виправляємих довільно розташованих в межах узагальненого кодового слова викривлень дорівнює (λ-1) символів.

Таким чином, запропоновані методи дозволяють забезпечити різні, в залежності від умов застосування, процедури контролю чи контролю та оперативного поновлення цілісності інформації в ТКПС при її несанкціонованих порушеннях з забезпеченням високої імітостійкості.

При організації процесів обміну інформацією слід врахувати наступне:

1. Методи контролю цілісності інформації, з їх вище визначеними особливостями, можуть бути легко пристосованими для організації обміну без використання механізмів зворотного вирішуючого зв'язку, що, як відомо [4], призводить до підвищення швидкості обміну (довжина блоку повинна відповідати довжині повідомлення, прийнятій у відповідному протоколі. Це просто реалізується, наприклад, в протоколі X.25).
2. Контроль цілісності інформації в блоках, довжина яких перевищує довжину повідомлення (наприклад, контроль у файлах, які передаються через ТКПС), дозволяє застосовувати принципи каскадних кодів, та забезпечувати цілісність в умовах впливу чи то завад великої тривалості (кількість викривлень перевищує корегуючі властивості внутрішнього коду), чи то тривалих (в тому ж розумінні) завмирань сигналів.
3. Оскільки використання ЛУ-коду дозволяє виявляти та виправляти викривлення, обумовлені будь-якими впливами, в тому числі і впливами на канальному рівні, то принципово можливим є використання модемів з найпростішими функціями, принаймні без використання функцій виявлення в узагальненому кодовому слові місця викривлення та його виправлення, що, на думку авторів, дозволить зменшити вартість ТКПС та підвищити швидкість обміну.

Слід, крім того, звернути увагу і на те, що обчислення контрольних ознак за запропонованими методами має усі ознаки, необхідні для деяких видів *цифрового підпису*, а отже можуть в певних випадках виконувати його функції.

Література: Игнатов В.А. Теория информации и передачи сигналов. – М.: Сов. радио, 1979. – 280 с. 2. Хетагуров Д.А., Руднев Ю.П. Повышение надежности цифровых устройств методами избыточного кодирования. – М.: "Энергия", 1974. – 290 с. 3. Мамиконов А.Г. и др. Достоверность и резервирование информации в АСУ. – М.: Энергоиздат, 1986. – 304 с. 4. Бойченко А.В. Про один алгоритм захисту інформаційного обміну в комп'ютерних мережах //Реєстрація, зберігання і обробка даних. - 1999. - № 1, том 1. с. 92 - 95. 5. Бунин С.Г., Василенко В.С. Сравнительная оценка СПД с решающей обратной связью и с использованием корректирующих кодов // УСиМ. – 1992. – № 9/10. – С.30 – 35. 6. Василенко В.С., Короленко М.П. Целостность информации в автоматизированных системах // Корпоративные системы. 1999.-№ 3.-с. 52-57.

УДК 681.31

МЕХАНІЗМИ КОНТРОЛЮ ЦІЛІСНОСТІ ІНФОРМАЦІЇ ТА ЇЇ ПОНОВЛЕННЯ

Василь Василенко, Михайло Короленко
ВАТ "КП ОТІ"

Анотація: **В статті запропоновано застосування для задач контролю та поновлення цілісності інформаційних наборів процедур завадостійкого корегуючого коду умовних лишків (ЛУ - КОДУ), які при високій імітостійкості хеш-функцій, які формуються для цього, забезпечують можливість більш оперативного поновлення інформації, цілісність якої є порушеною.**

Summari: **The article presents the application of the noise-immune correcting conditional residue code (CR-code) procedures, which in case of a high imitation resistance of the formed hash functions provide the possibility for the prompt recovery of the corrupted data integrity, to the tasks of data sets integrity control and recovery.**

Ключові слова: **цілісність інформації, контроль, поновлення, імітостійкість, завадостійкий корегуючий код.**
Key words: **data integrity, control, recovery, imitation resistance, noise-immune correcting code.**

Цілісність є однією з основних функціональних властивостей захищених систем [1-4], забезпечення якої дозволяє з певною гарантією говорити про можливість викрити чи запобігти спробам будь-якої несанкціонованої модифікації чи знищення програмних засобів, інформації та інших ресурсів комп'ютерних систем.

Як відомо, цілісність інформації забезпечується або механізмами розмежування доступу, або механізмами контролю цілісності інформації, або їх сукупністю.

В даній статті запропоновано ефективні механізми контролю цілісності а також контролю та поновлення цілісності інформації, які можуть бути застосованими чи окремо, чи в сукупності з механізмами розмежування доступу, що не відбивається на складності чи властивостях цих механізмів.

Під контролем цілісності інформації розуміється процес перевірки наявності викривлень цієї інформації, незалежно від причин їх походження (навмисні чи ненавмисні викривлення).

Під контролем та поновленням цілісності інформації розуміється процес перевірки наявності викривлень цієї інформації, незалежно від причин їх походження (навмисні чи ненавмисні викривлення), з наступною корекцією викривленої інформації.

1 Механізм контролю цілісності інформації

1.1 Побочний контроль цілісності інформації. Узагальнені та базові кодові слова

В основу розглянутих механізмів контролю цілісності інформації покладено механізм блочного контролю цілісності інформації, коли контрольована інформація розподіляється на блоки – базові та узагальнені кодові слова В якості базових кодових слів, з урахуванням технологічності реалізації відповідних алгоритмів, використовуються інформаційні набори певної довжини, наприклад у 32 байти. З декількох таких базових кодових слів шляхом їх перемежування глибиною λ створюється “узагальнене кодове слово”:

$$\mathbf{N} = n1 \cdot \lambda ,$$

де: N-число байтів в узагальненому кодовому слові (в блоці інформації);

n1 – число байтів в базовому кодовому слові (довжина базового кодового слова). Для наведеного прикладу n1 = 32.

Для контролю цілісності, а в подальшому і поновлення базових кодових слів потрібно мати додаткові (надлишкові) символи, які зберігають в собі в специфічному вигляді – у вигляді чи контрольної ознаки (в термінах завадостійкого кодування), чи хеш–функції (в термінах криптографічних перетворень) інформацію про це ж базове кодове слово за його станом на початок контролю.

Хеш–функція для контролю цілісності базових кодових слів механізму, що розглядається, розраховується [5] за правилами Коду умовних лишків (лишків умовних код, – ЛУ – код). При цьому здійснюється перетворення інформації, що контролюється – розраховується контрольна ознака, чи хеш–функція.

Хеш–функція базових кодових слів формується як s-байтове двійкове число, величина якого обчислюється як:

$$\mathbf{H}= \{p_k - \{[z \cdot p_k] \cdot R_k\}_{p_k}\}_{p_k} .$$

В цьому виразі:

змінна p_k дорівнює добутку s таких простих чисел – контрольних основ $p_{k1}, p_{k2}, ..., p_{ks}$ з діапазону восьмирозрядних чисел, щоб цей добуток перевищував добуток двох найбільших основ з їх набору p_i :

$$\prod_{I=1}^{I=s} p_{ki} \geq p_n \cdot p_{n-1} ;$$

змінна s визначає потрібну кількість контрольних основ і ,через це, - довжину хеш – функції;

величина z обчислюється як:

$$\mathbf{z}= \sum_{i=1}^{i=n2} \frac{\alpha_i \cdot m_i}{p_i} - \left[\sum_{i=1}^{i=n2} \frac{\alpha_i \cdot m_i}{p_i} \right],$$

де: змінна n2 при формуванні хеш–функції приймає значення n1(тобто n2 = n1);

позначка [X] означає обчислення цілої частини величини X;

змінна m_i - константа ЛУ – коду, – “вага” відповідного ортогонального базису R_i.

Після формування хеш–функції її значення записується чи після інформаційних наборів, чи в іншому місці файлу, який контролюється, та зберігається для наступного контролю цілісності базових кодових слів.

1.2 Формування хеш-функції на етапі запису інформації

Під хеш–функцією H, взагалі, розуміється відображення довільної інформації, що контролюється, (довільного набору даних) в образ фіксованої невеликої довжини за правилами того чи іншого перетворення.

Механізм забезпечення контролю цілісності, що розглядається, орієнтовано на файлову організацію розміщення інформації на її носії з файлами довільної довжини, яка умовно наведена на рисунку 1.

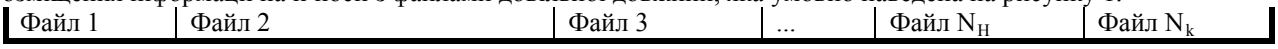

Рисунок 1 - Файлова структура розміщення інформації на носії

Передбачається, що інформація, яка контролюється, розміщена в файлах з номерами з 1 по N_H, а в файлі з номером N_k зберігається хеш–функція, яка була обчислена для інформації, що є записаною в файлах 1... N_H під час її запису на носій. При цьому результуюча хеш–функція носія H_H обчислюється як порозрядна сума по модулю 2 хеш–функцій інформаційних файлів:

$$H_н = \sum_{i=1}^{i=N_н} H_{фi},$$

де: $H_{ф\,i}$ - хеш–функція, що обчислена для і-го файлу (i = 1,..., N_H);

позначка Σ означає в цій формулі обчислення порозрядної суми по модулю 2.

Для обчислення хеш–функцій $H_{фi}$ інформаційних файлів кожен з цих файлів розподіляється на блоки інформації – узагальнені кодові слова (див. рисунок 2):

Рисунок 2 - Представлення інформації файла при формуванні його хеш–функції

Кожен з таких блоків включає до свого складу $32 \cdot \lambda$ байтів.

Результуюча хеш–функція і-го файла обчислюється, як порозрядна сума по модулю 2 хеш–функцій $H_{ГJ} = H_{УКСJ}$ усіх блоків інформації – узагальнених кодових слів і зберігається (записується) в кінці файла або після усіх файлів:

$$H_{Ф\,i} = \sum_{J=1}^{J=N_{ГБФ}} H_{ГJ},$$

де позначка Σ, як і раніше, означає в цій формулі обчислення порозрядної суми по модулю 2.

Чисельність блоків – узагальнених кодових слів в файлі приймає значення

$$N_{ГБФ}=N_{СФ}/(32\cdot\lambda),$$

якщо величина $N_{СФ}/(32\cdot\lambda)$ є цілою, та, що буде значно частіше,

$$N_{ГБФ}=[N_{СФ}/(32\cdot\lambda)]+1,$$

в іншому випадку. В цьому виразі:

$N_{СФ}$ – кількість символів в файлі, який контролюється;

32 – обрана довжина базових кодових слів (вона може бути і іншою);

позначка [X] означає, як і раніше, обчислення цілої частини величини X.

Хеш–функція кожного з узагальнених кодових слів інформації $H_{УКСJ}$ обчислюються, як конкатенація хеш–функцій його "базових кодових слів":

$$H_{УКСJ}= \overset{\lambda}{*}_{i=1} H_i,$$

де H_i - хеш–функція і – го базового кодового слова, а величина λ визначається нижче.

Для останнього з узагальнених кодових слів, у випадку, коли кількість базових кодових слів в ньому є величиною не цілою, меншою ніж λ, або коли довжина файлу є невеликою ($N_{СФ}< 32$), у якості хеш–функцій відсутніх базових кодових слів приймаються нульові хеш–функції, тобто хеш–функції, в яких кожен з s байтів, що входять до їх складу, є арифметичним нулем.

Довжина узагальнених кодових слів n1 та їх кількість $N_{ГБФ}$ залежать від довжини файлу, і потрібної довжини хеш–функції $H_{фi}$ файлу. Якщо довжина файлу перевищує 32 байти і довжина хеш–функції $H_{фi}$ визначена та дорівнює

$$R = |H_{фi}|,$$

наприклад, R=32 байтам, як це прийнято в міждержавному стандарті ГОСТ 34.311 – 95, то величини R, s, та λ взаємно визначаються з виразу

$$\lambda = R/s,$$

якщо величина R/s є цілим числом, або

$$\lambda = [R/s]+1,$$

в іншому випадку.

При цьому довжина хеш–функції, що обчислюється r, приймає, залежно від випадку, або значення

$$r = R, \text{ або } r = ([R/s]+1)\cdot s.$$

Не важко зрозуміти, що мінімальним значенням змінної s є величина, яка дорівнює трьом, оскільки величини p_n та p_{n-1} для забезпечення восьми бітових умовних лишків (символів інформаційного набору ,що контролюються) є дев'яти бітовими, тобто розрядність їх добутку перевищує 2 байти. Максимальним значенням змінної s є величина 29, , оскільки існує лише 29 взаємно простих восьми бітових чисел, причому з метою забезпечення добрих технологічних властивостей відповідних алгоритмів, значення s слід обирати з ряду 4, 8, 16. Тоді величина λ (при R=32 байтам) приймає одне із значень 8, 4, 2 відповідно.

В разі, коли довжина файлу не перевищує 32 байти, величини s, та λ слід обирати так, щоб виконувалась умова R=32 байтам. Неважко переконатись, що це можливо забезпечити до довжини файлу $N_{СФ}$=2, коли слід приймати n1 = 1, s = 16, λ = 2.

Для випадку коротких файлів, коли не можливо забезпечити в базових словах кількість інформаційних символів n1 рівною 32 байтам, значення усіх відсутніх байтів слід приймати рівними арифметичному нулю.

Після формування хеш–функції її значення записується після інформаційних наборів чи в іншому місці файлу, який контролюється, та зберігається для наступного контролю цілісності базових кодових слів.

2 Контроль цілісності інформації

Контроль цілісності інформації носія, яка є розміщеною на цьому носії чи на його локальній частині забезпечується шляхом порівняння хеш–функції відповідного носія, яка обчислюється під час контролю цілісності інформації, що зберігається з хеш–функцією, яка була обчисленою для цього ж носія під час запису інформації.

Якщо ці хеш–функції співпадають, то приймається рішення про те, що цілісність даного носія не є порушеною, і здійснюється контроль цілісності наступного носія.

В разі, коли ці хеш–функції не співпадають, приймається рішення про те, що цілісність даного носія є порушеною, і здійснюється відповідне інформування користувача. Після цього здійснюється контроль цілісності наступного носія.

Для організації контролю цілісності інформації з використанням відповідного програмного засобу контролю у складі програмних засобів доцільно організувати каталог файлів, для яких сформовані хеш–функції (ХФ) – ХФ - файли. Орієнтовний зміст каталогу (одного рядка для одного з файлів):

- Номер по порядку;
- Ім'я файлу;
- Дата формування ХФ;
- Ключ хешування;
- Значення ХФ;

Такий каталог необхідно зберігати в наборах, що захищені від доступу будь – яких користувачів, окрім власника інформації. Ключова інформація, а також ХФ – файли є об'єктами захисту Підсистеми захисту інформації.

Формування ХФ здійснюється власником інформації на його ключах в відповідному режимі для критичної інформації, нових програм, таблиць, наборів даних та т.п., що потребують контролю цілісності, а також при кожній зміні цієї інформації (при виконанні будь – яким авторизованим користувачем з повноваженнями - **дозвіл модифікації** (UPDATE) чи **можливість будь - якої роботи з набором даних** (ALTER) будь – яких операцій з цією інформацією).

Контроль цілісності цієї інформації доцільно здійснювати ***автоматично*** завжди після виконання будь – яким користувачем з повноваженнями - **дозвіл модифікації** (UPDATE) чи **можливість будь - якої роботи з набором даних** (ALTER) будь – яких операцій з інформацією, що потребує контролю цілісності (для цього ці об'єкти повинні мати необхідні ознаки, за якими здійснюється автоматичний запуск процедур контролю цілісності).

У першому випадку (після виконання будь – яким користувачем з повноваженнями - **дозвіл модифікації** (UPDATE) чи **можливість будь - якої роботи з набором даних** (ALTER) будь – яких операцій з інформацією, що потребує контролю цілісності) після контролю, в разі виявлення порушення цілісності цієї інформації без відому власника, здійснюється котроль правомірності змін в інформації та формування ХФ, чи, в необхідних випадках, поновлення інформації з усіма діями, які слід виконувати при несанкціонованому порушенні цілісності інформації.

3 Механізми контролю та поновлення цілісності інформації

Під контролем та поновленням цілісності інформації розуміється процес перевірки наявності викривлень цієї інформації, незалежно від причин їх походження (навмисні чи ненавмисні викривлення), з наступною корекцією викривленої інформації. Тобто, механізм контролю та поновлення цілісності інформації забезпечує, як контроль цілісності інформації, так і усунення викривлень цієї інформації.

Для контролю цілісності інформації можливим є застосування таких механізмів як:

- блочно – групового контролю інформації;
- контролю інформації по файлам;
- контролю інформації всього носія (картриджу, ГМД, логічного диску, усього ЖМД та т.п.);
- контролю інформації з оцінкою можливостей її поновлення, тобто з перевіркою, чи не перевищують викривлення інформації можливостей механізму по їх корекції.

4 Блочно – груповий контроль та поновлення інформації

Блочно – груповий контроль цілісності інформації носія, яка є розміщеною на цьому носії чи на його локальній частині, при реалізації цих механізмів, забезпечується або, як і при реалізації вище розглянутих механізмів контролю цілісності інформації, шляхом порівняння хеш–функції відповідного носія чи його частини, яка обчислюється під час контролю та поновлення цілісності інформації, що зберігається, з хеш–функцією, яка була обчисленою для цього ж носія чи його частини під час запису інформації, або з використанням властивостей ЛУ - коду.

Але, оскільки в даних механізмах потрібно реалізовувати додаткову, порівняно з розглянутими механізмами контролю цілісності інформації, задачу поновлення інформації, то природнім є те, що при цьому потрібно мати додаткову, необхідну для корекції викривлень інформацію. Ця додаткова інформація зосереджується у відповідних хеш–функціях, порядок і алгоритми обчислення яких повинні забезпечити як отримання, так і зберігання потрібної інформації, і тому відрізняються від викладених в розділі 1 порядку і алгоритмів обчислення хеш–функцій при реалізації механізмів контролю цілісності інформації. Ця відміна стосується, в основному, задачі збільшення потрібної при цьому надлишковості.

Механізм, що розглядається, забезпечення контролю та поновлення цілісності інформації, як і раніше, орієнтовано на файлову організацію розміщення інформації на її носії з файлами довільної довжини, яка умовно наведена на рисунку1.

Контрольована інформація, як і раніше, розміщується в файлах з номерами з 1 по $N_н$, а в файлі з номером N_k зберігається хеш–функція, яка була обчислена для інформації, що є записаною в файлах 1... $N_н$ під час її запису на носій. Але, на відміну від розглянутого раніше механізму контролю цілісності інформації, результуюча хеш–функція носія обчислюється як конкатенація хеш–функцій його файлів:

$$H_Н = \overset{i=N_K}{\underset{i=1}{*}} H_{фi},$$

де: позначка * означає обчислення конкатенації хеш–функцій усіх файлів носія;

$H_{фi}$- хеш–функція, що обчислена для i-го файлу (i = 1,..., $N_н$).

Для обчислення хеш–функцій інформаційних файлів, на відміну від механізму розділу 1.1.2, кожен з цих файлів розподіляється на групи блоків інформації. Кожна з таких груп включає до свого складу декілька (до 8) блоків, довжина яких, із умови забезпечення доброї технологічності програмних засобів, складає по 64 байти кожен. Такий розподіл на групи блоків дозволяє забезпечити поновлення цілісності великої чисельності, але малорозрядних викривлень. Максимальна кількість таких корегованих викривлень дорівнює чисельності груп блоків інформації в файлі, а їх довжина в байтах (в цьому випадку – мінімальна) не перевищує величини λ = 2, де λ, як і раніше, - глибина перемежування. Чисельність груп блоків інформації в файлі $N_{ГБФ}$, а також чисельність блоків в групі блоків $N_{БГ}$ залежить від довжини файлу (див. рисунки 3,4):

Група блоків 1	Група блоків 2	Група блоків 3	...	Група блоків $N_{ГБФ}$	Хеш–функція $H_{фi}$

Рисунок 3 - Представлення інформації файла при формуванні його хеш–функції

Кількість таких груп блоків $N_{ГБФ}$ обчислюється з виразу (тут і далі в цьому розділі кількість байтів в групі блоків взято рівно 512. Для інших з можливих довжин замість 512 слід підставити потрібне значення цієї довжини):

$$N_{ГБФ} = [N_{СФ}/512]+1.$$

Результуюча хеш–функція і-го файла обчислюються, як конкатенація хеш–функцій усіх груп блоків

інформації і зберігається (записується) в кінці файла або після усіх файлів:

$$H_{фi} = \overset{i=N_{ГБФ}}{\underset{i=1}{*}} H_{ГJ}$$

де позначка * означає обчислення конкатенації хеш–функцій відповідних груп блоків.
Кожна з груп блоків інформації складається з блоків інформації – узагальнених кодових слів (див. малюнок 4):

Блок 1	Блок 2	Блок 3	...	Блок 8	Хеш–функція $H_{ГJ}$

Рисунок 4 - Представлення інформації групи блоків при формуванні її хеш–функції

Кількість блоків – узагальнених кодових слів в групі визначається $H_{ГJ} \leq 8$.

Хеш–функції кожної з груп блоків інформації $H_{ГJ}$ обчислюються також , як конкатенація хеш–функцій її блоків інформації - узагальнених кодових слів:

$$H_{ГJ} = \overset{i=N_{ГJ}}{\underset{i=1}{*}} H_{УКСJ},$$

де $H_{УКСJ}$ - хеш–функція j – го узагальненого кодового слова.

Як і раніше для останніх, чи то груп блоків інформації, чи блоків інформації, у випадку, коли кількість узагальнених кодових слів $H_{ГJ}$ в цій групі є меншою ніж 8, або коли довжина файлу є невеликою (довжина файлу не перевищує 32 байти), у якості хеш–функцій відсутніх блоків чи байтів інформації приймаються або нульові хеш–функції, тобто хеш–функції, в яких кожен із s байтів, що входять до їх складу, є арифметичним нулем, або значення відсутніх байтів приймаються рівними арифметичному нулю.

Хеш–функція кожного з узагальнених кодових слів інформації обчислюються , як і раніше, як конкатенація хеш–функцій його "базових кодових слів" довжиною $N_{СБКС}$- символів кожне:

$$H_{УКСJ} = \overset{\lambda}{\underset{i=1}{*}} H_i,$$

де H_i - хеш–функція і – го базового кодового слова, порядок обчислення якої, окрім вибору величин s та λ, розглянуто раніше в розділі 1.1.2. При виборі величин s, в цьому випадку, слід виходити з необхідності забезпечення мінімально можливої надлишковості, яка, безумовно, забезпечується при мінімально можливої надлишковості в кожному з базових кодових слів, яка, як відомо, дорівнює 3 байтам. Глибину перемежування λ, в свою чергу, слід вибирати рівною 2.

Представлення інформації узагальнених кодових слів умовно, без врахування перемежування, представлено на рисунку 5,на якому БКС – базове кодове слово.

БКС 1	БКС 2	БКС 3	...	БКС λ	Хеш-функція $H_{УКСJ}$

Рисунок 5 - Представлення інформації узагальненого кодового слова при формуванні його хеш–функції

Таким чином, на етапі запису інформації в окремій частині носія буде сформована область, де є записаною сукупність - конкатенація хеш–функцій кожного з базових кодових слів. Це дозволяє організувати подальші контроль та, при необхідності, поновлення інформації.

5 Блочно – груповий контроль цілісності інформації

При організації блочно – групового контролю цілісності шляхом порівняння хеш–функцій відповідного носія чи його частини, які були обчисленими під час запису інформації, з хеш–функціями, які були обчисленими для цього ж носія чи його частини під час контролю інформації, необхідно, в повній відповідності до розглянутого в розділі 1.2.1.1 алгоритму, знов здійснювати обчислення хеш–функцій кожного з базових кодових слів.

Після обчислення хеш–функції чергового базового кодового слова, на відміну від згаданого алгоритму, цю хеш–функцію не записують на носій, а порівнюють з хеш–функцією, яка була обчисленою під час запису інформації.

Якщо ці хеш–функції співпадають, то приймається рішення про те, що цілісність даного базового кодового слова не є порушеною, і здійснюється контроль цілісності наступного базового кодового слова доти, поки не закінчиться контроль усього носія.

В разі, коли ці хеш–функції не співпадають, приймається рішення про те, що цілісність даного базового

кодового слова є порушеною, і здійснюється його поновлення шляхом використання інформації з резервних копій або, при умові здійснення додаткових обчислень величини z, - у відповідності з нижче викладеним Z - алгоритмом поновлення цілісності. Після цього здійснюється контроль цілісності наступного базового кодового слова, доти поки не скінчиться контроль усього файлу, а потім і носія.

При організації блочно – групового контролю цілісності шляхом використання властивостей ЛУ – коду, у відповідності з алгоритмом обчислення хеш–функцій базових кодових слів, здійснюється обчислення величини z, як:

$$z = \sum_{i=1}^{i=n2} \frac{\alpha_i \cdot m_i}{p_i} - \left[\sum_{i=1}^{i=n2} \frac{\alpha_i \cdot m_i}{p_i} \right],$$

де, на відміну від алгоритму обчислення хеш–функцій базових кодових слів змінна n2 приймає значення

$$n2 = n1+x,$$

n1, як і раніше – число байтів в базовому кодовому слові (довжина базового кодового слова);

x – кількість байтів в хеш–функції (від 3 до 29) даного базового кодового слова.

Таким чином, при обчисленні величин z для кожного з базових кодових слів на етапі контролю цілісності використовуються не лише символи контрольованої інформаційної частки файлу (носія) - базового кодового слова, а й символи хеш–функції, або, в термінах завадостійкого корегуючого ЛУ – коду, - символи контрольної ознаки цього базового кодового слова.

Отримане при цьому значення величини z порівнюється з константою ЛУ – коду:

$$Z<1/p_k,$$

де змінна p_k, як і раніше – контрольна основа ЛУ – коду.

Якщо ця нерівність задовольняється, то , то приймається рішення про те, що цілісність даного базового кодового слова не є порушеною, і здійснюється контроль цілісності наступного базового кодового слова, доти поки не скінчиться контроль усього носія.

В разі, коли ця нерівність не задовольняється, приймається рішення про те, що цілісність даного базового кодового слова є порушеною, і здійснюється його поновлення у відповідності з нижче викладеним Z - алгоритмом поновлення цілісності. Після цього здійснюється контроль цілісності наступного базового кодового слова, доти поки не закінчиться контроль усього носія.

6 Поновлення цілісності інформації базових кодових слів

Поновлення цілісності інформації при організації блочно – групового контролю цілісності шляхом порівняння хеш–функцій відповідного носія чи його частини, які були обчисленими під час запису інформації, з хеш–функціями, які були обчисленими для цього ж носія чи його частини під час контролю інформації, є можливим за рахунок використання відповідних резервних копій або додаткових обчислень величин z і це є особливістю цього поновлення та його, залежно від обставин, достоїнством чи певним недоліком.

Поновлення цілісності інформації при організації блочно – групового контролю цілісності з використанням властивостей ЛУ – коду не потребує використання резервних копій, а є суто розрахунковим з повним використанням інформації, яка зосереджена в надлишкових символах - хеш–функціях чи, в термінах ЛУ – коду, – в контрольних ознаках кожного з базових кодових слів.

У відповідності з Z – алгоритмом ЛУ – коду (назва алгоритму пов’язана з обчисленням змінної z) корекція викривленої змінної $\widetilde{\alpha}_i$, де позначка $\widetilde{X}$ означає наявність якогось викривлення в змінній X, відбувається згідно з виразом:

$$\alpha_i = \{ \widetilde{\alpha}_i - \{ [z \cdot p_i] \cdot R_i \}_{p_i} \}_{p_i},$$

в якому зміст усіх змінних та позначок співпадає з раніше визначеними.

В останньому виразі не визначеним є лише значення і – номеру викривленого символу $\widetilde{\alpha}_i$. Це значення знаходиться з системи нерівностей:

$$Z \cdot p_i - [Z \cdot p_i] < p_i/p_k, \ (i = 1, 2, ..., n1).$$

За розшукуємо значення і приймається номер того p_i, для якого задовольняється одна з n1 нерівностей цієї системи.

Цей механізм контролю та поновлення інформації дозволяє виявити порушення цілісності інформації в межах кожного з узагальнених кодових слів та виправити виявлені в ньому викривлення, довжина яких B_B може бути (при їх довільному розташуванні в межах узагальненого кодового слова) від одного до

$$B_B = [(\lambda-1) \cdot b_c+1]$$

двійкових символів (біт), де b_c - довжина символів в контрольованій інформації в бітах (для розглядаючих механізмів b_c =8). Тобто найбільш можлива довжина виправляємих довільно розташованих в межах узагальненого кодового слова викривлень дорівнює (λ-1) символів. Загальна кількість виправляємих викривлень такої довжини дорівнює кількості узагальнених кодових слів у складі файла.

7 Контроль та поновлення інформації по файлам

Для організації контролю та поновлення інформації по файлам інформація файла розміщується у відповідності до схеми, умовно представленої на рисунку 6.

Рисунок 6 - Представлення інформації файла при формуванні його хеш–функції

Як видно з рисунку 6, уся інформація файла розбивається на $N_{БКС}$ базових кодових слів, символи яких розміщаються з перемежуванням (на мал. 6 воно не показане) глибиною

$$\lambda_Ф = N_{БКС} = [N_{СФ}/N_{СБКС}]+1,$$

де: $N_{СФ}$ - кількість символів в складі контрольованого файла;

$N_{СБКС}$ - кількість символів в складі базового кодового слова.

В свою чергу, кількість символів в складі базового кодового слова $N_{СБКС}$ визначається, як і в попередніх механізмах.

Хеш-функція контрольованого файла формується, як конкатенація хеш–функцій усіх $\lambda_Ф = N_{БКС}$ базових кодових слів.

Нарешті, хеш–функції базових кодових слів при цьому обчислюється за таким же алгоритмом, як і в попередньому механізмі.

Цей механізм контролю та поновлення інформації дозволяє виявити порушення цілісності інформації в межах файлу та виправити виявлені в ньому викривлення, довжина яких $B_{ВФ}$ може бути (при їх довільному розташуванні в межах файлу) від одного до

$B_{ВФ} = (\lambda_Ф-1)\cdot b_c+1)$

біт, де b_c -, як і раніше, довжина символів в контрольованій інформації в бітах. Тобто найбільш можлива довжина виправляємих довільно розташованих в межах файлу викривлень дорівнює ($\lambda_Ф$ -1) символів.

8 Контроль та поновлення інформації всього носія

Для організації контролю та поновлення інформації всього носія інформація цього носія не залежно від кількості файлів на ньому розміщується у відповідності до схеми, умовно, без врахування перемежування, представленої на рисунку 7.

Рисунок 7 - Представлення інформації носія при формуванні його хеш–функції

Як видно з рисунка 7, уся інформація файла розбивається на $M_{БКС}$ базових кодових слів, символи яких розміщаються з перемежуванням (на мал. 5 воно не показане) глибиною

$\lambda_Н = M_{БКС} = [N_{СН}/N_{СБКС}]+1$,

де: $N_{СН}$ - кількість символів в складі контрольованого носія;

$N_{СБКС}$ - як і раніше, кількість символів в складі базового кодового слова.

В свою чергу, кількість символів в складі базового кодового слова $N_{СБКС}$ визначається, як і в попередніх механізмах.

Хеш-функція контрольованого носія формується, як конкатенація хеш–функцій його базових кодових слів.

Нарешті, хеш–функції базових кодових слів при цьому обчислюється за таким же алгоритмом, як і в попередніх механізмах.

Цей механізм контролю та поновлення інформації дозволяє виявити порушення цілісності інформації в межах носія та виправити виявлені в ньому викривлення, довжина яких $B_{ВН}$ може бути (при їх довільному розташуванні в межах носія) від одного до

$B_{ВН}=(\lambda_Н-1)\cdot b_c+1,$

біт, де b_c -, як і раніше, довжина символів в контрольованій інформації в бітах. Тобто найбільш можлива довжина виправляємих довільно розташованих в межах файлу викривлень дорівнює ($\lambda_Н$-1) символів.

9 Контроль та поновлення інформації з оцінкою можливостей її поновлення

Розглянуті механізми контролю та поновлення інформації є дуже ефективними (див. нижче) при умові, що кількість та довжина виявлених механізмами контролю викривлень відповідають можливостям механізмів поновлення по їх виправленню. Це є особливістю усіх існуючих завадостійких алгоритмів з корекцією викривлень.

Вочевидь, існує імовірність того, в деяких умовах, наприклад при злонавмисному порушенні цілісності інформації, така відповідність може бути зруйнованою. В таких умовах усі відомі завадостійкі алгоритми з корекцією викривлень будуть "виправляти" виявлені викривлення, вносячи при цьому додаткові викривлення, або, в кращому випадку, - не виявляти їх наявності і з такою можливістю доводиться поки що миритися.

Тому існує необхідність розробки таких механізмів контролю, які дозволяли б здійснювати оцінку можливостей механізмів поновлення по виправленню виявлених викривлень, що є напрямком подальших розробок.

10 Ефективність розглянутих механізмів

Звернемо увагу на те, що ***основним достоїнством розглянутих механізмів є*** не виключна простота процедур, які реалізуються, і, що само по собі є дуже цінним, а наявні ***криптографічні властивості ЛУ-коду***, можливість приховати здавалось би відкритий механізм розрахунку хеш-функцій від неавторизованих користувачів та можливих злонавмисників, що мають метою будь-яким чином порушити цілісність захищаємої інформації. Дійсно, досить тримати в таємниці набір основ для обчислення змінних та і самої хеш-функції, тобто вважати їх ключем перетворення і відповідним чином забезпечувати їх прихованість, як перед неавторизованим користувачем постає задача їх викриття, підбору, здобуття та т.п., тобто типова для криптоаналітика задача. Шляхом нескладних додаткових (до викладених механізмів) операцій, які зараз опускаються, ця задача переводиться в розряд задач з прямим перебором варіантів ключів, коли криптографічна стійкість визначається кількістю таких варіантів. В свою чергу ця кількість залежить, як відомо від довжини ключового набору (кількості основ, які використовуються), та від потужності поля взаємно простих чисел, з якої здійснюється їх вибір. До того ж доступна (відкрита) для аналізу інформація, яка міститься в хеш-функціях, не має однозначного зв'язку з ключами перетворення, оскільки при їх розрахунку використовуються операції за певними модулями, включаючи (для випадку контролю цілісності) і операції порозрядного додавання за модулем 2.

Останнє дає змогу визначити кількість варіантів ключів, яка представлена в таблиці 1 (див.[5]).

Таблиця 1

Довжина ключа (байти)	Механізми формування хеш–функцій для контролю цілісності інформації				
	ГОСТ 34.310 – 94	ГОСТ 28147-89	ЛУ - код		
			n1=28, s=4	n1=29, s=3	n1=32, s=4
32	-	10^{76}	$>3\cdot10^{76}$	$\approx10^{76}$	$\approx10^{76}$
64	$9{,}45\cdot10^{65}$	-	$>10^{135}$	$>10^{136}$	$>10^{136}$
128	$9{,}45\cdot10^{156}$	-	$\approx 10^{260}$	$\approx 10^{260}$	$\approx 10^{260}$

Примітка. Для другої та третьої колонок використані дані Інституту Інформаційних Технологій Харківського технічного університету радіоелектроніки (в другій колонці, виходячи з обсягу обчислень в $3\cdot10^5$ та $3\cdot10^{12}$ міпсороків відповідно). В шостій колонці наведено дані для найбільш доцільних довжини інформаційної частки базового кодового слова в 32 байти та довжині надлишкової частки базового кодового слова в 4 байти.

Як витікає з таблиці, розглянуті механізми суттєво переважають відомі, дозволені для застосування механізми з використанням міждержавних стандартів (ГОСТ 34.310 та ГОСТ 28147).

З урахуванням уже підкресленої простоти реалізації це дає змогу стверджувати про їх високу ефективність.

Література: 1. «Загальні положення щодо захисту інформації в комп'ютерних системах від несанкціонованого доступу» (НД ТЗІ 1.1 – 002 – 99); 2. «Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу» (НД ТЗІ 1.1 – 003 – 99); 3. «Критерії оцінки захищеності інформації в комп'ютерних системах від несанкціонованого доступу» (НД ТЗІ 2.5 – 004 – 99); 4. «Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу» (НД ТЗІ 2.5 – 005 – 99). 5. Василенко В.С., Короленко М.П. Целостность информации в автоматизированных системах // Корпоративные системы.-1999.-№ 3.-С.52-57.