (див. рис. 1) до збільшення початкового числа $A' < P$ на деяку величину $l_i \cdot R_i$. Тут l_i і $R_i = R/p_i$ – цілі числа. Дійсно, якщо вихідне число

$$A = \alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_n, \alpha_k$$

є викривленим по основі p_i і має вид

$$\widetilde{A} = \alpha_1, \alpha_2, \ldots, \widetilde{\alpha}_i, \ldots, \alpha_n, \alpha_к$$

де

$$\widetilde{\alpha}_i = \{\alpha_i + \Delta\alpha_i\} \pmod{p_i},$$

то це є еквівалентним наступному перетворенню

$$\widetilde{A} = (\alpha_1, \alpha_2, \ldots, \alpha_i, \ldots, \alpha_n, \alpha_k) + (0, 0, \ldots, \Delta\alpha_i, \ldots, 0, 0) =$$
$$= (\alpha_1, \alpha_2, \ldots, \{\alpha_i + \Delta\alpha_i\} \pmod{p_i}, \ldots, \alpha_{n1}, \alpha_k).$$

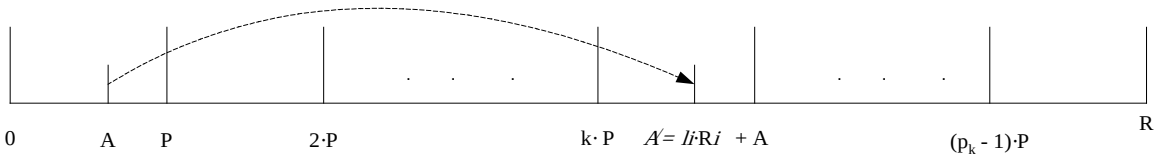

Рисунок 1 – До виходу викривленого числа за межі робочого діапазону

Величина викривлення перевищує величину робочого діапазону P

$$\Delta A = (0, 0, \ldots, \Delta\alpha_i, \ldots, 0, 0) > P,$$

оскільки тільки число виду

$$\Delta A = l_i \cdot R_i = l_i \cdot R/p_i$$

має всі лишки, окрім лишка по основі p_i, такими, що дорівнюють нулю. Але $\Delta A = l_i \cdot R_i > P = R/p_k$ тобто, навіть при $l_i = 1$ величина $R/p_i > R/p_k$ бо $p_k > p_i$.

Відтак, сума $\widetilde{A} = A' + \Delta A > P$, тобто викривлене число вийшло за межі робочого діапазону P і попало в діапазон [P, R).

Згадані алгоритми кодування-декодування як раз і використовують цей факт.

Використання для кодування - декодування z – алгоритму

Для виявлення викривлень в z - алгоритмі використовується відмічений вище факт, що викривлене число виходить за межі робочого діапазону, тобто

$$\tilde{A} \geq \mathrm{P}. \tag{4}$$

Скористаємось відомим співвідношенням для переводу чисел із СЛК в позиційну систему числення

$$\tilde{A} = \sum_{i=1}^{i=n+1} \alpha_i B_i - [(1/R) \sum_{i=1}^{i=n+1} \alpha_i B_i] \times R, \tag{5}$$

де: B_i – константа системи числення, її ортогональний базис, причому

$$B_i = R \cdot m_i / p_i, \ (i = 1, 2, \ldots, n + 1); \tag{6}$$

(n + 1) – число умовних основ, включаючи контрольну; m_i- ціле позитивне число, “вага” ортогонального базису B_i, таке при якому

$$m_i B_i \ (\mathrm{mod}\ p_i) = 1.$$

Підставивши вираз (5) в (4) з урахуванням (6), отримаємо

$$\sum_{i=1}^{i=n=1} \alpha_i R \cdot m_i / p_i - [(1/R) \sum_{i=1}^{i=n+1} \alpha_i R \cdot m_i / p_i] \times R > \mathrm{R}/p_k. \tag{7}$$

Скоротивши обидві частини (7) на R отримаємо, що в разі наявності викривлень,

$$z > 1/p_к, \tag{8}$$

де

$$z = \sum_{i=1}^{n+1} \alpha_i m_i / p_i - [\sum_{i=1}^{n+1} \alpha_i m_i / p_i]. \tag{9}$$

Вирази (8 – 9) визначають z – алгоритм декодування для ЛУ-коду, який лише визначає наявність викривлень. Цей алгоритм включає (n + 1) незалежних (при необхідності одночасних) операцій множення коду і-ої групи (i = 1, …, n +1) на відповідну константу і потім додавання (n +1) отриманих добутків.

Не розглядаючи побудови алгоритму, здатного не лише визначати наявність, але й виправляти викривлення, визначимо недоліки подібного підходу, які не дозволяють повною мірою покладатися на алгоритми даного класу. Для цього скористаємось наступними міркуваннями щодо виразу (9). Звернемо увагу на те, що змінна рі, яка входить у доданки, що присутні в цьому виразі у вигляді $\alpha_i m_i / p_i$, є майже завжди простим числом. Внаслідок цього переважна більшість доданків є дробовими числами із нескінченою довжиною. Оскільки пристроїв для здійснення операцій над такими числами не відомо, то отримати точні значення виразу (9) не можливо. Отже, при використання z – алгоритму декодування у відповідному корегуючому алгоритмі слід очікувати досить часто невірного результату, в силу чого можливим є лише обмежене його застосування. Вільним від цих вад є наступний алгоритм.

Використання для кодування-декодування алгоритму нулізації

Суть алгоритму нулізації зводиться до того, що як при кодуванні, так і при декодуванні числа

$$\tilde{A} = (\alpha_1, \alpha_2, \ldots, \{\alpha_i + \Delta\alpha_i\} \ (\mathrm{mod}\ p_i), \ldots, \alpha_n, \alpha_k)$$

по лишкам усіх n основ, що утворюють робочий діапазон α_i (i = 1, 2, …, n), послідовно формуються, так звані мінімальні числа виду

$$t_1 = (\alpha_1, \alpha_2^{/}, \alpha_3^{/}, \ldots, \alpha_n^{/}, \alpha_k^{/}),$$

$$t_2 = (0, (\alpha_2 - \alpha_2^{/}) \ (\mathrm{mod}\ p_2), \alpha_3^{(2)}, \ldots, \alpha_n^{(2)}, \alpha_k^{(2)}),$$

$$t_3 = (0, 0, (\alpha_3 - \alpha_3^{/} - \alpha_3^{(2)}) \ (\mathrm{mod}\ p_3), \alpha_4^{(3)}, \ldots, \alpha_n^{(3)}, \alpha_k^{(3)}),$$

……..

$$t_n = (0, 0, 0, \ldots, (\alpha_n - \sum_{j=1}^{n-1} \alpha_n^{(j)}) \ (\mathrm{mod}\ p_n), \alpha_k^{(n)}),$$

Кожне з таких мінімальних чисел може бути представленим у вигляді

$$t_i = v_i \cdot \prod_{j=1}^{i-1} p_i \ .$$

З урахуванням того, що в системі лишкових класів

$$t_i \,(\mathrm{mod}\, p_i) = \alpha_i^{i-1} = \{\alpha_i - \sum_{j=1}^{i-1} \alpha_i^{(j)} \} \,(\mathrm{mod}\, p_i) = v_i \cdot \prod_{j=1}^{i-1} p_i \ \ (\mathrm{mod}\, p_i),$$

величину v_i можна визначити як:

$$v_i = \{ \alpha_i^{i-1} / \prod_{j=1}^{i-1} p_i \} (\mathrm{mod}\, p_i) = \{(\alpha_i - \sum_{j=1}^{i-1} \alpha_i^{(j)}) / \prod_{j=1}^{i-1} p_i \} \,(\mathrm{mod}\, p_i)$$

для усіх лишків α_i з номерами $i > 1$, а для першого із лишків α_1 значення $v_1 = 1$.

Сума цих чисел $T = \sum_{i=1}^{n} t_i$ має наступні дві властивості [3]. По-перше, лишки цієї суми по всім основам, окрім p_k, завжди дорівнюють лишкам вихідного числа Ă. По-друге, величина цієї суми завжди є меншою ніж величина робочого діапазону T < P, тобто величина T лежить в межах робочого діапазону і для не викривлених чисел T = *A'*.

Неважко помітити, що *процес отримання величини* T= *A' є процесом кодування вихідного* числа ЛУ-кодом, тобто значення *A'* залежить лише від цього вихідного числа і не залежить від невідомої при кодуванні величини лишку по контрольній основі $p_к$. Цей лишок (контрольна ознака, що розшукується) α_k при цьому дорівнює сумі за модулем $p_к$ усіх проміжних величин $\alpha_k^{(i)}$ (i = 1, 2, ..., n) тобто

$$\alpha_k = (\sum_{i=1}^{n} \alpha_k^{(i)}) \,(\mathrm{mod}\, p_к).$$

При декодуванні ж віднімання від числа Ă величини T приводить до того, що отримана різниця

$$\tilde{A} - T = k \cdot P$$

має по всім основам, окрім контрольної, лишки, що дорівнюють нулю, а по контрольній

$$\gamma = (\alpha_k - (T \bmod p_k)) \,(\mathrm{mod}\, p_k) = (k \cdot P) \,(\mathrm{mod}\, p_k),$$

тобто при запису в лишкових класах має вид

$$\tilde{A} - T = (0, 0, \ldots, 0, \ldots 0, (k \cdot P) \,(\mathrm{mod}\, p_k)),$$

де k = 0, 1, 2, …, p_k.

Для не викривлених чисел, тобто при k = 0, величина γ = 0, для викривлених $\gamma \neq 0$. *Таким чином установлюється факт наявності чи відсутності викривлень*.

Для виявлення можливостей алгоритму щодо корегування викривлень нагадаємо:

1) відомий факт, що при $p_к > p_n \cdot p_{n-1}$ між величиною викривлення $\Delta\alpha_i$ і величиною γ є взаємно однозначна відповідність, що дає змогу сподіватися в тому, що отримавши γ можна якимось чином визначити місце і величину викривлення, тобто здійснити її виправлення;

2) на числовій осі величина викривлення $l_i \cdot R_i$ відображається точкою в деякому піддіапазоні "контрольного" діапазону [(P + 1), R); відповідно, процес викривлення початкового числа A відобразиться переміщенням точки A із робочого діапазону [0, P) в деякий інший піддіапазон.

Звернемо увагу на те, що в залежно від величини початкового числа (див. рис. 1), викривлене число (A_1 чи A_2) може попасти в один із суміжних діапазонів із номерами k або (k – 1). Зокрема, при

$$A = A_1 \leq k \cdot P - li \cdot Ri,$$

це буде (в уже прийнятих позначеннях) діапазон ((k – 1)·P, k·P), тобто діапазон із номером (k – 1), а при

$$A = A_2 > k \cdot P - li \cdot Ri$$

це буде діапазон (k·P, (k + 1)·P) , тобто діапазон із номером k.

Внаслідок операції нулізації із числа $A^{/}$, яке контролюється, віднімаються відповідно числа $T = A^{/} - (k - 1) \cdot P < P$, чи $T = A^{/} - k \cdot P < P$. При цьому по контрольній основі $q = p_k$ одержується результат γ такий, що відповідає лівій межі (див. рис. 2) піддіапазону [(k – 1)·P, k·P), тобто величині (k – 1)·P, або ж такий, що відповідає лівій межі піддіапазону [k·P, (k + 1)·P) , тобто величині k·P.

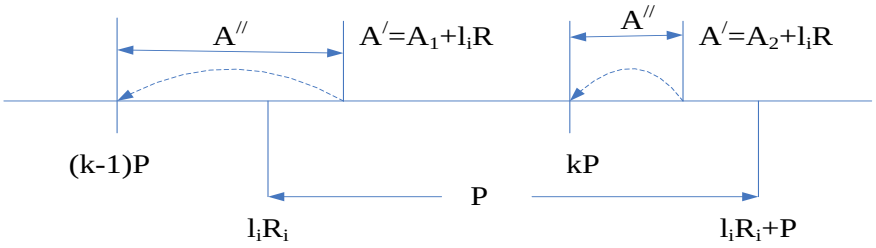

Рисунок 2 – Ілюстрація процесу нулізації

Тобто маємо

$$\gamma = \{k\cdot P\}_q, \text{ або } \gamma = \{(k-1)\cdot P\}_q.$$

Звідси, за правилами СЛК, отримаємо

$$k = \{\gamma/\{P\}_q\}_q, \text{ чи } (k-1) = \{\gamma/\{P\}_q\}_q. \quad (10)$$

Тобто, використовуючи вирази (10) завжди можна визначити номер того діапазону, в який потрапило викривлене число та результат нулізації – число k·P. Оскільки величина викривлення $l_i\cdot R_i$ і результат нулізації k·P є близькими, тобто їх різниця є меншою за величину робочого діапазону P, то це надає принципову можливість визначити місце викривлення.

Оскільки подальші міркування певним чином залежать від можливих співвідношень величин k·P та $l_i\cdot R_i$, розглянемо два наступних випадки.

В першому випадку, при $k\cdot P > l_i\cdot R_i$ значення R_i, яке характеризує величину і місце викривлення можна визначити із очевидної нерівності

$$k\cdot P - [k\cdot P/ R_i]\cdot R_i < P. \quad (11)$$

Підставимо у (11) замість R_i його значення у вигляді

$$R_i = P\cdot q/ p_i.$$

Тоді

$$k\cdot P - [k\cdot P/ Ri]\cdot Ri = k\cdot P - [k\cdot P\cdot p_i / (P\cdot q)]\cdot P\cdot q/ p_i = k\cdot P - [k\cdot p_i /q]\cdot P\cdot q/ p_i < P.$$

Розділивши обидві частини правої сторони останньої нерівності на величину P, отримаємо

$$k - [k\cdot p_i /q]\cdot q/p_i < 1.$$

Помножимо обидві частини останньої нерівності на величину p_i. Одержимо:

$$k\cdot p_i - [k\cdot p_i /q]\cdot q < p_i, \quad (12)$$

або

$$\{k\cdot p_i\}_q < p_i. \quad (13)$$

Звернемо увагу на те, що в (12), (13) вирази в квадратних дужках є не що інше як величина l_i:

$$[k\cdot P/ R_i] = [k\cdot p_i /q] = l_i. \quad (14)$$

Вирази (11) та еквівалентні їм вирази (12), (13) утворюють системи нерівностей по n (i = 1, 2, ..., n) нерівностей в кожній, в яких справедливим є лише одна нерівність для того номера i та значення основи p_i, по якій має місце викривлення.

Таким чином, внаслідок розв'язання будь-якої із систем нерівностей (7) – (13) щодо змінної p_i *місце викривлення стає виявленим*.

Для визначення ж його величини проаналізуємо величини $T = A^{/} - k\cdot P$, чи $T = A^{/} - (k-1)\cdot P$, які формуються по всім лишкам, окрім лишку по контрольній основі в ході операції нулізації числа, яке контролюється.

Як видно на рис. 2, вирази (7) – (13) є справедливими в разі, коли величина викривлення $l_i\cdot R_i < k\cdot P$. В цьому випадку величина сформованого в ході нулізації числа T є меншою вихідного числа A_2 на величину $(k\cdot P - l_i\cdot R_i)$, тобто

$$T = A^{/} - k\cdot P = A_2 - (k\cdot P - l_i\cdot R_i) < A_2,$$

(15)

та

$$\Delta \tilde{A} = (k\cdot P - li\cdot R_i),$$

а величина скорегованого числа має визначатися як:

$$A_2 = T + (k\cdot P - l_i\cdot R_i).$$

Тобто величина скорегованого значення лишку:

$$\alpha_i = \{ \tilde{\alpha}_i + \Delta \tilde{\alpha}_i) = \{ T + (k\cdot P - l_i\cdot R_i)\} \bmod p_i = \{ \tilde{\alpha}_i - \{l_i\cdot R_i\} \bmod p_i\} \bmod p_i,$$

або з урахуванням (14):

$$\alpha_i = \{ \tilde{\alpha}_i - \{[k\cdot p_i /q]\cdot R_i\} \bmod p_i\} \bmod p_i.$$

(16)

Приклад. Нехай в СЛК із основами 2, 3, 5, 17 вихідне число 18_{10} = 0, 0, 3, 1 внаслідок викривлення перетворилося на 0, 0, 0, 1 = 120_{10}.

Результат нулізації дає

$$\Gamma = 0, 0, 0, 1, \gamma = 1.$$

Звідси

$$k = \{\gamma/\{P\}q\}q = \{1/13\}17 = (1 + 3\cdot 17)/13 = 52/13 = 4.$$

Пошук місця викривлення із

$$\{k\cdot p_i\}q < p_i$$

для k = 4 дає

$$\{4\cdot 2\}_{17} < 2 – \text{не вірно},$$

$$\{4\cdot 3\}_{17} < 3 – \text{не вірно},$$

$$\underline{\{4\cdot 5\}_{17} < 5 – \text{вірно}},$$

тобто виявлене викривлення по основі p_3 =5.

Розрахунок скорегованого лишку по основі p_3:

$$\alpha_3 = \{0 - \{[20 /17]\cdot 42\} \bmod 5\} \bmod 5 = 5 - 2 = 3.$$

Видно, що корекція викривлення здійснена правильно.

В іншому випадку, коли результат нулізації – число (k – 1)·P (див. рис. 2) є меншим за величину викривлення $l_i\cdot R_i$, обрахування місця і величини викривлення за виразами (15) – (16) призведе до невірних результатів. Тоді, з урахуванням властивостей операцій в лишкових класах, для визначення місця та величини викривлення слід скористатися виразом:

$$q - \{(k - 1)\cdot p_i\}q < p_i.$$

(17)

В разі вірності цієї нерівності по одній із основ p_i, правомочним є висновок про те, що

$$\gamma = \{(k - 1)\cdot P\}_q,$$

а отже

$$k = \{\gamma/\{P\}_q\}_q + 1. \qquad (18)$$

Як видно на рис. 2, в цьому разі величина викривлення $l_i\cdot R_i > (k - 1)\cdot P$. Тоді величина сформованого в ході нулізації числа T є більшою вихідного числа A_2 на величину $[l_i\cdot R_i - (k - 1)\cdot P]$, тобто

$$T = A^{/} - (k - 1)\cdot P = A_1 + [l_i\cdot R_i - (k - 1)\cdot P] < A_1.$$

(19)

Останній вираз може бути представлений у вигляді

$$T = A^{/} - k\cdot P = A_1 - [(k - 1)\cdot P - l_i\cdot R_i].$$

Неважко помітити, що вирази (15) та (19) є тотожними, якщо вважати, що номер діапазону в обох випадках має значення – k. I, хоча значення викривлення при цьому

$$\Delta \tilde{A} = -(k \cdot P - l_i \cdot R_i),$$

величина скорегованого числа має визначатися, як і раніше, з виразу:

$$A_1 = T + ((k-1) \cdot P - l_i \cdot R_i).$$

Тобто величина скорегованого значення лишку:

$$\alpha_i = \{\tilde{\alpha}_i + \Delta\alpha_i\} = \{T + ((k-1) \cdot P - l_i \cdot R_i)\} \bmod p_i = \{\tilde{\alpha}_i - \{l_i \cdot R_i\} \bmod p_i\} \bmod p_i,$$

або з урахуванням (16) отримаємо, як і раніше,

$$\alpha_i = \{\tilde{\alpha}_i - \{[k \cdot p_i / q] \cdot R_i\} \bmod p_i\} \bmod p_i.$$

IV Узагальнені лишково – Хеммінгові (ЛХ) коди

<u>У лишково – Хеммінгових (ЛХ) кодах</u> двійкові базові кодові слова, розбиті на b-розрядні узагальнені символи, записуються у вигляді $\alpha_1, \alpha_2, \ldots \alpha_n$, де $\alpha_i \le 2^b - 1$, а $N = b \cdot n$. Так само, як і в двійковому коді Хеммінга (класична форма запису коду) узагальнені символи α_i з номерами $i = 2^j$ ($j = 0, 1, \ldots$) є перевірочними, решта символів – інформаційні. Причому для отримання перевірочних символів при кодування використовується алгоритм, аналогічний алгоритму для двійкового коду Хеммінга, але відносно до узагальнених символів. При цьому всі необхідні для кодування і декодування операції здійснюються за деяким модулем. Тобто, в ЛХ-коді для отримання першого перевірочного символу необхідно скласти по деякому модулю (одержати лишки від суми) всі узагальнені символи базового кодового слова, що мають в коді свого номера одиницю в першому (молодшому) розряді; для отримання другого перевірочного символу – скласти по модулю усі символи, що мають в коді свого номера одиницю в другому розряді і т. д.

Як модуль для отримання контрольних символів досить зручно використовувати величину $s = 2^b$, тобто

$$\alpha_1 = \{\alpha_3 + \alpha_5 + \alpha_7 + \ldots\ldots\ldots\ldots\}_s,$$
$$\alpha_2 = \{\alpha_3 + \alpha_5 + \alpha_6 + \alpha_7 + \ldots\ldots\ldots\}_s,$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$

При такому значенні модуля потрібна розрядність перевірочних символів не відрізняється від розрядності узагальнених символів b.

При декодування зберігається той же алгоритм розрахунку перевірочних α_i символів, що і при кодуванні, але при додаванні за модулем використовуються і контрольні символи. Знов одержані перевірочні символи порівнюються з відповідними перевірочними символами, обчисленими при кодуванні. При їх відповідності робиться висновок про відсутність викривлення, в решті випадків – про наявність викривлення.

Якщо приписати результатам порівняння значення 0, а результатам не порівняння – значення 1, то одержана сукупність нулів і одиниць утворює код, який також, як і в двійковому коді Хеммінга, є номером викривленого символу.

Приклад. Хай необхідно закодувати ЛХ-кодом восьмирозрядну ($N = 8$) послідовність 10001101. Якщо код орієнтований на виправлення двократних викривлень, то $b = 2$, кількість узагальнених символів $n = N/b = 4$. Як модуль для отримання контрольних символів використаємо величину $s = 4$. Відомо, що в коді Хеммінга при $n = 4$ потрібно три перевірочні символи $\alpha_1, \alpha_2, \alpha_4$, а інформаційними символами є $\alpha_3 = 10$, $\alpha_5 = 00$, $\alpha_6 = 11$, $\alpha_7 = 01$ Для отримання першого перевірочного символу складемо по модулю чотири $\alpha_3, \alpha_5, \alpha_7$.

$$\alpha_1 = \{\alpha_3 + \alpha_5 + \alpha_7\}_4 = 11$$

Аналогічно цьому

$$\alpha_2 = \{\alpha_3 + \alpha_6 + \alpha_7\}_4 = 10,$$
$$\alpha_4 = \{\alpha_5 + \alpha_6 + \alpha_7\}_4 = 00.$$

Після кодування одержано код

$$11.\ 10.\ 10.\ 00.\ 00.\ 11\ 01 = \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7,$$

який має бути записаним у запам'ятовуючий пристрій (ЗП), переданим в канал зв'язку і т. д.

Хай зчитаний або прийнятий з каналу зв'язку код має викривлення у п'ятій групі:

$$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \acute{\alpha}_5, \alpha_6, \alpha_7 = 11.10.10.00.01.11.01.$$

Після обчислення нових контрольних символів, одержимо

$$\alpha_1 = \{\alpha_3 + \acute{\alpha}_5 + \alpha_7\}_4 = 00$$
$$\alpha_2 = \{\alpha_3 + \alpha_6 + \alpha_7\}_4 = 10,$$

$$\alpha_4 = \{\acute{\alpha}_5 + \alpha_6 + \alpha_7\}_4 = 01.$$

Результати порівняння дадуть код 101, оскільки перший та третій перевірочні символи не співпадають. Це свідчить про виявлення помилки в п'ятому символі, що і було насправді.

Неважко визначити і величину викривлення. Дійсно, будь-який з перевірочних символів, наприклад α_i, при викривленні деякого інформаційного, наприклад α_j, що приймає участь у формуванні символу α_i, має величину

$$\acute{\alpha}_i = \{\alpha_c + \alpha_d + \ldots + \{\alpha_j + \Delta\alpha_j\} + \ldots\}_s = \{\alpha_i + \Delta\alpha_j\}_s.$$

Звідки

$$\Delta\alpha_j = \{\acute{\alpha}_i - \alpha_i\}_s. \qquad (20)$$

Для вищерозглянутого прикладу

$$\Delta\alpha_5 = \{\acute{\alpha}_1 - \alpha_1\}_4 = \{00 - 11\}_4 = 01,$$

або

$$\Delta\alpha_5 = \{\acute{\alpha}_4 - \alpha_4\}_4 = \{01 - 00\}_4 = 01.$$

Знаючи величину ($\acute{\alpha}_i$) і місце викривлення (i), легко здійснити корекцію, оскільки з (20) маємо

$$A_i = \{\acute{\alpha}_i - \Delta\alpha_j\}_s.$$

У нашому прикладі

$$\alpha_5 = \{\acute{\alpha}_5 - \Delta\alpha_5\}_4 = \{01 - 01\}_4 = 00,$$

що і є насправді.

Алгоритм декодування ЛХ- коду може бути ще спрощеним, якщо при кодуванні замість перевірочних символів α_i в записану або передану послідовність записувати величину

$$\Delta\alpha_i = \{s - \Delta\alpha_j\}_s.$$

Тоді для вже розглянутого прикладу (α_1 =11, α_2 =10, α_4 = 00) $\Delta\alpha_1$ = 01, $\Delta\alpha_2$ = 10, $\Delta\alpha_4$ = 00 і записувати (передавати) необхідно код:

$$\Delta\alpha_1, \Delta\alpha_2, \alpha_3, \Delta\alpha_4, \alpha_5, \alpha_6, \alpha_7 = 01.\ 10.\ 10.\ 00.\ 00.\ 11.\ 01.$$

Якщо зчитано або прийнято слово з тим же викривленням, що і раніше, тобто

$$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \acute{\alpha}_5, \alpha_6, \alpha_7, = 01.10.10.00.01.11.01,$$

то після декодування отримаємо

$$\Delta\alpha_1 = \{\alpha_1 + \alpha_3 + \acute{\alpha}_5 + \alpha_7\}_4 = 01,$$
$$\Delta\alpha_2 = \{\alpha_2 + \alpha_3 + \alpha_6 + \alpha_7\}_4 = 00,$$
$$\Delta\alpha_4 = \{\alpha_4 + \acute{\alpha}_5 + \alpha_6 + \alpha_7\}_4 = 01.$$

При цьому, якщо відмінним від нуля перевірочним символам приписати значення 1, а іншим – код 0, то одержимо код i = 101, що визначає місце викривлення, величина якого дорівнює значенню будь-якого ненульового перевірочного символу. Для розглянутого прикладу величина викривлення $\Delta\alpha_i$ = 01, корекція якого нескладна.

Ще більш простими в технічній реалізації алгоритмами кодування – декодування ЛХ- коду є, на думку автора, алгоритм, в яких замість операцій додавання за модулем s пропонується використовувати операції посимвольного додавання за модулем 2. Решту операцій алгоритмів кодування – декодування пропонується залишити такими ж, як і вище.

Приклад. Нехай, як і вище для виявлення і виправлення дворозрядних викривлень потрібно закодувати ЛХ-кодом восьмирозрядну (N = 8) послідовність 10001101. Тоді b = 2, n = N/b = 4. При цих умовах потрібно три перевірочні символи α_1, α_2, α_4, а інформаційними символами є α_3 = 10, α_5 = 00, α_6 = 11, α_7 = 01. Для отримання першого перевірочного символу складемо α_3, α_5, α_7 не за модулем s = 4, а порозрядно за модулем два.

$$\alpha_1 = \{\alpha_3 + \alpha_5 + \alpha_7\}_2 = 11$$

Аналогічно цьому

$$\alpha_2 = \{\alpha_3 + \alpha_6 + \alpha_7\}_2 = 00,$$
$$\alpha_4 = \{\alpha_5 + \alpha_6 + \alpha 7\}_2 = 10.$$

Після кодування одержано код

$$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7 = 11.\ 00.\ 10.\ 10.\ 00.\ 11\ 01$$

Нехай зчитаний або прийнятий з каналу зв'язку код має викривлення у п'ятій групі:

$$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \acute{\alpha}_5, \alpha_6, \alpha_7 = 11.00.10.10.01.11.01.$$

Після обчислення нових контрольних символів, одержимо

$$\alpha_1 = \{\alpha_3 + \acute{\alpha}_5 + \alpha_7\}_2 = 10$$
$$\alpha_2 = \{\alpha_3 + \alpha 6 + \alpha_7\}_2 = 00,$$
$$\alpha_4 = \{\acute{\alpha}_5 + \alpha_6 + \alpha_7\}_2 = 11.$$

Результати порівняння дадуть код 101, оскільки перший та третій перевірочні символи не співпадають. Це свідчить про виявлення помилки в п'ятому символі, що і було насправді.

Величина викривлення легко знаходиться шляхом порозрядного додавання за модулем 2 одного зі знову розрахованих ненульових контрольних символів (в цьому прикладі – це перший чи третій контрольний символ $\alpha_1 = 10$, $\alpha_4 = 11$) із його переданим значенням (в цьому прикладі – $\alpha_1 = 11$ чи $\alpha_4 = 10$). Тоді при використанні перших контрольних символів величина викривлення $\Delta\alpha_i = \{10 + 11\}_2 = 01$, а при використанні третіх контрольних символів $\Delta\alpha_i = \{11 + 10\}_2 = 01$. Видно, що одержано однакові значення викривлень, отже в подальшому можна обмежуватися використанням лише однієї пари контрольних символів, наприклад, першої.

Для корекції викривлення пропонується здійснити порозрядне додавання за модулем 2 викривленого символу, місце якого вже встановлено, з величиною викривлення $\Delta\alpha_i$.

У нашому прикладі $\alpha_5 = \{\acute{\alpha}_5 + \Delta\alpha_5\}_2 = \{01 + 01\}_2 = 00$, що і є насправді.

Приклад. Представимо послідовність 000000110000011000100000 у вигляді коду з довжиною символів 2 біта. Такий код буде орієнтований на виправлення двократних викривлень, b = 2, а кількість узагальнених символів m = N/b = 24/2 =12.

В цьому випадку,послідовність без урахування перевірочних символів матиме наступний вигляд:

$$\alpha_1\ \alpha_2\ \alpha_3\ \alpha_4\ \alpha_5\ \alpha_6\ \alpha_7\ a_8\ \alpha_9\ a_{10}\ a_{11}\ a_{12}\ a_{13}\ a_{14}\ a_{15}\ a_{16}\ a_{17} =$$
$$= \alpha_1\ \alpha_2\ 00.\ \alpha_4\, 00.\ 00.\ 11.a_8\, 00.\ 00.\ 01.10.00.10.00.\ a_{16}\, 00$$

Для визначення величин контрольних символів α_1, α_2, α_4, a_8, a_{16} формуємо систему перевірочних рівнянь:

$$\alpha_1 = \{\alpha_3 + \alpha_5 + \alpha_7 + a_9 + a_{11} + a_{13} + a_{15} + a_{17}\}_2 = 10,$$
$$\alpha_2 = \{\alpha_3 + \alpha_6 + \alpha_7 + a_{10} + a_{11} + a_{14} + a_{15}\}_2 = 00,$$
$$\alpha_4 = \{\alpha_5 + \alpha_6 + \alpha_7 + a_{12} + a_{13} + a_{14} + a_{15}\}_2 = 11,$$
$$a_8 = \{\alpha_9 + a_{10} + a_{11} + a_{12} + a_{13} + a_{14} + a_{15}\}_2 = 01,$$
$$a_{16} = \{a_{17}\}_2 = 00.$$

В наслідок цього держимо код:

10.00.00.11.00.00.11.01.00.00.01.10.00.10.00.00.00

Нехай зчитаний або прийнятий з каналу зв'язку код має викривлення у одинадцятій групі a_{11}, тобто

$$\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, a_8, a_9, a_{10}, a'_{11}, a_{12}, a_{13}, a_{14}\ a_{15}, a_{16}, a_{17} =$$
$$= 10.00.00.11.00.00.11.01.00.00.11.10.00.10.00.00.00$$

Системи синдромних рівнянь матимуть вигляд:

$$\alpha'_1 = \{\alpha_3 + \alpha_5 + \alpha_7 + a_9 + a_{11} + a_{13} + a_{15} + a_{17}\}_2 = 11,$$
$$\alpha'_2 = \{\alpha_3 + \alpha_6 + \alpha_7 + a_{10} + a_{11} + a_{14} + a_{15}\}_2 = 01,$$
$$\alpha'_4 = \{\alpha_5 + \alpha_6 + \alpha_7 + a_{12} + a_{13} + a_{14} + a_{15}\}_2 = 11,$$
$$a'_8 = \{\alpha_9 + a_{10} + a_{11} + a_{12} + a_{13} + a_{14} + a_{15}\}_2 = 10,$$
$$a'_{16} = \{a_{17}\}_2 = 00.$$

Результати порівнянь дадуть код 11010, оскільки перший, другий і четвертий перевірочні символи не співпадають. Це свідчить про те, що виявлено помилку в 11-ому символі. Неважко визначити і величину викривлення

$$\Delta a_{11} = \{a'_{11} - a_{11}\}_2 = \{01+11\}_2 = 10,$$

а отже здійснити його корекцію: $\alpha_{11} = \{\acute{\alpha}_{11} + \Delta\alpha_{11}\}_2 = \{11 + 10\}_2 = 01$.

З розглянутих прикладів видно, що остання модифікація лишково-Хеммінгового коду дає можливість виявлення та виправлення групових (пакетних) викривлень алгоритмів при більш простих для апаратурної реалізації алгоритмах кодування -декодування.

Висновок

Застосування розглянутих алгоритмів кодування – декодування узагальнених кодів дозволяє забезпечити виявлення та виправлення викривлень в b − розрядних узагальнених символах в кожному із базових кодових слів. З урахуванням перемежування глибиною λ довжина пакетів викривлень в узагальнених кодових словах, які можуть бути виправленими, може дорівнювати λ·b двійкових символів. Застосування таких кодів, на погляд автора, дозволить розв'язати сформульовану проблему щодо надійного забезпечення цілісності інформаційних об'єктів в умовах впливу пакетів викривлень значної тривалості.

Література: **1**. *НД ТЗІ 2.5-005-99 "Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу"*; **2**. *Василенко В. С.*

*Узагальнені завадостійкі коди в задачах забезпечення цілісності інформаційних об'єктів в умовах природних впливів. К. НТУУ "КПІ" // Правове, нормативне та метрологічне забезпечення Системи захисту інформації в Україні. Випус*к 2 (13)// 2006, с. 144–159. **3.** *Акушский И. Я., Юдицкий Д. И. Машинная арифметика в остаточных классах. // М.: Сов. радио*, 1966. – 421 с. **4.** *Василенко В. С., Будько М. М., Короленко М. П. Контроль и відновлення цілісності інформації в автоматизованих системах. К. НТУУ "КПІ" // Правове, нормативне та метрологічне забезпечення Системи захисту інформації в Україні. Випус*к 4// 2002, с. 119–128.

УДК 004.658.2

АУДИТ АНОМАЛЬНОГО ПОВЕДЕНИЯ ПОЛЬЗОВАТЕЛЕЙ БАЗ ДАННЫХ

Михаил Коломыцев, Светлана Носок
Национальный технический университет Украины «Киевский политехнический институту»

Анотації: **Вирішується задача розробки підходу до формування профілю типової поведінки, що враховує специфіку функціонування СУБД. Пропонується процедура виявлення аномальної поведінки на базі створеного профілю.**
Summary: **The task of developing an approach to typical behaviour profile that accounts for the database management system functioning pecularities is covered in his work. The procedure of anomalous behaviour detection on the basis of the developed profile is suggested.**
Ключові слова: **Аномальна поведінка, профіль користувача, байєсовський класифікатор.**

I Введение

Современные СУБД обладают развитыми возможностями обеспечения безопасности данных. Однако, в силу ряда причин, обусловленных сложностью информационных систем и изменчивостью обстановки их функционирования, становится возможным нарушение установленной политики безопасности. Зачастую [1], нарушителями политики безопасности становятся инсайдеры. Защита информации от инсайдеров является сложной задачей, поскольку они являются легитимными пользователями, действующими в рамках предоставленных им полномочий. В таких ситуациях особенное значение приобретает способность системы безопасности СУБД обнаруживать изменение в поведении пользователей, сигнализирующее о попытках злоумышленных действий. Преимуществом технологии обнаружения атак на сервер БД, основанной на обнаружении аномальной активности, в отличие от подхода с использованием сигнатур, является большая гибкость и возможность обнаруживать неизвестные атаки. В данной работе предлагается подход к обнаружению аномального поведения пользователей в базах данных, использующих ролевое управление доступом.

II Постановка задачи

Системы обнаружения аномального поведения основаны на том, что известны некоторые параметры, характеризующие правильное или допустимое поведение объекта наблюдения. В качестве таких параметров могут выступать, например, количественные показатели использования ресурсов сервера БД [2] или интенсивности обращений к ресурсам [5].Значения параметров, соответствующие нормальному поведению объекта наблюдения называется профилем. Выявления аномального поведения основано на сравнении текущих значений параметров активности с профилем. Параметры профиля вычисляются за достаточно большой период времени. Под текущими значениями параметров активности обычно понимаются значения, вычисленные на коротком интервале времени (применительно к СУБД – по одной или нескольким транзакциям), непосредственно предшествующем рассматриваемому моменту.

Хотя аномальное поведение не обязательно является следствием атаки на систему, с высокой долей вероятности оно свидетельствует о нарушении (умышленным или нет) политики безопасности. Поскольку действия злоумышленника обязательно чем-то отличаются от поведения обычного пользователя, в работах [3,4] методы обнаружения аномального поведения положены в основу систем обнаружения вторжений в базу данных.