

17. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. Detecting Hidden Information in FAT. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
18. Smirnov, O., Kuznetsov, A., Kuznetsova, K. Synthesis of Discrete Signals with Improved Correlation Properties. Монографія: In.: ISCI'2019: Information Security in Critical Infrastructures. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
19. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у Кібербезпека та інформаційні технології: монографія. – Х.: ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
20. Смірнов О.А., Дреева Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка. – Х.: Вид. Рожко С.Г. 2019. С. 123-139.

УДК 004

І.Мицак, магістр гр. КН-21М-1,4,

Центральноукраїнський національний технічний університет

ДОСЛІДЖЕННЯ ТА ПРОГРАМНА РЕАЛІЗАЦІЯ СИСТЕМИ ЗАБЕЗПЕЧЕННЯ ЦІЛІСНОСТІ ДАНИХ У ХМАРНИХ СЕРВІСАХ

У статті розроблено програмне забезпечення, яке призначено для системи забезпечення цілісності даних у хмарних сервісах. Метою розробки є дослідження та програмна реалізація системи забезпечення цілісності даних у хмарних сервісах. Об'єктом дослідження є процес забезпечення цілісності даних у хмарних сервісах. Предметом дослідження є методи забезпечення цілісності даних у хмарних сервісах. Методи дослідження базуються на методах забезпечення цілісності даних, методах математичної статистики, методах розробки програмного забезпечення. Результат роботи – програмна реалізація системи забезпечення цілісності даних у хмарних сервісах. В процесі роботи над програмною моделлю виконано аналіз існуючих апаратних та програмних засобів. В повній мірі описані всі компоненти розробленого програмного забезпечення.

комп'ютерні науки, цілісність даних, хмарні сервіси

Постановка проблеми. Сучасні хмарні сервіси – це ефективний засіб зберігання й передачі даних (у них найбільш низька удільна вартість зберігання інформації), потреби ринку дуже великі, і в продаж найчастіше надходять хмарні сервіси, що не відповідають закладеним у формат стандартам. Виробники допускають таке зниження якості лише тому, що коригувальна здатність, закладена в хмарних сервісах досить велика, і навіть не дуже якісний хмарний сервіс, швидше за все, буде й записуватися й зчитуватися, нехай і на знижених швидкостях.

Але основною проблемою такої “другосортності” є те, що “запас міцності” системи корекції помилок хмарних сервісів при роботі з ними вкрай низький, і будь-яка, навіть незначна подряпина на диску серверу хмарного сервісу може ушкодити дані. Імовірність збою збереженого файлу прямо пропорційна кількості займаних їм секторів, і якщо файл – це архів, що займає весь хмарний сервіс, для надійного архівного зберігання інформації потрібно дуже якісний носій.

Існує цілий ряд методів, які дозволяють вирішити цю задачу. Але посеред цього ряду особливо стоять методи завадостійкого кодування, які дозволяють, навіть при досить сильному ушкодженні тексту відтворити його.

Посеред методів завадостійкого кодування в останній час дуже активно розвиваються методи, які засновані на використанні циклічних кодів. Беззаперечною перевагою цих методів є дуже велика здатність виправляти помилки в ушкодженному тексті. Саме до таких

методів й відноситься метод Ріда-Соломона. Вони, у цей час широко використовуються в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення у випадку ушкоджень, у завадостійкому кодуванні.

Аналіз останніх досліджень і публікацій. При аналізі останніх досліджень і публікацій [1-10] було виявлено певні прогалини у забезпеченні системи забезпечення цілісності даних у хмарних сервісах.

Мета й завдання дослідження. Метою роботи є дослідження та програмна реалізація системи забезпечення цілісності даних у хмарних сервісах.

Для досягнення поставленої мети визначена програма дослідження, що складається з наступних завдань:

- Огляд існуючих систем забезпечення цілісності даних у хмарних сервісах.
- Дослідження системи забезпечення цілісності даних у хмарних сервісах.
- Програмна реалізація системи забезпечення цілісності даних у хмарних сервісах.

Об'єктом дослідження є процес забезпечення цілісності даних у хмарних сервісах.

Предметом дослідження є методи забезпечення цілісності даних у хмарних сервісах.

Методи дослідження базуються на методах забезпечення цілісності даних, методах математичної статистики, методах розробки програмного забезпечення.

Виклад основного матеріалу. Опис коду Ріда-Соломона

Коди Ріда-Соломона – недвійкові циклічні коди, що дозволяють виправляти помилки в блоках даних. Елементами кодового вектора є не біти, а групи біт (блоки). Дуже поширені коди Ріда-Соломона, що працюють із байтами (октетами).

У цей час широко використовується в системах відновлення даних з компакт-дисків, при створенні архівів з інформацією для відновлення у випадку ушкоджень, у завадостійкому кодуванні.

Код Ріда-Соломона був винайдений в 1960 році співробітниками лабораторії Лінкольна Массачусетського технологічного інституту Ірвином Рідом і Густавом Соломоном. Ідея використання цього коду була представлена в статті «Polynomial Codes over Certain Finite Fields». Перше застосування код Ріда-Соломона одержав в 1982 році в серійному випуску компакт-дисків. Ефективний алгоритм декодування був запропонований в 1969 році Елвином Берлекемпом і Джеймсом Мессі.

Формальний опис

Коди Ріда-Соломона є важливим частковим випадком БЧХ-коду, корінь полінома, що породжує, якого лежать у тім же полі, над яким і будується код ($m = 1$).

Коди Боуза-Чоудхури-Хоквінгхема (БЧХ коди) – у теорії кодування це широкий клас циклічних кодів, застосовуваних для захисту інформації від помилок. Відрізняється можливістю побудови коду із заздалегідь певними коригувальними властивостями, а саме, мінімальною кодовою відстанню.

Поліном, що породжує

Визначення поліномом, що породжує, циклічного (n, k) коду C називається такий ненульовий $g(x) = \sum_{i=0}^r g_i x^i$ поліном з C , ступінь якого найменша й коефіцієнт при старшому ступені $g_r = 1$.

Теорема 1

Якщо C – циклічний (n, k) код і $g(x)$ – його поліном, що породжує, тоді *ступінь* $g(x)$ дорівнює $r = n - k$ і кожне кодове слово може бути єдиним чином представлено у вигляді $c(x) = m(x)g(x)$, де ступінь $m(x)$ менше або дорівнює $k - 1$.

Теорема 2

$g(x)$ – поліном, що породжує, циклічного (n, k) коду є дільником двочлена $x^n - 1$

Наслідок: у такий спосіб як поліном, що породжує, можна вибирати будь-який поліном, дільник $x^n - 1$. Ступінь обраного полінома буде визначати кількість перевірочних символів r , число інформаційних символів $k = n - r$.

Матриця, що породжує

Поліноми $g(x), xg(x), x^2g(x), \dots, x^{k-1}g(x)$ лінійно незалежні, інакше $m(x)g(x) = 0$ при ненульовому $m(x)$, що неможливо.

Значить кодові слова можна записувати, як і для лінійних кодів, наступним чином:

$$\bar{m}G = (m_0, m_1, \dots, m_{k-1}) \begin{bmatrix} g(x) \\ xg(x) \\ \dots \\ x^{k-1}g(x) \end{bmatrix} = m(x)g(x), \quad (1)$$

де G є матрицею, що породжує, $m(x)$ – інформаційним поліномом.

Матрицю G можна записати в символній формі:

$$G = \begin{bmatrix} g_0 & g_1 & \dots & g_{r-1} & g_r & 0 & \dots & 0 \\ 0 & g_0 & \dots & g_{r-2} & g_{r-1} & g_r & \dots & 0 \\ \cdot & \cdot & \dots & \cdot & \cdot & \cdot & \dots & \cdot \\ 0 & 0 & \dots & 0 & g_0 & g_1 & \dots & g_r \end{bmatrix}$$

Перевірочна матриця

Для кожного кодового слова циклічного коду справедливо $c(x) = 0 \pmod{g(x)}$.

Тому $H = \begin{bmatrix} 1 & x & x^2 & \dots & x^{n-2} & x^{n-1} \end{bmatrix} \pmod{g(x)}$ можна записати як

Тоді:

$$\bar{c}H^T = \sum_{i=0}^{n-1} c_i x^i \pmod{g(x)}. \quad (2)$$

Нехай α – елемент поля $GF(q)$ порядку n . Якщо α – примітивний елемент, то його порядок дорівнює $q-1$, т.е. $\alpha^{q-1}=1, \alpha^i \neq 1, 0 < i < q-1$.

Тоді нормований поліном $g(x)$ мінімального ступеня над полем $GF(q)$, коріннями якого є $d-1$ підряд, що йдуть ступенів, $\alpha^{l_0}, \alpha^{l_0+1}, \dots, \alpha^{l_0+d-1}$ елемента α , є поліномом, що породжує, коду над полем $GF(q)$ $g(x) = (x - \alpha^{l_0})(x - \alpha^{l_0+1}) \dots (x - \alpha^{l_0+d-1})$; де l_0 – деяке ціле число (у тому числі 0 і 1), за допомогою якого іноді вдається спростити кодер. Звичайно покладається $l_0 = 1$. Ступінь багаточлена $g(x)$ дорівнює $d-1$.

Довжина отриманого коду n , мінімальна відстань d (мінімальна відстань d лінійного коду є мінімальним із всіх відстаней Хеммінга всіх пар кодових слів). Код містить $r = d - 1 = \deg(g(x))$ перевірочний символ, де $\deg()$ позначає ступінь полінома; число інформаційних символів $k = n - r = n - d + 1$. У такий спосіб $d = n - k - 1$ і код Ріда-Соломона є роздільним кодом з максимальною відстанню (є оптимальним у змісті границі Синглтона).

Кодовий поліном $c(x)$ може бути отриманий з інформаційного полінома $m(x)$, $\deg m(x) \leq k - 1$, шляхом перемноження $m(x)$ і $g(x)$: $c(x) = m(x)g(x)$

Властивості

Код Ріда-Соломона над $GF(q^m)$, що виправляє t помилок, вимагає $2t$ перевірочних символів і з його допомогою виправляються довільні пакети довжиною t і менше. Відповідно до теореми про границю Рейгера, коди Ріда-Соломона є оптимальними з погляду співвідношення довжини пакета й можливості виправлення помилок – використовуючи $2t$ додаткових перевірочних символів виправляються t помилок (і менш).

Теорема (границя Рейгера). Кожний лінійний блоковий код, що виправляє всі пакети довжиною t і менш, повинен містити щонайменше $2t$ перевірочних символів.

Виправлення багаторазових помилок

Код Ріда-Соломона є одним з найбільш потужних кодів, що виправляють багаторазові пакети помилок. Застосовується в каналах, де пакети помилок можуть утворюватися настільки часто, що їх уже не можна виправляти за допомогою кодів, що виправляють одиночні помилки. $(q^m - 1, q^m - 1 - 2t)$ -код Ріда-Соломона над полем $GF(q^m)$ з кодовою відстанню $d = 2t + 1$ можна розглядати як $((q^m - 1)m, (q^m - 1 - 2t)m)$ -код над полем $GF(q)$, що

може виправляти будь-яку комбінацію помилок, зосереджену в t або меншому числі блоків з m символів.

Найбільше число блоків довжини m , які може торкнутися пакет довжини l_i , де $l_i \leq mt_i - (m - 1)$, не перевершує t_i , тому код, що може виправити t блоків помилок, завжди може виправити й будь-яку комбінацію з r пакетів загальної довжини l , якщо $l + (m - 1) \leq mt$.

Практична реалізація

Кодування за допомогою коду Ріда-Соломона може бути реалізовано двома способами: систематичним і несистематичним.

При несистематичному кодуванні інформаційне слово множиться на якийсь поліном, що неприводиться, у полі Галуа. Отримане закодоване слово повністю відрізняється від вихідного й для добування інформаційного слова потрібно виконати операцію декодування й уже потім можна перевірити дані на зміст помилок. Таке кодування вимагає більші витрати ресурсів тільки на добування інформаційних даних, при цьому вони можуть бути без помилок.

При систематичному кодуванні до інформаційного блоку з k символів приписуються $2t$ перевірочних символів, при обчисленні кожного перевірочного символу використовуються всі k символів вихідного блоку.

У цьому випадку немає витрат ресурсів при добуванні вихідного блоку, якщо інформаційне слово не містить помилок, але кодер/декодер повинен виконати $k(n - k)$ операцій додавання й множення для генерації перевірочних символів. Крім того, тому що всі операції проводяться в полі Галуа, те самі операції кодування/декодування вимагають багато ресурсів і часу. Швидкий алгоритм декодування, заснований на швидкому перетворенні Фур'є, виконується за час порядку $l \ln n^2$.

Кодування

При операції кодування інформаційний поліном множиться на багаточлен, що породжує. Множення вихідного слова S довжини k на не приводиться поліном, що, при систематичному кодуванні можна виконати в такий спосіб:

– До вихідного слова приписуються $2t$ нулів, виходить поліном $T = Sx^{2t}$.

– Цей поліном ділиться на поліном, що породжує, G , перебуває залишок R , $Sx^{2t} = QG + R$, де Q – частка.

– Цей залишок й буде коригувальним кодом Ріда-Соломона, він приписується до вихідного блоку символів. Отримане кодове слово $C = Sx^{2t} + R$.

Кодер будується зі регістрів зсуву, суматорів і перемножувачів. Регістр зсуву складається з комірок пам'яті, у кожній з яких перебуває один елемент поля Галуа.

Наведений як приклад кодер Ріда-Соломона генерує 16 коригувальних байт, що дозволяє виправляти до 8 і виявляти до 16 помилок у кадрі даних.

Перемножувачі на константи $GF(0) \dots GF(15)$ у полі Галуа реалізуються в такий спосіб: спочатку вихідне число й константа перетворюються в індексну форму, потім складаються в межах байта без обліку переносу.

Результатом операції є результат додавання, перетворена обернено в поліноміальну форму.

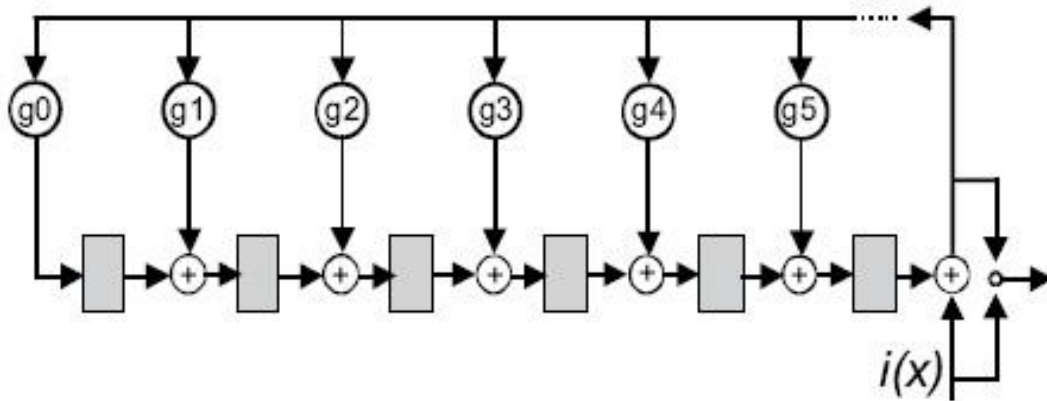


Рисунок 1 – Функціональна схема кодера Ріда-Соломона

При переході від однієї форми подання даних до іншої доцільно використовувати таблицю істинності розміром 256 байт, що становить ємність одного ЕАВ (Embedded Array Block – блок зосередженої пам'яті). Для реалізації кодера потрібно 16 таких перемножувачів, при цьому те саме число множиться на різні константи, що дозволяє використовувати для його перекладу в індексну форму один ЕАВ.

Для перекладу результатів у поліноміальну форму потрібно вже 16 таких таблиць, що вимагає застосування ІМС FPGA дуже великої ємності. У запропонованій схемі використовується тактування кодера із частотою, в 8 разів перевищуючу частоту надходження байт даних.

Це дає можливість використовувати дві пари «суматор – ЕАВ», мультиплексує константи на входах суматорів і дозволяючи роботу регістрів-накопичувачів у моменти появи відповідних даних на виходах засувки ЕАВ.

На структурній схемі кодера (рисунок 2) символу «С» відповідають дві константи $GF(n)$.

Символ «L» у логіці регістрів-накопичувачів відповідає наступний: вихід компаратора нуля (символи CMP0 і SYNC) дозволяє роботу схеми «АБО що виключає», на входи якої подаються вихід попереднього регістра й ЕАВ. Якщо ж вектор зворотного зв'язку дорівнює "0", схема пропускає дані з виходу попереднього регістра-накопичувача на вхід наступного.

У результаті кодер з урахуванням схеми синхронізації (на рисунку не показана) займає 255 LE (Logic Element – логічний елемент) і 3 ЕАВ, що дозволяє розмістити його в ІМС EPF10K10. Після оптимізації розміщення схеми на кристалі FPGA швидкодія схеми досягла 11,57 МГц (частота надходження байт даних, далі – байтова частота).

При використанні ІМС EPF10K20, у складі якої 6 ЕАВ, використовуючи 4 пари "суматор – ЕАВ", можна тактувати кодер із частотами, що перевищують байтову частоту не в 8, а в 4 рази, що дозволить підняти її до 25...30 МГц.

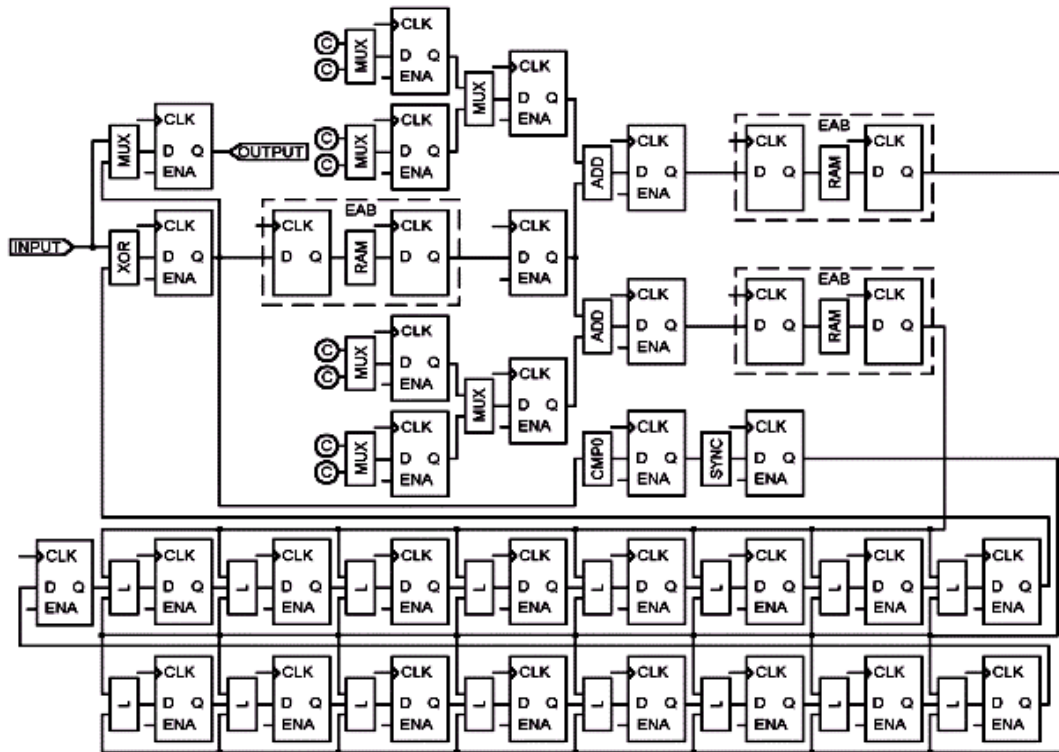


Рисунок 2 – Структура кодера Ріда-Соломона

Декодування

Декодер, що працює по авторегресивному спектральному методі декодування, послідовно виконує наступні дії:

- Обчислює синдром помилки.
- Будує поліном помилки.
- Знаходить корінь даного полінома.
- Визначає характер помилки.
- Виправляє помилки.

Обчислення синдрому помилки

Обчислення синдрому помилки виконується синдромним декодером, що ділить кодове слово на багаточлен, що породжує. Якщо при діленні виникає остача, то в слові є помилка. Остача від ділення є синдромом помилки.

Побудова полінома помилки

Обчислений синдром помилки не вказує на положення помилок. Ступінь полінома синдрому дорівнює $2t$, що багато менше ступеня кодового слова n . Для одержання відповідності між помилкою і її положенням у повідомленні будується поліном помилок.

Поліном помилок реалізується за допомогою алгоритму Берлекемпа-Мессі, або за допомогою алгоритму Евкліда. Алгоритм Евкліда має просту реалізацію, але вимагає більших витрат ресурсів. Тому частіше застосовується більше складний, але менш затратоємний алгоритм Берлекемпа-Мессі. Коефіцієнти знайденого полінома безпосередньо відповідають коефіцієнтам помилкових символів у кодовому слові.

Алгоритм Евкліда виконується наступним чином.

Нехай a і b суть цілі числа, не рівні одночасно нулю, і послідовність чисел $a, b, r_1 > r_2 > r_3 > r_4 > \dots > r_n$ визначена тим, що кожне r_k це остача від ділення передпопереднього числа на попереднє, а передостаннє ділиться на останнє націло, тобто

$$\begin{aligned} a &= bq_0 + r_1 \\ b &= r_1q_1 + r_2 \\ r_1 &= r_2q_2 + r_3 \end{aligned} \quad (3)$$

$$r_{n-1} = r_n q_n$$

Тоді (a,b) , найбільший загальний дільник a і b , дорівнює r_n , останньому ненульовому члену цієї послідовності.

Існування таких r_1, r_2, \dots , тобто можливість ділення з остачею m на n для будь-якого цілого m і цілого $n \neq 0$, доводиться індукцією по m .

Коректність цього алгоритму впливає з наступних двох тверджень:

– Нехай $a = bq + r$, тоді $(a,b) = (b,r)$.

– $(0,r) = r$. для будь-якого ненульового r .

Знаходження корня

На цьому етапі шукаються коріння полінома помилки, що визначають положення перекручених символів у кодовому слові. Реалізується за допомогою процедури Ченя, рівносильній повному перебору. У поліном помилок послідовно підставляються всі можливі значення, коли поліном звертається в нуль – коріння знайдені.

Визначення характеру помилки і її виправлення

По синдрому помилки й знайдених корінь полінома за допомогою алгоритму Форни визначається характер помилки й будується маска перекручених символів. Ця маска накладається на кодове слово за допомогою операції XOR і перекручені символи відновлюються. Після цього відкидаються перевірені символи й виходить відновлене інформаційне слово.

Застосування

У даній момент коди Ріда-Соломона мають дуже широку область застосування завдяки їхній здатності знаходити й виправляти багаторазові пакети помилок.

Запис і зберігання інформації

Код Ріда-Соломона використовується при записі й читанні в контролерах оперативної пам'яті, при архівуванні даних, запису інформації на жорсткі хмарні сервіси (ЕСС), запису у хмарних сервісах хмарні сервіси.

Навіть якщо ушкоджено значний обсяг інформації, зіпсовано кілька секторів дискового носія, то коди Ріда-Соломона дозволяють відновити більшу частину загубленої інформації. Також використовується при записі на такі носії, як магнітні стрічки й штрихкоди.

Запис у хмарних сервісах-ROM

Можливі помилки при читанні з диска з'являються вже на етапі виробництва диска, тому що зробити ідеальний хмарний сервіс при сучасних технологіях неможливо. Так само помилки можуть бути викликані подряпинами на поверхні диска, пилом і т.д.

Тому при виготовленні компакт-диску серверу хмарного сервісу, що читається, використовується система корекції CIRC (Cross Interleaved Reed Solomon Code). Ця корекція реалізована у всіх пристроях, що дозволяють зчитувати дані з CD дисків, у вигляді чипа із прошиванням firmware. Знаходження й корекція помилок заснована надмірності й переміщення (redundancy & interleaving). Надмірність приблизно 25% від вихідної інформації.

При записі на цифрові аудіокомпакт-хмарні сервіси (Compact Disc Digital Audio – CD-DA) використовується стандарт Red Book. Корекція помилок відбувається на двох рівнях C1 і C2. При кодуванні на першому етапі відбувається додавання перевіренних символів до вихідних даних, на другому етапі інформація знову кодується.

Крім кодування здійснюється також перемішування (переміщення) байтів, щоб при корекції блоки помилок розпалися на окремі біти, які легше виправляються. На першому рівні виявляються й виправляються помилкові блоки довжиною один і два байти (один і два помилкових символи відповідно).

Помилкові блоки довжиною три байти виявляються й передаються на наступний рівень. На другому рівні виявляються й виправляються помилкові блоки, що виникли в C2, довжиною 1 і 2 байти. Виявлення трьох помилкових символу є фатальною помилкою, не можуть бути виправлені.

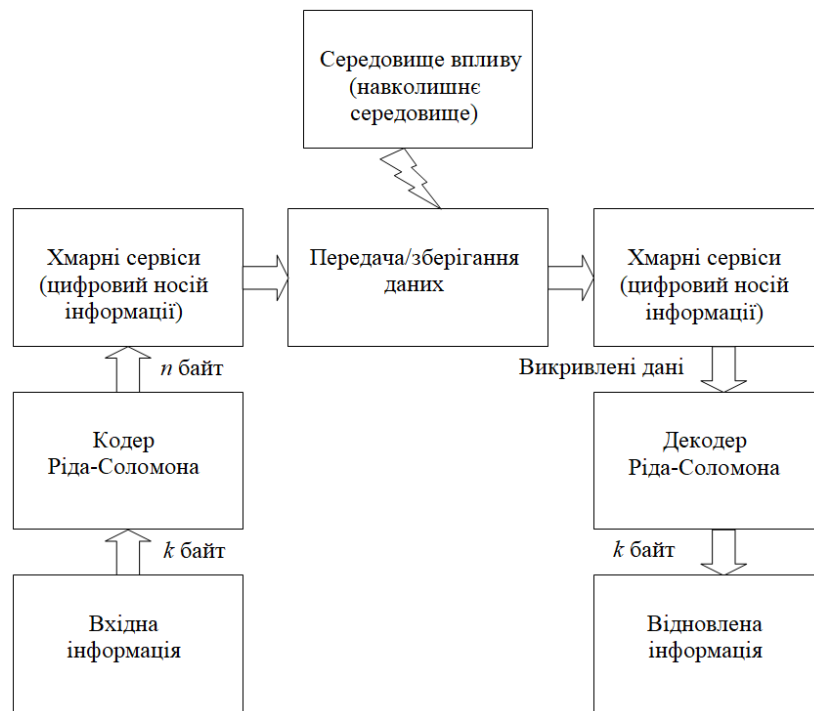


Рисунок 3 – Структурна схема системи

Бездротовий і мобільний зв'язок

Цей алгоритм кодування використовується при передачі даних по мережах WiMAX, в оптичних лініях зв'язку, у супутниковому й радіорелейному зв'язку. Метод прямої корекції помилок у минаючому трафіке (Forward Error Correction, FEC) ґрунтується на кодах Ріда-Соломона.

Розробка структурної схеми

Структурна схема системи забезпечення цілісності даних у хмарних сервісах зображена на рисунку 3.

З цієї схеми ми бачимо, що над вхідними даними, перед записом у хмарний сервіс, відбуваються перетворення кодеком Ріда-Соломона. Після кодування дані записуються на носій. У якості носія окрім хмарні сервіси може використовуватися любий інший носій інформації. Після цього інформація зберігається на носіїві. При зчитуванні інформації, розроблене програмне забезпечення декодує інформацію, яка зберігається на носіїві, і якщо потрібно, після проведення відповідних перевірок, проводить відновлення втраченої інформації. Якщо таке відновлення неможливе, то програма видає відповідне повідомлення.

Висновки. У статті наведені теоретичне узагальнення й рішення наукового завдання дослідження методів забезпечення цілісності даних у хмарних сервісах.

Рішення даного завдання полягало у вирішенні наступних задач:

- Був проведений огляд існуючих систем забезпечення цілісності даних у хмарних сервісах.
- Досліджена система забезпечення цілісності даних у хмарних сервісах.
- На основі отриманих результатів досліджень створена програмна реалізація системи забезпечення цілісності даних у хмарних сервісах.

Розроблені під час виконання випускної кваліфікаційної роботи за другим (магістерським) рівнем вищої освіти алгоритми дозволяють успішно вирішувати завдання забезпечення цілісності даних у хмарних сервісах.

Проведено аналіз предметної галузі в ході якого були виявлені об'єкти, взаємодія яких носить істотний характер для функціональної діяльності предметної галузі, і їхні основні характеристики; побудована алгоритм і вибраний середовище розробки.

Список літератури

1. Smirnov O.A., Alimseitova Zh., Adranova A., Akhmetov B., Lakhno V., Zhilkishbayeva G. «Models and algorithms for ensuring functional stability and cybersecurity of virtual cloud resources». *Journal of theoretical and applied information technology* Vol.98. No 21, 2020, P. 3334-3346. (Scopus).
2. Smirnov O., Kuznetsov A., Arischenko A., Chepurko I., Onikiychuk A., Kuznetsova T. «Pseudorandom sequences for spread spectrum image steganography». *CEUR Workshop Proceedings Volume 2654*, 2020, Pages 122-131. (Scopus).
3. Smirnov O., Kuznetsov A., Kovalchuk D., Kuznetsova T. «New technique for data hiding in cover images using adaptively generated pseudorandom sequences». *CEUR Workshop Proceedings Volume 2654*, 2020, Pages 1-14. (Scopus).
4. Smirnov O., Lutsenko M., Kuznetsov A., Kiian A., Kuznetsova T., «Biometric cryptosystems: overview, state-of-the-art and perspective directions». *Lecture Notes in Networks and Systems*, vol 152. Springer, Cham. 2021, pp 66-84. (Scopus).
5. Smirnov O., Kuznetsov A., Pushkar'ov A., Serhiienko R., Babenko V., Kuznetsova T., «Representation of Cascade Codes in the Frequency Domain». In: Radivilova T., Ageyev D., Kryvinska N. (eds) *Data-Centric Business and Applications. Lecture Notes on Data Engineering and Communications Technologies*, vol 48. Springer, Cham. 2021. pp 557-587. (Scopus).
6. Smirnov, O., Markovets, O. Vovk, N., Turchyn, Y., «Model of informational support for social network administrators' content creation». *CEUR Workshop Proceedings Volume 2616*, 2020, Pages 125-136. (Scopus).
7. Smirnov, O., Drieieva, H., Drieiev, O., Polishchuk, Y., Brzhanov, R., Aleksander, M. «Method of fractal traffic generation by a model of generator on the graph». *CEUR Workshop Proceedings Volume 2616*, 2020, Pages 366-379. (Scopus).
8. Smirnov, O., Shekhanin, K., Kuznetsov, A., Krasnobayev, V. «Detecting Hidden Information in FAT». *International Journal of Computer Network and Information Security (IJCNIS)*. Vol. 12, No. 3, 2020. PP.33-43. (Scopus).
9. Smirnov, O., Drieieva, H., Drieiev, O., Simakhin, V., Bondar, S., Odarchenko, R. «Managing multifractal properties of the binary sequence generated with the Markov chains», *CEUR Workshop Proceedings Volume 2608*, 2020, Pages 633-645. (Scopus).
10. Smirnov, O., Kuznetsov, A., Gorbacheva, L., Babenko, V., «Hiding data in images using a pseudo-random sequence», *CEUR Workshop Proceedings Volume 2608*, 2020, Pages 646-660., (Scopus).
11. Smirnov, O., Kuznetsov, A., Kolovanova, I., Kuznetsova, T., «Noise immunity of the algebraic geometric codes». *International Journal of Computing*; 2019, Volume 18, Issue 4 – Research Institute for Intelligent Computer Systems – 2019. – P. 393-407. (Scopus).
12. Smirnov, O., Ulichev, O., Meleshko, Y., Khokh, V., Goncharenko, I. «Method of Choosing Objects for Informational Influence in Social Networks during Information Campaign Based on the Analytic Hierarchy Process». *CEUR Workshop Proceedings*, Vol 2588, P. 215-227, 2019. (Scopus).
13. Smirnov, O., Krasnobayev, V., Yanko, A., Kuznetsova, T. «Methods of nulling numbers in the system of residual classes». *CEUR Workshop Proceedings*, Vol 2588, P. 90-106, 2019. (Scopus).
14. Smirnov, O., Kuznetsov, A., Kovalchuk, D., Pastukhov, M., Kuznetsova, K., Prokopovych-Tkachenko, D., «Discrete Signals with Special Correlation Properties», *CEUR Workshop Proceedings Volume 2353*, *CEUR Workshop Proceedings 2019*, Pages 618-629. (Scopus).
15. Smirnov, O., Kuznetsov, A., Kiian, A., Kuznetsova, K., Ivko, T., Prokopovych-Tkachenko, D., «Soft Decoding Based on Ordered Subsets of Verification Equations of Turbo-Productive Codes», *CEUR Workshop Proceedings Volume 2353*, *CEUR Workshop Proceedings 2019*, Pages 873-884. (Scopus).
16. Smirnov, O., Kuznetsov, A., Prokopovych-Tkachenko, D. «Hiding Data in Images Using a Pseudo-Random Sequence». *ISCI'2020: Information Security in Critical Infrastructures*. Collective monograph. Edited by Ivan D. Gorbenko, Victor A. Krasnobayev and Alexandr A. Kuznetsov. ASC Academic Publishing, USA, 2020. pp. 46-59. – ISBN: 978-1-7362833-0-1 (Hardback), ISBN: 978-1-7362833-1-8 (Ebook).
17. Smirnov, O., Kuznetsov, A., Shekhanin, K., Chepurko, I. *Detecting Hidden Information in FAT*. Монографія: In.: *ISCI'2019: Information Security in Critical Infrastructures*. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 412-429. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
18. Smirnov, O., Kuznetsov, A., Kuznetsova, K. *Synthesis of Discrete Signals with Improved Correlation Properties*. Монографія: In.: *ISCI'2019: Information Security in Critical Infrastructures*. Collective monograph. Edited by Ivan D. Gorbenko and Alexandr A. Kuznetsov, ASC Academic Publishing, USA, 2019, pp. 281-299. – ISBN: 978-0-9989826-8-7 (Hardback), ISBN: 978-0-9989826-9-4 (Ebook).
19. О.А. Смірнов, П.С. Усік, «дослідження перспектив використання технологічних рішень в мережах 5g» у *Кібербезпека та інформаційні технології: монографія*. – Х. : ТОВ «ДІСА ПЛЮС», 2020.С. 122-135.
20. Смірнов О.А., Дреєва Г.М., «Метод генерування фрактального трафіку за допомогою моделі генератора на графі» у *Інформаційна безпека та інформаційні технології: монографія / за заг. ред. В. С. Пономаренка*. – Х. : Вид. Рожко С.Г. 2019. С. 123-139.